EE5103/ME5403 Lecture Six
Predictive Control:
Solution to Tracking Problem

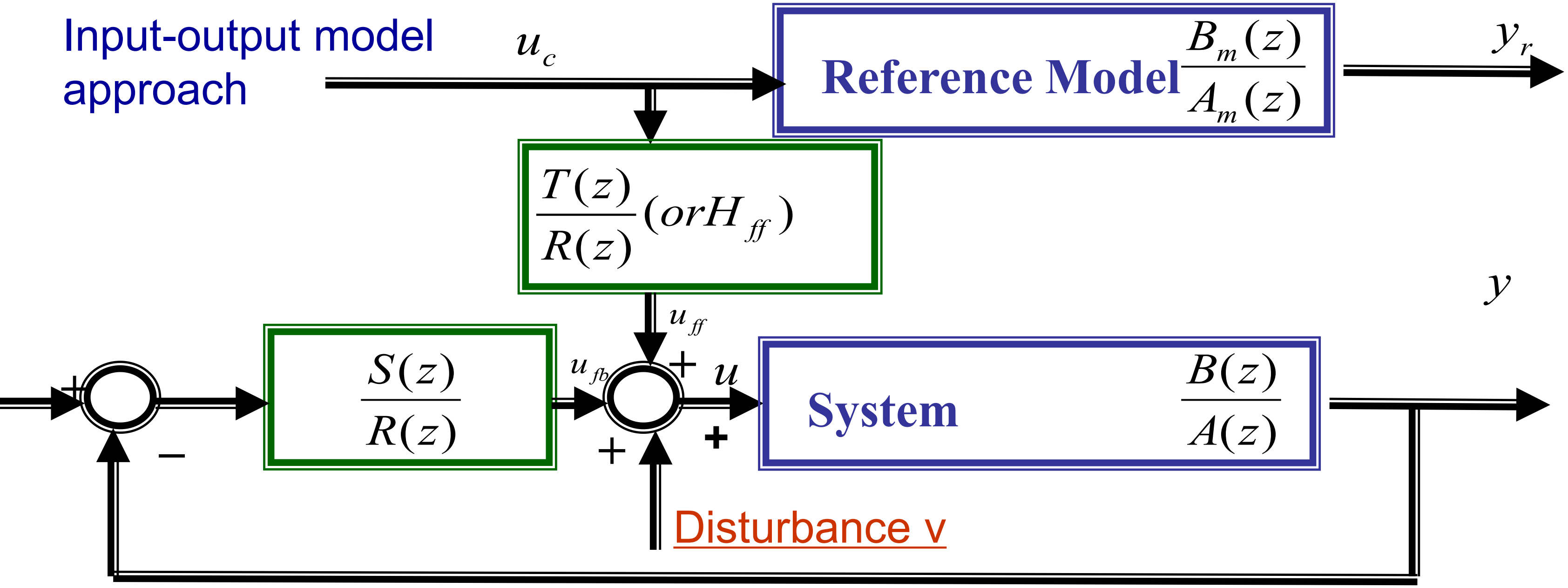
Xiang Cheng

Associate Professor

Department of Electrical & Computer Engineering
The National University of Singapore

Phone: 65166210 Office: Block E4-08-07
Email: elexc@nus.edu.sg

Input-output model approach



Occam's razor -- the simpler, the better.

Separation Property: Design feedback controller first, then build feedforward controller.

Step One: Figure out the design requirements on R(z).

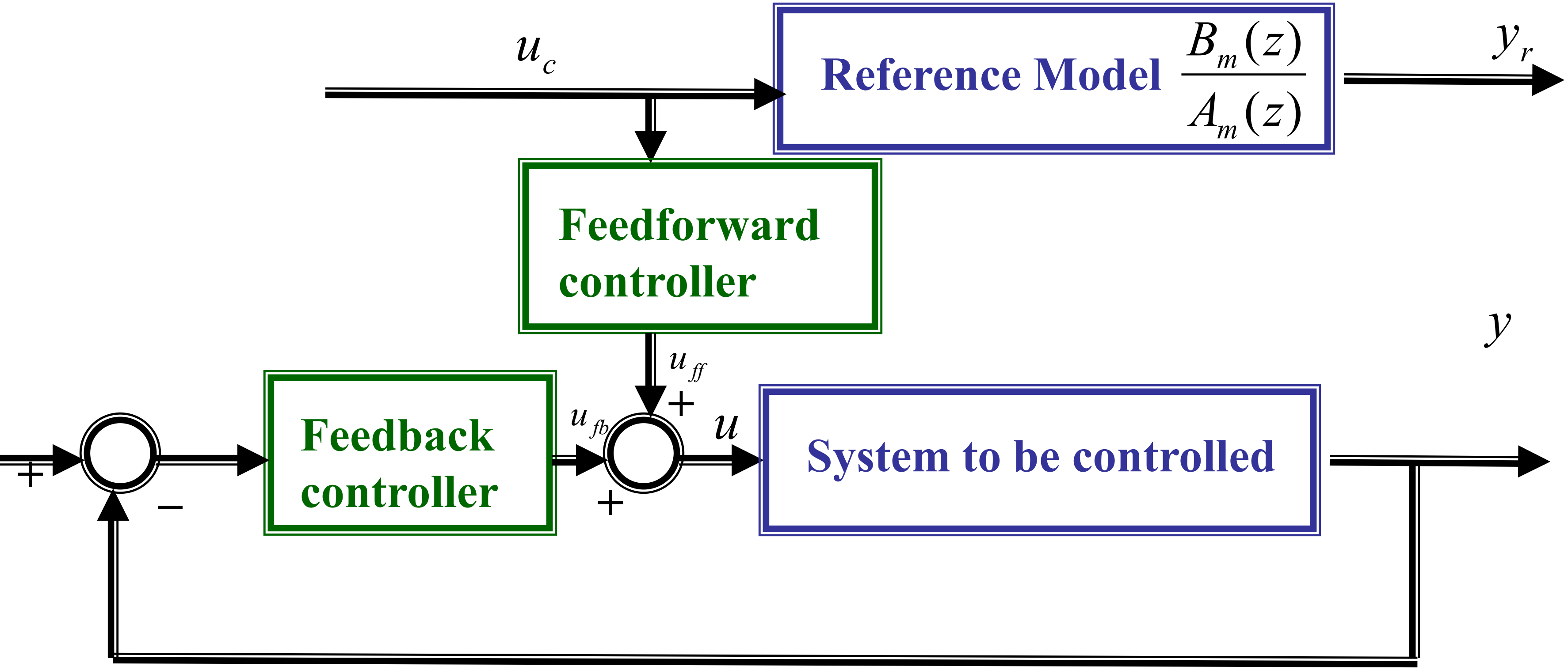
- If need to cancel zeros: R(z) must contain the zero factor, B(z)
- Disturbance rejection: R must contain the unstable factors related to the disturbance.
- If disturbance is constant, then R contain the unstable factor, (z-1).
- Causality conditions: Deg (R) >= Deg (S),

Step Two: Design R and S by solving the Diophantine equation $AR + BS = A_{cl}$

Step Three: Choose T or H_{ff} at the final stage, to satisfy other design requirements.

The order of controller can be increased in order to meet other design requirements.

Two-Degree-of-Freedom Controller

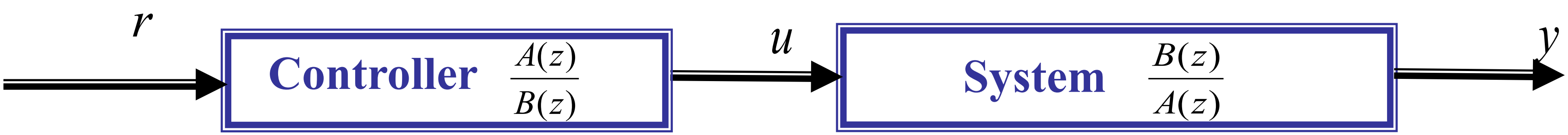


Objective: To make the closed loop model follow the reference model as close as possible.

This approach can be applied to most of the linear systems with models available.

- Can we make the output follow arbitrary command signal directly without using the reference model?

The answer is yes in theory, but with a price to pay.



•The inverse problem approach:

Assuming the desired output r(k) can be obtained, how to find out the corresponding input?

$$Y(z) = \frac{B(z)}{A(z)}U(z) \xrightarrow{Y(z)=R(z)} R(z) = \frac{B(z)}{A(z)}U(z)$$

•Controller:

$$U(z) = \frac{A(z)}{B(z)}R(z)$$

•Example: First order system

$$Y(z) = \frac{1}{z+a}U(z)$$

$$y(k+1) + ay(k) = u(k)$$

Let: $y(k) = r(k)$

$$r(k+1) + ar(k) = u(k)$$

•Controller:

$$u(k) = r(k+1) + ar(k)$$

•Does it work?

•**What is the T.F. from r to y?**

$$\frac{A(z)}{A(z)} = \frac{z+a}{z+a}$$

$$y(k+1) + ay(k) = u(k) \quad = r(k+1) + ar(k) \Longrightarrow Y(z) = \frac{z+a}{z+a}R(z)$$

•**When does it work? Or when can you cancel out the poles and zeros?** Stable!

•**When does it fail?** If the system is unstable!

•**Is it open loop controller, or a feedback controller?** Open loop controller

$$y(k+1)+ay(k)=u(k)$$ Open loop controller: $$u(k)=r(k+1)+ar(k)$$

How to change it into a feedback controller using the output signal?

Use the real output y(k) instead of the desired output r(k) wherever possible!

Feedback controller: $$u(k)=r(k+1)+ay(k)$$

Plug it into the system equation: $$y(k+1)+ay(k)=u(k)=r(k+1)+ay(k)$$

$$y(k+1)=r(k+1)$$

It works now!

What is the trick?

Re-write the controller: $$r(k+1)=-ay(k)+u(k)$$

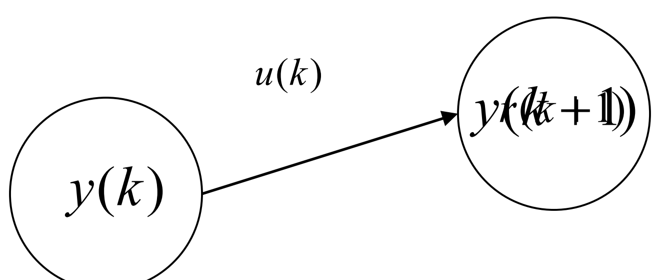
- If we re-write the system equation: $$y(k+1)=-ay(k)+u(k)$$

We can consider it as a predictor:

Use the present and past history to predict the future!

What control action at the present would bring the future output, y(k+1), to a desired value, r(k+1)?

The control input can be found out by matching predicted y(k+1) with r(k+1)!



General Case:

$$u \longrightarrow \boxed{\textbf{System} \quad \frac{B(z)}{A(z)}} \longrightarrow y$$

$$U \to \frac{B(z)}{A(z)} \to Y \quad \Longrightarrow \quad U \to \frac{B(z^{-1})}{A(z^{-1})} \to Y$$

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

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

$$B(q^{-1}) = q^{-d}(b_0 + b_1 q^{-1} + \cdots b_{n_1} q^{-n_1}), \qquad b_0 \neq 0$$

$$= q^{-d} B'(q^{-1})$$

Example: $$\frac{2}{z^2+1} \quad \Longrightarrow \quad \frac{z^{-2}2}{z^{-2}(z^2+1)} = \frac{2z^{-2}}{1+z^{-2}}$$

How to write down the difference equation?

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-d-1) + \cdots + b_{n_1} u(k-d-n_1)$$

•**What is the physical meaning of the time delay: d?**

•It is the time that you have to wait for the system to respond!

One-Step-Ahead Control: The case of time delay d=1

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n)=b_0u(k-1)+b_1u(k-2)+\cdots+b_{n_1}u(k-n_1-1)$$

Write down the equation in the predictor form:

$$y(k+1)+a_1y(k)+\cdots+a_ny(k-n+1)=b_0u(k)+b_1u(k-1)+\cdots+b_{n_1}u(k-n_1)$$

$$y(k+1)=-a_1y(k)+\cdots-a_ny(k-n+1)+b_0u(k)+b_1u(k-1)+\cdots+b_{n_1}u(k-n_1)$$

How to choose u(k) such that y(k+1)=r(k+1)?

$$r(k+1)=-a_1y(k)+\cdots-a_ny(k-n+1)+b_0u(k)+b_1u(k-1)+\cdots+b_{n_1}u(k-n_1)$$

Can you compute u(k) from above equation?

Controller: $$u(k)=\frac{1}{b_0}[r(k+1)+a_1y(k)+\cdots+a_ny(k-n+1)-b_1u(k-1)+\cdots-b_{n_1}u(k-n_1)]$$

$$y(k+1)=r(k+1)$$

Isn't it simple?

Yes.

How about d>1? Let's consider a simple example first:

$$y(k) - y(k-1) = u(k-2)$$

What is d? d=2

We want to have an equation with u(k) since we want to use it to compute the input.

How to convert the above equation so that it contains the input u(k)?

Forward shift both sides by 2! $y(k+2) = y(k+1) + u(k)$

Let $y(k+2) = r(k+2)$

$$r(k+2) = y(k+1) + u(k) \Longrightarrow u(k) = r(k+2) - y(k+1)$$

Can we implement the controller in reality?

No.

What's going wrong?

Is $y(k+2) = y(k+1) + u(k)$ in the predictor form?

No! We cannot use the future output to predict the future!

$y(k+2) = y(k+1) + u(k)$ is not a predictor!

$$y(k+2) = y(k+1) + u(k)$$ is not a predictor!

How to turn it into a predictor?

We need to deal with y(k+1) on the right hand side.

• **Can we predict the future value y(k+1) using the present and past information?**

Just backward shift the above equation by one!

$$y(k+1) = y(k) + u(k-1)$$

Then just replace y(k+1) with its prediction, we have

$$y(k+2) = y(k+1) + u(k) \qquad = y(k) + u(k-1) + u(k)$$

Is it a predictor now? Yes!

The next step is simple: just try to match the predicted output with the desired one!

$$y(k+2) = r(k+2)$$

$$r(k+2) = y(k) + u(k-1) + u(k) \quad \Longrightarrow \quad u(k) = r(k+2) - y(k) - u(k-1)$$

General Case: $A(q^{-1})y(k) = B(q^{-1})u(k) = q^{-d}B'(q^{-1})u(k)$

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-d-1) + \cdots + b_{n_1} u(k-d-n_1)$$

Can we express the output of the system at time t + d in the following predictor form?

$$y(k+d) = \alpha_0 y(k) + \cdots + \alpha_{n1} y(k-n_1) + \beta_0 u(k) + \cdots + \beta_{n2} u(k-n_2)$$

$$y(k+d) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k)$$

$$\alpha(q^{-1}) = \alpha_0 + \cdots \alpha_{n_1} q^{-n_1}$$

$$\beta(q^{-1}) = \beta_0 + \beta_1 q^{-1} + \cdots \beta_{n_2} q^{-n_2}, \beta_0 \neq 0$$

Predictor form: Only present and past values of the inputs and outputs (RHS) are used to predict the future (LHS)!

General Case:

$$y(k)+a_1 y(k-1)+\cdots+a_n y(k-n)=b_0 u(k-d)+b_1 u(k-d-1)+\cdots+b_{n_1} u(k-d-n_1)$$

We want to have an equation with u(k) since we want to use it to compute the input.

So we simply rewrite the original equation by forward shift of d steps:

$$y(k+d)+a_1 y(k+d-1)+\cdots+a_n y(k+d-n)=b_0 u(k)+b_1 u(k-1)+\cdots+b_{n_1} u(k-n_1)$$

$$y(k+d)=-a_1 y(k+d-1)\cdots-a_n y(k+d-n)+b_0 u(k)+b_1 u(k-1)+\cdots+b_{n_1} u(k-n_1)$$

Is this in the predictor form?

No, it has future outputs y(k+1), y(k+2), …, y(k+d-1)!

Is it possible to predict y(k+1)?

$$y(k+1)=-a_1 y(k)\cdots-a_n y(k-n+1)+b_0 u(k-d+1)+\cdots+b_{n_1} u(k-n_1-d+1)$$

How about y(k+2)?

$$y(k+2)=-a_1 y(k+1)-a_2 y(k)\cdots-a_n y(k-n+1)+b_0 u(k-d+2)+\cdots+b_{n_1} u(k-n_1-d+2)$$

Is it in the predictor form? No.

How to turn it into predictor?

Replace y(k+1) with its predictor form!

$$y(k+d-1) = -a_1 y(k+d-2)\cdots - a_n y(k+d-n-1) + b_0 u(k-1) + \cdots + b_{n_1} u(k-n_1-1)$$

How to predict y(k+d-1)?, how to deal with the future values of the outputs?

Replace y(k+1), y(k+2), y(k+3), .., y(k+d-2) with their predictor forms!

$$y(k+d) = -a_1 y(k+d-1) + \cdots - a_n y(k+d-n) + b_0 u(k) + b_1 u(k-1) + \cdots + b_{n_1} u(k-n_1)$$

We can always turn it to a predictor form by replacing the future values with their predictions:

$$y(k+d) = \alpha(q^{-1}) y(k) + \beta(q^{-1}) u(k)$$

$$y(k+d) = \alpha_0 y(k) + \cdots + \alpha_{n1} y(k-n_1) + \beta_0 u(k) + \cdots + \beta_{n2} u(k-n_2)$$

How to compute the input u(k)?

Match y(k+d) with the desired one r(k+d)!

$$r(k+d) = \alpha_0 y(k) + \cdots + \alpha_{n1} y(k-n_1) + \beta_0 u(k) + \cdots + \beta_{n2} u(k-n_2)$$

Let's check whether all the signals are bounded.

The resulting closed-loop system is described by

$$y(k+d) = r(k+d) \Longrightarrow y(k) = r(k); \quad k \geq d$$

$$A(q^{-1})y(k) = B(q^{-1})u(k) \Longrightarrow B(q^{-1})u(k) = A(q^{-1})r(k); \quad k \geq d+n$$

$$U(z) = \frac{A(z^{-1})}{B(z^{-1})} R(z)$$

What is the condition to guarantee that the input is bounded?

- **B(z) is stable!**

Do we need this condition for pole-placement or stabilization? No.

Do we need this for perfect match of the reference model? Yes.

- **The stable inverse condition is always the price to pay for perfect tracking!**

The idea of predictive control is not only simple, but can also be directly applied to nonlinear system.

Consider the following nonlinear system:

$$y(k+1) = ay^2(k) + bu(k) + nu(k)y(k)$$

Is it in the form of the predictor? Sure!

Match the predicted output with the desired output:

$$r(k+1) = ay^2(k) + bu(k) + nu(k)y(k)$$

Controller:

$$u(k) = \frac{r(k+1) - ay^2(k)}{b + ny(k)}$$

What is the output at k+1?

$$y(k+1) = r(k+1)$$

Can you design pole-placement controller directly for this system?

No. You cannot do it directly. You have to linearize it first.

Can we make the output follow arbitrary command signal without the reference model?

It is possible to solve the control problem without the reference model.

Did we put any constraint on the desired output r(k) in the solution?

No. The desired output r(k) can be arbitrary.

What is the condition for perfect tracking? B(z) is stable!

Does it mean that we can make this type of system do whatever we want?

Can you make the car run faster than the jet plane?

No. we cannot. **we did not put any limit on the cost of the control! The resulting control signal may be too large to implement!**

In theory, both 1 and 1,000,000 are bounded.
In real world, an input of 1 might be attainable, while 1,000,000 might be impossible.

The one-step-ahead feedback control law only minimizes the squared prediction error:

$$J_1(k+d) = \frac{1}{2}[y(k+d) - r(k+d)]^2$$

In reality, we need to achieve
a compromise between bringing y(k + d) to r(k + d) and the cost of the control effort.

This is the starting point of so called “model predictive control”(MPC),
which is one of the main topics in part II!

Break

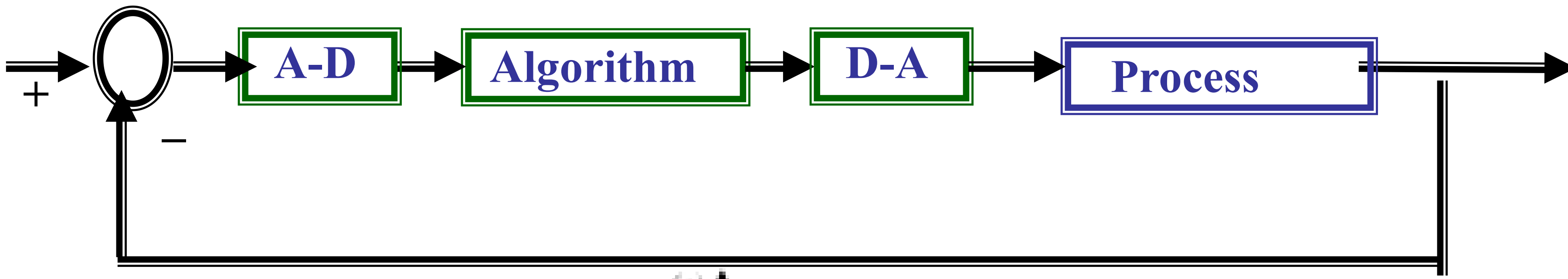
next generation robots

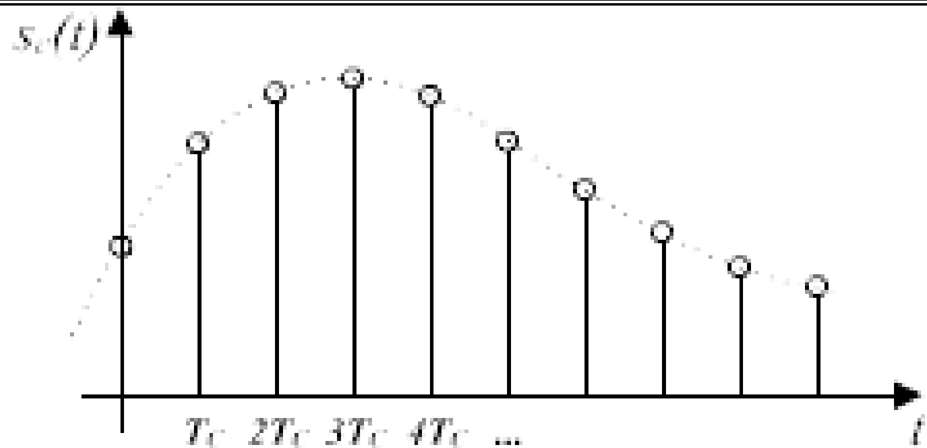
Review of Part I: Lectures One to Six

- In the first lecture, we revisited some of the fundamental concepts for systems and control. Most of them should be familiar to you except the concept of "state".

What is the state of a dynamic system?

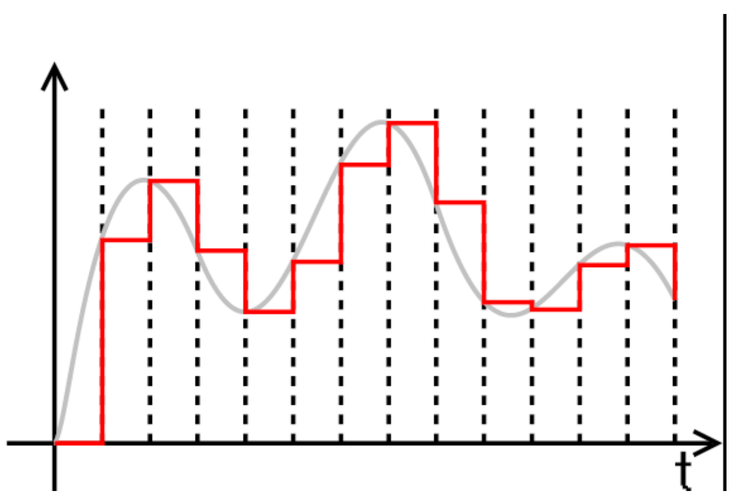
State at present: The information needed to predict the future assuming the current and all the future inputs are available.





<u>How to do the A-D?</u>

Uniform Sampling



<u>How to do the D-A?</u>

Zero-order Hold

State-space Model

$$\dot{x} = Ax + bu$$
$$y = cx$$

Laplace Transform

$$\Longrightarrow \quad sX(s) - x(0) = AX(s) + bU(s)$$

$$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}bU(s)$$

Inverse Laplace Transform

$$\Longrightarrow \quad x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}bu(\tau)d\tau$$

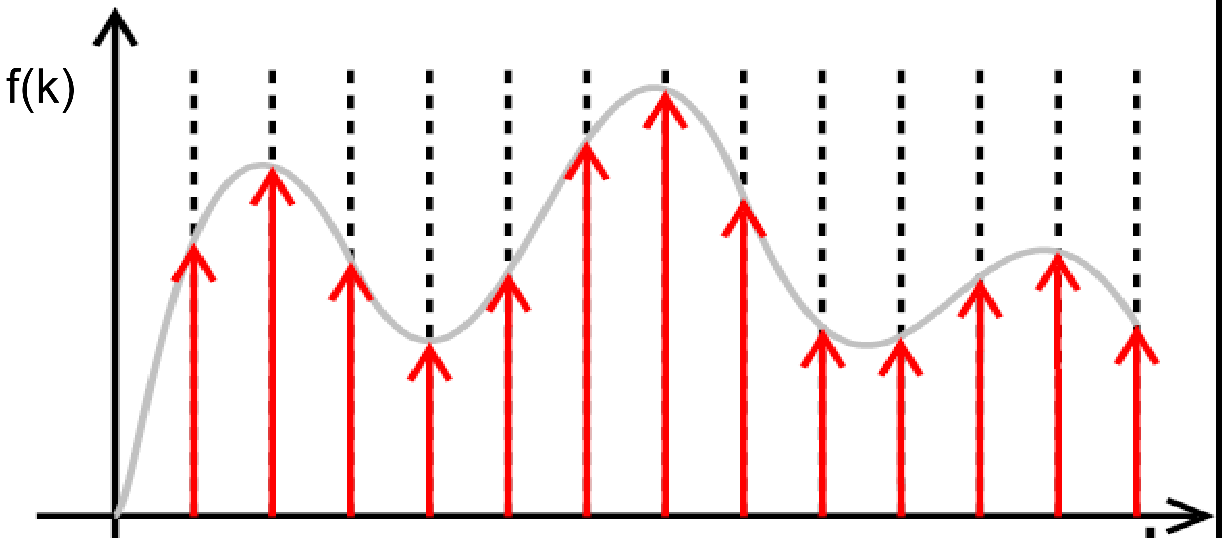
Zero-order hold

$$\Longrightarrow \quad x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$

$$\Phi = e^{Ah}$$

$$\Gamma = \left(\int_0^h e^{As}ds\right)b$$

How to calculate e^{At} ? $\quad (sI - A)^{-1} \Longrightarrow e^{At}$



Z-transform --- Equivalent of Laplace Transform!

$$F(z) = \sum_{k=0}^{\infty} f(k) z^{-k} = f(0) + f(1) z^{-1} + f(2) z^{-2} + \cdots + f(k) z^{-k} + \cdots$$

Most important property: how to relate x(k+1) and x(k) in z-domain

x(0)=0

$$z\{x(k+1)\} = z(X(z) - x(0)) \Longrightarrow z\{x(k+1)\} = zX(z)$$

$$q(forward - time - shift) \Leftrightarrow z$$

$$q^{-1}(backward - time - shift) \Leftrightarrow z^{-1}$$

Let's apply z-transform to the state-space equation

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$

$$\Longrightarrow$$

$$z(X(z) - x(0)) = \Phi X(z) + \Gamma U(z)$$
$$zX(z) - \Phi X(z) = zx(0) + \Gamma U(z)$$

$$X(z) = (zI - \Phi)^{-1} zx(0) + (zI - \Phi)^{-1} \Gamma U(z)$$

$$Y(z) = cX(z) = c(zI - \Phi)^{-1} zx(0) + c(zI - \Phi)^{-1} \Gamma U(z)$$

• <u>If the initial conditions are zero, x(0)=0</u>

$$Y(z) = c(zI - \Phi)^{-1} \Gamma U(z) = H(z)U(z)$$

• **<u>What is the transfer function?</u>**

$$U(z) \longrightarrow \boxed{H(z)} \longrightarrow Y(z)$$

$$H(z) = c(zI - \Phi)^{-1} \Gamma = \frac{Q(z)}{P(z)}$$

$$P(z) = \det\{zI - \Phi\}$$

$$U(z) \longrightarrow \boxed{H(z)} \longrightarrow Y(z)$$

<u>What can you do with the transfer functions?</u>

- Given a transfer function, you can write down the difference equations, and vice versa

$$q(forward - time - shift) \Leftrightarrow z$$

- Calculate the output, Y(z)=H(z)U(z) when initial conditions are zero.

But in general,
$$Y(z) = \underbrace{c(zI - \Phi)^{-1} zx(0)}_{\text{zero-input response}} + \underbrace{H(z)U(z)}_{\text{zero-state response}}$$

- Transfer function is the impulse response in z-domain.

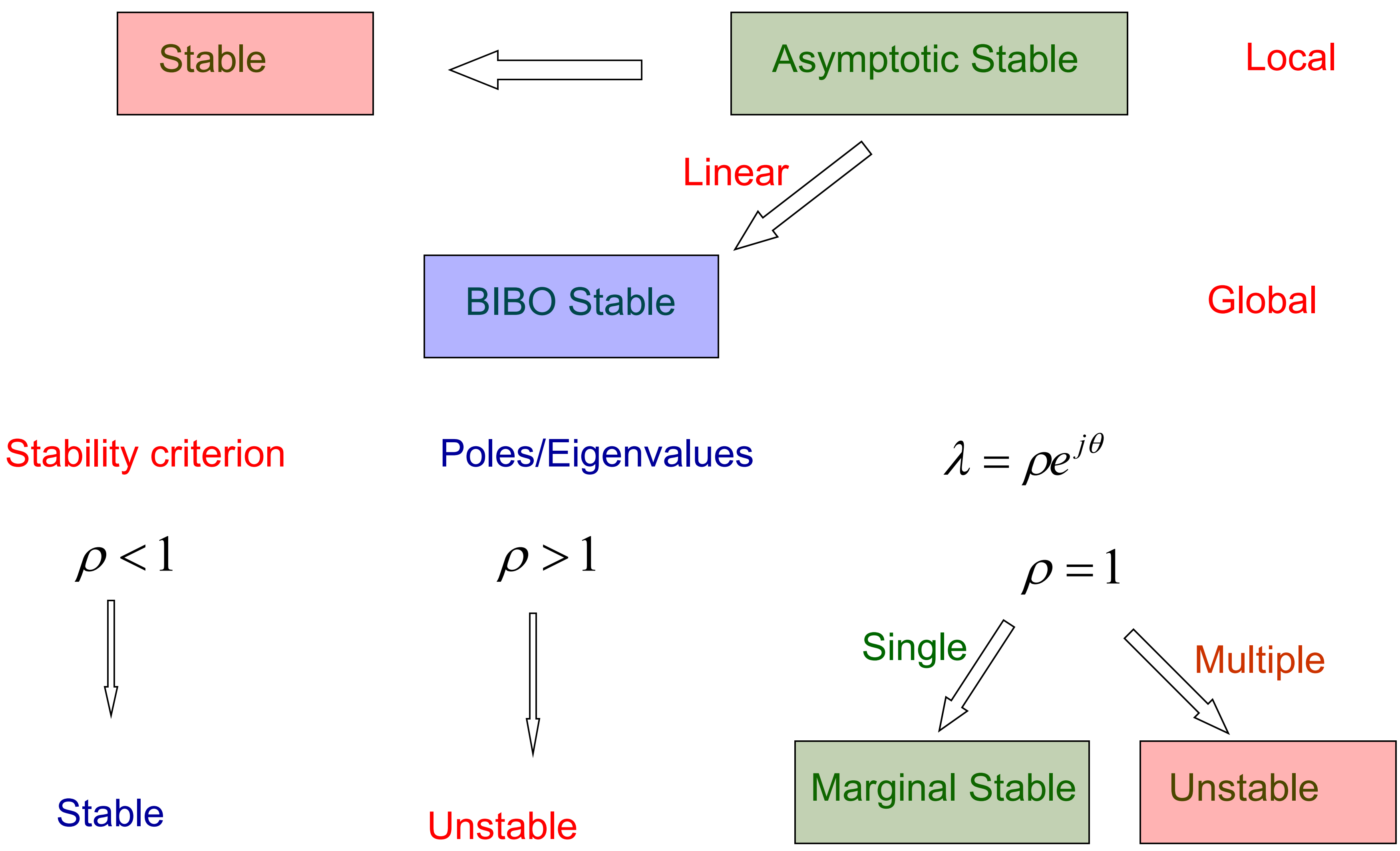
- The stability can be determined by checking poles of the transfer function.

- For stable system, some responses can be found from the response to the exponential:

$$a = 1 \Longrightarrow$$ Steady state gain: $H(1)$

$$a^k \longrightarrow \boxed{H(z)} \longrightarrow H(a)a^k \qquad H(a) = 0 \Longrightarrow$$ Signal blocking property! Disturbance Rejection!

Relationship between stability concepts



•How to check the stability without solving the equation?

JURY'S STABILITY TEST: $$A(z) = a_0 z^n + a_1 z^{n-1} + \ldots + a_n = 0$$

Get the coefficients: $$a_0 \quad a_1 \quad \cdots \quad a_{n-1} \quad a_n$$

Reverse the order: $$a_n \quad a_{n-1} \quad \cdots \quad a_1 \quad a_0 \qquad \times \frac{a_n}{a_0}$$

Eliminate the last element a_n $$a_0^{n-1} \quad a_1^{n-1} \quad \cdots \quad a_{n-1}^{n-1}$$

Repeat the process $$a_{n-1}^{n-1} \quad a_{n-2}^{n-1} \quad \cdots \quad a_0^{n-1} \qquad \times \frac{a_{n-1}^{n-1}}{a_0^{n-1}}$$

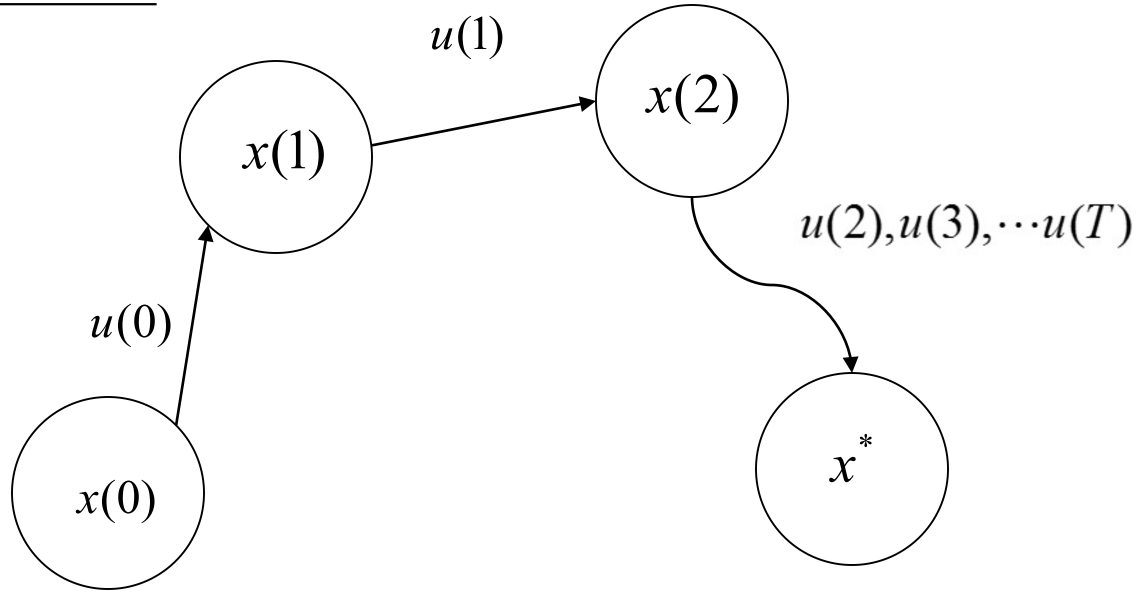
Stop when there is only one element left a_0^0

The system is stable if all the first elements

$$a_0 \quad a_0^{n-1} \quad a_0^{n-2} \quad \cdots \quad a_0^0$$ are positive!

Controllability

Is it possible to steer a system from any given initial state to any other state in finite time?



Controllability Matrix $\quad W_c = \left[\Gamma \quad \Phi\Gamma \quad \cdots \quad \Phi^{n-1}\Gamma\right]$

Geometrical Interpretation: $\{\Gamma, \Phi\Gamma, \Phi^2\Gamma, \cdots \Phi^{n-1}\Gamma\}$ are all the possible directions to go!

The Cayley-Hamilton Theorem !

$$\Phi^n = -a_1\Phi^{n-1} - a_2\Phi^{n-2} + \cdots - a_n I$$

Observability: $u(k) \longrightarrow \boxed{\quad\quad} \longrightarrow y(k)$

What is the state x(k)?

Can we determine the state x(k) from observations of inputs and outputs?

$$
\begin{aligned}
y(0) &= Cx(0) \\
y(1) &= Cx(1) = C\Phi x(0) \\
&\vdots \\
y(n-1) &= C\Phi^{n-1}x(0)
\end{aligned}
\quad\Longrightarrow\quad
\begin{pmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{pmatrix} x(0) = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{pmatrix}
$$

The state x(0) can be obtained if and only if

The observability matrix $W_0 = \begin{pmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{pmatrix}$ is non-singular (of full rank)

What is the geometrical interpretation of this observability matrix?

It is the set of all possible directions that the state vectors are projected.

If it can span the whole space, it means that we have the information of the state vector for all the directions, and the state can be reconstructed.

If it cannot span the whole space, the state information along certain direction will never be available. Thus the system is not observable.

•Controllable Canonical Form

$$z(k+1)=\begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} z(k) + \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(k)$$

$$y(k) = (b_1 \ \ldots \ b_n \ ) \, z(k)$$

Always Controllable

•Observable Canonical Form

$$z(k+1)=\begin{pmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{pmatrix} z(k) + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} u(k)$$

$$y(k) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} z(k)$$

Always Observable

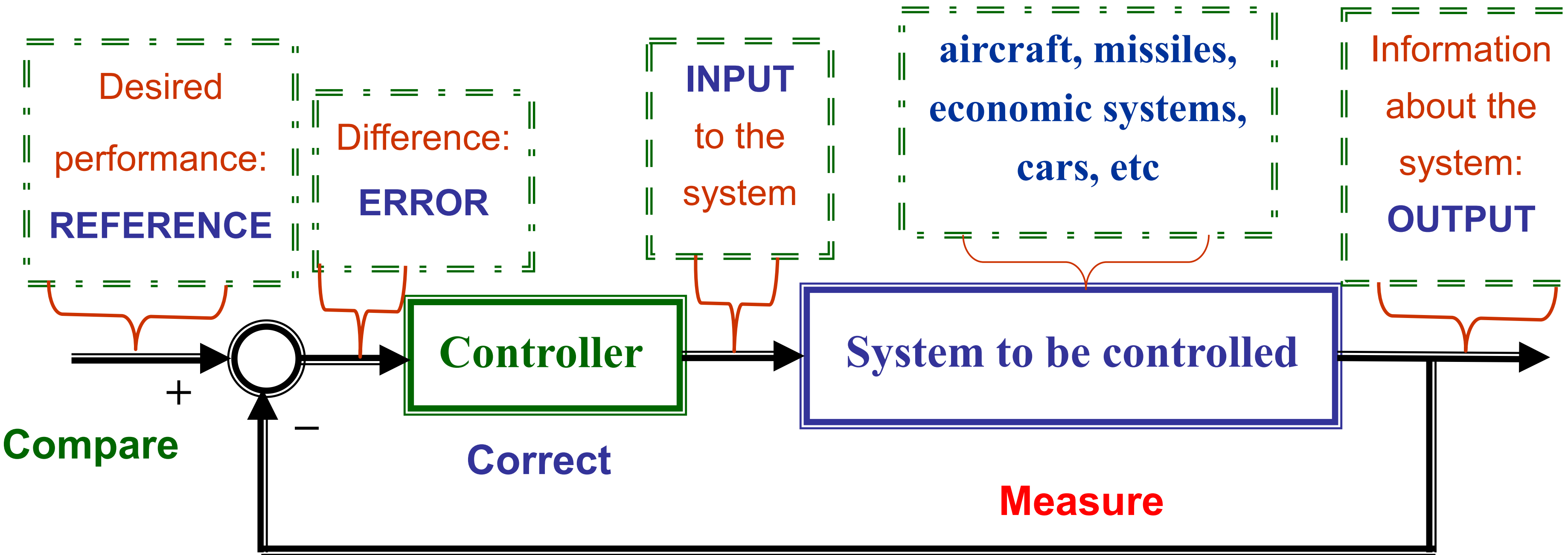
•The transfer function:

$$H(z) = \frac{b_1 z^{n-1} + b_2 z^{n-2} + \cdots + b_n}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_n}$$

Are these two canonical forms equivalent to each other?

Not always. Only if the system is both controllable and observable!

•What is a control system?



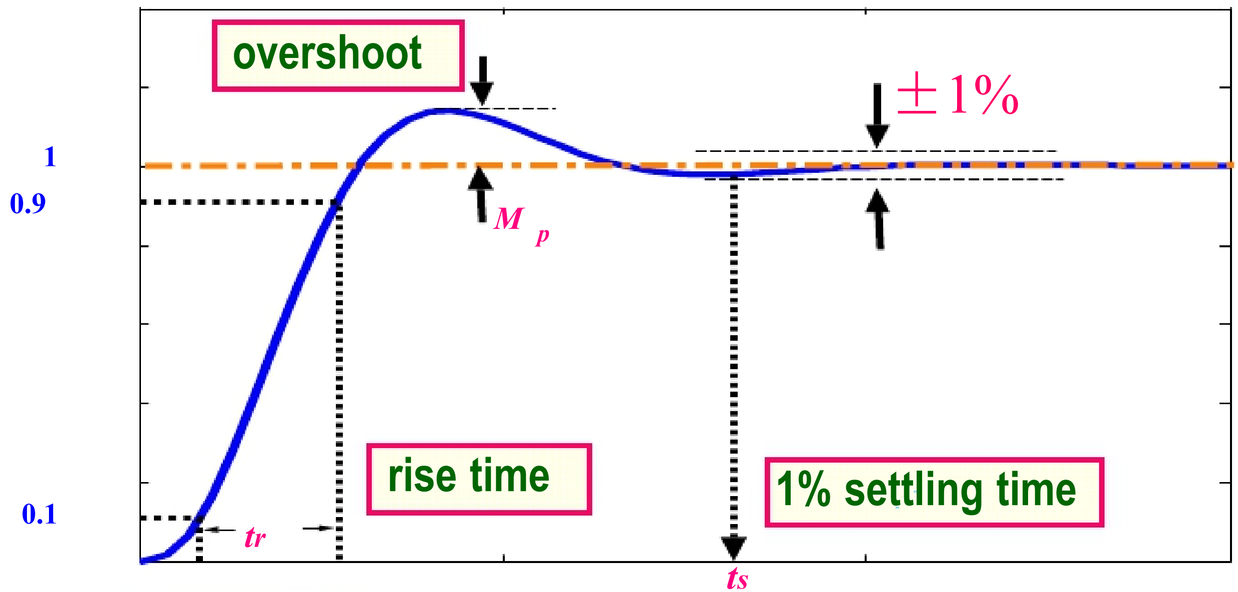
Objective: To make the system OUTPUT and the desired REFERENCE as close as possible, i.e., to make the ERROR as small as possible.

•**Feedback: Measure —Compare –-Correct**

How to specify the reference signal, or the desired output?

•The reference signal is specified by a well-behaved reference model.

Settling time, overshoot and rise time — Continuous time



$$t_r \cong \frac{1.8}{\omega_n}$$

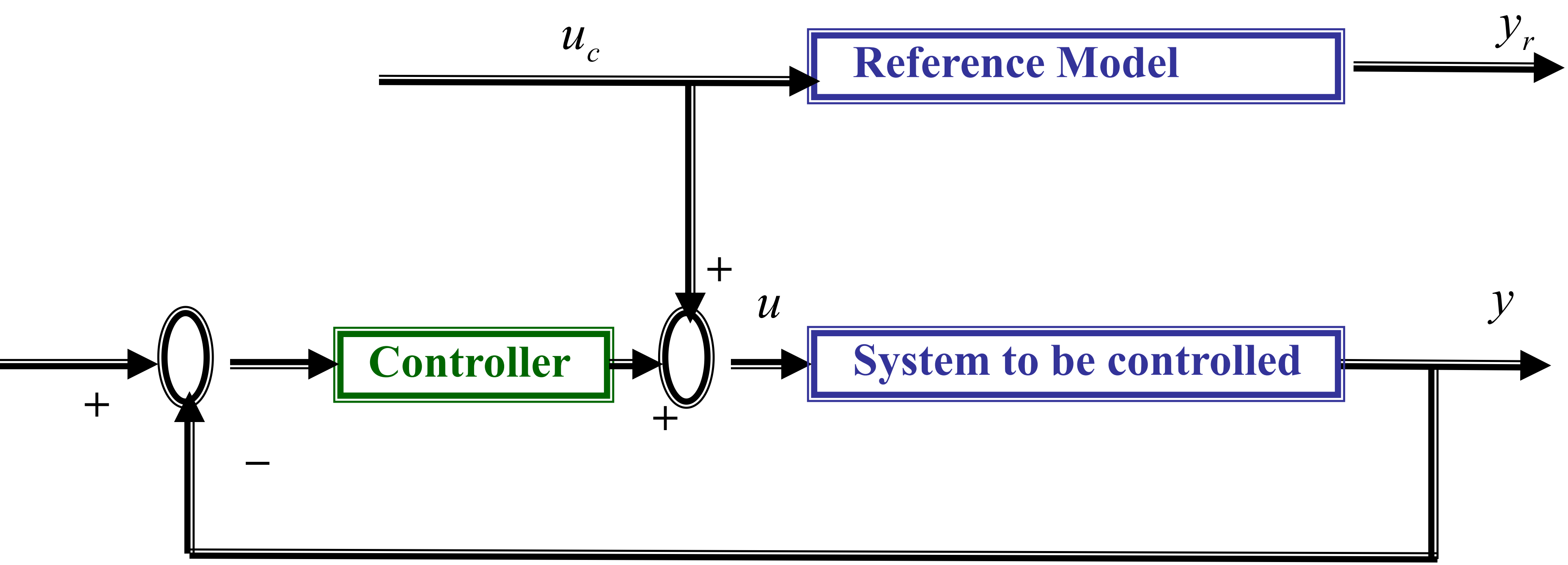
$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$t_s \cong \frac{4.6}{\zeta\omega_n}$$

$(t_s, M_p, t_r) \Leftrightarrow (\zeta, \omega_n) \Leftrightarrow$ Reference Model

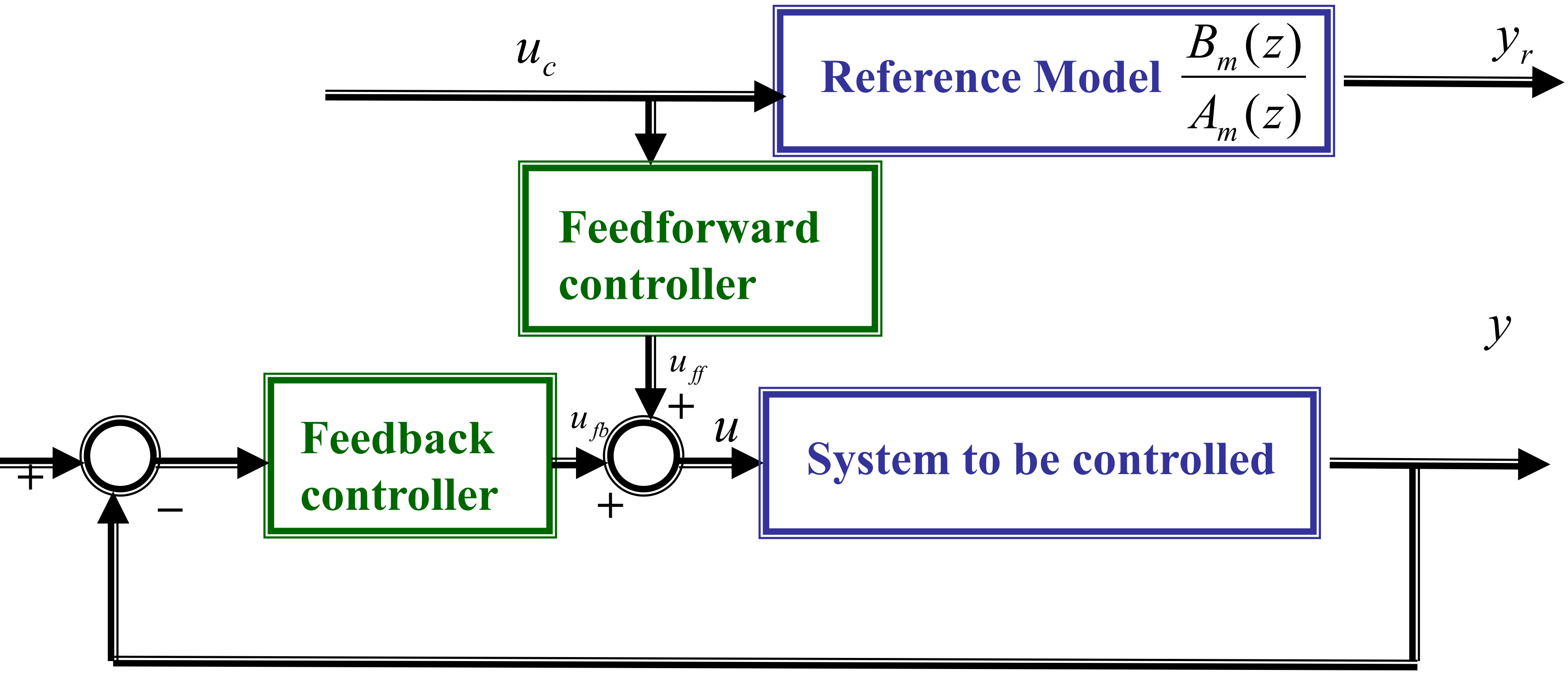
$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \Rightarrow \quad H_m(z) = \frac{B_m(z)}{A_m(z)}$$

• **Model-reference control:**



Objective: To make the closed loop model follow the reference model as close as possible.

Two-Degree-of-Freedom Controller

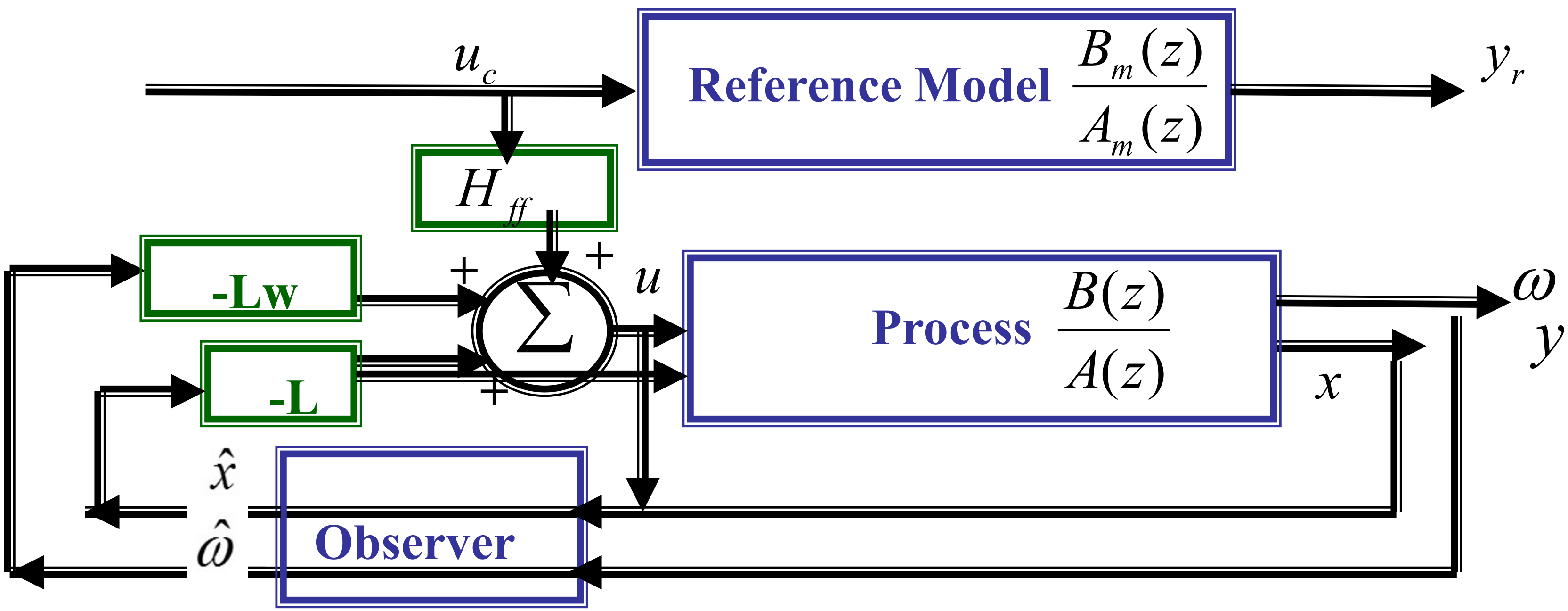


$$u = u_{fb} + u_{ff}$$

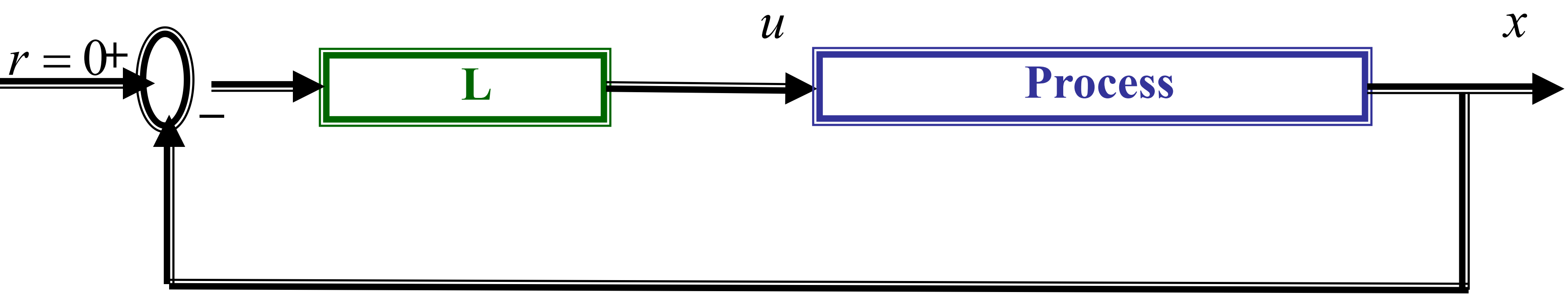
Pole placement	Zero placement
Match $A_m(z)$	**Match** $B_m(z)$

State-space approach

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$



Step One: Proportional Control --- State Feedback Control



A very simple controller: $$u(k) = -Lx(k)$$

Closed Loop System: $$x(k+1) = \Phi x(k) + \Gamma u(k) = (\Phi - \Gamma L)x(k)$$

If the system is controllable, the poles can be placed at anywhere:

- Ackermann's formula $$L = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} W_c^{-1} A_m(\Phi)$$

Can you do it without Ackermann's formula?

$$|zI - (\Phi - \Gamma L)| \Longrightarrow A_m(z)$$

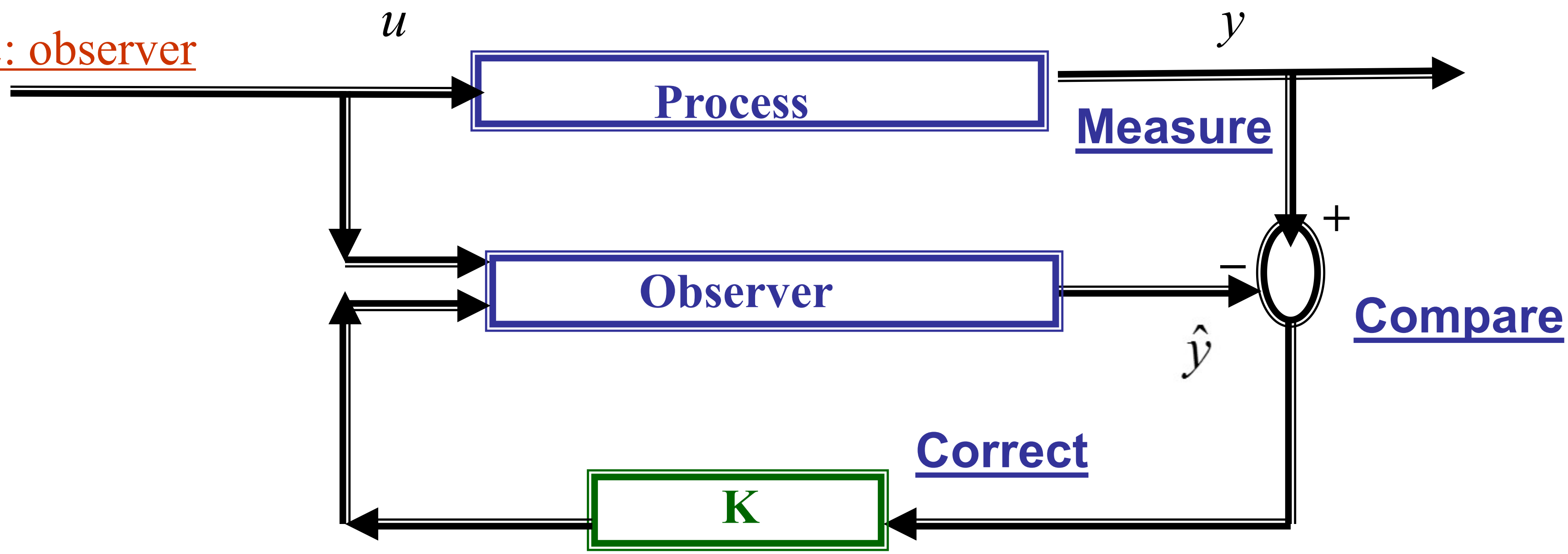
If the system is uncontrollable, can we still assign the poles to any desired locations?

Some of the poles cannot be changed!

- If the uncontrollable poles are stable, then we can still control the system to some degree.

- Stabilizable.

Step 2: observer



$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = c\hat{x}(k)$$

$$x(k+1) - \hat{x}(k+1) = \Phi(x(k) - \hat{x}(k)) - K(y(k) - \hat{y}(k))$$

$$e(k+1) = (\Phi - Kc)e(k)$$

How to choose K?

- Ackermann's formula

$$K = A_o(\Phi)(W_o)^{-1}[0 \quad \cdots \quad 0 \quad 1]^T$$

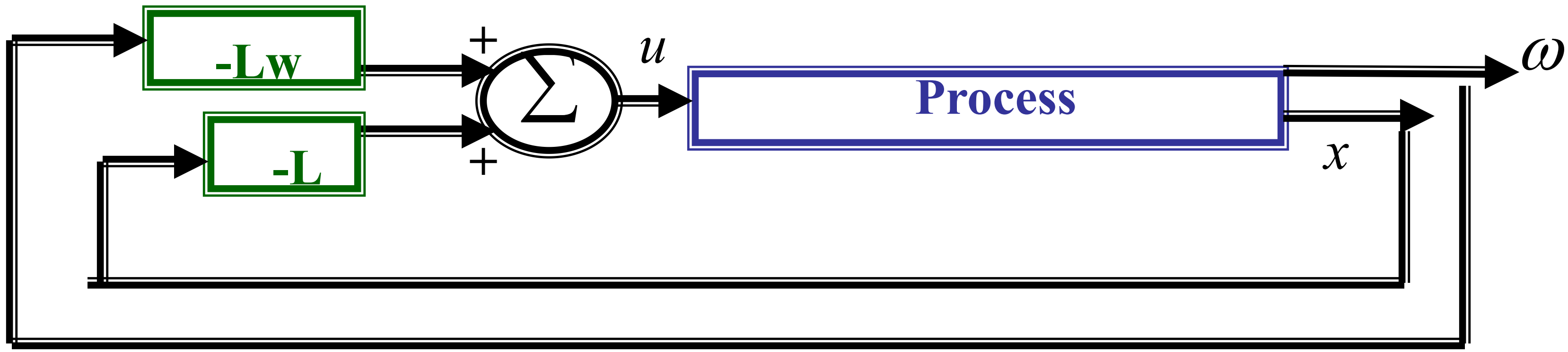
Another way is direct comparison:

$$|zI - (\Phi - Kc)| \Longrightarrow A_o(z)$$

Step 3: Output-feedback controller

$$u(k) = -L\hat{x}(k)$$

Disturbance Rejection: **What is the trick?** Treat disturbance as another state variable!



$$u(k) = -L_c z(k) = -Lx(k) - L_\omega \omega(k)$$

$$\begin{bmatrix} x(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Phi_{x\omega} \\ 0 & \Phi_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} - \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} \begin{bmatrix} L & L_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma L & \Phi_{x\omega} - \Gamma L_\omega \\ 0 & \Phi_\omega \end{bmatrix} \begin{bmatrix} x(k) \\ \omega(k) \end{bmatrix}$$

$$y(k) = \begin{bmatrix} c & 0 \end{bmatrix} z(k)$$

- How to choose Lw? Try to make $\Phi_{x\omega} - \Gamma L_\omega = 0$

What if it is impossible to make $\Phi_{x\omega} - \Gamma L_\omega = 0$?

You need to analyze the transfer function from disturbance to the output.

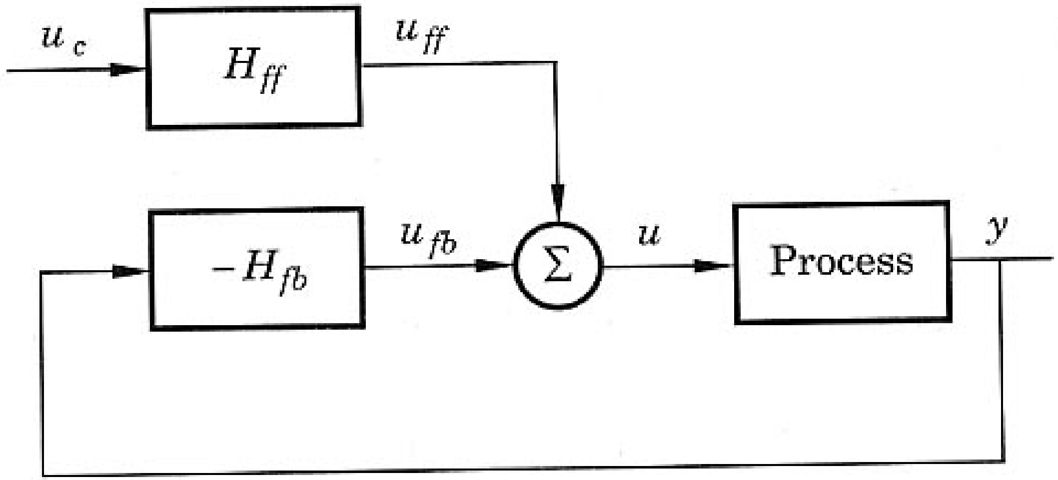
$$\omega(z) \longrightarrow \boxed{H_\omega(z)} \longrightarrow Y(z)$$

If the disturbance is constant, how to make its output ZERO?

Make the steady-state gain $H_\omega(1) = 0$

The disturbance can be estimated by the observer!

Tracking Problem: how to match the reference model?



The closed loop T.F. $\quad H_{ff}\dfrac{B(z)}{A_m(z)}$

- <u>How to design the feedforward controller?</u> $\quad H_{ff} = \dfrac{B_m(z)}{B(z)}$

- <u>Under what conditions is perfect tracking attainable? How to make sure that the inputs and outputs are bounded?</u>

$$\frac{Y(z)}{U_c(z)} = \frac{B_m(z)}{A_m(z)}$$

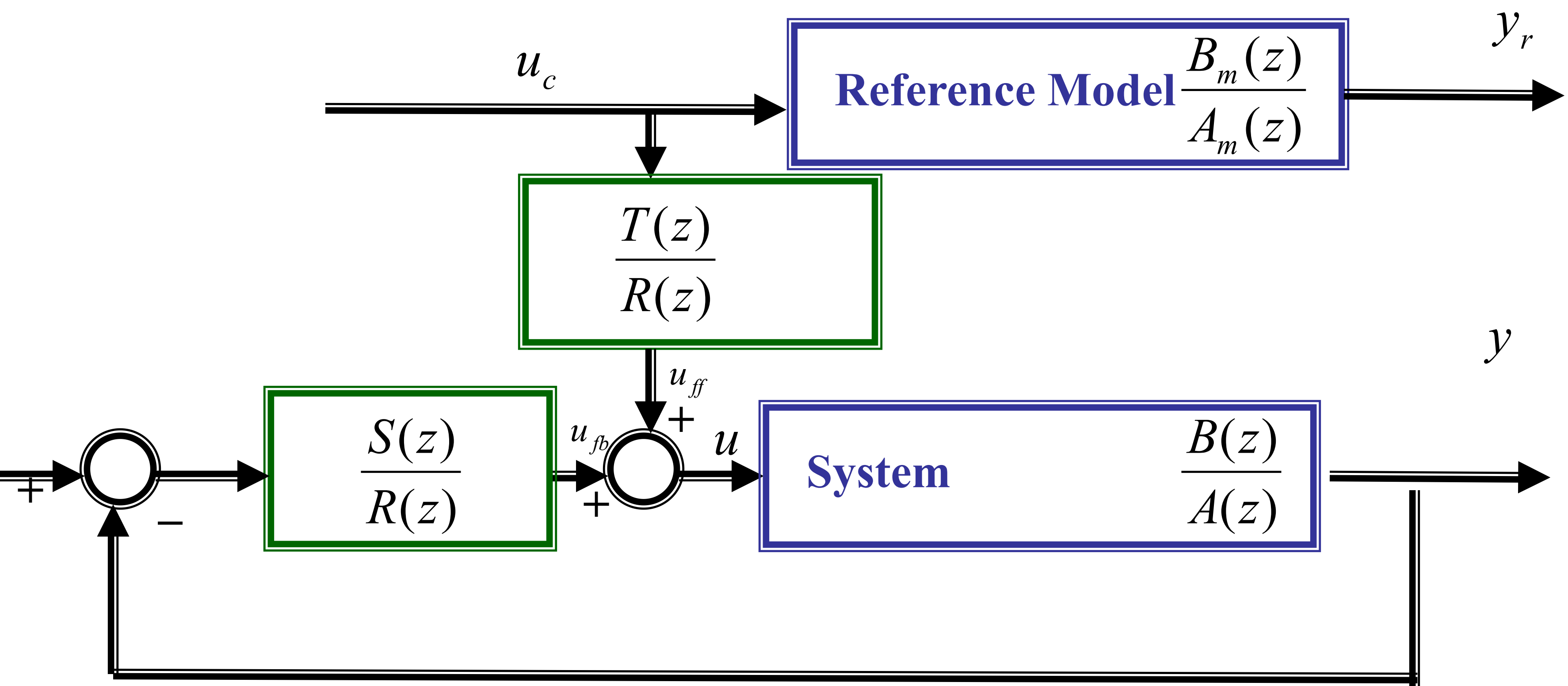
$$\frac{U(z)}{U_c(z)} = \frac{A(z)B_m(z)}{B(z)A_m(z)}$$

B(z) is stable!

<u>Do we need the condition of stable inverse B(z) for pole-placement or stabilization?</u>

No. We need it only for perfect tracking (zero-placement).

Input-output model approach: Zero cancellation



What's the closed loop transfer function from command signal Uc to y?

$$\frac{Y(z)}{U_{ff}(z)} = \frac{B(z)R(z)}{A(z)R(z)+B(z)S(z)} \qquad \xrightarrow{\frac{U_{ff}(z)}{U_c(z)} = \frac{T(z)}{R(z)}} \qquad \frac{Y(z)}{U_c(z)} = \frac{T(z)B(z)}{A(z)R(z)+B(z)S(z)}$$

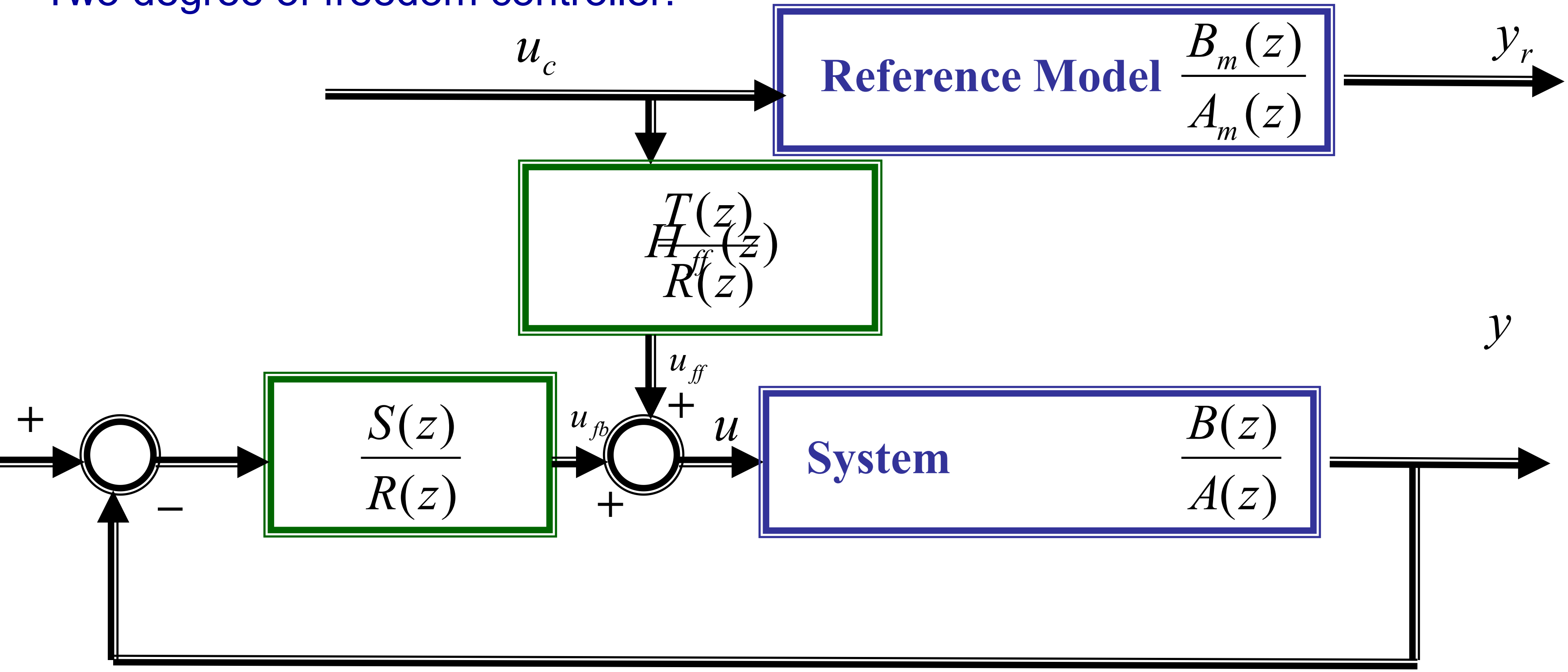
Can we cancel out B(z)? Yes if B(z) is stable! Design R such that it contains B.

Choose R(z) and S(z) such that $A_{cl}(z) = A(z)R(z)+B(z)S(z) = A_m(z)A_o(z)B(z)$

How to choose T(z) to match the reference model? $T(z) = B_m(z)A_o(z)$

$$\frac{Y(z)}{U_c(z)} = \frac{T(z)B(z)}{A(z)R(z)+B(z)S(z)} = \frac{B_m(z)A_o(z)B(z)}{A_m(z)A_o(z)B(z)} = \frac{B_m(z)}{A_m(z)}$$

Two degree of freedom controller:



What's the closed loop transfer function from command signal Uc to y?

$$\frac{Y(z)}{U_{ff}(z)} = \frac{B(z)R(z)}{A(z)R(z)+B(z)S(z)} \qquad \frac{U_{ff}(z)}{U_c(z)} = H_{ff}(z) \Longrightarrow \frac{Y(z)}{U_c(z)} = H_{ff}(z)\frac{B(z)R(z)}{A(z)R(z)+B(z)S(z)}$$

Choose R(z) and S(z) such that

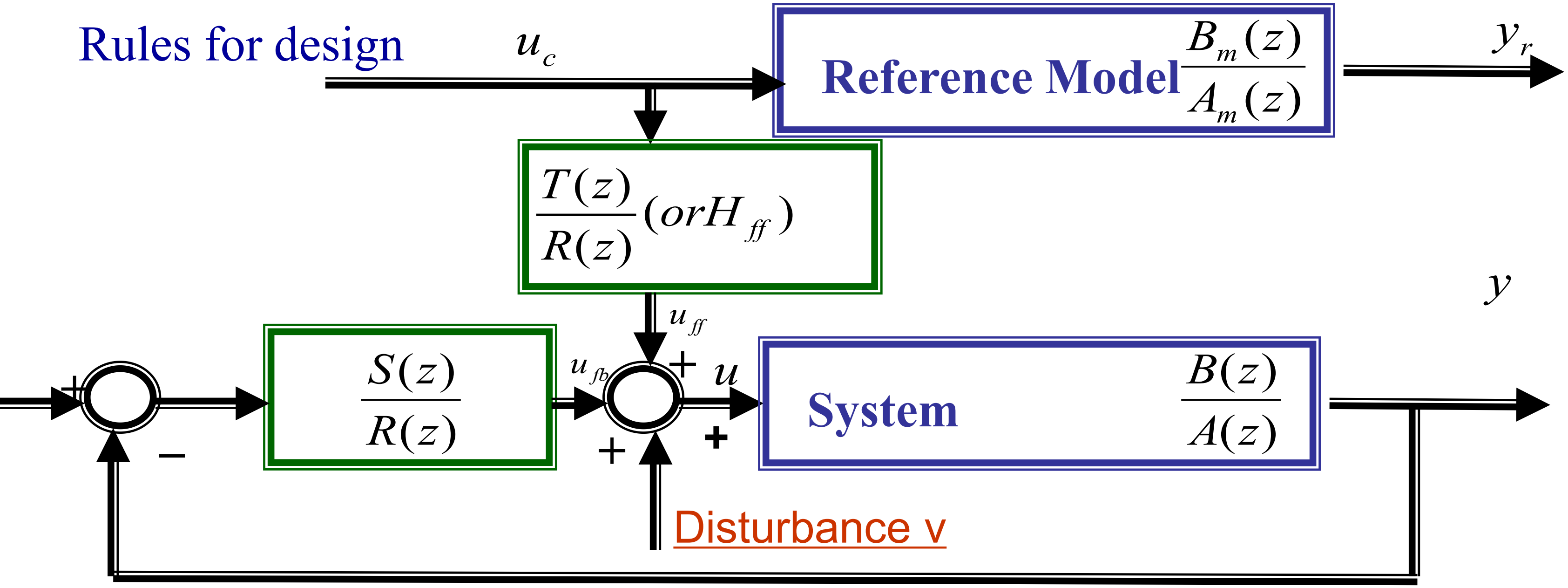
$$A_{cl}(z) = A(z)R(z)+B(z)S(z) = A_m(z)A_o(z)$$

$$\frac{Y(z)}{U_c(z)} = H_{ff}(z)\frac{B(z)R(z)}{A_m(z)A_o(z)}$$

Design $H_{ff}(z)$ to match $\frac{B_m(z)}{A_m(z)}$

$$H_{ff}(z)\frac{B(z)R(z)}{A_m(z)A_o(z)} = \frac{B_m(z)}{A_m(z)} \Longrightarrow H_{ff} = \frac{A_o(z)B_m(z)}{B(z)R(z)}$$

Rules for design



Occam's razor -- the simpler, the better.

Separation Property: Design feedback controller first, then build feedforward controller.

Step One: Figure out the design requirements on R(z).

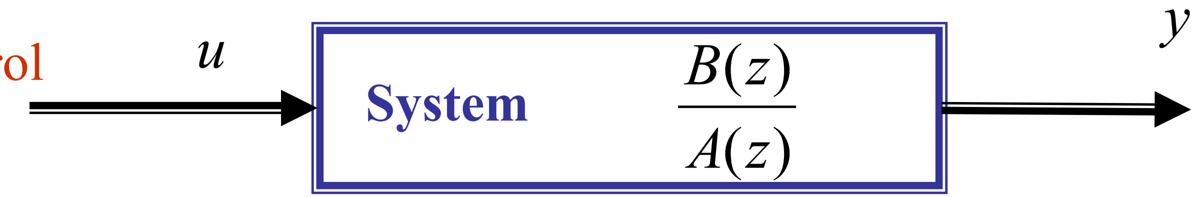
- If need to cancel zeros: R(z) must contain the zero factor, B(z)
- Disturbance rejection: R must contain the unstable factors related to the disturbance.
- If disturbance is constant, then R contain the unstable factor, (z-1).
- Causality conditions: Deg (R) >= Deg (S),

Step Two: Design R and S by solving the Diophantine equation $AR + BS = A_{cl}$

Step Three: Choose T or H_{ff} at the final stage, to satisfy other design requirements.

The order of controller can be increased in order to meet other design requirements.

Predictive control



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

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots a_n q^{-n}$$
$$B(q^{-1}) = q^{-d}(b_0 + b_1 q^{-1} + \cdots b_{n_1} q^{-n_1}), \qquad b_0 \neq 0$$
$$= q^{-d} B'(q^{-1})$$

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-d-1) + \cdots + b_{n_1} u(k-d-n_1)$$

$$y(k+d) = -a_1 y(k+d-1) - \cdots - a_n y(k+d-n) + b_0 u(k) + b_1 u(k-1) + \cdots + b_{n_1} u(k-n_1)$$

- The key idea is to construct the predictor model:

- Predictor model $y(k+d) = \alpha(q^{-1})y(k) + \beta(q^{-1})u(k)$

$$y(k+d) = \alpha_0 y(k) + \cdots + \alpha_m y(k-m) + \beta_0 u(k) + \cdots + \beta_m u(k-m)$$

- Controller Design: just match the prediction with the desired output!

$$y(k+d) = r(k+d)$$

$$r(k+d) = \alpha_0 y(k) + \cdots + \alpha_m y(k-m) + \beta_0 u(k) + \cdots + \beta_m u(k-m)$$

- The idea can be applied to nonlinear systems

One-step-ahead control

- System

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n)=b_0u(k-d)+b_1u(k-d-1)+\cdots+b_{n_1}u(k-d-n_1)$$

$$y(k+d)=-a_1y(k+d-1)-\cdots-a_ny(k+d-n)+b_0u(k)+b_1u(k-1)+\cdots+b_{n_1}u(k-n_1)$$

How to convert the model into the form of a predictor?

It is simple, just replace the future values with their predictions!

$$y(k+1)=-a_1y(k)\cdots-a_ny(k-n+1)+b_0u(k-d+1)+\cdots+b_{n_1}u(k-n_1-d+1)$$

$$y(k+2)=-a_1y(k+1)-a_2y(k)\cdots-a_ny(k-n+1)+b_0u(k-d+2)+\cdots+b_{n_1}u(k-n_1-d+2)$$

$$y(k+d-2)=-a_1y(k+d-3)-\cdots-a_ny(k+d-n-2)+b_0u(k-2)+b_1u(k-3)+\cdots+b_{n_1}u(k-n_1-2)$$

$$y(k+d-1)=-a_1y(k+d-2)-\cdots-a_ny(k+d-n-1)+b_0u(k-1)+b_1u(k-2)+\cdots+b_{n_1}u(k-n_1-1)$$

Finally we put it into the predictor form.

$$y(k+d)=\alpha_0y(k)+\cdots+\alpha_my(k-m)+\beta_0u(k)+\cdots+\beta_mu(k-m)$$

What is next ?

The one-step-ahead feedback control law only minimizes the squared prediction error:

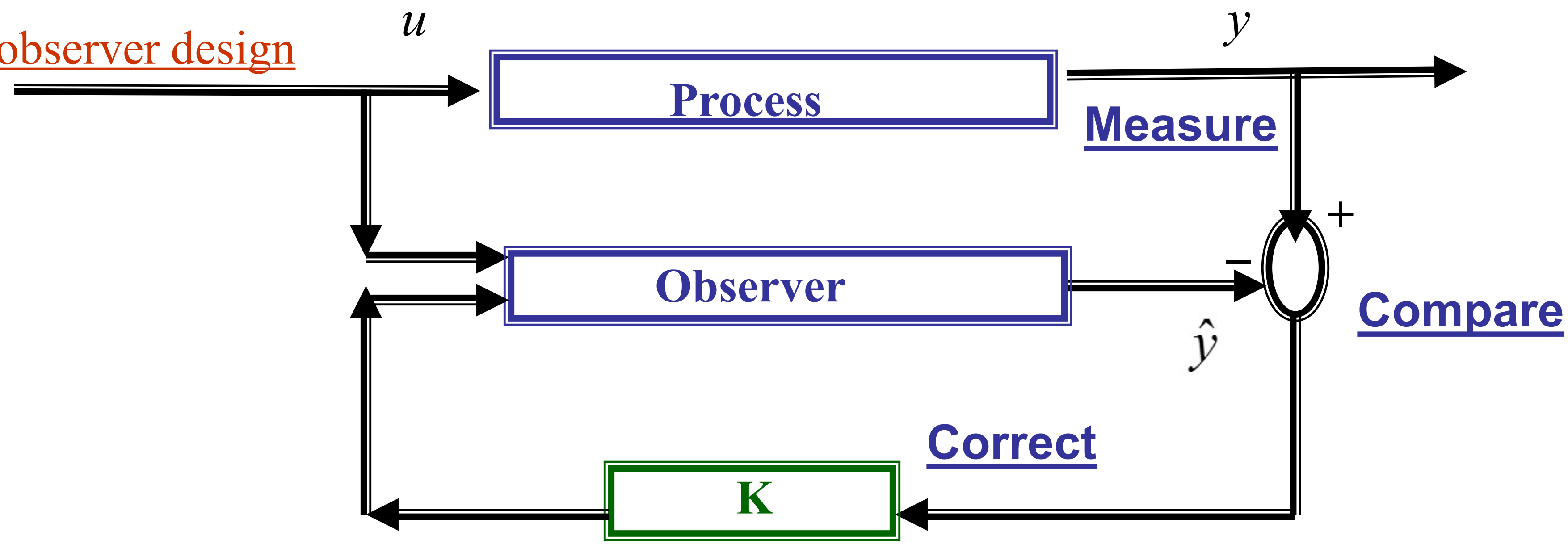
$$J_1(k+d) = \frac{1}{2}[y(k+d) - r(k+d)]^2 \quad \Rightarrow \quad y(k+d)=r(k+d)$$

The resulting control signal u(k) might be too big to be implemented!

In reality, we need to achieve
a compromise between bringing y(k + d) to r(k + d) and the cost of the control effort.

This is the starting point of so called "model predictive control"(MPC),
which is one of the main topics in part II!

The observer design



$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = cx(k)$$

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - \hat{y}(k))$$
$$\hat{y}(k) = c\hat{x}(k)$$

How to choose K?

For simplicity, we usually choose K to achieve deadbeat!

In reality, we never use deadbeat observer since it is sensitive to noise.

Then how to design K in real world application?

There is an optimal way to compute K, which is called Kalman Filter, which is another major topic for part II.

Real World Applications: Control systems are everywhere!

The future

The present

Ban on killer robots?

THANK YOU !