EE5101/ME5401: Linear Systems
Review of Part II: Control System Design

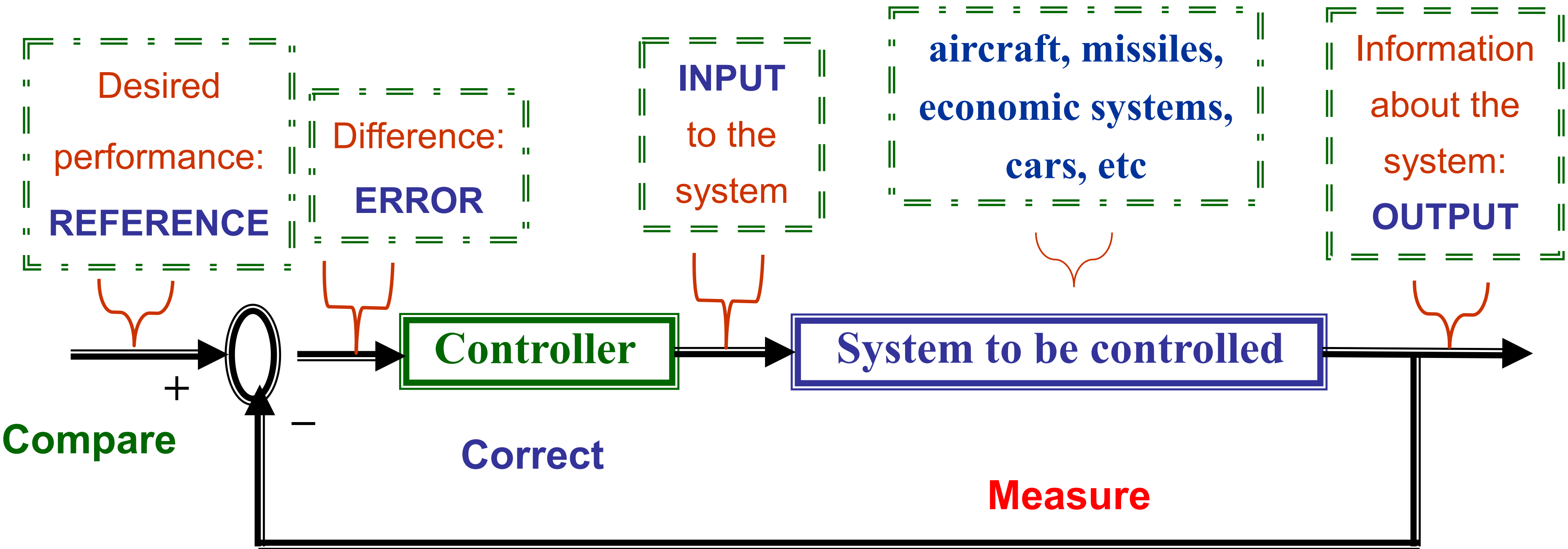
Xiang Cheng

Associate Professor

Department of Electrical & Computer Engineering
The National University of Singapore

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

•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.

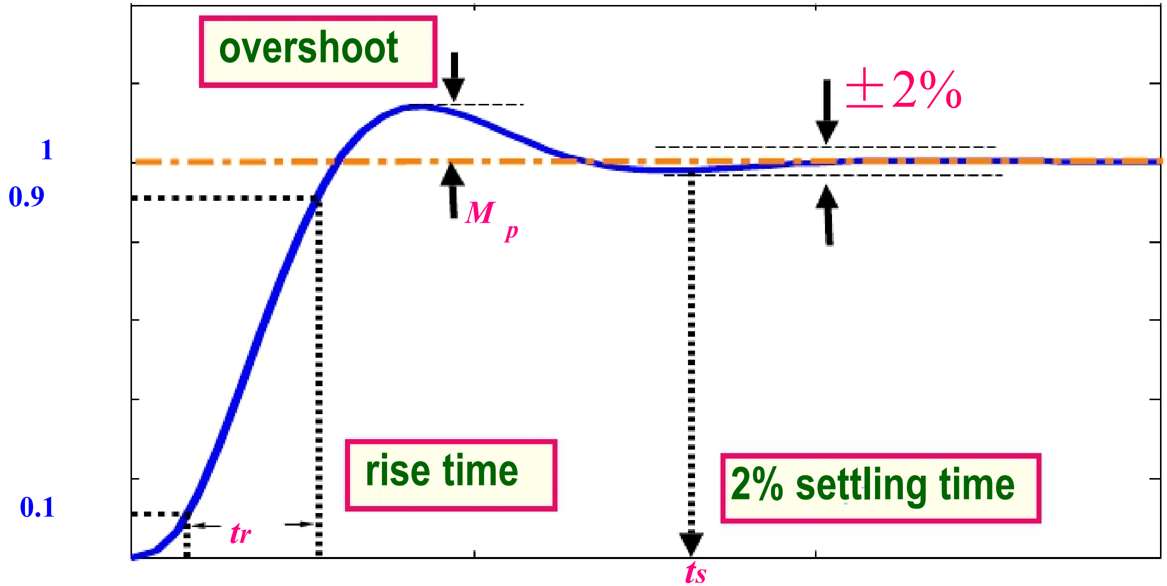
•Feedback: Measure —Compare –-Correct

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

•In practical control problems, there are certain performance specifications.

Settling time, overshoot and rise time — time domain specifications

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



$$t_r \cong \frac{1.8}{\omega_n} \qquad M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \qquad t_s \cong \frac{4.0}{\zeta\omega_n}$$

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

For higher order system, choose the second order system as the dominant mode, and place all the other poles far away (2 to 5 times faster) from the dominant one.

From the transient performance specifications, you can design the positions of the desired poles.

$$\phi_d(s) = \prod_{i=1}^{n}(s - \lambda_i) = s^n + \gamma_{n-1}s^{n-1} + \cdots + \gamma_1 s + \gamma_0$$

This is the desired closed loop characteristic polynomial.

Pole Placement

Given a plant
$$\dot{x} = Ax + Bu,$$
$$y = Cx.$$

Build a state feedback controller:

$$u = -Kx + Fr,$$

The closed loop: $\dot{x} = Ax + B(-Kx + Fr) = (A - BK)x + BFr$

Design K such that the characteristic polynomial of the closed loop

$$\det(sI - (A - BK)) = s^n + \gamma_{n-1}s^{n-1} + \cdots + \gamma_0$$

SISO system

The open loop plant $$\dot{x} = Ax + Bu,$$

The state feedback controller: $$u = r - k^T x.$$

The closed loop system $$\dot{x} = (A - bk^T)x + br$$

What is the condition on the system to place the poles to desired ones?

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

We used the controllable canonical form to derive the Ackermann's formula

$$k^T = [0,\ 0,\ \cdots,\ 0,\ 1]\mathrm{C}^{-1}\phi_d(A),$$

$$\phi_d(A) = \phi_d(s)\big|_{s=A} = A^n + \gamma_{n-1}A^{n-1} + \ldots + \gamma_0 I_n.$$

What if you forget Ackermann's formula?

The easiest way is directly comparing the coefficients of the two polynomials:

$$\det(sI - (A - bk^T)) = s^n + \gamma_{n-1}s^{n-1} + \cdots + \gamma_0$$

How many solutions can you get? Only one!

MIMO system

The open loop plant $\dot{x} = Ax + Bu,$

The state feedback controller: $u = -Kx + Fr.$

The closed loop system $\dot{x} = (A - BK)x + BFr$

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

Is the solution unique?

No. There are infinite number of solutions!

Why?

The easiest way is still comparing the coefficients of the two polynomials:

$$\det(sI - (A - BK)) = s^n + \gamma_{n-1}s^{n-1} + \cdots + \gamma_0$$

Then you will find that the number of design parameters in K is greater than the number of equations. That is why you can have infinite number of solutions.

Different solutions would give you the same poles. Will they lead to the same step responses?

No. The various solutions give you the same poles, but different zeros. So the transient responses are different from each other.

Pole Placement for MIMO system

One simple method is to force the MIMO to SISO

$$u = qv \quad \Rightarrow \quad \begin{aligned} \dot{x} &= Ax - Bu \\ &= Ax - Bqv \\ &= Ax + \underbrace{(Bq)}_{b} v \end{aligned}$$

Step 1. Choose the weight vector q such that the pair {A, Bq} is controllable.

Step 2. Use any single-input pole placement algorithm for the *pair* {A, Bq} to determine k^T such that $A - Bqk^T$

has the desired eigenvalues (closed-loop poles).

Overall, the required state feedback matrix is $K = qk^T$.

The algorithm is simple to compute.

But it does not fully exploit the freedom of the multiple inputs as it forces it into single input.

The third way of pole placement for MIMO system

<u>Algorithm For Full rank pole placement</u>

Given controllable $\{A, B\}$ and the desired $\phi_d(s)$.

(i) Obtain the controllable canonical form in the state x via

$$\bar{x} = Tx,$$

such that

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u$$

is in the controllable canonical form.

The most time-consuming part is computing T.

Once T is obtained, then we can get the controllable canonical form:

$$\bar{A} = TAT^{-1}, \quad \bar{B} = TB$$

How to compute the transformation matrix T?

First compute the controllability matrix

$$W_c = \{B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B\}$$

Check whether it is full rank or not. If it is controllable, then move on to the next step.

For single input system, the controllability matrix is a square matrix, all of the vectors are independent.

For MIMO system, we need to select the n independent vectors from the controllability matrix in the strict order from left to right.

In this way, we assure that all the inputs will play a part in the placement!

After the vectors are selected, it is important to re-group them in a square matrix C in the following form

$$C = \{b_1 \quad Ab_1 \quad \cdots \quad A^{d_1-1}b_1 \quad b_2 \quad Ab_2 \quad \cdots \quad A^{d_2-1}b_2 \cdots \quad b_m \quad Ab_m \quad \cdots \quad A^{d_m-1}b_m \quad \}$$

Where the indices d_i imply the number of vectors in C related to the i-th input, u_i

The controllable canonical form can then be computed from this matrix C!

Compute the inverse of C, C^{-1}

For multi-input case, we need to take out m rows from C^{-1} corresponding to the m inputs, and form T as

$$T = \begin{bmatrix} q_{d_1}^T \\ q_{d_1}^T A \\ \vdots \\ q_{d_1}^T A^{d_1-1} \\ q_{d_1+d_2}^T \\ q_{d_1+d_2}^T A \\ \vdots \\ q_{d_1+d_2}^T A^{d_2-1} \\ \vdots \end{bmatrix} \Rightarrow \bar{B} = TB = \begin{array}{c} \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 1 & \times & \cdots & \times \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \end{array} \begin{array}{l} \\ \\ \\ d_1^{th} row \\ \\ \\ \\ (d_1+d_2)^{th} row \\ \\ \\ \\ n^{th} row \end{array}$$

How many T can you find?

There is only one T!

(ii) Specify a desired closed-loop matrix A_d

such that $\det(sI - A_d) = \phi_d(s)$

and A_d is the same as $\bar{A}$ except non-trivial rows.

How many ways can you choose the desired closed-loop matrix A_d ?

There are many ways in this step! That is why the full rank solution to the placement for MI system is still not unique!

(iii) Compare the non-trivial rows of the closed loop canonical form

$$\overline{A} - \overline{B}\overline{K}$$

with the desired one A_d, and compute $\overline{K}$.

(iv) Compute the original feedback gain K

$$K = \overline{K}T.$$

The computation is much more complicated than the unity rank method.

But it can fully exploit the freedom of all the inputs! Therefore the overall performance is usually better than the unity rank method.

When we design the pole placement controller, the main focus is on how to meet the transient performance requirements.

Since there are many possible poles to meet the transient performance requirements, sometimes we want to choose the "best" one to strike the balance between speed and cost.
That is the motivation behind the optimal control -- LQR

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \mathbf{x}(0) \neq 0,$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t),$$

The objective is to bring the state from *non-zero initial value to zero*. This is the *regulation* problem. The problem is cast into **the following** *quadratic* **cost function,**

$$J = \frac{1}{2}\int_0^\infty \left(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\right)dt.$$

The LQR optimal control is to find the control law to minimize this cost function. The optimal control law turns out to be in the form of *linear* state feedback:

$$\mathbf{u} = -\mathbf{K}\mathbf{x},$$

What are the conditions to solve LQR problem?

Assumption 1. The system, (A, B), is controllable.

Why? What would happen if the system is not controllable?

If the uncontrollable mode is unstable, the unstable mode will blow up.

Assumption 2. The pair $(\mathbf{A}, \mathbf{H})$ is completely observable, where $\mathbf{H}$ is any matrix such that $\mathbf{H}^T\mathbf{H} = \mathbf{Q}$.

Why do we need this condition on Q?

Assumption 2 says that *all* the state variables will be "observed" by the performance index. Otherwise, those state variables which are not connected to the performance index may blow up.

Assumption 3. The matrix $\boldsymbol{R}$ *is positive definite, while the matrix* $\boldsymbol{Q}$ *is semi-positive definite.*

Why? What would happen if R is not positive definite?

If R is not positive definite, the input signals which are not included in the performance index may blow up!

How to Solve LQR problem?

System: $\dot{x} = Ax + Bu,$

Performance Index: $J = \frac{1}{2}\int_0^\infty \left(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\right)dt.$

Controller: $u = -Kx, \quad K = R^{-1}B^T P,$

where P is the symmetric positive definite solution of the algebraic Matrix Riccati equation:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0.$$

The difficult part for LQR is how to solve ARE.

For first or second order system, you can first assume a matrix P with symmetric elements, and then plug it into the ARE.

It normally results in a set of nonlinear equations, with multiple solutions.

Out of the multiple solutions, there is ONLY ONE solution which is positive definite. And you have to find it out.

How to Solve LQR problem?

A systematic way to solve ARE:

Step 1: Form the $2n$x$2n$ matrix: $$\Gamma = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix},$$

and find its n stable eigenvalues.

***Step 2*:** Let the eigenvector of Γ corresponding to **stable** λ_i , $i = 1,\dots, n,$ be

$$\begin{pmatrix} \nu_i \\ \mu_i \end{pmatrix}, i = 1, 2, \cdots, n.$$

where ν_i, μ_i are n-dimensional vectors.

Then, P is given by $$P = [\mu_1,\dots,\mu_n][\nu_1,\dots,\nu_n]^{-1}.$$

Servo Control

Both pole placement and LQR focus upon how fast we want the system to behave, or in other words, the transient response.

$$\text{Control Specifications} \begin{cases} Transient: \quad speed, \quad overshoot, \quad etc. \\ Accuracy: \quad e(t) = r(t) - y(t) \quad 0 \le t < \infty \\ Steady \quad State \quad Accuracy: \quad e(\infty) \end{cases}$$

How to achieve $e(\infty) = 0$
in face of disturbance w(t) and set-point change r(t)?

Design = Servo mechanism + Stabilization!

We present

- Output feedback for SISO system
- State feedback for MIMO System

Polynomial Approach to General SISO Servo Problem

Consider the unity output feedback system.

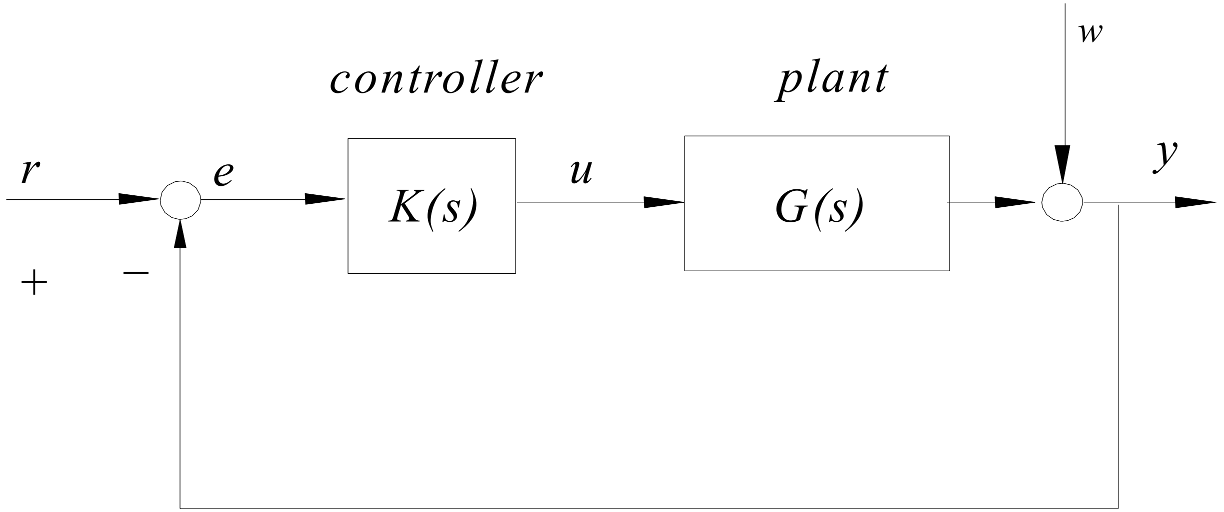
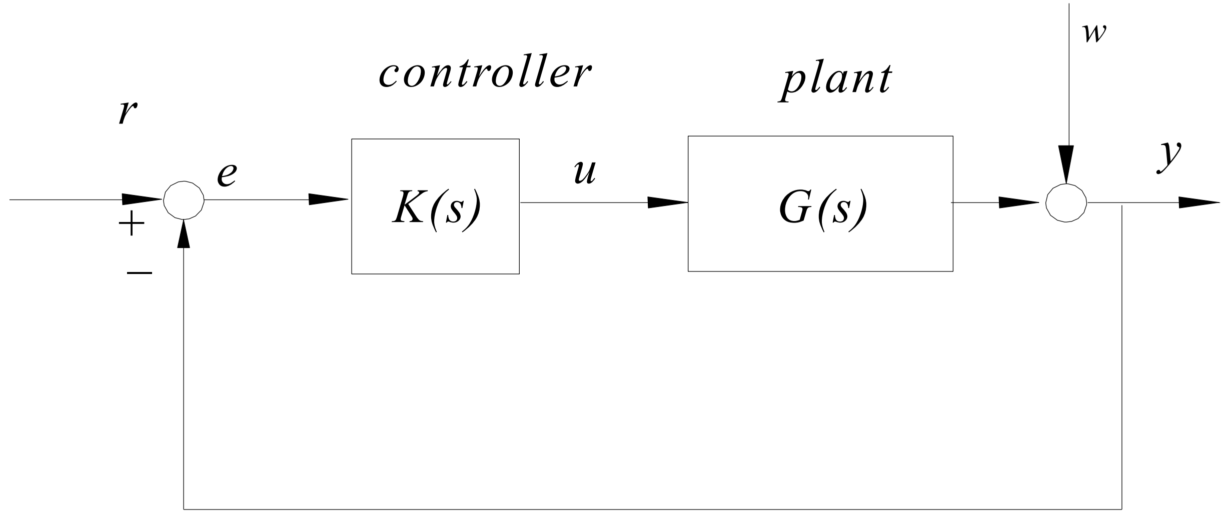


Figure 2 Unity output feedback system.

Objective: $e(\infty) = 0$ ***in face of r and w***



Let $$G(s) = \frac{N(s)}{D(s)}, \qquad K(s) = \frac{B(s)}{A(s)},$$

The order of the process is n. $\deg N(s) \le \deg D(s) = n$.

The closed loop transfer function H(s) is given by

$$H(s) = \frac{G(s)K(s)}{1+G(s)K(s)} = \frac{\dfrac{N(s)}{D(s)}\dfrac{B(s)}{A(s)}}{1+\dfrac{N(s)}{D(s)}\dfrac{B(s)}{A(s)}} = \frac{N(s)B(s)}{D(s)A(s)+N(s)B(s)}.$$

Design = Servo mechanism + Stabilization!

Let's first solve the pole-placement problem (stabilization)

$$D(s)A(s) + N(s)B(s) = P_c(s)$$

where $D(s)$ and $N(s)$ are known, the roots of $P_c(s)$ are the desired poles of the overall system to achieve, and $A(s)$ and $B(s)$ are unknown polynomials to be determined.

What is the condition for solving the pole placement problem?

D(s) and N(s) are coprime --- No common factors

If the common factor is stable, that stable pole cannot be changed. But the overall system is still stabilizable!

What is the order of the controller if the order of the system is n?

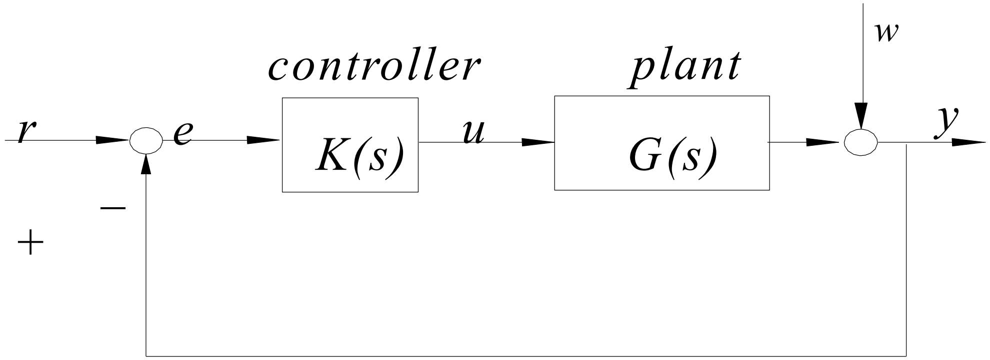
If the order of the controller is n-1, there is only one solution. If the order of the controller is larger than n-1, then the solution is not unique.

How to find the solution?

It is easy! Just plug in A(s) and B(s) into the equation, and compare the coefficients of the two polynomials.

$$D(s)A(s) + N(s)B(s) = P_c(s)$$

What is the Servo mechanism? How to make e(t) → 0?



$$E(s) = R(s) - Y_r(s) = R(s) - \frac{G(s)K(s)}{1+G(s)K(s)}R(s)$$

$$= \frac{1}{1+G(s)K(s)}R(s) = \frac{1}{1+\frac{N(s)}{D(s)}\frac{B(s)}{A(s)}}\frac{N_r(s)}{D_r(s)} = \frac{D(s)A(s)}{D(s)A(s)+N(s)B(s)}\frac{N_r(s)}{D_r(s)}$$

To make E(s) go to zero, the unstable factor in $D_r(s)$ needs to be cancelled out.
So we include $D_r(s)$ in A(s).
Similarly let's compute the output due to the disturbance:

$$Y_w(s) = \frac{1}{1+G(s)K(s)}W(s) = \frac{A(s)D(s)}{A(s)D(s)+B(s)N(s)}\frac{N_w(s)}{D_w(s)}$$

To make $Y_w(s)$ go to zero, the unstable factor in $D_w(s)$ needs to be cancelled out.
So we include $D_w(s)$ in A(s).

To achieve both goals, we include both unstable factors $D_w(s)$ and $D_r(s)$ in A(s).
That's why we need to get the least common denominator of R(s) and W(s), Q(s).
This is the servo mechanism.

Servo control design procedure for SISO system:

(i) Obtain plant coprime polynomial fraction as $G = N(s)/D(s)$.

(ii) Determine LCD, $Q(s)$ from the given types of disturbance W(s) and reference R(s).

(iii) Design $\tilde{K}$ to stabilize the generalized plant, $N/(DQ)$.

(iv) Form the servo controller as

$$K = \tilde{K}\frac{1}{Q} = \frac{B}{A}$$

From this design:

- The closed loop is stable by pole placement,
- Q(s) is inside A(s) such that the unstable factors of R(s) and W(s) disappear!

The problem with this design is that the order of the controller might be high because the order of the generalized plant is increased.

Simplified design procedure for Servo Control :

(i) Obtain plant coprime polynomial fraction as $G = N(s)/D(s)$.

(ii) Determine Q from the given types of disturbance and reference.

(iii) Skip the step of generalized plant. Directly include Q(s) as a factor of A(s) or in other words, design

$$A(s) = Q(s)A'(s)$$

Also note that if the poles of the open loop D(s) already contain some factors of Q(s), there is no need to include them again in A(s).

Cancelling out stable poles/zeros may also lead to simpler design.

(iv) Solve the pole placement problem:

$$D(s)Q(s)A'(s) + N(s)B(s) = P_c(s)$$

Always make sure that the controller B/A is proper!

Servo controller design for MIMO system

Consider an $m \times m$ plant:

$$\dot{x} = Ax + Bu + B_w w,$$

$$y = Cx,$$

The error is defined as $e = r - y.$

Suppose that the disturbance w and reference r are **both of step type**. We introduce one integrator to each channel:

$$v(t) = \int_0^t e\,(\mathrm{t})\,d\tau$$

$$\dot{v}(t) = e(t) = r - y(t) = r - Cx(t).$$

Combine the integrators with the state space model to form the augmented system:

$$\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} A & O \\ -C & O \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} B \\ O \end{pmatrix} u + \begin{pmatrix} B_w \\ O \end{pmatrix} w + \begin{pmatrix} 0 \\ I \end{pmatrix} r$$

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + \bar{B}_w w + \bar{B}_r r$$

Augmented system $$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}u + \bar{B}_w w + \bar{B}_r r$$

If the augmented system is controllable, it can be stabilized by the state feedback control law:

$$u = -K\bar{x} = -\begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$$

or

$$u(t) = -\underset{\substack{\uparrow \\ P}}{K_1} x - \underset{\substack{\uparrow \\ I}}{K_2} \int_0^t e \, d\tau,$$

which contains integral control. The feedback gain K may be determined by pole placement procedure or LQR.

Once the feedback system is stable, each signal in the system in response to step inputs will be constant in the steady state, and so is $v(t)$.

$$e(t) = \dot{v}(t) = 0.$$

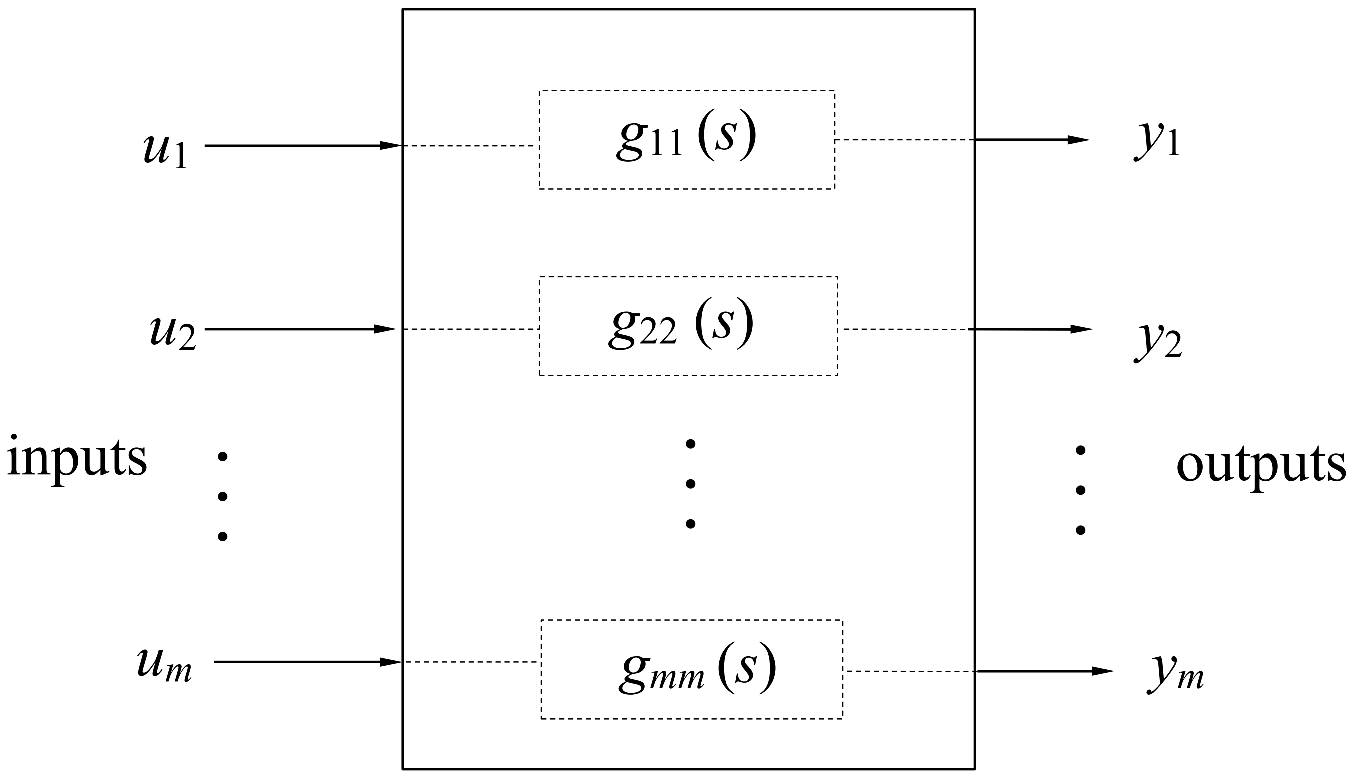
Thus, zero steady state error has been achieved.

Design = Servo mechanism + Stabilization!

Integrator is the magic tool to deal with constant reference. This idea of constructing the augmented system can be extended to deal with other types of reference signals and disturbances.

Controller design for SISO system is straight forward. But the design for MIMO could be very tricky. Is there a simpler way to deal with MIMO systems?

Therefore, people have been trying hard to find a way to convert the MIMO to SISO system such that they can design the control systems easily.



Decoupled system.

The decoupled system is easy to handle as we can treat it as SISO!

Decoupling by State Feedback

$$\dot{x} = Ax + Bu,$$
$$y = Cx,$$

where $u, y \in R^m, x \in R^n, B \in R^{n\times m}, C \in R^{m\times n},$ to be decoupled by the state feedback $u = -Kx + Fr.$

We need to design both K and F!

The resultant system is

$$\dot{x} = (A - BK)x + BFr,$$
$$y = Cx,$$

Open loop TF: $G(s) = C(sI - A)^{-1} B.$

Closed loop TF: $H(s) = C(sI - A + BK)^{-1} BF.$

The transfer function of the closed loop, H(s), is related to the open loop G(s). To derive the expression of the closed loop transfer function in terms of the open loop is the key to the solution to decoupling problem!

$$H(s) = G(s)\left[I + K(sI - A)^{-1} B\right]^{-1} F.$$

If H(s) is diagonal, then G(s) must be nonsingular.
This is the only condition for the system to be decoupled.

First we need to find a condition to assure that $G(s) = C(sI - A)^{-1}B$ is nonsingular.

Partition C into m row vectors.

$$C = \begin{bmatrix} c_1^T \\ c_2^T \\ \vdots \\ c_m^T \end{bmatrix}$$

Define, $\sigma_i, i = 1,2,\cdots,m$ *, as an integerby*

$$\sigma_i = \begin{cases} \min\left(j \middle| c_i^T A^{j-1} B \neq 0^T, j = 1,\ 2,\ \cdots,\ n \right); \\ \\ n, \quad if \quad c_i^T A^{j-1} B = 0^T, j = 1,\ 2,\ \cdots,\ n. \end{cases}$$

σ_i corresponds to the relative degree of the transfer function for each row.

You need to be very clear on how to compute this indicator, which is the key design parameter for decoupling.

$$G(s) = C(sI - A)^{-1}B$$

The key to decoupling problem is to express G(s) in the form of

$$G(s) = \begin{pmatrix} s^{-\sigma_1} & 0 & \cdots & 0 \\ 0 & s^{-\sigma_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s^{-\sigma_m} \end{pmatrix} \left[B^* + C^* (sI - A)^{-1} B \right],$$

where B^* and C^* are defined as

$$B^* = \begin{bmatrix} c_1^T A^{\sigma_1 - 1} B \\ c_2^T A^{\sigma_2 - 1} B \\ \vdots \\ c_m^T A^{\sigma_m - 1} B \end{bmatrix} \qquad C^* = \begin{bmatrix} c_1^T A^{\sigma_1} \\ c_2^T A^{\sigma_2} \\ \vdots \\ c_m^T A^{\sigma_m} \end{bmatrix}$$

σ_i corresponds to the relative degree of the transfer function for each row.

B* correspond to the leading coefficients of the zero polynomials.

The key to prove this is to express $(sI - A)^{-1}$ as infinite series:

$$(sI - A)^{-1} = s^{-1} \left[I - \frac{A}{s} \right]^{-1} = s^{-1} \left[I + As^{-1} + A^2 s^{-2} + \cdots \right]$$

How to design the decoupler u=-Kx+Fr?

Compute the matrix

$$B^* = \begin{pmatrix} c_1^T A^{\sigma_1 - 1} B \\ c_2^T A^{\sigma_2 - 1} B \\ \vdots \\ c_m^T A^{\sigma_m - 1} B \end{pmatrix}$$

If this B^* is nonsingular, let

$$C^* = \begin{bmatrix} c_1^T A^{\sigma_1} \\ c_2^T A^{\sigma_2} \\ \vdots \\ c_m^T A^{\sigma_m} \end{bmatrix}$$

Design the state feedback controller: $u = -Kx + Fr.$

$$F = B^{*-1}, \qquad K = B^{*-1} C^*.$$

Then, the closed-loop system transfer function matrix is given by

$$H(s) = diag\left(s^{-\sigma_1},\ s^{-\sigma_2},\ \cdots,\ s^{-\sigma_m}\right),$$

which is called an integrator decoupled system.

But the integrator decoupled system is unstable. Can we make it stable?

Decoupling with pole placement: One wishes to have the closed-loop transfer function matrix as

$$H(s) = \begin{pmatrix} \frac{1}{\phi_{f_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\phi_{f_2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\phi_{f_m}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{s^{\sigma_1} + \gamma_{11} s^{\sigma_1 - 1} + \cdots + \gamma_{1\sigma_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{s^{\sigma_m} + \gamma_{m1} s^{\sigma_m - 1} + \cdots + \gamma_{m\sigma_m}} \end{pmatrix}$$

It turns out that we just need to change the way to compute C*.

Define

$$C^{**} = \begin{bmatrix} c_1^T \phi_{f1}(A) \\ c_2^T \phi_{f2}(A) \\ \vdots \\ c_m^T \phi_{fm}(A) \end{bmatrix}$$

where $\phi_{f_i}(A) = A^{\sigma_i} + \gamma_{i1} A^{\sigma_i - 1} + \cdots + \gamma_{i\sigma_i} I,$

$$\phi_{f_i}(\mathrm{s}) = s^{\sigma_i} + \gamma_{i1} s^{\sigma_i - 1} + \cdots + \gamma_{i\sigma_i}$$

corresponds to the stable characteristic polynomial of the i-th input-output pair.

The state feedback controller $u = -Kx + Fr.$

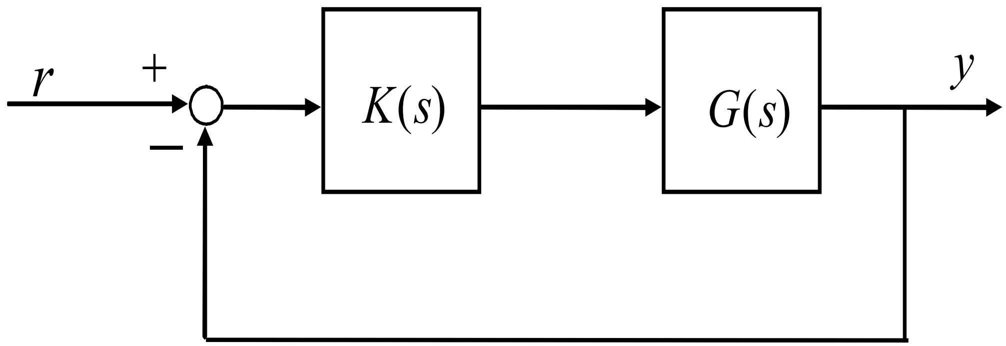
$$F = B^{*-1}, \qquad K = B^{*-1} C^{**}.$$

can make the resultant feedback system have the desired stable transfer functions.

Decoupling Control by Output Feedback

So we know how to decouple the system if the state space model is given. What if we only know the transfer function G(s)? Can we still decouple the system?

It turns out the solution is even simpler. Consider the unity feedback system,



Our task here is to design $K(s)$ such that the feedback system is internally stable and the closed-loop transfer function matrix,

$$H(s) = \left[I + G(s)K(s)\right]^{-1} G(s)K(s) \quad ,$$

is decoupled, or diagonal.

It can be shown that the closed loop is decoupled if and only if the open loop TF is decoupled.

Design procedure. The decoupling problem with stability by output feedback can be solved by designing $K(s)$ in two stages, i.e., $K(s) = K_d(s)K_s(s)$.

Step One: $K_d(s)$ is to make $G(s)K_d(s)$ diagonal and non-singular with no unstable pole-zero cancellations. In this step, $K_d(s)$ is not required to be proper.

Step Two: Whenever such a $K_d(s)$ is found in step one, it is always possible to design a diagonal controller $K_s(s)$ to stabilize the resultant $G(s)K_d(s)$ with SISO methods or pole placement technique loop by loop so that decoupling is not affected, and to assure that there are no unstable pole-zero cancellation between G(s) and K(s)=$K_d(s)K_s(s)$, and to make $K(s)$ proper.

The key step in decoupling control is to find $K_d(s)$.

Can we simply use $K_d(s) = G^{-1}(s)$?

In many cases, we cannot do this as it may lead to unstable pole-zero cancellation.

Design for decoupler $K_d(s)$: Suppose that $G(s)$ is non-singular. Write

$$G(s) = \frac{N(s)}{d(s)}$$

where $d(s)$ is the least common denominator of $G(s)$, and N(s) is polynomial matrix. If $\det(N)$ and $d(s)$ have no common unstable roots, then choose

$$K_d(s) = \text{adj}(N(s))$$

so that

- $G(s)K_d(s) = \dfrac{N(s)\bullet adj(N(s))}{d(s)} = \dfrac{\det(N(s))}{d(s)} I_m$ is decoupled

One problem with this simple method is that the order of the decoupled systems is large.

Refinement. One way to reduce the order of the decoupler is to elaborate the above method as follows. Express $G(s)$ as

$$G(s) = diag\left\{\frac{1}{d_1}, \quad \frac{1}{d_2}, \quad \cdots, \quad \frac{1}{d_m}\right\} N_r(s)$$

where d_i is the least common denominator of i-th row of G(s) so that

$N_r(s)$ is a polynomial matrix. Choose

$$K_d = adj(N_r(s)) \qquad \text{Then,}$$

$$G(s)K_d(s) = diag\left\{\frac{\det N_r(s)}{d_1}, \quad \frac{\det N_r(s)}{d_2}, \quad \cdots, \quad \frac{\det N_r(s)}{d_m}\right\}$$

You still need to design diagonal controller $K_s(s)$ to stabilize the resultant $G(s)K_d(s)$ with SISO methods so that decoupling is not affected, and to assure that there are no unstable pole-zero cancellation between G(s) and K(s)=$K_d(s)K_s(s)$, and to make $K(s)$ proper.

Why state estimation?

- The state space approach uses state feedback:

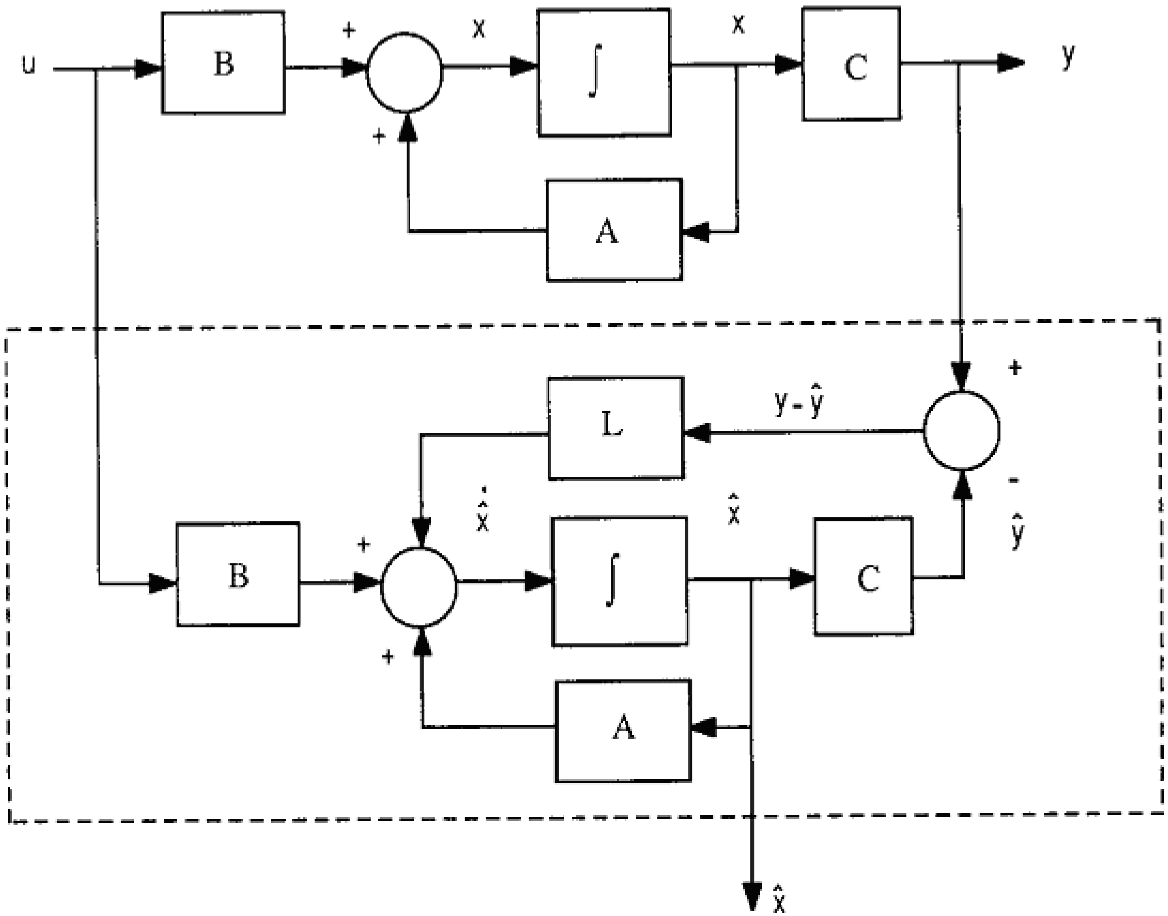
$$u = -Kx + Fr$$

- But usually, x is not available as all state variables are not measurable.

- Only y is measurable.

- To implement state feedback without $x(t)$ necessitates use of state estimation.

Full-order Observer



$$\dot{x} = Ax + Bu,$$
$$y = Cx.$$

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\left[y - \hat{y}\right]$$
$$\hat{y} = C\hat{x}$$

Observer = model + feedback correction mechanism

System

$$\dot{x} = Ax + Bu,$$
$$y = Cx.$$

Observer

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\left[y - \hat{y}\right]$$
$$\hat{y} = C\hat{x}$$

Estimation error $\tilde{x} = x - \hat{x}$

$$\dot{\tilde{x}} = (A - LC)\tilde{x}$$

$$\det\left[sI - \left(A - LC\right)\right] = \det\left[sI - \left(A - LC\right)\right]^{T} = \det\left[sI - (\tilde{A} - \tilde{B}\tilde{K})\right]$$

A design procedure is summarized as follows.

(i) Choose the observer poles 3-5 times faster than control poles;
(ii) Use a pole placement algorithm to obtain L.

Once the estimates are available, the controller is

$$u = -K\hat{x} + Fr$$

In full-order observer, we notice that some state variables are already available form the output signal. Can we reduce the redundancy?

Reduced-Order Observers

Idea: Since state variables are not unique and they can be transformed from one into another, we want to take advantage of the m state variables that are available through $y=Cx$ and construct an observer of order $n-m$, lower than n, to estimate the remaining (n-m) state variables.

Let $\xi = Tx$, where T is $(n-m)\times n$ with $(n-m)$ rows that are linearly independent of C.

$$\begin{bmatrix} y \\ \xi \end{bmatrix} = \begin{bmatrix} C \\ T \end{bmatrix} x \qquad \begin{bmatrix} C \\ T \end{bmatrix} \text{ is of } n\times n \text{ and non-singular.}$$

Using T as a design variable is one of the keys to the design of a reduced-order.

For the system:

$$\dot{x} = Ax + Bu,$$
$$y = Cx,$$

construct an observer as

$$\dot{\xi} = D\xi + Eu + Gy$$

such that

$\xi \to Tx$ as t goes to infinity.

The estimation error dynamics:

$$e = \xi - Tx$$
$$\frac{de}{dt} = De + (DT - TA + GC)x + (E - TB)u$$

If D is stable, and

$$DT - TA + GC = 0,$$
$$E - TB = 0,$$

then

$$\frac{de}{dt} = De$$

So, the estimation error will converge to zero.

Design: Determine *T, D, G* and *E* as follows.

(i) Find the constraints on T such that $\begin{bmatrix} C \\ T \end{bmatrix}$ is non-singular; but do not choose T directly at this step.

(ii) Choose D such that its eigenvalues have negative real parts, or desired decay rates; there are multiple solution in this step.

(iii) Solve $DT - TA + GC = 0$ for T and G; this is the key design equation. The solution is also not unique in this step.

(iv) Calculate E=TB.

Finally, one needs to reconstruct the state variables from the plant's output and the observer's output

$$\hat{x} = \begin{bmatrix} C \\ T \end{bmatrix}^{-1} \begin{bmatrix} y \\ \xi \end{bmatrix}$$

Once the estimates are available, the controller is

$$u = -K\hat{x} + Fr$$

Break

State-of-the-art control systems

Future Robots (6: 76:0)

How to design the control systems for real applications

Step one: Build a mathematical model of the process. Linearize the model around the operation point, and get the state space model.
This is actually the most challenging part. Because once a good model is available, the rest is simple. Just apply whatever you have learned from this module.

Step Two: Assume all the state variables are available, build the state feedback controller. No need to worry whether you can afford all the sensors or not !

Step Three: Talk to your boss about the budget and decide the type of sensors to monitor the system, and build observer to estimate those state variables without any sensors. If you are desperate to save money, then just use the cheapest sensor to measure one output!
For practical applications, Kalman Filter is usually preferred due to presence of noise.

At least one sensor (one output signal) is required! Otherwise, there is no feedback!

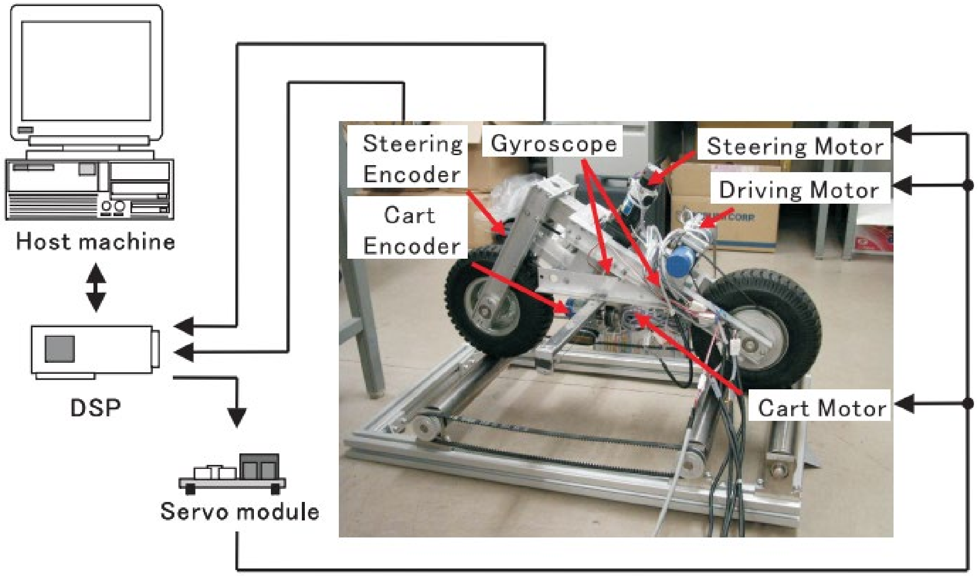
Step Four: Replace those state variables with their estimates and implement the controller.

There are three major factors you should consider when you tune the design parameters in your controller:

1. Speed --- Transient response
2. Accuracy --- Steady state error
3. Cost ---- Size of the control signals as well as the sensors needed.

Project Briefing

Control of a Stationary Self-Balancing Two-wheeled Vehicle



$$\dot{x} = Ax + Bu$$

$$y = Cx$$

$$x = \begin{bmatrix} d(t) & \phi(t) & \psi(t) & \dot{d}(t) & \dot{\phi}(t) & \dot{\psi}(t) \end{bmatrix}^T$$

6th order system
2 inputs
3 outputs

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 6.5 & -10 & -\alpha & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ 5 & -3.6 & 0 & 0 & 0 & -\gamma \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \beta & 11.2 \\ b_{51} & b_{52} \\ 40 & \delta \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad u = \begin{bmatrix} u_c(t), & u_h(t) \end{bmatrix}^T$$

Design Specifications.

The transient response performance specifications for all the outputs y in state space model are as follows:

1) The overshoot is less than 10%.
2) The 2% settling time is less than 5 seconds.

You can use the following formula to design the desired poles.

$$t_s = \frac{4}{\zeta\omega_n}, M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

But remember that there is no guarantee that the transient response will automatically satisfy the requirements if the poles are placed into desired region. Because this formula is only accurate for the standard second order system without any zeros. For the practical systems, it is not second order, and it also has zeros, which may cause some discrepancy.

Therefore, you need to simulate the control system using SIMULINK and plot out the step responses of the three outputs to verify whether the design specifications are met. Fine tuning is necessary to get a good solution.

Design Specifications.
The transient response performance specifications for all the outputs y in state space model are as follows:

1) The overshoot is less than 10%.
2) The 2% settling time is less than 5 seconds.

How to verify whether above design specifications are met in SIMULINK?

As you are building the blocks using SIMULINK in MATLAB, it is easy to simulate the step response.

For example, you first design the state feedback controller $u = -Kx$ using pole placement or LQR.

When you want to check the step responses of the three outputs, simply apply

$$u = -Kx + r$$

where r are step inputs. Since there are two inputs, you should apply only one step at a time and keep the other one as zero (r=[1,0], or r=[0,1]), and then observe the behavior of the three outputs.

You need to check 6 step responses and try to make all of them meet the requirements.

1) **Assume that you can measure all the six state variables, design a state feedback controller using the pole place method, simulate the designed system and show all the six state responses to non-zero initial state x_0 with zero external inputs. Discuss effects of the positions of the poles on system performance and monitor control signal size. In this step, both the disturbance and set point can be assumed to be zero. (10 points)**

- You need to use the transient performance specifications to design the desired poles .
- But the designed poles might not meet the transient response requirement, That's why you need to vary the poles (mainly the pair of the complex poles) and find out the ones which can do the job.
- For this part, you need to check whether the transient specifications are met by using the controller:

$$u = -Kx + r$$

where r is step input, as discussed earlier.

- For step response, the initial conditions is set as zero.
- For response to non-zero initial state, set external inputs r=0, and set the initial condition for the state variable.

With the same desired poles, different methods may lead to different zeros.

Full rank method is usually better than unity rank method.

2) Assume that you can measure all the six state variables, design a state feedback controller using the LQR method, simulate the designed system and show all the state responses to non-zero initial state with zero external inputs. Discuss effects of weightings *Q* and *R* on system performance, and also monitor control signal size. In this step, both the disturbance and set point can be assumed to be zero. **(10 points)**

The key to LQR:

1. How to solve ARE?
2. How to design Q and R to meet the design requirement?

- Since this is the 6th order system, it is recommended to use the eigenvalue and eigenvector method to solve the ARE. Of course with MATLAB.
- Don't worry, if LQR problem is set in your final exam, the order of the system will be lower than 3.
- There is no simple formula to relate the transient response to weight matrices Q and R. You need to vary Q and R and see its effect.

- For this part, you need to check whether the transient specifications are met by using the controller:

$$u = -Kx + r$$

where r is step input, as discussed earlier.

3) Assume you can only measure the three outputs. Design a state observer, simulate the resultant observer-based LQR control system, monitor the state estimation error, investigate effects of observer poles on state estimation error and closed-loop control performance. In this step, both the disturbance and set point can be assumed to be zero. (10 points)

- For this part, you can build either full-order or reduced order observer.

- When evaluating the performance, you only need to compare it to the result in 2) where states are all available and the case the initial condition is not zero, while r=0.

No need to evaluate the step response for this part.

4) Suppose we are only interested in the two outputs $d(t)$ and $\psi(t)$, then we get a 2-input-2-output system. Design a decoupling controller with closed-loop stability and simulate the step set point response of the resultant control system to verify decoupling performance with stability. In this step, the disturbance can be assumed to be zero. Is the decoupled system internally stable?

(10 points)

For this part, you can try either state feedback controller or output feedback controller, or both.

When you check whether the system is decoupled, set the reference r as [1,0] and [0,1], and observe whether only one output is affected or not. In this step, set the initial condition to be zero.

Internally stable means that all the six state variables are stable (not blowing up) when the initial state is not zero. You need to set the initial condition to be non-zero and apply the zero reference input r=0, and observe all the state variables.

5) Assume that the operating set point for the three outputs is

$$y_{sp} = -\frac{1}{10} C A^{-1} B \begin{bmatrix} -0.5 + (a-b)/20 \\ 0.1 + (b-c)/(a+d+10) \end{bmatrix}$$

where *a*, *b*, *c*, *d* are still the last four digits in your matriculation number, as defined above. Therefore, the objective of the controller is to maintain the output vector around this operating set point as close as possible.
Assume that you only have three cheap sensors to measure the output. Design a controller such that the plant (vehicle) can operate around the set point as close as possible at steady state even when step disturbances are present at the plant input. Plot out both the control and output signals. (10 points)

You can just follow the design procedure mentioned before:

- First you just assume that you can measure all the state variables, and design the servo control. If you can do this, then the rest is simple.
- Build an observer, and replace the states with their estimates in the controller.
- The controller not only can make all the outputs converge to the operating setting set point, but also meet the transient specifications. You need to verify both transient specifications and steady state requirements.

- The step disturbance can be injected as load disturbance (added to the input signal directly)

$$u = -Kx + r + w$$

6) We have learned about the multivariable integral control using state space model in Chapter 9. It is a classical way to solve the set point tracking problem even when a constant disturbance is involved. Now for the two-wheeled vehicle, can we maintain the three outputs at an **arbitrary constant set point** with zero steady-state error? You can try the integral control method or any other method you figure out. You can use simulations to test various set points and see the results. Please give a formal mathematical analysis/proof for your conclusion. **(10 points)**

In part 5), you have already built a controller to drive the plant to the given operating set point. Now you can try some other different set points to see whether your controller still succeeds. Or maybe you'd like to try other controllers (which are not limited to ones covered in my lecture notes) as well.

After some attempts, you may get a conclusion. To generalize your conclusion to an arbitrary set point, please provide a formal mathematical analysis. Hint: a simple mathematical proof will be enough.

This is the most difficult part of the mini-project because you cannot find the answer directly from the lecture notes. Extra efforts are needed to get it done.

Project Briefing

In MATLAB, there are many commands for control systems design such as "acker", "place", "lqr" and "care".

It is NOT acceptable to state in the report that you design the control system simply using these MATLAB commands in one single line.

In the process of trial and error experiments, you can use those commands to save the search time. Once you have found out the final design for the desired poles or Q and R to meet the requirements, you need to detail the calculation steps following the lecture notes step by step.

You should try your best to work out the numbers by hand and calculator.

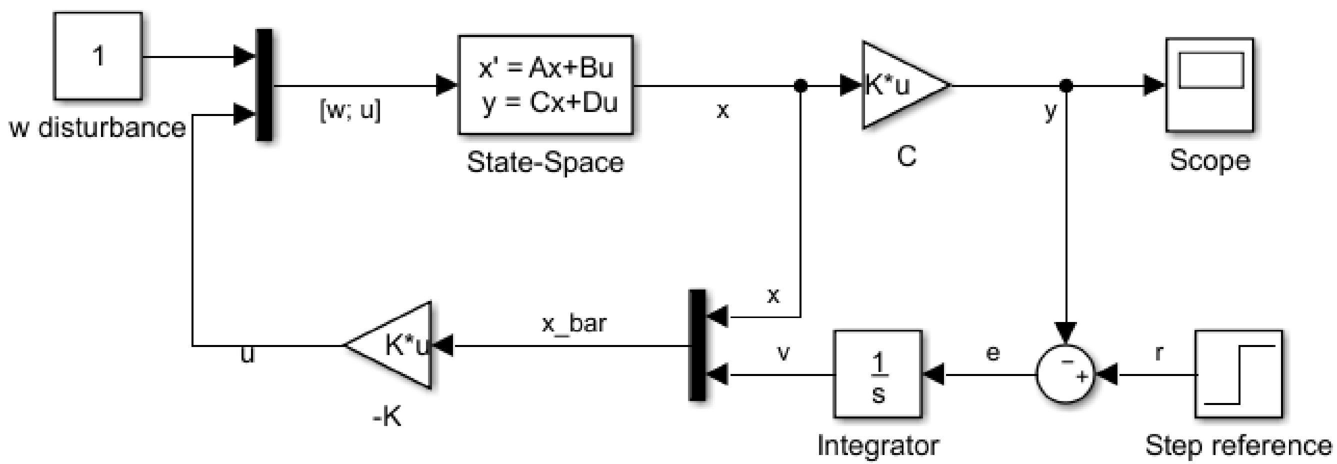
In this way, you will master the whole design procedure.

If you can only rely on MATLAB command to get the numbers, you will be in trouble in the final exam.

You'd better use Simulink to generate all those input and output signals

If you have no experience using Simulink in the past, you can learn it quickly.
A simple way to get familiar with Simulink is trying to reproduce some of the results shown in the lecture notes.

For instance, the Simulink model for showing the servo control result for example 5 (pages 66-68 in chapter 9) is given in the slides:



You just try it out and see whether you can get the same plots, and then play around with all those blocks to know how to specify the parameters for each.

Another important thing is to use "format long" in MATLAB to increase the numerical accuracy. The results from most of the numerical methods are not precise. For example, instead of "ZERO", you may get something like 10^(-14).

Tips for preparing the final exam:

There are past exam papers in the library, you should take a look.
There are 2 questions from part I and 2 from part II.

If you can work out the solutions to the first 5 questions of the mini-project and all the tutorial questions by yourself, you are guaranteed to score high for the two questions of part II in the final exam.

Due to time constraint, some topics will not be tested in the exam paper.

However, all the topics have equal probability to appear in the exam paper.

DO NOT use past exam papers to bet that some topics will never appear!

Tips for preparing the final exam:

Other tips:

1. We are well aware the time constraint of the exam. Therefore, the numerical calculations are simple. If your solutions are very complicated and tedious, then something must be wrong!

2. Do not leave it blank for the questions you do know how to solve. Try to write down something relevant.

3. The exam is closed book, but you are allowed to bring one A-4 size help sheet. The help sheet can be double-sided. You can handwrite or print course material. There is no limit on the font size in the printed paper, but you are not allowed to use a magnifier.

4. The Laplace transform table will be provided in the exam paper. No need to print it in the help sheet.

Q&A

This is it!

THANK YOU !