Robust Adaptive Controller Design

- Digital RST Controller (Recall)
- Pole Placement Technique
  - Desired closed-loop poles (Recall)
  - Solving Diophantine Equation (Recall)
  - Model Reference Control (MRC)
  - Robust Pole Placement by Q-Parametrization
- Adaptive Control Systems
- Parameter Adaptation Algorithm (Recall)
- Direct and Indirect Adaptive Control
- Switching Adaptive Control
- Gain-Scheduling Control

Discrete-Time Models

We consider SISO-LTI discrete-time models of the form :

$$y(k) = -\sum_{i=1}^{n_A} a_i y(k-i) + \sum_{i=0}^{n_B} b_i u(k-i)$$

Define q^{-1} a backward shift operator such that $q^{-1}y(k) = y(k-1)$, then

$$A(q^{-1})y(k) = B(q^{-1})u(k)$$

or $\quad y(k) = G(q^{-1})u(k) \quad$ with $\quad G(q^{-1}) = \dfrac{B(q^{-1})}{A(q^{-1})} \quad$ where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{n_A} q^{-n_A} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_B} q^{-n_B} \end{aligned}$$

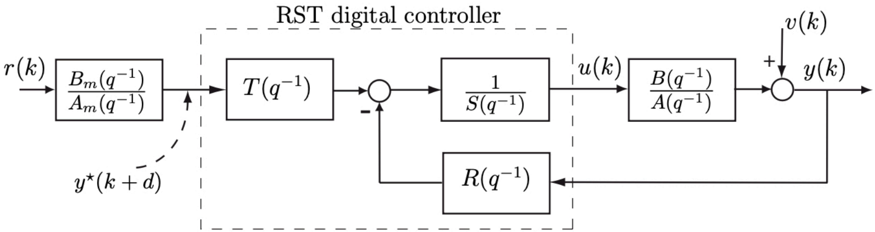
Definition (Delay d)

The number of first zero coefficients of $B(q^{-1})$ is called delay d. For sampled systems $d \geq 1$ (b_0 is always zero) and $B(q^{-1}) = q^{-d}B^\star(q^{-1})$ where $B^\star(q^{-1})$ has no leading zero coefficients.

RST Controller

A general form of a two-degree of freedom digital controller is given by :

$$R(q^{-1})y(k) + S(q^{-1})u(k) = T(q^{-1})y^{\star}(k+d)$$



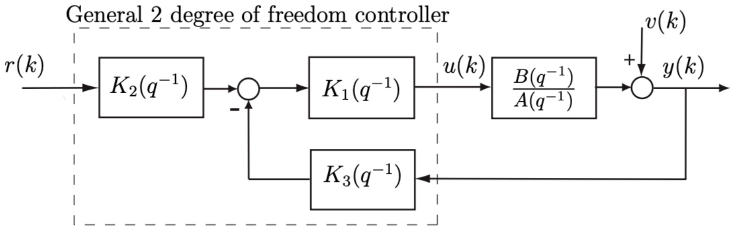
where $y^{\star}(k+d)$ is the *desired tracking trajectory* given with d steps in advance and

$$\begin{aligned} R(q^{-1}) &= r_0 + r_1 q^{-1} + \cdots + r_{n_R} q^{-n_R} \\ S(q^{-1}) &= 1 + s_1 q^{-1} + \cdots + s_{n_S} q^{-n_S} \\ T(q^{-1}) &= t_0 + t_1 q^{-1} + \cdots + t_{n_T} q^{-n_T} \end{aligned}$$

RST Controller

Advantages of RST controller :

- Can be easily implemented.
- It has two degrees of freedom (tracking and regulation dynamics can be designed independently).
- The other controller structures can be converted to an RST controller.



$$u(k) = K_1(q^{-1})\left[K_2(q^{-1})r(k) - K_3(q^{-1})y(k)\right] \quad \text{with} \quad K_i(q^{-1}) = \frac{N_i(q^{-1})}{D_i(q^{-1})}$$

is equivalent to

$$\begin{aligned} R(q^{-1}) &= N_1(q^{-1})N_3(q^{-1})D_2(q^{-1}) \\ S(q^{-1}) &= D_1(q^{-1})D_2(q^{-1})D_3(q^{-1}) \\ T(q^{-1}) &= N_1(q^{-1})N_2(q^{-1})D_3(q^{-1}) \end{aligned}$$

Pole Placement Technique

Objective :

Place the closed-loop poles on the desired places.

Closed-loop poles :

The roots of the characteristic polynomial $P(q^{-1})$ are the closed-loop poles.

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2} + \cdots$$

Desired closed-loop poles :

They should be chosen according to the desired performance.

Example (First-order polynomial)

Let $P(q^{-1}) = 1 + p_1 q^{-1}$. When $r(k) \equiv 0$, the free output response is defined by $y(k+1) = -p_1 y(k)$. Then $p_1 = -0.5$ leads to a relative decrease of 50% for the output amplitude at each sampling instant (choose p_1 between -0.2 and -0.8).

Pole Placement Technique

Example (Second-order polynomial)

Let $P(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2}$

1. Choose the time-domain performance (desired rise time, settling-time and overshoot for a step response).
2. Choose ζ (damping factor) and ω_n (natural frequency) of a second-order **continuous-time** model

   $$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

   that meets the time-domain performance.
3. Compute s_1 and s_2, the roots of $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$.
4. Compute p_1 and p_2 from :

   $$P(z^{-1}) = (z - e^{s_1 h})(z - e^{s_2 h}) = z^2 + p_1 z + p_2$$

Time-domain Performance

Overshoot : The overshoot M_p is a function of the damping factor :

$$M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}}$$

Settling-time : The time T_{set} for which the response remains within 2% of the final value :

$$e^{-\zeta\omega_n T_{set}} < 0.02 \qquad \text{or} \qquad \zeta\omega_n T_{set} \approx 4$$

Rise-time : The time it takes to rise from 10% to 90% of the final value. The following approximation can be used :

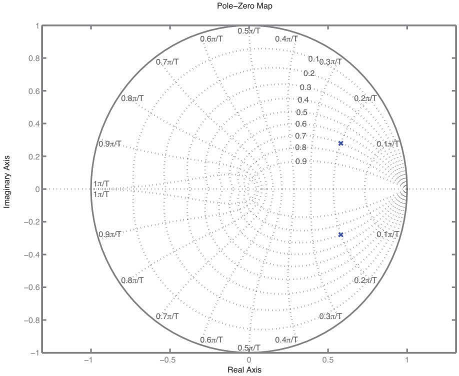
$$T_r \approx \frac{1.8}{\omega_n}$$

After computing ζ and ω_n, the desired $P(q^{-1})$ is computed by :

$$\begin{aligned} p_1 &= -2e^{-\zeta\omega_n h}\cos\left(\omega_n h\sqrt{1-\zeta^2}\right) \\ p_2 &= e^{-2\zeta\omega_n h} \end{aligned}$$

Time-domain Performance

Using `zgrid` of MATLAB :



Desired closed-loop poles with the loci for constant ζ and ω_n

The typical values for ζ and ω_n are :

$$\frac{0.25}{h} \leq \omega_n \leq \frac{1.5}{h} \quad ; \quad 0.7 \leq \zeta \leq 1$$

Time-domain Performance

Example

Compute the desired discrete-time closed-loop polynomial to have an overshoot of 10% and a settling time of $t_s = 1.2$ s. Suppose that the sampling period $h = 0.1s$.

1. For 10% overshoot we have : $e^{-\zeta\pi/\sqrt{1-\zeta^2}} = 0.1 \quad \Rightarrow \quad \zeta \approx 0.6$
2. The natural frequency is computed as $\omega_n \approx \frac{4}{\zeta T_{set}} = 5.55$.
3. The coefficients of the characteristic polynomial are :

$$\begin{aligned} p_1 &= -2e^{-\zeta\omega_n h}\cos\left(\omega_n h\sqrt{1-\zeta^2}\right) = -1.294 \\ p_2 &= e^{-2\zeta\omega_n h} = 0.513 \end{aligned}$$

that corresponds to the following desired poles :

$$z_{1,2} = 0.647 \pm j0.308$$

Dominant and Auxiliary Poles

The desired closed-loop polynomial can be divided into two polynomials defining the dominant and auxiliary closed-loop poles :

$$P(q^{-1}) = P_d(q^{-1})\,P_f(q^{-1})$$

Dominant closed-loop poles : Define the main dynamics of the closed-loop system in regulation and are computed based on the desired time-domain performance.

Auxiliary closed-loop poles : They introduce a filtering action in certain frequency regions in order to

- reduce the effect of the measurement noise ;
- smooth the variations of the control signal ;
- improve the robustness.

As a general rule, the "auxiliary poles" (called also the "observer poles"), are faster than the "dominant poles". It means that the roots of $P_f(q^{-1})$ should have a real part smaller than those of $P_d(q^{-1})$.

Regulation : Computation of R and S

Once $P(q^{-1})$ is specified, in order to compute

$$\begin{aligned} R(q^{-1}) &= r_0 + r_1 q^{-1} + \cdots + r_{n_R} q^{-n_R} \\ S(q^{-1}) &= 1 + s_1 q^{-1} + \cdots + s_{n_S} q^{-n_S} \end{aligned}$$

the following equation, known as "Bezout identity" (or Diophantine equation), must be solved :

$$A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1}) = P(q^{-1})$$

Theorem

The Diophantine equation has a unique solution with minimal degree for

$$\begin{aligned} n_R &= \textit{deg}\ R(q^{-1}) = n_A - 1 \\ n_S &= \textit{deg}\ S(q^{-1}) = n_B - 1 \\ n_P &= \textit{deg}\ P(q^{-1}) \leq n_A + n_B - 1 \end{aligned}$$

If and only if $A(q^{-1})$ and $B(q^{-1})$ are coprime.

Regulation : Computation of R and S

Question

$$\text{Given} \qquad G(q^{-1}) = \frac{0.2q^{-2}}{1 - 0.8q^{-1}}$$

Compute minimum order of $R(q^{-1})$ and $S(q^{-1})$:

(A) $n_R = 1, n_S = 1$ **(B)** $n_R = 0, n_S = 0$
(C) $n_R = 0, n_S = 1$ **(D)** $n_R = 0, n_S = 2$

Regulation : Computation of R and S

Question

$$\text{Given} \qquad G(q^{-1}) = \frac{0.2q^{-2}}{1-0.8q^{-1}}$$

Compute $R(q^{-1}) = r_0$ and $S(q^{-1}) = 1 + s_1 q^{-1}$ to place the closed loop poles at the roots of $P(q^{-1}) = 1 - 1.3q^{-1} + 0.5q^{-2}$.

(A)	$r_0 = 0.5, s_1 = -0.5$	**(B)**	$r_0 = -0.5, s_1 = -0.5$
(C)	$r_0 = -0.5, s_1 = 0.5$	**(D)**	$r_0 = -5.9, s_1 = -2.1$

Regulation : Computation of R and S

Example

Consider a discrete-time plant model given by :

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \qquad n_A = 2, \quad n_B = 2$$

Then $n_P \leq n_A + n_B - 1 = 3$, and $n_R = n_A - 1 = 1$, $n_S = n_B - 1 = 1$.
Let us take $n_P = 2$. Therefore, we should solve :

$$(1+a_1 q^{-1}+a_2 q^{-2})(1+s_1 q^{-1})+(b_1 q^{-1}+b_2 q^{-2})(r_0+r_1 q^{-1}) = 1+p_1 q^{-1}+p_2 q^{-2}$$

Then we have :

$$\begin{array}{l} a_1 + s_1 + b_1 r_0 = p_1 \\ a_2 + a_1 s_1 + b_2 r_0 + b_1 r_1 = p_2 \\ a_2 s_1 + b_2 r_1 = 0 \end{array} \quad \text{or} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & b_1 & 0 \\ a_2 & a_1 & b_2 & b_1 \\ 0 & a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} 1 \\ s_1 \\ r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} 1 \\ p_1 \\ p_2 \\ 0 \end{bmatrix}$$

Solving Diophantine Equation (Computing R and S)

A general solution to $A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1}) = P(q^{-1})$ is given by :

$$x = M^{-1}p$$

where $x^T = \begin{bmatrix} 1 & s_1 & \dots & s_{n_S} & r_0 & \dots & r_{n_R} \end{bmatrix}$ and

$$M = \left[\begin{array}{cccc|cccc} \multicolumn{4}{c}{\overbrace{\hphantom{1 \quad 0 \quad \cdots \quad 0}}^{n_B}} & \multicolumn{4}{c}{\overbrace{\hphantom{b_0 \quad 0 \quad \cdots \quad 0}}^{n_A}} \\ 1 & 0 & \cdots & 0 & b_0 & 0 & \cdots & 0 \\ a_1 & 1 & \ddots & \vdots & b_1 & b_0 & \ddots & \vdots \\ \vdots & a_1 & \ddots & 0 & \vdots & b_1 & \ddots & 0 \\ a_{n_A} & \vdots & \ddots & 1 & b_{n_B} & \vdots & \ddots & b_0 \\ 0 & a_{n_A} & \ddots & a_1 & 0 & b_{n_B} & \ddots & b_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n_A} & 0 & \cdots & 0 & b_{n_B} \end{array}\right]$$

M is called the Sylvester matrix and $p^T = \begin{bmatrix} 1 & p_1 & \dots & p_{n_P} & 0 & \dots & 0 \end{bmatrix}$.
Note that the inverse of M exists if and only if $A(q^{-1})$ and $B(q^{-1})$ are coprime polynomials (no simplifications between zeros and poles).

Regulation : Computation of R and S

Fixed terms in the regulator : The performance and robustness of the closed-loop system can be improved by introducing some fixed terms, $H_R(q^{-1})$ and $H_S(q^{-1})$, in the polynomial R and S as :

$$\begin{aligned} R(q^{-1}) &= H_R(q^{-1})R'(q^{-1}) \\ S(q^{-1}) &= H_S(q^{-1})S'(q^{-1}) \end{aligned}$$

Therefore, we need to solve the following equation :

$$A(q^{-1})H_S(q^{-1})S'(q^{-1}) + B(q^{-1})H_R(q^{-1})R'(q^{-1}) = P(q^{-1})$$

This can be done after replacing $A(q^{-1})H_S(q^{-1})$ by $A'(q^{-1})$ and $B(q^{-1})H_R(q^{-1})$ by $B'(q^{-1})$.

Then, for the minimal order solution we should have :

$$\begin{aligned} n_{R'} &= \deg\ R'(q^{-1}) = n_{A'} - 1 = n_A + n_{H_S} - 1 \\ n_{S'} &= \deg\ S'(q^{-1}) = n_{B'} - 1 = n_B + n_{H_R} - 1 \\ n_P &= \deg\ P(q^{-1}) \leq n_{A'} + n_{B'} - 1 = n_A + n_{H_S} + n_B + n_{H_R} - 1 \end{aligned}$$

Regulation : Choice of H_R and H_S

Choice of H_S

- Zero steady state error for a step disturbance :
  Integrator in the controller : $H_S = 1 - q^{-1}$
- Asymptotic rejection of a harmonic disturbance $v(k)$:

$$v(k) = \frac{1}{1 + \alpha q^{-1} + q^{-2}} \delta(k) \qquad \alpha = -2\cos(\omega h) = -2\cos(2\pi f h)$$

  Internal model principle : $H_S(q^{-1}) = 1 + \alpha q^{-1} + q^{-2}$

Choice of H_R

- opening the loop ($u = 0$) at a disturbance frequency f :

$$H_R(q^{-1}) = (1 + \alpha q^{-1} + q^{-2})$$

- Opening the loop at Nyquist frequency ($f = f_s/2 = 1/(2h)$) :

$$H_R(q^{-1}) = 1 + q^{-1}$$

Computation of T

The tracking performance is usually given by a tracking reference model :

$$H_m(q^{-1}) = \frac{B_m(q^{-1})}{A_m(q^{-1})}$$

The transfer function from the reference to the output is :

$$H_{cl}(q^{-1}) = \frac{B_m(q^{-1})T(q^{-1})B(q^{-1})}{P(q^{-1})A_m(q^{-1})}$$

Different dynamic for regulation and tracking :

In this case, $T(q^{-1}) = P(q^{-1})/B(1)$ cancels the regulation dynamic and make the steady-state gain of $H_{cl}(q^{-1})$ equal to 1. The tracking dynamic is imposed by the denominator of the reference model.

Same dynamic for regulation and tracking :

The reference model is chosen as $H_m(q^{-1}) = 1$ and $T(q^{-1})$, is chosen to have a steady-state gain of 1 for $H_{cl}(q^{-1})$. So we take : $T(q^{-1}) = P(1)/B(1)$. If the controller or the plant model has an integrator, i.e. $A(1)S(1) = 0$: Then $P(1) = A(1)S(1) + B(1)R(1) = B(1)R(1)$ and $T(q^{-1}) = R(1)$.

RST controller design : An Example

Example

Consider the following discrete-time second-order plant model :

$$G(q^{-1}) = \frac{0.1q^{-1} + 0.2q^{-2}}{1 - 1.3q^{-1} + 0.42q^{-2}}$$

The sampling period is $h = 1s$.

Design an RST controller such that :

- The tracking dynamics are close to the dynamics of a second-order continuous-time model with $\omega_n = 0.5$ rad/s and $\zeta = 0.9$.
- The regulation dynamics are close to that of a second-order continuous-time model with $\omega_n = 0.4$ rad/s and $\zeta = 0.9$.
- The steady state error for an output step disturbance is zero.

RST controller design : An Example

1 : With $\omega_n = 0.4$ rad/s and $\zeta = 0.9$, we obtain :

$$P(q^{-1}) = 1 - 1.3741q^{-1} + 0.4867q^{-2}$$

2 : Zero steady state error is obtained by $H_S(q^{-1}) = 1 - q^{-1}$.

3 : The following Bezout equation should be solved :

$$A(q^{-1})H_S(q^{-1})S'(q^{-1}) + B(q^{-1})R(q^{-1}) = P(q^{-1})$$

We have $n_{S'} = n_B - 1 = 1$ and $n_R = n_A + n_{H_S} - 1 = 2$ and

$$A'(q^{-1}) = A(q^{-1})(1 - q^{-1}) = 1 - 2.3q^{-1} + 1.72q^{-2} - 0.42q^{-3}$$

Therefore the Bezout equation in the matrix form becomes :

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -2.3 & 1 & 0.1 & 0 & 0 \\ 1.72 & -2.3 & 0.2 & 0.1 & 0 \\ -0.42 & 1.72 & 0 & 0.2 & 0.1 \\ 0 & -0.42 & 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 1 \\ s'_0 \\ r_0 \\ r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1.3741 \\ 0.4867 \\ 0 \\ 0 \end{bmatrix}$$

RST controller design : An Example

Solving the Bezout equation leads to

$$\begin{aligned} R(q^{-1}) &= 3 - 3.94q^{-1} + 1.3141q^{-2} \\ S(q^{-1}) &= (1 + s_1' q^{-1})(1 - q^{-1}) = 1 - 0.3742q^{-1} - 0.6258q^{-2} \end{aligned}$$

4 : The reference model $H_m(q^{-1})$ is computed by discretization of a second-order model with $\omega_n = 0.5$ rad/s and $\zeta = 0.9$:

$$H_m(q^{-1}) = \frac{0.0927q^{-1} + 0.0687q^{-2}}{1 - 1.2451q^{-1} + 0.4066q^{-2}}$$

Finally, the polynomial $T(q^{-1})$ is computed as :

$$T(q^{-1}) = \frac{P(q^{-1})}{B(1)} = 3.333 - 4.5806q^{-1} + 1.6225q^{-2}$$

If we wish to have the same dynamics for tracking and regulation, then $H_m(q^{-1}) = 1$ and $T(q^{-1}) = R(1) = 0.3741$.

Model Reference Control (MRC)

In this approach the zeros of the plant model in

$$H_{cl}(q^{-1}) = \frac{B_m(q^{-1})}{A_m(q^{-1})} \quad \frac{q^{-d}B^{\star}(q^{-1})T(q^{-1})}{A(q^{-1})S(q^{-1}) + q^{-d}B^{\star}(q^{-1})R(q^{-1})}$$

are cancelled by the closed-loop poles :

$$A(q^{-1})S(q^{-1}) + q^{-d}B^{\star}(q^{-1})R(q^{-1}) = B^{\star}(q^{-1})P(q^{-1})$$

This can be done if

- the zeros of $B^{\star}(q^{-1})$ are stable,
- complex zeros have a sufficiently high damping factor ($\zeta > 0.2$).

Remark

In discrete-time systems, unstable zeros can be the consequence of too fast sampling or a large fractional delay. This can be avoided by re-identification of a model with augmented delay or resampling.

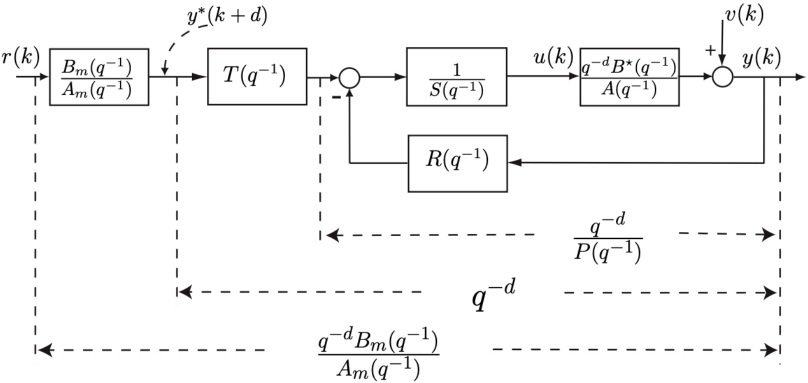
Model Reference Control (MRC)

In order to have a solution to Diophantine Equation we should have $S(q^{-1}) = B^{\star}(q^{-1})S'(q^{-1})$ and solve

$$A(q^{-1})S'(q^{-1}) + q^{-d}R(q^{-1}) = P(q^{-1})$$

with $\quad n_P \leq n_A + d - 1 \quad , \quad n_{S'} = d - 1 \quad , \quad n_R = n_A - 1$ and

$$M = \left[\begin{array}{cccc|cccc} \multicolumn{4}{c}{\overbrace{\hspace{8em}}^{d}} & \multicolumn{4}{c}{\overbrace{\hspace{8em}}^{n_A}} \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ a_2 & a_1 & \ddots & 0 & \vdots & 0 & \ddots & 0 \\ \vdots & a_2 & \ddots & 1 & 0 & \vdots & \ddots & 0 \\ a_{n_A} & \vdots & \ddots & a_1 & 1 & 0 & \ddots & 0 \\ 0 & a_{n_A} & \ddots & a_2 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n_A} & 0 & \cdots & 0 & 1 \end{array} \right]$$



Tracking : By choosing $T(q^{-1}) = P(q^{-1})$ the transfer function between the reference $r(k)$ and $y(k)$ will be :

$$H_{cl}(q^{-1}) = \frac{q^{-d}B_m(q^{-1})}{A_m(q^{-1})}$$

Model Reference Control

Example : Consider the following discrete-time second-order plant model :

$$G(q^{-1}) = \frac{0.2q^{-2} + 0.1q^{-3}}{1 - 1.3q^{-1} + 0.42q^{-2}}$$

Design an RST controller based on MRC technique for placing the closed loop dominant pole at 0.7.

- $B^\star(q^{-1}) = 0.2 + 0.1q^{-1}$ has a zero at -0.5 (inside the unit circle).
- Taking $P_d = 1 - 0.7q^{-1}$ and solving $AS + q^{-d}B^\star R = P_d B^\star$ gives $S = B^\star S'$. So we should solve $AS' + q^{-d}R = P_d$
- We have $n_{S'} = d - 1 = 1$ and $n_R = n_A - 1 = 1$ so we should solve :

$$(1 - 1.3q^{-1} + 0.42q^{-2})(1 + s_1' q^{-1}) + q^{-2}(r_0 + r_1 q^{-1}) = 1 - 0.7q^{-1}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.3 & 1 & 0 & 0 \\ 0.42 & -1.3 & 1 & 0 \\ 0 & 0.42 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ s_1' \\ r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.7 \\ 0 \\ 0 \end{bmatrix}$$

$$-1.3 + s_1' = -0.7 \qquad \Rightarrow s_1' = 0.6$$

Robust Pole Placement

A pole placement controller may not be implemented on the real system for the following reasons :

1. The controller may not be robust with respect to model uncertainty.
   - If an uncertainty weighting filter is available the robust stability condition should be verified. For multiplicative uncertainty we should have $\|W_2\mathcal{T}\|_\infty < 1$.
   - Robustness can be verified using the robustness margins like gain, phase and modulus margin M_m (the inverse of the infinity norm of the sensitivity function). $M_m \geq 0.5$ implies a gain margin of greater than 2 and a phase margin of greater than 29°.

$$\|\mathcal{S}\|_\infty = \max_{\omega} |\mathcal{S}(e^{-j\omega})| < 6\text{dB} \quad \equiv \quad M_m > 0.5$$

2. The control input may be too large and saturated in real experiment.
   - The magnitude of the transfer function between the external input and the control input should be reduced at high frequencies.
   - The dominant closed loop poles should be slowed down.

Robust Pole Placement

Example

Consider the following plant model with $h = 1s$:

$$G(q^{-1}) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}}$$

- Desired closed-loop poles : $z_{1,2} = 0.3 \pm j0.2$.
- Integrator in the controller : $H_S(q^{-1}) = 1 - q^{-1}$.

Solving the Bezout equation, we obtain :

$$\begin{aligned} R(q^{-1}) &= 1.4667 - 1.72q^{-1} + 0.6067q^{-2} \\ S(q^{-1}) &= 1 - 0.5667q^{-1} - 0.4333q^{-2} \end{aligned}$$

This controller gives :

- $M_m = \|\mathcal{S}\|_\infty^{-1} = 0.39 \quad ; \quad \|\mathcal{U}\|_\infty \approx 17$ dB.
- $|u(k)| > 2$ for an impulse output disturbance and $\|u\|_2 = 4.05$.
- Settling time of the output disturbance step response : 6 sec

Robust Pole Placement

Example

Slowing down the closed loop poles :

The dominant poles of the plant model have $\omega_n = 0.4926$ and $\zeta = 0.362$. We choose the same ω_n with $\zeta = 0.9$ to compute the desired closed-loop poles ($z_{1,2} = 0.6272 \pm j0.1368$).

Solving the Bezout equation, we obtain :

$$
\begin{aligned}
R(q^{-1}) &= 0.8721 - 1.29q^{-1} + 0.5231q^{-2} \\
S(q^{-1}) &= 1 - 0.6264q^{-1} - 0.3736q^{-2}
\end{aligned}
$$

This controller gives :

- $M_m = \|\mathcal{S}\|_\infty^{-1} = 0.566 \quad ; \quad \|\mathcal{U}\|_\infty \approx 10$ dB.
- $|u(k)| < 1.5$ for an impulse output disturbance and $\|u\|_2 = 1.92$.
- Settling time of the output disturbance step response : 12 sec

The new controller is more robust but it's slower.

Robust Pole Placement

Example

Shaping the input sensitivity function :
We add a fixed term $H_R(q^{-1}) = 1 + q^{-1}$ in the controller to reduce the input sensitivity function at high frequencies but we keep the same closed-loop poles as the original controller (fast poles). This leads to :

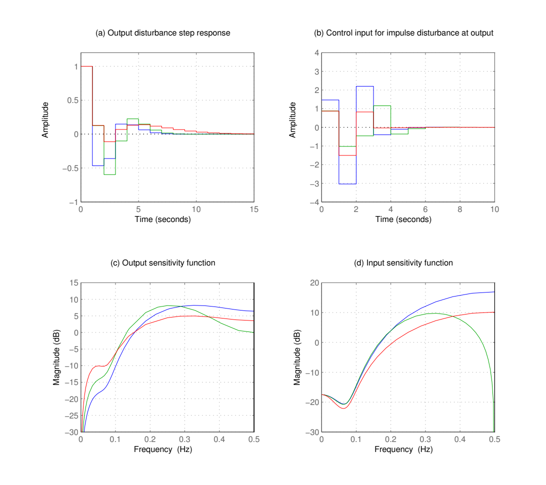
$$\begin{aligned} R(q^{-1}) &= 0.8740 - 0.2382q^{-1} - 0.6973q^{-2} + 0.4149q^{-3} \\ S(q^{-1}) &= 1 + 0.0260q^{-1} - 0.7297q^{-2} - 0.2964q^{-3} \end{aligned}$$

This controller gives :

- $M_m = \|\mathcal{S}\|_\infty^{-1} = 0.3913 \quad ; \quad \|\mathcal{U}\|_\infty \approx 10$ dB.
- $|u(k)| < 1$ for an impulse output disturbance and $\|u\|_2 = 1.87$.
- Settling time of the output disturbance step response : 7 sec

Robust Pole Placement

Example



Original controller (blue curves), slowing down the closed-loop poles (red curves), adding a fixed term $H_R(q^{-1}) = 1 + q^{-1}$ (green curves)

Q-Parametrization

Suppose that $R_0(q^{-1})$ and $S_0(q^{-1})$ are computed for a nominal model and a given $P(q^{-1})$. Then, the following set of controllers :

$$\begin{aligned} R(q^{-1}) &= R_0(q^{-1}) + A(q^{-1})Q(q^{-1}) \\ S(q^{-1}) &= S_0(q^{-1}) - B(q^{-1})Q(q^{-1}) \end{aligned}$$

where $Q(q^{-1}) = q_0 + q_1 q^{-1} + \cdots + q_{n_q} q^{-n_q}$, are also the (non minimal order) solutions of the Bezout equation :

$$\begin{aligned} AS + BR &= AS_0 - ABQ + BR_0 + BAQ \\ &= AS_0 + BR_0 \end{aligned}$$

The main advantage of this parameterization is that the sensitivity functions will depend linearly on the Q parameters.

$$\mathcal{S}(q^{-1}) = \frac{AS}{P} = \frac{A(S_0 - BQ)}{P} \qquad \mathcal{U}(q^{-1}) = \frac{AR}{P} = \frac{A(R_0 + AQ)}{P}$$

As a result, any norm of $\mathcal{S}$ or $\mathcal{U}$ is a convex function of the Q parameters.

Q-Parametrization (with fixed terms)

Suppose that $R_0(q^{-1}) = R_0'(q^{-1})H_R(q^{-1})$ and $S_0(q^{-1}) = S_0'(q^{-1})H_S(q^{-1})$ are computed for a nominal model and a given $P(q^{-1})$.
Then, the following set of controllers :

$$\begin{aligned} R(q^{-1}) &= R_0(q^{-1}) + A(q^{-1})H_R(q^{-1})H_S(q^{-1})Q(q^{-1}) \\ S(q^{-1}) &= S_0(q^{-1}) - B(q^{-1})H_S(q^{-1})H_R(q^{-1})Q(q^{-1}) \end{aligned}$$

are also the (non minimal order) solutions of the Bezout equation :

$$\begin{aligned} AS + BR &= AS_0 - ABH_S H_R Q + BR_0 + BAH_R H_S Q \\ &= AS_0 + BR_0 \end{aligned}$$

Sensitivity function shaping by convex optimization

The two norm of $\mathcal{U}$ can be minimized under a constraint on the modulus margin and the maximum amplitude of the input sensitivity function :

$$\begin{aligned} & \min_Q \|\mathcal{U}(Q)\|_2 \\ \text{subject to :} \quad & \|M_m \mathcal{S}(Q)\|_\infty < 1 \\ & \|\mathcal{U}(Q)\|_\infty < U_{\max} \end{aligned}$$

Robust Pole Placement

Example

Consider the following plant model with $h = 1s$:

$$G(q^{-1}) = \frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}}$$

- Desired closed-loop poles : $z_{1,2} = 0.3 \pm j0.2$.
- Integrator in the controller : $H_S(q^{-1}) = 1 - q^{-1}$.

Solving the Bezout equation, we obtain :

$$\begin{aligned} R(q^{-1}) &= 1.4667 - 1.72q^{-1} + 0.6067q^{-2} \\ S(q^{-1}) &= 1 - 0.5667q^{-1} - 0.4333q^{-2} \end{aligned}$$

This controller gives :

- $M_m = \|\mathcal{S}\|_\infty^{-1} = 0.39 \quad ; \quad \|\mathcal{U}\|_\infty \approx 17$ dB.
- $|u(k)| > 2$ for an impulse output disturbance and $\|u\|_2 = 4.05$.
- Settling time of the output disturbance step response : 6 sec

Robust Pole Placement

Example

This problem can be solved by standard convex optimization solvers that leads to :

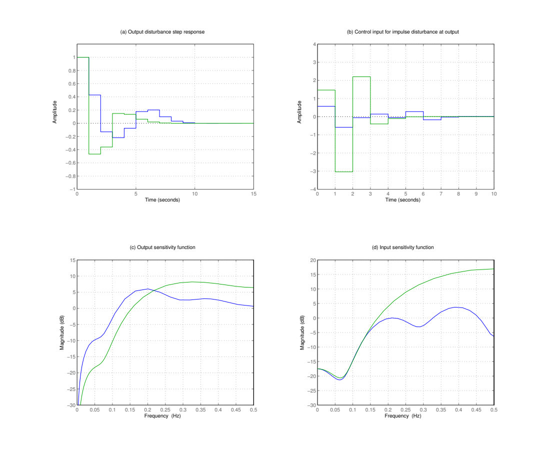
$$
\begin{aligned}
Q(q^{-1}) &= -0.895 + 0.307q^{-1} + 0.343q^{-2} + 0.245q^{-3} \\
R(q^{-1}) &= 0.572 - 0.07q^{-1} - 0.137q^{-2} - 0.054q^{-3} - 0.128q^{-4} + 0.172q^{-5} \\
S(q^{-1}) &= 1.0 + 0.329q^{-1} - 0.293q^{-2} - 0.497q^{-3} - 0.417q^{-4} - 0.123q^{-5}
\end{aligned}
$$

This controller gives :

- $M_m = \|\mathcal{S}\|_\infty^{-1} = 0.5 \quad ; \quad \|\mathcal{U}\|_\infty \approx 3.66$ dB.
- $|u(k)| < 1$ for an impulse output disturbance and $\|u\|_2 = 0.896$.
- Settling time of the output disturbance step response : 8 sec

Robust Pole Placement by Q parameterization

Example



Q parametrization (blue curves), initial controller (green curves)

Adaptive Control

Definition (Adaptive Control)

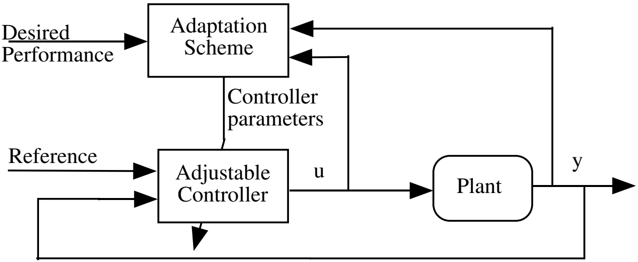
Covers a set of techniques which provide a systematic approach for automatic adjustment of controllers in *real time*, in order to achieve or to maintain a desired level of control system performance when the parameters of the plant dynamic model are unknown and/or change in time.

Unknown parameters : Adaptive control techniques can provide an automatic tuning procedure in closed loop for the controller parameters (the effect of the adaptation vanishes by time).

Time-varying parameters : To maintain an acceptable level of control system performance, an adaptive control approach has to be considered (non-vanishing adaptation).

Adaptive Control

Adaptive control : Implementation of the classical control and tuning procedure in real time.



The disturbances acting on a control system can be classified as :

a) disturbances acting upon the controlled variables (rejected by conventional feedback) ;
b) parameter disturbances acting upon the performance of the control system (rejected by adaptive control).

Adaptive Control

Direct Adaptive Control

- An error signal related to control performance is chosen.
- The controller parameter are adapted to minimize the error signal.

Indirect Adaptive Control

- The plant model is identified on-line.
- A controller is designed based on the new model.
- The updated controller is implemented on the system.

Switching Adaptive Control

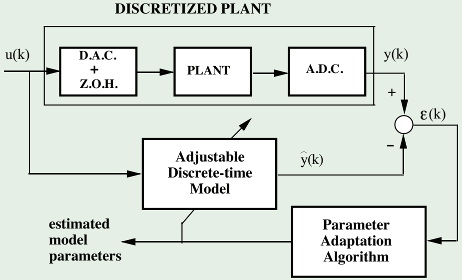
- A finite set of model are identified off-line.
- For each model a controller is designed off-line.
- The best model is chosen based on error signal on-line and the corresponding controller will be implemented.

Parameter Adaptation Algorithms

Parameter Adaptation Algorithm (PAA)

It is the main part of an adaptive control system and used for on line estimation of the parameters.

Example (PAA for model parameter estimation)



The prediction error is used by the PAA to modify, at each sampling instant, the plant model parameters.

Parameter Adaptation Algorithms

Integral type adaptation algorithm

$$\begin{bmatrix} \textit{New estimated} \\ \textit{parameters} \\ (\textit{vector}) \end{bmatrix} = \begin{bmatrix} \textit{Previous estimated} \\ \textit{parameters} \\ (\textit{vector}) \end{bmatrix} + \begin{bmatrix} \textit{Adaptation} \\ \textit{gain} \\ (\textit{matrix}) \end{bmatrix}$$

$$\times \begin{bmatrix} \textit{Measurement} \\ \textit{function} \\ \text{Observation vector} \\ (\textit{vector}) \end{bmatrix} \times \begin{bmatrix} \textit{Prediction error} \\ \textit{function} \\ \text{Adaptation error} \\ (\textit{scalar}) \end{bmatrix}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k)\phi(k)\epsilon(k+1)$$

$\hat{\theta}(k+1)$ New estimated parameters

$\hat{\theta}(k)$ Previous estimated parameters

$F(k)$ Adaptation gain

$\phi(k)$ Observation vector

$\epsilon(k+1)$ Adaptation error

Parameter Adaptation Algorithms

Consider the discrete-time model of a plant :

$$y(k+1) = -a_1 y(k) + b_1 u(k) = \theta^T \phi(k)$$

with

$$\theta^T = [a_1, b_1] \qquad \phi^T(k) = [-y(k), u(k)]$$

The adjustable *a priori* predictor will be :

$$\hat{y}^\circ(k+1) = \hat{y}[(k+1)|\hat{\theta}(k)] = -\hat{a}_1(k)y(k) + \hat{b}_1(k)u(k) = \hat{\theta}^T(k)\phi(k)$$

The *a posteriori* predicted output is given by :

$$\hat{y}(k+1) = \hat{y}[(k+1)|\hat{\theta}(k+1)] = \hat{\theta}^T(k+1)\phi(k)$$

Then the a priori and a posteriori prediction errors defined as :

$$\begin{aligned} \epsilon^\circ(k+1) &= y(k+1) - \hat{y}^\circ(k+1) \\ \epsilon(k+1) &= y(k+1) - \hat{y}(k+1) \end{aligned}$$

Recursive Least Squares Algorithm

Find a recursive algorithm which minimizes the *least squares* criterion :

$$\min_{\hat{\theta}(k)} J(k) = \sum_{i=1}^{k} [y(i) - \hat{\theta}^T(k)\phi(i-1)]^2$$

where $\hat{\theta}^T(k)\phi(i-1) = \hat{y}[i \mid \hat{\theta}(k)]$. The value of $\hat{\theta}(k)$ is obtained by solving :

$$\frac{\delta J(k)}{\delta \hat{\theta}(k)} = -2\sum_{i=1}^{k} [y(i) - \hat{\theta}^T(k)\phi(i-1)]\phi(i-1) = 0$$

Taking into account that :

$$[\hat{\theta}^T(k)\phi(i-1)]\phi(i-1) = \phi(i-1)\phi^T(i-1)\hat{\theta}(k)$$

one obtains :

$$\hat{\theta}(k) = \left[\sum_{i=1}^{k} \phi(i-1)\phi^T(i-1)\right]^{-1} \sum_{i=1}^{k} y(i)\phi(i-1) = F(k)\sum_{i=1}^{k} y(i)\phi(i-1)$$

which is not recursive !

Recursive Least Squares Algorithm

$$
\begin{aligned}
F^{-1}(k+1) &= \sum_{i=1}^{k+1} \phi(i-1)\phi^T(i-1) = F^{-1}(k) + \phi(k)\phi^T(k) \\
\hat{\theta}(k+1) &= F(k+1)\sum_{i=1}^{k+1} y(i)\phi(i-1) \\
&= F(k+1)\left[\sum_{i=1}^{k} y(i)\phi(i-1) + y(k+1)\phi(k)\right] \\
&= F(k+1)[F^{-1}(k)\hat{\theta}(k) + y(k+1)\phi(k)]
\end{aligned}
$$

Note that

$$
F^{-1}(k)\hat{\theta}(k) = F^{-1}(k+1)\hat{\theta}(k) - \phi(k)\phi^T(k)\hat{\theta}(k)
$$

Therefore :

$$
\begin{aligned}
\hat{\theta}(k+1) &= F(k+1)\left\{F^{-1}(k+1)\hat{\theta}(k) + \phi(k)[y(k+1) - \phi^T(k)\hat{\theta}(k)]\right\} \\
&= \hat{\theta}(k) + F(k+1)\phi(k)\epsilon^{\circ}(k+1)
\end{aligned}
$$

Recursive Least Squares Algorithm

Lemma (Matrix Inversion Lemma)

Let F and R be nonsingular matrices and H a full rank one, then the following identity holds :

$$(F^{-1} + HR^{-1}H^T)^{-1} = F - FH(R + H^T FH)^{-1}H^T F$$

Choose $H = \phi(k)$, $R = 1$ to obtain :

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}$$

Parameter Adaptation Algorithm

$$\epsilon^{\circ}(k+1) = y(k+1) - \hat{\theta}^T(k)\phi(k)$$

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\epsilon^{\circ}(k+1)$$

Choice of the Adaptation Gain

Least squares algorithm leads to the following adaptation gain :

$$F^{-1}(k+1) = F^{-1}(k) + \phi(k)\phi^T(k)$$

- The trace of F is non-increasing and converges asymptotically to zero.
- As a result the parameter estimates will not be adapted even if the true parameter changes. A large adaptation error will not change the parameters.
- So it can be used just for the case that the true parameters are fixed.

Generalized Adaptation Gain

The adaptation gain can be generalized by :

$$F^{-1}(k+1) = \lambda_1(k)F^{-1}(k) + \lambda_2(k)\phi(k)\phi^T(k)$$

$$0 < \lambda_1(k) \leq 1 \ ; \ 0 \leq \lambda_2(k) < 2 \ ; \ F(0) > 0$$

Note that $\lambda_1(k) < 1$ tends to increase the adaptation gain but $\lambda_2(k) > 0$ tends to decrease it.

Choice of the Adaptation Gain

Decreasing Gain : $\lambda_1(k) = \lambda_1 = 1 \quad ; \quad \lambda_2(k) = 1$

- The least squares criterion is minimized :

$$J(k) = \sum_{i=1}^{k} [y(i) - \hat{\theta}^T(k)\phi(i-1)]^2$$

- This type of profile is suited to stationary systems.

Constant forgetting factor : $\lambda_1(k) = \lambda_1\,;\ 0 < \lambda_1 < 1\,;\ \lambda_2(k) = 1$

- The typical values for λ_1, the *forgetting factor*, are 0.95 to 0.99.
- The criterion to be minimized will be :

$$J(k) = \sum_{i=1}^{k} \lambda_1^{(k-i)} [y(i) - \hat{\theta}^T(k)\phi(i-1)]^2$$

- This type of profile is suited to slowly time-varying systems.
- If the $\{\phi(k)\phi^T(k)\}$ sequence becomes null in the average (steady state case), $F(k)$ goes to infinity.

Choice of the Adaptation Gain

Variable forgetting factor : In this case $\lambda_2(k) = 1$ and the forgetting factor $\lambda_1(k)$ is given by : $\lambda_1(k) = \lambda_0\lambda_1(k-1) + 1 - \lambda_0$; $0 < \lambda_0 < 1$

- The typical values are : $\lambda_1(0) = 0.95$ to 0.99 ; $\lambda_0 = 0.5$ to 0.99
- The criterion minimized will be :

$$J(k) = \sum_{i=1}^{k} \left[\prod_{j=1}^{k} \lambda_1(j-i) \right] [y(i) - \hat{\theta}^T(k)\phi(i-1)]^2$$

- Recommended for stationary systems, since it avoids too fast decrease of the adaptation gain and accelerates the convergence.
- Other choice of $\lambda_1(k)$ (It goes to 1 if $\phi(k)\phi^T(k)$ becomes null).

$$\lambda_1(k) = 1 - \frac{\phi^T(k)F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}$$

- Another possible choice is (It goes to 1 when $\epsilon^\circ(k) \to 0$) :

$$\lambda_1(k) = 1 - \alpha \frac{[\epsilon^\circ(k)]^2}{1 + \phi^T(k)F(k)\phi(k)} \; ; \; \alpha > 0$$

Choice of the Adaptation Gain

Constant trace : In this case, $\lambda_1(k)$ and $\lambda_2(k)$ are automatically chosen at each step in order to ensure a constant trace of the gain matrix (constant sum of the diagonal terms) : $trF(k+1) = trF(k) = trF(0)$ where $F(0) = \delta n_p I$ and typically $0.1 < \delta < 4$.

- At each step there is a movement in the optimal direction of the RLS, but the gain is maintained approximately constant.
- This type of profile is suited to systems with time-varying parameters and for adaptive control with non-vanishing adaptation.

(Decreasing gain or Variable forgetting factor)+ constant trace : In this case, decreasing gain or variable forgetting factor algorithm is switched to constant trace when : $trF(k) \leq \delta n_p$.

- This profile is suited to time-varying systems and for adaptive control in the absence of initial information on the parameters.

Robust Parameter Estimation

The conventional assumptions for PAA are :

1. The true plant model and the estimated plant model have the same structure (the true plant model is described by a discrete-time model with known upper bounds for the degrees n_A, n_B).
2. The disturbances are zero mean and of stochastic nature (with various assumptions).
3. For parameter estimation in closed-loop operation, the controller
   a) has constant parameters and stabilizes the closed loop ;
   b) contains the internal model of the deterministic disturbance for which perfect state disturbance rejection is assured.
4. The parameters are constant or piece-wise constant.
5. The domain of possible parameters values is in general not constrained.

In practice these assumptions may be violated so PAA should be robust.

Robust Parameter Estimation

- **Filtering of input/output data :** Usually, we would like to estimate a model characterizing the low frequency behaviour of a plant, so we have to filter the high-frequency content of input-output, in order to reduce the effect of the unmodelled dynamics.
- **PAA with dead zone :** A dead-zone is introduced on the adaptation error such that the PAA stops when the adaptation error is smaller than the upper magnitude of the disturbance (because the adaptation error smaller than the bound of the disturbance is irrelevant).
- **PAA with projection :** In many applications, the possible domain of variation of the parameter vector θ is known (for example, the model is stable, or the sign of a component of θ is known). In such cases, the estimated parameters should be projected to a given convex domain.
- **Data normalization :** In PAA, the parameter estimates and the adaptation error remain bounded if the regressor ϕ containing plant inputs and outputs is bounded. Since we cannot assume its boundedness, we have to "normalize" the data by a norm of ϕ.

Direct Adaptive Control

Definition

Direct adaptive control covers those schemes in which the parameters of the controller are directly updated from a signal error (adaptation error) reflecting the difference between attained and desired performance.

- Direct adaptive control schemes are generally obtained by defining an equation for a signal error (adaptation error) which is a function of the difference between the tuned controller parameters and the current controller parameters.

$$\epsilon(k+1) \propto (\theta_C - \hat{\theta}_C(k))$$

- Although direct adaptive control is very appealing, it cannot be used for all types of plant model and control strategies (like pole placement, linear quadratic control or generalized predictive control).
- Model reference control can be applied if for any values of the plant parameters the finite zeros of the model are inside the unit circle.

Direct Adaptive Control

Model Reference Adaptive Control (MRAC) : Direct adaptive control using the MRC criterion will be used.

First we recall MRC for known parameters.

Model Reference Control : In this approach the zeros of the plant model in

$$H_{cl}(q^{-1}) = \frac{B_m(q^{-1})}{A_m(q^{-1})} \quad \frac{q^{-d}B^{\star}(q^{-1})T(q^{-1})}{A(q^{-1})S(q^{-1}) + q^{-d}B^{\star}(q^{-1})R(q^{-1})}$$

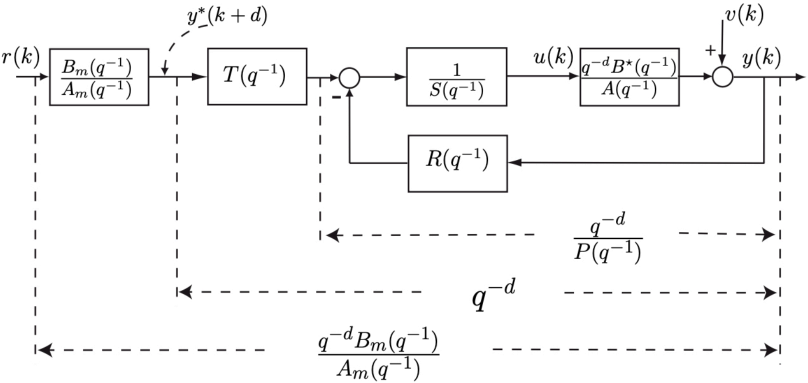
are cancelled by the closed-loop poles :

$$A(q^{-1})S(q^{-1}) + q^{-d}B^{\star}(q^{-1})R(q^{-1}) = B^{\star}(q^{-1})P(q^{-1})$$

This can be done if

- the zeros of $B^{\star}(q^{-1})$ are stable,
- complex zeros have a sufficiently high damping factor ($\zeta > 0.2$).

Model Reference Control (MRC)



Tracking : By choosing $T(q^{-1}) = P(q^{-1})$ the transfer function between the reference $r(k)$ and $y(k)$ will be :

$$H_{cl}(q^{-1}) = \frac{q^{-d}B_m(q^{-1})}{A_m(q^{-1})}$$

Model Reference Adaptive Control

The controller equation for known parameters is given by :

$$S(q^{-1})u(k) + R(q^{-1})y(k) = P(q^{-1})y^{\star}(k+d)$$

or in a regressor form as : $\theta_C^T \phi_C(k) = P(q^{-1})y^{\star}(k+d)$

where :

$$\begin{aligned} \phi_C^T(k) &= [u(k), \ldots, u(k-n_S), y(k), \ldots, y(k-n_R)] \\ \theta_C^T &= [s_0, \ldots, s_{n_S}, r_0, \ldots, r_{n_R}] \end{aligned}$$

Therefore, the control law in the adaptive case will be chosen as :

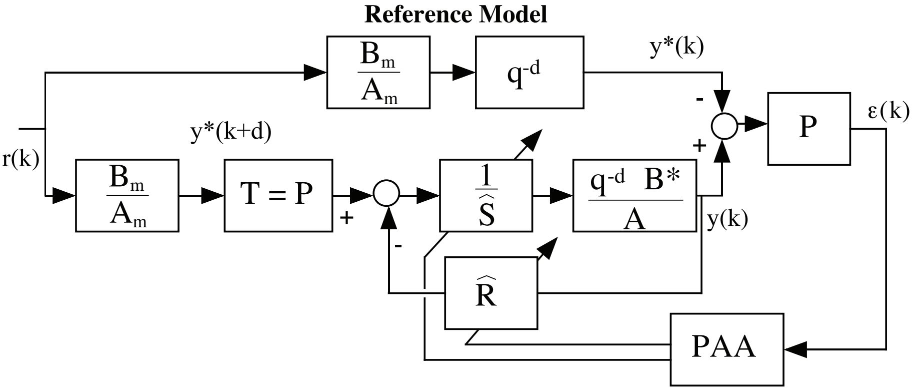
$$\hat{S}(k, q^{-1})u(k) + \hat{R}(k, q^{-1})y(k) = P(q^{-1})y^{*}(k+d)$$

$$\text{or} \qquad \hat{\theta}_C^T(k)\phi_C(k) = P(q^{-1})y^{\star}(k+d)$$

The effective control input will be computed as :

$$u(k) = \frac{1}{\hat{s}_0(k)} \left[P(q^{-1})y^{\star}(k+d) - \hat{S}^{\star}(k, q^{-1})u(k-1) - \hat{R}(k, q^{-1})y(k) \right]$$

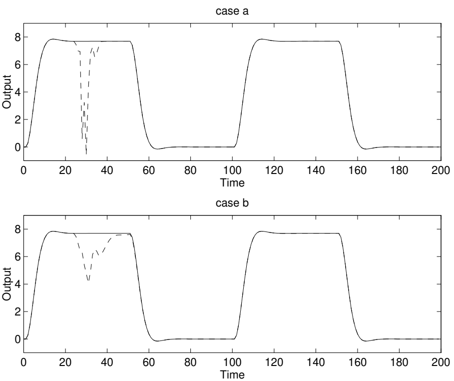
Model Reference Adaptive Control



The a priori adaptation error (linear in $\hat{\theta}_C$) is defined as :

$$\epsilon^{\circ}(k+d) = P(q^{-1})y(k+d) - P(q^{-1})y^{\star}(k+d) = [\theta_C - \hat{\theta}_C(k)]^T \phi_C(k)$$

Example

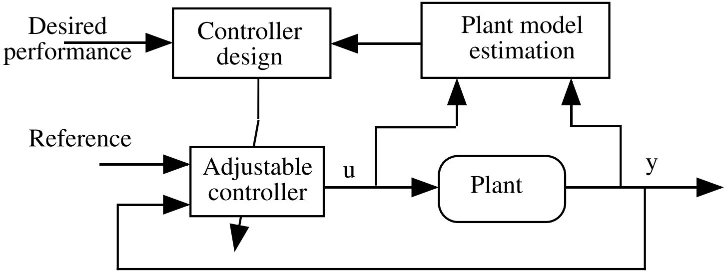


At $k = 25$, the model parameters change. Two different values of the regulation polynomial are considered : $P_1(q^{-1}) = 1$ (deadbeat control); $P_2(q^{-1}) = 1 - 1.262q^{-1} + 0.4274q^{-2}$ corresponds to a second-order system with $\omega_0 = 0.5$ rad/s and $\zeta = 0.85$.

Indirect Adaptive Control

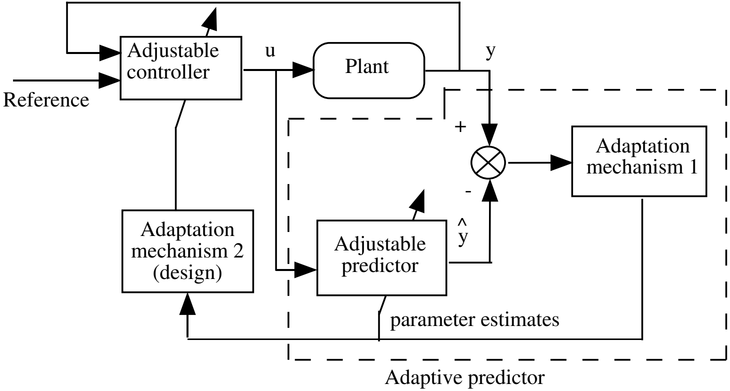
Adaptation of the controller parameters is done in two stages :

1. on-line estimation of the plant parameters ;
2. on-line computation of the controller parameters based on the current estimated plant model.



Certainty Equivalence Principle : The current plant model parameter estimates are used to compute the controller parameters as if they are equal to the true ones.

Indirect Adaptive Control



Separation Principle : The adaptive predictor gives a good prediction (or estimation) of the plant output (or states) when the plant parameters are unknown. An appropriate control for the predictor is computed and this control is also applied to the plant.

The stability of the closed-loop system is analyzed separately.

Indirect Adaptive Control (implementation)

	Strategy 1 :	**Strategy 2 :**
1	sample the plant output	sample the plant output
2	update the plant model parameters	compute the control signal based on the past controller parameters
3	update the controller parameters	send the control signal
4	compute the control signal	update the plant model parameters
5	send the control signal	update the controller parameters
6	wait for the next sample	wait for the next sample

Strategy 1 Add extra delay ; A posteriori adaptation should be analyzed.

Strategy 2 Smaller delay ; A priori adaptation should be analyzed.

Strategy 3 Update the estimates of the plant model parameters at each sampling instant, but update the controller parameters only every $N \gg 1$ sampling instants. More sophisticated controller can be designed (robust). The risk of getting non-admissible plant model is reduced.

Adaptive Pole Placement

Step 1 : Estimation of the plant model parameters

The plant output can be expressed as : $y(k+1) = \theta^T \phi(k)$ where :

$$\begin{aligned} \theta^T &= [a_1 \cdots a_{n_A}, b_d \cdots b_{n_B}] \\ \phi^T(k) &= [-y(k) \cdots - y(k-n_A+1), u(k-d+1) \cdots u(k-n_B+1)] \end{aligned}$$

The *a priori* output of the adjustable predictor is given by :

$$\hat{y}^\circ(k+1) = \hat{\theta}^T(k)\phi(k)$$

where : $\hat{\theta}^T(k) = [\hat{a}_1(k) \cdots \hat{a}_{n_A}(k), \hat{b}_d(k) \cdots \hat{b}_{n_B}(k)]$

Parameter Adaptation Algorithm

$$\begin{aligned} &\epsilon^\circ(k+1) = y(k+1) - \hat{\theta}^T(k)\phi(k) \\ &F^{-1}(k+1) = \lambda_1(k)F^{-1}(k) + \lambda_2(k)\phi(k)\phi^T(k); \\ &0 < \lambda_1(k) \leq 1 \ ; \ 0 \leq \lambda_2(k) < 2 \ ; \ F(0) > 0 \\ &\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\epsilon^\circ(k+1) \end{aligned}$$

Step 2 : Computation of the control law

We will use Strategy 1 for updating the controller parameters. The controller equation generating $u(k)$ is :

$$u(k) = -\hat{S}^{\star}(k, q^{-1})u(k-1) - \hat{R}(k, q^{-1})y(k) + \frac{1}{\hat{B}(k,1)}P(q^{-1})y^{\star}(k+d)$$

where :

$$\hat{S}(k, q^{-1}) = 1 + \hat{s}_1(k)q^{-1} + \cdots + \hat{s}_{n_S}(k)q^{-n_S} = 1 + q^{-1}\hat{S}^{*}(k, q^{-1})$$
$$\hat{R}(k, q^{-1}) = \hat{r}_0(k) + \hat{r}_1(k)q^{-1} + \cdots + \hat{r}_{n_R}(k)q^{-n_R}$$

and $\hat{S}(k, q^{-1}), \hat{R}(k, q^{-1})$ are solutions of :

$$\hat{A}(k, q^{-1})\hat{S}(k, q^{-1}) + \hat{B}(k, q^{-1})\hat{R}(k, q^{-1}) = P(q^{-1})$$

The admissibility condition for the estimated model is :

$$|\det M[\hat{\theta}(k)]| \geq \delta > 0$$

where $M[\hat{\theta}(k)]$ is the Sylvester matrix.

Robust Adaptive Pole Placement

Example

The continuous time plant to be controlled is characterized by the transfer function :

$$G(s) = \frac{2}{s+1} \cdot \frac{229}{(s^2 + 30s + 229)}$$

The system will be controlled in discrete time with a sampling period $h = 0.04$ s. For this sampling period the true discrete time plant model is given by :

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}}$$

with :

a_1	a_2	a_3
-1.8912	1.1173	-0.21225
b_1	b_2	b_3
0.0065593	0.018035	0.0030215

Robust Adaptive Pole Placement

Example

- The desired closed-loop poles used are as follows :

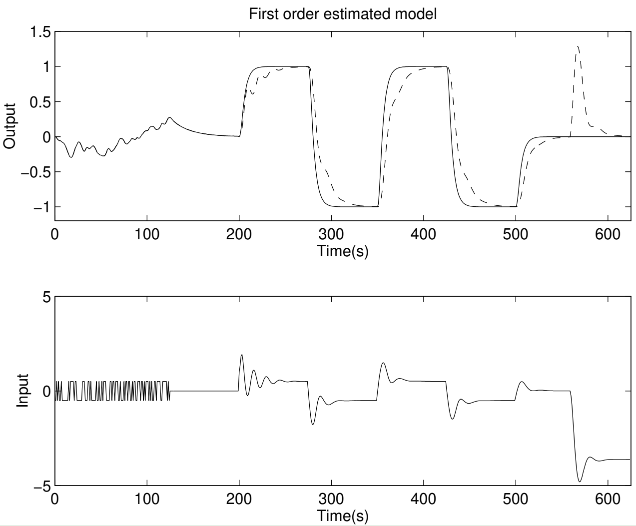
$$\begin{aligned} \text{for } n &= 1: \; P_1(q^{-1}) = (1-0.8q^{-1})(1-0.9q^{-1}) \\ \text{for } n &= 2,3: \; P_n(q^{-1}) = P_1(q^{-1})(1-0.2q^{-1})(1-0.1q^{-1}) \end{aligned}$$

- The controller has an integrator.
- Same dynamics has been used in tracking and regulation.
- For the case $n=2$, a filter $H_R(q^{-1}) = 1 + q^{-1}$ has been introduced.
- Variable forgetting factor + constant trace adaptation gain has been used ($F(0) = \alpha I$; $\alpha = 1000$ desired trace : $TrF(k) = 6$).
- The filter used on input/output data is : $L(q^{-1}) = 1/P(q^{-1})$.
- The normalizing signal $m(k)$ can be generated by :

$$m(k) = \max(\|\phi_f(k)\|, 1)$$

Robust Adaptive Pole Placement

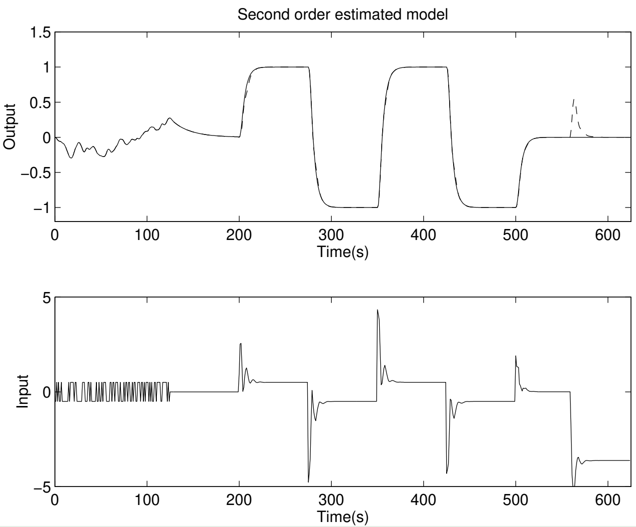
Example



First order estimated model

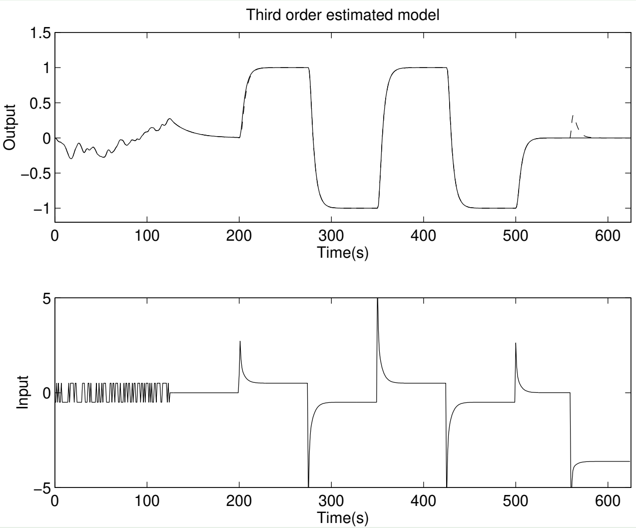
Robust Adaptive Pole Placement

Example



Second order estimated mode

Example



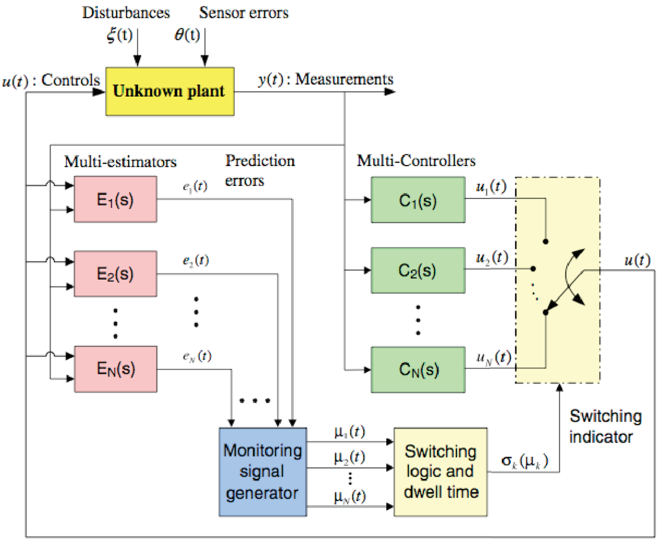
Third order estimated model

Indirect Adaptive Control

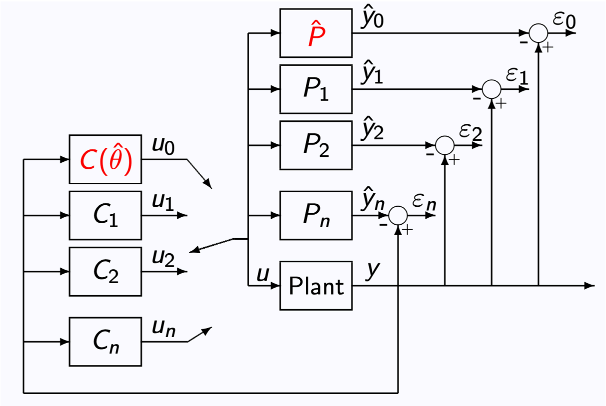
Concluding Remarks :

1. Indirect adaptive control algorithms emerged as a solution for adaptive control of systems featuring discrete time models with unstable zeros.
2. It offers the possibility of combining (in principle) any linear control strategy with a parameter estimation scheme.
3. The design of the controller based on the estimated plant models should be done such that some robustness constraints on the sensitivity functions be satisfied.
4. For each type of underlying linear control strategy used in indirect adaptive control a specific admissibility test has to be done on the estimated model prior to the computation of the control.
5. Robustification of the parameter adaptation algorithms used for plant model parameter estimation may be necessary in the presence of unmodelled dynamics and bounded disturbances.
6. Adaptive pole placement and adaptive generalized predictive control are the most used indirect adaptive control strategies.

Switching Adaptive control



Switching Adaptive control with tuning



After a parameter variation (a large estimation error)

- First the controller corresponding to the closest model (fixed model) is chosen (switching).
- Then the adaptive model is initialized with the parameter of this model and will be adapted (tuning).

Structure of Switching Adaptive Control

Plant : LTI-SISO (for analysis) with parametric uncertainty and unmodelled dynamics :

$$\bigcup_{\theta \in \Theta} P(\theta)$$

where $P(\theta) = P_0(\theta)[1 + W_2\Delta]$ with $\|\Delta\|_\infty < 1$. Other type of uncertainty can also be considered.

Multi-Estimator : Kalman filters, fixed models, adaptive models. If Θ is a finite set of n models, these models can be used as estimators (output-error estimator). If Θ is infinite but compact, a finite set of n models with and adaptive model can be used.

Multi-Controller : We suppose that for each $P(\theta)$ there exists $C(\theta)$ in the multi-controlller set that stabilizes $P(\theta)$ and satisfies the desired performances (the controllers are robust with respect to unmodelled dynamics).

Structure of Switching Adaptive Control

Monitoring Signal : is a function of the estimation error to indicate the best estimator at each time.

$$J_i(k) = \alpha\varepsilon_i^2(k) + \beta \sum_{j=0}^{k} e^{-\lambda(k-j)} \varepsilon_i^2(j)$$

with $\lambda > 0$ a forgetting factor, $\alpha \geq 0$ and $\beta > 0$ weightings for instantaneous and past errors.

Switching Logic : Based on the monitoring signal, a switching signal $\sigma(k)$ is computed that indicates which control input should be applied to the real plant. To avoid chattering, a minimum dwell-time between two consecutive switchings or a hysteresis is considered.

The dwell-time and hysteresis play an important role on the stability of the switching system.

Switching Logic

Dwell-Time : After each switching, the switching signal σ remains unchanged for $T_d h$ second. A large value for T_d may deteriorate the performance and a small value can lead to instability.

Hysteresis : A hysteresis cycle with a design parameter μ is considered between two switchings. It means that a switching to another controller will occur if the performance index concerning a model is improved by μJ_i. With hysteresis, large errors are rapidly detected and a better controller is chosen. However, the algorithm does not switch to a better controller in the set if the performance improvement is not significant.

- A combination of two logics (dwell-time and hysteresis) may also be considered.
- For both algorithm the minimum value of T_d and μ can be computed that guarantee the closed-loop stability.

Stability of Switching Adaptive Control

Trivial case :

- No unmodelled dynamics and no noise,
- the set of models is finite,
- parameters of one of the estimators matches those of the plant model,
- plant is detectable.

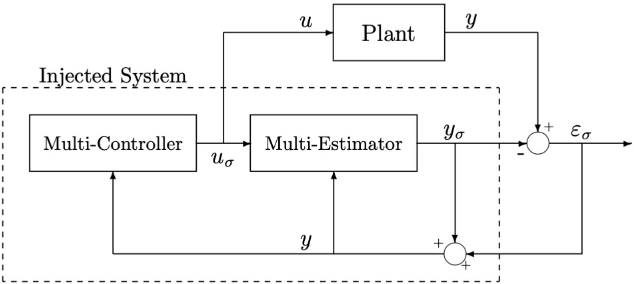
Main steps toward stability :

1. One of the estimation errors (say ε_i) goes to zero.
2. $\varepsilon_\sigma(k) = y(k) - y_\sigma(k)$ goes to zero as well.
3. After a finite time τ switching stops ($\sigma(\tau) = i \quad k \geq \tau$).
4. If ε_i goes to zero, θ_i will be equal to θ and the controller C_i stabilizes the plant $P(\theta)$:

   (*Certainty equivalence stabilization theorem*)

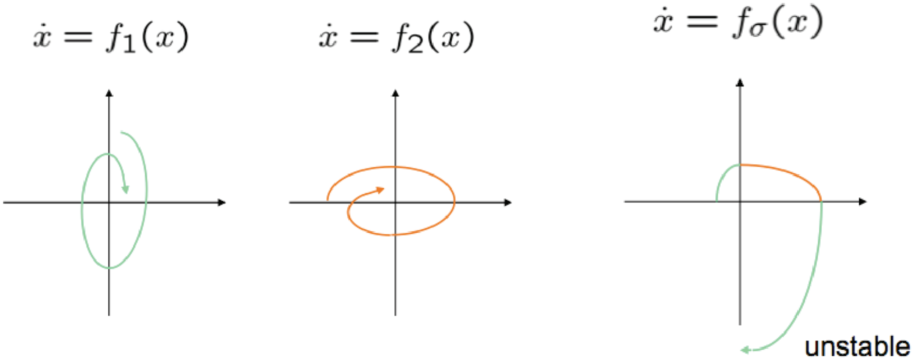
Assumptions : Presence of unmodelled dynamics and noise. Existence of some "good" estimators in the multi-estimator block. The plant P is detectable.

1. ε_i for some i is small.
2. ε_σ is small (because of a "good" monitoring signal).
3. All closed-loop signals and states are bounded if :
   The injected system is stable.



Stability of Switching Systems

Each controller stabilizes the corresponding model in the multi-estimator for frozen σ.
Question : Is the injected system stable for a time-varying switching signal $\sigma(k)$?



Is $f_\sigma(x)$ stable ? No

Stability of Switching Systems

Consider a set of stable systems :

$$\dot{x} = A_1 x, \quad \dot{x} = A_2 x, \quad \dots \quad \dot{x} = A_n x$$

then $\dot{x} = A_\sigma x$ is stable if : A_1 to A_n have a common Lyapunov matrix.
This can be verified by a set of Linear Matrix Inequalities (LMIs) :

Continuous-time	Discrete-time
$$\begin{aligned} A_1^T P + P A_1 &\prec 0 \\ A_2^T P + P A_2 &\prec 0 \\ &\vdots \\ A_n^T P + P A_n &\prec 0 \end{aligned}$$	$$\begin{aligned} A_1^T P A_1 - P &\prec 0 \\ A_2^T P A_2 - P &\prec 0 \\ &\vdots \\ A_n^T P A_n - P &\prec 0 \end{aligned}$$

- The stability is guaranteed for arbitrary fast switching.
- The stability condition is too conservative.

Concluding Remarks :

1. Large and fast parameter variations may lead to poor transient performance or even instability in classical adaptive control systems.
2. Adaptive control with switching can significantly improve the transient behaviour of adaptive systems.
3. The basic idea is to use a multi-estimator instead of a unique estimator in the adaptive control scheme. During the transients, one of the estimator can provide rapidly a good estimate of the plant output and an appropriate controller can be chosen.
4. The main issue in adaptive control with switching is the stability of the closed-loop system. It can be shown that a dwell-time can be computed that guarantees closed-loop stability.
5. The use of robust pole placement technique in adaptive control with switching guarantees quadratic stability of the injected system and consequently stability of the adaptive system with arbitrary fast switching.

Gain-Scheduled Controller Design

Motivation : A large class of nonlinear systems can be represented by a set of linear models that approximate the dynamics of the systems in different operating points.

- Such nonlinear behaviour cannot be controlled by classical linear control methods.
- Robust controllers with respect to multimodel uncertainty is a solution but may lead to poor performance.
- Direct and indirect adaptive control need a permanent persistently excitation signal and are difficult to implement.
- If the information about the operating point can be **measured**, called scheduling parameter, it can be included in the controller by making it dependent on these parameters.

Gain-Scheduled Controller Design

Gain-Scheduling method :

Step 1 : A finite grid of operating points is chosen within the whole range of operating points, then a controller is designed for each of these selected operating points based on the local model.

Step 2 : An interpolation between the controller parameters is done to get a gain-scheduled controller.

Advantages : Very good performance ; all classical controller design methods can be used ; easy implementation.

Drawback : The stability is not guaranteed for fast variations of the scheduling parameters.

Remark : Comparing with switching adaptive control, it does not need the multi-estimator part but some sensors to measure the scheduling parameters.

Gain-Scheduled Controller Design

Data-Driven Method : The two steps can be combined in one convex optimization problem.
Models : $G(j\omega, \theta)$ is a function of a scheduling parameter $\theta \in \Theta = [\theta_{\text{min}}\ ,\ \theta_{\text{max}}]$.
Gain-Scheduled Controller : $K(z, \theta) = X(\theta)Y^{-1}(\theta)$, where for a second order interpolation, we have

$$X(\theta) = X_0 + \theta X_1 + \theta^2 X_2$$
$$Y(\theta) = Y_0 + \theta Y_1 + \theta^2 Y_2$$

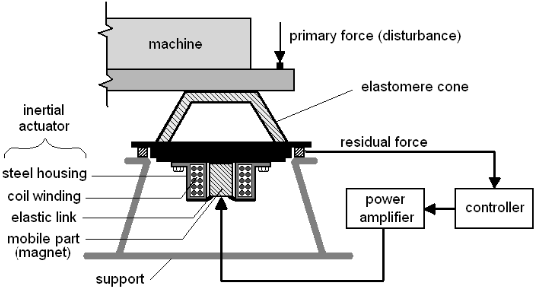
Gain-Scheduled $\mathcal{H}_\infty$ Controller

$$\min_{X,Y} \gamma$$

$$\begin{bmatrix} \gamma I & W_1 Y(\theta) \\ (W_1 Y(\theta))^* & \Phi(\theta)^*\Phi_c(\theta) + \Phi_c(\theta)^*\Phi(\theta) - \Phi_c(\theta)^*\Phi_c(\theta) \end{bmatrix} \succ 0 \qquad \forall \omega \in \Omega$$

and $\forall \theta \in \Theta$, where $\Phi(\theta) = Y(\theta) + G(\theta)X(\theta)$ and $\Phi_c(\theta) = Y_c(\theta) + G(\theta)X_c(\theta)$.
This problem is convex wrt the controller parameters but has infinite number of constraints (SIP). As before, it can be solved by gridding the frequency and the scheduling parameter.

Active Suspension Benchmark



- Active suspension system is used to reject the effect of disturbance.
- An inertial actuator creates vibrational forces to counteract the disturbances (like loudspeakers).
- A shaker used to generate the disturbances.

Active Suspension Benchmark

Control Objective :

- The disturbance consists of one sinusoidal signal with unknown time-varying frequency, which lies in an interval from 50 to 95Hz.
- The controller should reject the disturbance as fast as possible.
- The magnitude of input sensitivity function $|\mathcal{U}|$ should be less than 10 dB at high frequencies.
- The noise at other frequencies should not be amplified more than 6 dB ($M_m = 0.5$).

Linear controller design : If the frequency of the disturbance was known, it could be rejected using the internal model principle.

- A fixed term in the controller should be considered as follows :

$$F_y(z,\theta) = z^2 + \theta z + 1 \qquad f = \cos^{-1}(-\theta/2)/2\pi$$

Active Suspension Benchmark

Gain-scheduled controller design :

1. A very fine frequency grid with a resolution of 0.5 rad/s (5027 frequency points) is considered.
2. A resolution of 1Hz, which corresponds to 46 points in the interval $[-1.8478\,,\,-1.4686]$ is considered for the scheduling parameter θ.
3. The following controller structure is chosen $K(\theta) = X(\theta)Y^{-1}(\theta)$:

$$X(z,\theta) = \frac{X_0(z) + \theta X_1(z)}{(z-\alpha)^n} \quad ; \quad Y(z,\theta) = \frac{F_y(z,\theta)}{(z-\alpha)^n}$$

4. The following optimization problem is solved :

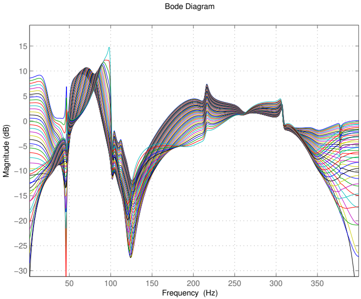
$$\min_{X,Y} \gamma$$

$$\left\| \begin{array}{c} W_1 \mathcal{S}(\theta) \\ W_3 \mathcal{U}(\theta) \end{array} \right\|_\infty < \gamma \quad ; \quad \|\mathcal{S}(\theta)\|_\infty < 2 \quad ; \quad \forall \theta \in \Theta$$

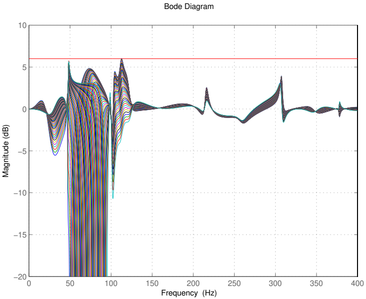
where $W_1 = F_y^{-1}$ and $W_3 = 1$.

Active Suspension Benchmark

Control performance for known and fixed disturbance frequency



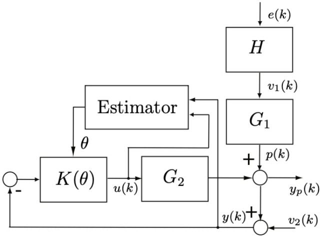
Input sensitivity function

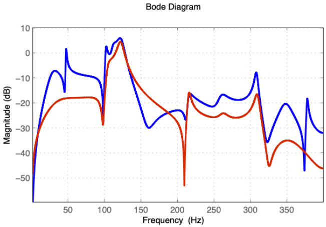


Output sensitivity function

Active Suspension Benchmark

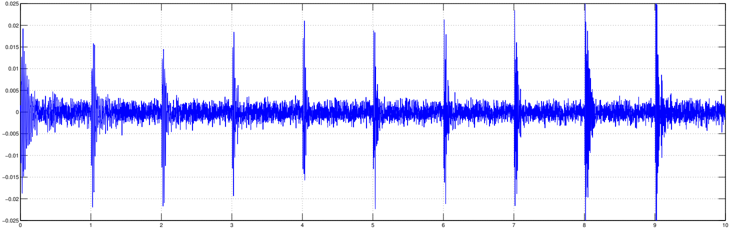
Estimator design : The disturbance frequency should be estimated on line.



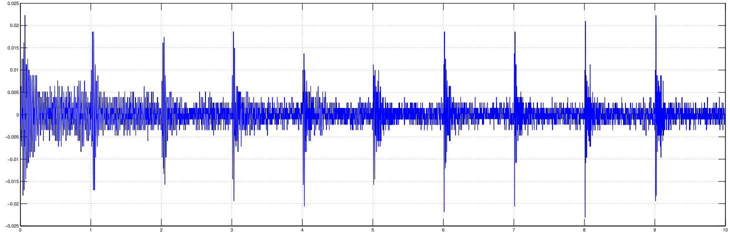


- If we model $p(k)$ as the output of an ARMA model with white noise as input, we have : $D_p(q^{-1})p(k) = N_p(q^{-1})e(k)$,
- Since $p(k)$ is not available, it is estimated using the measured signal $y(k)$ and the known model of the secondary path.
- Then a standard PAA is used to estimate θ.

Active Suspension Benchmark



Transient responses in simulation (disturbance frequencies from 50Hz to 95Hz)



Experimental Transient responses (disturbance frequencies from 50Hz to 95Hz)