EE16B, Spring 2018
UC Berkeley EECS

Maharbiz and Roychowdhury

Lectures 6B & 7A: Overview Slides

Controller Canonical Form
Observability
Controller Canonical Form (CCF)

- Recall prior example: \( \dot{x}[t+1] = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix} x[t] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t] \)
- char. poly.: \( \lambda^2 - a_2 \lambda - a_1 \): nice simple formula
- Generalization: Controller Canonical Form (CCF)

\[
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & 0 & 1 \\
a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n
\end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}
\]

- char poly: \( \lambda^n - a_n \lambda^{n-1} - a_{n-1} \lambda^{n-2} - \cdots - a_2 \lambda - a_1 \)
  - not difficult to show this (though a bit tedious)
  - apply determinant formula using minors to the last row
Feedback on CCF

- System: \[ \frac{d}{dt} \vec{x}(t) = A\vec{x}(t) + \vec{b}u(t), \text{ with } (A, \vec{b}) \text{ in CCF} \]

- apply feedback \( \vec{k}: A \mapsto A - \vec{b}\vec{k}^T \)

\[ A - \vec{b}\vec{k}^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ a_1 - k_1 & a_2 - k_2 & a_3 - k_3 & \cdots & a_{n-1} - k_{n-1} & a_n - k_n \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \]

- char poly: \[ \lambda^n - (a_n - k_n)\lambda^{n-1} - (a_{n-1} - k_{n-1})\lambda^{n-2} - \cdots - (a_2 - k_2)\lambda - (a_1 - k_1) \]

\[ \Rightarrow \text{its roots are the eigenvalues that determine stability} \]
Assigning Desired Roots

- Suppose you want $\lambda_1, \lambda_2, \ldots, \lambda_n$ to be the roots
- the char. poly. should equal: $\prod_{i=1}^{n} (\lambda - \lambda_i) = \lambda^n - (\sum_{i=1}^{n} \lambda_i) \lambda^{n-1} + \cdots + \lambda_1 \lambda_2 + \cdots + \lambda_{n-1} \lambda_n + \lambda_2 (\sum_{i=3}^{n} \lambda_i) + \cdots + \lambda_n \lambda_1$
- Expand out $\prod_{i=1}^{n} (\lambda - \lambda_i) = \lambda^n - (\lambda_1 + \lambda_2 + \cdots + \lambda_n) \lambda^{n-1} + \cdots + \lambda_1 \lambda_2 \cdots \lambda_n$
- equate coefficients against $\lambda^n - (a_n - k_n) \lambda^{n-1} - (a_{n-1} - k_{n-1}) \lambda^{n-2} - \cdots - (a_2 - k_2) \lambda - (a_1 - k_1)$

We just showed: if a system is in CCF, feedback can move its eigenvalues to any desired locations.
CCF and Eigenvalue Placement: Examples

- \( A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow A - \vec{b}k^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 - k_1 & 2 - k_2 & 3 - k_3 \end{bmatrix} \)

- char. poly.: \( \lambda^3 - (3 - k_3)\lambda^2 - (2 - k_2)\lambda - (1 - k_1) \)

- desired char. poly.: \( (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \)

\( \therefore \) say we want: \( \lambda_1 = \lambda_2 = \lambda_3 = 0 \Rightarrow (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \equiv \lambda^3 \)

- then \( k_3 = 3, k_2 = 2, k_1 = 1 \)

\( \therefore \) or, if we want: \( \lambda_1 = -1, \lambda_2 = -2, \lambda_3 = -3 \)

- \( (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) \equiv \lambda^3 + 6\lambda^2 + 11\lambda + 6 \)

- \( -(3 - k_3) = 6 \)
- \( -(2 - k_2) = 11 \Rightarrow \begin{cases} k_3 = 9 \\ k_2 = 13 \\ k_1 = 7 \end{cases} \)
Converting Systems to CCF

- But CCF seems a very special/restrictive form …
- … key question: what systems are in CCF?
- A: any controllable system can be converted to CCF!

Here’s how you do it:

1. Given any state-space system:
   \[
   \frac{d}{dt} \bar{x}(t) = A \bar{x}(t) + \bar{b}u(t)
   \]

2. Form its controllability matrix:
   \[
   R_n \triangleq \begin{bmatrix}
   \bar{b}, A\bar{b}, A^2\bar{b}, \ldots, A^{n-1}\bar{b}
   \end{bmatrix}
   \]

3. Compute its inverse:
   \[
   R_n^{-1}
   \]

4. Grab the last row of \( R_n^{-1} \): call it \( \bar{q}^T \)

   \[
   R_n^{-1} = \begin{bmatrix}
   \vdots \\
   q^T
   \end{bmatrix}
   \]

   (\( \bar{q} \) is a col. vector; \( \bar{q}^T \) is a row vector)
Converting Systems to CCF (contd.)

5. Form the basis transformation matrix $T \triangleq$

6. Define $\bar{z}(t) = T \bar{x}(t) \iff \bar{x}(t) = T^{-1} \bar{z}(t)$

7. Write the system in terms of $\bar{z}(t)$:

$$\frac{d}{dt} \bar{z}(t) = T A T^{-1} \bar{z}(t) + T \tilde{b} u(t)$$

$$\bar{x}(t) = T^{-1} \bar{z}(t)$$

8. $(\hat{A}, \tilde{b})$ will be in CCF!

- Proof: see the handwritten notes

T will be full rank, hence non-singular and invertible

$\tilde{q}^T A^{n-1}$

equivalent to the original system:

$u(t) \mapsto \bar{x}(t)$ is the same

similarity transformation
Example: co-operative car control

\[ \frac{dp_1}{dt} = v_1(t) \]
\[ \frac{dp_2}{dt} = v_2(t) \]
\[ \frac{dv_1}{dt} = a_1(t) \]
\[ \frac{dv_2}{dt} = a_2(t) \]

Define:
\[ x_1(t) = p_1(t) - p_2(t) - \delta \]
\[ x_1(t) + \delta \]
\[ x_2(t) = \frac{d(p_1 - p_2)}{dt} - \frac{d(v_1 - v_2)}{dt} \]

With IC = [0, 0]^T and \( u(t) = -\varepsilon \), the cars will hit each other in

\[ T = \sqrt{\frac{2\delta}{\varepsilon}} \]

\[ \frac{dx_1}{dt} = x_2(t) \]
\[ \frac{dx_2}{dt} = u(t) \]

\[ \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \]
Co-op. Car Control (contd.)

• introduce state feedback:
  • eigenvalues:
    • $\lambda_{1,2} = -\frac{k_2}{2} \pm \frac{1}{2} \sqrt{k_2^2 - 4k_1}$

• stabilization
  • $k_2 > 0, \quad k_1 > 0$ ensures eigenvalues have -ve real parts
  • small errors in the acceleration $u(t) \to$ only small changes to the desired distance $\delta$

• see handwritten notes for details
Controllable Systems can be Stabilized

- So far, we have shown that:
  - CCF systems can be stabilized by feedback
  - Controllable systems can be put in CCF

- **Controllable sys. can be stabilized by feedback**

- but not necessary to first convert to CCF to stabilize
  - just write out the char. poly. of $A - \vec{b}\vec{k}^T$ directly
    - will be a linear expression in $k_1, k_2, \ldots, k_n$
  - match coeffs. of $\lambda^k$ against those of $(\lambda - \lambda_1)(\lambda - \lambda_2) \cdots (\lambda - \lambda_n)$
    - will obtain a linear system of equations in $\vec{k}$: $M\vec{k} = \vec{r}$
  - solve $M\vec{k} = \vec{r}$ for $\vec{k}$ (usually numerically) determined by the entries of $A$, $b$, and by $\lambda_1$, $\ldots$, $\lambda_n$
Observability [Back to Discrete]

- Suppose we have just a **SCALAR output** $y[t]$
- i.e., don’t have access to all of $\tilde{x}[t]$ for feedback
- Can we recover $\tilde{x}[t]$ just from observations of $y[t]$?

More precisely:
- Suppose we know: $A$, $\hat{b}$, $\hat{c}^T$ and $u[t]$
- And can measure $y(t)$
- Can we recover $\tilde{x}[t]$?

If yes: the system is called **OBSERVABLE**
The Observability Matrix

- We know that:

\[ \bar{x}[t] = \begin{bmatrix} A^{t-1} & 0 \\ 0 & I \end{bmatrix} \bar{x}[0] + \sum_{i=1}^{t} A^{t-i} \bar{b} u[i - 1] \]

- Suppose \( u[t] = 0 \)

- Then \( \bar{x}[t] = A^{t-1} \bar{x}[0] \). Write out:

\[ y[t] = \bar{c}^T \bar{x}[t] \]

The observability matrix (\( n \times n \)) must be full-rank/non-singular/invertible to recover \( \bar{x}(t) \) uniquely from measurements of \( y(t) \).
Observability: An Example

Each application of $A$ rotates by $\theta$.

\[
\begin{bmatrix}
  x_1[t + 1] \\
  x_2[t + 1]
\end{bmatrix} = \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) \\
  \sin(\theta) & \cos(\theta)
\end{bmatrix} \begin{bmatrix}
  x_1[t] \\
  x_2[t]
\end{bmatrix}
\]

This is a "rotation matrix" - call it $A$. 

\[A^3 \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} \quad A^2 \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} \quad A \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix}\]
Observability: Example (contd.)

\[
\begin{bmatrix}
x_1[t + 1] \\
x_2[t + 1]
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
x_1[t] \\
x_2[t]
\end{bmatrix},
\]

\[
y[t] = x_1[t] = \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{x}[t]
\]

- **Observability matrix:**
  \[
  O \triangleq \begin{bmatrix}
  \leftarrow \bar{c}^T \rightarrow \\
  \leftarrow \bar{c}^T A \rightarrow
  \end{bmatrix} = \begin{bmatrix}
  1 & 0 \\
  \cos(\theta) & -\sin(\theta)
  \end{bmatrix}
  \]

- **Determinant of O:**
  \[
  \text{det}(O) = -\sin(\theta)
  \]
  - non-zero if \( \theta \neq 0, \pi, 2\pi, \cdots, i\pi \rightarrow \text{observable}
  - \( \theta = i\pi \rightarrow \text{not observable} \)
  - cannot recover \( x_2 \) uniquely
Observers

- Can we make a **system** that recovers $\bar{x}[t]$ from $y[t]$ in **real time**?
  - (we can use our knowledge of $A$, $\vec{b}$, $u[t]$ – and $y[t]$)
- **YES!** (if the system is observable – as it will turn out)
  - first: make a clone of the system
  - next: incorporate the **difference between the outputs of the actual system and the clone**

\[
\begin{align*}
\bar{x}[t+1] &= A\bar{x}[t] + \vec{b}u[t] \\
\bar{c}^T &
\end{align*}
\]

\[
\begin{align*}
\hat{x}[t+1] &= A\hat{x}[t] + \vec{b}u[t] \\
&\quad + \vec{l}(\bar{c}^T\hat{x}[t] - y[t]) \\
\end{align*}
\]

called an **OBSERVER**
Observers – Why/How They Work

- **Observer:** \( \hat{x}[t+1] = A\hat{x}[t] + \vec{b}u[t] + \vec{l}(c^T \hat{x}[t] - y[t]) \)
  - Define a *state prediction error*: \( \vec{e}[t] \Delta \hat{x}[t] - \vec{x}[t] \)
  - then we can derive (**move to xournal**):
    \[ \vec{e}[t+1] = (A + \vec{l}c^T)\vec{e}[t] \]
  - would like \( \vec{e}[t] \to 0 \) as \( t \) increases (i.e., \( \hat{x}[t] \to \vec{x}[t] \))
    \[ \Rightarrow \text{choose} \ \vec{l} \ \text{to make the eigenvalues of} \ A + \vec{l}c^T \text{ stable!} \]
  - strong analogy w controllability (recall \( A - \vec{b}k^T \))
    \[ \Rightarrow \text{evs of} \ A + \vec{l}c^T = \text{evs of} \ A^T + \vec{c}\vec{l}^T \to -\vec{c} \leftrightarrow \vec{b}, \ \vec{l}^T \leftrightarrow \vec{k}^T \]
  - i.e., can always make \( A + \vec{l}c^T \) stable if \( (A^T, -\vec{c}) \)
    is controllable (using previous controllability + feedback result)
Observers – Why/How (contd.)

- \((A^T, -\vec{c})\) controllable \(\rightarrow\) \\
  \[\begin{bmatrix}
  \vec{c}^T & A^T \vec{c} & \cdots & (A^T)^{n-2} \vec{c} & (A^T)^{n-1} \vec{c}
\end{bmatrix}\] must be full rank

- \(\rightarrow\) \\
  \[\begin{bmatrix}
  \vec{c}^T & A^T \vec{c} & \cdots & (A^T)^{n-2} \vec{c} & (A^T)^{n-1} \vec{c}
\end{bmatrix}^T\] must be full rank

\[\begin{bmatrix}
  \vdots \\
  \vec{c}^T A^{n-1} \\
  \vec{c}^T A^2 \\
  \vec{c}^T A \\
  \vec{c}^T
\end{bmatrix}\]

- Conclusion: if a system is observable, we can build an observer for it whose estimate \(\hat{x}[t]\) will approximate \(\bar{x}[t]\) more and more closely with \(t\)
Observer: Rotation Matrix Example

\[
\begin{bmatrix}
    x_1[t+1] \\
    x_2[t+1]
\end{bmatrix} =
\begin{bmatrix}
    \cos(\theta) & -\sin(\theta) \\
    \sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
    x_1[t] \\
    x_2[t]
\end{bmatrix},
\]

\[y[t] = x_1[t] = \begin{bmatrix} 1 & 0 \end{bmatrix} \vec{x}[t]\]

\[\begin{align*}
\text{example: } & \theta = \frac{\pi}{2} \rightarrow A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \rightarrow O = \frac{\vec{c}^T A}{\vec{c}^T} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}
\end{align*}\]

\[\text{side note: eigenvalues of A: } \pm j \rightarrow \text{BIBO unstable}\]

\[\text{let } \vec{l} = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}, \text{ then } A + \vec{l}\vec{c}^T = \begin{bmatrix} l_1 & -1 \\ 1 + l_2 & 0 \end{bmatrix}\]

\[\text{eigenvalues (see the notes): } \lambda_{1,2} = \frac{l_1}{2} \pm \frac{l_1^2 - 4(1 + l_2)}{2}\]

\[\text{and can easily show: } l_1 = \lambda_1 + \lambda_2, \quad l_2 = \lambda_1 \lambda_2 - 1\]

\[\text{i.e., can set } \vec{l} \text{ to obtain any desired eigenvalues}\]

\[\text{warning: if complex, ensure evs are complex conjugates}\]

\[\text{what will happen if you don’t?}\]
Observer: Rot. Matrix Example (contd.)

\[
\begin{bmatrix}
x_1[t + 1] \\
x_2[t + 1]
\end{bmatrix} =
\begin{bmatrix}
cos(\theta) & -sin(\theta) \\
sin(\theta) & cos(\theta)
\end{bmatrix}
\begin{bmatrix}
x_1[t] \\
x_2[t]
\end{bmatrix},
\]

\[y[t] = x_1[t] = \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{x}[t] \]

- \text{now try: } \theta = \pi \rightarrow A = \begin{bmatrix}
-1 & 0 \\
0 & -1
\end{bmatrix} \rightarrow \text{not observable (recall)}

- \text{eigenvalues (see the notes): } \lambda_1 = -1, \lambda_2 = l_1 - 1

\[A + \bar{c}^T = \begin{bmatrix}
-1 + l_1 & 0 \\
l_2 & -1
\end{bmatrix}\]

cannot be changed/stabilized using \( \bar{c} \)
Observability: The Continuous Case

• Observability for C.T. state-space systems
  • and implications for placing observer eigenvalues

• EXACTLY THE SAME CRITERIA

\[
\begin{bmatrix}
    \leftarrow & c^T & \rightarrow \\
    \leftarrow & c^T A & \rightarrow \\
    \leftarrow & c^T A^2 & \rightarrow \\
    \vdots \\
    \leftarrow & c^T A^{n-1} & \rightarrow \\
\end{bmatrix}
\]

• Stability for C.T. means Re(eigenvalues) < 0
Observers: Accurate Positioning

- Physical motion is inherently marginally stable
  - due to the relationship between position, velocity and acceleration
  \[ \dot{x} = v, \quad \dot{v} = a \]
  - small error in \(a\) → growing error in \(v\)
  - small error in \(v\) → growing error in \(x\)

- You are in a car in a featureless desert
  - you know the position where you started
  - you record your acceleration (along \(x\) and \(y\) directions)
  - to estimate your current position
    - you integrate accel./velocity to predict your current position
    - but inevitable small errors (eg, play in accelerator) make your predicted position more and more inaccurate (m. stability)
      - soon, your prediction becomes completely useless – miles from where you really are
  - NOT A VERY PRACTICALLY USEFUL WAY TO LOCATE YOURSELF
Observers for Positioning (contd.)

• Enter GPS
  • you have a GPS receiver and position calculator
    ➔ but GPS isn’t perfectly accurate either (though much better than our integration technique, aka “dead reckoning”)
      • can easily be a few 10s of feet off

• Can we combine dead reckoning and GPS
  • for better accuracy than GPS alone?
  • **YES**: feed GPS position data into an **observer**
    • stabilize the observer by choosing $\vec{l}$ wisely
    • even with perpetual small GPS and acceleration errors
      ➔ the observer’s estimate is far better than just the GPS alone!* 

• This is what all serious navigational systems use
  • with an additional twist: $\vec{l}$ keeps updating, becomes $\vec{l}[t]$
  • **this is the famous KALMAN FILTER**
    ➔ used in all rockets, drones, autonomous cars, ships, ...
Rudolf Kálmán
“inventor” of control theory: 1950s/60s

- state-space representations
- stability, controllability, observability and implications
- Kalman filter
  - initially received with “vast skepticism” - not accepted for publication!
  - later adopted by the Apollo rocket program, the Space Shuttle, submarines, cruise missiles, UAVs/drones, autonomous vehicles, ...
Who Invented Eigendecomposition?

1852 - 1858

James Joseph Sylvester (1814-97)
Arthur Cayley (1821-95)

also coined the term “matrix”

Cayley-Hamilton Theorem
Who Invented Matrices?

- known and used in **China** - before 100BC (!)
  - explained in *Nine Chapters of the Mathematical Art* (1000-100 BC) → used to solve simultaneous eqns; they knew about determinants
- 1545: brought from China to Italy (by **Cardano**)
- 1683: **Seki** (“Japan’s Newton”) used matrices
- developed in Europe by **Gauss** and many others
- finally, into its modern form by **Cayley** (mid 1800s)
Charles Proteus Steinmetz
inventor of the phasor

• “Complex Quantities and their Use in Electrical Engineering”, July 1893
• revolutionized AC circuit/transmission calculations

• suffered from hereditary dwarfism, hunchback, and hip dysplasia