integrated systems

AFOSR-TR-97 0310

Final Report

Reporting Period: 15 Oct. 1993 - 14 M

# Unified Controller Design for Intelligent Manufacturing Automation

*Sponsored by:*

**Defense Advanced Research Projects Agency**

DARPA Order No. FQ8671-9500762 A108/5

Monitored by AFOSR Under Contract No. F49620-94-C-0003

*Prepared by:*

Dr. Robert L. Kosut
**SC Solutions**
3211 Scott Blvd., Santa Clara, CA 95054
kosut@scsolutions.com

Dr. Gurcan Aral
**Integrated Systems, Inc.**
201 Moffett Park Drive, Sunnyvale, CA 94089
garal@isi.com

*Prepared for:*

**Defense Advanced Research Projects Agency**
Attention: Dr. Anna Tsao
3701 North Fairfax Drive
Arlington, VA 22203-1714

and

**AFOSR/NM**
**Directorate of Mathematics and Geociences**
Attention: Dr. Marc Jacobs
110 Duncan Avenue Suite B115
Bolling AFB, DC 20332-0001

ISI Report No. 5250-04

14 May 1997

Integrated Systems, Inc.
201 Moffett Park Drive
Sunnyvale, CA 94089

Final Report

Reporting Period: 15 Oct. 1993 - 14 May 1997

# Unified Controller Design for Intelligent Manufacturing Automation

*Sponsored by:*

**Defense Advanced Research Projects Agency**

DARPA Order No. FQ8671-9500762 A108/5

Monitored by AFOSR Under Contract No. F49620-94-C-0003

*Prepared by:*

Dr. Robert L. Kosut
**SC Solutions**
3211 Scott Blvd., Santa Clara, CA 95054
kosut@scsolutions.com

Dr. Gurcan Aral
**Integrated Systems, Inc.**
201 Moffett Park Drive, Sunnyvale, CA 94089
garal@isi.com

*Prepared for:*

**Defense Advanced Research Projects Agency**
Attention: Dr. Anna Tsao
3701 North Fairfax Drive
Arlington, VA 22203-1714

and

**AFOSR/NM**
**Directorate of Mathematics and Geociences**
Attention: Dr. Marc Jacobs
110 Duncan Avenue Suite B115
Bolling AFB, DC 20332-0001

ISI Report No. 5250-04

14 May 1997

**Integrated Systems, Inc.**
201 Moffett Park Drive
Sunnyvale, CA 94089
Tel: 408-542-1500 Fax: 408-542-1950

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 14, 1997

3. REPORT TYPE AND DATES COVERED
Final Report10-15-93 - 05-14-97

4. TITLE AND SUBTITLE
Unified Controller Design for Intelligent Manufacturing Automation

5. FUNDING NUMBERS
F49620-94-C-0003

6. AUTHOR(S)
Dr. Robert Kosut & Dr. Gurcan Aral

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SC Solutions & Integrated Systems, Inc.
201 Moffett Park Drive
Sunnyvale, CA 94089

8. PERFORMING ORGANIZATION REPORT NUMBER
5250-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects
3701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
FQ8671-9500762 A108/5

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The proposed objective was to develop a unified controller design methodology for manufacturing automation systems and to demonstrate the approach on a manufacturing process of interest to DARPA. The demonstration system selected was rapid thermal processing (RTP) of semiconductor wafers. This novel approach in integrated circuit manufacturing demands fast tracking control laws that achieve near uniform spatial temperature distributions. In order to ensure the final product quality, it is essential to maintain a uniform temperature profile despite uncertainties in both transient and steady-state phases of the process. Specific accomplishments included the development of mathematical and computational tools for heat transfer modeling, specifically conduction and multiband radiation, nonlinear model reduction, methods for robust thermal control, and an approach applicable to repetitive run-to-run feedforward learning control All the results were tested for feasibility on commercial RTP chambers.

14. SUBJECT TERMS

15. NUMBER OF PAGES
78

16. PRICE CODE

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UNLIMITED |

# Contents

# 1 Objectives & Accomplishments

## 1.1 Objectives

Our proposed objectives were twofold:

**Objective 1**

To develop a unified controller design methododology for manufacturing automation systems and to demonstrate the approach on a manufacturing process of interest to DARPA.

**Objective 2**

To provide specifications for computer-aided-control-engineering (CACE) design tools that appeal to the needs and skill level of process control engineers.

These objectives were aimed at the general process control industry, in the hopes of extending the scope of the DARPA initiative known as IPM (Intelligent Processing of Materials), *i.e.*, to merge recent advances in process modeling, sensor development, and computer-aided-control-engineering.

## 1.2 Accomplishments

The demonstration system selected was rapid thermal processing (RTP) of semiconductor wafers. This novel approach in integrated circuit manufacturing demands fast tracking control laws that achieve near uniform spatial temperature distributions. In order to ensure the final product quality, it is essential to maintain a uniform temperature profile despite uncertainties in both transient and steady-state phases of the process. Hence. the high performance requirements for RTP make it an excellent candidiate to meet both of our overall objectives. Another reason for selecting RTP is because of our physical proximity to Applied Materials Research, Inc. (AMAT). As a result we were able to test the feasibility of many of our ideas using AMAT facilities. [1]

Specific accomplishments are:

- Heat transfer modeling
- Model reduction
- Robust thermal control
- Feedforward learning control

A summary of our work in each of these areas is described in the next section.

[1] The data presented here and in our related publications is based on a generic RTP model and does not use the actual data obtained from any RTP system developed by AMAT.

# 2 Summary

Further details of the work summarized in this section can be found in a series of published papers which are included in the appendix of this report.

## 2.1 Heat transfer modeling

Predictive process models are crucial to accomplishing both path planning and feedback control especially when sdome of the key performance variables cannot be sensed *in situ*. In general, writing relations with the goal of completely describing the physics, micromechanics, or material science phenomena can produce different results from writing relations with the goal of using them to design robust paths and robust feedback controls.

Thermal systems are ubiquitous in material and semiconductor manufacturing systems. Thermal systems can be physically modeled using finite elements consisting of first order systems of ordinary differential equations. Most of the predominant thermal affects can be modeled in this way, e.g., radiation, conduction, and convection. The radiation affects are nonlinear involving quartic temperature terms. Radiation has a significantly larger effect in an RTP chamber than in a horizontal belt furnace principally because the RTP chamber is much smaller and the radiation effects diminish rapidly with distance. As a result, furnace thermal models are essentially linear whereas RTP systems are not, but can be well represented by linear models over a reasonably large region centered about each critical operating point.

Finite element thermal models can have thousands of states and even many more parameters, e.g., thermal coefficients. From the control perspective, the structure and accuracy of the model is most important. In addition, the large number of variables requires numerical software which is efficient and fast. The major goal in developing control software for these systems is to allow the user to establish variou design tradeoffs. e.g., the tradeoff between and uncertainty and performance.

We have developed a systematic procedure of generating nonlinear finite-element heat transfer models based on a set of node temperatures and associated heat transfer characteristics between the nodes. A circuit-theoretic approach is adopted due to the ease of constructing. modifying and verifying the models. With increasing resolution and hence large number of nodes and branches, the resulting interconnections can easily become cumbersome and even impossible to describe and manipulate without the aid of computers. Hence, a systematic approach to modeling is crucial to allow the engineer to truly concentrate on model validation rather than error-prone, laborious technicalities in generating the model.

For heat transfer models based on conduction, convection and radiation characteristics of components, the resulting nonlinear dynamic system equations can easily be described in terms of node equations. This approach relies on three concepts: graph. branch characteristics and a conservation law. Specifically, a graph is a collection of nodes and directed branches connecting these nodes. It is a means of formalizing the interactions based on the physical properties of the subsystems and incorporating prior knowledge/observations about the model.

In our work to date we have developed software which accepts as inputs the parameters associated with finite element thermal models. An additional feature of the software is that uncertainties in the model are directly associated with uncertain physical system parameters, e.g., thermal coefficients. The uncertain parameters can be either identified or used in a robust control design.

## 2.2 Robust thermal control

Rapid thermal processing of semiconductor wafers demands fast tracking control laws that achieve near uniform spatial temperature distributions. In order to ensure the final product quality, it is essential to maintain a uniform temperature profile despite uncertainties in both transient and steady-state phases of the process. We have developed solutions and associated software for the design of robust thermal control and optimization. For a given operating trajectory, the control problem is posed as determining a feedforward/feedback controller that minimizes the worst-case peak deviation of the performance variables subject to a particular class of bounded disturbances and parameter variations. Since the solution to this nonlinear problem is not known, a sequence of approximations in terms of the small-signal equivalents are used to pose linear control problems. A complete solution to the associated linear control design problem is derived. Considering problem sizes of interest, efficient computational solution methods were investigated and prototype tools developed to simplify repetitive performance/robustness tradeoff studies, as well as determining sensor/actuator locations and operating points.

Specifically, (1) we have implemented efficient methods of calculating optimal controller gains and performance/robustness tradeoffs; (2) we have developed a graphical user-interface that simplifies the design of these controllers, in such a way that the theoretical details are transparent to the user.

The results have taken the trial/error studies performed on the physical system to a simulation level study where worst-case performance can be quantified. Apart from the obvious advantages of having a representative off-line model of the real system, we have brought a systematic approach to quantify the following factors:

- best nominal uniformity at steady-state,
- best worst-case uniformity at steady-state, and
- best nominal transient uniformity during ramp-ups between steady-states.

This systematic approach of associating performance limitations with a particular chamber model has allowed us to bring answers to crucial questions involving physical chamber-design and implications on achievable closed-loop performance.

The results are dewcribed in detail in the following papers which are included in the appendix:

1. Robert L. Kosut and M. Güntekin Kabuli, "Robust control of thermal processes: Static performance," *Proceedings of the 2nd International Rapid Thermal Processing Conference*, pp. 296–297, Monterey, California, September 1994.
2. M. Güntekin Kabuli, Robert L. Kosut and Stephen P. Boyd, "Improving static performance robustness of thermal processes," *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 62–66, Lake Buena Vista, Florida, December 1994.
3. A. Emami-Naeini, M. G. Kabuli and R. L. Kosut, "Finite-Time Tracking with Actuator Saturation: Application to RTP Temperature Trajectory Following," *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 73–78, Lake Buena Vista, Florida, December 1994.

4. Robert L. Kosut and M. Güntekin Kabuli, "On Actuator-Sensor Selection in Thermal Processes," *Proceedings of the 34th IEEE Conference on Decision and Control*. New Orleans, Louisiana, December 13–15. 1995.

5. Robert L. Kosut and M. Güntekin Kabuli, "On operating point sensitivity of thermal processes," *Proc. 1996 Triennial IFAC World Congress*, San Francisco, California USA. July 1-5 1996.

## 2.3 Model Reduction

We investigated the use of the proper orthogonal decomposition (POD) method for model reduction which was used originally for approximating turbulent phenomena. There seems to be no end to the number of times it has been re-discovered, *e.g.*, it appears in work on weather prediction. and in pattern recognition where it is known as the Karhunen-Loeve expansion. More recently it has been used to reduce the dimensionality of the ODEs obtained from finite element analysis. In summary, the POD can be used iteratively to find a parsimonious set of basis functions for the finite element analysis.

The basic premise of the POD is the orthogonal decomposition of the spatial covariance of the instantaneous spatial solution profiles of a PDE – "snapshots" as they are often referred. In the context of time-dependent and in particular turbulent hydrodynamics, the POD method was used on spatial velocity correlations to identify coherent spatial structures. A combination of this hierarchy of structures with a Galerkin weighted residual discretization of the fundamental model equations provides a spatially and temporally accurate model of the PDE dynamics, provided that a sufficient number of modes has been retained. There is increasing evidence, through the study of several model problems. that this methodology can be a crucial engineering algorithmic tool for the reduction, analysis. design and control of distributed systems. beyond the context of fluid flow.

The POD method is quite general and essentially relies on the singular value decomposition. It works directly on the state, and does not distinguish outputs as do balanced/truncation methods involving Hankel singular values. Our experience with POD was successful for reducing high order finite element models of rapid thermal processing systems as described in the following paper which is included in the appendix:

H. Aling, R.L. Kosut, A. Emami-Naeini. and J. L. Ebert. "Nonlinear model reduction with application to rapid thermal processing." *Proc. 35th IEEE CDC.* pp. 4305-4310. Kobe. Japan. Dec. 1996.

## 2.4 Feedforward learning control

For RTP, the ability to *quickly* manipulate wafer temperature according to the commanded temperature profile is crucial. Sensor-based feedback can certainly improve the RTP reactor's temperature following capability. maintain tight temperature control at steady state, and reduce the effects due to equipment variations. However, the bandwidth of feedback control must be balanced with stability considerations which are often limited by the process characteristics such as time delay.

Feedforward control, on the other hand, can complement feedback control performance by promoting non-delay and anticipatory actions which, when properly designed. can lead to superior tracking or disturbance rejection. Combining feedback and feddforward control should lead to a robust. stable and yet agile temperature control system.

Traditional feedforward design is usually based on analytical methods that require fairly accurate modeling of the process and the FB control loop. Such knowledge is often not available or is subject to change overtime. To overcome this problem we have developed a method for using run-to-run data to modify a feedforward control signal. The result, which is principally useful for *repetitive* run-to-run resipes. is a learning algorithm. The approach has its roots in disturbance rejection of acoustic emmissions where the disturbance is uaually measured with a microphone. Our application is in the context of feedback control. The specific approach is designed for ease of tuning. as it can *learn* from the past experiences. Many manufacturing tasks are repetitive task-oriented and are potential applications.

The feedforward learning control approach is described in detail in the following publications which are included in the appendix:

1. K.M. Tao, R.L. Kosut and G. Aral, "Learning Feedforward Control," *Proc. American Control Conference*, Baltimore. MD, June - July 1994, pp. 2575 - 2579.
2. K.M. Tao, G. Aral, R.L. Kosut and M. Ekblad, "Feedforward Learning Methods in RTP Temperature Control," *Proc. 2nd. Int. Rapid Thermal Processing Conf.*, Monterey, CA, Sept. 1994, pp. 278 - 282.
3. K.M. Tao, R.L. Kosut, M. Ekblad and G. Aral, "Feedforward Learning Applied to RTP of Semiconductor Wafers," *Proc. 33rd IEEE Conf. Decision and Control*. pp. 67–72, Lake Buena Vista, FL, December 1994.
4. K.M. Tao. R.L. Kosut and M. Ekblad, "Feedforward Learning – Nonlinear Processes and Adaptation," *Proc. 33rd IEEE Conf. Decision and Control*. pp. 1060–1065, Lake Buena Vista. FL, December 1994.

## 3 Transitions

The feasibility of many of the techniques developed under this contract for modeling. control design, and optimization were successfully tested on one or more of Applied Materials RTP chambers.

# A Publications

## A.1 Robust control of thermal processes: Static performance

- Robert L. Kosut and M. Güntekin Kabuli, "Robust control of thermal processes: Static performance," *Proceedings of the 2nd International Rapid Thermal Processing Conference*, pp. 296–297, Monterey, California, September 1994.

Robert L. Kosut  M. Güntekin Kabuli
Integrated Systems Inc
3260 Jay Street, Santa Clara, CA 95054-3309

In this paper the steady-state sensitivity of static nonlinear heat transfer models subject to static feedforward/feedback control laws is considered. For a given operating point, the desired control problem is posed as determining a feedforward/feedback static controller that minimizes the worst-case peak deviation of the performance variables about a nominal point subject to a particular class of bounded disturbances and parameter variations. Since the solution to this nonlinear problem is not known, a sequence of approximations in terms of the small-signal equivalents are used to pose static linear control problems. A complete solution to the associated static linear control design problem is derived. Considering problem sizes of interest, efficient computational solution methods are investigated and prototype tools are developed to simplify repetitive performance/robustness tradeoff studies, as well as determining sensor/actuator locations and operating points.

## Introduction

Rapid thermal processing of semiconductor wafers demands fast tracking control laws that achieve near uniform spatial temperature distributions. In order to ensure the final product quality, it is essential to maintain a uniform temperature profile despite uncertainties in both transient and steady-state phases of the process.

In this paper we focus exclusively on the steady-state sensitivity of static nonlinear heat transfer models subject to static feedforward/feedback control laws. The approach relies on a static nonlinear heat transfer model, typically obtained by forming a mesh of branches that model conduction, convection and radiation between the nodes of the mesh. A systematic modeling approach based on the analysis of large scale nonlinear resistive networks is applied to obtain the equations that determine the operating points in terms of input and disturbance biases. For a more detailed treatment of the associated concepts utilized, see e.g., [1] for the circuit-theoretic approach and [2] for heat transfer topics. For a given operating point, the desired control problem is posed as determining a feedforward/feedback static controller that minimizes the worst-case peak deviation of the performance variables about a nominal point subject to a particular class of bounded disturbances and parameter variations. Since the solution to this nonlinear problem is not known, a sequence of approximations in terms of the small-signal equivalents are used to pose static linear control problems.

## Problem Description

Consider the feedback system shown in Figure 1 , where

- $w \in \mathbb{R}^{n_w}$ are the exogenous inputs consisting of both static disturbances and uncertain physical parameters,
- $z \in \mathbb{R}^{n_z}$ are the performance variables (not necessarily those that are sensed) such as critical temperatures and lamp (actuator) power levels,
- $y \in \mathbb{R}^{n_y}$ are the sensed variables, e.g., temperature readings from various sensors,
- $u \in \mathbb{R}^{n_u}$ are the actuator inputs, e.g., lamp powers.

The static linear plant $P \in \mathbb{R}^{(n_z+n_y)\times(n_w+n_u)}$ is obtained by linearization about a particular equilibrium or operating point. The actuator input consists of a static feedforward control $u_K \in \mathbb{R}^{n_u}$ and a static feedback $Ky$ with $K \in \mathbb{R}^{n_u \times n_y}$ .

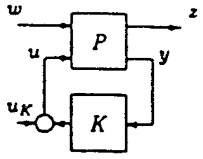


Figure 1: Static feedback system

It is convenient to partition $P$ as follows:

$$\begin{bmatrix} z \\ y \end{bmatrix} = P \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad .$$

Let $w_0 \in \mathbb{R}^{n_w}$ and $z_0 \in \mathbb{R}^{n_z}$ denote, respectively, nominal values of disturbance and performance variables. Let $\Delta_w$ and $\Delta_z$ be nonsingular diagonal scaling matrices such that
$w = w_0 + \Delta_w \eta_w$ and $z = z_0 + \Delta_z \eta_z$ .
The scaling matrices are chosen to reflect the relative size of uncertainty in the $w$-variables and the relative size of performance tolerance in the $z$-variables. The $\eta$-variables reflect uncertainty ( $\|\eta_w\|_\infty \le \lambda_w$ ) and performance ( $\|\eta_z\|_\infty \le \lambda_z$ ) . The norm $\|\cdot\|_\infty$ is defined as $\|x\|_\infty = \max_i |x_i|$ . For a given $\lambda_w$ and $\lambda_z$ , a design $(K, u_K)$ is said to be
$(\lambda_w, \lambda_z)$–feasible iff $\|\eta_z\|_\infty \le \lambda_z$ for all $\|\eta_w\|_\infty \le \lambda_w$ .

The specific design problem under consideration is the following:

- For a given $P$, $\lambda_w$, $w_0$, $z_0$ and nonsingular $\Delta_w$, $\Delta_z$ , determine arg $\min_{(K,u_K)} \max_{\|\eta_w\|_\infty \le \lambda_w} \|\eta_z\|_\infty$ . (1)

Due to the two free parameters $K$ and $u_K$ , one can pose four possible cases from problem (1) with appropriate restrictions on the parameters, namely:

1. open-loop ($u = 0$),
2. feedforward only ($u = u_K$),
3. feedback only ($u = Ky$) and
4. feedforward/feedback ($u = u_K + Ky$) .

Typical design specifications involve the following:

1. feasibility: determine (if possible) a $(\lambda_w, \lambda_z)$–feasible design, or
2. robustness: for a fixed $\lambda_z$, maximize $\lambda_w$ among $(\lambda_w, \lambda_z)$–feasible designs, or
3. performance: for a fixed $\lambda_w$, minimize $\lambda_z$ among $(\lambda_w, \lambda_z)$–feasible designs.

The feasibility, robustness and performance design problems stated above can all be expressed in terms of problem (1) by considering the tradeoff curve denoted by the graph

Research supported by ARPA under AFOSR contract F49620-94-C-0003 and Army Missile Command contract DAAH01-93-C-R193.

$$\left\{ \left( \lambda_w \, , \, \min_{(K,u_K)} \max_{\|\eta_w\|_\infty \le \lambda_w} \|\eta_z\|_\infty \right) \mid \lambda_w \ge 0 \right\} \qquad (2)$$

The graph in (2) denotes the boundary of feasible and infeasible designs on the $(\lambda_w, \lambda_z)$−plane. In general, the graph that partitions the feasible and infeasible regions is neither convex nor concave. In the following section, we describe a solution method for problem (1) and hence a method for deriving the graph in (2) .

## Solution

The general form of the solution to obtain the graph in (2) is given in the Appendix. The results can be derived using the static form of the Youla parametrization of all feasible controllers[5] . The details will be reported elsewhere.

For our purposes here, all possible four cases reduce to a problem of the form $\min_X \|T_1 + T_2 X T_3\|_{i,\infty}$ , where $\|\cdot\|_{i,\infty}$ denotes the matrix norm induced by the vector norm $\|\cdot\|_\infty$ . Efficient solution methods based on interior point methods are used to solve the associated linear program. In the process, $(n_u(n_y+1))$ control variables are sought by introducing $(n_z(n_w+1)+1)$ more slack variables. Typically $n_z n_w \gg n_u n_y$ ; hence, special structure and/or conditioning has to be utilized in order not to be burdened by the increase in the parameter dimension. The results in the following section are obtained by prototype optimization tools developed in MATRIX$_X$ based on the methods in [3] and [4] .

## Design Example

Consider the mesh in Figure 2. It represents a linear resistive network consisting of seven nodes and nine branches. All branch conductances are taken as unity; node 0 denotes the datum node (ambient temperature). Input and disturbance fluxes are denotes by $u$ and $w$ , respectively. The measured output is denoted by $y$ . Conservation equations written at each and every node except the datum node yield $y = P_{yw}w + P_{yu}u$ . For example, at node 5 , we have

$$w_2 = (T_5 - T_2) + (T_5 - T_4) + (T_5 - T_6) \quad ,$$

where $T$ denotes the node temperatures; conductances are unity for simplicity.

The description of the mesh (consisting of conduction, convection and radiation terms) and the derivation of the associated linearized model at the steady-state are all automated. For this example, the regulated variables are chosen as the six node temperatures. Hence, $n_w = 2$ , $n_u = 2$ , $n_z = 6$ and $n_y = 2$ . Thus $P \in \mathbb{R}^{8\times4}$ , $K \in \mathbb{R}^{2\times2}$ and $u_K \in \mathbb{R}^2$ . We seek at most $n_u(n_y+1) = 6$ control variables and introduce $n_z(n_w+1)+1 = 7$ slack variables.

The tradeoff curve in (2) is derived for five cases (see Figure 3) . One extra design approach is introduced to illustrate that the feedforward and feedback designs are not decoupled. In other words, minimizing over $u_K$ for $K = 0$ and then fixing the optimal $u_K$ value and then optimizing over $K$ is not equivalent to the simultaneous minimization over $u_K$ and $K$ .

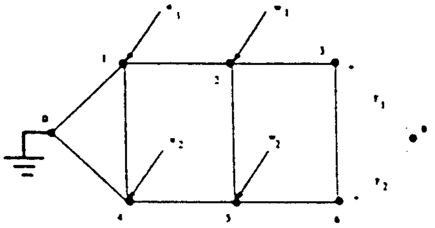


Figure 2: Sample mesh

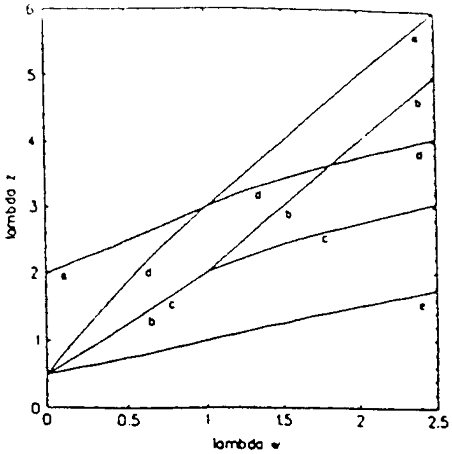


Figure 3: Tradeoff curves in (2) for the five cases:

a - open-loop
b - feedforward only
c - one at a time optimization: first feedforward and then feedback
d - feedback only
e - simultaneous optimization: feedforward and feedback

## Conclusions

Using the complete solution to the static linear control design problem, performance limitations are derived. Repetitive performance/robustness tradeoff studies can be performed easily for different sensor/actuator locations and operating points. The tools are specifically geared towards large scale resistive network problems. The problem solved here is used to illustrate the tool and methodology.

## References

[1] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, 1975.

[2] F. P. Incropera and D. P. DeWitt, *Introduction to Heat Transfer*, John Wiley & Sons, 1985.

[3] Yu. Nesterov and A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, 1993.

[4] L. Vandenberghe and S. Boyd, "Primal-dual potential reduction method for problems involving matrix inequalities," submitted to Math. Programming, 1993.

[5] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*, MIT Press, 1985.

## Appendix

Given $P$, $w_0$, $\lambda_w$, $z_0$ and nonsingular $\Delta_w$, $\Delta_z$ , it can be shown that:

$$(K, u_K) \;=\; \arg \min_{(K,u_K)} \max_{\|\eta_w\|_\infty \le \lambda_w} \|\eta_z\|_\infty \quad \text{and} \quad \det(I - P_{yu}K) \neq 0$$

if and only if

$$(K, u_K) \;=\; (Q(I + P_{yu}Q)^{-1} \, , \, (I + QP_{yu})^{-1}u_Q)$$

where

$$\begin{aligned} (Q, u_Q) &= \arg\min \| [\ \lambda_w A(Q) \;\; b(Q, u_Q)\ ] \|_{i,\infty} \\ A(Q) &= \Delta_z^{-1}(P_{zw} + P_{zu}QP_{yw})\Delta_w \\ b(Q, u_Q) &= \Delta_z^{-1}((P_{zw} + P_{zu}QP_{yw})w_0 - z_0 + P_{zu}u_Q) \\ \det(I + P_{yu}Q) &\neq 0 \end{aligned}$$

## A.2 Improving static performance robustness of thermal processes

- M. Güntekin Kabuli, Robert L. Kosut and Stephen P. Boyd, "Improving static performance robustness of thermal processes," *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 62–66, Lake Buena Vista, Florida, December 1994.

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

# Improving Static Performance Robustness of Thermal Processes*

M. Güntekin Kabuli and Robert L. Kosut
Integrated Systems Inc.
3260 Jay Street, Santa Clara, CA 95054-3309
kabuli@isi.com, kosut@isi.com

Stephen Boyd
Information Systems Laboratory
Stanford University, Stanford, CA 94305
boyd@isl.stanford.edu

### Abstract

A static (steady-state) robust control design problem is considered using a nonlinear model of a thermal system. For a given operating point, the control problem is to determine a feedforward/feedback static controller that minimizes the worst-case static peak performance deviation from nominal in the presence of bounded disturbances and parameter variations. It is desired to obtain the tradeoff between the size of the worst-case deviation and the size of the uncertainty set. A complete solution is derived for the static linear control design problem obtained from linearization about selected operating points. Efficient computational tools are developed to rapidly analyze numerous operating points and control configurations.

## 1 Introduction

Rapid thermal processing (RTP) systems demand fast tracking control laws that achieve near uniform spatial temperature distributions across the target, e.g., a semiconductor wafer, during both transient and steady-state phases of the process.

In this paper we only address the static (steady-state) problem using static feedforward/feedback control laws. The approach relies on a static nonlinear heat transfer model which includes parameter uncertainty. The form of the model found to be very convenient for robust control design is obtained by forming a mesh of branches that model conduction, convection and radiation between the nodes of the mesh. A systematic modeling approach based on the analysis of large scale nonlinear resistive networks can then be applied to obtain the equations that determine the operating points in terms of input and disturbance biases. This model structure is generic, since all thermal system models can be put in this form [1, 2] .

For a given operating point, the control problem is posed as designing a feedforward/feedback static controller that minimizes the worst-case peak deviation of the performance variables from a nominal point when subjected to bounded disturbances and parameter variations. Since the solution to this nonlinear problem is not known, a sequence of approximations in terms of the small-signal equivalents are used to pose static linear control problems. We derive a complete solution to the associated static linear control design problem. Considering problem sizes of interest, (e.g., 20 actuators, 20 sensors, 100 regulated variables, 100 exogenous disturbances) , efficient computational solution methods are investigated and prototype tools are developed to simplify comparative design studies resulting from different choices of operating points, actuators, sensors and control laws (feedforward and/or feedback).

The paper is organized as follows: in Section 2 we pose and solve the static linear *sensitivity* problem, i.e., parameter uncertainties are included as additional exogenous input perturbations. The tools required here are also needed for the robustness problem. An example of the sensitivity tradeoffs using the thermal mesh model is given in Section 3. Robustness results for real parametric uncertainties are given in Section 4. An example, using the developed tools, is given in Section 5. To conserve space, only a very brief discussion of the computational issues and methods is provided. Further details on the optimization methods and proofs can be obtained from the authors.

## 2 Sensitivity Tradeoff

### 2.1 Problem Description

Consider the feedback interconnection shown in Figure 1 , where $w$, $z$, $u$ and $y$ denote the exogenous inputs, controlled outputs, actuator inputs and measured outputs, respectively; $P \in \mathbb{R}^{(n_z+n_y)\times(n_w+n_u)}$ denotes a static linear plant, $K \in \mathbb{R}^{n_u\times n_y}$ denotes a static linear feedback controller. and $u_K \in \mathbb{R}^{n_u}$ denotes the static feedforward control.

*Research supported by ARPA under AFOSR contract F49620-94-C-0003 and Army Missile Command contract DAAH01-93-C-R193.

0-7803-1968-0/94$4.00©1994 IEEE

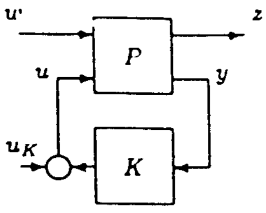


Figure 1: Static feedback system

To address the sensitivity to parameter variation, the exogenous input $w$ includes disturbances as well as parameter perturbations from nominal. As such it is convenient to explicitly account for the nominal as well as deviations. Hence, let the normalized exogenous input be given by

$$\eta_w = \Delta_w^{-1}(w - w_0)$$

where $w_0 \in \mathbb{R}^{n_w}$ is the nominal value and $\Delta_w$ is a diagonal scaling matrix. A normalized output $\eta_z$ is defined accordingly:

$$\eta_z = \Delta_z^{-1}(z - z_0) \ .$$

Motivated by the temperature uniformity requirements of RTP problems, we are naturally led to consider the "infinity" norm as the appropriate measure of signal size. Thus, for $x \in \mathbb{R}^n$ , the infinity norm is defined as $||x||_\infty = \max_{1 \le k \le n} |x_k|$ . Recall also that the induced matrix norm is the "max-row-sum", i.e.,

$$||A||_{i,\infty} = \max_{||x||_\infty \le 1} ||Ax||_\infty = \max_i \sum_j |a_{ij}| \ .$$

We can now state the fundamental design problem:

- **Optimal Design Tradeoff:** For a given $P$, $w_0$, $z_0$ , $\Delta_w$, and $\Delta_z$, find the optimal tradeoff between disturbance size $||\eta_w||_\infty$ and performance tolerance $||\eta_z||_\infty$, i.e., determine the graph:

$$\{(\lambda_w \ , \lambda_z) \mid \lambda_w \ge 0\}$$
$$\lambda_z = \min_{(K,u_K)} \max_{||\eta_w||_\infty \le \lambda_w} ||\eta_z||_\infty \qquad (1)$$

The graph in (1) of $\lambda_z$ vs. $\lambda_w$ gives the minimum relative change ($\lambda_z$) uniformly in all performance variables ($z$) for a relative uniform change ($\lambda_w$) in all exogenous input variables ($w$). Hence, the graph denotes the *boundary* between feasible and infeasible designs on the $(\lambda_w, \lambda_z)-$ plane. A typical tradeoff curve - the graph described in (1) - is shown by the solid line in Figure 2. The shaded region below this tradeoff curve is infeasible, i.e., there exists no combination of feedforward or feedback which can achieve the requested performance. Conversely, all $(1,1)-$feasible designs correspond to points on the segment between designs A and B in Figure 2 . Note that A corresponds to a performance design and C corresponds to a robust design, i.e., design C allows a much larger uncertainty for the requested specification $||\eta_z||_\infty \le \lambda_z = 1$. In general, the graph that partitions the feasible and infeasible regions is neither convex nor concave.

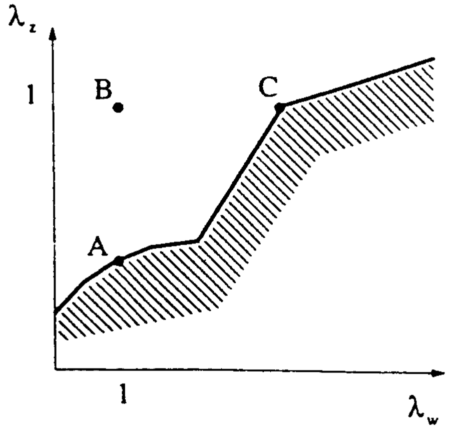


Figure 2: Graph in (1) denoting the boundary of the feasible region in the $(\lambda_w,\lambda_z)$-plane; shaded region is infeasible.

Due to the two free design parameters $K$ and $u_K$ , four possible problems can be posed, namely:

- open-loop ($u = 0$),
- feedforward only ($u = u_K$),
- feedback only ($u = Ky$),
- feedforward/feedback ($u = u_K + Ky$) .

Typical design specifications involve the following:

- feasibility: determine (if possible) a $(\lambda_w, \lambda_z)-$feasible design
- robustness: for a fixed $\lambda_z$, maximize $\lambda_w$ among $(\lambda_w, \lambda_z)-$feasible designs
- performance: for a fixed $\lambda_w$, minimize $\lambda_z$ among $(\lambda_w, \lambda_z)-$feasible designs.

The feasibility, robustness and performance design problems stated above can all be expressed in terms of the tradeoff curve denoted by the graph in (1) . In fact, it can be shown that the four possible choices of control ($u = 0$ , $u = Ky$ , $u = u_K$ , $u = u_K + Ky$ ) result in special cases of finding a solution to a problem of a max-row-sum norm minimization of the form:

$$\min_X ||T_1 + T_2 X T_3||_{i,\infty} \ . \qquad (2)$$

where the $T$'s and $X$ are defined in accordance with the four cases.

## 2.2 Positive Definite Programming

To efficiently solve the above max-row-sum problem we utilize the primal-dual potential reduction methods described in [4]. The computational tools we have developed resolve the following difficulties: 1) Initialization of primal and dual variables ; 2) Efficient approximate solutions to the (huge) least-squares problems to determine the analytic center ; 3) Perturbation of the updated dual parameters. To further explain these very important but esoteric issues is outside the scope for this paper. Interested readers can request details from the authors. For the intended applications $(n_z n_w) \gg (n_u n_y) \gg 1$ . The particular solution approach reduces the original $(n_z n_w + n_w + n_u n_y + n_u + 1)$-unknown least-squares problem to a $(n_u n_y + n_u)$-unknown least-squares problem. Hence the computational complexity is determined by the control variables: $(n_u n_y)$ for feedback and $n_u$ for feedforward.

# 3 Example: Sensitivity Tradeoff

The mesh in Figure 3 represents a resistive network consisting of seven nodes and nine branches. The mesh describes conduction, convection and radiation effects as non-linear resistive elements. (The authors have used an *Xmath* script to automate the mesh generation and the associated linearized model at the steady-state.) Following standard node analysis results for linear resistive networks (see e.g. [1]) , the steady-state heat-flux conservation equations arising from application of the Kirchhoff Current Law at each and every node (except the reference (datum) node) result in:

$$0 = A_c G A_c^T x + A_u u + A_w w, \quad y = Cx \quad . \qquad (3)$$

The node variables $x$ correspond to node temperature minus the ambient temperature, $u$ denotes the control input fluxes, and $w$ the disturbance fluxes. The measured node temperatures are denoted by $y$ . The matrix $[A_c \ A_u \ A_w]$ is defined by the incidence matrix that describes the interconnection of branches; its entries are 0's, 1's or $-1$'s . The matrix $G$ is diagonal consisting of nominal branch conductances. For this example, the regulated variables are chosen as the six node temperatures. Hence, $n_w = 2$ , $n_u = 2$ , $n_z = 6$ and $n_y = 2$ . Thus $P \in \mathbb{R}^{8\times4}$ , $K \in \mathbb{R}^{2\times2}$ and $u_K \in \mathbb{R}^2$ . We seek at most $n_u(n_y + 1) = 6$ control variables and introduce $n_z(n_w + 1) + 1 = 19$ slack variables. Following the notation in Section 2 , let $w_{0,i} = 1, i = 1,2$ ; $\Delta_w = I$ ; $z_{0,i} = 3, i = 1,\ldots,6$ ; $\Delta_z = I$ .

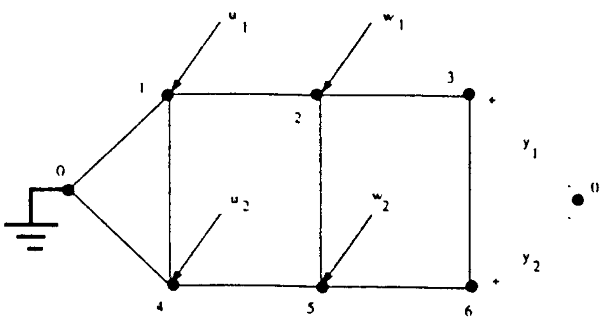


Figure 3: Sample mesh

The tradeoff curve in (1) is derived for five cases (see Figure 4) . One extra design approach is introduced to illustrate that the feedforward and feedback designs are not decoupled. In other words, minimizing over $u_K$ for $K = 0$ and then fixing the optimal $u_K$ value and then optimizing over $K$ is not equivalent to the simultaneous minimization over $u_K$ and $K$ .

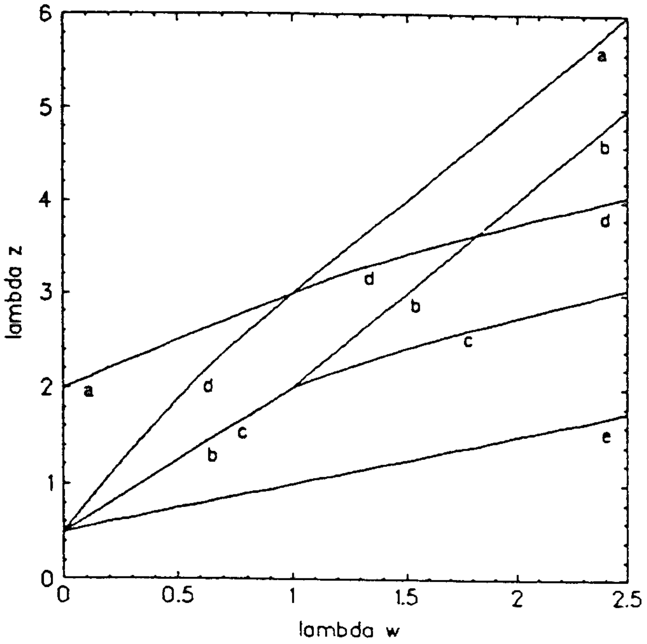


Figure 4: Tradeoff curves in (1) for the five cases:
a – open-loop
b – feedforward only
c – one at a time optimization: first feedforward and then feedback
d – feedback only
e – simultaneous optimization: feedforward and feedback

## 4 Robustness Analysis

The control design problem formulated in Section 2 is based on a nominal plant model $P : (w, u) \mapsto (z, y)$ , where $w$ includes perturbations in uncertain parameters. In this section, we consider the performance of the nominal control $u = u_K + Ky$ subject to perturbed plant models as shown in Figure 5 , where $w$ now includes only exogenous disturbances.

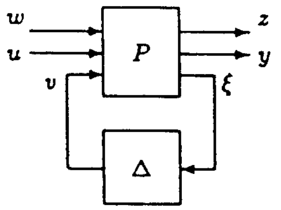


Figure 5: Perturbed plant model

Although we still consider linearized plant models, the uncertainty is maintained in its natural form (Figure 5) . In the example to follow we take $\Delta$ as a diagonal matrix whose entries correspond to uncertainties in the branch conductance matrix $G$ . Hence, the perturbed form of the linear resistive network (3) becomes:

$$0 = A_c(G + \Delta)A_c^T x + A_u u + A_w w \quad . \qquad (4)$$

In order to define the uncertainty set, let $e_i$ denote the $i$th standard basis vector. Its dimension is determined from the context. Now, the uncertainty structure and vertex set are defined as follows:

$$\mathcal{D} = \{\Delta \in \mathbb{R}^{n_v \times n_\xi} \mid \Delta_l \leq \Delta \leq \Delta_u\} \quad , \qquad (5)$$

$$\mathcal{V}_\mathcal{D} = \{\Delta \in \mathcal{D} \quad | \quad e_i^T \Delta e_j \in \{e_i^T \Delta_l e_j, e_i^T \Delta_u e_j\}, \quad 1 \leq i \leq n_v \; , \; 1 \leq j \leq n_\xi \; \} \quad .(6)$$

Let $H_{zw}(\Delta)$ denote the fractional form obtained for $u = Ky$ (for $u = u_K + Ky$ , $w$ gets augmented by one entry) . From Figure 5,

$$H_{zw}(\Delta) = H_{11} + H_{12}\Delta(I - H_{22}\Delta)^{-1}H_{21} \quad .$$

Let $\det(I - H_{22}\Delta) \neq 0$ for all $\Delta \in \mathcal{D}$ . Under these assumptions, it can be shown that:

$$\max_{\Delta \in \mathcal{D}} \|H_{zw}(\Delta)\|_{i,\infty} \; = \; \max_{\Delta \in \mathcal{V}_\mathcal{D}} \|H_{zw}(\Delta)\|_{i,\infty} \quad . \qquad (7)$$

This result means that the worst-case maximum row sum of the linear fractional form $H_{zw}(\Delta)$ over $\mathcal{D}$ is achieved at the vertex set $\mathcal{V}_\mathcal{D}$ . We utilize this result in the following section, in a design example. It can also be shown that $\min_{\Delta \in \mathcal{D}} \|H_{zw}(\Delta)\|_{i,\infty}$ is not necessarily achieved at the vertex set.

## 5 Example: Robustness Analysis

We consider a mesh consisting of 40 linear conductance branches and 25 temperature nodes. As in (3) , $[A_c, A_u, A_w] \in \mathbb{R}^{24 \times (40+5+3)}$ is the incidence matrix and consists of 0's, 1's and $-1$'s to denote the directed graph associated with the mesh. $u \in \mathbb{R}^5$ and $w \in \mathbb{R}^3$ denote the control input and exogenous input fluxes, respectively. $y \in \mathbb{R}^5$ denotes the measured node temperatures. The node temperatures satisfy (3) where $G = \text{diag}(g)$ . Nominal operating conditions (denoted by 0 subscripts) are determined to minimize the deviation from a uniform temperature profile across the measured nodes. The nominal operation is determined by

$$A_c \,\text{diag}(g_0)\; A_c^T x_0 + A_u u_0 + A_w w_0 = 0 \quad ,$$

where $g_0 > 0$ and $w_0 > 0$ . Let the performance variable $z$ be defined as $z = [y^T \; u^T]^T$ , where $z_0 > 0$ . The design problem is posed as follows: determine $u = u_K + Ky$ such that the performance measure

$$\varphi(u_K, K) = \max_{\substack{\|\Delta_w^{-1}(w - w_0)\|_\infty \leq 1 \\ \|\Delta_G^{-1}(g - g_0)\|_\infty \leq 1}} \|\Delta_z^{-1}(z - z_0)\|_\infty$$

is minimized for the feedback perturbed plant model in Figure 5 ; we choose $\Delta_w = \text{diag}(w_0)$ , $\Delta_G = 0.1\text{diag}(g_0)$ and $\Delta_z = \text{diag}(z_0)$ . Recall that by (7) , the performance measure is achieved at the vertices. Note that we allow a 10% uncertainty in all 40 conductance branches. Hence, for a given design, the exact measure could have been obtained by $2^{40}$ max row sum evaluations. To make the calculations reasonable, we will rate the designs according to 10% uncertainty in the *first* 10 conductance branches; hence determine a lower bound on $\varphi$ (but this is exact if only 10 parameters are varied) .

In the first design approach, we take $G = G_0$ and solve

$$(u_{K_1}, K_1) = \arg \min_{(u_K,K)} \max_{\substack{\|\Delta_w^{-1}(w - w_0)\|_\infty \leq 1 \\ G = G_0}} \|\Delta_z^{-1}(z - z_0)\|_\infty \, .$$

Thus, the control law is optimal under perfect plant modeling.

In the second design approach, the first order approximation to the plant

$$\begin{aligned} A_c \,\text{diag}(g_0)\; A_c^T(x - x_0) + A_u(u - u_0) + A_w(w - w_0) + \\ A_c \,\text{diag}(g - g_0)\; A_c^T x_0 \;=\; 0 \end{aligned} \qquad (8)$$

is used to solve for the optimal sensitivity controller

$$(u_{K_2}, K_2) = \arg \min_{(u_K,K)} \max_{\substack{\|\Delta_w^{-1}(w - w_0)\|_\infty \leq 1 \\ \|\Delta_G^{-1}(g - g_0)\|_\infty \leq 1}} \|\Delta_z^{-1}(z - z_0)\|_\infty \, .$$

In other words, the input to the $\Delta$ block in Figure 5 is fixed at $\xi_0 = A_c^T x_0$ , and the exogenous inputs $w$ are augmented by 40 more entries to account for $(g - g_0)$ in (8). The results are summarized in Table 1.

| | $g = g_0$ | $0.9g_0 \leq g \leq 1.1g_0$ |
|---|---|---|
| $\varphi(u_0, 0)$ | 6.99% | $\geq$ 13.37% |
| $\varphi(u_{K_1}, K_1)$ | 3.47% | $\geq$ 223.45% |
| $\varphi(u_{K_2}, K_2)$ | 3.67% | $\geq$ 6.63% |

Table 1: Performance ratings

Note that the first row in Table 1 corresponds to the openloop performance. The input is set to $u_0$ ; hence the performance measure reflects the relative change in $y$ about $y_0$ . At the nominal conductance parameters, worst-case deviation is 6.99% and when the first 10 conductances have 10% uncertainty, worst-case deviation is 13.37% . Note that the second column of Table 1 is a lower bound on the performance measure $\varphi$ since the parameter perturbations are restricted to the first ten branches, only. Recall that the first design did not take into account any parametric uncertainty. Hence the nominal performance is better than the open-loop; however, 10% parametric uncertainty can cause a deviation more than twice the nominal $x_0$ . In the second design, by augmenting the exogenous inputs $w$ with the first-order effect of the parametric changes, a more cautious nominal design is achieved. With uncertainty in the first ten parameters, the worst-case deviation is now half of open-loop deviation, although the nominal performance is slightly worse than the first nominal design.

## 6 Further Robustness Analysis

A little more notation is needed for this section. For $A \in \mathbb{R}^{n \times n}$ , $\rho_{\mathbb{R}}(A)$ denotes the maximum absolute value of the real eigenvalues of $A$ . For a real matrix $A$ , $\mid A \mid$ denotes the absolute value of $A$ , i.e., $e_i^T \mid A \mid e_j = \mid e_i^T A e_j \mid$ . $\Pi(A) = \rho_{\mathbb{R}}(\mid A \mid) = \rho(\mid A \mid)$ ; also referred to as the Perron eigenvalue of $A$ . Let $\mathbb{1} \in \mathbb{R}^n$ denote a vector of all 1's .

Let $H_{zw}(\Delta) = H_{11} + H_{12}\Delta(I - H_{22}\Delta)^{-1}H_{21}$ , where $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \in \mathbb{R}^{(n_z+n_\delta)\times(n_w+n_\delta)}$ and $\Delta$ is diagonal. Under these assumptions,

$$\begin{aligned}
\min_{\substack{\|H_{zw}(\Delta)\|_{i,\infty} \leq \gamma \\ \|\Delta\|_{i,\infty} \leq \frac{1}{\gamma} \\ \det(I - H_{22}\Delta) \neq 0}} \gamma
&= \max_{\substack{e \in \{e_1,\ldots,e_{n_z}\} \\ w \in \{-1,1\}^{n_w} \\ S \in \operatorname{diag}\{-1,1\}^{n_\delta}}} \rho_{\mathbb{R}}\left(\begin{bmatrix} e^T H_{11} w & e^T H_{12} \\ H_{21} w & H_{22} \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & S \end{bmatrix}\right) \\
&\leq \max_{\substack{e \in \{e_1,\ldots,e_{n_z}\} \\ w \in \{-1,1\}^{n_w}}} \Pi\left(\begin{bmatrix} e^T H_{11} w & e^T H_{12} \\ H_{21} w & H_{22} \end{bmatrix}\right) \\
&\leq \max_{1 \leq i \leq n_z} \Pi\left(\begin{bmatrix} e_i^T \mid H_{11} \mid \mathbb{1} & e_i^T H_{12} \\ \mid H_{21} \mid \mathbb{1} & H_{22} \end{bmatrix}\right) \\
&\leq \left\| \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \right\|_{i,\infty} .
\end{aligned}$$

A cheap lower bound to the optimal $\gamma$ value above can be obtained by evaluating $\rho_{\mathbb{R}}$ of a smaller number of matrices rather than the huge number $(n_z 2^{(n_w+n_\delta)})$ . Note that typically, $n_\delta \gg n_w$ . Using the Perron eigenvalues, coarser upper bounds on the optimal $\gamma$ can be obtained by $n_z 2^{n_w}$ and $n_z$ eigenvalue evaluations, respectively.

Note also that if the entries of $H$ are all positive, then the optimal $\gamma$ is given by

$$\min_{\substack{\|H_{zw}(\Delta)\|_{i,\infty} \leq \gamma \\ \|\Delta\|_{i,\infty} \leq \frac{1}{\gamma} \\ \det(I - H_{22}\Delta) \neq 0}} \gamma = \max_{1 \leq i \leq n_z} \Pi\left(\begin{bmatrix} e_i^T H_{11} \mathbb{1} & e_i^T H_{12} \\ H_{21} \mathbb{1} & H_{22} \end{bmatrix}\right).$$

## 7 Conclusion

The sensitivity and robustness to parameter uncertainty of operating points of thermal processes has been investigated using static feedforward/feedback control. The problem is motivated using a large-scale linear resistive network. Efficient computational tools are developed to handle a large number of nodes and branches. Successive design studies involving different operating points, actuator/sensor selections can be easily performed.

## References

[1] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, 1975.

[2] F. P. Incropera and D. P. DeWitt, *Introduction to Heat Transfer*, John Wiley & Sons, 1985.

[3] Yu. Nesterov and A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, 1993.

[4] L. Vandenberghe and S. Boyd, "Primal-dual potential reduction method for problems involving matrix inequalities," submitted to Math. Programming, 1993.

### A.3 Finite-Time Tracking with Actuator Saturation: Application to RTP Temperature Trajectory Following

- A. Emami-Naeini, M. G. Kabuli and R. L. Kosut, "Finite-Time Tracking with Actuator Saturation: Application to RTP Temperature Trajectory Following," *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 73–78, Lake Buena Vista, Florida, December 1994.

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

# Finite-Time Tracking with Actuator Saturation: Application to RTP Temperature Trajectory Following*

A. Emami-Naeini, M. G. Kabuli and R. L. Kosut
Integrated Systems, Inc.
3260 Jay Street, Santa Clara, CA 95054

## ABSTRACT

Precise trajectory following in the presence of actuator saturation constraints is important in performance of many control systems. An approximate finite-time tracking problem is formulated for a multivariable discrete-time linear time-invariant system. The actuator saturation constraints are taken into account explicitly. The problem is set up as a set of linear programming problems using a transfer matrix approach. The approach is applied to temperature profile tracking in rapid thermal processing (RTP) systems. It is known that fast and precise temperature tracking is crucial in performance of RTP systems. The actuator saturation manifests itself in terms of the power driving the lamps. Two rapid thermal processing examples are presented to illustrate the validity of the approach.

## 1 Introduction

Precise trajectory following is important in performance of many control systems . The problem of finite-time tracking in the presence of actuator saturation has been a generic problem in control theory. In servomechanisms, a velocity profile is to be followed precisely. In many thermal systems, a temperature profile is to be followed quickly and accurately. The performance in many of these systems is tied to how fast the desired trajectory can be followed. In almost all systems, actuators saturate because of limited dynamic range. For example, a valve saturates when it is completely open or closed. The control surfaces on an aircraft can be moved by a certain angle from their nominal positions. In thermal systems, heaters or lamps can be driven between minimum and maximum power settings. For example in rapid thermal processing (RTP), precise temperature trajectory following is crucial and the actuators (lamps) have a finite maximum power driving them. Furthermore, in these problems the minimum actuator setting is zero power so that the saturation nonlinearity is not symmetric. Therefore, the problem of fast tracking in the presence of actuator saturation has been a generic problem in control theory.

Kalman [1] was the first to realize that the problem can be addressed effectively if it is formulated in a discrete-time setting. Schmidt [2] solved the problem for low order single-input-single-output systems. This problem has been previously addressed by the present authors in reference [3] in a continuous-time setting with applications to control of flexible structures. Other related work are contained in [4], [5], [6]. In this paper, a solution to the finite-time tracking with actuator saturation is proposed for multivariable linear time-invariant discrete systems. The approach relies on the transfer matrix to formulate constraints on the set of admissible finite duration control signals that achieve precise point-to-point trajectory following. The problem is first formulated as an open-loop problem. The shape of the best input signal to achieve the finite-time tracking is derived. We then consider the closed-loop implementation of the problem so that the same control signal is produced at the input of the plant. The actuator saturation is taken into account explicitly. The problem is set up as a set of linear programming problems. This represents the first complete solution to the problem. The motivating example for the present research is fast temperature tracking in rapid thermal processing of semiconductor wafers. Therefore, a short description of RTP appears next before the problem formulation.

## 2 Rapid Thermal Processing

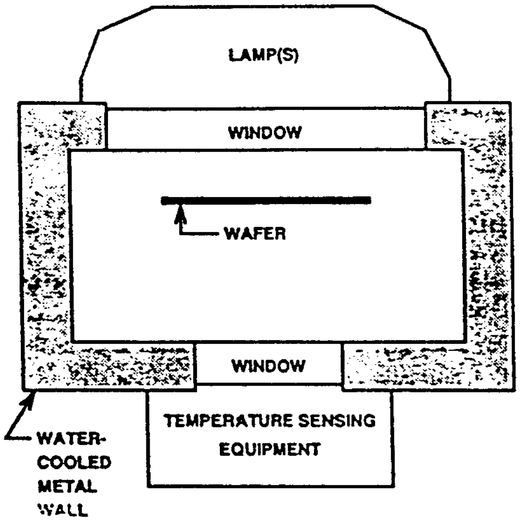
A variety of different steps are involved in semiconductor microelectronics manufacturing. These steps include oxidation, lithography, epitaxial film growth (epi), annealing, CVD, etc. Each of these steps is a distinct part of the process and uses associated processing equipment. An important state-of-the art technique to perform some of these steps is RTP. This technique has major advantages over conventional furnace-based batch thermal processing of semiconductor wafers. In the conventional furnace-based techniques, the processing step involves several hours, and the speed is limited by the large thermal masses of the walls. In contrast, in RTP only the wafer mass is heated or cooled and the RTP walls are water cooled and kept at room temperature. This cuts down the processing time to seconds. From a manufacturing point of view, RTP

*This research is supported in part by ARPA under AFOSR Contract F49620-94-C-0003.

0-7803-1968-0/94$4.00©1994 IEEE

fits naturally into the current cluster-tool concepts which promise IC fabrication lines which are more flexible, and much less capital intensive as compared to present billion dollar state-of-the-art fabs. RTP is an essential technology for single-wafer processing. RTP's viability has been demonstrated for process steps such as silicidation, RTCVD, and annealing [8]. It has been also proposed as an efficient way to clean wafers. The key enabling technology for application of RTP in a manufacturing setting is the precise temporal and spatial control of temperature. Sensors are now available [9] but control design needs further development.

Typical RTP systems are described in [7]. A diagram of a typical system is shown in Figures 1-2 [7]. The wafer is heated by radiation via a lamp array. In one design (Figure 2), rings of tungsten-halogen lamps arrays are used as heaters (actuators) and are separated from the chamber by a quarts window. The lamp array has a hexagonal packing. The wafer is heated only from the top-side. There are several other competing alternatives for the lamp design (e.g., a two array scheme [7]) . However, precise temperature control is required regardless of the lamp design. The lamp voltage requirements are from 20 to 200 volts but are dependent on the chamber geometry, etc. The chamber has a large number of inputs and outputs. It uses advanced pyrometers as temperature sensors (see Figure 3). The system is to follow a pre-defined temperature profile (ramp up, hold, cool down) such as shown in Figure 4, and accurate tracking of the temperature profile is required along with minimal overshoots at transitions and minimal spatial temperature variations during all phases of the profile. This must be done to insure that all wafers are processed the same way so as to achieve repeatability. Some designs keep the wafer fixed, whereas in others the wafer is rotated during the processing cycle. The wafer rotation can induce angular disturbances which must be rejected. Other disturbances in the system are low frequency ones, e.g., the heat transferred to the wafer from the quartz window when the lamps are first turned on. To meet requirements for $0.25\mu m$ devices, SEMATECH has established 1995 goals for temperature uniformity across the wafer of $\pm 3^\circ C$ for oxide and $\pm 5^\circ C$ for other processes. This must be so to ensure that processing is uniform across the wafer and thermal stress does not result in wafer defects (warping, slip). Accurate and repeatable temperature control is required starting at low temperatures and high temperatures. Temperature uniformity is required on all the wafer in spite of the fact that temperature is being measured only at a finite number of points. The control system must deal with actuator saturations (corresponding to maximum lamp power setting or intensity of 200 volts). This is a particularly important problem in RTP because of the wide ranges of process temperature setpoints. Other factors affecting system performance include wafer diameter, chamber geometry, gas flow uniformity and cooling of the chamber.



NOT SHOWN: QUARTZ PINS TO SUPPORT WAFER
GAS INJECT AND EXHAUST PORTS

Figure 1: Cross-section of generic RTP system.

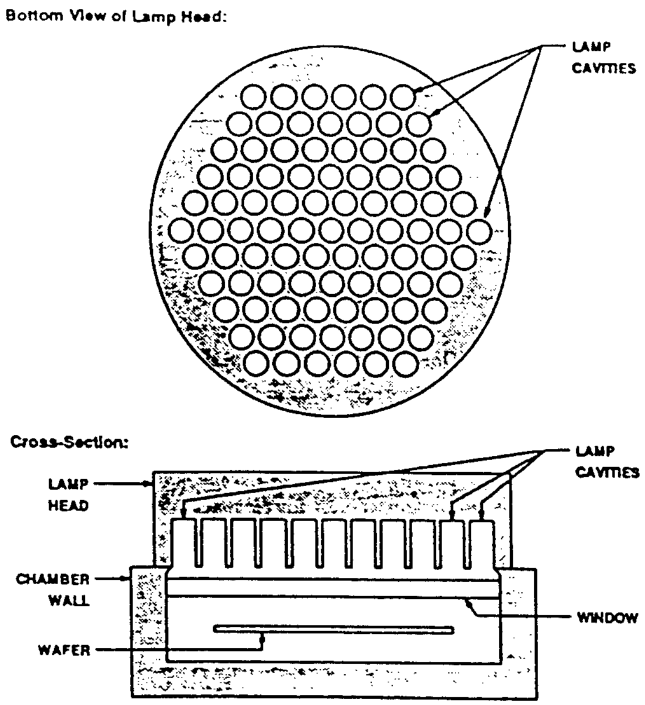


Figure 2: One-sided heating with hexagonal array of tungsten-halogen lamps.

Precise temperature control is critical to this promising technology. In these systems, many heaters affect the temperature at each location where it is measured. Control that explicitly accounts for the influence of each heat source on each temperature sensor is needed for acceptable performance. With such strong physical coupling, it is extraordinarily difficult to obtain acceptable control of the temperature profile using single loop conventional controllers commonly used in industrial applications. Moreover, since previous approaches relied heavily on precise calibration, slight changes in chamber design or wafer geometry can require substantial and time-consuming efforts in control redesign. The necessity for meeting extremely high performance specifications requires that the control system be optimal with respect to the specific process being controlled. As a result, model-based multivariable control system design is a must.

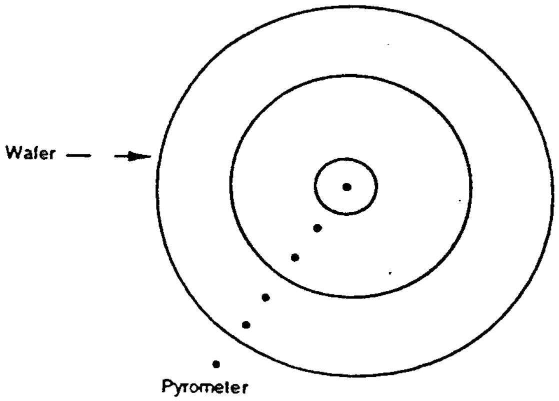


Figure 3: Sensor Locations

The approach presented in this paper will also allow us to explicitly account for actuator saturations that have been handled in an ad hoc fashion in current designs. In addition, the methodology is general so that it can be applied to other processes.

# 3 Problem Formulation and Main Result

Let the plant, $P$, have a minimal state-space description ($A$,$B$,$C$), with $n_c$ inputs, $n_o$ outputs and $n_s$ state variables. Hence,

$$x(k+1) = Ax(k) + Bu(k) \qquad (1)$$

$$y(k) = Cx(k). \qquad (2)$$

Assume that the plant, $P$, is at rest, i.e., $x_0 = 0$. We are interested in the rest-to-rest maneuvering of the system subject to actuator saturation. Specifically, let each of the inputs have a specified actuator constraint,

$$u_i \min < u_i < u_i \max \qquad i = 1,2,\ldots,n_c. \qquad (3)$$

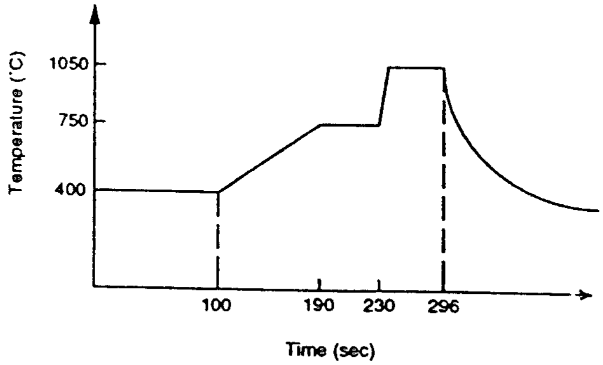


Figure 4: Typical RTP Cycle (for oxide growth).

For a given integer $N$, let $U_N$ denote the set of all control inputs bounded by Eq. (3) but of duration $N$ time steps. We say that the system tracks a reference input $r$ iff

$$y(k) = r(k) \qquad \text{for all} \quad k \geq N. \qquad (4)$$

The goal is to find $u \in U_N$ such that Eq. (4) holds.

Assume that the plant is internally stable and its transfer matrix has been written in the pole-residue form (assuming distinct poles),

$$P(z) = C(zI - A)^{-1}B = \sum_{i=1}^{n_s} \frac{H_i}{(z-\lambda_i)} \qquad (5)$$

where

$$H_i = Cv_iq_i^TB \qquad i = 1,2,\ldots,n_s \qquad (6)$$

and $v_i$, $q_i^T$ are the right and left eigenvectors of $A$ respectively. Suppose it is desired to follow a constant reference input. We may decompose the input $U(z)$ into two parts: an N-tap FIR and a steady-state part as follows,

$$U(z) = U_{FIR}(z) + U_{ss}(z). \qquad (7)$$

Specifically, we may represent the input as,

$$U(z) = \sum_{i=0}^{N} p_i z^{-i} + \hat{p}\frac{z^{-N}}{(z-1)} \qquad (8)$$

where $p_i$ and $\hat{p}$ are vectors of size $n_c$. The first part of the right hand side of Eq. (8) is a set of N-tap FIR filters whose effect vanishes exactly after N time steps. The second term is a vector of delayed steps. The control sequence is then,

$$u(k) = \sum_{k=0}^{N} p_k\delta_k + \hat{p}1(k-N-1) \qquad (9)$$

where $\delta_k$ is the unit pulse.

We may also decompose the output of the system into an N-tap FIR part and a steady-state component,

$$Y(z) = C(zI - A)^{-1}BU(z) = Y_{FIR}(z) + Y_{ss}(z). \quad (10)$$

The desired steady-state output is,

$$Y_{ss}(z) = \frac{z}{z-1} r_o \quad (11)$$

where $r_o$ is the constant output levels. We may substitute Eqs. (8) and (11) in Eq. (10) and equate the residues for the plant poles, $z = \lambda_j$, as well as the pole of the input signal at $z = 1$. For $z = 1$ we obtain,

$$\sum_{i=1}^{n_s} \frac{H_i}{1-\lambda_j} \hat{p} = r_o \quad (12)$$

and for each $\lambda_j$ we have,

$$\sum_{i=0}^{N} H_j \lambda_j^{-i} p_i + H_j \frac{\lambda_j^{-N}}{\lambda_j - 1} \hat{p} = 0 \qquad j = 1, \ldots, n_s. \quad (13)$$

Given $N$, the problem may then be formulated as a linear programming (LP) problem,

$$\max \quad \lambda$$

$$\mathcal{A}(N) \begin{bmatrix} p \\ \hat{p} \end{bmatrix} = \lambda B$$

$$u_{\min} < p_i < u_{\max} \qquad i = 0, 1, 2, \ldots, N \quad (14)$$
$$u_{\min} < \hat{p} < u_{\max},$$

where

$$p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_N \end{bmatrix} \quad (15)$$

and the actuation bounds are

$$u_{\min} = \begin{bmatrix} u_1 \min \\ u_2 \min \\ \vdots \\ u_{n_c} \min \end{bmatrix} \quad u_{\max} = \begin{bmatrix} u_1 \max \\ u_2 \max \\ \vdots \\ u_{n_c} \max \end{bmatrix}. \quad (16)$$

The scaling on the achievable steady-state value (relative to $r_o$) is $\lambda$. Hence the desired value of $\lambda$ is unity. The structure of $\mathcal{A}$ and $B$ matrices (shown for $n_c = 4$) are shown below,

$$\mathcal{A}(N) = \begin{bmatrix} 0 & 0 & \ldots & 0 & \sum_{k=1}^{n_s} \frac{H_k}{1-\lambda_k} \\ H_1 & H_1\lambda_1^{-1} & \ldots & H_1\lambda_1^{-N} & H_1 \frac{\lambda_1^{-N}}{\lambda_1 - 1} \\ H_2 & H_2\lambda_2^{-1} & \ldots & H_2\lambda_2^{-N} & H_2 \frac{\lambda_2^{-N}}{\lambda_2 - 1} \\ H_3 & H_3\lambda_3^{-1} & \ldots & H_3\lambda_3^{-N} & H_3 \frac{\lambda_3^{-N}}{\lambda_3 - 1} \\ H_4 & H_4\lambda_4^{-1} & \ldots & H_4\lambda_4^{-N} & H_4 \frac{\lambda_4^{-N}}{\lambda_4 - 1} \end{bmatrix}$$

$$B = \begin{bmatrix} r_o \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

A series of convex minimization problems can be solved by varying $N$ to obtain a solution to the finite-time tracking problem which satisfies the hard limits on the actuators. The design procedure is as follows:

Step 1. Select a large enough N (depending on plant dynamics).

Step 2. Solve the linear program. If feasible, decrease N and repeat until infeasible. If infeasible, increase N. Continue until a satisfactory solution is obtained.

Step 3. Form the input sequence, $u(k)$, and evaluate the tracking performance.

Note that at any $N$ for which $\lambda > 1$, scaling of the control signal by $\lambda$ achieves the desired tracking and satisfies the saturation constraint. However, the extremal solution (i.e. the smallest $N$) will correspond to the case where optimal $\lambda = 1$. The solution to this problem has an interpretation from the classical deadbeat control concepts. By construction, $U(z)$ will always contain zeros to cancel all the (stable) plant poles. The combination of $P(z)U(z)$ will have $N$ poles at the origin. Hence constant inputs will be tracked in $N$ time steps. The locations of zeros of $U(z)$ have a nice geometrical pattern as seen in the following examples. Note that steps 1-3 are performed off-line and the resulting input sequences are stored in a table for on-line use.

# 4 Example 1: SISO RTP Plant

This is a model of an RTP system represented by the first order plant

$$P(z) = \frac{2.3164}{(z - .9964)}.$$

The tracking is achieved in $N = 3$ time periods as shown in Fig. 5. The zeros of $U(z)$ are located at $0.4982 \pm j0.8629$ and at the plant pole at 0.9964. The zeros of $U(z)$ appear inside the unit circle in a Butterworth pattern.

# 5 Example 2: MIMO RTP Plant

This is a 4-input-4 output model of an RTP reactor as described in [11]. The poles of the system are at 0.9052, 0.9827, 0.9856 and 0.9972 and it has no finite transmission zeros. The plant model is specified so that the input is in percent power and the output is temperature in °$C$. The linear model is valid between 750°$C$ and 1050°$C$. It is assumed that the system is at 750°$C$ and is to be taken to 1050°$C$ in the fastest possible time and held there. The constraints on the actuators are power settings between 9% and 100%.

Several values of $N$ were tried (N=60, 65, 70). The tracking requirement is met for $N = 70$ as shown in Figure 6. Note that all the control signals shown in Figure 7 are of the "bang-bang-hold" variety. It is interesting to see that $u_2$ turns off quickly due to the coupling in the system. However, as soon as $u_1$ and $u_3$ turn off, $u_2$ is quickly turned back on. Since the sampling period is 0.05 second, the system tracks the 300°$C$ step in less than four seconds. This is well below the 10-15 seconds settling-time reported in [11].

The zeros of $U(z)$ have a Butterworth pattern and include all the four plant poles.

While the speed of tracking is the best possible, in the actual system, it is desired to have all the responses grow together to avoid wafer slip. Ways to deal with this are currently under investigation. The response shown in Figure 6 is, however, quite acceptable at low temperatures.

## 6 Closed-Loop Implementation

The finite-time tracking may be implemented in a closed-loop control structure to provide disturbance rejection and robustness with respect to modeling errors. Consider the closed-loop system shown in Figure 8 where $\hat{u}$ is the tracking command and $\hat{y} = P\hat{u}$. If the plant model is accurate, in the absence of disturbances, the same tracking performance is achieved as in the open-loop configuration. The feedback controller $K(z)$ provides disturbance rejection capability and is designed independently of $\hat{u}$. Since part of the actuation is now used for disturbance rejection, the design in the previous sections must be conservative. The designer then "backs off" on $u_{min}$ and $u_{max}$ to leave enough head room for disturbance rejection. One can choose a large enough N so that a value of $\lambda > 1$ is achieved. The open-loop control is decreased by a factor of $\lambda^{-1}$ to leave enough actuator authority for disturbance rejection.

## 7 Conclusions

This paper presents a complete solution to the rest-to-rest finite-time tracking problem in the presence of actuator constraints. While parts of the solution had existed in the literature, the problem is solved for the general multivariable case using the solution to a single linear program. Only constant signal tracking was discussed. However, the methodology is general and applies to tracking of signals that can be generated by a linear system. It was assumed that the open-loop system is stable. This is a generic property of process control systems for which the technique is intended. The technique may be used to establish the upper limit on tracking performance, i.e., the minimum tracking time achievable. The connection to the classical deadbeat control theory is interesting. Deadbeat control establishes the minimum tracking time achievable *without* actuator constraints. Even though the approach to the problem is open-loop, the final implementation may be done in a closed-loop fashion, i.e., using a combination of feedforward/feedback as in Figure 7. The technique is currently being applied to thermal systems (RTP, APCVD furnace).

## References

[1] R. E. Kalman, "Optimal Nonlinear Control of Saturating Systems by Intermittent Action," IRE Wescon Record, Part 4, pp. 130-135, 1957.

[2] S. F. Schmidt, "The Analysis and Design of Continuous and Sample-Data Feedback Control Systems with a Saturation Type Nonlinearity," Ph.D. Dissertation, Stanford University, June 1959.

[3] M. G. Kabuli and R. L. Kosut, "Fast Finite-Time Tracking with Saturating Actuators," in *Proc. Conf. Decision and Control*, pp. 2883-2884, 1991.

[4] M. L. Workman, "Adaptive Proximate Time-Optimal Servomechanisms," Ph.D. Dissertation, Stanford University, March 1987.

[5] M. L. Workman, R. L. Kosut and G. F. Franklin, "Adaptive Proximate Time-Optimal Control: Continuous-Time Case," *Proc. American Control Conf.*, pp. 589-594, June 1987.

[6] M. L. Workman, R. L. Kosut and G. F. Franklin, "Adaptive Proximate Time- Optimal Control: Discrete-Time Case," *Proc. Conf. Decision Control*, pp. 1548-1553, December 1987.

[7] S. A. Norman, "Wafer Temperature Control in Rapid Thermal Processing," Ph.D. Dissertation, Stanford University, July 1992.

[8] Texas Instruments Technical Journal, Vol.9, No. 5, September-October, 1992.

[9] F.L. Degertekin, J. Pie, Y.J. Lee and B.T. Khuri-Yakub and K.C. Saraswat, "In-Situ Temperature Monitoring in RTP by Acoustical Techniques," MRS Symp., San Francisco, April 1993.

[10] P. J. Gyugyi, Y. M. Cho, G.F. Franklin and T. Kailath, "Control of Rapid Thermal Processing: A System Theoretic Approach," in Proc. 1993 *IFAC World Congress.*

[11] C. Elia, "RTP Multivariable Temperature Controller Development," *Proc. Automatic Control Conference*, pp. 907-911, June 1994.

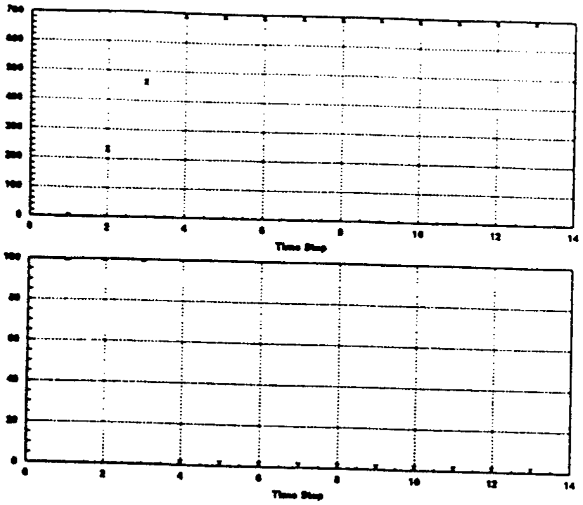


Figure 5: Transient response and control effort for Example 1.

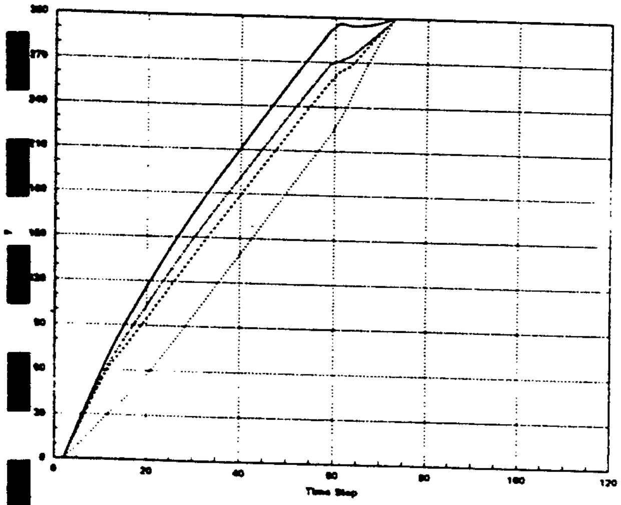


Figure 6: Transient response for Example 2.

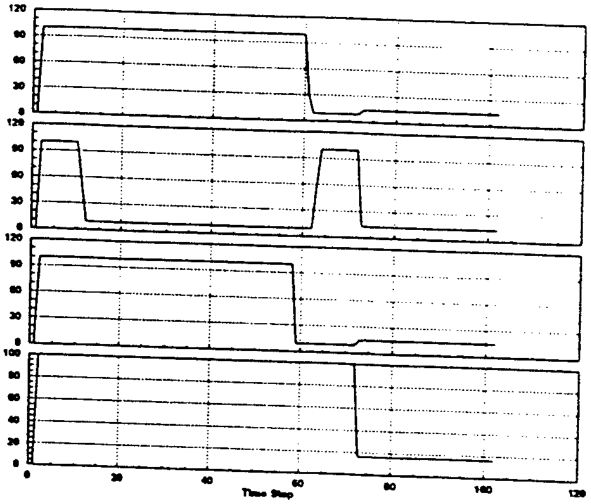


Figure 7: Control effort for Example 2.

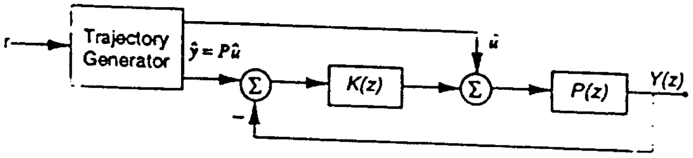


Figure 8: Closed-loop implementation

## A.4 On Actuator-Sensor Selection in Thermal Processes

- Robert L. Kosut and M. Güntekin Kabuli, "On Actuator-Sensor Selection in Thermal Processes," *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, Louisiana, December 13–15, 1995.

# On Actuator-Sensor Selection in Thermal Processes*

Robert L. Kosut M. Güntekin Kabuli

Integrated Systems Inc.
3260 Jay Street, Santa Clara, CA 95054-3309
email: kosut@isi.com , kabuli@isi.com

## Abstract

A first-principles based nonlinear algebraic model of a rapid thermal processing chamber is used to consider an actuator-sensor selection problem based on static (steady-state) performance. A two step systematic procedure is proposed and illustrated on an example. Actuator groupings are ranked according to the achievable best nominal uniformity levels. Once the actuator groupings are determined, the sensor locations are rated according to the best worst-case achievable closed-loop performance.

## 1 Introduction

Rapid thermal processing (RTP) is an efficient multi-chamber single-wafer processing approach in integrated circuit manufacturing. The chemical reaction recipes and high throughput goal, demand fast tracking control laws that achieve near uniform spatial temperature distributions across a semiconductor wafer, during both transient and steady-state phases of the process. Due to the radiative effects, small chamber volume and remote sensing restrictions, the desired performance specifications pose a challenging actuator-sensor selection problem. Multiple tungsten-halogen lamps act as the only heat source. There is no active cooling of the wafer. Surface properties of the wafers differ initially and also vary during processing. Temperature readings across the wafer by remote sensing techniques are a function of uncertain surface properties. In the course of chamber design, different actuator-sensor configuration choices due to lamp groupings and pyrometer target locations, need to be ranked. The ranking of these choices will be based on "worst-case" performance degradations under "best" possible control design.

## 2 Problem Description

Consider the algebraic steady-state (subject to constant inputs) model of an RTP chamber, denoted by $\mathcal{P}$ :

$$\mathcal{P}:(w,u)\mapsto(z,y)\begin{cases}0 = f(x,w,u)\\ y = g(x,w)\\ z = Cx\\ u_{\min}\le u \le u_{\max}\end{cases}, \qquad (1)$$

where $w\in\mathbb{R}^{n_w}$ , $u\in\mathbb{R}^{n_u}$ , $x\in\mathbb{R}^{n_x}$ denote the uncertain constant parameters, input values and steady-state cell temperatures, respectively. The minimum power levels are nonnegative (no active cooling). Vector inequalities are to be interpreted entry by entry. The model in (1) is obtained from a first-principles based dynamic model of an RTP chamber, where the nonlinearities $f$ , and $g$ are smooth, and determined by heat conservation equations (see e.g., [1]) . For a given $w$ and $u$ , there is a unique steady-state determined by (1) . The outputs $y\in\mathbb{R}^{n_y}$ denote the pyrometer readings and $z\in\mathbb{R}^{n_z}$ denotes unmeasured regulated cell temperatures across the wafer. For a given reference temperature $r\in\mathbb{R}_+$ , the performance goal is to minimize the worst-case spatial temperature error $(z-r\mathbf{1})$ subject to $w$ , where $\mathbf{1}$ denotes a vector of ones. The uncertain constant parameter vector $w$ (denoting uncertainties in emissivities, thermal conductance, thermal mass etc.) is in a known polytope $w_{\min}\le w\le w_{\max}$ . The effect of imposing integral action based on the pyrometer readings $(n_y\le n_u)$ will also be considered in answering the following questions:

- **What is the best uniformity predicted by the model ?**
- **What are the best lamp grouping and sensor locations for a range of $r$'s ?**

Since some of the $w$ entries represent characteristics of the chamber, an answer to the first question above serves a dual purpose: the limits of performance are determined and suggestions can be made for a possible chamber redesign to further improve the limits of performance. The second question above, is conceptually a special case of the previous, by merely grouping actuators, selecting sensors and redefining the model under study. It has more of a practical significance in that one can determine an implementation cost versus achievable uniformity trade-off in order to justify additional sensors and/or actuators in the control design. In the rest of the paper, we address these questions and propose solution methods.

## 3 Operating Point: Best Nominal Uniformity

Let the spatial uniformity error be quantified with its root-sum-square-error (Frobenius norm) and let $w_0$ denote a nominal parameter. For a given reference $r$ , the best steady-state uniformity is achieved at an input level $u=\hat{u}(r,w_0)$, where

$$\hat{u}(r,w_0) = \arg\min_{\substack{u_{\min}\le u\le u_{\max}\\ 0=f(x,w_0,u)\\ z=Cx}} \|z-r\mathbf{1}\| \ . \qquad (2)$$

Let $\hat{x}(r,w_0)$ denote the unique state associated with (2) . Using Newton-Raphson iterations based on quadratic program solutions, the operating point map $(\hat{u},\hat{x})(\cdot,\cdot)$ is evaluated for different references $r$ and/or parameters $w$ . The associated minimum determines the performance lower bound dictated by the plant model. *No combination of feedforward/feedback can achieve a nominal uniformity error smaller than this lower bound.*

### 3.1 Case Study: Nominal Uniformity

For the model under study, the dimensions in (1) are $n_x=115$, $n_w=4$, $n_u=21$ , $n_z=21$ and $n_y=27$ . Power levels are restricted to the interval $[0.02,1]$ . For a nominal $w_0$ , the full 21 actuator case and a particular 6 actuator grouping were compared for reference $r$ ranging from 600°C to 1200°C with 100°C increments. For each actuator grouping and reference, a problem as in (2) was solved. The results are shown in Figure 1 . By utilizing all of the available 21 actuators, the uniformity can be improved by a factor ranging from 5 to 10 , over the reference temperature range. As illustrated by the lower-right plot in Figure 1 , the first three pairwise groupings (the first six X's) are close to the first six of 21 independent solutions (filled circles). However, the last three groupings (denoted by 3,4 and 7 grouped X's) are considerably off, and the uniformity degradation is as high as a factor of 10 . The lower curve in the lower left plot in Figure 1 is a performance lower bound. By generating more curves associated with candidate actuator groupings (as done for the sample 6 grouping), one can decide on an acceptable grouping based on the implementation limitation and the degradation in uniformity from the performance limitation. This uniformity comparison can also be performed for different minimum and maximum power levels.

## 4 Operating Point: Best Worst-Case Uniformity

Sensitivity minimization at a given operating point based on the affine approximation of the model and control was performed:

*Research supported by ARPA under AFOSR contract F49620–94–C–0003.

**1.** For a given reference $r$ and a nominal $w_0$ , solve the problem in (2). Let the input state pair $(u_0, x_0)$ be the associated minimizer. Let $y_0$ and $z_0$ be the outputs and regulated variables at the operating point $(u_0, x_0)$ .
**2.** Obtain the steady-state affine approximation to the plant model in (1) about the operating point $(u_0, x_0)$. Augment with actuator and sensor noises to obtain the real matrix relating $(w_{aug}, u)$ to $(z_{aug}, y)$ .
**3.** Apply the affine control law $u = u_0 + u_K + K(y - y_0)$ and determine the feedforward/feedback terms by solving

$$\min_{(u_K, K)} \quad \|\Delta_2^{-1}(z_{aug} - \delta_2)\|_\infty$$

$$\|\Delta_1^{-1}(w_{aug} - \delta_1)\|_\infty \leq 1$$

with and without integral action constraint on a subset of $y$'s. Note that the maximum absolute value norm ($\| \cdot \|_\infty$) is used to conform to the interval uncertainty description of $w$ . The weights $\Delta_1$, $\Delta_2$ and the centers $\delta_1$, $\delta_2$ are design parameters. The feedforward term $u_K$ is redundant if the centers $\delta_1$ and $\delta_2$ are associated with the steady-state operating point. The problem can be transformed into a linear-program of the type

$$\min_{X} \quad \|T_1 + T_2 X T_3\|_{i,\infty}$$

$$( \ S P_{yu} X = I \ )$$

where the matrix variable $X$ is constrained by the integral action constraint (if there is any) imposed on the selected outputs $Sy$ , where $S$ denotes a selection matrix obtained by choosing the appropriate rows of an identity matrix. The induced matrix norm $\| \cdot \|_{i,\infty}$ is the maximum row sum. A detailed description of the transformation and efficient solution methods can be found in [2].

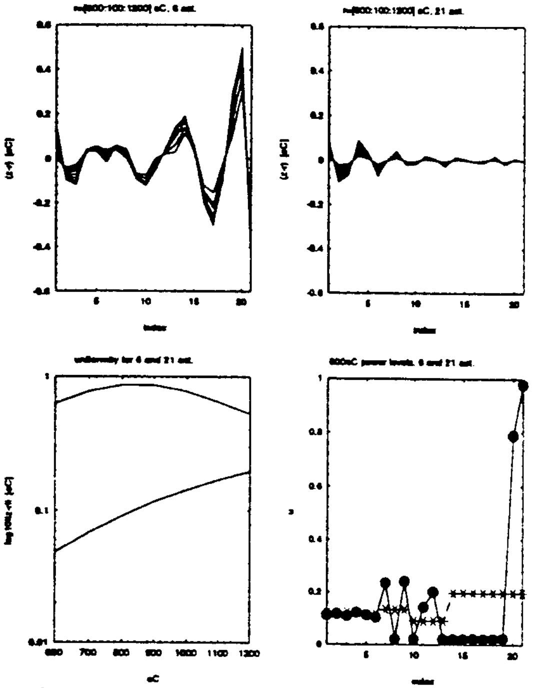


Figure 1: Nominal uniformity study for 6 and 21 actuator configurations.
**upper left:** the best nominal spatial temperature error distribution for the particular 6 actuator grouping.
**upper right:** the best nominal spatial temperature error distribution, all 21 actuators utilized.
**lower left:** comparison of 6 and 21 actuator groupings.
**lower right:** the optimal power levels at 800°C for the 6 (X's) and 21 (•'s) actuator groupings.

### 4.1 Case Study: Best Worst-Case Uniformity

For the model considered in Section 3.1 , a sensitivity study was performed at the reference $r = 1100°C$ . The operating points at the desired reference was computed utilizing all 21 actuators and solving the problem in (2) with power levels in the interval $[0.175, 0.85]$ . The plant was linearized about this operating point and augmentation was done as described in Section 4 . The centers were selected as the nominal values and disturbance weights were selected to reflect the following: ±5% variation in plant parameters, ±0.5°C sensor noise, and ±0.01 actuator noise. Six sensor locations were chosen and two sets of feedback values were computed with and without integral action constraint on the particular six measured outputs. Let $H_z$ and $H_u$ denote the closed-loop gain matrices from $w_{aug}$ to the regulated variables $z$ and actuator $u$ , respectively (including the associated weightings in $w_{aug}$) . Similarly, let $\widehat{H}_z$ and $\widehat{H}_u$ denote the associated closed-loop matrices for the design with integral action constraint. The subscripted $H$ matrices are all 21 by 31 ($n_z = 21$ , $n_u = 21$ , $n_y = 6$ , $n_{w_{aug}} = 4 + 6 + 21 = 31$) . Let $sgn(\cdot)$ denote the signum function; i.e., 1 for nonnegative and -1 for negative, evaluated entry by entry. Let $abs(\cdot)$ denote the absolute value function evaluated entry by entry. Since the design problem is posed in terms of the peak norm and the $w_{aug}$ weightings are already included in the subscripted $H$ matrices, the columns of the matrix $\left(H sgn(H)^T\right)$ correspond to the 21 spatial worst-case candidates and the vector $(abs(H)\mathbf{1})$ denote the spatial peak deviation at the worst-case $w_{aug}$ (see Figure 2 ) .

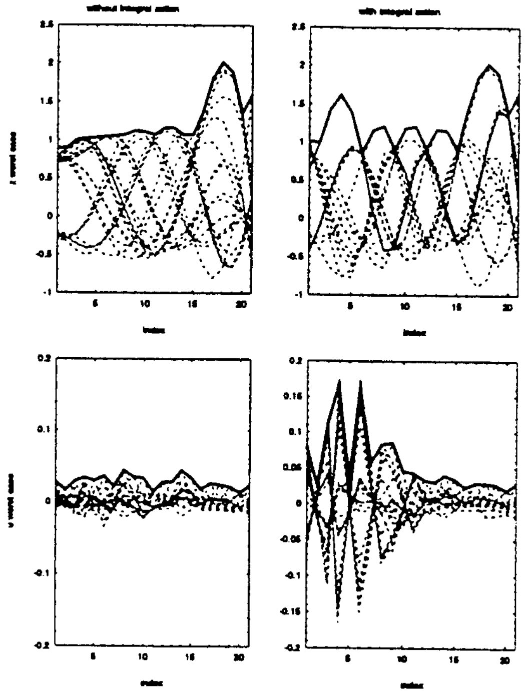


Figure 2: Best worst-case fluctuations about the nominal, with and without integral action constraint on the measured outputs used in feedback. $((\cdot)sgn(\cdot)^T)$ (dashed lines) and $(abs(\cdot)\mathbf{1})$ (solid line) evaluated at $H_z$ (upper left), $\widehat{H}_z$ (upper right), $H_u$ (lower left), $\widehat{H}_u$ (lower right).

## Concluding Remarks

Actuator-sensor selection was considered as a two step procedure: 1) actuator grouping and nominal structural parameter effects were investigated in terms of best nominal uniformity (Section 3), and 2) sensor selection and the effect of integral action constraints were investigated in terms of best worst-case uniformity (Section 4) under parametric uncertainty. The proposed approach is a systematic way of determining the effect of actuator/sensor selection in operating point sensitivity reduction. Case study results were used to illustrate the approach.

## References

[1] F. P. Incropera and D. P. DeWitt, *Introduction to Heat Transfer*, John Wiley & Sons, 1985.

[2] M. Güntekin Kabuli, Robert L. Kosut and Stephen P. Boyd, "Improving static performance robustness of thermal processes," *Proceedings of the 33rd IEEE Conference on Decision and Control*, pp. 62–66, Lake Buena Vista, Florida, December 1994.

## A.5 On operating point sensitivity of thermal processes

- Robert L. Kosut and M. Güntekin Kabuli, "On operating point sensitivity of thermal processes," *Proc. 1996 Triennial IFAC World Congress*, San Francisco, California USA, July 1-5 1996.

# ON OPERATING POINT SENSITIVITY OF THERMAL PROCESSES*

Robert L. Kosut M. Güntekin Kabuli

*Integrated Systems Inc.*
*3260 Jay Street, Santa Clara, CA 95054-3309 , U.S.A.*
email: kosut@isi.com , kabuli@isi.com

Abstract: First-principles based nonlinear dynamic model of a rapid thermal processing chamber is used to consider a control/structure interaction problem. The proposed approach ranks a particular chamber design according to steady-state wafer uniformity analysis for constant inputs as well as achievable transient uniformity during ramping using a baseline integral action controller. A computational toolset is developed to determine operating points, reduced-order small signal equivalent models, step- or ramp-tracking feedback controllers and a baseline dynamic response for a given chamber design. The approach is illustrated on a model.

Keywords: Steady state, Process control, Thermal equilibrium, Temperature profiles, Integral action.

## 1. INTRODUCTION

Rapid thermal processing (RTP) is a new approach in integrated circuit manufacturing; it is a fast and efficient multi-chamber single-wafer processing approach in contrast to the conventional slow and costly single-chamber multi-wafer processing. A typical RTP chamber volume is much smaller than that of a batch processing chamber; moreover, RTP chamber walls are cooled. Hence, successive single-wafer processing can be done rapidly, and chamber clean-up is not required between consecutive processes. The chemical reaction recipes and high throughput goal demand fast tracking control laws that achieve near uniform spatial temperature distributions across a semiconductor wafer, during both transient and steady-state phases of the process. Due to the radiative effects, small chamber volume and remote sensing restrictions, the desired performance specifications pose a challenging control/structure interaction problem.

In a typical RTP chamber, multiple tungsten-halogen lamps act as the only heat source. There is no active cooling of the wafer. Surface properties of the wafers may differ initially and will certainly vary during processing, as well. Temperature measurements across the wafer have to

*Research supported by ARPA under AFOSR contract F49620-94-C-0003.

be done remotely, and most remote sensing techniques rely on uncertain surface properties. A challenge in this control/structure interaction problem is to be able to rank different actuator/sensor configuration choices due to particular lamp groupings, pyrometer target locations and chamber specific parameters according to achievable transient and steady-state wafer spatial temperature uniformities. The proposed approach in this paper allows the user to determine operating points, reduced-order small signal equivalent models, step- or ramp-tracking feedback controllers and a baseline dynamic response for a given chamber design. The approach is illustrated on a model.

## 2. MODEL DESCRIPTION

A first-principles based dynamic model of the generic RTP system is denoted by $\mathcal{P}$ :

$$\mathcal{P}:(w,u)\mapsto(z,y)\left\{\begin{array}{rcl} \dot{x} & = & f(x,w,u) \\ y & = & g(x,w \\ z & = & Cx \\ \underline{u}\leq & u & \leq\overline{u} \end{array}\right. , \quad (1)$$

where $w\in\mathbb{R}^{n_w}$ denotes the uncertain parameters, $u\in\mathbb{R}^{n_u}$ denotes the input values and $x\in\mathbb{R}^{n_x}$ denotes the node temperatures. A subset of the states $x$ corresponds to the wafer states, denoted by $z\in\mathbb{R}^{n_z}$ . Pyrometer measurements are denoted by $y\in\mathbb{R}^{n_y}$ . The minimum power levels $\underline{u}$ are nonnegative, since there is no active cooling. The predetermined maximum power levels are denoted by $\overline{u}$ . Vector inequalities are to be interpreted entry by entry.

Typically, a dynamic nonlinear model as in (1) is obtained from first-principles using heat conservation equations associated with a large-scale nonlinear resistor-capacitor network consisting of branches that model conduction, convection and radiation; see e.g. (Incropera and DeWitt, 1985) .

In this paper, a generic RTP system model developed by Ebert, *et al.* (1995) will be used for illustration purposes. In this particular RTP system, the chamber is a water-cooled cylindrical cavity with five independently powered lamps ($n_u=5$). A thick quartz window below the lamps separate the lamp cavity from the wafer cavity. A thinner quartz plate below the quartz window serves as a showerhead to allow gas flow into the system. The silicon wafer is located below the showerhead. A guard ring near the edge of the wafer improves the temperature uniformity. The model is derived for low-pressure operation where gas flow and gas convection heat transfer are not important. The elements of the system are divided into nodes ($n_x=116$) and the associated conservation equations are derived. A two-band radiation model is implemented to accommodate the semitransparent quartz elements. The model is parametrized in terms of uncertain emissivities, thermal conductance, thermal mass, etc. The silicon wafer is divided into $n_z=21$ nodes.

## 3. STEADY-STATE ANALYSIS

In a typical reaction recipe, the wafer has to maintain a uniform spatial profile while tracking a desired piecewise-linear reference trajectory that represents multiple ramp and hold phases. For example, a wafer at room temperature is first heated up to and held at 500°$C$ for a prespecified duration and then ramped up with a prespecified ramp rate and held at 1100°$C$ for a prespecified duration, after which the wafer goes through a cool down phase. The overall time frame is in the order of seconds. During such reaction recipes, a chamber does not reach steady-state over the short durations when the reference is held constant. In fact, open-loop state responses to step inputs exhibit the different time-scales: e.g., fast wafer state responses versus slower quartz window state responses. After feedback control is applied, the wafer states are steered to their desired steady-state values over a faster time-scale then the rest of the states by imposing integral action on certain pyrometer measurements. The premise of the steady-state analysis in this section is the following: By clamping the pyrometer measurements at prespecified steady-state values, the wafer spatial temperature profile is kept close to the wafer steady-state profile. Hence it is crucial to quantify the achievable uniformity levels that the model in (1) predicts when $\dot{x}=0$ .

Let $S$ denote a selection matrix, where $\eta=S\left[z^T\ y^T\ u^T\right]^T$ denotes the appropriate entries selected from the wafer states, pyrometer measurements and lamp inputs, respectively. Let $\eta_0$ denote the associated desired nominal values. For a fixed parameter value $w_0$ and a particular choice of $S$ and $\eta_0$ and a diagonal nonnegative weighting matrix $\Delta$ , let the operating point $(u, x)$ be given by

$$(u,x) = \arg \min_{\substack{0 = f(x, w_0, u) \\ \underline{u} \leq u \leq \overline{u}}} \left\| \Delta(S \begin{bmatrix} Cx \\ g(x, w_0) \\ u \end{bmatrix} - \eta_0) \right\| . \tag{2}$$

In other words, the operating point determined by (2) minimizes the cost $\|\Delta(\eta - \eta_0)\|$ . The current toolset allows the choices of Frobenius norm or peak norm in the cost description. The problem in (2) is solved using Newton-Raphson iterations based on quadratic program solutions (Frobenius norm case) or linear program solutions (peak norm case). Note that, due to the first-principles based derivation of (1), for a fixed parameter $w$ and an input level $u$ , there is a unique equilibrium state $x$ .

By determining operating points from (2), one can answer critical chamber design specific questions such as the following:
*1) What is the best steady-state uniformity predicted by the model ?* Let $r \in \mathbb{R}_+$ denote a constant reference temperature, which is typically in a given interval $[\underline{r}, \overline{r}]$ determined by the reaction recipes. The performance goal in choosing an operating point is to minimize the spatial wafer temperature error $(z - r\mathbf{1})$ for a given $w$ , where $\mathbf{1}$ denotes a vector of ones. The uncertain constant parameter vector $w$ (denoting uncertainties in emissivities, thermal conductance, thermal mass, etc.) is in a known polytope determined by $\underline{w} \leq w \leq \overline{w}$ . For a given nominal $w_0$ and a reference $r$ , the minimum cost in (2) determines the nominal performance limitation. The achieved minimum uniformity error determines the performance lower bound dictated by the plant model; moreover, suggestions can be made for a possible chamber redesign to further improve the limits of performance. *No combination of feedforward/feedback can achieve a nominal uniformity error smaller than this lower bound.*
*2) What is the tradeoff between wafer uniformity and sensor uniformity ?* Since wafer states are not measured, the answer to *1)* above provides pyrometer reference values that can be used for integral action control. For a given set of sensor locations, uniformity in pyrometer measurements need not imply wafer uniformity.
*3) What are the best lamp groupings for a range of $r$'s ?* This question is a special case of *1)* above, by merely grouping lamps, and redefining the model under study. It has more of a practical significance in that one can determine an implementation cost versus achievable uniformity tradeoff in order to justify the need to install additional actuators driving the lamp groups for control.
*4) What is the effect of minimum and maximum power levels on steady-state uniformity ?* Necessarily, the power levels should comply with desired range of operation. The effect can be seen by changing $\underline{u}$ or $\overline{u}$ in (2). Also, a tradeoff over feasible solutions can be computed by appropriately assigning $\eta$ , $\eta_0$ and $\Delta$ in (2) .
The proposed approach to steady-state analysis is now illustrated on the generic RTP model.

### *3.1. Case Study*

A simple uniformity tradeoff was performed at $r = 1000°C$ for two regions on the wafer. Let $z_1$ denote the first 15 wafer states and $z_2$ denote the latter 6. In other words, the wafer is partitioned into two equal areas: an inner disk and an outer annular region. The cost in (2) was chosen as $\left\| \begin{bmatrix} \lambda(z_1 - r\mathbf{1}) \\ (1-\lambda)(z_2 - r\mathbf{1}) \end{bmatrix} \right\|_2$ . The admissible command levels were restricted to $[0, 1]$ for each lamp. For eight different values of $\lambda \in [0.2, 0.9]$ , the operating points were computed. The results are plotted in Figure 1.

The first strip in Figure 1 shows the tradeoff between $\|z_1 - r\mathbf{1}\|_2$ and $\|z_2 - r\mathbf{1}\|_2$ . The rightmost filled circle denotes the values at $\lambda = 0.2$ ; the filled circles to its left denote the values at 0.1 increments of $\lambda$ with $\lambda = 0.9$ corresponding to the leftmost filled circle in the top strip of Figure 1 . The second strip shows the associated lamp input levels. The last two strips show the spatial temperature uniformities $(z_1 - r\mathbf{1})$ and $(z_2 - r\mathbf{1})$ , respectively.

An evaluation of Figure 1 reveals some interesting chamber design specific properties at $1000°C$ operation. The overall peak-to-peak wafer uniformity cannot be improved further than 2°$C$ when cost is based on Frobenius norm. A confirmation of this intrinsic peak-to-peak limitation can be done at $\lambda = 0.5$ and changing the norm in the cost description to peak-norm. Since the operating point inputs are not violating the desired levels, increasing the maximum power levels will have no effect on improving steady-state uniformity. The inner disk uniformity can be marginally improved at the expense of the annular region uniformity, which results in a considerable 2.5°$C$ tilt. The spatial coordinates and the number of lamps is in fact a key factor for further improvement. By easily generating such tradeoff curves for different chamber parameters and geometries, a crucial control specific assessment can be made.

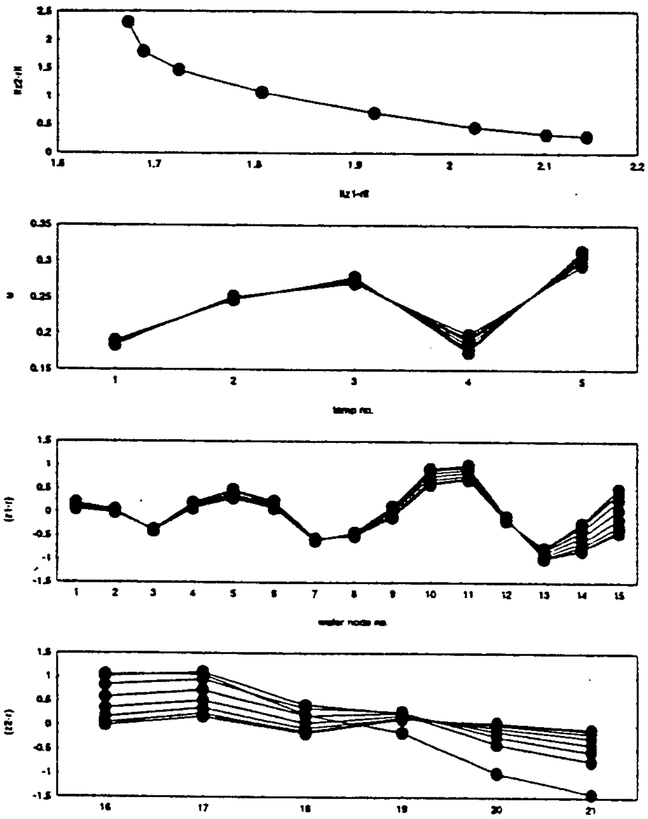


Figure 1: Steady-state uniformity tradeoff at 1000°$C$ for two regions on the wafer in the generic RTP model.

For $\lambda = 0.5$ (i.e., uniform weighting over the 21 wafer states) , a similar uniformity study based on (2) was done for $r = 500, 800, 1100$°$C$ . The optimal power inputs achieving the best steady-state spatial wafer uniformity is shown in the top strip in Figure 2 . The normalized lamp inputs (with respect to the first lamp) are plotted in the second strip in Figure 2 . The effect of grouping all of the five lamps into one actuator using the relative gains in Figure 2 will be illustrated later in Section 4.1 .

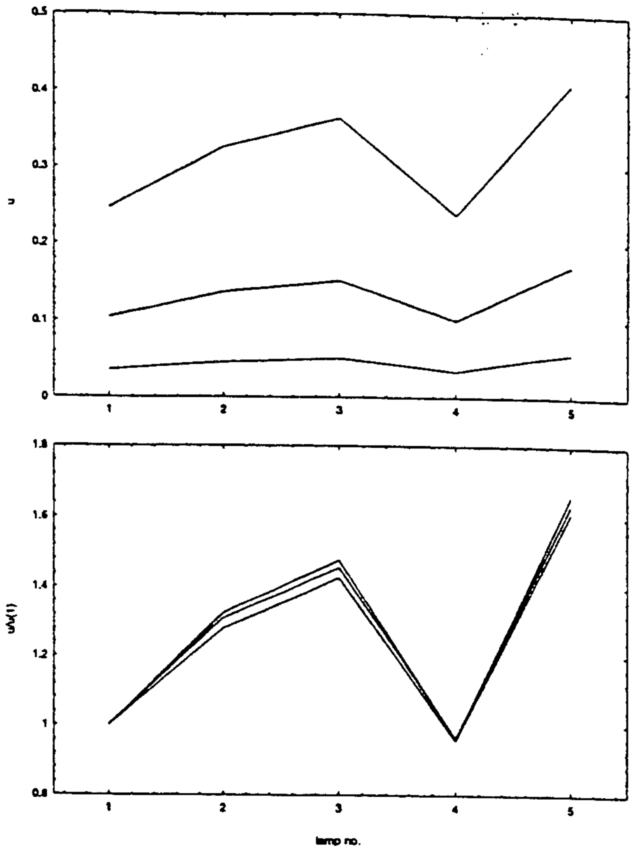


Figure 2: Optimal $u$ for best steady-state wafer profile at 500°$C$, 800°$C$ and 1100°$C$ (top). The normalized power levels $u/u_1$ (bottom).

## 4. TRANSIENT PERFORMANCE

As discussed in Section 3 , the performance limits of a particular chamber design cannot be determined by steady-state analysis only. This section focuses on designing a baseline controller to provide a dynamic response to evaluate achievable transient uniformity.

Let $(u_0, x_0)$ be an operating point determined by solving a problem of the type in (2) . Let $P_0$ denote the transfer-matrix of the small-signal equivalent of

(1) about $(u_0, x_0)$ from input $u$ to sensor $y$ .

Due the physical properties of the process chamber, $P_0$ is minimum phase, stable and strictly proper. The generic RTP process chamber model used for illustration is relative degree two. For speeding up design related iterations and for keeping the order of the feedback controller low, Hankel-singular-value reduction is performed on the high-order $P_0$ to obtain $P$ . All stabilizing controllers for $P$ in the unity-feedback configuration is given by $C = Q(I - PQ)^{-1}$ , where $Q$ is a stable transfer-matrix denoting the free design parameter. A simple design approach is adopted by solving $PQ = F$ for $Q$ , where $F$ denotes a desired closed-loop reference to plant output transfer-matrix with relative degree at least that of $P$ . Clearly, specifying $F$ to be diagonal with nonzero entries corresponds to a decoupling design (hence necessarily, $n_y \leq n_u$) . Imposing $F(0) = I$ results in integral action in all output channels, hence a step tracking design. Imposing an additional $F'(0) = 0$ results in a ramp-tracking design. For chamber design iterations, the adopted approach was a decoupling ramp-tracking design, where the only design freedom was the bandwidth determined by $F$. The resulting controller $C$ was simulated in feedback with the original nonlinear model in (1) with suitable offsets and actuator saturation limits. The resulting dynamic responses complement the tradeoff studies in Section 3 in rating a chamber design since achievable ramp-rates and step response limits can be determined with the available actuation limits.

### *4.1. Case Study*

A simple comparative study was performed on the generic RTP process chamber model, where the design goal was to ramp with a rate of $50°C/s$ from $500°C$ to $1100°C$ using a ramp-tracking controller.

An operating point of (1) was determined at $1100°C$ by solving (2) for minimum wafer spatial uniformity. The associated lamp input level is shown in the top curve in the top strip of Figure 2 . Using the same cost criterion, the operating point at $500°C$ was also computed to start the nonlinear dynamic simulation from steady-state conditions.

The small-signal equivalent about the $1100°C$ operating point was computed (i.e., 116-state $P_0$) . For illustration purposes, pyrometer target locations $i_1 = \{1\}$ and $i_5 = \{1, 5, 10, 15, 20\}$ were chosen to compare a 1-input 1-output control design performance with that of a 5-input 5-output one.

Let $S_{i_1}$ and $S_{i_5}$ denote the selection matrices associated with the index sets $i_1$ and $i_5$ , respectively. Let $P_1$ denote the 15-state reduced order model obtained from $S_{i_1} P_0 \delta_{1100}$ , where $\delta_{1100}$ denotes the normalized lamp levels at $1100°C$ shown in bottom plot in Figure 2 . In other words, $P_1$ is a reduced-order 1-input 1-output model by grouping all 5 lamps into one and using only the first pyrometer location. Similarly, let $P_5$ denote the 15-state reduced order model obtained from $S_{i_5} P_0$.

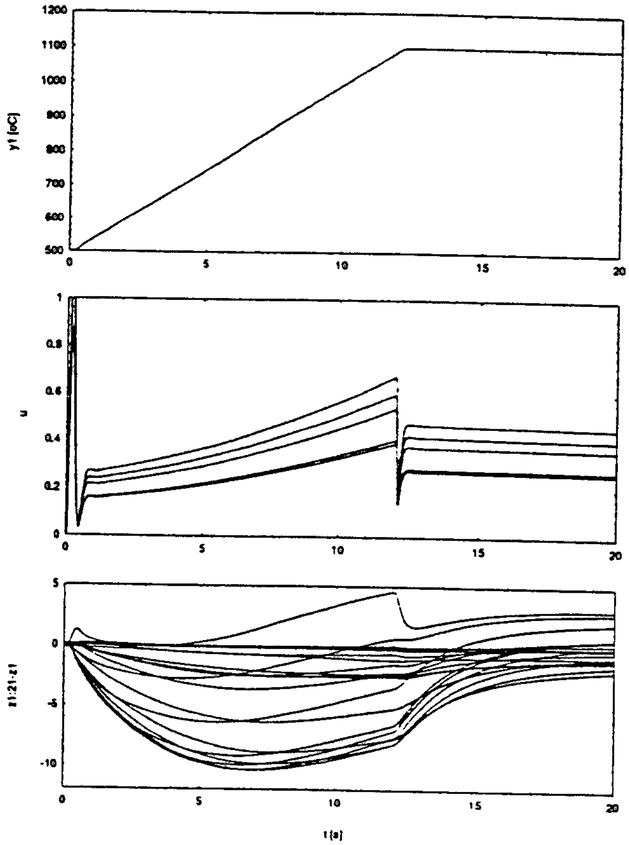


Figure 3: Closed-loop response using $C_1$ and the nonlinear model in (1). Center temperature response (top). Individual lamp inputs, with fixed ratios (center). Wafer spatial uniformity with respect to the center temperature, $(z - z_1 \mathbf{1})$ (bottom).

The associated 1-input 1-output controller $C_1$ and

5-input 5-output $C_5$ were determined by solving $P_1Q_1 = f$ and $P_5Q_5 = fI$ , where $f(s) = \frac{3\sigma^2 s+\sigma^3}{(s+\sigma)^3}$ to guarantee a decoupling ramp-tracking controller. The resulting feedback controllers were simulated with (1) by introducing appropriate offsets, saturation and initializations. The results are shown in Figure 3 and Figure 4 .

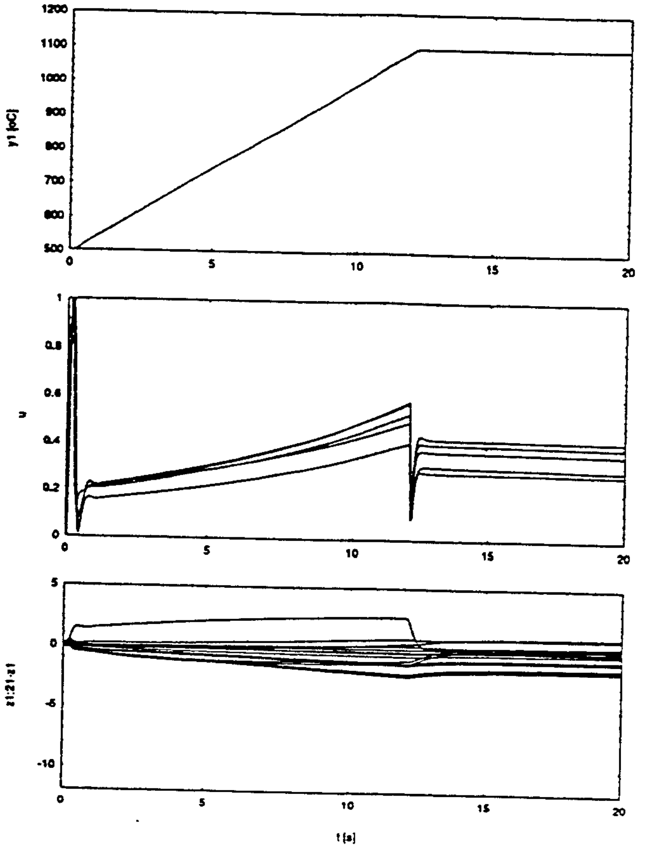


Figure 4: Closed-loop response using $C_5$ and the nonlinear model in (1). Center temperature response (top). Individual lamp inputs (center). Wafer spatial uniformity with respect to the center temperature, $(z - z_1\mathbf{1})$ (bottom).

Since integral action requires $P(0)Q(0) = I$ , and the actuation effort at steady-state is related to $Q(0)$ , good conditioning of $P(0)$ is generally adopted as a guideline. While this guideline is useful in weeding out cases with order of magnitude differences, it is not a complete answer to rank actuator groupings and sensor locations used in feedback. A systematic procedure was discussed using DC-gain matrices from $(w, u)$ to $(z, y)$ to extend the conditioning argument to a best worst-case closed-loop performance in (Kosut and Kabuli, 1995). The approach in Section 4 and the dynamic responses as in Figure 3 complement such condition number based approaches in choosing the pyrometer locations and/or lamp groupings.

## 5. CONCLUDING REMARKS

First-principles based nonlinear dynamic model of a generic rapid thermal processing chamber is used to illustrate a solution to a control/structure interaction problem. A computational toolset is developed to rank a particular chamber design according to steady-state wafer uniformity analysis for constant inputs as well as achievable transient uniformity while ramping using a baseline integral action controller. For different chamber design parameters, the user can determine operating points, perform steady-state uniformity tradeoffs, derive reduced-order small signal equivalent models, design step- or ramp-tracking feedback controllers and observe a baseline dynamic response for a given chamber design.

## ACKNOWLEDGEMENT

The authors thank Jon Ebert for developing the model used in this paper and for providing valuable insight.

## REFERENCES

Incropera, F. P. and D. P. DeWitt (1985). *Introduction to Heat Transfer*, John Wiley & Sons.

Ebert, J. L., A. Emami-Naeini and R. L. Kosut (1995). Thermal modeling of rapid thermal processing systems. *3rd International Rapid Thermal Processing Conference*, R. B. Fair, B. Lojek (eds), pp. 343–355, Amsterdam, The Netherlands.

Kosut, R. L. and M. G. Kabuli (1995). On actuator-sensor selection in thermal processes. *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 2482–2483, New Orleans, Louisiana, USA.

## A.6 Nonlinear model reduction with application to rapid thermal processing

- H. Aling, R.L. Kosut, A. Emami-Naeini, and J. L. Ebert, "Nonlinear model reduction with application to rapid thermal processing," *Proc. 35th IEEE CDC*, pp. 4305-4310, Kobe, Japan, Dec. 1996.

Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

# Nonlinear Model Reduction with Application to Rapid Thermal Processing *

H. Aling, R.L. Kosut, A. Emami-Naeini, J.L. Ebert †

**Abstract:** The Proper Orthogonal Decomposition (POD), also called ***snapshot*** method[1, 2], is a nonlinear model order reduction method where reduction of the size of the state space is achieved using a singular value decomposition of a matrix of snapshots of the state vector. This method has been shown to work well for a simple lumped physical model of a rapid thermal processing (RTP) chamber. Although a substantial reduction of the number of states is achieved, some numerical computations still need to be performed in the high-dimensional state which is computationally expensive. In this paper we will demonstrate how this can be avoided using aggregation of terms, resulting in a significant model simulation speed improvement.

**Notation:** The notation $x^k$ is used for both vectors and matrices where ***every individual element*** is raised to the power k , except for the case of invertible matrices where $k = -1$ it denotes the regular matrix inversion. Likewise, the . notation in front of operators on vectors means an elementwise operation such as $(a./b)_i = a_i/b_i$ , or $(a.b)_i = a_i b_i$ . We sometimes use the notation 1 for a vector of ones. For a vector of indices $\sigma$ , we will write $x_\sigma = \Pi_\sigma x$ for a vector consisting of the elements of $x$ indexed by the index vector $\sigma$ .

## 1 Introduction

We have recently reported successful results on nonlinear model reduction using the *Proper Orthogonal Decomposition* (POD) or ***snapshot*** method [3, 4] . The snapshot method was capable of providing a significant reduction in the number of integrators. However, the results did not harness the full potential of computational savings promised by the snapshot method in the sense that one nonlinear term still had to be computed in the original high-dimensional state space coordinates, rather than in the low-dimensional state space of the reduced-order model.

More specifically, we applied the snapshot method to the following simplified high-order lumped physical thermal model equations:

$$\dot{x} = M^{-1}\left[A_c x + A_r x^4 + Bu + C\right] \qquad (1)$$

Here, $M$ is the (diagonal) thermal mass matrix, $A_c$ and $A_r$ represent the thermal conductivity and radiative exchange matrices, respectively, $B$ is the lamp power input matrix, $u \in R^m$ is the lamp power, and $C$ is a constant term determined by the steady-state with lamp power equal to zero (see [5] for details).

The basis for the snapshot method is formed by a singular value decomposition of the snapshot matrix $X$ :

$$X = U\Sigma V^T = [U_1\ U_2]\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}[V_1^T\ V_2^T]$$

Here, $X = (x_1, \ldots, x_M)$ where the $x_i$ $(i = 1, \ldots, M)$ are the state snapshots at time $t_i$ and where $U_i$, $V_i$ and $\Sigma_i$ $(i = 1, 2)$ are determined by a suitable truncation of $\Sigma$ to the first $n$ singular values. If the state dimension is $N$ then $X$ is a $(N \times M)$ matrix.

Assuming that the snapshots are representative for the state vectors that occur during model simulation, we can approximate $x$ by $\hat{x} = U_1 z$ where $z$ is obtained by substituting this term in (1) :

$$\begin{aligned} \dot{z} &= U_1^T M^{-1}\left[A_c U_1 z + A_r (U_1 z)^4 + Bu + C\right] \\ &= A_c^r z + A_r^r (U_1 z)^4 + B^r u + C^r \qquad (2) \end{aligned}$$

where

$$\begin{aligned} A_c^r &= U_1^T M^{-1} A_c U_1 \in R^{n\times n} \\ A_r^r &= U_1^T M^{-1} A_r \in R^{n\times N} \\ B^r &= U_1^T M^{-1} B \in R^{n\times m} \\ C^r &= U_1^T M^{-1} C \in R^n \end{aligned}$$

*This work is supported by the Advanced Research Projects Agency (ARPA) under Contract No. N00014-94-C-0187.

†Integrated Systems, 201 Moffet Park Drive, Sunnyvale, CA 94089

It is obvious that when $\Sigma_2 = 0$ , then $x = \hat{x}$ . When $\Sigma$ is nonzero but small, $\hat{x}$ will be close to $x$ . Although the exact nature of this approximation is still under study, the method seems to work quite well for these particular model equations.

The number of computations required to simulate (2) is significantly less than that of (1) . However, computation of the term $A_r^r(U_1 z)^4$ remains a problem, especially if $N$ is large (typical values are $N = 5000$ and $n = 10$ [1] ) . This implies that potentially there is room for improvement in simulation speed by a factor of several hundreds if this term can be reduced to a lower-order expression. It turns out that this is indeed possible. How it can be achieved will be explained in the following sections.

**Caveat:** If the model (1) is obtained from a PDE via a Finite Elements approximation, then it is possible to avoid the problem addressed in this paper. However, it is often the case that (1) is obtained from a discrete lumped model with no PDE available, in which case the method given here is appropriate.

## 2 Truly Reduced-Order Modeling by Aggregation

One obvious way of avoiding the computation of $(U_1 z)^4$ and its premultiplication by $A_r^r$ is to expand $A_r^r(U_1 z)^4$ in powers of elements of the vector $z$ . Unfortunately the number of coefficients can be extremely large. For instance, if $n = 10$ , every element of $(U_1 z)^4$ contains many thousands of coefficients which makes this method very impractical.

Instead of analytically computing the coefficients, one could of course compute a polynomial approximation of $A_r^r(U_1 z)^4$ using a least squares fit of the form $\Theta_r \phi(z)$ where $\phi$ is a vector of polynomial terms in the elements of $z$ . We have tried this, using snapshot-based values of $z$ to construct matrices of regression variables, with the choice $\phi(z) = \left[\begin{array}{ccccc} 1 & z^T & z^{2^T} & z^{3^T} & z^{4^T} \end{array}\right]^T$ .

In this least squares fit we used the term $A_r^r(U_1 z)^4$ instead of $A_r^r x^4$ since the former is only an approximation of the latter, the exact term that we wish to approximate.

$$[A_r^r x^4, \ldots, A_r^r x^4] \approx \hat{\Theta}_r \left[\begin{array}{ccc} 1 & \ldots & 1 \\ z_1 & \ldots & z_M \\ z_1^2 & \ldots & z_M^2 \\ z_1^3 & \ldots & z_M^3 \\ z_1^4 & \ldots & z_M^4 \end{array}\right]$$

Unfortunately,
using the term $\hat{\Theta}_r \left[\begin{array}{ccccc} 1 & z^T & z^{2^T} & z^{3^T} & z^{4^T} \end{array}\right]^T$ as a substitute for $A_r^r(U_1 z)^4$ resulted in an extremely numerically-unstable simulation. Closer inspection revealed that the approximation was at best reasonable only close to the region where the snapshots had been taken. Obviously, expansion in straight polynomial terms is not very robust.

The next approach that could have been tried would be to replace the regression vectors 1, $z$, $z^2$ and $z^3$ with a variety of purely 4th order terms and try again. Since there are many such terms, this would have resulted in an endless search for the optimal set of regression variables.

Fortunately it is not necessary to do this. It turns out that there is a very natural and convenient choice of regression variables that works much better, namely one of the form $(\Pi_\sigma U_1 z)^4$ where $\sigma$ is a suitable index vector. If $\sigma$ is chosen as the vector of node indices corresponding with the nodes that undergo most of the excitation from the individually perturbed lamp power inputs such as the 5 lamp temperatures and 5 evenly distributed wafer temperatures, complemented with a single window and showerhead temperature, a near-perfect fit $\hat{\Theta}_r(\Pi_\sigma U_1 z)^4$ of $A_r^r x^4$ is obtained. For a physical interpretation of these variables we refer to Figure 1 where a diagram of the generic RTP model is displayed [5]. A diagram of the RTP chamber with the selected nodes is shown in Figure 1 . Using this term in the actual simulation yielded results that were almost indistinguishable from that of the original low-order model equations, requiring much less simulation time. An explanation of how this is possible follows.

With discretized thermal systems where the states represent the node temperatures, oftentimes many states display a similar behavior. More precisely, let us assume that the state vector $x$ can be approximated as $x = Sx_\sigma$ where $\sigma$ is an index vector of certain representative states, and $S$ is a matrix with only 1 nonzero element in every row. In other words, all temperatures are grouped and

[1] For practical reasons, we have limited our full-order model to $N = 116$ nodes. However, we believe that the same methodology would work for much higher model orders.

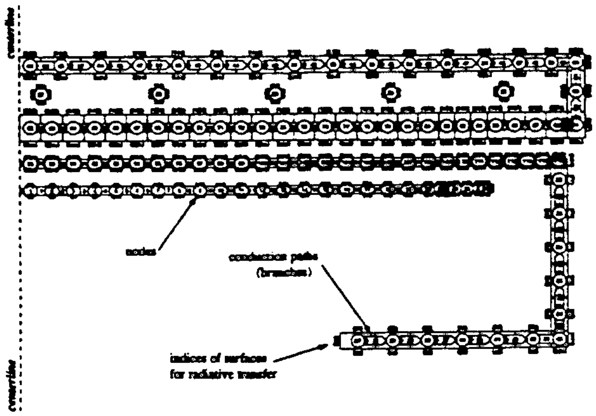


Figure 1: Schematic of the axisymmetric generic RTP system

within every group the signals differ only by a constant factor. We will refer to this property as the property of *aggregation*, or by saying that $x$ is *aggregated*. Under the assumption of aggregation,

$$A_r^r(U_1 z)^4 \approx A_r^r(Sx_\sigma)^4 = A_r^r S^4 x_\sigma^4 \approx A_r^r S^4(\Pi_\sigma U_1 z)^4$$

This suggests that it is possible to make a good least squares fit using the snapshot data of the form

$$A_r^r(U_1 z)^4 \approx \Theta_r(\Pi_\sigma U_1 z)^4$$

This approximation is orders of magnitude more efficient to compute since the number of columns of $\Theta_r$ is much less than that of $A_r^r$ . Note that the assumption of aggregation was made only for the purpose of finding a suitable set of regression vectors, and that it has no direct implication for the properties of the reduced-order model. However in cases where it does not quite hold, the least squares fit may still be able to correct the situation since it simply provides the best fit on the set of regression variables, regardless of the underlying assumptions.

We would like to emphasize that this approach requires only minor engineering judgement regarding the selection of nodes that are used in the least squares fit. This makes the method physically pleasing, while it is also quite objective and requires no quantitative model tuning as is the case with aggregation-based model reduction.

# 3 Application to a More General Model Structure

The model equations (1) are unduly restrictive, and it is necessary to investigate if true reduced-order modeling can be applied to more general RTP chamber models. For this, we will investigate the following set of model equations:

$$\begin{aligned}
\dot{x} &= M(x)^{-1}\left[-C(x) - R(x) + Bu - W(x)\right] \\
R(x) &= A_r^G\left((1-\beta(x)).x^4\right) + A_r^L\left(\beta(x).x^4\right) + A_r^0 \\
W(x) &= K_l(x - T_l) \\
C(x) &= A_c\,\mathrm{diag}\left(\Gamma(A_{cp}x)\right)A_c^T x \\
\Gamma(A_{cp}x) &= \kappa(A_{cp}x)./(A_{cp}x) \\
M(x) &= \mathrm{diag}(m.\eta(x)./x) \qquad (3)
\end{aligned}$$

Here, the variables have the following interpretation:

| | |
|---|---|
| $x$ | State vector (node temperatures) |
| $R$ | Radiation loss term |
| $W$ | Wall loss term |
| $C$ | Conduction loss term |
| $M$ | Thermal mass matrix |
| $\beta$ | Black-body radiation fraction |
| $A_c$, $A_{cp}$ | Sparse matrices representing branch conductivity |
| $\kappa$, $\eta$ | Scalar polynomials, used for branch conductivity and specific heat |
| $m$, $K_l$, $T_l$ | Constant vectors |

The difference between (3) and the simple model (1) is twofold: several matrices have been made temperature-dependent, and the radiative transfer has been separated into a two frequency-band region (related to $A_r^L$ and $A_r^G$) .

We take the same approach as in section 2 : write $x = U_1 z$ , then making least squares fits to approximate several terms to reduce all computations to low order using aggregation. Due to the dependence of the thermal mass matrix $M(x)$ on the state it is more convenient to keep the $M(x)$ term to the left hand side:

$$M(x)\dot{x} = -C(x) - R(x) + Bu - W(x)$$

Substituting $x = U_1 z$ ,

$$M(U_1 z)U_1\dot{z} = -C(U_1 z) - R(U_1 z) + Bu - W(U_1 z)$$

Under the assumption of aggregation, we can reduce the size of the computations by selecting the representative states indexed by $\sigma$ on both sides:

$$\Pi_\sigma M(U_1 z)U_1\dot{z} = -C_\sigma(U_1 z) - R_\sigma(U_1 z) + B_\sigma u - W_\sigma(U_1 z)$$

Now we can focus on approximation of the right hand side terms. Only the terms $R_\sigma(x)$ and $C_\sigma(x)$ require some further investigation.

At this point some remarks on the comparison between the generic reduced-order model equation (3) and that of the simplified generic model (2) are in order. With (2) , we were able to get $\dot{z}$ to the left hand side by premultiplying with $U_1^T$ . By virtue of the properties of the fact that $U_1$ is composed of the set of (orthogonal) left singular vectors, this premultiplication represents a nicely balanced way of obtaining the reduced-order model equations. However, with (3) we have to perform a left-inversion of the term $\Pi_\sigma M(U_1 z)U_1$ to achieve the same. The numerical conditioning of this inversion depends strongly on the selection $\sigma$ and may require adding in some more nodes.

We will therefore end up with two orders: one for the number of integrators (dimension of $z$) , and one for the number of selections (dimension of $\sigma$) to approximate the nonlinear behavior, and to precondition the left-inversion of (3) . Although it may seem as if the determination of $\sigma$ and the related left-inversion introduce too much subjectivity into this approach, it also gives us a unique chance to extract exactly that behavior that focuses on specific components of the system (wafer temperature) that are crucial for control design purposes. Much less weight is attached to other states that are only required to support the dynamic behavior of the reduced-order model, they are approximated in a fairly crude way.

## The Radiative Loss Term $R_\sigma(x)$

The selected components of $R(x)$ are given by

$$\begin{aligned} R_\sigma(x) = \Pi_\sigma A_\tau^G\ ((1-\beta(x)).x^4) +& \\ \Pi_\sigma A_\tau^L\ (\beta(x).x^4) + \Pi_\sigma A_\tau^0& \end{aligned}$$

where the subscript $_\sigma$ indicates selection of the components indexed by $\sigma$ . The expression for $\beta(u)$ where $u$ is a scalar is of the form

$$\begin{aligned} \beta(u) &= c_1 p(\frac{k}{u}) + c_2 p(2\frac{k}{u}) + c_3 p(3\frac{k}{u}) \\ p(u) &= e^{-u}(u^3 + 3u^2 + 6u + 6) \end{aligned}$$

Based on this, we will try the following least squares fits:

$$\begin{aligned} \Pi_\sigma A_\tau^G(1-\beta(x)).x^4) &\approx \Theta_\tau^G(1-\beta(x_\sigma)).x_\sigma^4) \\ \Pi_\sigma A_\tau^L(\beta(x).x^4) &\approx \Theta_\tau^L(\beta(x_\sigma).x_\sigma^4) \end{aligned}$$

where the values for $x$ and $\Pi_\sigma x$ are taken from the snapshots.

## The Conduction Loss Term $C(x)$

Reduction of $C(x)$ is the hardest part of the model reduction procedure. It should be noted that the temperature-dependence reflected by $\Gamma(A_{cp}x)$ is generally quite small, so in first instance one could approximate $\Gamma(A_{cp}x)$ by a constant and simply precompute the $(n \times n)$ matrix $A_c$ diag$(\Gamma)A_c^T U_1$ after which the job is completed. This might work quite well, especially since the radiation term is the dominant factor in RTP chamber models. In cases where this assumption is not justified one could proceed as follows.

First, we have to notice that $A_c$ and $A_{cp}$ are directly related to the branches between the nodes. More precisely, the columns of $A_c$ contain zeros everywhere except for two elements which have the values 1 and $-1$ . The term $A_c^T x$ is therefore represents temperature differences between nodes. Similarly, $A_{cp}$ contains zeros everywhere except for two elements that both have the value 1, therefore it represents average temperatures of nodes.

Because of the physical interpretation, we cannot select components based on $\sigma$ . Instead, we have to group the set of branches between nodes into subsets that display a similar behavior in the same way as what we did for the selection of nodes. This results in a selection $\tau$ by picking one representative element from every subset. The natural consequence of our approach is to fit the following model approximation:

$$\begin{aligned} \Pi_\tau A_c\ \text{diag}(\Gamma(A_{cp}x))A_c^T x \approx& \\ \Theta_c\ \text{diag}(\Gamma(\Pi_\tau A_{cp}x))\Pi_\tau A_c^T x& \qquad (4) \end{aligned}$$

where the matrices $A_{c,\tau}$ and $A_{cp,\tau}$ have been constructed in the same way as $A_c$ and $A_{cp}$ , based on branches between groups of signals instead of individual signals.

Doing this requires more engineering judgement than we would like to use, therefore we will first try a simpler approach. Assume that the conductivity term of a branch between two nodes $x_1$ and $x_2$ is described by $p((x_1 + x_2)/2)(x_1 - x_2)$ where $p(x)$ is a polynomial. This corresponds with one branch element of (4) . Assuming that the temperatures $x_1$ and $x_2$ are approximately the same, and that $p(x)$ can be approximated by a second order polynomial $p(x) = p_0 + p_1 x + p_2 x^2$ , we can write

$$p((x_1 + x_2)/2)(x_1 - x_2) \approx (p_0 + p_1 x_1 + p_2 x_1^2)x_1 \\ -(p_0 + p_1 x_2 + p_2 x_2^2)x_2$$

Here, we will substitute the elements of $x_\sigma$ for $x_1$ and $x_2$ for all branch conductivity approximations of this form. Again, we rely on the least squares fit to compensate for any approximation errors caused by our simplifying assumptions.

# 4 Results

## The Simplified Generic Model

We used 830 samples of the same dataset as was described in [3, 4] for the selection of snapshots. A selected set of signals of this dataset (lamp, wafer, showerhead and window temperatures) is depicted in Figure 2 , where the different characteristics of these four groups are clearly recognizable. The input for the dataset consists of a nominal input sequence to which a PRBS-based input disturbance was added for persistent excitation. This disturbance can be described as the sum of four individual PRBS sequences where each sequence has a bandwidth based on the dominant time constant of one of the four signal groups, with every component of the five lamp power inputs excited independently. Using the simplified generic model reduction described by (2) , several models were obtained for different choices of snapshots and model orders. These models were then validated on the nominal dataset which is stochastically independent of the dataset used for model reduction, and which therefore is a legitimate validation dataset. The results can be summarized by stating that a 12th order model based on 70 snapshots resulted in a 2 degrees RMS validation error which is an excellent result. This was based on a selection $\sigma$ of 5 lamp temperatures, 5 wafer temperatures, 1 showerhead and 1 window temperature (12 elements).

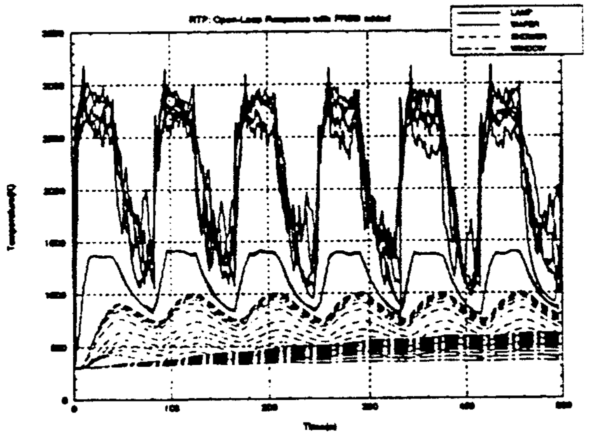


Figure 2: The PRBS-perturbed nominal trajectory

## The Generic Model

The more difficult task of reducing the generic model was accomplished in much the same way without great difficulty. The model order used was 12, while the selection $\sigma$ was based on 5 lamp temperatures, 5 wafer temperatures, 5 showerhead temperatures, 1 window and 1 wall temperature (17 elements). The 5 extra selections significantly improved the simulation of the other, non-selected wafer and showerhead temperatures. Figures 3 and 4 show the simulation results of the 21 wafer and 28 showerhead temperatures, respectively.

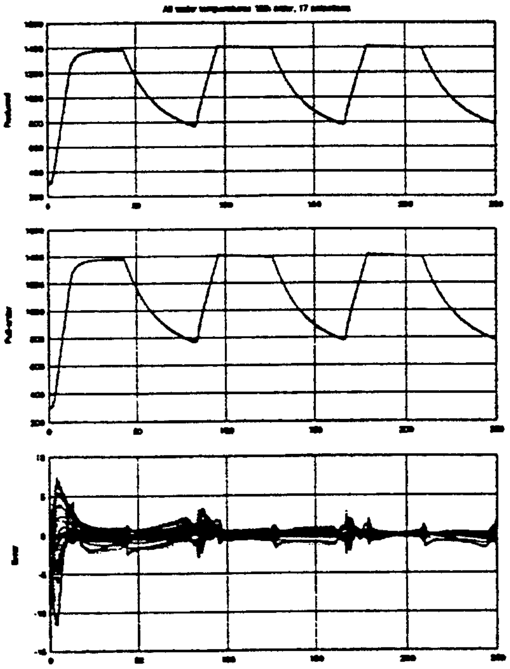


Figure 3: Wafer temperatures predicted by the reduced-order model

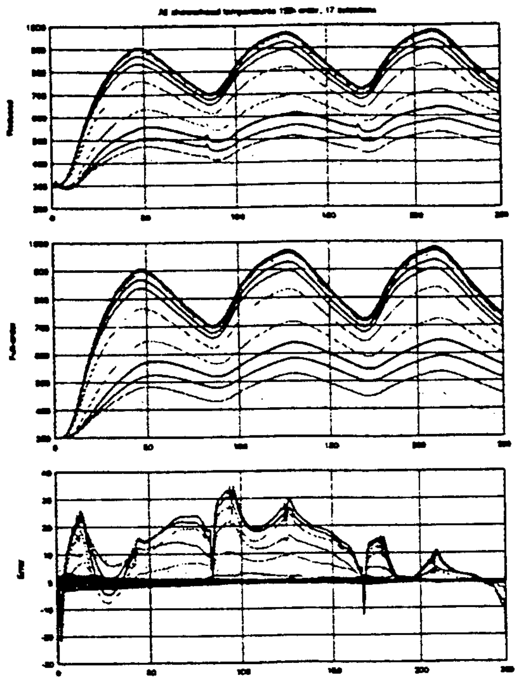


Figure 4: Showerhead temperatures predicted by the reduced-order model

The most important result is that the wafer temperatures are simulated with an error that is generally less than 2 degrees which is an excellent result. There are some off-center showerhead temperatures that deviate more than 20 degrees, but those states are only of secondary importance for our model reduction effort. Instead, it is shown that we can model the wafer temperatures quite well with a low-order, nonlinear model by making appropriate signal selections. This type of model reduction can be extremely useful for the implementation of controllers, by providing on-line nonlinear model estimates of the temperatures.

## 5 Summary and Conclusions

We have applied the snapshot method to the generic RTP chamber model and have demonstrated how the numerical efficiency can be improved substantially by reducing the nonlinear terms using least squares. The model reduction is based on a selection of singular values for the number of integrators, and a selection of aggregated states for reduction of the nonlinear terms. Some engineering judgement is required for a proper selection. When this is done correctly, low-order nonlinear models are obtained that are capable of predicting several states corresponding with nodes near the ones that were selected for aggragation very accurately. Although the precise nature of the approximations is a topic for further research, the approach seems intuitive and very promising for model reduction of high order lumped physical models.

## Acknowledgements

This is joint work with MIT and Princeton University. In particular, we wish to thank I.G. Kevrekidis (Princeton) and K. Jensen (MIT) for the helpful discussions that we had with them.

## References

[1] L. Sirovich, C.H. Sirovich: *Low dimensional Description of Complicated Phenomena,* Contemporary Mathematics, Vol. 99, 1989, pp. 277-305.

[2] L. Sirovich: *Turbulence and the Dynamics of Coherent Structures parts I, II and III,* Quarterly of Applied Mathematics, Vol. XLV, 1987, no. 3, pp. 561-571, pp. 573-582, pp. 583-590.

[3] *Application of a Nonlinear Model Reduction Method to Rapid Thermal Processing (RTP) Reactors*, H. Aling, J.L. Ebert, A. Emami-Naeini, R.L. Kosut, to appear in IFAC-96.

[4] *Application of Feedback Linearization to Model of Rapid Thermal Processing (RTP) Reactors*, H.Aling, J. Abedor, J.L. Ebert, A. Emami-Naeini, R.L. Kosut, 3rd International RTP Conference, August '95, Amsterdam, Proc. RTP-'95 pp. 356-366.

[5] *Thermal Modeling of Rapid Thermal Processing Systems*, J.L. Ebert, A. Emami-Naeini, R.L. Kosut, 3rd International RTP Conference, August '95, Amsterdam, Proc. RTP-'95 pp. 343-355.

## A.7 Learning Feedforward Control

- K.M. Tao, R.L. Kosut and G. Aral, "Learning Feedforward Control," *Proc. American Control Conference*, Baltimore, MD, June - July 1994. pp. 2575 - 2579.

Proceedings of the
American Control Conference
Baltimore, Maryland • June 1994

FA8 - 10:00

# Learning Feedforward Control*

K. Mike Tao, Robert L. Kosut and Gurcan Aral
Integrated Systems, Inc.
3260 Jay Street, Santa Clara, CA 95054

ABSTRACT

*Some new results in learning feedforward control are presented in this paper. The proposed procedures are applicable to processes with or without feedback control. Under the ideal situation, this* discrete-time *Learning Control (LC) scheme can perfect the task with one repeat. Convergence analyses are also included for noisy and imprecise learning. New algorithms are proposed to address these noisy learning issues. Interesting connections between the proposed approach and iterative image deblurring, LQ control and Kalman filtering are identified. With all the required process knowledge readily measurable, the proposed scheme is relatively simple to implement and can be used as an on-site feedforward* tuning *tool. Encouraging simulation results are included.*

## 1 Introduction

It is well established that a properly designed feedforward (FF) control can complement feedback (FB) control by promoting non-delay and anitcipatory actions, leading to superior tracking or disturbance rejection. Unlike the traditional FF design which is usually based on analytical modeling, therefore often difficult to *tune*, the Learning Control (LC) approach described is suitable for *on-site tuning* as it can *learn* from the past experiences. (Many manufacturing tasks are repetitive, task-oriented, and are potential applications of LC).

Unlike some of the published works, e.g., [1, 2], which use proportional, proportional + derivative, or state errors for iterative learning, this paper focusses on (discrete-time) linear time-invariant (LTI) *predictive control* type techniques which can lead to rapid convergence. Under ideal situations, this LC scheme can perfect the tracking task with one repeat (Section 2). The only required knowledge is the impulse responses which can be readily measured on-site. In that regard, our scheme is somewhat similar to the one in [3]. However, unlike [3], our results are applicable to multi-input, multi-output (MIMO) cases and are less restrictive in assumptions. It can also be shown that our results lead to the main convergence results in [3]. Convergence analyses for noisy and imprecise cases (Section 3) are included, and new algorithms are proposed to address these issues (Sections 4, 5, 6). Rapid learning is demonstrated by the simulation of a MIMO process. Interesting and important connections with iterative image deblurring [7], LQ control and Kalman filtering are identified. These connections allow a unifying *system inversion* viewpoint and the sharing of algorithmic ideas from seemingly unrelated fields.

## 2 The Basic Learning Algorithm

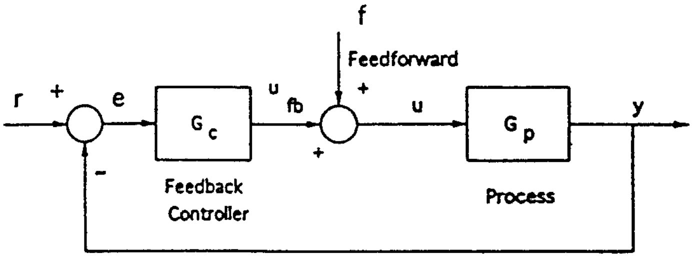


Figure 1: Combined Feedforward and Feedback Configuration

Consider a combined FB and FF control configuration as shown in Figure 1. The basic LC scheme and its variants can be described by:

$$f^{(i+1)} = f^{(i)} - Ke^{(i)} \tag{1}$$

where $e^{(i)}$ is the tracking error vector, [1] and $f^{(i)}$ the FF signal in the $i$-th iteration and $K$ is a *general* (linear) operator which effects LC. This LC operator, $K$, may include *time-advance* operations. [2] Realizing that both $f$ and the command $r$ contribute to $e$, straightforward analysis yields the following relationship in noiseless conditions:

$$f^{(i+1)} = (I - K\mathcal{M})f^{(i)} - K\Gamma r \tag{2}$$

where $\mathcal{M}$ is the transfer operator from $f$ to $e$, and $\Gamma$ is the transfer operator from the command $r$ to $e$:

$$\mathcal{M} = -(I + G_pG_c)^{-1}G_p \tag{3}$$

$$\Gamma = I - (I + G_pG_c)^{-1}G_pG_c \tag{4}$$

Notice that *if* $K$ is chosen as the inverse of $\mathcal{M}$, i.e., the transfer from $f$ to $e$, then exact convergence with one repeat is possible. This is because

$$\begin{aligned} K &= \mathcal{M}^{-1} && (5)\\ f^{(i+1)} &= (I - \mathcal{M}^{-1}\mathcal{M})f^{(i)} - \mathcal{M}^{-1}\Gamma r \\ &= G_p^{-1}r && (6) \end{aligned}$$

where $f = G_p^{-1}r$ is the "perfect" FF control signal for reproducing the desired trajectory $r$. [3]

A more relaxed condition for convergence is that

$$|\lambda_i(I - K\mathcal{M})| < 1 \tag{7}$$

or

$$0 < \lambda_i(K\mathcal{M}) < 2 \tag{8}$$

*Work supported in part by ARPA under AFOSR Contract No. F49620-94-C-0003.

[1] Under certain conditions, the signal, $\tilde{u}_{fb}$, instead of the tracking error $e$, may be used for iterative learning, where $\tilde{u}_{fb}$, is the *deviation* of the FB controller output from its initial steady state. This is because if the FF is doing a perfect job for the transient, then the FB controller would not have to labor.

[2] Time-advancing is necessary to implement correct "credit" assignment for learning. This is because that $f$ applied at time $t$ will not affect the process or $e$ until time $t + 1$ or later.

[3] The meanings and existence of $G_p^{-1}$, and the interpretation of the algorithm when $G_p^{-1}$ does not exist are explained at the end of this section.

where $\lambda_i$ denotes the eigenvalues. This relaxed condition allows the use of a broader scope of algorithms ranging from cases where $\mathcal{M}^{-1}$ is not known exactly (e.g., due to measurement imprecision or process nonlinearity) to gradient descent type of optimization algorithms described in Section 5. Now, instead of converging in one repeat, it will take multiple repeats to achieve satisfactory results. [4]

Computationally, this basic algorithm involves the following:

$$
\begin{aligned}
-e &= -\begin{bmatrix} e(2) \\ e(3) \\ e(4) \\ \cdot \\ \cdot \\ \cdot \\ e(N+1) \end{bmatrix} \\
&= \begin{bmatrix} h_1 & 0 & 0 & \dots & 0 \\ h_2 & h_1 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ h_N & h_{N-1} & h_{N-2} & \dots & h_1 \end{bmatrix} \begin{bmatrix} \delta f(1) \\ \delta f(2) \\ \delta f(3) \\ \cdot \\ \cdot \\ \delta f(N) \end{bmatrix} \\
&= M\delta f \qquad (9) \\
f^{(i+1)} &= f^{(i)} + \delta f^{(i)} \\
&= f^{(i)} - M^{-1}e^{(i)} \qquad (10)
\end{aligned}
$$

where $h_1, h_2, h_3, ...h_N$ is the closed-loop impulse response sequence from $f$ to $e$. Its state-space equivalent would be $CB, CAB, CA^2B, ..., CA^{N-1}B$ where [A,B,C] stands for a state-space representation of the $f$ to e process. Matrix $M$ is a time-domain realization of the transfer operator, $\mathcal{M}$.

Using the impulse response representation directly facilitates on-site measurement and simpler computation. Alternatively, one may replace the impulse responses in the matrix by step responses and solve for $\Delta\delta f(t) = \delta f(t+1) - \delta f(t)$ instead. [5]

This procedure is validated using a simulated 3 x 3 LTI MIMO process stabilized with decentralized PI controllers. Figure 2 illustrates the system performance in tracking a ramp command. The top plots show the tracking performance with feedback control only; the bottom plots the performance after one repeat of the task. The impulse response matrix is estimated by differencing the "measured" step responses. The simulation is done using the graphics oriented control design software SystemBuild and MATRIXx. The solid lines indicate the command trajectories and the dashed lines the actual process responses. It is apparent that in this idealized situation, the basic learning approach achieves perfect tracking in one learning repeat. Issues with noisy cases are addressed in Sections 3, 4, 5 and 6. Before closing this section, some remarks are in order:

**Remark 1:**
In the rest of the paper, the *class* of learning scheme will be formalized as:

$$
\begin{aligned}
f^{(i+1)} &= f^{(i)} + \delta f^{(i)} \qquad (11) \\
\delta f^{(i)} &= -K\mathcal{E}^{(i)}_{f^{(i)}} \qquad (12)
\end{aligned}
$$

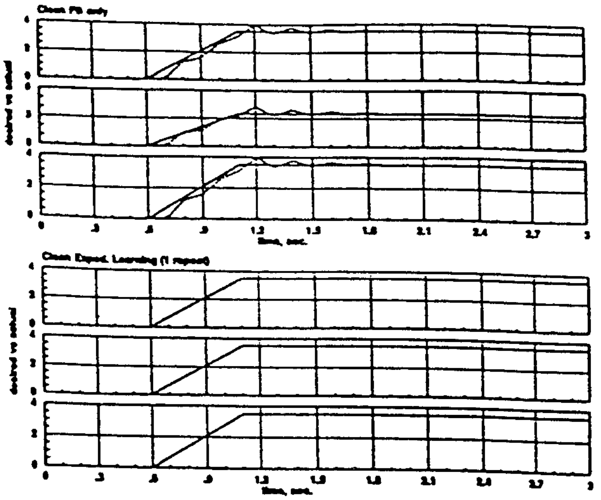


Figure 2: FF LC via the Basic Learning Algorithm

where $\mathcal{E}^{(i)}_{f^{(i)}}$ is used to represent the error (i.e., $e^{(i)}$ or $\tilde{u}^{(i)}_{fb}$) observed in i-th task repeat with $f^{(i)}$ applied. This more expressive notation is needed in Secton 4 where certain noise-averaging learning algorithms are developed.

**Remark 2:**
The use of $G_p^{-1}$ assumes the existence of process "right inverse" [4]. [6] A necessary condition is that the number of process inputs must be no less than the number of outputs. With the existence of $G_p^{-1}$, the existence of $\mathcal{M}^{-1}$ for the closed-loop is assured.

When the $M$ matrix is 1-delay, [7] the matrix $h_1$ (or $CB$ in state-space notation) has rank n. The $M$ matrix in Eq. (9) is invertible. This (restrictive) condition is assumed by some of the published work, e.g., [1, 3]. When the total delay is greater than 1, the $M$ matrix as is presented in Eq. (9), would be rank deficient. In that case, one can either solve the equation as is, but in a least-squares sense, or shift the error vector, $e$, on the left-hand-side so that it would start with $t = L + 1$.

In either case, the notion of least-squares is important because it allows flexibility in seeking meaningful, practical solutions including the various forms of constrained optimization solutions discussed later. Constraining the magnitude of control is needed when the process is *non-minimum phase* or is subject to tight physical constraints. Constraining the rate of control is useful to avoid control rate saturation and to reduce the noise effect.

## 3 The Basic Algorithm with Uncertainties

In this section, two types of uncertainties are considered: process

[4] Noting that with $\mathcal{M}$ in the lower triangular form as in Eq. (9), so is $\mathcal{M}^{-1}$. If $K$ is restricted to be lower triangular, then so is $K\mathcal{M}$. The $\lambda$'s of a lower triangular matrix are the diagonals. This leads to the main convergence result, Proposition 3.1, in [3].

[5] Certainly, this procedure does not prohibit one from using more elaborate system identification techniques. The impulse or step response matrices can be generated by a model obtained through parametric identification.

[6] The concept of L-delay inversion by Massey and Sain [5] is of direct relevance here. A discrete-time process $G_p$ is L-delay right invertible if it can *perfectly* track any given command after L-delays. Notice that L is not unique and the smallest L is often of the greatest interest. It is also known that if a process is right invertible then it must be so with no greater than p delays where p is the dimension of the state space – similar property exists for controllability and observability. The following test [5] can determine the minimum delay L:

*A discrete-time process is L-delay right invertible if and only if $rank(M_L) = rank(M_{L-1}) + n$ where n is the process output dimension, and $M_L$ is the $(nL \times mL)$ leading principal minor of M in Eq. (9).*

[7] For presentation, we have assumed that the discrete-time process has an inherent 1 delay due to sampling. This is reflected in Eq. (9) that the error vector $e$ starts from $t = 2$. In reality, the process could have pure time-delays and/or L-delays.

or measurement noise and imprecise impulse response model.

First consider the effect of the process noise, $v$ (added to the output $y$). With the process noise, the learned $f$ obeys the following recursion:

$$f^{(i+1)} = (I - KM)f^{(i)} - K\Gamma r + K\Gamma v^{(i)} \tag{13}$$

where $\Gamma$ is the transfer operator from $r$ to $\mathcal{E}$ (i.e., $e$ or $\tilde{u}_{fb}$), and $v^{(i)}$ is the process noise for the $i$-th repeat.

When K is precisely chosen as $\mathcal{M}^{-1}$, the above recursion reduces to [8]

$$f^{(i+1)} = G_p^{-1}[r - v^{(i)}] \tag{14}$$

The output tracking error after the $(i+1)$th repeat is

$$\begin{aligned} e^{(i+1)} &= r - y^{(i+1)} && (15) \\ &= \Sigma[v^{(i+1} - v^{(i)}] && (16) \end{aligned}$$

where $\Sigma = (I + G_pG_c)^{-1}$ is the transfer function from the process noise, $v$, to process output, $y$.

*Notice that any repeatable disturbance in $v$ is eliminated by this learning procedure. However, non-repeatable random noise is amplified by a factor of $2^{1/2}$ on the standard deviation.*

Next, consider the effect of imprecise impulse response measurements, $\hat{M}$. Assuming $K = \hat{M}^{-1}$, and $KM = I + B_M$ yields

$$I - KM = -B_M \tag{17}$$

and

$$\begin{aligned} f^{(i+1)} &= -B_M f^{(i)} - K\Gamma[r - v^{(i)}] && (18) \\ &= -B_M f^{(i)} - KMM^{-1}\Gamma[r - v^{(i)}] && (19) \\ &= -B_M f^{(i)} + (I + B_M)G_p^{-1}[r - v^{(i)}] && (20) \\ &= G_p^{-1}r + B_M(G_p^{-1}r - f^{(i)}) - (I + B_M)G_p^{-1}v^{(i)} && (21) \end{aligned}$$

According to Eq. (20), the $f$ dynamics represent a "stable" learning process when $B_M$ is dissipative, i.e.,

$$|\lambda_i(B_M)| < 1 \tag{22}$$

(where $\lambda_i$ are the eigenvalues,) and $r$ and $v$ are finite. [9]

Comparing Eq. (21) with Eq. (14), it is noticed that impulse response modeling error slows down the convergence and also increases the potential effect of the noise.

## 4 Some Noise Reduction Techniques

It is shown above that process noise or measurement error can potentially degrade the learning performance. This section devotes further attention to these issues.

First of all, conventional signal conditioning and noise filtering techniques should be considered in practical implementations. [10] Unconventional statistical techniques such as *jack-knife sampling* [6] may be used. [11] Next, the basic algorithm may be modified to account for the noisy conditions. Four (4) such techniques are described in this paper. Two are discussed in Section 5: one uses rate constraints to assure the smoothness of the learned $f$; another uses "soft" inversion to avoid amplifying noise. The one based on a Kalman filter formulation is described in Section 6. A noise-averaging technique which can be built into the basic learning algorithm is described in *this* section.

From an optimization viewpoint, the basic learning algorithm eliminates the most recently observed learning error, $e^{(i)}$. The *noise-averaging formulation*, however, replaces $e^{(i)}$ with the *statistically averaged learning error* which takes into consideration all the previous learning trials. Of course, as $f$ evolves with the learning process, portions of the previouse learning errors are due to different $f$. After correcting for the differences in $f$ using the $M$ model, it is possible to keep track of a statistically averaged learning error which assumes a common $f$, but with multiple realizations of the process noise averaged out. Choosing $f$ according to the following Statistically Averaged Inversion (SAI) algorithm minimizes this averaged learning error:

$$f^{(i+1)} = f^{(i)} - M^{-1}(\mathcal{E}^{(i)}_{f^{(i)}}/(i+1)) \tag{23}$$

for $i = 0, 1, 2, \ldots$ . This algorithm has the flavor of *Stochastic Approximation*, and it can be shown that the noise effect on the learned $f$ is asymptotically eliminated as learning continues (provided that the model $M$ is known precisely). It is noticed, however, the correction gain decreases as $i$ increases (i.e., $1/(i+1) = 1, 1/2, 1/3, \ldots$), and the learning process eventually becomes open-loop. This observation along with Eq. (20) suggests that in the case where $M$ is not known precisely, part of the initial error on $f$ may remain as uncorrected bias.

To strike a balance between eliminating the deterministic FF error and minimizing the noise effect, the following Exponentially Averaged Inversion (EAI) algorithm is in order.

$$f^{(i+1)} = f^{(i)} - \hat{M}^{-1}(\gamma\mathcal{E}^{(i)}_{f^{(i)}}) \tag{24}$$

where $0 \leq \gamma \leq 1$ is the exponential averaging constant. When $\gamma = 1$, the algorithm becomes the basic inversion algorithm.

Alternatively, the following variation allows one to choose $\gamma$ *adaptively*:

$$f^{(i+1)} = f^{(i)} - \hat{M}^{-1}(\frac{1}{(i+1)} + \gamma\frac{i}{(i+1)})\mathcal{E}^{(i)}_{f^{(i)}} \tag{25}$$

Notice that the first component in the parenthesis represents pure statistical averaging, whereas the second component is a model error correction term discounted by $\gamma$ $(0 < \gamma < 1)$. $\gamma$ can be chosen according to the ratio between the *excess* $||\mathcal{E}||^2$ and $||\mathcal{E}||^2$, where the *excess* $||\mathcal{E}||^2$ is estimated against the expected noise level and therefore is an indication of modeling error. This choice is similar to the way that the (optimal) Wiener filter is determined.

Finally, it is noted here that the $\gamma$ in EAI can be used to model the *uncertainty* in the initial $f$. If one has high "confidence" in $f^{(0)}$, then small $\gamma$ should be used. However, a more precise account of "confidence" is outlined in the Kalman filter formulation described in Section 6.

## 5 Gradient Algorithms & Constrained Optimization

The basic LC algorithm and its variants described so far require matrix inversion at each step. In optimization theory, this amounts to descending along the Newton's direction. Often, direct system inversion may be less desirable for practical reasons.

[8] Notice that $G_p^{-1}r$ is the "perfect" FF control signal for reproducing r at the process output.

[9] Eq. (22) is the same condition as Eq. (7).

[10] *Non-causal* filtering can be used because of the "off-line" nature of LC.

[11] For example, fast sampling around a regular sample instant provides a pseudo-ensemble of measurements from which a good average reading may be derived without repeating the same experiment multiple times.

In such cases, "soft" inversion via Optimal Gradient Descent is an alternative to exact inversion.

The Exponentially Averaged version of the Optimal Gradient Descent Learning is summarized here. Assuming a post $i$-th repeat situation, $f^{(i)}$, $\delta f^{(i-1)}$, and $\mathcal{E}^{(i)}_{f^{(i)}}$ are given where $\mathcal{E}^{(i)}_{f^{(i)}}$ is the observed learning error with $f^{(i)}$ applied during the $i$-th repeat. Also known is the *averaged* learning error from the previous repeat: $[\mathcal{E}]^{(i-1)}_{f^{(i-1)}}$. The procedure is to first update the *averaged* learning error, i.e., to compute $[\mathcal{E}]^{(i)}_{f^{(i)}}$.

$$[\mathcal{E}]^{(i)}_{f^{(i)}} = (1-\gamma)([\mathcal{E}]^{(i-1)}_{f^{(i-1)}} + \hat{M}\delta f^{(i-1)}) + \gamma\mathcal{E}^{(i)}_{f^{(i)}} \qquad (26)$$

whereas for $i-1=0$, $[\mathcal{E}]^{(0)}_{f^{(0)}} = \mathcal{E}^{(0)}_{f^{(0)}}$.

With this updated *averaged* learning error, the gradient, $g$, of this error squared is computed with respect to $f$, and $f$ is then updated along the negative gradient direction with the optimal step size, $\mu$:

$$\begin{aligned} f^{(i+1)} &= f^{(i)} - \mu g && (27)\\ g &= \hat{M}'[\mathcal{E}]^{(i)}_{f^{(i)}} && (28)\\ \mu &= (g'g)/(g'\hat{M}'\hat{M}g) && (29) \end{aligned}$$

When $\gamma = 1$ this algorithm becomes the (basic) Optimal Gradient Learning algorithm. When $\gamma = 1/(i+1)$, it becomes the Statistically Averaged version.

*Numerically* iterating (infinitely) many times with Optimal Gradient *between* task repeats amounts to exact inversion. However, exact inversion is often not a good idea in the face of noise and uncertainties – a principle also found in iterative image deblurring [7] and adaptive signal processing [9]. [12] In such cases, "soft" inversion may be performed by iterating with optimal gradient descent a *finite* number of times.

Figure 3 illustrates the performance of the Optimal Gradient method in the present of process noise. The top plots are the FB-only control performance (with the solid lines indicating the command and dashed line the noisy process response). The bottom plots show the the Optimal Gradient Learning results after 5 repeats with 10 numerical iterations between task repeats. Again, in this noisy case, the Optimal Gradient learning is able to improve over the feedback-only performance. The built-in noise-averaging mechanism (as described in Section 4) and the "soft" inversion feature are the only noise reduction methods used in this example. With signal conditioning and filtring, additional protections can be expected.

In many practical applications, the mangnitude and/or the rate of control are limited, e.g., by valve position and slew rate. The following outlines formulations and solutions to constrained optimization problems in a (batch) linear quadratic (LQ) setting. [13]

### With (Soft) Constraint on the Magnitude of Control

$$min.J = min.\frac{1}{2}(||\mathcal{E}^{(i+1)}||^2_Q + f^{(i+1)'}Rf^{(i+1)}) \qquad (30)$$

where Q is a weighting matrix on the learning error and R is the control penalty weighting matrix. Q and R should be (at least) non-negative definite and $(M'QM+R)$ should be positive definite. The inversion based solution to this problem is

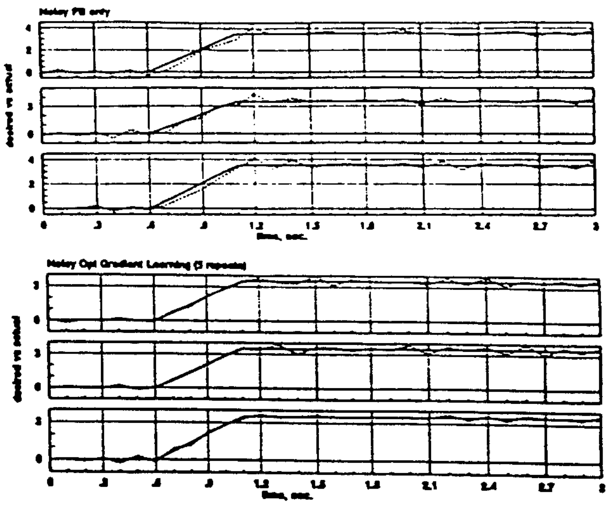


Figure 3: FF via Optimal Gradient Learning (Noisy)

$$\begin{aligned} \delta f^{(i)} &= -(M'QM+R)^{-1}[M'Q\mathcal{E}^{(i)} + Rf^{(i)}] && (31)\\ f^{(i+1)} &= f^{(i)} + \delta f^{(i)} && (32) \end{aligned}$$

Similarities between this constrained problem and the *regularization* solution found in (iterative) image deblurring and restoration problems [7] are worthnoting. In that context, the image is blurred by a transfer function and is also corrupted by noise. To undo the blurring involves convolving the noise-corrupt blurred image with the inverse of the blurring transfer function. Since the inverse of the blurring transfer function tends to amplify the noise, the problem becomes ill-conditioned. In [7], the advantage of "soft" inversion over exact inversion in image restoration is analyzed and demonstrated, in terms of noise conditioning.

Furthermore, *regularization* techniques are used to add constraint on the deblurred image – usually smoothness constraints of some kind, e.g., the $R$ matrix could represent a Laplacian type operator to penalize nonsmooth components [7]. The case *here* is slightly different in that it is to limit the magnitude of the solution, $f$. This particular "regularization" may prevent either violation of physical constraints or numerical difficulty when M is ill-conditioned or when the process is of non-minimum phase. On the other hand, the inclusion of *rate* constraints on the learned $f$ can be motivated by the concerns over the corrupting noise in $\mathcal{E}$ as well as over the actuator rate limitations.

### With (Soft) Constraints on the Magnitude and the Rate of Control

$$min.J = min.\frac{1}{2}(||\mathcal{E}^{(i+1)}||^2_Q + f^{(i+1)'}R_1 f^{(i+1)} + \Delta[f]^{(i+1)'}R_2\Delta[f]^{(i+1)}) \qquad (33)$$

where $R_1$ and $R_2$ are the weighting matrices on the magnitude and the "rate" of control, respectively, and $\Delta[f]$ is the control increment vector defined as

$$\begin{aligned} \Delta f(t) &= f(t+1) - f(t) && (34)\\ \Delta[f] &= \begin{bmatrix} f(0) & \Delta f(1) & . & . & . & \Delta f(N-1) \end{bmatrix}^T && (35) \end{aligned}$$

Notice that

$$f = T\Delta[f] \qquad (36)$$

[12] This connection points to some *robust* techniques in the image and signal processing literature [10, 11]. Also, in [3] a robust technique is provided.

[13] Hard constraints may also be formulated using linear or quadratic programming (LP or QP) type techniques.

$$T = \begin{bmatrix} I & 0 & . & 0 \\ I & I & . & 0 \\ . & . & . & . \\ I & I & . & I \end{bmatrix} \qquad (37)$$

The inversion based solution to this problem is:

$$\begin{aligned} \Delta[\delta f]^{(i)} &= -(\bar{M}'Q\bar{M} + \bar{R})^{-1}[\bar{M}'Q\mathcal{E}^{(i)} + \bar{R}\Delta[f]^{(i)}] && (38) \\ \delta f^{(i)} &= T\Delta[\delta f]^{(i)} && (39) \\ f^{(i+1)} &= f^{(i)} + \delta f^{(i)} && (40) \end{aligned}$$

where $\bar{M} = MT$ and $\bar{R} = T'R_1T + R_2$. By using appropriate $R_2$, not only can one constrain the rate of control for physical reasons, but also the undesired effect due to the corrupting noise.

### Soft-Constrained Optimal Gradient Algorithms

With constraint on the magnitude of control:

$$\begin{aligned} f^{(i+1)} &= f^{(i)} - \mu g && (41) \\ g &= \hat{M}'Q\mathcal{E}^{(i)} + Rf^{(i)} && (42) \\ \mu &= (g'g)/(g'\hat{M}'\hat{M}g + R) && (43) \end{aligned}$$

With constraints on both the magnitude and the rate of control:

$$\begin{aligned} \Delta[f]^{(i+1)} &= \Delta[f]^{(i)} - \mu\bar{g} && (44) \\ u^{(i+1)} &= T\Delta[f]^{(i+1)} && (45) \\ \bar{g} &= \hat{\hat{M}}'Q\mathcal{E}^{(i)} + \bar{R}\Delta[f]^{(i)} && (46) \\ \mu &= (\bar{g}'\bar{g})/(\bar{g}'\hat{\hat{M}}'\hat{\hat{M}}\bar{g} + \bar{R}) && (47) \end{aligned}$$

## 6 The Kalman Filtering Approach

In this section, we briefly show that the LC problem can also be formulated as a Kalman filtering problem. Consider the "perfect" feedforward signal, $f^*$ as the state vector, then the following state-space description holds:

$$\begin{aligned} f^{*(i+1)} &= f^{*(i)} + w^{(i)} && (48) \\ e^{*(i)} &= M_r f^{*(i)} - \Sigma v^{(i)} && (49) \end{aligned}$$

where $e^{*(i)}$ is the (post $i$-th repeat) optimal tracking error, $v^{(i)}$ is the additive noise of the process and $w^{(i)}$ is a random term to model any uncertainty or variations associated with $f^*$. $\Sigma = (I + G_pG_c)^{-1}$ and $M_r$ is a composite transfer function that relates $f$ and r to e:

$$\begin{aligned} \bar{e} &= M_r f && (50) \\ &= Mf + b_r && (51) \end{aligned}$$

where $\bar{e}$ is the "clean" tracking error (i.e., if noise does not exist), $b_r$ is the portion of $e$ that is due to $r$. With this decomposition, it follows that:

$$\begin{aligned} e^* - e &= M_r(f^* - f) && (52) \\ &= M(f^* - f) && (53) \\ e - Mf &= -Mf^* - \Sigma v && (54) \end{aligned}$$

where we have used the fact that $0 = M_r f^*$ and $e^* = -\Sigma v$. This leads to a *new* state-space description which facilitates a Kalman filter solution:

$$\begin{aligned} f^{*(i+1)} &= f^{*(i)} + w^{(i)} && (55) \\ z^{(i)} &= e^{(i)} - Mf^{(i)} && (56) \\ &= -Mf^{*(i)} - \Sigma v^{(i)} && (57) \end{aligned}$$

Then, the Kalman filter takes on the following form:

$$\begin{aligned} \hat{f}^{*(i+1)} &= \hat{f}^{*(i)} + K^{(i)}(z^{(i)} - \hat{z}^{(i)}) && (58) \\ \hat{z}^{(i)} &= -M\hat{f}^{*(i)} && (59) \end{aligned}$$

where $K^{(i)}$ is the Kalman gain. Since

$$\begin{aligned} z^{(i)} &= e^{(i)} - Mf^{(i)} && (60) \\ f^{(i)} &= \hat{f}^{*(i)} && (61) \end{aligned}$$

the Kalman filter equation is simply:

$$\hat{f}^{*(i+1)} = \hat{f}^{*(i)} + K^{(i)}e^{(i)} \qquad (62)$$

Note that this is *not* the standard Kalman filter as the evolution is not a function of time but rather a function of the learning cycle. Noting that the Kalman filter in essence implements a form of "optimal" system inversion, this Kalman filter connection does provide a common ground for unifying the various procedures.

## 7 Summary

With all the required process knowledge readily measurable, the proposed procedures are relatively simple with the ease of on-site *tuning*, and are applicable to SISO/MIMO, with or without FB. Not only are encouraging simulation results obtained, but we have also obtained very encouraging preliminary *experimental* results conducted on a semiconductor wafer Rapid Thermal Processing (RTP) reactor [12]. Moreover, the basic scheme has also been extended to *nonlinear* processes with good results [13].

## References

[1] J.E. Kurek and M.B. Zaremba, "Iterative Learning Control Synthesis Based on 2-D System Theory", *IEEE Trans. Automat. Control*, Vol 38, No. 1., January 1993, pp. 121 - 125.

[2] M. Togai and O. Yamano, "Learning Control of Robot Manipulators," *SIAM Conf. Geometric Modeling and Robotics*, Albany, NY, July 1985.

[3] T. Ishihara, K. Abe and H. Takeda, "A Discrete-Time Design of Robust Iterative Learning Controllers," *IEEE Trans. Syst., Man, Cybern.*, Vol. 22, No. 1, Jaunary-February 1992, pp. 74 - 84.

[4] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Oxford: Pergamon, 1982.

[5] J.L. Massey and M.K. Sain, "Inverses of Sequential Circuits," *IEEE Trans. Computers*, Vol. C-17, No. 4, April 1968, pp. 330 - 337.

[6] P. Diaconis and B. Efron, "Computer-Intensive Methods in Statistics," *Scientific American*, May 1983, pp. 116 - 130.

[7] J. Biemond et al., "Iterative Image Deblurring," *Proc. IEEE*, Vol. 78, No. 5, May 1990, pp. 856 - 883.

[8] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley, 1973.

[9] K.M. Tao, "Statistical Averaging and PARTAN – Some Alternatives to LMS and RLS," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, March 1992, pp. IV-25 - IV-28.

[10] R.K. Ward and B.E.A. Saleh, "Deblurring Random Blurr," *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 35, No. 10, 1987, pp. 1494 - 1498.

[11] S.A. Kassam and H.V. Poor, "Robust Signal Processing for Communication Systems," *IEEE Communications Magazine*, January 1983, pp. 20 - 28.

[12] K.M. Tao, R.L. Kosut, M. Ekblad and G. Aral, "Feedforward Learning Applied to RTP of Semiconductor Wafers," (submitted for publication), February 1994.

[13] K.M. Tao, R.L. Kosut and M. Ekblad, "Feedforward Learning – Nonlinear Processes and Adaptation," (submitted for publication), February, 1994.

## A.8 Feedforward Learning Methods in RTP Temperature Control

- K.M. Tao, G. Aral, R.L. Kosut and M. Ekblad, "Feedforward Learning Methods in RTP Temperature Control," *Proc. 2nd. Int. Rapid Thermal Processing Conf.*, Monterey, CA, Sept. 1994, pp. 278 - 282.

# Feedforward Learning Methods in RTP Temperature Control[*]

K. Mike Tao, Gurcan Aral, Robert L. Kosut, and Mark Ekblad
Integrated Systems, Inc.
3260 Jay Street, Santa Clara, CA 95054
e-mail:mtao@isi.com

## ABSTRACT

Some recent results of using (repetitive) Learning Control (LC) to determine the appropriate feedforward (FF) control action are reviewed in this paper. This LC approach only requires information that is readily measureable on-site, and is designed for the ease of tuning, as it can *learn* from the past experiences. It is applicable to a wide range of single or multi-variable, linear or smoothly nonlinear *repetitive* processes. Many manufacturing tasks are repetitive task-oriented and are potential applications of this LC procedure. Applications of these FF LC methods to RTP temperature control are the focus of this paper. Both simulation and experimental results are included.

Although the experiment is exploratory in nature, the results are very encouraging and deserve serious considerations. Applying this LC approach has resulted in a speed-up of a well-tuned FB loop by a factor of 8 which amounts to more than 20 seconds saving in one processing step - quite significant for RTP. Additionally, the experiment has demonstrated the applicability of the LC theory in a real-world manufacturing setting.

## 1 Introduction

For Rapid Thermal Processing (RTP) of semiconductor wafers, the ability to *quickly* manipulate wafer temperature according to the commanded temperature profile is crucial. Sensor-based feedback (FB) control can certainly improve the RTP reactor's temperature following capability, maintain tight temperature control at steady state, and reduce the effects due to equipment variations. However, the speed of FB control must be balanced with stability considerations which are often limited by the process characteristics such as time delay.

Feedforward (FF) control, on the other hand, can complement the FB control performance by promoting non-delay and anticipatory actions which, when properly designed, can lead to superior tracking or disturbance rejection. Combining FB and FF, one could have a robust, stable and yet agile temperature control system.

Traditional FF design is usually based on analytical methods that require fairly accurate modeling of the process and the FB control loop. Such knowledge is often not available or is subject to change overtime. Furthermore, the so designed FF control is often difficult to *tune*.

The LC approach described in this paper and in [1, 2, 3], on the other hand, only requires the FB loop characteristics that are readily measureable on-site, and is designed for the ease of tuning, as it can *learn* from the past experiences. Many manufacturing tasks are repetitive task-oriented and are potential applications of LC. This LC approach is applicable to a wide range of processes, single or multi-variable, linear or smoothly nonlinear.

In this paper, we review these very recent results including experimental results of applying the basic LC scheme to an RTP reactor designed for semiconductor wafer manufacturing. A factor of 8 speed-up, or more than 20 seconds saving in one single processing step, is realized by using FF LC.

The rest of the paper is organized as follows. First, the basic LC scheme and its variants are reviewed and summarized in Section 2. In Section 3, the RTP experiment is described and shown with its 100° C temperature step setpoint following capability. With a *tuned* FB only control, it takes about 25 seconds or so to reach the new tempearature target at the high-end of the temperature range. We show that, with the application of FF LC, this to target time is reduced to less than 3 seconds - a factor of 8 speed-up or a saving of more than 20 seconds for this one processing step which is quite significant for RTP!

Section 4 further explains how this basic LC scheme can be extended to a class of smoothly *nonlinear* processes, and demonstrates rapid convergence for a simulated nonliner RTP process. Methods for *implementing* FF LC in a *practical system* are discussed in Section 5.

## 2 The Basic Learning Algorithm and Variants

Consider a combined FB and FF control configuration as shown in Figure 1. The basic LC scheme and its variants can be described by:

$$f^{(i+1)} = f^{(i)} - K e^{(i)} \tag{1}$$

where $e^{(i)}$ is the tracking error vector [1] and $f^{(i)}$ the FF control signal in the i-th iteration and $K$ is a *general* (linear) operator which facilitates LC. This LC operator, $K$, may include *time-advance* operations, since the entire $e^{(i)}$ is known after the i-th iteration.

Realizing that both $f$ and the command $r$ contribute to $e$, a straightforward analysis yields the following relationships in noiseless conditions:

$$\begin{aligned} f^{(i+1)} &= (I - K\mathcal{M})f^{(i)} - K\Gamma r & (2)\\ e^{(i+1)} &= (I - \mathcal{M}K)e^{(i)} & (3) \end{aligned}$$

where $\mathcal{M}$ is the transfer operator from $f$ to $e$, and $\Gamma$ is the transfer operator from the command $r$ to $e$.

[*] Work supported in part by ARPA under AFOSR Contract No. F49620-94-C-0003.

Notice that if $K$ is chosen as the inverse of $\mathcal{M}$, i.e., the transfer operator from $f$ to $e$, then exact convergence with one task repeat is possible, and

$$K = \mathcal{M}^{-1} \qquad (4)$$

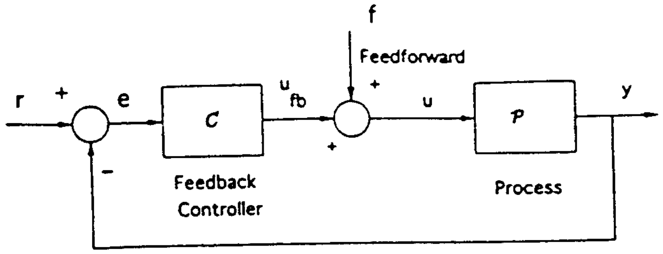


Figure 1: Combined Feedforward and Feedback Configuration

$$f^{(i+1)} = (I - \mathcal{M}^{-1}\mathcal{M})f^{(i)} - \mathcal{M}^{-1}\Gamma r \qquad (5)$$
$$= \mathcal{P}^{-1}r \qquad (6)$$
$$e^{(i+1)} = 0 \qquad (7)$$

where $f = \mathcal{P}^{-1}r$ is the "perfect" FF control signal for reproducing the command trajectory $r$. ($\mathcal{P}^{-1}$ assumes the existence of process "right inverse" [4]. A *necessary* condition is that the number of process inputs must be no less than the number of outputs. With the existence of $\mathcal{P}^{-1}$, the existence of $\mathcal{M}^{-1}$ for the closed-loop is assured. When the right inverse does not exist, least-squares solutions can be sought according to a computational formulation described later.)

A more relaxed condition for convergence is that

$$|\lambda_i(I - \mathcal{M}K)| < 1 \qquad (8)$$

or

$$0 < \lambda_i(\mathcal{M}K) < 2 \qquad (9)$$

where $\lambda_i$ denotes the eigenvalues. This relaxed condition allows the use of a broader scope of algorithms [1] ranging from cases where $\mathcal{M}^{-1}$ is not known exactly (e.g., due to measurement imprecision or process nonlinearity) to gradient descent type of optimization algorithms. Now, instead of converging in one repeat, it will take multiple repeats to achieve satisfactory results.

Computationally, this basic algorithm involves the following (assuming the closed-loop FB system is *linear time-invariant*).

$$-e = -\begin{bmatrix} e(2) \\ e(3) \\ e(4) \\ \cdot \\ \cdot \\ \cdot \\ e(N+1) \end{bmatrix} \qquad (10)$$

$$= \begin{bmatrix} h_1 & 0 & 0 & \dots & 0 \\ h_2 & h_1 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ h_N & h_{N-1} & h_{N-2} & \dots & h_1 \end{bmatrix} \begin{bmatrix} df(1) \\ df(2) \\ df(3) \\ \cdot \\ \cdot \\ \cdot \\ df(N) \end{bmatrix} \qquad (11)$$

$$= Mdf \qquad (12)$$

$$(13)$$

$$f^{(i+1)} = f^{(i)} + df^{(i)} \qquad (14)$$
$$= f^{(i)} - M^{-1}e^{(i)} \qquad (15)$$

where $h_1, h_2, h_3, \dots h_N$ is the closed-loop impulse response sequence from $f$ to $e$. Its state-space equivalent would be $CB, CAB, CA^2B, \dots, CA^{N-1}B$ where $[A, B, C]$ stands for a state-space representation of the $f$ to $e$ process. [2] Matrix $M$ is a time-domain realization of the transfer operator, $\mathcal{M}$. [3]

Using the impulse response representation directly facilitates on-site measurement and simpler computation. It also allows greater flexibility in representing the process dynamics than a parametric model.

When the process is not right invertible, the $M$ matrix would not be invertible. In that case, a least-squares (LS) solution can still be sought. Constraints can be further used in the LS formulation to help shape the FF solution. For instance, by adding penalty on the magnitude of FF control, one can discourage excessively large control which may result from a non-minimum phase process or an excessively demanding transient command. Likewise, the rate of the learned FF control can also be constrained to avoid actuator rate saturation and minimize the potential effect of noise [1]. Extension to include hard constraints can be made using linear programming (LP) or quadratic programming (QP) type of formulations.

This procedure is validated using a simulated 3 x 3 multi-input multi-output (MIMO) process stabilized with decentralized PI controllers. Figure 2 illustrates the system performance in tracking a ramp command. The top plots show the tracking performance with feedback control only; the bottom plots show the performance after one repeat of the task. The impulse response matrix is estimated by differencing the "measured" step responses. The simulation is done using the graphics oriented control design software, SystemBuild and MATRIXx. The solid lines indicate the command trajectories and the dashed lines the actual process responses. It is apparent that in this idealized *linear time-invariant* situation, the basic LC approach can achieve perfect tracking in one learning repeat. Good performance is also obtainable in a noisy environment by taking proper noise reduction measures discussed in [1].

A question in order is: would this LC approach work well in a rugged industrial manufacturing environment? A major portion of this paper is devoted to reporting some preliminary experimental results of applying LC to wafer temperature tracking in an RTP reactor.

## 3 The RTP Reactor and Wafer Temperature Control

The reactor, on which the experiments were conducted, uses rapid thermal processing (RTP) technology to process one wafer at a time. According to a pre-designed temperature command profile, the wafer is raised to high temperatures in a few stages stages during the course of the process and then cooled down. It is crucial to maintain tight temperature control at the set temperature, but it is also very important to quickly respond to the temperature command *without overshoot*. The FB controller is designed with an anti-windup PI and a lead/lag compensator in the forward path. The lead/lag is introduced to offset some of

[1] Under certain conditions, the FB controller output $u_{fb}$ may be used instead of the tracking error $e$. Actually, it is the *deviation* of $u_{fb}$ from its initial steady state that should be used for iterative learning. This is because if the FF is doing a perfect job for the transient, then the FB controller would not have to labor.

[2] If $u_{fb}$ is used for learning, then the impulse response sequence should be from $f$ to $u_{fb}$.

[3] Equations (10) and (11) have assumed that the only time delay is the sample delay, i.e., 1 delay. If additional process delays exist, some of the leading $h$'s would be zero and should be removed from the equations, together with the corresponding leading $e$'s.

the phase lag due to the thermal capacitance induced process delay. (Process delays of about 1.25 seconds to 1.5 seconds are observed.) The FB controller is also gain-scheduled throughout the whole operational range to maximize the performance uniformity. However, at the high end of the temperature range, it is more difficult to achieve high speed and stability simultaneously, and stability is given higher priority.

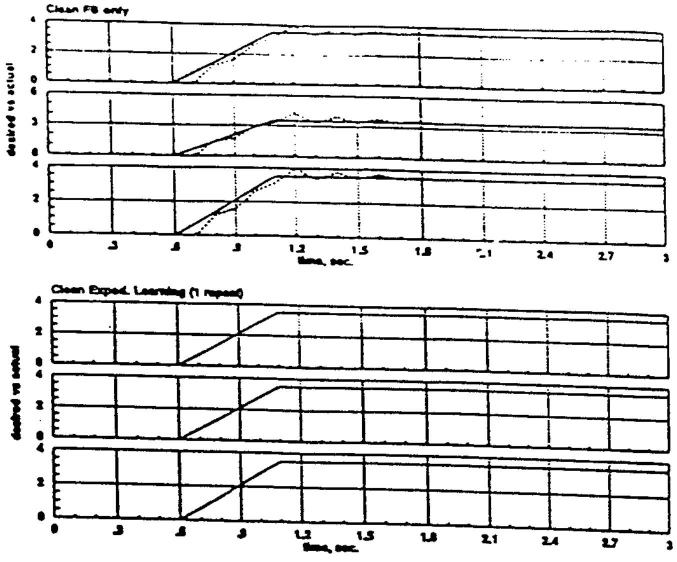


Figure 2: Simulated Feedforward LC (MIMO case)

Figure 3 shows a "high-end" wafer temperature response to a step command from 800° C to 900° C. The settling time is somewhere around 25 seconds – sluggish compared to the "low-end" performances. The following discussions show how we use this information to design a FF control signal that drastically speeds up the temperature response.

To carry out the FF learning procedure, one needs to measure the impulse response characterizing the transfer function from the FF signal to the temperature tracking error, $e$, or to the FB controller output, $u_{fb}$. One way to take such measurements is to inject FF steps and measure the associated responses. That is, special system identification experiments would have to be conducted.

For the single-input single-output linear case, however, the transfer function from the command to wafer temperature is equivalent to the transfer from the FF signal to the FB controller output except for a difference in sign. Therefore, we use the temperature step response obtained above to derive the FF to FB impulse response. The step response, however, is contaminated by the square-wave type measurement disturbances. These square-wave disturbances are inevitably introduced by the existing sensor.

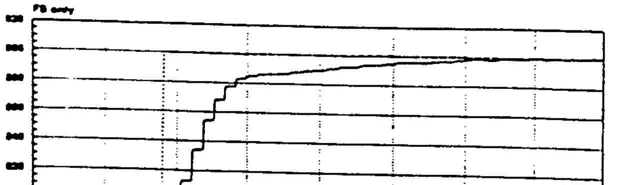


Figure 3: Wafer Temperature Response to Temperature Step Command (FB-Only Control)

These disturbances must be removed in order to obtain a meaningful impulse response reading. This noise removal can be carried out by using (*non-causal*) low-pass filtering or by fitting a low-order system to the step response. The particular noncausal filtering we used is a form of jack-knife *sub*sampling [6] and (cubic) spline interpolation scheme. Jack-knife sampling is effective for estimating ensemble average without actually conducting multiple experiments.

Next, the filtered step response is time differenced and sign reversed to produce an estimate of the FF to FB impulse response. Due to time differencing, the "tail" of the impulse response still exhibits some oscillations from the residual noise. To improve the steady-state performance of the FF solution, the tail is further smoothed out. [4] The result is shown in Figure 4.

Another needed ingredient is the "error" signal which is the difference between the *desired* and the *actual*. What is the *desired*? Since the step command is still driving the FB loop, and for practical reasons we certainly do not want the temperature to respond like a step, a *reference model* is introduced to generate the desired response. A first-order *reference model* with rise time of about 2 seconds is used to provide the *desired temperature response*. However, since we use the FB output as an indication of how well the system performs, a *desired FB* signal must be computed. This desired FB signal is generated by feeding the difference between the step command and the desired temperature response to a *copy* of the FB controller. This FB controller's output is then the *desired FB*.

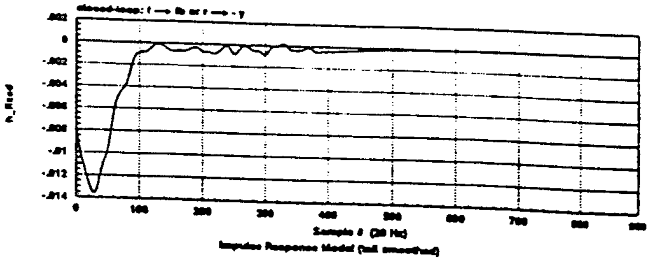


Figure 4: The Estimated Impulse Response Model

With the desired and the actual FB, a FB output based error signal is obtained. (This error signal is time-advanced according to Eq. (11) and the comment thereafter to account for the process and sample delay.) This time-advanced error signal and the impulse response vector were then used by the LC algorithm to compute a FF signal. The algorithm computes an appropriate FF signal that would minimize this error according to the impulse response model. For unconstrained situation, this minimization can be accomplished by exact inversion. However, it is known that exact inversion is often not a good idea in the face of uncertainties, a principle also found in iterative image deblurring [7] and adaptive signal processing [8]. We instead use soft inversion by numerically iterating with Optimal Gradient Descent [5, 1].

Thirty (30) *numerical* iterations are performed between process repeats. The resultant FF signal is then injected with proper time synchronization. The result is a significantly faster response. The (to 90%) rise time is about three (3) times shorter than that of FB only control and yet *without* overshoot. However, it is noticed that the rise time is still longer than the *desired*, and the wafer temperature starts the transition a bit too early (about 0.8 second or so). A few factors could have contributed to this, including measurement and numerical errors and perhaps process nonlinearity that would affect the fidelity of impulse re-

[4] It is noted that this "tail" problem could be avoided, if the impulse response is generated by a parametric model fitted to the data.

sponse (linear) modeling

Using the same FF signal but delaying it by 0.5 second, a very good result is obtained and is displayed in Figure 5. Notice the calculated anticipatory action of the FF signal that precedes the step command. The wafer temperature starts the upward transition just on time and hits the target in less than 3 seconds with no overshoot. Comparing to the 25 seconds or so settling time with FB only control, this means a saving of more than 20 seconds for just one processing step which is quite significant for RTP.

Finally, "robustness" of the combined FF and FB scheme is revealed in the following two cases. If the (first) FF is delayed by 1 second (i.e., *twice as much* as the 0.5 second delay), the wafer temperature is still well behaved and responds fast, but with about 7° C overshoot (for this 100° step). This provides a "feel" for the robustness of FF control. Another robustness indicator is the *repeatability* run which was performed three days after the first run was made and showed remarkable consistency. In between, the reaction chamber was taken apart once for examination and that apparently did not impact the performance. The relatively tight closed-loop FB control should be (at least partially) credited for the apparent robustness. This is one of the reasons for using the combined FF and FB configuration.

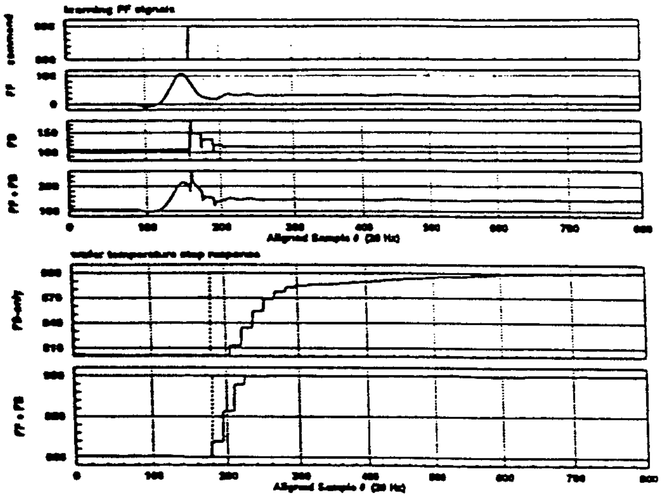


Figure 5: Wafer Temperature with FF Learning

## 4 LC for Nonlinear Processes

In this section, we briefly describe an extension of the basic LC scheme to a class of smoothly *nonlinear* processes. For nonlinear processes, the notion of "impulse response" is not exact, and varies with the process state and input. Using the "impulse response" measured at a certain process condition would not, in general, adequately describe the process behavior throughout a (temperature) control task. The model used for this study is a nonlinear RTP model which is fitted to some measured data to reflect nonlinear process gain and time constant. Direct application of the basic LC scheme (with full strength) has resulted in learning oscillations and apparent divergence. Although with *reduced strength*, the basic learning scheme would still converge, but only after *many iterations*.

Instead, using a *prediction error* based scheme described in [3], the "impulse response" *model* can be *adapted* during the iterative learning process, resulting in rapid convergence. Figure 6 shows such an adaptive learning sequence that converges to the desired reference response in 2 or 3 adaptations. (The feedback only response which is not shown is an order of magnitude slower.)

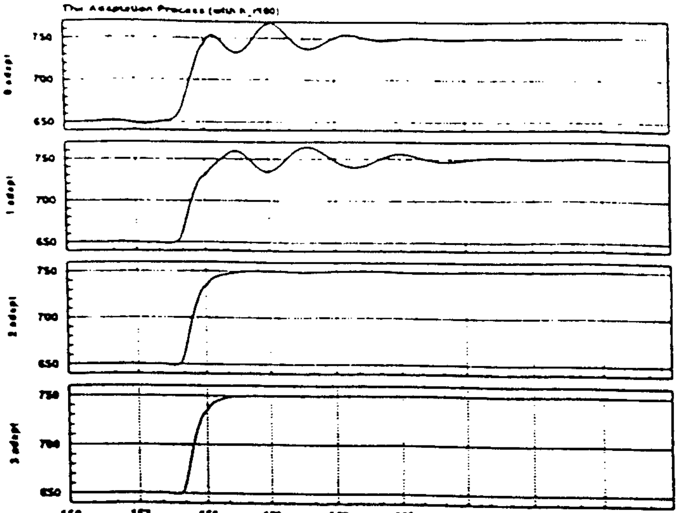


Figure 6: An Adaptive LC Sequence – for a nonlinear RTP process

## 5 Implementing LC

The LC scheme discussed so far is *signal synthesis* based, i.e., an appropriate FF signal, $f$, is determined for a specific task and recipe. A FF data base can be constructed for a set of recipes. When faced with a *new* recipe, however, one would have to either *interpolate* from the FF data base, or likely relearn the FF signal. Then, the FF data base needs to be updated accordingly.

A more effective method for *implementing* LC in a practical system is to use a dynamic FF filter which is driven by the recipe command to generate the appropriate FF signal. In this way, the need for a (growing) FF data base and for relearning is eliminated. Based on the signal synthesis based LC, methods for realizing a *dynamic filter* based implementation have been developed and are described in [9].

## 6 Summary

The feedforward Learning Control (LC) methodology is shown to be applicable to the control of *repetitive* industrial and maufacturing processes in general and of the wafer temperature in a typical RTP reactor, in particular. The preliminary experiments have demonstrated a speed-up by a factor of 8, or 20 seconds saving in a single processing step, quite significant for RTP.

With all the required ingredients readily measurable, the learning scheme, as decribed here and in [1], is suitable for on-site tuning and learning. The very encouraging experimental results seem to strongly support the applicability of the theory and deserve serious considerations for practical deployment.

## References

[1] K.M. Tao, R.L. Kosut and G. Aral, "Learning Feedforward Control," *Proc. American Control Conference*, Baltimore, MD, June - July 1994, pp. 2575 - 2579.

[2] K.M. Tao, R.L. Kosut, M. Ekblad and G. Aral, "Feedforward Learning Applied to RTP of Semiconductor Wafers," *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, December 1994 (to appear).

[3] K.M. Tao, R.L. Kosut and M Ekblad, "Feedforward Learning - Nonlinear Processes and Adaptation," *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, December 1994 (to appear).

[4] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Oxford: Pergamon Press, 1982.

[5] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley, 1973.

[6] P. Diaconis and B. Efron, "Computer-Intensive Methods in Statistics," *Scientific American*, May 1983, pp. 116 - 130.

[7] J. Biemond et al., "Iterative Image Deblurring," *Proc. IEEE*, Vol. 78, No. 5, May 1990, pp. 856 - 883.

[8] K.M. Tao, "Statistical Averaging and PARTAN - Some Alternatives to LMS and RLS," *Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, March 1992, pp. IV-25 - IV-28.

[9] K.M. Tao and R.L. Kosut, *Implementaing Feedforward Learning Using Dynamic Filters*, Research Memo., Integrated Systems, Inc., Santa Clara, CA, July 1994.

## A.9 Feedforward Learning Applied to RTP of Semiconductor Wafers

- K.M. Tao, R.L. Kosut, M. Ekblad and G. Aral, "Feedforward Learning Applied to RTP of Semiconductor Wafers," *Proc. 33rd IEEE Conf. Decision and Control*, pp. 67–72, Lake Buena Vista, FL, December 1994.

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

# Feedforward Learning Applied to RTP of Semiconductor Wafers *

K. Mike Tao, Robert L. Kosut, Mark Ekblad and Gurcan Aral
Integrated Systems, Inc.
3260 Jay Street, Santa Clara, CA 95054
e-mail:mtao@isi.com

## ABSTRACT

Recent algorithmic results in (feedforward) Learning Control (LC) [1] are applied to the control of semiconductor wafer temperature in a Rapid Thermal Processing (RTP) reactor. Although the first attempt is experimental in nature, the results are very encouraging and desçrve serious consideration.

Applying this LC approach has resulted in a speed-up of a well-tuned FB loop by a factor of 8 which amounts to more than 20 seconds saving in one processing step -- quite significant for RTP. Additionally, the experiment has demonstrated the applicability of the LC theory in a real-world manufacturing setting. Many repetitive manufacturing tasks are potential applications of this LC procedure.

## 1 Introduction

For Rapid Thermal Processing (RTP) of semiconductor wafers, the ability to *quickly* manipulate wafer temperature according to the commanded temperature profile is crucial. Sensor-based feedback (FB) control can certainly improve the RTP reactor's temperature following capability, maintain tight temperature control at steady state, and reduce the effects due to equipment variations. However, the speed of FB control must be balanced with stability considerations, and is often limited by the process characteristics such as time delay.

Feedforward (FF) control, on the other hand, can complement the FB control performance by promoting non-delay and anticipatory actions which, when properly designed, can lead to superior tracking or disturbance rejection. Combining FB and FF, one could have a robust, stable and yet agile temperature control system.

Traditional FF design is usually based on analytical methods that require fairly accurate modeling of the process and the FB control loop. Such knowledge is often not available or is subject to change overtime. Furthermore, the so designed FF control is often difficult to *tune*.

The LC approach described in [1], on the other hand. only requires the FB loop characteristics that are readily measureable on-site, and is designed for the ease of tuning, as it can *learn* from the past experiences via task repetition. (Many manufacturing tasks are repetitive task-oriented and are potential applications of LC.) This LC approach is therefore applicable to a wide range of processes, single or multivariable, linear or smoothly nonlinear. [1]

In this paper, we report our experimental results of applying the basic LC scheme to an RTP reactor designed for semiconductor wafer manufacturing. A factor of 8 speed-up, or more than 20 seconds saving in one processing step, is realized by using learning FF control.

The rest of the paper is organized as follows. First. the basic LC scheme and its variants are reviewed and summarized in Section 2. In Section 3, the RTP process is described and shown with its 100° C temperature step following capability. With a *tuned* FB only control, it takes about 25 seconds or so to reach the new tempearature target. We show, in Section 4. that with the application of FF LC, this to target time is reduced to less than 3 seconds - a factor of 8 speedup or a saving of more than 20 seconds for this one processing step which is quite significant for RTP! The details of conducting this FF LC experiment are also included in Section 4. The robustness of the results is confirmed by a repeatibility test.

*Work supported by ARPA under AFOSR Contract No. F49620-94-C-0003.

[1] This simple and effective LC scheme has also been extended to the control of a class of nonlinear processes. Details of that extension are described in a separate paper [2].

0-7803-1968-0/94$4.00©1994 IEEE

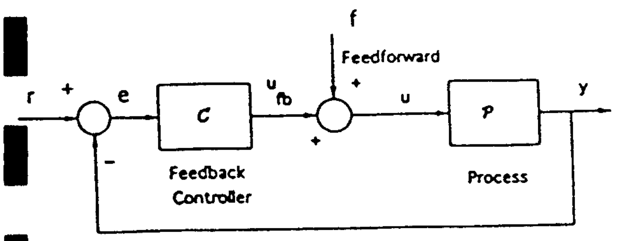


Figure 1: Combined Feedforward and Feedback Configuration

## 2 The Basic Learning Algorithm and Variants

Consider a combined FB and FF control configuration as shown in Figure 1. The basic LC scheme and its variants can be described by:

$$f^{(i+1)} = f^{(i)} - Ke^{(i)} \tag{1}$$

where $e^{(i)}$ is the tracking error vector [2] and $f^{(i)}$ the FF signal in the $i$-th iteration and $K$ is a *general* (linear) operator which facilitates LC. This LC operator, $K$, may include time-advance operations to allow proper ***credit assignment*** for the learning process.

Realizing that both $f$ and the command $r$ contribute to $e$, straightforward analysis yields the following relationships in noiseless conditions:

$$\begin{aligned} f^{(i+1)} &= (I - K\mathcal{M})f^{(i)} - K\Gamma r & (2) \\ e^{(i+1)} &= (I - \mathcal{M}K)e^{(i)} & (3) \end{aligned}$$

where $\mathcal{M}$ is the transfer operator from $f$ to $e$, and $\Gamma$ is the transfer operator from the command $r$ to $e$:

$$\begin{aligned} \mathcal{M} &= -(I + \mathcal{P}\mathcal{C})^{-1}\mathcal{P} & (4) \\ \Gamma &= I - (I + \mathcal{P}\mathcal{C})^{-1}\mathcal{P}\mathcal{C} & (5) \end{aligned}$$

Notice that if $K$ is chosen as the inverse of $\mathcal{M}$, i.e., the transfer operator from $f$ to $e$, then exact convergence with one repeat is possible. This is because if $K = \mathcal{M}^{-1}$,

$$\begin{aligned} f^{(i+1)} &= (I - \mathcal{M}^{-1}\mathcal{M})f^{(i)} - \mathcal{M}^{-1}\Gamma r & (6) \\ &= \mathcal{P}^{-1}r & (7) \\ e^{(i+1)} &= 0 & (8) \end{aligned}$$

where $f = \mathcal{P}^{-1}r$ is the "perfect" FF control signal for reproducing the command trajectory $r$. ($\mathcal{P}^{-1}$ assumes the existence of process "right inverse" [3, 1]. A *necessary* condition is that the number of process inputs must be no less than the number of outputs. With the existence of $\mathcal{P}^{-1}$, the existence of $\mathcal{M}^{-1}$ for the closed-loop is assured. When the right inverse does not exist, least-squares solutions can be sought according to the following computational formulation.)

A more relaxed condition for convergence is that

$$|\lambda_i(I - MK)| < 1 \tag{9}$$

or

$$0 < \lambda_i(MK) < 2 \tag{10}$$

where $\lambda_i$ denotes the eigenvalues, and matrix $M$ is a time-domain (Toeplitz) realization of the transfer operator, $\mathcal{M}$, as shown below in Eq. (12). This relaxed condition allows the use of a broader scope of algorithms ranging from cases where $\mathcal{M}^{-1}$ is not known exactly (e.g., due to measurement imprecision or process nonlinearity) to gradient descent type of optimization algorithms. Now, instead of converging in one repeat, it will take multiple repeats to achieve satisfactory results.

Computationally, this basic algorithm involves the following (assuming the closed-loop FB system is ***linear time-invariant***):

$$-e = -\begin{bmatrix} e(2) \\ e(3) \\ e(4) \\ . \\ . \\ . \\ e(N+1) \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} h_1 & 0 & 0 & \dots & 0 \\ h_2 & h_1 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & \dots & 0 \\ . & . & . & \dots & . \\ h_N & h_{N-1} & h_{N-2} & \dots & h_1 \end{bmatrix} \begin{bmatrix} df(1) \\ df(2) \\ df(3) \\ . \\ . \\ . \\ df(N) \end{bmatrix}$$

$$= Mdf \tag{12}$$

$$\begin{aligned} f^{(i+1)} &= f^{(i)} + df^{(i)} & (13) \\ &= f^{(i)} - M^{-1}e^{(i)} & (14) \end{aligned}$$

where $h_1, h_2, h_3, \dots h_N$ is the *measureable* closed-loop impulse response sequence from $f$ to $e$. Its state-space equivalent would be $CB, CAB, CA^2B, \dots, CA^{N-1}B$

[2] Under certain conditions, the signal, $\tilde{u}_{fb}$, instead of the tracking error, $e$, may be used for iterative learning, where $\tilde{u}_{fb}$ is the ***deviation*** of the FB controller output from a reference $\bar{u}_{fb}$.

where [A,B,C] stands for a state-space representation of the $f$ to $e$ process. [3] [4]

When the process is not right invertible, the $M$ matrix would not be invertible. In that case, a least-squares (LS) solution can still be sought. Constraints can be further used in the LS formulation to help shape the FF solution. For instance, by adding penalty on the magnitude of FF control, one can discourage successively large control which may result from a non-minimum phase process or an overly demanding transient command. Likewise, the rate of the learned FF control can also be constrained to avoid actuator rate saturation and minimize the potential effect of noise. Extension to include hard constraints can be made using linear programming (LP) or quadratic programming (QP) type of formulations.

This procedure is validated using a simulated 3 x 3 multi-input multi-output (MIMO) process stabilized with decentralized PI controllers. Figure 2 illustrates the system performance in tracking a ramp command. The top plots show the tracking performance with feedback control only; the bottom plots show the performance after one repeat of the task. The impulse response matrix is estimated by differencing the "measured" step responses. The simulation is done using the graphics oriented control design software, SystemBuild and MATRIXx. The solid lines indicate the command trajectories and the dashed lines the actual process responses. It is apparent that in this idealized *linear time-invariant* situation, the basic LC approach can achieve perfect tracking in one learning repeat. Good performance is also obtainable in a noisy environment by taking proper noise reduction measures discussed in [1].

A question in order is: would this LC approach work well in a rugged industrial manufacturing environment? The rest of the paper is devoted to reporting some preliminary experimental results of applying LC to wafer temperature tracking in an RTP reactor.

## 3 The RTP Reactor and Wafer Temperature Control

The reactor, on which the experiments were conducted, uses the rapid thermal processing (RTP) technology to process one wafer at a time. According to a pre-designed temperature command profile, the wafer is raised to high temperatures in a few stages during the course of the process and then cooled down. It is crucial to maintain tight temperature control at the set temperature, but it is also very important to quickly respond to the temperature command *without overshoot*. The FB controller is designed with an anti-windup PI and a lead/lag compensator in the forward path. The lead/lag is introduced to offset some of the phase lag due to the thermal capacitance induced process delay. (Process delays of about 1.25 seconds to 1.5 seconds are observed.) The FB controller is also gain-scheduled throughout the whole operational range to maximize the performance uniformity. However, at the high end of the temperature range, it is more difficult to achieve high speed and stability simultaneously, and stability is given higher priority.

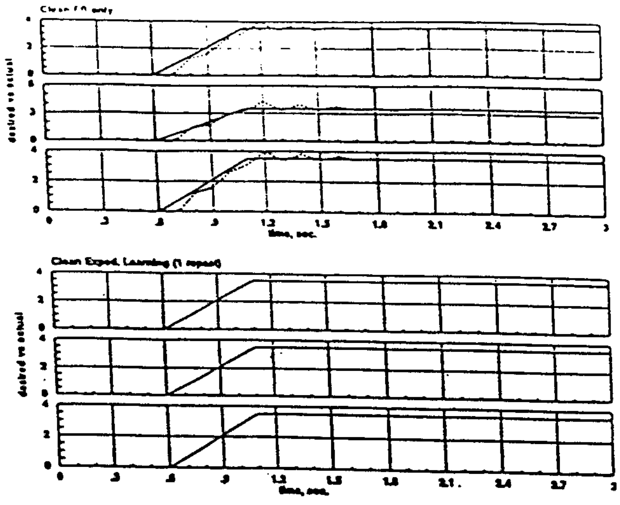


Figure 2: Simulated Feedforward LC (MIMO case)

Figure 3 shows a "high-end" wafer temperature response to a step command from 800° C to 900° C. The settling time is somewhere around 25 seconds - sluggish compared to the "low-end" performances. The next Section discusses how we use this information to design a FF control signal that drastically speeds up the temperature response.

## 4 Feedforward LC Applied to the RTP Reactor

To carry out the FF learning procedure, one needs to measure the impulse response characterizing the transfer function from the FF signal to the temperature tracking error, $e$, or to the FB controller output, $u_{fb}$. One way to take such measurements is to inject FF

[3] If $u_{fb}$ is used for learning, then the impulse response sequence should be from $f$ to $u_{fb}$.

[4] Equations (11) and (12) have assumed that the only time delay is the sample delay, i.e., 1 delay. If additional process delays exist, some of the leading $h$'s would be zero and should be removed from the equations, together with the corresponding leading $e$'s.

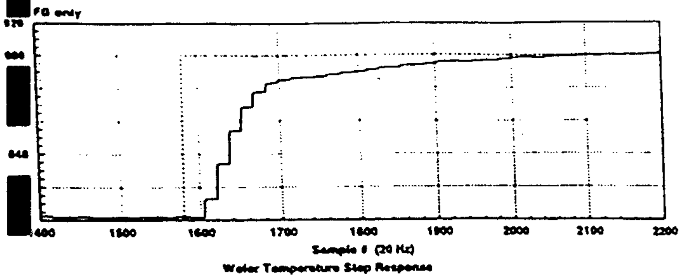


Figure 3: Wafer Temperature Response to Temperature Step Command (FB-Only Control)

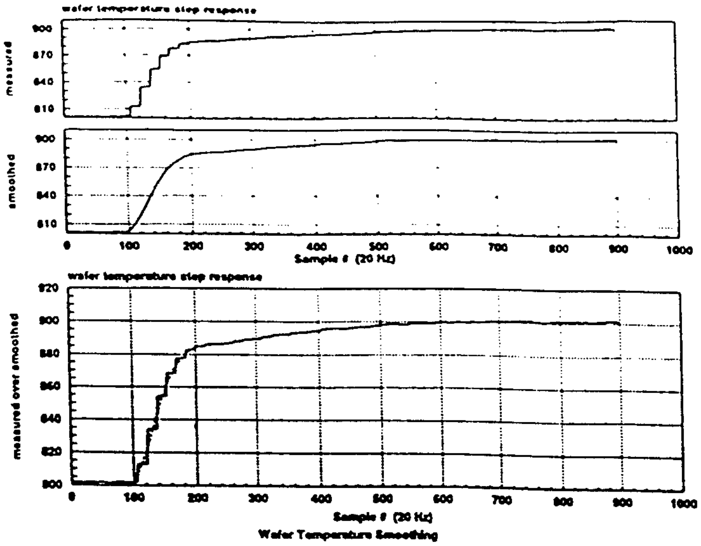


Figure 4: Step Response Smoothing

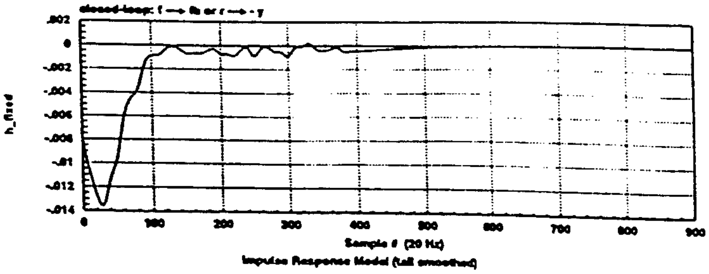


Figure 5: The Estimated Impulse Response Model

steps and measure the associated responses. That is, special system identification experiments would have to be conducted.

For the single-input single-output linear case, however, the transfer function from the command to wafer temperature is also equivalent to the transfer from the FF signal to the FB controller output except for a difference in sign. Therefore, we use the temperature step response obtained above to derive the FF to FB impulse response. The step response, however, is contaminated by the square-wave type measurement disturbances. These square-wave disturbances are inevitably introduced by the existing sensor.

These disturbances must be removed in order to obtain a meaningful impulse response reading. This noise removal can be carried out by using (non-causal) low-pass filtering or by fitting a low-order system to the step response. The particular filtering we used is a jack-knife subsampling and (cubic) spline interpolation scheme. The contaminated step response (sampled at a 20 Hz rate) is first subsampled every 10 samples. Then, at each subsampled point, the temperature reading is replaced by the average reading around that point (i.e., the average of readings at the original 20 Hz rate around that point). Mean averaging with a window of size 15 is used. This is a form of jack-knife sampling [5] which is effective for estimating ensemble average without actually conducting multiple experiments. These subsampled and averaged data points are then interpolated back using the (cubic) spline. This filtering scheme performs very well. Figure 4 compares the step response before and after smoothing. (Other non-causal filtering schemes such as taking the average of forward and backward Butterworth low-pass filtered results may also be adopted.)

Next, the filtered step response is time differenced and sign reversed to produce an estimate of the FF to FB impulse response. Due to time differencing, the "tail" of the impulse response still exhibits some oscillations from the residual noise. To improve the steady-state performance of the FF solution, the tail is further smoothed out. The resultant impulse response model (with 1.25 second process time delay removed) is shown in Figure 5. [5]

Another needed ingredient is the "error" signal which is the difference between the *desired* and the *actual*. What is the *desired*? Since the step command is still driving the FB loop, and for practical reasons we cer-

[5] It is noted that this "tail" problem could be avoided, if the impulse response is generated by a parametric model fitted to the data.

tainly do not want the temperature to respond like a step, a *reference model* is introduced to generate the desired response. A first-order *reference model* with rise time of about 2 seconds is used to provide the *desired temperature response*. However, since we use the FB output as an indication of how well the system performs, a *desired FB* signal must be computed. This desired FB signal is generated by feeding the difference between the step command and the desired temperature response to a *copy* of the FB controller. This FB controller's output is then the *desired FB*. This is because, in our configuration, though the desired temperature is the output of a first-order reference model, the (step) command is still driving the real-time FB controller. When and if the desired temperature trajectory is achieved, the FB controller would see an "ideal" error which is the difference between the command and the desired temperature trajectory.

With the desired and the actual FB, a FB output based error signal is obtained. (This error signal is time-advanced according to Eq. (11) and the footnote comment thereafter to account for the process and sample delay.) This (time advanced) error signal and the impulse response vector were then used by the LC algorithm to compute a FF signal. The algorithm computes an appropriate FF signal that would minimize this error according to the impulse response model. For unconstrained situation, this minimization can be accomplished by exact inversion. However, it is known that exact inversion is often not a good idea in the face of uncertainties, a principle also found in iterative image deblurring [6] and adaptive signal processing [7]. We instead use *soft* inversion by numerically iterating with Optimal Gradient Descent [4, 1].

Thirty (30) *numerical* iterations are performed between process repeats. The resultant FF signal (shown in Figure 6) is then injected with proper time synchronization. Notice the calculated anticipatory action of the FF signal that precedes the step command. The result is a significantly faster response as shown in Figure 6. The (to 90%) rise time is about three (3) times shorter than that of FB only control and yet without overshoot. However, it is noticed that the rise time is still longer than the *desired*, and the wafer temperature starts the transition a bit too early (about 0.8 second or so). A few factors could have contributed to this, including measurement and numerical errors and perhaps process nonlinearity that would affect the fidelity of impulse response modeling.

Using the same FF signal but delaying it by 0.5 second, a very good result is obtained and is displayed in Figure 7. The wafer temperature starts the upward transition just on time and hits the target in less than 3 seconds with no overshoot. Comparing to the 25 seconds or so settling time with FB only control, this means a saving of more than 20 seconds for just one processing step which is quite significant for RTP.

Alternatively, in a more systematic manner as the iterative LC algorithm suggests, the error signal from the the first try can be used by the LC algorithm to come up with a refined FF for the second task repeat. This procedure has been carried out numerically and yielded a refined FF signal. However, due to a real-time implementation error, (which biases any FF signal by its initial non-zero values,) the refined FF is realized with a step bias which, of course, prevents us from getting the correct result. [6] However, based on qualitative observation and quantitative anlaysis (that estimates the bias effect), the refined FF generated by the LC algorithm would have worked if implemented correctly. Future experiments are needed to validate this.

Finally, "robustness" of the combined FF and FB scheme is revealed in the following two cases. If the (first) FF is delayed by 1 second (i.e., *twice as much* as the 0.5 second delay), the wafer temperature is still well behaved and responds fast, but with about 7° C overshoot (for this 100° step). This provides a "feel" for the robustness of FF control. Another robustness indicator is the *repeatability* run which was performed three days after the first run was made and showed remarkable consistency. In between, the reaction chamber was taken apart once for examination and that apparently did not impact the performance. The relatively tight closed-loop FB control should be (at least partially) credited for the apparent robustness. This is one of the reasons for using combined FF and FB configuration.

## 5 Summary

Feedforward learning is applied to control the wafer temperature in a typical RTP reactor in a rugged industrial environment. The preliminary experiments have demonstrated a speed-up by a factor of 8, or 20 seconds saving in one processing step, quite significant for RTP.

With all the required ingredients readily measurable, the learning scheme, as decribed here and in [1], is suitable for on-site tuning and learning. The very encouraging experimental results seem to strongly sup-

[6] This is not a problem for the first try, because the first FF signal starts with values very close to zero.

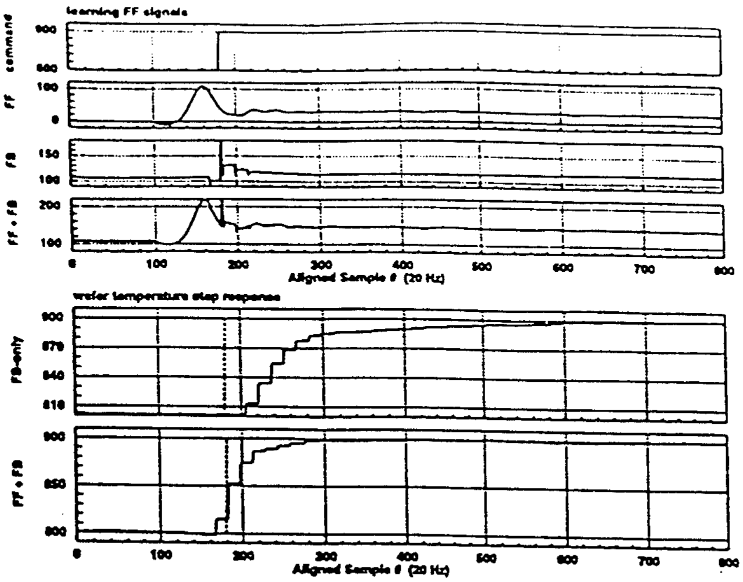


Figure 6: Wafer Temperature with FF Learning - 1

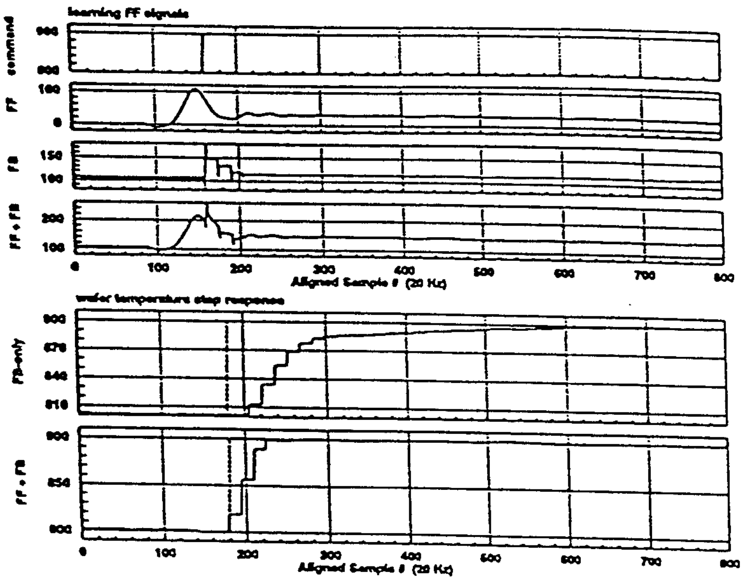


Figure 7: Wafer Temperature with FF Learning - 2

port the applicability of the theory and deserve serious considerations for practical implementations.

The LC scheme discussed so far is *signal synthesis* based, i.e., appropriate FF *signals* are determined and memorized for specific tasks. An *implementation* method for practical systems is to use dynamic FF *filters* driven by the recipe commands to generate the appropriate FF signals, thereby effecting generalization capability and memory efficiency [8, 9]. It is interesting to note that these FF filters can be determined using an extension of the LC scheme described herein.

## References

[1] K.M. Tao, R.L. Kosut and G. Aral, "Learning Feedforward Control," *Proc. American Control Conference*, June - July 1994, pp. 2575 - 2579.

[2] K.M. Tao, R.L. Kosut and M. Ekblad, "Feedforward Learning – Nonlinear Processes and Adaptation," *Proc. 33rd IEEE Conf. Decision and Control*, Lake Buena Vista, FL, December 1994.

[3] R.V. Patel and N. Munro, *Multivariable System Theory and Design*, Oxford: Pergamon Press, 1982.

[4] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley, 1973.

[5] P. Diaconis and B. Efron, "Computer-Intensive Methods in Statistics," *Scientific American*, May 1983, pp. 116 - 130.

[6] J. Biemond et al., "Iterative Image Deblurring," *Proc. IEEE*, Vol. 78, No. 5, May 1990, pp. 856 - 883.

[7] K.M. Tao, "Statistical Averaging and PARTAN – Some Alternatives to LMS and RLS," *Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, March 1992, pp. IV-25 - IV-28.

[8] K.M. Tao and R.L. Kosut, *Implementing Feedforward Learning Using Dynamic Filters*, Research Memo, Integrated Systems, Inc., Santa Clara, CA, July 1994.

[9] K.M. Tao, G. Aral, R.L. Kosut and M. Ekblad, "Feedforward Learning Methods in RTP Temperature Control," *Proc. 2nd. Int. Conf. Rapid Thermal Processing*, Monterey, CA, August - September, 1994.

## A.10 Feedforward Learning – Nonlinear Processes and Adaptation

- K.M. Tao, R.L. Kosut and M. Ekblad, "Feedforward Learning – Nonlinear Processes and Adaptation," *Proc. 33rd IEEE Conf. Decision and Control*, pp. 1060–1065, Lake Buena Vista, FL, December 1994.

WP-3 5:30

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

# Feedforward Learning – Nonlinear Processes and Adaptation *

K. Mike Tao, Robert L. Kosut and Mark Ekblad
Integrated Systems, Inc.
3260 Jay Street, Santa Clara, CA 95054
e-mail:mtao@isi.com

## ABSTRACT

A new feedforward learning control (LC) scheme is reported in this paper. This new scheme is applicable to a class of smoothly *nonlinear* (repetitive) control tasks. *Rapid* reductions in tracking error are demonstrated using a single-input, single-output (SISO), nonlinear model of the Rapid Thermal Processing (RTP) wafer manufacturing process.

This scheme preserves the simplicity of the basic LC scheme reported in [1] by using the (measurable) impulse response model. Since the "impulse response" of a nonlinear process varies as the process trajectory moves along, *adaptation* is used to adjust the impulse response model between task repeats. Analytical justifications for the proposed adaptation are provided using a nonlinear equation solving analogy. Extension to the mult-input and mutioutput (MIMO) case is also included.

## 1 Introduction

It is well established that feedback (FB) control can provide a measure of robustness against process variations and disturbances. The speed of FB control, however, must be balanced with stability considerations which are often limited by the process characteristics such as time delay. A properly designed feedforward (FF) control, on the other hand, can complement FB control by promoting non-delay and anticipatory actions, leading to superior tracking or disturbance rejection. Combining FB and FF, one could have a robust, stable and yet agile control system.

To automatically determine the appropriate FF actions, a basic Learning Control (LC) scheme and its variants have been reported in [1] for *repetitive* control tasks including many manufacturing type processes. Based on very simple and practical ideas, this basic LC scheme can lead to rapid learning of FF signals for (almost) linear time-invariant (LTI) processes, SISO or MIMO. The only required information is the process loop impulse response which can be readily measured on site, making this a practical on-site *tuning* tool.

In this paper, this simple and effective LC scheme is extended to the control of a class of nonlinear processes. Retaining the simplicity of "impulse response" modeling, this scheme uses between-task adaptation to adjust the impulse response model to successively achieve good approximations of the underlying nonlinear process dynamics. An interesting *dual* relationship between the adaptation mechanism and the basic LC allows the use of the same least-squares inversion type algorithm for a dual purpose. Demonstrated on a nonlinear Rapid Thermal Processing (RTP) semiconductor manufacturing process model, this new adaptive LC scheme shows rapid convergence after 3 adaptation task repeats.

The rest of the paper is organized as follows. In Section 2, the basic LC scheme and its variants are reviewed. In Section 3, the nonlinear RTP model is described first, and then the basic LC scheme is applied. It is shown that the measured "impulse response" of this nonlinear process model varies considerably with the size of the impulse input. Consequently, the basic LC scheme which uses a single fixed "impulse response" does not perform well. In fact, one of the impulse responses led to iterative learning divergence. Section 4 first shows that with the new adaptive LC scheme, the temperature trajectory tracking is (nearly) perfected after 3 adaptation task repeats. Then, in Section 5, this adaptive scheme is formulated using a *prediction error* minimization view for the SISO case. Minimizing the prediction error, the adaptation scheme adjusts the impulse response model between task repeats. The computation algorithm is a *dual* to the basic LC, thereby allowing software reuse. Section 6 offers an insightful illustration of this adaptive LC process by using a "1D" nonlinear algebraic equation solving analogy. This illustration links this adaptive process to successive approximation methods used in solving nonlinear equations [2, 6]. Section 7 extends this adaptive LC scheme to the multi-input multi-output (MIMO) case.

## 2 The Basic Learning Algorithm and Variants

Consider a combined FB and FF control configuration as shown in Figure 1. The basic LC scheme and its variants can be described by:

$$f^{(i+1)} = f^{(i)} - Ke^{(i)} \tag{1}$$

where $e^{(i)}$ is the tracking error vector [1] and $f^{(i)}$ the FF signal in the $i$-th iteration, and $K$ is a general (linear) operator which facilitates LC. This LC operator, $K$, may include time-advance operations to allow proper *credit assignment* for the learning process.

Realizing that both $f$ and the command $r$ contribute to $e$, straightforward analysis yields the following relationships in noiseless conditions:

$$\begin{aligned} f^{(i+1)} &= (I - K\mathcal{M})f^{(i)} - K\Gamma r & (2)\\ e^{(i+1)} &= (I - \mathcal{M}K)e^{(i)} & (3) \end{aligned}$$

where $\mathcal{M}$ is the transfer operator from $f$ to $e$, and $\Gamma$ is the

*Work supported by ARPA under AFOSR Contract No. F49620-94-C-0003.

[1] Under certain conditions, the signal $\tilde{u}_{fb}$, instead of the tracking error , $e$, may be used for iterative learning, where $\tilde{u}_{fb}$ is the *deviation* of the FB controller output from a reference $u_{fb}$.

0-7803-1968-0/94$4.00©1994 IEEE

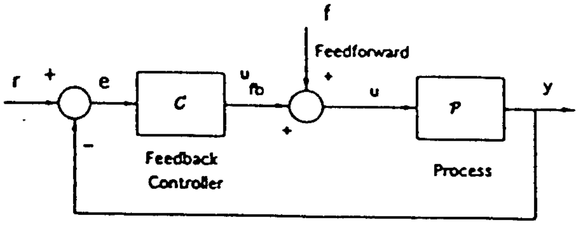


Figure 1: Combined Feedforward and Feedback Configuration

transfer operator from the command $r$ to $e$:

$$\mathcal{M} = -(I + \mathcal{P}\mathcal{C})^{-1}\mathcal{P} \tag{4}$$

$$\Gamma = I - (I + \mathcal{P}\mathcal{C})^{-1}\mathcal{P}\mathcal{C} \tag{5}$$

Notice that if $K$ is chosen as the inverse of $\mathcal{M}$, i.e., the transfer operator from $f$ to $e$, then *exact convergence with one repeat is possible.* This is because if $K = \mathcal{M}^{-1}$,

$$\begin{aligned} f^{(i+1)} &= (I - \mathcal{M}^{-1}\mathcal{M})f^{(i)} - \mathcal{M}^{-1}\Gamma r & (6)\\ &= \mathcal{P}^{-1}r & (7)\\ e^{(i+1)} &= 0 & (8) \end{aligned}$$

where $f = \mathcal{P}^{-1}r$ is the "perfect" FF control signal for reproducing the command trajectory $r$. [2]

A more relaxed condition for tracking convergence is that $|\lambda_i(I - MK| < 1$ or $0 < \lambda_i(MK) < 2$. where $\lambda_i$ denotes the eigenvalues, and matrix $M$ is a time-domain (Toeplitz) realization of the transfer operator, $\mathcal{M}$, as shown in Eq. (10) below. This relaxed condition allows the use of a broader scope of algorithms ranging from cases where $\mathcal{M}^{-1}$ is not known exactly (e.g., due to measurement imprecision or process nonlinearity) to gradient descent type of optimization algorithms. Now, instead of converging in one repeat, it will take multiple repeats to achieve satisfactory results.

Computationally, this basic algorithm involves the following (assuming the closed-loop FB system is LTI):

$$-e = -\begin{bmatrix} e_2 \\ e_3 \\ e_4 \\ \cdot \\ \cdot \\ \cdot \\ e_{N+1} \end{bmatrix} \tag{9}$$

$$\begin{aligned} &= \begin{bmatrix} h_1 & 0 & 0 & \dots & 0 \\ h_2 & h_1 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ h_N & h_{N-1} & h_{N-2} & \dots & h_1 \end{bmatrix} \begin{bmatrix} df_1 \\ df_2 \\ df_3 \\ \cdot \\ \cdot \\ \cdot \\ df_N \end{bmatrix} \\ &= Mdf & (10)\\ f^{(i+1)} &= f^{(i)} + df^{(i)} & (11)\\ &= f^{(i)} - M^{-1}e^{(i)} & (12) \end{aligned}$$

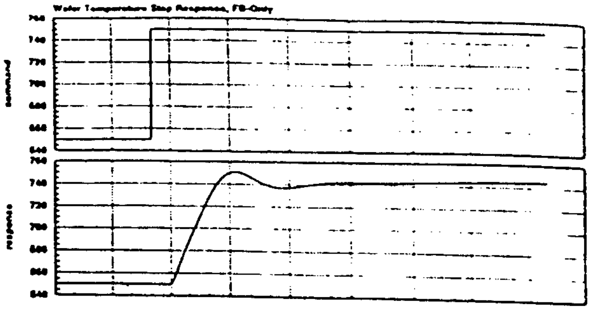


Figure 2: Step Response of the RTP Model under FB-only Control

where $h_1, h_2, h_3, \dots h_N$ is the closed-loop impulse response sequence from $f$ to $e$ (which can be readily measured on-site). [3] [4] Notice that the above formulation is equally applicable to SISO and MIMO LTI problems as the simulation results in [1] demonstrate.

Variants of this basic computational scheme are described in [1], addressing the issues of process invertibility, phase nonminimality, process and measurement uncertainties, actuator constraints, etc. These methods are various constrained or regularized least-squares (LS) type solution procedures that are considered useful and robust for handling these practical issues.

## 3 The Basic LC Applied to A Nonlinear RTP Model

In this section, we apply the basic LC scheme which is designed for LTI processes to a *nonlinear* single-input single-output (SISO) RTP process model which is under stable, *gain scheduled* "linear" FB control. This RTP model exhibits a 1 second time delay, and is commanded to go from 650° C to 750° C "as quickly as possible" and preferrably with *no overshoot.* The step response of the FB loop is shown in Figure 2. Apparently that to avoid overshoot, the FB controller takes a long time to eliminate the steady-state error. This performance is only marginally acceptable, and improvement is highly desired. To apply the basic LC to this nonlinear FB-loop, an "impulse response" measurement has to be made. For a nonlinear process, the notion of "impulse response" is imprecise and, at best, a first-order approximation (in the sense of a Volterra series expansion [4].) It is expected to vary as a function of the process state and input. The degree of these variations depends on the degree of the nonlinearity.

In this study, the $f$ to $u_{fb}$ "impulse response" model is used. [5] Figure 3 shows three estimated "impulse responses" (corresponding to the transfer from the FF signal to the FB controller ouput). The first two are estimated by inputting a FF step of 10% and 30% power command, respectively into the model. Then, the "impulse responses" are estimated by differencing the

[2] $\mathcal{P}^{-1}$ assumes the existence of process "right inverse" [3, 1]. A *necessary* condition is that the number of process inputs must be no less than the number of outputs. With the existence of $\mathcal{P}^{-1}$, the existence of $\mathcal{M}^{-1}$ for the closed-loop is assured. When the right inverse does not exist, least-squares solutions can be sought [1] according to the computational formulation to follow.

[3] If $\bar{u}_{fb}$ is used for learning, then the impulse response sequence should be from $f$ to $u_{fb}$.

[4] Equations (9) and (10) have assumed that the only time delay is the sample delay, i.e., 1 delay. If additional process delays exist, some of the leading $h$'s would be zero and should be removed from the equation, together with the corresponding leading $e$'s.

[5] This is because for single-input single-output (linear) processes, the transfer function from the command $r$ to the wafer temperature $y$ is the same as the transfer function from FF ($f$) to FB ($u_{fb}$) with an exception of the sign. Knowing the $r$ to $y$ temperature step response from the FB-only performance (and hoping that the nonlinearity is not too severe), one can estimate the FF to FB impulse response without having to conduct special system identification experiments. In practical applications, these savings could be meaningful.

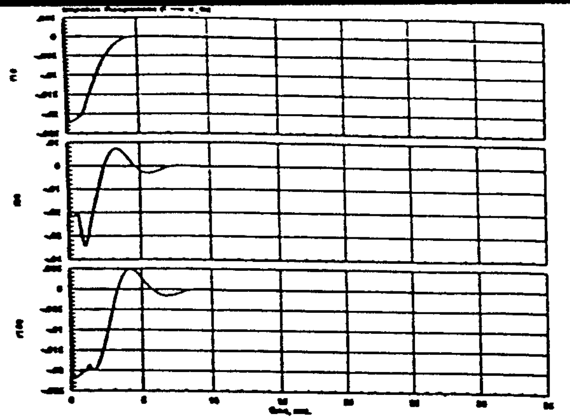


Figure 3: Three (3) Estimated "Impulse Responses" for the Nonlinear RTP Model

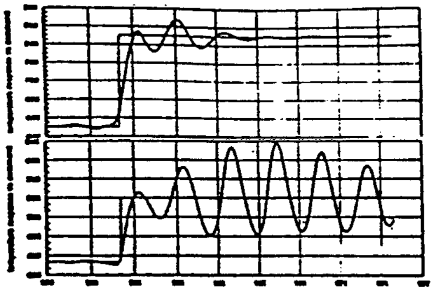


Figure 4: Basic LC Results using "Impulse Response" Model – $h_{r100}$

step responses. The third "impulse response," $h_{r100}$, is *inferred* from the actual temperature response to the temperature step command (650° C to 750° C) according to the above (footnote) discussion. It is apparent that these three "impulse responses" exhibit quite different gain and dynamic characteristics. Since the third "impulse response" is the one that in practice can save a system identification experiment, we use it in the simulation for studying the basic LC. Figure 4 shows the results after one task repeat and after three task repeats. It is apparent that with this particular "impulse response," $h_{r100}$, the basic LC seems to diverge. In practical applications, however, one may use only *a fraction of* the FF signal recommended by the basic LC algorithm to increase the chance of convergence. In fact, with this "divergent" $h_{r100}$, good results are obtained after *many* iterations, *if* the basic LC recommended FF signal is discounted by 50% at each iteration. Certainly, one could ask questions such as: what is the minimum amount of discount that would still guarantee convergence, etc. The *real issue*, however, is that the many iterations means potentially many calibration runs which would make the method less attractive. Instead, using the proposed new adaptive learning FF control method, in the next section we present some very attractive results with convergence virtually achieved after two or three adaptations.

## 4 Adaptive LC Applied to the Nonlinear RTP Model

Using the adaptation scheme to be described in the *next* section, rapid convergence is demonstrated here with the nonlinear RTP model. Figure 5 (upper) shows the results after 4 task repeats (or 3 adaptations). [6] The FB-only response is also shown in the upper plot to contrast the performance improvement effected by the adaptive LC. The temperature response has virtually conformed to the *desired response* which is that of a first-order system with a 0.5 second time constant. The desired response is displayed in the lower plot for the ease of comparison.

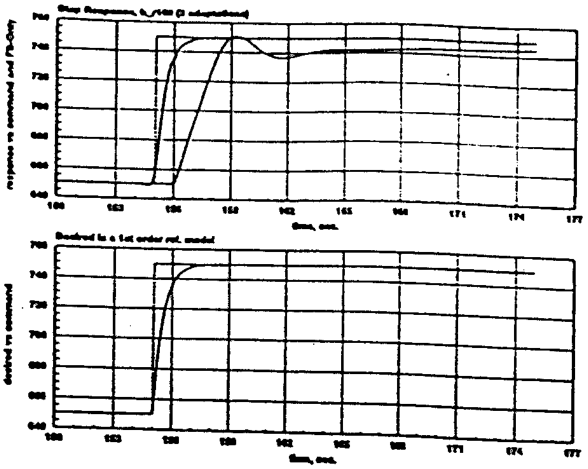


Figure 5: Adaptive Learning Control Result: nonlinear RTP model (after 4 task repeats)

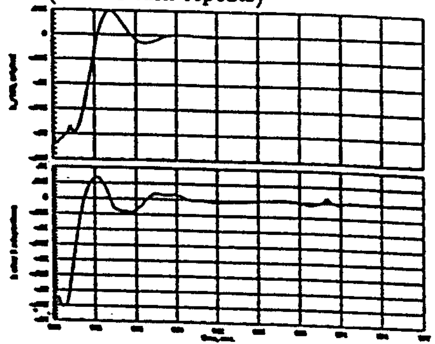


Figure 6: Impulse Response Models: before and after adaptation

Figure 6 compares the (final) *adapted* "impulse response" with the original "impulse response," $h_{r100}$, which is used prior to adaptation by the basic LC algorithm to perform the first task repeat. Figure 7 shows the sequence of intermediate results with 0, 1, 2 and 3 adaptations (or equivalently, with 1, 2, 3 and 4 repeats). It is noticed that the learning process has converged rapidly after 2 adaptations. With this kind of convergence, this technique can be very useful in practical manufacturing processes that exhibit a considerable degree of nonlinearity as this RTP model. By investing in a few calibration runs, one may gain substantial improvement over many production runs. Once a near optimal FF signal is established, this technique can also be used for fine tuning as the equipment drifts over periods of time.

Before discussing the details of this new adaptation scheme in the next Section, a case subject to periodic measurement disturbance (with random phase) is presented here to demonstrate the algorithm's robustness in a noisy environment. Figure 8 shows the results after 4 repetitive applications of the adaptive LC algorithm (i.e., 3 adaptations). The learned response is plotted against the (step) command, the response of the reference model, and the FB-only response subject to the same random-phased disturbance condition. Apparently, the learned response tracks the reference response quite well in the face of the disturbance. It is noticed that the adaptive LC does not seem to aggravate the disturbances, and the steady-state disturbance level remains about the same as that in the FB-only case. Note

[6] Since learning is "from scratch," i.e., the initial FF is zero, no adaptation takes place until after the first task repeat and therefore there are only 3 adaptations in 4 task repeats. (Since learning has stopped after 4 repeats, no adaptation is made after the 4th repeat either.)

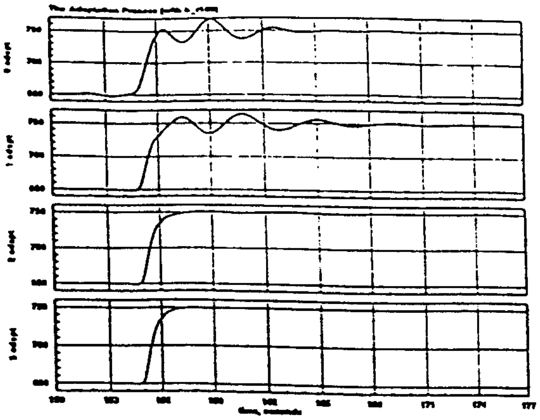


Figure 7: Adaptive Learning Control Results: the learning sequence

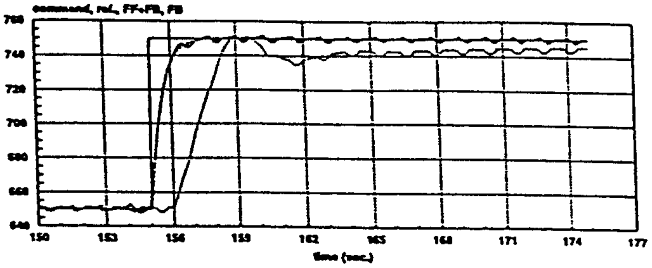


Figure 8: Adaptive Learning Control: a noisy case

that in this simulation, no special filtering is used. The robust performance is a result of 1) adapting only the early portion of the impulse response model, [7] and 2) using "soft" inversion instead of "exact" inversion for LC computations between task repeats to avoid amplifying noise. [8] In practical applications, proper filtering [1, 7] of the signals would add extra protection against disturbances and noise. [9]

## 5 The New Adaptation Scheme – SISO Case

Since the notion of "impulse response" for a nonlinear process is imprecise, the model would vary as a function of the process state and input. The idea then is to *adaptively* adjust the impulse response *model* as the learning process proceeds. The basis for adaptation is the mismatch between what is expected and what one actually gets. For the ease of presentation, we focus on single-input single-output (SISO) case in this section. Extension to the MIMO case is considered in Section 7.

Figure 9 illustrates this *prediction error* based adaptation scheme. After the $i$-th task repeat, the basic LC scheme is used to determine the FF increment, $df^{(i)}$, for the next repeat based on the *best* impulse response model, $\hat{h}^{(i)}$, that one has at the time. Applying this FF increment, $df^{(i)}$, to the process, one can expect (*predict*) a new (tracking) error according to the current impulse response model:

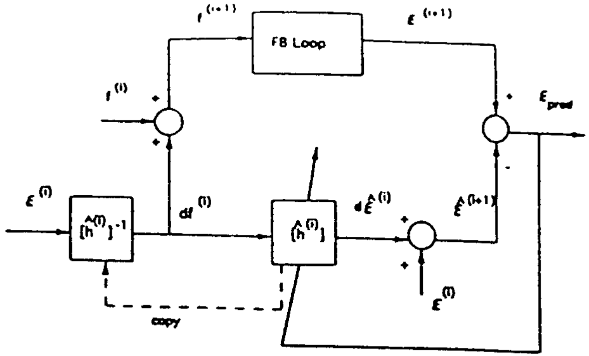


Figure 9: Prediction Error based Adaptive LC Scheme

$$\hat{\mathcal{E}}^{(i+1)} = \mathcal{E}^{(i)} + [\hat{h}^{(i)}]df^{(i)} \tag{13}$$

where $\mathcal{E}^{(i)}$ is the actual measured (tracking) error (i.e., $\tilde{u}_{fb}$ in our case) from the $i$-th task repeat, and $[h]$ stands for the Toeplitz matrix form of $h$ as shown in Equation (10) which actually performs *convolution* between $h$ and $df$. Before convergence is achieved, the expected error for the $(i+1)$-th task repeat will be different from the actual observed error due (primarily) to the errors associated with the impulse response model. This difference, termed *prediction error*, is defined as:

$$\mathcal{E}_{pred}^{(i+1)} \equiv \mathcal{E}^{(i+1)} - \hat{\mathcal{E}}^{(i+1)} \tag{14}$$

Substituting for $\hat{\mathcal{E}}^{(i+1)}$,

$$\begin{aligned} \mathcal{E}_{pred}^{(i+1)} &= \mathcal{E}^{(i+1)} - \mathcal{E}^{(i)} - [\hat{h}^{(i)}]df^{(i)} && (15) \\ &= \mathcal{E}^{(i+1)} - \mathcal{E}^{(i)} - [df^{(i)}]\hat{h}^{(i)} && (16) \end{aligned}$$

where we have used the fact that the convolution between $h$ and $df$ is commutative to arrive at the second equality. This is a *key technical step* leading to the following adaptation algorithm:

$$\mathcal{E}_{pred} = \begin{bmatrix} \bar{e}_2 \\ \bar{e}_3 \\ \bar{e}_4 \\ . \\ . \\ . \\ \bar{e}_{N+1} \end{bmatrix} \tag{17}$$

$$= \begin{bmatrix} df_1 & 0 & 0 & \dots & 0 \\ df_2 & df_1 & 0 & \dots & 0 \\ df_3 & df_2 & df_1 & \dots & 0 \\ . & . & . & \dots & . \\ df_N & df_{N-1} & df_{N-2} & \dots & df_1 \end{bmatrix} \begin{bmatrix} d\hat{h}_1 \\ d\hat{h}_2 \\ d\hat{h}_3 \\ . \\ . \\ . \\ d\hat{h}_N \end{bmatrix} \tag{18}$$

and adapt $\hat{h}$ according to:

$$\hat{h}^{(i+1)} = \hat{h}^{(i)} + d\hat{h} \tag{19}$$

Solving this matrix equation, Eq. (18), can be approached by exact inversion, constrained least-squares, or "soft" inversion using a finite number of *numerical* iterations with optimal gradient descent, etc. [10] For optimal gradient descent, the gradient (with respect to $\hat{h}$) is readily expressed as:

$$\begin{aligned} g &= \frac{1}{2}\frac{\partial(\mathcal{E}_{pred}^{(i+1)})^2}{\partial\hat{h}^{(i)}} && (20) \\ &= -[df^{(i)}]'\mathcal{E}_{pred}^{(i+1)} && (21) \end{aligned}$$

[7] This is a simple robustification procedure discussed in the next Section.

[8] Also, see Section 5 for more details on this.

[9] In [7], experimental work of applying LC to wafer temperature control in a typical RTP reactor is discussed. Considerable sensor disturbances are encountered and successfully removed. The operation condition, however, is linear enough that the basic LC proves to be quite adequate.

[10] The details are described in our previous paper [1]. (In the face of uncertainties, *soft* inversion is often preferred to exact inversion – a principle also found in iterative image deblurring [8] and adaptive signal processing [9].)

Hence, the Optimal Gradient algorithm for the adaptation of $\hat{h}$ is [5]:

$$\hat{h}^{(i+1)} = \hat{h}^{(i)} - \mu g \tag{22}$$

$$= \hat{h}^{(i)} + \mu[df^{(i)}]'\mathcal{E}_{pred}^{(i+1)} \tag{23}$$

$$\mu = (g'g)/(g'[df^{(i)}]'[df^{(i)}]g) \tag{24}$$

*Numerically* iterating with the Optimal Gradient Algorithm finitely many times performs *soft* inversion for Eq. (18) at each task repeat.

Notice that this matrix equation for $h$ adaptation is exactly the *dual* to the basic LC algorithm which solves for $df$ (Eq. (10)). The roles of $dh$ and $df$ are exchanged in these two *dual problems*. This allows the *reuse* of the software developed for the basic LC in the adaptation of $h$. It also allows the variant LC *algorithms* [1] to be used in $h$ adaptation. For instance, the rate-constrained LC algorithm might be useful in the adaptation of $h$ to discourage measurement noise from entering the adapted $h$. However, this rate constraining should be used primarily on the "tail" portion of the impulse response model, $\hat{h}$. This is because the early portion of $\hat{h}$ has high signal-to-noise ratio (SNR) and usually exhibits fast transient response due to the dominant process dynamics, and therefore should not be penalized for rate of change.

Implemented in a simplified manner, one may choose to adapt only the early portion of the $\hat{h}$ model and leave the rest intact. This way, the potential contamination by measurement noise is minimized (provided that a reasonable "tail" portion of the initial $\hat{h}$ is available). For example, if only the first $k$ terms of $\hat{h}$ are adapted, the adaptation matrix equation becomes:

$$\begin{bmatrix} \bar{e}_2 \\ \bar{e}_3 \\ \bar{e}_4 \\ \cdot \\ \cdot \\ \cdot \\ \bar{e}_{k+1} \\ \cdot \\ \cdot \\ \bar{e}_{N+1} \end{bmatrix} \simeq \begin{bmatrix} df_1 & 0 & 0 & \dots & 0 \\ df_2 & df_1 & 0 & \dots & 0 \\ df_3 & df_2 & df_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ df_N & df_{N-1} & df_{N-2} & \dots & df_{N-k} \end{bmatrix} \begin{bmatrix} d\hat{h}_1 \\ d\hat{h}_2 \\ d\hat{h}_3 \\ \cdot \\ \cdot \\ \cdot \\ d\hat{h}_k \end{bmatrix} \tag{25}$$

This simple scheme actually works quite effectively with noise according to the simulated results. It also seems to *robustify* the adaptation scheme even in a noiseless environment. This is because sometimes "tail swings" may occur in the intermediate runs before convergence is achieved. Being able to hold fast onto the steady-state portion of the $\hat{h}$ model would encourage consistently good steady-state convergence.

# 6 Justifications of the Adaptation Scheme

In this Section, we offer a simplified but insightful illustration of the workings of this adaptation scheme. By reducing the dimension of the $h$ vector to 1, the problem becomes that of solving a single-variable nonlinear algebraic equation. In that context, it then becomes clear that the proposed adaptation *is* the secant method [6]. The secant method is an effective method for solving nonlinear algebraic equations and is convergent if the initial trial solution is close enough to the true solution. Visualizing the iterative solution process provides deeper understanding of this adaptation scheme and helps sketch out possible formal proofs of convergence by qualifying the underlying nonlinearity. Extending from the nonlinear algebraic analogy to a dynamic setting can be accomplished using the concepts from operator theory and functional analysis [2].

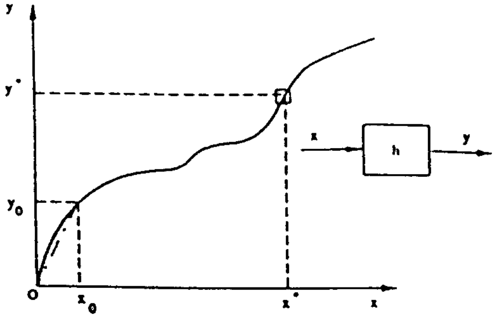


Figure 10: A Nonlinear Algebraic Equation Solving Analogy

Considering a one-dimensional case, Figure 10 depicts a problem of finding the right input, $x^*$, that will produce the desired outcome, $y^*$, on a nonlinear curve. We decide to use *linear model* templates to successively approach the right solution. To get started, we conduct a "system identification" experiment by putting in input, $x_0$, and observing the corresponding $y$ value on the nonlinear curve, i.e., $y_0$. Based on this experiment, we build the first *linear* model, $\hat{h}^{(0)}$, which is the equation describing a line passing through the origin with slope of $y_0/x_0$. Using this initial model, we now proceed to find $x^*$ by determining what $x$ value would give $y^*$ as output. By solving the equation, $\hat{h}^{(0)}(x) = y^*$, it is determined that $x_1$ would be the "right" $x$ value. However, the nonlinear curve responds to $x_1$ with a value, $y_1$ that is quite different from the desired $y^*$ as Figure 11 illustrates. The difference between the actual, $y_1$, and the expected, $y^*$, is used (together with the amount of $x$ incremented, i.e., $x_1$) to determine the modification that should be applied to $\hat{h}^{(0)}$. After this modification (adaptation), the updated linear model, $\hat{h}^{(1)}$, is the line passing through the origin and the point, $(x_1, y_1)$. Extending this line to find the value of $x$ that would produce the desired $y^*$, we obtain $x_2$.

However, instead of reaching the desired $y^*$ we obtain the value $y_2$ by applying $x_2$ to the nonlinear curve. Notice that $y_2$ is still considerably less than the desired $y^*$. The difference between $y_2$ and $y^*$ is the *prediction error* which is used (together with the amount of $x$ incremented, i.e., $x_2 - x_1$) to determine the adaptation needed for $\hat{h}^{(1)}$. The updated linear model, $\hat{h}^{(2)}$, is the line passing through the points, $(x_1, y_1)$ and $(x_2, y_2)$. Using this updated linear model, it is recommended that $x_3$ be used to reach the desired goal.

Repeating this prediction and correction cycle one more time, we have another new linear model, $\hat{h}^{(3)}$, which is the line passing through points $(x_2, y_2)$ and $(x_3, y_3)$. Now this new linear model, $\hat{h}^{(3)}$, recommends $x_4$ as the input for achieving our goal. This time, the recommendation is very close to being optimal, (since the result of applying $x_4$ to the nonlinear curve is very close to the desired response, $y^*$,) and the process has practically converged.

It is noted that the above procedure is identical to the so-called *secant method* in algebraic equation solving [6]. The secant method is a *locally* convergent method with a respectable order of 1.62 convergence rate. The fact that the secant method does *not* require the evaluation of any *derivatives* (i.e., gradients) is of special advantage here, because it translates to the adaptive LC scheme *not* requiring any special *system identifica-*

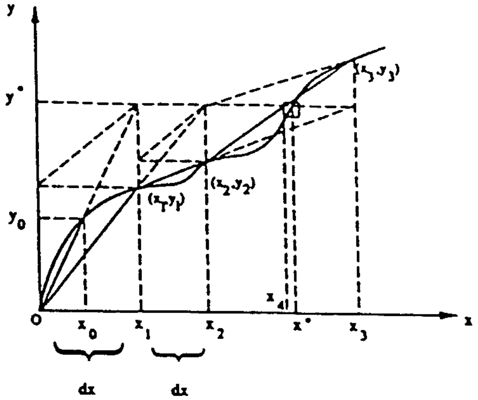


Figure 11: The Nonlinear Algebraic Equation Solving Analogy (in action)

*tion experiments* except for the first run.

## 7 The New Adaptation Scheme – MIMO Case

In this section, we extend the SISO adaptation algorithm to the multi-input multi-output (MIMO) case. For MIMO cases, there are some added technical complexities that make it more challenging and illuminating.

First of all, the *key technical step* in the development of SISO adaptation, i.e., from Eq. (15) to (16), no longer holds. This is because the MIMO nature makes $h$ a matrix and $df$ a composite "long" vector which contains segments of multiple-input $df$'s. "Commutation" is no longer as straight forward as it is in Eqs. (15) and (16). Instead, the following equation (i.e., essentially Eqs. (15) and (18) combined)

$$\begin{bmatrix} \tilde{e}_2 \\ \tilde{e}_3 \\ \tilde{e}_4 \\ \cdot \\ \cdot \\ \cdot \\ \tilde{e}_{N+1} \end{bmatrix} = \begin{bmatrix} d\hat{h}_1 & 0 & 0 & \dots & 0 \\ d\hat{h}_2 & d\hat{h}_1 & 0 & \dots & 0 \\ d\hat{h}_3 & d\hat{h}_2 & d\hat{h}_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ d\hat{h}_N & d\hat{h}_{N-1} & d\hat{h}_{N-2} & \dots & d\hat{h}_1 \end{bmatrix} \begin{bmatrix} df_1 \\ df_2 \\ df_3 \\ \cdot \\ \cdot \\ df_N \end{bmatrix} \tag{26}$$

can be transformed to:

$$\begin{bmatrix} \tilde{e}'_2 \\ \tilde{e}'_3 \\ \tilde{e}'_4 \\ \cdot \\ \cdot \\ \tilde{e}'_{N+1} \end{bmatrix} = \begin{bmatrix} df'_1 & 0 & 0 & \dots & 0 \\ df'_2 & df'_1 & 0 & \dots & 0 \\ df'_3 & df'_2 & df'_1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ df'_N & df'_{N-1} & df'_{N-2} & \dots & df'_1 \end{bmatrix} \begin{bmatrix} d\hat{h}'_1 \\ d\hat{h}'_2 \\ d\hat{h}'_3 \\ \cdot \\ \cdot \\ \cdot \\ d\hat{h}'_N \end{bmatrix} \tag{27}$$

Let us use a 2 x 2 MIMO process for illustration. Notice that originally $\tilde{e}_k$ is a 2 x 1 column vector corresponding to the 2 output "prediction" errors at time $k$. Now, since $\tilde{e}'_k$ is a 1 x 2 row vector, the left hand side of Eq. (27) has become an $N \times 2$ matrix with the first column denoting the first output "prediction" errors and the second column the second output "prediction" errors. Likewise, the second entity on the right-hand-side of the equation has become an $2N \times 2$ matrix. The first column corresponds to the impulse response models of the first output (due to the 2 inputs). These observations suggest that the MIMO adaptation could be performed *one output at a time*. That is, the above matrix equation can be *split* into 2 sets of $[N \times 1] = [N \times 2N][2N \times 1]$ vector-matrix equations which can be (readily) solved using the SISO adaptation procedure.

However, noting that there are $2N$ $\hat{h}$'s to adapt, whereas there are only $N$ equations, this split formulation seems underdetermined. (This is partly due to the redundancy in impulse response modeling.) Fortunately, this problem can be overcome by invoking the *robust* SISO adaptation scheme that only a portion of the $\hat{h}$ model needs adaptation. Therefore, this MIMO adaptation problem can be readily solved one output at a time using the *robust* SISO adaptation scheme.

One more technical detail is that even with more equations than unknowns (by adapting only a portion of $\hat{h}$), this problem can still be ill-conditioned. This happens when the columns of, say, $[df_1, df_2, df_3, ..., df_N]'$ are (nearly) identical thereby rendering the $N \times 2N$ matrix in Eq. (29) (nearly) singular. Physically, this means when the $df$'s for the 2 inputs are nearly the same, the *multi-input aspect* of the MIMO process is not adequately *probed* to allow reliable identification of individual input contributions. This phenomenon, called *multi-collinearity* [10], in multi-variable regression is known to lead to erroneous models. In practice, when the process does not respond to the 2 inputs equally, this may not be a problem. This is because then the 2 $df$'s are likely to be different. On the other hand, if the process is fairly input symmetric, it may be necessary to *perturb* the 2 $df$'s so that adequate excitations are available for reliable adaptation.

## 8 Summary

A new *adaptive* learning feedforward control scheme is introduced in this paper. This new scheme retains all the key features of the previous basic LC scheme [1], such as modeling simplicity and applicability with or without FB control, SISO or MIMO. Above and beyond, this new scheme extends the applicability of LC to include a class of smoothly *nonlinear* processes in an effective way. Rapid reductions in tracking errors are demonstrated using a nonlinear RTP manufacturing model. Computational algorithms are given along with robustness considerations. Analytical justifications are given using an analogy with the secand method used in algebraic equation solving.

## References

[1] K.M. Tao, R.L. Kosut and G. Aral, "Learning Feedforward Control," *Proc. American Control Conference*, Baltimore, MD, June - July, 1994, pp. 2575 - 2579.

[2] T.L. Saaty, *Modern Nonlinear Equations*, NY: Dover, 1981.

[3] J.L. Massey and M.K. Sain, "Inverses of Sequential Circuits," *IEEE Trans. Computers*, Vol. C-17, No. 4, April 1968, pp. 330 - 337.

[4] R. Johansson, *System Modeling and Identification*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[5] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Reading, MA: Addison-Wesley, 1973.

[6] A. Ralston, *A First Course in Numerical Analysis*, NY: McGraw-Hill, 1965.

[7] K.M. Tao, R.L. Kosut, M. Ekblad and G. Aral, "Feedforward Learning Applied to RTP of Semiconductor Wafers," *Proc. 33rd. IEEE Conf. Decision and Control*, Lake Buena Vista, FL, December 1994.

[8] J. Biemond et al., "Iterative Image Deblurring," *Proc. IEEE*, Vol. 78, No. 5, May 1990, pp. 856 - 883.

[9] K.M. Tao, "Statistical Averaging and PARTAN – Some Alternatives to LMS and RLS," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, March 1992, pp. IV-25 - IV-28.

[10] M.S. Younger, *A Handbook for Linear Regression*, North Scituate, MA: Duxbury Press, 1979.