ME 233 Advanced Control II

Lecture 23

Direct Adaptive Pole Placement, and Tracking Control

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).

Direct Adaptive Control

1. Plants with <u>minimum phase zeros</u> and <u>no disturbances</u>:

   - Controller design (review)

   1. Controller PAA
   2. Adaptive Controller

2. Plants with <u>minimum phase zeros</u> and <u>constant disturbances</u>:

- Read section: ***Direct adaptive control with integral action for plants with stable zeros*** in the ME233 class notes, part II.

Deterministic SISO ARMA models

SISO ARMA model

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

Where all inputs and outputs are scalars:

- $u(k)$ control input
- $y(k)$ output

d is the ***known*** pure time delay

Deterministic SISO ARMA models

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

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 $B(q^{-1})$ is anti-Schur

Control Objectives

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

- **Closed-loop pole polynomial:**

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

Where:

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

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

Control Objectives

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

- **Reference model:**

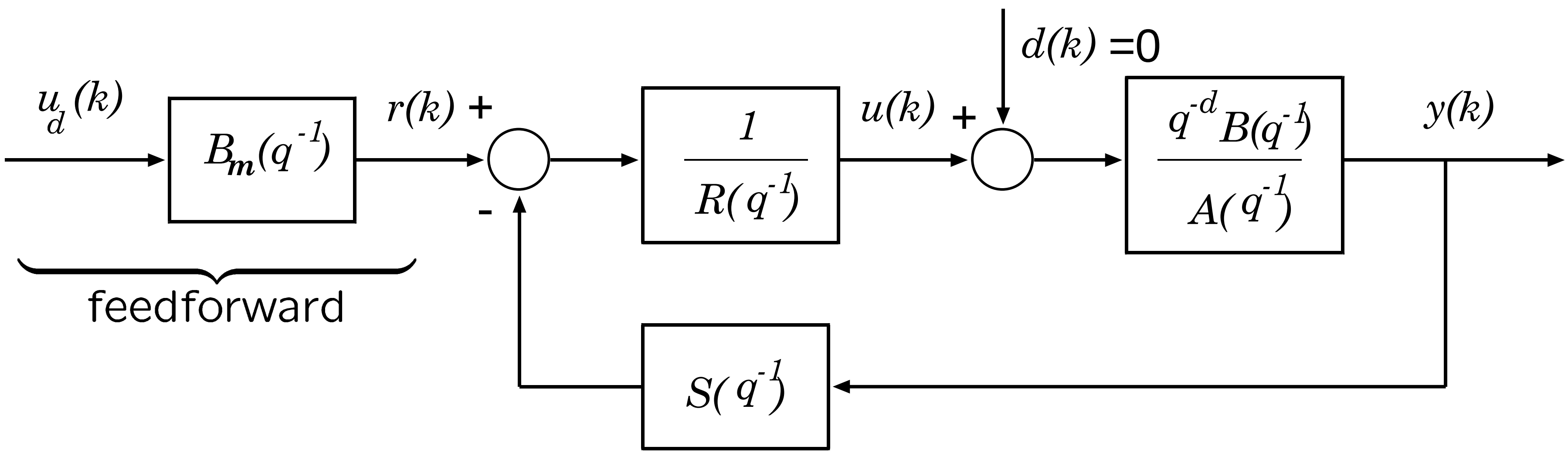
$$\boxed{A_c^{'}(q^{-1}) y_d(k) = q^{-\mathrm{d}}\, B_m(q^{-1})\, u_d(k)}$$

Where:

- $u_d(k)$ **known** reference input control input sequence
- $A_c^{'}(q^{-1})$ (from the previous slide)
- $B_m(q^{-1})$ zero polynomial, chosen by the designer

Control Law

- Feedback and feedforward actions:



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

$$r(k) = q^{+\mathrm{d}} A_c'(q^{-1}) y_d(k) \;=\; B_m(q^{-1}) u_d(k)$$

$$\frac{q^{-\mathrm{d}} B_m(q^{-1})}{A_c'(q^{-1})} u_d(k)$$

Feedforward is causal

Feedback Controller

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

$$A_c'(q^{-1}) = A(q^{-1})\, R'(q^{-1}) + q^{-\mathsf{d}}\, S(q^{-1})$$

Closed-loop poles

Plant poles

Plant pure delays

$$R(q^{-1}) = R'(q^{-1})\, B(q^{-1})$$

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

Diophantine equation

$$A_c'(q^{-1}) = A(q^{-1})\,R'(q^{-1}) + q^{-\mathsf{d}}\,S(q^{-1})$$

Solution:

$$R'(q^{-1}) \;=\; 1 + r_1'q^{-1} + \cdots + r_{n_r'}\,q^{-n_r'}$$

$$S(q^{-1}) \;=\; s_o + s_1q^{-1} + \cdots + s_{n_s}q^{-n_s}$$

$$\boxed{\begin{aligned} n_r' \;&=\; \mathsf{d} - 1 \\ n_s \;&=\; \max\{\, n - 1\,,\, n_c' - \mathsf{d}\,\} \end{aligned}}$$

Feedback Controller

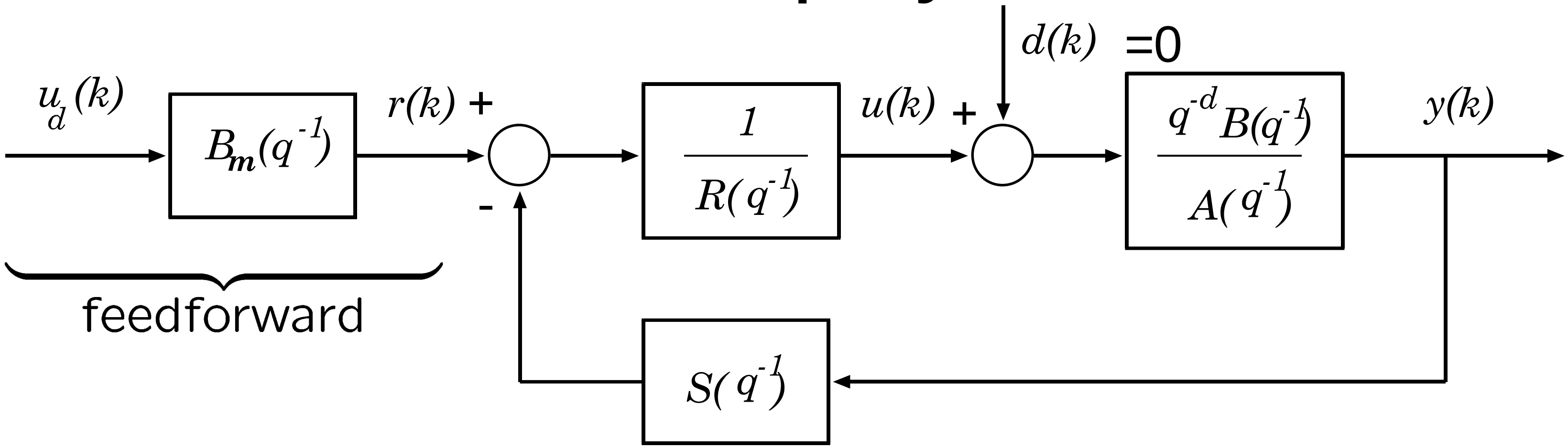
$$u(k) = \frac{1}{R(q^{-1})}\left[r(k) - S(q^{-1})y(k)\right]$$

where

$$R(q^{-1}) = R^{'}(q^{-1})\, B(q^{-1})$$

$$\boxed{\begin{aligned} n_r^{'} &= \mathsf{d} - 1 \\ n_s &= \max\{\, n - 1\,,\ n_c^{'} - \mathsf{d}\,\} \\ n_r &= n_r^{'} + m \end{aligned}}$$

Closed-loop dynamics



$$
\begin{aligned}
A_c'(q^{-1})\, y(k) &= q^{-\mathsf{d}}\, r(k) \\
&= q^{-\mathsf{d}} B_m(q^{-1}) u_d(k) \\
&= A_c'(q^{-1})\, y_d(k)
\end{aligned}
$$

$$
\boxed{A_c'(q^{-1})\, \{ y(k) - y_d(k) \} = 0}
$$

Direct Adaptive Control

1. Plants with minimum phase zeros and no disturbances:
   - Controller design

   1. Controller PAA

   2. Adaptive Controller

Controller parameters

We want to identify the controller polynomials

$$R(q^{-1}) \qquad\qquad S(q^{-1})$$

directly, where

$$R(q^{-1}) = R^{'}(q^{-1})\, B(q^{-1})$$

$$R(q^{-1}) = \underbrace{r_o}_{=b_o} + r_1\, q^{-1} + \cdots + r_{n_r}\, q^{-n_r}$$

$$S(q^{-1}) = s_o + s_1\, q^{-1} + \cdots + s_{n_s}\, q^{-n_s}$$

Controller parameters

Start with the Diophantine equation

$$A_c'(q^{-1}) = A(q^{-1})\,R'(q^{-1}) + q^{-\mathsf{d}}\,S(q^{-1})$$

Multiply both sides by *y(k)*

$$A_c'(q^{-1})\,y(k) = R'(q^{-1})\,A(q^{-1})\,y(k) + q^{-\mathsf{d}}\,S(q^{-1})\,y(k)$$

Controller parameters

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

Insert plant dynamics

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

$$A_c'(q^{-1})\,y(k) = q^{-\mathsf{d}}\left[\underline{R'(q^{-1})\,B(q^{-1})}\,u(k) + S(q^{-1})\,y(k)\right]$$

$$\boxed{A_c'(q^{-1})\,y(k) = q^{-\mathsf{d}}\left[R(q^{-1})\,u(k) + S(q^{-1})\,y(k)\right]}$$

PAA – version 1

$$A_c'(q^{-1})\,y(k) = q^{-\mathrm{d}}\left[R(q^{-1})\,u(k) + S(q^{-1})\,y(k)\right]$$

Filter by $1/A_c{}'(q^{-1})$ (*normally a low-pass filter*)

$$\boxed{y(k) = R(q^{-1})\,u_f(k-\mathrm{d}) + S(q^{-1})\,y_f(k-\mathrm{d})}$$

$$y_f(k) = \frac{1}{A_c'(q^{-1})}\,y(k)$$

$$u_f(k) = \frac{1}{A_c'(q^{-1})}\,u(k)$$

PAA – version 1

$$y(k) = R(q^{-1})\, u_f(k - \mathrm{d}) + S(q^{-1})\, y_f(k - \mathrm{d})$$

Is linear in the controller parameters:

$$\boxed{y(k) = \phi_f^T(k - \mathrm{d})\theta_c}$$

$$\theta_c = \begin{bmatrix} s_o & \cdots & s_{n_s} & r_o & \cdots & r_{n_r} \end{bmatrix}^T \in \mathcal{R}^{N_c}$$

$$N_c = n_s + n_r + 2$$

PAA – version 1

Plant dynamics:

$$\boxed{y(k) = \phi_f^T(k - \mathsf{d})\theta_c}$$

$$\theta_c = \begin{bmatrix} s_o & \cdots & s_{n_s} & r_o & \cdots & r_{n_r} \end{bmatrix}^T \in \mathcal{R}^{N_c}$$

$$\phi_f(k) \;=\; \frac{1}{A_c'(q^{-1})}\phi(k)$$

$$\phi(k) \;=\; \begin{bmatrix} y(k) & \cdots & y(k-n_s) & u(k) & \cdots & u(k-n_r) \end{bmatrix}^T$$

$$N_c = n_s + n_r + 2$$

PAA – version 1

Plant dynamics:

$$y(k) = \phi_f^T(k-\mathsf{d})\theta_c$$

RLS PAA:

$$e^o(k+1) = y(k+1) - \phi_f^T(k-\mathsf{d}+1)\hat{\theta}_c(k)$$

$$e(k+1) = \frac{\lambda_1(k)}{\lambda_1(k) + \phi_f^T(k-\mathsf{d}+1)F(k)\phi_f(k-\mathsf{d}+1)}e^o(k+1)$$

$$\hat{\theta}_c^o(k+1) = \hat{\theta}_c(k) + \frac{1}{\lambda_1(k)}F(k)\phi_f(k-\mathsf{d}+1)e(k+1)$$

$$F(k+1) = \frac{1}{\lambda_1(k)}\left[F(k) - \lambda_2(k)\frac{F(k)\phi_f(k-\mathsf{d}+1)\phi_f^T(k-\mathsf{d}+1)F(k)}{\lambda_1(k) + \lambda_2(k)\phi_f^T(k-\mathsf{d}+1)F(k)\phi_f(k-\mathsf{d}+1)}\right]$$

PAA – version 2

$$A_c'(q^{-1})\,y(k) = q^{-\mathrm{d}}\left[R(q^{-1})\,u(k) + S(q^{-1})\,y(k)\right]$$

$$\boxed{\eta(k) = A_c'(q^{-1})\,y(k)}$$

filtered output signal

$$\boxed{\eta(k) = \phi^T(k-\mathrm{d})\theta_c}$$

$$\theta_c = \begin{bmatrix} s_o & \cdots & s_{n_s} & r_o & \cdots & r_{n_r} \end{bmatrix}^T \in \mathcal{R}^{N_c}$$

$$\phi(k) \;=\; \begin{bmatrix} y(k) & \cdots & y(k-n_s) & u(k) & \cdots & u(k-n_r) \end{bmatrix}^T$$

PAA – version 2

Plant dynamics:

$$\boxed{\eta(k) = \phi^T(k - \mathsf{d})\theta_c}$$

RLS PAA:

$$e^o(k+1) = \eta(k+1) - \phi^T(k - \mathsf{d} + 1)\hat{\theta}_c(k)$$

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

$$\hat{\theta}_c^o(k+1) = \hat{\theta}_c(k) + \frac{1}{\lambda_1(k)} F(k)\phi(k - \mathsf{d} + 1)e(k+1)$$

$$F(k+1) = \frac{1}{\lambda_1(k)} \left[F(k) - \lambda_2(k) \frac{F(k)\phi(k - \mathsf{d} + 1)\phi^T(k - \mathsf{d} + 1)F(k)}{\lambda_1(k) + \lambda_2(k)\phi^T(k - \mathsf{d} + 1)F(k)\phi(k - \mathsf{d} + 1)} \right]$$

PAA – version 1 Vs version 2

- $A_c'(q^{-1})$ is normally a ***high-pass*** filter
- $1/A_c'(q^{-1})$ is normally a ***low-pass*** filter

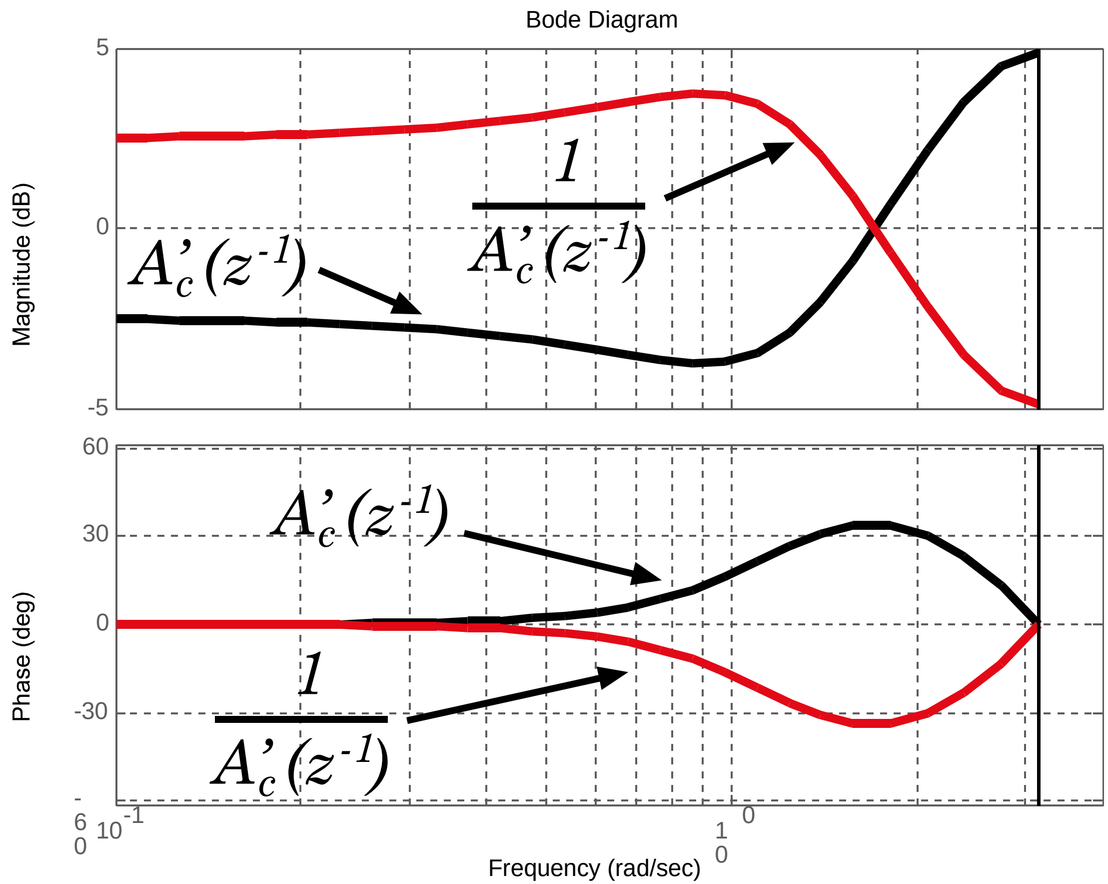
Example

$$A_c'(q^{-1}) = (1 - .5q^{-1})^2$$

Version 1 is preferable

$$\phi_f(k) = \frac{1}{A_c'(q^{-1})}\phi(k)$$

filters high frequency noise



PAA projection

PAA: Projection

$$\hat{r}_o(k)$$

$$\hat{\theta}_c(k) = \begin{cases} \hat{\theta}_c^o(k) & \text{if } \hat{r}_o^o(k) \geq b_{mino} \\ \left[\ \hat{s}_o^o(k) \cdots \hat{s}_{n_s}^o(k)\ b_{mino}\ \cdots\ \hat{r}_{n_r}^o(k)\ \right]^T & \text{if } \hat{r}_o^o(k) < b_{mino} \end{cases}$$

Replace $\hat{r}_o^o(k)$ by b_{mino} if it becomes too small.

Control law will divide by $\hat{r}_o(k)$. Thus, the projection algorithm prevents the control action from becoming too large.

PAA Gain matrix

Gain matrix:

$$F(k{+}1) = \frac{1}{\lambda_1(k)}\left[F(k) - \lambda_2(k)\frac{F(k)\phi_f(k-\mathsf{d}+1)\phi_f^T(k-\mathsf{d}+1)F(k)}{\lambda_1(k)+\lambda_2(k)\phi_f^T(k-\mathsf{d}+1)F(k)\phi_f(k-\mathsf{d}+1)}\right]$$

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

are adjusted so that the maximum singular value of $F(k)$ is uniformly bounded, and

$$0 < K_{\min} \leq \lambda_{\min}\{F(k)\} \leq \lambda_{\max}\{F(k)\} < K_{max} < \infty\,.$$

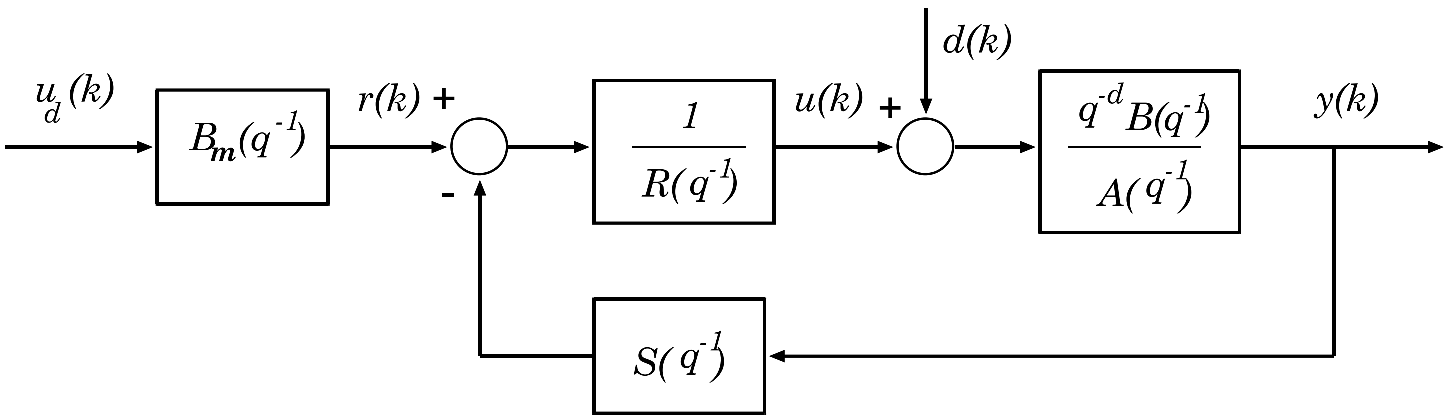
Direct Adaptive Control

1. Plants with minimum phase zeros and no disturbances:
   - Controller design

   1. Controller PAA

   2. Adaptive Controller

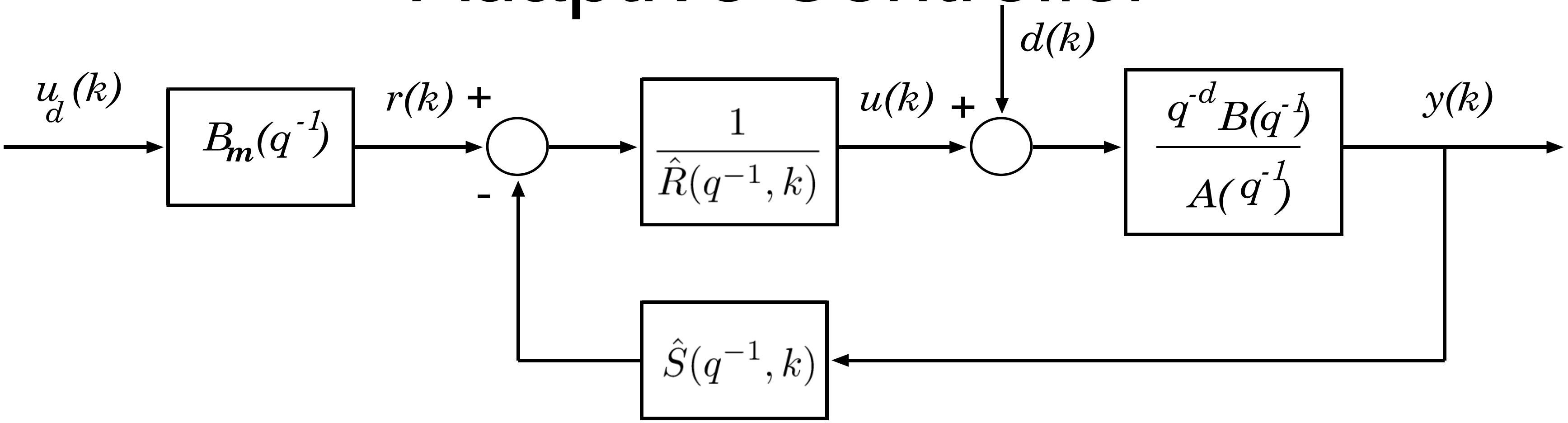
Fixed Controller



$$R(q^{-1})\,u(k) = B_m(q^{-1})\,u_d(k) - S(q^{-1})y(k)$$

Use this equation to solve for u(k)

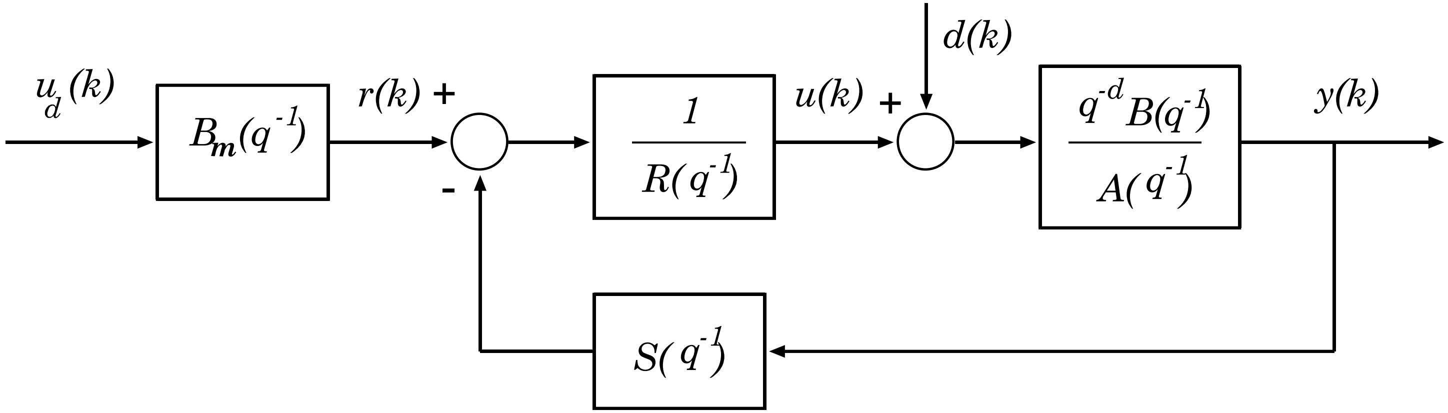
Adaptive Controller



$$\widehat{R}(q^{-1},k)\,u(k) = B_m(q^{-1})\,u_d(k) - \widehat{S}(q^{-1},k)y(k)$$

Use this equation to solve for u(k)

Fixed Controller

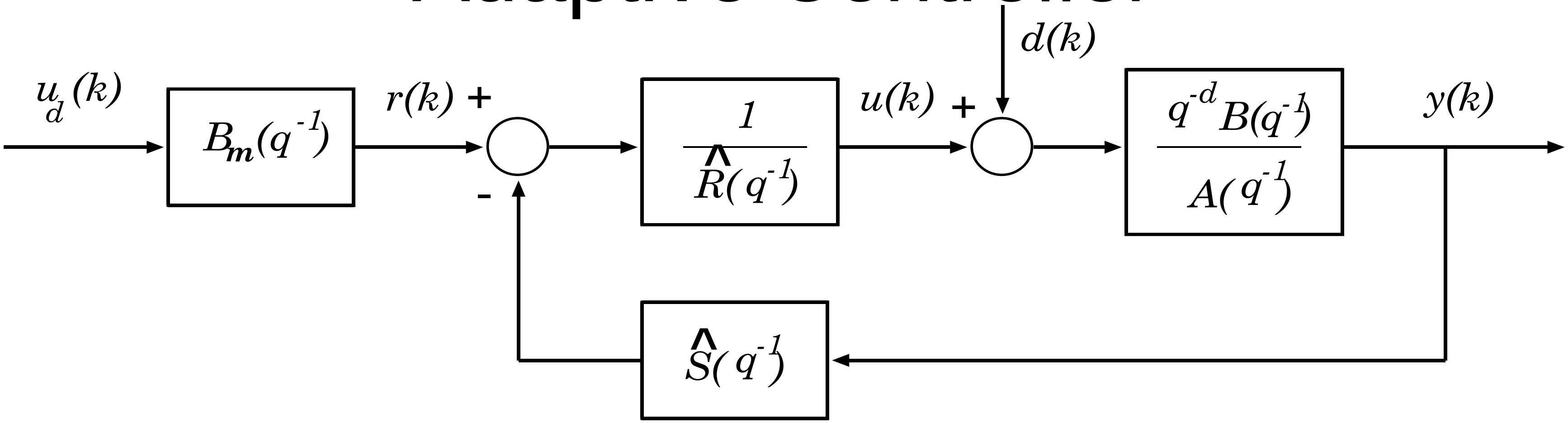


$$R(q^{-1})\,u(k) = B_m(q^{-1})\,u_d(k) - S(q^{-1})y(k)$$

$$S(q^{-1})y(k) + R(q^{-1})\,u(k) = B_m(q^{-1})\,u_d(k)$$

$$\boxed{\phi^T(k)\theta_c = B_m(q^{-1})u_d(k)}$$

Adaptive Controller



$$\widehat{R}(q^{-1}, k)\, u(k) = B_m(q^{-1})\, u_d(k) - \widehat{S}(q^{-1}, k) y(k)$$

$$\boxed{\phi^T(k)\widehat{\theta}_c(k) = B_m(q^{-1}) u_d(k)}$$

Direct Adaptive Control

1. Plants with minimum phase zeros and no disturbances:

    - Controller design

    1. Controller PAA
    2. Adaptive Controller

2. Plants with minimum phase zeros and constant disturbances:

- Read section: ***Direct adaptive control with integral action for plants with stable zeros*** in the ME233 class notes, part II.