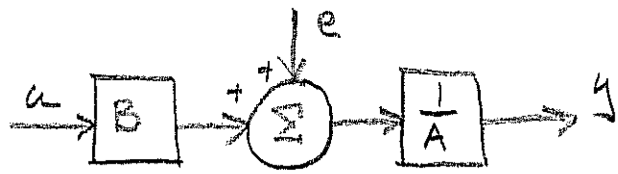
Lesson 10: ARX Models

Objectives:
- Projection
- Least-Squares

1.) ARX Models:



n = # input observations
m = # output observations
N = # samples collected

$$A(z)Y(z) = B(z)U(z) + E(z)$$

$$\hat{y}[k] = \frac{b_0 u[k] + b_1 u[k-1] + \cdots + b_m u[k-m]}{a_1 y[k-1] + a_2 u[k-2] + \cdots + a_n u[k-n]}$$

→ Z-Transform yields z-domain transfer function:
- → Poles and Zeros
  - → Roots of denominator & numerator.
  - → Encapsulate frequencies & damping.
- → Derive DFT (DTFT) of transfer function:

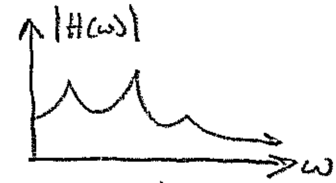
$$z = e^{i\omega T}$$

- Why not build DFT of Transfer Function directly?

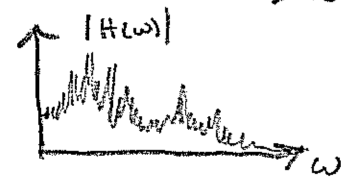
$$DFT(h[n]) = \frac{DFT(y[n])}{DFT(u[n])}$$

Problems: 1.) Data is noisy.

We want:



We get:



2.) Large models: $n = m = N$

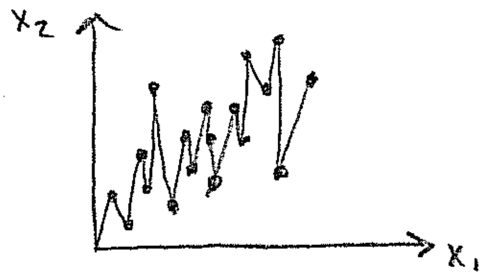
N is often $O(10^3 - 10^5)$

– Models should be smaller than the data set.

2.) Model size Reduction:

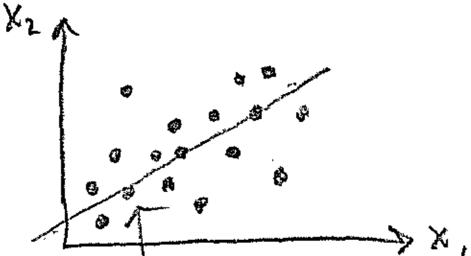
Options: 1.) Downsampling
2.) Windowing / Averaging
3.) Projection / Least-squares ⟵ This one

Consider data models:



Model Size = Data Set Size

(Not very clever)



Linear regression

Model size = 2 < Data Set size

(Actually, Affine)

Projection:
High order ⟶ Low order

EXAMPLE

n	x_1	x_2
1	0	1
2	5	8
3	3	4
4	5	6
5	7	10
6	2	4
7	3	7
8	1	5

$x_2 = f(x_1) = c_0 + c_1 x_1$

$$\begin{vmatrix} c_0 & & = 1 \\ c_0 + 5\,c_1 & & = 8 \\ c_0 + 3\,c_1 & & = 4 \\ c_0 + 5\,c_1 & & = 6 \\ c_0 + 7\,c_1 & & = 10 \\ c_0 + 2\,c_1 & & = 4 \\ c_0 + 3\,c_1 & & = 7 \\ c_0 + 1\,c_1 & & = 3 \end{vmatrix}$$

8 eqns, 2 unknowns
↳ use least-squares!

Least-squares soln:

$\bar{A}\,\bar{x} = \bar{b}$ $\quad$ $\bar{A}$ = Data matrix (8x2)
$\bar{x}$ = Vector of least-squares solution (2x1)
$\bar{b}$ = Vector of outputs (8x1)

Square ⟶ $[\bar{A}^T \bar{A}]\bar{x} = \bar{A}^T \bar{b}$

$$\underset{2\times1}{\bar{x}} = \underset{2\times2}{[\bar{A}^T \bar{A}]^{-1}}\ \underset{2\times8}{\bar{A}^T}\ \underset{8\times1}{\bar{b}}$$

SIZE?
8x8?
2x2?

EXAMPLE CONTINUED:

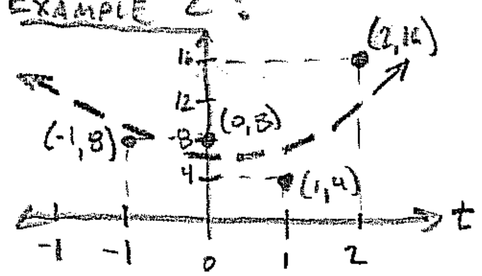
$$\begin{bmatrix} 1 & 0 \\ 1 & 5 \\ 1 & 3 \\ 1 & 5 \\ 1 & 7 \\ 1 & 2 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 4 \\ 6 \\ 10 \\ 4 \\ 7 \\ 3 \end{bmatrix} = \begin{bmatrix} 8 & 26 \\ 26 & 122 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 43 \\ 184 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1.54 \\ 1.18 \end{bmatrix}$$

$$\boxed{\text{Linear Model: } X_2 = 1.54 + 1.18\, x_1}$$

→ More Complex Models ok? Yes!

EXAMPLE 2:



$f(t) = c_0 + c_1 t + c_2 t^2$
parabolic regression

$$\bar{A}\bar{x} = \bar{b}$$

$$\begin{vmatrix} c_0 + t_1 c_1 + t_1^2 c_2 \\ c_0 + t_2 c_1 + t_2^2 c_2 \\ c_0 + t_3 c_1 + t_3^2 c_2 \\ c_0 + t_4 c_1 + t_4^2 c_2 \end{vmatrix} \rightarrow \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 8 \\ 4 \\ 16 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 2 & 6 \\ 2 & 6 & 8 \\ 6 & 8 & 18 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 36 \\ 28 \\ 78 \end{bmatrix}$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 3 \end{bmatrix}$$

$$\boxed{\text{Parabolic Model: } \quad f(t) = 5 - t + 3t^3}$$

→ We can use same process to find ARX coefficients! $a_1, a_2, \ldots, a_n$, $b_0, b_1, \ldots, b_m$

3.) Least-squares for ARX:

Basic form: $\bar{\Phi}\hat{\bar{\Theta}} = \bar{b}$

$(\bar{A}\bar{x} = \bar{b})$

n = # output observations
m = # input observations
N = # samples collected
$s = n + p + 1$
define: $p = \max(m, n)$

$$y[k] = \frac{b_0 u[k] + b_1 u[k-1] + \cdots + b_m u[k-m]}{a_1 y[k-1] + a_2 y[k-2] + \cdots + b_n u[k-n]}$$

$$\hat{\bar{\Theta}} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}_{s \times 1}$$ ← we want this

$$\bar{b} = \begin{bmatrix} y[p+1] \\ y[p+2] \\ \vdots \\ y[k] \\ \vdots \\ y[N] \end{bmatrix}_{(N-p)\times 1}$$ ← output signal, minus some of the initial points.

What if the system is at rest for $k < 0$?

$$\bar{\Phi} = \begin{bmatrix} -y[p] & -y[p-1] & \cdots & -y[p+1-n] & u[p+1] & u[p] & \cdots & u[p+1-m] \\ -y[p+1] & -y[p] & \cdots & -y[p+2-n] & u[p+2] & u[p+1] & \cdots & u[p+2-m] \\ & & & \vdots & & & & \\ -y[k-1] & -y[k-2] & \cdots & -y[k-n] & u[k] & u[k-1] & \cdots & u[k-m] \\ & & & \vdots & & & & \\ -y[N] & -y[N-1] & \cdots & -y[N-n] & u[N] & u[N-1] & \cdots & u[N-m] \end{bmatrix}_{(N-p)\times s}$$

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}$$

$$H(z) = z^{(n-m)} \frac{b_0 z^m + b_1 z^{m-1} + b_2 z^{m-2} + \cdots + b_m}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_n}$$

Roots Give Poles

→ * Uses "batch" of data samples → "Batch" Least-Squares