

Module 04

Linear System Theory

Ahmad F. Taha

**CE 4240/5240 — Introduction to
Infrastructure Systems Engineering**

Email ahmad.taha@vanderbilt.edu

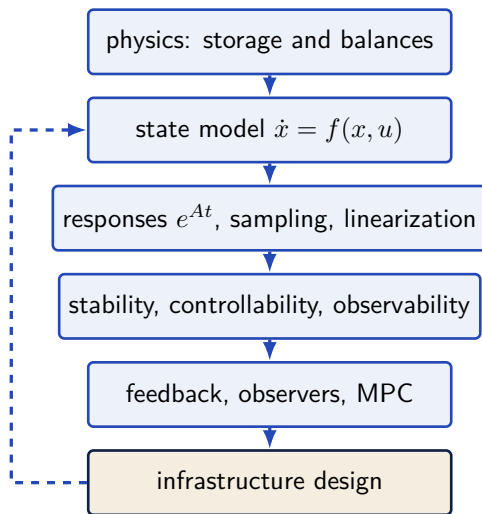
Webpage <https://ahmadstatespace.github.io/>

Fall 2026



Outline and objectives

- ① Build continuous- and discrete time state models from physical balances
- ② Compute e^{At} and solve $\dot{x} = Ax + Bu$ in the cases that matter
- ③ Linearize and discretize a model, and know where the result is valid
- ④ Decide stability from eigenvalues or from a decreasing energy function
- ⑤ Test controllability, stabilizability, observability, and detectability
- ⑥ Design state feedback, observers, observer based control, and constrained predictive control
- ⑦ Read an infrastructure decision as a change to A , B , or C



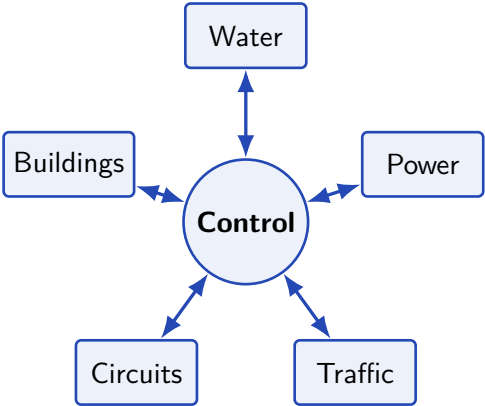
Notation

Symbol	Meaning
$x \in \mathbb{R}^n$	state vector (n states)
$u \in \mathbb{R}^m$	control input (chosen)
$d \in \mathbb{R}^r$	disturbance (not chosen)
$y \in \mathbb{R}^p$	measured output
$\dot{x}$	time derivative dx/dt
t, k	continuous time, sample index
T_s	sample time; $x_k = x(kT_s)$
A, B, E	state, input, disturbance matrices
C, D	output, feedthrough matrices
$A_d, B_d, \dots$	the discrete time counterparts

Symbol	Meaning
$\bar{x}, \bar{u}$	equilibrium state and input
$\delta x = x - \bar{x}$	deviation from equilibrium
$\lambda_i(A)$	eigenvalues (modes) of A
$j = \sqrt{-1}, \text{Re}(\cdot)$	imaginary unit, real part
I	identity matrix
$\ x\ = \sqrt{x^T x}$	Euclidean norm
$P \succ 0 (\succeq 0)$	symmetric positive (semi)definite
∇V	gradient of a scalar function V
$\text{rank}(\cdot), \text{eig}(\cdot)$	rank, set of eigenvalues
K, L	feedback gain, observer gain

Dynamic systems and control

- A dynamic system has **memory**, carried by stored water, charge, momentum, heat, or vehicles
- The **state** is the smallest internal description needed to predict future motion once inputs are known
- **Control** uses chosen inputs to change that future motion
- **Feedback** adjusts the input after seeing measurements

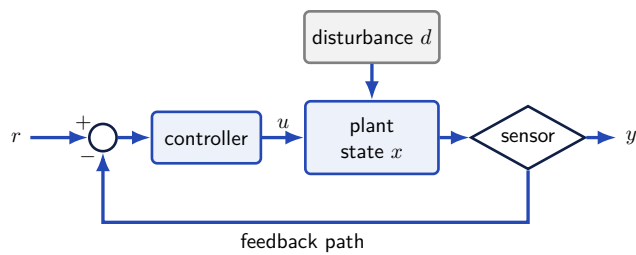


Two questions we will keep asking

Can the inputs move the state where we want it? Can the sensors tell us where the state is?

Plant, state, input, output, disturbance

- The **plant** is the physical system
- The **state** x stores its internal condition
- The **input** u is chosen; the **disturbance** d is not
- The **output** y is what sensors report
- The **reference** r is the desired behavior
- The **controller** maps r and y to u



- Open loop: u follows a preset plan and ignores y .
- Closed loop: u reacts to the error $r - y$.

Vocabulary across infrastructure systems

System	State x	Input u	Output y	Disturbance d
Traffic corridor	vehicle counts per link	ramp meter rates	flow and occupancy	demand and incidents
Water network	tank levels and node pressures	pump speeds and valve settings	level and pressure readings	customer demand
Room climate	wall and air temperature	heating power	thermostat reading	weather and open doors
RLC circuit	capacitor voltage, inductor current	source voltage	sensed voltage	load change

Open loop	Feedback	Model based control
<ul style="list-style-type: none"> Input follows a schedule Simple; no sensors Exposed to error and disturbance 	<ul style="list-style-type: none"> Input uses measurements Corrects unexpected behavior Needs sensors and a stability check 	<ul style="list-style-type: none"> Predicts before acting Exposes constraints and limits Only as good as the model

Four model classes

	Continuous time	Discrete time
Nonlinear	$\dot{x} = f(x, u, d), \quad y = g(x, u)$	$x_{k+1} = f_d(x_k, u_k, d_k),$ $y_k = g_d(x_k, u_k)$
Linear	$\dot{x} = Ax + Bu + Ed, \quad y = Cx + Du$	$x_{k+1} = A_d x_k + B_d u_k + E_d d_k,$ $y_k = C_d x_k + D_d u_k$

- **Linear** means responses add and scale (superposition)
 - **Time invariant** means the matrices, or f , do not depend on the clock
 - **Continuous** models come from physics; **discrete** models come from sampling or from step by step balances
- f and g are the state and output maps; the subscript d marks the discrete time versions
 - Linearization moves a model up one row; sampling moves it one column to the right
 - Every later test (stability, controllability, observability) has a version in each cell

LTI state space model

$$\dot{x} = Ax + Bu + Ed \quad y = Cx + Du$$

For $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $d \in \mathbb{R}^r$, $y \in \mathbb{R}^p$

$$A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m}, \quad E \in \mathbb{R}^{n \times r}$$

$$C \in \mathbb{R}^{p \times n}, \quad D \in \mathbb{R}^{p \times m}$$

- Row i of A is the balance of state i ; entry A_{ij} says how state j pushes on state i
- Column j of B is the direction in state space that actuator j pushes
- Row i of C is the combination of states that sensor i reads
- D is an instantaneous input to output path, usually zero for physical plants

Why state space

- Initial conditions are explicit
- Many inputs and outputs, one model
- The same object supports stability, control, estimation, and constraints
- Infrastructure changes are edits to A, B, C

LTI stands for *linear time invariant*: constant matrices.

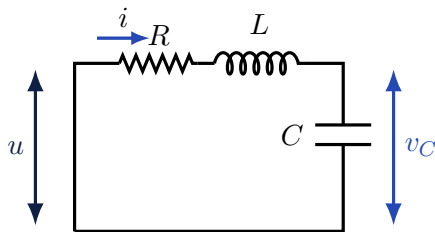
Modeling workflow

- 1 Draw a physical boundary
- 2 Identify stored quantities: each becomes a state
- 3 Name chosen inputs and uncontrolled disturbances
- 4 Write one balance law per stored quantity
- 5 Choose the measured outputs
- 6 Check units, signs, equilibria, and limiting cases

Store	Natural state	Balance law
Capacitor	voltage v_C	$C\dot{v}_C = i$
Inductor	current i	$L\dot{i} = v_L$
Tank	level h or volume	$\dot{V} = q_{\text{in}} - q_{\text{out}}$
Mass	velocity $\dot{q}$	$m\ddot{q} = \sum F$
Spring	displacement q	$F = kq$
Room	temperature θ	$c\dot{\theta} = \sum \text{heat}$
Road link	vehicle count n	$\dot{n} = \text{in} - \text{out}$

C, L, m, k, c are capacitance, inductance, mass, stiffness, and heat capacity.

Example: RLC circuit



u : source voltage (input); i : loop current; v_C : capacitor voltage; R, L, C : resistance, inductance, capacitance.

Kirchhoff's voltage law and the capacitor law

$$u = Ri + L\dot{i} + v_C, \quad i = C\dot{v}_C$$

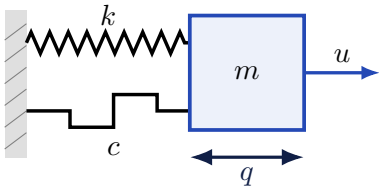
- i stores magnetic energy $\frac{1}{2}Li^2$
- v_C stores electric energy $\frac{1}{2}Cv_C^2$

Choose $x_1 = i$ **and** $x_2 = v_C$ then we can write:

$$\dot{x}_1 = -\frac{R}{L}x_1 - \frac{1}{L}x_2 + \frac{1}{L}u, \quad \dot{x}_2 = \frac{1}{C}x_1$$

$$\dot{x} = \underbrace{\begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 1/L \\ 0 \end{bmatrix}}_B u, \quad y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{C_y} x$$

Example: mass spring damper



q : displacement; $\dot{q}$: velocity; u : applied force; m, c, k : mass, damping, stiffness.

Newton's law: $m\ddot{q} = u - c\dot{q} - F_{\text{spring}}(q)$

States: $x_1 = q$, $x_2 = \dot{q}$ (two stores: spring and mass)

Linear spring $F_{\text{spring}} = kq$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u$$

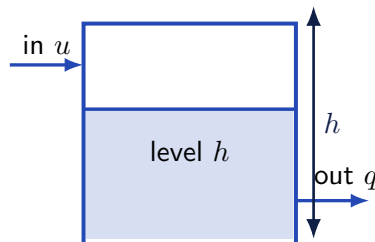
Hardening spring $F_{\text{spring}} = kq + \alpha q^3$, $\alpha > 0$

$$m\ddot{q} + c\dot{q} + kq + \alpha q^3 = u$$

$$\dot{x} = f(x, u) = \begin{bmatrix} x_2 \\ (u - cx_2 - kx_1 - \alpha x_1^3)/m \end{bmatrix}$$

The cubic term makes the restoring force grow faster than the displacement.

Example: single tank



S : tank cross section area; a : outlet orifice area; g : gravity; ρ : fractional release per sample.

Gravity drained tank (continuous, nonlinear) Volume balance and Torricelli's law (outflow grows with the square root of the level):

$$S\dot{h} = u - q, \quad q = a\sqrt{2gh}$$

$$\dot{h} = \frac{1}{S} \left(u - a\sqrt{2gh} \right)$$

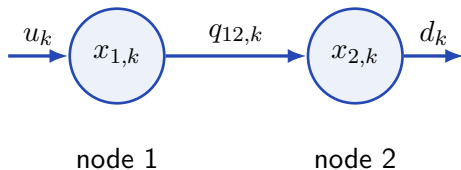
Reservoir with proportional release (discrete, linear)
Over one sample the flows are constant; in normalized storage units

$$x_{k+1} = x_k + u_k - q_k, \quad q_k = \rho x_k$$

$$x_{k+1} = (1 - \rho)x_k + u_k$$

- The stored amount is the state because it carries memory; u_k is the chosen inflow and q_k the release
- With $1 - \rho = 0.95$ and inflow scaled by 0.1 this is the model $h_{k+1} = 0.95h_k + 0.1u_k$ used later

Example: two tank network



$x_{i,k}$: storage at node i ; u_k : pumped inflow;
 d_k : demand drawn at node 2; $q_{12,k}$: flow
on the pipe; $\gamma > 0$: pipe conductance.

General network with incidence matrix
 $M \in \mathbb{R}^{n \times n_e}$ (one column per pipe, +1 at
the tail and -1 at the head) and
 $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_{n_e})$:

$$x_{k+1} = (I - M\Gamma M^T)x_k + B_d u_k + E_d d_k$$

$M^T x$ lists the level differences across pipes.

Edge flow follows the level difference

$$q_{12,k} = \gamma (x_{1,k} - x_{2,k})$$

Each node keeps its own balance

$$x_{1,k+1} = x_{1,k} + u_k - q_{12,k}$$

$$x_{2,k+1} = x_{2,k} + q_{12,k} - d_k$$

Stack $x_k = [x_{1,k} \ x_{2,k}]^T$:

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 - \gamma & \gamma \\ \gamma & 1 - \gamma \end{bmatrix}}_{A_d} x_k + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B_d} u_k + \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{E_d} d_k$$

- Each row of A_d is one node balance
- Off diagonal entries are pipe interactions
- Removing the pipe sets $\gamma = 0$ and decouples the tanks

Linearity and time invariance

Linearity is superposition. For $\dot{x} = -2x + u$ from $x(0) = 0$

$$x[u](t) = \int_0^t e^{-2(t-\tau)} u(\tau) d\tau$$

so $x[au_1 + bu_2] = ax[u_1] + bx[u_2]$ for any scalars a, b . A unit step gives $x(t) = (1 - e^{-2t})/2$.

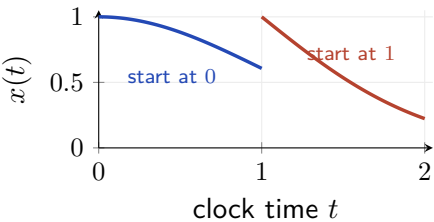
The map $f(x, u) = x^2 + u$ fails the test:

$$f(2x, 2u) = 4x^2 + 2u \neq 2f(x, u)$$

Time invariance: the clock does not matter.

For $\dot{x} = -2x$ starting at $x = 1$, one second later the state is e^{-2} whether the experiment starts at clock time 0 or 5.

For $\dot{x} = -tx$ from $x(0) = 1$, $x(1) = e^{-1/2}$. Start the same one second experiment at clock time 1: $x(2)/x(1) = e^{-3/2}$. The clock changed the result.



State transition formula

Solve the continuous LTI model

$$\dot{x} = Ax + Bu, \quad x(0) = x_0$$

Multiply by e^{-At} and integrate

$$\frac{d}{dt} (e^{-At}x) = e^{-At}Bu$$

$$x(t) = \underbrace{e^{At}x_0}_{\text{natural}} + \underbrace{\int_0^t e^{A(t-\tau)}Bu(\tau) d\tau}_{\text{forced}}$$

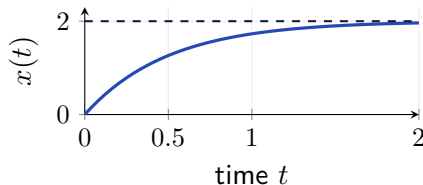
$$y(t) = Cx(t) + Du(t)$$

The natural response depends on where we start; the forced response depends on what we do.

The matrix exponential carries the state motion

$$e^{At} = I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \dots$$

- $e^{A \cdot 0} = I$ and $\frac{d}{dt}e^{At} = Ae^{At}$
- $Av = \lambda v$ gives $e^{At}v = e^{\lambda t}v$: modes decay or grow as scalar exponentials
- $e^{A(t+s)} = e^{At}e^{As}$ and $(e^{At})^{-1} = e^{-At}$



Unit step response of $\dot{x} = -2x + 4u$, $x(0) = 0$:
 $x(t) = 2(1 - e^{-2t})$ approaches the dashed value 2 with
 time constant $1/2$ s.

Computing e^{At} by diagonalization

If A has n linearly independent eigenvectors v_i with $Av_i = \lambda_i v_i$

$$A = T\Lambda T^{-1}, \quad T = [v_1 \quad \cdots \quad v_n]$$

$$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$$

then $A^k = T\Lambda^k T^{-1}$ term by term in the series, so

$$e^{At} = T \text{diag}(e^{\lambda_1 t}, \dots, e^{\lambda_n t}) T^{-1}$$

- Columns of T are the mode shapes
- Each mode evolves on its own as $e^{\lambda_i t}$
- Complex $\lambda_i = \sigma \pm j\omega$ give $e^{\sigma t} \cos \omega t$ and $e^{\sigma t} \sin \omega t$

Example.

$$A = \begin{bmatrix} -1 & 2 \\ 0 & -3 \end{bmatrix}, \quad \lambda_1 = -1, \quad \lambda_2 = -3$$

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$e^{At} = T \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-3t} \end{bmatrix} T^{-1} = \begin{bmatrix} e^{-t} & e^{-t} - e^{-3t} \\ 0 & e^{-3t} \end{bmatrix}$$

- Check: $t = 0$ gives I ; the derivative at 0 gives A
- The off diagonal entry is the difference of the two modal decays: state 2 feeds state 1 through $A_{12} = 2$

Eigenvalues set the rates of system evolution and eigenvectors set the directions.

Repeated eigenvalues

Suppose $A = \lambda I + N$ with N nilpotent, $N^r = 0$.
Since λI and N commute

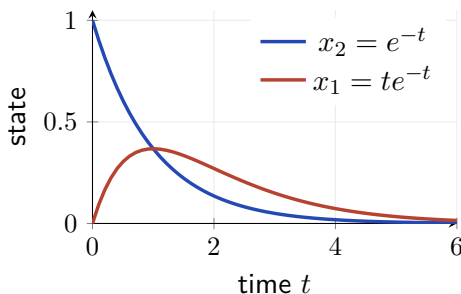
$$e^{At} = e^{\lambda t} e^{Nt}, \quad e^{Nt} = I + Nt + \cdots + \frac{N^{r-1}t^{r-1}}{(r-1)!}$$

Example.

$$A = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} = -I + \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_N, \quad N^2 = 0$$

$$e^{At} = e^{-t} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

From $x_0 = [0 \ 1]^T$: $x(t) = e^{-t} [t \ 1]^T$, so x_1 first rises, then both decay.



- $te^{\lambda t}$ still decays when $\text{Re } \lambda < 0$
- At $\lambda = 0$ the factor t grows without

$$\text{bound: } e^{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

Computing e^{At} numerically

Cayley Hamilton shortcut. Every 2×2 matrix satisfies its characteristic equation, so every power of A is a combination of I and A :

$$e^{At} = \alpha(t)I + \beta(t)A$$

For distinct eigenvalues $\lambda_1 \neq \lambda_2$

$$\beta(t) = \frac{e^{\lambda_1 t} - e^{\lambda_2 t}}{\lambda_1 - \lambda_2}, \quad \alpha(t) = \frac{\lambda_1 e^{\lambda_2 t} - \lambda_2 e^{\lambda_1 t}}{\lambda_1 - \lambda_2}$$

Obtained by requiring the formula to hold on each eigenvector: $e^{\lambda_i t} = \alpha + \beta \lambda_i$.

Scaling and squaring. Direct summation of the series is inaccurate when $\|At\|$ is large. Libraries use

$$e^{At} = \left(e^{At/2^s} \right)^{2^s}$$

- ① Choose the integer s so that $At/2^s$ has small norm
- ② Approximate the small exponential by a Padé rational function
- ③ Square the result s times
 - This is the method behind `expm` in MATLAB, NumPy, and Julia
 - Always sanity check: $e^{A \cdot 0} = I$ and $\frac{d}{dt} e^{At} \big|_0 = A$

Constant inputs and equilibria

For constant $u(t) = \bar{u}$ and invertible A , the forced integral can be evaluated:

$$x(t) = e^{At}x_0 + A^{-1}(e^{At} - I)B\bar{u}$$

The **equilibrium** $\bar{x}$ solves $0 = A\bar{x} + B\bar{u}$, so $\bar{x} = -A^{-1}B\bar{u}$ and

$$x(t) = \bar{x} + e^{At}(x_0 - \bar{x})$$

Define the **deviation variables** $\delta x = x - \bar{x}$, $\delta u = u - \bar{u}$. They obey the same model, $\delta \dot{x} = A\delta x + B\delta u$, with the equilibrium at the origin.

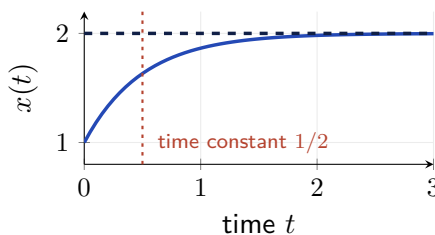
- Stability questions are asked about $\delta x = 0$
- Constraints must be shifted by $\bar{x}$ and $\bar{u}$ too

Worked scalar case. $\dot{x} = -2x + 4u$,
 $x(0) = 1$, $u \equiv 1$

$$\bar{x} = -(-2)^{-1}(4) = 2$$

$$x(t) = 2 + e^{-2t}(1 - 2) = \boxed{2 - e^{-2t}}$$

Checks: $x(0) = 1$, $\dot{x}(0) = -2(1) + 4 = 2$,
 $x(\infty) = 2$.



Time constant $1/|\lambda| = 0.5$ s; the error to equilibrium falls to 2% after about four time constants.

RLC circuit: closed form response

With $R = 2\ \Omega$, $L = 1\ \text{H}$, $C = 0.5\ \text{F}$

$$A = \begin{bmatrix} -2 & -1 \\ 2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C_y = [0 \quad 1]$$

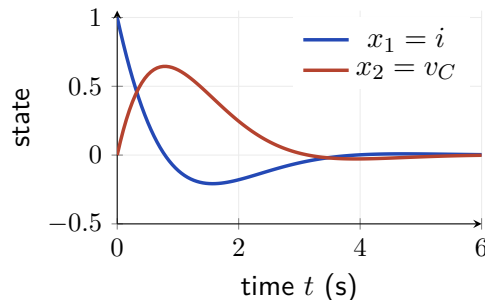
Eigenvalues: $\lambda^2 + 2\lambda + 2 = 0 \Rightarrow \lambda = -1 \pm j$

Write $A = -I + N$ with $N = \begin{bmatrix} -1 & -1 \\ 2 & 1 \end{bmatrix}$ and

note $N^2 = -I$, so $e^{Nt} = I \cos t + N \sin t$:

$$e^{At} = e^{-t} \begin{bmatrix} \cos t - \sin t & -\sin t \\ 2 \sin t & \cos t + \sin t \end{bmatrix}$$

- e^{-t} : the resistor removes stored energy at rate set by R/L
- $\sin t, \cos t$: energy sloshes between L and C at $\omega = 1\ \text{rad/s}$
- $R = 0$ removes the decay and leaves a sustained oscillation



Zero input response from $x_0 = [1 \ 0]^T$: an initial current of 1 A charges the capacitor, then everything decays.

Reading the matrices: $B_1 = 1/L$ says the source changes current directly; $A_{21} = 1/C$ couples current into capacitor voltage.

Discrete time response

For $x_{k+1} = A_d x_k + B_d u_k$, repeated substitution gives

$$x_1 = A_d x_0 + B_d u_0$$

$$x_2 = A_d^2 x_0 + A_d B_d u_0 + B_d u_1$$

$$x_k = A_d^k x_0 + \sum_{i=0}^{k-1} A_d^{k-1-i} B_d u_i$$

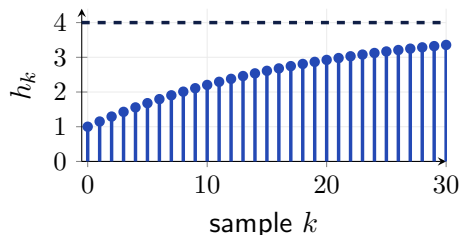
For constant $u_k = \bar{u}$ the equilibrium solves $\bar{x} = A_d \bar{x} + B_d \bar{u}$ and

$$x_k = \bar{x} + A_d^k (x_0 - \bar{x})$$

Sampled reservoir. $h_{k+1} = 0.95h_k + 0.1u_k$,
 $h_0 = 1$, $u_k \equiv 2$

$$\bar{h} = 0.95\bar{h} + 0.2 \Rightarrow \bar{h} = 4$$

$$h_k = \bar{h} + 0.95^k (h_0 - \bar{h}) = 4 - 3(0.95)^k$$



Each sample keeps 95% of the distance to equilibrium: $h_{10} \approx 2.20$, $h_{30} \approx 3.36$.

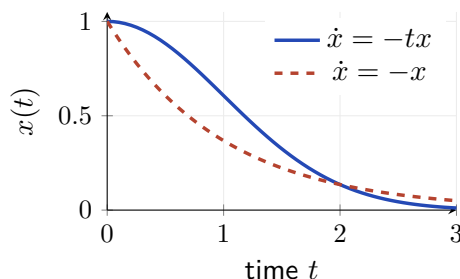
Time varying systems

Consider the scalar LTV model $\dot{x} = -tx + u(t)$.
Multiply by the integrating factor $e^{t^2/2}$:

$$\frac{d}{dt} \left(e^{t^2/2} x(t) \right) = e^{t^2/2} u(t)$$

$$x(t) = e^{-t^2/2} x_0 + \int_0^t e^{-(t^2-\tau^2)/2} u(\tau) d\tau$$

- The weight depends on t and τ separately, not only on the elapsed time $t - \tau$
- For matrices $A(t)$, values at different times need not commute, so $e^{\int A}$ is generally wrong
- The general object is the state transition matrix $\Phi(t, \tau)$, usually computed numerically



Both start at 1; the LTV model decays slowly at first, then faster than any fixed exponential as the coefficient $-t$ grows.

Sampled models and Euler discretization

- Sensors report at sample times $t = kT_s$
- A digital controller computes u_k and holds it until the next sample (zero order hold, ZOH)
- Simulation and predictive control advance in steps

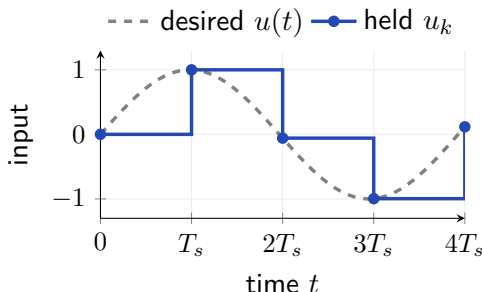
Forward Euler (approximate). Replace the derivative of $\dot{x} = Ax + Bu$ by a difference:

$$\dot{x}(kT_s) \approx \frac{x_{k+1} - x_k}{T_s}$$

$$x_{k+1} = (I + T_s A)x_k + T_s B u_k$$

- Cheap and intuitive
- Accuracy degrades as T_s grows
- A large T_s can make a stable model look unstable

The next slide gives the exact sampled model for a held input; Euler is its first order Taylor approximation.



Exact discretization: zero order hold

If $u(t) = u_k$ on $kT_s \leq t < (k+1)T_s$, apply the state transition formula over one sample:

$$x_{k+1} = e^{AT_s} x_k + \left(\int_0^{T_s} e^{A\tau} d\tau \right) B u_k$$

$$\boxed{A_d = e^{AT_s}}, \quad \boxed{B_d = \int_0^{T_s} e^{A\tau} B d\tau}$$

The output equation is unchanged: $C_d = C$, $D_d = D$.
If A is invertible, $B_d = A^{-1}(A_d - I)B$.

Expand to see Euler inside: $A_d = I + T_s A + \frac{1}{2} T_s^2 A^2 + \dots$ and $B_d = T_s B + \frac{1}{2} T_s^2 AB + \dots$.
Euler keeps only the first order terms.

One matrix exponential computes both

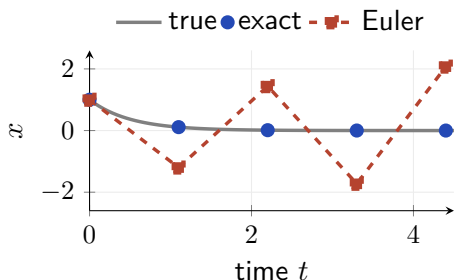
$$\exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T_s\right) = \begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}$$

- Works for singular A (integrators, pure storage)
- Exact at the sample instants
- The eigenvalues map as $\lambda_i(A_d) = e^{\lambda_i(A)T_s}$

Sample time and discretization error

For $\dot{x} = -2x + 4u$: exact $A_d = e^{-2T_s}$,
 $B_d = 2(1 - e^{-2T_s})$; Euler $A_d = 1 - 2T_s$,
 $B_d = 4T_s$.

T_s	Euler A_d	Exact A_d	Exact B_d
0.1	0.8	0.8187	0.3625
0.5	0	0.3679	1.2642
1.1	-1.2	0.1108	1.7784



Zero input response with $T_s = 1.1$: Euler's $A_d = -1.2$ alternates in sign and grows.

Mass spring damper with $m = 2$, $c = 3$,
 $k = 8$ (so $\dot{x} = Ax + Bu$ with)

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -1.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

Exact ZOH model at $T_s = 0.1$ s:

$$A_d = \begin{bmatrix} 0.9810 & 0.0922 \\ -0.3690 & 0.8427 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.00237 \\ 0.04612 \end{bmatrix}$$

- B_d is small in position and larger in velocity: a force changes velocity first, position second
- $A_d \approx I + T_s A = \begin{bmatrix} 1 & 0.1 \\ -0.4 & 0.85 \end{bmatrix}$; the difference is the second order correction
- Rule of thumb: T_s at most a tenth of the fastest time constant of interest

Linearization

For $\dot{x} = f(x, u)$, $y = g(x, u)$ choose an operating point $(\bar{x}, \bar{u})$ with $f(\bar{x}, \bar{u}) = 0$ and write $x = \bar{x} + \delta x$, $u = \bar{u} + \delta u$. A first order Taylor expansion gives

$$\delta \dot{x} \approx A \delta x + B \delta u, \quad \delta y \approx C \delta x + D \delta u$$

$$A = \left. \frac{\partial f}{\partial x} \right|_{\bar{x}, \bar{u}}$$

$$B = \left. \frac{\partial f}{\partial u} \right|_{\bar{x}, \bar{u}}$$

$$C = \left. \frac{\partial g}{\partial x} \right|_{\bar{x}, \bar{u}}$$

$$D = \left. \frac{\partial g}{\partial u} \right|_{\bar{x}, \bar{u}}$$

$\partial f / \partial x$ is the $n \times n$ Jacobian with entries $\partial f_i / \partial x_j$.

Discrete models linearize the same way.

For $x_{k+1} = f_d(x_k, u_k)$ with fixed point $\bar{x} = f_d(\bar{x}, \bar{u})$

$$A_d = \left. \frac{\partial f_d}{\partial x} \right|_{\bar{x}, \bar{u}}, \quad B_d = \left. \frac{\partial f_d}{\partial u} \right|_{\bar{x}, \bar{u}}$$

- The model is in deviation variables:
 $\delta x = 0$ is the equilibrium
- Different operating points give different A and B
- The neglected terms are second order in $\|\delta x\|, \|\delta u\|$

Order of operations

Continuous model available: $f \rightarrow (A, B) \rightarrow (A_d, B_d)$, linearize at the intended equilibrium, then sample at the actual T_s . If DT model is available then differentiate f_d directly.

Linearization examples

Gravity drained tank

$$\dot{h} = f(h, u) = \frac{1}{S} \left(u - a\sqrt{2gh} \right)$$

Equilibrium: $\bar{u} = a\sqrt{2g\bar{h}}$ (inflow equals outflow)

$$A = \frac{\partial f}{\partial h} = -\frac{a\sqrt{2g}}{2S\sqrt{\bar{h}}}, \quad B = \frac{\partial f}{\partial u} = \frac{1}{S}$$

Measured outflow $y = a\sqrt{2g\bar{h}}$:

$$C = \frac{\partial y}{\partial h} = \frac{a\sqrt{2g}}{2\sqrt{\bar{h}}}, \quad D = 0$$

With $S = 10$, $a = 0.1$, $g = 9.81$, $\bar{h} = 1$:

$$A \approx -0.0221, \quad B = 0.1, \quad C \approx 0.221$$

Time constant $1/|A| \approx 45$ s; a higher $\bar{h}$ gives a slower tank.

A linearization is a family of models indexed by the operating point.

Hardening spring $m\ddot{q} + c\dot{q} + kq + \alpha q^3 = u$,
 $x = [q \ \dot{q}]^T$

$$f(x, u) = \begin{bmatrix} x_2 \\ (u - cx_2 - kx_1 - \alpha x_1^3)/m \end{bmatrix}$$

Equilibrium at rest: $\bar{x} = [\bar{q} \ 0]^T$, $\bar{u} = k\bar{q} + \alpha\bar{q}^3$

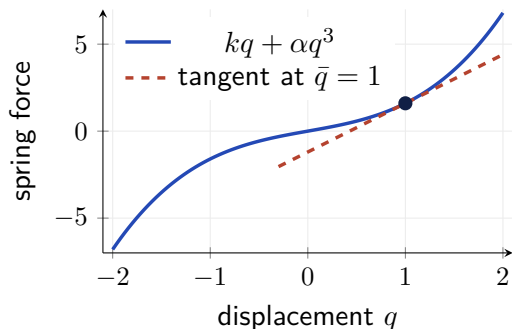
$$A = \begin{bmatrix} 0 & 1 \\ -(k + 3\alpha\bar{q}^2)/m & -c/m \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}$$

The **local stiffness** is

$$k_{\text{local}} = k + 3\alpha\bar{q}^2$$

so the same spring is stiffer, and the local natural frequency $\sqrt{k_{\text{local}}/m}$ higher, at larger displacements.

Validity of local models



$k = 1, \alpha = 0.6$: the tangent slope $k_{\text{local}} = 2.8$ matches the curve near $\bar{q} = 1$ and is off by more than half the force at $q = 2$.

- The omitted terms are second order: error grows like $\|\delta x\|^2$
- Saturation, switching, and dead bands are not differentiable; treat them explicitly
- A sampled linear model has two approximations: linearization and the hold

Validation checklist

- Simulate the nonlinear and the local model from the same start
- Test the largest expected disturbance and reference change
- Confirm that constraints keep δx in the trusted region

Stability definitions

Continuous time

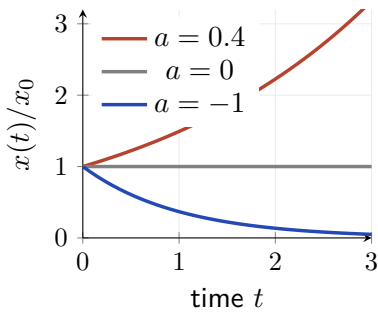
$$\dot{x} = ax, \quad x(t) = e^{at}x_0$$

$a < 0$	decays to zero
$a = 0$	stays constant
$a > 0$	grows

Discrete time

$$x_{k+1} = a_d x_k, \quad x_k = a_d^k x_0$$

$ a_d < 1$	decays to zero
$ a_d = 1$	persists
$ a_d > 1$	grows



Definitions for vectors (about the equilibrium $x = 0$)

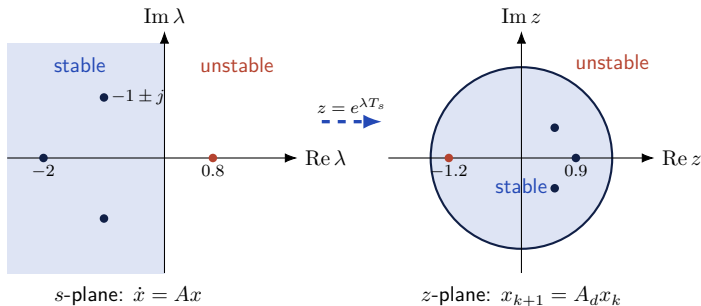
- **Stable**: trajectories starting close stay close
- **Asymptotically stable**: stable and $x(t) \rightarrow 0$
- **Exponentially stable**: there exist $M \geq 1, \alpha > 0$ with

$$\|x(t)\| \leq M e^{-\alpha t} \|x(0)\|$$

- **Unstable**: some trajectory leaves every neighborhood

For LTI systems, asymptotic and exponential stability coincide.

Eigenvalue stability regions



continuous: $\text{Re } \lambda_i(A) < 0 \quad \forall i$

discrete: $|\lambda_i(A_d)| < 1 \quad \forall i$

- Real negative λ : monotone decay with time constant $1/|\lambda|$
- Pair $\sigma \pm j\omega$: oscillation at ω inside the envelope $e^{\sigma t}$
- Sampling maps λ to $e^{\lambda T_s}$: the imaginary axis becomes the unit circle
- Repeated eigenvalues on the boundary can grow like t , as

$$\text{in } e^{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} t} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

Energy based stability

Eigenvalues decide stability for LTI models but say little about

- nonlinear dynamics and saturation
- uncertain parameters
- regions of attraction under constraints

An **energy like function** $V(x)$ can replace them if it is

- zero at the equilibrium and positive elsewhere
- decreasing along every trajectory

$$V(x) > 0 \text{ for } x \neq 0, \quad \dot{V}(x) < 0$$

Mass spring damper, no input:

$$m\ddot{q} + c\dot{q} + kq = 0$$

Total stored energy

$$V(q, \dot{q}) = \underbrace{\frac{1}{2}kq^2}_{\text{spring}} + \underbrace{\frac{1}{2}m\dot{q}^2}_{\text{kinetic}}$$

Differentiate along the motion, then use the force balance $m\ddot{q} = -c\dot{q} - kq$:

$$\dot{V} = kq\dot{q} + m\dot{q}\ddot{q} = \dot{q}(kq + m\ddot{q}) = \boxed{-c\dot{q}^2 \leq 0}$$

- The spring and mass store energy; the damper removes it at rate $c\dot{q}^2$
- $\dot{V} = 0$ only when $\dot{q} = 0$, and the motion cannot stay there unless $q = 0$, so the energy keeps falling to zero

Physics hands us a Lyapunov function: the total stored energy in a dynamic system.

Lyapunov's direct method

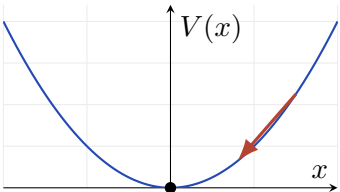
Theorem (Lyapunov)

For $\dot{x} = f(x)$ with $f(0) = 0$, suppose a continuously differentiable V satisfies, for all $x \neq 0$ in a region around the origin,

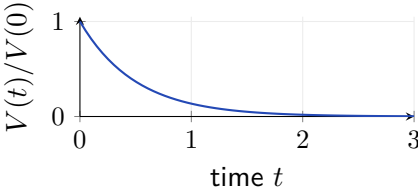
$$V(0) = 0, \quad V(x) > 0, \quad \dot{V}(x) = \nabla V(x)^T f(x) < 0$$

Then the origin is asymptotically stable in that region.

- Conditions holding only near 0: local conclusion
- Conditions holding everywhere and $V(x) \rightarrow \infty$ as $\|x\| \rightarrow \infty$: global conclusion
- $\dot{V} \leq 0$ alone gives stability without convergence; convergence needs an extra argument
- Discrete time: replace $\dot{V}$ by $V(x_{k+1}) - V(x_k)$ (next slide)



For $\dot{x} = -x$ and $V = x^2/2$:
 $\dot{V} = x\dot{x} = -x^2 = -2V$, so
 $V(t) = V(0)e^{-2t}$.



The state rolls downhill in V , and it's great that we never need the trajectory itself!1

Lyapunov equations

Continuous LTI $\dot{x} = Ax$, try $V(x) = x^T P x$ with $P = P^T \succ 0$:

$$\dot{V} = \dot{x}^T P x + x^T P \dot{x} = x^T (A^T P + P A) x$$

Pick any $Q = Q^T \succ 0$ and solve the linear matrix equation

$$A^T P + P A = -Q$$

Then $\dot{V} = -x^T Q x < 0$.

- A solution $P \succ 0$ exists for every $Q \succ 0$ if and only if A is stable
- Level sets $x^T P x = c$ are nested ellipsoids that trajectories cross inward

Discrete LTI $x_{k+1} = A_d x_k$, $V_k = x_k^T P x_k$:

$$V_{k+1} - V_k = x_k^T (A_d^T P A_d - P) x_k$$

$$\boxed{A_d^T P A_d - P = -Q}, \quad Q \succ 0$$

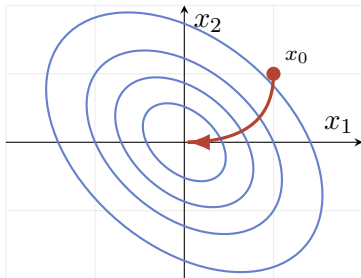
- $P \succ 0$ exists for every $Q \succ 0$ if and only if all $|\lambda_i(A_d)| < 1$
- Measures energy loss per sample instead of per second
- Both equations are linear in P : $n(n+1)/2$ unknowns, solved by `lyap` and `dlyap`

Lyapunov examples

$A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$, $Q = I$, $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$. The three entries of $A^T P + P A = -I$ give

$$-2p_{11} = -1, \quad p_{11} - 3p_{12} = 0, \quad 2p_{12} - 4p_{22} = -1$$

$$P = \begin{bmatrix} 1/2 & 1/6 \\ 1/6 & 1/3 \end{bmatrix} \succ 0, \quad \dot{V} = -x^T x$$



Level sets $x^T P x = c$ and the trajectory from $x_0 = [2 \ 2]^T$, which crosses every ellipse inward.

Beyond linearization. For

$$\dot{x} = -x^3$$

the linearization at 0 is $\delta \dot{x} = 0 \cdot \delta x$: eigenvalue 0, no verdict.

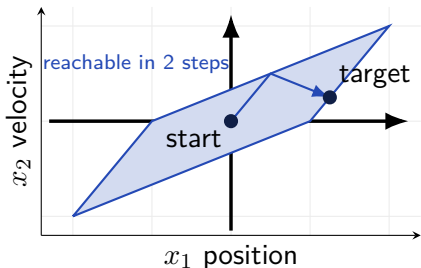
Choose $V = x^2/2$:

$$\dot{V} = x \dot{x} = -x^4 < 0 \quad \text{for } x \neq 0$$

and $V \rightarrow \infty$ as $|x| \rightarrow \infty$, so the origin is **globally asymptotically stable**.

- Convergence is slower than any exponential near zero:
 $x(t) = x_0 / \sqrt{1 + 2x_0^2 t}$
- Linearization is inconclusive exactly when a mode sits on the boundary
- The energy argument still works and does not need the solution

Reachability



Sampled double integrator (position and velocity, unit sample time)

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A_d} x_k + \underbrace{\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}}_{B_d} u_k, \quad |u_k| \leq 1$$

After two steps from the origin

$$x_2 = A_d B_d u_0 + B_d u_1 = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} u_0 + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_1$$

- The two columns span the plane: any target is reachable if u is unbounded
- With $|u_k| \leq 1$ the reachable set is the shaded parallelogram; the drawn path uses $u_0 = 1, u_1 = -0.5$

– **Reachability**: which states can inputs create?

– **Controllability**: which states can inputs drive to zero? For LTI systems one rank test answers both.

Controllability rank test

For $x_{k+1} = Ax_k + Bu_k$ after n steps

$$x_n - A^n x_0 = \begin{bmatrix} A^{n-1}B & \cdots & AB & B \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix}$$

The columns are the directions inputs can create.
Define the **controllability matrix**

$$\mathcal{C} = [B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B] \in \mathbb{R}^{n \times nm}$$

Rank test

(A, B) is controllable if and only if $\text{rank}(\mathcal{C}) = n$

- B : direct actuator directions; $AB, A^2B, \dots$: how dynamics spread them
- Powers beyond A^{n-1} add nothing (Cayley Hamilton)
- The same test applies to $\dot{x} = Ax + Bu$

RLC circuit

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

$$\mathcal{C} = [B \quad AB] = \begin{bmatrix} 1 & -2 \\ 0 & 2 \end{bmatrix}, \quad \det \mathcal{C} = 2 \neq 0$$

Controllable: the source changes the current directly, and current changes the capacitor voltage through $A_{21} = 2$.

Two decoupled tanks ($\gamma = 0$, pump on tank 1 only)

$$A_d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

Rank 1: tank 2 is unreachable. Reconnect the pipe ($\gamma > 0$) and $AB_d = [1 - \gamma \quad \gamma]^T$ restores full rank.

Rank answers yes or no; *how hard* is a separate question.

Equivalent controllability tests

Test	Condition	Interpretation
Rank	$\text{rank}[B \ AB \ \cdots \ A^{n-1}B] = n$	input directions span the state space
PBH	$\text{rank}[\lambda I - A \ B] = n$ for every eigenvalue λ of A	no mode is disconnected from the input
Eigenvector	$w^T B \neq 0$ for every left eigenvector w ($w^T A = \lambda w^T$)	every mode feels some actuator
Gramian	$W_c(T) \succ 0$ for some $T > 0$ (defined below)	every direction is reachable with finite energy

PBH (*Popov, Belevitch, Hautus*) is the most useful in practice:

- It is checked mode by mode
- Failing modes are identified, not just counted
- Restricting it to unstable modes gives stabilizability

Gramian over the horizon $[0, T]$:

$$W_c(T) = \int_0^T e^{A\tau} B B^T e^{A^T \tau} d\tau \in \mathbb{R}^{n \times n}$$

Discrete version:

$$W_c(N) = \sum_{i=0}^{N-1} A_d^i B_d B_d^T (A_d^T)^i.$$

Small eigenvalues of W_c flag directions that are reachable only with large inputs.

Controllability Gramian

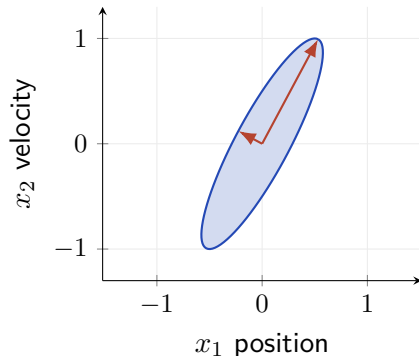
Goal: steer $\dot{x} = Ax + Bu$ from x_0 to x_f at time T .
 Let $\Delta x = x_f - e^{AT}x_0$ (what the input must add to the natural motion). If $W_c(T)$ is invertible, the input with least energy $\int_0^T u^T u dt$ is

$$u^*(t) = B^T e^{A^T(T-t)} W_c(T)^{-1} \Delta x$$

and the energy it uses is

$$\int_0^T u^{*T} u^* dt = \Delta x^T W_c(T)^{-1} \Delta x$$

- Directions along large eigenvalues of W_c are cheap
- Directions along small eigenvalues are reachable in principle but expensive
- A longer horizon T enlarges W_c and lowers the cost



Continuous double integrator, $T = 1$:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad W_c(1) = \begin{bmatrix} 1/3 & 1/2 \\ 1/2 & 1 \end{bmatrix}$$

The ellipse $\Delta x^T W_c^{-1} \Delta x \leq 1$ is the set reachable with unit energy. Semi axes $\sqrt{1.268} = 1.13$ and $\sqrt{0.066} = 0.26$: changing velocity is cheap; changing position without velocity is not.

State feedback and pole placement

Choose the input as a linear function of the state plus a reference r :

$$u = -Kx + r \Rightarrow \dot{x} = (A - BK)x + Br$$

Pole placement

If (A, B) is controllable, the eigenvalues of $A - BK$ can be placed anywhere (complex ones in conjugate pairs) by choice of $K \in \mathbb{R}^{m \times n}$.

Worked design. $A = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$,

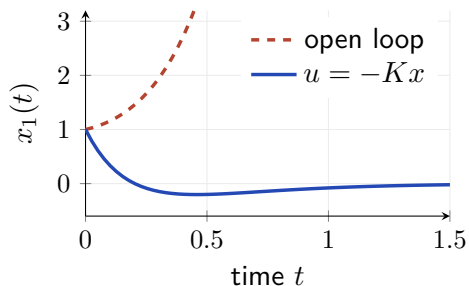
$K = [k_1 \ k_2]$. Open loop eigenvalues 4 and -2 : unstable.

$$\det(\lambda I - A + BK) = \lambda^2 + (k_1 - 2)\lambda + (3k_2 - k_1 - 8)$$

Target poles $-3, -5$:

$$(\lambda + 3)(\lambda + 5) = \lambda^2 + 8\lambda + 15. \text{ Match coefficients:}$$

$$k_1 - 2 = 8, \quad 3k_2 - k_1 - 8 = 15 \Rightarrow K = \begin{bmatrix} 10 & 11 \end{bmatrix}$$



Response from $x_0 = [1 \ 0]^T$. Closed loop: $x_1 = 3e^{-5t} - 2e^{-3t}$, and the input starts at $u(0) = -10$.

- Faster poles mean larger K and larger u
- Check actuator limits after choosing K
- For $n > 2$ use `place(A,B,poles)` or, better, an LQR weighting

Discrete feedback and pole selection

For $x_{k+1} = A_d x_k + B_d u_k$ use
 $u_k = -K x_k + r_k$:

$$x_{k+1} = (A_d - B_d K) x_k + B_d r_k$$

- Closed loop eigenvalues must lie inside the unit circle
- A continuous pole λ corresponds to the discrete pole $e^{\lambda T_s}$
- A pole at $z = 0$ is a *deadbeat* mode: it vanishes after one sample
- The same place routine works with (A_d, B_d)

λ	$T_s = 0.1$	$T_s = 0.5$
-3	0.741	0.223
-5	0.607	0.082
$-1 \pm j$	$0.905e^{\pm j0.1}$	$0.607e^{\pm j0.5}$

Choosing the poles

- Decay rate: the slowest pole sets the settling time (about $4/|\operatorname{Re} \lambda|$)
- Damping: keep $|\operatorname{Im} \lambda|/|\operatorname{Re} \lambda|$ modest to avoid overshoot
- Effort: gains grow with the distance between open- and closed loop poles
- Robustness: very fast poles amplify model error and measurement noise
- Sampling: discrete poles faster than the sample rate cannot be realized

Reference tracking

$u = -Kx + r$ does not make y follow r by itself. Scale r by $N_r = -[C(A - BK)^{-1}B]^{-1}$ (square case), or add integral action.

Stabilizability

Definition

(A, B) is **stabilizable** if some K makes $A - BK$ stable.

- Controllable $\Rightarrow$ stabilizable
- A stable A is stabilizable with $K = 0$, even if $B = 0$
- An uncontrollable pair can still be stabilizable
- Failure needs an *unstable and uncontrollable* mode

PBH test on unstable modes only

$$\text{rank}[\lambda I - A \quad B] = n$$

for every eigenvalue with $\text{Re } \lambda \geq 0$ (continuous) or $|\lambda| \geq 1$ (discrete).

Why: split the state into controllable and uncontrollable parts. A change of coordinates $z = T^{-1}x$ gives

$$\begin{bmatrix} \dot{z}_c \\ \dot{z}_u \end{bmatrix} = \begin{bmatrix} A_c & A_{cu} \\ 0 & A_u \end{bmatrix} \begin{bmatrix} z_c \\ z_u \end{bmatrix} + \begin{bmatrix} B_c \\ 0 \end{bmatrix} u$$

- (A_c, B_c) is controllable, so feedback can place $\text{eig}(A_c - B_c K_c)$
- No input reaches z_u ; its eigenvalues $\text{eig}(A_u)$ are fixed
- Stabilizable $\iff A_u$ is stable

Detectability, next section, is the mirror image: unobservable modes must be stable. Together they are the minimum requirements for observer based control and for a well posed MPC.

Stabilizability examples

Uncontrollable but stabilizable

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

$$C = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \text{rank } C = 1$$

The input reaches only x_2 . The uncontrollable mode is $\lambda = -2$, already stable. With $K = [0 \quad 3]$

$$A - BK = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix}$$

Both poles at -2 : stabilizable, although we could not have moved the first one.

Cannot be stabilized

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

The first state obeys $\dot{x}_1 = x_1$ with no input in its equation. For any $K = [k_1 \ k_2]$

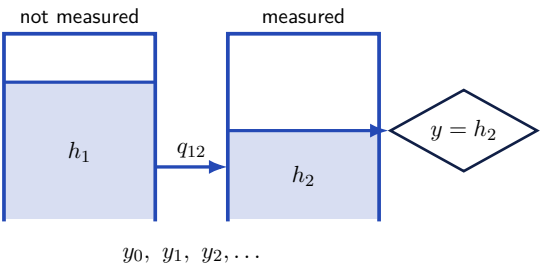
$$A - BK = \begin{bmatrix} 1 & 0 \\ -k_1 & -2 - k_2 \end{bmatrix}$$

is lower triangular, so $\lambda = 1$ stays an eigenvalue.

PBH at $\lambda = 1$: $\text{rank} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 1 \end{bmatrix} = 1 < 2$. Fails.

Observability: intuition

- A level sensor on tank 2 does not read tank 1
- But tank 1 changes the flow into tank 2
- The sequence $y_0, y_1, y_2, \dots$ therefore carries information about both levels
- Close the connecting pipe and tank 1 leaves no trace in the sensor



Observability

Different initial states must leave different output histories. Then the measurements pin the state down.

Observability rank test

For $x_{k+1} = Ax_k + Bu_k$, $y_k = Cx_k$ with known inputs, subtract the input contributions from the outputs:

$$\begin{bmatrix} y_0 \\ y_1 - CBu_0 \\ \vdots \\ y_{n-1} - \dots \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}}_{\mathcal{O} \in \mathbb{R}^{np \times n}} x_0$$

Rank test

(A, C) is observable if and only if $\text{rank}(\mathcal{O}) = n$; then x_0 is the unique solution and every later x_k follows.

- Inputs change the numbers but not the rank
- Same test for $\dot{x} = Ax$, $y = Cx$
- Unobservable subspace:
 $\mathcal{N}(\mathcal{O}) = \{x : \mathcal{O}x = 0\}$, the states invisible to every sensor

Two connected tanks, sensor on tank 2 only

$$A_d = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.7 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\mathcal{O} = \begin{bmatrix} C \\ CA_d \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.2 & 0.7 \end{bmatrix}, \quad \det \mathcal{O} = -0.2$$

Both levels are observable: the second row is the coupling from tank 1 to tank 2.

Pipe closed (A_d diagonal): $CA_d = \begin{bmatrix} 0 & 0.7 \end{bmatrix}$, so

$$\mathcal{O} = \begin{bmatrix} 0 & 1 \\ 0 & 0.7 \end{bmatrix}, \quad \text{rank } \mathcal{O} = 1$$

$\mathcal{N}(\mathcal{O})$ is the x_1 axis: tank 1 is hidden.

Observability tests and detectability

Test	Condition	Interpretation
Rank	$\text{rank}[C; CA; \dots; CA^{n-1}] = n$	output history separates all states
PBH	$\text{rank} \begin{bmatrix} \lambda I - A \\ C \end{bmatrix} = n$ for every eigenvalue λ	no mode is hidden from the sensors
Eigenvector	$Cv \neq 0$ for every right eigenvector v of A	every mode shows up in y
Gramian	$W_o(T) = \int_0^T e^{A^T \tau} C^T C e^{A \tau} d\tau \succ 0$	every direction produces output energy

Detectability

(A, C) is **detectable** if some $L \in \mathbb{R}^{n \times p}$ makes $A - LC$ stable. Equivalently, every unobservable mode is stable: the PBH test needs to hold only for $\text{Re } \lambda \geq 0$ (continuous) or $|\lambda| \geq 1$ (discrete).

Duality. (A, C) is observable exactly when (A^T, C^T) is controllable, and detectable exactly when (A^T, C^T) is stabilizable. Every controllability tool has an observability twin, obtained by transposing.

Observable: reconstruct everything. Detectable: whatever you cannot reconstruct dies out on its own.

Luenberger observer

For $\dot{x} = Ax + Bu$, $y = Cx$, run a copy of the model and correct it with the output residual $y - C\hat{x}$:

$$\dot{\hat{x}} = \underbrace{A\hat{x} + Bu}_{\text{predict}} + \underbrace{L(y - C\hat{x})}_{\text{correct}}$$

$\hat{x}$ is the estimate; $L \in \mathbb{R}^{n \times p}$ is the observer gain.

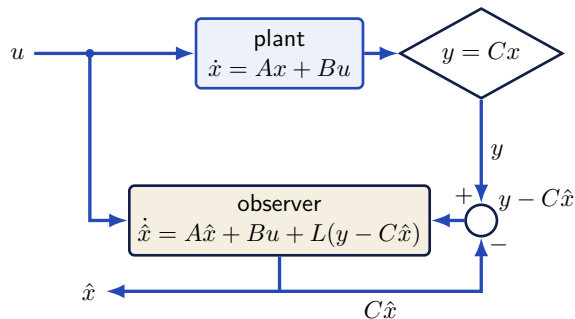
Estimation error $e = x - \hat{x}$:

$$\dot{e} = (Ax + Bu) - (A\hat{x} + Bu) - L(Cx - C\hat{x})$$

$$\dot{e} = (A - LC)e$$

The input cancels: the error dynamics are autonomous and $e \rightarrow 0$ whenever $A - LC$ is stable.

- Observable (A, C) : $\text{eig}(A - LC)$ can be placed freely
- Detectable only: the hidden modes decay at their own rate



The observer needs only u , y , and a model.

Observer design: make the error motion
 $\dot{e} = (A - LC)e$ **stable and fast enough.**

Observer gain design

Worked design. $A = \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix}$, $C = [0.5 \quad 1]$,

$$L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$$

$$A - LC = \begin{bmatrix} 1 - 0.5l_1 & 3 - l_1 \\ 3 - 0.5l_2 & 1 - l_2 \end{bmatrix}$$

$$\det(\lambda I - A + LC) = \lambda^2 + (0.5l_1 + l_2 - 2)\lambda + (2.5l_1 + 0.5l_2 - 8)$$

Observer poles $-3, -5$ need $\lambda^2 + 8\lambda + 15$:

$$0.5l_1 + l_2 = 10, \quad 2.5l_1 + 0.5l_2 = 23$$

$$L = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

By duality $L = \text{place}(A^T, C^T, \text{poles})^T$: the observer problem is pole placement for the transposed pair.

Sampled measurements. Predict with the model, correct with the new sample:

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k + L(y_k - C_d \hat{x}_k)$$

Error $e_{k+1} = (A_d - LC_d)e_k$; require $|\lambda_i(A_d - LC_d)| < 1$.

Choosing observer speed

- Faster observer poles remove initial error quickly
- Very fast poles mean large L , which amplifies sensor noise and model mismatch
- Rule of thumb: observer poles 2 to 5 times faster than the controller poles
- With noise statistics available, the Kalman filter chooses L optimally

Observer based control and separation

Full state feedback needs x ; usually we have only y . Feed back the estimate instead:

$$\boxed{u = -K\hat{x}}, \quad \dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

With $\hat{x} = x - e$

$$\dot{x} = Ax - BK(x - e) = (A - BK)x + BKe$$

$$\dot{e} = (A - LC)e$$

The controller acts on the estimate; the observer drives the estimation error to zero; the term BKe is the price of not knowing x , and it fades with e .

In coordinates $[x^\top \ e^\top]^\top$ the closed loop is

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}}_{A_{cl}} \begin{bmatrix} x \\ e \end{bmatrix}$$

Block triangular, so

$$\boxed{\text{eig}(A_{cl}) = \text{eig}(A - BK) \cup \text{eig}(A - LC)}$$

Separation principle

Design K as if x were measured and L as if there were no controller. Stabilizability of (A, B) and detectability of (A, C) suffice for a stable closed loop.

Observer based control: example

Double integrator (position x_1 , velocity x_2 , force u , position sensor)

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Controller poles $-2, -3$:

$$\det(\lambda I - A + BK) = \lambda^2 + k_2\lambda + k_1$$

$$K = \begin{bmatrix} 6 & 5 \end{bmatrix}$$

Observer poles $-5, -6$:

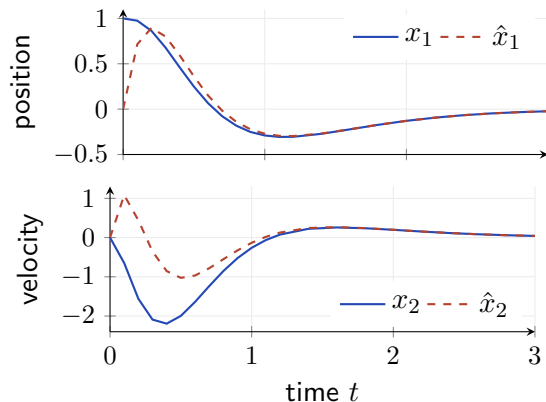
$$\det(\lambda I - A + LC) = \lambda^2 + l_1\lambda + l_2$$

$$L = \begin{bmatrix} 11 \\ 30 \end{bmatrix}$$

Implemented equations

$$u = -6\hat{x}_1 - 5\hat{x}_2$$

$$\dot{\hat{x}}_1 = \hat{x}_2 + 11(y - \hat{x}_1), \quad \dot{\hat{x}}_2 = u + 30(y - \hat{x}_1)$$

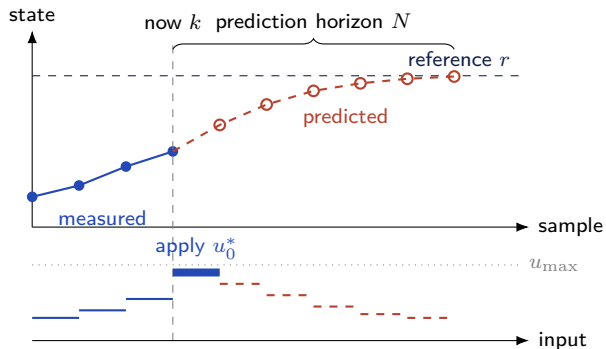


True state from $x_0 = [1 \ 0]^T$, observer started at $\hat{x}_0 = 0$. The estimates lock on within about 0.7 s (observer poles), then the position settles at the controller rate.

Receding horizon control

At every sample k

- ① Measure or estimate the current state x_k
- ② Predict N future states for a candidate input sequence $u_0, \dots, u_{N-1}$
- ③ Minimize tracking error plus control effort subject to the constraints
- ④ Apply only the first optimized input u_0^*
- ⑤ Shift the window and repeat at $k+1$



N is the **prediction horizon**. Feedback enters through step 1: the plan is recomputed from the measured state.

Stacked predictions

For $x_{i+1} = Ax_i + Bu_i$ starting from the measured x_0 (here A, B are the discrete time matrices):

$$\begin{aligned}
 x_1 &= Ax_0 + Bu_0, & x_2 &= A^2x_0 + ABu_0 + Bu_1, & \dots \\
 \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{X \in \mathbb{R}^{Nn}} &= \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathcal{A} \in \mathbb{R}^{Nn \times n}} x_0 + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\mathcal{B} \in \mathbb{R}^{Nn \times Nm}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U \in \mathbb{R}^{Nm}}
 \end{aligned}$$

- $X = \mathcal{A}x_0 + \mathcal{B}U$ is *affine* in the decision vector U : the dynamics disappear as explicit constraints
- The block rows of $\mathcal{B}$ are the reachability matrices of the previous section, so $\mathcal{B}$ has full column rank when (A, B) is controllable and $N \geq n$
- Known disturbances add a term $\mathcal{E}D$ built the same way from E

Linear MPC as a QP

Cost over the horizon (reference r , weights $Q, P \succeq 0, R \succ 0$)

$$J = \sum_{i=0}^{N-1} \ell(x_i, u_i) + V_f(x_N)$$

$$\ell(x, u) = (x - r)^T Q (x - r) + u^T R u \quad (\text{stage cost})$$

$$V_f(x) = (x - r)^T P (x - r) \quad (\text{terminal cost})$$

Subject to, for $i = 0, \dots, N - 1$,

$$x_{i+1} = Ax_i + Bu_i$$

$$x_{\min} \leq x_{i+1} \leq x_{\max}, \quad u_{\min} \leq u_i \leq u_{\max}$$

- Q prices state error, R prices effort, P prices where the horizon ends
- Affine dynamics and box constraints keep the problem convex; on/off decisions or products of variables would break it

Condensed form. Substitute $X = \mathcal{A}x_0 + \mathcal{B}U$ with $\bar{Q} = \text{diag}(Q, \dots, Q, P)$, $\bar{R} = \text{diag}(R, \dots, R)$, and $\bar{r}$ the stacked reference:

$$H = 2 (\mathcal{B}^T \bar{Q} \mathcal{B} + \bar{R}) \succ 0$$

$$f(x_0) = 2 \mathcal{B}^T \bar{Q} (\mathcal{A}x_0 - \bar{r})$$

$$\min_U \frac{1}{2} U^T H U + f(x_0)^T U \quad \text{s.t.} \quad GU \leq g + Sx_0$$

G, g, S collect the input bounds and the state bounds rewritten through $\mathcal{A}$ and $\mathcal{B}$.

- Nm decision variables, solved in milliseconds for small systems
- Only f and the right hand side depend on x_0 : H and G are built once
- Without constraints, $U^* = -H^{-1}f(x_0)$: a linear state feedback

Tank MPC example

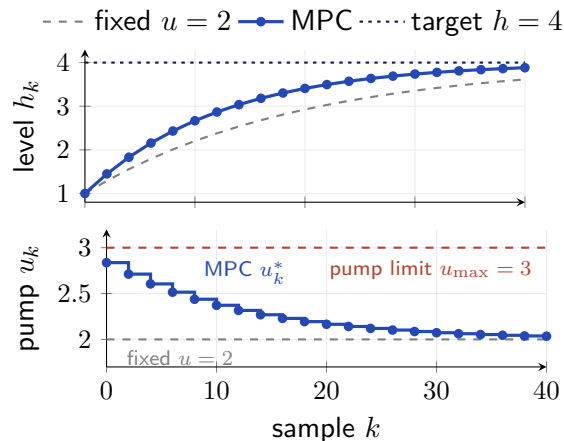
Sampled reservoir $h_{i+1} = 0.95h_i + 0.1u_i$,
target level 4, nominal pump 2, $N = 10$:

$$\min_{u_0, \dots, u_9} \sum_{i=0}^9 [(h_i - 4)^2 + 2(u_i - 2)^2] + (h_{10} - 4)^2$$

s.t. $0.5 \leq h_i \leq 5$ (service and overflow)

$0 \leq u_i \leq 3$ (pump range)

- Holding $h = 4$ needs $0.05 \cdot 4 = 0.1u$, i.e. $u = 2$: the nominal flow keeps the target but fills slowly ($h_{10} = 2.20$, $h_{40} = 3.61$)
- MPC starts at $u_0^* \approx 2.84$ and lowers the flow toward 2 as the level nears 4 ($h_{10} = 2.67$, $h_{40} = 3.88$)
- The flow above nominal closes the gap; the weight 2 on $(u - 2)^2$ keeps it under the pump limit 3



Nonlinear MPC and feasibility

Nonlinear MPC. With the gravity drained tank sampled by Euler,

$$h_{i+1} = h_i + \frac{T_s}{S} \left(u_i - a\sqrt{2gh_i} \right)$$

the predictions are nonlinear in the future levels, and the program is no longer a QP.

- A nonlinear solver may stop at a local minimum
- Initialization (warm start from the previous plan) and scaling matter
- Linearizing along the previous plan gives a QP; repeating this is *successive* or *sequential* QP
- Validate the final controller on the nonlinear model

Why a good open loop plan is not enough.

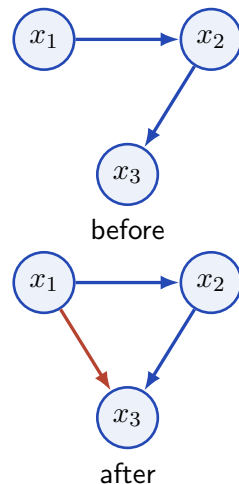
- **Recursive feasibility:** after applying u_0^* the next problem must have a solution. A terminal set $x_N \in \mathcal{X}_f$ where a local controller is known to work guarantees it
- **Stability:** a terminal cost $V_f = x^T P x$ with P from the Lyapunov equation of that local controller makes J^* a Lyapunov function
- **Soft constraints:** allow a penalized slack $s \geq 0$ in $x \leq x_{\max} + s$ so disturbances cannot make the problem infeasible
- **Robustness:** tighten constraints by a disturbance margin
- **Fallback:** a backup action (last plan shifted, or a simple feedback) when the solver fails

Infrastructure changes and system matrices

Design choice	Model change	Control question
Add a pipe, line, or road	new entries in A ; denser sparsity	do disturbances decay or spread
Add a pump or ramp meter	new column in B	can the critical modes be moved
Add a sensor	new row in C	can hidden modes be reconstructed
Add storage	new state; new couplings in A	does the extra memory improve resilience

- A new edge changes eigenvalues and modal damping
- It can create an actuator path (controllability) or a sensor path (observability)
- It can also carry a fault or congestion into a larger region

$$A_{\text{before}} = \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ 0 & * & * \end{bmatrix} \quad A_{\text{after}} = \begin{bmatrix} * & 0 & 0 \\ * & * & 0 \\ * & * & * \end{bmatrix}$$



The new edge $1 \rightarrow 3$ is a new nonzero entry A_{31} (shown in red on the left).

Physical planning fixes stability, controllability, and observability before operation begins.

Actuator and sensor placement

Actuator placement (which column to add to B)

- Must: PBH rank for every unstable mode (stabilizability)
- Better: raise $\lambda_{\min}(W_c)$, the cheapest bound on the hardest direction
- Better: lower the worst case energy $\max_{\|\Delta x\|=1} \Delta x^T W_c^{-1} \Delta x = 1/\lambda_{\min}(W_c)$
- Respect pump, valve, or meter capacity and cost

Sensor placement (which row to add to C)

- Must: PBH rank for every unstable mode (detectability)
- Better: raise $\lambda_{\min}(W_o)$, the output energy of the least visible direction
- Better: reduce estimation error covariance (Kalman filter view)
- Keep detectability after any single sensor failure

Gramians as scores

$W_c = \int_0^T e^{A\tau} B B^T e^{A^T \tau} d\tau$ and $W_o = \int_0^T e^{A^T \tau} C^T C e^{A\tau} d\tau$ depend on the candidate B or C . Common scalar scores: $\lambda_{\min}$ (worst direction), trace (average energy), $\log \det$ (volume of the reachable or observable ellipsoid). Each score is a different notion of *how well*, once rank says *whether*.

Infrastructure decisions as optimization

Ramp metering (operation)

$$\min_{r_0, \dots, r_{N-1}} \sum_{k=0}^{N-1} (q_k - q_{\text{safe}})^2 + \eta r_k^2$$

$$\text{s.t. } q_{k+1} = Aq_k + Br_k + Ed_k$$

$$0 \leq r_k \leq r_{\max}$$

q_k : link vehicle counts; r_k : metering rates; d_k : demand; q_{safe} : target counts; $\eta > 0$: effort weight.

Linear dynamics, convex cost: a QP.

Pump installation (planning + operation)

$$\min_{z, u} c^T z + \sum_{k=0}^{N-1} u_k^T R u_k$$

$$\text{s.t. } x_{k+1} = A(z)x_k + B(z)u_k$$

$$z \in \{0, 1\}^p$$

$z_j = 1$ builds candidate pump j at cost c_j ; $B(z)$ keeps only the built columns.

Binary z times continuous u : a mixed integer program.

Sensor placement (planning)

$$\min_{s \in \{0, 1\}^p} \sum_{j=1}^p c_j s_j$$

$$\text{s.t. } (A, C(s)) \text{ detectable}$$

$$\lambda_{\min}(W_o(s)) \geq \underline{w}$$

$s_j = 1$ installs candidate sensor j at cost c_j ; $C(s)$ stacks the selected rows; $\underline{w}$ is a required observability level.

Combinatorial; greedy and relaxation methods work well.

Operation problems are convex but planning problems mixed integer problems and hence nonconvex.

Joint planning and operation

Let binary $z \in \{0, 1\}^q$ select infrastructure and let u operate it:

$$\min_{z, u, x} c_{\text{build}}^T z + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k)$$

subject to

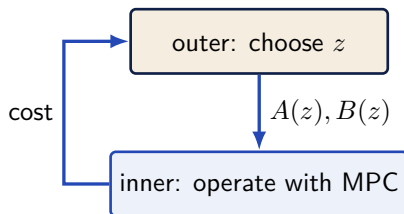
$$x_{k+1} = A(z)x_k + B(z)u_k$$

$$x_k \in \mathcal{X}(z), \quad u_k \in \mathcal{U}(z)$$

plus stability, stabilizability, or detectability requirements on $(A(z), B(z), C(z))$.

c_{build} : construction costs; $\mathcal{X}(z), \mathcal{U}(z)$: state and input limits that depend on what is built (a new tank raises x_{max} ; a new pump raises u_{max}).

- Fix z : the operation problem is a convex QP, the MPC of the previous section
- Free z : products $A(z)x_k$ and the binaries make the planning problem nonconvex
- Decompose: an outer search over z (branch and bound, heuristics) with the inner QP as its evaluator
- Rank and Gramian conditions can prune candidate z before any simulation



Module recap

Models and responses

- States are stored quantities that carry memory
- $x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu d\tau$;
eigenvalues set rates, eigenvectors set directions
- Exact sampling is $A_d = e^{AT_s}$; Euler is its first order shadow
- Linearization is a family of models indexed by the operating point; report its validity region

Stability

- Left half plane in continuous time, unit disk in discrete time
- A decreasing energy function proves stability without solving, and $A^TP + PA = -Q$ builds one for LTI systems

Control and estimation

- Controllability:
 $\text{rank}[B \ AB \ \cdots \ A^{n-1}B] = n$;
stabilizability asks only about unstable modes
- Observability:
 $\text{rank}[C; CA; \cdots; CA^{n-1}] = n$;
detectability asks only about unstable modes
- $u = -Kx$ places poles;
 $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$
reconstructs x ; the two designs separate
- MPC turns constraints into a QP solved at every sample

Infrastructure

- A pipe edits A , a pump adds a column to B , a sensor adds a row to C :
planning is control design done early