Module 2-1: Pole Placement

Linear Control Systems (2020)

Ding Zhao

Assistant Professor

College of Engineering

School of Computer Science

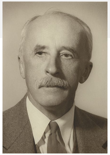
Carnegie Mellon University

Table of Contents

Table of Contents

PID Control

- Algorithms: $u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{de(t)}{dt}$
- Transfer function: $L(s) = K_p + K_i/s + K_d s$



Minorsky, 1885-1970

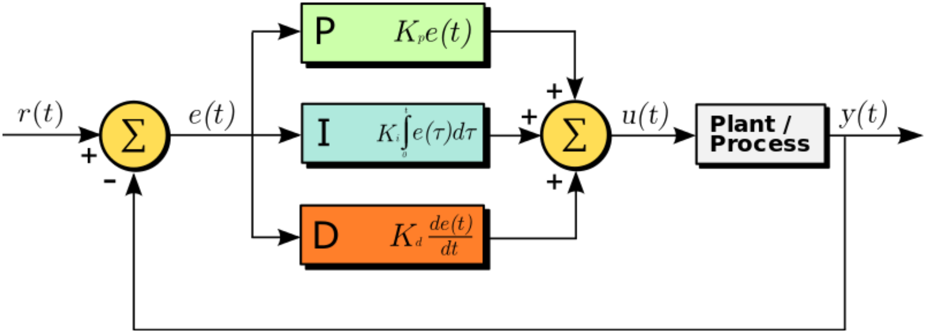


image Courtesy a Arturo Urquizo

Intuition of PID

Take the example of lateral control of a car. Define Cross Track Error (CTE) as the distance of the car from trajectory.

Recap: Linear CONTROL Systems - A Brief History

Control:continuously operating dynamical systems

- **1920s**: Minorsky worked on automatic controllers (PID) for steering ships
- **1930s**: Nyquist/Bode (Bell Lab) developed methods for analyzing the stability of controlled systems
- **1940s**: Frequency response methods made it possible to design linear closed-loop Norbert Wiener (Cybernetics)
- **1950s**: Root-locus method due to Evans was fully developed
- **1960s**: State space methods linear model-based (MIMO) optimal/stochastic/ adaptive control Rudolf Kalman (Apollo)
- **1980s**: Machine learning controls nolinear model-free genetic algorithm, neural network reinforcement learning control
- **Present**: modern control theory in engr,bio,med,econ,social etc.

A Breif History

- In 1922, Minorsky helped in the installation and testing of automatic steering on board the battleship USS New Mexico. In relation to this work Minorsky authored a paper introducing the concept of Integral and Derivative Control. This paper is one of the earliest formal discussions on control theory in the English language.
- 1924–1934 Nicolas Minorsky was a Professor of Electronics and Applied Physics at Penn.
- Navy requests to investigate anti-rolling devices on ships. The ability to stabilize a ship such as an aircraft carrier would be extremely useful during the landing of airplanes.
- 1934 to 1940, Minorsky worked on roll stabilization of ships for the navy, designing an activated-tank stabilization system into a 5-ton model ship, later on dubbed as "USS Minorsky".
- A full-scale version of the system was tested in the USS Hamilton but exhibited control stability problems. Very promising results were beginning to appear when the outbreak of the Second World War interrupted further development as the Hamilton was called to active duty and the 5 ton model was put into storage.
- In 1942, Ziegler and Nichols introduced tuning rules.

Tuning method: Ziegler–Nichols Method

The ultimate gain K_u: set K_I and K_D to 0, increase K_P until the output of the control loop has stable and consistent oscillations to get K_u and the oscillation period T_u. Z-N yields an aggressive gain and overshoot – some applications wish to instead minimize or eliminate overshoot, and for these this method is inappropriate. In this case, use the last two rows.

Ziegler–Nichols method[1]

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$T_u/1.2$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$T_u/8$	–	$K_uT_u/10$
classic PID[2]	$0.6K_u$	$T_u/2$	$T_u/8$	$1.2K_u/T_u$	$3K_uT_u/40$
Pessen Integral Rule[2]	$7K_u/10$	$2T_u/5$	$3T_u/20$	$1.75K_u/T_u$	$21K_uT_u/200$
some overshoot[2]	$K_u/3$	$T_u/2$	$T_u/3$	$0.666K_u/T_u$	$K_uT_u/9$
no overshoot[2]	$K_u/5$	$T_u/2$	$T_u/3$	$(2/5)K_u/T_u$	$K_uT_u/15$

Tuning method: Twiddle Algorithm

```
# Choose an initialization parameter vector
p = [0, 0, 0]
# Define potential changes
dp = [1, 1, 1]
# Calculate the error
best_err = robot(p)
threshold = 0.001
while sum(dp) > threshold:
    for i in range(len(p)):
        p[i] += dp[i]
        err = robot(p)

        if err < best_err:
        # There was some improvement
            best_err = err
            dp[i] *= 1.1
```

```
else:  # There was no improvement
    p[i] -= 2 * dp[i]
    # Go into the other direction
    err = robot(p)

    if err < best_err:
    # There was an improvement
        best_err = err
        dp[i] *= 1.05
    else:  # There was no improvement
        p[i] += dp[i]
        # As there was no improvement,
        # the step size in either direction,
        # the step size might simply be too big.
        dp[i] *= 0.95
```

Courtesy at Martin Thoma

Table of Contents

Feedback Control

- Benefits of feedback
  1. Stabilize unstable systems
  2. Improve transient response (speed)
  3. Reject disturbances
  4. Increase robustness to modeling errors
- We will consider "state feedback" $\Rightarrow u = Kx$
  - For systems that do not output all the states, we will estimate unmeasured states with an observer

SISO State Feedback

Consider the LTI SISO system

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

and $u = Kx + Ev$ (Feed forward. We will see how to design it later.)
$\Rightarrow \dot{x} = (A + bK)x + bEv, y = (c + dK)x + dEv$
where, $A + bK = A_{fb}$

- We know that the stability of the system depends on the eigenvalues of A_{fb}
- Goal: Given a set of desired eigenvalues $\{\lambda_d\}$, design K s.t. $A + bK$ has eigenvalues λ_d.

Recap: Controllable Canonical Forms for the SISO System

Given a SISO system: $G(s) = \dfrac{b_n s^n + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & \cdots & \cdots & -a_{n-1} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} b_0 - b_n a_0 & b_1 - b_n a_1 & \cdots & b_{n-1} - b_n a_{n-1} \end{bmatrix} x + b_n u$$

We can also arrive at this form via a similarity transformation using M_c.

$$M_c = P P_c^{-1}$$

Feedback with Controllable Canonical Form

- The easiest way to design SISO feedback controllers is to start with controllable canonical form (of course, given the system is controllable).
- Let

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \tilde{x} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

Let $u = \tilde{K}\tilde{x} = \begin{bmatrix} \tilde{k}_0 & \tilde{k}_1 & \cdots & \tilde{k}_{n-1} \end{bmatrix} \tilde{x}$

$\Rightarrow \tilde{B}\tilde{K} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}^T \begin{bmatrix} \tilde{k}_0 & \tilde{k}_1 & \cdots & \tilde{k}_{n-1} \end{bmatrix}$

Pole Placement with Controllable Canonical Form

$$\Rightarrow \tilde{A} + \tilde{B}\tilde{K} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 + \tilde{k}_0 & -a_1 + \tilde{k}_1 & -a_2 + \tilde{k}_2 & \cdots & -a_{n-1} + \tilde{k}_{n-1} \end{bmatrix}$$

- From Controllable Canonical Form, the characteristic equation for $\tilde{A} + \tilde{B}\tilde{K}$ is

$$\Delta(s) = s^n + (a_{n-1} - \tilde{k}_{n-1})s^{n-1} + \cdots + (a_1 - \tilde{k}_1)s + (a_0 - \tilde{k}_0)$$

$\Rightarrow$ By choosing $\tilde{K}$, we can get any $\Delta(s)$ we want $\Rightarrow$ any eigenvalues!

Controllable Canonical Form

- We still need to convert $\tilde{K}$ to K to work with the original system form (similarity transformation)
- Let the original system be x and the Controllable Canonical Form system be $\tilde{x}$. Then $x = M\tilde{x}$ with $M = PP_c^{-1}$. Given $\tilde{K}$ designed for $\tilde{A}$, we can calculate K.

$$
\begin{aligned}
u &= Kx = \tilde{K}\tilde{x} = KM\tilde{x} \\
K &= \tilde{K}M^{-1}
\end{aligned}
$$

Example

- Place poles of

$$\dot{x} = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \text{ at -5 and -6}$$

- Solution:

$$\Delta(s) = (s-1)(s-2) - 12 = s^2 - 3s - 10 \qquad \dot{\tilde{x}} = \begin{bmatrix} 0 & 1 \\ 10 & 3 \end{bmatrix} \tilde{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$P = \begin{bmatrix} 1 & 4 \\ 1 & 6 \end{bmatrix}, P_c = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}, P_c^{-1} = \begin{bmatrix} -3 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow M = PP_c^{-1} = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix}, M^{-1} = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ 3 & -1 \end{bmatrix}$$

$$(\lambda+5)(\lambda+6) = \lambda^2 + 11\lambda + 30 = \lambda^2 + (-3-k_1)\lambda + (-10-k_0)$$

$$\Rightarrow \tilde{k}_1 = -14, \tilde{k}_0 = -40 \Rightarrow \tilde{K} = \begin{bmatrix} -40 & -14 \end{bmatrix}$$

$$\Rightarrow K = \begin{bmatrix} -40 & -14 \end{bmatrix} M^{-1} \Rightarrow K = \begin{bmatrix} -1 & -13 \end{bmatrix}$$

Example

- It's not hard to verify that the closed loop system has the desired eigenvalues.
- Clearly if the system is controllable, we can place the poles... what if it's not?

Example

$$\dot{x} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 3 \end{bmatrix} u$$

Controllable? No - by observing the Jordan blocks.

- Solution:

Let $K = \begin{bmatrix} k_0 & k_1 \end{bmatrix}$ and $u = Kx \Rightarrow A + bK = \begin{bmatrix} -2 + k_0 & 3k_1 \\ k_0 & -2 + 3k_1 \end{bmatrix}$

$$\Rightarrow |\lambda I - (A + bK)| = \left| \begin{bmatrix} \lambda + 2 - k_0 & -3k_1 \\ -k_0 & \lambda + 2 - 3k_1 \end{bmatrix} \right|$$

$$= (\lambda + 2 - k_0)(\lambda + 2 - 3k_1) - 3k_0 k_1 = (\lambda + 2)^2 - (k_0 + 3k_1)(\lambda + 2)$$

$$= (\lambda + 2)(\lambda + (2 - k_0 - 3k_1))$$

In general, one or more eigenvalues of an uncontrollable system will be unaffected by state feedback.

Table of Contents

SISO State Observers

- We have used the full state vector in feedback, but really only measure y.
- An observer takes as inputs u and y returns an estimator of x.
- Why not just use $\dot{\tilde{x}} = A\tilde{x} + Bu$, $\tilde{y} = C\tilde{x} + Du$, $\tilde{x}(t_0) = x(t_0)$?
  - This only replies on model-based prediction, it can be used used in a short period of time, but we also need to correction to deal with modeling errors/disturbance.
  - Only replying on prediction is akin to ruining with eyes closed.

Dynamics of SISO State Observers

- Let's try the following to get y involved

$$\begin{aligned}
\dot{\tilde{x}} &= A\tilde{x} + Bu + L(y - \tilde{y}) \\
&= A\tilde{x} + Bu + L(y - (C\tilde{x} + Du)) \\
&= (A - LC)\tilde{x} + (B - LD)u + Ly
\end{aligned}$$

- Define the error $e = x - \tilde{x}$

$$\begin{aligned}
\Rightarrow \dot{e} &= Ax + Bu - (A - LC)\tilde{x} - (B - LD)u - L(Cx + Du) \\
&= Ae - LCe = (A - LC)e \\
\Rightarrow \dot{e} &= (A - LC)e
\end{aligned}$$

- Based on stability theory, $e \to 0$ as $t \to \infty \Leftrightarrow A - LC$ has poles in the LHP

Recap: Observable Cononical Form for the SISO System

$$\dot{x} = \begin{bmatrix} -a_{n-1} & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -a_1 & 0 & 0 & \cdots & 1 \\ -a_0 & 0 & \cdots & \cdots & 0 \end{bmatrix} x + \begin{bmatrix} b_{n-1} - b_n a_{n-1} \\ \vdots \\ b_1 - b_n a_1 \\ b_0 - b_n a_0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} x + b_n u$$

Similarly: we could arrive at the observable canonical via similarity transformation: $M_o = Q^{-1} Q_o$

Pole Placemnet for SISO Observers in Observable Canonical Form

We can arbitrarily place the poles ⇔ (A,C) is observable.

$$\tilde{A} - \tilde{L}\tilde{C} = \begin{bmatrix} -\tilde{a}_{n-1} & 1 & 0 & \cdots & 0 \\ -\tilde{a}_{n-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ -\tilde{a}_1 & 0 & 0 & \cdots & 1 \\ -\tilde{a}_0 & 0 & 0 & \cdots & 0 \end{bmatrix} - \begin{bmatrix} \tilde{l}_{n-1} \\ \vdots \\ \tilde{l}_0 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\tilde{a}_{n-1} - \tilde{l}_{n-1} & 1 & 0 & \cdots & 0 \\ -\tilde{a}_{n-2} - \tilde{l}_{n-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ -\tilde{a}_1 - \tilde{l}_1 & 0 & 0 & \cdots & 1 \\ -\tilde{a}_0 - \tilde{l}_0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Can achieve arbitrary pole placement via choice of $\tilde{L}$! To solve original eigenvalue problem, convert via $L = M\tilde{L}$ with $M = Q^{-1}Q_o$.

Example

- Design an observer for $A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}$ with poles at -10, -20
- Solution:

$$\begin{aligned}
\Delta(s) &= (s-1)(s-2) - 12 = s^2 - 3s - 10 \\
\Delta_{fb}(s) &= (s+10)(s+20) = s^2 + 30s + 200 \\
&= s^2 + (-3 + \tilde{l}_1)s + (-10 + \tilde{l}_0) \\
&\Rightarrow \tilde{L} = \begin{bmatrix} 33 \\ 210 \end{bmatrix}
\end{aligned}$$

Now convert back to original variables

$$\tilde{A} = \begin{bmatrix} 3 & 1 \\ 10 & 0 \end{bmatrix}, \; M = Q^{-1}Q_o = \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

$$L = M\tilde{L} = \begin{bmatrix} 33 \\ 92 \end{bmatrix}$$

Observer Structure

For SISO, the observer can be expressed as a 2-input ($[u, y - \tilde{y}]$) and (n+1)-output system ($[\tilde{x}, \tilde{y}]$)

-

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + Bu + L(y - \tilde{y}) \\ &= A\tilde{x} + \begin{bmatrix} B & L \end{bmatrix} \begin{bmatrix} u & y - \tilde{y} \end{bmatrix}^T \\ y_o = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} &= \begin{bmatrix} I_n \\ C \end{bmatrix} \tilde{x} + \begin{bmatrix} 0_n \\ D \end{bmatrix} u \end{aligned}$$

Table of Contents

SISO Feedback Control with Observer

By using an observer and performing state feedback on $\tilde{x}$, we can build an output feedback controller.

$$u = K\tilde{x}$$
$$\Rightarrow \dot{x} = Ax + BK\tilde{x}, y = Cx + Du$$

$$\dot{\tilde{x}} = A\tilde{x} + Bu + L(y - \tilde{y})$$
$$\tilde{y} = C\tilde{x} + DK\tilde{x}$$
$$\Rightarrow \dot{\tilde{x}} = A\tilde{x} + BK\tilde{x} + L(Cx + DK\tilde{x} - (C\tilde{x} + DK\tilde{x}))$$
$$= (A - LC + BK)\tilde{x} + LCx$$

- Now stack $\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A & BK \\ LC & A - LC + BK \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$

SISO Feedback Control

- Transfom via $M_e = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}$. $M_e^{-1} = M_e$

$$
\begin{aligned}
\begin{bmatrix} x(t) \\ e(t) \end{bmatrix} &= \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \\
\Rightarrow \begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} &= \begin{bmatrix} I & O \\ I & -I \end{bmatrix} \begin{bmatrix} A & BK \\ LC & A - LC + BK \end{bmatrix} \begin{bmatrix} I & O \\ I & -I \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \\
&= \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}
\end{aligned}
$$

Separation Principle

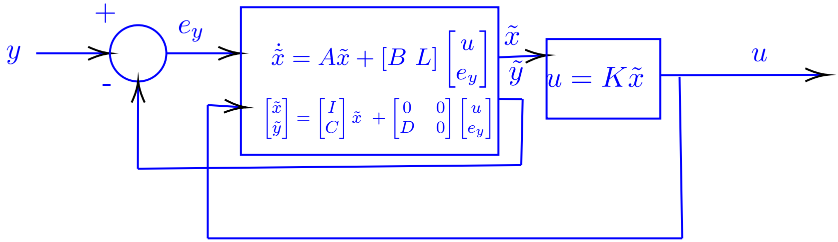
Can show through properties of the determinant that the characteristic equations is:

$$\Delta(s) = |sI - (A - LC)| \cdot |sI - (A + BK)|$$

$\Rightarrow$ The eigenvalues of the augmented system are the same as the separate controller/observer.
-This is called the **separation principle**. Observers & controllers can be designed separately.
-However, we should be careful to make the observer faster (3-5 times to the left on complex plane) than the controller.

Structure of the Full Controller

The full controller takes the form



Controller Transfer Functions

-Let's now look at the controller TF to compare with classical methods,

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + BK\tilde{x} + L(y - (C\tilde{x} + DK\tilde{x})) \\ &= (A + BK - LC - LDK)\tilde{x} + Ly \\ u &= K\tilde{x} \end{aligned}$$

$\Rightarrow C(s) = \frac{U(s)}{Y(s)} = K(sI - (A + BK - LC - LDK))^{-1}L$

-For a full state observer, the controller has order n

Table of Contents

Reduced Order Observer

-In some cases, the output contains direct measurement of a subset of the state variables - no need to estimate those.

-The transformation $M^{-1} = \begin{bmatrix} q \text{ linearly independent rows of } C \\ n-q \text{ additional ind. rows} \end{bmatrix}$ transforms any system into the form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$y = \begin{bmatrix} I_{q \times q} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du = x_1 + Du$$

Reduced Order Observer

Note that x_1 is measured by y & we can rewrite as

$$\begin{aligned} A_{12}x_2 &= \dot{x}_1 - A_{11}x_1 - B_1u \\ \dot{x}_2 &= A_{22}x_2 + B_2u + A_{21}x_1 \\ \text{Define: } \overline{u} &\triangleq A_{21}x_1 + B_2u \\ \overline{y} &\triangleq \dot{x}_1 - A_{11}x_1 - B_1u \\ \Rightarrow \dot{x}_2 &= A_{22}x_2 + \overline{u} \\ \overline{y} &= A_{12}x_2 \end{aligned}$$

We can now observe x_2 via

$$\dot{\tilde{x}}_2 = \overbrace{A_{22}\tilde{x}_2 + \overline{u}}^{\text{prediction}} + \overbrace{L(\overline{y} - \tilde{\overline{y}})}^{\text{correction}} = (A_{22} - LA_{12})\tilde{x}_2 + \overline{u} + L\overline{y}$$

where $\tilde{\bar{y}} = A_{12}\tilde{x}_2$ and $\bar{y}$ can be obtained from direct measurement $\bar{y} = \dot{x}_1 - A_{11}x_1 - B_1u$

Reduced Order Observer

-Now look at the error

$$\begin{aligned}\dot{e} &= \dot{x}_2 - \dot{\tilde{x}}_2 \\ &= A_{22}x_2 + \overline{u} - (A_{22}\tilde{x}_2 + \overline{u} + L(A_{12}x_2 - A_{12}\tilde{x}_2)) \\ &= A_{22}(x_2 - \tilde{x}_2) - LA_{12}(x_2 - \tilde{x}_2) \\ &= (A_{22} - LA_{12})e\end{aligned}$$

Check e-values of $A_{22} - LA_{12}$ for convergence. If (A, C) observable, we could always find L to stablize the system.

Reduced Order Observer

However, what if we don't have access to $\dot{x}_1$?
Define: $z = \tilde{x}_2 - Lx_1$

$$\begin{aligned}
\dot{z} &= \dot{\tilde{x}}_2 - L\dot{x}_1 \\
&= (A_{22} - LA_{12})\tilde{x}_2 + \overline{u} + L\overline{y} - L\dot{x}_1 \\
&= (A_{22} - LA_{12})\tilde{x}_2 + \overline{u} - L(A_{11}x_1 + B_1 u) \\
\dot{z} &= (A_{22} - LA_{12})z + [(A_{22} - LA_{12})L + A_{21} - LA_{11}]x_1 + (B_2 - LB_1)u
\end{aligned}$$

We could then use it to estimate z and then get $\tilde{x}_2 = z + Lx_1$

Reduced Order Observer

Design L to place the observer eigenvalues.
Design state feedback as before with

$$\begin{bmatrix} x_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} y - Du \\ z + Lx_1 \end{bmatrix}$$

$$u = k \begin{bmatrix} x_1 \\ \tilde{x}_2 \end{bmatrix} = K \begin{bmatrix} y - Du \\ z + Lx_1 \end{bmatrix}$$

Table of Contents

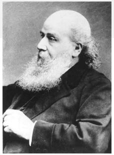
MIMO Systems

MIMO systems

- For multi-input controllers and multi-output observers, the pole placement problem does not have a unique solution.
- Fundamentally, need to solve the eigenvalue problem to set eigenvalue of $A + BK$
- Note that the observer design problem can be written in the same form by placing eigenvalue of $A^T - C^T L^T$
- (Dual Problem)

  The observability of $\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$ $\Leftrightarrow$ Controllability of $\begin{cases} \dot{x} = A^T x + C^T u \\ y = B^T x + Du \end{cases}$
- The python $scipy.signal.place\_poles$ command solves the MIMO problem as well

Design Controllers for MIMO Systems



Sylvester, 1814-1897

By hand, use a similarity transformation and solve a Lyapunov equation

1. Create an $n \times n$ matrix F with desired eigenvalues with no overlapping eigenvalues of A (restriction on this method)
2. Select an arbitrary $p \times n$ matrix $\overline{K}$ such that $\left(F, \overline{K}\right)$ is observable
3. Solve for T in $AT - TF = -B\overline{K}$ (Sylvester equation), if T is singular, go back to 2
4. Use $K = \overline{K}T^{-1}$ as feedback controller

Proof: Similarity transformation! $AT - TF = -BKT$, $(A + BK)T = TF$
$\Rightarrow A + BK = TFT^{-1} \Rightarrow$ We want $A + BK$ is similar to F

Example

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 1 & 2 & 3 \\ 2 & 1 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 2 \\ 0 & 2 \end{bmatrix}$$

$\lambda_d = -4 \pm 3j, \; -5 \pm 4j$

Write F in Controllable Canonical Form

$$\begin{aligned} \Delta(s) &= (s+4+3j)(s+4-3j)(s+5+4j)(s+5-4j) \\ &= s^4 + 18s^3 + 146s^2 + 578s + 1025 \end{aligned}$$

$$\Rightarrow F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1025 & -578 & -146 & -18 \end{bmatrix}$$

Example

Let $\overline{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ (can check $rank \left(\begin{bmatrix} \overline{K} \\ \overline{K}F \\ \overline{K}F^2 \\ \overline{K}F^3 \end{bmatrix} \right) = 4$ Now use Python

($scipy.linalg.solve\_sylvester$) to solve the Sylvester equation

$\Rightarrow K = \overline{K}T^{-1} = \begin{bmatrix} -0.2256 & 0.0641 & 0.008 & 0.5027 \\ -146.73 & -40.66 & -9.61 & -0.39 \end{bmatrix}$ Can check ...

$numpy.linalg.eig(A + BK) = -4 \pm 3j, \ -5 \pm 4j$

Example

Control is not unique! with $\overline{K} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ get

$K = \begin{bmatrix} -267.98 & -237.25 & -22.50 & 44.44 \\ -13.73 & -5.12 & 0.91 & 0.33 \end{bmatrix}$, same eigenvalues!

Design Observer for MIMO Systems

An analogous procedure exists for observers:

1. Create an $n \times n$ matrix F with desired eigenvalues $\neq$ eigenvalues of A (restriction on this method)
2. Select an arbitrary $n \times q$ matrix $\overline{L}$ such that $\left(F, \overline{L}\right)$ is controllable
3. Solve Sylvester equation $TA - FT = \overline{L}C$, if T is singular, go to (2)
4. Use $L = T^{-1}\overline{L}$ as observer gains