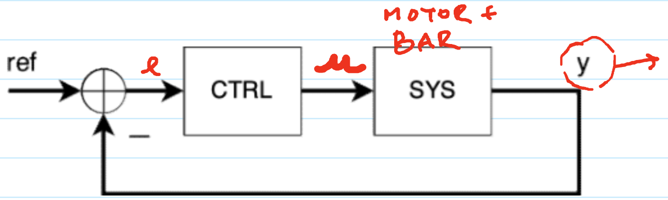
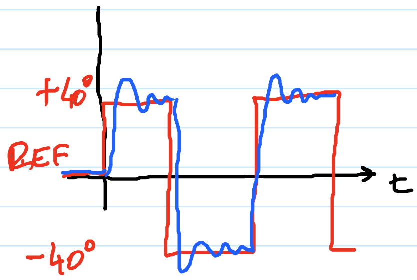
Computer Control Lab Project 2020/21 - Session 1

Monday, 19 October 2020 14:41

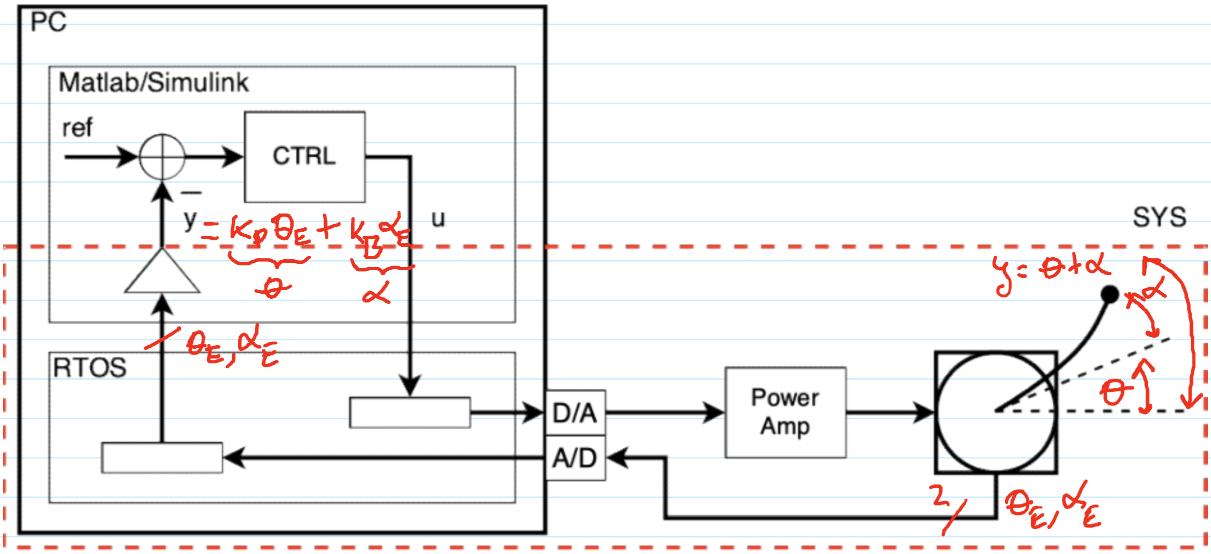
Conceptual configuration for the control system



Angle of the tip of the bar



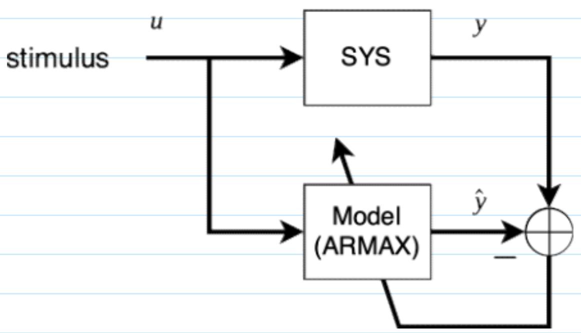
With actual hardware



Controlller design

1. PID ("heuristical")
2. Optimal control design (LQR, LQE, LQG...)
    a. Model-based design
    b. Need a state-model for our system to design the controller
    c. Task 1: Identify the system
    d. Task 2: Design the controller

Classical system identification framework

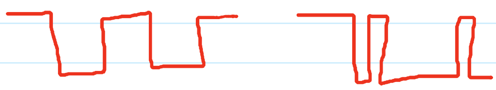


Successful learning:
Y and \hat{y} are similar for a given stimulus

Task 1: System identification

1. Create stimulus

SQUARE PRBS



M-FILE

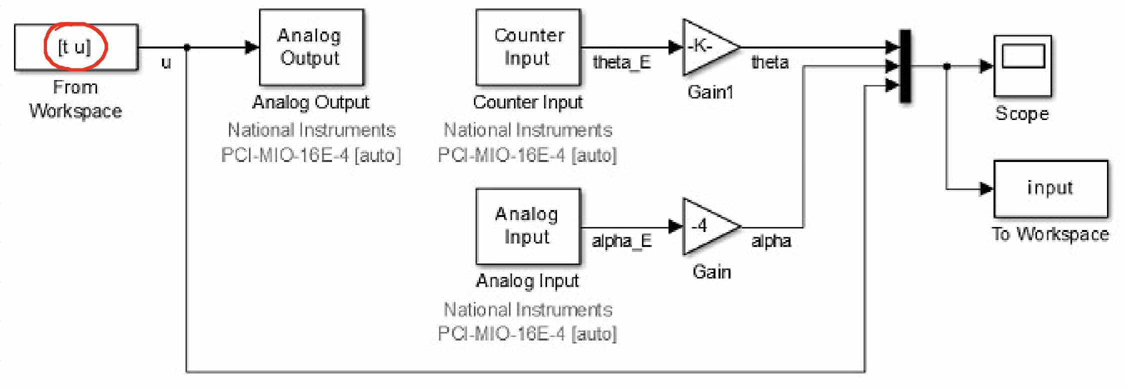
T = 120; (total duration for the stimulus)
f = 0.4; fs = 100;
(switching frequency)
t = (0:1/fs:T);
u = square(2*pi*f*t); (create square wave)
u = idinput(length(t), 'prbs', [0 B])

0.1 ~ 0.2

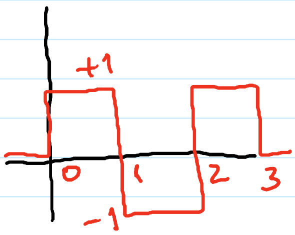
2. Apply the stimulus and record output

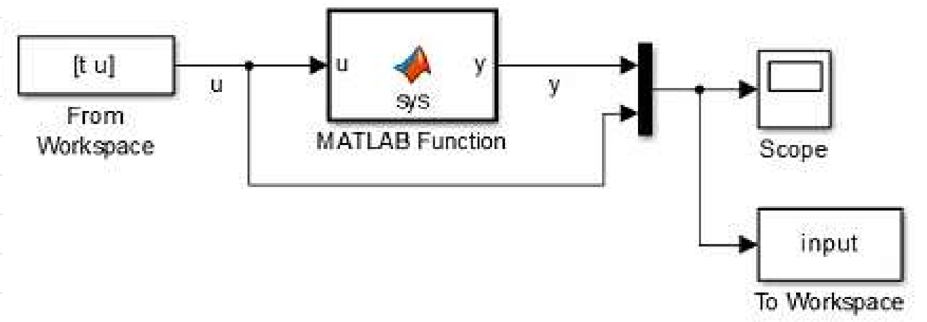
SIMULINK

HISTORY



$$[t \ u] = \begin{bmatrix} 0 & 1 \\ 1 & -1 \\ 2 & 1 \\ 3 & 0 \end{bmatrix}$$

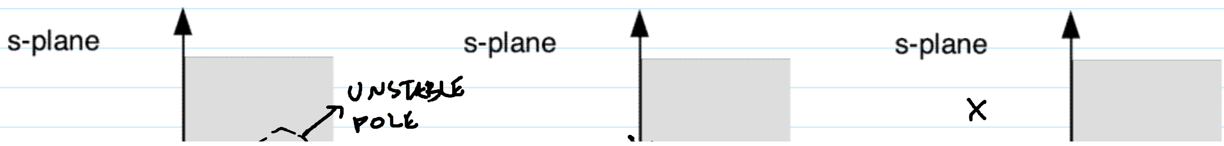


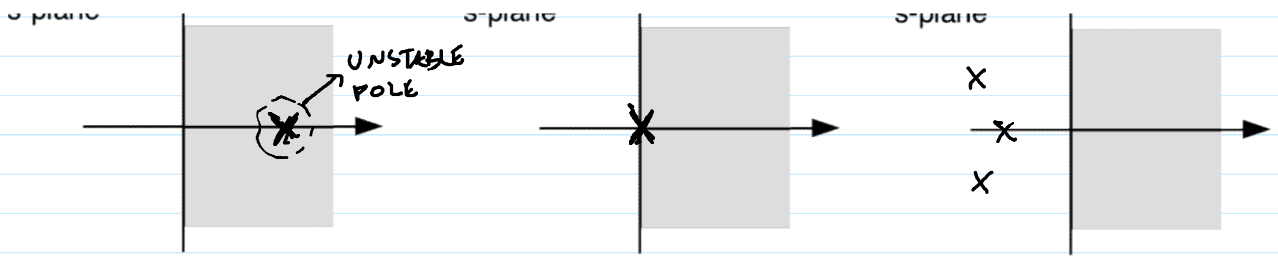


3. Preprocessing

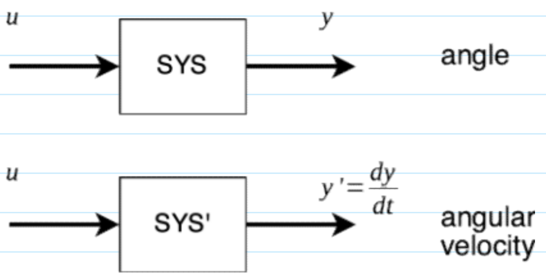
$$y = \underbrace{K_P \theta_E}_{\theta} + \underbrace{K_B \alpha_E}_{\alpha}$$

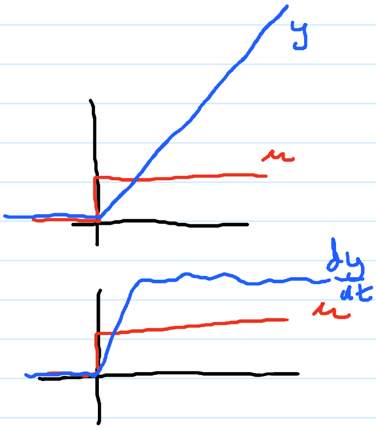
CAUSAL, UNSTABLE CRITICALLY STABLE CAUSAL, STABLE





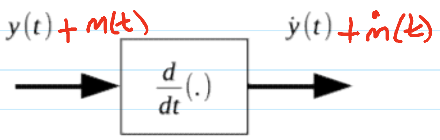
Modified system





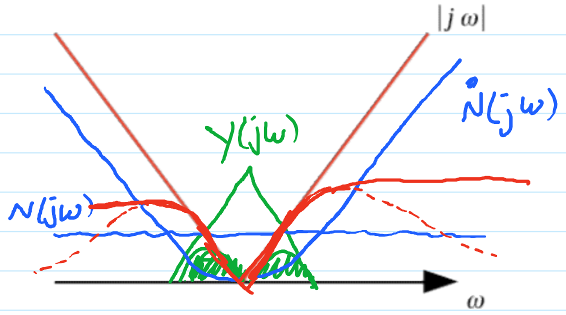
y' -- Modified output (angular velocity)

$$y'(n) = y(n) - y(n-1) \longrightarrow Y'(z) = \underbrace{(1-z^{-1})}_{\text{TRANSFER FUNCTION}} Y(z)$$



$$\lambda \rightarrow 0.8 \ldots 0.95$$

$$Y'(z) = \underbrace{(1-z^{-1})\left(\frac{1-\lambda}{1-\lambda z^{-1}}\right)}_{\text{TRANSF. FUNCTION FOR PRACTICAL DIFFERENTIATOR}} Y(z) \qquad = \frac{N(z)}{D(z)}$$



$$y' = \text{FILTER}(\underbrace{[\ldots]}_{\text{COEFFS. } N(z)}, \underbrace{[\ldots]}_{\text{COEFFS. } D(z)}, y)$$

$D(z) \rightarrow [1 \; -0.9]$

$\lambda = 0.9$

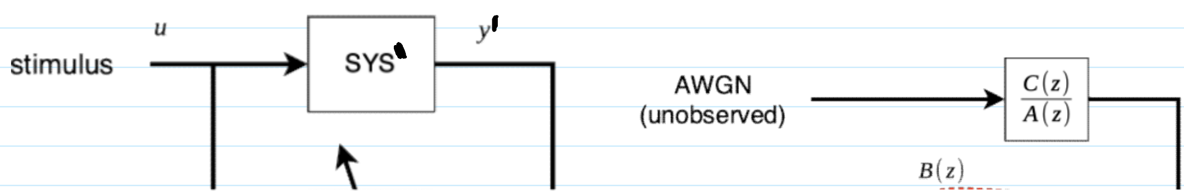
$$\frac{N(z)}{D(z)} = \frac{0.1(1-z^{-1})}{1-0.9z^{-1}}$$

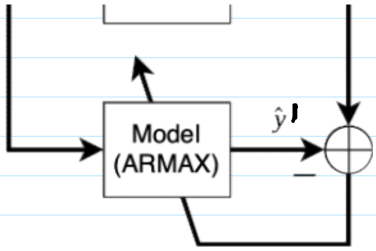
$N(z) \rightarrow 0.1*[1 \; -1]$

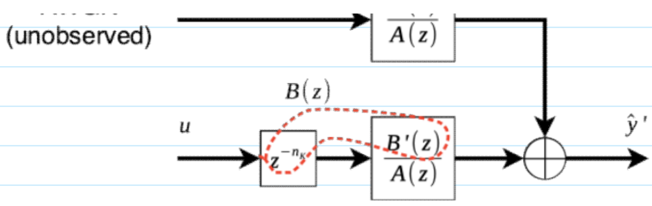
Burn-in + detrend

- Discard (mechanical) transient = delete ~10s of data from acquisition
  - fs = 100; y'(1:10*fs) = []; u(1:10*fs) = []

4. Identification







≈4 ≈1

z = [y' u];
M = armax(z,[nA nB nC nK])

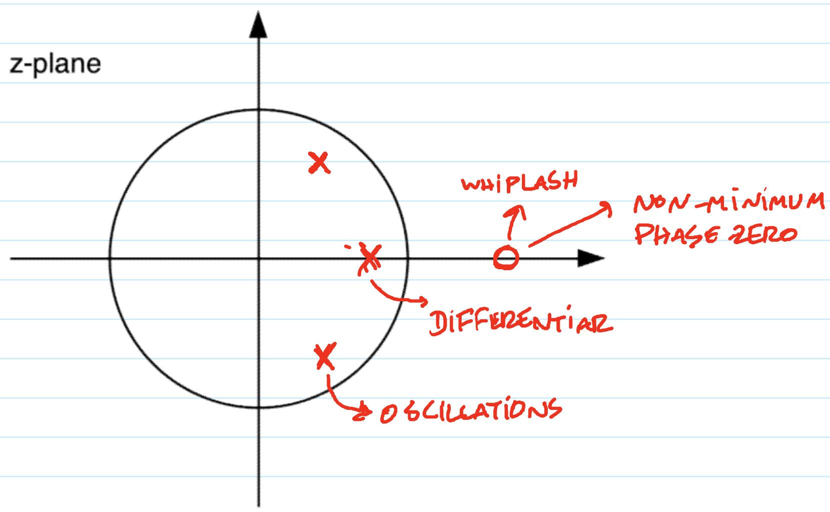
<4

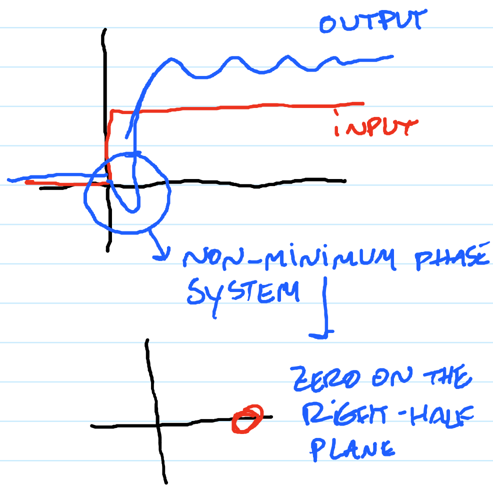
$$n_A: \ A(z) = 1 + a_1 z^{-1} + \dots + a_{n_A} z^{-n_A}$$

$$n_B, n_K: \ B(z) = z^{-n_K}\underbrace{\left(b'_1 + \dots + b'_{n_B} z^{-(n_B-1)}\right)}_{B'(z)}$$

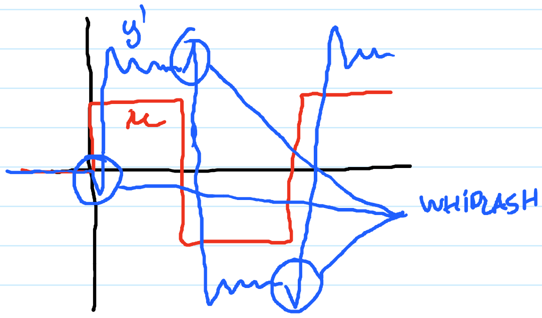
$$n_C: \ C(z) = 1 + c_1 z^{-1} + \dots + c_{n_C} z^{-n_C}$$

Desirable features for the pole-zero diagram of the identified transfer function



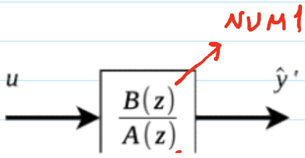


The whiplash effect

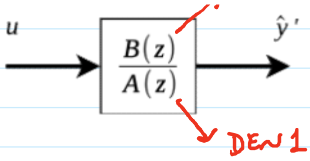


z1 = [y1' u1]; (training set)
z2 = [y2' u2]; (test set)
M = armax(z1, [nA nB nC nK]);
(see Fit parameter inside M object: 0% -> bad 100% - excellent)
[~, fit] = compare(z2, M);

5. Check results

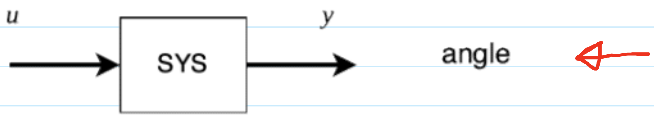


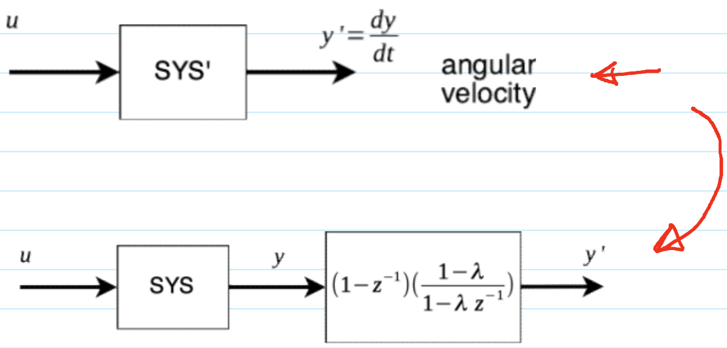
5. Check results



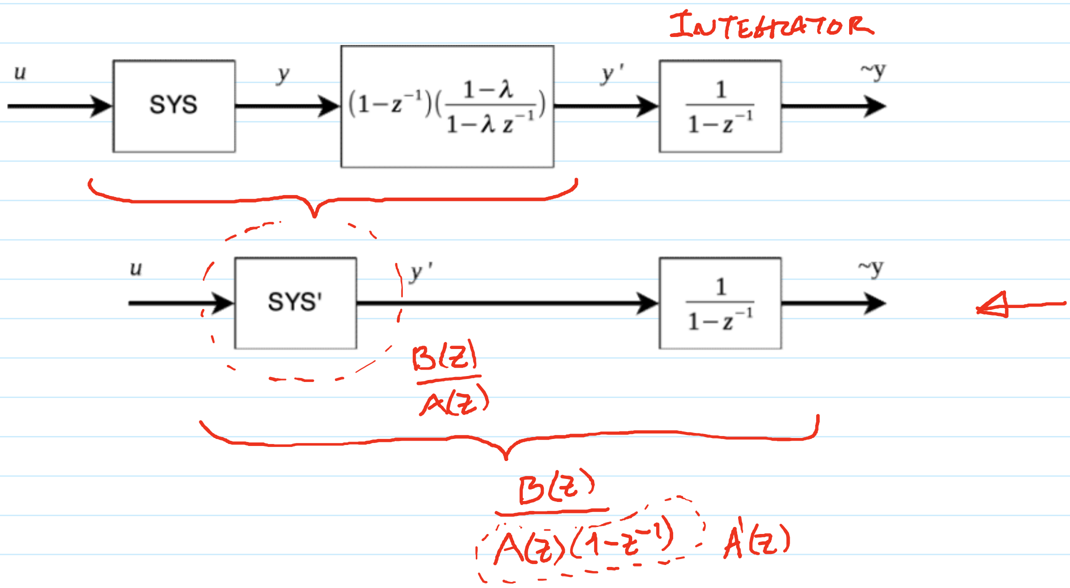

```
[den1, num1] = polydata(M);

y^' = idsim(u, M);
or...
y^' = filter(num1, den1, u);

zplane(num1, den1); (build pole-zero diagram)
```

6. Postprocessing





Undoing the effect of the differentiator




```
den = conv(den1, [1 -1])
```