Goals:
• Introduce control design concepts and classical “design patterns”
• Describe the design of state feedback controllers for linear systems
• Define reachability of a control system and give tests for reachability

Reading:
• Åström and Murray, Feedback Systems, Ch 7
Control Design Concepts

System description: single input, single output system (MIMO also OK)

\[
\dot{x} = f(x, u) \quad x \in \mathbb{R}^n, \ x(0) \text{ given} \\
y = h(x, u) \quad u \in \mathbb{R}, \ y \in \mathbb{R}
\]

Stability: stabilize the system around an equilibrium point
- Given equilibrium point \( x_e \in \mathbb{R}^n \), find control “law” \( u=\alpha(x) \) such that
  \[
  \lim_{t \to \infty} x(t) = x_e \text{ for all } x(0) \in \mathbb{R}^n
  \]

Reachability: steer the system between two points
- Given \( x_0, x_f \in \mathbb{R}^n \), find an input \( u(t) \) such that
  \[
  \dot{x} = Ax + Bu \quad \text{takes } x(t_0) = x_0 \to x(T) = x_f
  \]

Tracking: track a given output trajectory
- Given \( r = y_d(t) \), find \( u=\alpha(x,t) \) such that
  \[
  \lim_{t \to \infty} (y(t) - y_d(t)) = 0 \text{ for all } x(0) \in \mathbb{R}^n
  \]
Design Patterns for Control Systems

“Classical” control (1950s...)

- Reference input shaping
- Feedback on output error
- Compensator dynamics shape closed loop response
- Uncertainty in process dynamics \( P(s) \) + external disturbances \( (d) \) & noise \( (n) \)

Goal: output \( y(t) \) should track reference trajectory \( r(t) \)

Design typically done in “frequency domain” (final 3 weeks of CDS 110)

“Modern” (state space) control (1970s...)

- Assume dynamics are given by linear system, with known \( A, B, C, D \) matrices
- Measure the state of the system and use this to modify the input
- \( u = -K x + k_r r \)

Goal unchanged: output \( y(t) \) should track reference trajectory \( r(t) \) [often constant]
State Space Controller Design for Linear Systems

\[
\dot{x} = Ax + Bu \quad x \in \mathbb{R}^n, \; x(0) \text{ given}
\]

\[
y = Cx + Du \quad u \in \mathbb{R}, \; y \in \mathbb{R}
\]

**Goal:** find a linear control law \( u = -Kx + k_r r \) such that the closed loop system

\[
\dot{x} = Ax + Bu = (A - BK)x + Bk_r r
\]

is stable at equilibrium point \( x_e \) with \( y_e = r \).

**Remarks**

- If \( r = 0 \), control law simplifies to \( u = -Kx \) and system becomes \( \dot{x} = (A - BK)x \)
- Stability based on eigenvalues \( \Rightarrow \) use \( K \) to make eigenvalues of \((A - BK)\) stable
- Can also link eigenvalues to *performance* (eg, initial condition response)
- Question: when can we place the eigenvalues anyplace that we want?

**Theorem** The eigenvalues of \((A - BK)\) can be set to arbitrary values if and only if the pair \((A, B)\) is “reachable”.

**Python:**

\[
K = \text{ct.place}(A, B, \text{eigs}) \quad \text{Reachability: } \begin{bmatrix} B & AB & A^2B & \ldots & A^{n-1}B \end{bmatrix} \text{ full rank}
\]

(Will cover in more detail in W6)
Example: Predator Prey

System dynamics (more detailed, continuous time model)

\[
\frac{dH}{dt} = (r + u)H \left(1 - \frac{H}{k}\right) - \frac{aHL}{c + H}, \quad H \geq 0,
\]
\[
\frac{dL}{dt} = b \frac{aHL}{c + H} - dL, \quad L \geq 0.
\]

- Phase portrait: stable limit cycle with unstable equilibrium point at \( H_e = 20.6, \ L_e = 29.5 \)
- Can we design the dynamics of the system by modulating the food supply ("u" in "r + u" [formerly \( b_H(u) \)] term)

Q1: can we move from some initial population of lynxes and hares to a specified population in time \( T \) by modulation of the food supply? [= trajectory generation]
- Eg: need large amount of food for 1872 Olympics

Q2: can we stabilize the lynx population around a desired equilibrium point (eg, \( L_d = \sim 30 \))?
- Try to keep lynx and hare population in check

Approach: try to stabilize using state feedback law

\[
u = -k_1(H - H_e) - k_2(L - L_e)
\]
Equilibrium point calculation

\[
\frac{dH}{dt} = (r + u)H \left( 1 - \frac{H}{k} \right) - \frac{aHL}{c + H}
\]

\[
\frac{dL}{dt} = b \frac{aHL}{c + H} - dL
\]

- \( x_e = (20.6, 29.5), \ u_e = 0, \ L_e = 29.5 \)

Linearization

- Compute linearization around equilibrium point, \( x_e \):

\[
A = \frac{\partial f}{\partial x} \bigg|_{(x_e, u_e)} \quad B = \frac{\partial f}{\partial u} \bigg|_{(x_e, u_e)} \quad \frac{dx}{dt} \approx A(x - x_e) + B(u - u_e) + \text{higher order terms}
\]

- Redefine local variables: \( z = x - x_e, \ v = u - u_e \)

\[
\frac{d}{dt} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -\frac{acL_e}{(c+H_e)^2} - \frac{2H_e r}{k} + r & -\frac{aH_e}{c+H_e} \\ \frac{abcL_e}{(c+H_e)^2} & \frac{abH_e}{c+H_e} - d \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} H_e \left( 1 - \frac{H_e}{k} \right) \\ 0 \end{bmatrix} v
\]

- Reachable? YES, if \( a, b \neq 0 \) (check \([B \ AB]) \Rightarrow can use feedback to place eigenvalues
Example #2: Stabilization via eigenvalue assignment

\[
\begin{bmatrix}
\frac{dz_1}{dt}
\end{bmatrix} = 
\begin{bmatrix}
-\frac{abcL_e}{(c+H_e)^2} - \frac{2H_r c + \lambda}{k} + r \\
-\frac{aH_e}{c+H_e} - d
\end{bmatrix}
\begin{bmatrix}
z_1 \\
z_2
\end{bmatrix} + 
\begin{bmatrix}
H_e \left(1 - \frac{H_r}{k}\right)
\end{bmatrix} v
\]

Control design:

\[ v = -Kz = -k_1 (H - H_e) - k_2 (L - L_e) \]
\[ u = ue + K (x - x_e) \]

Place poles at stable values

- Choose \( \lambda = -0.1, -0.2 \)
- Python: \( K = \text{place}(A, B, [-0.1; -0.2]) \)

Key principle: *design of dynamics*

- Use feedback to create a stable equilibrium point

More advanced: control to desired value \( r = L_d \) (Wed)

![Control system diagram]

![Population graph]

![Phase plane diagram]
Implementation Details

Eigenvalues determine performance
- For each eigenvalue $\lambda_i = \sigma_i + j\omega_i$, get a contribution of the form
  $$y_i(t) = e^{-\sigma_i t}(a \sin(\omega_i t) + b \cos(\omega_i t))$$
- Repeated eigenvalues can give additional terms of the form $t^k e^{\sigma_i + j\omega_i} \Rightarrow$ be careful

Use observer (estimator) to determine the current state if you can’t measure it (W5)
- Estimator looks at inputs and outputs of plant and estimates the current state
- Can show that if a system is observable then you can construct and estimator
- Use the estimated state as the feedback $u = K\hat{x}$

Next week: basic theory of state estimation and observability
- Kalman filtering = optimal observers in presence of noise (W5: very briefly; more in CDS 212)
Linear Quadratic Regulator (LQR)

Rather than placing eigenvalues, can also solve optimal control problem:

\[
\dot{x} = Ax + Bu \quad x = \mathbb{R}^n \\
x(0) \text{ given} \quad u \in \mathbb{R}^p \quad J = \int_0^\infty (x^T Q x + u^T R u) \, dt
\]

Can show that optimal controller is in the form \( u = -K x \) [CDS 212]

\[
u = -K x, \quad K = R^{-1} B^T P \
\]

\[
PA + A^T P - PBR^{-1} B^T P = -Q
\]

--- State feedback (constant gain)

--- Algebraic Riccati equation

Remarks

- In Python, \( K = \text{ct.lqr}(A, B, Q, R) \)
- Require \( R > 0 \) but \( Q \geq 0 \) (+ must satisfy “observability” condition [W5])
- Alternative form: minimize “output” \( y = H x \)

\[
L = \int_0^\infty x^T H^T H x + u^T R u \, dt = \int_0^\infty \| H x \|^2 + u^T R u \, dt
\]

- Require that \((A, H)\) is observable. Intuition: if not, dynamics may not affect cost \( \Rightarrow \) ill-posed. We will study this in more detail when we cover observers
Variation: Integral Action

State feedback limitations

- Control design depends on reasonably good model (OK)
- Constant disturbance signal generates constant error: if $d \neq 0$ then $u = -Kx + d$ will push system away from equilibrium value

Approach: use integral feedback to give zero steady state (output) error

\[
\frac{d}{dt} \begin{bmatrix} x \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix} \quad \text{integral of (output) error}
\]

- Now design LQR controller for extended system (including integrator weight)
  \[ u = -Kx - k_i z + d \]
  equilibrium value $\Rightarrow y = r \Rightarrow 0$ steady state error

- Controller automatically compensates for constant disturbances by adjusting the input to make the output be identically zero
- Can adjust to incorporate feedforward gain $k_r$ [Wed]
Example: LQR for Predator Prey (w/ Integral Action)

Predator-prey dynamics with increased food supply

\[
\begin{align*}
\frac{dH}{dt} &= (r + u)H \left(1 - \frac{H}{k}\right) - \frac{aHL}{c + H}, \\
\frac{dL}{dt} &= b \frac{aHL}{c + H} - dL,
\end{align*}
\]

- Suppose \( r \) (food supply) increases
- Acts like constant disturbance on \( u \)
- Result: increased lynx population

\textbullet{} Note that hare population only depends on \( a, b, c, d \) (second equation), not \( r \) (!)

Solution: apply integral feedback to compensate for additional food supply

- \( u = -Kx - k_i z \), \( \dot{z} = L - L_e \)
- Controller now adjusts \( z \) to exactly cancel out change in \( r \)
- Design of all gains (\( K \) and \( k_i \)) can be done via eigenvalue placement or LQR
Summary: State Space Feedback

Stability: stabilize the system around an equilibrium point

- Given equilibrium point $x_e \in \mathbb{R}^n$, find control “law” $u = \alpha(x)$ such that
  \[
  \lim_{t \to \infty} x(t) = x_e \text{ for all } x(0) \in \mathbb{R}^n
  \]

\[u = Kx, \quad K = R^{-1}B^TP\]

\[PA + A^TP - PBR^{-1}B^TP = -Q\]

\[K = \text{ct.place}(A, B, \text{eigs})\]

\[K = \text{ct.lqr}(A, B, Q, R)\]