CDS 110/ChE 105: Lecture 4.1
State Feedback

Richard M. Murray

22 April 2024

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

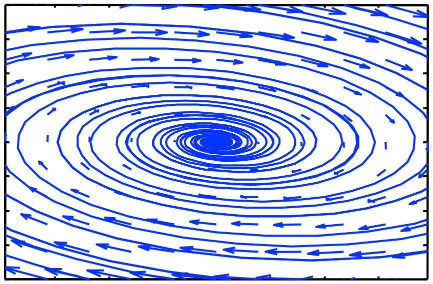
System description: single input, single output system (MIMO also OK)

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

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

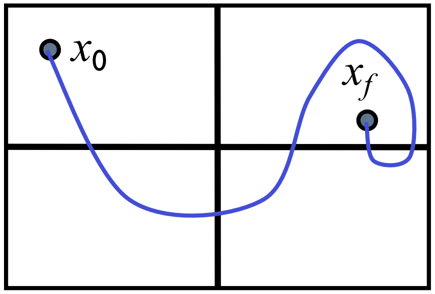
$$\lim_{t\to\infty} x(t) = x_e \text{ for all } x(0) \in \mathbb{R}^n$$



Reachability: steer the system between two points

- Given $x_o, x_f \in \mathbb{R}^n$, find an input $u(t)$ such that

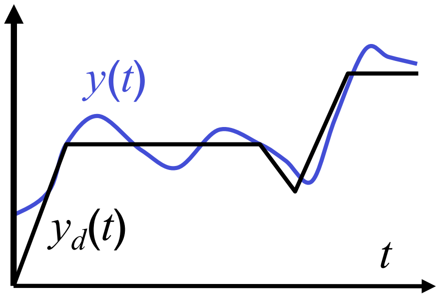
$$\dot{x} = Ax + Bu \ \text{ takes } x(t_0) = x_0 \to x(T) = x_f$$



Tracking: track a given output trajectory

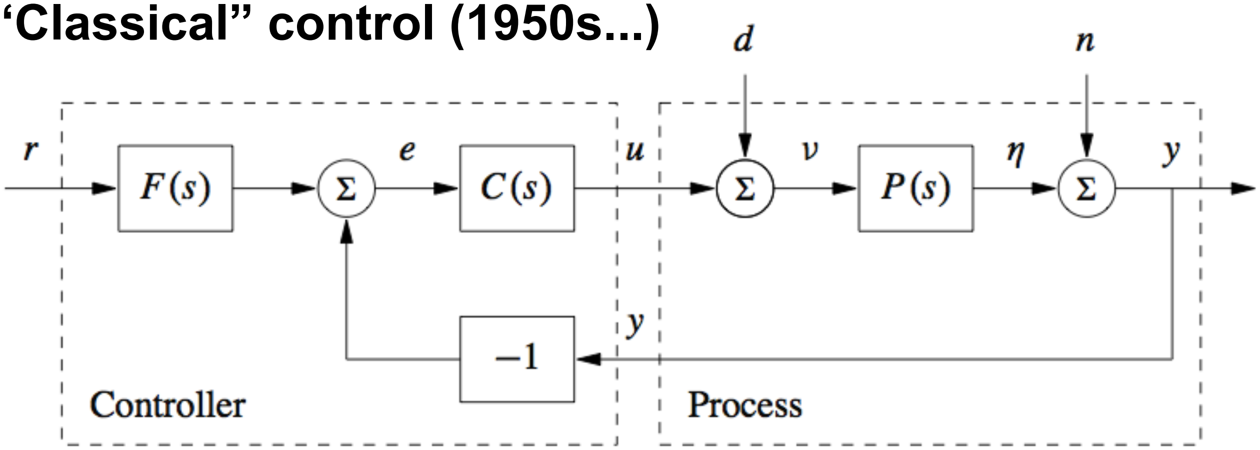
- Given $r = y_d(t)$, find $u=\alpha(x,t)$ such that

$$\lim_{t\to\infty} \big(y(t) - y_d(t)\big) = 0 \text{ for all } x(0) \in \mathbb{R}^n$$



Design Patterns for Control Systems

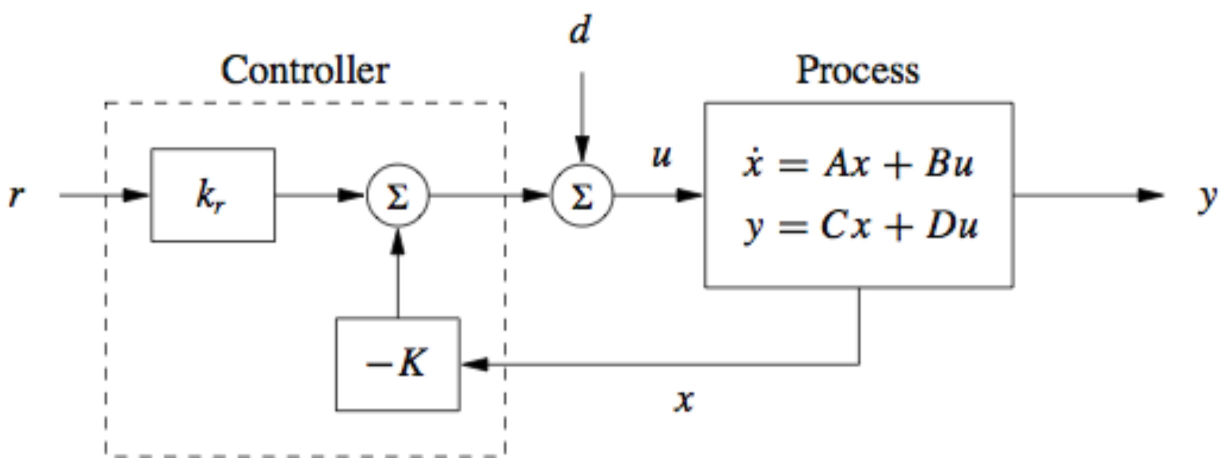
"Classical" control (1950s...)



- 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

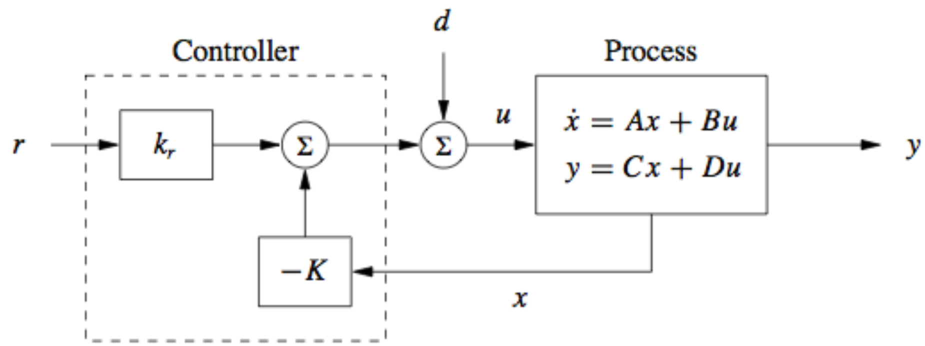
$$\dot{x} = Ax + Bu \qquad x \in \mathbb{R}^n,\ x(0) \text{ given}$$
$$y = Cx + Du \qquad u \in \mathbb{R},\ y \in \mathbb{R}$$

$$x(T) = e^{AT}x_0 + \int_{\tau=0}^{T} e^{A(T-\tau)}Bu(\tau)d\tau$$

Goal: find a linear control law $u = -K\,x + k_r\,r$ 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 $u = -K\,x$ and system becomes $\dot{x} = (A - BK)x$
- Stability based on eigenvalues ⇒ 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 = ct.place(A, B, eigs) Reachability: $\begin{bmatrix} B & AB & A^2B & \dots A^{n-1}B \end{bmatrix}$ full rank

(will cover in more detail in W6)

Example: Predator Prey

System dynamics (more detailed, continuous time model)

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

- 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]

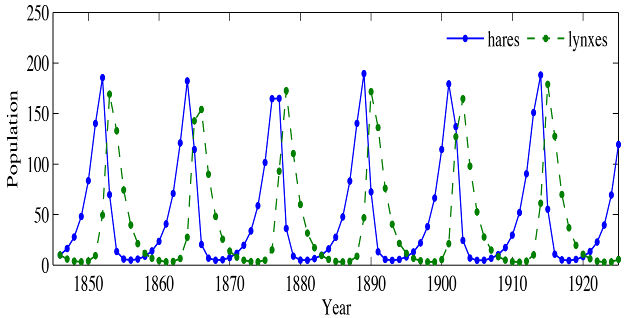
- Eg: need large amount of food for 1872 Olympics

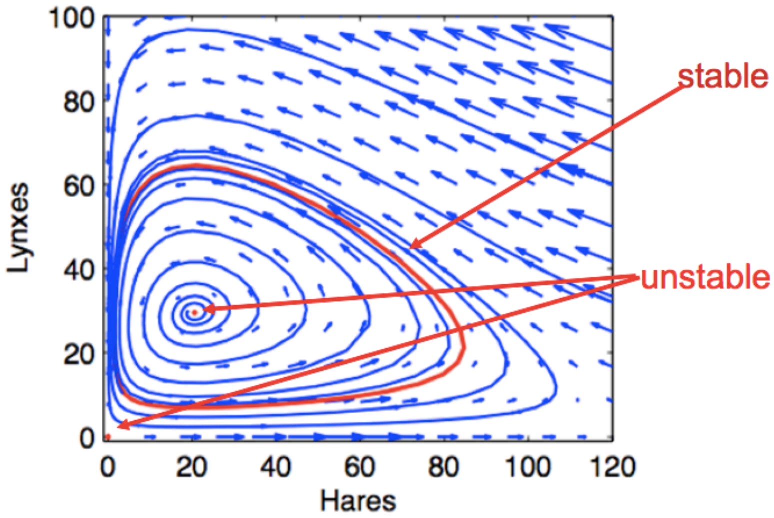
Q2: can we *stabilize* the lynx population around a desired equilibrium point (eg, L_d = ~30)?

- Try to keep lynx and hare population in check

Approach: try to stabilize using state feedback law

$$u = -k_1 (H - H_e) - k_2 (L - L_e)$$





Predator Prey Stabilization: Problem Setup

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

```
# Define system dynamics
predprey = ct.nlsys(…, outputs=['L', 'H'])

# Find the equilibrium point
xe, ue = ct.find_eqpt(predprey, [20, 30], [0])

# Generate the linearization around the eq point
linsys = predprey.linearize(xe, ue)

# Create control law
K = place(linsys.A, linsys.B, [-0.1, -0.2])
```

Linearization

- Compute linearization around equilibrium point, x_e:

$$A = \left.\frac{\partial f}{\partial x}\right|_{(x_e,u_e)} \qquad B = \left.\frac{\partial f}{\partial u}\right|_{(x_e,u_e)} \qquad \frac{dx}{dt} \approx A(x-x_e) + B(u-u_e) + \text{higher order terms}$$

- Redefine local variables: $z = x - x_e$, $v = u - u_e$

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

- Reachable? YES, if a, $b \neq 0$ (check [B AB]) ⇒ can use feedback to place eigenvalues

Example #2: Stabilization via eigenvalue assignment

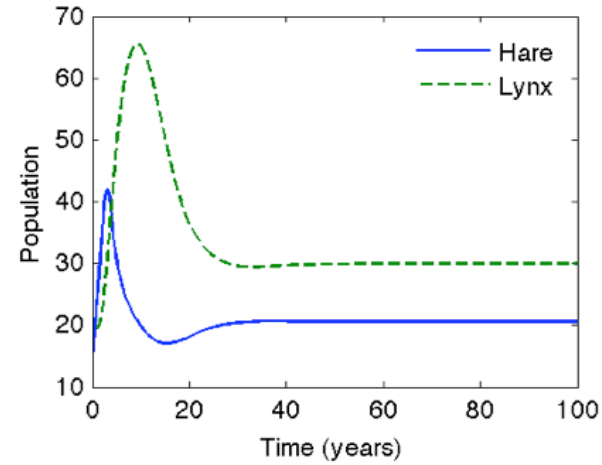
$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -\frac{acL_e}{(c+H_e)^2} - \frac{2H_e r}{k} + r & -\frac{aH_e}{c+H_e} \\ \frac{abcL_e}{(c+H_e)^2} & \frac{abH_e}{c+H_e} - d \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} H_e\left(1 - \frac{H_e}{k}\right)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

Control design:

$$v = -Kz = -k_1(H - H_e) - k_2(L - L_e)$$
$$u = u_e + K(x - x_e)$$

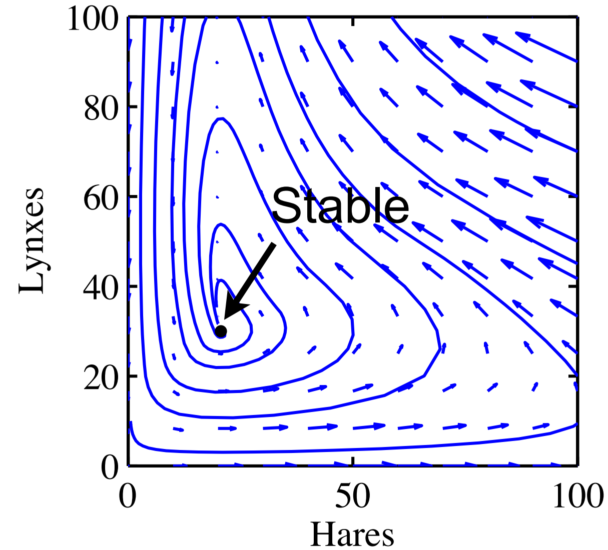
Place poles at stable values

- Choose λ = -0.1, -0.2
- Python: K = place(A, B, [-0.1; -0.2])

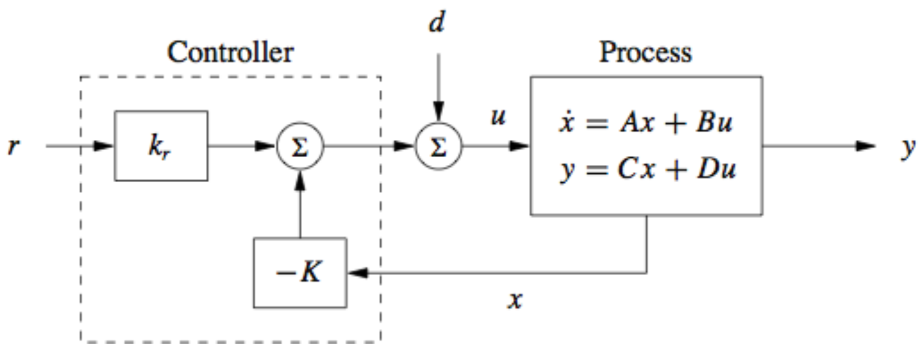


Key principle: *design of dynamics*

- Use feedback to create a stable equilibrium point



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

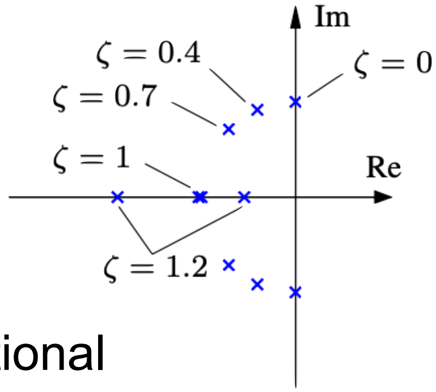


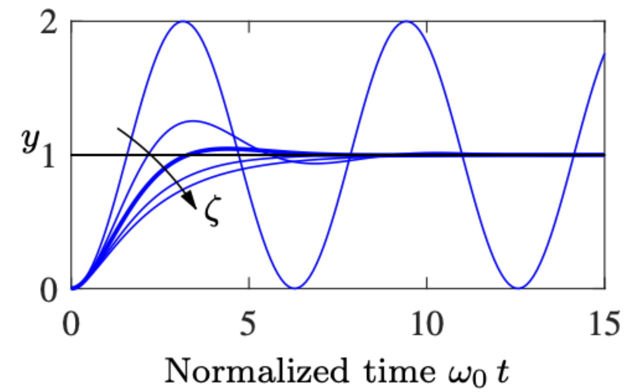
Implementation Details

Eigenvalues determine performance

- For each eigenvalue $\lambda_i = \sigma_i + j\omega_i$, get a contribution of the form

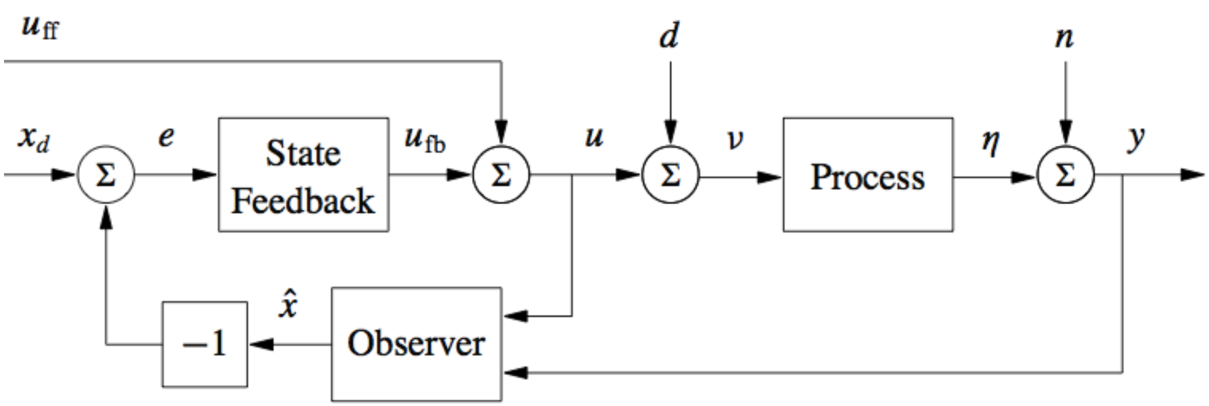
$$y_i(t) = e^{-\sigma_i t}\big(a\sin(\omega_i t) + b\cos(\omega_i t)\big)$$





- Repeated eigenvalues can give additional terms of the form $t^k e^{\sigma + j\omega}$ ⇒ 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:

$$\dot{x} = Ax + Bu \quad x = \mathbb{R}^n$$
$$x(0) \text{ given} \qquad u \in \mathbb{R}^p$$

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

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

$$u = -Kx, \qquad K = R^{-1} B^T P$$ ← State feedback (constant gain)

$$PA + A^T P - PBR^{-1} B^T P = -Q$$ ← Algebraic Riccati equation

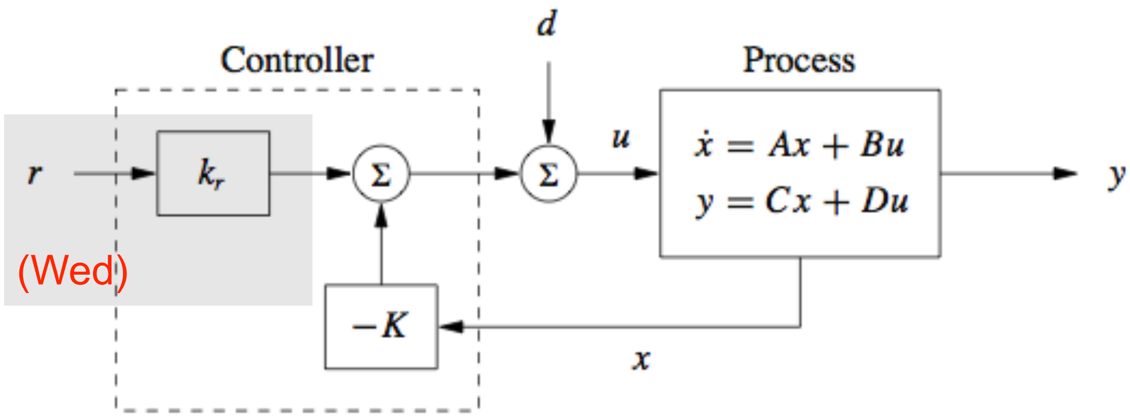
Remarks

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

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

- Require that (A, H) is observable. Intuition: if not, dynamics may not affect cost ⇒ 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 eq pt

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

$$\frac{d}{dt}\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix}$$

← 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

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

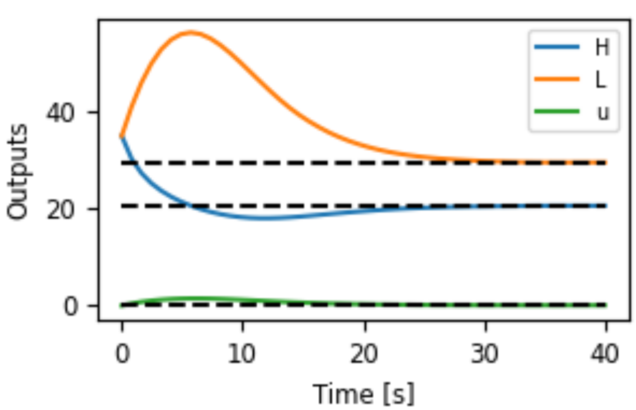
- Suppose r (food supply) increases
- Acts like constant disturbance on *u*
- Result: increased lynx population

- Note that hare population only depends on a, b, c, d (second equation), not r (!)

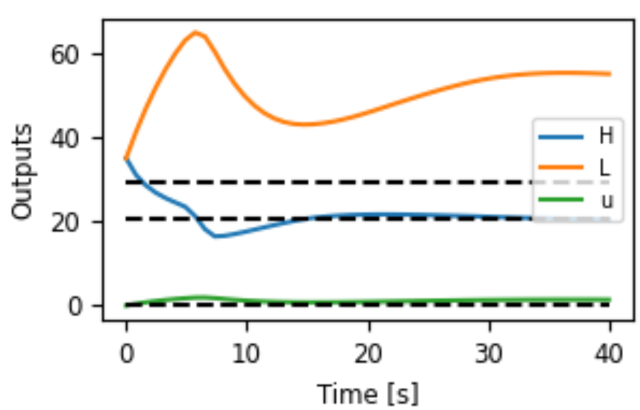
Solution: apply integral feedback to compensate for additional food supply

- $u = -Kx - k_\mathrm{i} z, \quad \dot{z} = L - L_\mathrm{e}$
- 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

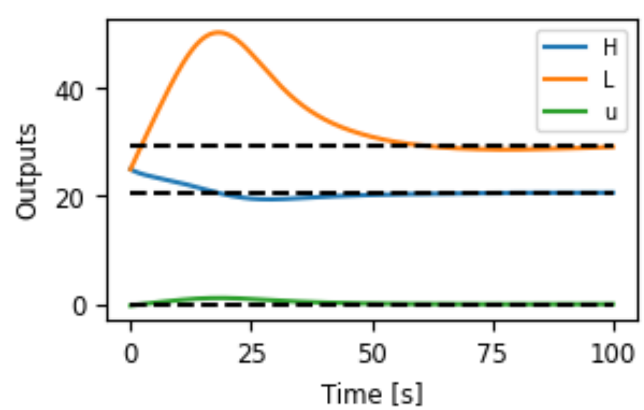
I/O response with eval placement, r = 1.6



I/O response w/ eval placement, r = 1.65



I/O response w/ integral action, r = 1.65

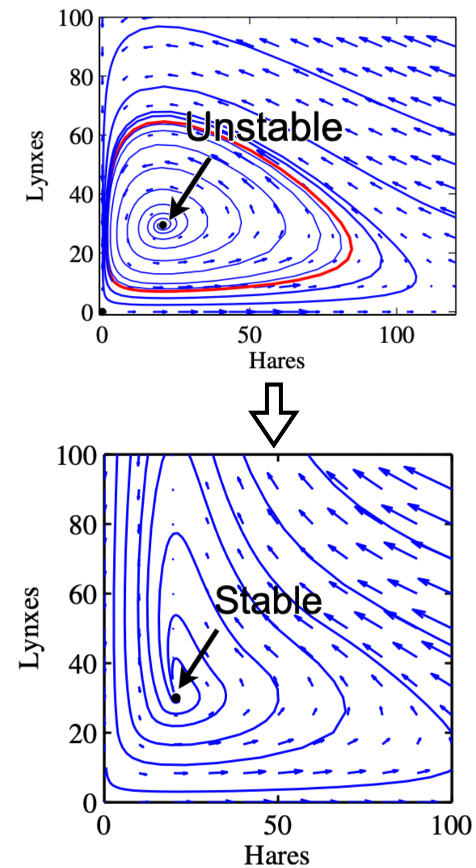


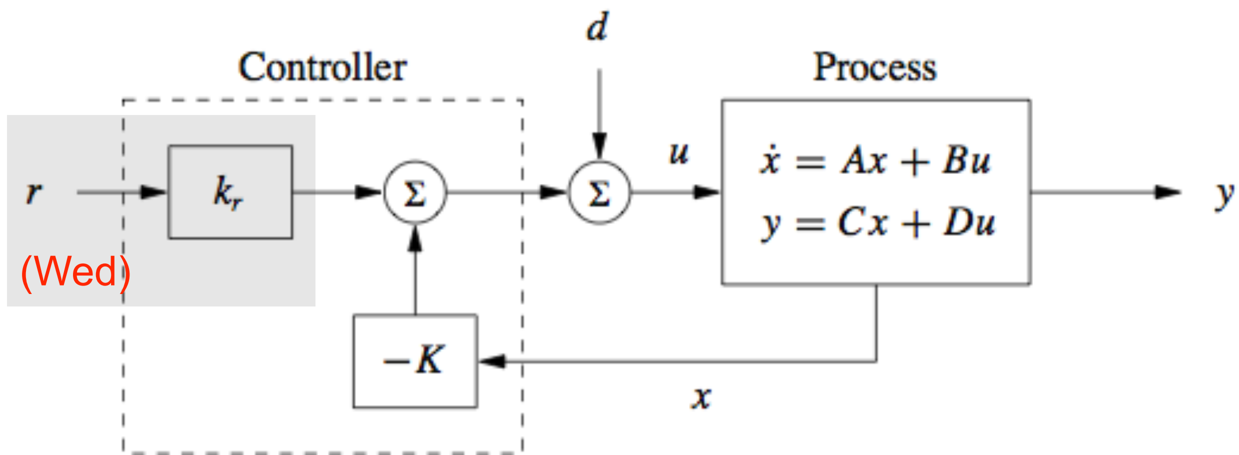
Summary: State Space Feedback

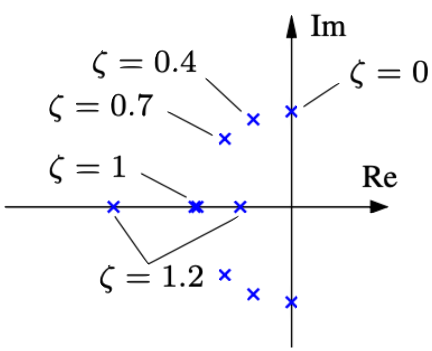
Stability: stabilize the system around an equilibrium point

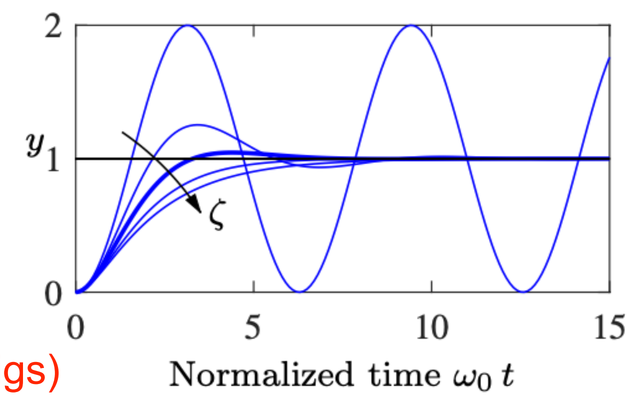
- Given equilibrium point $x_e \in \mathbb{R}^n$, find control "law" *u*=α(*x*) such that

$$\lim_{t \to \infty} x(t) = x_e \text{ for all } x(0) \in \mathbb{R}^n$$









K = ct.place(A, B, eigs)

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

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

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

K = ct.lqr(A, B, Q, R)