ME 233 Advance Control II

Lecture 21

Indirect Adaptive Pole Placement, Disturbance Rejection and Tracking Control

Adaptive Control

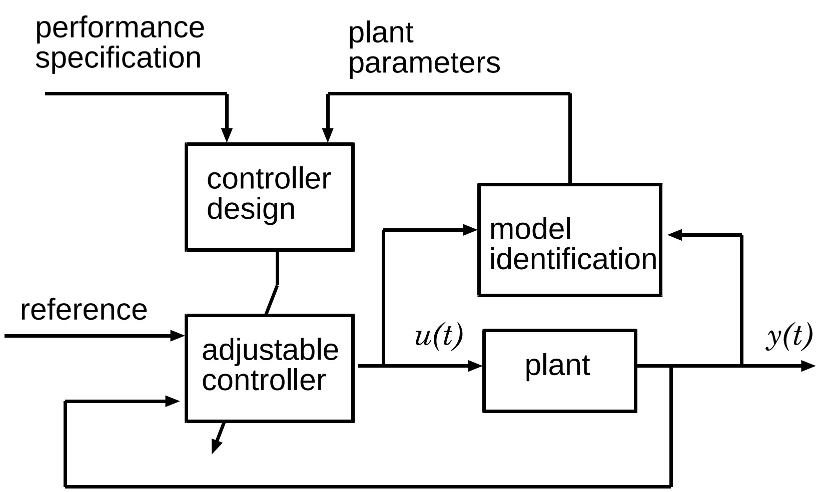
Adaptive Control Principle

Controller parameters **are not constant,** rather, they are adjusted in an online fashion by a ***Parameter Adaptation Algorithm (PAA)***

When is adaptive control used?

- Plant parameters are unknown
- Plant parameters are slowly time varying

Self-Tuning Regulator (STR):



Self-tuning Regulator Approach

- Control Design Procedure:
  - Pole-placement, tracking control and deterministic disturbance rejection for ARMA models (Lecture 15).
- Model Identification:
  - Series-parallel with Recursive Least Squares (RLS) identification with or without forgetting factor.

Certainty Equivalence Principle

1. Use pole-placement, tracking control and deterministic disturbance rejection controller synthesis methodology.
2. Design controller assuming that plant parameters are known.
3. Estimate plant parameters using RLS PAA.
4. Controller parameters are re-calculated at every sample instance by assuming that the latest plant parameters estimates are the real parameters.

Direct vs. Indirect Adaptive Control

- Both use pole-placement, tracking control and deterministic disturbance rejection controller synthesis methodology.
- **Indirect adaptive control:**
  1. Plant parameters are estimated using a RLS PAA.
  2. Controller parameters are calculated using the certainty equivalence principle.
  - **Use with plants that have non-minimum phase zeros.** (Plant unstable zeros are not cancelled).
- **Direct adaptive control:**
  1. Controller parameters are updated directly using a RLS PAA.
  - **Use with plants that do not have non-minimum phase zeros.** (Plant zeros are cancelled).

Outline

1. Review lecture 15: Pole-placement, tracking control and deterministic disturbance rejection for ARMA models.
2. Formulate the plant's Parameter Adaptation Algorithm (PAA).
3. Implement an indirect adaptive controller, using the certainty equivalence principle.
4. For plants with minimum phase zeros, we will simplify the indirect adaptive controller.

Deterministic SISO ARMA models

SISO ARMA model

$$A(q^{-1})\, y(k) = q^{-d}\, B(q^{-1})\, [u(k) + d(k)]$$

Where all inputs and outputs are scalars:

- $u(k)$ control input
- $d(k)$ deterministic but unknown disturbance
- $y(k)$ output

Deterministic SISO ARMA models

$$A(q^{-1})\,y(k) = q^{-\mathrm{d}}\,B(q^{-1})\,[\,u(k) + d(k)]$$

Where polynomials:

$$A(q^{-1}) \;=\; 1 + a_1 q^{-1} + \cdots + a_n q^{-n}$$

$$B(q^{-1}) \;=\; b_o + b_1 q^{-1} + \cdots + b_m q^{-m}$$

are co-prime and d is the ***known*** pure time delay

Deterministic SISO ARMA models

The zero polynomial:

$$B(q^{-1}) = B^s(q^{-1})\,B^u(q^{-1})$$

Without loss of generality, we will assume that

$B^s(q^{-1}) = 1 + \cdots + b^s_{m_s}\,q^{-m_s}$ is Schur

$B^u(q^{-1}) = b_o + \cdots + b^u_{m_u}\,q^{-m_u}$ zeros which we **do not** wish to cancel

$$m_s + m_u = m$$

i.e. the polynomial $B^s(q^{-1})$ is monic

Control Design

1. Pole Placement: The poles of the closed loop system must be placed at specific locations in the complex plane.

- **Closed loop pole polynomial:**

$$\boxed{A_c(q^{-1}) = B^s(q^{-1})\,A_c^{'}(q^{-1})}$$

Where:

- $B^s(q^{-1})$ cancelable plant zeros
- $A_c^{'}(q^{-1})$ monic Schur polynomial chosen by the designer

$$A_c^{'}(q^{-1}) = 1 + a_{c1}^{'} q^{-1} + \cdots + a_{c n_c^{'}}^{'} q^{-n_c^{'}}$$

Control Objectives

2. **<u>Tracking</u>**: The output sequence $y(k)$ must follow a ***reference*** sequence $y_d(k)$ which is known

- **Reference model:**

$$A_m(q^{-1})y_d(k) = q^{-\mathrm{d}}\,B_m(q^{-1})\,u_d(k)$$

Where:

- $y_d(k)$ **reference output sequence,** which is known in advance (i.e. $y_d(k+L)$ is available at instance k for some $L>0$).
- $A_m(q^{-1})$ monic Schur polynomial chosen by the designer
- $B_m(q^{-1})$ polynomial chosen by the designer

Control Objectives

3. <u>Disturbance rejection</u>: The closed loop system must reject a class of **<u>persistent</u>** disturbances $d(k)$

- **Disturbance model:**

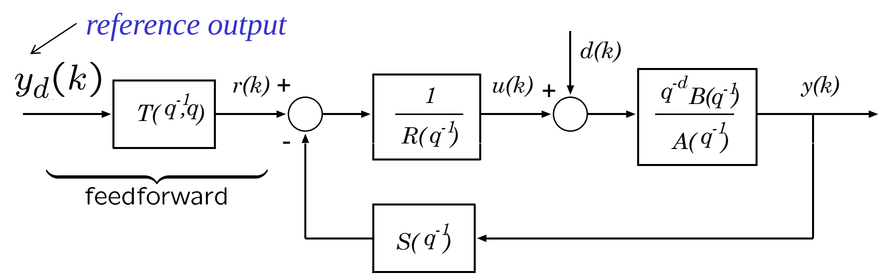
$$\boxed{A_d(q^{-1})d(k) = 0}$$

Where

- $A_d(q^{-1})$ is a ***known*** annihilating polynomial <u>with roots on the unit circle</u>
- $A_d(q^{-1}),\ B(q^{-1})$ are co-prime

Control Law

- Feedback and feedforward actions:



$$u(k) = \frac{1}{R(q^{-1})}\left[r(k) - S(q^{-1})y(k)\right]$$

$$r(k) = T(q^{-1},q)\, y_d(k)$$

Feedforward action (a-causal)

Feedback Controller

Diophantine equation: Obtain polynomials $R^{'}(q^{-1}),\ S(q^{-1})$ which satisfy:

$$\boxed{A_c^{'}(q^{-1}) = A_d(q^{-1})\, A(q^{-1})\, R^{'}(q^{-1}) + q^{-\mathrm{d}}B^{u}(q^{-1})\, S(q^{-1})}$$

Close loop poles minus cancelled zeros

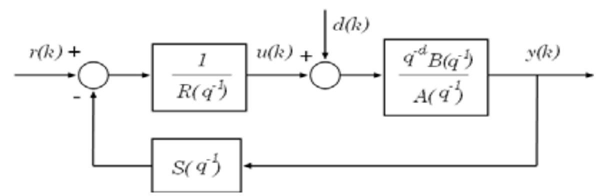
Disturbance annihilating polynomial

Plant poles

Unstable plant zeros

$$\boxed{\begin{aligned} R(q^{-1}) &= R^{'}(q^{-1})\, A_d(q^{-1})\, B^{s}(q^{-1}) \\ A_c(q^{-1}) &= B^{s}(q^{-1})\, A_c^{'}(q^{-1}) \end{aligned}}$$

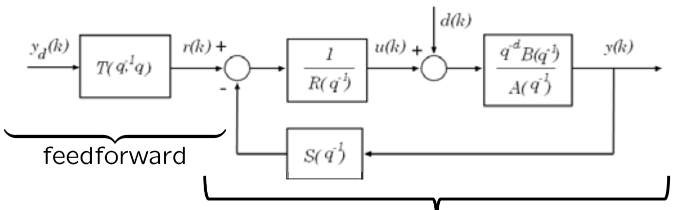
Feedback Controller



$$\boxed{y(k) = \frac{q^{-\mathrm{d}}B^{u}(q^{-1})}{A_c^{'}(q^{-1})}\, r(k)}$$

$$+\underbrace{\frac{q^{-\mathrm{d}}B(q^{-1})\, R^{'}(q^{-1})}{A_c(q^{-1})}\, A_d(q^{-1})d(k)}_{\to 0}$$

Zero-phase error feedforward



$$\frac{q^{-d}B^u(q^{-1})}{A_c'(q^{-1})}$$

$$\boxed{T(q^{-1},q) = A_c'(q^{-1})\, q^d \, \frac{B^u(q)}{[B^u(1)]^2}}$$

Outline

1. Review lecture 16: Pole-placement, tracking control and deterministic disturbance rejection for ARMA models.
2. Formulate the plant's Parameter Adaptation Algorithm (PAA).
3. Implement an indirect adaptive controller, using the certainty equivalence principle.
4. For plants with minimum phase zeros, we will simplify the indirect adaptive controller.

Certainty Equivalence Principle

1. Use pole-placement, tracking control and deterministic disturbance rejection controller synthesis methodology.

2. Estimate plant parameters using RLS PAA:
   - Polynomial estimates: $\hat{A}(q^{-1},k)$ $\hat{B}(q^{-1},k)$

3. Controller polynomials $R'(q^{-1})$ $S(q^{-1})$
   Feedforward compensator $T(q,q^{-1})$

   are computed using $\hat{A}(q^{-1},k)$ $\hat{B}(q^{-1},k)$

Parameter Adaptation Algorithm (PAA)

1. Use a series-parallel RLS algorithm to estimate plant parameters.

2. Pre-filter input $u(k)$ and output $y(k)$ using the disturbance annihilating polynomial, to prevent parameter biasing.

3. Use ``parameter projection" to prevent unbounded control input

PAA: sequence pre-filtering

Plant dynamics:

$$A(q^{-1})\, y(k) = q^{-\mathsf{d}}\, B(q^{-1})\, [u(k) + d(k)]$$

Disturbance:

$$A_d(q^{-1}) d(k) = 0$$

Filtered input and output sequences:

$$\boxed{\begin{array}{rcl} y_f(k) & = & A_d(q^{-1})\, y(k) \\ u_f(k) & = & A_d(q^{-1})\, u(k) \end{array}}$$

PAA: sequence pre-filtering

Multiply plant dynamics by annihilating polynomial:

$$A(q^{-1})\, \underbrace{A_d(q^{-1}) y(k)}_{y_f(k)} = q^{-\mathsf{d}}\, B(q^{-1}) \left[\underbrace{A_d(q^{-1}) u(k)}_{u_f(k)} + \underbrace{A_d(q^{-1}) d(k)}_{=0} \right]$$

$$\boxed{A(q^{-1})\, y_f(k) = q^{-\mathsf{d}}\, B(q^{-1}) u_f(k)}$$

PAA: series parallel RLS

Filtered plant dynamics

$$A(q^{-1})\, y_f(k) = q^{-\mathsf{d}}\, B(q^{-1}) u_f(k)$$

Can be written as

$$\boxed{y_f(k) = \phi(k-1)^T\, \theta}$$

Where:

$$\theta = \left[\begin{array}{cccccc} a_1 & \cdots & a_n & b_o \cdots & b_m \end{array}\right]^T \in \mathcal{R}^{n+m+1}$$

$$\phi(k-1) = \left[\begin{array}{cccccc} -y_f(k-1) & \cdots & -y_f(k-n-1) & u_f(k-\mathsf{d}) & \cdots & u_f(k-\mathsf{d}-m) \end{array}\right]^T$$

PAA: parameter projection

Assume that we know:

1. Minimum magnitude of DC gain of $B^u(q^{-1})$

$$\boxed{|B^u(1)| \;\geq\; B^u_{\mathsf{min}} > 0}$$

2. Sign and minimum value of leading coefficient of $B^s(q^{-1}) = b_o + \cdots + b^s_{m_s}\, q^{-m_s}$

$$\boxed{b_o \;\geq\; b_{mino} > 0}$$

Series parallel RL with projection

PAA:

$$
\begin{aligned}
e^o(k+1) &= y_f(k+1) - \phi^T(k)\hat{\theta}(k) \\
e(k+1) &= \frac{e^o(k+1)}{1+\phi^T(k)F(k)\phi(k)} \\
\hat{\theta}^o(k+1) &= \hat{\theta}(k) + F(k)\phi(k)\, e(k+1) \\
F(k+1) &= \frac{1}{\lambda_1(k)}\left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \phi^T(k)F(k)\phi(k)}\right]
\end{aligned}
$$

$$
\begin{aligned}
0 &< \lambda_1(k) \leq 1 \\
0 &\leq \lambda_2(k) < 2
\end{aligned}
$$

$\hat{\theta}^o(k+1)$: A-priori parameter estimate (prior to projection)

Series parallel RL with projection

PAA: Projection

$\hat{b}_o(k)$

$$
\hat{\theta}(k) = \begin{cases} \hat{\theta}^o(k) & \text{if } \hat{b}^o_o(k) \geq b_{mino} \\ \left[\ \hat{a}^o_1(k) \cdots \hat{a}^o_n(k)\ b_{mino}\ \cdots\ \hat{b}^o_m(k)\ \right]^T & \text{if } \hat{b}^o_o(k) < b_{mino} \end{cases}
$$

Replace $\hat{b}^o_o(k)$ by b_{mino} if it becomes too small.

Outline

1. Review lecture 16: Pole-placement, tracking control and deterministic disturbance rejection for ARMA models.
2. Formulate the plant's Parameter Adaptation Algorithm (PAA).
3. Implement an indirect adaptive controller, using the certainty equivalence principle.
4. For plants with minimum phase zeros, we will simplify the indirect adaptive controller.

Indirect Adaptive Controller

After each PAA iteration:

- Update $\hat{A}(q^{-1},k)$ $\hat{B}(q^{-1},k)$ polynomials:

$$
\begin{aligned}
\hat{A}(q^{-1},k) &= 1 + \hat{a}_1(k)\, q^{-1} + \cdots + \hat{a}_n(k)\, q^{-n} \\
\hat{B}(q^{-1},k) &= \hat{b}_o(k) + \hat{b}_1(k)\, q^{-1} + \cdots + \hat{b}_m(k)\, q^{-m}
\end{aligned}
$$

- Factorize $\hat{B}(q^{-1},k)$ polynomial:

$$
\hat{B}(q^{-1},k) = \hat{B}^s(q^{-1},k)\hat{B}^u(q^{-1},k)
$$

$\boxed{\hat{B}^s(q^{-1},k) \text{ : monic}}$

Indirect Adaptive Controller

3) Calculate controller polynomials:

$$\widehat{R}'(q^{-1},k) \qquad \widehat{S}(q^{-1},k)$$

by solving the Diophantine equation:

$$A_c'(q^{-1}) \;=\; A_d(q^{-1})\,\underline{\widehat{A}(q^{-1},k)}\,\widehat{R}'(q^{-1},k) + q^{-d}\,\underline{\widehat{B}^u(q^{-1},k)}\widehat{S}(q^{-1},k)$$

Plant parameter polynomial estimates are used instead of actual polynomials

Indirect Adaptive Controller

4) Calculate feedforward filter: $\widehat{T}(q^{-1},q,k)$

$$\widehat{T}(q^{-1},q,k) = \frac{\widehat{B}^u(q,k)\,B_m(q^{-1})}{[\bar{B}^u(k)]^2}$$

Where:

$$\bar{B}^u(k) = \begin{cases} \widehat{B}^u(1,k) & \text{if } \;|\widehat{B}^u(1,k)| \geq B^u_{\min} \\ B^u_{\min} & \text{if } \;|\widehat{B}^u(1,k)| < B^u_{\min} \end{cases} .$$

Replace $\widehat{B}^u(1,k)$ by $B^u_{\min}$
if it becomes too small.

Indirect Adaptive Controller

5) Calculate polynomial: $\widehat{R}(q^{-1},k)$

$$\widehat{R}(q^{-1},k) \;=\; A_d(q^{-1})\,\widehat{B}^s(q^{-1},k)\,\widehat{R}'(q^{-1},k)$$

Notice that , $A_d(q^{-1})$, $\widehat{B}^s(q^{-1},k)$ and $\widehat{R}'(q^{-1},k)$ are monic.

Thus,

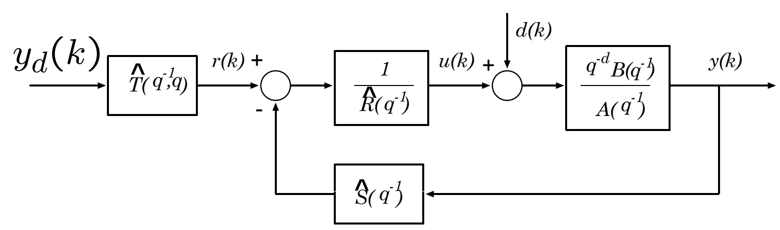
$$\widehat{R}(q^{-1},k) \;=\; 1 + q^{-1}\,\widehat{r}_1(k) + \cdots + q^{-n_r}\widehat{r}_{n_r}(k)$$

is also monic

Indirect Adaptive Controller

6) Adaptive control law is given by:

$$\boxed{\widehat{R}(q^{-1},k)\,u(k) = \widehat{T}(q^{-1},q,k)\,y_d(k) - \widehat{S}(q^{-1},k)y(k)}$$



Outline

1. Review lecture 16: Pole-placement, tracking control and deterministic disturbance rejection for ARMA models.

2. Formulate the plant's Parameter Adaptation Algorithm (PAA).

3. Implement an indirect adaptive controller, using the certainty equivalence principle.

4. For plants with minimum phase zeros, we will simplify the indirect adaptive controller.

Indirect Adaptive Controller with Stable Zeros

After each PAA iteration:

1. Update $\hat{A}(q^{-1},k)$ $\hat{B}(q^{-1},k)$ polynomials:

$$\hat{A}(q^{-1},k) \;=\; 1 + \hat{a}_1(k)\,q^{-1} + \cdots + \hat{a}_n(k)\,q^{-n}$$

$$\hat{B}(q^{-1},k) \;=\; \hat{b}_o(k) + \hat{b}_1(k)\,q^{-1} + \cdots + \hat{b}_m(k)\,q^{-m}$$

- (no need to factorize $\hat{B}(q^{-1},k)$)

Indirect Adaptive Controller with Stable Zeros

2) Calculate controller polynomials:

$$\hat{R}'(q^{-1},k) \qquad \hat{S}(q^{-1},k)$$

by solving the Diophantine equation:

$$A_c'(q^{-1}) \;=\; A_d(q^{-1})\,\underline{\hat{A}(q^{-1},k)}\,\hat{R}'(q^{-1},k) + q^{-d}\,\hat{S}(q^{-1},k)$$

Plant parameter polynomial estimates are used instead of actual polynomials

Indirect Adaptive Controller with Stable Zeros

- Feedforward filter $T(q^{-1},q)$ is ***constant***

$$T(q^{-1},q) = q^d A_c'(q^{-1})$$

Thus, there is no need to update it at every sample step.

Indirect Adaptive Controller with Stable Zeros

3) Calculate polynomial: $\widehat{R}(q^{-1},k)$

$$\widehat{R}(q^{-1},k) \;=\; A_d(q^{-1})\,\widehat{B}(q^{-1},k)\,\widehat{R}'(q^{-1},k)$$

Notice that both $A_d(q^{-1})$ and $\widehat{R}'(q^{-1},k)$ are monic.

Thus,

$$\widehat{R}(q^{-1},k) \;=\; \underbrace{\widehat{r}_o(k)}_{=\,\widehat{b}_o(k)} + q^{-1}\,\widehat{r}_1(k) + \cdots + q^{-n_r}\widehat{r}_{n_r}(k)$$

This coefficient is always $\geq\; b_{mino}$

Indirect Adaptive Controller with Stable Zeros

4) Adaptive control law is given by:

$$\boxed{\widehat{R}(q^{-1},k)\,u(k) = A_c'(q^{-1})\,y_d(k+d) - \widehat{S}(q^{-1},k)y(k)}$$