Concepts, Methods and Techniques in Adaptive Control

Guy A. Dumont and Mihai Huzmezan
Department of Electrical and Computer Engineering
University of British Columbia,Vancouver, B.C. Canada

IEEE ACC – Anchorage – May 8, 2002

Outline

1. Introduction
2. Development of Adaptive Control
3. Self-Tuning Control
4. Iterative Control
5. Multimodel Adaptive Control
6. Moving Towards The Multivariable Case
7. Nonlinear Adaptive Control
8. Available Commercial Products
9. Conclusions

Introduction

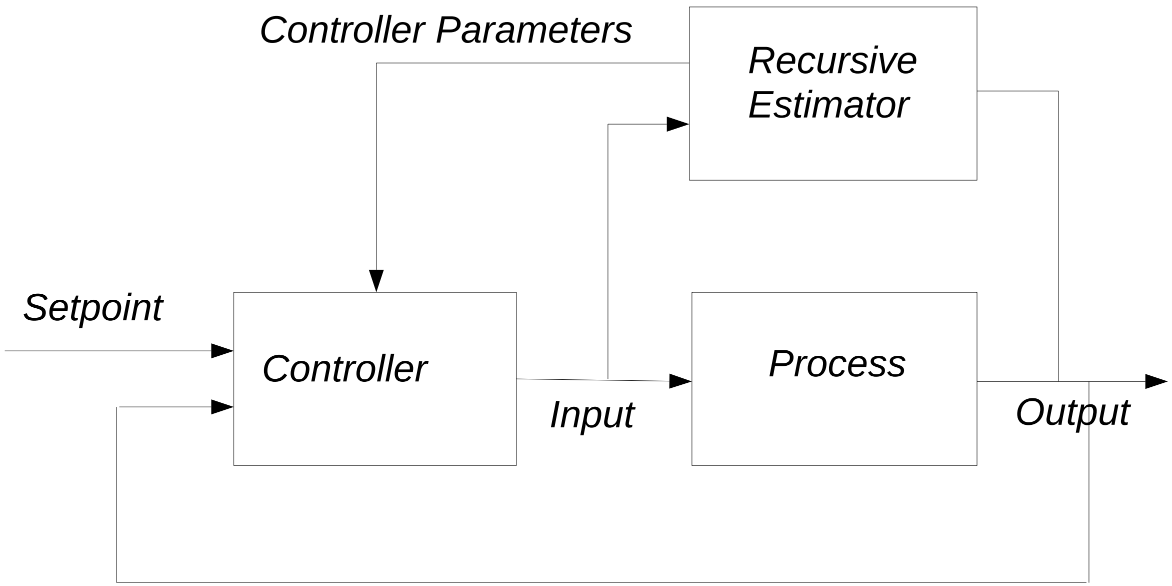
- To adapt:

  "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"

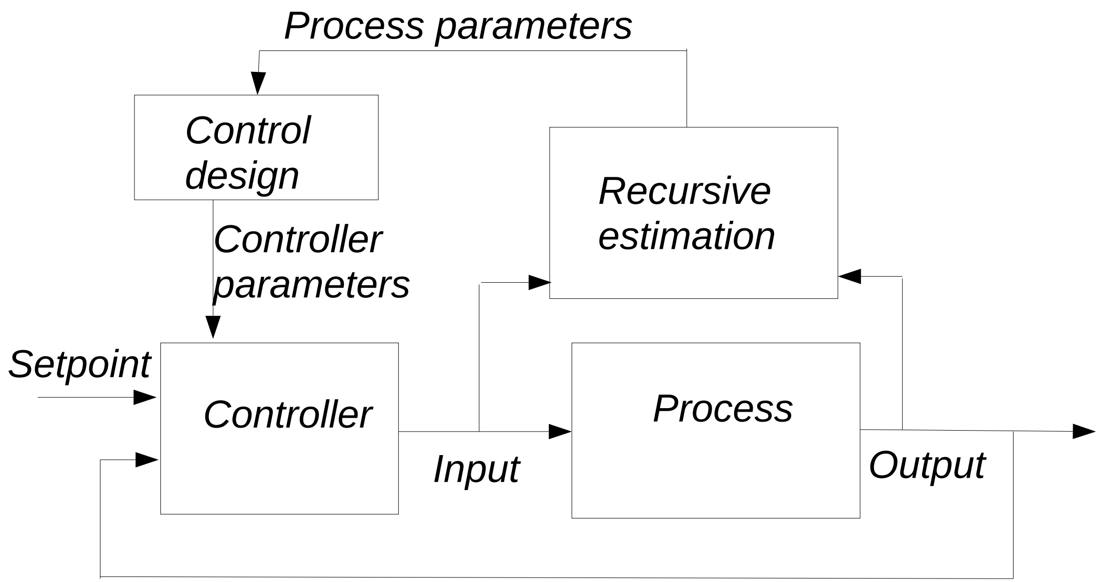
- When is a controller adaptive?

  "a non-adaptive controller is based solely on a-priori information whereas an adaptive controller is based also on a posteriori information"

Direct self-tuning control



Indirect self-tuning control



Self-Tuning Control

- Model-based tuning consists of two operations:
  - Model building via identification
  - Controller design using the identified model
- Self-tuning control can be thought of as an automation of this procedure when these two operations are performed on-line

Recursive Identification

- 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 behavior 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

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

Recursive Identification

The corresponding recursive least-squares algorithm with forgetting is then

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

- In choosing λ one has to compromise between fast tracking and long-term quality of estimates
- In the absence of excitation, then $P(t+1) \approx P(t)/\lambda$ will lead to grow exponentially causing wild variations in parameter estimates
- Variable forgetting factor has been proposed using the prediction error $\varepsilon(t)$

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

Recursive Identification

- The exponential Forgetting and Resetting Algorithm (EFRA) allows tracking of time-varying parameters while guaranteeing boundedness of covariance matrix

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

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

- Then $\sigma_{min} I \leq P(t) \leq \sigma_{max} I \quad \forall t$ where

$$
\sigma_{min} \approx \frac{\beta}{\alpha - \eta} \qquad \sigma_{max} \approx \frac{\eta}{\gamma} + \frac{\beta}{\eta} \qquad \eta = \frac{1-\lambda}{\lambda}
$$

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

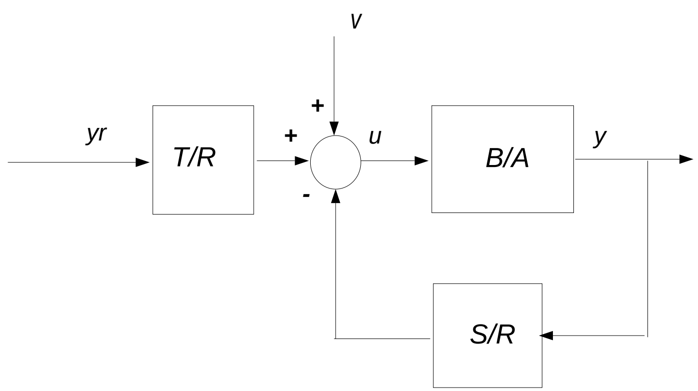
Identification in Closed Loop

- When the identification is done in closed loop, the identifiability of the process may be problematic.
- Two ways to guarantee closed-loop identifiability:
  1. The first one is ensure that a sufficiently exciting signal independent of y is injected into the loop, typically at setpoint or at the control input.
  2. The second way is to switch between different regulators, i.e. for a SISO loop it is sufficient to switch between two different controllers.
- The latter situation is actually very favorable for the adaptive control situation where the controller is constantly changing.
- There are also subtle interactions between identification and control in a closed-loop situation that affect the frequency distribution of the estimation variance and bias.
- It can be shown that when identification has to be performed under an output variance constraint, then a closed-loop experiment is optimal.

Pole Placement Design

- The two-degree-of-freedom controller is

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



- The desired response is

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

Pole Placement Design

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

$$
\begin{aligned}
y(t) &= \frac{BT}{AR+BS} y_r(t) + \frac{BR}{AR+BS} v(t) \\
u(t) &= \frac{AT}{AR+BS} y_r(t) + \frac{BS}{AR+BS} v(t)
\end{aligned}
$$

The closed-loop characteristic polynomial is thus

$$
\boxed{A_c = AR + BS}
$$

- This equation is known as the Diophantine equation, the Bezout identity
- It plays a central role in many control aspects of modern control theory

Pole Placement Design

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
- Desirable to cancel only stable and well damped zeros. Write

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

  - B^+ contains well damped stable zeros that can be cancelled
  - 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.

$$R = B^+ R'$$

Pole Placement Design

Then

$$\frac{B^+B^-T}{AB^+R' + B^+B^-S} = \frac{B^-B'_mA_0}{A_mA_0}$$

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

$$\boxed{AR' + B^-S = A_0A_m}$$

and

$$\boxed{T = B'_mA_0}$$

Pole Placement Design

- The Diophantine equation has no solution if A and B^- have common factors

- When A and B^- are coprime, 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

Indirect Self-Tuning Pole Placement

- Given A_0, A_m, B'_m and n, at each sampling time do the following:
  - Estimate $\hat{A}$ and $\hat{B}$
  - Perform pole-placement procedure as described previously)

- Note:
  - Pole placement often done in a determistic framework, i.e. little perturbation on the process. It is then important to ensure sufficient excitation by frequent setpoint changes
  - Need for finding common factors between $\hat{A}$ and $\hat{B}$
  - Need to factor $\hat{B}$ as $\hat{B} = \hat{B}^+ \hat{B}^-$

Direct Self-Tuning Pole Placement

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

Consider the Diophantine equation

$$A_0 A_m = A R' + B^- S$$

Multiply both sides by $y(t)$

$$A_0 A_m y(t) = R' A y(t) + B^- S y(t) = R' B u(t) + B^- S y(t)$$

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

$$A_0 A_m y(t) = B^- [R u(t) + S y(t)]$$

This could be considered a process model. Note however, that it is nonlinear in the parameters.

Youla-Kucera Parameterization

- The key feature is to use a parameterization that makes the closed-loop sensitivity functions **linear** in a design variable.
- This leads to new insights and to a control synthesis method
- The open-loop inversion provides an interesting framework for control design

Youla-Kucera Parameterization

- In open-loop control, let the input U be generated form the reference R by a transfer function Q, i.e.

$$U = QR$$

- This leads to an input-output transfer function from R ro Y as

$$T = GQ$$

- This shows that T is 1 only at the frequencies where Q inverts the plant model G.
- A key point is that $T = GQ$ is **linear** in Q
- In conventional feedback control with a controller C, the closed-loop complementary sensitivity is

$$T = \frac{GC}{1 + GC}$$

i.e. is **nonlinear** in C

Youla-Kucera Parameterization

- The above equation makes tuning $C(s)$ difficult
- Comparing the two equations, we see that if we write

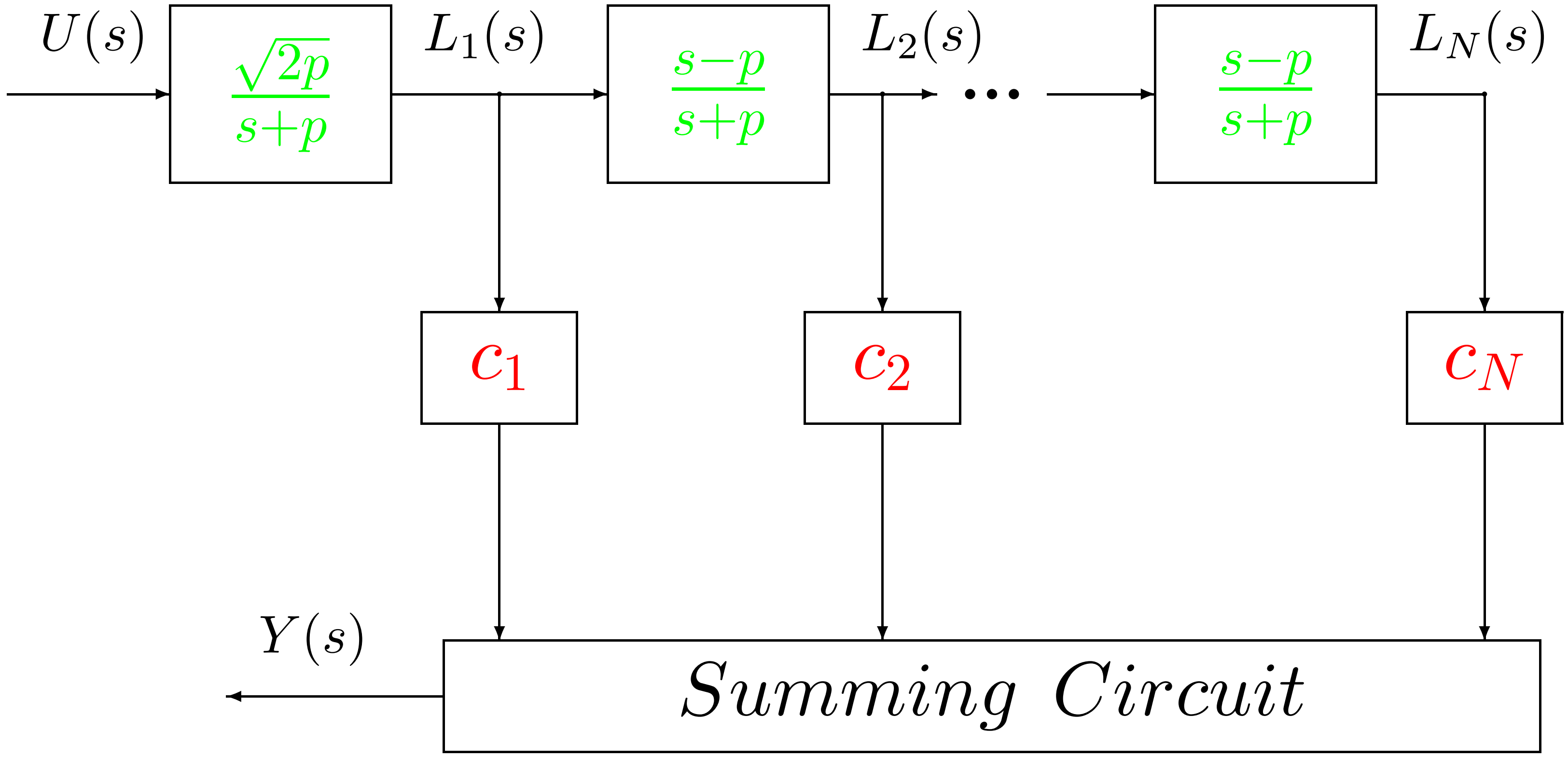
$$Q = \frac{C}{1 + GC}$$

then we can invert this relationship to find the controller $C(s)$

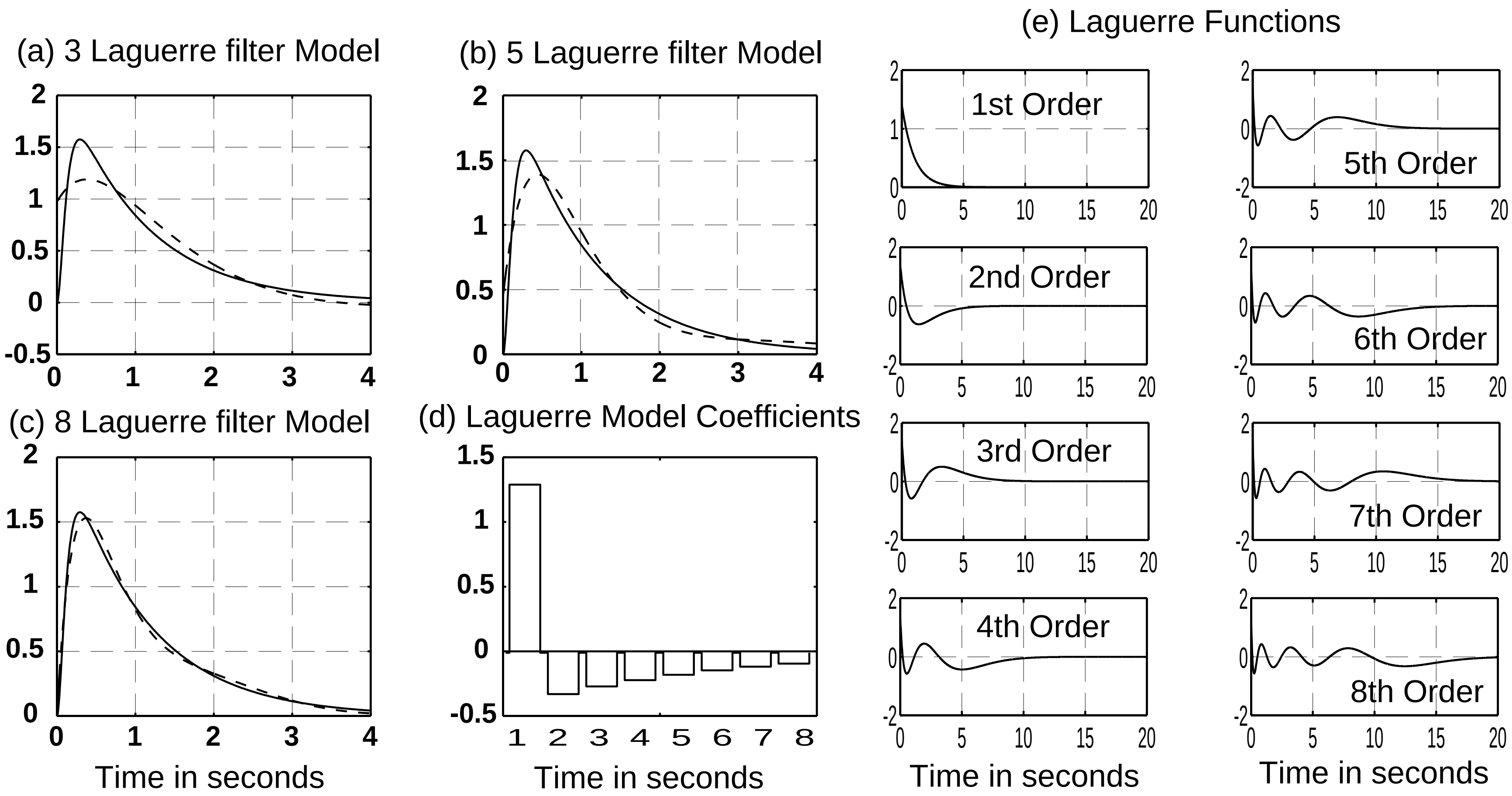
$$C = \frac{Q}{1 - QG}$$

- The procedure is then in two steps:
  1. Use Q as the design variable in $T = GQ$
  2. Use the above equation to determine the corresponding controller C
- Note that the relationship between C and Q is one-to-one.

Continuous Laguerre Functions



Example of Laguerre Modelling



Identification of Laguerre Models

Consider the real plant described by

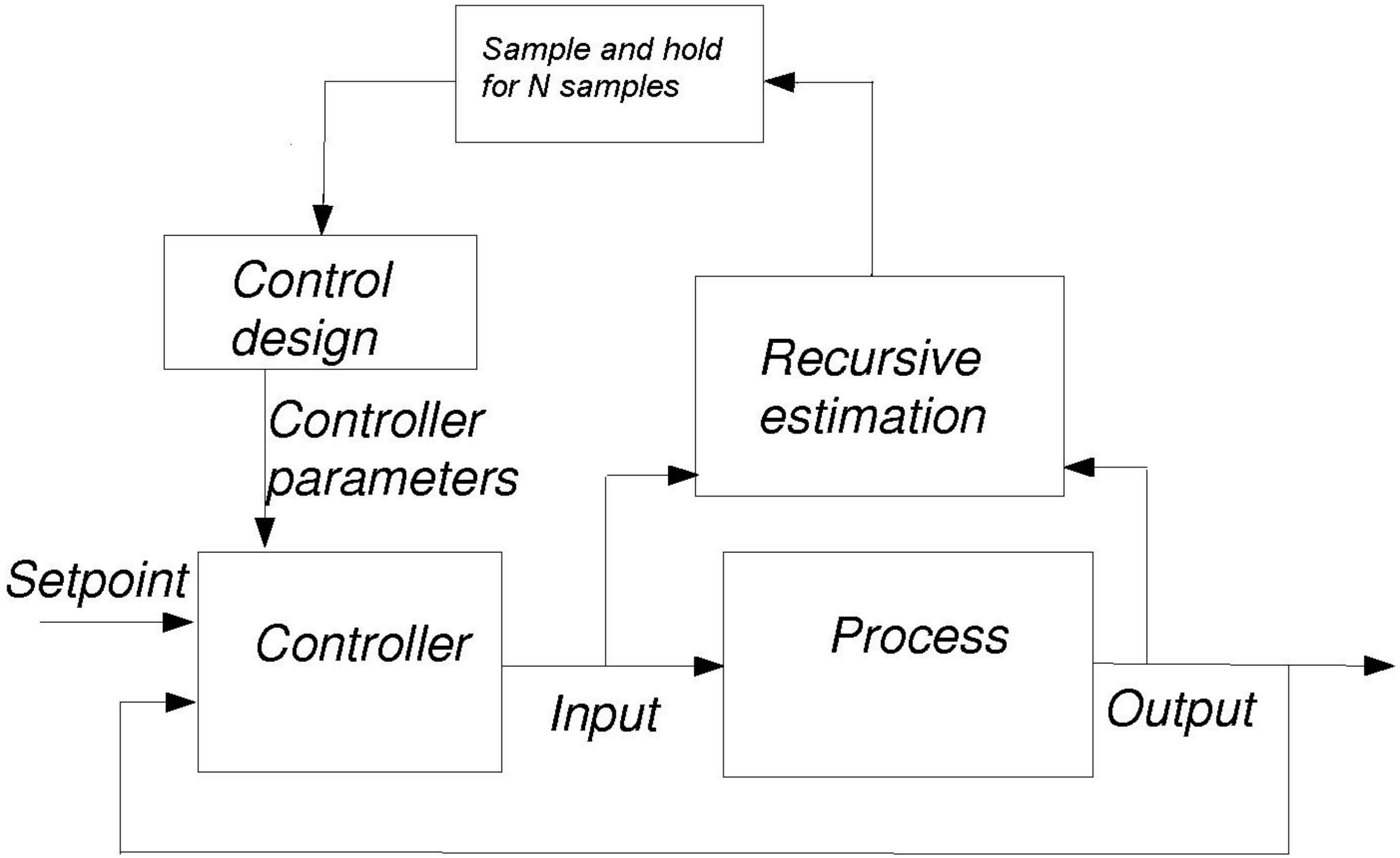
$$y(t) = \sum_{i=1}^{N} c_i L_i(q) + \sum_{i=N+1}^{\infty} c_i L_i(q) + w(t)$$

where $w(t)$ is a disturbance.
This model has an output-error structure, is linear in the parameters, and gives a convex identification problem.

- Even if $w(t)$ it coloured 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 $i = N + 1, \cdots, \infty$.

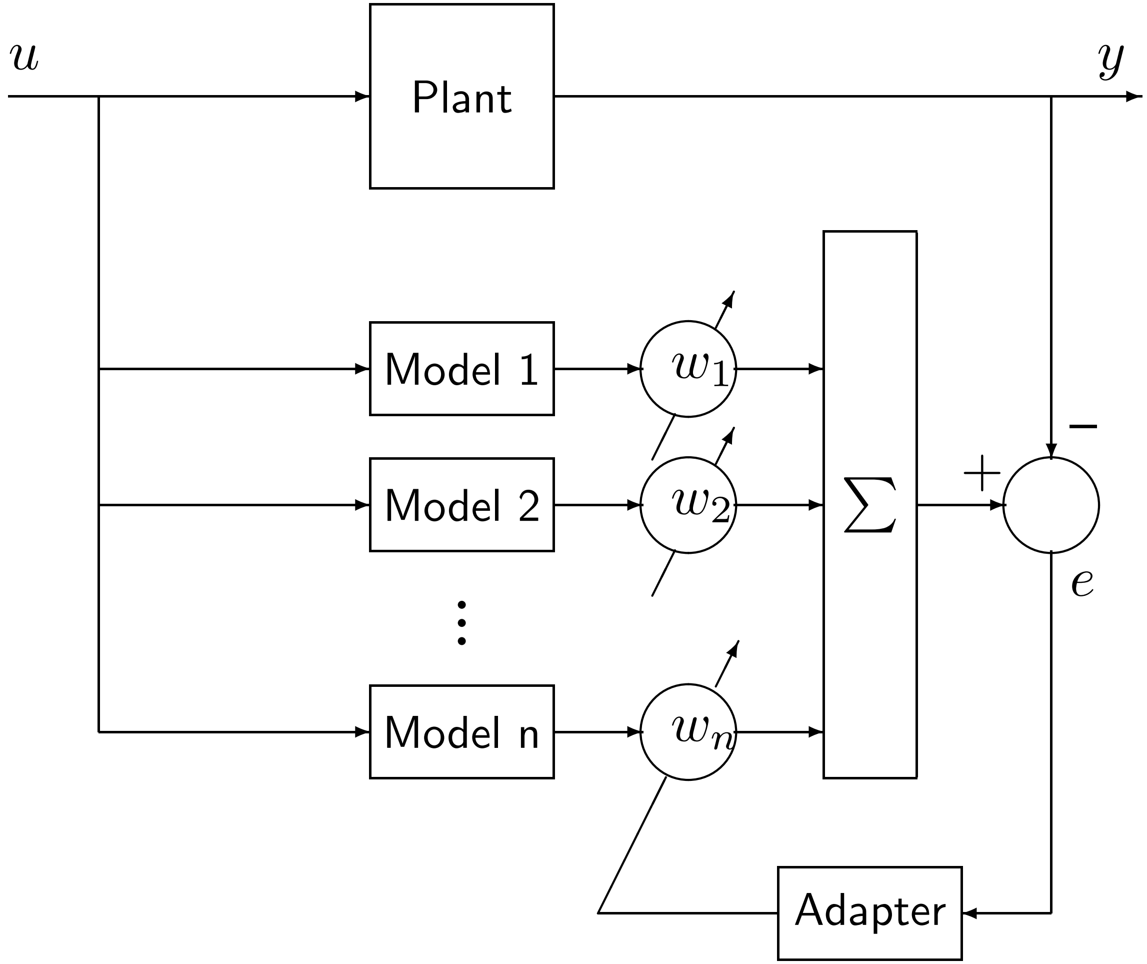
Iterative Control



Iterative Control

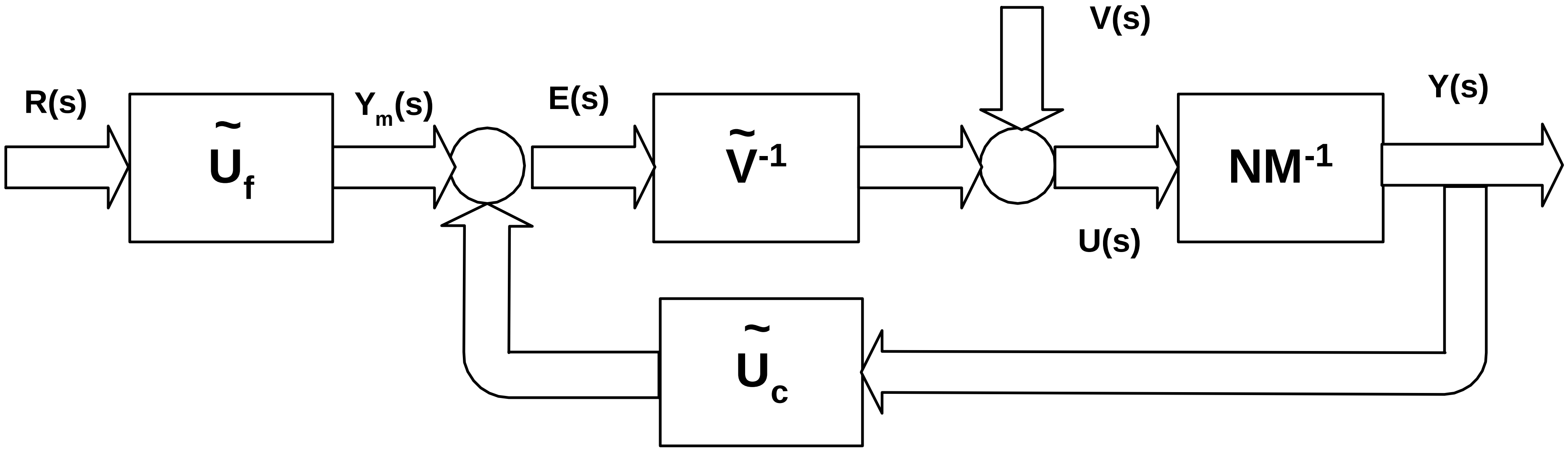
- Alternative to indirect adaptive control but with infrequent updating of control parameters
- The basic idea of iterative control is to observe the system under fixed feedback for a sufficient long period, after which identification and control re-design is performed.
- The re-designed controller is then implemented and the process repeated until a criterion indicative of satisfactory performance is met.
- Fundamental difference is that with adaptive control identifiability is guaranteed by use of a time-varying parameter, in iterative control an external perturbation must be sent to the plant to ensure persistent excitation.
- Identification criterion should be similar to the criterion use to assess the difference between desired and achieved performance.

Multimodel Adaptive Control



Multivariable Adaptive Control

- Using coprime factors and the Bezout Identity multivariable adaptive control is not hard to comprehend:



Coprime Factors Formulae For Adaptive Control

- Typical closed loop equations:

$$Y(s) = N(s)M^{-1}(s) \quad (1)$$

$$Y_m(s) = \tilde{U}_f(s)R(s) = N_m(s)M_m^{-1}(s)R(s) \quad (2)$$

$$E(s) = Y_m(s) - \tilde{U}_c(s)Y(s) \quad (3)$$

$$U(s) = \tilde{V}^{-1}(s)E(s) + V(s) \quad (4)$$

- The controller: $C(s) = \tilde{V}^{-1}\tilde{U}_c$
- The Plant/Controller Bezout identity is: $\tilde{V}(s)M(s) + \tilde{U}_c(s)N(s) = I$.
- If such a factorization exists it is equivalent to $C(s)$ being a stabilizing controller for $P(s)$

Making Our Life Easy I

- Closed loop dependencies:

$$Y(s) = NM^{-1}(I + \tilde{V}^{-1}\tilde{U}_c NM^{-1})^{-1}\tilde{V}^{-1}\tilde{U}_f R(s) \tag{5}$$

- Based on the Bezout identity we can simplify:

$$NM^{-1}(I + \tilde{V}^{-1}\tilde{U}_c NM^{-1})^{-1}\tilde{V}^{-1} = N(\tilde{V}M + \tilde{U}_c N)^{-1} = N \tag{6}$$

- In conclusions: $Y(s) = NU_f R(s)$ and since $(I - \tilde{U}cN) = \tilde{V}M$ the error is $E(s) = \tilde{V}M\tilde{U}_f R(s)$ and the command can be computed as $U(s) = MU_f R(s)$.

Making Our Life Easy II

- Now assuming: $R(s) = 0$:

$$
\begin{aligned}
Y(s) &= NM^{-1}(I + \tilde{V}^{-1}\tilde{U}_c NM^{-1})^{-1}V(s) && (7)\\
&= N(M + \tilde{V}^{-1}\tilde{U}_c N)^{-1}V(s) && (8)\\
&= N(\tilde{V}M + \tilde{U}_c N)^{-1}\tilde{V}^{-1}V(s) && (9)\\
&= \tilde{N}\tilde{V}^{-1}V(s) && (10)
\end{aligned}
$$

- Applying the superposition principle:

$$
\begin{aligned}
Y(s) &= N\tilde{U}_f R(s) + N\tilde{V}^{-1}V(s)\\
U(s) &= M\tilde{U}_f R(s) + M\tilde{V}^{-1}V(s) && (11)
\end{aligned}
$$

Back to Adaptive Control:
The Multivariable Indirect Self Tuner

- The indirect self tuning controller takes involves two steps:
  1. plant NM^{-1} identification
  2. obtaining a solution $\tilde{V}^{-1}\tilde{U}_c$ of the Bezout identity

- To achieve the step 2 the indirect self tuner can use a wide variety of controller design methodologies (e.g. linear quadratic, minimum variance, predictive, frequency loop-shaping based etc.)

- Such design methods combined with visibility of the model are appealing features for the industrial control community.

Back to Adaptive Control: The Multivariable Direct Self Tuner

- For direct self tuners the starting point is again the Bezout identity.
- Post multiplying this identity with $Y(s)$ we obtain:

$$\begin{aligned} Y(s) &= (\tilde{V}M + \tilde{U}_c N)Y(s) && (12) \\ &= \tilde{V}MY(s) + \tilde{U}_c NY(s) && (13) \end{aligned}$$

- Using the plant dynamics written as: $MY(s) = NU(s)$ we obtain:

$$Y(s) = \tilde{V}NU(s) + \tilde{U}_c NY(s) \quad (14)$$

or:

$$Y(s) = (I - \tilde{U}_c N)^{-1}\tilde{V}NU(s) \quad (15)$$

Remarks On:
The Multivariable Direct Self Tuner

- Equation $Y(s) = \tilde{V}NU(s) + \tilde{U}_c NY(s)$ is in fact the process model parameterized in the controller coprime factors.

- If the above model is identified the controller is obtained without design based on some identified model.

- The bottleneck is the nonlinearity in N, which makes the identification task and the design of the direct self tuner intractable at times.

Feedback Linearization Plays An Essential Role In Nonlinear Adaptive Control

- Adaptive control of nonlinear systems is a difficult topic.
- No general nonlinear control methods are available.
- The idea of obtaining a modified plant which exhibits a linear characteristics and use it to derive a controller is a natural approach to nonlinear control.
- This method can be extended to adaptive control via tuning on-line some of its parameters.
- **Our vision is to use:**
  1. High fidelity models
  2. Nonlinear model approximation techniques
  3. Nonlinear grey box identification
  4. Constrained Model Based Predictive Control

  to achieve this goal.

High Fidelity Models Are An Essential Ingredient

Assumptions:

- 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 with required fidelity.
- Generally the tuning of these parameters can be obtained through nonlinear output error identification performed in a recursive fashion.
- Shamma et. al. pioneered a nonlinear model approximation technique that embeds the plant nonlinearities without interpolating between point-wise linearization.
- Quasi-LPV models require the scheduling variable to be a state of the model.
- This approach is mostly suited for systems exhibiting state nonlinearities.

Developing the Quasi-LPV Model (I)

We start with a nonlinear model written in a form for which the nonlinearities depend only on the scheduling variable α:

$$\frac{d}{dt}\begin{bmatrix}\alpha \\ q\end{bmatrix} = f(\alpha) + \begin{bmatrix}A_{11}(\alpha) & A_{12}(\alpha) \\ A_{21}(\alpha) & A_{22}(\alpha)\end{bmatrix}\begin{bmatrix}\alpha \\ q\end{bmatrix} + \begin{bmatrix}B_{11}(\alpha) \\ B_{21}(\alpha)\end{bmatrix}\delta \tag{16}$$

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

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

Developing the Quasi-LPV Model (II)

Providing that there exist continuously differentiable functions $q_{eq}(\alpha)$ and $\delta_{eq}(\alpha)$ we write the system in the following form:

$$\frac{d}{dt}\begin{bmatrix}\alpha \\ q - q_{eq}(\alpha)\end{bmatrix} = \begin{bmatrix}0 & A_{12}(\alpha) \\ 0 & A_{22} - \frac{d}{d\alpha}q_{eq}(\alpha)A_{12}(\alpha)\end{bmatrix}\begin{bmatrix}\alpha \\ q - q_{eq}(\alpha)\end{bmatrix} + \begin{bmatrix}B_{11}(\alpha) \\ B_{21}(\alpha) - \frac{d}{d\alpha}q_{eq}(\alpha)B_{11}(\alpha)\end{bmatrix}(\delta - \delta_{eq}(\alpha)) \tag{17}$$

Developing the Quasi-LPV Model (III)

The problem generated by the existence of an inner loop required to compute $\delta_{eq}(\alpha)$ can be avoided by adding an integrator at the plant input:

$$\frac{d}{dt}\begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \\ \delta - \delta_{eq}(\alpha) \end{bmatrix} = \begin{bmatrix} 0 & A_{12}(\alpha) & B_{11}(\alpha) \\ 0 & A_{22} - \frac{d}{d\alpha}q_{eq}(\alpha)A_{12}(\alpha) & B_{21}(\alpha) - \frac{d}{d\alpha}[q_{eq}(\alpha)]B_{11}(\alpha) \\ 0 & -\frac{d}{d\alpha}[\delta_{eq}(\alpha)]A_{12}(\alpha) & -\frac{d}{d\alpha}\delta_{eq}(\alpha)B_{11}(\alpha) \end{bmatrix} \times \begin{bmatrix} \alpha \\ q - q_{eq}(\alpha) \\ \delta - \delta_{eq}(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \nu \qquad (18)$$

Essential remarks regarding this method

- This final form is actually the representation used for identification and control.
- 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
- 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.
- 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.

The controller

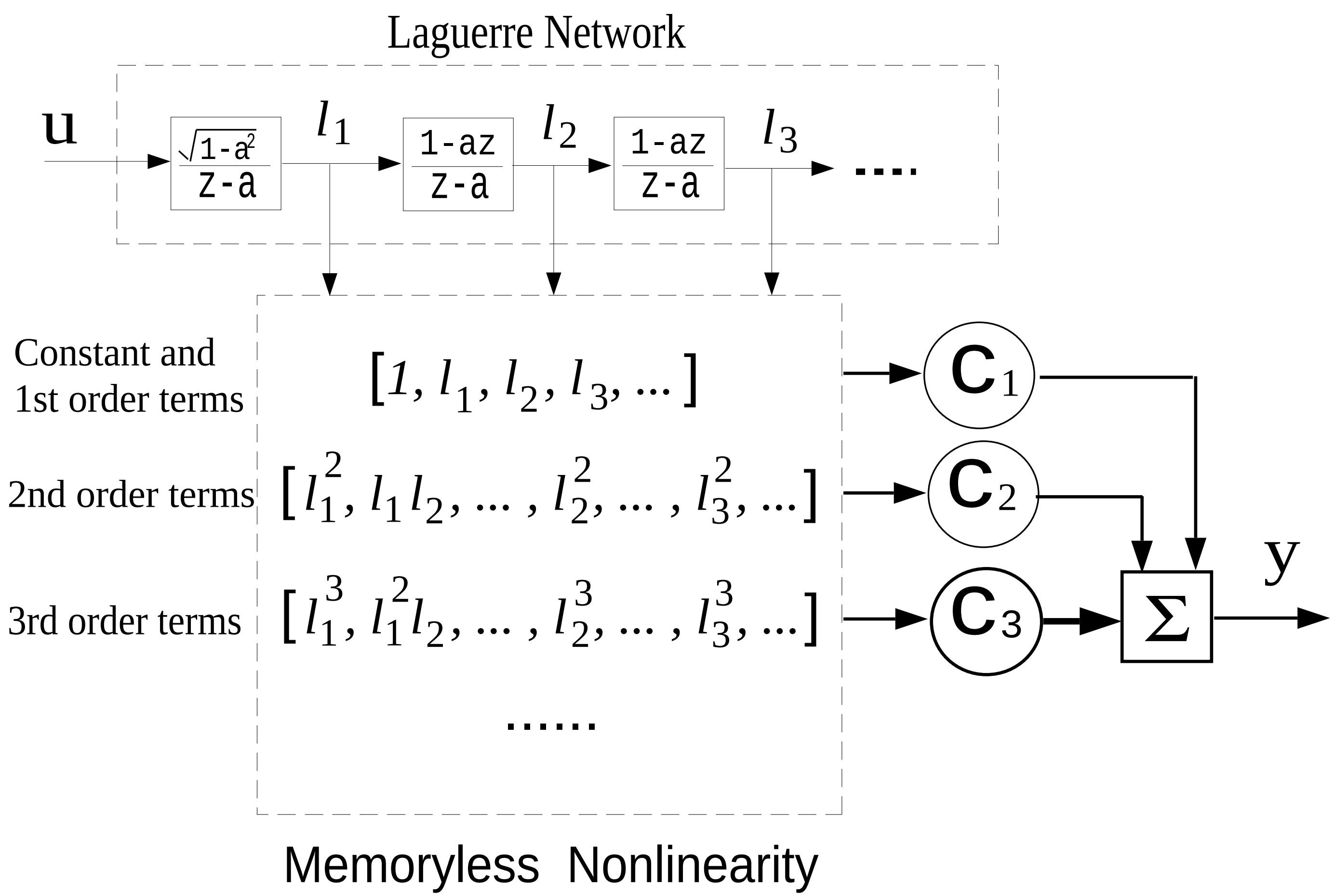
- Freezing the high fidelity model with respect to the parameter vector yields a linear model which is used by the controller as its internal model.

- A constrained model based predictive controller can be providing the plant inputs.

- Two strategies have been considered for use during setpoint changes which cause significant α variations:

  1. No *a priori* trajectory information (a single model can be used across the whole prediction horizon, but changed at each current time step in accordance with the measured α value)
  2. *A priori* trajectory information (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.

Alternatives to Quasi-LPV Modelling and Greybox Identification: Laguerre Modelling



Conclusions

- Adaptive has a long history
- Rich collection of algorithms
- **With careful engineering, adaptive control can provide significant benefits**

Commercial Controllers Are Embracing More And More Adaptive Control Ideas

- BCI Autopilot www.bciautopilot.com
- Brainwave www.brainwave.com
- Connoiseur & Exact www.foxboro.com
- CyboCon www.cybocon.com
- Intune www.controlsoftinc.com
- Knowledgescape www.kscape.com
- QuickStudy www.adaptiveresources.com