Discussion 6A

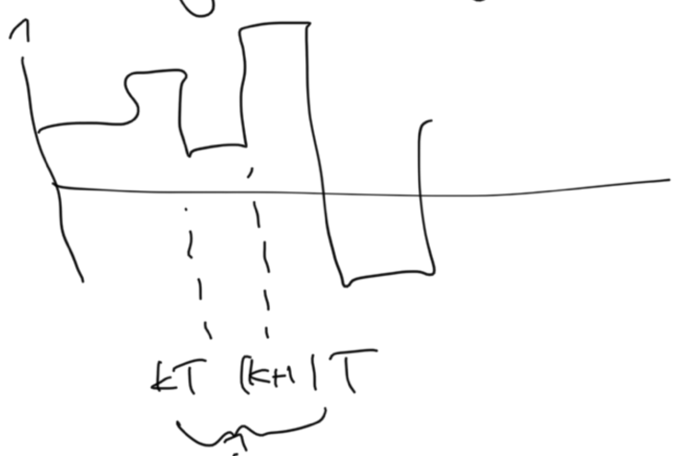
System Identification

* Least Squares
* System ID

---

0. Discretization

→ solving diff eqs w/ constant inputs



solve diff eq just in this interval

$$\frac{dx(t)}{dt} = \lambda x(t) + u u(t)$$

→ From Week 1, know how to solve!

From kT to $kT+T$

$$u(t) = u[k] = \text{const.}$$

"Trick": in this specific time interval, have a diff eq like from week 1 w/

constant inputs

Solution: $x(t) = x(kT)\,e^{\lambda(t-kT)} + \mu\, u[k] \displaystyle\int_{kT}^{t} e^{\lambda(t-\tau)}\,d\tau$

$\xrightarrow{CT} x(kT) = x[k] \leftarrow DT$

$x[k+1] = e^{\lambda T} x[k] + (1 - e^{\lambda T}) u[k]$

[For $\frac{dx}{dt} = \lambda\,(x(t) - u(t))$] ← special case

How to solve in vector case?

a) diagonalization

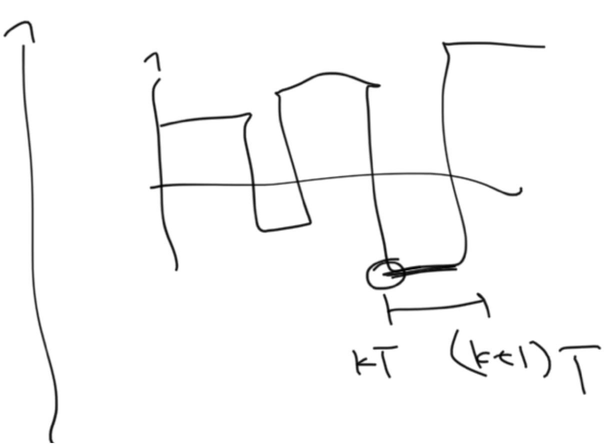
→ change basis so you have a system of scalar constant-input diff eqs like we already saw

b) non-diagonalizable

→ can do by direct integration (FTC)

→ or have to use matrix exponential (out of scope)

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} u(t)$$



$$\frac{dx}{dt} = v(t)$$

$$\frac{dv}{dt} = \frac{u(t)}{M} \longrightarrow \text{Solve for } t \in (kT, (k+1)T)$$

$$\int_{v(kT)}^{v((k+1)T)} dv = \int_{kT}^{(k+1)T} \frac{u(t)}{M}\, dt = \int_{kT}^{(k+1)T} \frac{u[k]}{M}\, dt$$

$$v((k+1)T) - v(kT) = \frac{u[k]}{M}((k+1)T - kT) = \frac{u[k]}{M} T$$

$$v[k+1] - v[k] = \frac{u[k]}{M} T$$

$$v[k+1] = v[k] + \frac{u[k]}{M} T$$

(Discretize $x(t)$ similarly — see Dis 5C)

I. Least Squares and SVD Recap

Problem Statement:

$$\vec{y} = A\vec{x} + \vec{e}$$

$A \in \mathbb{R}^{m \times n}$, $m > n$, rank n

(tall matrix, lin. ind. columns)

Goal: Find $\vec{x}^* = \min_{\vec{x}} \|\vec{e}\|^2 = \min_{\vec{x}} \|A\vec{x} - \vec{y}\|^2$

$\vec{x}$ $\vec{x}$

* LSA POV:

$$\vec{x} = (A^T A)^{-1} A^T \vec{y}$$

* LSB POV: SVD!

$$\vec{x}^* = \min_{\vec{x}} \| A\vec{x} - \vec{y} \|^2 = \| U \Sigma V^T \vec{x} - \vec{y} \|^2$$

Note $\| U \vec{x} \| = \| \vec{x} \|$

unitary transformation is length-preserving

$$= \| U^T U \Sigma V^T \vec{x} - U^T \vec{y} \|^2$$

$$= \| \Sigma V^T \vec{x} - U^T \vec{y} \|^2$$

$$= \left\| \begin{bmatrix} \Sigma_c \\ 0 \end{bmatrix} V^T \vec{x} - \begin{bmatrix} U_c^T \\ U_2^T \end{bmatrix} \vec{y} \right\|^2$$

$$= \left\| \begin{bmatrix} \Sigma_c V^T \vec{x} - U_c^T \vec{y} \\ - U_2^T \vec{y} \end{bmatrix} \right\|^2$$

can't change (fixed)
w/ choice of $\vec{x}$

We <u>can</u> choose $\vec{x}^*$ s.t. this top term is 0

$$\vec{x}^* = V\Sigma_c^{-1}U_c^T\vec{y}$$

* 16A and 16B eqns are equal (they should be)
* SVD POV can help elucidate geometric nature of problem / relation to matrix Fundamental subspaces

$$\left.\begin{matrix} U_c & U_2 \\ \underbrace{\vec{u}_1 \cdots \vec{u}_r}_{\text{span } Col(A)}, & \underbrace{\vec{u}_{r+1} \cdots \vec{u}_m}_{\text{span } Null(A^T)} \end{matrix}\right\} \text{span } \mathbb{R}^m$$

(From $A\vec{v}_i = \sigma_i \vec{u}_i$)

Consider $\vec{y}^* = A\vec{x}^* = U\Sigma V^T(V\Sigma_c^{-1}U_c^T\vec{y})$

$$= U\Sigma\Sigma_c^{-1}U_c^T\vec{y}$$

$$= U\begin{bmatrix}\Sigma_c \\ 0\end{bmatrix}\Sigma_c^{-1}U_c^T\vec{y}$$

$$= U\begin{bmatrix}I \\ 0\end{bmatrix}U_c^T\vec{y}$$

$$= [U_c\ U_2]\begin{bmatrix}I \\ 0\end{bmatrix}U_c^T\vec{y}$$

$$= (U_cI + \cancel{U_2(0)})U_c^T\vec{y}$$

$$= U_cU_c^T\vec{y}$$

What is this?

$$= \begin{bmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{bmatrix}\left(\begin{bmatrix} \vdash u_1^T \dashv \\ \vdots \\ \vdash u_r^T \dashv \end{bmatrix}\right)\vec{y}$$

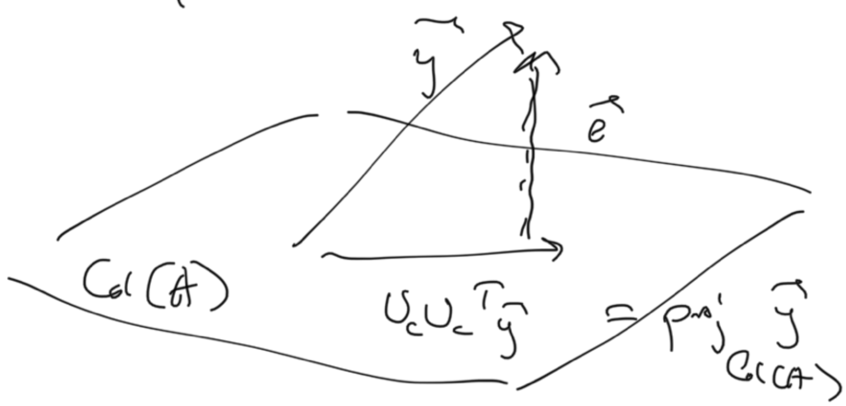
$$= \begin{bmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{bmatrix} \begin{bmatrix} u_1^T y \\ \vdots \\ u_r^T y \end{bmatrix}$$

$$= \begin{bmatrix} | & & | \\ u_1 & \cdots & u_r \\ | & & | \end{bmatrix} \begin{bmatrix} \langle u_1, y \rangle \\ \vdots \\ \langle u_r, y \rangle \end{bmatrix}$$

$$= \langle \vec{u}_1, \vec{y} \rangle \hat{u}_1 + \langle \vec{u}_2, \vec{y} \rangle \vec{u}_2 + \cdots + \langle \vec{u}_r, \vec{y} \rangle \vec{u}_r$$

Projection of $\vec{y}$ onto the columns of U_c

$\Longrightarrow$ projection onto the Col(A)!

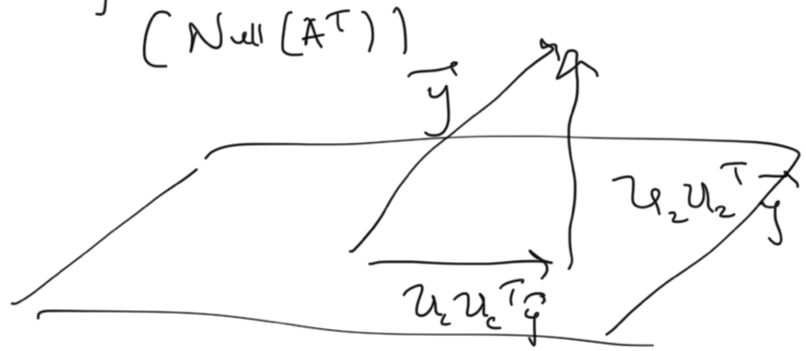


In fact,

$$\vec{y} = I\vec{y} = UU^T\vec{y} = [U_c \; U_2] \begin{bmatrix} U_c^T \\ U_2^T \end{bmatrix} \vec{y}$$

$$= U_c U_c^T \vec{y} + U_2 U_2^T \vec{y}$$

projection of $\vec{y}$ onto the $\perp$ complement of Col(A)



When we choose $\vec{x}^*$ s.t.

$$\|\vec{e}\| = \left\| \begin{bmatrix} 0 \\ U_2^T\vec{y} \end{bmatrix} \right\| = \left\| U \begin{bmatrix} 0 \\ U_2^T\vec{y} \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} U_c & U_2 \end{bmatrix} \begin{bmatrix} 0 \\ U_2^T\vec{y} \end{bmatrix} \right\|$$

$$= \| U_2 U_2^T \vec{y} \|$$

only leaving $\perp$ complement's contribution in the error

\# ASIDE: similar exercise w/ min norm solution

$$\vec{x}^* = \min_{\vec{x}} \|\vec{x}\| \quad \text{s.t.} \quad A\vec{x} = \vec{y}$$

(wide matrix, lin. ind. rows)

$$\|\vec{x}\| = \|V^T\vec{x}\| = \left\| \begin{bmatrix} V_c^T\vec{x} \\ V_2^T\vec{x} \end{bmatrix} \right\|$$

$$\left.\underbrace{\vec{v}_1 \ldots \vec{v}_r}, \underbrace{\vec{v}_{r+1} \ldots \vec{v}_n}^{T}\right\} \text{span } \mathbb{R}^n$$

$Col(A^T)$ $\quad$ $Null(A)$

To minimize, $\vec{x}^*$ S.T. $V_2^T \vec{x}^* = 0$

$$\begin{bmatrix} \langle \vec{v}_{r+1}, \vec{x}^* \rangle \\ \vdots \\ \langle \vec{v}_n, \vec{x} \rangle \end{bmatrix} = 0$$

$\vec{x}^*$ is $\perp$ to Null (A)

(No component of $\vec{x}^*$ lies in Null (A))

If we did change of basis

$$\vec{x}^*_V = V^T \vec{x}^* = \begin{bmatrix} c_1 \\ \vdots \\ c_r \\ c_{r+1} \\ \vdots \\ c_n \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} c_1 \\ \vdots \\ c_r \end{matrix}}\right\} \rightarrow \perp \text{ basis for } Col(A^T) \\ \left.\vphantom{\begin{matrix} c_{r+1} \\ \vdots \\ c_n \end{matrix}}\right\} \rightarrow \perp \text{ basis for } Null(A) \end{matrix}$$

Coordinates of $\vec{x}^*$ in the V basis

Then minimum norm solution, which has no component lying along the Null(A), looks like:

$$\vec{x}^*_V = \begin{bmatrix} c_1 \\ \vdots \\ c_r \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$ all 0!

* LS Example : Polynomial Fitting

Know output follows:

$$y = a_0 + a_1 x + a_2 x^2$$

e.g. $\Delta x = v_0 t + \frac{1}{2} a t^2$

x	y
0	0
0.5	4
1	10
1.5	18
2	30

$\rightsquigarrow$ $A\vec{x} + \vec{e} = \vec{y}$?

Unknown params: a_0, a_1, a_2

Measure: y_1, y_2, y_3, y_4, y_5

"Input": x_1, x_2, x_3, x_4, x_5

$$\left\{ \begin{array}{l} y_1 = a_0 + a_1 x_1 + a_2 x_1^2 \\ \vdots \\ y_5 = a_0 + a_1 x_5 + a_2 x_5^2 \end{array} \right.$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \\ 1 & x_5 & x_5^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

$$\rightsquigarrow \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0.25 \\ 1 & 1 & 1 \\ 1 & 1.5 & 2.25 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ 10 \\ 18 \\ 30 \end{bmatrix}$$

$$D \qquad \vec{p} \approx \vec{y}$$

$$\hat{\vec{p}} = (D^T D)^{-1} D^T \vec{y}$$

$$a_0 = 0.17$$
$$a_1 = 4.5$$
$$a_2 = 5.14$$

I was roughly using $\Delta x \approx 5t + \frac{1}{2} 10 t^2$

4. System ID

Just LS applied to state space models!

a) $x[k+1] = a x[k] + b u[k] + e[k] \leftarrow$ noise, disturbance, error

Observe initial state $x[0]$:

input $u[0]$, output $x[1]$

input $u[1]$, output $x[2]$

$\vdots$

$$x[1] = a x[0] + b u[0] + e_1$$
$$x[2] = a x[1] + b u[1] + e_2$$
$$\vdots$$
$$x[n+1] = a x[n] + b u[n] + e_n$$

$$\begin{bmatrix} x[1] \end{bmatrix} \quad \begin{bmatrix} x[0] & u[0] \end{bmatrix} \begin{bmatrix} a \end{bmatrix}$$

$$\begin{bmatrix} \vdots \\ x[n+1] \end{bmatrix} = \begin{bmatrix} x[1] & u[1] \\ & \vdots \\ x[n] & u[n] \end{bmatrix} [\,b\,] + \vec{e}$$

$\vec{y}$ $\qquad D \qquad \vec{p}$

$$\vec{y} \approx D\hat{\vec{p}}$$

Know how to solve!

$$\boxed{\vec{p}^{*} = (D^T D)^{-1} D^T \vec{y}}$$

Best fitting parameters for identifying the system

b) Vector Case

Messy symbolically, but essentially, unrolling vector system into a bunch of scalar eqns

$$\begin{bmatrix} x_0[k+1] \\ x_1[k+1] \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0[k] \\ x_1[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k]$$

$$x_0[k+1] = 0$$

$$x_1[k+1] = x_1[k] + u[k]$$

$$A = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{n1} & & \cdots & a_{nn} \end{bmatrix}}_{n \times n}$$

since $\vec{x} \in \mathbb{R}^n$

$$B = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \quad \text{since } \vec{u} \in \mathbb{R}^m$$

$$\begin{array}{c} n \times m \\ \end{array}$$

$$\begin{bmatrix} x_1[k+1] \\ \vdots \\ x_n[k+1] \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1[k] \\ \vdots \\ x_n[k] \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix} \begin{bmatrix} u_1[k] \\ \vdots \\ u_m[k] \end{bmatrix}$$

Look at one row:

$$x_j[k+1] = \sum_{j=1}^{n} a_{ij} x_j[k] + \sum_{j=1}^{m} b_{ij} u_j[k]$$

$$= \underbrace{\begin{bmatrix} x_1[k] & \cdots & x_n[k] & u_1[k] & \cdots & u_m[k] \end{bmatrix}}_{m+n \text{ entries in total}} \begin{bmatrix} a_{j1} \\ \vdots \\ a_{jn} \\ b_{j1} \\ \vdots \\ b_{jm} \end{bmatrix}$$

Can do this for all $j = 1 \ldots n$

$\Rightarrow$ create a giant stacked vector of parameters

$$\vec{p} \in \mathbb{R}^{(m+n)n}$$

$$D \in \mathbb{R}^{nK \times n(m+n)}$$

K = # of time steps you observe

(not including initial state $\vec{x}[0]$)

For D to be a "tall" matrix,

$$nK \geq n(m+n)$$

$$\Rightarrow K \geq m+n$$

(Need more than $m+n$ timesteps observed) to do LS / System ID

Worksheet

(1) System ID and Linear Control

$$x[t+1] = \lambda x[t] + g(u[t])$$

a) Approx $g(u)$ to $O(2)$ about $u^* = 0$

$\Longrightarrow$ Taylor expand to order 2 about $u^* = 0$

(before, plain linearization $\longrightarrow$ order 1)

Recall: $f(x) \approx f(x^*) + f'(x^*)(x-x^*)$

$$+ \frac{f''(x^*)(x-x^*)^2}{2!}$$

Truncate here to get $O(2)$

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x^*)(x-x^*)^n}{n!} \qquad + \frac{f'''(x^*)(x-x^*)^3}{3!}$$

$$+ \dots$$

In this case, $x^* \Longrightarrow u^* = 0$

$$f(x) \Longrightarrow g(u)$$

$$g(u) \simeq g(0) + g'(0)(u-0) + \frac{g''(0)}{2}(u-0)^2$$

$$= \beta_0 + \beta_1 u + \beta_2 u^2$$

We see: $\boxed{\begin{aligned} \beta_0 &= g(0) \\ \beta_1 &= g'(0) \\ \beta_2 &= \frac{g''(0)}{2} \end{aligned}}$

Plug into $x[t+1] \simeq \lambda x[t] + \beta_0 + \beta_1 u[t] + \beta_2 u^2[t]$

b) $x[0] = 0$

Input $u[0], \ldots, u[N-1]$

$\downarrow$

Observe $x[1] \ldots x[N]$

LS estimates of $\lambda, \beta_0, \beta_1, \beta_2$?

$$\Longrightarrow \vec{p} = \begin{bmatrix} \lambda \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$x[1] = \lambda x[0] + \beta_0 + \beta_1 u[0] + \beta_2 u^2[0]$$

$$\vdots$$

$$x[N]$$

$$\vec{y} \approx D\vec{p}$$

$$\begin{bmatrix} x[1] \\ \vdots \\ x[N] \end{bmatrix} \approx \begin{bmatrix} x[0] & 1 & u[0] & u^2[0] \\ \vdots & \vdots & \vdots & \vdots \\ x[N-1] & 1 & u[N-1] & u^2[N-1] \end{bmatrix} \begin{bmatrix} \lambda \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

$$\boxed{\hat{\vec{p}} = (D^T D)^{-1} D^T \vec{y}}$$

where $D, \vec{y}$ defined above

② System ID

$$\vec{x}(t+1) = A\vec{x}(t) + Bu(t) + \vec{w}(t)$$

$$A = \begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \quad B = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \quad x(t) = \begin{bmatrix} x_0(t) \\ x_1(t) \end{bmatrix}$$

a) Assume $w(t) = 0$
(perfect model)

$x_0[t+1] = ?$

$x_1[t+1] = ?$

Scalar eqns $\Longrightarrow$ unravel $\underline{Ax + Bu}$

$$\boxed{\begin{aligned} x_0[t+1] &= a_0 x_0[t] + a_1 x_1[t] + b_0 u[t] \\ x_1[t+1] &= a_2 x_0[t] + a_3 x_1[t] + b_1 u[t] \end{aligned}}$$

b) How many inputs / timesteps do we need to observe?

$u[0], u[1] \cdots u[m]$ $\qquad m = ?$

observe $\downarrow$

$\vec{x}[1] \quad \cdots \quad \vec{x}[m+1]$

$m \stackrel{?}{=} 4$

6 unknowns $\longrightarrow$ need minimum 6 eqns

- Each time step gives us 2 eqns
- $\boxed{m = \frac{6 \text{ eqn}}{2 \text{ eqns / timestep}} = 3 \quad \text{timesteps}}$

From lecture:

$\vec{x} \in \mathbb{R}^n, \ \vec{u} \in \mathbb{R}^m$

$\Longrightarrow n = 2, \ m = 1$

Then $\vec{p} \in \mathbb{R}^{(m+n) \cdot n} = \mathbb{R}^6$

$D \in \mathbb{R}^{nK \times n(m+n)} = \mathbb{R}^{2K \times 6}$

So to be "tall",

$2K \geq 6 \Longrightarrow \underline{K \geq 3} \Longrightarrow m = 3$ timesteps minimum

c) Set up a LS problem $D\vec{p} \approx \vec{y}$

$$D = \begin{bmatrix} x_0[0] & x_1[0] & u[0] & 0 & 0 & 0 \\ x_0[1] & x_1[1] & u[1] & 0 & 0 & 0 \\ x_0[2] & x_1[2] & u[2] & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} x_0[3] & x_1[3] & u[3] & 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

What should $\vec{y}$, $\vec{p}$ be?

$$\boxed{\vec{p} = \begin{bmatrix} a_0 \\ a_1 \\ b_0 \\ a_2 \\ a_3 \\ b_1 \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} x_0[1] \\ x_0[2] \\ x_0[3] \\ x_0[4] \\ x_1[1] \\ x_1[2] \\ x_1[3] \\ x_1[4] \end{bmatrix}}$$

$$x_0[1] = a_0 x_0[0] + a_1 x_1[1] + b_0 u[0]$$

Because symmetry of D, you could flip the order for both $\vec{y}$ and $\vec{p}$

$$\vec{y} = \begin{bmatrix} x_1[1] \\ \vdots \\ x_1[4] \\ x_0[1] \\ \vdots \\ x_0[4] \end{bmatrix} \qquad \vec{p} = \begin{bmatrix} a_2 \\ a_3 \\ b_1 \\ a_0 \\ a_1 \\ b_0 \end{bmatrix}$$

also a valid solution

d) $\boxed{\vec{p}^* = (D^T D)^{-1} D^T \vec{y}}$

Impt: Assumed columns of D are lin. independent

$\rightarrow$ thus we can invert $D^T D$

e) What could go wrong? WCGW?

$\boxed{\text{Could be that } D \text{ is not invertible!}}$

$$D = \begin{bmatrix} x_0[0] & x_1[0] & u[0] & \\ x_0[1] & x_1[1] & u[1] & 0 \\ x_0[2] & x_1[2] & u[2] & \\ x_0[3] & x_1[3] & u[3] & \\ & 0 & & " \end{bmatrix}$$

$x_0[0] = x_1[0]$
$x_0[1] = x_1[1]$ $\implies \vec{x}_0 = \vec{x}_1$ $\ddot{\frown}$

"Control Law": $u[t] = x_0[t]$

(Plug observed output back into the input)

$$D = \begin{bmatrix} x_0[0] & x_1[0] & u[0] = x_0[0] & \\ x_0[1] & x_1[1] & u[1] = x_0[1] & 0 \\ x_0[2] & x_1[2] & u[2] = x_0[2] & \\ x_0[3] & x_1[3] & u[3] = x_0[3] & \\ & 0 & & " \end{bmatrix}$$

the same column!

By poor choice of inputs, we messed up the system ID (have to be careful!)