Run-to-Run Control in SEMICONDUCTOR MANUFACTURING
Preface

The goal of this book is to provide a practical guide to the understanding, implementation, and use of run-to-run (R2R) control in semiconductor manufacturing as well as manufacturing in general. The target audience is intentionally wide and includes technology directors and strategists, technical managers, control engineers, process engineers, systems designers, integrators, and users. The aim of the authors is to provide insight into the development, integration, application, enhancement, and operation of R2R control. In addition, the book points to new directions in R2R process control, some of which have only recently been discussed in the literature. These directions point to avenues of opportunity for developing even more effective R2R control strategies for the fabricator of the future.

WHO SHOULD USE THIS BOOK

The benefits of R2R control implementation are wide-ranging and affect the many levels of the manufacturing hierarchy. As such, this book is structured to provide benefit to readers at each of these levels. For example, the following is a sample of who might utilize this book as a guide and aid in implementing an effective R2R control initiative either on a single tool, or facility-wide in a fabrication facility:

- A corporate-level technical strategist would utilize the book as a resource to:
  1. Collect convincing evidence indicating that R2R control will provide significant competitive advantage.
  2. See that proven R2R control solutions are available.
  3. Read that benefits, such as $C_{pk}$ and yield, have been proven and quantified.
  4. Plan a strategy for integration.
- A facility director would utilize the book for directing facility-wide R2R control development and deployment. Specifically, the facility director would utilize the book to define plans for:
  1. Identifying target applications for R2R control.
  2. Performing the necessary requirements analysis and identifying equipment, metrology, control, and integration deficiencies.
  3. Identifying the “control problem” for each candidate process, including process quality metrics.
  4. Determining the controllability of each candidate process.
  5. Developing stand-alone control solutions for each candidate process.
  6. Integrating these control solutions for a fab-wide R2R control solution.
• The process engineer for each process would utilize the book to develop an effective control solution for his/her process. The book would aid the process engineer in:
  1. Process input and output parameter selection and refinement.
  2. Process identification for control.
  3. Development of an industrial-quality solution that addresses requirements of parameter bounds, discretization, parameter weighting, process and metrology noise rejection, etc.
  4. Development, integration, and testing of the control software solution.

HOW TO READ THIS BOOK

We have put this book together with the intent that it be of use to the beginning reader in R2R control as well as the specialist seeking detailed information on R2R control methods and/or recent directions and developments. In order to achieve this we have divided the text into six parts plus a conclusion. The Introduction and first chapter of each part* should be read first by the reader new to the subject. These first chapters have been chosen because they are, on the whole, more introductory and less burdened with technicalities than later chapters in the same part. Specialist readers may, of course, pick and choose as they wish.

* Excerpt for Part 6: Advanced Topics.
Editors

James Moyne, Ph.D., is an Associate Research Scientist in the Electrical Engineering and Computer Science Department at the University of Michigan, and is President and co-founder of MiTeX Solutions, Inc., Canton, Michigan. (MiTeX Solutions was acquired by Brooks Automation, Inc. in June 2000.) James received his B.S.E.E. and B.S.E. in math, and his M.S.E.E. and Ph.D. in electrical engineering from the University of Michigan. He has over 30 refereed publications in the areas of discrete control, advanced process control, databases, sensor bus technology, and communications, and is the author of the patent on the Generic Cell Controller run-to-run control enabling technology. He is also the author of a number of SEMI (semiconductor manufacturing) international standards in the areas of sensor bus systems and communications, and has received four SEMI outstanding achievement awards and a technology transfer award.

James lives in Canton, Michigan, where his hobbies include writing music and playing the keyboard and sax. He is a published poet, has released a solo album of New Age music, and is a member of Cornerstone, which is currently working on its second Rock/Pop album.

Enrique del Castillo, Ph.D., is an Associate Professor in the Harold and Inge Marcus Department of Industrial and Manufacturing Engineering at the Pennsylvania State University. He holds a Ph.D. in industrial engineering from Arizona State University, and a Master of Engineering in operations research and industrial engineering from Cornell University. Dr. Castillo’s research interests include quality engineering and applied statistics, with particular emphasis on response surface methodology and time series control. He has over 35 papers in journals such as IIE Transactions, Journal of Quality Technology, Metrika, Communications in Statistics, International Journal of Production Research, European Journal of Operational Research, and Journal of the Operational Research Society. He has been awarded an NSF CAREER grant for research in semiconductor manufacturing process control. Dr. Castillo is an Associate Editor of the IIE Transactions on Quality and Reliability Engineering journal and a member of the editorial board of the Journal of Quality Technology.

Arnon Max Hurwitz, Ph.D., is Managing Director of Qualtech Productivity Solutions, South Africa, and Vice President of MiTeX Solutions, Inc., Canton, Michigan. He gained his M.S. in applied statistics from Oxford University, England, and his Ph.D. in mathematical statistics from the University of Cape Town, South Africa. Dr. Hurwitz lectured at the Graduate School of Business, Cape Town, and at Guilford College, North Carolina, and was Head of the Mathematics Department at Oak Ridge Military Academy, North Carolina. He was Quality Engineer at Corning
Glass, Inc., Telecommunications Division, in Wilmington, North Carolina, and Corporate Statistician for HIMONT USA, Inc., in Houston, Texas. He was a Senior Statistician and a Senior Project Manager at the U.S. semiconductor industry consortium SEMATECH in Austin, Texas. In 1997 he became Vice President of MiTeX Solutions, Inc., and also founded Qualtech Productivity Solutions, a corporation supplying statistical analytic services to finance and industry.

Arnon lives near Cape Town, South Africa, and works internationally. He has published a number of statistical and control-theoretic works in leading U.S. journals, and has contributed chapters to several books. He has a wife and three children. His hobbies are reading, writing, fly-fishing, and trying to play the fiddle.
Acknowledgment

The authors would like to thank those who contributed additional chapters for this book, and those who co-authored chapters, for their excellent and invaluable contributions. Names and contact information for all contributors follow, and are listed in alphabetical order.

In addition, we would like to thank Nora Konopka of CRC Press for her unflagging support and enthusiasm for the project, as well as all those at CRC Press who translated a complex manuscript into publishable form.

Of course, technical work of this nature has required the support, over quite a number of years, of many different entities and personalities, not the least of which were the research institutions that gave the technology birth — namely Massachusetts Institute of Technology and the University of Michigan — and the institutes that funded further research. In this case it was the semiconductor research corporation SEMATECH International, its member companies, and the tool vendors — all members of SEMI/SEMATECH — who made their machines and their expert staffs available for our numerous site experiments. There are too many names involved to mention, and we thank them one and all.

Lastly, we thank the reviewers of our book for their valuable comments and suggestions which added significantly to the value of the text.

James Moyne, Enrique Del Castillo, and Arnon Max Hurwitz
Contributors

Duane S. Boning
Microsystems Technology Laboratories
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
e-mail: boning@mtl.mit.edu

Jonathan Chapple-Sokol
IBM Microelectronics Division
1000 River Road B975/E
Essex Junction, VT 05452 USA
e-mail: chapple@us.ibm.com

Nauman A. Chaudhry
One Oracle Drive
Nashua, NH 03062 USA
email: nchaudhr@us.oracle.com

Argon Chen
Graduate Institute of Industrial
Engineering
National Taiwan University
1, Roosevelt Road
Sec.1, Taipei, Taiwan 106
e-mail: achen@ccms.ntu.edu.tw

Jin-Jung Chen
Department of Mechanical Engineering
National Taiwan University
Taipei, Taiwan

John Colt
IBM Microelectronics Division
1000 River Road B975/E
Essex Junction, VT 05452 USA
e-mail: a442991@us.ibm.com

Enrique Del Castillo
Department of Industrial &
Manufacturing Engineering
Pennsylvania State University
207 Hammond Building
University Park, PA 16802 USA
e-mail: exd13@psu.edu

Chadi El Chemali
Electrical Engineering & Computer
Science
University of Michigan
2360 Bonisteel Avenue
Ann Arbor, MI 48109 USA
e-mail: ccel@umich.edu

Ruey-Shan Guo
Department of Industrial Management &
Business Administration
National Taiwan University,
50, Lane 144, Sec. 4
Keelung Road
Taipei, Taiwan
e-mail: rsguo@ccms.ntu.edu.tw

Arnon Max Hurwitz
Qualtech Productivity Solutions
Sanclare Building,
Dreyer Street
Claremont 7735 South Africa
e-mail: qualtech@iafrica.com

Kareemullah Khan
RA1-303
Intel Corportaion
5200 N.E. Elam Young Parkway
Hillsboro, OR 97124 USA
e-mail: kareemullah.khan@intel.com

© 2001 by CRC Press LLC
Foreword

Mark Melliar-Smith, President and CEO, and Randy Goodall, Associate Director of Productivity and Infrastructure
International SEMATECH, Austin, Texas

Control is essential for all manufacturing, but every day in the semiconductor industry more than a quadrillion transistors, each with dimensions ranging from a fraction of a micrometer down to tens of atoms, must be profitably fabricated. To manage the exponentially rising cost of meeting this manufacturing challenge, process equipment from each technology generation is increasingly pressed into service to support the next generation. In the early years of the twenty-first century, new processes for nearly every aspect of transistor fabrication, from thin gate and source/drain to interconnecting metal and dielectrics, must be introduced in semiconductor factories around the world. The International Technology Roadmap for Semiconductors shows starkly that the timing for these technology changes is so short that we will necessarily “test them in combat.” Active control mechanisms, such as the run-to-run methods described in this book, are mandatory if the industry is to keep pace with the world’s demand for new electronic products.

During this period of unprecedented technological advancement, semiconductor companies are also initiating the new generation of larger, 300 mm silicon wafers. The product value of even a single 300 mm wafer containing more than 1000 advanced chips dictates the use of active process control with a new level of urgency. Although a staggering challenge, the 300 mm transition also brings with it a new opportunity. International SEMATECH’s member companies have collectively and comprehensively set a high bar for equipment performance in all areas, including factory communications and recipe management. Process control implementations should now become more straightforward.

Widespread deployment of run-to-run control has been somewhat of a struggle because of the required critical mass of software and communications capability necessary in both process equipment and factory systems. Equally scarce were the people to drive the development, engineering, and adoption of these techniques. With the new generation of software-savvy engineers at both semiconductor and supplier companies, this is changing. In addition, new sensors and other lower-cost measurement options are becoming available to reduce the time, money, and logistics needed to support cost-effective control implementations.

Interest in better control of process equipment arose at SEMATECH in the early 1990s. The run-to-run method was then, and continues to be, the least equipment-invasive control scheme that demonstrates real benefit. It is gratifying to see the ideas supported by SEMATECH reach a level of maturity and industry acceptance that supports treatment in a book of their own. This volume represents the continuance
of nearly a decade of active work and collaboration by these authors, in both the academic setting of their respective universities as well as the industry context of SEMATECH. Their long-time focus on the specific control problems of semiconductor manufacturing imbue this effort with relevance and practicality. We believe this book will find an industry ready for standard approaches and solutions for run-to-run control. Our hope is that it accelerates the emergence of the new era of sophistication in controlling the manufacturing marvel that is the semiconductor industry.
Table of Contents

Introduction
James Moyne and Arnon M. Hurwitz

PART 1: FOUNDATION FOR CONTROL

Chapter 1
Process Control in the Semiconductor Industry
Taber H. Smith, Duane S. Boning, and James Moyne

Chapter 2
Process Control and Optimization Methods for Run-to-Run Application
Enrique Del Castillo and Arnon M. Hurwitz

PART 2: R2R CONTROL ALGORITHMS

Chapter 3
Basic R2R Control Algorithms
William Moyne

Chapter 4
Learning and Optimization Algorithms for an Optimizing Adaptive Quality Controller
Enrique Del Castillo

Chapter 5
An Adaptive Run-to-Run Optimizing Controller for Linear and Nonlinear Processes
Arnon M. Hurwitz and Enrique Del Castillo

Chapter 6
A Comparative Analysis of Run-to-Run Control Algorithms in the Semiconductor Manufacturing Industry
Zhe Ning, James Moyne, Taber Smith, Duane Boning, Enrique Del Castillo, Jinn-Yi Yeh, and Arnon M. Hurwitz
PART 3: INTEGRATING CONTROL

Chapter 7
Existing and Envisioned Control Environment for Semiconductor Manufacturing
James Moyne and Joe White

Chapter 8
Design Requirements for an Integrative R2R Control Solution
James Moyne

Chapter 9
The Generic Cell Controller
James Moyne

Chapter 10
Derivation of a Piggyback Run-to-Run Control Solution Design
James Moyne

Chapter 11
Integrated Run-to-Run Control Solution Examples
James Moyne

Chapter 12
Design and Optimization of an Optimizing Adaptive Quality Controller,
Generic Cell Controller Enabled Solution
Enrique Del Castillo, Jinn-Yi Yeh, James Moyne, and Victor Solakhian

PART 4: CUSTOMIZATION METHODOLOGY

Chapter 13
Case Study: Furnace Capability Improvement Using a Customized Run-to-Run Control Solution
Arnon Hurwitz and James Moyne

Chapter 14
Process Recipe Optimization
Enrique Del Castillo
PART 5: CASE STUDIES

Chapter 15
Multizone Uniformity Control of a CMP Process Utilizing a Pre- and Postmeasurement Strategy
*James Moyne, Chadi El Chemali, Kareemullah Khan, Rock Nadeau, Paul Smith, John Colt, Jonathan Chapple-Sokol, and Tarun Parikh*

Chapter 16
Control of Photolithography Alignment
*Nital S. Patel and Robert Soper*

Chapter 17
Age-Based Double EWMA Controller and Its Application to a CMP Process
*Argon Chen and Ruey-Shan Guo*

PART 6: ADVANCED TOPICS

Chapter 18
Advancements in Chemical Mechanical Planarization Process Automation and Control
*James Moyne*

Chapter 19
An Enhanced Exponentially Weighted Moving Average Controller for Processes Subject to Random Disturbances
*Ruey-Shan Guo, Argon Chen, and Jin-Jung Chen*

Chapter 20
Enabling Generic Interprocess Multistep Control: the Active Controller
*Nauman Chaudhry, James Moyne, and Elke A. Rundensteiner*

PART 7: SUMMARY AND CONCLUSIONS

List of Acronyms
Dedication

This book is dedicated to our wives,
Jennifer, Monica, and Mary Frances,
who make it all worthwhile.
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Active Controller</td>
</tr>
<tr>
<td>AEC</td>
<td>advanced equipment control</td>
</tr>
<tr>
<td>APC</td>
<td>advanced process control</td>
</tr>
<tr>
<td>ARL</td>
<td>average run length</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>American Telephone and Telegraph</td>
</tr>
<tr>
<td>BCAM</td>
<td>Berkeley Computer-Aided Manufacturing</td>
</tr>
<tr>
<td>CDM</td>
<td>(SAN) common device model</td>
</tr>
<tr>
<td>CIM</td>
<td>computer-integrated manufacturing</td>
</tr>
<tr>
<td>CMP</td>
<td>chemical mechanical planarization or chemical mechanical polishing</td>
</tr>
<tr>
<td>COO</td>
<td>cost of ownership</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>Cpk</td>
<td>process capability</td>
</tr>
<tr>
<td>CSRS</td>
<td>Control Systems Requirements Specification</td>
</tr>
<tr>
<td>CSSWG</td>
<td>Control Systems Specification Working Group</td>
</tr>
<tr>
<td>CTE</td>
<td>center-to-edge (nonuniformity)</td>
</tr>
<tr>
<td>CVD</td>
<td>chemical vapor deposition</td>
</tr>
<tr>
<td>DBMS</td>
<td>database management system</td>
</tr>
<tr>
<td>d-EWMA</td>
<td>double exponentially-weighted moving average</td>
</tr>
<tr>
<td>DOE</td>
<td>design of experiments</td>
</tr>
<tr>
<td>DOF</td>
<td>depth of focus</td>
</tr>
<tr>
<td>ECA</td>
<td>event–condition–action (rules)</td>
</tr>
<tr>
<td>ECS-TF</td>
<td>Equipment Control Systems Task Force (of SEMI)</td>
</tr>
<tr>
<td>EPC</td>
<td>engineering process control</td>
</tr>
<tr>
<td>E-R</td>
<td>entity-relationship (database modeling)</td>
</tr>
<tr>
<td>EWMA</td>
<td>exponentially-weighted moving average</td>
</tr>
<tr>
<td>FDC</td>
<td>fault detection and classification</td>
</tr>
<tr>
<td>FWI</td>
<td>full wafer interferometry</td>
</tr>
<tr>
<td>GCC</td>
<td>Generic Cell Controller</td>
</tr>
<tr>
<td>GEM</td>
<td>Generic Equipment Model</td>
</tr>
<tr>
<td>GM</td>
<td>gradual mode (algorithm)</td>
</tr>
<tr>
<td>GMt</td>
<td>time-based (extended) gradual model (algorithm)</td>
</tr>
<tr>
<td>GUI</td>
<td>graphical user interface</td>
</tr>
<tr>
<td>HSMS</td>
<td>high-speed message service</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>I300I</td>
<td>International 300-mm Initiative</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>ILD</td>
<td>interlevel dielectric</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>IMA</td>
<td>integrated moving average</td>
</tr>
<tr>
<td>IMC</td>
<td>internal model control</td>
</tr>
<tr>
<td>KIRC</td>
<td>knowledge-based interactive run-to-run controller</td>
</tr>
<tr>
<td>MES</td>
<td>manufacturing execution system</td>
</tr>
<tr>
<td>MIMO</td>
<td>multiple input, multiple output</td>
</tr>
<tr>
<td>MMSE</td>
<td>minimum mean squared error</td>
</tr>
<tr>
<td>MSD</td>
<td>mean square(d) deviation</td>
</tr>
<tr>
<td>MSE</td>
<td>mean square error</td>
</tr>
<tr>
<td>NCS</td>
<td>(SAN) Network Communication Standard</td>
</tr>
<tr>
<td>NP</td>
<td>non-product (wafers)</td>
</tr>
<tr>
<td>NSF</td>
<td>National Science Foundation</td>
</tr>
<tr>
<td>OAQCC</td>
<td>optimizing adaptive quadratic controller</td>
</tr>
<tr>
<td>OBEM</td>
<td>Object-Based Equipment Model</td>
</tr>
<tr>
<td>OEE</td>
<td>overall equipment effectiveness</td>
</tr>
<tr>
<td>OEM</td>
<td>original equipment manufacturer</td>
</tr>
<tr>
<td>OES</td>
<td>optical emission spectroscopy</td>
</tr>
<tr>
<td>OMT</td>
<td>object modeling technique</td>
</tr>
<tr>
<td>PCC</td>
<td>predictor–corrector controller</td>
</tr>
<tr>
<td>PFM</td>
<td>Process Flow Specification Manager</td>
</tr>
<tr>
<td>PID</td>
<td>proportional integral differential</td>
</tr>
<tr>
<td>PLS</td>
<td>partial least squares</td>
</tr>
<tr>
<td>PM</td>
<td>process maintenance</td>
</tr>
<tr>
<td>R2R</td>
<td>run-to-run</td>
</tr>
<tr>
<td>RHS</td>
<td>right-hand side</td>
</tr>
<tr>
<td>RLS</td>
<td>recursive least squares</td>
</tr>
<tr>
<td>RR</td>
<td>removal rate</td>
</tr>
<tr>
<td>SAN</td>
<td>Sensor Actuator Network</td>
</tr>
<tr>
<td>SDM</td>
<td>(SAM) specific device model</td>
</tr>
<tr>
<td>SECS</td>
<td>SEMI Equipment Communication Standard</td>
</tr>
<tr>
<td>SEMATECH</td>
<td>Semiconductor Manufacturing Technology</td>
</tr>
<tr>
<td>SEMI</td>
<td>Semiconductor Equipment and Materials International</td>
</tr>
<tr>
<td>SIA</td>
<td>Semiconductor Industry Association</td>
</tr>
<tr>
<td>SISO</td>
<td>single input, single output</td>
</tr>
<tr>
<td>SPC</td>
<td>statistical process control</td>
</tr>
<tr>
<td>SRC</td>
<td>Semiconductor Research Corp.</td>
</tr>
<tr>
<td>TI</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>USD</td>
<td>United States dollars</td>
</tr>
<tr>
<td>VLSI</td>
<td>very large-scale integration</td>
</tr>
<tr>
<td>WECO</td>
<td>Western Electric Co. (SPC rules)</td>
</tr>
<tr>
<td>WMSE</td>
<td>weighted mean-squared error</td>
</tr>
</tbody>
</table>
Introduction

James Moyne and Arnon Max Hurwitz

The semiconductor manufacturing industry is arguably the fastest evolving major industry in the world. Success in the industry requires constant attention to the state of the art in process tools, process chemistries and physics, and techniques for processing and process improvement. The two major fronts along which product advancements are made in this industry are minimum feature size and wafer dimension. At the time of this writing, the “state-of-the-art” minimum feature size was in the 250 to 180 nm range, while processing on 300 mm wafers was becoming more prevalent.

As feature sizes shrink and wafer sizes increase, the industry must innovate to maintain acceptable product yield, throughput, and overall equipment effectiveness (OEE). Some manufacturing capability attributes, such as non-product wafer (NPW) usage and wafer scrap, must actually be improved in the transition to larger wafer sizes because of the increased value of 300 mm wafers (raw and processed). For example, one user reported that a raw 300 mm wafer cost approximately $1500 to $2000.* Although the cost of a raw 300 mm wafer may ultimately drop to a few hundred dollars, the value of a processed 300 mm wafer increases during its many process steps, ultimately representing more than 1000 devices worth $10 to $100 each. Faults introduced in any stage of manufacturing will often only show up in final electronic testing, and the consequent device loss may be (cost-wise) quite devastating — especially with these large-diameter wafers.

Although a number of solutions, including improved equipment design and process innovation, will continue to aid in making these transitions cost effective, it has become clear that they are no longer sufficient. Specifically, it has become generally accepted that process and wafer quality sensing and subsequent process tuning will be required to complement these equipment and process improvements. The main form of process tuning that is being implemented as a standard process and equipment control solution in the industry is run-to-run (R2R) control. As will be shown throughout this book, R2R control is now a proven and available technology, and has become a critical component of the success of existing and next-generation fabrication facilities.

In this introduction we provide important information that lays a foundation for understanding the concepts, motivation, and directions presented throughout the text. Specifically, in the following sections we provide a definition of R2R control, explore the qualities of a VLSI process candidate for R2R control, describe basic characteristics of R2R control systems, provide a brief history of the development of R2R control as a component of advanced process control, and summarize the layout of the book.

1 WHAT IS RUN-TO-RUN CONTROL?

Run-to-run control is a form of discrete process and machine control in which the product recipe with respect to a particular machine process is modified *ex situ*, i.e., between machine “runs,” so as to minimize process drift, shift, and variability. This type of control is event-driven, where the events include the determination and reporting of pre- and/or postprocess *ex situ* metrology data, and the requirement of the tool to begin processing. The input/output structure of a typical R2R control solution is shown in Figure 1. Note the granularity of control could be wafer-to-wafer, or batch-to-batch, etc.

A typical scheme for utilization of R2R control for a CMP polishing tool is illustrated in Figure 2. (Note: GCC stands for Generic Cell Controller, a control structure to be discussed in Part 3 of this book.) Note that the metrology and automation scheme for R2R control can vary widely. For example, the metrology is generally limited to *ex situ* metrology, but could include *in situ* equipment state.
and wafer state information. The (ex situ) premetrology capability may or may not be available. The metrology could be in-line or off-line, i.e., it could be directly integrated (mechanically and electrically) into the process line, or could exist as a stand-alone metrology station. The metrology could be fully integrated into a single tool, or could be integrated into the process line as both a postmetrology capability for a process and a premetrology capability for the subsequent downstream process.

2 VLSI TOOLS: THE EXAMPLE OF CMP

As VLSI (very large scale integration) technology advances, the feature sizes of both the underlying devices and the underlying metal line widths decrease. With this decrease comes increased transistor speed and density, but also a need for more layers of metal interconnect. Thus interconnect technology is the center of much of today’s VLSI research.

One of the major problems with fabricating additional layers of metal interconnect is that the topography of the silicon wafer becomes increasingly nonplanar as levels of metal are added. This, coupled with the demand for increasingly smaller geometries, has led to some problems previously unseen. First, due to the clarity of image needed for submicron geometries, the focal depth of lithography machines has decreased. This reduced focal depth results in some of the topography of the wafer being out of focus when other parts are in focus (see Figure 3). This is unacceptable as geometries shrink.

In addition to lithography concerns, the nonplanar surface can lead to difficult processing as the aspect ratio of the valleys of the wafer becomes great enough that the interconnect metal is unable to fill and cover these areas. This effect can lead to circuit failure due to metal fatigue or lack of connection entirely. Figure 4 shows a typical nonplanar process as well as an ideal one.

There are many techniques used to increase planarity. Most involve applying a level of glass or oxide in an attempt to fill the valleys that can lead to trouble later on. The problem is that the peaks are also extended to some extent, so it is very difficult to achieve planarity through this process alone. The process of etching peaks

FIGURE 3 Enlarging lithography system.
from the dielectric has also been attempted; again, this suffers from the inability to etch peaks while leaving valleys unchanged. CMP (chemical mechanical planarization) solves this problem by using a combination of chemical etching and physical abrasion to achieve global planarization.

CMP has its roots in the silicon wafer production machines used to polish the wafers before processing. These machines provided a wealth of information that led to the CMP machines of today. The basic process is the same for both. Wafers are loaded into a vacuum grip carrier that can rotate. This is then pressed against an abrasive pad that can also rotate. The lower pad is much bigger than the wafer, and is continually coated with a chemical slurry by a nozzle. Figure 5 shows a schematic of a simple CMP machine.

Through the use of CMP, near ideal planarization can be achieved. This has allowed VLSI manufacturers to increase the number of interconnect layers. It has also aided reliability by reducing the mechanical strain in metal lines resultant from nonplanarity.

CMP is not without its flaws. In addition to its high cost, it has nonuniformity issues that are the center of much CMP research. Nonuniformity issues can arise both within a wafer and between two wafers. Within-wafer uniformity is measured by comparing the relative thicknesses of the wafers along various sites located radially from the center. The reason for this approach rather than a more uniform pattern is that CMP involves rotating the wafer, which makes all sites that are radially equal the same thickness. Figure 6 shows the two methods of measurement.
Uniformity between wafers is measured by comparing the average thickness between two wafers. This measurement is related to the overall drift in a machine. This drift can have many sources; among them are pad wear and changes in slurry composition.

3 CHARACTERISTICS OF R2R CONTROL SYSTEMS

Although there is a wide range of R2R control system scenarios and solutions, there are three basic characteristics common to all R2R control systems.

- Some form of postprocess quality measurement data is available. This measurement data may be in the form of *ex situ* postprocess metrology (traditional), but could also include *in situ* data compiled during the process. Note that the data may or may not be available for every wafer, batch, or control event. Note also that preprocess measurement data availability is not a requirement.

- A dynamic model of the process is maintained (explicitly or implicitly) in the controller that relates the postprocess quality data to tunable process “recipe” inputs. Using this model, the controller is able to provide “suggestions” for process improvement as necessary based on postprocess quality data values. The model is dynamic in that it attempts to track drifts in equipment and process quality parameters on a run-to-run basis.

- Process improvement control actions, i.e., process input parameter adjustments are instituted once during each “run” based on suggestions by the controller. A “run” may be defined as a single wafer process event, batch process event, etc. The control actions are generally instituted prior to the commencement of a run event (i.e., between “runs”); however, this is not necessarily required.

A typical R2R control solution was shown in Figure 2. Here, a wafer-to-wafer control scheme is utilized to provide thickness and uniformity control of a chemical mechanical planarization (CMP) process through R2R actuation of time and back-pressure. Only postmetrology data is utilized, 50% of all wafers are measured, and there is a two-wafer delay in actuation (i.e., data collected on wafer $n$ is utilized first in the control of wafer $n + 2$).
4 HISTORY OF THE DEVELOPMENT OF R2R CONTROL

Early algorithmic and system integration development for run-to-run (R2R) control in the semiconductor industry was pioneered by researchers at MIT and at the University of Michigan as well as by workers at various semiconductor manufacturing corporations in the U.S., most notably at Texas Instruments. In the early 1990s some of this key research began to be sponsored by the Austin, Texas-based semiconductor research consortium, SEMATECH.* References to SEMATECH, its research, and its affiliations may be found at the Web site http://www.sematech.org.

Continuous feature size reduction and wafer size increase have forced the players in the semiconductor industry to innovate to remain competitive. The innovations that have taken place to maintain and improve manufacturing capability attributes have historically been in the form of (primarily) equipment design improvement, process refinement, and, in a few cases such as chemical mechanical planarization (CMP), new process conception and development. These innovations, for the most part, could be characterized as process-centric; that is, a specific tool and/or its specific process is improved with the expectation that it will lead to improvement in overall factory throughput and yield.

In the mid-1980s the industry began to become more cognizant of other critical avenues for maintaining and improving process and product metrics. One of these avenues was improved process and product visibility, namely a much-improved ability to view and understand aspects of the process and the wafer during processing. This attempt at increased visibility was focused not only in the traditional off-line process identification and development areas, but also on-line as part of the fabrication scheme, i.e., attempting to ascertain the state of the wafer and process during processing (in situ sensing) and the state of the wafer directly after processing (ex situ metrology). Research and development organizations such as the Semiconductor Research Corp. (SRC) and SEMATECH identified projects to develop metrology technology (both in situ and ex situ) as well as physical and empirical process models. Research investigators also began to look seriously for the first time at each process as part of a total factory solution. Another force that appeared and provided guidance toward the evolution of the industry in this area was the Semiconductor Industry Association (SIA). The SIA began publishing a roadmap that attempted to capture the state of the art in the semiconductor industry, “hot” areas where research and development should be focused in the near future, and basic timelines for achieving milestones in these areas.

With increased process and product data in hand, process engineers began to investigate ways to put this sensory and metrology data to use. A natural first step was to utilize the additional data to enhance existing process alarming and control mechanisms. The predominant mechanism used in the industry at that time (and to this day) is statistical process control (SPC). SPC is a method for detection of statistically significant data patterns based on an assumption of a Gaussian distribution of data. Mean and variance parameters are determined for various data parameters through data collection and analysis. The sensory data are then monitored with

* SEMATECH has since changed its name to International SEMATECH.
respect to these mean and variance parameters and alarm events are generated. Specifically, a number of detection rules, called the Western Electric Rules, are applied to the data. When conditions of a rule are met, an alarm is generated.

While SPC is useful to detect and verify process stability and correctness, it is not technically a “control” solution. This is because SPC provides a mechanism for detecting aberrations but does not include mechanisms to correct for these aberrations. As a consequence of this limitation, in the late 1980s and early 1990s researchers in the industry began to complement sensory and metrology research with a focus on utilizing the data to suggest corrective action. This effort, termed “advanced process control” (APC), was spearheaded in the industry by efforts at SEMATECH and SRC. The primary consideration in control solution development in the semiconductor manufacturing arena that differentiated it from other arenas was the problem of lack of available sensory capability, especially for in situ process and product identification. The semiconductor industry is characterized by physically and chemically prohibitive environments (e.g., high-temperature processes, corrosive and hazardous processing chemistries, plasma environments, etc.), making sensor development and use in many of these environments difficult or impossible. This, combined with the fact that many process tools are not designed for the addition of in situ sensors (e.g., requiring the addition of quartz windows for in situ optical sensing), resulted in inadequate sensing for in situ process identification and control. This in turn resulted in labeling the control effort in the industry as “sensor-driven control,” where sensor development drove and directed control development rather than the reverse.

The lack of adequate (especially in situ) sensing in semiconductor manufacturing provided control systems researchers with unique challenges and opportunities for advancement. The first innovations in semiconductor manufacturing APC were the partitioning of the controlled process into the in situ control of the equipment environment, the in situ control of the equipment environment operating on the wafer, and the ex situ control of the final product for that process. This gives a hierarchical, nested partitioning of the control problem, as illustrated in Figure 7 (from a factory operations perspective) and Figure 8 (from a controls perspective).

Each level of control is characterized by a set of sensory/actuation capabilities (i.e., a process visibility and control capability) and control timing requirements. For example, referring to Figure 7, in controlling the equipment environment, the process is continuous. Thus the control should be continuous, or the time discretization should be sufficiently small so that the environment doesn’t drift significantly between control events. Similarly, the wafer processing environment is also continuous. However, timing requirements are generally more relaxed than with the equipment environment because the wafer effect from the equipment environment is cumulative and is impacted by factors such as the time constant of the process chamber (i.e., the time required for an equipment environment actuation event to begin to noticeably impact the wafer environment). At the outermost loop of the control system depicted in Figure 8, the ex situ control environment is strictly event-driven (and not continuous).

Wafers are analyzed after processing and corrective advice is fed back to the tool for future wafer processing. Note that the outermost loop depicted in Figure 8
does not necessarily represent the highest control loop in the control hierarchy. Specifically, interprocess control could be implemented that “wraps” control around a group of processes.

The partitioning of the control problem is powerful because it allows the development of effective control solutions when “good” process visibility is limited to one or two levels.

With the industry accepting the hierarchical nested control solutions approach, research and development could now be focused on a particular level of control. Terminology somewhat specific to the industry was attached to each of these control levels. \textit{In-situ} process environment and wafer environment control were collectively
termed *real-time, in situ, or time-critical* control, while *ex situ* process control was termed *run-to-run (R2R)* control.* In partitioning the control problem and control research, it became clear very quickly that, due to the available sensing capability and process knowledge, R2R control represented the first primary area where process improvements could be readily achieved.

During the 1990s innovation in R2R control development and deployment progressed on three basic fronts described below; this multidimensional approach represents a model for advancement that is also applicable to the other levels of process control.

1. *Sensor and Actuator Innovation*: Progress along this front provides increased process and equipment visibility.
2. *Control Algorithm Solution Innovation*: The development, analysis, and parameterization of control algorithm solutions are necessary to address the often unique process visibility and R2R controllability problems of the industry.
3. *Integration and Automation Innovation*: The most often overlooked front is the development of integration and automation techniques for rapid deployment (re)configuration and reuse, and assimilation of R2R control as part of multilayer control solutions.

Progress along these three fronts was paramount in determining the progress of the various APC thrusts. Indeed, it was probably the determining factor that pushed R2R to the forefront as the first widely implemented APC element. Specifically, the following are a few of the important factors that pushed R2R control to the forefront ahead of *in situ* control and interprocess control:

- *Sensor and Actuator Innovation*: R2R control is implemented where there is the highest level of process visibility (or at least capability for visibility) and actuation capability. As sensors can be implemented *ex situ*, issues of harsh processing environments and lack of equipment design for process/product inspection are nonexistent. Actuation is simply process recipe modification between runs. Thus R2R control is, in most cases, just an extension of process identification, design, implementation, and tuning (i.e., R2R control reflects uploading/downloading and tuning of recipes between runs similar to run-to-run actions of an operator). Additionally, process engineers have much more knowledge and confidence in operating and tuning a process R2R rather than either *in situ* or at an interprocess level. This is because process knowledge in a fabrication facility is primarily process-centric and run-based. A process engineer knows his/her process well in terms of the product it produces. He/she is generally less aware of the impact of up- or downstream processes, and does not have as much knowledge of process dynamics during processing. Simply put,

* The term “run-by-run” or “RbR” control is also used.
the process engineer knows his/her process from the perspective of the product it produces, and this is the perspective from which R2R control is applied.

- **Control Algorithm Solution Innovation:** R2R control addresses a relatively straightforward process-centric control problem. Simple and intuitive discrete control solutions can be applied in many cases. Further, as R2R control addresses process tuning that is already conducted *ad hoc* by operators and process engineers, it is already part of the semiconductor manufacturing industry culture.

- **Integration and Automation Innovation:** R2R control is event-driven and, thus, in most cases, not time-critical. In addressing integration and automation, attention could be focused on the solution architecture rather than response time. This has led to the development of generic, portable, and reusable software solutions that exist on common computer hardware/software platforms. Further, because of the non-time-critical nature of R2R control, stand-alone, nonintrusive prototype R2R systems can be deployed in advance of the fully integrated systems to verify and quantify the advantage of R2R. With these stand-alone systems, the user serves as the communication link between the controller and both the tool and metrology systems, and as a validation mechanism for controller advances. With the stand-alone implementation intermediate step, a smooth migration path is provided for acceptance of R2R control in the fabrication facility.

With the advantages of available sensory and actuation capability, straightforward control solutions, and integration and automation strategy giving a smooth migration path to fully automated R2R control, the industry began to focus more heavily on R2R control as the first advanced process control (APC) paradigm.

## 5 TEXT LAYOUT

This book is organized so the reader can quickly map his needs to the required parts and chapters within those parts. The remainder of the book is subdivided into seven parts; each part addresses an important aspect of R2R control development, deployment, and assessment. At the beginning of each part, a brief description is provided as to the contents of that part. This is followed by chapters addressing specific topics.

Part 1 addresses foundational material including an overview of process control in the industry and an overview of process control and optimizations. The two fundamental components of a successfully deployed R2R control solution, namely the control algorithm(s) and the integration methodology, are addressed in Parts 2 and 3, respectively; these parts also contain brief examples of R2R control solution deployment. Methodologies for customization of R2R control solutions to actual industry control problems are provided in Part 4. Part 5 contains actual detailed case studies of R2R control solution deployment. Advanced “next generation” topics in R2R control and semiconductor manufacturing process control in general are discussed in Part 6. Part 7, entitled “Conclusions,” provides a summary of what has
been presented and offers final thoughts on R2R control solution development, deployment, and evolution.

One unfortunate characteristic of the field of R2R Control, and indeed, semiconductor manufacturing in general, is the extensive use of acronyms. To address this problem, a complete listing of acronyms used in this text is provided at the end of the book.

6 SUMMARY

This introduction serves to open the door to the subject matter of this book as well as to general thinking about R2R control systems and their integration into manufacturing systems. It should be clear by now that the technology discussed is completely generic with respect to manufacturing control and integration. Even though the experience of the authors, and all quoted examples, pertains to the semiconductor industry, there is no reason why any other industry cannot take advantage of this form of control. In this sense, R2R control is similar to the well-known SPC (statistical process control).

To summarize some key features of R2R control, one may cite its wide applicability to tools of all types, its dependence on certain algorithms that again have wide generic applicability, its need for a coherent integration path in manufacturing, and certain common measurement requirements. In exploring these features and developing solutions, researchers and implementers have been able to take the technology as presented in this book from a merely academic exercise into the world of actual tool, and factory, application.

It is the authors’ hope that this text will not only serve to spread the use of R2R control further into semiconductor manufacturing, but will also inspire and motivate engineers, managers, and scientists in all branches of manufacturing to apply R2R control in their particular areas.
Part 1

Foundation for Control

Process control is gaining recognition in the semiconductor industry as a means to compensate for equipment changes due to inconsistent operation, process drifts, and process shifts. In the Introduction to this book an overview was provided of R2R control concepts and the history of the development of R2R control. In Part 1, practical and theoretical components that comprise the foundation for R2R control are presented.

In Chapter 1 several factors that led to the recent growth of R2R process control are reviewed. The motivation for migration from traditional statistical process control (SPC) techniques to adaptive control is explained. Problems are discussed that led to the early interest in process control and the resulting solution of these problems. Examples are drawn from semiconductor processes such as plasma etching, chemical mechanical planarization (CMP), and metal sputter deposition. Several issues are outlined that arose from the initial interest in adaptive process control, including a lack of commercial solutions, a common framework, \textit{in situ} sensors and on-line metrology, and detailed studies of algorithm performance. A description is given of how these issues were addressed as the new control technology matured, including the development of a framework for control, the rise of commercial applications, the development of new algorithms, and the introduction of new metrology. Chapter 1 also explains why methodologies that have obeyed the “keep it simple” rule have achieved widespread use. Suggestions are made for the future of process control in the semiconductor industry. Process control application benefits are summarized. Possible future trends in control are suggested that indicate a move toward methodologies that replace “black-box” systems with those that incorporate advanced models tightly coupled with the process, as well as methods that provide multistep process optimization and control.
In Chapter 2, an overview of R2R optimization techniques is given. Three methods are reviewed, namely (1) design of experiments (DOE)/response surface methods (RSM), (2) the Ultramax® sequential optimization software, and (3) the optimizing adaptive quadratic controller (OAQC), which is described in greater detail in later chapters. The stability and robustness of exponentially weighted moving average (EWMA)-based controllers, both single-weighted and double-weighted, are discussed in detail, and methods for tuning these controllers are described. Other important R2R control techniques are briefly reviewed. One of the goals of this chapter is to give an up-to-date review of available R2R methods.
1 Advanced Process Control in the Semiconductor Industry

Taber H. Smith, Duane S. Boning, and James Moyne

1.1 INTRODUCTION

Advanced process control, or APC, has evolved rapidly in the semiconductor industry during the 1980s and 1990s, with R2R control emerging as the first technologically viable product of that evolution. In the Introduction to this book a detailed definition of R2R control is provided and the evolution of R2R control is summarized from the point of view of capabilities of the industry and organizations that had a significant impact on the evolution of APC. In this chapter we provide an in-depth view of the evolution of advanced process control (toward R2R control) in the semiconductor industry from a technical perspective, describe the issues that are guiding the maturation of R2R control, and detail the benefits that can be achieved with effective R2R control.

Specifically, in Section 1.2 we provide an historical summary of the evolution of process control from alarm-based statistical process control toward model-based control solutions such as R2R control. Further, we describe technical issues that have proved to be a hindrance to this evolution and, subsequently, the widespread acceptance of R2R control. In Section 1.3 we summarize advancements that have been made in the field of R2R control that have helped to address these issues. Many of these advancements are described in detail in later chapters. We follow this summary with a discussion of the future of APC in Section 1.4. Here, we focus our attention on the benefits of R2R control (current and future), as well as probable directions for R2R control solution enhancement. This chapter concludes with a summary of the information presented.

The main purpose of this chapter is to provide the reader with a snapshot of the technical issues — past, present, and future — that are shaping the field of R2R control in semiconductor manufacturing. With this information, the reader is provided with a foundation for understanding the detailed aspects of R2R control that are described in the chapters that follow.
1.2 INITIAL STAGES OF CLOSED-LOOP PROCESS CONTROL

1.2.1 EARLY SOLUTIONS: STATISTICAL PROCESS CONTROL

As semiconductor processing entered the late 1980s, process variability had been substantially decreased. However, new problems were beginning to emerge. Many processes were exhibiting steady drift in equipment performance. Such drifts were often caused by the build-up of material on the interior components of the tools, or gradual wear of components. For example, the deposition rate in a metal sputtering process is highly correlated to the life of the components within the tool. The resulting drift in the deposition thickness is shown in Figure 1.1.

Although not intended for process adjustment purposes, statistical process control (SPC) is often used to compensate for such problems. SPC is a technique aimed at monitoring deviations from statistical control, a state of a process where measurements follow a stable, uncorrelated process. Derivations are usually identified through applying a set of rules or filters to the data. Adjustment to the process, i.e., removal of the assignable cause of variation, is usually left unmodeled, under the assumption that a process engineer will try to fix the problem at its root. Removal of the cause of variation avoids further occurrences of this problem in the future.

One set of violation rules used to determine the level of statistical control is known as the Western Electric Company (WECO) Rules.* A subset of these rules follows:

---

* WECO rules are usually not recommended in the SPC literature since they may cause a considerable increase in the number of false alarms given by the monitoring scheme.
1. Last point of data is greater than three standard deviations away from the process target.
2. Two of last three data points are greater than two standard deviations away from the target.
3. Four of last five data points are greater than one standard deviation away from the target.
4. Last eight data points are all above or all below the target.

In semiconductor manufacturing the process output is often shifted back to the target by a simple adjustment to the process (in the sputter deposition case, this is achieved by adjusting the deposition time). The amount of the adjustment is typically equal to the sample mean of the data over the violation set (i.e., the last five data points for WECO rule #3). As can be seen in Figure 1.2, the performance of this method can be particularly poor in the sense that tools oftentimes experience regular shifts or drifts in their outputs, and typical SPC detection methods such as the WECO rule set are not very adept at distinguishing these shifts or drifts from process noise. Although many process drift and shift conditions are eventually caught by SPC, the problems are often only solved by warming up or otherwise “seasoning” the tool, or by simply adjusting the processing time. In addition, the frequent occurrences of these events were beginning to require significant manual monitoring efforts. Other processes randomly drifted away from the target output, but then drifted back the other way, continually wandering about the target.

Processes with these characteristics call for active adjustment strategies based on control engineering principles, as opposed to the use of ad hoc SPC adjustments. Furthermore, SPC implicitly assumes that adjusting a process is a very expensive

**FIGURE 1.2** SPC control of a drifting process using tuning with WECO rules.
activity to be utilized only when there is strong evidence that a process has been affected by an extraneous source of variation. If adjustments are fairly simple and inexpensive, as in many R2R control applications, a continuous “run-to-run” adjustment scheme is preferable over SPC (see Reference 1 for a discussion on the difference between SPC and engineering process control, EPC).

1.2.2 NEW SOLUTIONS — RUN-TO-RUN PROCESS CONTROL

In response to the need for continuous process tuning solutions, run-to-run (R2R) process control algorithms began to emerge in university and industrial research. These methods were the first closed-loop feedback controllers to be used at the process level in the semiconductor industry. They are similar to SPC in that they monitor process parameters such as the deposition rate. However, unlike SPC techniques, these methods make continual changes to the process, usually based on dynamic modeling of process parameters. Many semiconductor fabrication facilities (fabs) were already making frequent (though somewhat ad hoc) changes to processes in order to compensate for drifts and shifts in the process outputs detected via techniques such as SPC. Therefore, these controllers provided a natural step from SPC to closed-loop feedback control.

One class of R2R controllers simply replaced the manual adjustment in SPC with an automatic adjustment. This effort was generally aimed at automating current practice within a fab. However, the controlled results were similar to those shown in Figure 1.2. In some cases these controllers were expanded to include a group of related processes. For example, SPC was used to control thicknesses of several processes by utilizing a simple relationship between the deposition rate for thickness and a common offset caused by equipment drift. This type of control was often very effective for a large class of shifting processes in the industry. A second class of
R2R controllers was based on the exponentially weighted moving average (EWMA). This scheme is shown in Figure 1.3, and is explained in detail in Chapter 3.

Briefly, an EWMA was used to track changes in the process using the following recursive algorithm:

\[ a[n] = w \cdot y[n] + (1 - w) \cdot a[n-1] \]  \hspace{1cm} (1)

where \( a[n] \) is the EWMA estimate of the process output, \( w \) is the EWMA weight, and \( y[n] \) is a measurement of the process output or parameter to be estimated on run. It can be seen that a higher EWMA weight means more recent measurements are weighted more heavily. This weighted average of the process offset is used to update the model of the process. This dynamic model is then used to adjust the equipment settings to control the process outputs.

As an example, an EWMA controller may be used to track one or more process outputs, such as deposition rate, and adjust the process inputs, such as process time, to control one or more process outputs, such as the final film thickness. The results of using an EWMA controller to control the deposition thickness of the sputter deposition process above are shown in Figure 1.4. This figure illustrates that the continual tuning of the process results in fewer regions where the control is off target. Note also that there would be fewer points outside typical specification limits.

Acceptance of this class of feedback control was initially slow, largely due to suspicion that frequent changes in the process settings would cause unseen changes in critical film properties. However, several works demonstrated the effectiveness of
the use of R2R process control,\textsuperscript{2–12} and acceptance of R2R process control began to grow.

1.2.3 \textbf{New Issues Facing R2R Control Adoption}

Those interested in fab-wide implementations of advanced process control, and specifically R2R control, realized there were several major barriers to be overcome. In particular, many fabs wished to implement R2R process control, but believed they neither had nor wanted the necessary skills to develop a commercial system in order to implement it. Unfortunately, there was no commercial solution that they could turn to for implementing R2R control or providing R2R control expertise. In addition, it was soon recognized that adopting and integrating R2R control systems on different tools in large numbers over a long period of time would require a substantial upgrade to existing factory systems. The lack of a consistent standardized or proven approach to integration of R2R process control presented a time and financial resource barrier to implementation. Many tools also did not provide enough measurement information to make R2R process control an effective tool. Another perceived problem was that, in order to obtain the critical parameters required to adequately control the process, a large amount of metrology and \textit{in situ} sensor development was necessary. Finally, many felt that the existing algorithms might not be sufficient to control some processes, and there were concerns regarding the stability, optimality, and robustness of these methods.

The issues that impeded the early adoption of R2R control are summarized in Table 1.1. Note that this table also (briefly) summarizes the issues, presents concepts and solutions that have been forwarded to address the issues, and provides references to other chapters in this book where these concepts and solutions are described in

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
\textbf{Issue} & \textbf{Solutions} & \textbf{References in Book} \\
\hline
Lack of commercial solutions & Third-party solution providers with standard solutions (see below) & Chapters 11, 13, 15 \\
No infrastructure for integration or automation & Standard framework for specification and integration: & Chapters 7–10 \\
& \quad \begin{itemize} \item Generic Cell Controller \item APC framework and enablers \end{itemize} & Chapters 8, 9, 11, 12 \\
Few on-line metrology and \textit{in situ} sensors & New sensors and sensor integration standards & Chapters 7, 10, 20 \\
Inadequate algorithms & Verification of quality of existing algorithms & Chapters 1, 7 \\
& Algorithm comparison: establishing domains of applicability & Chapters 11, 13, 15, 16 \\
& Improved algorithms & Chapter 6 \\
& & Chapters 4, 5, 17, 18, 19 \\
\hline
\end{tabular}
\caption{Summary of Issues Impeding Early Adoption of R2R Control, Concepts and Solutions, and References}
\end{table}
detail. Further, an elaboration on these issues impeding the early adoption of R2R control is provided in the remainder of this section, while a summary of some of the concepts and solutions advanced to address the issues is provided in Section 1.3.

1.2.3.1 Issue: Lack of Commercial Solutions

Implementing control in a production environment proved much more difficult than implementing it in a laboratory or research facility. This is due to a number of factors. For example, many of the tools have only coarse adjustments on the equipment settings, thus increasing the variability due to control actions. Further, these tools are often shut down for maintenance, which causes large shifts in the process outputs after being brought back up. Consider the deposition rate history of a sputter deposition process shown in Figure 1.5. The abrupt changes in the deposition rate are due to changes in the process kit. Notice that the starting values are very different, and the drift in the deposition rates are slightly different for each process kit.

At the time that R2R control solutions began to appear (early 1990s), measurements were almost exclusively performed off-line, and these were often slow, inconsistent, or skipped by operators. This caused delayed, inconsistent, and infrequent measurements of the process outputs. As illustrated by our sputter deposition example, this can have a large effect on the measured deposition thickness. Most of the controller implementations developed at that time gave little attention to these minor details. However, as shown in Figures 1.6 and 1.7, neglecting these details can cause a large increase in the variability of the controlled output. Figure 1.6 shows a drifting process with no delays, inconsistencies, or skipped measurements. Note that the

![Figure 1.5 Deposition rate of a sputter deposition process over several process kit changes.](image-url)
FIGURE 1.6 Control of a drifting process with no nonperiodic time intervals, infrequent measurements, or inconsistent delays.

FIGURE 1.7 Control of a drifting process with nonperiodic time intervals and infrequent measurements with inconsistent delays.
controlled output is consistently on target. On the other hand, Figure 1.7 shows the resulting control when measurements are often delayed, inconsistent, and skipped.

Another problem with implementing laboratory results in practice was that, for the most part, little attention was paid to cost-effective, repeatable solutions for integration and automation. The industry in general will not accept open-loop or manual forms of R2R control as long-term solutions. Specifically, R2R control will only be accepted if the user is not burdened with additional data entry at the controller (e.g., entering metrology information) or at the tool (e.g., entering tool process parameter update information).

In addition, the manual operation of most tools often left room for data-entry, user-interface, and security problems within control software. For example, operators manually entering data would sometimes use an incorrect metrology recipe, incorrectly enter data from the metrology tool, or make other minor errors. However, recipe changes made by the controller based on these measurements could result in several runs being significantly different from the target output.

Further, the nature of the process disturbances (e.g., drift, shifts, or wandering noise) often changes, and many process engineers lacked the time or ability to continually retune all the controllers in the manufacturing environment. This often resulted in the controllers running in a suboptimal mode. The lack of commercial solutions to address these issues was thus a serious impediment to the implementation of R2R techniques in the manufacturing environment.

1.2.3.2 Issue: No Enabling Technologies or Infrastructure for Integration or Automation

Probably the single largest barrier to the implementation of R2R process control in production facilities was the lack of a cost-effective enabling technology and infrastructure that would provide for (1) solutions to be rapidly prototyped and configured that were flexible and process-independent to achieve cost-effectiveness; (2) integration of software components, including third-party solutions, in a timely and cost-effective manner to achieve the necessary customization and address unique end-user requirements as is typical with implementation of a new technology; (3) communication capabilities where tools and sensors could be connected and data could be uploaded, transferred, and saved to support fully automated solutions; and (4) commercial quality applications to obtain the required level of software quality.

Specifically, an enabling technology was needed that would allow developers to rapidly integrate and configure the necessary software components (including control algorithms, data logging and presentation modules, communication drivers, and data filters) to quickly prototype and customize R2R control solutions. This enabling technology further had to provide for portability of these solutions between hardware/software platforms and semiconductor processes to achieve the necessary cost and technology leverage. It also had to provide a level of flexibility for rapid reconfiguration (to new process paradigms) and rapid upgrade (to maintain competitive advantage in the early stages of a technology market).

An infrastructure or framework was also needed to provide a consistent base upon which current and future generations of controllers, including R2R control,
real-time control, and endpoint detection modules, could be built. The large amount of research into on-line metrology and *in situ* sensors, as well as tool communication, required standard methods for data upload, download, and storage. This was necessary so that the factory system would be able to upload, download, store, and share information; information regarding process recipes was particularly important. However, many tool, sensor, metrology, and factory systems use different operating systems. Data formats for many of these pieces were completely different. In addition, tools in the semiconductor industry were generally not highly automated. This was largely due to the rapid pace at which tool technology had changed within the industry and the high cost of automation. Typical tool lifetime was on the order of 2 to 5 years, with costs typically being on the order of millions of dollars. Additional efforts to configure and program a tool for automation were generally considered cost prohibitive, time prohibitive, or not possible due to limited automation capabilities of the tool or a limited automation skill set on site. This lack of automation contributed to tool utilization typically on the order of 30%, and clearly pointed to a need for a basic equipment and control framework.

1.2.3.3 Issue: Few On-Line Metrology and *In Situ* Sensors

In many complex processes, the lack of information available about the process state and the state of the wafer during processing presented a major barrier to the effectiveness of R2R process control. It was found that, for processes like plasma etching, the complex interactions of the tool (e.g., the build-up of material on the chamber walls) and the plasma chemistry make monitoring and controlling the process difficult. These processes were often difficult to model and control with *ex situ* measurements. Because changing process settings resulted in unseen changes in the thin film properties, scanning electron microscope (SEM) and other time-consuming measurements were often the only means to develop and control processes. In order to control such complex processes, more information about the state of the wafer and plasma chemistry was needed.

In other processes, such as chemical mechanical polishing (CMP), processing occurs in a much less controlled environment. In CMP, the wafer is pressed face down on a polishing pad. Even though the chemical and physical mechanisms may not be as complex as in plasma etching, obtaining access to the chemical and physical mechanisms taking place during processing is extremely difficult (as explained in the Introduction section of this book). Initially, this made *ex situ* measurements the only information available on the process.

The lack of information available and the lack of access to the wafer being processed forced R2R process control to rely largely on *ex situ* measurements. However, many felt that the speed at which these measurements were performed would decrease the usefulness of R2R process control in a manufacturing environment, and that if R2R was to be successful, the information necessary for control had to be taken on the tool without operator involvement. This was particularly true for tools that required frequent measurements (which were normally those most likely in need of control). R2R process control for processes that relied heavily on *ex situ* metrology
were most likely to suffer a loss in throughput if measurements had to be taken frequently. In addition, frequent measurement and recipe changes increased operator error in the measurement data entry stage, as well as in the recipe adjustment stage. These issues strongly suggested that, in many cases, R2R process control without automated on-line metrology or in situ sensors was not practical or cost effective.

1.2.3.4 Issue: Inadequate Algorithms

Interestingly, algorithm development at this stage was generally ahead of implementation. This was largely due to the enormous task of creating an infrastructure, developing on-line metrology and in situ sensors, and configuring tools for automation. At this stage most of the problems involving algorithms were those related to implementation in production environments. However, a large amount of research was still needed in order to address theoretical questions of stability, optimality, and robustness, as well as practical issues of input/output bounding and weighting, advice parameter discretization, and noise filtering.

1.3 MATURATION OF CLOSED-LOOP PROCESS CONTROL

As R2R process control began to mature, a large amount of work was focused on addressing the problems (summarized above) that were faced by the early adopters of R2R process control. In particular, a generic enabler was developed to provide for rapidly configurable, integrated control solutions;* this enabling technology has been utilized in a large number of R2R applications in the industry from the beginning of the R2R maturation process through present day. Somewhat later, a standard framework was set up, upon which commercial applications could be developed; interestingly, this framework specified an integration environment very similar to that utilized by the generic enabler. The promise of R2R process control, real-time process control, and fault detection and diagnosis sparked efforts to develop on-line metrology and in situ sensors by both industry and academia. Finally, several works were beginning to address stability, optimality, and robustness of R2R process control algorithms, and practical issues associated with algorithm customization and deployment (see Parts 2 and 6 of this book for additional references). In addition, novel control techniques were beginning to address a wider range of control problems (see Part 6 of this book for additional references).

In the remainder of this section, the efforts that contributed to the maturation of closed-loop process control, and specifically R2R control, are described in more detail.

1.3.1 Enabling Technologies for R2R Control Solutions

In the late 1980s a research effort at the University of Michigan, sponsored in part by the Semiconductor Research Corp. (SRC), focused on identifying (1) barriers to

* See Chapters 9 through 11.
advanced process control adoption in the industry, (2) user requirements for inte-
grated control solutions, and (3) a basic roadmap for developing and deploying APC
in the industry. An early result of that effort was the rather startling conclusion
that the major barrier to APC integration was not control technology, but rather
integration technology. Specifically, it was noted that there was very little reuse of
software in APC solutions, despite the fact that software was rapidly becoming a
larger component of tool cost and tool quality assessment. Further, it was noted that
the industry was suffering from a “not invented here” mentality, where software
solutions were often times proprietary and not developed by software engineers, and
there was no coordinated attempt to look beyond the semiconductor industry for
software solutions. It was also concluded that R2R control was the best initial
candidate for APC development and deployment due to (1) the infant state of
metrology and sensor technology at the in situ level, (2) a lack of consistent software
platforms at the equipment control level, (3) a process-centric — rather than interpro-
cess — mentality and knowledge base, and (4) a general unwillingness on the part of
users and OEMs* to deploy multivariate control solutions at the (invasive) equipment
control level or at the factory control level (where little was known about process-to-
process interaction, and few resources were available to support integration).

The researchers at the University of Michigan worked with industry and iden-
tified requirements for integrated control systems in semiconductor manufactur-
ing.** They then surveyed the software technology base, and utilized this foundation
to develop the Generic Cell Controller.*** This enabling technology for integrated
control was developed to provide flexible, rapidly configurable, and portable inte-
grated control solutions for the industry. Its main features include a well-defined
and modular object-oriented integration environment, and a method for incorporating
control schemes in the (portable and flexible) data of a database (rather than in
procedural code or in scripts). Since the first GCC implementation was demonstrated
in 1989, a number of industrial R2R control solutions have been reported in literature
that utilize the GCC technology.**** Novel R2R control results reported in
literature with these GCC solutions include first etch process control, first CMP
thickness control, first CMP uniformity control, first multivariate vapor phase epitaxy
multivariate control, and first fully automated control solution.

During the mid- to late 1990s, the advanced equipment control (AEC) advanced
process control (APC) framework was set up by SEMATECH and several of its
member companies that were interested in promoting the development of commercial
hardware and software for semiconductor process control applications.***** The
framework, which is based in large part on earlier work in Texas Instruments’
Microelectronic Manufacturing Science and Technology (MMST) program, is out-
lined in Figure 1.8. This framework provided a standard around which applications
from various companies could be built and integrated. These applications include
monitoring and control programs for R2R process control, fault detection and

---

* Original equipment manufacturers.
** The GCC concept is described in detail in Chapter 9.
*** GCC solution examples are reported throughout this text, especially in Chapters 11, 13, 15, and 18.
**** Requirements for R2R control solutions are presented in Chapter 8.
***** See Chapters 7, 10, and 20 for additional discussion of the APC framework.
classification systems, and SPC monitoring packages. In addition, separate applications for data sampling plans, data collection, and data history maintenance could be added. The framework also includes a structure for low-level equipment, sensor, and metrology interfaces. Because a large portion of the benefits of R2R control are derived from integration and automation, the standard interfaces for data storage and retrieval outlined by the AEC/APC framework provide a basis for this to happen. The standard allows semiconductor companies to adopt those packages and technologies that are best suited to their needs. In addition, the framework outlines interfaces for collecting and storing data from equipment and sensors, as well as for sharing these data with applications for fault detection and classification, R2R process control, and factory-level computer integrated manufacturing (CIM) systems. These standards, then, if adopted, could provide IC companies with the ability to obtain individual products from the vendors of their choice, while providing vendors the ability to sell and market individual tools on the platform of their choice.

1.3.2 Commercial Software

As APC and especially R2R control migrated from the laboratory to industry, commercial software solutions began to appear. These solutions generally fell into two categories, namely those developed as single point solutions, and those developed to be reusable and portable to multiple applications. Solutions in the latter category, by and large, utilized enabling technologies similar to those described in Section 1.3.1.21,22 The rise of commercial solutions accelerated in the late 1990s due to a number of factors, including (1) the basic realization by the industry that APC
would eventually become an integral component of semiconductor manufacturing, (2) the move toward 300 mm tooling and the opportunities for adding capabilities both at the OEM and user levels, and (3) the substantial effort devoted to creating a standard framework upon which applications could be built.

The rise of commercial R2R control solutions has seen an accompanying rise in solution features. For example, many of these controllers directly accommodate equipment maintenance events and provide solutions to rapidly adjust for such changes.* Many control solutions also include the ability to control multiunit processing tools within a single package. This often means separate controllers for each unit, but allows the operators to easily control specific chambers of specific tools within one common graphical user interface (GUI). Commercial packages often contain built-in SPC or limit monitoring of values entered by operators. In addition, many systems include separate interfaces and permissions for operators, engineers, and administrators. Packages often significantly remove operator involvement by incorporating communication with both the tool and on-line or ex situ metrology, resulting in a fully automated solution.** These packages often offer the ability to set up arbitrary inputs from various metrology tools on-line, in situ, and ex situ. This allows a great amount of flexibility for the production facility to buy one package for R2R control from one vendor, another package for database support from another vendor, and work out the details of the automation of specific equipment on a tool-by-tool basis. Finally, some commercial vendors are providing control “solutions,” rather than software packages alone. These vendors provide software, consulting, and may also completely integrate the controller into an existing factory system. This eliminates or helps reduce the need for on-site control experts.

1.3.3 New Algorithms

Since the development of initial R2R process control algorithms in the late 1980s and early 1990s, many works have provided in-depth analysis of existing control methods, making them more widely accepted and easier to use.23–30 Several methods have been developed to address many of the practical issues involved with implementing R2R process control in the manufacturing environment.30 Methods have been developed to effectively accommodate delays and inconsistencies in the data returning from ex situ metrology.26 Other works have focused on the need to effectively tune the controllers,27 while still others have focused on the comparative evaluation of control algorithms.31*** There has also been a significant effort to bring standard techniques from the controls field, including proportional integral differential (PID) control, internal model control (IMC), and robust control.28

In contrast to these developments, there has been a significant amount of work to develop new methods for controlling semiconductor processes. These developments have included new algorithms such as neural network controllers,32–34 adaptive controllers,35,36 and other techniques. These works have contributed significantly to

* See, for example, Chapter 18.
** See, for example, Chapter 11.
*** See, for example, Chapter 6.
the number of different tools available for addressing the wide range of situations
present in the manufacturing environment. Many aspects of algorithm advancement
are covered in Part 2 (Chapters 3 through 6), and in portions of Parts 5 and 6
(specifically Chapters 17 to 20) of this book.

1.3.4 NEW METROLOGY

A significant effort in the early 1990s was dedicated to developing in situ sensors
and on-line metrology. This work has focused largely on the development of in situ
sensors for real-time endpoint detection and process control, as well as on-line
metrology for automated R2R process control. As a result, a large number of in situ
sensors have appeared for use in plasma etching, including temperature sensors,37,38
full-wafer interferometry (FWI),39 optical emission spectroscopy (OES),40 RF mon-
itors,41 scatterometry,42 and others. These sensors have been used in many applica-
tions, including endpoint detection on etch processes43 and R2R control of plasma
etching.44 Sensors have also been developed for chemical mechanical polishing
(CMP), including motor current sensing, interferometric sensing through a window
in the pad,45 thermal imaging of the pad,46 and other techniques.

Several on-line metrology tools have also been developed. These systems offer
significant throughput advantages with or without R2R process control. In particular,
on-line epitaxial thickness measurement tools have been developed and used for
R2R process control.47 An on-line metrology tool for chemical mechanical polishing
has been developed and used for R2R process control using several different tech-
niques.12,48,49 It has been shown that the cost of ownership (COO) for the CMP
process can be greatly reduced using this tool.12,48,49 It is likely that many more
metrology tools will be developed in the near future, and will provide the opportunity
for further advancements in process control and wafer processing.50,51 R2R control
could benefit from many forms of metrology, in-line (integrated) and off-line, but
also in situ, as R2R control solutions are developed to utilize in situ measurements.*

1.3.5 ACCEPTANCE OF RUN-TO-RUN CONTROL

With the development of enabling technologies and framework specifications, com-
mercial control software, new algorithms, and new sensors and metrology, advanced
process control, and specifically R2R control have matured significantly. R2R control
is now widely accepted as a means for production fabrication facilities to improve
processing quality, increase throughput, and decrease cost. The many works dem-
onstrating improvements gained by the use of R2R control over a wide range of
semiconductor processes** have resulted in advanced process control being recog-
nized as a key factor to achieving the performance and reliability requirements that
future semiconductor manufacturing will require. With this in mind, the future of
APC, and specifically R2R control in the semiconductor industry, will now be
discussed.

* See Chapter 18.
** See, for example, Chapters 11, 13, 16, 17, 18, and 19.
1.4 THE FUTURE OF ADVANCED PROCESS CONTROL

It has been suggested that feature size decreases and wafer size increases will not continue at a pace necessary to produce a twofold improvement in performance-per-dollar every 18 to 24 months.\textsuperscript{52} If this is true, then the twofold improvement in performance-per-dollar must be achieved in other areas. Thus, the opportunity for increased tool operating efficiency must be pursued using on-line metrology, \textit{in situ} sensors, process control, and endpoint detection to significantly contribute to cost and performance improvements.\textsuperscript{12,53} In this section some of the possible benefits that advanced process control could soon provide are discussed, an R2R control solution example is presented to illustrate some of these benefits, and suggestions are given for areas where future research and development is needed to maximize the impact of APC on the semiconductor industry.

1.4.1 BENEFITS OF ADVANCED PROCESS CONTROL

1.4.1.1 Benefit: Increased Throughput

For many years it has been suggested that automation with advanced process control will provide significant improvements in throughput, resulting in reductions in cost. The increased throughput will likely be derived from the combination of on-line metrology speed-ups (e.g., measurement and wafer processing occurring in parallel) and the elimination of operator wait times (e.g., performing and inspecting metrology results manually). In some applications there are expected to be added benefits such as the removal of processing steps. For example, Figure 1.9 illustrates that a significant increase in throughput and a significant decrease in the number of clean steps (i.e., cleaning after polishing and before reworking in CMP) can be achieved through the addition of sensors and process control.\textsuperscript{48} Since OEE is largely governed by idle equipment, downtime, and setup time, lessening of inefficient operation or “wait” time could address about 15\% of the loss in OEE.\textsuperscript{12,53}

R2R control could also extend the period between some maintenance operations, especially those associated with replacing consumable sets. Further, control algorithms customized to the process could tune the process immediately after a consumable set change, thereby reducing tool reconfiguration and increasing OEE.* Note also that additional gains in this area will likely come from multiple-process-step control solutions.**

While R2R process control may increase efficiency, the impact that increases in setup and maintenance times could have on OEE should be carefully considered. Setup and maintenance times often offset or overshadow the throughput increases due to improved operational efficiency. Because the added parameters of more complex controllers can make tuning and debugging difficult and time-consuming, minimizing the complexity of a controller is critical to reducing the added setup and maintenance times and maximizing the benefits. As a result, controllers that have

\* See, for example, Chapter 18.
\** See, for example, Chapter 15.
followed the “keep it simple” approach have achieved the largest success.\textsuperscript{11,52} The majority of commercial products are thus based on relatively simple integral controllers,\textsuperscript{12,19,23} and have focused on resolving a wide range of practical implementation issues. This trend will likely continue until more complex controllers demonstrate significant improvements in critical areas.

\textbf{FIGURE 1.9} Two CMP processes: one with a high-quality, low-throughput process using \textit{ex situ} metrology, and one with a higher quality, high-throughput process using on-line metrology and R2R control.
1.4.1.2 Benefit: Reduced Non-Product Wafers

An issue that has always plagued manufacturers attempting to improve process cost of ownership and throughput while maintaining yield is the lost processing time and cost associated with processing non-product (NP) wafers. These wafers are usually required as part of the qualification of a process, e.g., after a process maintenance (PM) event where a consumable set is replaced. For example, a CMP pad replacement is often followed by a qualification period where NP wafers are used to “break in” the pad. SPC techniques can be used to verify that the process is qualified and the likelihood of misprocessing has been reduced to a level where processing of product may resume. However, these techniques can be very inefficient, resulting in lost OEE due to the extra time required to process NP wafers, and lost tool COO due to the cost to the NP wafers and any pad life lost due to processing NP wafers.

Advanced process control can reduce the requirement for NP wafers by more quickly qualifying the process. This capability is largely a function of the control algorithm used and, more specifically, the capability of the control algorithm to model the process shift associated with the PM event and any transient process behavior that directly follows this event. This topic is explored further in Chapter 18.

1.4.1.3 Benefit: Improved Wafer-to-Wafer and Lot-to-Lot Variability

It has been suggested that process control will greatly improve wafer-to-wafer or lot-to-lot processing quality. It is likely that this area is where the greatest benefits of APC will be derived, and will contribute to significant improvements in manufacturing processes. A number of works suggest that APC techniques reduce process variability, and more specifically, process capability.\textsuperscript{3–5,7–10,12,15,19,23,29,47–49} Process capability is a function of process variability and process accuracy, i.e., closeness to target.\textsuperscript{*} APC improves process accuracy utilizing pre- and postprocess measurement, modeling of process and equipment trends, and suggesting subsequent process parameter adjustments (i.e., feedback control). Process variability is generally reduced through pre-process measurement and subsequent process parameter adjustment (i.e., feedforward control). Thus both pre- and postprocess metrology play a role in improving process accuracy. Of the two, the impact of premetrology is often underestimated, even by control engineers. Indeed, in scenarios where product variability dominates over equipment or process variability and trends, utilization of premetrology has been reported to account for up to 80\% of the improvement in Cpk achieved with APC.\textsuperscript{54}

The improved process capability can also result in reductions in process margins. These reduced process margins could create the opportunity for new design rules that lead to performance improvements. For example, tighter specifications on the post-CMP within-die variability could lead to possible reoptimization of the following lithography steps. These reoptimizations could lead to improvements in line width variations, which in turn could allow tighter tolerances on device sizes. The result would be higher performance-per-dollar devices. Improved process margins

\textsuperscript{*} Process capability is defined in Chapter 11.
could also result in cost and waste reduction, as shown in Figure 1.10. In the CMP process, the amount of deposited oxide for the inter-level dielectric (ILD) thickness is equal to the amount of oxide that will be removed in order to achieve planarization, plus the amount of variability of the final oxide thickness due to wafer-to-wafer and lot-to-lot variability. Improving the control of the final film thickness reduces this variability, and hence reduces the required amount of deposited oxide. In doing so, this reduces the amount of processing in the CMP step, as well as the deposition step. In both cases the result is reduced waste, power consumption, chemical usage, and, ultimately, cost, as well as increased throughput.

1.4.1.4 Benefit: Reduced Within-Wafer and Within-Die Variability

One of the most important improvements that R2R process control could address is that of within-wafer and within-die variability. The difficult task of controlling within-wafer and within-die nonuniformities will become particularly important as wafer sizes move toward 300 mm, where wafer-level uniformity is expected to be a major problem. This wafer-level nonuniformity will compound any within-die nonuniformity superimposed on the wafer-level nonuniformity. In many applications, such as CMP, this within-die variability is generally quite large, often several times larger than the across-wafer nonuniformities. Unfortunately, little work has been done to incorporate these factors into R2R control algorithms. Doing so often requires a significant understanding of the process being controlled, and thus control engineers are unable to address these more complex factors when controlling a process. It is also often the case that process engineers are unwilling to allow dynamic changes in the process recipe in order to achieve improvements in these effects. This resistance is generally due to process audit requirements, or a fear of scrapping production wafers because of unseen problems in important wafer properties that are not being directly controlled but are the result of changes in the process. Another reason few controllers have addressed the within-wafer and within-die nonuniformities is that there is often a need for more accurate process models in order to control a process. The rapid pace of the semiconductor industry results in process models being far behind current technology. These problems result in a retreat to...
simply adjusting process time, or small process parameters that affect only the lot-to-lot or wafer-to-wafer variability.

1.4.1.5 Benefit: Reduced Operating Errors

Finally, it is suggested that R2R process control will reduce misprocessing and other operating errors. This reduction in operating errors could be achieved via automated metrology upload and automatic recipe generation and download. This automation is facilitated by utilizing appropriate control-enabling technologies and adhering to a framework for integrated control, as described in Section 1.3.1. Reducing operator errors through adoption of automated process control could drive a large improvement in processing quality, particularly in the fab ramp-up period, where frequent changes and tweaking are made to processes. The reduction in processing error during this period is critical to the overall cost effectiveness of a new fab and its ultimate profitability. This area is one that has received a lot of attention but is difficult to demonstrate. Much more work is needed in this area in order to help solidify the benefits of using APC in complex multistep fabrication processes. With the maturation of R2R control and APC systems in general, the industry in a much better position to begin further study of these benefits.

1.4.2 Example of APC Benefits: R2R Control of a CMP Process

The CMP process has been a prime early beneficiary of R2R control and serves as an excellent example to illustrate the many benefits of R2R process control. A schematic of a typical CMP R2R control solution is provided in the Introduction section of this book. In general, current CMP R2R control solutions monitor post- and possibly preprocess wafer thickness and center-to-edge uniformity, and modify process parameters such as time, backpressure, carrier speed, and platen speed, in a run-by-run fashion based on advice from the control solution. A typical example of an uncontrolled CMP process is included in Figure 1.11 (Callout #1). Note that the process is characterized by the following features: (1) a fairly continuous degradation in pad removal rate over its life, (2) fairly regular pad replacement events triggered either by the execution of a predetermined number of process runs or — as in this example — the occurrence of an SPC alarm on process remaining thickness, (3) a pronounced “spike” in removal rate directly after a pad replacement event, and (4) a pad break-in phase directly after pad replacement where the pad is qualified either by utilizing a predetermined number of NP wafers or by utilizing NP wafers until the process is within acceptable limits.

The individual benefits of applying R2R control — and specifically tool and process specific R2R control — are also illustrated in Figure 1.11 (Callouts #2 through #5). The first important improvement noted in the controlled solution is that the process drift due to pad wear is eliminated during the useful life of the pad (Callout #2). This, in turn, results in reduced process variability and improved

* See, for example, Chapter 11.
process capability. The second related result is that pad life is extended through adjustment of the process parameters to keep the product measured quality parameters within SPC limits (Callout #3). The benefits of extending pad life include not only reduced cost of consumables, but also increased OEE due to the reduced frequency of PM events to replace the pad. Note that these control benefits, illustrated in Callouts #2 and #3 in Figure 1.11, can generally be achieved without tailoring the control solution to the PM pad replacement event.

Another significant improvement noted with the controlled solution is the reduced requirement for test wafers (Callout #4). This portion of Figure 1.11 illustrates that, with the controlled solution, the magnitude of the “spike” in removal rate, associated with a new pad, is reduced. Further, the tool is brought to a process-capable state more quickly, resulting in the use of fewer NP wafers and increased OEE as a result of reduced pad qualification time. Note that this class of control benefits can generally be achieved only if the controller includes a model of the pad change event. In the control solution depicted in Figure 1.11 the controller contains a model of the expected removal rate shift associated with a pad change. An enhancement to this control solution is described in Chapter 18, where the controller models not only the process shift associated with this PM event, but also the subsequent process dynamics usually associated with pad break-in.
It is important to note that this example, while specific to CMP, could easily be extrapolated to other processes. For example, a chemical vapor deposition (CVD) process can be characterized by drift associated with source depletion, and shift associated with a source replacement PM event. An etch process can be characterized by drift associated with chamber wall seasoning (polymer buildup), and shift associated with a chamber clean PM event. Thus, if the appropriate metrology capability is available to measure process quality, and if process-specific control models can be developed, these processes can also experience the same class of benefits demonstrated for the CMP process.

### 1.4.3 Areas for Future Research and Development

Many of the R2R algorithm solutions currently being successfully applied to R2R process control in semiconductor manufacturing are described in detail in Part 2 of this book. In the remainder of this section, we will describe three possible avenues for future R2R control algorithm advancement and discuss each possibility in terms of the benefits outlined above, the effort required to achieve these benefits, and the likely success of achieving them.

#### 1.4.3.1 Complex, Adaptive, Self-Tuning Controllers

There has been a fair amount of research activity into process control strategies with complex tuning algorithms. These works address issues such as the self-tuning of parameters, optimality of controller parameters, rapid correcting algorithms, robust process control, and other issues. These works often follow the work done in the controls field and provide a natural progression for work in APC. The controls field has dealt with many of these problems over the last few decades, and this provides a nice vehicle for applying control algorithm developments and proofs of stability and optimality to existing processes. It is often suggested that these methods should be incorporated as soon as possible, and this would create the greatest increase in process control performance in the semiconductor industry. In light of this, we now consider the possible benefits of this approach, the impact these benefits would have, and the likelihood of achieving them.

Consider the question of stability of a semiconductor process using R2R control. Currently, most processes to which R2R control is being applied are generally “well-behaved” in the sense that relatively simple model-based control solutions — such as dynamic linear approximation solutions — can be effective over the process range specified for a tool. For example, typical CMP control models approximate monotonic, though multivariate, relationships between inputs and outputs. Further, the drifts in these processes are gradual, while shifts can generally be detected using SPC techniques. Thus, using a combination of R2R control and limits monitoring, R2R control solutions have been demonstrated to remain stable in typical process scenarios. Therefore, stability of R2R control is not seen as a major issue in semiconductor processing.

Reliability, on the other hand, is critically important to the operating time, efficiency, cost, and profitability of a piece of equipment in a semiconductor facility.
This is another reason why controllers that have obeyed the “keep it simple” rule mentioned earlier have had large success. Controllers of this type generally provide good control because the drift rates of most semiconductor processes are small relative to the amount of noise in the process. The simplicity of these controllers allows them to be easily understood by the many individuals working with the processes, while at the same time allowing the control engineers to focus on the many manufacturing issues (e.g., nonperiodicity of the measurements) that are critical for reliable operation. The maturation of the APC infrastructure is likely to improve on many of these issues, but these manufacturing issues will nevertheless play an important part in the selection of any control system. In addition, most increases in complexity focus on improving the performance of the controller. This can sometimes result in a less user-friendly, less reliable system with a minimal improvement in quality. However, addressing robust control for use in the semiconductor industry may be an important exception to this rule of thumb, and could result in significant improvements in both the reliability and quality of APC systems.

Optimality is also of noted importance in semiconductor process control. Many control problems result from poor process models (either too simple or too complex) and improperly tuned controllers. Many classical control techniques are based on dynamic system modeling, and variations of these control approaches focus on improving the control of wafer-to-wafer and lot-to-lot variability by tuning the parameters of the controller.

However, addressing optimization of the parameters of a controller, while important, is not always of primary concern in many APC applications in semiconductor manufacturing. This is because such “optimized” APC solutions are usually aimed at addressing wafer-to-wafer and lot-to-lot process variability; such sources of variation are, however, not always the largest to be considered. For example, in plasma etching a large amount of etch rate variability is often observed. This variability may be caused by device dependencies in the process that are the result of the amount of open area on the different devices. So, in actuality, the variation is the result of a lack of effective control or understanding relating to the device dependencies in the process. The same may easily be true in other processes, such as CMP.

Consider a CMP process, and the within-die plot of 25 (postpolish) site thicknesses shown in Figure 1.12. Note that the range of these measurements is on the order of 3000 Å. If we now consider a single site within the die plotted for several dies across the wafer, as shown in Figure 1.13, we see that the (postpolish) within-wafer variation range is only on the order of 500 Å. Further, consider the plot of the (postpolish) mean thickness of a single device that was processed using a simple EWMA controller shown in Figure 1.14. Here the wafer-to-wafer and lot-to-lot variation range, which excludes effects due to the device, is also on the order of 500 Å.

The causes of the within-die variation, methods to measure and model these variations, and mechanisms for assessing the impact of these variations on processing are often unknown to process control engineers. This problem is often due to a lack of a detailed understanding of the semiconductor process at hand, and the lack of process visibility that can be provided by in situ sensors and metrology. Evidence,
such as that cited above, suggests that focusing on improving our understanding of the processes, how to measure the sources of variation, and how to build fundamental process models for use in control would have a much larger impact on controlling
semiconductor processes than would efforts focused on improving control parameter optimization techniques for current algorithms — or incorporating more complex algorithms to control problems of wafer-to-wafer or lot-to-lot variability.

### 1.4.3.2 Tool- and Process-Specific Models and Controllers

A major problem for many semiconductor processes lies in the control of device dependencies. The die-level signature profiles for different devices often look completely different. This results in metrics such as the post-process mean thickness being very different for different devices, and makes controlling the processing of these different devices extremely hard. Current controllers only approximate these effects by utilizing different models for different devices, or by using correction factors. Using multiple models makes it difficult to track tool changes in the models of devices other than the one being run. Therefore, running one product for a long period before switching to another product can cause a serious degradation in the quality of the second product. The device correction factors used to correct this are an improvement, but inaccuracies in their values can result in high variability or can require complex methods for updating model parameters such as deposition rate. In addition, the device correction factors are often specific to the exact locations of the measurements taken on the devices. While the monitored locations may be controlled correctly, there is little indication of what is happening in the rest of the device. Additionally, any change in the measurement plan can cause shifts or other undetected errors in the resulting product wafers.

**FIGURE 1.14** The wafer-to-wafer and lot-to-lot variation of the postpolish oxide thickness average of measurements taken at one location on 22 dies, plotted over four lots of 24 wafers.
Other problems with process lie in qualifying and stabilizing the process after a PM event such as replacing a consumable set. As an example, a polish pad replacement event on a CMP tool is almost always followed by a pad qualification phase. The nonproductive tool time associated with this pad “break-in” event reduces throughput and OEE and requires the use of NP wafers. A control solution that does not model the pad replacement transient signature does not address this problem, and in some cases can actually increase the process shift associated with the PM event.

Significant opportunities exist to correct these problems by incorporating process-specific models into existing control schemes. With respect to the device dependency problem outlined above, models that provide approximations of the fundamental response of the process at hand, particularly with respect to device dependencies, could greatly increase a controller’s knowledge about the controlled devices. Correlating measured values with those expected from the process-specific model would allow prediction and virtual monitoring of unmeasured regions of the controlled wafers. Such fundamental models, while not necessarily precise, would allow the user to greatly expand the degree at which different devices are monitored and controlled. With respect to the post-PM process qualification problem outlined in the previous paragraph, control solutions that model process and equipment shifts and subsequent transients associated with common PM events can reduce the time and NP wafer requirements for tool requalification, thereby increasing throughput and OEE and improving COO.*

1.4.3.3 Multistep or Full-Flow Process Controllers

Finally, it is often the case that process control efforts are local; controlling parameters of one process on one tool neglects the larger picture of the manufacturing process as a whole. The main reason for this process-centric control effort is that process and control knowledge is also largely process-centric. Relatively little work has been done to combine issues from multiple steps in the semiconductor manufacturing process into a process control effort aimed at solving problems related to a series of steps. Multiple-step process control has the potential to have a large impact on the industry, and very recently efforts have begun to focus on evaluating the viability and effectiveness of multistep process control. For example, in Chapter 15 a method is described for precompensation of CMP uniformity target based on downstream etch uniformity data. In Chapter 20, techniques and results for multiple-step feedforward control between lithography and etch processes are reviewed, and a generic, multiple-step control-enabling technology is described. These efforts represent the first steps in an area of control where the possible benefits could be particularly rewarding. Solutions at this level would allow for focus on a total factory control solution rather than process-centric, independent control solutions. In addition, many integration issues are centered around device dependencies in processing. As noted above, device dependencies arise in many processing steps and interact between multiple processing steps. It is these interdependencies between multiple process steps and device layouts that lead to significant manufacturability,

* See, for example, Chapter 18.
reliability, and performance losses. Combining the process-centric fundamental models mentioned above with multiple-step control methods is yet another area for significant improvement in control.

A particular example of such a device dependency is that of postpolish, within-die variation in CMP. This variation has a direct impact on the depth of focus (DOF) of the following lithography steps. This DOF has a large impact on the variability of the interconnect lines, and hence the performance of the resulting device. Post-CMP measurements of the variation of the focusing mechanism in lithography tools are often taken. Unfortunately, there is often little or no communication of these results back to the CMP area. There exists an opportunity for future control algorithms to take advantage of the APC framework in order to utilize these types of process interactions to significantly improve the overall semiconductor manufacturing process. Addressing control from a total factory solution standpoint will allow for the incorporation of these interdependencies into the control strategy, and better correlation of the controller assessment and tuning to the factory yield (rather than individual process parameters).

1.5 SUMMARY

In this chapter we have reviewed the progression of control in the semiconductor industry from statistical process control to run-to-run process control, which was driven by the reduction of random tool processing variability and the increase in steady long-term process drifts. We discussed how this trend toward R2R control uncovered many new issues. These new issues were largely manufacturing related, and several key barriers to process control were addressed. The resulting advancements included: (1) the development of enabling technologies and a framework around which process control could be developed, (2) the development of several commercial applications for run-to-run control, (3) the introduction of many new sensors aimed at real-time and run-to-run process control, and (4) several advances in process control algorithms. Finally, we addressed the future of APC by first discussing its many benefits, including increased throughput and OEE, reduced NP wafers, improved process variability and capability, and reduced operator error. We then summarized the future trends in process control for the semiconductor industry. Specifically, we suggested that there will be a shift away from complicated tuning algorithms with simple process models, toward more process-focused control approaches with fundamental process models. Further, we outlined how multiple-step process control methods could greatly increase the processing quality over several complicated or difficult-to-control steps. We have suggested that these approaches provide promise in dealing with the difficult task of controlling within-wafer and within-die variation in the face of device-dependent processing.

The material presented in this chapter provides motivation for exploring (1) the capabilities and limitations of current R2R control algorithm approaches being applied in the industry (Part 2); (2) enabling technologies and frameworks for cost-effective deployment, integration, and reuse (Part 3); (3) examples of solution approaches and results (Parts 3 through 5); and (4) areas for advancement (Part 6).
With this motivation in hand we hope we have established a foundation for the understanding of the material in the remainder of this book.

ACKNOWLEDGMENTS

We would like to thank the NSF/SRC Engineering Research Center for Environmentally Benign Semiconductor Manufacturing for supporting this work.

REFERENCES

2 Process Control and Optimization Methods for Run-to-Run Application

Enrique Del Castillo and Arnon M. Hurwitz

2.1 INTRODUCTION

In a typical semiconductor manufacturing process, within-run (or batch) variation is usually controlled by automatic controllers built into the equipment. Batch sizes may be as small as one wafer in some processes. A run-to-run (R2R) controller is necessary since specifications can change from batch to batch, the equipment may experience aging or wearing-out phenomena, maintenance operations can change the operating conditions of the process, or process disturbances may enter the system suddenly. This implies that equipment controllers cannot be kept operating at a fixed recipe. Thus, an R2R controller is needed to act as a supervisor, indicating whether a recipe change is needed and suggesting a new recipe for use in the next batch.

Frequently, R2R controllers are model-based. Linear regression techniques are used to estimate initial equipment models from experimental data. Here, the process is seen as a “black box,” and models derived in this way are empirical in nature, i.e., the models are not based on first physics/chemistry principles. We hasten to add that this is not always the case in R2R control, as some first-principle state-space controllers have been developed and implemented successfully. However, due to the rapid technological change, empirical statistical models are widely used in relatively new, poorly understood processes.

The development phases of a black box R2R control system are depicted in Figure 2.1. During qualification of a new process or equipment, experimental and statistical techniques are used for obtaining initial models and an initial optimal recipe. Then, the R2R controller tunes or adjusts these initial models and recipe, trying to keep the process at the optimized level (which becomes the target value in case targets were not previously available). Due to their simplicity and robustness, R2R controllers based on the exponential weighted moving average (EWMA) statistic are probably the most frequently used in industry.

The goal of this chapter is to provide a technical review of R2R control and optimization methods. An overview of R2R optimization techniques is first provided in Section 2.2. The stability and robustness of EWMA-based controllers are discussed
in detail in Section 2.3, where methods for tuning these controllers are also described. Other important R2R control techniques are briefly reviewed in Section 2.4. A review of the literature up to 1995 was provided by Del Castillo and Hurwitz. One of the goals of this chapter is to give a more up-to-date review of available methods.

2.2 R2R PROCESS OPTIMIZATION TECHNIQUES

For the qualification of a process and the development of mathematical models that relate the process inputs to the corresponding outputs (responses), several techniques have been reported in the literature, ranging from statistical methods such as design of experiments (DOE) and response surface methods (RSM) to fuzzy logic models to the use of artificial neural network.

DOE and RSM are widely used in industry for process characterization and optimization, not only in semiconductor manufacturing. Given the relevance of these methods, Chapter 14 provides an introduction to RSM techniques that will be useful to process engineers working in semiconductor manufacturing. In this chapter we discuss in detail the use of the Ultramax method, a commercial software developed by Ultramax Corp.

Sachs et al illustrate the use of this software in semiconductor R2R processes. Ultramax has been described as a Bayesian formulation of a ridge regression estimator that works also in the rank-deficient case (i.e., the algorithm can be launched even without observations). The appendix of this chapter provides a detailed description of the inner works of the Ultramax algorithm.

An advantage that Ultramax displays over the usual “tweak and maintain” SPC algorithms is its ability to “search” and characterize a feasibility region while seeking an optimum; the run-to-run (sequential) optimization is accomplished by a “backwards and forwards” motion around the primary optimization region or point. The utilization of this search facility prevents the algorithm from becoming trapped in
a relative maximum or minimum, and allows Ultramax to navigate through the relative optima.

Ultramax is multiple input, multiple output (MIMO), and thus can support a task such as the following application to a reactive ion etch (RIE) plasma system. The first problem formulation was “optimize selectivity subject to polysilicon etch rate greater or equal to 250 Å/min.” An initial “optimum” recipe was derived via classic design of experiments (DOE). The problem formulation was then entered into Ultramax (in similar fashion to a DOE setup) and the sequential optimizer was allowed to optimize the process on a run-to-run basis. The experimental results in Figure 2.2 show that etch selectivity was improved by almost a factor of two over a span of 20 runs, while the minimum etch constraint was not violated.

A second experiment (Figure 2.3) shows the effect of a deliberate system perturbation at run number 20. Here we see that the initial degradation in selectivity in the few runs following the perturbation is followed by a period of system recovery, where the selectivity is both improved and stabilized.

In experiments involving a planarization processes it was confirmed that Ultramax will act as a good R2R system optimizer for locating an optimum, or for recovering from a system perturbation. Ultramax works remarkably well for a system that requires quite a bit of “driving” toward a better operating point, and will function robustly in a production environment through its recognition of input constraints as well as output target ranges (specifications).

## 2.3 EWMA-BASED R2R CONTROLLERS

The EWMA statistic has been used for a long time for quality monitoring purposes. Its use as the basis of a recipe adjustment technique is relatively more recent. The

![Figure 2.2](image-url)
work of Box and Jenkins\textsuperscript{4} on minimum variance process control under the assumption of IMA(1,1) noise is probably the first reference to this type of statistic for process adjustment purposes. More recently, Sachs and his co-workers\textsuperscript{33} introduced an R2R controller for semiconductor manufacturing purposes based on a single EWMA statistic. This controller was termed a “gradual mode” controller to distinguish it from a “rapid mode” controller (not based on the EWMA) that recommends more drastic recipe changes if there is statistical evidence the process is badly out of control.

For a time series of measurements \(\{X_t, X_{t-1}, \ldots\}\) where \(t\) denotes the run number, the EWMA is given by

\[
\text{EWMA}_t = \lambda X_t + (1-\lambda)\text{EWMA}_{t-1}
\]

where the weight parameter \(\lambda\) (\(0 < \lambda \leq 1\)) and some initial condition for the recursion \(1\) have to be chosen. Evidently, the weight given to a measurement that occurred \(j\) runs ago is

\[\lambda(1-\lambda)^j.\]

Therefore, it can be seen that the weights decrease geometrically (exponentially if time is continuous) with the age of data. For a process that exhibits dynamic changes in its model structure, such “discounting” of older data can be very useful. Intuitively, in such a system, giving more weight to more recent measurements makes sense for run-to-run control.

In this section, stability, robustness, and tuning aspects of EWMA-based R2R controllers are presented. Our presentation is limited to the gradual-mode MIT controller (based on a single EWMA equation) and the predictor–corrector controller (based on two EWMA equations).

\[
\text{FIGURE 2.3}
\]

Selectivity

System Perturbation

Run number

\[0 \quad 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \quad 35\]

\[5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11\]
2.3.1 **Single EWMA-Based R2R Controllers**

Rather than deriving an optimal control law based on some optimality criteria (e.g., minimize output variance), the structure of EWMA-based controllers is first specified and then justified by looking at how robust these controllers are with respect to structural or parametric changes in the assumed models.

The single EWMA-based controller assumes the simple linear regression model

\[ Y_t = \alpha + \beta U_{t-1} + \varepsilon_t \]  

(2)

where the \( \{\varepsilon_t\} \) constitute a white noise sequence (i.e., they have zero mean, constant variance, and are not autocorrelated), \( U_t \) denotes the controllable factor setting at the end of run \( t \) (beginning of run \( t+1 \)), and \( Y_t \) denotes the measured response at time \( t \). The parameter \( \alpha \) is called the offset or bias and represents the mean value of the response when the controllable factor equals zero (frequently, \( U_t \) is scaled into the \((-1, 1)\) range according to standard DOE coding conventions). The parameter \( \beta \) is called the process gain and it is usually estimated off-line using DOE and regression techniques conducted during the qualification of the process. We will use the notation \( b = \hat{\beta} \) to denote the gain estimate.

If model (2) were known to be the true system description and \( T \) denotes the target response value, then the minimum mean square error (MMSE) control strategy is simply to fix the controllable factor at the value

\[ U_t = \frac{T - \alpha}{b} \]  

(3)

which constitutes a feedforward controller. In practice, model (2) is not an accurate description of the process due to disturbances. For this reason, Sachs and his co-workers\(^{33}\) propose to reestimate the parameter \( \alpha \) recursively from run to run using the EWMA expression

\[ \hat{\alpha}_t = a_t = \lambda(Y_t - bU_{t-1}) + (1-\lambda)a_{t-1} \quad (0 < \lambda \leq 1) \]  

(4)

and the feedback control law is

\[ U_t = \frac{T - a_t}{b}. \]  

(5)

Equations (4) and (5) are referred to as “bias controllers.”

EWMA-based controllers are simple instances of internal model controllers.\(^{15}\) In an internal model controller (Figure 2.4), the differences between the prediction of a model and the actual process response value are fed back to the controller via a filter that attenuates large model errors and adds robustness against unmodeled disturbances or unmodeled process dynamics. The controller is frequently the inverse
of the model. For a single EWMA-based controller, the “model” is simply the estimate of the gain, the process is simply given by the true gain $\beta$, the EWMA filter given by Eq. (4) is used on the feedback loop, and the controller takes the difference $T - a_t$ as input, multiplies this times the inverse of the model, and generates the control action given by Eq. (5).

2.3.1.1 Stability and Robustness Properties of Single EWMA-Based Controllers

Ingolfson and Sachs investigated the gradual mode controller for the case when the process obeys a slightly more complicated model:

$$Y_t = \alpha + \beta U_{t-1} + \delta t + \epsilon_t$$

where $\delta t$ is a deterministic trend component. If Eq. (6) is the true process description, these authors show that the EWMA-controlled system will be asymptotically stable if and only if

$$|1 - \lambda \xi| < 1$$

where $\xi = \beta/b$ is a measure of the quality of the gain estimate. Furthermore, assuming condition (7) holds, they show that the asymptotic mean square deviation (AMSD) from target is given by

$$\text{AMSD}_{dr}(\lambda) \equiv \lim_{t \to \infty} E \left[ (Y_t - T)^2 \right] = \frac{2 \sigma^2}{2 - \lambda \xi} + \frac{\delta^2}{\xi^2 \lambda^2}$$

where $\text{Var}(\epsilon_t) = \sigma^2$. Ingolfson and Sachs show that the gradual mode controller is a discrete integral controller of the form
with $K_0 = -\lambda/b$ and $K_I = T/b$. These controllers are very well known and compensate for process shifts and offsets. The robustness of this type of controller has been emphasized in a recent book by Box and Luceño.\textsuperscript{6}

An important question for EWMA-based controllers is how to select the $\lambda$ weight in Eq. (4), i.e., how to tune the controller. Smith and Boning\textsuperscript{34} show that the weight that minimizes the AMSD (given by Eq. (8)) is given by the real root of

$$\sigma^2 \xi^3 \lambda^3 - \delta^2 \xi^2 \lambda^2 + 4 \delta^2 \xi \lambda - 4 \delta^2 = 0 \quad (9)$$

that satisfies the stability condition. To solve Eq. (9), the probably unknown process parameters $\sigma$, $\xi$ and $\delta$ are needed, and this limits the practical application of this result. In general, most authors agree that the larger the relative drift $\delta/\sigma$, the closer $\lambda$ should be to one.

It is interesting to investigate what would happen if the single EWMA controller is applied to a process in a perfect state of “statistical control,” namely, a process such that

$$Y_t = \alpha + \epsilon_t$$

if $U_t$ is always zero. In a recent paper, Del Castillo\textsuperscript{14} shows that the inflation in AMSD over the minimum possible variance for $Y_t$ (given by $\sigma^2$) is given by

$$\frac{\text{AMSD}_{\text{DT}}(\lambda)}{\sigma^2} = \frac{2}{2 - \lambda \xi} \cdot (10)$$

which extends similar results by Box and Luceño,\textsuperscript{6} who analyze the case $\xi = 1$, i.e., the known gain case. This gives evidence in favor of using small values of $\lambda$ (around 0.1). In this way we have the assurance the controller will compensate against process shifts and drift for a small price to pay (in terms of inflation of AMSD), in the order of 10%, even if we grossly misidentify the process gain by a factor of two. Note that even if $\xi \neq 1$, the EWMA controller still contains “integral action,” and this will compensate for offsets or shifts. A good part of the “robustness” of these controllers (and also of PI controllers) comes from the integral action.

### 2.3.1.2 Other Process Disturbances

The performance of the single EWMA applied to disturbances other than a deterministic trend has been studied recently.\textsuperscript{14} For the case where the process model is

$$Y_t = \alpha + \beta U_{t-1} + N_t \quad (11)$$
with

\[ N_t = N_{t-1} + \delta + \epsilon_t \]  

i.e., a random walk with drift disturbance, it is shown that the stability condition is the same as before, namely, condition (7). Furthermore,

\[ \text{AMSD}_{RWD}(\lambda) = \frac{\sigma^2}{\xi\lambda(2 - \xi\lambda)} + \frac{\delta^2}{\xi^2\lambda^2} \]  

is shown to be minimized by using an EWMA weight of

\[ \lambda_{RWD} = \frac{4\delta^2 - \sigma^2 - \sigma\sqrt{8\delta^2 + \sigma^2}}{2(\delta^2 - \sigma^2)^2} \]  

a weight that always satisfies the asymptotic stability condition. If, on the contrary, the disturbance follows an IMA(1,1) process,\(^5\)

\[ N_t = N_{t-1} - \theta\epsilon_{t-1} + \epsilon_t \]

where \(\theta\) is a moving average parameter, then it is shown that

\[ \text{AMSD}_{IMA}(\lambda) = \frac{1 + \theta^2 - 2(1 - \xi\lambda)\theta}{\xi\lambda_i(2 - \xi\lambda_i)} \sigma^2 \]

which is minimized by setting

\[ \lambda_{IMA} = \frac{1 - \theta}{\xi}. \]

Note that if \(\xi = 1\), then \(\lambda_{IMA} = 1 - \theta\) provides the minimum possible AMSD, as shown by Box et al.\(^5\)

### 2.3.2 Double EWMA-Based R2R Controllers

Butler and Stefani\(^8\) notice that for a severe drift per time unit, the offset of the controlled response, as given by the second RHS term in either (8) or (13), can be too large even when large weights \(\lambda\) are used. To avoid such undesirable situation, they propose to introduce a second EWMA equation in the controller,

\[ D_t = \lambda_2(Y_t - b u_{t-1} - a_{t-1}) + (1 - \lambda_2)D_{t-1} \quad (0 \leq \lambda_2 \leq 1) \]
and the controller is then given by

\[ U_t = \frac{T - a_t - D_t}{b} \]  

where \( a_t \) is as in Eq. (4). Double EWMA-based controllers (also called “predictor–corrector” controllers) are internal model controllers where a double EWMA filter is used on the feedback loop. Del Castillo\textsuperscript{13,14} has recently shown that if the disturbance is any of the following,

- Deterministic trend
- Random walk with drift
- IMA(1,1)

the system controlled with a double EWMA will be asymptotically stable if and only if the following two conditions are satisfied:

\[ \left| 1 - 0.5 \xi (\lambda_1 + \lambda_2) + 0.5 z \right| < 1 \]  

\[ \left| 1 - 0.5 \xi (\lambda_1 + \lambda_2) - 0.5 z \right| < 1 \]

where \( z = \sqrt{\xi^2 (\lambda_1 + \lambda_2)^2 - 4 \lambda_1 \lambda_2 \xi} \). For the listed disturbances, if the process is stable, then, asymptotically, it will be on target on the average, thus the AMSD equals the asymptotic variance. Expressions for \( \text{AVAR} \) are very complicated.\textsuperscript{13} However, if it can be assumed the process gain is known (i.e., \( \xi = 1 \)), the asymptotic variance for a stable system under a deterministic trend disturbance is

\[ \text{AVAR}(Y) = \frac{\sigma^2}{(\lambda_1 - \lambda_2)^2} \left( \frac{\lambda_1^2 \lambda_2 + \lambda_2 (\lambda_1 - \lambda_2)^2}{2 - \lambda_2} + \frac{\lambda_1 (\lambda_1 - \lambda_2)^2 + \lambda_1 \lambda_2^2}{2 - \lambda_1} \right) + \sigma^2. \]

The stability conditions for this controller define a circular region on the \( \lambda_1, \lambda_2 \) plane (see Figure 2.5).

From Figure 2.5 and Eq. (19), it can be seen that to minimize the asymptotic variance and obtain a stable response, we should use small weights \( \lambda_1 \) and \( \lambda_2 \). The problem with this recommendation, originally given in the internal model control literature, is that the transient effect will be too large for small weight parameters. A measure of the severity of the expected transient up to a specified run number \( m \) is given by the average mean square deviation:

\[ \text{MSD} = \frac{1}{m} \sum_{t=1}^{m} E[Y_t - T]^2 \equiv \text{MSD}. \]
Del Castillo\textsuperscript{14} provides an expression for $\overline{\text{MSD}}$ for the algebraically simpler case of $\xi = 1$ (known gain). A spreadsheet optimization model (downloadable from http://www.ie.psu.edu/people/faculty/castillo/research.htm) is available to minimize a weighted sum of the asymptotic variance and $\overline{\text{MSD}}$:

$$\min_{\lambda_1, \lambda_2} \gamma_1 \text{AVAR}(Y_t) + \gamma_2 \overline{\text{MSD}} \quad (20)$$

subject to

$$0 < \lambda_1 < 1$$
$$0 < \lambda_2 < 1$$

where $\gamma_1$ and $\gamma_2$ (with $\gamma_1 + \gamma_2 = 1$) are two weights selected by the user. In Del Castillo\textsuperscript{13,14} it was recommended to simply give equal weight to each objective. A more complete analysis for different objective weights can be performed from which a process engineer can choose a particular set of optimal EWMA weights that solve problem (20) according to his/her preferences. The value of $m$ should be selected based on the number of runs the process will operate. For example, consider the case when $\alpha = 2$, $\delta = 0.1$, and $\sigma = 1.0$, that is, the case when we have a small relative drift $\delta/\sigma$. If $m = 100$ runs are going to be conducted, Figure 2.6 shows the optimal ($\lambda_1$, $\lambda_2$) weights that solve Eq. (20) optimally.
The plot reveals that unless we give more weight to the transient effect than to the AVAR objective (i.e., unless $\gamma_2 > \gamma_1$), the optimal solution to (20) calls for using $\lambda_2 = 0$, which implies the use of a single EWMA-based controller. In other words, for small relative drift, single EWMA control is sufficient. This situation is further emphasized if $m = 20$ is used instead (Figure 2.7). Note that the extreme cases ($\gamma_1 = 1, \gamma_2 = 0$, and $\gamma_1 = 0, \gamma_2 = 1$) in Figures 2.6 and 2.7 lead to the same $(\lambda_1, \lambda_2)$ solutions. In the first extreme case, all weight is given to the asymptotic variance, and this is not a function of $m$. The second extreme case is more striking, since we are giving all weight to $\overline{MSD}$, which is clearly a function of $m$.

For large relative drift $\delta/\sigma$, Figures 2.8 and 2.9 reveal that a double EWMA-based controller is necessary (i.e., $\lambda_2 > 0$ for all cases when $\gamma_2 > 0$). As before, the extreme cases yield the same EWMA weight solutions regardless of $m$.

More information on single and EWMA controllers can be found in Chapter 3 of this book.

2.4 OTHER R2R CONTROL METHODS

As mentioned before, a review of R2R control methods was provided in Reference 11 for up to 1995. Some newer developments are briefly discussed in this section.

Leang and Spanos$^{22,23}$ propose to look at sequences of semiconductor manufacturing processes. They propose two control approaches, one in which feedback/feedforward techniques are applied to each pair of adjacent processes. They also sketch in$^{24}$ a global control approach that will improve the overall capability of the sequence. Their approach can handle multiple inputs and outputs at each process step.

Stefani et al.$^{35}$ propose to use a standard DOE/RSM approach to build quadratic multivariate equipment models of the form

![Figure 2.6](image-url)
where the $X$'s denote the controllable factors. These authors propose an adaptation technique for tuning initial models of the form of (21) that selects “offset” parameters $o_i$ and “gain” parameters $g_i$ in

\[ Y = \beta_0 + \sum \beta_i X_i + \sum \beta_i^2 X_i^2 + \sum \sum \beta_{ij} X_i X_j \]  

(21)
The offset and gain parameters are obtained by minimizing the sum of squared errors of each response added up and weighted by each response prediction error variance. That is, response $k$ receives a weight $1/\sigma_k^2$. This tuning technique, in effect, generalizes the concept of “bias tuning” on which EWMA controllers are based, to the case where not only the bias (offset) but also the gains are adapted in a full quadratic RSM model.

Baras and Patel\cite{1} present a worst-case R2R controller based on an ellipsoidal algorithm that estimates a set of possible parameter values of RSM model. Within that set, a recipe is obtained by taking a worst case (minimax) approach.

Work related to using information from an initial response surface model is that of Hamby et al.\cite{17} They consider a multiple input, single output run-to-run system described by a first-order linear regression equation and controlled by a single EWMA controller. Their approach can be explained more simply by referring to the single input, single output (SISO) case. From the stability condition $|1 - \lambda \xi_j| < 1$, the authors develop the interesting concept of probability of stability $P(|1 - \lambda \xi_j| < 1)$ where the density of $b = \hat{\beta}$ (obtained from regression analysis) is used for integration and therefore can be computed analytically. A second interesting concept used to determine $\lambda$ is the probability of performance, $P_i = P(\text{AMSD}(\lambda) < \kappa)$, which is computed by Monte Carlo simulation of the density of $b$. The value of $\lambda$ that maximizes $P_p$ is selected.

Equipment models with adaptation based on recursive least-square estimation and self-tuning control techniques have been proposed for the SISO case,\cite{11} and the

\[ Y = \beta_0 + \sum \beta_i (X_i + o_i) + \sum \beta_j (g_i X_i) + \sum \sum \beta_{ij} (X_i + o_i)(g_j X_j) + \sum \beta_{ii} (X_i + o_i)^2 \sum \beta_{ii} g_i X_i^2. \]
MIMO unconstrained case. These papers are the predecessors of the OAQC algorithm described elsewhere in this book.

Finally, we should mention that internal model R2R controllers based on first principles — as opposed to based on empirical black-box models — have also been developed.

REFERENCES

APPENDIX. HOW THE ULTRAMAX® SOFTWARE WORKS

The software program called Ultramax* is a multivariate process optimizer that builds a model of a running process in a sequential manner. Such an optimizer is similar in its intent and action to the well-known EVOP strategy as described in Box and Draper.3 Ultramax is, however, software-based, whereas EVOP is not. This section describes in some detail the algorithm underlying the Ultramax optimizer.

The first step in running an application of the optimizer is to enter the problem formulation into the software. This is akin to the design stage of an experimental design–execute–analyze project. A list of input factors (design variables) and output factors (responses) is given along with their operating constraints. Additionally, prior regions are named; the prior region of a variable gives the area inside of which that (input or output) variable has, in the past, occurred.

An objective or goal function is also defined; this may be a single output or a function of outputs (for example, a sum of loss functions). Output constraints are typically the process or product specifications that we desire, and the input constraints define the low to high regions of input factor allowances.

Three features of Ultramax are of particular note.

1. The optimization can begin with little or no data. This is of great value in production when, as is usually the case, no screening experiments have been run ahead of the optimization. If such data are available, they can be entered into the Ultramax database and the program will use them appropriately.

2. As experimental data are added more parameters are estimated and, if enough runs have been accomplished, the data are selected and weighted according to a heuristic method. In this latter case, all (model) coefficients of a second-order Taylor series approximation to the underlying process response surface are the parameters that are estimated.

3. The program recommends one or more advices to be taken as future run settings (recipes). The advice may be strictly adhered to, modified, or dropped if necessary. The advice given is a result of optimizing on the goal function while at the same time respecting all input and output constraints; an advice is a prediction based on the latest version of the response surface that the software has at its disposal. In this sense one can regard it as a feedforward, model-based adaptive optimizer.

* Ultramax is a registered trademark of Ultramax Corp., Cincinnati, Ohio. Thanks to Dr. Carlos Moreno of the Ultramax Corp. for his assistance in this regard.
Item 1 above relates to the estimation algorithm, and we discuss it in some detail below. Item 2 relates to the weighting, or screening, of the accumulated database, a topic that we will not discuss. Item 3 is the well-known topic of constrained optimization (see Press et al.) and needs no further discussion here.

Ultramax estimates a process response surface as a second-order Taylor series approximation. Let us adopt the standard regression model formulation:

$$ y = X\beta + \varepsilon $$  \hspace{1cm} (22)

with \( X \) and \( n \times p \) matrix, \( \beta \) a \( p \times 1 \) vector, \( \varepsilon \) an \( n \times 1 \) error vector which is normally distributed with zero mean and covariance matrix \( \sigma^2 I_n \). Let the ordinary least squares (OLS) estimator of \( \beta \) be \( b \), where

$$ b = (X'X)^{-1}X'y $$  \hspace{1cm} (23)

The Ultramax version of \( b \) is \( B \), with

$$ B = (X'X + K)^{-1}X'y $$  \hspace{1cm} (24)

where \( K \) is a diagonal matrix, \( \text{diag}(k_i) \), with \( k_i \) constants, \( (i = 1, \ldots, p) \).

\( B \) reverts to \( b \) when enough data have finally been observed. As we can see, \( B \) is similar to a ridge-type estimator of the kind proposed by Hoerl and Kennard where \( k_i = k \) for all \( i \). Hoerl and Kennard developed their estimator for \( X \) of full rank, whereas we see that \( B \) is proposed for the rank-deficient case as well.

**A RANK-DEFICIENT BAYES ESTIMATOR**

For \( X \) of full rank it has long been recognized that a correction to the \( b \) of Eq. (23) in the form of a small, positive quantity added to each diagonal element of \( X'X \) has certain desirable features, particularly if \( X'X \) is ill-conditioned, that is, has some very small eigenvalues. Piegorsch and Casella have traced this development as far back as 1944 with Duncan and Levenberg, as well as to 1963 with Marquart. It is of some interest to note that Levenberg’s development was in response to problems of estimating first-order Taylor series approximations to non-linear functions problems that arose due to the omission of second- and higher-order terms.

For convenience, let us assume that the independent variables of (22) are scaled so that \( X'X \) is in correlational form, that is, that diagonal elements of \( X'X \) are unity and the off-diagonals are the same correlations. The Hoerl–Kennard ridge regression estimator mentioned above is

$$ \beta^* = \left( X'X + kI_p \right)^{-1}X'y $$  \hspace{1cm} (25)

where \( k > 0 \) is a constant that is chosen by graphical means or by some other plausible criteria.
As is well known (see, e.g., Graybill\textsuperscript{16}), there exists a \( p \times p \) matrix \( Q \) such that 
\[ Q'XQ' = \Lambda = \text{diag}(\lambda_i) \] with the \( \lambda_i \) being the eigenvalues of \( X'X \). Now suppose that \( Q \) is chosen such that \( \lambda_i \geq \lambda_j \) for all \( i > j \), and that \( \lambda_{r+1}, \ldots, \lambda_p \) are all assumed to be zero for \( \text{rank}(X'X) = r \leq p \).

Partition \( Q' \) into \( \{Q'_r : Q'_{p-r}\} \), where \( Q'_r \) is \( p \times r \) and \( Q'_{p-r} \) is \( p \times (p-r) \), and define \( \Lambda^2 = \text{diag}(\lambda_1, \ldots, \lambda_r) \).

If we define \( (X'X)_r = Q'_r \Lambda^2 Q_r \), then \( b^+ = Q'_r \Lambda^{-2} Q_r X b'y \) is a generalized inverse estimator for \( \beta \) since \( (X'X)_r = Q'_r \Lambda^{-2} Q_r \) is a generalized inverse estimator for \( X'X \).

This fact was noted by Marquart,\textsuperscript{27} who also suggested a combined generalized inverse-ridge estimate for \( \beta \) as
\[
b^{+*} = \left\{ (X'X)_r + kI_p \right\}^{-1} X'y
\] (26)

An estimator that is close to (26) in form may be derived by considering a set of parametric constraints of \( \beta \) of the form \( H\beta = 0 \), where \( H = Q_{p-r} \). Brown\textsuperscript{7} notes that the estimator \( b^* = X^*y \), where \( X^* \) is the Moore–Penrose generalized inverse for \( X'X \), may be viewed as the limit of the Hoerl–Kennard ridge estimator as \( k \to 0^+ \).

A Bayesian approach to (26) is to adopt a Bayesian prior for \( \beta \) such that the prior expectation of \( H\beta \) is zero. One such form, given by Goldstein and Smith,\textsuperscript{21} is
\[
\Phi(\beta) \propto \exp \left\{ -\frac{1}{2\sigma^2} \right\} \left( H\beta \right)^2 \left( H\beta \right)'
\] (27)

The Bayes estimator with respect to quadratic loss is given by the posterior mean of \( \beta \), and is
\[
b^\Phi = \left\{ (X'X) + \left( \sigma/\sigma_\beta \right)^2 H'H \right\}^{-1} X'y
\] (28)

\[
= \left\{ Q' \left[ \Lambda^2 + \left( \sigma/\sigma_\beta \right)^2 \begin{pmatrix} 0 & 0 \\ 0 & I_{p-r} \end{pmatrix} \right] Q \right\}^{-1} X'y
\] (29)

If \( (\sigma/\sigma_\beta)^2 \ll \lambda_i \), for all \( i, \ldots, r \), then
\[
b^\Phi = \left\{ Q' \left[ \Lambda^2 + \left( \sigma/\sigma_\beta \right)^2 I_p \right] Q \right\}^{-1} X'y
\] (30)

\[
= \left\{ X'X + \left( \sigma/\sigma_\beta \right)^2 I_p \right\}^{-1} X'y
\] (31)

That is, we have a plausible estimator that will work even in the rank-deficient case.
Note that this type of estimator is akin to the wider class of estimators derived by Lindley and Smith. Lindley and Smith’s derivations turn on a concept called “exchangeability,” which is here evidenced in (27) by the presence of a common $\sigma_\beta$ for the coefficient vector $\beta$. The presence of exchangeability can here be justified by our device of $X'X$ in correlational form.

**The Ultramax Formulation**

Turning to the Ultramax formulation, we learn that $X$ of Eq. (24) is in deviational, not correlational, form, that is, centered but not standardized. Let us write this special case of $X$ as $x$; that is, for any column of $x$, $x_j = (x_{ij} - m_j)$, where $x_{ij}$ are the raw observations on independent factors or second-order terms, and $m_j$ is its sample mean. Let $s_j$ be its observed sample standard deviation, and let $M = \{ \text{diag}(s_j \sqrt{n})^{-1} \}$ be a $p \times p$ diagonal matrix, $n > 2$. Then $M'x'M = (r_{jk})$, the $p \times p$ matrix of correlations. Substituting $xM$ for $X$ in Eq. (31), we obtain

$$b^\phi \equiv \left\{ (xM)'(xM) + \left( \sigma/s_\beta \right)^2 I_p \right\}^{-1} (xM)'y$$

(32)

Noting that $M' = M$, and $M^{-1} = \text{diag}(s_j \sqrt{n})$, we can derive

$$b^\phi \equiv M^{-1} \{ x'x + K \}^{-1} x'y$$

(33)

with $K = \text{diag}\{ n(\sigma s/\sigma_\beta)^2 \}$, thus

$$b^B = \{ x'x + K \}^{-1} x'y$$

(34)

is an appropriate Bayesian shrinkage estimator for full or rank-deficient cases.
Part 2

R2R Control Algorithms

Several different algorithmic approaches have been proposed in the last few years for R2R process control. The earliest work in this area is that of Sachs and his co-workers at MIT who proposed to use a feedback controller based on the so-called exponentially weighted moving average (EWMA) statistic. In the past, the EWMA statistic had been used widely in industry for process monitoring purposes, but in R2R control it is used as part of a simple feedback controller that suggests when and how to adjust a process. In Part 2 of this book, R2R control algorithms that are based on the EWMA statistic and algorithms based on some other approaches are presented.

Specifically, in Chapter 3, two basic R2R algorithm approaches are described that utilize EWMA data filtering. Included in this chapter is the “gradual mode” controller of Sachs et al., which uses a single EWMA equation, and the so-called “predictor-corrector” controller, developed at Texas Instruments, which uses two coupled EWMA statistics. These two algorithms have been effectively utilized in practical R2R control application in the semiconductor industry; examples of such applications are presented in Chapters 11, 13, and 15.

Chapters 4 and 5 detail a different class of controllers based on a further development of the R2R adaptive control concept. The inner works of an R2R algorithm, termed OAQC, are described. The OAQC can act both as a controller (if the responses should track given targets) or as an optimizer (if responses need to be maximized or minimized). As was mentioned in Chapter 2, the application of optimization techniques usually precede the use of control techniques, since the first are used to find an initial recipe during the qualification of a tool. Thus, there is an inherent advantage in an R2R algorithmic solution that incorporates both optimization and control.
Another important distinction can be made between the EWMA controllers of Chapters 3 and the OAQC and similar controllers of Chapters 4 and 5. EWMA run-to-run algorithms are based on a linear first-order approximation around the current recipe to a possibly nonlinear and drifting process. The EWMA statistics act as a filter of large system/model deviations caused by the nonmodeled dynamics, providing a robust approach to control (see Chapter 2). In a variety of semiconductor processes such local linear approximation becomes necessary. The OAQC and similar methods (e.g., the Ultramax method, see Chapter 2) can be seen as providing second-order (quadratic) polynomial approximations of the process responses around the current recipe. The parameters in these quadratic polynomials are continuously tuned by a learning mechanism (see Chapter 4), and this allows tracking of nonlinear changes in the process that occur from run to run.

The algorithms presented in Chapters 3 through 5, while addressing R2R control using a variety of approaches, do not represent the complete set of algorithms available for R2R control. For example, neural network and expert-systems-based solutions have been proposed. While each of these algorithms has its own strong points, and effective R2R control can often be accomplished using basic R2R control techniques (see Chapters 1 and 2), a detailed comparative analysis nevertheless should be attempted as part of the process control solution development and deployment process. Chapter 6 provides one of the few studies that compares the performance of various R2R control algorithms presented in the literature. More importantly, it identifies aspects of an R2R control algorithm that should be explored as part of the selection process.
3 Basic R2R Control Algorithms

William Moyne

3.1 INTRODUCTION

In the Introduction and Part 1 of this book we illustrated the many current benefits and discussed potential future benefits of R2R control. These benefits can be realized to a large extent utilizing straightforward and uncomplicated control algorithms. Having an understanding of these algorithms will provide the reader with ammunition to help develop simple but effective R2R control solutions.

The focus of this chapter is to explain two of the more basic R2R control algorithms used today in the semiconductor industry, namely the “EWMA gradual mode” and the “predictor–corrector controller (PCC).” Both of these algorithms are multiple input, multiple output (MIMO) first-order polynomial control approaches. That is, they approximate the system they are controlling as a set of polynomial equations containing only constant and first-order terms. This may at first seem limiting, but with these algorithms it is assumed that R2R control will be applied to a relatively stable process, subjected to noise and drift, and other techniques will be used when the process becomes unstable (mechanisms such as SPC could be utilized to identify the need to transition to these other techniques — see Chapter 1). The process drift may be monotonic or cyclical. Once the nature of the system has been established, both the EWMA gradual mode and PCC algorithms act as piece-wise linear approximators over many runs. Using this strategy, complex models can be linearized around an optimal point and, as such, are presented to the controller to maintain that point.

3.2 ALGORITHM FORMULATION

Since both of the algorithms being described in this chapter are polynomial-based and linear, they can be represented using standard linear equation techniques. Specifically, the algorithm uses an ‘m’-by-‘n’ (inputs-by-outputs) linear model with an additional constant term.*

\[ Y = Ax + c \]  

* Equations will use the following notation: Arrays will be capitals, vectors will be lower case, and indexing within a vector or matrix will be lower case with subscripts. In addition, the special subscript \( t \) will be reserved for time or run number information.
where \( y \) = system output,
\( x \) = input (recipe),
\( A \) = slope coefficients for equation,
\( c \) = constant term for linear model.

This matrix notation can be expanded into the familiar simultaneous equations notation. Each output represents a target of control, and each input represents an adjustable parameter in the recipe.

\[
y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m + c_1
\]
\[
\ldots
\]
\[
y_n = a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m + c_n
\]  \hspace{1cm} (2)

The algorithm operates under the assumption that the underlying process is locally approximated by the first-order polynomial model, and that this polynomial model can be maintained near a local optimal point solely by updating the constant term \( c \). In order to allow maximum flexibility for algorithmic development, the computational engine associated with the R2R algorithm is divided into two parts:

- Model update
- Recipe update

This division applies to both the EWMA gradual mode and predictor corrector algorithms. In the remainder of this section we will describe the methods used for both model and recipe update in each of these algorithms.

### 3.2.1 Model Update

Updating the model is the first step in the control process. Currently (with these two algorithms) this entails updating the constant term used in the polynomial model equation. This step of the control process determines how aggressive the controller acts, as well as its ability to handle different conditions such as drift.

#### 3.2.1.1 Model Update in the EWMA Gradual Mode Algorithm

The EWMA gradual mode algorithm uses one of the simplest methods of model update (see Chapter 1). As its name implies, it filters historical data with an exponentially weighted moving average (EWMA) filter to prevent overcontrol. A single weighting factor \( \alpha \) is used.

\[
c_t = \sum_{i=1}^{t} \alpha(1-\alpha)^{t-i} (y_i - Ax_i)
\]  \hspace{1cm} (3)

Although (3) would provide the desired EWMA weighting, it also requires data from all previous runs. Luckily, this can be simplified using the additive nature of the series to generate an iterative expression for the constant term update:
Using an EWMA filter to smooth the control action on a linear process has been shown to provide good results in a number of applications.\textsuperscript{1-5} The simplicity of the algorithm also makes it a natural starting point for an R2R control strategy.

### 3.2.1.2 Model Update in the Predictor–Corrector Control (PCC) Algorithm

The PCC algorithm is an expansion on the EWMA gradual mode that adds an explicit model for drift. Drift is present in many VLSI processes that can “age.” Examples include pad wear on a chemical mechanical planarizer, or buildup on the wall of a plasma etcher (examples of this type of drift are described further in the Introduction section of this book). The PCC algorithm uses two parameters, $\alpha$ and $\beta$, to weight noise and drift, respectively. EWMA weighting is used for both the constant term update and for the drift estimation.

\begin{align*}
  n_i &= \alpha(y_i - Ax_i) + (1 - \alpha)n_{i-1} \\
  d_i &= \beta(y_i - Ax_i - c_{i-1}) + (1 - \beta)d_{i-1} \\
  c_i &= n_i + d_i
\end{align*}

where

$\begin{align*}
  n &= \text{estimation of noise for run} \\
  d &= \text{estimation of drift for run} \\
  A &= \text{slope coefficients for model} \\
  y &= \text{measured output of the system} \\
  x &= \text{input (recipe)} \\
  c &= \text{constant term for model} \\
  \alpha &= \text{EWMA weighting for noise estimation} \\
  \beta &= \text{EWMA weighting for drift estimation}
\end{align*}$

Simulations of PCC vs. EWMA on processes with and without drift show that PCC provides better drift response with no noticeable penalty when drift is absent. Changes in the drift rate, however, can lead to potential overshoot based on the time averaging of the PCC drift estimator. Figure 3.1 shows a comparison between PCC and EWMA control under both drift and noise conditions.\textsuperscript{5}

### 3.2.2 Recipe Update

Once a suitable constant term has been chosen for each of the model equations separately, the task of determining a new recipe must be addressed. This solution must take into consideration many conditions and constraints that affect the process. Although, in the actual controller, the final recipe is calculated in the presence of all conditions and constraints, they will be discussed separately. The remainder of
this section will be devoted to presenting the algorithm used for fitting a solution to the numerous outputs. Parameters and constraints that can affect this solution, and methodologies for dealing with them, are discussed in Section 3.3.

**3.2.2.1 Curve Fitting**

At the heart of the R2R recipe algorithm is a matrix least-squares routine. Least-squares is a method for determining the optimal solution (curve fit) for an over-determined (#outputs > #inputs) system. The method has the favorable property of providing the “best” solution even if an “exact” solution does not exist. In this case, “best” refers to the solution that minimizes the squared error between itself and the exact solution, and “exact” refers to input values to the model that generate the desired target value exactly. Care must be taken when formulating the problem. The absolute scale of the inputs can cause certain inputs to be favored over others when an optimal solution is chosen. This is beneficial when used to modify the behavior of the controller, but is not desirable if it is not controlled (see input/output weights discussed in sections 3.3.3 and 3.3.4). To prevent unwanted bias, all inputs can be normalized (as shown in Eq. (6) below) to between –1 and 1 before any computation.

**FIGURE 3.1** PCC vs. EWMA.
where \( x_n \) = normalized recipe
\( x_{\text{min}} \) = lower bound for recipe
\( x_{\text{max}} \) = upper bound for recipe.

Based on the formulations of the problem and possible boundary constraints, the least-squares solution can take on three forms:

- Exact solution
- Overdetermined
- Underdetermined.

Figure 3.2 illustrates examples of the three possible forms of the solution to a control problem. Each of these must be solved in a different manner. For the underdetermined case (Figure 3.2a), the system has two inputs (x1 and x2) and one target output (3). This problem would normally lead to an infinite number of solutions (represented by a line). Since all solutions are “correct,” it would serve the purpose of the algorithm.
to simply pick one of the values. This extra degree of freedom is instead used to bring the solution as close to the previous recipe as possible, minimizing least-squares distance. This not only reduces the solution to a single value, but also has the positive effect of minimizing the extent of the changes to the input parameters of the system.

Figure 3.2b illustrates the effects that two conditions have on a two-input system. These constraints, which are represented by two lines, create a problem that has only one solution. This solution (represented by the intersection of the two lines) satisfies both conditions exactly. Due to the lack of freedom in the problem, the previous solution information is not used.

With the addition of a third condition, an overdetermined problem arises (Figure 3.2c). In this formulation there are more conditions than degrees of freedom in the inputs, so there is no exact solution. A least-squares algorithm is used to minimize the error between the target for each output and the final solution. Again, the previous solution is not used to avoid further constraining the problem.

In order to provide a flexible environment that can handle any input/output combination, the algorithm must first determine which case is occurring, and then generate the solution accordingly. This could require three separate computational routines, but luckily this can be reduced to one. Since the least-squared solution is guaranteed to be the best, it can be used to solve the exact and overdetermined cases directly and the underdetermined case with some preprocessing of the data. The mathematical formulation of the recipe update problem for each of these cases is described in more detail in the following subsections.

3.2.2.1.1 Exact solution
If the number of inputs \( n \) to a system is equal to the number of outputs \( m \), then there is exactly one solution that satisfies the desired outputs. The calculation of this solution is straightforward:

\[
y = Ax + c
\]

\[
x = A^{-1}(y - c)
\]

Note that here are two uses for the symbol \( y \) in the equations used for control. First, it represents the output of the system. This is what is measured as the real value of the system output. This value is primarily used to update the constant term \( c \), as discussed earlier. Second, it is used to denote the target that is desired for that output. This second use is how it is used in the remainder of this chapter. The two meanings are similar in that they are the observed and ideal values, respectively, of the system output.

3.2.2.12 Overdetermined
There are two events that could lead to an overdetermined problem. The first is that the problem was formulated with fewer inputs than outputs \( n < m \). Second, the controller could have originally been underdetermined or exactly determined, but
input bounds forced certain inputs to be locked, thus decreasing the number of controllable inputs.

Once an overdetermined case is encountered, a least-squares error fit is applied. This ensures, in a least-squares sense, that the solution places the output as close as possible to the target. The calculation of the solution is as follows:

\[ y = Ax + c \]
\[ A^T (y - c) = A^T Ax \]  \hspace{1cm} (8)
\[ x = \left( A^T A \right)^{-1} A^T (y - c) \]

3.2.2.1.3 Underdetermined

In contrast to the overdetermined case, the underdetermined case is encountered when the number of inputs exceeds the number of outputs \((m > n)\). This is often the case in a process. Several inputs can be modified to help maintain a certain output, so the possible solutions are infinite.

Although being able to reach target is always desirable, the choice of the “best” solution from the set of all possible solutions must be done in a consistent manner. Again we turn to least squares. This time, however, instead of merely obtaining an answer that hits the target, we can also select an answer that is closest to the previous recipe while still exactly solving the original problem. In this way we can ensure both that our output is guaranteed to be correct, and that the associated inputs are modified as little as possible.

The actual formulation of the problem is a little more complex than the other cases. It involves the use of a Lagrange multiplier \((\lambda)\) to take the two constraints and merge them into a single equation. This method of obtaining “best” results for underdetermined systems has been used with many of the very first R2R control solutions applied to actual process control, and continues to be used today. The calculation of the updated recipe parameters “\(x\)” for the underdetermined case is as follows:

\[ \min \| x - x_0 \|^2 \bigg|_{Ax=b} \]
\[ L = \frac{1}{2} (x - x_0)^T (x - x_0) + \lambda^T (Ax - b) \]
\[ \frac{dL}{dx} = (x - x_0)^T + \lambda^T A = 0 \]  \hspace{1cm} (9)
\[ x - x_0 = -A^T \lambda \]
\[ Ax - Ax_0 = -AA^T \lambda \]
\[ Ax = Ax_0 - AA^T \lambda = b \]
where $b = t - c$ (Target – constant term)

$x_0$ = recipe from previous run

$\lambda$ = Lagrange multiplier

$A$ = slope coefficients for model

$L$ = equation to minimize

### 3.3 CUSTOMIZATION OF ALGORITHMS TO PRACTICAL APPLICATIONS: PARAMETERS AND CONSTRAINTS

The formulations given in the previous section provide “correct” mathematical solutions to control problems. In the industrial application of R2R control, however, there are a number of practical issues that must be addressed by the algorithm that may alter the solution advice generated by these algorithms. These additional constraints and parameters must be taken into consideration along with those given by the R2R control equations themselves before a final solution can be found. This is what often separates a theoretical solution from a “feasible” or “practical” solution.\(^\text{10}\) When the controller incorporates these constraints and parameters, the recipe “advice” generation is altered in a predictable manner. Although these parameters can complicate an otherwise simple control approach, they can also provide for valuable operator influence to complement the theoretical controller.

A list of some of the constraints and parameters that impact (e.g., limit or bias) the answers provided by the theoretical solutions of Section 3.2 is as follows:

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Bias parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input bounds</td>
<td>Output weights</td>
</tr>
<tr>
<td>Input resolution</td>
<td>Input weights</td>
</tr>
</tbody>
</table>

In the remainder of this section, these constraints and parameters are described in more detail, and algorithm enhancements are presented to incorporate the impact of these factors into the mathematical formulations presented in Section 3.2.

### 3.3.1 INPUT BOUNDS

The R2R controller is generally designed to control actual machinery and, as a result, must account for limitations in the range of possible settings that an input may have. These limitations could be machine imposed (e.g., specified in a user manual), or
operator imposed (e.g., input range for a parameter that has been qualified for a process). One way to address input bounding with control is to simply determine the optimal recipe without input bounds, then fix all input R2R control advices that exceed these bounds to the closest valid (within bounds) setting. This approach provides the necessary constraints, but generally results in a less than optimal setting for the equipment. It is important that the final recipe is chosen in the presence of these constraints. To achieve this the R2R algorithm can be modified to use an iterative approach as shown in Figure 3.3.

This approach differs from the one-pass approach in one key area. After the variables have been modified to respect their maximum ranges, these variables are removed from the system and the process is repeated. This reduces the possibility of a nonoptimal solution, but does not guarantee an optimal solution. It is provided as a computationally inexpensive alternative to a full optimization that can at least guarantee valid if not optimal results.

### 3.3.2 INPUT RESOLUTION

A major problem faced when applying R2R control to a real process is input resolution. Even a perfectly modeled system can suffer from this. Control decisions based on infinite resolution must be rounded by the operator (or the machine) to acceptable increments. This can often lead to unacceptable control, and as a side-effect can give false information back to the controller algorithm: namely that the suggested recipe was used when, in fact, a rounded version was used.
As a first step to addressing this problem, a simple iterative method is proposed to provide resolution control. Inputs are ordered from least to most adjustable (using their input weights — see Section 3.3.4) and then sequentially rounded and removed from the equation. The remaining inputs are then adjusted to obtain the best solution for the new problem. This is repeated until all inputs have been rounded. Figure 3.4 shows a diagram of the method used.

### 3.3.3 Output Weights

It is often the case that the desired target of a process cannot be reached given the constraints of the system. If this is the case, a decision must be made as to the relative importance of each output. The default is, of course, equal weighting, but this may not be desirable. For example, if a process has two outputs, thickness and uniformity, the operator may want optimal thickness with a secondary requirement of good uniformity. The weights could also be set inversely proportional to the variance of the output variable. This would put greater importance on those variables with low variance, e.g., those that can be more accurately controlled. The controller accommodates output weighting by applying an output weighting matrix $W$ to the system.

$$W = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & w_{mn} \end{bmatrix}$$  \hspace{1cm} (10)
where \( w_1 \ldots w_m \) are the relative weights for outputs \( 1 \ldots m \).

The system equation for the output becomes

\[
W_y = W a x + W c
\]

\[
W(y - c) = W a x
\]

\[
(WA)^T W(y - c) = (WA)^T W a x
\]

\[
(A^T W^T W a)^{-1} A^T W^T W(y - c) = x
\]

The weighting works by biasing the magnitude of certain outputs so that when a least-squared solution is calculated, outputs with higher weights contribute a greater penalty to the solution if they are off target. Thus, higher-weighted outputs are set closer to their targets than other outputs. Application of output weights in an exact or underdetermined system has no effect on the output; in both cases there is no reason to sacrifice one output to obtain another, as there is an infinite solution set.

Other bias terms related to direct output weights are the model update weights. These weights (\( \alpha \) for EWMA, \( \alpha \) and \( \beta \) for PCC control) determine the aggressiveness of the controller for each of the outputs. These parameters can be used to minimize the impact of certain noisy outputs on the model update and recipe generation of the controller, while increasing the impact of more stable outputs. The result is that a system can quickly adapt to changing conditions while being somewhat resistant to process noise. An added benefit of these parameters is that they provide these biased noise filtering capabilities regardless of the type of control problem (i.e., underdetermined, exact, or overdetermined).

### 3.3.4 Input Weights (Input Adjustability)

Although the inputs to the system can be normalized to ensure consistent operation, weights can also be applied to these inputs to add yet another level of control. Input weights enable the user to set the adjustability of the inputs. That is, heavily weighted input variables are adjusted with greater magnitude relative to lightly weighted variables to achieve process control.

Application of the weighting is achieved by adjusting the normalized input variables so that the least-squared distance incurred by each variable (distance of new control advice \( x_t \) from \( x_{t-1} \) where \( t \) is the run number) is adjusted by its input weight. This should not be confused with the output weighting mechanism discussed in the previous subsection. In the underdetermined systems where the model indicates a set of solutions where all outputs are met, the recipe is determined with the added constraint of being as close to the old recipe as possible. This can be biased by the relative weighting of the inputs. Inputs that are weighted heavily are forced to be the least adjustable due to their relatively large effect on the error calculation for the recipe (i.e., the difference between the target and value predicted by the suggested recipe applied to the model). A matrix \( V \) and its inverse \( V^{-1} \) are used to apply the input weighting.
where \( v_1 \ldots v_n \) are the relative weights for inputs 1 \ldots n. Note that the input weighting has no effect on both overdetermined and exact solution problems. In those cases, the inputs are not factored into the calculation of the error for the final solution, so the magnitude of the inputs, which is the key to their weighting, is irrelevant.

Once a weight matrix has been defined it must be applied in such a manner as to ensure that the formulation of the problem leads to a correct solution. In order to achieve this, the weight must be applied to both the recipe and the slope (first-order) term. First, the application of the weight term \( V \) to the recipe \( x \) modifies the least-squared error generated by these inputs when determining the solution closest to the previous solution (see Section 3.2.1.3). The side-effect of this weighting is that the new output generated by these inputs is not consistent with the original problem formulation. To remedy this, the slope term \( A \) is weighted with the inverse of the recipe weight. The system equation for the output then becomes

\[
Y = Ax + c
\]

\[
Y = (A \cdot V^{-1}) \cdot (V \cdot x) + c
\]

\[
y = A^* x^* + c
\]

This new formulation can be used in place of the original variables to provide the necessary weighting. The problem is then treated as before (see Section 3.2.1.3), but with the new scaled values.

\[
\min \| x^* - x_0 \|^2 \bigg|_{A^* x^* = A x = b}
\]

The solution, however, is based on these scaled values, so it must be scaled back to the original domain.

\[
x = V^{-1} \cdot x^*
\]

### 3.4 Conclusions and Future Improvements

Two basic R2R algorithms have been introduced in this chapter. Both the EWMA gradual model and PCC algorithms utilize a MIMO linear polynomial modeling approach with R2R updates of the constant terms of the equations. The PCC additionally adds an explicit model for drift. These basic R2R algorithms have been proven effective in a variety of semiconductor manufacturing applications (see, for example, Chapters 1, 11, 13, and 15); a key element of the successful application of these algorithms, however, is their customization/enhancement to address practical
issues such as input and output bounding, discretization, and weighting, as described in Section 3.3.

There are a number of other algorithms that can be applied to R2R control; some of these algorithms are detailed in the remainder of this section of the book. Many new algorithms will undoubtedly be developed in the near future. Since the two algorithms presented in this chapter are modularized into two components, the model update stage and the recipe update stage, these algorithms can benefit from advancements at either stage. A key component of the successful utilization of these enhancements, however, will be the ability of these enhanced solutions to accommodate the practical issues in R2R control introduced in Section 3.3.

ACKNOWLEDGMENTS

Much of the material presented in this chapter is derived from Reference 11 and is reprinted with permission.

REFERENCES

4 Learning and Optimization Algorithms for an Optimizing Adaptive Quality Controller

Enrique Del Castillo

4.1 INTRODUCTION

The OAQC (optimizing adaptive quality controller) is a process optimization and control software tool recently developed for application in run-to-run (R2R) manufacturing environments. Its most recent version is the result of the evolution of R2R applications of adaptive control techniques for linear unconstrained single input, single output (SISO) systems;\(^5\) linear unconstrained multiple input, multiple output (MIMO) systems;\(^3\) and nonlinear, constrained MIMO systems.\(^8\) The OAQC has been implemented on the NextStep (Mach) and Windows NT platforms. The NextStep OAQC has been integrated with the Generic Cell Controller (GCC, see Reference 14 and Chapter 11 of this book) whereas the WinNT version is not integrated as of this writing. This chapter will specify which feature applies to each version. When no specification is made, the feature applies to both versions.

Many R2R controllers are based on response surface models that are obtained through experimentation during the “qualification” of a process. After qualification, an R2R controller will determine recipes at run \(t\) based on the estimated model parameters at that run. For a SISO system, a common response model is simply

\[
Y(x)_t = \alpha + \beta x_{t-1} + \delta t + \varepsilon_t
\]

where \(\alpha\) and \(\beta\) are parameters, \(Y\) is the response, \(x\) is the level of the controllable factor, \(\delta t\) models a deterministic drift, and the \(\{\varepsilon_t\}\) constitutes a white noise sequence with variance \(\sigma^2\). The so-called EWMA controllers\(^2,15,16\) popular in R2R applications and used as a “benchmark” in the area of R2R control, modify the estimate of \(\alpha + \delta t\) by computing the estimate \(a_t + D_t(t)\) and updating the estimates \(a_t\) and \(D_t\) at every run \(t\) using EWMA equations. The estimate of the process gain, \(b = \hat{\beta}\), is obtained during qualification experiments and is usually not updated during the control phase.
As mentioned in Chapter 2, these controllers are in fact *internal model controllers*\(^{10}\) and their properties have been studied in considerable detail.\(^4,11,16\)

Rather than performing the two phases, “qualification” and “control,” separately, the OAQC can start without prior response models and simultaneously build models and optimize a process, keeping the performance at its optimum as long as it is possible. In contrast to EWMA controllers, estimation (i.e., adaptation) of all parameters, including the gains, can take place. This chapter describes the learning and optimization characteristics of the optimizing adaptive quality controller. Section 4.2 discusses the optimization problems solved by the OAQC. In Section 4.3 we describe the OAQC learning algorithm. The important concept of a local model is explained in Section 4.4. Finally, some additional features of the OAQC controller are discussed in Section 4.5.

### 4.2 OPTIMIZATION

To better understand the OAQC approach for process optimization, it is useful first to look at how design of experiments (DOE) and response surface methodology (RSM) typically work.

When building a response surface model of a response, \(\hat{Y}(x)\), DOE approaches recommend different settings (recipes) \(x\) based on some optimality criteria for the design or for the estimated responses such as orthogonality, rotatability, or D-optimality. These criteria do not consider whether one wishes to maximize or minimize a particular response; they concentrate only on model-fitting properties. Suppose that, after factor screening experiments have been conducted, there is a single response \(Y(x)\) that is known to be affected by \(n\) controllable factors, where \(x \in \mathbb{R}^n\). The experimental design problem is then to find

\[
X = \begin{bmatrix}
x_1' \\
x_2' \\
\vdots \\
x_k'
\end{bmatrix} = \begin{bmatrix}
x_{11} & x_{12} & \cdots & x_{1n} \\
x_{21} & x_{22} & \cdots & x_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kn}
\end{bmatrix}
\]

such that some criteria for \(X\) is optimized. For example, for D-optimality, \(|X'X|\) is maximized, which implies that the volume of the confidence ellipsoid of the parameter estimates is minimized (\(\text{Var}(\hat{\theta})\) is “minimized” in this sense). Note that this minimization is achieved regardless of our goals or objectives for \(Y\). If we ignore the response objectives, D-optimality is a useful design criterion when there are several bounds or constraints that limit the possible values that \(x\) can take.

Once an experiment is designed, the \(k\) trial recipes are run and the parameters \(\theta \in \mathbb{R}^k(k \geq l)\) are estimated using some variant of the least-squares method:

\[
\min_{\theta} \sum_{i=1}^{k} (Y_i - \hat{Y}_i(\theta; x_i))^2
\]
where the $x$’s are fixed and the optimization is over $\theta$-space. The end result is a parametric model $\hat{Y}(\hat{\theta}; x)$. The last step in the qualification of a process using DOE and RSM techniques is the optimization of the recipe $x$, namely

$$\min_x \left[ \hat{Y}(\hat{\theta}; x) - T \right]^2 \quad \text{“target is best”}$$

or

$$\min_x \hat{Y}(\hat{\theta}; x) \quad \text{“smaller the better”}$$

or

$$\max_x \hat{Y}(\hat{\theta}; x) \quad \text{“larger the better”}$$

Here, the optimization is over the recipe (factor) $x$ space, perhaps subject to one or more constraints in $x$.\(^6,7\) Note that during this step $\hat{\theta}$ remains fixed. Actually, RSM usually starts with a series of steepest ascent/descent searches based on a first-order model until a second-order model can be fit. For simplicity of presentation, we assume here that $Y$ has been determined to be of second-order after steepest ascent searches.

An R2R controller like the EWMA controllers will take the models $\hat{Y}(x)$ and will try to keep the responses $Y$ at the optimal performance in the presence of process noise and drift. As can be seen, building models for R2R control consists of three optimization processes performed sequentially: (1) an experimental design is defined based on some optimality criteria, (2) a model-fitting (e.g., least-squares) minimization step is performed, and (3) a recipe-finding optimization step is performed. The OAQC, instead of solving these three problems in series, attempts to solve them simultaneously in an incremental way at each run, speeding up the qualification phase of a process and reducing extreme disruptions to the process while the OAQC learning algorithms are running.

### 4.2.1 Simultaneous Response Optimization and Model Fitting

For ease of presentation, assume again that there is a single response $Y$ that is known to depend on $n$ controllable factors. It is desired to keep $Y$ as close as possible to a target $T$. However, the input–output relation is unknown and a model $\hat{Y}$ initially may not be available. The OAQC solves, at each run $t$, the following problem:

$$\min_{x_{r+1} \in \Omega_t} \left( 1 - \lambda_t \right) \left[ \hat{Y}(x_{r+1}) - T \right]^2 - \lambda_t \left( \text{Var}(\hat{Y}(x_{r+1}))/\sigma^2 \right)$$

(1)

where the first term is the squared deviation from target of the one-run-ahead predicted response, and the second term is the scaled variance of the current estimated
response. The optimization is carried over the factor space within a set $\Omega$ defined by factor bounds. The constants $\{\lambda_t\}$ are such that $0 \leq \lambda_t \leq 1$, and give relative weights to the two objectives in the function: (1) optimization of the response (first term) and (2) model fitting (second term). As $t$ increases, the value of $\lambda_t$ decreases from 1 to 0. Notice that if $\lambda_t = 1$, the second term is equivalent to a conditional D-optimality criterion (see Reference 13), i.e., the next run factors are selected where current prediction is worse as measured by the response variance. Performance measure (1) is implemented only in the OAQC WinNT. For the corresponding performance criterion used by the NextStep version, see Reference 8.

4.3 LEARNING ALGORITHM

At each run, the solution to problem (1) provides a new set of factors $x_{t+1}$ to try next. Initially, optimization of the objective function (1) will tend to favor factor settings that will lead to obtaining good response models, as opposed to factor settings that achieve desired response performance. Learning is conducted through the estimation of the response parameters using a multivariate recursive least-squares (RLS) algorithm.\(^\text{12}\) The RLS algorithm finds $\hat{\theta}$ such that it minimizes $\frac{1}{N} \sum_{i=1}^{N} e_i^2$, where $e_t = y_t - \hat{\theta}' \phi$, is the residual at time $t$ and $\phi$ is a $l \times 1$ vector of regressors (typically, the $x$'s). The algorithm is

$$K_t = \frac{P_{t-1} \phi_t}{1 + \phi_t' P_{t-1} \phi_t}$$  \hspace{1cm} (2)

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t (y_t - \phi_t' \hat{\theta}_{t-1})$$  \hspace{1cm} (3)

$$P_t = [I_n - K_t \phi_t'] P_{t-1}$$  \hspace{1cm} (4)

where $K_t$ is an $(l \times 1)$ vector of gains or weights and $P_t$ is an $(l \times l)$ matrix proportional to the matrix of covariances of the parameter estimates at run $t$.

Note that we do not specify any structural requirement on $\theta$ and $\phi$. Therefore, response models can be linear, linear plus 2-factor interactions, or fully quadratic polynomials, depending on the process engineer’s needs. Models can also contain a drift parameter. Once the model orders are defined, the dimension $l$ is adjusted in the recursive estimation algorithm accordingly.

4.4 LOCAL VS. GLOBAL RESPONSE SURFACE MODELS

The OAQC algorithm\(^*\) constructs local models that are updated as each run is conducted. This is in sharp contrast to traditional RSM-based run-to-run control techniques where a response model is first fit, then used by an R2R controller. In a

---

\(^*\) Sections 4.4 and 4.5 apply only to the Windows NT OAQC.
classical R2R controller, inadequate performance can occur if the controller recommends factor settings outside of the region where the response model was fit in factor space, or if the model–system mismatch is large. Rather than using such a “global” model that hardly can be valid over a large region in the factor space, the OAQC constructs response models that are valid locally. Local models are achieved by the OAQC using a variety of techniques. First, an “initial region” of factor settings, $\Omega_1$, is selected by the process engineer. This will limit the values that controllable factors can take, and will usually be located around a baseline $x_0$ where the process has been operating in the past. As more runs are conducted, if the OAQC determines that better operating conditions might exist outside $\Omega_1$, the initial region is dropped and the OAQC will move outside of the initial region. At this point the OAQC will select controllable factors that (1) satisfy only operational region ($\Omega_2$) constraints, and (2) satisfy a local model constraint, defined as an ellipsoid of controllable factors $x$ computed based on the most recent recipes. The centroid of this ellipsoid moves according to an EWMA, giving more weight to the more recent factor settings. The operational region should be such that $\Omega_1 \subset \Omega_2$. This defines limits a process engineer does not want to exceed in the interest of safety. The main advantage of using an initial region and a local model constraint is that changes in the operating conditions of the process are not abrupt, and therefore the process is minimally disrupted from its regular operation during learning/optimization. If the initial region limits are dropped, the local model constraint guarantees that the next run will not differ greatly from the most recent experimental runs. These goals are similar to the goals in evolutionary operation techniques. Figure 4.1 illustrates the idea of local models and an initial region.

In the figure, there are two factors ($x_1, x_2$) that affect the response. Two quadratic models were sequentially fitted, both of them based on information collected within an initial region. Eventually, the OAQC finds a small region (depicted as Model 3) where a better performance is achieved outside the initial region $\Omega_1$, but inside of $\Omega_2$. Recently, Hurwitz and Del Castillo studied the performance of the OAQC-WinNT learning and optimization algorithms. They considered a simulated, $3 \times 2$ highly nonlinear CMP process with large trends, a particularly difficult system to control. It was found that the OAQC keeps the responses consistently closer to their targets with a performance considerably better than a simple open-loop operation of the process when not much room for improvement was available over the open-loop scenario. For an account of the performance of the OAQC, see Chapter 5 in this volume.

### 4.5 RESET ACTION

During a control session, desired targets may change or a response type may need to be changed. The OAQC defines four types of responses, in a way similar to Taguchi*:

1. **target** (the closer to target the better)
2. **max** (the larger the better)

* The OAQC NextStep version allows only target responses subject to constraints.
3. min (the smaller the better)
4. constraints only (response should satisfy constraints only).

Suppose that at some point in an optimization/control session, considerably different (new) targets are desired. When the targets change significantly, any control algorithm should contain an adequate “reset” action that will adapt future recipes to the new desired performance. A reset action should also be triggered when a response type changes, for example, from max type to target type. In the OAQC, a reset action is triggered if any of the changes or transitions shown in Figure 4.2 occur.

We point out that some of the transitions are unlikely to occur in practice (e.g., changing a response from “max” to “min” or vice-versa). Once one of the transitions in Figure 4.2 occurs, the OAQC will try to find a new recipe $x$ that meets the new objectives. If unsuccessful, it will “explore” the factor space to fit a better model and get new recipes. This will occur if it is concluded that the local models are no longer locally valid.

### 4.6 OTHER FEATURES

This chapter describes the main details of the optimization and learning capabilities of the OAQC. There are many other features that are useful in run-to-run applications that we briefly describe here.*

---

* Features 1 and 2 apply only to the WinNT OAQC.
1. **Uncontrollable and controllable factors.** A subset of the input factors that affect the responses may not be controllable; however, they might affect the responses of the process and they may be observable. Including these uncontrollable factors in the response models will improve their prediction capabilities. The OAQC enables definition of these two types of factors. The sets $\Omega_1$ and $\Omega_2$ will be defined by controllable factors only. Once a subset $x_c \subset x$ of controllable factors is defined, the optimization of Eq. (1) is carried out by leaving the uncontrollable factors $x_u$ fixed and optimizing (1) only with respect to $x_c$.

2. **Nonoptimizable responses.** It may be of interest to measure and model some response without optimizing it, perhaps simply for monitoring purposes. This can be done by defining it as a “constraint only” response and *not* defining any constraint. Thus, prediction models will be built for such responses, but no optimization will take place with respect to them. In this way, the ability to build regression models (of up to second order) of nonoptimizable responses of interest is incorporated into the system. These responses, however, must depend on the same factors $x$ as the optimizable responses.

3. **Factor and responses weights.** Not all controllable factors are equally easy to adjust, and it is frequently desirable to define relative weights that will make a factor less likely to be varied from its previous run. Similarly, not all optimizable responses will be equally important, and the OAQC allows us to define relative weights for each response. For example, once models have been fit with enough precision by optimizing Eq. (1), we will have $\lambda_i \to 0$. Then, a standard way of incorporating response weights is

$$\left(\hat{Y}_{r+1} - T\right) \Gamma \left(\hat{Y}_{r+1} - T\right)^\prime$$
where \( \hat{Y} \) and \( T \) are \( p \times 1 \) vectors of estimated (optimizable) responses and response targets, respectively, and \( \Gamma \) is a \( p \times p \) diagonal matrix of response weights, i.e., the \( i \)-th diagonal element of \( \Gamma (i = 1, 2, \ldots, p) \) is the relative weight given to response \( i \). The higher the weight given to a response, the more penalized its deviations from target are, and the more strict control of that response we will have.

4. **Initial models.** The OAQC can start its optimization/learning routine without prior information about the responses. However, if prior models are available (perhaps based on previous process data), these can be used in two different ways: (1) as initial models that will speed up the optimization/learning process, and (2) for simulation purposes. For the learning and estimation process, a vector \( \theta_0 \) of initial parameter estimates may be defined together with its associated precision matrix \( P_0 \). The more confidence we have in the estimates \( \theta_0 \), the smaller the diagonal elements in \( P_0 \) should be. This corresponds to a Bayesian interpretation of the recursive least-squares estimation algorithm.\(^{12}\) The OAQC allows us to simulate MIMO systems with responses of up to quadratic order with linear drift and additive normally distributed white noise. The OAQC optimization and control algorithm can then be applied to the simulated responses as if they were the true (unknown) system responses.

**REFERENCES**

5 An Adaptive Run-to-Run Optimizing Controller for Linear and Nonlinear Processes

Arnon Max Hurwitz and Enrique Del Castillo

5.1 INTRODUCTION

Manufacturing processes encounter many hindrances to fulfilling operational equipment efficiencies. In the Introduction to this book we noted that two causes of trouble on the manufacturing floor are typically the batch-to-batch or “run-to-run” (R2R) drift of the actual output performances from the desired level, and also an inherent inability of the process to deliver consistent high quality. This last-stated problem has usually been tackled in an off-line manner by statistically designed optimization experiments. Designed experiments are very effective for developmental work, but are seldom used in an actual production situation. The first-stated problem (i.e., R2R control) of tool drift, or shift, has traditionally been managed by the tool operator tweaking various recipe setpoints of the process, such as temperature. The problem here lies in varying operator experience levels and attitudes to control. In addition, it is typical that only one input gets tweaked, whereas, in truth, a number of inputs affect the various process outputs in some multivariable, cross-correlated manner.

A need is thus identified for an automated R2R control solution that is self-adapting to optimized control of a process. In this chapter we address the dual “optimization-control problem” by introducing a multiple input, multiple output (MIMO) R2R controller that can act as an optimizer and/or a controller. Specifically, in the first part of this chapter we briefly discuss the original “linear” model control approach to R2R where model development is done off-line using traditional experimental design. This approach, which is detailed in Chapter 3, has been proven effective, and a commercial application is mentioned.

We then present a solution to the dual optimization-control problem by introducing a multiple input, multiple output (MIMO) R2R controller that can act as an optimizer and/or a controller. This controller — the optimizing adaptive quadratic controller (OAQC) — can develop equipment models on-line if none are available (see also Chapter 4). Once a satisfactory model is determined, OAQC will switch to R2R control mode. The OAQC can also be used in either pure optimization or in pure batch control mode.
At the conclusion of this chapter we introduce a commercial integration platform control called the generic cell controller (GCC), which enables “plug-and-play” of production R2R control solutions such as OAQC solutions. The GCC is described in Part 3 of this book, and the integration of the OAQC in the GCC integration platform is presented in Chapter 12.

5.2 “LINEAR” R2R CONTROL

The evolution of linear constrained multiple input, multiple output (MIMO) R2R control is reviewed in Del Castillo and Hurwitz.\(^5\) In its simplest form, for a single input, single output (SISO) system, a common response model is

\[
y(x) = \alpha + \beta x + \delta t + \epsilon,
\]

where \(y\) is the estimated response, \(\alpha\) and \(\beta\) are parameters, \(x\) is the level of the controllable factor, \(\delta t\) is a deterministic drift with \(t\) being time, and \(\{\epsilon_t\}\) a white noise sequence with variance \(\sigma^2\). This model — called a gradual mode model — is used as a local approximation to a drifting, possibly nonlinear, response surface as shown in Figure 5.1.

So-called EWMA controllers\(^{11}\) model \((\alpha + \delta t)\) as \(c\) and update it using an exponentially weighted moving average, i.e.,

\[
c_t = \lambda (y_t - \beta x_t) + (1 - \lambda)c_{t-1}
\]

The basic mechanics of this simple R2R control scheme is as follows: It is desired that the output \(Y\) equals target \(T\) which occurs if the controllable input is set at \(X\) since \(y = c + bX\), where \(y\) estimates \(E[Y]\), \(b\) is the estimated linear slope, and \(c\) the estimated \(Y\)-intercept (Figure 5.2). If there is evidence of a process shift, \(c\) is reestimated using the above EWMA method (Figure 5.3). \(X\) is then recalculated to give \(X'\) so that \(Y\) again equals its target \(T\), that is, \(X' = (T - c)/b\) (Figure 5.4).

**FIGURE 5.1** “Gradual mode” linear model.
The parameter $b$ is typically estimated off-line using formal designed experiments, such as Box–Behnken or two-level factorial designs. The parameter $b$ then is assumed fixed. This leads to a control scheme that appears, at first sight, to be very simple. However, any commercial realization has to go much further.

Boning et al. pointed out some practical issues that the design of a R2R controller should consider. Among these are the importance of considering input constraints; the weighting of inputs (as some may be required to vary less frequently than others); the weighting of outputs (as some outputs are more important than others); the resolution or “granularity” of the input settings; and accountability of uncontrollable inputs.

A solution for the above has been found and incorporated into a commercial controller. This controller is installed in the most recent three-table CMP tool in the semiconductor industry (see Figure 5.5).

FIGURE 5.2 Input $X$ gives $E[Y]$ at target $T$.

FIGURE 5.3 $C$ is reestimated if process shifts.

FIGURE 5.4 $X$ is recalculated so that $Y = T$.

FIGURE 5.5 A three-table CMP tool.

The parameter $b$ is typically estimated off-line using formal designed experiments, such as Box–Behnken or two-level factorial designs. The parameter $b$ then is assumed fixed. This leads to a control scheme that appears, at first sight, to be very simple. However, any commercial realization has to go much further.

Boning et al. pointed out some practical issues that the design of a R2R controller should consider. Among these are the importance of considering input constraints; the weighting of inputs (as some may be required to vary less frequently than others); the weighting of outputs (as some outputs are more important than others); the resolution or “granularity” of the input settings; and accountability of uncontrollable inputs.

A solution for the above has been found and incorporated into a commercial controller. This controller is installed in the most recent three-table CMP tool in the semiconductor industry (see Figure 5.5).

* By MiTeX Solutions, Inc. of Ann Arbor, MI, in the Strasbaugh, Inc. “Symphony” planarizer.
5.3 NONLINEAR R2R CONTROL

If the process response surface is fairly “flat” in the region of control model approximation, then a linear controller of the type given above should be expected to work well. Experience in CMP, epitaxial deposition, and etch has borne this out. However, there exist processes where response surface nonlinearities are features. Even processes that exhibit linearity over one region of operation might easily show nonlinearity when the operating window is moved to a different part of the response space.

An illustration of a model nonlinear in the inputs is a “Hammerstein” model. A SISO example of this type is

\[ y_t = ax_{t-1}^2 + bx_{t-1} + c + \varepsilon_t \]  

(3)

This model is linear in its parameters, and may thus be estimated by a least-squares technique such as recursive least squares (RLS). Another approach to a nonlinear R2R control solution is to use neural nets for the response model. This will work if enough data are available to construct the neural weightings. In production processes with many different regimes/recipes, it is not always easy to get sufficient data for neural modeling, and a more parsimonious approach is useful. One such approach devised by Del Castillo and Yeh and called the “optimizing adaptive quadratic controller” (OAQC) is now described.

The OAQC assumes that equipment behavior can be modeled according to a second-order MIMO Hammerstein model of the form

\[ y_t = y(0) + Nz_{t-1} + Mt + (I_p - CB)\varepsilon_t \]  

(4)

where \( z_t = (u_t, u_t^2, u_t^{(i)}u_t^{(j)}, i < j) \) is a vector of length \( 2n + (n(n-1)/2) \) that contains the quadratic expansion of \( u_t \); \( y_t \) is a \( p \times 1 \) vector of quality characteristics; \( y(0) \) is a \( p \times 1 \) vector of intercepts; \( u_t \) is a vector of controllable factors; \( t \) denotes a vector containing the time index \( t \) in its \( p \) components, \( \{\varepsilon_t\} \) is a sequence of multivariate white noise random vectors; and \( B \) is the one-lag backward shift operator. This model is general enough for most equipment response surfaces.

A minimum means square error forecast developed by Del Castillo may be applied to the model given in (4) to give a forecast equation:

\[ \Upsilon_{t+1|t} = Ly_t + M(t+1) + Nz_t \]  

(5)

where \( \Upsilon_{t+1|t} \) is the one-step-ahead forecast of \( y_t \), and the \( (p \times 2n + (n(n-1)/2)) \) matrix \( N \) contains parameters for both first-order and quadratic terms. The quadratic expansion provides a second-order polynomial approximation to the system nonlinearity. Online parameter estimates of \( L, M, \) and \( N \) are provided by a recursive least-squares (RLS) algorithm, and then (5) may be used to derive the control rule for the current run.
The OAQC uses a multivariate version of the Clarke and Gawthrop\textsuperscript{3} one-step controller performance index,

\[ J = \left( \bar{Y}_{r+1|F} - T \right)' W \left( \bar{Y}_{r+1|F} - T \right) + (u_r - u_{r-1})' \Gamma (u_r - u_{r-1}) \]  

(6)

which is minimized subject to the input and output constraints:

\[ L_u \leq u_r \leq U_u \]  

(7)

\[ L_y \leq Y_{r+1|F} \leq U_y \]  

(8)

where \( T \) denotes a \( p \times 1 \) vector of targets, \( W \) is a \( p \times p \) diagonal matrix with entries representing the relative priorities of each response, and \( \Gamma \) is a \( n \times n \) diagonal matrix of cost coefficients — the larger a cost coefficient, the closer the factor \( i \) will remain to its previous setting. \( J \) is minimized with respect to the controllable factors \( u_r \).

### 5.4 THE OAQC NONLINEAR CONTROLLER

The traditional form of R2R control as described above proceeds according to the following three steps: (1) an experimental design based on some design optimality criterion is performed, (2) a model is fitted using some form of minimization (e.g., least squares), and (3) a search-and-locate optimization step is done to find an “optimal” process recipe. The OAQC attempts to address these three steps simultaneously (rather than sequentially) in an incremental way at each run, thereby speeding up the qualification of the process and reducing process disruptions.

For ease of presentation, assume that there is a single response \( Y \) that is dependent on \( n \) controllable factors. It is desired to keep \( Y \) as close as possible to target \( T \). The exact input–output relationship is, however, unknown. The OAQC solves the following problem at each run:

\[
\min \left( (1 - \lambda_r) \left( \bar{Y}_{r+1|F} (x_{r+1}) - T \right)^2 - \lambda_r \left( \text{Var} Y_r (x_{r+1}) / \sigma^2 \right) \right) 
\]

(9)

where the minimization occurs as \( \{ x_{r+1} \in \Omega_r \} \), a region of factor space defined by input factor bounds, and \( \bar{Y} \) is the estimated response.

The first term of (9) is the square deviation of the one-step-ahead predicted response, and the second term is the scaled variance of the current estimated response. The constants \( \{ \lambda_r \} \) are such that \( 0 \leq \lambda_r \leq 1 \), and give relative weights to the two objectives in (9), namely the optimization of the response (first term) and model fitting (second term). Note that if \( \lambda_r = 1 \), then the second term is equivalent to a conditional D-optimality,* a useful criterion in the presence of factor bounds.

* D-optimality \( \Rightarrow |X'X| \) is maximized \( \Rightarrow \text{Var} (\bar{Y}) \) minimized, where \( X = (x_j) \), the matrix of \( j = 1, \ldots, n \) factor inputs over \( i = 1, \ldots, k \) trials.
At each run the solution to problem (9) provides a new set of factor settings $x_{t+1}$ to try next. Initially, optimization of the objective function (9) will tend to favor factor settings that will lead to good response surface models as opposed to factor settings that achieve desired response performance. Learning proceeds via estimation of the response parameters using a multivariate recursive least squares (RLS) algorithm.  

5.5 OAQC: SIMULATED TRIALS

The OAQC software has a simulation facility that is used to demonstrate optimization and control against “open loop” (i.e., no control) in two different settings. In both trials a three-input, two-output quadratic CMP process model with linear drift is installed in the simulator as the “truth.” This model is given by Khuri. 

\[
y_1 = 2756.5 + 547.6u_1 + 616.3u_2 - 126.7u_3 - 1109.5u_1^2 - 286.1u_2^2 + 989.1u_3^2 - 52.9u_1u_2 - 156.9u_1u_3 - 550.3u_2u_3 - 10t + \epsilon_{1t}
\]

\[
y_2 = 746.3 + 62.3u_1 + 128.6u_2 - 152.1u_3 - 289.7u_1^2 - 32.1u_2^2 + 237.7u_3^2 - 28.9u_1u_2 - 122.1u_1u_3 - 140.6u_2u_3 - 1.5t + \epsilon_{2t}
\]

where $\epsilon_{1t} \sim N(0,60^2)$ and $\epsilon_{2t} \sim N(0,30^2)$. The $u_i$ were constrained to the $(-1,1)$ range. $y_1$ is removal rate, $y_2$ is within-wafer nonuniformity.

The first trial introduced as a given (prior), a fairly good initial model for the controller to start with. The second trial started with no prior model at all. The model given in the first trial was

\[
y_1 = 2500 + 400u_1 + 500u_2 - 100u_3 - 800u_1^2 - 200u_2^2 + 1000u_3^2 - 40u_1u_2 - 100u_1u_3 - 100u_2u_3 - 7t
\]

\[
y_2 = 600 + 50u_1 + 100u_2 - 100u_3 - 200u_1^2 - 50u_2^2 + 300u_3^2 - 30u_1u_2 - 100u_1u_3 - 100u_2u_3 - 3t
\]

In the first trial, a target of 2700 was set for the output $y_1$, while a constraint on $y_2$ between $-100$ and 700 was set as a goal. Both optimization and control were requested. Fifty (50) simulation runs were executed.

The graph of Figure 5.6, with “target” lines at 700 and 2700 added, shows the results for $y_1$ and $y_2$ (the outputs resulting from optimization and control) as well as the open loop or “uncontrolled” outputs. As can be seen, after the optimization phase ended — at about run 13 — the controlled outputs stayed closer to the target (or upper constraint in the case of $y_2$) than the uncontrolled outputs.
The second trial was executed for 50 simulated runs and in all details was identical to the first trial except that no prior model was specified, and $y_1$ was given no target, but desired constraints of $2700 \leq y_1 \leq 100,000$ were imposed. As can be seen again, in Figure 5.7, once the optimization phase is over, the controlled responses better match their constraints than for the open responses.

These graphical intuitions may be confirmed by examining a table of summary statistics (Table 5.1) for the trial runs (excluding the first 15 “optimization” runs in all cases). The means ($\mu$) of the controlled outputs ($y_1^c$, $y_2^c$) are closer to or more consistently within the limiting constraints than are the open responses ($y_1^o$, $y_2^o$).

There is not much to choose between when examining the response standard deviations ($\sigma$) except that in the case of $y_2$, the open responses have smaller variations. This is probably caused by the control action itself pushing against difficult-to-accomplish constraint limits due to the rather severe nature of the assumed trends.

The mean square deviation (MSD) for the $y_1$’s is taken as the square root of the average of $\Sigma(y_1^* - 2700)^2$, and again for $y_2$ but using the sum $\Sigma(y_2^* - 700)^2$, where “*” is either $c$ for controlled or $o$ for open loop. In all cases the controlled responses show improvement over the uncontrolled responses.

FIGURE 5.6 3 × 2 R2R simulation with prior model, $y_1$ target = 2700, $y_2$ constraints.

FIGURE 5.7 3 × 2 R2R simulation with no prior model, $y_1$ constraints and $y_2$ constraints.
In trial #1, the $y_1^c$ mean of 2570 is closer to the target (2700) than the mean for $y_1^o$. The $y_2^c$ mean of 713 is also closer to the constraint (700) than the mean for $y_2^o$. MSDs in both cases are also smaller for the controlled than the open case. For trial #2, control performance remains superior to open loop except, as before, in the case of $\sigma$ for $y_2$.

It should be noted that in the Khuri model, all signs in both model equations are the same for matching terms except for the signs of the trend terms. This makes the system inherently more difficult to control: a few differing signs would give added latitude to the controller in the face of opposing drifts.

## 5.6 CONTROLLER IMPLEMENTATION

A step in the commercialization path of any software is some means to embody it within a robust factory control framework. We note that Moyne and McAfee\(^9\) have given a structure for generic cell controller technology that is ideal for this type of R2R control, and that has also been commercialized.\(^*\) The GCC technology is detailed in Part 3 of this book (specifically Chapters 9 through 11). An integration path to ultimate system linkage has thus been well established in addition to the control theoretic foundation. An integrated GCC-enabled solution is shown in Figure 5.8. The GCC system accepts pre- and postmetrology process inputs and performs process optimization, modeling and/or control depending on the capabilities provided by the optimization/control algorithm incorporated in the particular GCC solution. As detailed in Chapter 12, an OAQC solution has been incorporated into a GCC system, thereby providing the adaptive control capabilities described in this chapter in an industrial quality solution.

---

* See [http://www.mitexsolutions.com](http://www.mitexsolutions.com).
5.7 CONCLUSIONS

Run-to-run control is now an industrially proven batch control strategy that has a considerable body of theoretical work behind it as well as a tested integration path. In this chapter we have shown that R2R control can be extended to nonlinear systems, and to systems about whose response patterns little or nothing is known, by the use of the optimizing adjusting quadratic controller. This approach integrates process model estimation, model updating, and recipe optimization in one seamless on-line entity.

ACKNOWLEDGMENTS

The authors are grateful to Strasbaugh, of San Luis Obispo, California, for permission to use diagrams appearing on their commercial Web site. Thanks are also due for the helpful comments of Dr. James Moyne of the University of Michigan. An earlier version of the material presented in this chapter was presented at the 1999 International Conference on Quality Manufacturing.

REFERENCES

6 A Comparative Analysis of Run-to-Run Control Algorithms in the Semiconductor Manufacturing Industry

Zhe Ning, James R. Moyne, Taber H. Smith, Duane S. Boning, Enrique Del Castillo, Jinn-Yi Yeh, and Arnon M. Hurwitz

6.1 INTRODUCTION

In Chapters 3 through 5 we described a number of R2R control algorithms. This chapter provides a comparative analysis of R2R control algorithms, focusing on their ability to provide robust and stable control in the face of linear and quadratic drift. This chapter is not meant to be an exhaustive analysis of algorithm robustness and stability, but rather an illustration of the pros and cons of the various R2R algorithm candidates, and a description of a methodology for comparative evaluation of R2R control algorithms. Specifically, the chapter is organized as follows. Following this introduction, background information is provided on each of the algorithms that are investigated, as well as the multialgorithm R2R control solution enabler. The testing benchmarks are then presented, followed by simulation results of the application of the algorithms to these benchmarks. This chapter concludes with a comparative analysis and a discussion of additional tests that could be performed to further qualify the algorithms.

6.2 BACKGROUND

The apparatus required for testing multiple R2R control algorithms and realizing multiple algorithm control solutions necessarily includes the algorithms themselves as well as the multialgorithm control system enabler. In the following subsections, summary information on each of these elements is provided for completeness. Note that more detailed treatment of the polynomial-based R2R control algorithms can be found in Chapters 3 through 5, while the multialgorithm solution is detailed in Chapter 9.
6.2.1 Run-to-Run Control Algorithms

6.2.1.1 The EWMA “Gradual Mode” R2R Controller

The EWMA “gradual mode” controller (GM) is a linear approximation model-based controller with an exponential weighted moving average (EWMA) “forgetting factor” that provides for control of noisy processes.\(^3\)\(^4\) The GM begins with a linear model of the process,

\[
y_t = Ax_t + c_t, \tag{1}
\]

where \(y_t\) = vector of \(n\) process outputs
\(x_t\) = vector of \(m\) process inputs (e.g., machine settable parameters)
\(c_t\) = vector of \(n\) constants
\(A\) = an \(n\) by \(m\) matrix (of first-order coefficients)

The controller updates the offset term \(c_t\) on an R2R basis using an EWMA approach:

\[
c_t = \alpha (Y_{t-1} - Ax_{t-1}) + (1-\alpha) c_{t-1}, \tag{2}
\]

where \(0 \leq \alpha \leq 1\). By adjusting \(c_t\) in (1), the GM is able to track and compensate for gradual changes in the process as well as filter out random talk noise in the process.

A controller “module” implementation of the GM was developed for integration into R2R control systems. In order to make the GM implementation practical on the factory floor, a number of features have been added to accommodate control of typical semiconductor processes. These features include (1) recipe parameter bounding to force the controller to search for a control solution in a user-definable recipe region, (2) input weighting to allow the user to trade off recipe advices between different inputs, (3) output weighting to allow the user to provide preferential control for important outputs, (4) an input discretization capability to accommodate the discrete nature of equipment settings, and (5) controller tuning parameters to provide maximal controller performance in the particular controller environment. Note that all of these features are dynamically settable in the implementation developed.\(^5\)

6.2.1.2 The Time-Based EWMA R2R Controller (GMt)

The GMt controller extends the GM controller to more accurately provide control in environments where some or all of the process quality measures (outputs) cannot be accurately approximated as a linear function of the process inputs.\(^4\) As an example, consider the following multivariate process control problem:

\[
(\text{Amount Removed, Nonuniformity}) = f (\text{Time, Speed, Pressure}) \tag{3}
\]

A linear solver cannot accurately model situations where an output parameter such as Amount Removed is a function of an input parameter such as Time multiplied by a linear function of other input parameters (i.e., Speed and Pressure).
The GMt provides a more accurate model and controller for systems of this type by breaking the system outputs down into two sets, namely primary outputs, which are directly impacted by primary inputs, and secondary outputs, which are functions of primary output(s) and secondary inputs. The GMt then provides control by implementing a two-step linear solution process that approximates a nonlinear solver. In the first step, the controller uses the EWMA GM method (as shown in (1) and (2)) to compensate fully for those primary outputs that do not have corresponding secondary outputs, and to partially compensate (in a weighted fashion) for those primary outputs that also have corresponding secondary outputs. Thus, for the system described in (3), the GMt would first compensate for Removal Rate in part and Nonuniformity in full by adjusting Speed and Pressure (in the manner of (1) and (2)), and then (in the second step) adjust Time to compensate the rest of the way for Amount Removed.

As with the GM, a controller “module” implementation of the GMt has been developed for integration into R2R control systems that contains all of the “industrial quality” features implemented in the GM application.

6.2.1.3 The Knowledge-Based Interactive Run-to-Run Controller

The knowledge-based interactive run-to-run controller (KIRC) is a machine learning algorithm for R2R control that uses leaves in a classification decision tree to suggest control actions for process improvement. KIRC generates nonlinear response surfaces from experimental data using neural networks. Points are calculated on each response surface and discretized into output classes to form training examples. The size of the outputs classes is typically determined by statistical process control limits. The GID3 algorithm generates a decision tree by using an information entropy metric to recursively partition the operating space with attribute tests. The starting operating point is chosen from the largest leaf in the decision tree where all outputs are inside the target range. As gradual drifts or sudden changes occur in the process, the output values may leave the region of the target leaf. The decision tree is searched for a neighboring leaf that matches the current output classification. By comparing the attribute ranges of (1) the leaf matching the new process state and (2) the original target leaf, a control action is computed to move the process back in the target output region.

Most of the work with KIRC has focused on compensating for process shifts caused by disturbances during a reactive ion etching process in a hexode reactor. An EWMA drift compensation component has recently been added to the algorithm so that control actions can be taken before a predicted gradual drift moves the process outside the control limits.

6.2.1.4 The Optimizing Adaptive Quality Controller (OAQC)

The optimizing adaptive quality controller (OAQC) is designed to seek and maintain optimum operating conditions for a multiple input, multiple output (MIMO) process. The controller is to be used in a run-to-run manner. Work on this controller originated from previous developments in self-tuning MIMO controllers.
The OAQC is comprised of the following elements:

1. A recursive multivariate least-squares estimation algorithm.
2. A nonlinear constrained optimization algorithm.
3. Various utilities, including experimental designs for fitting quadratic responses and obtaining initial estimates, multivariate SPC charts that can be added to the controller and act as deadbands or be used to supervise the recursive estimation procedure, diverse tuning functions that allow the process engineer to modify the controller, history plots for inputs and outputs, and discretization of the inputs.

The basic idea of an optimizing controller is to provide on-line estimation of a nonlinear (in the controllable factors) regression model (a Hammerstein model) that is optimized in order to obtain the best control action. Once optimal operating conditions have been achieved, the OAQC will seek to maintain the process under control running at such conditions. Therefore, the OAQC acts both as an “optimizer” and as a “controller.” If equipment models are available from previously conducted off-line experiments, the corresponding parameter estimates can be entered into the OAQC to speed up the optimization phase.

**Figure 6.1** shows a block diagram of the multivariate controller. The controller contains a multivariate recursive estimation algorithm to update the model parameters. Also, and most important from a practical point of view, the proposed OAQC explicitly considers input and output constraints.

### 6.2.2 Multiple Algorithm System Enabler

#### 6.2.2.1 The Generic Cell Controller (GCC)

The Generic Cell Control (GCC) is a discrete control enabling mechanism that utilizes distributed object-based communication and a relational database (as
opposed to procedural code) to store and execute sequential control information. The theory of operation of the GCC is documented in literature.\cite{9,10}

Because the GCC provides for modular and configurable control solutions, it is well suited as an enabler for R2R control implementations that make use of multiple control algorithms operating in a complementary fashion.\cite{1,11} An example of such an implementation is shown in Figure 6.2. Note that the implementation contains a rule-based decision-making module that utilizes metrology information along with process history and a rule base to determine the optimization or control algorithm(s) to invoke for each run. Note further that the rule base contains rules that identify the domain of applicability of the various algorithms; information on these domains (such as that presented in this chapter) could be used to develop these rules.

### 6.3 ALGORITHM TEST BENCHMARKS

There is a large variety of multivariate processes in the semiconductor industry that are candidates for R2R process control. These processes exhibit many qualities over their input parameter space of operation as well as over time (i.e., from run to run), that could impact (and be impacted by) an R2R controller. For example, the process response surfaces may be linear or nonlinear. Further, the processes may be subject to different drifts, noise, and other disturbances. Thorough testing of R2R algorithms for the semiconductor manufacturing industry would then necessarily include an exhaustive benchmarking of the algorithms to these process qualities. As a first step, results are presented on algorithm operation against the following two benchmarks:

1. A linear process with linear drift buried in white noise.
2. A full quadratic, severely nonlinear process with linear drift buried in white noise.

As a test vehicle, process models originating from a CMP process are used as an example process. These models each have four inputs (Platen Speed, Back-Pressure, Polish Head Downforce, and Profile) and two outputs (Removal Rate and
Nonuniformity). Factors are included in the models to account for process noise and run-to-run process drift. No discretization limits in the inputs are imposed.

Specifically, the linear process utilized has the form

\[ y[n] = C + Ax + \omega[n] + \delta n, \]  

where \( n \) is run number and \( \omega[n] \) is normally distributed white noise with zero mean and covariance matrix \( \Lambda \). We will use

\[
C = \begin{bmatrix}
-1382.60 \\
-627.32
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
50.18 & -6.65 & 163.4 & 8.45 \\
13.67 & 19.95 & 27.52 & 5.25
\end{bmatrix}
\]

\[
\delta = \begin{bmatrix}
-17 \\
1.5
\end{bmatrix}
\]

\[
\Lambda = \begin{bmatrix}
665.64 & 0 \\
0 & 5.29
\end{bmatrix}
\]

The second-order process has the form

\[ y[n] = C + f(u[n]) + \omega[n] + \delta n, \]  

where drift and noise are the same as in (4) and \( f(u[n]) \) is a full second-order polynomial function of the inputs with the following form:

\[ f(u[n]) = \sum_{i=0}^{3} \sum_{j=0}^{3} \beta(i,j)u(i)u(j) \]

We will use

\[
\beta = \begin{bmatrix}
1386.5 & 381.02 & -112.19 & 3778.8 & -21.301 & 8.7159 & 24.953 \\
1520.8 & 2365.6 & 2923.5 & 281.66 & -3.9419 & -1.0754 & 1.406 \\
37.082 & -17.642 & -11.974 & -164.99 & 28.150 & 249.17 & 0.025067 \\
0.33797 & -72.274 & -94.222 & -26.175 & -13.505 & 36.691 & 32.929
\end{bmatrix}
\]

In both cases, the targets are 1700 Å for Removal Rate and 150 Å for Nonuniformity. For simplicity, only Removal Rate control is evaluated in our comparison.
The process designed to test the performance of the GMt controller against the GM controller has the following form:

\[ y[n] = C + A x + \omega[n] + \delta n, \quad \text{and} \]

\[ \text{Amount Removed} = \text{Removal Rate} \times \text{Time}, \]

where Removal Rate is one of the outputs in the vector \( y[n] \) and noise and drift terms in (6) are the same as in (4). The outputs for this process are Amount Removed and Nonuniformity with targets of 1700 Å and 150 Å, respectively.

### 6.4 TEST RESULTS

Given the test benchmarks described in the previous section, controlled responses from different control algorithms (introduced in Section 6.2) are presented in this section. As a quantitative comparison metric, we use a weighted mean-squared error (WMSE). This quantity is computed as the squared error of each output from target summed over the run, divided by the respective process targets and total run number, and summed together.

#### 6.4.1 CONTROL OF THE LINEAR PROCESS

Figure 6.3 shows the controlled responses (Removal Rate) from GM, OAQC, and KIRC controllers in controlling a linear process with drift and noise. The weighted mean-squared errors (for 30 runs) were \( 3.2 \times 10^{-2} \) for GM controller, \( 2.50 \times 10^{-2} \) for OAQC and \( 2.52 \times 10^{-2} \) for KIRC, indicating that all three algorithms kept very good control of the process with linear drift.

#### 6.4.2 CONTROL OF THE QUADRATIC PROCESS

Figure 6.4 shows the controlled responses (Removal Rate) from GM, OAQC, and KIRC controllers in controlling a full quadratic (nonlinear) process with linear drift and noise. Here we see that the OAQC, which is based on a nonlinear model, keeps good control of the process, while both GM and KIRC lost control of the process after eight runs. The WMSE for OAQC was \( 3.67 \times 10^{-2} \).

#### 6.4.3 GMt CONTROLLER VS. GM CONTROLLER

Controlled responses for the process given in (6) and (7) (Section 6.3) are displayed in Figure 6.5. Both the GMt and GM controllers were run 40 times on the process.

\[
C = \begin{bmatrix} -138.21 \\ -627.32 \end{bmatrix}
\]

\[
A = \begin{bmatrix} 5.018 & -0.665 & 16.34 & 0.845 \\ 13.67 & 19.95 & 27.52 & 5.25 \end{bmatrix}
\]
The weighted mean-squared errors were $2.434 \times 10^{-2}$ for the GMt controller and 12.521 for the GM controller. As can be seen from the figure, the GM controller lost control of the process after 36 runs due to its linearized approximation of the nonlinear process, while the GMt controller maintained a good control of the process during the entire period of runs by providing a two-step linear solution.

**FIGURE 6.3** Control of linear process.
A review of the test results presented in Section 6.4 indicates that all three algorithms (GM, OAQC, and KIRC) provide good control of the simulated linear process (with linear drift and white noise); the differences between the control provided are minimal. With the simulated full-quadratic non-linear process, both the GM and KIRC controllers were unable to provide control beyond the first few runs due to the severe non-linearity in this test case; however, the OAQC provided control for the duration of the simulated process, with a relatively small WMSE.

The results presented in Section 6.4.3 illustrate the utility of the GMt control approach for a class of non-linear systems. As illustrated in Figure 6.5, the GMt is able to provide control for some processes where the GM controller eventually fails.

**FIGURE 6.4** Control of quadratic (non-linear) process.

### 6.5 COMPARATIVE ANALYSIS

A review of the test results presented in Section 6.4 indicates that all three algorithms (GM, OAQC, and KIRC) provide good control of the simulated linear process (with linear drift and white noise); the differences between the control provided are minimal. With the simulated full-quadratic non-linear process, both the GM and KIRC controllers were unable to provide control beyond the first few runs due to the severe non-linearity in this test case; however, the OAQC provided control for the duration of the simulated process, with a relatively small WMSE.

The results presented in Section 6.4.3 illustrate the utility of the GMt control approach for a class of non-linear systems. As illustrated in Figure 6.5, the GMt is able to provide control for some processes where the GM controller eventually fails.
A comparative analysis of four algorithms (GM, GMt, OAQC, and KIRC) has been presented in this chapter. The algorithms were evaluated in the control of both linear and non-linear processes that experienced linear R2R process drift as well as white noise. The results indicate that all algorithms were able to provide good control of the linear test process, while only the OAQC was able to provide control of the severely non-linear process. (There are many classes of non-linear drift processes for which the GM is able to provide a level of control.) Additionally, it has been shown that the GMt provides good control for a class of non-linear control problems such as those where the measured output is related to an intermediate process variable by a multiplicative factor such as time.

Although the test results presented here help to identify the domains of acceptable operation of the tested algorithms, they represent only a first step in determining the suitability of a control algorithm to a particular process. There are many other factors that must be considered before making the final evaluation of the control algorithm. For example, the non-linear model utilized in the test process represents only one case of non-linear control; the specifics of the non-linear process and the region of operation within the process model will have a significant impact on the suitability of the control algorithm. Also, the characteristics of the process noise may impact algorithm performance. Further, the impact of model error should be evaluated. Finally, the maturity of the algorithm implementation, capability for integration, and features (such as bounding of inputs and other practical factors)

![GM Control of Non-Linear Process](image1)

![GMt Control of Non-Linear Process](image2)
should be considered in any comprehensive evaluation. In summary, a complete algorithm solution comparative evaluation should address the qualification of the controllers with respect to factors such as non-linear process drift, various noise patterns, controller model error, and usability in an industrial setting.

ACKNOWLEDGMENTS

Much of the material presented in this chapter is derived from Reference 1, and is reprinted with permission. The authors gratefully acknowledge the contributions of William Moyne and Victor Solakhian in developing the GM, GMt, and GCC software, and Matt Hankinson for developing the KIRC software and providing KIRC algorithm test results.

REFERENCES

Part 3

Integrating Control

In Part 1 we noted that integration and automation innovation are key factors that pushed R2R control to the forefront ahead of in situ control and interprocess control in the movement toward advanced process control (APC) implementation. Indeed, the integration of control is often the most overlooked challenge in pursuing APC development and deployment. Attention must be paid to integration aspects throughout the APC process or the control solutions developed will be too costly or cumbersome to be useful.

In Part 3 of this book we explore the methods and solutions for integrating control. Information presented in this part is useful to anyone trying to implement and/or integrate R2R control. It addresses issues such as solution robustness and portability, design for flexibility and future enhancement, and identification of metrics of capability and success. Specifically, in Chapter 7 we define the existing and future-envisioned control environment in semiconductor manufacturing that might incorporate R2R control. This environment is defined in terms of specifications from three sources that are shaping the control integration environment for the industry: (1) the Semiconductor Industry Association and the SIA roadmap, (2) Semiconductor Equipment and Materials International (SEMI) and SEMI standards, and (3) SEMATECH and the SEMATECH Control Systems Requirements Specification (CSRS) output. For each source, the specific impact on R2R control integration is discussed. Following this discussion we identify the design requirements for an integrateable* R2R solution in Chapter 8. We then turn our attention in Chapter 9 to a specific R2R control enabler design and solution, called the Generic Cell Controller (GCC), which conforms to these design requirements. We include a

* Both “integrateable” and “integrable” are commonly utilized in the industry to mean “capable of integration.”
description of the GCC concept, provide a straightforward example of a GCC implementation, and discuss the utilization of the GCC flexible control system concept to realize a multialgorithm R2R control solution.

The information in Chapters 7 through 9 presents external and internal R2R controller design requirements, and provides examples of enabling technologies. However, it doesn’t suggest a complete design for an integrated R2R control solution. We address this problem in Chapter 10 by presenting a design for a “piggyback” R2R control solution that conforms to the integration requirements identified in Chapter 8. This is followed, in Chapter 11, by two examples of R2R control solution integration: (1) a “piggyback” tool retrofit solution, and (2) a fully integrated solution for a next-generation tool. These example presentations include descriptions of integrated solution design and software user interface, methodology for deployment, and evaluation in terms of process improvement. One other important aspect for R2R control solution development and integration is the development of the control algorithm solution to the control environment. This issue is addressed in Chapter 12 with a case study of the design and implementation of a GCC-enabled optimizing adaptive quality controller R2R control solution.
The industry as a whole has been pursuing the identification, specification, and standardization of control integration for semiconductor manufacturing along many fronts. The three major players in this arena are the Semiconductor Industry Association (SIA), Semiconductor Equipment and Materials International (SEMI), and SEMiconductor MANufacturing TECHnology (SEMATECH). The SIA is an organization of leaders in the semiconductor manufacturing industry. Members of the SIA have been instrumental in the process or addressing the technology needs of the industry by establishing precompetitive partnerships and consortiums such as the Semiconductor Research Corp. (SRC) in 1982, SEMATECH in 1987, and the International 300-mm Initiative (I300I) in 1996. The SIA membership also publishes a National Technology Roadmap for Semiconductors, which will be discussed further in Section 7.1. SEMI is a global trade association, founded in 1970, that represents the semiconductor and flat panel display equipment and materials industries. With over 2000 members, the primary goal of SEMI is to help its members expand their global marketing opportunities and relationships by providing industry-specific information and educational resources. SEMI is the primary standards organization for the semiconductor industry. SEMATECH is a nonprofit technology development consortium originally created to reinvigorate the U.S. semiconductor industry, but it has evolved since 1998 into an international research consortium.

In this chapter we provide a discussion of significant contributions to control integration by the Semiconductor Industry Association (SIA Roadmap), SEMI (standards), and SEMATECH (Control Systems Requirements Specification).

7.1 THE SEMICONDUCTOR INDUSTRY ASSOCIATION AND THE SIA ROADMAP

The SIA has provided direction for research, development, and deployment of control solutions in semiconductor manufacturing by presenting a roadmap for advancement. The National Technology Roadmap for Semiconductors is a 15-year projection of
the integrated circuit technology characteristics required to maintain the historical rate of performance and cost improvements. In providing this projection along with an analysis of the current state of the art, the roadmap identifies the research needs that must be fulfilled to realize the required technology advancements. The roadmap addresses the many aspects of semiconductor manufacturing from front-end processes through assembly and packaging.

Run-to-run control technology is addressed in the roadmap as a component of factory integration. The roadmap clearly identifies R2R control as a technology whose development is underway and whose full deployment should occur by the year 2003. The roadmap also identifies R2R control as an integral component of a larger advanced process control (APC) framework. Developers should continue to look to the SIA roadmap for indications of future directions in R2R control technology development, deployment, and integration.

7.2 SEMICONDUCTOR EQUIPMENT AND MATERIALS INTERNATIONAL AND SEMI STANDARDS

Semiconductor Equipment and Materials International (SEMI) has been pursuing standards in the area of communication and integration ever since the development of the SEMI Equipment Communication Standard (SECS) in the late 1970s. From that time to the present, SEMI experts have focused on defining communication mechanisms between the various elements of the control and operations hierarchy in the fabrication facility. Figure 7.1 is an illustration of various SEMI standards and SEMI standards efforts mapped into the logical semiconductor facility hierarchy. Note that this figure includes existing standards supported by SEMI as well as

---

**FIGURE 7.1** SEMI standards and standards efforts mapped into control hierarchy.
working standards specifications under development by SEMI members. The following is a brief description of SEMI communication control standards relevant to the integration of R2R control.4

SEMI Equipment Communications Standard (SECS — E4 and E5)*: SECS is a standard for communications between intelligent equipment and a host. The standard has two parts that define the communication protocol interface (SECS-I) and the messages exchanged (SECS-II). SECS-I, which specifies point-to-point communications over a (slow) R2-232 interface, is somewhat obsolete; high-speed message service (HSMS) is a high-speed Ethernet-based replacement for SECS-I, but retains many of the deficiencies of SECS-I (e.g., point-to-point only communications, host-equipment master/slave only operation, etc.). The Computer Integrated Manufacturing (CIM) framework specification and Object Based Equipment Model (OBEM) point to future alternatives to SECS (see below).

Generic Model for Communications and Control of SEMI Equipment (GEM — E30): The GEM standard defines the generic behavior of semiconductor equipment as viewed through a communications link in terms of SECS-II messages communicated over that link. The GEM standard impacts equipment control and equipment to “host” communications. Note that GEM does not comply with the distributed object-based control description paradigm; this paradigm is being pursued through OBEM, which may become a replacement for GEM (see below). Recently, the GEM standard has been enhanced to include an application note defining a methodology for process parameter update.5 This enhancement is important to the standardization of R2R control implementation because it defines a common way for providing R2R control advice information to a tool.

Object-Based Equipment Model (OBEM — under review)**: The OBEM specification defines concepts, behavior, and message services to realize equipment control systems. The OBEM standard defines the operation of the equipment control systems, including utilization of sensor actuator network (SAN)-compliant device-level systems and visibility to higher level control systems. Specifically at the equipment control level, requirements addressed by the OBEM standard (either directly or through reference) include integration of control algorithms at equipment control levels, real-time in situ control capabilities, synchronization of data, object-based solution, visibility of equipment objects to higher level systems (i.e., data accessibility), aspects of volume data management, integration of device control algorithms, and remote monitoring. OBEM achieves specification for these requirements by providing an aggregate description of an equipment class and identifying structure and behavior that provides links to other detailed object-oriented SEMI specifications.2,6

Sensor Actuator Network (SAN — E54): This suite of standards defines concepts, behavior, and message services to facilitate device level communication over a sensor/actuator bus, thereby integrating sensors and actuators into the equipment control system.4,7 Examples of devices include mass flow device, particle counter, capacitance manometer, etc. Components of the standard include sensor/actuator

* SEMI standards in equipment automation/software are delineated by an “E” number designation.
** At the time of this writing, the OBEM specification was under review as a provisional SEMI standard.2
network (i.e., sensor bus) communication standards (NCSs), a common device model (CDM), and specific device models (SDMs).

**CIM Application Framework and APC Framework (under review):** This suite of framework documents specifies a common object-oriented environment for integrating applications and sharing information in a CIM factory domain. These specifications are derived from a SEMATECH CIM framework specification. They define a “plug-in” environment for the definition, partitioning, and integration of CIM applications such as schedulers and work-in-progress trackers. The plug-in’s interfaces are specified in terms of exposed object attributes and methods. The APC framework component specifies framework applications related to control. Figure 7.2 provides an illustration of some of the components of the APC Framework, and shows the integration of the APC Framework with the other frameworks in the suite. Figure 7.3 shows the fundamental components in an APC framework-compliant advanced process controller.

Referring to Figure 7.1, each of the aforementioned standards is part of the hierarchical control solution specified for the industry. An example of a typical R2R control solution that might utilize these standard specifications is shown in Figure 7.4. Note that in this example (1) a SAN is utilized to collect data within an in-line metrology system; (2) the metrology system communicates metrology information
of interest to an R2R controller via SECS messaging; (3) the R2R control solution communicates to the metrology system via SECS and to the tool via GEM (SECS); (4) the internal architecture, as viewed from above, is APC framework compliant; (5) the equipment presents a GEM/SECS interface to both the R2R control element and a host controller;* and (6) the equipment implements R2R control advices (i.e., recipe modification suggestions) via sensor/actuator commands implemented over the equipment SAN.

* There are many different architectures for providing connectivity between equipment and both an R2R controller and a factory host. This two-equipment-port solution is merely one example and is not meant to suggest a preferred architecture.
7.3 SEMATECH AND THE CONTROL SYSTEMS REQUIREMENTS SPECIFICATION

The efforts by the SIA, and especially SEMI, clearly address aspects of definition of the integration environment for APC, including R2R control. However, they don’t provide a concise “big picture” that illustrates how all of the standard elements fit together, and specifically how R2R control should be integrated. SEMATECH realized this deficiency and established a Control Systems Specification Working Group (CSSWG) in early 1996 to pursue a Control Systems Requirements Specification (CSRS) for the industry.\(^\text{10}\) In researching these requirements through interviews of users, integrators, OEMs, standards leaders, and CSSWG members, it became clear that the specification of methods and components for adding a new control capability into a control system is a fundamental requirement that should be addressed in a CSRS. Specifically, users, working with integrators and OEMs, would like the flexibility to easily and reliably add, delete, or modify a sensor, algorithm, application, or control capability in an equipment control system. In further identifying this capability, the following guidelines were identified:

1. The specification should be defined so that the capability could be understood and added at the user skill level — that is, a skill level that doesn’t require technical intervention of the OEM or third-party integrator.
2. The specification should address current-generation and next-generation systems. It should provide a migration path so that current solutions can be reused to a large extent in next-generation CSRS-compliant systems.
3. The specification should reference SEMI standards wherever possible and should be aligned with SEMI directions in standardization.
4. The specification should be aligned with current efforts in advanced control systems specification such as APC Framework and the International 300-mm Initiative (I300I).\(^\text{8,9,11}\)

The two primary components of the CSRS are (1) a specification for the enhancement of existing equipment control systems, and (2) a specification of future control systems. The first component was developed for users, integrators, and OEMs to utilize as when they wish to add a control capability (sensor, algorithm, application, etc.) to an existing system. The CSRS indicates the standards and specifications to which the sensor supplier, OEM, and integrator should adhere so that capability can be easily added to the system. The second component was developed to be utilized as an aid for envisioning future equipment control systems and could thus be an aid in the specification of these systems.

The list of requirements of control systems addressed by the CSRS is shown in Table 7.1. As shown in the table, these requirements are specified across a number of domains to ensure interoperability in each of these domains. Specifically, the CSRS is divided into control and reliability domains. The control domain specifies the structure and operation of the entire factory control systems. The reliability domain addresses issues such as equipment up-time, maintenance, etc.
In mapping the list of requirements to the available standards provided by SEMI, a CSRS for current and near-term next-generation systems is defined. This definition is summarized in Figure 7.5. Referring to this figure, note that the following are specified for a run-to-run control solution:

1. Use of GEM/SECS for communication to equipment (and to metrology system as necessary).
2. Capability for migration to OBEM connectivity to equipment.
3. Implementation of R2R controller as a “piggyback” solution (see below).
4. Internal R2R controller architecture that is APC Framework “compatible” with capability for migration to framework compliance.

### TABLE 7.1
Control Systems Requirements Issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Specification for synchronization/time-stamping of data</td>
<td>Control</td>
</tr>
<tr>
<td>2. Real-time (in situ), run-to-run, and factory-level control capabilities</td>
<td>Control</td>
</tr>
<tr>
<td>3. Specifications for combining run-to-run and in situ control</td>
<td>Control</td>
</tr>
<tr>
<td>4. Specifications for combining run-to-run and factory-level control</td>
<td>Control</td>
</tr>
<tr>
<td>5. Defining equipment, run-to-run, and factory-level control systems utilizing a standard object-based paradigm</td>
<td>Control</td>
</tr>
<tr>
<td>6. Specifications for sensor integration into the control system (models, communication protocols, etc.)</td>
<td>Control</td>
</tr>
<tr>
<td>7. Specifications for integration of third-party) control algorithms at equipment and run-to-run level (without affecting integrity of control system or tool)</td>
<td>Control, Reliability</td>
</tr>
<tr>
<td>8. Defining data accessibility/visibility at equipment and run-to-run control levels</td>
<td>Control</td>
</tr>
<tr>
<td>9. Specs. for remote monitoring</td>
<td>Control</td>
</tr>
<tr>
<td>10. Specifying wafer state information</td>
<td>Control</td>
</tr>
<tr>
<td>11. Supporting wafer-driven processing</td>
<td>Control</td>
</tr>
<tr>
<td>12. Specifications for volume data management</td>
<td>Control</td>
</tr>
<tr>
<td>13. High-speed network compliance</td>
<td>Control</td>
</tr>
<tr>
<td>14. Defining a successor to SECS-II GEM that delivers higher throughput</td>
<td>Control</td>
</tr>
<tr>
<td>15. Specifications defining visibility of machine operations</td>
<td>Control, Reliability</td>
</tr>
<tr>
<td>16. Deterministic performance requirements</td>
<td>Control, Reliability</td>
</tr>
<tr>
<td>17. Configuration management requirements</td>
<td>Control, Reliability</td>
</tr>
<tr>
<td>18. Specifications for equipment transition between up/down time, maintenance, etc.</td>
<td>Reliability</td>
</tr>
<tr>
<td>19. Specifications for equipment maintenance</td>
<td>Reliability</td>
</tr>
<tr>
<td>20. Specifications for testability requirements</td>
<td>Reliability</td>
</tr>
<tr>
<td>21. Specifications for fault detection and end-point control</td>
<td>Control, Reliability</td>
</tr>
<tr>
<td>22. Alignment with the APC Framework efforts</td>
<td>Control</td>
</tr>
<tr>
<td>23. Alignment with 300-mm ECS efforts</td>
<td>Control</td>
</tr>
<tr>
<td>24. Alignment with I300I efforts</td>
<td>Control</td>
</tr>
<tr>
<td>25. ...</td>
<td></td>
</tr>
</tbody>
</table>

In mapping the list of requirements to the available standards provided by SEMI, a CSRS for current and near-term next-generation systems is defined. This definition is summarized in Figure 7.5. Referring to this figure, note that the following are specified for a run-to-run control solution:

1. Use of GEM/SECS for communication to equipment (and to metrology system as necessary).
2. Capability for migration to OBEM connectivity to equipment.
3. Implementation of R2R controller as a “piggyback” solution (see below).
4. Internal R2R controller architecture that is APC Framework “compatible” with capability for migration to framework compliance.
These specifications define elements of the internal structure and communication capability of a “piggyback” controller. A piggyback controller refers to a class of controllers that are integrated with traditional equipment control systems and perform selected control functions that are generally beyond the capability of these traditional control systems. For example, at the real-time level, a piggyback controller running a real-time operating system and utilizing high-speed data interface cards may utilize \( \text{in situ} \) measurement data such as optical emissions spectral data and \( V_{\text{bias}} \) (bias voltage) data to stabilize an etch process through adjustment of power, pressure, and flow equipment inputs (see Figure 7.6). The equipment controller purchased with the system is still utilized to control the sequential operation of the tool (e.g., pumpdown, gas flow, light plasma, etc.); however, as it is not equipped to modify equipment inputs during a run, this task is accomplished by the piggyback controller.

In the R2R control regime, a piggyback controller must be integrated with existing equipment and metrology and perform automatic updates of equipment input parameters as necessary to achieve R2R process control. In evaluating a capability for practical integration (in terms of cost and time), a number of design requirements can be identified for the integrateable R2R control solution. These design requirements are detailed in the next chapter.

### 7.4 SUMMARY

Insight has been provided in this chapter into contributions of three players in the industry that are shaping the roadmap of R2R control development and integration.
The SIA provides a timeline for the development and deployment of R2R control and gives an understanding of the positioning of R2R control with respect to other advancements in the industry. SEMI provides guidance toward interoperability and interchangeability of R2R solutions through standards developed generally in a bottom-up, “grass-roots” fashion. SEMATECH, through its CSRS effort, provides a level of organization to SEMI communication and control standards, and provides a roadmap for users, OEMs, and integrators to specify and verify requirements of control systems for both retrofit and next-generation applications.

These three players have contributed valuable input to the specification of requirements for control solutions in the industry. The derivation of these design requirements for R2R control solutions is provided in the next chapter, followed (in Chapters 9 and 10, respectively) by the description of an enabling technology and an R2R control solution design that meet these design requirements.

ACKNOWLEDGMENT

Much of the material presented in this chapter is derived from Reference 8, and is reprinted with permission. The authors also acknowledge the contributions of the members of the Control Systems Requirements Working Group sponsored by International SEMATECH.

REFERENCES

11. CIM Global Joint Guidance for 300 mm Semiconductor Factories, Release One, International 300-mm Initiative (I300I) and Japan 300-mm Semiconductor Technology Conference (J300) (December 1997).
8 Design Requirements for an Integrative* R2R Control Solution

James Moyne

The lack of immediate widespread acceptance of R2R technology in the semiconductor manufacturing area, despite its proven capabilities, indicates that adequate R2R controller design requirements have not been met for the painless integration of industrial quality R2R control. Design requirements for integrated cost-effective R2R control solutions are presented in this chapter.1-4

Clearly the primary concern of integrators is the short- and long-term cost of developing, integrating, utilizing, and maintaining systems utilizing R2R control. To a large extent, these issues impacting cost must dictate the design of the controller.1 In order to understand the issues that impact the cost of R2R control it is necessary to describe the technical and practical semiconductor manufacturing R2R control environment. First and foremost, semiconductor processes are complex, not well understood (thus physical models are generally nonpredictive), and very dynamic. Further, the control technology applied to these processes is basically sensor-driven, and there is generally an insufficient number of sensors and actuators at each level of control. Thus, there is a tendency to rely on empirical methods for control. A number of empirical control algorithms have been developed for applications to R2R control in semiconductor manufacturing; many have been discussed in detail in Part 2 of this book. Algorithm implementations and other supporting software elements (communication drivers, user interface modules, etc.) have been made available from a number of sources, both commercial and noncommercial. However, in general, the domain of applicability of each algorithm is limited or not well understood.2,3

This unique control environment imparts a number of requirements on practical and integratble R2R control solutions. A fundamental set of these requirements is listed in Table 8.1. Additional information on each requirement is provided in the following sections.4

8.1 PROCESS INDEPENDENCE

As noted above, the R2R control solutions for the semiconductor industry are generally empirical in nature. Further, a set of empirical algorithms have been proved effective for R2R control in a number of process scenarios. The R2R control environment should

* Sometimes referred to as “integrateable.”
take advantage of this multiprocess control capability and provide a level of process independence of the control solution. This will enhance the cost-effectiveness of R2R control by providing a high level of cross-process portability and reusability of the control solution. Design requirements of an R2R control system that will maximize process independence include (1) separation (modularization) of process-specific control capability from generic control capability, (2) capability for (re)configuration of process control I/O for adaptation to a new process, (3) dynamic internal R2R controller control scheme (see below) for rapid reconfiguration to new process control schemes, and (4) capability for plug-and-play of third party modules (see below) for incorporation of new required features in a process control environment.

8.2 PLUG-AND-PLAY INTEGRATION OF EXTERNAL SOFTWARE MODULES

The R2R control solution must be able to interoperate with software programs such as process model builders, optimizers, control algorithms, equipment and metrology interfaces, data presentation and storage mechanisms, and user interfaces. Thus, the control solutions must define a generic interface to all such types of software programs. These types of programs, or modules, should be able to dynamically connect and disconnect to/from the control environment without any code modification. There should be a module interface, which facilitates the passing of arbitrary data, determined at run time between the controller and the module. The module interface must also allow users to develop custom modules, and third-party developers to produce shrink-wrapped modules that can be incorporated in the control scheme dynamically (i.e., so that the control solution can be updated dynamically at run time). Note that this generic and dynamic software interface requirement is critical to achieving the requirement of process independence described above.

8.3 DYNAMIC CONTROL SCHEME

The controller's source of control knowledge (i.e., the knowledge of navigation through execution of software modules to achieve control tasks) should be persistent
yet dynamic, capable of being updated in mid-process, and capable of being adapted to many different processes by the user. One way to achieve these goals is to have the control knowledge stored in a database rather than in static code. Storing the control knowledge in a database makes the knowledge capable of being changed or modified quickly, easily, and at any time (including during run time). Further, it provides a level of portability of the control scheme.

8.4 COMPLEMENTARY OPERATION OF MULTIPLE CONTROL AND OPTIMIZATION METHODS

In analyzing the control algorithm alternatives applied to the full spectrum of process conditions in the semiconductor manufacturing industry, it is clear that none of the available R2R control algorithms cover the entire spectrum of process optimization and control (see Part 2 of this book). This fact imparts two requirements on the R2R control solution if it is to be applicable industry wide. First, there must be a capability for plug-and-play of control algorithm modules (see above) so that the “best” algorithm can be utilized in the control scheme for a process. Second, in many cases the controller must utilize a framework that supports the complementary utilization of a number of sequential optimization and control algorithms. This framework must contain a mechanism that can capture the state of the system after each process run and determine which of the available control algorithms is best suited to the process.

8.5 ABILITY TO PROVIDE R2R CONTROL WITH OR WITHOUT IN SITU CONTROL

In the Introduction to this book (see Figures 7 and 8) we noted that R2R control is merely a component of an envisioned multilevel control system that includes real-time equipment and process control as well as potentially soft-real-time process control components operating in conjunction with the sequential R2R control component. Currently, in many cases, these in situ control components are nonexistent; thus, the R2R control component must be capable of providing control in the absence of these components. However, in other cases in situ control elements (notably endpoint capabilities) are available. The R2R controller should be able to utilize the in situ capabilities in conjunction with R2R control capabilities to provide improved process control. Further, the controller should have a reconfigurable dynamic control scheme (see above) so that in situ control capabilities can be easily added (as they become available) to complement the control scheme.

8.6 PLATFORM INDEPENDENCE

Due to the software development and corporate environments, there are no clearly identifiable software or hardware platform standards (de facto or otherwise) in the semiconductor industry. At the time of this writing, the majority of applications operating at the (non-real-time) level of R2R control utilized the Pentium* (or

* Intel Corp.
similar) hardware platform running Windows NT.* However, there are a few organizations that produce or utilize rival hardware or software platforms demanding solutions for alternative platforms such as UNIX. A key source of cost leverage of any R2R solution, then, is platform independence or, at the very least, a level of portability of solutions between Windows NT and UNIX.\textsuperscript{1}

8.7 USER FRIENDLINESS AND CONTROL INTEGRATION MIGRATION PATH

Though not a quantifiable design requirement, user friendliness is perhaps the most important requirement for the acceptance of R2R control technology in the mainstream of semiconductor manufacturing. In most cases sequential control represents a new technology in fabs. Since processes are not well understood in general, and the control solutions are generally based on empirical data, there is a valid level of hesitation among process engineers and operators in accepting R2R control. There are a number of requirements of R2R control solutions that address this user friendliness issue and provide for a migration path toward full acceptance of R2R control as an integral part of mainstream semiconductor manufacturing. The most important of these requirements is the separation of details of the control algorithm operation in the controller from the overall R2R operation of the controller. The operator or process engineer should not be burdened with a requirement of understanding or even viewing control algorithm operation when he/she is utilizing the system in a production environment. The controller should be presented as a very simple I/O solution, similar to that depicted in Figures 1 and 2 in the Introduction to this book, where, for each control run, the controller utilizes a small amount of metrology data (e.g., average thickness and uniformity) and generates an advice consisting of tuning suggestions for a small number of process parameters (e.g., time and pressure). Note that this I/O solution is the same no matter what control algorithm is utilized; therefore, the portion of the GUI associated with controller operation should remain the same, regardless of the control algorithm utilized. One way to provide this level of consistency is to separate the user interface into operation modes such as Operate and Configure, where the overall R2R operation is viewed in the Operate mode and the algorithm details are viewed and configured in the Configure mode. A security log-in system could then be used to insulate the operator for accessing the Configure mode.

Another user interface requirement is that it be rapidly (re)configurable. Unfortunately, despite efforts at standardization of user interfaces by the semiconductor industry, there are many “standard” user interface forms that vary widely from company to company. A requirement of any R2R control solution is that its GUI be generally configurable to the look and feel specified by the user. Some parameters and ideas for GUI design and configurability are provided in SEMI and SEMATECH user interface specifications.\textsuperscript{5,6} These specifications suggest another important requirement of user interfaces, namely that they comply with (SEMI) standard specifications wherever possible.

\* Microsoft Corp.
8.8 SUMMARY

Some of the important requirements of R2R control solutions for semiconductor manufacturing have been detailed in this chapter. They address not only the technical capability of the solution, but also its economic feasibility. While not all of these requirements apply to every application, they should be utilized in the assessment process of an R2R control solution.

In the following chapters in this part of the book these requirements are utilized in the design and assessment of R2R control-enabling technologies, and R2R control integrated solutions.

REFERENCES

The Generic Cell Controller, or GCC, represents a solution to address the requirements of R2R control in the semiconductor manufacturing industry (as detailed in Chapter 8). The GCC design and a GCC-enabled R2R control solution are presented in this chapter. Specifically, the design of the GCC is presented, in Section 9.1, as a solution to any form of discrete control (including R2R control). Simple examples are used to illustrate the application of the GCC concept to control. A specific solution of GCC application to R2R control is presented in Section 9.2. This presentation includes a detailed description of the GCC operating environment, software components, and process optimization and control environment. This is followed in Section 9.3 by a detailed mapping of the solution design to the design requirements of Chapter 8.

9.1 GENERIC CELL CONTROLLER DESIGN

9.1.1 INTRODUCTION

A perception of the layout of the automated facility is a prerequisite to a study of its control structure. The widely accepted vision of the fully automated manufacturing facility is a hierarchical logical structure that is paralleled by a hierarchical control structure. At the top of the manufacturing hierarchy, a factory controller acts as a global production facility controller, and as a gateway to higher levels of company management. At the bottom of the hierarchy, the “leaves” of the tree are the various types of equipment in the facility involved in product manufacturing. Each piece of equipment has an equipment controller that provides for detailed operation of the equipment, and provides for network access to the facility hierarchy.

Between the facility controller and the equipment controllers, numerous cell control elements are networked in a hierarchy and divide the complicated task of wafer processing. Generally, this hierarchy consists of two or three physical levels. In any manufacturing facility a cell controller is defined as follows:

A cell controller is a unit in a facility that accepts commands from a parent (controller) and implements these commands by instructing children units under its control. These children may also be cell controllers, or may be equipment controllers with sufficient intelligence to interpret the commands sent from a parent cell controller.
Note the distinction between a cell controller — which is not involved in direct equipment control — and an equipment controller. The interface to the cell controller is, to some extent, a function of its role in the facility. Efforts in CIM framework component standardization seek to standardize the interface to these controllers as well as communications between control elements over the factory backbone.4-6

In understanding the above description of a cell controller, it is clear that, although the tasks assigned to the various cell controllers in the facility may vary, cell controllers have common basic functionality. The implementation of a cell controller design that isolates the generic functionality common to all cell controllers would result in a significant reduction of programming effort, as well as in cost in facility startup and facility update.2 Software portability would be increased while system complexity would be decreased.

The remainder of this section contains a description of a Generic Cell Controller (GCC) design for facility automation.1-3 A requirements analysis and list of design assumptions are first presented. This is followed by a general description of the design and a more detailed description of the GCC components and component interactions. A special capability of the GCC, namely its ability to be taught “on the fly” to adapt to a new control environment, is then discussed. Following a summary of the GCC design description, this section concludes with examples (in semiconductor manufacturing) that illustrate GCC operation. Note that throughout this section the GCC design is presented in generic terms and not applied to R2R control; this approach is taken to give the reader a better understanding of the range of applicability of GCC concepts. The specific application of GCC to R2R control is presented in Section 9.2.

9.1.2 GCC Design Goals and Assumptions

In determining design specifications for a generic cell controller in the semiconductor facility, careful consideration must be given to aspects of the facility environment and to the goals of facility automation implementers. The automated semiconductor manufacturing facility is a hybrid environment consisting of multivendor controllers, networks, and equipment.2,7 The product line and even aspects of the facility structure and operation may be ever-changing (i.e., dynamic within the limits imposed by the hierarchical facility definition). The facility control structure must support this hybrid dynamic environment in real time and at a minimum cost.7

Therefore, the following design goals are realized for GCC design:

1. The controller design must accommodate the hybrid and dynamic operational environment. Included in this goal is the implication that the controller design must not be a function of the facility hardware, software, and networking base.

2. The controller design specification should provide for the maximization of generic attributes of the cell controller so that development costs are minimized as redundancy of programming effort is minimized.

3. The cell controller performance should be maximized with respect to development speed, operating speed, software size, memory requirements,
etc. A minimum performance requirement is the ability to function in a sequential control environment.

To completely specify the role of the cell controller in the facility, a few assumptions about facility structure and operation must be made. The overall facility structure must be specified. For this design effort, a hierarchical facility structure is assumed. Thus, it will be assumed that a cell controller functions in the hierarchical facility structure as described above. Important properties of the hierarchical structure that should be considered in attempting a cell controller design include:

1. The controller accepts commands from, at most, one parent and implements each command by passing a set of commands to one or more children under its control.
2. The scope of influence of a controller is its immediate parent and immediate children.\(^2\) (Any cell controller design must reflect a clear understanding of the internal operations of the controller as well as its interaction with its immediate parent controller or children (other generic cell controllers and/or equipment controllers)).

It should be noted that the assumption of a hierarchical facility as described above is not a prerequisite for the realization of this GCC design. This point will become clear as the description of the design proceeds. The assumption of a hierarchical facility is made so that the GCC description can be more easily understood.

Assumptions are also made with respect to facility-wide communication. The first assumption is that any communication protocol in the facility between cell controllers or between cell controller and equipment controller satisfies the following additional properties:

3. The communication protocol provides for communication of application-level information to/from the controller that has error-free communication transmissions.
4. The communication protocol will support the communication (between controller applications) of any message that conforms to the specified facility message format (see next paragraph).
5. The communication protocol provides end-to-end delivery of messages between controller applications in a “reasonable” length of time (reasonable such that delays in message delivery in general do not grossly affect the timely operation of the controller).
6. The communication protocol functionality is partitioned such that a clear separation exists between the functionalities of the application and the functionalities of the communication protocol.

Another assumption made concerning facility communication is that a facility-wide message format specification exists. This specification should indicate a well-defined message information format and should allow for the communication of any and all
necessary information between controllers in a facility. As an example, one appropriate message format specification for semiconductor manufacturing is SECS (see Chapter 7).\textsuperscript{6} The CIM framework standards efforts are also aimed at defining standardized interfaces and interactions between control components.\textsuperscript{4-6}

Other assumptions are made concerning the development environment. It is assumed that the cell controller hardware environment is such that it does not impose any controller design limitations (such as memory space and hardware interface compatibility). (Of course this assumption is not valid in a practical implementation, but is useful in developing generic design specifications.) It is also assumed that because of the structure, potentially large quantity, and dynamic property of the control information to be stored, any data storage and management in the cell controller will make use of a relational database management system (DBMS) (although this assumption does not restrict the eventual incorporation of other database systems such as object-oriented database application systems\textsuperscript{8}).

\subsection{9.1.3 General Description of GCC}

With the foundation for the design in place, the design process may begin. An initial concern is that the design goals for operation in a heterogeneous environment and the design goal of a generic cell controller tend to conflict (because the more heterogeneous the environment, the greater the number of factors that must be dealt with to achieve a generic controller design). Any cell controller design that maximizes generic functionality should provide for the isolation of that functionality from the expected hybrid of communication protocols and hierarchical facility architectures. To provide for this isolation, the cell controller design should be modular; generic modules should exist in the cell controller design to provide for the truly generic aspects of functionality of the controller, while nongeneric modules should also exist in the design to isolate the majority of the cell controller functionality from the specifics of facility communication and facility structure.

The isolation from facility communication is provided by an interpreter module that converts information from a cell controller communication network interface into information with a common message format (and vice-versa). Note that messages converted by the cell controller interpreter module into the common message format are passed on to a message parser for use by the rest of the generic cell controller. Note also that the communication network could be any network that satisfies conditions 3 through 6 above. Because of the modular nature of the communication protocols (see assumption 6), the function of the interpreter module generally is to strip off protocol control information from incoming messages or to encapsulate outgoing messages with protocol control information.

The necessary isolation from changes in the facility structure already exists as restrictions have been placed on the facility structure concerning hierarchy and scope of influence (see assumptions 1 and 2). Thus, the components of interaction with the cell controller are well known and definable from a generic perspective.

A conclusion of this analysis, then, is that a generic environment is created for the development of a cell controller specification for use in the specified hierarchical
facility and with the implementation of the appropriate communication interpreters
isolating the protocol-specific aspects of communication.

When developing a cell controller specification in this generic environment, a
first step is to characterize the desired functionality of the controller. A controller
with the design described here reacts to three sources of stimuli (also called events):
(1) messages received from above (the parent controller), (2) messages received
from below (a child cell controller or equipment controller), and (3) an internally
triggered event (such as a timeout on a message transaction). The method of reaction
to these sources of stimuli is also (generically) definable as follows:

The cell controller, initially in an idle state, receives stimulus in the form of either a
message of a specified format or an internal event indication. The cell controller parses
the information, interprets the information, and takes action as a result. The cell
controller then returns to an idle state.

There are many cell controller designs that would potentially achieve the desired
functionality, however, the inclusion of performance and cost considerations in the
design requirements greatly reduces the number that are acceptable. Usually software
complexity is proportional to its size, so a specification that minimizes application
software size is desirable. A modular specification is also desirable because, as stated
earlier, it provides isolation from facility structure and communication. Modular
design then results in increased flexibility and portability of the design, and decreased
update costs. From the performance standpoint, sequential control performance
consistent with process events in the factory is a requirement. Finally, and perhaps
most importantly, it must be emphasized that the specification should be generic
and therefore applicable to any cell controller in the hierarchical facility.

The cell controller design schematically represented in Figure 9.1 incorporates
all of the above analysis and design considerations. The overall operation of this
cell controller design can be summarized as follows:

The cell controller is a reactive device and therefore begins in an idle state. It responds
to message events received from the parent or children controllers, or to internally
triggered events. All message events received are stripped of control information
(specific to the communication protocol) by I/O interpreter modules, and the generic
portion of the cell controller is presented with internal or message “event” indications
structured in a common facility-wide message format. A message parser makes use of
the well-defined structural properties of the message format to extract the event data
from the message. A main program kernel module called a “conductor” matches the
event data received from the message parser or from an internal event to an entry in
the controller database. (The database is structured such that the event formats are
embedded in the data dictionary; therefore, the database is capable of storing the event
data of any event — externally or internally generated — that complies with the
specified format.) Through relations implied by the database, the conductor determines
the unique action to take as a result of an event received. Also through relations implied
by the database, the appropriate action may be invoked. to invoke this action the
conductor first scans the database to determine the necessary routines (or methods) to
call as well as parameters to pass to the routines, and an order to call the routines. The
conductor calls these routines in the specified order and with the specified parameters. A routine may update the database. A routine may also send message data (derived from the database) to the message parser to be formatted and forwarded to the I/O interpreter so that the message data may be communicated in the specified format and appropriate communication protocol to a parent or child controller. Upon return of control to the conductor from the routines called, the cell controller returns to an idle state.

The event posting and servicing style of operation of the controller is illustrated by the flow diagram of Figure 9.2. In analyzing the figure it is clear that the macroscopic view of controller information flow is uncomplicated and well defined because (1) the sources of controller stimulation “events” are well defined as is the message format, and (2) much of the control information is embedded in the database structure. It is also clear that the system is heavily data-driven; indeed, all events result from data received (via a formatted message) or from an internal event (e.g., timeout on data expected). Thus, the database component of the controller plays a large role in determining controller operation.

9.1.4 Specific Description of GCC Modules

The following is a detailed description of the various modules of the GCC introduced in the previous section and illustrated in the GCC block diagram of Figure 9.1.

9.1.4.1 I/O Interpreter and Message Parser

These GCC modules perform data conversion between the facility-wide message format and the internal GCC event format. In this way, the I/O interpreter isolates
most of the functionality of the controller from communication protocol-specific information. This allows for a more generic controller specification. As an example, in the automated semiconductor manufacturing arena, the common message format is SECS-II. The controller interpreter module provides a two-way translation from SECS information over a communication network to SECS-II formatted messages. The message parser in turn formats and deciphers SECS-II formatted message information to and from GCC event messages. Note that the communication network could be any Ethernet, RS-232, etc., network. The functionality of the communication hardware (e.g., an RS-232 communication board) may be considered as part of the functionality of the I/O interpreter. The functionality of the I/O interpreter and message parser that is important to the GCC algorithm is that these modules, working together, present/accept application level information to/from the GCC that is independent of the physical and application communication protocol through which it was/will be communicated.

As a specific example, suppose an equipment controller is reporting its model number and software revision number to the GCC using a SECS Stream 1 Function 2 message. Suppose the model number is “EQUIP1” and the revision number is “Ver1.0.” The I/O interpreter module would provide support of communication of data into the GCC via an RS-232 or Ethernet interface, and present the message parser module with the SECS-II message syntax. This incoming SECS message would be of the format:

Message (Stream 1, Function 2)
List of two items
   Item 1 == Model number == “EQUIP1”
   Item 2 == Version number == “Ver1.0”.

FIGURE 9.2 Flow diagram of cell controller operation (internal event is stated here as “Timeout Indication”).
The information communicated to the parser would be the same SECS-II syntax regardless of the lower-level communication medium (e.g., RS-232 or Ethernet) utilized.

**9.1.4.2 Main Program Kernel Module (or “Conductor”)**

This module coordinates all controller activity. Its functionality is quite simple and well defined. This module is event-driven, reacting to message events from its parent and children, and internal events. For each stimulus event, the module uses the database to interpret the event and determine the action to be taken (i.e., the routines/methods in modules to be called). The conductor then calls the specified routine(s) in the order and with the parameters specified by the database. Upon return from these routine(s) the conductor returns to a state of rest/idle. Figure 9.2 is a flow diagram illustrating conductor functionality. The elements of the flow diagram may be explained as follows.

1. **Start/Idle**: The conductor is a reactive, event-driven module. Thus, for the case where no event stimulus is received and all (previous) stimulus reaction transients have ended, this module remains in an idle state, continuously checking for an event stimulus (see next item).

2. **Stimulus to Module**: When in an idle state, the conductor continuously checks for an incoming stimulus (external message or internal event). All message events received have been parsed by the message parser and saved as message information in a structure defined in the conductor. If no event is posted, the conductor continues to poll. If an event is detected the conductor “wakes up” and services the event (see item 3 below).

   Incoming event detection may be accomplished in a number of ways, such as with a mailbox mechanism. With this method, a memory location (variable) may be designated as a mailbox, accessible by the message parser module, an internal timer module, and the conductor. The mailbox is normally empty (contains a null value) and is polled by the conductor. When an event, external or internal, is to be posted, the mailbox flag is raised and an address in the mailbox points to the variable instance containing the event information. Upon receiving the information (upon its next poll) the conductor clears the mailbox message/timeout pointer, thereby lowering the mailbox flag.

3. **Match Event to Database**: As stated in the general description of the GCC, the conductor services event stimuli by matching the information of the event to information in the GCC database and, through relations implied by the database, taking the appropriate action. Thus, when the conductor has detected that an event stimulus has occurred, it must first attempt to find a match between information associated with the event and an entry in the database. The way in which event information is stored in the database (i.e., the database schema) is dependent on a variety of factors including the message format specification, the DBMS used, as well as user storage, performance, and interface objectives. Regardless of the
storage format, the conductor must contain an algorithm that will search through the database and indicate a match between stored information and the current message or timeout associated with the event. This algorithm will be a function of the database schema as well as the event data structure and the message protocol.

The interaction of the conductor with the database is described further below. Also, later in this section, examples are presented that illustrate the matching of a message to the database (where the messaging format is SECS), and the matching of a timeout internal event to the database. For these examples the database schema and message and timeout event data structures are detailed.

4. **Determine Action to Take**: Each message and timer instance stored in the database is linked to a single action identified by an action number. After the conductor has found a match between the incoming event (message or timeout) and an entry in the database, it retrieves the associated action number.

5. **Determine Routine(s) to Call (in Order) and Parameters to Pass**: In the database, each action (identified by an action number) is associated with (is linked to) one or more routines (each identified by its routine number), and a set of parameters to be passed to each routine. Note that in an object-oriented paradigm these “routines” may be equated to methods. After the main program has determined an action number in response to the incoming event (see item 4 above), it retrieves the associated routine numbers and parameters. It then compiles an ordered list of routines to call, and the parameters to pass with each call. An example of such a list is indicated in Table 9.1. In this example the action to be taken is action number 5. The purpose of this action is to read the values of two sensors on a piece of equipment and store them for later use. Four routines are to be called to implement this action: (1) “Initialize() — no parameters” — wakes up the piece of equipment; (2) “GetData(Ch, 1, 2)” — sends a message to the equipment requesting the values of channels 1 and 2; (3) “PollACK(Val_1, Val_2)” polls the equipment for a response message containing the sensor values “Val_1” and “Val_2”; and (4) “UpdatesDB(1, Val_1, 2, Val_2)” updates the database with the new values of channels 1 and 2.

6. **Another Routine to Call?/Call Routine(Parameters)**: From the compiled, ordered list of routines to call (see item 5 above), the conductor takes the necessary steps to implement the action indicated (see item 4 above). This is accomplished by calling each routine in the list in order with the indicated parameters. In the example relating to Table 9.1, the routines “Initialize,” “GetData,” “PollACK,” and “UpdateDB” are called in order and the values of equipment channels 1 and 2 are read and updated in the local database. After control has returned from the last routine called in the ordered list, the conductor returns to an idle state, polling for the next event stimulus.
9.1.4.3 Database

The database is the central source of information for the controller. The database contains the necessary information so that the conductor can match any valid incoming event to a database entry. The database also relates event information to actions to be taken and routines to be called. The general structure of such a database is indicated by the Entity Relationship (E-R) diagram of Figure 9.3 (which is a simplification of the detailed E-R diagram of the actual controller database). The Entity Relationship technique is commonly used to model data storage systems. A brief tutorial on the E-R modeling technique is given in Appendix A (found at the end of this chapter). An in-depth presentation of the E-R approach is found in Teorey. Referring now to Figure 9.3, the structure and use of the database will be described in detail.

The database contains the seven entities “Routine,” “Invocation,” “InternalEvents,” “InternalEvent Instance,” “Action,” and “Message.” A set of candidate relations (or tables) for a relational database schema can be derived from the model using well-documented techniques. In the basic implementation each of the entities would be a table in the GCC relational database. Further, the relationship “Invoked By” would also be a table in the database. Relationships between these entities would be implemented as foreign keys in the relational tables. Table 9.2 is an example of such a basic schema. Note that Keys and Foreign Keys of tables are

### Table 9.1
Ordered List of Routines to Call from the Conductor

<table>
<thead>
<tr>
<th>Action#</th>
<th>Order#</th>
<th>Routine#</th>
<th>RoutineName</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>Initialize</td>
<td>None</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>5</td>
<td>GetData</td>
<td>Ch. 1, 2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>11</td>
<td>PollACK</td>
<td>Val_1, Val_2</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>UpdateDB</td>
<td>1, Val_1, 2, Val_2</td>
</tr>
</tbody>
</table>

![Figure 9.3](image)
indicated. In the following paragraphs, each entity/relationship resulting in a table (or tables) in Table 9.2 is described in detail.

1. **Message**: The Message entity is perhaps the most complex entity in the database. The database table or tables corresponding to this entity would store information about every message event stimuli that the GCC is to receive. This includes messages from its parent controller and messages from child controller(s). Upon detecting a message event, the conductor services the event by first attempting to find a match between the message event and an instance of the Message entity in the database. If a match is found, the conductor can determine (also from the Message entity instance) a unique action number that serves as a key in determining an action to take to service the event (see also below).

   The database must provide the capability to store the information of any message to be sent or received by the controller. This requirement impacts on the design of the message storage portion of the database. The database schema incorporates the entire message structure definition into its data dictionary. The result is that any message with that message structure definition can be stored in the database, however, the structure of the database is independent of the message data itself, and is therefore generic to the facility.

2. **InternalEvent**: This relation contains information on all possible internal events that could have instances managed by the GCC. Note that this table is not accessed by the conductor in servicing an event, as it does not contain direct information on currently active internal events (see item 3 below). The existence of this table, however, may be necessary as it
contains information on internal events that may not be presently active. Thus deletion of this table could result in loss of information. As an example, a “Conversation Timer” may be defined and may be used to monitor the timely receipt of a reply message in response to a request message sent. The entry in the InternalEvent table may contain a description of the timer as well as other parameters such as a Timer# and the default timeout value. If, at some time, there are no active instances of this timer (i.e., if there are no outstanding conversations with the GCC) then there would be no instances of the timer in the InternalEvent_Instance table (see next item). Deletion of the InternalEvent table in this situation would result in critical loss of information.

3. **InternalEvent_Instance**: The InternalEvent_Instance table contains information on internal events being monitored in the system. This table would contain the foreign key of InternalEvent#, indicating the type of internal event of which this is an instance. This relation also contains the foreign key of Action# indicating a “one-to-many” relationship between the Action entity (described below) and the InternalEvent_Instance entity. Note that the foreign key of Action# cannot be null, i.e., every internal event monitored must correspond to an action (note that the action may be “no action”).

   The InternalEvent_Instance and Message entities provide similar functionality to the GCC system of mapping the event to a single action. Thus, if internal and external events are defined utilizing a common format, the tables corresponding to these entities can be combined into a single “Event” table.

4. **Action**: In servicing an event, the conductor first attempts to find a match with an instance of the Message (or InternalEvent_Instance) entity. If a match is found, the message or timer database record contains a value for foreign key parameter Action#. This value corresponds to an instance of the Action entity (see Table 9.2).

5. **Invoked_By**: The Invoked_By table results from the ternary relationship between Action, Routine, and Invocation (see Figure 9.3). From this table, the conductor can determine, for a desired action, the routines to call, the parameters to pass to each routine, and the order in which to call the routines. To accomplish this, the conductor queries the database for all entries in the Invoked_By table that contain the desired value for the Action# parameter. These entries are then ordered by the value of the Invocation# parameter. The routines (identified by their Routine#) are then called in order with the parameters indicated by the Parameter_Pointer value.

6. **Routine**: This table results from the Routine entity and contains information on each of the routines in the GCC. Note that the Routine table does not have to be accessed by the conductor in servicing an event as the necessary information on routines to call (e.g., Routine#, Parameter_Pointer, etc.) can be found in the Invoked_by table. The existence of the Routine table,
however, may be necessary as it contains information on routines that
may not be presently involved in the Invoked_by relationship. Thus,
deletion of this table could result in loss of information.

7. **Invocation_Num**: This table results from the Invocation entity and, in this
example implementation, only contains a single parameter used in the
Invoked_By table to indicate an ordering of routines associated with an
action. As this table does not provide any additional significant informa-
tion, it would be deleted from a revised database schema.

As a final note to the description of the database module, examples are presented
at the end of this section that illustrate the servicing of message and timeout events.
Included with these examples is an illustration of the function of matching the
message/timeout to the database, where the messaging format is SECS and the
database schema is as indicated above.

### 9.1.4.4 Internal Event Monitor Module

This optional module manages the internal GCC events. For instance, it may manage
timeout events and pass the appropriate information to the conductor.

### 9.1.4.5 Routines

The routines contain the functionality necessary to implement the details of control-
ler actions. As noted earlier, routines may be components or methods of program
modules. The routines are called by the conductor, as shown in Figure 9.1. Using
information stored in the database, the conductor determines which routines to call,
the order in which to call the routines, and the parameters to pass to the routines.
The execution of a routine may result in any combination of the following four
actions.

1. The database is modified. Entries may be added, deleted, or modified
   (within the obvious limitations that the modification is permitted and data
   integrity is maintained). For example, the data parameter of a message
   may be altered to reflect a recipe change. A timer instance may be created
to reflect the fact that a request message has just been received (and no
   reply yet sent). Note that routines are the only system components that
can alter the data in the database because all actions (including database
updates) are implemented through calls to routines.

2. A message(s) is sent to a higher- or lower-level controller(s). The routine
   may access the database to determine the data to be sent in the message.
The routine then passes the information to the message parser module
where it is formatted into a message conforming to the specified facility-
wide format (see Figure 9.1). Note that routines are the only system
components that invoke the sending of a message to any source outside
the controller.
3. An event is posted to the GCC. The routine may generate an event to the GCC. This event is serviced per normal GCC operation.
4. Elements outside the GCC are impacted, i.e., the routine interacts with elements not defined within the realm of the GCC (see below).

The routines then are defined to have the role of implementing the details of actions that result from the servicing of an event to the controller.

Examples of some of the roles of routines include:

1. Initiating a timer instance. For example, a message has been sent to a child controller and a timely reply is expected. A routine would establish a timer for the conversation in the Timer Instance database table. The timer would be monitored by the Internal Event module.
2. Deleting a timer instance. In servicing a timeout event, a routine would probably be called to remove the appropriate timer event entry from the InternalEvent_Instance database table.
3. Retransmitting a message. If, after a message has been sent to a remote controller, a negative acknowledgment message event or a timeout event is detected, a retransmit routine might be called to service the event. This module would retransmit the initial message.
4. Updating message information in the database. If a message has been received that indicates a change in the state of a machine or process, the information in the database may be modified by a routine to reflect the change in the state of the system and the change in expected incoming information.
5. Generating a message event to the GCC. For example, as part of an equipment diagnostic action event service, a routine may determine that an alarm has occurred at the equipment and may generate a message to the GCC to shut down the equipment.
6. Reading and/or writing from/to an external data structure, i.e., a data structure that is not directly part of the GCC database. For example, a routine might be directed to update an expert systems knowledge database, or a routine might print results to a log file.

Note that, within the definition of routines, a possible role of a routine is to serve as an interactive window to application modules (such as expert systems, simulation software, etc.) or directly to a user. (Note that these modules could be operating concurrently with the generic controller modules.) As an example of the use of an interactive application module, a user interface routine might be invoked in a situation where a correctly formatted message received does not have a match in the database. The user interactive routine might be called to determine the course of action from the user and to update portions of the database. Thus, the cell controller information can be modified while in operation such that it adapts to new stimuli; the user interactive window interface is used to teach the controller how to react to new stimuli. This learning mechanism is considered to be an important feature of
the GCC. A detailed discussion of such a learning mechanism for the GCC is discussed in the next subsection.

It should also be noted that issues of serializing message events are not addressed here, as this issue is, to a large extent, implementation-specific. As an example, a GCC solution implemented on a high-bandwidth processor may allow for parallel servicing of GCC events, with appropriate database access serialized to ensure database information integrity. On the other hand, a more simplified GCC implementation may queue events as they are received and address them serially.

9.1.5 A Learning Mechanism for the GCC

This section contains a description of a special capability of the GCC, namely its ability to be taught “on the fly,” i.e., while in operation. This capability of the GCC is given special attention for many reasons, including (1) the feature and its implementation are not necessarily intuitively obvious from the description of the GCC in previous sections, (2) the capability is rare in current (non-GCC) controller implementations as most current implementations do not have a dynamic control algorithms (because the control algorithms are “hard coded” rather than incorporated into a database), and (3) the capability is very desirable in flexible manufacturing systems, as these systems are dynamic with respect to product and product flow, and possibly equipment.

The GCC is defined as reacting to an event by trying to find a match between the event and an entry in the database, and then, through relations implied by the database, calling routine(s) in a specified order with specified parameter(s) to service the event. In many situations, however, especially when a system is in its infancy, many unexpected events may occur for which there is no match in the database. The GCC can be structured to detect these events and learn from an expert on the fly, i.e., while in operation, how to service these events. This is accomplished with a set of routines that serve as an interactive window to an expert user. These routines are called as part of an action linked to any message or timeout event that does not have a match in the database. The routines inform the expert of the anomalous event and query the expert for the appropriate action (new or existing) to take to service the event. Assuming the response from the expert is valid (i.e., it satisfies existing system and database integrity rules) the information is entered into the database. The GCC has “learned” how to service the event; if the event occurs at a later time, the GCC has developed the necessary knowledge to service the event. Figure 9.4 is an expanded view of the GCC flow diagram of Figure 9.2, illustrating the system learning capability.

It should be noted that the expert described in the previous paragraph does not necessarily have to be a human expert. It could be a knowledge base, artificial intelligence system, or any other qualified decision-making body.

9.1.6 Implementation of GCC Learning Mechanism

The implementation of a GCC learning mechanism could vary greatly from system to system. As noted above, the expert could be any (human or nonhuman) decision-making body. In early GCC implementations, however, the expert will most likely
be human. A GCC interface will most likely be menu-driven, guiding the expert through the update process.

As an example, when an unknown message event is detected, the expert may be greeted with a menu system to guide him/her through the GCC control knowledge update process. Note that the expert may bypass the learning process by allowing the GCC to return to the idle state without servicing the event. The expert may also bring the system to a halt if the event warrants such extreme action. In the general case, however, the expert is expected to teach the GCC to service the event on the fly.

If the user chooses to teach the GCC on the fly, the system should allow the expert (through the user interface) to view the unknown message in detail. Note that the message can be easily inserted into the existing database, as it complies with the facility-wide message format. The expert should also be allowed to view the list of currently available actions and the routines, parameters, etc., linked to them. The expert could then be prompted to enter an Action# to be associated with (linked to) the message event. The expert may chose an existing Action# to service the event. At this point, assuming the entry passes any system and database integrity checks, the learning process is complete as the existing event is already linked to a set of routine calls. However, if the expert chooses to create a new action to service the event, the GCC must learn the routine calls to be associated with this new action.

FIGURE 9.4 Generic Cell Controller flow diagram expanded to illustrate system learning capability.
Thus, for this case the user interface should allow the expert to view existing routines. The expert enters a sequence of routines to be called and the parameters to pass to these routines to implement this new action. After the sequence has been entered, and assuming the entries pass any system and database integrity checks, the learning process is complete. The GCC is now equipped to service this and future instances of the event.

It is important to note that the time needed to teach the GCC to service an event may vary greatly depending on the type of event, the type of teacher, as well as the teacher-to-GCC interface. In some cases the learning process may be too slow to effectively service the event that initiated the learning sequence. Thus, the learning interface may also contain options so that the current event instance may be treated differently from future occurrences of the event. For example, the GCC may be instructed to inform the operator of an error in response to a current event; however, the appropriate service response is formatted through the learning process as the appropriate response to future occurrences of that event type.

9.1.7 Examples

Examples presented here serve to illustrate generic cell controller operation. They also serve to illustrate the various functionalities of routines that might exist in a generic cell controller. For all examples, the generic cell controller operation is analyzed in a scenario that might occur in a semiconductor manufacturing facility. The facility-wide message format is SECS-II. Note that these examples do not address providing an R2R control capability utilizing a GCC-enabled solution; examples of this type are provided in Chapter 11.

9.1.7.1 Example 1 — Etching a Wafer

In this example, a generic cell controller, initially in an idle state (see Figure 9.2), receives a message from its parent to etch 500 Å of SiO₂ on a wafer. The generic cell controller reacts to the message as described above, finding a match to the message in the database and, through relations implied by the database, taking the appropriate action. The portion of this generic cell controller database that pertains to this example is shown in Figure 9.5. Note that some implementation-specific parameters have been added to the database tables. Specifically, an “Active?” parameter has been added to the message table. This parameter is used to indicate whether a message is expected; for this implementation the generic cell controller should only take the action indicated by the message table entry if that entry is tagged as active. In the following paragraphs a detailed description is included of the actions taken by this generic cell controller to service this message.

This example begins with a parent controller generating an event to the generic cell controller in the form of an SECS-II message indicating that 500 Å of SiO₂ should be etched in a wafer residing in Equipment#1. The command enters the generic cell controller through an I/O interpreter module where it is stripped of all protocol-specific control information (see Figure 9.1). The SECS-II message is then parsed by the message parser and presented to the conductor. The conductor services
FIGURE 9.5 Portion of a GCC database pertaining to Example #1 — Etching a Wafer.
the event by first searching the Message portion of the database (i.e., tables Message, List, and Item*) to find a match with the incoming data. Assume that in comparing the message received the database records in Figure 9.5, a match exists with Message# 7. From the Message table, the conductor determines that Action#4 should be taken to service the message event. The Action table indicates that Action#4 is “REQ_LOAD_RECIPE_ETCH_500_RIE#1,” i.e., the action will result in a variety of tasks being completed, including the issue of a request to Reactive Ion Etcher #1 to download a recipe to etch 500 Å.

To implement the above action the conductor makes five routine calls. The conductor determines the routines to call, the order in which to call the routines, and the parameters to pass to the routines, from the Invoked_By table shown in Figure 9.5. The sequence of routine calls for this action are:

1. RemoteCommandACK(SUCCESS, 0);
2. PPLoadInquire(“Etch500”, 1000, 1);
3. ActivateDBMessage(7, 2, OK, 1);
4. TimerAdd(CONVERS, 7, 2, 99, 0, 1);

This sequence of routines, when executed, first sends an acknowledgment message to the parent controller indicating that the etch job will be initiated. A message is then sent to the appropriate equipment controller requesting that it accept a download of a recipe to perform the etch. An acknowledgment (accept or deny) is expected as a response to this request; thus, the appropriate accept and deny messages in the database are tagged ACTIVE to indicate that they are expected. As the reply is expected in a timely fashion, a conversation timer instance is added to the InternalEvent_Instance table in the database. Finally, the etch request event is logged.

A detailed description of the activities associated with the calling of each of the routines is provided in Appendix B.

After control returns to the conductor from the last routine call, the event is considered serviced and the generic cell controller returns to an idle state (see Figure 9.2). Note that although the completion of these tasks (through routine calls) results in the etch request event being serviced, the etch job itself if not complete. The completion of the etch job would require additional stimuli from the equipment (e.g., a Stream 7 Function 2 response to the Process Program Load Inquire message); this could generate an action to download the recipe (see Action #15 in Figure 9.5) and send a final job report message to the parent controller.

9.1.7.2 Example 2 — Servicing a Timeout

In this example, a generic cell controller is initially in an idle state, polling for a message received or a timeout indication. A conversation timer instance has been

---

* “List” and “Item” refer to the components of an SECS message. A Message consists of zero or more lists and zero or more items. A list consists of zero or more items. Items are the usable data components of the message.
created previously in servicing an earlier event. The InternalEvent_Module, in scanning the InternalEvent_Instance table, deduces that the conversation timer instance has timed out. The generic cell controller reacts to the timeout event as described above, finding a match to the timeout indication in the database and, through relations implied by the database, taking the appropriate action.

As an example, if the InternalEvent Module detects that timer instance 3 of type Timer#4 has timed out, the module passes the appropriate information to the conductor through a structured variable. The conductor services the event by first searching the InternalEvent_Instance table in the database to find a match with the incoming data. The conductor then determines the appropriate action number from the database relations. The ordered execution of routines called to carry out this action could cause, for example, the issuing of a report to the parent controller indicating that a timeout has occurred with the appropriate identifying parameters.

9.1.8 GCC Design Summary

In this section, a GCC design has been presented. It is important to note that the GCC design is applicable to a number of sequential control scenarios in addition to R2R control. Specifically, the design is applicable in areas where flexibility, portability, and capability for integration are important. For example, a GCC framework could be utilized as an enabler for an advanced process control station that includes Fault Detection and Classification, Statistical Process Control, process monitoring and diagnostics, and off-line data analysis, in addition to R2R control.

The R2R control arena, though, represents the first arena where the approaches such as the GCC have proved effective. In the next section the application of the GCC concept to R2R control solutions is discussed in detail.

9.2 Generic Cell Controller Implementation

The GCC solution presented in this section is comprised of a suite of software applications that work together in the manner of the GCC operational model described in Section 9.1 to provide an automated R2R control capability. As shown in Figure 9.6, the system consists of the GCC kernel and a set of modules (these modules contain the “routines” defined in Section 9.1). One of the modules is a GCC GUI, which presents an abstract view of the cell. It shows the high-level commands that the GCC implementation can accept on behalf of the cell, together with the list of modules containing the routines that will actually carry out the work to execute high-level commands.

9.2.1 Module Descriptions

The following is a description of the components that comprise the GCC R2R solution system. Although the description is, to some extent, implementation-specific, it serves to provide insight into the design specifications that are necessary to achieve the solution requirements identified in Chapter 8. Further, the design described contains components that may not be required in some R2R control
applications; the specific implementation utilized would probably implement a subset of components described here. Note that the operation of practical systems that utilize a subset of the components defined here are described later in this book (e.g., Chapter 11 and Part 5).

9.2.1.1 GCC Kernel

The GCC Kernel contains the Main Program Module or “conductor” as described in Section 9.1. It waits for events and reacts with corresponding actions. When the GCC receives a command, an event is posted and the GCC consults its database to determine a corresponding action, which is an ordered list of invocations. An invocation is a message sent to a target. The targets in this case are the various modules connected to the GCC.

9.2.1.2 The GCC Graphical User Interface

The GCC solution provides a three-mode graphical user interface, as shown in Figure 9.7. By separating the GUI at the high level into Operate, Setup, and Access modes, and by limiting access to the setup mode based on the log-in security level, the control-specific aspects of the solution are separated from the operation-specific aspects. GCC GUI design requirements and compliance with standards are discussed in Chapter 8, as well as later in this chapter.

9.2.1.3 GCC Modules

The GCC modules identify themselves and the cell to which they belong by name, as a character string. The database can be modified or modules can be interchanged,
FIGURE 9.7 Illustration of a three-mode graphical interface (traversed through mode buttons at the bottom of the screen — Operate mode shown here).
thereby allowing dynamic binding of the software modules to the distributed GCC application suite. What this means is that the GCC solution can be modified during run time through the plug-and-play of module application modules and the corresponding modification to the data in the GCC database (as necessary) to specify the utilization of these modules.

All modules have certain common attributes and behaviors. If one runs a module with the kernel not running, one will notice that the module automatically tries to find the kernel and bind to it. If the module cannot find the kernel, it allows one to either

1. Try a search for the kernel again.
2. Specify a host name (either a specific name, or have it dynamically search for the kernel running on any local host within a local area network).
3. Run the module in stand-alone model. (For passive modules, such as a GUI module, the passive run mode has no utility; however, for an active module, such as an algorithm module, it allows for utilization of the module, for example, to configure process control models.)

### 9.2.1.4 GCC Module-Control Algorithm

This module or set of pluggable modules provides dynamic algorithmic solutions for sequential process parameter modification based on previous process outputs, possibly previous and current process inputs, and historical process data. For example, the algorithm could dynamically model a first-order approximation of the response surface model of the process at the process operating point (see Chapter 3). Regardless of the specific algorithm, its run-time “operate” behavior is as follows: given pre- and postprocess metrology as available, it updates the dynamic model and delivers the appropriate parameter modifications to achieve specific targets. “Setup” behavior includes a capability for creation, modification, and loading of control model configurations, viewing of model parameter evolution and configuration, possibly moving back the controller to a specified control run (i.e., deleting a portion of the recent controller evolution), turning the model adaptation behavior on and off, turning the control behavior on and off, executing the control behavior in a single-step mode, and configuring a process simulator.

### 9.2.1.5 GCC Module–Equipment Interface

This module provides an interface to the equipment. For example, it encodes and parses SECS communications to support download of equipment parameter updates. It also provides a capability to maintain synchronization with the tool application.

### 9.2.1.6 GCC Module–Metrology Interface

This module provides an interface to the metrology system. For example, it could encode and parse SECS communications to support upload of (pre- and/or postprocess) metrology parameters. It also provides a capability to maintain synchronization with the metrology application. Note that in cases where the metrology is fully
integrated (hardware and software) into the tool, the tool and metrology tool components may utilize the same physical interface. In this case the Tool and Metrology Interface GCC modules may share I/O resources.

9.2.1.7 GCC Module–History

This module produces dynamic graphical or tabular information of the event traces and control history of the process. Specifically, it displays and logs the controller inputs, controller outputs, and control run number. Note that this module does not impact the operation of the control system in providing R2R control advices.

9.2.1.8 GCC Control Rule Database

The GCC database data contains the control routing scheme that defines the methodology for servicing events to the GCC through invoking methods in the various GCC modules. Specifically, the database defines, for each event, the module invocation order, and the methods and I/O parameter types associated with each invocation.

9.2.2 Process Optimization and Control Scheme

The process optimization and control scheme utilized by the control solution is illustrated in Figure 9.8.12 This multibranch selection mechanism is enabled by the GCC methodology, and allows for the complementary utilization of process control

---

FIGURE 9.8 Process optimization and control scheme.
and optimization methods for R2R control. Each branch consists of a single optimization or control algorithm (e.g., a commercially available algorithm). These branches are utilized in complementary fashion to enhance the robustness of the controller. In order to achieve complementary utilization of these algorithms, a mechanism must be utilized to identify which combination of the available process analysis branches should be invoked so that the (statistically/heuristically) optimal or “best” advice for the process recipe will result. Further, after the process has been analyzed by the one or more indicated algorithm branches, a “best” set of advices (recipe) must be derived that results from comparing the weighted advices of the selected branch algorithms. This “best” recipe is the main output of the R2R controller.

The branch selection algorithm utilized by the R2R control paradigm addresses a number of issues in combining algorithms. First, the knowledge of the state of the process that could be utilized to select R2R control algorithms is diverse, oftentimes vague, and generally difficult to capture in crisp form. Second, the same can be said of the knowledge pertaining to the domain of applicability of each branch algorithm. Indeed, this knowledge is also vague. Third, as there is no common taxonomy for defining the domain of applicability, it is difficult to combine the knowledge into a deterministic knowledge base. Fourth, as there are differing degrees of confidence associated with the advices provided by algorithms, a weighting capability must be associated with algorithm advices in a multialgorithm scheme.

The branch-selection algorithm that is part of the implementation description presented in this section is based on a fuzzy logic multibranch solution. This solution addresses the aforementioned issues by utilizing fuzzy logic applied to a knowledge base to determine branch selection. A review of the requirements of the branch selection algorithm and the available methods for its implementation reveals that fuzzy logic is indeed an ideal mechanism for the development and use of the required knowledge base. Specifically, the fuzzy system is attractive for many reasons. First, it makes effective use of vague or nonexact information in conjunction with deterministic information, and it can provide suggestions with a limited amount of knowledge. This is important because, in many cases, the available knowledge on branch selection for R2R control is somewhat vague and limited. Second, the developed fuzzy system effectively captures knowledge in “human language” format (a form in which much of R2R process control knowledge exists). Third, the system is capable of suggesting the “best” alternative(s) in situations where there may be many viable solutions. This is important because in many cases the domain of applicability of R2R algorithm branches overlap. Finally, the developed system can relate degrees of confidence with suggested solutions. Thus, it inherently provides a mechanism for the weighting of advices from each branch invoked for a particular process run.

The fuzzy mechanism developed incorporates both fuzzy and nonfuzzy knowledge into a data knowledge base. This knowledge base is incorporated/linked into the database of the GCC R2R control enabler, and the resulting system is thus able to enforce routing information relating to which control thread(s) to invoke for a particular process run (see Figure 9.8). Thus, the database contains a schema for the storage of fuzzy and nonfuzzy rules as well as the interaction with the GCC
database. It also contains a “fuzzifier” that categorizes process run data as necessary so that it may be utilized by the available rules.\textsuperscript{13}

The rule syntax allows expression of rules that advise \textit{for} or \textit{against} an action. As an example, Figure 9.9 is an illustration of a valid rule base. The rule base contains rules that relate the usefulness of one of two algorithms (a linear approximation control algorithm and a quadratic approximation optimization algorithm) to the correction of process error. Note that each rule contains a predicate, operator, action, and certainty factor (a number between zero and one indicating the confidence or believability of the rule).

In order to handle information that is somewhat contradictory (e.g., rules 2 and 4 in Figure 9.9) we utilize a method introduced by Chaudhry et al.\textsuperscript{13} With this method, all rules associated with a particular action (i.e., the action of choosing a particular branch) are partitioned into two sets, those recommending \textit{for} the action and those recommending \textit{against} the action. For each of these sets a \textit{confidence} is derived (a number between zero and one) by applying fuzzy set theory. Thus, an \textit{Upper Confidence} level advising \textit{for} an action and a \textit{Lower Confidence} level advising \textit{against} an action are both derived.

After both confidences are derived, we then provide the following technique for making a decision of branch selection.\textsuperscript{12}

1. Upper and Lower Confidence levels computed are combined into a tuple $(X_1, X_2)$, where $X_1$ represents the degree of support for an action, and $X_2$ represents one minus the degree of support against an action ($0 \leq X_1, X_2 \leq 1$).
2. The tuple is then plotted as shown in Figure 9.10. This graph is a two-dimensional representation of the support associated with an action. Thus, the tuple associated with an action plotted on the graph depicts the degree to which the action is confirmed or refuted, and the degree to which the rule set associated with the action is contradictory or supplying a low amount of information. In Figure 9.10, rules strongly advise for choosing algorithm “A” and against algorithm “B.” Rules associated with algorithm “C” are contradictory (i.e., some are expressing strong support for the algorithm while others are expressing strong support against its utilization). There is little confidence in the knowledge-base information associated

\begin{verbatim}
Rule 1: if error > sigma, then quadOpt, 1.0
Rule 2: if error > sigma, then not linearApp, 1.0
Rule 3: if error < and error > 2, then quadOpt, 0.7
Rule 4: if error < and error > 2, then linearApp, 0.4
Rule 5: if error < 2 and error > 1, then quadOpt, 0.5
Rule 6: if error < 2 and error > 1, then linearApp, 0.7
Rule 7: if error < 1, then not quadOpt, 1.0
Rule 8: if error < 1, then linearApp 1.0
\end{verbatim}

\textbf{FIGURE 9.9} Fuzzy rule base for a two-branch controller.

© 2001 by CRC Press LLC
with algorithm “D.” Placement of points “E” and “F” indicate intermediate levels of confirmation and rejection support.

3. Steps 1 and 2 are applied to rule sets associated with other actions (that suggest other branches).

4. The graph is partitioned into three regions as shown in Figure 9.10: a triangular region of strong confirmation recommendation, a triangular region of strong rejection recommendation, and the remaining area representing weaker recommendation. Note that the two partitioning lines are lines of equal confidence.

5. A rule is applied to the graph to determine which action(s) to take, i.e., which optimization and/or control algorithm(s) to invoke. An example of such a rule might be:

   \[
   \text{if (there is at least one action in the strong confirmation region)} \]
   \[
   \text{then (take all actions in the strong confirmation region)} \]
   \[
   \text{else if (there is at least one action not in the strong rejection region)} \]
   \[
   \text{then (take the action closest to the strong confirmation region)} \]
   \[
   \text{else (take no action)} \]

   Note that if this rule were applied to the event depicted in Figure 9.10, then algorithm “A” would be invoked for the current process run.

In summary, the branch selection method developed utilizes fuzzy logic theory to recommend optimization and/or control branches to be invoked for a particular
process run. The method is flexible in that it supports fuzzy rules, such as rules that might be attained from the process engineer, as well as nonfuzzy rules. Further, the method is adaptive as it can incorporate new rules (relating to existing or new branch selection decisions) on-the-fly. This property combined with the GCC learning mechanism capability results in an R2R control framework that is very dynamic and adaptable.

9.3 MAPPING THE GCC DESIGN TO R2R CONTROLLER DESIGN REQUIREMENTS

The design requirements for integrateable R2R control were presented in Chapter 8. The GCC concept was developed to address these requirements. In this section we analyze the GCC design and implementation to illustrate that this approach does indeed address the design requirements. We provide this analysis by restating each requirement and describing how this requirement is addressed with GCC-enabled solutions. The mapping exercise is presented here to illustrate its utility as a tool to evaluate the capability of an integrated R2R control solution.

1. **Process Independence** — The GCC has a well-defined generic interface to the process and equipment being controlled. Because the control scheme mechanism is dynamic, the GCC can be programmed with control knowledge for any process. Because software modules and equipment interface modules (including third-party modules) can be integrated into the system, any appropriate process or equipment-specific modules can be integrated into the generic R2R control scheme. In other words, the modularity of the system also allows for the modularization of generic capability from process-specific capability. The dynamic capability allows for quick (re)configuration to new processes and new process control schemes.

2. **Plug-and-Play Integration of External Software Modules** — As defined above, the GCC provides a well-defined interface for dynamic integration of external software modules. These modules include control algorithms, data loggers, equipment and metrology interfaces, etc. The use of the object-oriented approach in some GCC implementations, which will be discussed further in Chapter 11, further simplifies the plug-and-play capability. The key to this capability is the utilization of the GCC database as the (sole) interface point between modules and the elimination of all direct module-to-module interaction.

3. **Dynamic Control Scheme** — When the GCC receives an event (message or timeout), the control scheme determines the action to be taken in order to service that event. That control action consists of a sequence of messages (i.e., routine calls) sent to modules to carry out the desired task. In other words, the control scheme is a list of event-action pairs, where an event represents a high-level command or message to the GCC, and the action is a list of module invocations. The control scheme is stored as the data in a database rather than in static code (see Section 9.1); thus, it is
persistent, portable, and able to be modified, even during the execution of a process if necessary. This dynamic control scheme is also critical to the GCC’s qualities of process-independence, and plug-and-play integration of external modules.

4. Complementary Operation of Multiple Control and Optimization Methods — While multiple control algorithms are not always required to achieve effective R2R process control today, the GCC has the ability to utilize several controllers and optimizers concurrently, as defined in Section 9.2, taking the advice given by the most appropriate controller or optimizer to use for the current run.

5. Ability to Provide R2R Control with or without In Situ Control — The GCC is designed as a major component of a multilevel control system that includes in situ equipment and process control operating in conjunction with sequential control components such as R2R control. These control loops operate concurrently in a hierarchical feedback fashion, as shown in Figure 9.7 and Figure 9.8 in the Introduction to this book. At the factory level, the GCC R2R control elements interface with factory-level control (potentially enabled through the GCC engine or with the active controller, see Part 6 of this book), also in a complementary fashion.

6. Platform Independence — The GCC implementation utilizes an open CORBA development and runtime platform. It provides a graphical, object-oriented user and software-development environment that is operating-system-independent, i.e., applications developed on one platform can be recompiled and utilized on another supported platform (current operating environments supported include Windows NT*, UNIX, and MacOS**).  

7. User Friendliness and Control Integration Migration Path — While the evaluation of user friendliness is somewhat subjective, the GCC solution has a number of qualities that enhance the user friendliness of the solution. For example, the ability to incorporate third-party modules, which provide a necessary functionality but also may have familiar user interfaces and familiar operation, facilitates user friendliness. The modular design of the system enhances user friendliness by allowing for modularization of the user interface as well as system understanding, and minimizing the complexity associated with modular interaction. The fully object-oriented, event-driven design of the system also enhances user friendliness because it results in operation that is more intuitive, and can be more easily represented graphically. Other factors that enhance user friendliness and the ability to provide a control integration migration path include (1) independence of the internal control solution from the I/O characteristics of the integration environment (see Section 9.1), (2) capability for (re)configuration of the system to evolving control needs, (3) capability for GUI reconfiguration, and (4) compliance of the implementation with SEMI standards for communications and architecture.6

---

* Microsoft Corp.
** Apple Computer Corp.
9.4 SUMMARY

The generic controller approach to portable and configurable sequential systems has been presented in this chapter. The GCC is applicable to a number of control paradigms; however, it is especially suited to address the requirements of R2R control solutions in semiconductor manufacturing. The example presented in Section 9.2 illustrates many of the advantages of the GCC when applied to R2R control, such as the capability of incorporating third-party modules to achieve desired functionality while maintaining flexibility of the control scheme. This allows, for instance, the complementary utilization of multiple control algorithms to achieve more robust control (as described in Section 9.2.2). It also allows the utilization of a “learning mechanism,” as described in Sections 9.1.5 and 9.1.6, to enhance the control scheme in response to new or unforeseen events.

The qualities of the GCC or any other solution enabler complying with the R2R control system requirements of Chapter 8 allows for the specification of (cost- and quality-) effective designs for R2R control. One such design, applicable especially to retrofit R2R control applications, is presented in Chapter 10. This is followed, in Chapter 11, by examples of utilization of GCC-enabled R2R control solutions in process control.

APPENDIX A: OVERVIEW OF ENTITY RELATIONSHIP (E-R) THEORY AND CONSTRUCTS

The E-R approach to database modeling provides semantics for conceptual design of databases.10–11 With the E-R approach, database information is represented in terms of entities, attributes of entities, and relationships between entities, where the following definitions apply. A common form of modeling semantics corresponding to each definition is illustrated in Figure 9.11.

![Figure 9.11 Entity relationship modeling semantics.](image-url)
1. **Entity**: A principle object about which information is collected. For example, in a database containing information about personnel of a company, an entity might be “Employee.” In E-R modeling an entity is represented with a box.

2. **Attribute**: A label that gives a descriptive property to an entity, e.g., name, color, etc. Two types of attributes exist. Identifier attributes distinguish among occurrences of an entity, e.g., social security number. Descriptor attributes merely define an entity occurrence, e.g., gender, weight, etc. In E-R modeling an attribute is represented with an oval tied to the entity (box) to which it pertains. In many cases, attributes are not included in the E-R model.

3. **Relationship**: A relationship is a connectivity exhibited between entity occurrences. Relationships may be one to one, one to many, or many to many, and participation in a relationship by an entity may be optional or mandatory. For example, in the database containing information about personnel of a company, a relation “married to” among Employee entity occurrences is one to one (if it is stated that an employee has, at most, one spouse). Further, participation in the relation is optional, as there may exist unmarried employees. As a second example, if company policy dictates that every employee have exactly one manager, then the relationship “managed by” among Employee entity occurrences is many to one (many employees may have the same manager), and mandatory (every employee must have a manager). In E-R modeling a relationship is represented with a diamond if it relates one or two entities, and is represented with an n-sided polygon if it relates n entities (where n is greater than two). Connectivity in a relationship is denoted with shading of the diamond; a connectivity of “one” is denoted with the appropriate portion of the diamond unshaded while a connectivity of “many” is denoted with the appropriate portion of the diamond shaded. Optionality of entity participation in a relationship is indicated with a ring around the line segment between the entity and the relationship.

A detailed description of E-R model semantics, extensions, and E-R modeling techniques may be found in Teorey.10

**APPENDIX B: DESCRIPTION OF ACTIVITIES ASSOCIATED WITH ROUTINE CALLS ASSOCIATED WITH EXAMPLE 1 — ETCHING A WAFER**

The following is a detailed description of the activities associated with the calling of each of the routines associated with “Action #4” in Example #1 (Section 9.1.7).

**Routine#27: RemoteCommandACK(SUCCESS, 0):** This routine prepares the data to send a Remote Command Acknowledge message (Stream 2 Function 22) to Port#0 (for export to the parent controller). The first routine parameter is the data
item to be sent in the message while the second parameter indicates the port number to which the message is to be sent. For this call to the routine, the data item to be sent with the message is SUCCESS, which is defined to be a 2-byte signed integer with a value of zero. The message and destination information is passed to the message parser by this routine, which directs the formatting of a Stream 2 Function 23 message. The routine then returns control to the conductor. The message parser will format the message into the proper SECS-II syntax and forward it to the I/O interpreter module associated with Port#0 for transmission to the parent controller.

Routine#29: PPLoadInquire("Etch500", 1000, 1): This routine prepares the data to send a Process Program Load Inquire message (Stream 7 Function 1) to initiate the download of a recipe to an equipment controller. The first two routine parameters, respectively, are the Process Program ID and Length data items to be sent in the message, while the third parameter indicates the port number to which the message is to be sent. For this call to the routine, the routine passes the "Etch500" and 1000 data items to the message parser and directs the formatting of a Stream 7 Function 1 message. The routine then returns control to the conductor. The message parser will format the message into the proper SECS-II syntax and forward it to the I/O interpreter module associated with Port#1 for transmission to the RIE#1 Equipment controller.

Routine#2: ActivateDBMessage(7, 2, 1, 1): This routine modifies an entry in the Message table, tagging it as active. A tag (described above as the “Active?” parameter) is used with this implementation to indicate whether a message is expected. If a message is expected, it is tagged “YES.” If an expected message arrives, the foreign key of Action# indicates the action to take to service the message event. However, if a match is found between an incoming message and a message tagged “NO,” the message is unexpected; the generic controller may invoke a user interface routine in this instance to learn how to deal with the unexpected event. The four routine parameters, respectively, are the Stream#, Function#, Data, and Port#. In this instance the above PPLoadInquire() routine call has resulted in the scheduling of a Stream 7 Function 1 message to be sent to the RIE#1 Equipment Controller. To be compliant with the SECS protocol, the recipient of the message is expected to respond with a Stream 7 Function 2 message in a timely fashion. The generic cell controller should be made aware that such a response is expected, i.e., the appropriate database message table entry(s) should be tagged as active. This is the function of the ActivateDBMessage routine. For this call to the routine, the message portion of the database would be searched for the Stream 7 Function 2 message, associated with Port#1. The routine assumes that a single match will be found. When a match is found, the routine modifies the database, tagging the entry as “ACTIVE,” as shown in Figure 9.5 (bubble call-out annotation). The routine then returns control to the conductor.

Routine#4: TimerAdd(CONVERS, 7, 2, 99, 0, 1): This routine adds an entry to the InternalEvent_Instance table (not shown in Figure 9.5). The parameters of the routine call indicate, respectively, the type of timer, the Stream and Function numbers of the message associated with the timer, the Action# to be associated with the timer, the timeout value (zero indicates use default — the default value may be found in
the Timer table), and the Port# associated with any message timer. For this particular call, a conversation timer is to be entered where the reply expected is a Stream 7 Function 2 message arriving via Port#1. The routine searches the InternalEvent table for the (requested) default value and creates a new entry in the InternalEvent_Instance table with the indicated parameters along with TimeStart (the current time). The routine then returns control to the conductor.

Routine#3: LogEvent(“Etch.Request.500.Ang.SiO2.RIE1”): For this implementation all events are logged. This routine opens up a log file, creates a new line at the end of the file, and enters the text “Etch.Request.500.Ang.SiO2.RIE1” along with the current date and time. The routine then returns control to the conductor. Note that this routine does not impact the generic cell controller database and does not result directly in the sending of any SECS-II formatted messages.

ACKNOWLEDGMENTS


REFERENCES


In Chapters 7, 8, and 9 we defined requirements and described technologies for providing integrateable R2R control solutions in semiconductor manufacturing. Utilizing these requirements and technologies, there are a number of plausible R2R control solution designs. The particular design depends to a large extent on the processing environment and, specifically, the capability for modification of the existing equipment and factory control systems.

In this chapter we explore an R2R control solution design in a retrofit environment where the equipment and factory control systems are in place and there is a desire to add an automated R2R capability to the system with minimal modification to the existing software. Thus, the only available interface to the equipment, metrology, and factory host elements is generally through SEMI standard GEM/SECS communication links. The SEMATECH Control Systems Requirements Specification (CSRS — see Chapter 7) provides elements of an integration requirements specification for APC whose main component is SEMI standards. In applying these requirements to existing systems in which R2R control is to be integrated, we see that a piggyback solution is suggested (a definition is provided in Chapter 7). In this chapter we provide a specification for this piggyback solution in terms of the basic internal architecture and the integration environment. Note that this is not a standard solution, but merely one solution design that meets the requirements for integration identified in Chapter 8. We conclude this chapter by identifying advantages and disadvantages of this solution and suggesting alternative design possibilities.

10.1 REQUIREMENTS OF A PIGGYBACK R2R CONTROL SOLUTION DESIGN

The SEMATECH CSRS provides elements of an integration requirements specification for APC. The CSRS for current and near-term next-generation systems provides a partial requirements specification using existing standards that specify in a common and reusable way the addition, deletion, or modification of sensors, algorithms, applications, and control capabilities in semiconductor processing systems at the user skill level. This CSRS is considered a partial solution because the standard
(plus additional, not-yet-standard specifications) set does not yet exist to specify a complete CSRS. A migration path to future CSRS systems is also provided to maintain alignment with the envisioned CSRS.²

Although the CSRS and the related SEMI standard set is incomplete with respect to completely specifying an R2R integration solution, they can be utilized to identify requirements and design parameters for adding an R2R capability to an existing system.

The requirements of a piggyback R2R controller can be generally stated as follows:

1. The system must integrate a metrology capability that captures one or more process quality parameters on a run-to-run basis. An example is a postprocess thickness sensor that could evaluate the remaining thickness of a layer and the uniformity of that layer. A second example is a metrology system that delivers an in situ “footprint,” or process/wafer parameter trace, for a process run from which process and product health may be derived.
2. The process must utilize the integrated metrology and provide automated run-to-run control process improvement via the automatic adjustment of recipes on the tool. The recipe input parameters to be adjusted are determined by the control systems integrator working within the constraints supplied by the user and the equipment.
3. The piggyback controller configuration must operate within the existing control, computer-integrated manufacturing (CIM), and manufacturing execution system (MES) environments. That is, there should be minimal, if any, reprogramming of the equipment control system or factory-level CIM and MES systems required.
4. The software solution must be reliable and maintainable and allow for integration of third-party software.
5. The solution must utilize existing SEMI standards wherever possible and provide a migration path to the envisioned long-term CSRS.⁵

### 10.2 RUN-TO-RUN PIGGYBACK CONTROLLER DESIGN

In order to achieve these requirements, the following design parameters, based on the CSRS for current and near-term next-generation solutions, are specified:²,³

1. The controller shall utilize GEM to communicate with the metrology and process tools. Additionally, if supported, the controller shall also utilize any applicable specific equipment model (SEM) standard such as the pending metrology SEM standard.³ Current process and metrology system tools should support GEM. GEM provides a mechanism for gathering metrology data. It also provides a mechanism for adjusting process program
parameters via equipment constant settings;* this capability is not required for GEM compliance, but is specified as an application note in GEM to support process parameter modification for APC.**

2. The internal structure of the controller shall be of a form that can migrate to an APC Framework-compliant solution.*** What this means is that the software structure of the controller should be object-oriented, should utilize CORBA or a similar object communication mechanism to support communication between functional modules, and should define a pluggable capability for the functional modules (i.e., a well-defined, object-oriented interface to each module). In order to support this internal structure in a GEM communication environment, software conversion modules or wrappers should be provided that convert the internal CORBA communication to GEM messaging as necessary to achieve run-to-run control.

3. The controller should provide a GEM pass-through capability, thereby maintaining equipment connectivity with the host, MES and factory level

---

* There are a number of methods that could be utilized to update process program parameters to achieve APC. Depending on the capabilities of the tool equipment control system software, equipment constants could be used to relate parameters of the updated process program. Equipment constants could also be used to relate suggested modifications to process program parameters from the stored process program, i.e., the constants contain only the +/- “tweaks.” The former method is preferred because it ensures data integrity between the controller and tool. Alternatively, the entire process program could be downloaded, but this results in an enormous amount of communication overhead. Finally, remote commands or variable parameters could be used. The reader should refer to the latest versions of appropriate SEMI standards (e.g., GEM) as well as equipment control systems specifications when determining which process program parameter update method to utilize.

** This effort is being pursued by the SEMI Equipment Control Systems Task Force (ECS-TF). For more information on this task force, refer to the CSRS document, or contact SEMI at www.semi.org.

*** Common Object Request Broker Architecture (CORBA) is a concept published by the Object Management Group (OMG), which specifies a common architecture for distributed object systems.
CIM system. The pass-through operation is illustrated in Figure 10.1 and is described in more detail in Section 10.3. The controller communicates with the equipment, metrology system, and host in such a way as to provide the host with an interface that is the same as the original equipment and metrology systems interface.

The resulting specification for the piggyback system addresses all of the system requirements identified, is realizable with today’s technology and standards, and is compliant with the CSRS specification, Version 2.0.2

10.3 PASS-THROUGH OPERATION

A key component of the controller design is the implementation of the SECS “pass-through” capability. As noted in Section 10.1, the piggyback controller configuration must operate within the existing CIM and MES systems, i.e., there should be no reprogramming of the factory-level CIM and MES systems required. What this means is that the SECS interface factory host presented by the controller should be the same as that presented by the tool and/or metrologer. In other words, the factory host software should not require modification and should see (interface to) a system that looks the same as a tool and/or metrologer without a piggyback controller inserted. In order to achieve this transparent capability, the following requirements must be met (using the tool as the example):*

1. The piggyback control must provide a synchronization capability between the communication state of the tool, as specified in the SEMI GEM standard, and the communication state presented to the host.2,5 The piggyback control system must move the tool communication system to a COMMUNICATING state, as shown in Figure 10.2, before it can effect run-to-run control.** Since the factory host did not move the tool into this state, it should still perceive the tool as being in the NOT COMMUNICATING state. Thus, the piggyback controller must provide an interface to the factory host indicating a NOT COMMUNICATING state until such time as the host wishes to communicate with the tool; at that time the controller will guide the factory host, utilizing the state model presented in Figure 10.2, to synchronization with the tool to the COMMUNICATING state.

2. The piggyback control must provide a synchronization capability between the control state of the tool and the control state presented to the host. The piggyback control system must move the tool communication system to an EQUIPMENT ON-LINE state as shown in Figure 10.3 before it can effect run-to-run control. Since the factory host did not move the tool into

* The same requirements apply to the pass-through capability to be provided between the metrology system and the factory host.

** The state table illustrations in this subsection utilize Harel notation to depict (nested) states and state transitions.10 Note that these equipment behavior models are specified in the GEM standard.5
this state, it should still perceive the tool as being in the UNKNOWN state until such time as it begins to bring the equipment ON-LINE. Thus, the piggyback controller must provide an interface to the factory host indicating the appropriate default state, and guide the factory host, utilizing the state model presented in Figure 10.3, to synchronization with the tool.

3. The piggyback controller must simultaneously support R2R control messaging along with factory host-to-tool messaging. This is accomplished utilizing a priority-based scheme. R2R control messages are of a lower priority than pass-through messages. Thus, an R2R message transaction can only be initiated when there are no outstanding pass-through transactions. If a pass-through transaction is initiated while an R2R transaction is open, both transactions must be serviced concurrently.
4. The piggyback controller may have to provide a “heartbeat” capability with the tool and the host. This capability, usually accomplished with periodic S1F1/F2 transactions, verifies the health of the SECS link. If the equipment supports a heartbeat capability, the host must also be presented with this heartbeat capability.

5. SECS transaction timing issues are complicated by the insertion of the piggyback controller between the factory host and the tool. SECS T1 and T2 timers can be supported at tool-to-host direct link levels (i.e., can remain unchanged); however, the T3 (transaction) timeout value may have to be lengthened to take into consideration any delay added by the piggyback controller. In most cases, however, T3 is set to a large value (10’s of seconds) and the piggyback delay is insignificant.

10.4 MIGRATION OF THE PIGGYBACK SOLUTION

A specification for a piggyback controller that supports R2R control that has been presented utilizes GEM at all interfaces and provides a pass-through capability between the tool and the host. The internal structure of the specified controller is such that can easily migrate to an APC Framework-compliant solution when such solutions are fully realizable.

The piggyback solution, though representing a method for rapidly incorporating an APC capability, is not considered a final solution for that capability. As implementations verify the capability, this capability may be migrated into the tool or up into the factory control system as illustrated in Figure 10.4, while the piggyback controller may be retained for rapid deployment and testing of new APC capabilities. Adherence to the CSRS in developing the piggyback controller and migrating to the final solution will facilitate the migration and result in a more cost-effective and reliable system.

![FIGURE 10.4](image) Migration of the piggyback control capability.
10.5 ADVANTAGES AND DISADVANTAGES OF THE PIGGYBACK APPROACH, AND ALTERNATIVES

The piggyback design just presented represents just one solution design for integrating an R2R control capability into an existing system. This design has a number of advantages, including the following:

1. It utilizes existing (SECS/GEM) interfaces on the equipment, metrology, and host components.
2. It maintains the connectivity between the equipment and the host; in many cases the host software and host operation are not impacted by the addition of the R2R control component.
3. A well-defined mechanism is specified for integration of R2R controller SECS/GEM commands with the existing communication between the host and equipment.
4. The internal architecture of the controller is such that a migration path is available toward integration directly into a next-generation tool, or onto a (APC Framework-compliant) factory backbone network.\textsuperscript{11}

Experience in utilizing this pass-through piggyback form of control solution has also revealed a number of disadvantages to this approach:

1. Because the piggyback solution is part of the equipment-to-host communication link, a failure in the piggyback system would generally cause a failure in that line. This problem may be addressed by providing a relay-invoked hardware bypass system similar to that shown for the real-time control piggyback system in Figure 7.6 of Chapter 7.
2. The fact that the R2R controller is modifying recipe parameters on the tool could lead to synchronization and traceability problems between the tool and the host. For example, over time, parameters being controlled may be altered significantly, which may impact the operation of host-level applications such as schedulers. It is important that the R2R controller actuation capability be limited so that it does not impact the host operation, or that the appropriate GEM event reports be set up so that the host is informed when parameters of interest are altered to the point that they may change host operation. Note that this may already be common practice in systems where the host downloads the baseline recipe for a process, but the operator is allowed to modify the recipe within certain bounds.
3. Although the piggyback specification defines a run-to-run (i.e., wafer-to-wafer or batch-to-batch) control solution, it does not indicate when, within a run, the control modification should take place. For example, if the metrology and run-to-run modeling and control for run $n$ is completed while run $n + 1$ is being processed, should the control advice be executed on run $n + 1$ as soon as it is available, or should the control recipe modification only occur between runs (i.e., after run $n + 1$ has completed,
but before run \(n + 2\) commences)? Unfortunately, considering the wide variety of process types and equipment solutions for a particular process, there is no clear answer to this question. One plausible solution might be to implement the control actuation capability to mimic the actuation capability provided to the user (through the equipment user interface) for the particular process instance.

4. The GEM standard specifies a point-to-point, host-to-equipment communication line. Therefore, because both the factory host and R2R controller are communicating with the equipment, the piggyback solution technically violates the GEM standard. This issue can be addressed by migrating the R2R controller to the factory backbone or into the equipment as shown in Figure 10.4. Note that this issue is also addressed in the APC Framework environment where host-to-equipment communications are peer-to-peer and multiple peer-to-peer connections are allowed. \(^7_8\)

5. Although the GEM standards provide mechanisms for the reporting and modification of recipe (“process”) parameters, it doesn’t explicitly require that process parameter modification be supported. Recent efforts within SEMI are focused on addressing this issue by providing application note additions to GEM to specify preferred methods for “process program parameter modification.” \(^6\)

6. Space in a cleanroom fabrication environment is generally very costly. The cost of the footprint of additional computer (including monitor) may be unacceptable in many cases. This issue may be addressed by having the R2R control element exist logically as a separate application entity in either the equipment control system or in the factory control system, as shown in Figure 10.4. Depending on the desired method for interaction between applications on a single hardware platform, there may be a need for GEM connectivity to the R2R controller. This issue is discussed later in this chapter.

The piggyback solution presented here is provided as one example of a solution for adding an R2R control capability to an equipment system. The following is a list of alternative solution approaches:

1. *Second GEM Port on Tool:* Using the same communication capability defined for the piggyback controller, an integrated R2R control capability can be achieved by adding a second GEM port to the tool. The advantage of this approach is that the pass-through capability required for the piggyback controller defined in this section may not have to be implemented, since connectivity between tool and factory host is supported through the original GEM port. Further, each GEM port has a single host that is consistent with the GEM specification. The obvious disadvantage is that, in most cases, tools do not support a second GEM port; thus, a significant amount of communication software enhancement may be required at the tool. Also, since the R2R host has no capability for monitoring the communications between the tool and factory host, there is a potential for
conflict between the two hosts. For example, each host could provide different process parameter values for the same run.

2. Additional Non-GEM Port on Tool: If a pass-through capability is not required for the piggyback controller, there may be no requirement that the R2R integration capability be GEM-compliant. If this is the case, a second communication port can be configured on the tool to support communications for R2R control. The advantage of this solution over a GEM port is that a much simpler communication capability can be utilized that only supports messaging required to implement R2R control. Another advantage of this solution, in addition to those identified for alternative 1 above, is that more common interface development tools, such as Ethernet TCP/IP builders, could be used. In addition to the disadvantages defined for alternative 1, the primary disadvantage of this alternative is that the solution is nonstandard.

3. Fully Integrated Equipment-Level Solution: As illustrated in Section 10.4 (see Figure 10.4), a possible migration path for the piggyback controller is to move the capability into the equipment controller. An important consideration is the impact of the piggyback controller operation on the reliability of the equipment controller software. One way to minimize this impact is to structure the R2R system as a separate set of applications that communicate with the equipment control application(s) via distributed object techniques such as CORBA. The advantages of such a solution are numerous. The footprint and additional screen required for the piggyback system are eliminated. The interface between the tool and host is simplified. The connectivity to the tool (applications) and metrology can also be simplified. The functionality, synchronization, etc., between the tool and R2R controller can be enhanced. The main disadvantage of this approach is the effort required to integrate the R2R and equipment control applications on a single (computer) operating system platform. The communication integration can actually be more straightforward than with many piggyback scenarios. There is an added effort, however, required to integrate the GUIs.

4. Factory-Level Backbone Solution: As also illustrated in Section 10.4 (see Figure 10.4), a possible migration path for proven piggyback R2R control solutions is up onto the factory backbone. This concept is supported in SEMI APC standardization efforts, where elements of the interface of this APC “plug-in” are defined. The advantages of this approach are numerous and are illustrated in Figure 10.5, using a Generic Cell Control (GCC)-enabled R2R control solution as an example (see Chapter 11). The backbone solution allows greater access to the capability by tools as well as users. As shown in Figure 10.5, this allows metrology to be reused as premetrology and postmetrology, thereby reducing metrology system costs. The R2R controller can be set up to provide R2R control for multiple tools (e.g., in this example, through multiple instantiations of the GCC R2R controller class in GCC implementations). Perhaps the largest advantage, though, is the visibility of the controller to factory level, multiprocess
data, and process targets. This allows the control solution to be tuned to factory-level control as opposed to equipment-centric control. This concept of R2R control as part of a total factory solution is discussed further in Part 6 (Chapter 18). The main disadvantage of this solution is the effort required to integrate the R2R solution on the factory backbone. This effort is compounded by the fact that there is a relative lack of standardization of communications at this level (the APC framework specification is not widely implemented and does not completely specify the operation of the controller at the factory level). Note that in many instances, the “optimal” integration strategy for the R2R capability is to limit the interaction to the factory database that contains the metrology data and process recipes. The controller simply accesses the metrology data and updates the appropriate recipe process parameters. This approach limits the requirement of interaction of the controller with tool and metrology systems, thereby providing a highly nonintrusive enhancement to the factory.

10.6 SUMMARY

A piggyback controller design has been presented in this chapter that supports flexible R2R control in the semiconductor manufacturing SEMI standard environment. Further, a migration path is supported for integration of the R2R capability at the factory level or equipment level, while retaining the piggyback capability for testing new advanced process control capabilities. Alternative piggyback controller designs have been presented to various interface and performance requirements imposed by equipment and/or factory systems.

In the next chapter we explore the application of R2R controller design by describing two integrated R2R control solution examples, namely a GEM/SECS piggyback solution and a fully integrated (at the equipment level) solution.
ACKNOWLEDGMENT

Some of the material presented in this chapter is derived from Reference 2, and is reprinted with permission.

REFERENCES

1. Moyne, J., “Integration of Run-to-Run Control into Existing and Next Generation Chemical-Mechanical-Planarization Tools,” SEMATECH AEC/APC Workshop IX, Lake Tahoe (September 1997).
2. SEMATECH Control Systems Requirements Specification V2.0, SEMATECH Technology Transfer Document #96123222B-ENG (December 1997).
In Chapters 8 through 10 we presented (1) key requirements for the internal structure of the R2R controller, (2) an example R2R solution-enabling technology meeting these requirements, (3) a design for an R2R piggyback controller, and (4) design alternatives including fully integrated solutions at the factory and equipment level. In this chapter we provide brief examples of two R2R control solution designs presented, namely a piggyback solution, and a fully integrated equipment solution.

In both examples, the process being controlled is chemical mechanical planarization (CMP, or “polishing”), which is described in detail in the Introduction to this book. For purposes of discussion in this chapter, the CMP process being controlled is described as follows: the process is basically a surface planarization method in which a wafer is affixed to a carrier and pressed face-down on a rotating platen holding a polishing pad. Typical process control metrics are remaining thickness and radial uniformity control, while tunable inputs include polish time, platen speed, carrier speed, downforce, backpressure (between wafer and carrier), and various conditioning parameters. It is the relationship between process control metrics and tunable inputs that motivates process control.

The fundamental R2R control-enabling technology utilized is the Generic Cell Controller (GCC — see Chapter 9), and all solution requirements defined in Chapter 8 are met. The GCC solution incorporates (1) interface modules for interfacing to the equipment, metrology, and factory host; (2) a control module; (3) a history module for data logging and presentation; and (4) an alarms module that monitors R2R data with respect to control and specification limits, and reports limits violations to the equipment. The interface modules in the piggyback solution provide a pass-through capability (external to the GCC) and prioritize pass-through communication over R2R control communication. The internal architecture of the GCC is CORBA,* and the GCC database provides the control scheme for servicing events through routing information between the various modules.

* Common Object Request Broker Architecture, see Chapter 10.
11.1 R2R CONTROL AND THE CMP PROCESS

Chemical mechanical planarization (CMP) is accepted as a critical component of semiconductor manufacturing. As the migration to 300-mm wafers and smaller features continues, issues such as increased equipment reliability and up-time, increased throughput, and reduced scrap will have even greater impact on the cost of ownership (COO). As a result, efforts will continue to be focused on improving these parameters to maintain competitive advantage.

Two major sources of lost productivity in CMP tools are process drift and operator error. Process drift, away from target thickness and uniformity, leads to lost wafers (scrap) and equipment downtime for consumable replacement. Operator error results in lost wafers and equipment downtime, while operator unavailability results in unnecessary equipment idle time and lost production. According to one SEMATECH study, these factors collectively account for up to 50% of overall equipment effectiveness. Thus, there is potential for significant benefit through addressing these factors.

The application of process automation and model-based R2R process control in CMP wafer fabrication has been recognized as having the potential to significantly impact these factors. The fully integrated and automated R2R solutions presented here for CMP process control greatly reduce process drift along with operator error and unavailability, thereby reducing scrap, equipment downtime, and cost of consumables, while increasing yield and throughput. In the remainder of this chapter the design of automated process control solutions for current (piggyback) and next-generation (fully integrated) CMP tools is presented, followed by an analysis of results and conclusions.

11.2 SOLUTION DESIGN

The automated R2R process control solution has been developed as a retrofit for currently existing CMP tools (Figure 11.1) and as an integral component of next-generation tools (Figure 11.2). The retrofit solution is an example of a piggyback solution, while the next-generation design is an example of a fully integrated solution (alternative 3 in Chapter 10, Section 10.5). Either system solution consists of three major components, namely the tool, metrology unit, and controller. In the retrofit solution the R2R controller resides on a separate computer and utilizes a SECS/GEM interface with the tool for communication. In the integrated solution, the R2R controller resides as an application on the equipment controller computer and utilizes the CORBA distributed object methodology to communicate with ActiveX* equipment controller applications. The single platform integrated software architecture is illustrated in Figure 11.3.

Automated R2R control is achieved because the tool, metrology, and control components each have inherent characteristics that collectively provide the run-to-run control capability. Specifically, the tool is a Strasbaugh 6DS-SP or 6ED “Symphony”

* ActiveX is a product of Microsoft Corp.
polisher that has a capability for recipe modification via a remote entity.* In the case of the 6DS-SP retrofit solution, this capability is achieved through the tool GEM interface, while in the next-generation Symphony tool, an internal object-based interface is used (see Figure 11.1 and Figure 11.2, respectively). The metrology unit is a Nova Measuring Instruments thickness sensor that computes average remaining thickness.

* 6DS-SP and 6ED “Symphony” are products of Strasbaugh, San Luis Obispo, California.
thickness and wafer uniformity and has an SECS interface for reporting this information.* The controller is a MiTeX Solutions multivariate run-to-run controller that is capable of simultaneously controlling thickness and uniformity to target.** The controller utilizes the GCC enabling technology described in Chapter 9. The controller implementation automatically accepts both preprocess and postprocess metrology information from the metrology unit. This is accomplished directly through the SECS interface in the retrofit solution, and indirectly through the equipment controller in the next generation solution (see Figure 11.1). The controller is capable of utilizing both premetrology and/or postmetrology measurements, and is capable of providing a level of R2R control regardless of whether these measurements are available for a particular run. The controller utilizes model-based control techniques and a two-stage linear approximation control algorithm with EWMA filtering (see Chapter 3) to derive recipe improvements (e.g., time and backpressure), and automatically delivers these recipe advices to the CMP tool. In the retrofit solution, parameter updates are delivered according to the specification of the GEM application, not for updating recipe parameters. In this retrofit solution, the controller further provides a pass-through communication capability to a GEM compatible host (see Figure 11.1), thereby allowing a host to maintain full GEM communications with the tool (and potentially the metrology unit). Note that this capability is not required in the next-generation fully integrated solution for reasons identified in Chapter 10, Section 10.5.

The R2R controller achieves CMP process control by utilizing metrology data along with dynamic process models and a process history to determine optimal recipe parameters (i.e., advices) for the next process cycle. Further, the control system utilizes an alarms “plug-in” module to monitor both metrology and recipe advice data with respect to control and specification limits. As shown in Figure 11.4, this fully configurable alarms module links limits violations to actions that can be performed by the tool (such as “alarm, finish cycle, and stop”). The controller then

* Nova Measuring Instruments, Rehovoth, Israel.
FIGURE 11.4 Excerpt of user interface to configurable alarms-based control module (piggyback solution).
automatically downloads both recipe optimization advices and any alarming action data to the tool and informs the tool to start processing the next wafer.

Figure 11.5 shows the user monitor interface to the run-to-run controller when performing automated run-to-run process control. Note that communication interfaces to the metrology and tool elements display the uploaded metrology data and downloaded tool recipes, respectively, while a graphical display keeps track of tool input and metrology output history. Note also that details of control algorithm operation are not presented at the operator interface login level, so the controller is viewed as a simple I/O “black box” at this level.

One novel aspect of this solution is that it provides for fully automated CMP process operation. That is, an operator is not required to monitor or operate the tool. The alarms monitor and control capability provides for automatic response to any alarm condition, while the run-to-run process tuning capability keeps the process within specification limits for significantly longer periods of time. The communication capability between the metrology system, control system, and tool completes the automation process while maintaining a pass-through communication capability between the tool and a host.

11.3 R2R ARCHITECTURE

The R2R enabler for both the retrofit and next-generation solutions is the GCC (see Chapter 9). In fact, *the same internal architecture is utilized for both solutions*, and this same solution has been applied to the control of other process types (see, for example, Reference 9). The GCC solution is implemented as a suite of application “modules,” including third-party modules, that are coordinated by the GCC kernel to provide automated R2R control utilizing the GCC paradigm. Some of the modules utilized in this solution (also see Chapter 9) include (1) the tool interface, which
downloads control advices and alarm indications — see (5) below — to the tool;
(2) the metrology interface, which uploads pre- and postwafer metrology data from
the metrology system; (3) the control module, which utilizes two-stage dynamic
linear approximation modeling and EWMA filtered model evolution (see Chapter 3
and 6); (4) a history module, which displays and logs all data received and generated
by the GCC system; (5) an alarms module, which monitors metrology data received
and control advices generated against control and specification limits and determines
appropriate alarm action requests to be sent to the tool; and (6) a GUI module, which
provides a user interface platform for viewing the operation of the system and
accessing the individual module GUls.

The GCC defines a generic module interface, which allows modules to dynam-
ically connect and disconnect from the GCC without any code modification. This
module interface facilitates the passing of arbitrary data, determined at runtime,
between the GCC and the module. It also allows users to develop custom modules
or third-party developers to produce shrink-wrapped modules. This generic and
dynamic interface to software modules contributes to the GCC’s quality of process
independence, and allows the rapid customization of the same basic solution to both
retrofit and next-generation environments as defined above.

The GCC module interface relies on a distributed objects architecture that allows
objects to send messages to other objects in other tasks, or have messages executed
in other threads of the same task. In general, an object sends a message to a remote
object by communicating its own address space with a proxy for the remote object.
The proxy assumes the identity of the remote object; it has no identity of its own.
The application is able to regard the proxy as if it were the remote object. Note that
proxy does not require access to the remote object’s class. It isn’t a copy of the
object, but a substitute for it. Its main function is to make a remote object appear
as if it were local.

The GCC mechanism relies on a special object, called a message event object,
to exclusively provide for interaction between the GCC kernel (“conductor”) and
the various modules. The message event object travels to the modules in an order
specified by the GCC database to execute the plan to service a message or timeout
event. The object carries with it the required methods to invoke and appropriate data
to send to and receive from the module. The common message object approach
further simplifies the module interface to the GCC.

In migrating from the piggyback solution to the fully integrated solution, the
object-oriented modular architecture allows the R2R control system to act as a
separate set of applications residing on the (Symphony) equipment controller. These
applications interact utilizing the CORBA approach. The primary advantages of this
approach are (1) the interface between the R2R control system and the equipment
controller is well-defined and relatively small, thereby tending to minimize the
software integration task; (2) the operation and any failure of the R2R control system
has minimal impact on the equipment controller application(s), thus minimizing any
negative impact on software reliability; and (3) the modular application design and
interaction allows independent maintenance and upgrade of the R2R control and
equipment control systems. One potentially difficult task in integrating R2R control
and equipment control applications is the integration of the user interface components.
Adherence to standard user interfaces, as described in Chapter 8, as well as object-oriented design and flexible object-oriented user interface design tools, lessens the user interface integration task. As an example, Figure 11.6 and Figure 11.7 are user interface screen shots from the Symphony equipment controller illustrating the integration of the R2R component. Note that the user interface of the piggyback solution is maintained (see Figure 11.4 and Figure 11.5, respectively).
11.4 RESULTS OF DEPLOYMENT

The run-to-run component of the control solution described above has been applied in production to simultaneously control both process remaining thickness and uniformity for CMP oxide removal processes. The results (Figure 11.8) compare the process with and without R2R control. Specifically, they show that (1) the controller provides for rapid compensation (tuning) of the standard recipe to bring the process in tune and better achieve multivariate process targets, and (2) the controller compensates for process drift over time and keeps the process in tune, thereby continuing to better achieve multivariate process targets. There are a number of metrics that can be used to quantify this benefit. One of the more common is process capability or $C_{pk}$, which is a measure of process accuracy and variability. The results shown in Figure 11.8 indicate that $C_{pk}$ is improved by over 50% for both thickness and uniformity when compared to the uncontrolled case.* Thus, the effects of the run-to-run control solution include significantly improved process quality, improved

* $C_p$ is a simple process capability index that relates the spread of the specification limits (i.e., the difference between the upper and lower specification limits, USL, and LSL) to the variation of the process, represented by six standard deviations or 6 Sigma. Thus $C_p = (USL - LSL) / (6 \text{ Sigma})$. $C_{pl}$ and $C_{pu}$ relate the process limits to the process average, divided by 3 Sigma. That is, $C_{pl} = (Av - LSL) / 3 \text{ Sigma}$ and $C_{pu} = (USL - Av) / 3 \text{ Sigma}$. $C_{pk}$ is the minimum of $C_{pl}$ and $C_{pu}$. Note that a higher $C_{pk}$ is always preferred.13
process efficiency (due especially to lack of need for rework), and lower scrap that results from process limits violation.

Results of applying the fully automated run-to-run control solution are shown in Figure 11.9. This figure illustrates that the controller automatically modifies time and backpressure individually for each wafer spindle to achieve control of both process remaining thickness and uniformity. The system also monitors both metrology and recipe suggestions against control and specification limits, and generates alarms as necessary to the tool to automatically shut down the tool in the event of a specification limit violation.

11.5 DISCUSSION

Although results presented above have focused on quantifying the process capability and reduced scrap benefits of this R2R control solution, a number of other benefits are also apparent. Specifically, additional benefits resulting from the run-to-run control component of the solution include lower number of required test wafers and increased life of consumables. Test wafer requirements can be reduced because the controller can adjust the process quickly during the pad break-in phase, allowing the transition to product wafers much sooner in the pad break-in curve. This advantage has been further amplified with a “new pad” feature that has been added to the controller.12 This R2R enhancement, which is described in detail in Chapter 18, gives the controller the capability to remember the initial characteristics of the pad (e.g., removal rate) and allows the controller to revert to this remembered state when the pad is replaced.

Another important, but as yet not quantified, benefit of run-to-run control in this example is the increase in consumable life (i.e., pad life). This occurs because the process is continually adjusted within established bounds to compensate for pad wear; thus, the number of runs between SPC-triggered pad replacement events is increased.

The benefits arising from the automation aspect of this example solution have not been quantified. However, noting the hands-off capability for operation with the automated control system in place, it is projected that process performance degradation due to operator error and lack of operator availability (i.e., idle operator) would be virtually eliminated.

11.6 CONCLUSIONS REGARDING R2R CONTROL IMPLEMENTATIONS

A fully automated advanced process control solution for CMP has been described in this chapter. This example application of automated process control clearly illustrates not only the advantages, but the requirement of automated process control as an integral component of next-generation tools. The solution has been developed for application to both existing (retrofit) and next-generation tools and has been shown to significantly impact cost of ownership of these tools. Specifically, this control solution has been shown to reduce process variability, reduce scrap, increase
FIGURE 11.9 The fully automated CMP R2R process control solution; inputs — a) and b) — are modified to keep outputs — c) and d) — on target.
life of consumables, and increase yield and throughput. It is further anticipated that this capability will also be proven to reduce equipment downtime and reduce test wafer requirements. The impact of the automation aspect has not yet been quantified; however, with run-to-run operator intervention drastically reduced, it is likely that the yield and throughput losses due to operator-induced idle and downtime will be reduced.

The controller has been developed as a fully object-oriented solution and is compatible with existing standards and trends in semiconductor manufacturing. It address all of the design requirements for an integrated R2R control solution identified in Chapter 8; most of these requirements are met due to utilization of the GCC-enabling technology implemented in an object-oriented environment. Further, due in part to this object-oriented structure, the controller has been developed so that it can be easily enhanced to support future control environments such as multistep process control, interprocess feedforward control, and combined in situ and run-to-run control.

ACKNOWLEDGMENTS

Portions reprinted with permission from proceedings of Advanced Semiconductor Manufacturing Conference: SEMICON Taiwan ’98 (see Reference 1). The development of an R2R control capability for the Strasbaugh tools presented in this chapter required a significant collaborative effort between the tool supplier (Strasbaugh) and the control solution integrator (MiTeX Solutions). The authors would like to acknowledge John Curry and Tim Weaver from Strasbaugh, and Victor Solakhian and Richard Gwizdak from MiTeX Solutions, for their efforts in making possible the solutions presented in this chapter.

REFERENCES

4. SEMATECH AEC Workshop VIII, Santa Fe, NM (October 1996).
12 Design and Optimization of an Optimizing Adaptive Quality Controller, Generic Cell Controller-Enabled Solution

Enrique Del Castillo, Jinn-Yi Yeh, James Moyne, and Victor Solakhian

12.1 INTRODUCTION

In Chapter 11 we presented the design and implementation of two R2R control solutions. In this chapter we present a third R2R control solutions example, focusing on the use of the control algorithm and the integration of the algorithm with the control enabling mechanism. The control algorithm utilized is the optimizing adaptive quality controller (OAQC), as described in Chapter 4, and the control-enabling technology is the Generic Cell Controller (GCC) as described in Chapter 9.

The OAQC acts both as a controller and an optimizer, maintaining optimum operating conditions for multiple input, multiple output processes. The controller will suggest a recipe of operating conditions to be implemented in the next run of a particular process. The OAQC algorithm has been implemented in two software platforms: (1) NextStep-OpenStep (Mach), and (2) Windows NT.

The objectives of this chapter are (1) to illustrate the NextStep implementation of the OAQC with a simple case study, and (2) to present the integration of the OAQC controller with the GCC. The chapter is organized as follows: Section 12.2 presents a case study using the NextStep software implementation of the OAQC. Then, Section 12.3 shows how the integration between the OAQC and the GCC was achieved.

12.2 USE OF THE OAQC: A CASE STUDY

The OAQC (NextStep/OpenStep) system can run in a stand-alone mode or integrated with the GCC system. This section provides a case study that illustrates the stand-alone form of operation. A $4 \times 2$ chemical mechanical planarization (CMP) process will be used for this purpose. This illustration is based on equipment models developed in Chapter 6. Controllable factors (scaled in the $(-1, 1)$ coding convention) are platen speed ($\upsilon_1$), back pressure ($\upsilon_2$) polishing downforce ($\upsilon_3$), and the profile of the conditioning system ($\upsilon_4$). Each factor is constrained to lie on the $(-1, 1)$
range. The two responses of interest are removal rate \( (y_1) \) and within-wafer nonuniformity \( (y_2) \). The fitted equipment models are:

\[
y_1 = 3233.25 + 1833.08 \, u_1 - 96.83 \, u_2 + 3005.34 \, u_3 - 614.51 \, u_4 - 5453.06 \, u_1^2 \\
+ 54.47 \, u_2^2 + 1597 \, u_3^2 + 83.43 \, u_4^2 - 705.68 \, u_1 \, u_2 - 1532.67 \, u_1 \, u_3 - 3959.76 \, u_1 \, u_4 \\
+ 563 \, u_2 \, u_3 + 934.39 \, u_2 \, u_4 + 0.3 \, u_3 \, u_4 - 17 \, t + \varepsilon_{1,t}
\]

and

\[
y_2 = 326.5 - 5483.85 \, u_1 - 169.4 \, u_2 + 338.87 \, u_3 + 176.62 \, u_4 - 1009.13 \, u_1^2 - 6.72 \, u_2^2 \\
+ 89.98 \, u_3^2 + 0.76 \, u_4^2 - 2890.96 \, u_1 \, u_2 - 12060.42 \, u_1 \, u_3 - 628.2 \, u_1 \, u_4 \\
- 270.1 \, u_2 \, u_3 + 137.59 \, u_2 \, u_4 + 395.15 \, u_3 \, u_4 + 1.5 \, t + \varepsilon_{2,t}
\]

where \( \varepsilon_{1,t} \sim N(0, 25.8^2) \) and \( \varepsilon_{2,t} \sim N(0, 2.3^2) \). For illustration, Eqs. (1) and (2) were simulated and correspond to the system to be controlled. We will proceed as if nothing is known about the system parameters with exception of prior intercept estimates for each response. This will provide a realistic setting in which equipment models are not available but only some basic information about the responses is.

### 12.2.1 ENTERING THE OAQC INITIAL SETTINGS

After starting the system, the OAQC graphical user interface (GUI), as shown in Figure 12.1, appears on the screen.

The user must enter some basic information through the GUI prior to running the OAQC, such as the number of factors (3), the number of responses (3), positive ranges for all factors (in this case \(-1 \leq \mu_i \leq 1\)), non-zero intercept estimates for each response (we entered 3233 for \( y_1 \) and 326 for \( y_2 \)), and non-zero minimum resolutions for each factor (we entered 1 for \( \mu_1 \), and 0.1 for \( \mu_2, \mu_3, \) and \( \mu_4 \)). The minimum resolutions refer to the minimum change to a controllable factor (i.e., the “discretization” of the inputs). In addition to entering this basic information, the following steps must be performed:
1. Change order of the models. Assuming the process engineer suspects there is curvature for all responses, full quadratic models are selected by clicking on all three types of effects (linear, two-factor interaction, and pure quadratic), as shown in Figure 12.2.

2. Change response constraints. The constraints were set at $y_1 > 1700$ and $y_2 < 250$.

3. Change response targets and weights. The target values were set at 1700 for $y_1$ and 150 for $y_2$ (by default, all response weights are set equal to one).

4. Change gain estimates. Assume approximate initial models given and quadratic models selected. The initial models were specified as follows:

$$y_1 = 3233 + 1800 u_1 - 90 u_2 + 3000 u_3 - 600 u_4 - 5000 u_1^2 + 50 u_2^2$$

$$+ 1500 u_3^2 + 80 u_4^2 - 700 u_1 u_2 - 1500 u_1 u_3 - 4000 u_1 u_4$$

$$+ 500 u_2 u_3 + 900 u_2 u_4 + 0.3 u_3 u_4 - 10 t$$

and

$$y_2 = 326 - 5000 u_1 - 170 u_2 + 300 u_3 + 170 u_4 - 1000 u_1^2 - 6 u_2^2$$

$$+ 90 u_3^2 + u_4^2 - 2800 u_1 u_2 - 12000 u_1 u_3 - 600 u_1 u_4$$

$$- 270 u_2 u_3 + 140 u_2 u_4 + 400 u_3 u_4 + t$$

Thus, the OAQC is given a priori information about the system to be controlled. Figure 12.3 shows the change gain estimates window where the user can enter those gain estimates in Eqs. (3) and (4).
5. Change tuning parameters. A three-level factorial experiment was selected for the first ten runs. Figure 12.4 shows the Change Tuning Parameters window.

6. Multivariate exponential weighted moving average (EWMA) SPC chart. This option implements the Lowry et al. chart. It asks for initial estimates for the variances of the responses (the default is all variances equal to 1.0, and all covariance equal to 0.01), SPC EWMA weight (0.3 is the default), and the constant upper limit (10 is the default). All defaults were
used in this option. The Multivariate EWMA SPC chart windows is shown in Figure 12.5. By default, the chart is used just for monitoring out-of-control events. If used as a deadband, the OAQC will be invoked (and a new recipe will be computed) only when there is evidence of an out-of-control state; otherwise, the previous recipe is kept.

12.2.2 **Find Next Recipe**

The user should click on the “Find Next Recipe” button whenever it is desired to invoke the OAQC’s optimizer to find the suggested recipe to use for the next run. After entering initial settings, bounds, constraints, intercepts, and perhaps parameter estimates, a normal iteration will then consist of the following:

1. Clicking “Find Next Recipe”
2. Reading the solution
3. Entering the recipe in the equipment
4. Obtaining the metrology
5. Entering the metrology information in the “Update Observed Responses (metrology)” option

Then, we repeat the same steps run by run until the control session terminates. The recipe and metrology information is automatically displayed in “history plots” that show the levels of the controllable factors and of the responses for each run. Figures 12.6 and 12.7 show the inputs and outputs, respectively, of a 50-wafer (runs) session for this CMP process. In this simulation, target values were set at 1700 for $y_1$ and 150 for $y_2$, and constraints were set at $y_1 > 1700$, $y_2 < 250$, and $-1 \leq \mu_i \leq 1$. The open-loop responses (what we would get if the recipe were maintained constant
at the zero level) are also shown in Figure 12.8. The constraints on the outputs together with the strict target values are useful in this example as removal rate is usually a “larger the better” response and nonuniformity a “smaller the better” response. The EWMA chart (Figure 12.8) indicates that there is no “out of control” signal. Thus, this $4 \times 2$ CMP process is in control.
12.3 INTEGRATION OF THE OAQC WITH THE GENERIC CELL CONTROLLER (GCC)

In Chapter 9 we showed that the GCC-R2R controller (GCC-R2R) is a discrete control mechanism that utilizes a relational database to store and execute sequential control information (a sequence of command–action pairs used to control a manufacturing process). The GCC implementation described here further uses object-oriented technology and distributed object-based communications to implement a suite of client/server applications designed to work in a distributed computing environment provided by the NextStep/OpenStep operating system. The solution is portable to many hardware and software platforms (Intel, Motorola, SUN Sparc, Power PC processors; Mach, Windows NT, Solaris operating systems; and Sybase, Oracle, OpenBase, and other industry standard database servers).  

Specifically, the GCC utilizes a relational database along with a GccEvent object to direct execution order and data flow, respectively, as illustrated in Figure 12.9. The GccKernel queries the database to determine a module execution order required to service a control command. The GccEvent objects travel between the modules and serve as containers for all the GCC data needed to relate the current state of the run-to-run control system. In this way the modules may execute the desired actions. The modules then perform their intended actions and the results of those actions are stored back to the GccEvent. Thus, the instance of GccEvent class is passed from one module to another as directed by the GccKernel as an argument of a command (message to an object). Each module contributes some data (such as the recipe, measured or simulated output, etc.) to the instance.

![Figure 12.9](image.png)  
**FIGURE 12.9** The GCC-R2R data flow diagram.
This version of the GCC-R2R contains a set of modules that, collectively, are capable of providing R2R control of a variety of manufacturing processes. This set includes three control algorithm modules:

1. The linear approximation multivariate “Gradual Mode” (GM) control algorithm (MIT)
2. The time-based extended GM (GMt) control algorithm (MIT)
3. The optimizing adaptive quality controller (OAQC) (PSU)

Since the GCC-R2R provides a modular and configurable control environment, it is suited for managing the required communications between a control algorithm module, a processing equipment interface module, a metrology equipment interface module, and other modules (such as a history-plot module) that are part of the control solution. Figure 12.10 shows the general schema of the GCC. The heart of the GCC is the GccKernel, which coordinates all controller activity (by coordinating command and data flow between the various R2R control modules running on any computer in a network). The GCC-R2R accepts commands either from a parent interprocess controller or from the process engineer/operator (via the graphical user interface) and performs corresponding actions fetched from the database. Those actions involve the coordination of any number of the aforementioned modules to achieve a desired result.

Of the many features of GCC-enabled R2R control solutions described in Chapters 9 and 10, the example provided in this chapter illustrates the feature of capability for integration of third-party software, namely the OAQC algorithm solution, into the GCC module. The OAQC controller can be used either in stand-alone mode or as part of the GCC. The stand-alone mode usage was described in Section 12.3.
The GCC mode of operation is somewhat different. Once a user has entered and saved the initial settings (see 12.2.1) for the process being controlled, it is not necessary to use the OAQC user interface. As the OAQC is used as part of the GCC system, the GccKernel takes control of all its activities. It sends to the OAQC module commands in order to get a suggested recipe for the next run or to simulate the process and metrology outputs. This is implemented via a distributed objects paradigm (see Figures 12.9 and 12.10). As part of the execution of sequential control, information is first fetched from the database for a high-level command. The GccKernel then sends messages to the OAQC system object and gets the results encapsulated in the GccEvent distributed object, which is used by the GCC as an information medium.

The following is a real example of commands sent to the GCC modules by the GccKernel to implement high-level “planarize”* command for one control “loop.” Each command explanation is commented to its right.

```plaintext
[OAQC getRecipe:] /* Get the process recipe for this run */
[GccTool recipeDownload:] /* Download that recipe to the equipment (either automatically via a SECS interface or manually with the aid of the equipment operator) */
[GccTool start:] /* Start the process */
[GccMetrology getMetrology:] /* Enter in metrology data (either automatically via a SECS* interface or manually through a Graphical User Interface) */
[OAQC control:] /* Determine a process recipe for the next run */
[GccHistory plot:] /* View the process history either in tabular form or graphically */
[GccKernel runAgainIfNecessary:] /* Repeat the loop according to the “repeat number” set by the operator */
```

There are numerous advantages to the GCC-integrated form of OAQC operation. This form of operation

1. Can be integrated into a hierarchical control system within a semiconductor manufacturing factory
2. Allows automatic uploading of metrology data and automatic downloading of recipes to the equipment
3. Provides a database management system for analyzing historical input–output data
4. Enables the OAQC to run in parallel with other controllers
5. Provides different authorization levels for users

* “Planarize” is a typical command process action for a chemical mechanical planarization (CMP) tool; see Introduction.

SEMI Equipment Communication Standard, a communication protocol commonly utilized in the semiconductor manufacturing industry. See Reference 10 for a complete description, and Chapter 9 for examples.
12.4 SUMMARY

A design for integration of an OAQC algorithm implementation into a GCC-enabled R2R control solution has been described in this chapter. As a complement to the examples in Chapter 11, which focused on the integration of R2R control into the factory and the operation and evaluation of the controller in that environment, this chapter focuses on the algorithm implementation and the integration of the algorithm into the R2R control solution. The solution developed is illustrated to be flexible and allows for operation of the algorithm in stand-alone and integrated model. The integration meets the requirements of integrated control defined in Chapter 8, thus the solution, if deployed, should provide for effective R2R control in a factory environment, similar to the solutions described in Chapter 11.

REFERENCES

Part 4

Customization Methodology

Thus far in this book we have provided motivation for R2R control in the industry (see Introduction), described elements of the foundation of process control (Part 1), presented information on R2R control algorithms (Part 2), and described R2R control deployment and integration-enabling mechanisms (Part 3). While each of these components is important to the R2R control solution, successful utilization of R2R control in practice additionally requires adherence to a procedure for process control deployment that includes process identification, control solution customization, integration, test, and validation. In Part 4 of the book we focus on the methodology for R2R control solution deployment and customization.

We present this methodology in Chapter 13 through a case study based on an example of an R2R implementation on a production tool. The device type, input/output detail, measurement units, and exact numerical results have been changed to protect the confidentiality of the customer. The implementation now refers to a “furnace for producing a ceramic used in the manufacture of superconducting wire.”

The effort was motivated by the customer’s desire to improve the performance of their ceramic furnaces without requiring substantial, costly hardware upgrades. A team made up of expert personnel from the customer and the control vendor was organized to carry out the initial implementation and show process improvement under R2R control.

The main feature of the chapter is the establishment of a well-defined methodology for implementing R2R process control. Detailed discussions were first conducted with the customer’s personnel to identify the control problem, and a design of experiment (DOE) on the furnace was executed so that a model of the process could be derived for control purposes. From the results of the experiment, regression models of the process were developed. Utilizing these as control models within the R2R controller, the controller was deployed on the most stable furnace in the plant;
any improvement in furnace performance could then be seen as a bottom-line advantage for the whole furnace line.

Results were gathered for 60 runs (furnace cycles) and compared to previous runs of the same furnace without the controller present. Notable improvement ratios were observed.

The operation of a run-to-run system usually takes place in two phases or stages. In the first stage, a new piece of equipment or tool is “qualified,” which means that experiments are conducted in order to characterize and possibly also optimize the performance of the tool. This qualification stage is based on design of experiment (DOE) techniques. DOE is widely known and practiced in modern process engineering, and enjoys many popular references such as References 1 and 2; it is therefore not further discussed. Any optimization part of a tool qualification uses a set of experimental optimization techniques called response surface methods. Once the process has been qualified, equipment models and process targets become available, and the goal of an R2R controller is to keep the process responses at the optimized levels in the presence of drift and disturbances that occur from run to run. This constitutes the second stage of the operation. Although some optimizing controllers can provide optimization and control capabilities in the same tool set (e.g., the OAQC algorithm described in Chapters 4 and 5), the use of classical response surface methods (RSM) is much more widespread. RSM methods are also the topic of Chapter 14.

REFERENCES

13 Case Study: Furnace Capability Improvement Using a Customized R2R Control Solution

Arnon Hurwitz and James Moyne

Control algorithms and enabling technologies for R2R control are described in Parts 2 and 3 of this book, respectively. In this chapter a case study of developing, deploying, and evaluating an R2R control solution for a furnace process is presented. The R2R control solution development and deployment effort was successfully carried out on a production tool in a production environment. The effort utilized a team with members drawn from the tool owner (the “customer”) and the vendor of the control solution. In presenting this case study, the device type, input/output detail, measurement units, and exact numerical results have been changed to protect the confidentiality of the customer; at the same time we have closely paralleled the implementation steps and the improvement ratios of the original outcome. The implementation now refers to a “furnace for producing a ceramic used in the manufacture of superconducting wire.”

The purpose of this chapter is to chronicle the complete R2R control customization and deployment process. Specifically, it addresses gathering user requirements, design and execution of process characterization experiments, development and customization of R2R control models, deployment, evaluation of results, and integration. The reader may wish to use elements of this chapter as a template for developing and deploying his/her own R2R control solution.

13.1 PROBLEM SETUP

The effort was motivated by the customer’s desire to improve the performance of his ceramic furnaces by using R2R control. A team made up of expert personnel from the customer and the control vendor* was organized to carry out the initial implementation and show process improvement under R2R control. The vendor’s personnel had expertise in R2R control solution development, deployment, and integration; the customer’s personnel were expert in the needs and the operation of the process.

* MiTeX Solutions, Canton, Michigan.
Utilizing a methodology (presented below) defined by the vendor and agreed to by the customer, the R2R control system deployment effort was initiated; detailed discussions were conducted with the customer’s personnel to identify the control problem, and a design of experiment (DOE) on the ceramic furnace was executed so that a model of the process could be derived for control purposes.

From the results of the experiment, as well as additional discussions with the customer, regression models of the process were developed. From these models the 4 × 4 multivariate R2R control model shown in Table 13.1 was derived. In this model, the process outputs to be controlled were $Y =$ ductility measurement, $Z_i =$ superconductivity measurement $i; (i = 1, 2, 3)$, $R_i =$ furnace temperature ramp profile $i; (i = 1, 2, 3)$, $E =$ amount of trace element additive, and $C_j$ a constant; $j = 1, 2, 3, 4$.

Utilizing this control model, the controller* was set up by entering the model coefficients and other parameters detailed below. The controller was then deployed on the most stable furnace in the plant; any improvement in furnace performance could then be seen as a bottom-line advantage to using R2R control.

The R2R controller was deployed and results gathered for 60 runs (furnace cycles) and compared to previous runs of the same furnace without the controller present. The results are summarized as follows:

1. The ratio of noncontrolled standard deviation for output $Y$ to controlled $Y$ is 2.29.
2. The ratio of noncontrolled standard deviation for output $Z_1$ to controlled $Z_1$ is 1.92. Similar ratios apply for $Z_2$ and $Z_3$.

In summary, R2R control cut the standard deviation of both $Y$ and $Z$ performance parameters in half.

### 13.2 BACKGROUND: DESCRIPTION OF PRODUCTION TOOL

In the case study application, the customer was interested in improving the capability of his furnaces used to produce ceramic material for superconducting wire. An informal analysis of the process indicated that the deployment of R2R control might provide significant process improvement. The furnace type in question has the construction shown in Figure 13.1.

---

* The MiTeX Generic Cell Controller run to-run (GCC-R2R) controller.
The furnace is divided into three zones with a separate crucible of ceramic material placed in each zone. Each zone is heated directly with its own set of induction coils, and indirectly by the adjacent coil(s). Each coil can be ramped up to full power with an independently controlled ramp gradient \( R_i \) for Zone \( i \). The ceramic material for a load was mixed in one batch with a trace element additive \( E \) and then evenly divided into three crucible batches for simultaneous insertion into the three furnace zones at the start of each cycle, or “run.” The percentage of the trace element for any one furnace run could be varied between strict specification limits from run to run, but could not be varied separately for the material in the three separate crucibles for any single run.

On completion of a furnace cycle the ceramic material was removed, drawn into wire, and tested for a physical property \( Y \) called “ductility.” As the \( Y \) test was expensive and destructive, only one test was made from a piece of wire taken from the material of the first furnace zone (Zone 1). A nondestructive test for “conductivity” \( Z_i \) was made on a sample of wire from each separate zone.

There were thus four controlled inputs \{\( E, R_1, R_2, R_3 \)\} and four measured outputs \{\( Y, Z_1, Z_2, Z_3 \)\}.

### 13.3 METHODOLOGY FOR CONTROL DEPLOYMENT

In order to ensure that a consistent approach was followed when deploying a control solution, the control solution vendor established a methodology to be followed.\(^1\) This allows for checking and corrective action as the implementation project proceeds. The steps of the methodology are laid out in Table 13.2.

### 13.4 APPLICATION OF METHODOLOGY

The above steps of the methodology were implemented as specified in Table 13.2. Details are provided in the following subsections.

#### 13.4.1 QUANTIFY PROBLEM WITH PROCESS ENGINEERS

In the first step of the application of this procedure, extensive discussions were conducted with customer personnel to identify process quality metrics and process
quality improvement goals. From these discussions it was agreed that the ideal run-to-run controller would provide multivariate (concurrent) control of ductility \( Y \) as well as three-zone superconductivity \( Z_1, Z_2, Z_3 \).

The process engineers further related that \( Y \) was a fairly well-behaved signal and a strong (qualitatively stated) function of the additive \( E \) level, although \( Y \) was also notorious for strong downward drift. In fact, the furnace had to be taken apart and rebuilt after every 20 to 30 cycles to “reset the drift factor back to zero.”

Conductivity \( Z \) was less well behaved* than \( Y \), and its correlation to input variables (induction ramp \( R_i \), and \( E \) level) was not as well understood.

As stated in Section 13.1 above, there were four controllable inputs \( \{E, R_1, R_2, R_3\} \) and four measured outputs \( \{Y, Z_1, Z_2, Z_3\} \). The input factors were constrained by safety and other considerations to lie within certain upper and lower bounds as shown in Table 13.3.

---

* Higher relative variability, and metrology noise.
13.4.2 Design of Experiment(s)

Using the information provided by the engineers, an experiment was designed to quantify the process control problem. This entails designing and running a formal experiment to optimally collect data that can be used to construct a linear model of the process. This model is subsequently used in the R2R controller to guide the process.

Given the resources and constraints of the situation at hand, it was decided to design a basic experiment with eight “experimental” runs plus four runs set at the process standard recipe setpoints — a total of twelve runs. The experimental matrix was based on a half-fraction of a two-level, four-input-factor design; that is, a $2^{4-1}$ fractional factorial design.* The design matrix is given in Table 13.4.

The “Pattern” column indicates the high (+) and low (–) settings of the inputs (+) according to the bounds given in Table 13.2.

13.4.3 Execution of the Experimental Design

The customer’s project team engineers and line operators executed the experiment. Note that the standard process recipe settings (i.e., “0000”) were run at roughly even intervals throughout the experiment, and started and ended the experiment. The results from these standard-setting runs were used to check that no abnormal tool behavior occurred during the experiment. The 12 runs were completed over a short time period, but if they had been run over a long period when tool drift could manifest itself, the standard run results could be used to de-trend the final experimental outcomes before the subsequent model-building. The results of measuring $Y$, and $Z_1$, $Z_2$, and $Z_3$, are given in Table 13.5.

---

* For details about factorial designs, see Reference 2.

---

**TABLE 13.4**

Experimental Design Matrix

<table>
<thead>
<tr>
<th>Pattern</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0000</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>18.1</td>
</tr>
<tr>
<td>2 –+++</td>
<td>159</td>
<td>161</td>
<td>161</td>
<td>16.3</td>
</tr>
<tr>
<td>3 ++–</td>
<td>161</td>
<td>159</td>
<td>161</td>
<td>16.3</td>
</tr>
<tr>
<td>4 0000</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>18.1</td>
</tr>
<tr>
<td>5 ––––</td>
<td>159</td>
<td>159</td>
<td>159</td>
<td>16.3</td>
</tr>
<tr>
<td>6 ++++</td>
<td>161</td>
<td>161</td>
<td>159</td>
<td>16.3</td>
</tr>
<tr>
<td>7 –++–</td>
<td>159</td>
<td>161</td>
<td>159</td>
<td>20.0</td>
</tr>
<tr>
<td>8 0000</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>18.1</td>
</tr>
<tr>
<td>9 ++++</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>20.0</td>
</tr>
<tr>
<td>10 –++–</td>
<td>159</td>
<td>159</td>
<td>161</td>
<td>20.0</td>
</tr>
<tr>
<td>11 ++–</td>
<td>161</td>
<td>159</td>
<td>159</td>
<td>20.0</td>
</tr>
<tr>
<td>12 0000</td>
<td>160</td>
<td>160</td>
<td>160</td>
<td>18.1</td>
</tr>
</tbody>
</table>

* For details about factorial designs, see Reference 2.
Note also that the (nonstandard) experimental runs* are arranged in random order through the design matrix. If the experiment were to be replicated, the standard runs would stay in their unaltered run order; however, the other experimental runs must have their order randomly shuffled to prevent introduction of any experimental bias or unwanted systematic structure.

### 13.4.4 Data Analysis

The results of Table 13.5 were entered into a statistical analysis program,** and linear regression models were requested for \( Y \) and the \( Z_i \)'s. The next set of three tables shows the output for the analysis of the \( Y \) data.

Table 13.6 shows the coefficients (parameters) of the model estimated from the experimental data for \( Y \). Writing down the model using these coefficients, we obtain

\[
Y = 10.65 + 1.4 * E - 0.01 * R1 + 0.03 * R2 - 0.01 * R3 \tag{1}
\]

---

* Rows: 2, 3, 5, 6, 7, 9, 10, 11.

** JMP by the SAS Institute of Cary, North Carolina.
From Table 13.6, only the inputs $R_2$ and $E$ have coefficients with $t$-statistics that are significantly different from zero; the coefficients for $R_1$ and $R_3$ are marginal (about 77% significance level for the $t$ ratios). We can, therefore, try a model for $Y$ that has only $R_2$ and $E$ on the right-hand side (RHS). The new parameter estimates are listed in Table 13.7.

Table 13.7 shows that the significance level for $R_2$ collapses; only $E$ remains as significant. One could have a control equation for $Y$, which is driven by $E$ alone, and it might work quite well. However, the engineers were convinced that the form given in Eq. (1) was preferable for control, and that the signs and sizes of the coefficients for all three $R$ inputs were correct. For this reason the controller was programmed with Eq. (1) for $Y$. As it turned out, this gave good $Y$ control in the trials.

The goodness of fit of Eq. (1) for $Y$ is also determined as a whole by the R-squared adjusted statistic, which was greater than 99%. This means that the model given in (1) “explains” more than 99% of the variation observed in the $Y$ of the experimental data.

Table 13.8 addresses possible lack of model fit to the data. The model we have fitted for $Y$ is strictly a linear one in the input factors; if the true nature of $Y$ contained quadratic or other nonlinear terms we would expect the fit to be not as good, and to show up formally as a significant lack-of-fit $F$-ratio. In Table 13.7 the probability of such a lack of fit is seen to be small because the lack-of-fit $F$-ratio statistic has a high probability (61.01%) of occurring by chance alone.

Next, we need to derive the equations for the conductivity $Z_1$, $Z_2$, and $Z_3$. Only $R_1$ was found to be significant for $Z_1$. Table 13.9 shows the estimation results for $Z_1$ regressed against $R_1$.

The chosen model for $Z_1$ is thus

$$Z_1 = 3.9 + 0.04 \times R_1$$  \hspace{1cm} (2)
R-squared adjusted for model (2) was 83%; no lack of fit was detected. Similarly, the models for $Z_2$ and $Z_3$ were

$$Z_2 = 6.8 + 0.03 \times R_2$$  \hspace{1cm} (3)$$

$$Z_3 = 0.5 + 0.06 \times R_3$$  \hspace{1cm} (4)$$

R-squared adjusted for model (3) was 34%, and for model (4), 55%. These indicate a weak explanatory power; however, the individual $t$-statistics showed significance for both ramp ($R_i$) parameters and, in addition, the signs of the parameters were in accordance with engineering theory and consistent with (2). No lack of fit was detected* for either (3) or (4). The final model system chosen for the controller can be seen in Table 13.10. Note that the values of the constant terms are not given explicit values here: they are initialized at the start of a control run to bring the tool outputs to target, and thereafter they are estimated on a run-to-run basis by the control software itself.

### 13.4.5 CONTROLLER PARAMETER SPECIFICATION

A minimal set of control parameters needs to be specified before the controller can be deployed for action. A list of these parameters follows:

1. The weights of the input parameters, the relative weights of which reflect preference for adjustment (higher weight implies greater preference).
2. The weights of the output parameters, the relative weights of which reflect priority in control.

* Although in the case of models (3) and (4) this could mean noise-obscured structure.

---

**TABLE 13.9**

Parameter Estimates for $Z_1$

<table>
<thead>
<tr>
<th>Term</th>
<th>Estimate</th>
<th>Std. Error</th>
<th>$t$ Ratio</th>
<th>Prob. &gt; $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>3.9323351</td>
<td>0.884476</td>
<td>4.45</td>
<td>0.0012</td>
</tr>
<tr>
<td>$R_1$</td>
<td>0.0404374</td>
<td>0.005528</td>
<td>7.32</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>

**TABLE 13.10**

Control Model System

$$Y = C_1 + 1.4 \times E - 0.01 \times R_1 + 0.03 \times R_2 - 0.01 \times R_3$$

$$Z_1 = C_2 + 0.04 \times R_1$$

$$Z_2 = C_3 + 0.03 \times R_2$$

$$Z_3 = C_4 + 0.06 \times R_3$$
3. The smoothing parameters (EWMA weights, or alphas — see Part 2 of this book), one for each output parameter, the weights of which, lying between 0 and 1, reflect more or less smoothing of input metrology noise; the closer to zero, the more smoothing.

4. The output targets for the product type.

5. The upper and lower control bounds for the input parameters. These may be the same as the bounds for the designed experiment, and are usually contained within the bounds of the designed experiment.

6. The resolution, or “granularity,” of each input parameter; this value indicates the smallest control adjustment possible for that input.

7. The control model equations.

Utilizing these control parameter specifications, the control system model of Table 13.10 is augmented by the specifications of Tables 13.11 and 13.12 to complete the specification for this application.

Table 13.11 indicates that all output parameters are considered of equal importance. Also, more metrology noise is expected for the $Z$ outputs, and so an alpha closer to zero is specified. The alpha for $Y$ is high, indicating high confidence in the precision of the measured value for $Y$ and in the controller action.

### Table 13.11

- **Output Parameter Specifications**

<table>
<thead>
<tr>
<th>Output Parameter</th>
<th>Weight</th>
<th>EWMA (alpha)</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>1</td>
<td>0.7</td>
<td>37.3</td>
</tr>
<tr>
<td>$Z1$</td>
<td>1</td>
<td>0.4</td>
<td>10.5</td>
</tr>
<tr>
<td>$Z2$</td>
<td>1</td>
<td>0.4</td>
<td>10.5</td>
</tr>
<tr>
<td>$Z3$</td>
<td>1</td>
<td>0.4</td>
<td>10.5</td>
</tr>
</tbody>
</table>

### Table 13.12

- **Input Parameter Specifications**

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Upper Bound</th>
<th>Lower Bound</th>
<th>Resolution</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>16.3</td>
<td>20.0</td>
<td>0.1</td>
<td>2.0</td>
</tr>
<tr>
<td>$R1$</td>
<td>159</td>
<td>161</td>
<td>0.01</td>
<td>1.0</td>
</tr>
<tr>
<td>$R2$</td>
<td>159</td>
<td>161</td>
<td>0.01</td>
<td>1.0</td>
</tr>
<tr>
<td>$R3$</td>
<td>159</td>
<td>161</td>
<td>0.01</td>
<td>1.0</td>
</tr>
</tbody>
</table>

13.5 Customization and Delivery of Controller

The parameters of Tables 13.10, 13.11, and 13.12 were entered into the controller software. In addition, certain custom additions such as password protection were
enabled for the specific needs of the customer. The controller was then delivered to the customer’s site and set up for the test runs.

13.6 ON-SITE TESTING, TRAINING, TECHNICAL SUPPORT DURING TESTING

The controller was deployed at the customer site on the “most stable” furnace. Because R2R control serves to reduce process shift and drift, performance improvement of a stable process would serve to identify a lower bound on the impact of R2R on process capability for this type of process. The testing proceeded as follows.

1. Vendor’s personnel visited the customer test site and installed an R2R controller on the target furnace. The controller was deployed on a stand-alone computer with communication between the metrology tool, controller, and process tool achieved through human interaction.
2. Vendor’s personnel remained at the site for a period of one week. During that time the vendor completed training of three shifts of customer’s personnel (process engineers and technicians) on the use of the R2R control system, and delivery of a customized user manual for the R2R control system that covered controller operation as well as model setup for change of process. The training process entailed two hours of hands-on training followed by four hours of monitoring by vendor personnel.
3. The stand-alone test controller was left at the vendor site for a period of six weeks. During that time the vendor was available for 24-hour phone technical support. A day’s visit was also made after three weeks to check the progress during the test period.
4. Customer’s personnel delivered to the vendor the process recipe logs and metrology data collected during the test period. The data for 60 control runs are included in the Appendix of this chapter. The customer also delivered a sample of metrology data for a fixed recipe (i.e., noncontrolled) run of 60 furnace cycles that occurred prior to the deployment of the controller.

13.7 RESULTS ANALYSIS

The controller eliminated the typical downward drift of the $Y$ parameter and enabled the process to run for 60 cycles without noticeable degradation in performance. The standard deviation of the $Y$ measurement was, in comparison to the noncontrolled $Y$, cut by more than half. The standard deviation for the Z outputs was also cut almost by half. Table 13.13 summarizes the improvements for $Y$ and $Z_1$. Results for $Z_2$ and $Z_3$ were similar. Graphs of the output metrology for $Y$ clearly show the improvement (Figure 13.2). For $Z_1$ (Figure 13.3), there is no trend to begin with, but the controller appears to reduce the number of process interventions, and thus leads to a more stable, less noisy process. Similar results can be noted for all three $Z$ outputs (Figure 13.4).
The actual moves made by the R2R controller, in the form of instructions for tweaking the inputs $E$ and $R_i$ ($i = 1, 2, 3$), are shown in Figures 13.5 and 13.6. Note that while $R_1$ tended to begin and end the 70-run campaign at about the same average level, $R_2$ and $R_3$ ended somewhat higher than where they began.

These good results were communicated to the customer in the form of a presentation to all who had been involved in the project, as well as interested managers from different production branches.

TABLE 13.13
Performance Improvement with R2R Control: $Y$ and $Z_1$

<table>
<thead>
<tr>
<th>Metric</th>
<th>$Y$</th>
<th>$Z_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation (controlled runs)</td>
<td>0.32</td>
<td>0.14</td>
</tr>
<tr>
<td>Standard deviation (non-controlled)</td>
<td>0.73</td>
<td>0.27</td>
</tr>
<tr>
<td>Standard dev. ratio (noncontr./contr.)</td>
<td>2.29</td>
<td>1.92</td>
</tr>
</tbody>
</table>

FIGURE 13.2 Controlled vs. non-controlled $Y$ performance.

FIGURE 13.3 Controlled vs. non-controlled $Z_1$ performance.
At the conclusion of the test, a determination should be made to deploy the controller in production. A deployment plan and integration path must be decided.

13.8 DEPLOYMENT, INTEGRATION

At the conclusion of the test, a determination should be made to deploy the controller in production. A deployment plan and integration path must be decided. *R2R controller*
integration is covered in depth in Chapter 10, and especially Chapter 11. In terms of the case study, the R2R controller solution had been installed in other tools and the integration path was clear enough to the vendor personnel. The integration can be made completely “closed loop” if there exists automated metrology for the output parameters. If automated metrology does not exist for all or some of the output parameters, then some human interaction will remain necessary; this is feasible, provided the run-to-run cycle leaves sufficient time for the human interaction to occur.

In the case of the induction furnace, several hours elapsed between runs, and this would make it possible to do metrology and/or data entry by hand in sufficient time before the occurrence of the next run. It is also possible, and has been shown for other tools,* for several runs to elapse before a new recipe needs to be generated. This time lapse also allows for human interaction in the control loop. Ideally, however, full automation is preferred, and this is possible, provided that a proper production robust integration entity is available.**

### 13.9 CONCLUSIONS

The successful development, customization, and deployment of an R2R control solution has been presented in this chapter. The development and deployment effort utilized is an orderly, systematic methodology that is process-independent. Thus, this methodology could be and has been applied to a wide variety of R2R control problem scenarios. Deployment of this specific methodology has yielded numerous successes in CMP, etch, and deposition control. With respect to the case study, performance improvement of the magnitude demonstrated opens the door to tool upgrades costing far less than buying the latest and most expensive tool. This is especially desirable if there are a number of similar tools needing to be replaced or upgraded in any one site. Thus, R2R controller technology gives production managers a new option for cost–benefit decision making.

### REFERENCES


---

* For example, in the case of semiconductor planarization devices.
** In the present case study, the MiTeX Generic Cell Controller (GCC) serves this purpose.
APPENDIX: DATA FROM PRODUCTION RUN WITH
AND WITHOUT R2R CONTROL
INPUTS

---TARGETS---

Run #

E

R1

R2

R3

Y

Z

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

17.5
17.5
17.5
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
17.8
17.8
18.0
18.0
18.0
18.0
18.0
18.0
17.8
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.0
18.3
18.0
18.0
18.0
18.0

160.00
160.00
160.00
160.00
160.00
160.00
160.00
160.00
160.00
160.13
160.13
160.13
160.13
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.00
160.00
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.13
160.00

160.00
160.00
160.00
160.13
160.13
160.13
160.13
160.40
160.13
160.13
160.13
160.13
160.13
160.13
160.27
160.13
160.13
160.13
160.00
160.00
160.13
160.13
160.13
160.13
160.13
160.13
160.27
160.27
160.27
160.27
160.27
160.27
160.27
160.40
160.40
160.40
160.40
160.40
160.40
160.40
160.40
160.40
160.53
160.53
160.40
160.40

160.00
160.00
160.00
160.00
160.00
160.00
160.00
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.73
159.73
159.73
159.73
159.73
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
159.87
160.00
160.00
160.00
160.00
160.00
160.00
160.13
160.13
160.13
160.13
160.13
160.13
160.27

37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3
37.3

10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5
10.5

© 2001 by CRC Press LLC

OUTPUTS

Cntrl Y Cntrl Z1 Cntrl Z2 Cntrl Z3 No-ctrl Y No-ctrl Z1
36.71
36.67
36.53
37.11
37.11
37.47
37.36
37.32
37.43
37.54
37.50
37.43
37.32
37.29
37.58
37.54
37.43
37.54
36.82
37.03
37.65
37.58
37.47
37.65
37.29
36.71
37.00
37.36
37.25
37.25
37.36
37.43
37.43
37.47
37.43
37.36
37.29
37.61
37.61
37.65
36.49
37.65
37.79
37.50
37.50
37.25

10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.01
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.95
10.45
10.45
10.45
10.45
10.45
10.45
10.51
10.45
10.23
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.62
10.45
10.45
10.45
9.79

10.40
10.51
10.40
10.12
10.51
10.73
10.56
10.01
10.45
10.34
10.51
10.45
10.45
10.40
11.06
10.40
10.56
10.45
10.18
10.29
10.34
10.29
10.34
10.67
10.29
10.40
10.34
10.34
10.34
10.45
10.34
10.29
10.29
10.29
10.51
10.56
10.40
10.40
10.45
10.67
10.29
10.34
10.45
10.56
10.40
10.40

10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
10.45
11.11
10.45
10.45
10.45
10.07
10.45
10.45
10.12
10.45
10.45
10.45
10.45
10.40
10.45
10.45
10.40
10.45
10.45
9.79
10.45
10.45
10.45
10.45
10.45
10.07
10.45
10.40
10.45
10.45
10.18
10.45
10.45

37.08
37.82
37.01
36.43
37.72
36.40
37.41
36.29
37.31
36.40
37.51
37.20
37.17
36.63
36.04
36.56
37.06
36.99
35.90
35.91
36.02
35.84
35.95
35.93
35.73
36.21
35.81
35.77
35.74
36.57
35.67
35.49
35.45
35.42
36.67
36.40
35.82
35.79
36.26
35.37
35.17
35.28
36.12
36.12
35.54
35.51

10.39
10.50
10.39
10.12
10.50
11.72
10.56
10.01
10.45
10.34
10.51
10.45
10.45
10.40
11.06
10.40
10.57
10.46
10.19
10.30
10.35
10.30
10.35
10.69
10.30
10.41
10.36
10.36
10.36
10.47
10.36
10.31
10.31
10.31
10.53
10.59
10.42
11.42
10.48
10.70
10.32
10.37
10.48
10.60
10.43
10.43


<table>
<thead>
<tr>
<th>Run #</th>
<th>E</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>Y</th>
<th>Z</th>
<th>Ctrl Y</th>
<th>Cntrl Z1</th>
<th>Cntrl Z2</th>
<th>Cntrl Z3</th>
<th>No-ctrl Y</th>
<th>No-ctrl Z1</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>18.0</td>
<td>160.27</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.21</td>
<td>10.45</td>
<td>10.56</td>
<td>10.45</td>
<td>36.01</td>
<td>10.60</td>
</tr>
<tr>
<td>48</td>
<td>18.0</td>
<td>160.27</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.29</td>
<td>10.73</td>
<td>10.29</td>
<td>10.45</td>
<td>34.93</td>
<td>10.32</td>
</tr>
<tr>
<td>49</td>
<td>18.0</td>
<td>160.13</td>
<td>160.40</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.47</td>
<td>10.45</td>
<td>10.73</td>
<td>10.45</td>
<td>34.90</td>
<td>10.77</td>
</tr>
<tr>
<td>50</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.54</td>
<td>10.45</td>
<td>10.56</td>
<td>10.45</td>
<td>35.91</td>
<td>10.60</td>
</tr>
<tr>
<td>51</td>
<td>18.0</td>
<td>160.13</td>
<td>160.40</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.32</td>
<td>10.45</td>
<td>10.45</td>
<td>10.45</td>
<td>35.84</td>
<td>10.49</td>
</tr>
<tr>
<td>52</td>
<td>18.0</td>
<td>160.13</td>
<td>160.40</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.29</td>
<td>10.34</td>
<td>10.07</td>
<td>10.45</td>
<td>34.75</td>
<td>10.11</td>
</tr>
<tr>
<td>53</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.14</td>
<td>10.45</td>
<td>10.12</td>
<td>10.45</td>
<td>34.71</td>
<td>10.16</td>
</tr>
<tr>
<td>54</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.43</td>
<td>10.51</td>
<td>10.45</td>
<td>10.45</td>
<td>35.73</td>
<td>10.50</td>
</tr>
<tr>
<td>55</td>
<td>17.8</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>36.78</td>
<td>10.45</td>
<td>10.56</td>
<td>10.45</td>
<td>35.73</td>
<td>10.61</td>
</tr>
<tr>
<td>56</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.54</td>
<td>10.45</td>
<td>10.34</td>
<td>10.23</td>
<td>34.79</td>
<td>10.39</td>
</tr>
<tr>
<td>57</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.39</td>
<td>10.45</td>
<td>10.51</td>
<td>10.45</td>
<td>35.90</td>
<td>10.55</td>
</tr>
<tr>
<td>58</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.29</td>
<td>10.45</td>
<td>10.45</td>
<td>10.45</td>
<td>35.59</td>
<td>10.50</td>
</tr>
<tr>
<td>59</td>
<td>18.0</td>
<td>160.13</td>
<td>160.53</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>37.32</td>
<td>10.73</td>
<td>10.51</td>
<td>10.40</td>
<td>35.83</td>
<td>10.56</td>
</tr>
<tr>
<td>60</td>
<td>18.0</td>
<td>160.00</td>
<td>160.40</td>
<td>160.27</td>
<td>37.3</td>
<td>10.5</td>
<td>36.16</td>
<td>10.45</td>
<td>10.51</td>
<td>10.45</td>
<td>35.79</td>
<td>10.56</td>
</tr>
</tbody>
</table>
14 Process Recipe Optimization

Enrique Del Castillo

The operation of a run-to-run system usually takes place in two phases or stages. In the first stage, a new equipment is “qualified,” which means that experiments are conducted in order to find a recipe that optimizes the performance of the equipment. The qualification stage is based on design of experiments techniques and the use of a set of experimental optimization techniques called response surface methods. Once the process has been qualified, equipment models and process targets become available and the goal of an R2R controller is to keep the process responses at the optimized levels in the presence of drift and disturbances that occur from run to run. This constitutes the second stage of the operation. Although some optimizing controllers can provide optimization and control capabilities in the same tool set (e.g., the OAQC algorithm), the use of classical response surface methods is much more widespread and, therefore, these methods are described in this chapter.

14.1 Determining the Optimal Region to Run a Process

The optimal region to run a process is usually determined after a sequence of experiments is conducted and a series of empirical models are obtained. In many engineering and science applications, experiments are conducted and empirical models are developed with the objective of improving the responses of interest. From a mathematical point of view, the objective is to find the operating conditions (or factor levels) $X_1, X_2, \ldots, X_k$ that maximize or minimize the system responses $Y_1, Y_2, \ldots, Y_r$. In experimental optimization, different optimization techniques are applied to the fitted response equations $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_r$. Provided the fitted equations approximate adequately the true (unknown) system responses, the optimal conditions of the model will be “close” to the optimal operating conditions of the true system.

The experimental optimization of response surface models differs from classical optimization techniques in at least three ways:

1. Experimental optimization is an iterative process, that is, experiments conducted in one set of experiments result in fitted models that indicate where to search for improved operating conditions in the next set of experiments. Thus, the coefficients of the fitted equations (or the form of the fitted equations) may change during the optimization process. This is
in contrast to classical optimization where the functions to optimize are supposed to be fixed and given.

2. The response models are fitted from experimental data that usually contain random variability due to uncontrollable or unknown causes. This implies that an experiment, if repeated, will result in a different fitted response surface model that might lead to different optimal operating conditions. Therefore, sampling variability should be considered in experimental optimization. In contrast, in classical optimization techniques the functions are deterministic and given.

3. The fitted responses are local approximations, implying that the optimization process requires the input of the experimenter (a person familiar with the process). This is in contrast with classical optimization, which is always automated in the form of some computer algorithm.

The experimental optimization of a single response is usually conducted in two phases or steps, following the advise of Box and Wilson (see Figure 14.1). The first phase consists of a sequence of line searches in the direction of maximum improvement. Each search in the sequence is continued until there is evidence that the direction chosen does not result in further improvements. The sequence of line searches is performed while there is no evidence of lack of fit for a simple first-order model of the form

\[ \hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k \]  

(1)

The second phase is performed when there is lack of linear fit in Phase I and, instead, a second-order or quadratic polynomial regression model of the form

\[ \hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_k X_k + b_{11} X_1^2 + b_{22} X_2^2 + \cdots + b_{kk} X_k^2 + b_{12} X_1 X_2 + \cdots + b_{k-1,k} X_{k-1} X_k \]

exhibits adequate fit. Not all responses will exhibit quadratic fit, and in such cases Phase I is stopped when the response of interest cannot be improved any further.

The direction of the gradient, \( \mathbf{g} \), is given by the values of the parameter estimates, that is, \( \mathbf{g}' = (b_1, b_2, \ldots, b_k) \). Since the parameter estimates \( b_1, b_2, \ldots, b_k \) depend on the scaling convention for the factors, the steepest ascent (descent) direction is also scale-dependent. That is, two experimenters using different scaling conventions will
follow different paths for process improvement. This does not diminish the general validity of the method, since the region of the search, as given by the signs of the parameter estimates, does not change with scale. An orthogonal scaling convention, however, is recommended. The orthogonal coded factors $x_i$ in terms of the factors in the original units of measurement, $X_i$, are obtained from the relation

$$
x_i = \frac{X_i - (X_{low} + X_{high})/2}{(X_{high} - X_{low})/2} \quad i = 1, 2, \ldots, k
$$

This coding convention is recommended since it provides better parameter estimates, and therefore a more reliable search direction. The coordinates of the factor settings on the direction of steepest ascent separated a distance $\rho$ from the origin are given by

$$
\text{maximize} \quad b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_k x_k \\
\text{subject to:} \quad \sum_{i=1}^{k} x_i^2 \leq \rho^2
$$

This problem can be solved with the aid of an optimization solver (e.g., like the solver option of a spreadsheet). However, in this case this is not really needed, as the solution is a simple equation that yields the coordinates

$$
x_i^* = \rho \frac{b_i}{\sqrt{\sum_{j=1}^{k} b_j^2}} \quad i = 1, 2, \ldots, k.
$$

(2)

An engineer can compute this equation for different increasing values of $\rho$ and get different factor settings all on the steepest ascent direction.

### 14.2 Optimization of Multiple Response Processes

In the multiple response case, finding process operating conditions that simultaneously maximize (or minimize, as desired) all the responses is quite difficult, and often impossible. Almost inevitably, the process engineer must make some trade-offs in order to find process operating conditions that are satisfactory for most (and hopefully all) the responses. Reference 4 explains what to do with multiple responses during “Phase I” of an RSM study. This chapter focuses on Phase II optimization.

#### 14.2.1 The Desirability Function Approach

The desirability function approach is one of the most widely used methods in industry for dealing with the optimization of multiple response processes. It is based on the
idea that the “quality” of a product or process that has multiple quality characteristics, with one of them out of some “desired” limits, is completely unacceptable. The method finds operating conditions \( x \) that provide the “most desirable” response values.

For each response \( Y_i(x) \), a desirability function \( d(Y_i) \) assigns numbers between 0 and 1 to the possible values of \( Y_i \), with \( d(Y_i) = 1 \) representing a completely desirable or ideal response value. The individual desirabilities are then combined using the geometric mean, which gives the overall desirability \( D \):

\[
D = \left( d_1(Y_1) \times d_2(Y_2) \times \cdots \times d_k(Y_k) \right)^{1/k}
\]

where \( k \) denotes the number of responses. Notice that if any response \( i \) is completely undesirable (\( d(Y_i) = 0 \)) then the overall desirability is zero. In practice, fitted response models \( \hat{Y}_i \) are used in the method.

Depending on whether a particular response \( Y_i \) is to be maximized, minimized, or assigned to a target value, different desirability functions \( d(Y_i) \) can be used. A useful class of desirability functions was proposed by Derringer and Suich.7 Let \( L_i, U_i, \) and \( T_i \) be the lower, upper, and target values desired for response \( i \), where \( L_i \leq T_i \leq U_i \). If a response is of the “target is best” kind, then its individual desirability function is

\[
d_i(\hat{Y}_i) = \begin{cases} 
0 & \text{if } \hat{Y}_i(x) < L_i \\
\left( \frac{\hat{Y}_i(x) - L_i}{T_i - L_i} \right)^s & \text{if } L_i \leq \hat{Y}_i(x) \leq T_i \\
\left( \frac{\hat{Y}_i(x) - U_i}{T_i - U_i} \right)^s & \text{if } T_i \leq \hat{Y}_i(x) \leq U_i \\
0 & \text{if } \hat{Y}_i(x) > U_i
\end{cases}
\]

where the exponents \( s \) and \( t \) determine how strictly the target value is desired. For \( s = t = 1 \), the desirability function increases linearly toward \( T_i \), for \( s < 1, t < 1 \), the function is convex, and for \( s > 1, t > 1 \), the function is concave (see example section below for an illustration).

If a response is to be maximized instead, the individual desirability is instead defined as

\[
d_i(\hat{Y}_i) = \begin{cases} 
0 & \text{if } \hat{Y}_i(x) < L_i \\
\left( \frac{\hat{Y}_i(x) - L_i}{T_i - L_i} \right)^s & \text{if } L_i \leq \hat{Y}_i(x) \leq T_i \\
1.0 & \text{if } \hat{Y}_i(x) > T_i
\end{cases}
\]

where in this case \( T_i \) is interpreted as a large enough value for the response. Finally, if we want to minimize a response, we could use
where \( T_i \) represents a small enough value for the response.

The desirability approach consists of the following steps:

1. Conduct experiments and fit response models for all \( k \) responses.
2. Define individual desirability functions for each response.
3. Maximize the overall desirability \( D \) with respect to the controllable factors.

**Example.** Derringer and Suich\(^7\) present the following multiple response experiment arising in the development of a tire tread compound. The controllable factors are \( x_1 \), hydrated silica level; \( x_2 \), silane coupling agent level; and \( x_3 \), sulfur level. The four responses to be optimized and their desired ranges are as follows:

- PICO abrasion index, \( Y_1 \) \( 120 < Y_1 \)
- 200% modulus, \( Y_2 \) \( 1000 < Y_2 \)
- Elongation at break, \( Y_3 \) \( 400 < Y_3 < 600 \)
- Hardness, \( Y_4 \) \( 60 < Y_4 < 75 \)

The first two responses are to be maximized, and the value \( s = 1 \) was chosen for their desirability functions. The last two responses are “target is best” with \( T_3 = 500 \) and \( T_4 = 67.5 \). The values \( s = t = 1 \) were chosen in both cases. The following experiments were conducted according to a central composite design:

<table>
<thead>
<tr>
<th>Run no.</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( Y_1 )</th>
<th>( Y_2 )</th>
<th>( Y_3 )</th>
<th>( Y_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.00</td>
<td>-1.00</td>
<td>-1.00</td>
<td>102</td>
<td>900</td>
<td>470</td>
<td>67.5</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>-1.00</td>
<td>-1.00</td>
<td>120</td>
<td>860</td>
<td>410</td>
<td>65</td>
</tr>
<tr>
<td>3</td>
<td>-1.00</td>
<td>1.00</td>
<td>-1.00</td>
<td>117</td>
<td>800</td>
<td>570</td>
<td>77.5</td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>1.00</td>
<td>-1.00</td>
<td>198</td>
<td>2294</td>
<td>240</td>
<td>74.5</td>
</tr>
<tr>
<td>5</td>
<td>-1.00</td>
<td>-1.00</td>
<td>1.00</td>
<td>103</td>
<td>490</td>
<td>640</td>
<td>62.5</td>
</tr>
<tr>
<td>6</td>
<td>1.00</td>
<td>-1.00</td>
<td>1.00</td>
<td>132</td>
<td>1289</td>
<td>270</td>
<td>67</td>
</tr>
<tr>
<td>7</td>
<td>-1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>132</td>
<td>1270</td>
<td>410</td>
<td>78</td>
</tr>
<tr>
<td>8</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>139</td>
<td>1090</td>
<td>380</td>
<td>70</td>
</tr>
<tr>
<td>9</td>
<td>-1.63</td>
<td>0.00</td>
<td>0.00</td>
<td>102</td>
<td>770</td>
<td>590</td>
<td>76</td>
</tr>
<tr>
<td>10</td>
<td>1.63</td>
<td>0.00</td>
<td>0.00</td>
<td>154</td>
<td>1690</td>
<td>260</td>
<td>70</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>-1.63</td>
<td>0.00</td>
<td>96</td>
<td>700</td>
<td>520</td>
<td>63</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>1.63</td>
<td>0.00</td>
<td>163</td>
<td>1540</td>
<td>380</td>
<td>75</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>0.00</td>
<td>-1.63</td>
<td>116</td>
<td>2184</td>
<td>520</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>0.00</td>
<td>0.00</td>
<td>1.63</td>
<td>153</td>
<td>1784</td>
<td>290</td>
<td>71</td>
</tr>
<tr>
<td>15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>133</td>
<td>1300</td>
<td>380</td>
<td>70</td>
</tr>
<tr>
<td>16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>133</td>
<td>1300</td>
<td>380</td>
<td>68.5</td>
</tr>
</tbody>
</table>
Using ordinary least squares and standard diagnostics, the fitted responses were the following:

\[
\hat{Y}_1 = 139.12 + 16.49x_1 + 17.88x_2 + 2.21x_3 - 4.01x_1^2 - 3.45x_2^2 - 1.57x_3^2 \\
+ 5.12x_1x_2 - 7.88x_1x_3 - 7.13x_2x_3 \quad \text{(adj. } R^2 = 0.6903)\];

\[
\hat{Y}_2 = 1261.13 + 268.15x_1 + 246.5x_2 - 102.6x_3 - 83.57x_1^2 - 124.82x_2^2 \\
+ 199.2x_3^2 + 69.37x_1x_2 - 104.38x_1x_3 - 94.13x \quad \text{(adj. } R^2 = 0.4562)\];

\[
\hat{Y}_3 = 68.91 - 1.41x_1 + 4.32x_2 + 0.21x_3 + 1.56x_1^2 + 0.058x_2^2 - 0.32x_3^2 \\
- 1.62x_1x_2 + 0.25x_1x_3 - 0.12x_2x_3 \quad \text{(adj. } R^2 = 0.7466)\).

Note that no interactions were significant for response 3, and that the fit for response 2 is quite poor.

Optimization of \(D\) with respect to \(x\) was carried out using the Design Expert software. Figure 14.2 shows the individual desirability functions \(d(Y_i)\) for each of the four responses. The functions are linear since the values of \(s\) and \(t\) were selected equal to one. A dot indicates the best solution found by the Design Expert solver.

The best solution is \(x^* = (-0.10, 0.15, -1.0)\) and results in \(d_1(\hat{Y}_1) = 0.34 (\hat{Y}_1(x^*) = 136.4), d_2(\hat{Y}_2) = 1.0 (\hat{Y}_2(x^*) = 1571.1), d_3(\hat{Y}_3) = 0.49 (\hat{Y}_3(x^*) = 450.46)\) and \(d_4(\hat{Y}_4) = 0.76 (\hat{Y}_4(x^*) = 69.26)\). The overall desirability for this solution is 0.596. All responses are predicted to be within the desired limits.

Figure 14.3 shows a 3-D plot of the overall desirability function \(D(x)\) for the \(x_2 - x_3\) plane when \(x_1\) is fixed at \(-0.10\). The function \(D(x)\) is quite “flat” in the vicinity of the optimal solution, indicating that small variations around \(x^*\) are not predicted to change the overall desirability drastically. However, it is quite important to perform confirmatory runs at the estimated optimal operating conditions. This is particularly true in this example given the poor fit of the response models (e.g., \(\hat{Y}_2\)).

### 14.2.2 Mathematical Programming Approach

The analysis of multiple response systems usually involves some type of optimization problem. When one response can be chosen as the “primary,” or most important, response, and bounds or targets can be defined on all other responses, a mathematical programming approach can be taken. If this is not possible, the desirability approach should be used instead.
In the mathematical programming approach the primary response is maximized or minimized, as desired, subject to appropriate constraints on all other responses. The case of two responses (“dual” responses) has been studied in more detail by some authors and is presented first. Then, the case of more than two responses is illustrated.

### 14.2.2.1 Dual Response Systems

The optimization of dual response systems (DRS) consists of finding operating conditions $x$ that

$$\text{optimize } \hat{Y}_p(x)$$

subject to:

$$\hat{Y}_s(x) = T$$

$$x^T x \leq \rho^2$$

where $T$ is the target value for the secondary response and $\rho$ is the radius of a spherical constraint that limits the region in the controllable factor space where the search should be undertaken. The value of $\rho$ should be chosen with the purpose of avoiding solutions that extrapolate too far outside of the region where the experimental
data were obtained. For example, if the experimental design is a central composite design, choosing $\rho = \alpha$ (axial distance) is a logical choice. Bounds of the form $L \leq x_i \leq U$ can be used instead if a cuboidal experimental region was used (e.g., when using a factorial experiment).

In a DRS, the response models $\hat{Y}_p$ and $\hat{Y}_s$ can be linear, quadratic, or even cubic polynomials. A nonlinear programming algorithm has to be used for the optimization of a DRS. For the particular case of quadratic responses, an equality constraint for the secondary response, and a spherical region of experimentation, specialized optimization algorithms exist that guarantee global optimal solutions. In such case, the algorithm DRSALG^6 can be used (download from http://www.stat.cmu.edu/jqt/29-3), but a Fortran compiler is necessary.

In the more general case of inequality constraints or a cuboidal region of experimentation, a general purpose nonlinear solver must be used and several points should be tried to avoid local optima. This is illustrated in the next section.

14.2.2.2 More Than Two Responses

Example. Three components ($x_1$, $x_2$, $x_3$) of a propellant need to be selected to maximize a primary response burning rate ($Y_1$) subject to satisfactory levels of two secondary responses, namely, the variance of the burning rate ($Y_2$) and the cost ($Y_3$). The three components must add up to 100% of the mixture. The fitted models were
\[
\hat{Y}_1 = 35.4x_1 + 42.77x_2 + 70.36x_3 + 16.02x_1x_2 + 36.33x_1x_3 + 136.8x_2x_3 \\
+ 854.9x_1x_2x_3 \\
\hat{Y}_2 = 3.88x_1 + 9.03x_2 + 13.63x_3 = 0.1904x_1x_2 - 16.61x_1x_3 - 27.67x_2x_3 \\
\hat{Y}_3 = 23.13x_1 + 19.73x_2 + 14.73x_3
\]

The optimization problem is therefore

\[
\text{maximize} \quad \hat{Y}_1(x) \\
\text{subject to} \quad \hat{Y}_2(x) \leq 4.5 \\
\hat{Y}_3(x) \leq 20 \\
x_1 + x_2 + x_3 = 1.0 \\
0 \leq x_1 \leq 1 \\
0 \leq x_2 \leq 1 \\
0 \leq x_3 \leq 1
\]

We can use Microsoft Excel’s “solver” to solve this problem. The table below shows an Excel spreadsheet that has been set up with the problem above. Cells B1:B3 contain the decision variables (cells to be changed), cell E1 is to be maximized, and all the constraints need to be entered appropriately. The figure shows the spreadsheet after the solver completes the optimization. The solution is \(x^* = (0.212, 0.343, 0.443)\), which provides \(\hat{Y}_1 = 106.62\), \(\hat{Y}_2 = 4.17\), and \(\hat{Y}_3 = 18.23\). Therefore, both secondary responses meet the desired bounds. The solver should be run from a variety of starting points (i.e., try different initial values in cells B1:B3 prior to starting the solver) to avoid local optima. Once again, confirmation experiments should be conducted at the estimated optimal operating conditions.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Factors</td>
<td>Responses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>x1</td>
<td>0.21233</td>
<td>Y1(x)</td>
<td>106.6217</td>
</tr>
<tr>
<td>3</td>
<td>x2</td>
<td>0.343725</td>
<td>Y2(x)</td>
<td>4.176743</td>
</tr>
<tr>
<td>4</td>
<td>x3</td>
<td>0.443946</td>
<td>Y3(x)</td>
<td>18.23221</td>
</tr>
<tr>
<td>5</td>
<td>Additional constraint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>x1 + x2 + x3</td>
<td>1.000001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES

Part 5

Case Studies

In Parts 1 through 4 of this book the set of tools necessary to develop, deploy, and customize R2R control solutions has been presented. Specifically, foundational elements have been presented in Part 1, algorithms in Part 2, integration-enabling mechanisms in Part 3, and customization methodologies and strategies in Part 4. In Part 5 we illustrate the application of R2R control tools with control solution deployment case studies. Note that, while case studies of application of R2R control solutions have also been presented in earlier chapters (notably Chapters 1 and Chapter 11), they have generally been presented to highlight one aspect of the R2R control tool set (e.g., the integration technology in Chapter 11). Here, we focus on the deployment efforts themselves.

Specifically, in Chapter 15 the development of an R2R control solution for a CMP (chemical mechanical planarization) process is described. In this application the CMP control solution is developed as the first component of a multistep control solution in a (frequently occurring) CVD*–CMP–Lithography–Etch sequence. The major challenge that must be addressed in this application is the modeling and control of CMP uniformity so as to improve line yield. An innovative approach is described whereby the uniformity target is adjusted to a nonzero value to focus on yield loss due to outer “lip” radial nonuniformity, and to “precompensate” for anticipated downstream etch nonuniformity. This chapter provides insight into the R2R control solution development and customization process, as well as the Genetic Cell Controller (GCC) R2R control solution integration methodology.

In Chapters 16 and 17 focus is placed on customization of EWMA R2R control algorithms as a step in the development and deployment process. The EWMA algorithms, which are discussed in detail in Chapters 1 through 3, are used extensively

* Chemical vapor deposition.
in R2R control. There is, however, a well-recognized need to extend EWMA controllers to address the following two problems: (1) how to choose the EWMA weight parameters and (2) how to modify the basic EWMA algorithms to account for unequal times between measurements (i.e., when the sampling period is not always one run).

The two case studies presented in Chapters 16 and 17 address the two problems above, respectively, and give a practical demonstration of applying the resulting techniques in two different process control environments. In Chapter 16 adaptive control techniques are presented for selecting the tuning parameter of single EWMA controllers. The resulting adaptive EWMA controller is applied to align a stepper photolithography tool. The relevance of the alignment problem is carefully detailed, and two adaptive EWMA controllers are presented. The EWMA controllers are equipped with special filters that allow the user to obtain the desired performance. The adaptation mechanism is protected from “bad” measurements by the application of a spike filter, namely a filter similar to an SPC chart filter.

In Chapter 17 a modification is made to the predictor–corrector double EWMA controller (see Chapters 2 and 3) to allow the controller to handle metrology observations obtained at unequally spaced points in time. The resulting “age-based” controller has the theoretical advantage of being easier to analyze, which produces the practical advantage of being simpler to tune. An application of the aged-based EWMA controller to a CMP process is documented in detail. The performance of the proposed controller is compared, using simulated CMP processes, with that of the standard predictor–corrector EWMA algorithm.
15 Multizone Uniformity Control of a CMP Process Utilizing a Pre- and Postmeasurement Strategy

James Moyne, Chadi El Chemali, Kareemullah Khan, Rock Nadeau, Paul Smith, John Colt, Jonathan Chapple-Sokol, and Tarun Parikh

15.1 INTRODUCTION

As wafer sizes increase to 300 mm, manufacturers must maintain process capability and yield and reduce nonproduct wafer (NPW) usage. Equipment and process design improvements were invoked in the past to address these issues. However, it has become clear that, as noted throughout this book, advanced process control (APC) has become a critical component of the solution for the future, with R2R control the most widely pursued form of APC in the semiconductor manufacturing industry. One process that has repeatedly been shown to benefit from R2R control is chemical mechanical planarization (CMP). The CMP process is described in the Introduction to this book, and CMP R2R control solutions are described in Chapter 11. In the move to 300-mm manufacturing, uniformity control (in conjunction with thickness control) will be a necessary component of R2R control solutions.

The aim of this chapter is to detail the latest advancements in CMP process uniformity modeling and R2R control, and also to illustrate the methodology of developing, deploying, and evaluating an R2R control solution for a semiconductor process. To achieve these goals, the development and deployment of a multizone approach to the modeling and control of CMP uniformity is presented in the remainder of this chapter. The final control solution utilizes both pre- and postprocess metrology data and provides uniformity control to a multizone optimization metric as part of a multiprocess control solution. Specifically, following this introduction, background information is presented in Section 15.2 on a multiprocess control framework that is being implemented to provide yield improvement of a contact process. This includes a specific description of the multiprocess control strategy, the CMP process environment, and the R2R control-enabling technology being utilized. The multizone CMP process uniformity modeling approach is then presented in Section 15.3 along with...
a description of the final uniformity control solution developed. This is followed, in
Section 15.4, with a presentation of results of applying the final uniformity control
solution. Issues associated with the inclusion of the control solution as part of a total
factory solution are then presented in Section 15.5. This chapter concludes with a
summary of the results presented and a discussion of potential future work to extend
the control scheme to include CVD and lithography processes.

15.2 BACKGROUND

15.2.1 CMP Process Uniformity Control

The first CMP R2R uniformity control of CMP was reported on a Strasbaugh CMP
tool and presented in Chapter 11. That work shows that significant improvements in
process capability can be achieved through multivariate (thickness and uniformity)
control of a CMP oxide process. Recently, uniformity control reported on other CMP
tool types further illustrated this advantage. In all cases the uniformity control focus
has been on radial uniformity with a center-to-edge (CTE) metric utilized to quantify
the radial uniformity. Closer analysis of post-CMP process nonuniformity, however,
reveals significant higher-order radial nonuniformity components such as center “dim-
ple” and outer “doughnut” regions. These and other nonradial nonuniformity charac-
teristics, such as across-wafer gradient nonuniformity, are due in part to upstream
chemical vapor deposition (CVD) processing. Understanding and modeling these
nonuniformity characteristics is a key component to developing an improved unifor-
mity control solution for CMP. Providing for detection and weighted control of these
higher-order nonuniformity characteristics is especially important when developing an
R2R control solution as part of a total factory control strategy and solution.

15.2.2 Processing and Control Environment

The CMP uniformity modeling and control work described in this chapter is part of
a multiprocess control solution being developed to improve yield of a contact process
at the IBM Microelectronics facility in Burlington, Vermont. The target contact
process line has a typical CVD, CMP, lithography, etch (RIE) sequence. The control
solution is being developed in a number of phases, with the first phase focused CMP
and RIE R2R process control. The initial envisioned control scheme for (Phase I)
measurement and control is illustrated in Figure 15.1. Note that, with this scheme,
R2R control solutions are envisioned for both CMP and etch processes, with pre-
and postprocess measurement utilized along with interprocess feedforward and feed-
back information flow between the two control solutions. For both control solutions,
the process quality metrics being controlled are postprocess thickness and uniformity,
with these metrics indirectly verified at the postetch step through electrical testing.

In developing this multiprocess control scheme, preliminary results have shown
that, while repeatable results on etch process uniformity were observed, satisfactory
models for controlling etch uniformity could not be obtained (through design of
experiments analysis). Specifically, the analysis did not yield a discernable relation-
ship between process uniformity and total power and power ratio of a split coil
reactor. Thus, the control scheme was modified with the RIE process, R2R, and
feedforward to RIE control components eliminated. The resulting interprocess control solution operates in the following manner:

1. Etch process uniformity is determined at infrequent intervals through pre- and postmeasurement analysis at the etch process.
2. An etch process nonuniformity metric is determined and fed back to the CMP R2R controller.
3. The CMP R2R controller utilizes pre- and postprocess metrology and provides for control of rate variation of CMP removal and radial nonuniformity.
4. The CMP R2R control targets are adjusted to precompensate for the etch process nonuniformity and maximize postetch process yield.

The CMP R2R controller does not necessarily optimize the CMP process, but rather operates as part of a total factory solution to provide the best CMP process for that process line.

15.2.3 Control Enabling Solution

The main objective of this project is to improve process yield through reusable factory-wide integrated control solutions. In order to achieve this objective, the R2R control solution had to meet a number of integration requirements:

Requirement #1: The solution must be generic, i.e., it can be reconfigured to control the various processes on the line. In the first phase of the project, the solution is targeted at CMP and etch processes, but should be sufficiently generic to also be applicable to CVD and lithography processes in future phases of the project.

Requirement #2: The solution has to be able to utilize not only postprocess metrology, but also preprocess metrology and upstream process information from (potentially) multiple sources.
Requirement #3: The solution should be capable of utilizing control algorithms that are suited to the control of the target processes.

Requirement #4: The solution should integrate with the currently existing facility manufacturing execution system (MES), but also provide a migration path to a future APC framework compliant solution.

The R2R enabling technology that is being utilized for the project, called the Generic Cell Controller (GCC), addresses these four requirements. (Detailed information on the GCC concept and solutions is provided in Chapters 9 through 12.) Specifically, with respect to requirement #1, the GCC is a flexible control solution enabler that is fully object-oriented, process-independent, and suitable for factory-wide distributed implementation. GCC-enabled R2R control solutions have been demonstrated to provide control of a number of semiconductor manufacturing processes including CMP, vapor phase epitaxy, and etch. With respect to requirement #2, the GCC technology has been shown to provide fully integrated process control solutions that utilize both post- and premetrology. As for requirement #3, an EWMA linear approximation algorithm with nonlinear extensions was chosen; this algorithm has been utilized effectively for CMP and vapor phase epitaxy process control (see also Chapters 3 and 13). Finally, with respect to requirement #4, Figure 15.2 shows the architecture of the GCC solution when enabled for APC framework-compatible application. Due to the distributed object-oriented nature of the technology, it is configurable to existing MES systems, and can be migrated to (future) APC framework systems as necessary.

The proposed GCC R2R solution integration strategy is shown in Figure 15.2. A GCC solution is instantiated from a single class for each process to be controlled. The class has access to upstream and downstream metrology and target information as necessary to address interprocess control. Both client–server TCP/IP and fully object-oriented interfaces to the GCC station are provided so that the GCC R2R solution can operate in the current client–server integration environment, and can migrate to a (future envisioned) APC framework compliant environment.

FIGURE 15.2 GCC architecture for APC framework integration.
15.3 CMP UNIFORMITY MODELING AND CONTROL

The target CMP tool is a Westech 372M.* A schematic of this machine is shown in Figure 15.3. In this CMP process the wafer is affixed to a wafer carrier via backside air and pressed face-down on a rotating platen holding a polishing pad. A slurry with abrasive material (e.g., alkaline slurry of colloidal silica for oxide or silicon polishing) is dripped onto the rotating platen during polish. The slurry chemically attacks the wafer surface, converting the silicon top layer to a hydroxylated form, which is more easily removed by the mechanical abrasive of the pad.

The control of the CMP process is difficult due to variation and degradation of consumable parts, inconsistency of the slurry, variation in pad physical properties, and the lack of in situ sensors. The main difficulty arises in achieving a reliable film thickness due to change in removal rate over time and the within-wafer uniformity of the polish. Polish rates differ at the center and edge of the wafer due to nonconstant relative pad velocity from the edge to the center, nonuniform slurry and by-product transport under the wafer, wafer bowing due to pressure, or machine drift in time of any of these parameters. In practice, dummy wafers are used to condition and/or calibrate the tool before or after each lot of wafers. The objective is therefore to use an R2R process control to reduce or eliminate monitor wafer usage, and to maintain the performance of the CMP processes by adjusting the removal rate and uniformity of the film thickness. Achieving this goal requires the identification of appropriate input parameters to the CMP process, finding suitable metrics for the process outputs (i.e., removal rate and uniformity), and formulating the appropriate models to be implemented in an R2R controller.8

15.3.1 CMP Uniformity Model Derivation and Testing

Radial CMP polishers impart consistent radial nonuniformity features on a wafer. Figure 15.4 shows a map of a typical oxide wafer after undergoing radial polishing.

* SpeedFam — IPEC Corp.
Different regions can be identified with typical radial polishing, largely due to the fact that polish rates in the center and edge of the wafer are lower than the rates in the in-between region. The film deposition rate from the CVD process, which generally shows a radial pattern, can be substantially different in the center of the wafer because of its singular nature. If the rate of the deposition process in the outside region of the wafer is faster than the mean rate, the outer edge “lip” region will be significantly thicker following polishing because of the combined effects of CVD and CMP. Thus, the film nonuniformity across the wafer can be broken up for modeling.

An initial effort at modeling the CMP nonuniformity attempted to provide a level of isolation of these features by breaking the uniformity metric up into concentric zones, as shown in Figure 15.5. The profile in each zone is approximated to a linear fit, and the slope of this fit is taken as uniformity metric for that particular zone. Minimizing each of the slopes contributes to the improvement of the uniformity.

With a multizone uniformity metric defined, a $2^3$ factorial design of experiments (DOE) was performed to attempt to model process removal rate and uniformity. The inputs to the process are Platen Speed, Carrier Speed, and Backside Air Pressure. This DOE design has eight factor-level combinations, each replicated twice, and five center points. The normalized levels used for each input are shown in Table 15.1. The wafers were measured before and after processing; measurements were taken at 45 sites on the wafer, as shown in Figure 15.6 (6 mm at the wafer edge are excluded).

In order to develop the models for removal rate and uniformity, standard least-squares regression techniques were applied to formulate the predictive model. The analysis yielded the following model for removal rate:

$$RR = -15.66 + 59.68 \, PS + 29.57 \, CS - 16.99 \, PS \, CS,$$  

(1)
where $RR$ is the removal rate, $PS$ is the platen speed, and $CS$ is the carrier speed. The model fits with $R^2 = 98.7\%$. *The regression models for the uniformity slopes

* The $R^2$ value is a measure of how well the model is able to predict the variation in the response.
in Zones 1 to 3 also showed good regression fit $R^2$ but low $F$-ratio* and high $P$-value.**

The significance of the effects are therefore subject to a high $\alpha$ risk. No model was derived for Zone 4 because the $F$-ratios were close to 1. The lack of model in Zone 4 was unfortunate since, as noted above, control in this outer zone is critical to improving yield. This provided motivation for adjusting the uniformity modeling strategy, as described later in this section.

\[
s_i = 11.46 - 9 \; CS - 6.64 \; BA + 11.64 \; PS \; CS
\]  
(2)

\[
s_2 = -10.65 + 2.78 \; CS + 8.75 \; PS + 10.59 \; BA - 15.01 \; PS \; BA
\]  
(3)

\[
s_3 = -3.05 - 21.65 \; PS - 31.08 \; CS + 21.64 \; PS \; CS + 14.89 \; PS \; BA
\]  
(4)

where $s_i$ is the slope at zone $i$, $i = 1, 2, 3$. The models fit with $R^2 = 86.5\%$, $R^2 = 90.8\%$, and $R^2 = 88.7\%$, respectively.

Process R2R control experiments were run at IBM to test the validity of the models for thickness and nonuniformity. Two uniformity zones were selected to be controlled: Zone 1 and Zone 3. Zone 2 was not selected because it was relatively flat throughout the DOE, and there was a desire to keep the control problem over-determined and more manageable. The linear approximation algorithm introduced in Section 15.2 was utilized in the R2R controller. As an example, with this algorithm applied to thickness control, the removal rate derived from the DOE analysis (Eq. 1) is approximated to be constant. The removed thickness model used is described by the following equation:

\[
RmvdT[n] = RR[n] \cdot t[n]
\]  
(5)

where $RmvdT$ is Removed Thickness, $t$ is polish time, $RR$ is Removal Rate constant, and $n$ refers to current run. This model was implemented in the GCC R2R controller with the RR constant updated at each run to compensate for drift and noise, according to an EWMA filtering mechanism:¹²

\[
RR[n] = \alpha \left( \frac{RmvdT[n - 1]}{t[n - 1]} \right) + (1 - \alpha) \; RR[n - 1]
\]  
(6)

where $\alpha$ is an EWMA weighting factor ($0 \leq \alpha \leq 1$), the value of which is selected based on considerations of noise, drifts, shifts, and model error.¹⁴

---

* $F$-ratio is the statistic used to evaluate whether the parameters are statistically significant. The lower the $F$-ratio for a parameter, the less likely that parameter is statistically significant.
** $P$-value is the probability that the $F$-value fails the test and the model parameters are not statistically significant. A $P$-value less than or equal to a chosen $\alpha$ risk indicates that the model parameters are significant, where $\alpha$ risk is the probability of declaring that a model parameter is significant when it is not.
The suggested time for the next run \( n \) is then calculated as

\[
t[n] = (\text{ST}[n] - \text{Target})/\text{RR}[n],
\]

where \( \text{ST} \) is the pre-CMP film thickness of the wafer to be polished.

The results of applying model-based control to the multivariate control of thickness and (multizone) uniformity revealed a good capability for thickness control, but an inability to provide uniformity control. As an example, Table 15.2 summarizes the model predicted vs. measured slopes in Zones 1 and 3 for four control runs. Clearly, the model does not accurately predict the uniformity in these zones.

As a result of a subsequent analysis of the uniformity modeling problem, it was hypothesized that the main reason for the poor modeling of the slopes is the existence of a nonradial cross-wafer gradient, which has been traced to the upstream CVD process. Figure 15.7 shows the film thickness profile along the \( x \) and \( y \) axes of a wafer before and after the CMP process, respectively. The individual measurements are plotted with error bars. The (nonradial) cross-wafer gradient at the upstream CVD process is closely correlated with post-CMP process cross-wafer gradient. The nonradial cross-wafer gradients invalidate the radial uniformity symmetry assumption and add variability to the DOE data, thus hiding the true level of significance of the input parameters on the slopes’ values.

A second DOE was performed to address the CMP uniformity control problem. Prior to this DOE, maintenance was performed on the upstream CVD process and the gradient problem was drastically reduced. Further, a simplified single radial uniformity metric was used. Specifically, uniformity was measured as the center-to-edge slope of the thickness profile, where the slope was determined through regression analysis. This simplified approach to uniformity modeling was taken because it was felt that satisfactory uniformity control could still be achieved through control to a nonzero slope that minimizes the yield loss effect due to nonuniformity in the outer lip region (see Figure 15.12 later in this chapter). Analysis of data from the second DOE yielded the following models for removal rate and center-to-edge radial uniformity slope:

<table>
<thead>
<tr>
<th>Run</th>
<th>Slope 1 (meas.)</th>
<th>Slope 1 (pred.)</th>
<th>Slope 3 (pred.)</th>
<th>Slope 3 (pred.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.20</td>
<td>7.4600</td>
<td>-4.990</td>
<td>-19.25</td>
</tr>
<tr>
<td>2</td>
<td>5.2974</td>
<td>0.34</td>
<td>1.4760</td>
<td>-3.26</td>
</tr>
<tr>
<td>3</td>
<td>4.6302</td>
<td>1.1974</td>
<td>-0.72</td>
<td>-2.534</td>
</tr>
<tr>
<td>4</td>
<td>3.8337</td>
<td>4.45</td>
<td>-11.27</td>
<td>-3.96</td>
</tr>
</tbody>
</table>

Note: meas. = measured slopes; pred. = predicted slopes.
where RR is the removal rate and CTE is center-to-edge slope. Note that the above models are coded to utilize normalized inputs (between +1 and 1). The Removal Rate model has a high value of $R^2 = 90\%$, and the PS input is significant with a $P$-value less than 0.0001. The CTE model has an $R^2 = 66.7\%$, which is relatively low; however, the effects of Platen Speed and Carrier Speed are more significant than in with the first DOE, with $P$-values of 0.0038 and 0.0105, respectively. Therefore, the two models are statistically valid and accurate enough to predict the removal rate and uniformity slope of the film oxide.

### 15.4 RESULTS: MULTIVARIATE CONTROL OF CMP

The models derived from the second DOE analysis, (8) and (9), were utilized to configure the GCC R2R control solution containing a two-stage linear approximation EWMA control algorithm. This controller, referred to as the GMt algorithm,\textsuperscript{12} is an extension of the EWMA algorithm described in Chapter 3. It provides more accurate control of systems that cannot be accurately approximated as linear functions of the process inputs. In the CMP process, for example, the removed thickness metric is a function of the removal rate multiplied by the process variable time, i.e., $t$, as

\begin{align*}
RR &= 20.2 + 1.91 \ PS \\
CTE &= -6.550.5 \ PS + 0.34 \ CS
\end{align*}
shown in Eq. (5). We see that, in this case, the process quality parameter, *amount removed* (RmvdT), cannot be approximated by a linear function of the CMP machine parameters, PS and CS, because these parameters impact the true process variable, i.e., the RR. The GMt algorithm is a suitable control solution in this case because it models the relationship between the input parameters, PS and CS, and the primary output, RR, and then models the multiplicative relationship between this primary output and the secondary output, the RmvdT.

The GMt provides a control solution for our models by breaking the system outputs down into two sets, namely primary output(s) and secondary inputs, as shown in Figure 15.8. Therefore, the GMt implements a two-step linear solution process that approximates a nonlinear solver. In the first step, the controller uses the GCC R2R EWMA method to compensate fully for the primary outputs that do not have corresponding secondary outputs, and to partially compensate (in a weighted fashion) for those primary outputs that also have corresponding secondary outputs. The GMt then completes the compensation on the secondary outputs by adjusting secondary inputs.

For the CMP control, the removal rate thickness model given in Eq. (8) and the center-to-edge uniformity model given in Eq. (9) are used in the primary stage with PS and CS as primary inputs and RR and CTE as primary outputs, as shown in Figure 15.8. In the first stage of the controller, the GMt compensates for RR drift and for CTE. In the second stage, a secondary input of *time* is adjusted to compensate for the secondary output, the *amount removed*.

The results of application of this control solution at the IBM Microelectronics facility are shown in Figures 15.9 and 15.10. In each figure the controlled output metrics of remaining thickness and nonuniformity slope, respectively, are plotted along with the desired target output and an estimation of the uncontrolled output. The uncontrolled output value for each metric is estimated through applying the difference between the current controlled recipe and the starting recipe to the process model for the metric. Specifically, the uncontrolled thickness value is estimated as

\[
\text{Thickness}_{\text{Uncontrolled}} = \text{Actualrem} + \Delta\text{rem}
\]  

where *Actualrem* is the actual amount of oxide removed, i.e., the difference between premetrology measurement and postmetrology measurement of oxide thickness.
\( \Delta \text{rem} \) is the amount of thickness to be added or removed if no controller exists. \( \Delta \text{rem} \) is given by

\[
\Delta \text{rem} = (\text{Actual}/\Delta \text{RR} + \Delta \text{RR}) \times (t_N - t_S)
\]  

(11)

where \( t_N \) is the nominal time recipe and \( t_S \) is the actual time recipe suggested by the controller. \( \Delta \text{RR} \) is the amount to be added or removed to the removal rate if no controller exists. From Eq. (8), \( \Delta \text{RR} \) is calculated as

\[
\Delta \text{RR} = 1.01 \Delta \text{PS}
\]  

(12)

The uncontrolled uniformity value is estimated as

\[
\text{Uniformity}_{\text{Uncontrolled}} = \text{Actual} \text{CTE} + \Delta \text{CTE}
\]  

(13)

where \( \text{Actual} \text{CTE} \) is the actual amount of the slope uniformity, i.e., the difference between premetrology measurement and postmetrology measurement of center-to-edge-metric. \( \Delta \text{CTE} \) is the amount of slope uniformity to be added or removed if no controller exists. From Eq. (9), \( \Delta \text{CTE} \) is calculated as:

\[
\Delta \text{CTE} = 0.34 \Delta \text{CS} - 0.5 \Delta \text{PS}
\]  

(14)
where $\Delta CS$ is the difference between the controlled and nominal carrier speed recipe and $\Delta PS$ is the difference between the controlled and nominal platen speed recipe.

The results presented in Figures 15.9 and 15.10 indicate the multivariate control of both thickness and uniformity. This is further illustrated by comparing the post-process wafer map of the final control experiment, Figure 15.11, with that of a typical uncontrolled process run (Figure 15.4).

### 15.5 ISSUES: A CMP CONTROL SOLUTION AS PART OF A TOTAL FACTORY CONTROL SOLUTION

To date, run-to-run control research has been focused for the most part on the control of single processes. A key goal of the control development effort presented in this chapter, however, is that the control solution be part of a total factory control solution. While this goal imparts requirements of configurability, reusability, etc., on the control solution enabler (as presented in Section 15.2), it also raises other issues with respect to design of the factory control solution. A discussion of the more prominent issues is provided in this section.

#### 15.5.1 CMP Uniformity Control for Yield Maximization

As shown in Section 15.2, CMP radial process nonuniformity is characterized by a number of zonal features. The efforts in CMP uniformity control to date have been
focused on minimizing a single CTE metric. While this approach has been shown to provide increased process capability, the effort described in this chapter has been focused on establishing a relationship between nonuniformity targets and yield, and configuring the CMP controller to those targets. Of the four uniformity zones described in Section 15.3 and illustrated in Figure 15.5, IBM researchers have qualitatively determined that nonuniformity in the outer lip region is the largest source of yield loss. Thus, in providing a uniformity control solution, the target must be weighted to the minimization of nonuniformity due to this outer lip region. Considering that the uniformity control solution developed is limited to controlling one uniformity parameter, namely the slope of radial nonuniformity, it is this target that must be adjusted. Luckily, the outer lip region is characterized consistently by an upward sloping profile; thus, the impact of this region on the entire process can be minimized by targeting a slightly downward sloping radial uniformity gradient as shown in Figure 15.12.

15.5.2 CMP Uniformity Control for Precompensation of Etch Nonuniformity

Establishing controller performance metrics at the factory level rather than the process-centric level impacts the formulation of the control solution. Specifically, by attending to overall yield, particular control solutions at a process may be configured to control “nonoptimal” process-centric targets so that they may provide “precompensation” for downstream processes. This precompensation method from CMP to etch is illustrated graphically in Figure 15.13. The etch process does not have a developed R2R control capability. However, through pre- and postetch metrology, the etch process uniformity is characterized. This information is fed back to the CMP process and the CMP R2R process control targets are adjusted to precompensate.
for etch nonuniformity. In the current control solution described in this chapter, the precompensation feedback is static, i.e., the CMP R2R controller targets are adjusted manually as necessary to provide precompensation. Further, the precompensation assumes that a specific etch tool is chosen for wafers processed on a particular CMP tool. Providing a dynamic configurable precompensation capability, and addressing precompensation as a component of chamber matching with banks of tools, are topics of future work (see Section 15.6).

15.5.3 CMP Uniformity Analysis for CVD Gradient Nonuniformity Limits Monitoring

Analysis of the first CMP process DOE (see Section 15.3) revealed that the pre-CMP process nonuniformity contains two distinct components: (1) a radial nonuniformity component, and (2) an across-wafer gradient component. While straightforward techniques can be used in many cases to deconvolute these two components, the choice of metrology strategy in the first DOE (see Figure 15.6) and lack of

FIGURE 15.12 CMP uniformity targeting for yield maximization.

FIGURE 15.13 Applying precompensation at CMP for postetch uniformity.
knowledge of wafer alignment resulted in insufficient data to carry out the deconvolution. This issue was addressed in the second DOE with a focus on minimizing across wafer nonuniformity at CVD, and a new pre-CMP metrology measurement pattern to better detect the axis of across-wafer nonuniformity. However, the results of the DOE experiments also revealed another use for pre-CMP measurement, namely control and/or limits monitoring of uniformity imparted at the upstream CVD process. Specifically, across-wafer gradient nonuniformity at CVD is largely due to nonuniform spacing between the reactant source (a showerhead) and the wafer. The factors affecting this type of nonuniformity cannot be “tuned” on an R2R control base. However, the equipment can be taken off line and the problem can be mechanically corrected. Therefore, while pre-CMP measurements of gradient nonuniformity cannot be used as feedback for a CVD R2R control solution, they can be used in a limits-monitoring capacity to initiate process maintenance activities at the CVD.

Radial nonuniformity at CVD is due to a number of factors, including the distance between the deposition source and the wafer. It is hypothesized that this type of nonuniformity is controllable to a degree, and thus pre-CMP radial nonuniformity data could be utilized as part of a CVD radial uniformity R2R control scheme. Development of such an R2R controller is being considered as a future effort in this project (see Section 15.6).

15.6 CONCLUSIONS AND FUTURE WORK

In this chapter we have presented a multizone approach to the modeling and control of CMP radial nonuniformity. The work focuses on CMP control as part of a total factory control solution rather than a process-centric solution. While the analysis does not reveal a capability for control of the individual radial zones, it does provide data for the development of uniformity metrics and control solutions that are tied to maximization of yield in the process line rather than minimization of nonuniformity at the CMP process. The solution has been applied to a Contact process at IBM’s Microelectronics facility. This solution is configurable and process-independent, is capable of accommodating pre- and postmetrology information, and is designed to be utilized as an integral part of a total factory (interprocess and multiprocess) control solution. The results of application of the CMP controller indicate a capability for simultaneous control of both CMP process thickness and uniformity. The results also indicate that (1) the CMP nonuniformity target for the controller should be set to a nonzero value to compensate for weighting of the outer lip area, which is a major contributor to yield loss; (2) in order to maximize line yield, the CMP nonuniformity target should also be adjusted to precompensate for radial nonuniformity characterized at the downstream etch process; and (3) with the appropriate measurement scheme in place, CMP pre- or postmetrology could be utilized to determine contributions to radial and gradient nonuniformity from the upstream CVD process. This information could be fed back to the CVD process as part of a process maintenance system and R2R control scheme for CVD.

Future efforts should be focused on further development of the total factory control solution. CVD process maintenance alarming and R2R process control
elements could be used to reduce yield loss due to CVD misprocessing, and to provide a more consistent uniformity profile for CMP processing. Further, depending on the level of controllability of CVD radial process uniformity, CVD R2R control could be used as a radial uniformity precompensation tool for both CMP and etch. Other efforts should be focused on continuing to provide generic and configurable control solution enablers. Specifically, it is proposed that the current GCC generic R2R process control solution enabler (which currently supports pre- and postmetrology input) be enhanced to provide dynamic adjustment to the control process due to any number of upstream advices, and any number of downstream precompensation requests. Development of such an enabler would provide an avenue for rapid implementation, configuration, and test of multiprocess, factory-wide control schemes.

ACKNOWLEDGMENT

Portions reprinted with permission from the proceedings of the 46th International Symposium of the American Vacuum Society. The authors would like to thank Victor Solakhian of the University of Michigan for controller software development, Jason Silbergleit of IBM Microelectronics for RIE process support, and John Taylor of Compugenesis for providing a GCC-to-CMP tool SECS interface solution.

REFERENCES

1. Rozich, W., “Minimizing the Risk of 300mm Wafer Introduction into Semiconductor Manufacturing,” SEMATECH AEC/APC Symposium XI, Vail, CO (September 1999).
4. SEMATECH AEC/APC Symposium XI Proceedings, Vail, CO (September 1999).
7. Jarrett Campbell, W., “Run-to-Run Control of Within-Wafer Uniformity in Oxide Chemical Mechanical Planarization,” SEMATECH AEC/APC Symposium XI, Vail, CO (September 1999).
16 Control of Photolithography Alignment

Nital S. Patel and Robert A. Soper

16.1 INTRODUCTION

A modern semiconductor device exists in three dimensions; throughout the course of its manufacture, a chip experiences the patterning of approximately two dozen different layers, each of which must be precisely positioned with respect to the one beneath.\(^1\) This positioning (or alignment) is necessary in order to ensure the correct operation of the final device. Sufficiently large alignment errors can result in many different types of failures; the exact nature depends on the specific levels involved. As an example, if the misalignment of the contact layer to the implanted substrate (the well) is too great, shorts or diode leakage can result (see Figure 16.1).

The rapid pace of advancement in consumer electronics has been driven in large part by the rapid increase in computing power available from computer chips of ever-decreasing size. These two driving forces have required that computer chips become both more complex and more compact simultaneously. Together, such demands place stringent requirements on alignment accuracy.

Until very recently, alignment requirements could be met using statistical process control (SPC) techniques, wherein the alignment input settings were held more or less fixed, and the sources of alignment variation were identified and removed.* Unfortunately, as alignment requirements have become tighter and the known sources of variation have been removed, the quality of alignment control afforded by SPC has ceased to be sufficient. SPC is particularly unsuited to the task of controlling alignment in the face of unexplained disturbances because of its inability to take corrective action. Combating disturbances has become a significant drain on the photolithography process engineers who are responsible for alignment. The engineers, in effect, are the controllers in informal feedback loops around the alignment process.

A need is thus identified to provide automated control of the alignment process. This chapter describes a methodology for run-to-run control of photolithography alignment. By replacing the engineers in these feedback loops with run-to-run controllers, a number of benefits can be realized. First and foremost, the engineers’

* SPC techniques were introduced in Chapter 1.
time is freed to pursue more important activities. Alignment is essentially a non-value-added process, and the less effort expended on it, the better. In some cases the quality of the alignment can be improved by continual corrective action. A collateral benefit is uniformity of control action. When faced with an alignment failure, individual engineers will use their own personal experience as a guide in determining the type and magnitude of the correction to make. As the levels of experience of process engineers vary widely within a company, even within a single process engineering group, the style of alignment correction will vary correspondingly. With a run-to-run controller in place, all corrections are completely predictable and repeatable. This predictability also makes the manufacturing process for a device more transferable from one facility to another.

This chapter is structured as follows. In the remainder of this introduction section, the alignment process is introduced, including discussions of misalignment measurement and compensation, and sources of measurement error. This is followed in Section 16.2 by a detailed discussion of the metrology technique, and in Section 16.3 by a presentation of the R2R control solution. This chapter concludes with a brief summary and some final thoughts.

16.1.1 Alignment Process

Before actually exposing the photoresist on a pattern level that requires alignment, the stepper takes a series of actions intended to provide accurate alignment — in effect, performing internal feedforward control. There are three separate actions: measuring existing misalignment in the layer to which the current layer is being aligned, modeling the raw misalignment data to determine the optimum corrections to make to match the current layer as closely as possible to the existing layer, and applying alignment corrections during exposure.

16.1.2 Misalignment Measurement

Alignment quality is generally defined in a relative sense; i.e., the alignment of one layer to the next is expressed as the difference between their positions. The stepper is the only piece of photolithography equipment that uses an absolute coordinate system and is therefore able to measure the misalignment of a single layer without another layer to use as a reference. This capability (provided by the extremely accurate interferometer-controlled stepper stage) is used to good effect in controlling alignment.
Each pattern level to which another level is to be aligned contains a set of structures that the stepper uses to determine absolute position. Several (typically one to four) of these structures are measured in each of a number of fields (approximately six to ten) around the wafer (Figure 16.2). These measured locations are compared to the desired locations defined by the device layout. The differences between the actual and desired locations are recorded as raw \( X \) and \( Y \) alignment errors.

16.1.3 Stepper Compensation

Once measured, the alignment errors are modeled by a set of linear stage- and wafer-level parameters. These model parameters correspond to the stepper input variables that can be manipulated to control the ultimate layer-to-layer alignment.

After measuring the incoming wafer alignment errors and modeling them into correctable components, the stepper combines the calculated optimum input values with any offsets specified by the user to arrive at the final input settings. These settings are then used in exposing the current wafer. The process of alignment measurement, modeling, and compensation is repeated for each wafer independently. A run-to-run controller would interact with the stepper by determining the values of these user-specified offsets. Typically, they would be sent to the stepper automatically through some sort of network connection, but they could also be entered manually.

16.1.4 Process Biases

In theory, the feedforward action afforded by measuring the inherent alignment errors on the wafers before exposure should result in perfect alignment; however, this is not the case for several reasons. The stepper’s alignment measurement system relies on optical technology (broad- or narrow-band visible light); the alignment mark is covered by, and is itself made up of, optically transparent materials that have a thickness on the order of the illuminating wavelength. As a result, diffraction and interference phenomena can bias the alignment readings. In addition, the stepper’s
alignment metrology system is subject to changes in behavior over time due to mechanical wear, miscalibration, maintenance, etc. Finally, the input settings are defined relative to arbitrary baselines, which can be changed in order to optimize other aspects of stepper operation.

16.2 ALIGNMENT METROLOGY

16.2.1 Metrology Technique

Measuring the alignment of one layer to another after patterning is accomplished with a separate piece of equipment that reads a composite mark containing two elements, one patterned in each layer. A typical mark comprises two nested boxes — the outer box is etched in the substrate and the smaller, inner box is patterned in photoresist at the current step.

Multiple copies of these box-in-box structures are printed in each stepper field; a common arrangement is to have a box-in-box structure in each corner of the field, with a number of fields measured around the wafer. The choice of number of fields (and number of wafers per lot) to measure is driven by many, sometimes competing, factors, e.g., desired throughput, measurement noise, desired precision, etc.

An alignment metrology tool uses pattern recognition techniques to locate the edges of each box and calculate the amount by which the inner box is off center with respect to the outer box, in both the X and Y directions.

The alignment metrology tool fits the same alignment model that was used previously in the stepper to the raw misalignment data. The values of the modeled parameters (averaged across the wafers measured) are the process measurements that are sent to the run-to-run controller for feedback to the stepper.

16.2.2 Sources of Error

As in the stepper, final alignment measurement is also optical and, therefore, sensitive to variation in material properties such as refractive index and thickness. Since the technique depends on detecting the edges of the box-in-box structure, it is also influenced strongly by edge contrast, roughness, etc., which can be changed by process variations in other areas. It is not uncommon for metal pattern levels to cause measurement difficulties; the grainy substrate scatters light, confusing the edge detection algorithm, and the etched outer boxes are frequently deformed (refer to Figure 16.3). The run-to-run control system must be designed with these challenges in mind, and must be able to reject “fliers” that do not represent the true state of the process. This is a nontrivial requirement that plays a significant role in determining the ultimate utility of the controller in a real manufacturing environment.

16.3 CONTROLLER DESIGN

The previous sections presented the principles by which the stepper and alignment metrology tools operate. In this section, the impact of these on controller design is first considered. Having done so, two designs are presented that are currently being
employed in multiple fabs within Texas Instruments (TI). The first is based on integral (EWMA-based) control, and the second design adds additional filtering to better trade off noise attenuation and response times. The effect of stepper lens distortion, and varying die sizes, is seen in the appearance of device dependencies. Specifically, it is observed that certain devices exhibit different markshift behavior from others. At present, these dependencies are relatively small (<20%) compared to the specs, and given the need to make the system manufacturing-friendly, such dependencies are ignored. Note that trying to weed out and correct for such dependencies, especially in an environment where multiple new devices are introduced each week, is both cumbersome and results in large data sets. The latter results in software performance issues that impact lot login and logout times (cycle times) in the fab. However, such device dependencies do impact the noise characteristics seen by the controller. This noise is now driven by product mix. In addition, there are metrology issues dealing with recipe maintenance and spikes (e.g., due to the metrology structures being affected by other processes). An incorrectly written recipe will cause a device to fall out, and the recipe will be corrected. However, multiple lots of such a device could have run prior to the recipe error being detected. This results in a temporary increase in noise magnitude. Figure 16.4 shows the effect of device dependencies on process noise. At the same time, a stepper could run any one of a multitude of pattern levels. These different levels see different drift rates by virtue of nonuniform sampling. For example, a level that has multiple runs a day would generally require less aggressive correction than one that has a run every other day. Also, the same pattern level could exhibit different noise characteristics across different steppers (Figure 16.5).

Given the fact that there could be thousands of overlay loops (one for each variable, per pattern level, per stepper), it is virtually impossible to manually reoptimize these controllers in order to cope with such variations in noise and disturbance characteristics. Instead, adaptive controllers are employed to effectively deal with this problem (as well as fan-out). The designs presented will focus primarily on the loop for x-markshift. Control loop design for the other variables is done similarly. The design of the controllers is based on the algorithm presented in Reference 3. This reference also provides an example of applying the algorithm developed for controlling

![Distorted (left) and normal (right) box-in-box alignment metrology structures.](image)
overlay. Here, a simplified version of the tuning algorithm is presented. This simplification is made noting that the typical disturbance encountered in overlay control is that of a shift vs. drift (especially after stepper maintenance). Most of the sophistication required for the algorithm in Reference 3 was due to the need to optimally deal with drifts. In case of shifts, the scheme can be simplified considerably by noting that the asymptotically optimal value of the controller gain is 0. The markshift loop can be modeled as follows:

\[ y_{k+1} = \theta u_k + d_k + w_k, \quad k = 0, 1, 2, \ldots \quad (1) \]

where \( y_{k+1} \) is the offset measured at metrology, \( u_k \) is the offset input to the stepper, \( d_k \) is the disturbance that the controller is required to compensate for, \( w_k \) is (high-frequency) noise, and \( \theta \approx 1 \) is the stepper gain. An EWMA-based controller is considered first. The results are then extended to second-order controllers due to their ability to provide better noise attenuation while preserving the response times.

### 16.3.1 The EWMA Case

For the EWMA case, the controller is given by

\[ x_{k+1} = x_k - \lambda_k \left( y_{k+2-\tau} - u_{k-\tau} \right), \quad x_0 = 0, \quad k = 0, 1, 2, \ldots \quad (2) \]

\[ u_k = y \cdot \text{round} \left( \frac{1}{\gamma} x_k \right) \]

where \( \tau_k \geq 0 \) is the (variable) delay to metrology, \( \gamma \) is the precision of the stepper input, \( x_k \in \mathbb{R} \) is the controller state, and \( \lambda_k \) is the controller gain. The value of \( \lambda_k \) is determined by the following recursion:

\[ \mu_{k+1} = \mu_k + \epsilon \left( y_{k+1-\tau} + u_k - \mu_k \right) \quad (3) \]
FIGURE 16.5 Example of interstepper differences. The same pattern level exhibits different noise characteristics on steppers A and B.
\[ \xi_{k+1} = \xi_k + \varepsilon \left( (y_{k+1+\tau_k} - u_{k-\tau_k} - u_k)^2 - \varepsilon_k \right), \mu_0^2 \leq \xi_0 \]

\[ \lambda_k = \lambda \frac{\delta^2 + 2\mu_k^2}{\delta + \mu_k^2 + \xi_k}, k = 0, 1, 2, \ldots \]

with \(1 > \varepsilon, \delta > 0\) with \(\varepsilon, \delta \approx 0\) and \(0 < \lambda \leq 2\). Equations (2) and (3) have been written in a form that essentially takes the delay out of the loop, and minimizes its impact on stability. Equation (3) is similar to the one derived in Reference 3, except that in this case the value of \(\lambda_k\) is forced to lie in the open interval \((0, \lambda)\). The value of \(\lambda\) is chosen to satisfy gain margin requirements in order to ensure closed-loop stability. In addition to this, a spike filter is employed to prevent the controller (2) from updating on data from maverick lots. This filter works by looking at the measurements from consecutive lots and allows the controller state \(x_k\) to update only if this difference is less than some given value. One way to implement this is as follows (4), where the controller Eq. (2) is modified with the addition of an extra state \((\tilde{x}_k)\) to reject spikes having a magnitude greater than \(\Delta > 0\).

\[ x_{k+1} = x_k - \lambda_k \left( y_{k+1+\tau_k} - u_{k-\tau_k} + x_k \right) \cdot \text{round} \left( \frac{\Delta}{\Delta + \text{abs}(y_{k+1+\tau_k} - \tilde{x}_k)} \right) \]

\[ \tilde{x}_{k+1} = y_{k+1+\tau_k} \]

\[ u_k = \gamma \cdot \text{round} \left( \frac{1}{\gamma} x_k \right). \]

**Figure 16.6** shows an example of an EWMA-based x-markshift controller for a mix-and-match (previous layer was patterned on a different stepper) implant level. Mix-and-match levels see additional noise due to stepper mismatch (both in lens.

![Graph showing behavior of x-markshift for a mix-and-match implant level.](image-url)
distortions and reference baselines). In the example, the controller rejects a shift that occurs at approximately run 20. The dashed line shows a simulation of what the output would have been without the input correction (refer to (1)).

16.3.2 Second-Order Controller

In certain situations (i.e., for certain pattern levels), one is unable to achieve the required amount of high-frequency noise attenuation while still preserving the response speed by using an EWMA-based controller alone. To this end, the EWMA-based controller is augmented with a low-pass filter to yield the following controller equations (the spike filter is retained):

\[
x_{k+1} = x_k + \left[\begin{array}{cc}
-\frac{\lambda_k}{2} & -\frac{\lambda_k}{2} \\
1 - e^{-2\lambda_k} & 3e^{-2\lambda_k} - 1
\end{array}\right] x_k - \left[\begin{array}{c}
\frac{\lambda_k}{2} \\
4 e^{-2\lambda_k} - 1
\end{array}\right] (y_{k+1-\tau_k} - u_{k-\tau_k})
\]

\[
\text{round}\left(\frac{\Delta}{\Delta + \|y_{k+1-\tau_k} - \tilde{x}_k\|}\right)
\]

\[
\tilde{x}_{k+1} = y_{k+1-\tau_k}
\]

\[
u_k = \gamma \cdot \text{round}\left(\frac{1}{\gamma}[1 1] x_k\right)
\]

where \([x, \tilde{x}]^T \in \mathbb{R}^3\) are the controller states (the base controller is second order), and once again the delay is taken out of the loop. Equation (5) is the discretized (assuming a zero-order hold) form of a critically damped continuous time controller having the following Laplace transform:

\[
C(s) = \frac{\lambda_k^2}{s(s + 2\lambda_k)}, \quad \lambda_k > 0.
\]

The additional high-frequency attenuation achieved by this controller (5) is apparent from the Bode plot shown in Figure 16.7, as is the extra phase lag. As seen in Figure 16.8, the controller exhibits the noise attenuation properties of a low-gain EWMA-based controller, while achieving response times equivalent to a high-gain EWMA. Stability for the closed-loop system (assuming unity stepper gain) is guaranteed provided that the roots of the following (characteristic) equation are less than 1 in magnitude:

\[
4z^2 + \left(2\lambda_k - 3e^{-2\lambda_k} - 5\right)z + 3e^{-2\lambda_k} - 2\lambda_k e^{-2\lambda_k} + 1 = 0
\]
This is true provided $0 < \lambda_k < 5$. However, the maximum allowable value of $\lambda_k$ is typically chosen to be much smaller than this to achieve sufficient gain and phase margins. For example, setting $\lambda_k = 0.4$ yields gain and phase margins of 21 dB and 71°, respectively. Increasing this to 0.8 results in a phase margin of 65° and the introduction of undesirable underdamped behavior. This underdamping is the result of discretization.

In this case also, the gain $\lambda_k$ is allowed to vary according to Eq. (3). Furthermore, due to the phase lag, large adaptation rates can be chosen (i.e., the gain is allowed to fluctuate more) since the controller response is less sensitive to fluctuations in the gain (as compared to the EWMA case). Figure 16.9 shows the performance of this controller for control of x-markshift (arbitrarily scaled) at metal-1 pattern. Figure 16.9 (top) also shows simulated data obtained by backing out the controller corrections from the measured misalignment employing the model (1). The corresponding controller gain ($\lambda_k$) is shown in Figure 16.9 (bottom). It is seen that the controller gain remains high when the stepper experiences a shift/drift disturbance. After this disturbance is compensated for, the controller gain returns to a lower value. The maximum controller gain allowed was 0.4, and at no instance is this value exceeded (as predicted by theory).

The adaptive scheme proves very useful in fan-out and maintenance of the controllers. In addition to allowing identical controllers to be employed across multiple pattern levels and steppers, the system also results in zero maintenance.
FIGURE 16.8 Step response comparison of EWMA-based and second-order controllers.

FIGURE 16.9 Behavior of x-markshift for metal-1 pattern: % error (top), and controller gain (bottom).
The EWMA-based and second-order controllers are both being employed to control lithography overlay in TI, with different fabs preferring one controller structure over the other. Based on experience, it appears that the EWMA case works best for fabs with low volume and low product mix, which have less device and metrology recipe-induced noise, and need very aggressive control. For a high-mix, high-volume environment, the second-order controllers prove superior due to their additional filtering capability. The initial lag in the controller response is inconsequential in such a setting, and in fact helps prevent spurious control action.

16.4 CONCLUSIONS

Two examples of adaptive feedback controllers being employed to control lithography overlay in TI fabs have been presented in this chapter. The first example was that of an EWMA-based controller, and the second example built on this by incorporating additional low-pass filtering. It is expected that with scaling device geometries and tightening alignment requirements, lens distortion will contribute significantly to the overall alignment error budget. This will prompt a move toward device-dependent overlay control. Efficient software will be required to implement such control schemes in a manufacturing-friendly fashion, particularly in high-mix ASIC (application-specific integrated circuits) fabs. Other issues that will arise is the need to ensure adequate sampling across various pattern levels in order to further enhance system performance.

ACKNOWLEDGMENTS

The authors would like to acknowledge the following people who have contributed to development and deployment of overlay feedback control in TI: Stephanie Hilbun, Eddie Brooks, Doug Ballard, Adriana Sanchez, and Steve Jenkins.

REFERENCES

17 Age-Based Double EWMA Controller and Its Application to a CMP Process

Argon Chen and Ruey-Shan Guo

17.1 INTRODUCTION

The basic R2R control algorithm solutions presented in Chapter 1, and especially Chapter 3, are based on the R2R control scheme originally proposed by Sachs et al.\(^1\) In this scheme, the EWMA statistic is used as an estimate of the process deviation from its target. However, the controller based on the EWMA statistic is not sufficient for controlling a wearing out process. The predictor–corrector controller (PCC) algorithm, also described in Chapter 3, was developed to enhance the EWMA R2R controller’s capability.\(^2,3\) In this chapter, we first reexamine the fundamentals of the PCC formulations and propose an adjustment that is cleaner and more pervasive in controlling processes subject to both random shifts and drifts, and takes into account process age. We then study the application of this adjusted algorithm to the control of the CMP process. We refer to this adjusted PCC scheme as an age-based double EWMA (d-EWMA) scheme. As shown throughout this book (see, for example the Introduction), the CMP process is an ideal candidate for R2R control and therefore represents an ideal candidate for demonstrating the benefits of the d-EWMA scheme.

This chapter is organized as follows. In the remainder of this section we provide background information on the EWMA statistic. In Section 17.2 we examine the fundamentals in the PCC formula and propose an adjustment that is cleaner and more pervasive. The adjusted PCC formula is then further refined in Section 17.3 to take the process age into consideration. The methodology is illustrated in Section 17.4 using the CMP process as a case study.

17.1.1 BACKGROUND: THE EWMA STATISTIC

The use of exponentially weighted moving average (EWMA) statistic for estimating process deviations has been widely studied and adopted in practice. Box and Jenkins\(^4\) show that a controller based on the EWMA statistic is a minimum mean square error (MMSE) controller when the underlying process disturbance follows the IMA(1,1) (first-order integrated moving average) process. In practice, the EWMA statistic has
been shown to be quite effective, even for processes subject to disturbances other than the IMA process. In particular, applications in the semiconductor process industry, known as R2R process control, have shown that the EWMA statistic is also capable of bringing processes with linear drift under control.\textsuperscript{1,5-7} Suppose that the process output \( Y_t \) can be controlled linearly by an input variable \( X_t \) and is subject to a natural process disturbance \( \varepsilon_t \, (\sim \mathcal{N}(0, \sigma^2)) \) and a systematic disturbance \( \delta_t \). The process model can be written as:

\[
Y_t = \alpha + \beta X_t + \varepsilon_t + \delta_t,
\]

where \( \alpha \) is the linear process model’s intercept term and \( \beta \) is the system gain that translates the input variable’s size to the process output’s responded size. The EWMA statistic is then used to estimate the size of the process’s intercept (\( \alpha \)) plus its systematic deviation at time \( t + 1 \) (\( \delta_{t+1} \)):

\[
a_t = w(Y_t - \beta X_t) + (1 - w)a_{t-1}
\]

\[
= \sum_{j=1}^{t} \left[ w(1 - w)^{t-j} (Y_j - \beta X_j) \right]
\]

where the weight \( w \) is usually set between 0 and 1 and \( a_t \) is an exponentially weighted average of the historic deviations \( \alpha + \delta_j \)'s. Therefore, the process output at time \( t + 1 \) is estimated to be

\[
\hat{Y}_{t+1} = a_t + \beta X_{t+1}.
\]

In order to keep the process output at a predetermined target level (\( T \)), we obtain the process recipe at time \( t + 1 \):

\[
X_{t+1} = \frac{T - a_t}{\beta}.
\]

Such a controller is insufficient for processes subject to systematic wear-out. Several authors\textsuperscript{2,3} have addressed this problem and propose using two EWMA formulas: one for estimating “step-change” deviation and the other for estimating the process “drift” speed.

\[
a_t = w_1(Y_t - \beta X_t) + (1 - w_1)a_{t-1}
\]

\[
p_t = w_2(Y_t - \beta X_t - a_{t-1}) + (1 - w_2)p_{t-1}
\]
where \( w_1 \) and \( w_2 \) are the weights for the first and second EWMA equations, respectively, and \( p_t \) is to estimate the size of the process drift from \( t \) to \( t + 1 \). Thus, the process recipe at \( t + 1 \) should be

\[
X_{t+1} = \frac{T - (a_t + p_t)}{\beta}
\]  

(6)

Such a control scheme is referred to as predictor corrector control (PCC) scheme, termed by Butler and Stefani.©

17.2 PCC AND DOUBLE EWMA FORMULA

The PCC formula in (5) has two purposes: one is to estimate the intercept plus the systematic deviation \( (a_t) \) and the other is to estimate the process drift speed \( (p_t) \). Suppose that the process is subject to a linear drift only. That is,

\[
Y_t = \alpha + \beta X_t + \varepsilon_t + c\sigma t
\]

\[
= (\alpha + c\sigma t) + \beta X_t + \varepsilon_t
\]  

(7)

where the process output is systematically drifting away by a size of \( c\sigma \) per unit time. To see how the estimates \( a_t \) and \( p_t \) work, we can examine the process’s steady state as time approaches infinity (Appendix A):

\[
\lim_{t \to \infty} E(a_t) = \alpha - \frac{c\sigma}{w_1} + c\sigma(t + 1)
\]  

(8)

\[
\lim_{t \to \infty} E\left( p_t = \frac{c\sigma}{w_1} \right)
\]

This result somehow surprises us, since in this PCC formula \( a_t \) is not really an estimate for \( \alpha + \delta_{t+1} \) and \( p_t \) is not really the estimate for the drifting speed.

The steady-state recipe becomes

\[
\lim_{t \to \infty} X_{t+1} = \frac{T - \left[ \alpha - \frac{c\sigma}{w_1} + c\sigma(t + 1) + \frac{c\sigma}{w_1} \right]}{\beta} = \frac{T - [\alpha + c\sigma(t + 1)]}{\beta}
\]  

(9)

The expected process output at time \( t + 1 \) is then

\[
\lim_{t \to \infty} E(Y_t) = T
\]  

(10)
Here, we show that the system with PCC controller is a stable system and the expected process output converges to the desired target.

The estimate for $\alpha + \delta_t$ in the PCC formula can be further rewritten as (Appendix B)

$$
a_i + p_i = (w_1 + w_2 - w_1 w_2) \sum_{i=1}^{t} e_i + w_1 w_2 \sum_{i=1}^{t} \sum_{j=1}^{i} e_j + a_0
$$

(11)

$$
= w_1 \sum_{i=1}^{t} e_i + w_2 \sum_{i=1}^{t} \sum_{j=1}^{i} e_j + a_0
$$

where $e_j = Y_j - T$. This is, in effect, an integral-double-integral (I-II) controller, in contrast to the proportional integral derivative (PID) controller. In this I-II controller, the control action is proportional to the summation of the output errors and to the summation of summations of the output errors. The I-II controller can shown to be an MMSE controller for processes subject to IMA(2,2) disturbances. In (11) we can observe that the controller’s integral constant is $w_1 = w_1 + w_2 - w_1 w_2$ and the double-integral constant is $w_2 = w_1 w_2$. We define a control space of the I-II controller as the space filled up by all possible settings of control constants $(w_1, w_2)$, as shown in Figure 17.1. In Figure 17.1, the controller allows both $w_1$ and $w_2$ to be set between 0 and 1. For PCC controller, we usually set $w_1$ and $w_2$ between 0 and 1 as well. This, however, leads to a limited control region in the $(w_1, w_2)$ control space, as in the shaded area shown in Figure 17.1.

Now we would like to propose an adjustment of the PCC formula such that the control region can be extended to the entire space and the steady-state estimates are cleaner. We refer to this adjusted PCC formula as double EWMA (d-EWMA) formula as distinct from the original PCC formula. This d-EWMA formula is

$$
a_i = w_1 (Y_i - \beta X_i) + (1 - w_1) (a_{i-1} + p_{i-1})
$$

$$
p_i = w_2 (Y_i - \beta X_i - a_{i-1}) + (1 - w_2) p_{i-1}
$$

(12)
As can be observed in (12), the only difference from the original PCC formula is that in the first EWMA formula for \( a_t \), we add \( p_{t-1} \) into the formula. As a result, the I-II controller becomes (Appendix C)

\[
a_t + p_t = w_1 \sum_{i=1}^{t} e_i + w_2 \sum_{i=1}^{t} \sum_{j=1}^{i} e_j + (t+1)p_0 + a_0
\]  

(13)

That is, \( w_1 = w_1 \) and \( w_\text{II} = w_2 \) and with \( 0 < w_1 < 1 \) and \( 0 < w_2 < 1 \) we will have an I-II controller that fills up the entire control space.

With this d-EWMA formula, we can also obtain a cleaner steady state:

\[
\lim_{t \to \infty} E(a_t) = \alpha + c\sigma t
\]

\[
\lim_{t \to \infty} E(p_t) = c\sigma
\]

(14)

This result is much cleaner than the original PCC formula. Moreover, the two estimates \( a_t \) and \( p_t \) here have very clearly defined meanings. In this d-EWMA formula, \( a_t \) represents an estimate for \( \alpha + \delta_t \) and \( p_t \) an estimate for the size of the process drift from \( t \) to \( t+1 \). Together \( a_t + p_t \) is the estimate for \( \alpha + \delta_{t+1} \). The steady-state recipe at time \( t+1 \) becomes

\[
\lim_{t \to \infty} X_{t+1} = \frac{T - (a_t + p_t)}{\beta} = \frac{T - \alpha - c\sigma (t+1)}{\beta}
\]

This is exactly the same as (9) and thus the process output with d-EWMA control will also converge to the desired target \( T \). Though both controllers (PCC and d-EWMA) are “unbiased” controllers, the advantages of d-EWMA controller over the PCC controller are twofold:

1. The d-EWMA controller is a direct form of I-II controller and its control space fills up the entire I-II control space.
2. Unlike the PCC controller, \( a_t \) and \( p_t \) in the d-EWMA formula have clear definitions.

To illustrate these two advantages, we simulate a random drift process where \( T = 3500, a_0 = \alpha = 3000, \beta = 10, \) and \( p_0 = 0 \). The process drift speed \( c \) is a random number over time, i.e.,

\[
c_t = \begin{cases} 
c_{t-1} & \text{with } p = 0.99 \\
\sim N(0.02, 0.05) & \text{with } p = 0.01
\end{cases}
\]

A typical process with such random drift disturbances is shown in Figure 17.2.
To evaluate the control efficiency, we use a normalized mean squared error (MSE/σ²) as the performance measure. MSE is defined as

\[ MSE = \frac{\sum_{i=1}^{n} (Y_i - T)^2}{n} \]

And MSE/σ² is a measure normalized against the variance (σ²) of the natural disturbance (εᵣ). Figure 17.3 shows the contour plot for the control efficiency over the control space. As can be seen, the optimal I-II control setting \( w_I \cong 0.115 \) and \( w_{II} \cong 0.005 \) falls outside the control region (shaded area) of the original PCC controller. That is, if we use the PCC controller and restrict the values of \( w_1 \) and \( w_2 \) in the interval of (0, 1), the PCC controller would never be as effective as an optimal I-II controller (or d-EWMA controller).

FIGURE 17.2 A typical random drift process.
17.3 AGE-BASED DOUBLE EWMA CONTROLLER

In many applications, the data sampling time is not equally spaced. This leads to an invalid result in the d-EWMA formula. However, the data collected often comes with the process’s age at the time of sampling. In this section we will develop an age-based d-EWMA formula in which the unequally spaced data can be accommodated along with the age data.

First, let \( t_i \) denote the process age at the \( i \)th sampling. In Eq. (12), the two EWMA equations need to be modified to take the age into consideration. The term \( a_i \) now becomes the estimate for the intercept \((\alpha)\) plus the systematic deviation at the \( i \)th sample \((\delta_i)\). The first EWMA equation in (12) should be modified to

\[
a_i = w_i (Y_i - \beta X_i) + (1 - w_i)\left[a_{i-1} + (t_i - t_{i-1})p_{i-1}\right]
\]

\(15\)
where \( p_{i-1} \) is an estimate for the drifting size per unit time at the \((i - 1)\)th sample. From sample \( i - 1 \) to sample \( i \) the process has grown \((t_i - t_{i-1})\) older. Since the process continues to drift away at a rate of \( p_{i-1} \) between the two samples, by the \( i \)th sample the process has further deviated for an amount of \((t_i - t_{i-1})p_{i-1} \). This explains the second term of the EWMA formula in (15). We also need to modify the second EWMA formula for estimating the drifting speed at the \( i \)th sample \((p_i)\):

\[
p_i = w_2 \left( \frac{Y_i - \beta X_i - a_{i-1}}{t_{i-1} - t_i} \right) + (1 - w_2)p_{i-1}
\]

(16)

where \( Y_i - \beta X_i - a_{i-1} \) represents the amount the process drifts away between sample \( i - 1 \) and sample \( i \). The recipe at the \((i + 1)\)th run should be therefore set at

\[
X_{i+1} = \frac{T - \left[ a_i + (t_{i+1} - t_i)p_i \right]}{\beta}
\]

(17)

to keep the process output on target.

It should be noted that such a modification to accommodate the process age is only possible for the d-EWMA formula. For the original PCC formula, the modification becomes almost impossible because of its unclear definitions of \( a_i \) and \( p_t \).

## 17.4 APPLICATION TO CMP PROCESS WITH AGING PAD AND DISC

A simple strategy for controlling the CMP process is to predict the R2R process removal rate and then adjust the polishing time based on the prediction.\(^5\,8\,9\) EWMA and PCC (or double EWMA) techniques are the two most often used prediction techniques. Based on the age-based d-EWMA controller presented earlier, we will design an age-based R2R prediction technique that takes into account the ages of the abrasive pad and brushing disc. We will show how this proposed technique could improve the prediction capability and, thus, the control efficiency through actual CMP production data. Figure 17.4 shows a typical CMP process.

In a CMP process, both the abrasive pad and the brushing disc are wearing out quickly. Because of the combination of chemical and mechanical processes during polishing, the wear-out process becomes quite irregular. Simple EWMA prediction of the removal rate is not sufficient. d-EWMA is therefore needed to capture the changes of the removal rate. The d-EWMA prediction of removal rate at \((i + 1)\)th observation \((\hat{R}_{i+1})\) can be expressed as

\[
a_i = w_1 R_i + (1 - w_1) \hat{R}_i
\]

(18)

\[
p_i = w_2 (R_i - a_{i-1}) + (1 - w_2)p_{i-1}
\]

(19)
These are d-EWMA prediction equations corresponding to Eq. (12) without considering the process age. Similar to the d-EWMA controller, the first EWMA Eq. (18) is to estimate the level of the removal rate ($a_i$) and the second EWMA Eq. (19) is to capture its changing speed ($p_i$). In Eq. (20), the removal rate at observation $i + 1$ is then predicted by adding together the level estimate $a_i$ and the anticipated change ($p_i$) from observation $i$ to $i + 1$.

In practice, the age of the abrasive pad and the brushing disc can be acquired along with the removal rate data. Figure 17.5 shows the trend of the removal rate over a disc’s lifetime.

\[
\hat{R}_{i+1} = a_i + p_i
\]
over an entire lifetime of one disc. During the lifetime of a disc, preventive maintenance (PM) was performed seven times. Each time the abrasive pad was replaced with a new one.

It can be observed that the removal rate is changing in different speeds during a disc’s lifetime. In the beginning of a disc (e.g., the first pad), the removal rate seems to fall drastically. However, the drop of the removal rate becomes less as the disc gets older. Observing the last two pads in Figure 17.5, the removal rate becomes a constant during these two pads. This observation helps us understand how the ages of pad and disc affect the removal rate. The removal rate basically falls as the pad gets older. And the falling speed of the removal rate during each pad is again affected by the disc’s age. The older the disc, the less the removal rate falls. Based on this observation, we can design an age-based d-EWMA prediction scheme, similar to Eqs. (15) to (17), that takes into consideration the age of the pad and the age of the disc.

First, the initial changing speed of the removal rate \( p_0 \) for each abrasive pad is estimated based on the age \( \tau \) of the disc:

\[
p_0(\tau) = d + b\tau, \tag{21}
\]

where \( d \) is the changing speed of the removal rate for a brand-new disc \( (\tau = 0) \) and \( b \) represents how the changing speed changes as the disc ages.

Second, given the initial changing speed of the removal rate for each new pad, the formula in Eqs. (18) to (20) is then revised to accommodate the age \( t \) of the pad. Let \( t_i \) denote the age of the pad at the \( i \)th observation. We propose the following age-based d-EWMA prediction scheme:

\[
a_i = w_1 R_i + (1 - w_1) \hat{R}_i \tag{22}
\]

\[
p_i = w_2 \left( \frac{R_i - a_{i-1}}{t_i - t_{i-1}} \right) + (1 - w_2)p_{i-1} \tag{23}
\]

\[
\hat{R}_{i+1} = a_i + (t_{i+1} - t_i)p_i \tag{24}
\]

The data shown in Figure 17.5 is then used to estimate the model parameters \( (d, b, w_1, \text{ and } w_2) \) in Eqs. (21) to (24). Figure 17.6 shows this estimated model. The model estimated using data in Figure 17.5 can be now be used to predict the removal rate for other discs. Figure 17.7 shows the raw removal rate data for four discs. Figure 17.8 shows the prediction for Disc 3 using the model estimated from Disc 1.

We summarize the performance of EWMA, d-EWMA, and age-based d-EWMA prediction schemes by comparing their prediction mean squared error (MSE) in Table 17.1.

It is recommended that when the age data for the abrasive pad and the brushing disc are available, the proposed age-based d-EWMA scheme should be used to
predict the removal rate. The prediction improvement is about 17% better than the EWMA scheme, and 11% better than the d-EWMA scheme.

17.5 CONCLUSIONS

In this chapter we presented an adjustment of the original PCC controller. This adjustment has a cleaner form and was shown to be more pervasive in the form of
the I-II controller. We refer to this adjusted PCC controller as a d-EWMA controller. This cleaner form of d-EWMA controller enabled us to accommodate the process age into the formula. An age-based d-EWMA controller was therefore developed. The performance of this age-based d-EWMA controller was then illustrated through the example of CMP removal rate prediction. The results show that the age-based d-EWMA controller is indeed the most effective.

ACKNOWLEDGMENTS

We would like to thank Mr. J.-J. Chen, Mr. C.-P. Tung, Mr. Y.-L. Chou, and Ms. C.-L. Lin for their help in preparing the figures. We also like to thank Dr. Jowei Dun and Mr. S.-A. Wu of TSMC who provided the precious CMP data for this study.
REFERENCES


APPENDIX A

PCC controller:

\[ a_t = w_1(Y_t - bX_t) + (1 - w_1)a_{t-1} = w_1(\alpha + c\sigma t + \varepsilon_t) + (1 - w_1)a_{t-1} \]

\[ p_t = w_2(Y_t - bX_t - a_{t-1}) + (1 - w_2)p_{t-1} = w_2(\alpha + c\sigma t + \varepsilon_t - a_{t-1}) + (1 - w_2)p_{t-1} \]

Using matrix expression, we obtain

\[
\begin{bmatrix}
a_t \\ p_t
\end{bmatrix} =
\begin{bmatrix}
1 - w_1 & 0 \\
-w_2 & 1 - w_2
\end{bmatrix}
\begin{bmatrix}
a_{t-1} \\ p_{t-1}
\end{bmatrix} +
\begin{bmatrix}
w_1\alpha + w_1\varepsilon_t + w_1c\sigma t \\ w_2\alpha + w_2\varepsilon_t + w_2c\sigma t
\end{bmatrix}.
\]

Let \( U(t) = \begin{bmatrix} a_t \\ p_t \end{bmatrix} \)

\( A = \begin{bmatrix} 1 - w_1 & 0 \\ -w_2 & 1 - w_2 \end{bmatrix} \)

\( B(t) = \begin{bmatrix} w_1\alpha + w_1\varepsilon_t + w_1c\sigma t \\ w_2\alpha + w_2\varepsilon_t + w_2c\sigma t \end{bmatrix} \)

Then, \( U(0) = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \)

\( A = \begin{bmatrix} 1 - w_1 & 0 \\ 0 & 1 - w_2 \end{bmatrix} \)

\( \frac{1}{w_2 - w_1} \begin{bmatrix} 1 - w_1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 - w_2 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \)

We obtain

\[
A^t =
\begin{bmatrix}
\frac{1}{w_2 - w_1} & 0 \\
(1 - w_1)' & 0
\end{bmatrix}
\begin{bmatrix}
0 \\
1 - w_2'
\end{bmatrix}
\begin{bmatrix}
\frac{1}{w_2} \\ (1 - w_2)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{1}{w_2 - w_1} & 0 \\
(1 - w_2)' - (1 - w_1)' & (1 - w_2)'
\end{bmatrix}
\]

and, thus

\[ U(t) = AU(t - 1) + B(t) \]

\[ = A[AU(t - 2) + B(t - 1)] + B(t) \]

\[ = A^tU(0) + \sum_{i=1}^{t} A^{t-i}B(i) \]

\[ = \begin{bmatrix}
\frac{1}{w_2 - w_1} & 0 \\
(1 - w_2)' - (1 - w_1)' & (1 - w_2)'
\end{bmatrix}
\begin{bmatrix}
\alpha \\ 0
\end{bmatrix} \]
\[
\sum_{i=1}^{t} \left[ \frac{w_2}{w_2 - w_1} \left( (1 - w_1)^{i-1} \right) \right] + 0 \left( 1 - w_2 \right)^{i-1} \\
\begin{bmatrix}
w_1 \alpha + w_1 \varepsilon_i + w_1 c \sigma i \\
w_2 \alpha + w_2 \varepsilon_i + w_2 c \sigma i 
\end{bmatrix}
\]

Finally,

\[
a_t = \alpha (1 - w_1) + \sum_{i=1}^{t} (1 - w_1)^{i-1} (w_1 \alpha + w_1 \varepsilon_i + w_1 c \sigma i) \\
p_t = \frac{\alpha w_2}{w_2 - w_1} \left( (1 - w_2) - (1 - w_1) \right) \\
\sum_{i=1}^{t} \left[ \frac{w_2}{w_2 - w_1} \left( (1 - w_1)^{i-1} \right) \right] + \left( 1 - w_2 \right)^{i-1} (w_2 \alpha + w_2 \varepsilon_i + w_2 c \sigma i)
\]

As \( t \) approaches infinity,

\[
\lim_{t \to \infty} E(a_t) = \alpha - \frac{c \sigma}{w_1} + c \sigma (t + 1) \\
\lim_{t \to \infty} E(p_t) = \frac{c \sigma}{w_1}
\]

**APPENDIX B**

\[
a_t = w_1 (e_t + p_{t-1}) + a_{t-1} = w_1 \sum_{i=1}^{t} e_i + w_1 \sum_{i=0}^{t-1} p_i + a_0 \\
p_t = w_2 e_i + p_{t-1} = w_2 \sum_{i=1}^{t} e_i + p_0 \\
a_t + p_t = (w_1 + w_2) \sum_{i=1}^{t} e_i + w_1 \sum_{i=0}^{t-1} p_i + a_0 + p_0 \\
= (w_1 + w_2) \sum_{i=1}^{t} e_i + w_1 \sum_{i=0}^{t-1} (w_2 \sum_{j=1}^{i} e_j + p_0) + a_0 + p_0
\]
\[(w_1 + w_2)\sum_{i=1}^{\infty} e_i + w_1 w_2 \sum_{i=0}^{\infty} \sum_{j=1}^{i} e_j + (w_1 t + 1) p_0 + a_0\]

\[(w_1 + w_2 - w_1 w_2)\sum_{i=1}^{\infty} e_i + w_1 w_2 \sum_{i=0}^{\infty} \sum_{j=1}^{i} e_j + (w_1 t + 1) p_0 + a_0\]

**APPENDIX C**

\[a_i = w_1 e_i + p_{i-1} + a_{i-1} = w_1 \sum_{i=1}^{\infty} e_i + \sum_{i=0}^{\infty} p_i + a_0\]

\[p_i = w_2 e_i + p_{i-1} = w_2 \sum_{i=1}^{\infty} e_i + p_0\]

\[a_i + p_i = (w_1 + w_2) \sum_{i=1}^{\infty} e_i + \sum_{i=0}^{\infty} p_i + p_0 + a_0\]

\[= (w_1 + w_2) \sum_{i=1}^{\infty} e_i + \sum_{i=0}^{\infty} \left( w_2 \sum_{i=1}^{\infty} e_i + p_0 \right) + a_0 + p_0\]

\[= (w_1 + w_2) \sum_{i=1}^{\infty} e_i + w_2 \sum_{i=0}^{\infty} \sum_{j=1}^{i} e_j + (t + 1) p_0 + a_0\]

\[= w_1 \sum_{i=1}^{\infty} e_i + w_2 \sum_{i=0}^{\infty} \sum_{j=1}^{i} e_j + (t + 1) p_0 + a_0\]
Part 6

Advanced Topics

In the Introduction section of this book we provided a chronology of the development of R2R control. The information presented in Parts 1 through 5 provides a snapshot of the current state of the art of the many aspects of R2R control. While the case studies and results of deployment presented throughout the book (notably Chapters 11, 13, and 15 to 17) confirm that R2R control technology is indeed a mature capability that is ready for industry-wide deployment, there are numerous areas where additional research and development could lead to new and better solutions. Part 6 of this book is devoted to selected advanced topics in R2R control.

As is illustrated in many of the examples in this book, the CMP process appears to have benefited the most from R2R control. In Chapter 18, advances in R2R control of the CMP (chemical mechanical planarization) process are described; these enhancements to R2R control tools will pave the way to more effective and economical CMP R2R control solutions. Solutions that are described in Chapter 18 include (1) a “New Pad” feature that provides for the enhancement of the R2R control algorithm to accommodate process shifts due to pad replacement, (2) a methodology for utilizing in situ endpoint data for R2R process uniformity control, and (3) a solution for utilizing CMP R2R control as part of a multiprocess control strategy.

The enhancements to the EWMA algorithms that are described in Chapters 16 and 17 illustrate the benefit that can still be achieved from improving algorithms to address practical issues. This concept is extended in Chapter 19, where an enhanced EWMA controller is presented that has the capability of choosing the control parameter dynamically in response to the underlying process disturbances. There are two modules in this controller, namely the dynamic-tuning loop trigger module and the run-by-run feedback control module. In the dynamic-tuning loop trigger module, two EWMA control charts are used sequentially to determine if there is a large or
medium shift in the process output, and to trigger a new dynamic-tuning loop accordingly. In the run-by-run feedback control module, the control parameter and control model are retuned sequentially and a new process recipe is generated to compensate for the process output’s deviation from the target. Simulation results validate that the enhanced EWMA controller is superior to the traditional EWMA controller with fixed control parameters.

In the Introduction section, and again in Chapter 7, we noted that R2R control is really just one (and the first) component in an envisioned multilevel hierarchical control strategy. In Chapter 20 this idea is explored in detail and an interprocess control solution is presented that incorporates R2R control at the lower control levels to provide a factory-wide strategy for control. The solution, called the “active controller,” has the required properties of being generic, portable, and configurable. Many of these qualities arose from adapting design requirements and solutions that have been utilized for R2R control (see Chapters 8 and 9). Chapter 20 thus provides a vision for a multilevel, total factory control solution.
Advancements in Chemical Mechanical Planarization Process Automation and Control

James Moyne

18.1 INTRODUCTION

Maintaining acceptable yields in the semiconductor manufacturing and display industries requires constant attention to the state of the art in process tools, process chemistries and physics, and techniques for processing and process improvement. As feature sizes shrink and wafer sizes increase, the industry must continually innovate to maintain acceptable product yield, throughput, and overall equipment effectiveness (OEE). Some manufacturing capability attributes, such as nonproduct (NP) wafer usage and wafer scrap, must actually be improved in the transition to larger wafer sizes because of the increased cost of 300-mm wafers. As an example, one user reported that a processed 300-mm wafer costs approximately $5000 USD, while a raw wafer cost $2500 to $2000.¹

A number of techniques, including improved equipment design and process innovation, continue to aid in this cost-effective transition to 300-mm wafers and smaller linewidth technologies. However, as detailed in the Introduction of this book, traditional avenues within this industry are no longer sufficient with the focus turning to process and equipment parameter sensing, process identifying, and dynamic process tuning to complement equipment and process improvements.¹ ²

The CMP process is an excellent case study of this movement toward process automation and control (as demonstrated throughout this book). CMP was a relatively late arrival to semiconductor and display device processing, but its use is highly motivated through its “requirement” for .35 mm and below processing.¹ ³ ⁴

Since the advent of the first CMP control solutions (see Introduction and Chapter 1), process automation and control of CMP has advanced in a number of directions, including (1) automated control, (2) multivariate control, (3) enhanced control techniques to address practical process limitations and to address specific cost issues such as NP wafer requirements, (4) control solutions that combine R2R control with endpoint techniques, and (5) R2R control solutions that are incorporated as part of a total factory solution. This chapter provides a summary of the recent advancements in process automation and control for CMP, focusing on model-based
R2R control solutions. Specifically, in the next section, a background of advancements in CMP process automation and control up to the current year is presented. While many of these advancements are considered mature and are available on some CMP tools today, it is important to note that research and development efforts may still be needed to apply these capabilities to other CMP tools. This background section is followed by a discussion of current efforts in CMP process automation and control. This chapter concludes with a brief summary and a discussion of potential future directions in CMP process automation and control.

18.2 BACKGROUND

As mentioned above, CMP became an ideal process candidate for enhancement through process automation and control because (1) it is clearly an important component of .35 mm and below processing; (2) the process is relatively new and not well understood, and process engineers are generally open to process improvement through automation and control; (3) in-line and (later) in situ metrology technology became available. Because of the generally prohibitive in situ sensing environment of CMP, many early sensing results in CMP were in ex situ (multipoint) thickness measurement. Thus, in attempting to “close the loop” and provide process control of CMP, R2R control solutions were pursued.

The typical application of R2R control to CMP is described in the Introduction as well as Chapter 11 and Chapter 15 of this text. The first reported results in CMP R2R control came out of a three-year project sponsored by SEMATECH, whose goal was to provide reusuable solutions for CMP R2R control. Significant results that came out of that work for a CMP process include (1) thickness control, (2) thickness plus uniformity (multivariate) control strategies, (3) demonstration of process capability (Cpk) improvement, and (4) demonstration of pad-life extension. Other significant results that came out of that work for process control solutions include (1) a configurable R2R process control solution; (2) a multivariate, first-order, nonlinear R2R control algorithm solution with exponential weighted moving average (EWMA) noise filtering; (3) R2R control algorithm enhancements to support process input weighting (based on tunability), output weighting (to achieve multi-parameter optimization functions), and input boundary and discretization conditions; (4) a comparative analysis of R2R control algorithms; and (5) mechanisms for R2R process control automation.

Building on the results of the SEMATECH effort, a tool supplier and control systems supplier continued the R2R control research effort and produced significant results of (1) thickness + uniformity (multivariate) control and Cpk improvements of over 50%, and (2) a fully automated R2R process control solution for CMP. The solution developed as well as results achieved are described in detail in Chapter 11. Other, more recent results have reported uniformity control of other CMP tool types. In each case, different parameters were identified and utilized

* Cpk, or “process capability,” is described in Chapter 11.
to control uniformity (see Table 18.1), suggesting that uniformity control solutions vary significantly between tool types.

Two fronts along which CMP R2R control development has been achieving significant milestones are automation and integration. The capabilities and characteristics of R2R systems that facilitate automation and ease of integration are described in detail in Part 3 of this book.

## 18.3 CURRENT EFFORTS IN CMP PROCESS AUTOMATION AND CONTROL

The results of development and deployment of R2R automation and control solutions for CMP over the past few years have demonstrated their effectiveness in reducing process variability, increasing process capability ($C_{pk}$), increasing life of consumables and decreasing process maintenance events, and reducing operator error through automation. These results combined with advancements in sensing, integration, and control technologies have allowed the CMP process automation and control field to branch out into a number of new directions. While the full benefit of these pioneering efforts has not yet been realized, they do provide insight into the further potential benefits of CMP process automation and control. In this section a few of these efforts are described and prospects for process benefit are discussed.

### 18.3.1 Algorithm Enhancement: “New Pad” Feature

Recently, significant results have been achieved in advancing the state of the art of practical R2R control algorithms solutions for semiconductor and display manufacturing. For example, higher-order modeling and control algorithms have been developed that incorporate a learning capability, thus reducing the need for process identification ahead of control system deployment. Solutions have been demonstrated that support changing metrology strategies with both pre- and postmetrology. Stability analysis of EWMA algorithms has allowed for better determination of range of operation. A multialgorithm approach to control has been developed to provide a wider range of controllability of systems. Interprocess feedforward and feedback techniques have been proposed to reduce R2R variability (see also Chapter 15).
Some of the recent algorithm enhancements reported have been focused on tailoring the control capability to the CMP process. For example, the R2R control solutions have recently been developed to be more robust to specific process shifts in CMP processing. Figure 18.1 provides an illustration of the benefit of this type of improvement. The “new pad” feature shown increases equipment effectiveness by quickly bringing the process within acceptable limits after a pad replacement event, thereby reducing the number of required NP wafers. The operation of an EWMA R2R controller in adjusting for pad replacement can be understood by examining the basic equation for R2R adjustment of CMP process time for the current run $t_{n+1}$, based on results for the previous run $n$ and starting thickness measured for the current run, $ST_{n+1}$ (see also Chapter 2). The removal rate, $RR_{n+1}$, is modeled as

$$RR_{n+1} = \alpha (RmvdT_n/t_n) + (1-\alpha)RR_n,$$

where $RmvdT_n$ is the thickness of the material removed with the previous polish, and $\alpha$ is an EWMA weighting factor ($0 \leq \alpha \leq 1$), the value of which is selected based on considerations of noise, drifts, shifts, and model error. The suggested time for the next run $t$ is then calculated as

$$t_{n+1} = \frac{(ST_{n+1} - \text{Target})}{RR_{n+1}}.$$

In adjusting the control parameters to a new pad event, it is important to note that the removal rate of a new pad on the first wafer, $RR_{np+1}$, cannot be accurately estimated from the removal rate from the previous run as shown in (1), because the main process trend of pad wear modeled with the old pad does not apply to the new pad.
pad. The best statistical estimate of $RR_{np0}$ is some function (e.g., moving average) of $RR_{np+1}$ determined from previous calculations of actual removal rate on first wafer after a pad replacement. Then, assuming that the difference between $RR_{np+1}$ for the current pad and $RR_{np0}$ is much larger than the measurement and process noise factors, the removal rate estimate can be quickly adjusted to the new pad by setting the EWMA coefficient $\alpha$ to 1 for the first run. Thus, from (1), we have

$$RR_{np+2} = RmvdT_{np+1}/t_{np+1}.$$  \hspace{1cm} (3)

Utilizing this strategy, the improvement in thickness control over a strategy that does not utilize pad change event knowledge is shown in Figure 18.1. Note that the new pad thickness removed “spike” is reduced. The magnitude of the spike that remains is a function of the difference between $RR_{np}$ for the current pad and $RR_{np0}$. Note also that a significant transient exists for a number of runs as the removal rate model settles to the actual removal rate. This transient can be significantly reduced through adjustment of the removal rate model estimate by a factor that captures the exponential-like decay of removal rate over the first few wafers after a pad replacement event. For example, the following equation set could be utilized in place of (1) and (3) to model removal rate for the first $i$ wafers after a pad replacement:

$$RR_{np+1} = RR_{np0}$$

$$RR_{np+2} = e^{-x} \left[ RmvdT_{np+1}/t_{np+1} \right]$$  \hspace{1cm} (4)

$$RR_{np+i} = e^{-x} \left\{ \left[ \alpha_{np} \left( RmvdT_{np+i-1}/t_{np+i-1} \right) \right] + \left[ \left( 1 - \alpha_{np} \right) RR_{np+i-1} \right] \right\}, \text{ for } i \geq 3,$$

where $\alpha_{np}$ is an EWMA weighting factor (which may differ from $\alpha$ in (1)) and $x$ is associated with the decay constant (in wafers) of the removal rate profile for a new pad.

The benefits of employing an effective new pad strategy are numerous, including reduction of NP (test) wafer requirements, reduction of length of PMs associated with new pad replacement and tool qualification, and increased overall equipment effectiveness (OEE). Key issues are repeatability of removal rate decay after pad replacement, determination of end of pad decay profile (e.g., appropriate run in which to switch from (4) to (1) for removal rate model). Note also that the “new pad” model enhancement feature may also be applicable to other semiconductor and display process modeling scenarios such as etch modeling after a clean operation, or sputter deposition modeling after replacement of a deposition source.

### 18.3.2 Combining In Situ Endpointing with R2R Uniformity Control

A common misconception in semiconductor and display process control is that *in situ* control is a replacement for R2R control. The truth is that *in situ* control and R2R control are complementary and will coexist in the factory of the future.\(^{10,22}\) One example of the complementary nature of this hierarchical control structure is the
Combination of endpointing and R2R control in CMP to better achieve process thickness and uniformity targets. The apparatus, shown in Figure 18.2, is applicable to both oxide and metal polishing, and can utilize motor current or optical endpointing along with R2R control to achieve both thickness and uniformity control. The system operates by utilizing the optical or motor current endpointing to achieve thickness targets. The R2R control system provides accurate estimates of endpoint times, thereby reducing the need for endpoint monitoring until very near endpoint, and reducing the chance for misreading endpoint traces (e.g., stopping on the wrong cycle of an optical interference endpoint signal).

More importantly, the R2R controller can be adapted to utilize endpoint trace data to determine uniformity characteristics of the polished wafer. For example, the slope of a motor current trace is related to the uniformity of the wafer. With respect to optical endpointing, if the laser monitors the wafer through a hole in the platen, it can be set to monitor a track across the wafer, thereby providing endpoint information at different points on the wafer; this data can then be used to determine a uniformity profile. Utilizing this uniformity information, the R2R controller can then suggest process input adjustments on a run-to-run basis to achieve uniformity targets. In many cases it may also be used to suggest optimal endpointing algorithms.

**18.4 MULTIZONE UNIFORMITY YIELD MODELING AND CONTROL**

Previous results in CMP uniformity control focus on controlling center-to-edge (CTE) nonuniformity to an “optimal” target of zero. Recent efforts in CMP process uniformity analysis and modeling, however, suggest that a model of CMP process uniformity can be broken into radial zones, and an optimal yield target is generally not a CTE value of zero. The analysis of the multizone uniformity yield problem and derivation of optimal yield targets is detailed in Chapter 15. Process modeling for control to yield targets as opposed to process targets is expected to be a future direction of semiconductor manufacturing R2R control.

**18.5 CMP R2R CONTROL AS PRECOMPENSATION STRATEGY**

Previous advancements in CMP process control have focused strictly on the CMP process and optimizing the CMP tool to remove evenly across the entire wafer to a
desired thickness. Recently, efforts have been focused on incorporating CMP process control as part of a factory-wide scheme to improve yield of the process at the postetch step. The target contact process line has a typical CVD, CMP, lithography, etch (RIE) sequence. The control solution is being developed in a number of phases, with the first phase focused on the development of CMP and RIE R2R process control solutions, with pre- and postprocess measurement utilized along with interprocess feedforward and feedback information flow between the two control solutions. For both control solutions, the process quality metrics being controlled are postprocess thickness and uniformity, with these metrics indirectly verified at postetch process through electrical testing.

In developing this multiprocess control scheme, preliminary results have shown that, while repeatable results on etch process uniformity were observed, satisfactory models for controlling etch uniformity could not be obtained (through design of experiments analysis). Thus the control scheme was modified with the RIE process R2R and feedforward to RIE control components eliminated. The resulting interprocess control solution operates in the following manner (see also Figure 18.3 and Reference 22):

1. Etch process uniformity is determined at infrequent intervals through pre- and postmeasurement analysis at the etch process.
2. An etch process nonuniformity metric is determined and fed back to the CMP R2R controller.
3. The CMP R2R controller utilizes pre- and postprocess metrology and provides for control of CMP process removal thickness and radial nonuniformity.
4. The CMP R2R control targets are adjusted to precompensate for the etch process nonuniformity and maximize postetch process yield.

The CMP R2R controller thus does not necessarily optimize the CMP process, but rather operates as part of a total factory solution to provide the best CMP process for that process line. Note that the impact of CMP process targets on overall line yield must be taken into consideration in determining CMP uniformity targets that produce the optimal overall line yield.
18.6 CONCLUSIONS AND FUTURE DIRECTIONS

The CMP process industry has benefited from significant advancements in CMP process automation and control over the past seven years. Conversely, CMP has served as an excellent case study and showcase for the development and demonstration of the capabilities of process automation and control in semiconductor and display manufacturing. Significant achievements in CMP process automation and control include thickness R2R control, thickness plus uniformity R2R strategies and (later) control, demonstration of process capability (Cpk) improvement, demonstration of pad-life extension, and R2R control with post- and premetrology capabilities. Results applicable to generic R2R control that resulted from efforts utilizing CMP as a process research vehicle include a configurable R2R process control solution; a multivariate, first-order, nonlinear R2R control algorithm solution with exponential weighted moving average (EWMA) noise filtering; R2R control algorithm enhancements to support process input weighting (based on tunability); output weighting (to achieve multiparameter optimization functions), and input boundary and discretization conditions; and mechanisms and solutions for R2R process control automation.

Over the past year CMP process automation and control research has yielded results and potential advancements in a number of new areas, including algorithm enhancements such as the “new pad” feature (to minimize NP wafer usage after PMs), solution designs that combine R2R uniformity control with endpoint technology (reducing the need for ex situ measurement), multizone uniformity modeling and yield control, and precompensation (interprocess) control targeting strategies for CMP for improved line yield.

Future process automation and control directions for CMP will be increasingly focused on CMP processing as part of a multilevel total factory solution. Thus the emphasis will be on optimizing the factory output as opposed to the CMP process output, and the choice of process control targets and metrology strategies will be made based on these factory-level directives. The CMP process control solution will have to be automated to meet increasing demands on throughput and to reduce operator error. The automated control solution will be multilevel with in situ control providing for endpointing and in situ parameter adjustment. The R2R controller will provide input to the in situ control system, especially to provide adjustment parameters for process shift events such as PMs and product shifts. The R2R controller will also control parameters that cannot (yet) be controlled at an in situ level, and will accommodate and utilize interprocess feedforward and feedback information to achieve factory-level control targets. R2R control solutions will also be enhanced accordingly to provide R2R plus interprocess control configurable algorithm solutions and enablers that can be integrated at the CMP equipment level or distributed over the factory computer integrated manufacturing system.

ACKNOWLEDGMENT

Portions reprinted with permission from proceedings of the Third International Symposium on Chemical Mechanical Polishing in IC Device Manufacturing: 196th Meeting of the Electrochemical Society.21
REFERENCES

19 An Enhanced EWMA Controller for Processes Subject to Random Disturbances

Ruey-Shan Guo, Argon Chen, and Jin-Jung Chen

19.1 INTRODUCTION

As explained throughout this book, semiconductor manufacturing processes are subject to small and large special disturbances such as process drifts or shifts. In many cases the causes of disturbances are known, but it is either impossible or too expensive to remove them. For example, variations of raw material quality may be difficult to reduce. Another example is the disturbance caused by the machine maintenance or changes in process settings. In such cases, when the resulting output deviations can be compensated by adjusting the processing recipe, process control techniques such as R2R control will be useful. Since the feedback control action is exercised after observing the process output on a run-by-run* basis, there is no input–output transient (dynamics) effect involved, as seen in the real-time control problems. Usually, a “continuous” run-by-run feedback control strategy is often adopted if the following conditions apply:

- The causes of variation are difficult to remove
- The adjustment of the process recipe is easy and fast
- The adjustment cost is relatively inexpensive
- The quality loss due to the output deviation from the target is significant.

The use of the exponentially weighted moving average process controller (EWMA controller, see Parts 1 and 2 of this book) for processes subject to small disturbances has been widely studied and adopted in practice. Box and Jenkins² proved that the EWMA controller is a minimum mean square error (MMSE) controller when the underlying process disturbance is an IMA(0,1) (first-order integrated moving average) process. In the semiconductor process industry, several applications

* “Run-by-run” and “run-to-run” (R2R) are used interchangeably in this chapter.
of the EWMA R2R controller are shown to be effective, even when the underlying disturbances don’t follow the IMA pattern but follow an approximately linear drift pattern.\textsuperscript{1,10,12} For example, the chemical mechanical polishing (CMP) process usually suffers from a drift in the removal rate due to the wear of the polishing pad. Such a phenomenon is also seen in the deposition rate of the metal sputtering process due to the consumption of the metal target. Other R2R control work can be found in References 8, 9, and 13–15.

Although effective in many applications, the EWMA controller still has some deficiencies. To be more specific:

- It is mostly used to compensate for small disturbances, and there is no control strategy focusing on random disturbances with different shift sizes.
- The control efficiency strongly depends on the choice of the control parameter, and in many cases it is determined empirically.
- There is no capability to change the control parameter dynamically in response to the change of the process disturbances.

To overcome these deficiencies, several approaches have been proposed in recent years. Sachs et al.\textsuperscript{10} proposed a Bayesian-based algorithm instead of the EWMA algorithm to compensate for large disturbances. Ingolfsson and Sachs\textsuperscript{6} analyzed the stability of the EWMA controller and provided a guideline for choosing the control parameter under different process drifts. Smith and Boning\textsuperscript{11} proposed a self-tuning EWMA controller for processes with drift. Del Castillo\textsuperscript{3,4} also presented a self-tuning controller that uses recursive least squares to continuously estimate the process parameters. He showed that the self-tuning controllers have many advantages over controllers with fixed control parameters. The controllers he proposed are based on a process model that follows linear stochastic processes.

In this chapter we focus on processes subject to random disturbances of different shift sizes, and propose an enhanced EWMA-based control algorithm to address the control of this class of processes. Specifically, the goals of this chapter are to

1. Present a straightforward control strategy for practical applications.
2. Extend the EWMA controller to compensate for process disturbances with different shift sizes.
3. Provide the capability to dynamically change control parameters in response to disturbance changes.

In the following sections, we first overview the system architectures for both the EWMA controller and the enhanced EWMA controller. The two main modules of the enhanced controller, the Dynamic-Tuning Loop Trigger module and the R2R Feedback Control module, are then detailed in Sections 19.3 and 19.4, respectively. We then use simulations to validate the advantage of the enhanced EWMA controller over the conventional EWMA controller in Section 19.5. Conclusions are presented in Section 19.6.
19.2 SYSTEM ARCHITECTURE

19.2.1 EWMA CONTROLLER

To better explain the enhanced EWMA controller, we need to review the control algorithm of the current EWMA controller first (see Chapter 3 for an in-depth discussion). Assume that a typical process during production stage is already optimized through design of experiments. In this case, the input–output process behavior can be approximated by a linear model as

\[ Y_t = \alpha + \beta X_t + \epsilon_t \]  

(1)

where
- \( Y_t \) = output measurement (\( t \) is the current run no.)
- \( \alpha \) = process model’s constant term
- \( \beta \) = process model’s slope term
- \( X_t \) = equipment settings
- \( \epsilon_t \) = white noise \( \epsilon_t \sim N(0, \sigma^2) \)

If there are special process disturbances such as process drifts and process shifts (in addition to the underlying white noises) superimposed onto the process, the process model will become

\[ Y_t = \alpha + \beta X_t + \epsilon_t + \delta_t \]

(2)

where \( \delta_t \) are the special process disturbances.

To reduce the impact of these special process disturbances on the output value, an EWMA controller is used. Figure 19.1 illustrates the system architecture of a current EWMA controller, and its main concepts are briefly explained below.

1. Assume that the underlying process model in (1) is estimated by a control model with the following form:

\[ \hat{Y}_t = a_{t-1} + bX_t \]

(3)

where
- \( \hat{Y}_t \) = predicted output value
- \( a_{t-1} \) = control model’s constant term
- \( b \) = control model’s slope term (assumed = \( \beta \) and is fixed for all runs)

If we compare (2) with (3), we have \( a \equiv \alpha + \delta \), providing that \( \beta \) is correctly estimated by \( b \) through design of experiments. This means that the special process disturbances result in the changes of the constant term of the control model.
2. Collect the output measurement $Y_t$ after one run and calculate the residual between the measured output value and the predicted value:

$$e_t = Y_t - \hat{Y}_t$$

(4)

3. Fine-tune the constant term of the control model using the calculated residual. To estimate its value more accurately, the constant term of the control model is updated using the EWMA algorithm. Notice that $W$ is the control parameter to decide the weighting schemes on the historical data and $a_0 = \alpha$.

$$a_t = W e_t + a_{t-1}$$

$$= W(Y_t - bX_t) + (1 - W)a_{t-1}$$

(5)

$$= \sum_{j=1}^{t} \left[ W(1-W)^{t-j} \left( Y_j - bX_j \right) \right] + (1-W)^{t} \alpha$$

4. Obtain the new control model for the next run.

$$\hat{Y}_{t+1} = a_t + bX_{t+1}$$

(6)
5. Obtain the equipment settings (recipe) for the next run.

\[ X_{t+1} = \frac{T - a}{b}, \text{ where } T \text{ is the target value} \]  

(7)

6. Repeat steps 2 to 5 on a run-by-run basis.

19.2.2 Enhanced EWMA Controller

Figure 19.2 illustrates the system architecture of the enhanced EWMA controller. As can be seen, there are two modules: the Dynamic-Tuning Loop Trigger module
and the Run-by-Run Feedback Control module. In the Dynamic-Tuning Loop Trigger module, control charts are first used to determine if there is a large or medium process shift. If there is a large or medium process shift and the error is controllable, then the control parameter \((W)\) must be reset to a higher value and starts a new dynamic-tuning loop. In the Run-by-Run Feedback Control module, the control parameter is tuned based on the current run’s sequence in the dynamic-tuning loop. Once the control parameter is retuned, the control model will be retuned accordingly. A new recipe is then generated so that the observed deviation can be compensated for in the next run of the process. If the generated recipe is not feasible in practice, we have to stop the process.

### 19.3 Dynamic-Tuning Loop Trigger Module

In our proposed enhanced EWMA controller, a baseline EWMA controller will be always utilized to compensate for smaller disturbances. In addition, large and medium disturbances will be dealt with through the controller’s dynamic-tuning mechanism. Thus, in the Dynamic-Tuning Trigger module, the goals are to detect if there is a large or medium process shift and to reset the value of the control parameter. Here, two EWMA control charts are used as the detection tools. As shown in Figure 19.2, the residual for the current run is first calculated and plotted on the EWMA control charts. If there is an out-of-control signal and the error is controllable, a dynamic-tuning loop is triggered.

In general, control charts are designed to detect a change as fast as possible while minimizing the false alarm rate. We will use the following criteria to judge the performance of the control charts:

- **Robustness**: long average run length (\(ARL_0\)) between false alarms. The larger the \(ARL_0\), the better the robustness.
- **Sensitivity**: short average run length (\(ARL_1\)) between the shift and the alarm. The smaller the \(ARL_1\), the better the sensitivity.

Typically, there are four types of control charts, and their difference can be characterized by the weighting scheme on the historical data (Figure 19.3).²

- **Shewhart control chart**: each control point uses only the current measurements, so a 100% weight is assigned to the current data.
- **CUSUM control chart**: each control point uses all the historical data by assigning equal weights to all historical data.
- **Moving average control chart**: each control point uses the latest \(n\) \((n = 4\) in the example) data points by assigning equal weights to the latest \(n\) points.
- **EWMA control chart**: each control point uses all the historical data, but the more recent the data, the higher the weight. As shown in (8), the weight is exponentially decreased as the data point ages.
For each of various control chart designs, tradeoffs have to be made between robustness and sensitivity. Since the Shewhart control chart is usually more sensitive to a large disturbance, but at the cost of a higher false alarm rate, we need to design our two EWMA control charts with better sensitivity and robustness. Table 19.1 shows the \( ARL_0 \) (the average run length when the shift size = 0) and \( ARL_1 \) (the average run length when the shift size > 0) for the Shewhart control chart and two EWMA control charts. Here the EWMA results are calculated by using the Markovian approximation.

As can be seen, the EWMA large control chart with \( W_l = 0.6 \) and control limit (CL) = 3.285\( \sigma \) (here, \( \sqrt{W_l/(2-W_l)} \)) is as good as the Shewhart control chart at detecting a large shift (3\( \sigma \)), but at a better robustness. The EWMA medium control chart with \( W_m = 0.33 \) and CL = 3.25\( \sigma_m \) (here, \( \sqrt{W_m/(2-W_m)} \)) is better than the Shewhart control chart at detecting a medium shift (2\( \sigma \)) in terms of sensitivity and robustness. Here the average run length for the EWMA large control chart is roughly equal to 2 after a 3\( \sigma \) shift \( (d = 2 \text{ in Figure 19.2}) \). The average run length for the EWMA medium control chart is roughly equal to 3.9 after a 2\( \sigma \) shift \( (d = 4 \text{ in Figure 19.2}) \).

19.4 RUN-BY-RUN FEEDBACK CONTROL MODULE

The control parameter \( W \) also determines the controller’s sensitivity and robustness. It is known that a larger value of \( W \) will increase the controller’s sensitivity to the process deviation and is able to capture the size of the deviation more quickly,
whereas a smaller $W$ helps stabilize the controller’s response to white noises. It seems reasonable to use a large value of the control parameter right after the detection of a large shift (for sensitivity purposes), and to decrease the values gradually afterward (for robustness purposes).

Now, suppose that a sudden shift (or a step change) occurs in the process ($\delta_t = \Delta_{t^*} = S \Delta_{t^*}$ in (2)). That is,

$$Y_t = \alpha + \beta X_t + S \Delta_{t^*} + \varepsilon_t$$

where $S =$ size of the process shift

$$\Delta_{t^*} = \begin{cases} 1 & \text{when } t \geq t^* \\ 0 & \text{when } T < t^* \end{cases}$$

Assume for now that the change point $t^*$ is known. In order to compensate for the sudden shift ($S$) quickly, $W$ of the EWMA controller in Eq. (5) should be set larger immediately after the change point to capture the shift size. However, the step change occurs only once and the process mean remains unchanged after $t^*$. If the value of $W$ remains large, the controller becomes oversensitive to the white noises. To overcome the trade-off between large and small $W$ and to design a more effective controller for processes subject to sudden shifts, we propose a controller with a dynamically adjusted control parameter:

$$a_t = W_t (Y_t - bX_t) + (1 - W_t) a_{t-1}$$

(10)

In Eq. (10), the value of the control parameter ($W_t$) is dynamically adjusted over time. The task now is to derive a method for adjusting $W_t$.

Here, we first consider the case where the change point is known and we do not have prior knowledge of how the process will be shifting away. In this case, we rely only on the sample data presently obtained from the process to estimate the shift.

### TABLE 19.1
Average Run Length for Shewhart and Two EWMA Control Charts

<table>
<thead>
<tr>
<th>Shift size</th>
<th>ARL</th>
<th>Shewhart: $W = 1$</th>
<th>EWMA large: $W_f = 0.6$</th>
<th>EWMA medium: $W_m = 0.33$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 \sigma$</td>
<td>$ARL_0$</td>
<td>370.40</td>
<td>1003.44</td>
<td>1007.44</td>
</tr>
<tr>
<td>$1 \sigma$</td>
<td>$ARL_1$</td>
<td>43.89</td>
<td>31.55</td>
<td>15.87</td>
</tr>
<tr>
<td>$2 \sigma$</td>
<td>$ARL_1$</td>
<td>6.30</td>
<td>4.55</td>
<td>3.90</td>
</tr>
<tr>
<td>$3 \sigma$</td>
<td>$ARL_1$</td>
<td>2.00</td>
<td>2.06</td>
<td>2.27</td>
</tr>
<tr>
<td>$4 \sigma$</td>
<td>$ARL_1$</td>
<td>1.19</td>
<td>1.36</td>
<td>1.67</td>
</tr>
</tbody>
</table>
size. In the appendix to this chapter, a theorem is provided to determine the optimum control parameter $W_t$, which minimizes the mean squared deviation (MSD$_{t+1}$) at run $t + 1$. MSD$_{t+1}$ is defined as follows:

\[
\text{MSD}_{t+1} = \left[ E(Y_{t+1} - T)^2 \right]
\] (11)

The following results for are is obtained:

\[
W_t^\text{opt} = \frac{1}{t - t^* + 1}
\] (12)

\[
\text{MSD}_{t+1}^\text{opt} = \left[ \frac{1}{t - t^* + 1} + 1 \right] \sigma^2
\] (13)

In Eq. (12), $W_t$ decreases over time and diminishes to zero. This, however, is not desirable, since very often the process is subject not only to shifts, but also to drifts and other smaller disturbances. Therefore, a minimum value of $W_t$ is required to keep the baseline EWMA controller working for compensating for such disturbances. The control equation becomes

\[
a_t = W_t(Y_t - bX_t) + (1 - W_t)a_{t-1} \quad \text{for } t \geq t^*
\] (14)

\[
W = \begin{cases} 
\max \left( W_t, W_b \right) & \text{if dynamic tuning loop is not completed} \\
W_b & \text{if dynamic tuning loop is completed}
\end{cases}
\] (15)

where $W_b$ is the required minimum value of the control parameter.

In reality, the change point $t^*$ is unknown. EWMA charts are therefore installed to detect the shifts. As mentioned earlier, two EWMA control charts are used in our approach: one for detecting a larger shift and the other for detecting a medium shift. However, the signaling of either control chart only takes place some time after the actual change point. The time delay can be roughly estimated by the average run length ($d$) of the control chart. Suppose now the control chart detects the shift at the $k$th run (Figure 19.4). Then, the change point can be estimated to be

\[
\hat{t}^* = k - d + 1
\] (16)

**FIGURE 19.4** The run length for the control chart to detect a process shift.
Equations (12), (14), and (15) can be then applied, followed by the recipe generation and model prediction as presented in Eqs. (6) and (7). The complete Run-by-Run Feedback Control module is shown in Figure 19.2.

From Eqs. (12) and (16), the control parameter of the enhanced EWMA controller has a starting value:

\[ W_k = \frac{1}{k - \hat{t}^* + 1} = \frac{1}{d} \]  \hspace{1cm} (17)

That is, the controller will take into consideration \( d \) data points prior to the detection point \( k \) while estimating \( a_t \). The control parameter continues to diminish over time until it reaches \( W_k \) (Figure 19.5) and is fixed at that minimum required level (0.1 in Figure 19.5) afterward. This control cycle holds on until one of the EWMA charts signals again. The alarm given by the control chart indicates the occurrence of a shift (large or medium) and triggers the need of adjusting the control parameter again to capture the step change. The dynamically adjusting scheme of the control parameter as described above is thus restarted, and Eqs. (4) and (12) are recalculated starting from the estimated change point \( \hat{t}^* \).

Figure 19.5 shows the changes of the control parameter over time. As can be seen, the value of \( W \) starts with 1/2 (triggered by the first EWMA chart for detecting a large shift) and decreases until it reaches a minimum value of 0.1. The cycle lasts until a new signal is given by the second control chart. The new signal triggers a new cycle of the dynamically adjusting \( W \). This time, it starts with 1/4 and again diminishes to 0.1.

19.5 SIMULATION VALIDATION

19.5.1 PROCESS DISTURBANCE MODELS

To validate the superior performance of the enhanced EWMA controller over the traditional EWMA controller with fixed control parameter, simulations are performed.
First we need to define the special process disturbances. Two types of disturbances are simulated, namely, process drifts and shifts, and modeled in Eq. (18):

\[ \delta_t = c\sigma t + \tau_t \]  

(18)

where \( c\sigma t \) is the process drift and \( \tau_t \) is the process shift. Assume, also, that the drift follows a linear drift pattern and deviates from the target at the speed of \( c\sigma \) (\( c \) is a constant and \( \sigma \) is the standard deviation of white noise) per unit time. As for the shift, it is modeled as a random shift with the following pattern:

\[
\tau_t = \begin{cases} 
\tau_{t-1} & \text{with probability } 1 - p \\
\tau_{t-1} + S & \text{with probability } p
\end{cases}
\]  

(19)

where \( p \) is the probability of the shift occurrence. The shift size \( S \) is a random variable and follows a normal distribution with mean \( \mu_S \) and variance \( \sigma_S^2 \), i.e., \( S \sim N(\mu_S, \sigma_S^2) \). To simplify the simulation, we further assume that \( \sigma_S = \sigma \), so \( S \sim N(\mu_S, \sigma^2) \).

### 19.5.2 A Simple Example

Next, a simple example is used to illustrate the dynamic-tuning feature in the presence of random disturbances. In this simulation example, a process is assumed to have the following model form:

\[
Y_t = \alpha + \beta X_t + \epsilon_t + \delta_t = 3000 + 10 X_t + \epsilon_t + \delta_t
\]  

(20)

\[ \epsilon_t \sim N(0, \sigma^2) \sim N(0, 50^2) \]  

(21)

\[ \delta_t = c\sigma t + \tau_t = 0.02(50)t + \tau_t \]  

(22)

\[
\tau_t = \begin{cases} 
\tau_{t-1} & \text{with probability } 1 - p \\
\tau_{t-1} + S & \text{with probability } p
\end{cases}
\]  

(23)

\[ p = 0.05, \quad S \sim N(3\sigma, \sigma^2) \sim N(150, 50^2) \]  

(24)

Let’s further assume the target output is 3500. Now the simulation is performed with the above parameters for 2000 runs in one simulation cycle. With different random numbers, the simulation is repeated 200 cycles. Results of the first 71 runs in one cycle are shown in Figure 19.6a,b,c,d. Notice that there are three shifts occurring during the first 71 runs. The shift sizes and the locations are 3.3\( \sigma \) at run 17, 3.1\( \sigma \) at run 45, and 2.2\( \sigma \) at run 62. In Figure 19.6b, the alarms given by the first EWMA control chart at the 18th run and the 46th run indicate the occurrence of large shifts and trigger the need to increase the value of the control parameter in
order to capture the step changes. In Figure 19.6c, the alarm given by the second EWMA control chart at the 65th run also indicates the occurrence of a medium shift. Figure 19.6d shows the changes of the control parameter over time. As can be seen, the values of $W$ reset with $1/2$ at the 18th run and 46th run and reset with $1/4$ at the 65th run. Then $W$ decreases until it reaches a minimum value $0.15$ ($W_b = 0.15$ in this case). As shown in Figure 19.6a, when a shift occurs the enhanced EWMA controller can estimate the change of the process mean value and compensate for the deviation more quickly than the EWMA controller with fixed control parameter.

19.6 MONTE CARLO SIMULATIONS

To validate the performance of the enhanced EWMA controller, Monte Carlo simulations are performed under various random disturbances. In these Monte Carlo simulations process models are assumed to be the same as in (20), (21), and (22). The superimposed random shifts in (23) are simulated in four cases, and their occurrence probability ($p$) and magnitude ($S$) are summarized in Table 19.2. For example, in the first case the random shift has a low probability of occurrence and smaller shift magnitude. In the fourth case, the random shift has a high probability of occurrence and larger shift magnitude.
In these simulations, we compare the performance of the enhanced EWMA controller with the EWMA controller with fixed control parameter under the four cases of disturbances. As for the control performance characterization, the normalized mean squared error is used as the performance index. The smaller the value, the better the performance. The normalized mean squared error is denoted as $\text{MSE}/\sigma^2$ and is defined as

$$\text{MSE}/\sigma^2 = \left\{ \frac{1}{n} \sum_{t=1}^{n} (Y_t - T)^2 \right\} / \sigma^2$$

Here the $\text{MSE}/\sigma^2$ index is calculated based on the simulation results of 2000 runs and 200 simulation cycles. Results of the study ($\text{MSE}/\sigma^2$ index against $W_b$) are

![Graph showing the comparison of performance between the enhanced EWMA controller and the EWMA controller with fixed control parameter under different cases of disturbances.](image)

**TABLE 19.2**

Four Simulated Cases in the Monte Carlo Simulations

<table>
<thead>
<tr>
<th>Case</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>0.005</td>
<td>0.005</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$S$</td>
<td>$N(0, \sigma^2)$</td>
<td>$N(3\sigma, \sigma^2)$</td>
<td>$N(0, \sigma^2)$</td>
<td>$N(3\sigma, \sigma^2)$</td>
</tr>
</tbody>
</table>

In these simulations, we compare the performance of the enhanced EWMA controller with the EWMA controller with fixed control parameter under the four cases of disturbances. As for the control performance characterization, the normalized mean squared error is used as the performance index. The smaller the value, the better the performance. The normalized mean squared error is denoted as $\text{MSE}/\sigma^2$ and is defined as

$$\text{MSE}/\sigma^2 = \left\{ \frac{1}{n} \sum_{t=1}^{n} (Y_t - T)^2 \right\} / \sigma^2$$

Here the $\text{MSE}/\sigma^2$ index is calculated based on the simulation results of 2000 runs and 200 simulation cycles. Results of the study ($\text{MSE}/\sigma^2$ index against $W_b$) are
shown in Figures 19.7a,b,c,d. We also plot the average of Figures 19.7a,b,c,d in Figure 19.7e. Figure 19.7e represents the situation where the disturbance information is unknown and all types of disturbances are possible.

Based on these simulation results, we have the following conclusions:

- When there are only small process drifts or small shifts, as happened in Cases 1 and 3, the baseline EWMA controller is very effective in controlling these small disturbances. As a result, the enhanced EWMA controller doesn’t trigger the dynamic-tuning capability and uses the minimum control parameter in most of the runs. Therefore, a similar control performance in Figures 19.7a and 19.7c is observed.
- When there are large process shifts, as happened in Cases 2 and 4, the enhanced EWMA controller is much better than the EWMA controller with fixed control parameter. As shown in Figures 19.7b and 19.7d, the performance of the EWMA controller with fixed control parameter highly depends on the choice of the control parameter. The range of the optimal control parameter is quite large, ranging from 0.3 in Case 2 to 0.6 in Case 4. The performance of the enhanced EWMA controller, on the other hand, is less dependent on the minimum control parameter.
- The superiority of the enhanced EWMA controller is validated by the better average performance as shown in Figure 19.7e.

19.7 CONCLUSIONS

In this chapter, an enhanced EWMA controller for processes subject to small and large random disturbances has been presented. The controller uses two EWMA control charts to trigger a dynamic-tuning loop and adjusts the control parameter in response to the disturbances. Through the simulation study, we reach the following conclusion:

- The enhanced EWMA controller is very effective in controlling processes subject to small and large random disturbances.
- The enhanced EWMA controller adjusts its control parameter dynamically in response to the process changes.
- The enhanced EWMA controller is better than the EWMA controller with fixed control parameters.

The enhanced controller presented thus represents an ideal extension to the EWMA solutions presented in Chapter 3 and utilized throughout this book (e.g., in Chapters 1, 6, 11, 13, and 15). Utilizing this extension, the domain of applicability of the EWMA solutions can be expanded and control can be provided for systems that are subject not only to well-behaved drift, but also larger random disturbances.
FIGURE 19.7 (a) Simulation results of Case 1. (b) Simulation results of Case 2. (c) Simulation results of Case 3. (d) Simulation results of Case 4. (e) Average results of Cases 1, 2, 3, and 4.
FIGURE 19.7 (continued)
APPENDIX

**Theorem:** Given a process model in Eq. (9), an EWMA controller with a dynamically adjusted control parameter in Eq. (10) minimizes MSD if \( W_t = \frac{1}{t - t^* + 1} \) for all \( t \geq t^* \) and the observation at the time \( t^* \) is known.

**Proof:** Given the control parameter \( W_t = \frac{1}{t - t^* + 1} \) for any \( t^* \leq t < \tau \) and assuming \( b = \beta \):

\[
a_t = W_t (Y_t - bX_t) + (1 - W_t) a_{t-1}
\]

\[
= W_t (\alpha + S + \varepsilon_t) + (1 - W_t) \left[ W_{t-1} (Y_{t-1} - bX_{t-1}) + (1 - W_{t-1}) a_{t-2} \right].
\]

Substituting \( W_{t-1} = \frac{1}{\tau - t^*} \) into the above equation, we obtain:

\[
a_t = W_t (\alpha + S + \varepsilon_t) + (1 - W_t) \left[ \frac{1}{\tau - t^*} (\alpha + S + \varepsilon_{t-1}) + W_{t-1} (Y_{t-1} - bX_{t-1}) + (1 - W_{t-1}) a_{t-2} \right].
\]

Again, substituting \( W_{t-2} = \frac{1}{\tau - t^* - 1} \) into the above equation, we obtain

\[
a_t = W_t (\alpha + S + \varepsilon_t) + (1 - W_t) \left[ \frac{1}{\tau - t^*} (\alpha + S + \varepsilon_{t-1}) + \frac{\tau - t^* - 1}{\tau - t^*} W_{t-1} (Y_{t-1} - bX_{t-1}) + \frac{\tau - t^* - 2}{\tau - t^* - 1} a_{t-3} \right].
\]
By recursively substituting $W_{\tau-3} = \frac{1}{\tau-t^* -2}, \ldots, W_{t^*} = 1$ into the equation, we finally have

$$a_t = \alpha + S + W_t \epsilon_t + (1 - W_t) \frac{1}{\tau-t^*} \sum_{i=t^*}^{\tau-1} \epsilon_i.$$

The recipe at $\tau + 1$ becomes

$$X_{\tau+1} = \frac{T - a_\tau}{b},$$

$$= \frac{t - \alpha - S - W_t \epsilon_t - (1 - W_t) \frac{1}{\tau-t^*} \sum_{i=t^*}^{\tau-1} \epsilon_i}{b}.$$

The process output and its MSD at $\tau + 1$ are

$$Y_{\tau+1} = \alpha + bX_{\tau+1} + S + \epsilon_{\tau+1} = T + \epsilon_{\tau+1} - W_t \epsilon_t - (1 - W_t) \frac{1}{\tau-t^*} \sum_{i=t^*}^{\tau-1} \epsilon_i$$

and

$$MSD_{\tau+1} = E\left[(Y_{\tau+1} - T)^2\right]$$

$$= \sigma^2 + W_t^2 \sigma^2 + (1 - W_t)^2 \frac{1}{\tau-t^*} \sigma^2.$$

By minimizing $MSD_{\tau+1}$ with respect to $W_t$, we obtain the optimal control parameter at $\tau$,

$$W_{\tau}^{opt} = \frac{1}{\tau-t^*+1}$$

and the optimal MSD,

$$MSD_{\tau+1}^{opt} = \left[\frac{1}{\tau-t^*+1} + 1\right] \sigma^2.$$

We have shown that given $W_t = \frac{1}{t-t^*+1}$ for $t^* \leq t < \tau$, $W_{\tau}^{opt} = \frac{1}{\tau-t^*+1}$. That is, if $W_{t^*}^{opt} = \frac{1}{t^*-t^*+1}$, then recursively we can obtain $W_{t^*}^{opt} = \frac{1}{t^*-t^*+1}$ for $t^* \leq t < t^*$. Now, we show that $W_{t^*}^{opt} = 1$. Since

© 2001 by CRC Press LLC
the new recipe for the next run is

\[ X_{t^*+1} = \frac{T - a_{t^*}}{b} \]

and the next run’s output is

\[ Y_{t^*+1} = \alpha + bX_{t^*+1} + S + \epsilon_{t^*+1} \]

Given the output at time \( t^* \) is observed to be \( y_{t^*} \), MSD for the next run becomes:

\[ MSD_{t^*+1} = E \left[ \left( Y_{t^*+1} - T \right)^2 \mid Y_{t^*} = y_{t^*} \right] \]

\[ = E \left[ \left( (1-W_{t^*})(T-y_{t^*}) - \epsilon_{t^*} + \epsilon_{t^*+1} \right)^2 \mid Y_{t^*} = y_{t^*} \right] \]

\[ = (1-W_{t^*})^2 (T-y_{t^*})^2 + \sigma^2 + \sigma^2 \]

Minimizing \( MSD_{t^*+1} \) with respect to \( W_{t^*} \), we obtain optimal control parameter:

\[ W_{t^*}^{opt} = 1 \text{ and } MSD_{t^*+1}^{opt} = [1+1] \sigma^2. \]

Therefore,

\[ W_t^{opt} = \frac{1}{t-t^*+1} \text{ and } MSD_{t+1}^{opt} = \left[ \frac{1}{t-t^*+1} + 1 \right] \sigma^2 \text{ for } \forall \ t \geq t^*. \]
REFERENCES


20 Enabling Generic Interprocess Multistep Control: the Active Controller

Nauman Chaudhry, James Moyne, and Elke A. Rundensteiner

20.1 INTRODUCTION

Control of a semiconductor manufacturing facility can take place at various levels throughout the facility and can take different forms. As shown in Figures 7 and 8 in the Introduction to this book, there are various levels of control that can be configured in a hierarchical complementary fashion to better achieve process goals. As R2R control becomes more mature and widely accepted, attention will become more focused on the integration of layers of control above (interprocess) and below (in situ) the R2R control level. The implementation of generic, configurable R2R control solutions, as defined in Part 3 of this book, will simplify the integration task. Similarly, development of generic and configurable solutions at other control layers will facilitate their complementary and incremental utilization in the factory, thereby easing the migration to a multilevel control scheme depicted in Figure 8 of the Introduction.¹

This chapter addresses the migration to multilevel control in the semiconductor industry by presenting a methodology for implementing generic and configurable interprocess multistep control.² ³ Multistep (also called interprocess) control can be carried out whenever the fabrication process consists of several steps. In multistep feedforward control, deviations in the processing of a wafer in one step are compensated for by adjusting the processing in some/all of the steps that are yet to be carried out on that particular wafer. In multistep feedback control, a cell which has already processed a wafer receives advice from a “downstream” cell so as to adjust its model for “better” processing in future runs. In other words, for multistep feedforward control, a cell carrying out a step on a wafer gives advice to cells which are “downstream” for the processing of this wafer, whereas in multistep feedback control the direction of flow of advice is “upstream” with respect to the processing of the present wafer, as shown in Figure 20.1.

Control of semiconductor processes over multiple steps has been identified as an important feature of the semiconductor manufacturing facilities of the future,⁴
and research has been conducted for developing control algorithms for carrying out such control.\textsuperscript{5,6} However, the development of generic enabling mechanisms, which can be used to implement multistep feedback and feedforward control, has not been adequately addressed (see Section 20.4 for further discussion). To fill this gap, we present in this chapter an adaptable, portable, and generic software enabler for multistep control in manufacturing. This enabler, called the \textit{Active Controller}, uses an \textit{active database system} to carry out multistep control, and provides the capability of defining control rules that serve to provide an adaptable and portable mechanism for multistep control.

The remainder of this chapter is organized as follows. In Section 20.2, the requirements for implementing a generic multistep controller are described. An introduction to some of the basic concepts in active databases and an in-depth description of the Active Controller is contained in Section 20.3, and in Section 20.4 the issue of integrating the Active Controller with computer integrated manufacturing (CIM) environments is discussed. In Section 20.5 the current research in multistep control and CIM environments is summarized, while conclusions are presented in Section 20.6.

### 20.2 PROVIDING MULTISTEP CONTROL

Factory computer integrated manufacturing (CIM) systems are comprised of several applications that provide the various CIM functions.\textsuperscript{7} In this section we discuss those applications whose functionality is relevant for providing multistep control. We will use this discussion to derive requirements for a generic multistep controller.

#### 20.2.1 SEMICONDUCTOR CIM ENVIRONMENT

In a typical CIM implementation,\textsuperscript{8} the processing of wafers requires interaction between various applications, the following of which are important for providing multistep control:

- The processing to be carried out to fabricate wafers is specified using an application termed \textit{Process Flow Specification Manager} (PFM).\textsuperscript{8} PFM
provides a computer-aided specification environment for defining process flows. A process flow for a particular product specifies the process steps to be carried out on the wafer, and the sequence in which these steps are to be executed to achieve the desired processing of the wafer.

• The process flow is given as an input to a scheduler. The scheduler decides when a wafer is to be processed by a piece of equipment and directs the factory resources to effect these decisions.

• Processing and control at individual pieces of equipment is carried out by equipment controllers. An equipment controller provides mechanisms for the specification and manipulation of recipes which are used for carrying out processing at an equipment when instructed by the scheduler. An equipment controller has information about required equipment controllables (settings/recipe) to achieve specified goals (effects) in the process flow.

20.2.2 Requirements for a Multistep Controller

For any wafer undergoing processing in the facility, the multistep controller needs to be able to detect when and if multistep control is required, and carry out the appropriate compensating action. To provide this functionality, the multistep controller needs to be informed of the result of each processing step completed by an equipment controller. Depending upon this result and relevant previous result data, the multistep controller may suggest changes to the existing process flow for the wafer so as to compensate for any processing errors. In addition, the multistep controller may also respond to temporal and other events (e.g., machine shutdowns, etc.) to carry out appropriate actions as needed. To provide these functions, the multistep controller should be able to

• Understand the process flow specification,
• Monitor events of interest in the facility,
• Retain or access information about the processing carried out on individual wafers,
• Keep historical information necessary to make control decisions about equipment in the facility.

In addition to the functional requirements listed above, the following design goals are identified for the multistep controller:

• Since modification of control knowledge is an ongoing necessity in a semiconductor manufacturing environment,\(^9\) the multistep control mechanism should be adaptable to allow easy extension/adaptation to handle new control situations.
• The controller should be portable so that it can be easily reused for different fabrication processes.
• The controller should be generic in that it is not tied to a particular CIM implementation and can be integrated with various CIM environments.
20.3 ACTIVE CONTROLLER FOR MULTISTEP CONTROL

The Active Controller, a multistep controller developed to fulfill the requirements for multistep control (as discussed in the previous section), is described in this section. Since active database technology is at the core of the Active Controller, we first give an introduction to relevant basic concepts of active database systems. We then discuss the design and implementation of the Active Controller, and research toward extending active database technology to provide advanced features for the general domain of process automation and control.

20.3.1 Active Database Technology

Conventional database systems are passive: they only execute queries or transactions explicitly submitted by a user or an application program. Recent trends in database technology have attempted to extend this conventional functionality to accommodate advanced applications. Active database systems represent one of the important enhancements of conventional database systems, where the database system is transformed to be active, i.e., to provide the ability to monitor the database state and to react to predefined situations without explicit user action or application requests. Situations to be observed and actions to be initiated are managed by the database system. Active database systems thus provide common application-independent functionality for complex dynamic control structures. Instead of having each application individually survey events and conditions and schedule the appropriate reactions (i.e., consistency checks, notification procedures, etc.), both the monitoring of events and the scheduling of the actions is incorporated into the database system.

The desired behavior of active database systems is expressed in terms of event–condition–action (ECA) rules. The rule syntax can be described as

\[
ON \text{ event} \quad IF \quad \text{condition} \quad DO \quad \text{action},
\]

with the following semantics: when the event occurs, if the condition is true, then carry out the action. Events signal situations inside or outside the database system. Conditions are Boolean expressions that are checked as preconditions of actions. Actions are executable routines within or outside the database system.

The basic behavior of an active database system is illustrated in the following example:

**Example:** Consider a simple rule defined to keep track of the value of temperature of an oven and raise an alarm if the value exceeds a limit.

\[
ON \text{ current temperature updated for oven} \\
IF \text{ current temperature} > \text{ high temperature limit} \\
DO \text{ signal high temperature alarm}.
\]

Now, whenever an application updates the temperature value in the database, the active database system will check if the temperature value exceeds the specified limit, and if so will raise a high temperature alarm. Notice that if the policy is to be changed (e.g., the high temperature limit is to be modified), this will require only modifying the rule.
20.3.2 Using Active Databases for Multistep Control

Multistep control is performed using ECA rules as follows: rules encoding the desired control algorithms are defined on the process specification. Execution of a process step at a machine or a composition of several such process steps constitutes the (composite) event for the Active Controller rules. The conditions of these rules are the scenarios in which multistep control can be used to improve the processing of the product, i.e., the situations in which processing at certain machines can be adjusted to compensate for errors in processing at certain other machines. These conditions are defined in terms of relevant process parameters, such as predicted process mean, specification limits, etc., and parameters of the manufacturing process, e.g., acceptable yield loss, etc. Firing of the rules causes appropriate actions to be taken to change the processing to be carried out in the future, and thus compensate past processing errors.

A suitable active rule mechanism which allows expression of a range of conditions for multistep control provides for a portable implementation of multistep control, since the specification of the control behavior via active rules means that same controller can be reused for a different process by defining a suitable rule base. This mechanism is easily adaptable, since by adding appropriate rules the controller can be adapted to handle new situations. This is in contrast with the use of an application program written for event detection and condition checking, because in this case moving from one process to another may require significant modification of the application program.

It should be noted that since expert systems provide the capability of rule definition and execution, a multistep control enabler could be developed using expert systems technology instead of using active database systems. Use of expert systems, though, would require the overhead of coupling the expert system to the (passive) CIM database. Additionally, active database systems are more powerful than expert systems. Expert systems typically have condition–action rules, which are evaluated and executed in recognize–act (“if–then”) cycles. Active databases have event–condition–action rules, which are a superset of the expert system condition–action rules and can provide more efficient processing of rules. Other advantages of active database systems over expert systems stem from the fact that active database systems provide all the capabilities of passive database systems. Database mechanisms such as indexing can be utilized for condition checking, making rule execution more efficient than in expert systems. Active database systems have in fact been suggested as suitable platforms for building large and efficient expert systems.

20.3.3 Active Controller Design and Sample Implementation

An Active Controller solution has been implemented using the Ode active object-oriented database system developed by AT&T. The implementation has been carried out using O++, Ode’s programming language, and comprises about 10K lines of O++ source code. A graphical user interface (GUI) has been developed for
20.3.3.1 Modeling the Process Flow

In current semiconductor manufacturing CIM systems, process flows to fabricate a wafer are specified by a hierarchical object-oriented data structure. Process flows are composed from flows, processes, steps, and equipment, as shown in Figure 20.2, using the object modeling technique (OMT). In this model, a step is the basic unit of processing and is defined as what can be accomplished in one chamber. A step is carried out on one piece of equipment. However, the same equipment may be used by more than one step. The next higher processing abstraction is a process. A process is defined as a sequence of other processes or steps. The highest processing abstraction is a flow, which is defined as a sequence of processes.

Each flow, process, and step may have a set of effects, which describe the expected impact of the flow, process, or step on the wafer. The effects associated with a step are used to create settings/recipe for the equipment on which the step is being carried out. However, these effects can be overridden by a process that contains this step. Similarly, a flow may use its set of effects to override effects associated with its processes.

The current software implementation of the process-flow schema in the Active Controller includes classes corresponding to flow, process, step, and effect (and subclasses of these classes). Equipment is simulated by asking the user of the Active Controller to input the results of process-flow execution.

Example #1: A screen shot of daVinci’s representation of a process-flow in the Active Controller database appears in Figure 20.3. In the Active Controller GUI, a flow is represented by an oval, a process by a rhombus, and a step by a rectangle. The effects of a step are shown in the rectangle under the name of the step. The figure shows a...
flow instance cmos_3 which is composed of one process mosfet_n1. mosfet_n1 is composed of a process gate_definition4, which in turn is composed of various steps and processes, such as deposit1, photolito20, measure4, and etch3.

20.3.3.2 Defining Rules for Multistep Control

When a wafer is to be fabricated, an instance of the corresponding flow is selected from the database and the resultant process-flow is executed. The execution of the process-flow results in the execution of the processes of which the flow is composed. Similarly, the execution of a process, in turn, causes the execution of the processes and steps of which this process is composed. The fabrication of a wafer thus results in the execution of a number of processes and steps at different levels of nesting in the process-flow.

As the execution of a process-flow proceeds, the need to control the wafer fabrication frequently requires changes to the original processing sequence specified in the process-flow. To provide this capability we need the ability to specify control actions for the process specification. We use the rule definition facilities of Ode to specify the multistep control knowledge in terms of active rules defined on the process flow structure. These control rules can trigger appropriate control actions during process execution whenever the need arises, and can modify processing by, for example, requiring reexecution of a part of the process-flow. Simpler modifications consist of tuning the process by changing just the effects associated with one or more steps in the processing sequence. Such modifications can be easily accommodated by causing the appropriate attributes of the relevant step objects (e.g., settings) to be changed.
Example #2: Consider that a photolithography sequence (see Figure 20.3) is being executed and a statistically significant error violating the control limit is observed in the photoresist linewidth. The rule encoding to detect this condition, \texttt{photoresist_error} defined on class \texttt{measure} is shown in Figure 20.4a and the controller operation to verify this condition is illustrated in Figure 20.4b. The event associated with the rule is \texttt{after measure\_execute} and hence will be raised after the execution of the complete (postlithography) measure step. When an out-of-limits condition is detected (Figure 20.4c), the appropriate rule fires (Figure 20.4d) and a new recipe is generated for the (downstream) etch step by modifying the etch target (Figure 20.4e).

Example #3: Consider that a photolithography sequence is being executed and a statistically significant error violating the control limit is observed in the reflectance of the wafer after the spin-coat and bake step. A new recipe is generated for the exposure step by modifying the exposure time to compensate for the reflectance error; the formulation for recipe adjustment for example is given in Reference 20. In the active controller implementation, a rule \texttt{spin\_cb\_error} is defined on the class \texttt{spin\_coat}. The event associated with the rule is \texttt{after spin\_cb\_execute} and hence will be raised after the execution of the spin-coat and bake step. To check for alarm condition a function \texttt{error\_check} is called, and if it returns TRUE, indicating an alarm has been raised, the spin-coat step informs the photolitho step of this error. In case this error has been reported, the rule \texttt{spin\_cb\_error} defined on the \texttt{expose\_resist} class is fired before the execution of the expose step. This results in modifying the exposure time. This modification is achieved by calling the \texttt{modify} method for \texttt{effects}, which can query the database to determine the results of the spin-coat step and call an external function to find out the new recipe.

In addition to the simple modification of changing only the recipe associated with a particular step (as in the previous two examples), the Active Controller can also implement more complex control actions such as rework, repair, etc. Actions such as rework require modification to the actual sequence of steps that were to be carried out. The Active Controller can be used to implement such control actions.³

20.3.3.3 Active Controller Features

Ode provides powerful rule definition capabilities. The condition part allows calling functions in addition to querying the database, while the action part can contain programming language statements and allow for interaction with foreign functions. This capability can be used to integrate third-party analysis software with the Active Controller implementation.

Ode also provides a rich event language with multiple operators (e.g., conjunction, negation, ordering, etc.) for composing events. Events can be defined to be raised either before or after the execution of a method. Events can also be defined to be raised with respect to transactions. When defining a class, the user specifies the events that need to be monitored for instances of this class. Ode keeps a history of these events of interest. This capability, in conjunction with the event composition operators, means that the Active Controller implementation provides an extremely powerful control rule definition capability. It is thus possible, for example, to use the Active Controller to define rules that fire based on historical trends or on events that occur at different points in the processing of a particular wafer. Note that the
Figure 20.4 (a) Rules Defined (on “gate_definition”, “measure”, and “etch”) for correcting for photoresist linewith error. (b) Mechanism for evaluating photoresist linewith error. (c) Detecting linewith error and rule firing. (d) Figure 20.4d. Active Control: action associated with rule firing; adjustment of etch recipe. (e) Active Control: correcting for linewith error by adjustment of etch recipe.
FIGURE 20.4 (continued)
Active Controller implementation provides all the functional capabilities identified in Section 20.2. In addition, the use of active database technology in the Active Controller helps achieve the design goals of adaptability and portability. The third design goal of genericity of the Active Controller implementation is linked to emerging trends in CIM environments and software technology, and is further discussed in Section 20.4.

20.3.4 Extending Active Database Technology for Process Control

Active database technology is still in the process of maturing. In particular, unlike traditional (passive) database systems for which well-defined methodologies for building database applications exist, a key difficulty in developing applications using active databases is the lack of methodological support for active rule definition. It has also been noted that another hindrance in the employment of active database technology is the difficulty in defining and maintaining rule sets. Therefore, for active database technology to be an attractive vehicle for the implementation of advanced applications, the implementor needs to be guided toward defining rules that are easy to analyze and have predictable behavior.

To make active database technology suitable for developing control applications, solutions have been developed for these different problems for the domain of process control by exploiting particular characteristics of the process specification model (i.e., Figure 20.2). A detailed description of this research is given in Reference 3. Here we give a brief summary of some of the salient features.
To overcome the absence of design methodologies for defining active rules, specific guidelines for rule definition are developed. Programming and maintaining rules that arbitrarily span many classes can be an extremely difficult task, since the rule programmer has to ensure that a newly defined rule does not cause unforeseen interaction with other rules already defined for other classes. In addition, such rule definition violates the object-oriented principle of encapsulation. Guidelines have been developed that require each rule to have a well-defined scope of influence. To limit the scope of interaction of rules, each rule has to be defined on a particular process (or flow or step) class. The rule can only refer to properties (i.e., attributes, methods, or other rules) that are directly accessible from the process class it is defined on. Since we remain within the scope of the process on which the rule is defined, the resulting definition of process classes respects the principle of encapsulation, and this leads to rules whose interaction is easier to analyze. We have applied these design guidelines in defining control rules for the Active Controller, and have found the resulting rule sets to have the aforementioned properties of maintainability and predictability.

### 20.4 INTEGRATING THE ACTIVE CONTROLLER WITH CIM ENVIRONMENTS

Having described the design and implementation of the Active Controller, we now look at the issue of integration of this controller in CIM environments. In this area, the SEMATECH CIM Framework provides a comprehensive specification for an application framework for CIM in semiconductor facilities. The heart of the CIM framework is a set of semiconductor manufacturing abstractions and services that are typically embodied as applications. The CIM Framework can be viewed as a set of integrated frameworks, providing functionalities at different levels in the factory. These frameworks include Enterprise Framework, MES Framework, APC Framework, and Equipment Integration Framework (see Figure 7.2 of Chapter 7).

The Active Controller can be envisaged as an additional application in the APC Framework providing multistep control while drawing upon the services provided by several applications, such as the Specification/PFM application in the MES Framework, the Process History application in Equipment Integration Framework, etc. Object technology is at the core of the design and implementation of the Active Controller. Hence, as the CIM framework matures, it should be possible to tailor the Active Controller implementation to fit in as an application drawing upon and providing services to the CIM Framework.

The Active Controller needs to interact closely with the PFM. To modify the processing of the wafer, the Active Controller has to act on the process flow specification and thus requires knowledge of this specification. This information is maintained by the PFM. One possible design alternative is to add the functionality of the Active Controller to the PFM. However, to keep the Active Controller as a pluggable application, the Active Controller has been designed to duplicate the needed process-flow information in the Active Controller database, and then the Active Controller as an advisory application to the PFM.
20.5 RELATED RESEARCH

20.5.1 Research in Multistep Control

Recently, research has also been carried out to provide control of sequences of interrelated processes. A control system for photolithography sequences is described in Reference 5. Two process control methodologies are used for multistep control. In one of these, a local controller corrects the shortcomings of the present machine by generating customized recipes at the next process step. In the other control methodology, a global controller finds optimum specification for the upstream processes to ensure that the outputs of the final process can meet their specifications. Results indicate a significant improvement in the overall capability of the process sequence.

Another research effort in multistep control is reported in Reference 6. Feedforward and feedback techniques are applied to a processing sequence involving the four basic steps of silicon oxidation, aluminum metallization, lithography, and aluminum etch. Feedforward control is carried out at two points. The stepper focus in the exposure substep of the photolithography step is varied depending upon the result of the oxide thickness of the oxidation step. Adjustments are also made in the etching step, depending upon the oxide thickness and the size of the resist pattern. The study demonstrated the potential of using multistep control by achieving the stated goal of keeping the capacitance of a fabricated capacitor fairly constant (within 1%). A recent research effort in feedforward control is described in Reference 23, where a feedforward controller adjusts the recipe of an etch step to compensate the deviations in a photolithography process.

The control strategies described in Reference 6, as well as the strategies described in References 5 and 23, can be implemented by using the Active Controller. In fact, the control rules shown in the examples in Section 20.3 implement some of the control strategies described in these papers. Since the Active Controller provides an adaptable vehicle for implementation of control strategies, results obtained from future research in the area of multistep control could also be easily incorporated in the Active Controller via the definition of appropriate active rules.

20.5.2 Research in CIM Environments

Among research efforts devoted to developing semiconductor manufacturing systems, the Berkeley Computer Aided Manufacturing (BCAM) system developed at the University of California, Berkeley, is to our knowledge the only system that provides multistep control functionality. BCAM software utilizes equipment models for supervisory control. The models are used by the software for process simulation and recipe generation. This recipe generation is used to accommodate product specifications. BCAM stores and retrieves equipment recipes and models in a recipe database. Research within BCAM has also focused on the provision of multistep control as described in References 5 and 20, and discussed in the previous section. The Active Controller solution can be considered to be complementary to this research effort since the Active Controller provides a clean and easy platform for implementing these various control algorithms.
20.6 CONCLUSIONS

In this chapter an interprocess control-enabling mechanism called the Active Controller was described. The solution is a generic, adaptable, and portable software enabler for multistep control in manufacturing facilities. The Active Controller provides multistep control through the use of active database technology. It keeps track of relevant processing events and data, and when the conditions for multistep control hold, executes appropriate actions to compensate for the errors in processing.

The Active Controller implementation, with its capability for the definition of complex rules over a history of processing events and ability to invoke user-provided analysis routines, provides for a generic and adaptable implementation that can be used to implement various algorithms for multistep control. The same controller can be reused for a different process by defining suitable rules for the new process and adding these rules to the active database. This mechanism is flexible and adaptable since, by adding appropriate rules, it may be extended to handle new situations.

Active Controller research had its foundations in R2R control and extended concepts of reusability and flexibility to higher levels in the factory control hierarchy. Similarly, research is also being conducted in developing real-time control solutions for specific processes (see, for example, Reference 25), thereby addressing levels of below R2R control. If these control solutions at all levels continue to be developed with forethought to multilevel integration, a true multilevel generic control solution for semiconductor manufacturing will soon be realized.

ACKNOWLEDGMENTS


REFERENCES


14. Ode 4.0 User Manual, AT&T Bell Laboratories, Murray Hill, NJ.


Part 7

Summary and Conclusions

This text has served to provide a detailed look into the many aspects of R2R control in semiconductor manufacturing from an historical perspective through a description of basic foundational tools, algorithms, enabling technologies, and customization methodologies, to case studies and advanced topics. In this final part of the book we provide a high-level summary of the material presented in the book, give a snapshot of the current status of R2R control development and deployment in the industry, and identify general future directions for R2R control. As part of the discussion of future directions we will highlight a few new research and development efforts that came into fruition near the final stages of editing this text; these efforts serve to illustrate that the level of activity in both R2R control research and development continues to grow. We also identify particular conferences and publications that could serve as avenues for further research, and provide some final thoughts on this book and its role in facilitating R2R control understanding and deployment.

1 SUMMARY

The R2R control field is expansive and involves many dimensions of research and development. In organizing this book, we quickly came to realize that the wealth of research results in each of these dimensions could support its own complete text. Thus, in providing a text that addresses the entire semiconductor manufacturing R2R control field, we chose to provide the reader with a foundation of information in each dimension, along with more in-depth treatises of important aspects of each dimension. In this way the reader is provided with a foundation to further investigate the aspects of R2R control in which he/she is most interested.

Although the R2R control field could be partitioned in a number of ways, we chose a partitioning that seems to reflect a natural grouping of research efforts and
researchers. These partitions roughly correspond to the “Parts” of this book. In the remainder of this section we provide a brief summary for each of these parts.

1.1 Foundation

In Part 1 of the book, practical and theoretical components that comprise the foundation for R2R control were presented. R2R control in the semiconductor manufacturing industry arose from both the successes and limitations of SPC. SPC provided the first solution that utilized process observation to affect the process. R2R control emerged as a new alternative to SPC in the early 1990s.

A number of issues hindered early widespread acceptance of R2R control. These included lack of commercial solutions, no infrastructure for integration or automation, few on-line metrology and in situ sensors, and possibly inadequate algorithms. All of these issues were addressed as part of the maturation process of R2R control, resulting in increasing acceptance of this technology.

The benefits of this technology in the semiconductor industry are seen as increased throughput, reduced non-product wafers, improved wafer-to-wafer and lot-to-lot variability, reduced within-wafer and within-die variability, and reduced operator error. Possible future directions in the R2R control field include increased use of complex, adaptive, self-tuning controllers, tool- and process-specific models/controllers, and multistep or full-flow process controllers. Better understanding of the properties of existing R2R control solutions (along the lines described in Chapter 2) will lead to improved tuning methods and better performance.

1.2 R2R Control Algorithms

A wide range of R2R control algorithms were described in Part 2. Two basic algorithms, called the “gradual mode” and “predictor–corrector,” utilize linear approximation modeling combined with EWMA data filtering. These algorithms are widely utilized in the semiconductor manufacturing R2R control industry, and have been proven effective in a number of process control scenarios. The OAQC algorithm provides a quadratic modeling capability and can be utilized either as a process optimizer or a process controller. This capability all but eliminates the need for a DOE preceding the deployment of the controller.

These algorithms represent just a few of the R2R control algorithm alternatives available; however, these approaches represent the majority of R2R control implementations in the industry. A comparison of these algorithms reveals that, in most relatively well-behaved scenarios, the basic linear approximation algorithms perform as well or nearly as well as the quadratic solutions. A quadratic (or other nonlinear) control solution proves necessary, however, if the system is highly nonlinear.

1.3 Integrating Control

The existence of methodologies and technologies for effective integration and automation of R2R control represents a major hurdle to its acceptance. This topic was addressed in Part 3 with a detailed description of methods and solutions for integrating control. Three major entities facilitating the development of solutions for
R2R control integration are the Semiconductor Industry Association, which provides a roadmap for APC development, acceptance, and adoption; SEMATECH, which has defined a Control Systems Requirements Specification for the industry; and SEMI, which continues to support development of standards for communications and integration.

A detailed requirements analysis for integrated R2R control solutions has been conducted for the industry, and an R2R control enabling technology, called the Generic Cell Controller (GCC), has been developed that addresses these requirements. The GCC has been effectively utilized as an enabling solution for a large number of applications, including CMP, vapor phase epitaxy, and etch. In providing a migration path for the integration of R2R control into existing and next generation systems, a “piggyback” R2R control design has been developed that provides for the addition of a R2R control capability to an existing system with minimal change to that system (i.e., virtually nonintrusive at the equipment and factory level). Utilizing a technology such as the GCC to implement this design, the piggyback solution can be ported to a fully integrated solution for next generation tools. An example of this process illustrates the cost and technology leverage that can be obtained in the migration from piggyback to fully integrated solutions.

1.4 CUSTOMIZATION METHODOLOGY

In Part 4 a methodology for deployment of R2R control is presented. The components of the methodology include (1) quantifying the problem with process engineers, (2) DOE for process characterization and R2R control model development, (3) DOE execution, (4) subsequent data analysis, (5) R2R controller specification, (6) customization and delivery of R2R system, and (7) testing, training, and technical support.

Customization of R2R control solutions is almost always required; this motivates the need for an open and flexible solution-enabling technology such as that described in Part 3. A key part of the methodology is the development and qualification of response surface models of the process. Classical methods are available that can be applied. If control solutions that include an optimization component are used, such as the OAQC solution described in Part 2, several of the response-surface quantification steps may be omitted from this deployment methodology.

1.5 CASE STUDIES

A number of case studies of R2R control are included throughout this book as vehicles to illustrate a particular feature of the R2R control field that is the topic of that part. In Part 5 the focus is on the case studies themselves. Here we illustrate the effectiveness of R2R control, and provide insight into possible future directions of R2R control research and development.

For example, an R2R control solution being deployed at IBM* is a multiprocess solution that includes a CMP component and possibly a CVD component, along with feedback from the upstream etch process. This multiprocess solution is providing

* IBM Microelectronics, Burlington, Vermont.
R2R control as a total factory solution rather than a process-centric solution, with the control solution quality parameters including total line yield rather than just individual process quality parameters.

Other case studies illustrate how practical application of R2R control can motivate and guide algorithm enhancement. In a photolithography application, an adaptive version of the basic EWMA control solution was used to provide effective control in an environment where there are both small and larger process shifts and drifts. In addressing a CMP process control scenario with both aging pad and disc, a predictor–corrector EWMA algorithm was enhanced to provide control in an environment where metrology observations are obtained at unequally spaced points in time.

1.6 **Advanced Topics**

The R2R control field is rapidly expanding in a number of directions. A few of these directions are explored in Part 6. One clear future direction is the movement from process-centric R2R control to a total factory solution. R2R control solutions must be flexible so that targets are set based on factory goals rather than process goals.

Facilitating flexible solutions at this interprocess level requires an assessment of factory level integration and enabling requirements, and utilizing the sequential control and integration techniques that are suitable for factory-wide systems. The Active Controller represents such a solution for the interprocess control arena; this event-driven control solution utilizes active database technology to provide a flexible and portable sequential control environment.

Another future direction in the R2R control field is the enhancement of R2R control solutions to include process-specific modeling features. For example, a control solution that models a pad replacement process shift in a CMP process will produce a solution that provides for reduced test wafer requirements and increased OEE through reduced process (re)qualification time.

2 **Current Status of R2R Control in Semiconductor Manufacturing**

At the time of this writing, R2R control solutions are being widely reported, originating from users, metrology suppliers, and integrated control solution suppliers.\(^1\)\(^-\)\(^3\)

As noted throughout this book, success is being reported in control of all front-end process types. Thus, to some extent, the focus with respect to R2R has shifted from demonstrating viability of R2R control, to demonstrating R2R control as an integral part of total factory solutions.\(^4\) In this regard, enabling technologies such as the Generic Cell Controller and framework specifications will remain in the spotlight as tools to facilitate factory-wide deployment of R2R control.* Examples of these are provided in Chapters 15 and 20.

Although research into enhancement of control algorithms has produced a number of alternatives, the dominant algorithm types being utilized in the industry continue

---

* See Part 3 and Chapter 1 of this book, respectively.
to be variants of the first linear approximation algorithms with EWMA filtering as
described in Chapter 3. This will probably continue to be the case because, as noted
in Chapter 1, the “keep-it-simple” approach seems to be effective, and other aspects
of R2R control acceptance, such as integration, are lagging behind algorithm devel-
opment. It is important that the research in algorithm development and refinement
continue, however, because, as integration issues are rapidly being solved, algorithm
refinement could move back toward the top of a priority list. In this regard, it is
critical that integrated control enabling solutions support a level of “plug-and-play”
of algorithms for rapid and cost effective upgrade of control solutions.*

A dominant enabling technology for R2R control as a third-party solution continues
to be the Generic Cell Control.** While a large number of solutions have been reported
in the user community that don’t utilize the GCC approach, much of the GCC
literature base for R2R control focuses on integrated and portable solutions.*** The GCC
compatibility with the APC framework (see Chapters 9 and 10) should allow this
technology to continue as an enabler for R2R and other APC solutions well into the
future.5 The APC framework will continue to play an increasingly important roll in R2R
control deployment, due in part to the increasing acceptance of R2R control as part of
a total factory solution (see above), and the fact that the APC Framework is being
specified as a SEMI standard.****

The research and development arena for a large majority of activity in R2R
process control continues to be front-end processing. This focus has arisen for both
technical and historical reasons. As noted in the Introduction to this book, the CMP
process was an early target for R2R control investigation because at the time it was
a new technology that suffered from process drift and shift as well as a lack of in situ
sensory capability. CMP continues to be an ideal candidate for R2R control; however,
application has quickly spread to other processes in the front-end community. This
is due mainly to the relative similarity between CMP and other front-end processes,
and the fact that the front-end community shares the same literature base. At the
time of this writing there was no known effort dedicated to deploying R2R control
for back-end processes such as assembly and test; however, it is reasonable to assume
that these processes also have drift and shift characteristics that could make them
ideal candidates for R2R control. Thus, movement of R2R control into this area is
a probable future direction in the industry, as described below.

3 THE FUTURE OF R2R CONTROL IN
SEMICONDUCTOR MANUFACTURING

Although R2R control solutions are being deployed throughout the industry, the
field of R2R control in semiconductor manufacturing, which is only about 10 years
old, could yet be considered in its infancy. Results reported to this point have served

* See, for example, Chapters 9 to 11.
** The GCC is described in Part 3.
*** See, for example, Chapter 11.
**** SEMI is an acronym for Semiconductor Equipment and Materials International, and is described
in Chapter 7.
to provide effective control solutions for various processes in the industry, and they have also opened new doors, showing us that we have only begun to realize the potential of APC. In the remainder of this section a few of the new opportunities for expansion of R2R control and APC are discussed.

3.1 **R2R Control as Part of a Factory-Wide APC Solution**

The case study presented in Chapter 15 illustrated a trend toward viewing R2R control as part of a factory-wide solution. Factory-wide deployment of R2R control and, more importantly, factory-wide access to control data, will serve to open up new research and development areas that will extend the reach and capability of advanced process control. One such area is enabled by the (electronic) publishing of metrology data for a process at the factory level. This allows upstream and/or downstream process control solutions to utilize this data in their control strategies. As an example, CMP pre-metrology data can be used as part of the CMP control scheme to control CMP thickness and radial nonuniformity. However, through the use of deconvolution techniques, it can further be used in the upstream CVD control scheme to identify the magnitude of cross wafer gradient or “wedge” nonuniformity contributed by the deposition process.  

Another research and development area opened up by factory-wide data access results from the accessibility of control algorithms and parameters. This allows the development of factory-wide, rather than process-centric, control algorithms that focus on process line yield optimality, rather than process optimality. It also allows for the incorporation of higher level control loops as part of a multilevel hierarchical control scheme. Further, it allows for the use of factory level control enabling solutions that support incremental improvements to the factory-wide control scheme.

3.2 **Integration of APC Components**

R2R control is clearly a mature APC technology; however, other APC technologies are emerging in the industry. For example, fault detection and classification (FDC) is receiving an increasing level of attention and could become a critical component of some process solutions. A high-level schematic representation of an FDC system is given in Figure 1. A typical FDC system utilizes *in situ* process data along with possibly *ex situ* metrology data to identify and classify tool and process faults. Tools such as time series analysis or neural networks can be utilized to establish and qualify the relationships between the data collected and the fault classes. Other APC technologies such as maintenance scheduling and automated process modeling are also receiving attentions as components of an open APC solution suite.

As these individual APC components mature, it is becoming more and more evident that they can be utilized in a complementary fashion if the underlying enabling technologies and frameworks facilitate the structured exchange of process and control information among the components. While the enabling technology and frameworks described in Chapters 9 and 1, respectively, do address this issue, it is
expected that future APC efforts will be devoted to further specifying and evaluating the interaction of the various APC technologies via these enablers and frameworks.

4 FURTHER READING

The references found throughout this book list the primary conferences and publications that present the latest results and innovations in R2R control for semiconductor manufacturing. A summary of these conferences and publications, and the probable content of material presented, is provided in Table 1.

The single most important conference associated with APC in the semiconductor manufacturing industry is the SEMATECH Advanced Equipment Control Symposium. As the name implies, this workshop has served as a forum for presentation

---

**TABLE 1**

Summary of Sources of Further Reading

<table>
<thead>
<tr>
<th>Resource</th>
<th>Internet Reference</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Vacuum Society</td>
<td><a href="http://www.vacuum.org">www.vacuum.org</a></td>
<td>Symposums and Journal</td>
</tr>
<tr>
<td>CMP for Multilevel Interconnection Conference (CMP-MIC)</td>
<td><a href="http://www.imic.org">www.imic.org</a></td>
<td>Conference</td>
</tr>
<tr>
<td>Electrochemical Society</td>
<td><a href="http://www.electrochem.org">www.electrochem.org</a></td>
<td>Symposums and Journal</td>
</tr>
<tr>
<td><em>IEEE Transactions on Semiconductor Manufacturing</em></td>
<td><a href="http://www.ieee.org">www.ieee.org</a></td>
<td>Journal</td>
</tr>
<tr>
<td>SEMATECH Advanced Equipment Control Workshop</td>
<td><a href="http://www.sematech.org">www.sematech.org</a></td>
<td>Workshop</td>
</tr>
<tr>
<td>SEMI Standards</td>
<td><a href="http://www.semi.org">www.semi.org</a></td>
<td>Equipment Automation and Software Standards Conference</td>
</tr>
<tr>
<td>SEMI/IEEE Advanced Semiconductor Manufacturing Conference (ASMC)</td>
<td><a href="http://www.semi.org">www.semi.org</a></td>
<td></td>
</tr>
</tbody>
</table>
of innovative, though oftentimes preliminary, APC results, and obtaining feedback and direction.

The American Vacuum Society and the Electrochemical Society also regularly sponsor symposiums that contain a wealth of submissions devoted to semiconductor manufacturing process control. Examples of other conferences where papers devoted to semiconductor manufacturing process control regularly appear include the SEMI/IEEE Advanced Semiconductor Manufacturing Conference (ASMC), and the CMP for Multilevel Interconnection Conference (CMP-MIC).

Journals that regularly contain papers devoted to aspects of semiconductor manufacturing process control include *IEEE Transactions on Semiconductor Manufacturing* and *IEEE Transactions on Components, Packaging, and Manufacturing Technology — Part C*. Journals of the American Vacuum Society and the Electrochemical Society also occasionally contain papers devoted to APC topics.

Another important source of APC innovation is the SEMI standards meetings. As noted in Chapter 7, many aspects of integrated R2R control such as “piggyback” communications and the APC Framework are being pursued as standards within SEMI. Many of the task forces within SEMI devoted to the development of these standards have defined solutions to integrated process control and, more importantly, are shaping the path for APC integration and acceptance.* Thus, any developer interested in any aspect of APC for semiconductor manufacturing should include research into the latest relevant SEMI standards as part of any literature survey.

## 5 FINAL THOUGHTS

R2R control continues to be an exciting field for research and development in semiconductor manufacturing. The benefits that have been experienced to this point represent the tip of the iceberg. The road to total acceptance and utilization of R2R control in aspects of semiconductor manufacturing is necessarily long; however, as researchers have paid attention to generic, portable, and expandable solutions, a clear migration path to integrated factory-wide R2R control has been opened up. Any research effort, though, is only effective if the results are communicated to the industry in a timely and organized fashion. In this book, we put a level of organization around this rich research pool of R2R control and presented the results in a fashion that would allow a developer to understand and take advantage of the many facets of R2R control research. In doing so, it is our hope that this book is a step on the path toward R2R control being fully accepted as an integral component of a total factory solution for semiconductor manufacturing.

---

* The primary group addressing APC within SEMI is the Information and Control Committee; various subcommittees and task forces are under its leadership. Further information can be found at [www.semi.org](http://www.semi.org).
REFERENCES

13. A number of articles on Fault Detection and Classification can be found in *IEEE Transactions on Control Systems Technology*.