Concepts, Methods and Techniques in Adaptive Control

Guy A. Dumont[1] and Mihai Huzmezan [2]
Department of Electrical and Computer Engineering,
University of British Columbia,
V6T 1Z4, Vancouver, B.C., Canada

Abstract

This tutorial paper looks back at almost 50 years of adaptive control trying to establish how much more we need to secure for the industrial community an adaptive controller which will be used and referred to with the same ease as PID controllers are now. Since the first commercial adaptive controller, significant progress in the design and analysis of these controllers has been achieved. Various forms of adaptive controllers are now readily available targeting a significant range of industries from process to aerospace. A general overview of adaptive control will allow the audience to place on the map several industrial architectures for such controllers, all with the aim of bridging the gap between academic and industrial views of the topic. Such a presentation of design and analysis tools currently opens a more philosophical question "Has the critical mass in adaptive control been reached?"

Keywords: Adaptive Control,

1 Introduction

According to the Webster's dictionary, "to adapt" means: "To adjust oneself to particular conditions; To bring oneself in harmony with a particular environment; To bring one's acts, behavior in harmony with a particular environment", while adaptation means: "Adjustment to environmental conditions; Alteration or change in form or structure to better fit the environment".

For a control system, the plant constitutes the environment. Plant nonlinearities can be found in most of the processes from flight to process control. For instance, in steel rolling mills, paper machines or rotary kilns, the dynamics can change due to nonlinear actuators or sensors (e.g. nonlinear valves, pH probes), flow and speed variations raw material variability or wear and tear. For an aircraft, nonlinearities are mainly correlated with the compressibility of the air and the turbulent flow around control and lift surfaces. In ship steering, changing wave characteristics represent a major challenge.It is well recognized that linear feedback can cope fairly well with parameter changes within certain limits.

Following this the immediate question is "When is a controller adaptive?". A short possible answer was offered by G. Zames during a presentation made at the 35th Conference in Decision and Control, Kobe, Dec. 1996: "a non-adaptive controller is based solely on a-priori information whereas an adaptive controller is based also on a posteriori information". When the process changes with operating conditions in a known, predictable fashion gain scheduling can provide the required performance. Therefore in cases like: flight control systems, compensation for production rate changes, compensation for paper machine speed the controller parameters can be made to vary in a predetermined fashion by the operating conditions.

The main focus of this paper is connected with a narrower definition of adaptive control, which is centered around a fixed-structure controller with adjustable parameters, i.e. the controller possesses a mechanism for automatically adjusting them, based on posterior information. From an academic perspective adaptive control theory essentially deals with finding parameter adjustment algorithms that offer global stability and convergence guarantees. Linear controllers with guaranteed robust performance properties appear to a designer as the natural choice for simple, certifiable and easy to implement controllers. Our experience shows that if the use of a fixed controller cannot achieve a satisfactory compromise between robust stability and performance, then and only then, should adaptive control be used. The use of adaptive control is justified on complex systems exhibiting time-varying dynamics.

In this realm, as academics but also engineers with strong industrial ties, our immediate advice is to use the simplest technique that satisfies the specifications, using the words of great Einstein, we should strive to "make things as simple as possible, but no simpler".

[1]e-mail: guyd@ece.ubc.ca
[2]e-mail:huzmezan@ece.ubc.ca

This paper looks back at almost 50 years of adaptive control trying to establish how much more we need to do in order to secure for the industrial community an adaptive controller that will be used and referred to with the same ease as PID controllers are now. Since the first commercial adaptive controller, significant progress in the design and analysis of these controllers has been achieved. Various forms of adaptive controllers are now readily available, targeting a significant range of industries from process to aerospace.

A general overview of adaptive control will allow the audience to understand the architectures of such controllers, all with the aim of bridging the gap between academic and industrial views on this topic.Leaving behind for a second the historical evolution of adaptive control, the present reality is that a number of active control methods are directly linked to the fundamental idea of adjusting a parameter of the controller based on further evolutions of the plant. Therefore, in no particular order we can enumerate adaptive control approaches like: model reference adaptive control, self/auto tuning, model following, multi-model identification and control, pseudo inverse projection, dual control or even feedback linearization.

In this paper one of our major goals is to present succinct information on a number of methods, techniques and concepts used in adaptive control and which are now embedded in commercial products or are ready for implementation. Such information will permit the industrial readers to better evaluate their needs.

The paper will be structured as follows:

Section ?? will be allocated to several challenges posed by most adaptive controllers such as: passive versus active learning, opportunities for analytic solutions, practical methods for having the controllers with the adaptation mechanism on at all times, control of processes with rapidly time-varying dynamics and the commissioning of adaptive controllers.

2 Development of Adaptive Control

The history of adaptive control goes back nearly 50 years, [30]. The development of adaptive control started in the 1950's with the aim of developing adaptive flight control systems, although that problem was eventually solved by gain scheduling. Among the various solutions that were proposed for the flight control problem, the one that would have the most impact on the field was the so-called model-reference adaptive system (MRAS). Figure 1 depicts a typical MRAS where the specifications are in terms of a reference model and the parameters of the controller are adjusted directly to achieve those specifications. Although the original

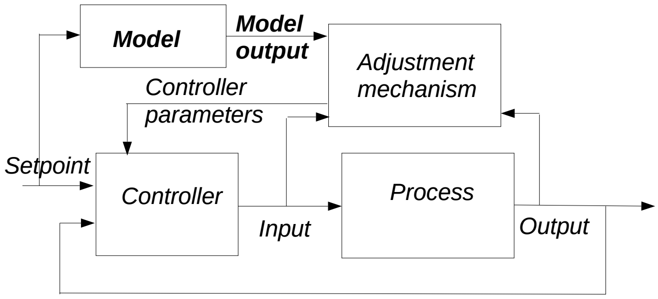

Figure 1: Model-reference adaptive control

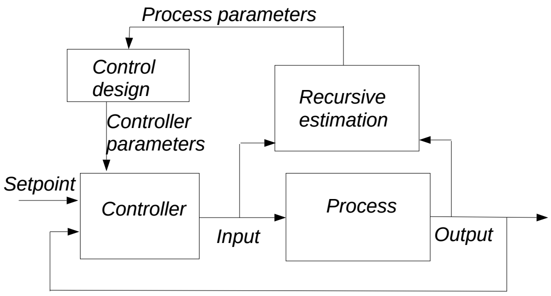

Figure 2: Indirect self-tuning control

algorithm proved unstable, it lead to the development during the 1970's and 1980's of algorithms with guaranteed stability, convergence and robustness properties. In 1957 Bellman developed dynamic programming. In 1958 Kalman [?] developed the self-optimizing controller "which adjusts itself automatically to control an arbitrary dynamic process". This would provide the foundation for the development of the self-tuning regulator by Peterka [?] and Åström and Wittenmark [?]. Self-tuning control consists of two operations:

- Model building via identification
- Controller design using the identified model

Self-tuning control can be thought of as an on-line automation of the off-line model-based tuning procedure performed by control engineers. Figure 2 depicts the architecture of an indirect self-tuning controller where these two operations are clearly seen. It is possible to reformulate the self-tuning problem in a way that the model estimation step essentially disappears, in which case the controller parameters are directly adapted, this is the so-called direct self-tuner, closer in spirit to the MRAS.

In 1957, Bellman [?] invented dynamic programming which he later applied to adaptive control [?]. Not very much later in 1960 Feldbaum [?] developed the dual controller in which the control action serves a dual purpose as it is "directing as well as investigating". In a major departure from the MRAS an STC schemes

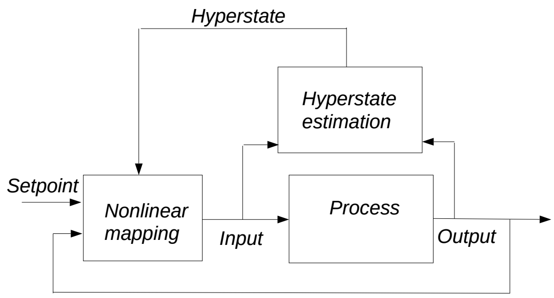

Figure 3: Dual control

which relied on the so-called certainty-equivalence principle (somewhat of a euphemism to say that those schemes ignore the uncertainty on the estimates by treating them as true values), the dual controller explicitely accounts for the uncertainty and attempts to reduce it. Figure 3 depicts the architecture of dual control which uses nonlinear stochastic control theory and amalgamates both parameters (and their uncertainties) and state variables into a hyperstate, which yields the control signal via a nonlinear mapping. The dual controller can handle very fast parameter changes and will constantly seek the best compromise between the regulation performance, caution and probing. Unfortunately, the solution to the dual control problem is untractable for most systems.

Following this initial wave of adaptive controllers the 1970's and 1980's saw rapid development in the field. In the early 1980's the first convergence and stability analysis proofs appeared, followed by a systematic robustness analysis.

3 Model Reference Adaptive Control

The model reference adaptive control is in fact a class of direct self tuners since no explicit estimate or identification of the plant is made. In exchange the controller parameters are identified directly. This approach leaves no room for checking the model quality.

As simple way to produce a model reference adaptive controller is to start with a time varying matrix of gains $K(t)$. This methodology applies to several approaches among which the MIT rule is probably the most classic one. In the MIT rule case the gain is chosen to minimize the following loss function $J(K(t)) = 0.5e^2(t)$. To make $J(K(t))$ small we should change $K(t)$ in the direction of the negative gradient:

$$\frac{dK(t)}{dt} = -\gamma \frac{\partial J(K(t))}{\partial K(t)} = -\gamma e(t) \frac{\partial e(t)}{\partial K(t)} \quad (1)$$

where $\partial e(t)/\partial K(t)$ is the partial derivative called sensitivity derivative of the system.

As an example let us consider the problem of a SISO plant for which its gain is unknown (i.e. $P(s) = kP_0(s)$, where $P_0(s)$ is what we call nominal plant). We apply the MIT rule to find the controller parameter θ when the gain k is unknown. The plant model is $P_m(s) = k_0P_0(s)$, where K_0 is a given constant. The defined error in this case is:

$$e(t) = y(t) - y_m(t) = kP(l)\theta r(t) - k_0P(l)r(t) \quad (2)$$

where $y(t)$, $y_m(t)$, $r(t)$ and $l = d/dt$ are the plant output, plant model output and reference, tuning parameter and differential operator, respectively. The sensitivity derivative is:

$$\frac{\partial e(t)}{\partial \theta} = kP(l)r(t) = \frac{k}{k_0} y_m(t) \quad (3)$$

The MIT rule gives the following tuning for θ:

$$\frac{d\theta}{dt} = -\gamma' \frac{k}{k_0} y_m(t)e(t) = \gamma y_m(t)e(t) \quad (4)$$

Note that for a correct value of γ sign knowledge of k sign is required.

In the industrial world there were reports of instability based on the basic MIT rule. It has been understood that the choice of the adaptation gain is critical and its value depends on the signal levels. Normalizing the signals will create the required independence for this algorithm. So the MIT rule has to be modified as follows:

$$\frac{d\theta}{dt} = \gamma \phi e(t) \quad (5)$$

where $\phi = \partial e(t)/\partial \theta$. The adjustment rule:

$$\frac{d\theta}{dt} = \frac{\gamma \phi e(t)}{\alpha + \phi^T \phi} \quad (6)$$

where $\alpha > 0$ is introduced to avoid zero division when $\phi^T\phi$ is small. In the above θ can be a vector of parameters.

4 Self-Tuning Control

4.1 Recursive identification for adaptive control

All methods that use the least–squares criterion

$$V(t) = \frac{1}{t} \sum_{i=1}^{t} [y(i) - x^T(i)\hat{\theta}]^2$$

identify the average behaviour of the process. When the parameters are time varying, it is desirable to base the identification on the most recent data rather than on the old one, not representative of the process anymore. This can be achieved by exponential discounting of old data, using the criterion

$$V(t) = \frac{1}{t}\sum_{i=1}^{t}\lambda^{t-i}[y(i) - x^T(i)\hat{\theta}]^2$$

where $0 < \lambda \leq 1$ is called the forgetting factor. The new criterion can also be written

$$V(t) = \lambda V(t-1) + [y(t) - x^T(t)\hat{\theta}]^2$$

Then, it can be shown (Goodwin and Payne, 1977) that the RLS scheme becomes

$$\begin{aligned}
\hat{\theta}(t+1) &= \hat{\theta}(t) + K(t+1)[y(t+1) - x^T(t+1)\hat{\theta}(t)] \\
K(t+1) &= P(t)x(t+1)/[\lambda + x^T(t+1)P(t)x(t+1)] \\
P(t+1) &= \left\{P(t) - \frac{P(t)x(t+1)x^T(t+1)P(t)}{[\lambda + x^T(t+1)P(t)x(t+1)]}\right\}\frac{1}{\lambda}
\end{aligned}$$

In choosing λ, one has to compromise between fast tracking and long term quality of the estimates. The use of the forgetting may give rise to problems.

The smaller λ is, the faster the algorithm can track, but the more the estimates will vary, even the true parameters are time-invariant.
A small λ may also cause blowup of the covariance matrix P, since in the absence of excitation, covariance matrix update equation essentially becomes

$$P(t+1) = \frac{1}{\lambda}P(t)$$

in which case P grows exponentially, leading to wild fluctuations in the parameter estimates.
One way around this is to vary the forgetting factor according to the prediction error ε as in

$$\lambda(t) = 1 - k\varepsilon^2(t)$$

Then, in case of low excitation ε will be small and λ will be close to 1. In case of large prediction errors, λ will decrease.

The Exponential Forgetting and Resetting Algorithm (EFRA) due to Salgado, Goodwin and Middleton[1] allows tracking of time-varying parameters while guaranteeing boundedness of the covariance matrix:

$$\begin{aligned}
\varepsilon(t+1) &= y(t+1) - x^T(t+1)\hat{\theta}(t) \\
\hat{\theta}(t+1) &= \hat{\theta}^T(t) + \frac{\alpha P(t)x(t+1)}{\lambda + x^T(t+1)P(t)x(k+1)}\varepsilon(t) \\
P(t+1) &= \frac{1}{\lambda}\left[P(t) - \frac{P(t)x(t+1)x^T(t+1)P(t)}{\lambda + x(t+1)^T P(t)x(t+1)}\right] \\
&\quad +\beta I - \gamma P(t)^2
\end{aligned}$$

[1]M.E. Salgado, G.C. Goodwin, and R.H. Middleton, "Exponential Forgetting and Resetting", *International Journal of Control*, vol. 47, no. 2, pp. 477–485, 1988.

where I is the identity matrix, and α, β and γ are constants.

With the EFRA, the covariance matrix is bounded on both sides:

$$\sigma_{min} I \leq P(t) \leq \sigma_{max} I \qquad \forall t$$

where

$$\sigma_{min} \approx \frac{\beta}{\alpha - \eta} \qquad \sigma_{max} \approx \frac{\eta}{\gamma} + \frac{\beta}{\eta}$$

with

$$\eta = \frac{1-\lambda}{\lambda}$$

With $\alpha = 0.5$, $\beta = \gamma = 0.005$ and $\lambda = 0.95$, $\sigma_{min} = 0.01$ and $\sigma_{max} = 10$.

4.2 Prototype algorithms

Prototype single-input, single-output algorithms will be presented, along the line of [**?**]. Consider the simple process described by

$$A(q)y(t) = B(q)(u(t) + v(t)$$

where A and B are coprime, $\deg A = n$, the relative degree is $d = \deg A(q) - \deg B(q)$, and $v(t)$ is a load disturbance. The two-degree-of-freedom controller

$$R(q)u(t) = T(q)y_r - S(q)y(t)$$

will be used to obtain a closed-loop system as

$$A_m(q)y_m(t) = B_m(q)y_r(t)$$

With this two-degree-of-freedom controller, the closed-loop system is

$$y(t) = \frac{BT}{AR+BS}y_r(t) + \frac{BR}{AR+BS}v(t)$$

The closed-loop characteristic polynomial is thus

$$A_c = AR + BS$$

This equation is known as the Diophantine equation or the Bezout identity and it plays a central role in many control aspects of modern control theory. The design problem is thus to find R, S and T such that

$$\frac{BT}{AR+BS} = \frac{BT}{A_c} = \frac{B_m}{A_m}$$

Generally $\deg(AR + BS) > \deg A_m$. It means that BT and $AR + BS$ have a common factor A_0. As it is desirable to cancel only stable and well damped zeros, write

$$B = B^+B^-$$

where B^+ contains well damped stable zeros that can be cancelled and B^- contains unstable and poorly damped zeros that should not be cancelled, i.e.

$$B_m = B^-B'_m$$

It follows that B^+ is a factor of R, i.e.

$$R = B^+ R'$$

Then

$$\frac{B^+ B^- T}{AB^+ R' + B^+ B^- S} = \frac{B^- B'_m A_0}{A_m A_0}$$

Thus, the solution to the design problem is obtained from:

$$AR' + B^- S = A_0 A_m$$

and

$$T = B'_m A_0$$

The Diophantine equation has no solution if A and B^- have common factors. When A and B^- are coprime, it has an infinite number of solutions:

$$R = R_0 + B^- U$$

$$S = S_0 - AU$$

where R_0 and S_0 are solutions and U is an arbitrary polynomial. All those solutions have the same closed-loop properties, but differ by their noise rejection properties. Causality of the controller imposes

$$\deg S \leq \deg R$$

$$\deg T \leq \deg R$$

$$\deg A_m - \deg B'_m \geq \deg A - \deg B^+$$

It is also natural to look for minimum-degree solution

$$\deg S = \deg A - 1$$

$$\deg A_c = \deg B^+ + \deg A_0 + \deg A_m \geq 2 \deg A - 1$$

or

$$\deg A_0 \geq 2 \deg A - \deg B^+ - \deg A_m - 1$$

For minimum degree pole placement control, choose:

$$\deg A_m = \deg A$$

$$\deg B_m = \deg B$$

$$\deg A_0 = \deg A - \deg B^+ - 1$$

Then the pole placement algorithm is

1. Factor B as $B = B^+ B^-$ with B^+ monic
2. Find R' and S with $\deg S = \deg A - 1$ from

$$AR' + B^- S = A_0 A_m$$

3. With $T = B'_m A_0$ and $R = B^+ R'$, the controller is

$$Ru = Ty_r - Sy$$

Note that LQG controller can be obtained as the solution of the Diophantine equation

$$P(q)C(q) = A(q)R(q) + B(q)S(q)$$

The closed-loop characteristic polynomial is $P(q)C(q)$ where the stable polynomial $P(q)$ is obtained from the following spectral factorization:

$$rP(q)P(q^{-1}) = \rho A(q)A(q^{-1}) + B(q)B(q^{-1})$$

Indirect adaptive pole-placement
Given A_0, A_m, B'_m and n, at each sampling time do the following:

- Estimate $\hat{A}$ and $\hat{B}$ (also $\hat{C}$) if in a stochastic framework using RLS (or e.g. AML in the stochastic case)
- Perform pole-placement procedure as described previously using $\hat{A}$ and $\hat{B}$.

Note that pole placement is often done in a determistic framework, i.e. there is little perturbation on the process. It is then important to ensure sufficient excitation by frequent setpoint changes. We also need to find common factors between $\hat{A}$ and $\hat{B}$ and to factor $\hat{B}$ as $\hat{B} = \hat{B}^+ \hat{B}^-$

Direct adaptive pole placement

$$Ay(t) = Bu(t)$$

$$A_m y(t) = B_m u(t)$$

Consider the Diophantine equation

$$A_0 A_m = AR' + B^- S$$

Multiply both sides by $y(t)$

$$A_0 A_m y(t) = R' Ay(t) + B^- Sy(t) = R' Bu(t) + B^- Sy(t)$$

Because $R'B = R'B^+B^- = RB^-$

$$A_0 A_m y(t) = B^- [Ru(t) + Sy(t)]$$

This could be considered a process model. Note however, that it is nonlinear in the parameters. Only when all zeros are cancelled, i.e. when $B^- = 1$ does it become linear in the parameters.

Indirect vs. direct adaptive control

4.3 Self-Tuning vs. Auto-Tuning

- **Self-tuning**
  - Continuous updating of controller parameters

- Used for truly time-varying plants

- **Auto-tuning**
  - Once controller parameters near convergence, adaptation is stopped
  - Used for time invariant or very slowly varying processes
  - Used for periodic, usually on-demand tuning

4.4 A practical adaptive predictive controller

The set of Laguerre functions is particularly appealing to describe stable dynamic systems because it is simple to represent and is similar to transient signals. It also closely resembles Padé approximants. The continuous Laguerre functions, a complete orthonormal set in $L_2[0,\infty)$, can be represented by the simple and convenient ladder network shown in Figure 4 and can be described by:

$$F_i(s) = \sqrt{2p}\frac{(s-p)^{i-1}}{(s+p)^i} \quad , \quad i = 1,..,N \tag{7}$$

where i is the order of the function ($i = 1,..N$), and $p > 0$ is the time-scale. Based on the continuous network compensation method, the Laguerre ladder network of Fig. 4 can be expressed in a stable, observable and controllable state-space form as,

$$l(t+1) = Al(t) + bu(t) \tag{8}$$

$$y(t) = c^T l(t) \tag{9}$$

with $l^T(t) = [\ l_1(t) \quad l_2(t) \quad \ldots \quad l_N(t)\ ]^T$, and $c^T = [c_1 \ \ c_2 \ldots c_N]$. The l_i's are the outputs from each block in Fig. 4, and $u(t)$, $y(t)$ are the plant input and output respectively. A is a lower triangular $N \times N$ matrix where the same elements are found respectively across the diagonal or every subdiagonal , b is the input vector, and c is the Laguerre spectrum vector. The vector c gives the projection of the plant output onto the linear space whose basis is the orthonormal set of Laguerre functions. Some of the advantages of using the above series representation are that,(a) because of its resemblance to the Padé approximants time-delays can be very well represented as part of the plant dynamics, (b) theoretically the model order N does not affect the coefficients c_i, and (c) extensions to multivariable schemes do not require the use of interactor matrices (Zervos and Dumont, 1988b).

Most industrial applications of control use a discrete time model of the plant. It is possible to define a set of z-transfer functions that are orthonormal and have a structure similar to the continuous Laguerre filters :

$$L_i(z) = \frac{\sqrt{1-a^2}}{z-a}\left(\frac{1-az}{z-a}\right)^{i-1} \tag{10}$$

The above model can be identified using a simple least-squares type algorithm. Consider the real plant described by

$$y(t) = \sum_{i=1}^{N} c_i L_i(q) + \sum_{i=N+1}^{\infty} c_i L_i(q) + w(t) \tag{11}$$

where $w(t)$ is a disturbance. It is obvious that this model has an output-error structure, is linear in the parameters, and gives a convex identification problem. Because of that, and of the orthonormality of the Laguerre filter outputs (obtained if the plant is perturbed by a white noise or a PRBS), it is trivial to show that:

- Even if $w(t)$ it colored and non-zero mean, simple least-squares provide consistent estimates of the c_i's.
- The estimate of the nominal plant, i.e. of c_i, for $i = 1, \cdots, N$ is unaffected by the presence of the unmodelled dynamics represented by c_i, for $i = N+1, \cdots, \infty$.

Wahlberg (1991) shows that the the mapping $(1 + ae^{i\omega})(e^{i\omega} + a)$ improves the condition number of the least-squares covariance matrix. Furthermore, the implicit assumption that the system is low-pass in nature reduces the asymptotic covariance of the estimate at high frequencies. For recursive least-squares, Gunnarsson and Wahlberg(1991) show that the mean square error of the transfer function estimate can be approximated by

$$\hat{\pi}(e^{i\omega}) = \frac{1}{2}\left(N(1-\lambda)\frac{1-a^2}{|e^{i\omega}-a|^2}\frac{\Phi_v(e^{i\omega})}{\Phi_u(e^{i\omega})} + \frac{\mu^2}{1-\lambda}r_1(e^{i\omega})\right)$$

Note that the case $a = 0$ corresponds to a FIR model. The MSE is seen to be proportional to the number of parameters. Compared with a FIR model, a representation in terms of an orthonormal series representation will be more efficient, will require less parameters and thus will give a smaller MSE. Furthermore, the disturbance spectrum is scaled by

$$\frac{1-a^2}{|e^{i\omega}-a|^2}$$

thus reducing the detrimental effect of disturbances at high frequencies.

5 Dual Control

6 Iterative Control

n the early 1990's, several auhors have proposed a new approach, the so-called iterative control design approach which can be seen as an alternative to adaptive

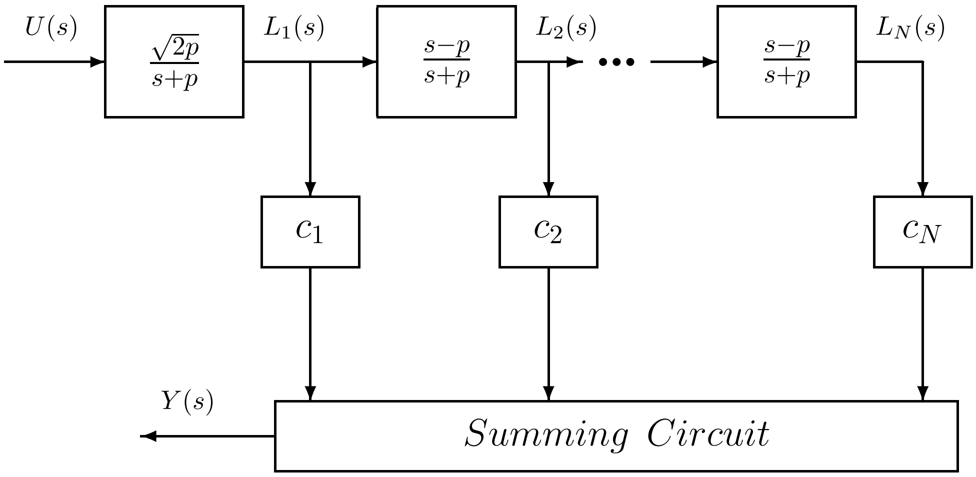

Figure 4: Representation of plant dynamics using a truncated continuous Laguerre ladder network.

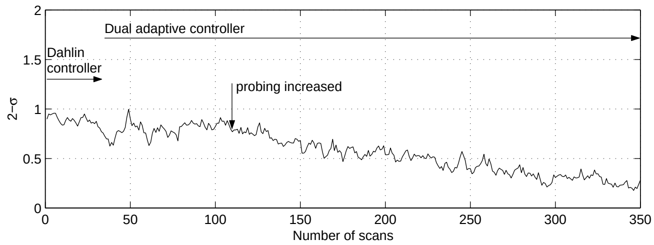

Figure 5: Trial 3, the benefit of probing is illustrated. From top: profile 2σ; some gain estimate profiles; some actuator profiles; some CD coat weight profiles

control, see for instance [?, ?, ?, ?, ?]. Because adaptive controllers combine identification with a time-varying controller, they results in systems with complex behaviours that are difficult to analyse. The basic idea of iterative control is to observe the system under fixed feedback for a sufficient long period, after which identification and control re-design is performed. The redesigned controller is then implemented and the process repated until a criterion indicative of satisfactory performance is met. Thus, at any given time, the system is under linear time-invariant control and is easy to analyse. It is like adaptive control with the control update only being done at the end of the experiment. Because a fixed controller is used during identification, an external perturbation has to be sent to the loop in order to guarantee closed-loop identifiability.

Although most iterative control design techniques that have been proposed rely on the identification of a process model to redesign the controller, in the spirit of indirect adaptive control, it is also possible to obtain iterative control design techniques that can be termed model-free in the sense that they estimate the controller parameters directly, in the spirit of direct adaptive control, see for instance [?, ?].

Iterative control design techniques have already met with a certain amount of success in industrial applications, particularly in iterative tuning of PID controllers, [?].

7 Multivariable Adaptive Control

We consider that a normalized coprime factor approach can be the self explanatory. Both indirect (e.g. achieved through an explicit separation between plant and controller parameters) and direct (e.g.wihtout an explicit identification of the plant parameters) self tuning are control methods with an ability to tune its own parameters.

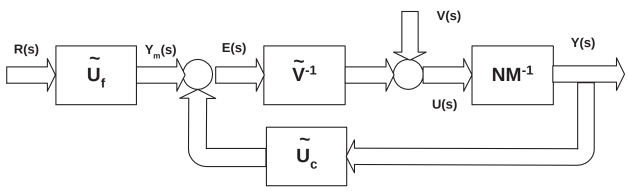

Figure 6: Coprime Factor Formulation of Self Tuners

Starting with a simple right coprime factorization of the plant [2] $P(s) = N(s)M^{-1}(s)$ and a 2 degrees of freedom controller implemented in an observer form [?]

[2] p is a transmission pole of $P(s)$ if $det(M(s)) = 0$ and z is a transmission zero of $P(s)$ if $det(N(s)) = 0$, order n of the system is $\delta det(N(s))$, definitions which correspond to a minimal realization of $P(s)$

as in Figure 6 we observe the following dependencies:

$$
\begin{aligned}
Y(s) &= N(s)M^{-1}(s) && (12)\\
Y_m(s) &= \tilde{U}_f(s)R(s) = N_m(s)M_m^{-1}(s)R(s) && (13)\\
E(s) &= Y_m(s) - \tilde{U}_c(s)Y(s) && (14)\\
U(s) &= \tilde{V}^{-1}(s)E(s) + V(s) && (15)
\end{aligned}
$$

The controller implementation is $C(s) = \tilde{V}^{-1}\tilde{U}_c$ based on a left coprime factorization that satisfies the Bezout identity $\tilde{V}(s)M(s) + \tilde{U}_c(s)N(s) = I$. If such a factorization exists it is equivalent to $C(s)$ being a stabilizing controller for $P(s)$.

Dropping the s from all transfer matrices in closed loop and assuming the load disturbance $V(s) = 0$ for the closed loop presented in Figure 6 the following dependencies can be observed:

$$Y(s) = NM^{-1}(I+\tilde{V}^{-1}\tilde{U}_cNM^{-1})^{-1}\tilde{V}^{-1}\tilde{U}_fR(s) \quad (16)$$

Based on the Bezout identity:

$$NM^{-1}(I+\tilde{V}^{-1}\tilde{U}_cNM^{-1})^{-1}\tilde{V}^{-1} = N(\tilde{V}M+\tilde{U}_cN)^{-1} = N \quad (17)$$

we have $Y(s) = NU_fR(s)$ and since $(I - \tilde{U}cN) = \tilde{V}M$ the error can be expressed as $E(s) = \tilde{V}M\tilde{U}_fR(s)$ and therefore the command as $U(s) = MU_fR(s)$.

Using again Figure 6, the Bezout identity and assuming that $R(s) = 0$ we have:

$$
\begin{aligned}
Y(s) &= NM^{-1}(I+\tilde{V}^{-1}\tilde{U}_cNM^{-1})^{-1}V(s) && (18)\\
&= N(M+\tilde{V}^{-1}\tilde{U}_cN)^{-1}V(s) && (19)\\
&= N(\tilde{V}M+\tilde{U}_cN)^{-1}\tilde{V}^{-1}V(s) && (20)\\
&= \tilde{N}\tilde{V}^{-1}V(s) && (21)
\end{aligned}
$$

Then applying the superposition principle:

$$
\begin{aligned}
Y(s) &= N\tilde{U}_fR(s) + N\tilde{V}^{-1}V(s)\\
U(s) &= M\tilde{U}_fR(s) + M\tilde{V}^{-1}V(s) && (22)
\end{aligned}
$$

The typical indirect self tuning takes place in two stages: i) plant NM^{-1} identification ii) obtaining a solution $\tilde{V}-1\tilde{U}_c$ (i.e. the controller) of the Bezout identity[3]. For the second step it is worth mentioning the exceptional ability of the indirect self tuner to use a wide variety of controller design methodologies (e.g. linear quadratic, minimum variance, predictive, frequency loop-shaping based etc.). These design methods combined with visibility of the model which are appealing features for the industrial control community.

[3]Remember that if such a factorization exists the controller $\tilde{V}-1\tilde{U}_c$ is stabilizing for the plant NM^{-1}

For direct self tuning regulators the starting point is again the Bezout identity. Post multiplying this identity with $Y(s)$ we have:

$$
\begin{aligned}
Y(s) &= (\tilde{V}M+\tilde{U}_cN)Y(s) && (23)\\
&= \tilde{V}MY(s)+\tilde{U}_cNY(s) && (24)
\end{aligned}
$$

and further using the plant dynamics written as $MY(s) = NU(s)$

$$Y(s) = \tilde{V}NU(s) + \tilde{U}_cNY(s) \quad (25)$$

or as model:

$$Y(s) = (I-\tilde{U}_cN)^{-1}\tilde{V}NU(s) \quad (26)$$

Equation (**??**)is in fact the process model parameterized in the controller coprime factors. Hence if the above model is identified the controller is obtained without design based on some identified model. The only problem is that Equation (**??**) is nonlinear in N, which makes the identification task and hence obtaining the direct self tuner more difficult.

An example of indirect self tuning - Model following adaptive control

One of the common appearances that indirect self tuning algorithms are taking is model following. The closed loop presented in Figure 6 together with the equation (22) can be linked to model following based on the assumption that $V(s) = MN^{-1}Ym(s)$. This model following scheme is presented in Figure **??**.

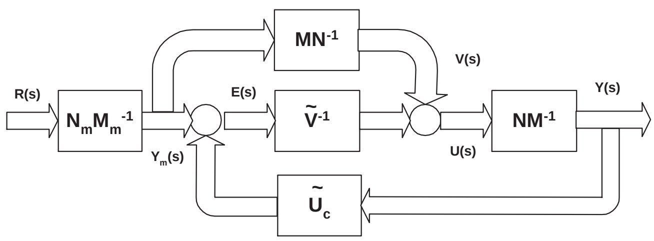

Figure 7: Coprime Factor Formulation of Self Tuners (Model Following Equivalent)

For this structure is easy to observe the requirement for an accurate plant model. This can be achieved through on-line identification. Note that the choice of the model to be followed has to account for plant limitations due to unstable zeros (which correspond to an inverse response), time delays and unstable poles. Moreover input constraints which can lead to the control of a highly nonlinear system need to be indirectly embedded in this model. Bad choices for the $N_mM_m^{-1}$ model will result in a closed loop system with poor sensitivity.

In essence model following means in general the adjustment of a constant feedback gain, assuming that this gain is used in the nominal system, so that the time varying system approximates the ideal, which is usually linear, in some sense. This control technique was employed by many researchers initially on aircraft models ranging from F-8 to F-16 and F-18. Most of the simulations performed were considered successful, providing an improved system response in the case when model parameters are varying.

Adaptive model following involves the redesign of the control law by a parameter estimation of the new system and a re-optimization. The resulting controller can be either a total new design or an augmented version of the initial controller, depending on the strategy.

Many schemes are employing the conventional controller in parallel with the adaptive one, each of them being used in different situations. For instance the nominal one is employed during the normal operations or when the adaptive controller exhibits failures, as opposed to the adaptive one which is introduced in the algorithm when parameters vary and hence adaptation is needed.

8 Nonlinear Adaptive Control

8.1 Nonlinear Laguerre modelling

The Laguerre methodology can be extended to nonlinear systems, following the work of Schetzen (1980). The nonlinear Laguerre model is a special case of a Wiener model, where the linear dynamic part represented by a series of Laguerre filters is followed by a memoryless nonlinear mapping. Such a nonlinear model can be derived from the Volterra series input-output representation, where the Volterra kernels are expanded via truncated Laguerre functions. A finite-time observable nonlinear system can be approximated as a truncated Wiener-Volterra series:

$$y(t) = h_0(t) + \sum_{n=1}^{N} \int \cdots \int h_n(\tau_1, \cdots, \tau_n) \prod_{i=1}^{n} u(t-\tau_i) d\tau_i$$

For instance, truncating the series after the second-order kernel:

$$\begin{aligned} y(t) &= h_0(t) + \int_0^\infty h_1(\tau_1) u(t-\tau_1) d\tau_1 + \qquad (27) \\ & \int_0^\infty \int_0^\infty h_2(\tau_1, \tau_2) u(t-\tau_1) u(t-\tau_2) d\tau_1 d\tau_2 \end{aligned}$$

Assuming that the Volterra kernels are in $L_2[,\infty)$, they can be expanded and approximated as:

$$h_1(\tau_1) = \sum_{k=1}^{N} c_k \phi_k(\tau_1) \qquad (28)$$

$$h_2(\tau_1, \tau_2) = \sum_{n=1}^{N} \sum_{m=1}^{N} c_{nm} \phi_n(\tau_1) \phi_n(\tau_2) \qquad (29)$$

Using Laguerre functions, this second-order nonlinear system can be expressed as the nonlinear state-space model:

$$\dot{l}(t) = Al(t) + bu(t) \qquad (30)$$

$$y(t) = c_0 + c^T l(t) + l^T(t) D l(t) \qquad (31)$$

where $c = \{c_k\}$ and $D = \{c_{nm}\}$. Note that since the Volterra kernels are symmetric, $c_{nm} = c_{mn}$ and thus D is symmetric. A discrete model can be derived in a similar form. Note that this model is linear in the parameters, and can thus be easily identified.

8.1.1 Feedback linearization:

There is much current interest in the adaptive control of nonlinear systems. This is a difficult problem since no general methods are available. The idea of obtaining a modified plant which exhibits a linear characteristics and use it to derive a controller is probably the most natural approach to nonlinear control. The method is called feedback linearization and can be extended to adaptive control via tuning on-line some of its parameters. For a clear understanding we are presenting feedback linearization applied to a simple SISO system with two states:

$$\dot{x}_1(t) = x_2(t) + f(x_1(t)) \qquad (32)$$

$$\dot{x}_2(t) = u(t) \qquad (33)$$

where $f(\dot{)}$ is the output nonlinearity as a differentiable function.

Introducing new coordinates $\zeta_1(t) = x_1(t)$ and $\zeta_2(t) = x_2(t) + f(x_1(t))$ we rewrite the above system as:

$$\dot{\zeta}_1(t) = \zeta_2(t) \qquad (34)$$

$$\dot{\zeta}_2(t) = \zeta_2(t)\dot{f}(\zeta_1(t)) + u(t) \qquad (35)$$

Using the control law

$$u(t) = -a_2\zeta_1(t) - a_1\zeta_2(t) - \zeta_2(t)\dot{f}(\zeta_1(t)) + v(t) \qquad (36)$$

we get a linear closed loop described by:

$$\dot{\zeta}(t) = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \zeta(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t) \qquad (37)$$

The transformation that links the original input to the closed loop input is:

$$u(t) = -a_2 x_1(t) - (a_1 + \dot{f}(x_1(t)))(x_2(t) + f(x_1(t))) + v(t) \qquad (38)$$

Moving in the direction of adaptive control we rewrite the original system based on the unknown parameter θ as:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) + \theta f(x_1(t)) \quad (39)\\ \dot{x}_2(t) &= u(t) \quad (40) \end{aligned}$$

Applying the well known certainty equivalence principle we have the following mapping between the original input and the closed loop input:

$$u(t) = -a_2 x_1(t) - (a_1 + \dot{f}(x_1(t)))(x_2(t) + \hat{\theta} f(x_1(t))) + v(t) \quad (41)$$

which unfortunately gives an error equation nonlinear in the parameter θ. The solution is to use the modified coordinates $\zeta_1(t) = x_1(t)$ and $\zeta_2(t) = x_2(t) + \hat{\theta} f(x_1(t))$, where $\hat{\theta}$ is an estimation of θ. The system above becomes:

$$\begin{aligned} \dot{\zeta}_1(t) &= \dot{x}_1(t) = x2(t) + \theta f(x_1(t)) \quad (42)\\ &= \zeta_2(t) + (\theta - \hat{\theta}) f(\zeta_1(t)) \quad (43)\\ \dot{\zeta}_2(t) &= \dot{\hat{\theta}} f(x_1(t) + \hat{\theta}(x_2(t) + \theta f(x_1(t))) \dot{f}(x_1(t)) + u(t) \end{aligned}$$

The control law becomes:

$$\begin{aligned} u(t) &= -a_2 \zeta_1(t) - a_1 \zeta_2(t) - \hat{\theta}(x_2 + \hat{\theta} f(x_1(t))) \dot{f}(x_1(t)) - \\ &\quad - f(x_1(t)) \dot{\hat{\theta}} + v(t) \quad (44) \end{aligned}$$

which results in the following linear closed loop:

$$\dot{\zeta}(t) = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \zeta(t) + \begin{bmatrix} f(\zeta_1(t) \\ \hat{\theta} f(\zeta_1(t)) \dot{f}(\zeta_1(t)) \end{bmatrix} \hat{\theta} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v(t) \quad (45)$$

A generalization of this approach to MIMO systems is given in [?].

8.1.2 Quasi-LPV adaptive control: This section suggests a new nonlinear adaptive strategy based on four ingredients:

- High fidelity models
- Nonlinear model approximation techniques
- Nonlinear identification
- Constrained Model Based Predictive Control

This approach is motivated by a number of problems encountered with other active approaches surveyed.

High fidelity dynamic models are increasingly built for complex plants. This has been the case in the aerospace and process control industries for many years. Most of the nonlinear models available require in general, extra tuning to reflect a specific process within required fidelity. Automatic tuning of these parameters can be obtained through nonlinear output error identification performed in recursive fashion.

Before embarking onto this path a nonlinear model approximation technique is required to produce a suitable model. In [?] a quasi-LPV model that embeds the plant nonlinearities without interpolating between point-wise (Jacobian) linearization is presented. The main characteristic of these models, compared with the usual LPV way of representing systems, is that the scheduling variable is a state of the model. The quasi-LPV approach is mostly suited for systems exhibiting output nonlinearities. Such nonlinearities enable us to write the system in form of equations (46). A principal requirement for a nonlinear system to be transformed into a quasi-LPV system is that the number of available equations has to be equal to the number of states plus the number of outputs minus the number of scheduling variables. When it is impossible to embed all the system nonlinearities in the output then the transformations used in producing the quasi-LPV model, see equation (48), have to be approximated up to first order terms in all the states except the scheduling parameters.

To develop the quasi-LPV model we start with a nonlinear model written in a form for which the nonlinearities depend only on the scheduling variable α:

$$\frac{d}{dt}\begin{bmatrix} \alpha \\ q \end{bmatrix} = f(\alpha) + \begin{bmatrix} A_{11}(\alpha) & A_{12}(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha) \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} B_{11}(\alpha) \\ B_{21}(\alpha) \end{bmatrix} \delta \quad (46)$$

A family of equilibrium states, parameterized by the scheduling variable α, is obtained by setting the state derivatives to zero:

$$0 = f(\alpha) + A(\alpha) \begin{bmatrix} \alpha \\ q_{eq}(\alpha) \end{bmatrix} + B(\alpha) \delta_{eq}(\alpha)$$

Providing that there exist continuously differentiable functions $q_{eq}(\alpha)$ and $\delta_{eq}(\alpha)$, we are able to write the system (46) in the following form [?]:

$$\begin{aligned} &\tfrac{d}{dt}\begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \end{bmatrix} = \\ &\begin{bmatrix} 0 & A_{12}(\alpha) \\ 0 & A_{22} - \frac{d}{d\alpha} q_{eq}(\alpha) A_{12}(\alpha) \end{bmatrix} \begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \end{bmatrix} + \quad (47)\\ &\begin{bmatrix} B_{11}(\alpha) \\ B_{21}(\alpha) - \frac{d}{d\alpha} q_{eq}(\alpha) B_{11}(\alpha) \end{bmatrix} (\delta - \delta_{eq}(\alpha)) \end{aligned}$$

This form gives a different α-dependent family than would be obtained by point-wise linearization. In order to use (47), the function $\delta_{eq}(\alpha)$ must be known. This can be estimated by using an 'inner loop' [?] but, because of model uncertainty, this can reduce the robustness of the main control loop in a way which is difficult to predict at the design stage. Like in [?] we avoid the problem generated by the existence of an inner loop required to compute $\delta_{eq}(\alpha)$ by adding an integrator at the plant input. As a result we have the quasi-LPV form for the system dynamics:

$$\frac{d}{dt}\begin{bmatrix}\alpha \\ q-q_{eq}(\alpha) \\ \delta-\delta_{eq}(\alpha)\end{bmatrix} = \begin{bmatrix}0 & A_{12}(\alpha) & B_{11}(\alpha) \\ 0 & A_{22}-\frac{d}{d\alpha}q_{eq}(\alpha)A_{12}(\alpha) & B_{21}(\alpha)-\frac{d}{d\alpha}[q_{eq}(\alpha)]B_{11}(\alpha) \\ 0 & -\frac{d}{d\alpha}[\delta_{eq}(\alpha)]A_{12}(\alpha) & -\frac{d}{d\alpha}\delta_{eq}(\alpha)B_{11}(\alpha)\end{bmatrix} \times \begin{bmatrix}\alpha \\ q-q_{eq}(\alpha) \\ \delta-\delta_{eq}(\alpha)\end{bmatrix} + \begin{bmatrix}0\\0\\1\end{bmatrix}\nu \tag{48}$$

This final form is actually the representation used for identification and further control purposes. The quasi-LPV model of the plant gives the clarity required by industry since it retains a physical meaning for the elements of the model LTV matrices. It is advisable to have the scheduling variable as a system output rather than an estimate.

The matrices $A(\alpha)$ and $B(\alpha)$ depend in a nonlinear fashion on a number of model parameters. The nonlinear output error identification method is employed to produce representations for the model LTV matrices, hence the quasi-LPV model runs in parallel with the nonlinear plant providing the state equilibrium values $q_{eq}(\alpha)$, $\delta_{eq}(\alpha)$ together with the $A(\alpha)$, $B(\alpha)$, C, D matrices depending on the current value of the scheduling parameter α.

Freezing the high fidelity model with respect to the parameter vector yields a linear model which is used by the controller as its internal model. An important advantage for real-time implementation is that this is a computationally inexpensive way of obtaining, in adaptive fashion, a linear internal model which approximates the nonlinear one.

We envisage a constrained model based predictive controller providing the inputs to the plant. Giving it enough degrees of freedom (a large enough set of control inputs) enables it to keep the plant close to the required trajectory. The structure of the controller consists of standard modules. The states, including α, are assumed measurable. If α remains constant, the $MBPC$ controller stabilizing this model will drive the states to zero which means that the δ input to the plant will be set at the true δ_{eq} trim value. The internal model will be the corresponding LTI system obtained at each value of the scheduling parameter α.

The main question raised while implementing the adaptive mechanism was which model should be used to provide predictions. Two strategies have been considered for use during setpoint changes which cause significant α variations:

- No *a priori* trajectory information. A single model was used across the whole prediction horizon, but changed at each current time step in accordance with the measured α value.

- *A priori* trajectory information available. This allows the internal model to vary over the prediction horizon, but one needs to predict α over the prediction horizon in order to do this. It is important to base this prediction on the desired trajectory, rather than the achieved one. This might look as a gross approximation but it retains the QP structure of the optimisation problem. Basing it on the achieved trajectory would lose even the convexity property, so should be avoided if at all possible.

Note that these two strategies become the same in the special case when both the control horizon and the prediction horizon are one. It is advisable to take one of the above suggested paths since otherwise, obtaining the models through the plant future dynamics, the resulting cost function is no longer quadratic in the command increment. This method avoids a nonlinear constrained optimization for which it is hard to guarantee a global solution.

9 Commercial Adaptive Controllers

9.1 BrainWave

9.2 Connoisseur

9.3

10 Conclusions

References

[1] Batch Control and Management. *Control Engineering*, (March 30):20–24, 1998.

[2] I. Akesson. Adaptive Automatic Control of Reaction Speed in Exothermic Batch Chemical Reactor. In *Proceedings of the Workshop*, pages 99–103. International Federation of Automatic Control, 1987.

[3] K.J. Astrom, C.C. Hang, and B.C. Lim. A New Smith Predictor for Controlling A Process with An Integrator and Long Dead-time. *IEEE Transactions on Automatic Control*, 39:343–354, 1994.

[4] K.J. Astrom and B. Wittenmark. *Adaptive Control.* Addison Wesley, second edition, 1995.

[5] R. Berber. Control of Batch Reactors: A Review. *Transactions Institute of Chemical Engineering*, (74):3–20, 1996.

[6] M. Cabassud, M.V. Le Lann, A. Chamayou, and G. Casamatta. *Modelling and Adaptive Control of a Batch Reactor*, volume 116. Dechema Monographs, 1989.

[7] C.T. Chen and S.T. Peng. A Simple Adaptive Control Strategy for Temperature Trajectory Tracking in Batch Processes. *The Canadian Journal of Chemical Engineering*, 76:1118–1126, 1998.

[8] D. Clarke. *Advances in Model-Based Predictive Control*, pages 3–21. Oxford University Press, 1993.

[9] D.W Clarke and C. Mohtadi. Properties of Generalized Predictive Control. *Automatica*, 25(6):859–875, 6 1989.

[10] J.A. de Dona, G.C. Goodwin, and M.M. Seron. Connections between Model Predicitve COntrol and Anti-Windup Strategies for Dealing with Saturating Actuators. In *Proceedings of the* 5^{th} *Europeean Control Conference, Karlsruhe, Germany*, 1999.

[11] R. C. Dorf and R. H. Bishop. *Robust Process Control*. Addison Wesley, eight edition, 1999.

[12] G.A. Dumont and C.C. Zervos. Adaptive Controllers based on Orthonormal Series Representation. In *Proceedings of the* 2^{nd} *IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, Sweden*, 1986.

[13] A.M. Fileti and J.A. Pereira. Adaptive and Predictive Control Strategies for Batch Distillation. *Computers Chemical Engineering*, 21:1227–1236, 1997.

[14] C.E. Garcia. Quadratic Dynamic Matrix Control of Nonlinear Processes - An Application to a Batch Reaction Process. In *National Meeting*. AIChE, 1984.

[15] G.C.Goodwin and K.Sin. *Adaptive Filtering, Prediction and Control*. Enlewoods Cliffs NJ, Prentice Hall, 1984.

[16] P. Jarupintusophon, M.V. Lelann, M. Cabassud, and G. Casamatta. Realistic Model-Based Predictive and Adaptive-Control of Batch Reactors. *Computers Chemical Engineering*, 18:445–461, 1994.

[17] A. Jutan and A. Uppal. Combined Feedforward-Feedback Servo Control Scheme for an Exothermic Batch Reactor. *Industrial Engineering Chemical Process Design and Development*, 23:597–602, 1984.

[18] J.W.Head. Approxiamtion to transients by means of Laguerre Series. *Proceedings of the Cambridge Philosophical Society*, 52:640–651, 1956.

[19] V. Koncar, M.A. Koubaa, X. Legrand, P. Bruniaux, and C. Vasseur. Multirate Predictive Control with Nonlinear Optimizations Application to a Thermal Process for Batch Dyeing. *Textile Research Journal*, 67(11):788–792, 1997.

[20] A. Krishnan and K.A. Kosanovich. Multiple model-based controller design applied to an electrochemical batch reactor. *Journal of Applied Electrochemistry*, 27:774–783, 1997.

[21] K.S. Lee, I.S¿ Chin, H.J. Lee, and J.H. Lee. Model Predictive Control Technique Combined with Iterative Learning for Batch Processes. *AIChE Journal*, 45(10):2175–2187, 2000.

[22] Y.W. Lee. *Statistical Theory of Communication*. John Wiley and Sons, New York, first edition, 1960.

[23] G. Marroquin and W.L. Luyben. Practical Control Studies of Batch Reactor Using Realistic Mathematical Model. *Chemical Engineering Science*, pages 993–1003, 1973.

[24] M. Morari and E. Zafiriou. *Robust Process Control*. Prentice-Hall: Englewood, NJ, first edition, 1989.

[25] A.S. Morse. Supervisory control of families of linear set point controllers. In *Proceedings of the 32nd Conference on Decision and Control*, San Antonio, Texas, Dec 1993. IEEE.

[26] Z. Nagy and S. Agachi. Model Predictive Control of a PVC Batch Reactor. *Computers Chemical Engineering*, 21(6):571–581, 1997.

[27] K.S. Narendra and J. Balakrishnan. Adaptive Control Using Multiple Models. *IEEE Transactions on Automatic Control*, 42(2):171–187, 1997.

[28] J.A. Ramirez, R.M.Loperena, I. Cervantes, and A. Morales. A Novel Proportional-Integral-Derivative Control Configuration with Application to the Control of Batch Distillation. *Industrial Chemical Engineering Research*, 39:432–404, 2000.

[29] M.E. Salgado, G.C. Goodwin, and R.H. Middleton. Exponential Forgetting and Resetting. *International Journal of Control*, 47(2):477–485, 1988.

[30] K.J. Åström. Adaptive Control Around 1960. *IEEE Control Systems Magazine*, 16(3):44–49, 1996.

[31] T. Takamatsu, S. Shioya, and Y. Okada. Adaptive Internal Model Control and Its Applications to Batch Polymerization Reactor. In *Proceedings of the Workshop*, pages 109–114. International Federation of Automatic Control, 1986.

[32] C.C. Zervos. *Adaptive Control Based on Orthonormal Series Representation*. PhD thesis, University of British Columbia, 1988.