# 12. Linear Systems with Inputs and Outputs

- Systems with inputs and outputs
- Solution
- Impulse and step matrices
- Mass-spring example
- Discretization of continuous-time linear systems
- Example: forces applied to a mass
- Example: computing the state
- Solution of discrete-time LDS
- Convolution and impulse response
- Block Toeplitz matrices
- Example: controlling a hovercraft
- Comments on controlling a hovercraft

# The Key Points of This Section

- if we have a *continuous-time* linear dynamical system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

  we can discretize it to make a *discrete-time* LDS in the form

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

  which gives the behavior at sample times

- discrete-time systems are just like continuous-time ones;
  - they have a modal decomposition
  - stability is determined by eigenvalues of $A$; but we need $|\lambda| < 1$ not $\mathrm{Re}(\lambda) < 0$
  - they map $u$ to $y$ via convolution

- for control and estimation, we can form least-squares problems using the *block Toeplitz matrix* which maps $u$ to $y$

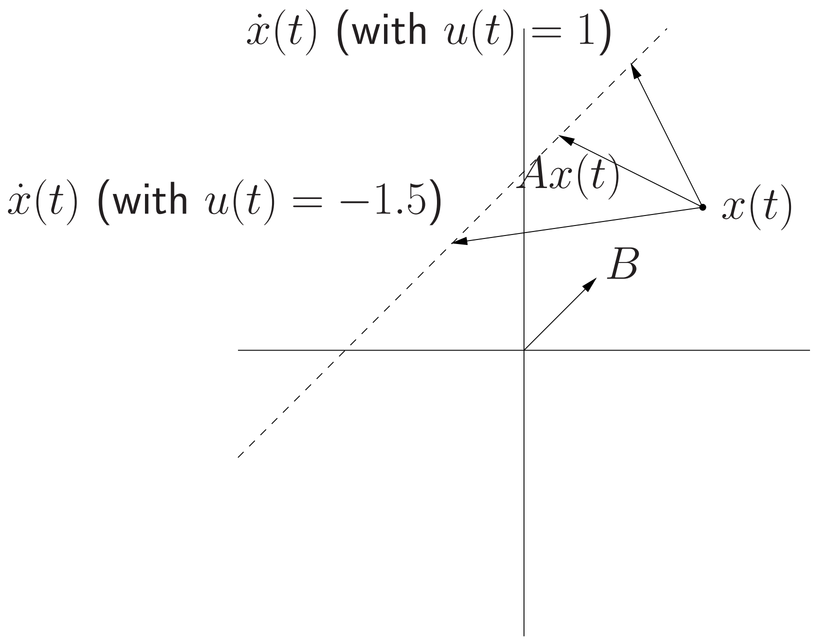
# Inputs & outputs

Continuous-time time-invariant LDS has form

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

- $Ax$ is called the *drift term* (of $\dot{x}$)
- $Bu$ is called the input term (of $\dot{x}$)

picture, with $B \in \mathbb{R}^{2\times 1}$:



# Interpretations

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

write $\dot{x} = Ax + b_1 u_1 + \cdots + b_m u_m$, where $B = [b_1 \ \cdots \ b_m]$

- state derivative is sum of autonomous term ($Ax$) and one term per input ($b_i u_i$)
- each input $u_i$ gives another degree of freedom for $\dot{x}$ (assuming columns of $B$ independent)

write $\dot{x} = Ax + Bu$ as $\dot{x}_i = \tilde{a}_i^T x + \tilde{b}_i^T u$, where $\tilde{a}_i^T$, $\tilde{b}_i^T$ are the rows of $A$, $B$

- $i$th state derivative is linear function of state $x$ and input $u$

# Solution

The solution is

$$x(t) = e^{tA}x(0) + \int_0^t e^{(t-\tau)A}Bu(\tau)\,d\tau$$
$$y(t) = Ce^{tA}x(0) + \int_0^t Ce^{(t-\tau)A}Bu(\tau)\,d\tau + Du(t)$$

- check via differentiation
- at time $t$, *current* state $x(t)$ and output $y(t)$ depend on *past* input ($u(\tau)$ for $\tau \leq t$)
- that is, mapping from input to state and output is *causal* (with fixed *initial* state)

# Impulse matrix

The *impulse matrix* is

$$h(t) = Ce^{tA}B + D\delta(t)$$

with $x(0) = 0$, $y = h * u$, *i.e.*,

$$y_i(t) = \sum_{j=1}^{m} \int_0^t h_{ij}(t-\tau)u_j(\tau)\, d\tau$$

**interpretations:**

- $h_{ij}(t)$ is impulse response from $j$th input to $i$th output
- $h_{ij}(t)$ gives $y_i$ when $u(t) = e_j\delta$
- $h_{ij}(\tau)$ shows how dependent output $i$ is, on what input $j$ was, $\tau$ seconds ago
- $i$ indexes output; $j$ indexes input; $\tau$ indexes time lag

# Step matrix

The *step matrix* or *step response matrix* is given by

$$s(t) = \int_0^t h(\tau)\, d\tau$$

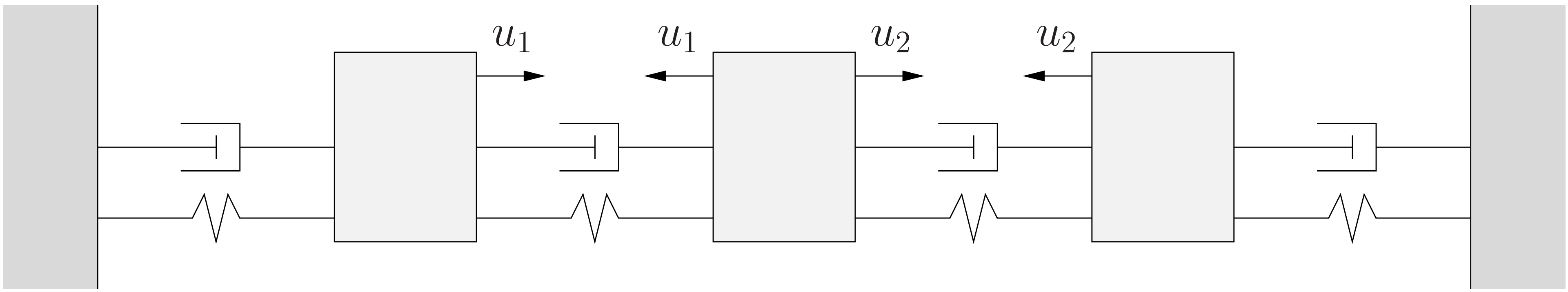
**interpretations:**

- $s_{ij}(t)$ is step response from $j$th input to $i$th output
- $s_{ij}(t)$ gives $y_i$ when $u = e_j$ for $t \geq 0$

for invertible $A$, we have

$$s(t) = CA^{-1}\left(e^{tA} - I\right)B + D$$

# Mass-Spring Example




- unit masses, springs, dampers
- $u_1$ is tension between 1st & 2nd masses
- $u_2$ is tension between 2nd & 3rd masses
- $y \in \mathbb{R}^3$ is displacement of masses 1,2,3
- $x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$

# Mass-Spring Example

system is:

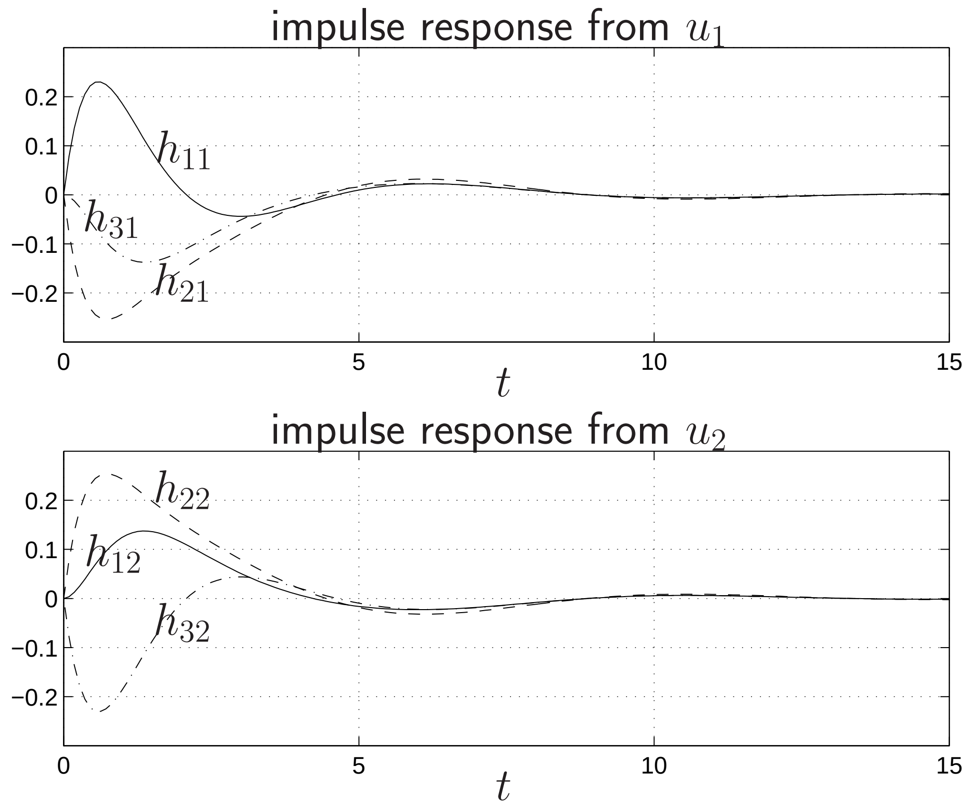
$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & -2 & 1 & 0 \\ 1 & -2 & 1 & 1 & -2 & 1 \\ 0 & 1 & -2 & 0 & 1 & -2 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

eigenvalues of $A$ are

$$-1.71 \pm j0.71, \qquad -1.00 \pm j1.00, \qquad -0.29 \pm j0.71$$

# Mass-Spring Example

impulse matrix:

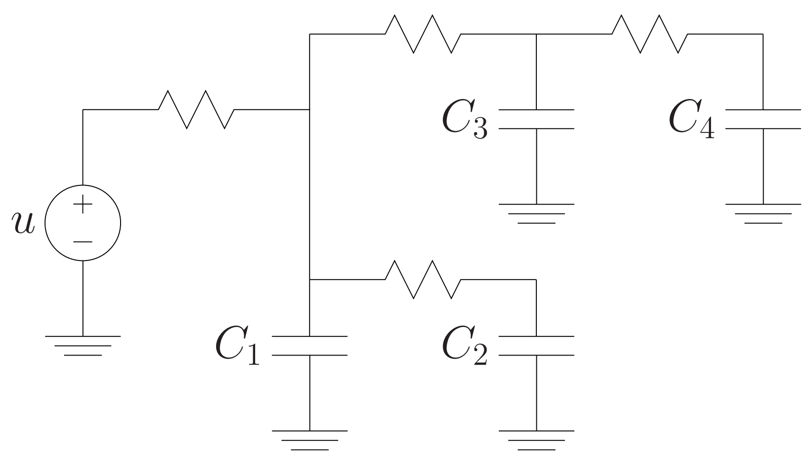



roughly speaking:

- impulse at $u_1$ affects third mass less than other two
- impulse at $u_2$ affects first mass later than other two

# Circuit Example

interconnect circuit:




- $u(t) \in \mathbb{R}$ is input (drive) voltage
- $x_i$ is voltage across $C_i$
- output is state: $y = x$
- unit resistors, unit capacitors
- step response matrix shows delay to each node

# Circuit Example

system is

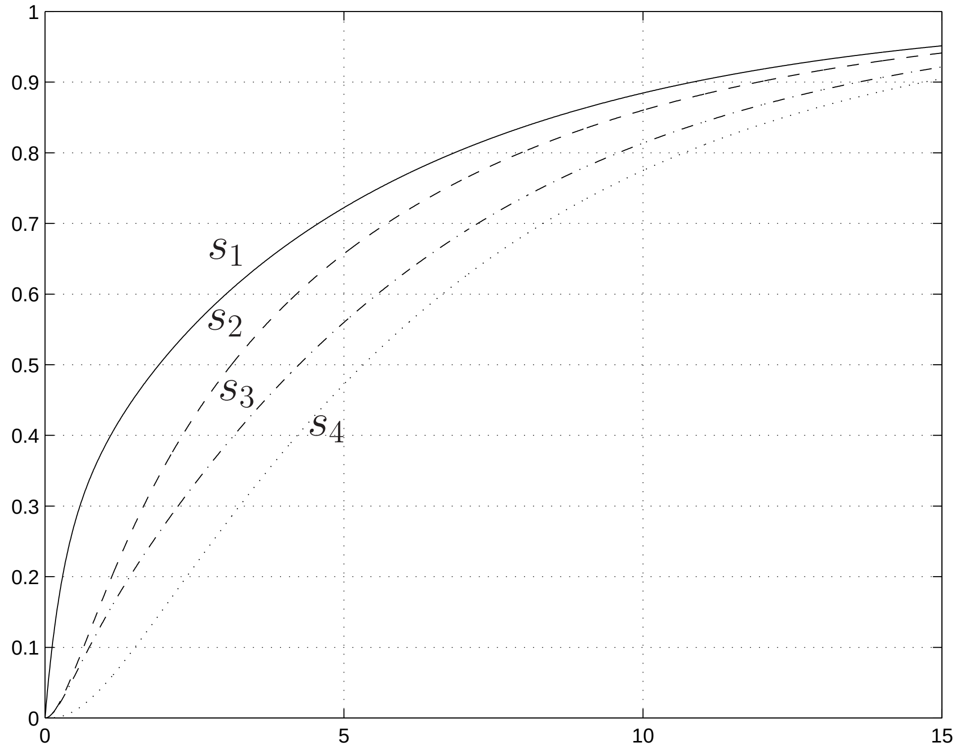
$$\dot{x} = \begin{bmatrix} -3 & 1 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u, \qquad y = x$$

eigenvalues of $A$ are

$$-0.17, \qquad -0.66, \qquad -2.21, \qquad -3.96$$

# Circuit Example

step response matrix $s(t) \in \mathbb{R}^{4\times 1}$:




- shortest delay to $x_1$; longest delay to $x_4$
- delays $\approx 10$, consistent with slowest (*i.e.*, dominant) eigenvalue $-0.17$

# DC or static gain matrix

The static gain matrix is

$$H_0 = -CA^{-1}B + D$$

- static gain matrix describes system under *static* conditions, *i.e.*, $x$, $u$, $y$ constant:

  $$0 = \dot{x} = Ax + Bu, \qquad y = Cx + Du$$

  eliminate $x$ to get $y = H_0 u$

- if system is stable,

  $$H_0 = \int_0^\infty h(t)\,dt = \lim_{t\to\infty} s(t)$$

  if $u(t) \to u_\infty \in \mathbb{R}^m$, then $y(t) \to y_\infty \in \mathbb{R}^p$ where $y_\infty = H_0 u_\infty$

# DC or static gain matrix

DC gain matrix for spring mass example:

$$H(0) = \begin{bmatrix} 1/4 & 1/4 \\ -1/2 & 1/2 \\ -1/4 & -1/4 \end{bmatrix}$$

DC gain matrix for example 2 (RC circuit):

$$H(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

## Change of coordinates

start with LDS

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

change coordinates in $\mathbb{R}^n$ to $\tilde{x}$, with $x = T\tilde{x}$, then

$$\dot{\tilde{x}} = T^{-1}\dot{x} = T^{-1}(Ax + Bu) = T^{-1}AT\tilde{x} + T^{-1}Bu$$

hence LDS can be expressed as

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u$$
$$y = \tilde{C}\tilde{x} + \tilde{D}u$$

where

$$\tilde{A} = T^{-1}AT, \qquad \tilde{B} = T^{-1}B, \qquad \tilde{C} = CT, \qquad \tilde{D} = D$$

## discrete-time systems

a discrete-time LDS has the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

# Discretization

we have the continuous-time linear dynamical system (LDS)

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \qquad\qquad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

we will *sample* $x(t)$

- sample times $0, h, 2h, 3h, \ldots$
  $h > 0$ is the *sampling interval*
- $k$'th sample is $x_d(k) = x(kh)$, and $y_d(k) = y(kh)$

we have

$$\begin{aligned} x_d(k+1) &= x(kh+h) \\ &= e^{Ah}x(kh) + \int_{kh}^{kh+h} e^{A(kh+h-\tau)}Bu(\tau)\,d\tau \\ &= e^{Ah}x_d(k) + \int_0^h e^{A(h-s)}Bu(kh+s)\,ds \end{aligned}$$

# discretization

suppose input signal $u$ is *constant between sampling times;* i.e., on intervals $[kh, (k+1)h)$

$$u(t) = u_d(k) \qquad \text{for } kh \le t < (k+1)h$$

then

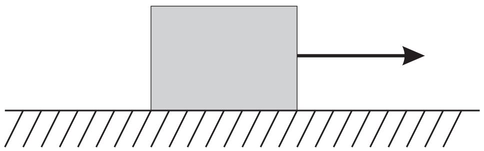
$$\begin{aligned} x_d(k+1) &= e^{Ah} x_d(k) + \int_0^h e^{A(h-s)} B u(kh+s)\, ds \\ &= e^{Ah} x_d(k) + \left( \int_0^h e^{As} B\, ds \right) u_d(k) \end{aligned}$$

so we have the *discretized system*

$$\boxed{\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_d u_d(k) \qquad\qquad x_d(0) = x_0 \\ y_d(k) &= C_d x_d(k) + D_d u_d(k) \end{aligned}}$$

$$A_d = e^{Ah} \qquad B_d = \int_0^h e^{As} B\, ds \qquad C_d = C \qquad D_d = D$$

# example: force on mass



Newton's law gives continuous-time LDS

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

let's compute the discretization

$$\begin{aligned} A_d &= e^{Ah} \\ &= I + Ah + \frac{1}{2}A^2h^2 + \cdots \\ &= I + Ah \\ &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \end{aligned}$$

# example: force on mass

also

$$
\begin{aligned}
B_d &= \int_0^h e^{As} B \, ds \\
&= \int_0^h \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \, ds \\
&= \int_0^h \begin{bmatrix} s \\ 1 \end{bmatrix} \, ds = \begin{bmatrix} \frac{1}{2}h^2 \\ h \end{bmatrix}
\end{aligned}
$$

so the discretization is

$$
\begin{aligned}
x_d(k+1) &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x_d(k) + \begin{bmatrix} \frac{1}{2}h^2 \\ h \end{bmatrix} u_d(k) \\
y_d(k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_d(k)
\end{aligned}
$$

# computing the discretization

we want to compute

$$A_d = e^{Ah} \qquad B_d = \int_0^h e^{As} B \, ds$$

useful fact: $\dfrac{d}{dt} e^{At} = A e^{At}$

$$\begin{aligned} B_d &= \int_0^h e^{As} B \, ds = A^{-1} \int_0^h A^{-1} e^{As} B \, ds \\ &= A^{-1} \Big[ e^{At} B \Big]_0^h = A^{-1}(e^{Ah} - I)B \end{aligned}$$

so, if $A$ is invertible

$$\boxed{B_d = A^{-1}(e^{Ah} - I)B}$$

# example: computing the state

what is $x_d(4)$?

$$\begin{aligned} x_d(4) &= A_d x_d(3) + B_d u_d(3) \\ &= A_d\big(A_d x_d(2) + B_d u_d(2)\big) + B_d u_d(3) \\ &= A_d^2 x_d(2) + A_d B_d u_d(2) + B_d u_d(3) \\ &\;\vdots \\ &= A_d^4 x_d(0) + \begin{bmatrix} A_d^3 B_d & A_d^2 B_d & A_d B_d & B_d \end{bmatrix} \begin{bmatrix} u_d(0) \\ u_d(1) \\ u_d(2) \\ u_d(3) \end{bmatrix} \end{aligned}$$

if $h = 1$, then

$$A_d^k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \qquad B_d = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$$

so $A_d^k B_d = \begin{bmatrix} \frac{1}{2} + k \\ 1 \end{bmatrix}$, and

$$\begin{bmatrix} A_d^3 B_d & A_d^2 B_d & A_d B_d & B_d \end{bmatrix} = \begin{bmatrix} \frac{7}{2} & \frac{5}{2} & \frac{3}{2} & \frac{1}{2} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

# Solution of Discrete-time LDS

solution of discrete-time LDS is just given by linear equations

$$x(t+1) = Ax(t) + Bu(t)$$

iterating this gives

$$x(t) = A^t x(0) + \begin{bmatrix} A^{t-1}B & A^{t-2}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(t-1) \end{bmatrix}$$

we can also write this as a sum

$$\boxed{x(t) = A^t x(0) + \sum_{\tau=0}^{t-1} A^{(t-1-\tau)} Bu(\tau)}$$

# convolution

in discrete-time, convolution takes the form

$$y(t) = \sum_{\tau=0}^{t} H(t-\tau)u(\tau)$$

- $H(t)$ is a matrix for each $t$
- the sequence $H(0), H(1), \ldots$ is called the *impulse response* of the system
- for state-space LDS, we have when $x(0) = 0$,

$$y(t) = \sum_{\tau=0}^{t-1} CA^{(t-1-\tau)}Bu(\tau) + Du(t)$$

  so the impulse response is

$$H(t) = \begin{cases} D & \text{if } t = 0 \\ CA^{t-1}B & \text{if } t > 0 \end{cases}$$

- $H_{ij}(0), H_{ij}(1), \ldots$ is the response of output $i$ to an impulse applied to input $j$

## block Toeplitz matrices

we have

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(t) \end{bmatrix} = \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & & \ddots & \\ CA^{t-1}B & CA^{t-2}B & \dots & CB & D \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(t) \end{bmatrix} + \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^t \end{bmatrix} x(0)$$

- this matrix gives the output sequence $y(0), y(1), \dots$ in terms of the input sequence $u(0), u(1), \dots$ and the initial state $x(0)$
- *block Toeplitz* means blocks are constant along diagonals from top-left to bottom right
- we can use this to find controllers and estimators

# example: hovercraft

a hovercraft, mass 1, with thrusters in directions

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad v_2 = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} \qquad v_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

has dynamics

$$x(k+1) = \begin{bmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}h^2 & 0 \\ h & 0 \\ 0 & \frac{1}{2}h^2 \\ 0 & h \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_2 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(k)$$

here

- $x_1, x_2 =$ position, velocity in **x**-direction
  $x_3, x_3 =$ position, velocity in **y**-direction
- $h =$ sample time; we'll use $h = 1$.
- $u_i(k)$ power applied to thruster $i$ at time $k$

## example: hovercraft

we would like to drive it through the positions

$$y(20) = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \qquad y(40) = \begin{bmatrix} 10 \\ -1 \end{bmatrix} \qquad y(70) = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

at the above times

we have

$$y(t) = \sum_{\tau=0}^{t-1} CA^{t-1-\tau}Bu(\tau) + Du(t)$$

this gives the rows of

$$\begin{bmatrix} y(20) \\ y(40) \\ y(70) \end{bmatrix} = A_{\mathsf{act}} \begin{bmatrix} u(0) \\ \vdots \\ u(70) \end{bmatrix}$$

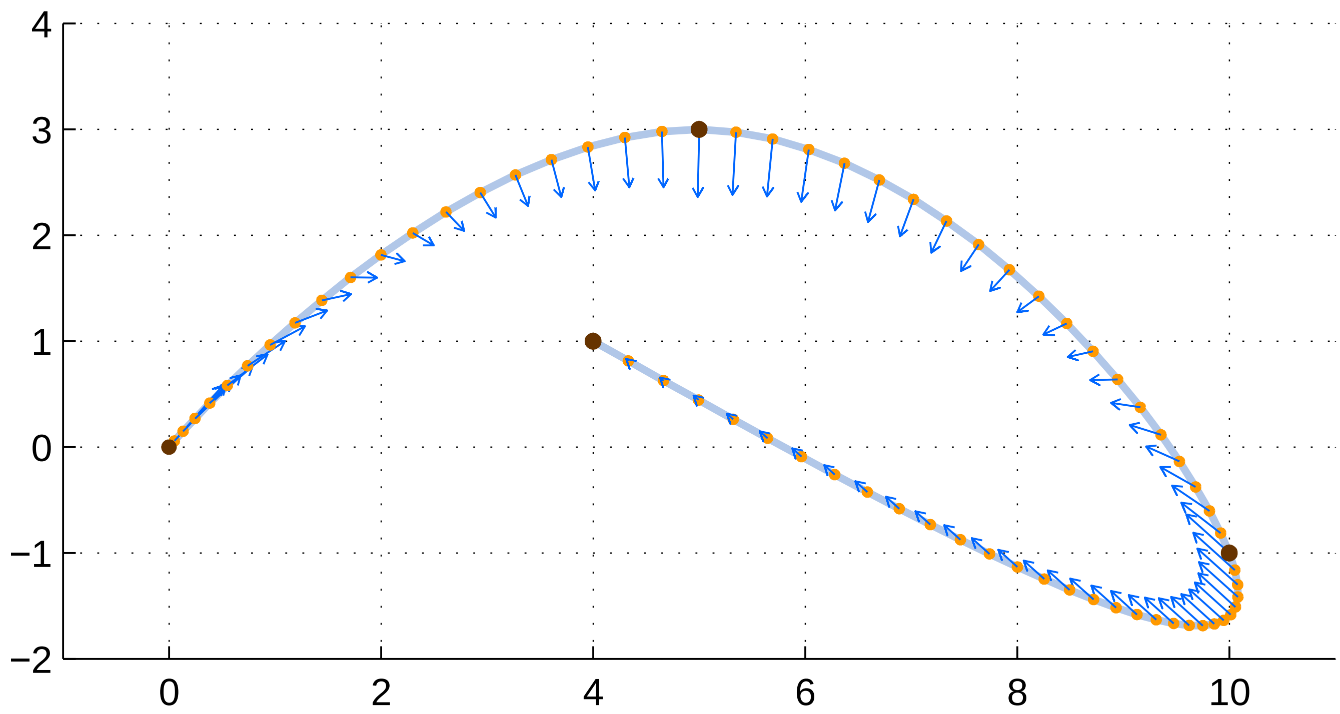
here $A_{\mathsf{act}}$ is $6 \times 213$.

# example: hovercraft

let's find the minimum norm sequence of thrusts that meets the specifications

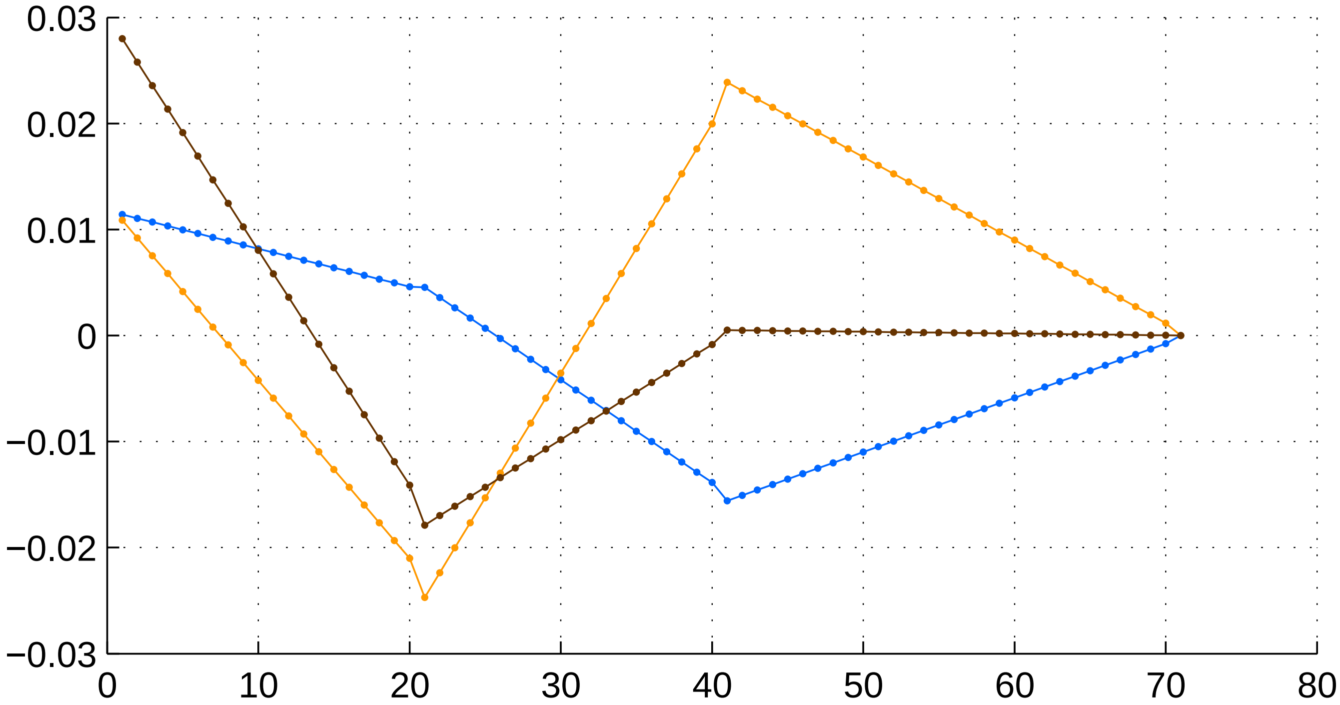
$$\begin{bmatrix} u(0) \\ \vdots \\ u(70) \end{bmatrix} = A_{\text{act}}^{\dagger} \begin{bmatrix} 5 \\ 3 \\ 10 \\ -1 \\ 4 \\ 1 \end{bmatrix}$$

hovercraft trajectory is



# example: hovercraft

sequence of thrust inputs is



## comments on this approach

- discrete-time systems have the wonderful property that

  the relationship between the sequence of inputs and the sequence of outputs is just a linear equation

- so we can apply

  least squares to find the inputs, given measurements of the outputs
  minimum-norm solutions to steer the output along some desired trajectory

- but
  - as the number of time-steps increases, so does the size of the least-squares problems
  - computation time grows as $(\text{number-of-time-steps})^3$
  - $A_{\text{act}}$ contains all powers of $A$ from 1 to the number of time-steps; so *condition numbers* grow very large
  - worst of all: controllers and estimators are *open-loop*

- we'll fix all this