RECALL:

DETERMINISTIC MODEL $\sum_{k=1}^{N} \left(M(k; U, \underbrace{x_{INIT}, p}_{\theta}) - y(k) \right)^2$

$\begin{cases} OE \longrightarrow \text{NONLIN LS} \\ OE + FIR \longrightarrow \text{LLS} \end{cases}$

$\{$ E.E., ARX $\longrightarrow$ LLS
(LIP)

$y(k) = \underbrace{h(\ldots)}_{\text{DET PART}} + \varepsilon(k)$

FIR IS SPECIAL CASE
OF ARX
OF OE

$$\boxed{y(k) = -\sum a_i y(k) + \sum b_i u(k) + \varepsilon(k)} \text{ (ARX)}$$

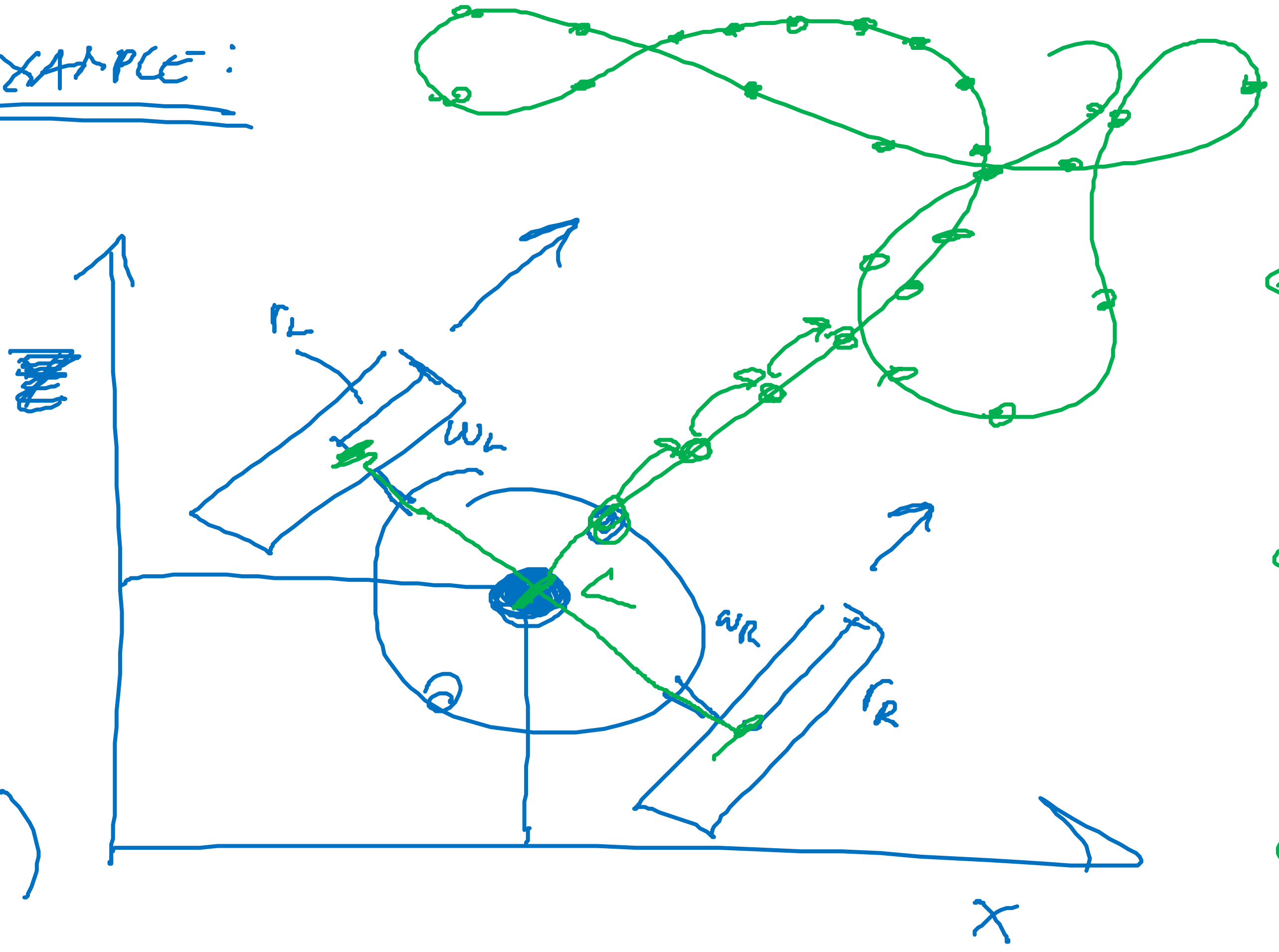
PEM ARX_PEM

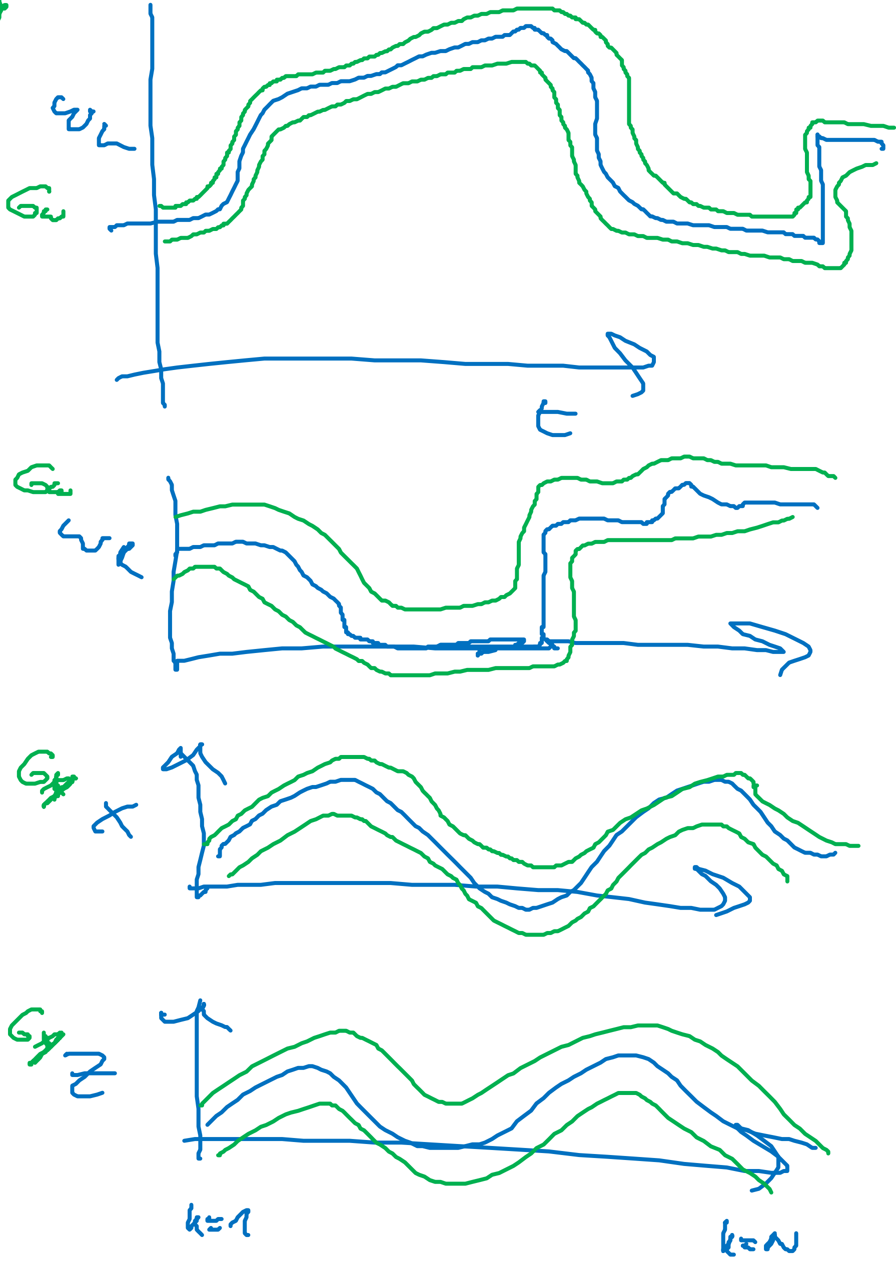
$\boxed{\text{IOE}}$

$$\boxed{\arg\min_{\varepsilon_N^u, x_{INIT}, p} \sum_{k=1}^{N} \frac{\left(M(k; U + \varepsilon_N^u, x_{INIT}, p) - y(k)\right)^2}{\sigma_y^2} + \frac{(\varepsilon_u(k))^2}{\sigma_u^2}}$$

OE IS SPECIAL CASE OF IOE WITH $\sigma_u^2 = 0$

LONG EXAMPLE:





INPUTS

$$u = \begin{pmatrix} \omega_L \\ \omega_R \end{pmatrix}$$

OUTPUTS

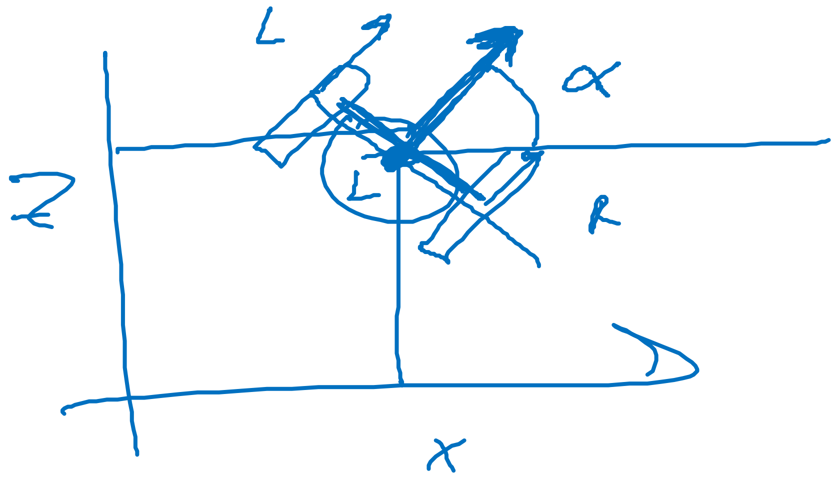
$$y = \begin{pmatrix} x \\ z \end{pmatrix}$$

SAMPLING TIME Δt

UNKNOWN PARS:

$$p = \begin{pmatrix} r_L \\ r_R \\ L \end{pmatrix}$$

TASK: GIVEN u_N, y_N, FIND p



$x_{INIT} = \begin{pmatrix} x \\ z \\ \alpha \end{pmatrix}$ AT $t=0$

α "STATE"

DERIVE CONT. TIME MODEL

GENERAL: $\boxed{\dot{x} = f(x,u)}$ x STATE
$\boxed{y = g(x,u)}$ u INPUT
y OUTPUT

$\omega_L \cdot r_L$ GROUND SPEED ON LEFT SIDE (ASSUME: NO SLIP)

$\omega_R \cdot r_R$ " RIGHT "

$$\frac{d}{dt}\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \cdot \frac{(\omega_L \cdot r_L + \omega_R \cdot r_R)}{2}$$

$$\frac{d}{dt}\alpha = -\frac{(\omega_L \cdot r_L - \omega_R r_R)}{L}$$

(ASSUME L GIVEN r_L, r_R, L)

HOW TO GENERATE $\boxed{M(u; U, x_{INIT}, p)}$?

INTERMEDIATE STEP: HOW TO GO FROM CONT. TIME TO DISCRETE TIME?

FROM CONT. TO DISCR. TIME

$$\boxed{\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}}$$

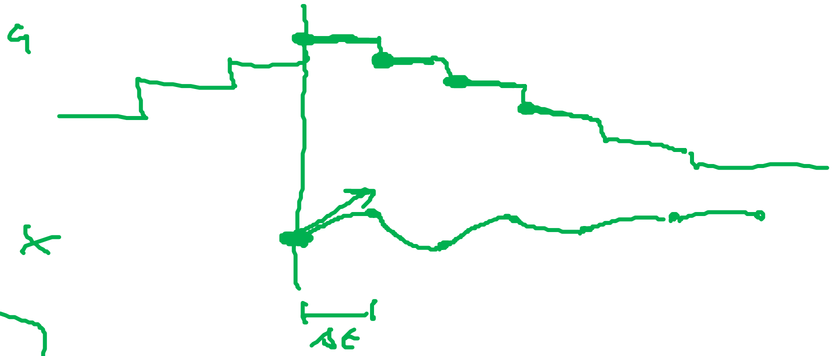
$$\boxed{x = \begin{pmatrix} x \\ z \\ \alpha \end{pmatrix}} \qquad u = \begin{pmatrix} \omega_L \\ \omega_R \end{pmatrix}$$

$$y = \begin{pmatrix} x \\ z \end{pmatrix}$$

$$\boxed{f = \begin{bmatrix} \cos\alpha \cdot \frac{1}{2}(\omega_L r_L + \omega_R r_R) \\ \sin\alpha \cdot \quad '' \\ -\frac{1}{L}(\omega_L r_L - \omega_R \cdot r_R) \end{bmatrix}}$$

SOME ASSUMPTIONS / APPROXIMATIONS:

- CONSTANT u ON SAMPLING INTERVALS



- NUMERICAL INTEGRATION

  E.G. EULER INTEGRATION

$$\boxed{x(t+\Delta t) = x(t) + \Delta t \cdot f(x,u)}$$

$$\boxed{\frac{x(t+\Delta t) - x(t)}{\Delta t} = f(x,u)} \simeq \boxed{\dot{x} = f(x,u)}$$

MORE ACCURATE:
- MULTIPLE SMALLER STEPS
- HIGHER ORDER INTEGRATOR
  E.G. RUNGE-KUTTA (RK4)
  MORE EFFICIENT IN PRACTICE

ODE45 (IMPLEMENTS ADAPTIVE RK METHOD)

EXAMPLE

TAKE $\boxed{\begin{matrix} x_k & \omega_{L,k} \\ z_k & \omega_{R,k} \\ \alpha_k & \end{matrix}}$ COMPUTE $\boxed{\begin{matrix} x_{k+1} \\ z_{k+1} \\ \alpha_{k+1} \end{matrix}}$

$x_k \equiv x(k)$ ETC.

Δt FIXED

for $k=1,\dots,n$

$$\boxed{\begin{aligned} x_{k+1} &= x_k + \Delta t \cdot \cos \alpha_k \cdot \frac{1}{2} (\underline{\underline{\omega_{L,k}}} \cdot r_L + \underline{\underline{\omega_{R,k}}} \cdot r_R) \\ z_{k+1} &= \dots \\ \alpha_{k+1} &= \alpha_k + \Delta t \cdot \left(-\frac{1}{L}\right) \cdot (\underline{\underline{\omega_{L,k}}}\, r_L - \underline{\underline{\omega_{R,k}}} \cdot r_R) \end{aligned}}$$

end $\to M(k; \dots)$

$$\boxed{M(k; U, x_{INIT}, p) = \begin{pmatrix} x_k \\ z_k \end{pmatrix}}$$

$$\boxed{U = \begin{bmatrix} \omega_{L,1} \\ \omega_{R,1} \\ \omega_{L,2} \\ \omega_{R,2} \\ \vdots \end{bmatrix} \quad x_{INIT} = \begin{bmatrix} x_1 \\ z_1 \\ \alpha_1 \end{bmatrix}, \quad p = \begin{bmatrix} r_L \\ r_R \\ L \end{bmatrix}}$$

BETTER:
- MORE TIME STEPS
- RK4

ATTENTION: FOR STIFF MODELS YOU NEED IMPLICIT INTEGRATORS.

FORMULATE IOE - ML PROBLEM

$$\hat{\Theta}_{ML} = \arg\min_{\tilde{U}, x_{INIT}, p} \sum \frac{1}{G_y^2} \left\| \left(M(k; \cancel{U+E_N}\ \tilde{U}, x_{INIT}, p) \right) - \begin{pmatrix} x_k \\ z_k \end{pmatrix} \right\|_2^2 + \frac{1}{G_W^2} \left\| \begin{pmatrix} \tilde{\omega}_{L,k} \\ \tilde{\omega}_{R,k} \end{pmatrix} - \begin{pmatrix} \omega_{L,k} \\ \omega_{R,k} \end{pmatrix} \right\|_2^2$$

$$\Theta = \begin{bmatrix} \tilde{U} \\ x_{INIT} \\ p \end{bmatrix} = \begin{bmatrix} \tilde{\omega}_{L,1} \\ \tilde{\omega}_{R,1} \\ \vdots \\ x_1^{INIT} \\ z_1^{INIT} \\ \varphi_1^{INIT} \\ r_L \\ \vdots \end{bmatrix}$$

$$\tilde{U} = \begin{pmatrix} \vdots \\ \tilde{\omega}_{L,k} \\ \tilde{\omega}_{R,k} \\ \vdots \end{pmatrix} = U + E_N^U$$

USE "Lsqnonlin" IN MATLAB

IT SOLVES

$$\boxed{\min_{\Theta} \frac{1}{2} \|F(\Theta)\|_2^2}$$

WITH $F: \mathbb{R}^{n_\Theta} \to \mathbb{R}^{n_F}$

$$\boxed{n_\Theta = 2 \cdot N + 3 + 3}$$

$$\boxed{n_F = 4 \cdot N}$$

$$F(\Theta) = \begin{bmatrix} \left(\frac{\tilde{\omega}_{L,1} - \omega_{L,1}}{G_W} \right) \\ \left(\frac{\tilde{\omega}_{R,1} - \omega_{R,1}}{G_W} \right) \\ \left(\frac{M^x(1; \tilde{U}, x_{INIT}, p) - x_1}{G_y} \right) \\ \left(\frac{M^z(1; \tilde{U}, x_{INIT}, p) - z_1}{G_y} \right) \\ \vdots \end{bmatrix}$$

$M^x(1, \ldots) = x_1^{INIT}$

$M^x(2; \quad) = \ldots$

$M^x(3; \ldots) = \ldots$

FIRST PRINCIPLES MODELS

"PHYSICAL"

- NEED LOTS OF KNOWLEDGE
- GIVE INSIGHT,
- CAN BE GENERALIZED,
- CAN DO PREDICTIONS (LONG)

⋮

"WHITE BOX MODEL"

"GREY BOX MODELS"

HAVE BLACK BOXES IN A WHITE BOX...

BLACK BOX MODELS

(MACHINE LEARNING)

- NEED NO PHYSICAL KNOWLEDGE
- FAST TO FORMULATE
- CAN DO (SHORT TERM) PREDICTIONS
- CAN CHOOSE "NICE" MODEL CLASS, E.G. LIP

- FIR (SPECIFY n_b)
- ARX (SPEZIFY n_b, n_a, n_{DELAY})
- (NEURAL NETWORK)
- (SUPPORT VECTOR MACHINE)

WIDELY USED IN CONTROL ENGINEERING, LTI SYSTEMS (FIR, ARX, ...)

← ⊖ →