POLIMI
DATA SCIENTISTS

MIDA 2

Course Notes

Edited by:

Matteo Sacco

These notes have been made thanks to the effort of Polimi Data Scientists staff.

Are you interested in Data Science activities?

Follow PoliMi Data Scientists on Facebook!

Polimi Data Scientist is a community of students and Alumni of Politecnico di Milano.

We organize events and activities related to Artificial Intelligence and Machine Learning, our aim is to create a strong and passionate community about Data Science at Politecnico di Milano.

Do you want to learn more?
Visit our website and join our Telegram Group! !

MIDA 2

1. Black Box non-parametric systems identification of I/O systems using state space models

Representations

1. State Space Representation

$F_{nxn}(state\ matrix)$ $G_{nx1}(input\ matrix)$
$H_{1xn}(output\ matrix)$ $D_{1x1}(I/O\ matrix)$

D = 0 for **strictly proper systems**, where the output only depends on the state.
Output : Input
State representation is not unique

$F' = TFT^{-1}$ $G' = TG$
$H' = HT^{-1}$ $D' = D$

2. Transfer Function representation (I/O representation)

$$y(t) = \boldsymbol{W}(\boldsymbol{z})\, u(t) = \frac{\boldsymbol{B}(\boldsymbol{z})}{\boldsymbol{A}(\boldsymbol{z})}\boldsymbol{z}^{-\boldsymbol{k}}\, u(t) = \frac{b_0 + b_1 z^{-1} + \cdots + b_p z^{-p}}{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}} z^{-k}$$

n = order of the system

3. Impulse Response representation

It can be proven that I/O relation from u(t) and y(t) can be written as

$$y(t) = \sum_{k=0}^{+\infty} \omega(k) u(t-k)$$

Where $\omega(k)$ is the I.R of the system and *y(t)* is the convolution of the I.R. with the input signal.

SS -> TF

$$y(t) = \boldsymbol{H}(\boldsymbol{ZI} - \boldsymbol{F})^{-1}\boldsymbol{G}\, u(t)$$

TF -> SS

SS is not unique.
Control realization technique.

$$W(z) = \frac{b_0 z^{n-1} + \cdots + b_{n-1}}{z^n + a_1 z^{n-1} + \cdots + a_n}$$

Assumptions:

- Monic DEN (a_0 = 1)
- Strictly proper system (b_{-1} = 0)

Formulas:

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & -a_1 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \begin{matrix} H = [b_{n-1} & b_{n-z} & \cdots & b_0] \\ D = 0 \end{matrix}$$

TF -> IR

Infinite long division of NUM and DEN of W(z).

IR -> TF

Trough the Z-transform

$$W(z) = \sum_{t=0}^{+\infty} \omega(t)\, z^{-t}$$

We would need **infinite points** of I.R. and a **noise free** I.R.

SS -> IR

$$\omega(0) = 0, w(n) = HF^{n-1}\square\ \square(\square - \square)$$

4SID (Subspace based State Space Systems Identification)

4SID starts with the measurement of the system output in the "impulse experiment".

$$\textbf{Observability}: Rank(O) = Rank\left(\begin{bmatrix} H \\ \vdots \\ HF^{n-1} \end{bmatrix}\right) = n$$

$$\textbf{Controllability}: Rank(R) = Rank([G \quad \cdots \quad F^{n-1}G]) = n$$

$$\textbf{Hankel Matrix}: H_n = \begin{bmatrix} \omega(1) & \cdots & \omega(n) \\ \vdots & \ddots & \vdots \\ \omega(n) & \cdots & \omega(2n-1) \end{bmatrix} = \begin{bmatrix} HG & \cdots & HF^{n-1}G \\ \vdots & \ddots & \vdots \\ HF^{n-1}G & \cdots & HF^{2n-1}G \end{bmatrix} = O \cdot R$$

1. Noise Free

Step 1: Build the biggest full-Rank Hankel Matrix, H_n

Step 2: Take H_{n+1} and factorize it into two rectangular matrices $(n+1 \text{ x } n)(n \text{ x } n+1)$

Which are O_{n+1} and R_{n+1}.

$$H_{n+1} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,n+1} \\ \vdots & \ddots & \vdots \\ h_{n,1} & \cdots & h_{n,n+1} \\ h_{n+1,1} & \cdots & h_{n+1,n+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \\ ? & ? & ? \end{bmatrix} \begin{bmatrix} h_{1,1} & \cdots & h_{1,n+1} \\ \vdots & \ddots & \vdots \\ h_{n,1} & \cdots & h_{n,n+1} \end{bmatrix} = \underset{(n+1)\text{x}\, n}{O_{n+1}} \cdot \underset{n\,\text{x}(n+1)}{R_{n+1}}$$

Step 3: $O_1 = O_{n+1}(1{:}n, :), O_2 = O_{n+1}(2{:}n+1, :)$

$\hat{F} = O_1^{-1} \cdot O_2$ $\qquad$ $\hat{G} = R_{n+1}(:,1)$

$\hat{H} = O_{n+1}(1, :)$ $\qquad$ $\hat{D} = \omega(0)$

This method remained unused until a new tool was developed **S**ingular **V**alue **D**ecomposition: a technique for *Data compression* and *Optimal separation of signal from noise*.

2. With Noise

We get the samples, n = 100 ÷ 1000 = q + d – 1

$$\widetilde{H}_{qd} = \begin{bmatrix} \widetilde{\omega}(1) & \cdots & \widetilde{\omega}(d) \\ \vdots & \ddots & \vdots \\ \widetilde{\omega}(q) & \cdots & \widetilde{\omega}(q+d-1) \end{bmatrix}$$

- If q ≈ d : the method has better accuracy
- If q < d : the method is computationally less intensive

Rule of thumb: 0.6d < q < d

Step 2: SVD of $\underset{qxd}{\widetilde{H}_{qd}} = \underset{qxq}{\widetilde{U}}\ \underset{qxd}{\widetilde{S}}\ \underset{dxd}{\widetilde{V}^T}$

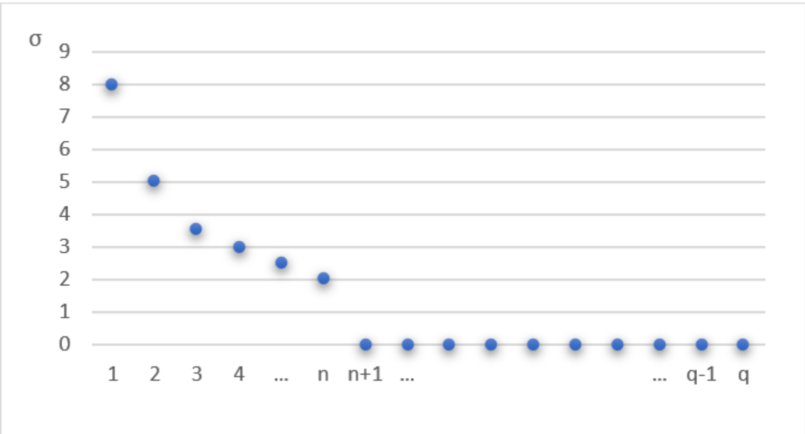
$\widetilde{U}, \widetilde{V}$: are unitary matrices (p15)

$\widetilde{S} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \sigma_q & 0 \end{bmatrix}$ $\sigma_1, \ldots, \sigma_n$ are the singular value of $\widetilde{H}_{qd}$

SVD is similar to a **diagonalization** of a rectangular matrix.

$$SV(M) = \sqrt{Eig(MM^T)} = \sqrt{Eig(M^TM)}$$

Step 3: Plot the singular values and "cut off" the 3 matrices.



In the ideal case there is a perfect separation between signal and noise.

With some empirical test we can select a good compromise between **complexity**, **precision** and **over fitting**, to choose an appropriate *n*.

Rank reduction:

$$\widetilde{H}_{qd} = \widetilde{U}\widetilde{S}\widetilde{V}^T = \widehat{U}\widehat{S}\widehat{V}^T + H_{res}$$

$$\hat{U} = \tilde{U}(:,1:n)$$
$$\hat{S} = \tilde{S}(1:n,1:n)$$
$$\hat{V} = \tilde{V}(1:n,:)$$

$$rank(\tilde{H}_{qd}) = q$$
$$\hat{H}_{qd} = \hat{U}\hat{S}\hat{V}^T$$
$$rank(\hat{H}_{qd}) = n \ll q$$

Step 4: Estimation of $\{\hat{F}, \hat{G}, \hat{H}\}$ using $\hat{H}_{qd}$

$$\hat{S}^{½} = \begin{bmatrix} \sqrt{\sigma_1} & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & \sqrt{\sigma_n} & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$
$$\hat{O} = \hat{U}\,\hat{S}^{½}$$

$$\hat{H}_{qd} = \hat{U}\hat{S}\hat{V}^T = \hat{U}\hat{S}^{½}\hat{S}^{½}\hat{V}^T$$
$$\hat{R} = \hat{S}^{½}\,\hat{V}^T$$
$$\hat{H}_{qd} = \hat{O}\,\hat{R}$$

$$\hat{H} = \hat{O}(1,:)$$
$$\hat{G} = \hat{R}(:,1)$$

$$\hat{O}_1 = \hat{O}(1:q-1,:)$$
$$\hat{O}_2 = \hat{O}(2:q,:)$$
$$\hat{O}_1\,\hat{F} = \hat{O}_2 \rightarrow \hat{F} = \cancel{\hat{O}_1^{-1}\hat{O}_2}$$
$$\rightarrow \hat{F} = \left(\hat{O}_1^T\hat{O}_1\right)^{-1}\hat{O}_1^T\hat{O}_2$$

Conclusion: we have estimated a model $\{\hat{F}, \hat{G}, \hat{H}\}$ in a **non-parametric**, **constructive** way.

Remark-1/5: something similar can be done for generic (non-impulsive) input

Remark-4/5: 4SID is a constructive method that can be implemented in a **fully automatic** way except for 2 steps that need **supervision**

- *q* and *d* selection (not critical)
- choice of *n* (can be automatic using **cross validation** method)

2. Parametric Black Box System Identification of I/O Systems (using a frequency domain approach)

Generic *parametric* identification method

1. **Collect Data**, experiment design and data pre-processing
1. Select a priori a **class/family** of **parametric models**
2. Select a priori a **performance index**
3. **Optimization step**: minimize J(θ) w.r.t. θ

General Idea:

- Make a set of ***single sinusoid*** excitation, ***single-tune*** experiments.
- From each experiment → estimate a single point of the **frequency response**
- **Fit** the estimated and modeled freq. response to obtain the optimal model

1. Experiment Design Step

In the **Experiment Design** step we first have to select a set of excitation frequencies $\{\omega_1, \omega_2, \dots, \omega_H\}$, usually evenly spaced and where ω_H must be select according to the **bandwidth** of the control system. $A_1 \sin(\omega_1 t), \dots, A_H \sin(\omega_H t)$

Remark: amplitudes (A_1, ..., A_N) can be constant or more frequently decrease as the frequency increases to comply with the power constraints on the input actuator.

If the system is LTI (linear time invariant) the freq. response theorem says that the response to a sinusoid is a sinusoid of the same frequency. However, in real application this is not the case because of:

- Noise on output measurement
- Noise on the system (not directly on output
- (small) non-linear effects (neglectable)

In **pre-processing** of **I/O data** we want to extract from $y_i(t)$ a **perfect sinusoid** of the right frequency ω_i.

The **model** of the output signal is

$$\hat{y}_i(t) = \boldsymbol{B_i} \sin(\omega_i t + \boldsymbol{\varphi_i})$$

$$\hat{y}_i(t) = \boldsymbol{a_i} \sin(\omega_i t) + \boldsymbol{b_i} \cos(\omega_i t)$$

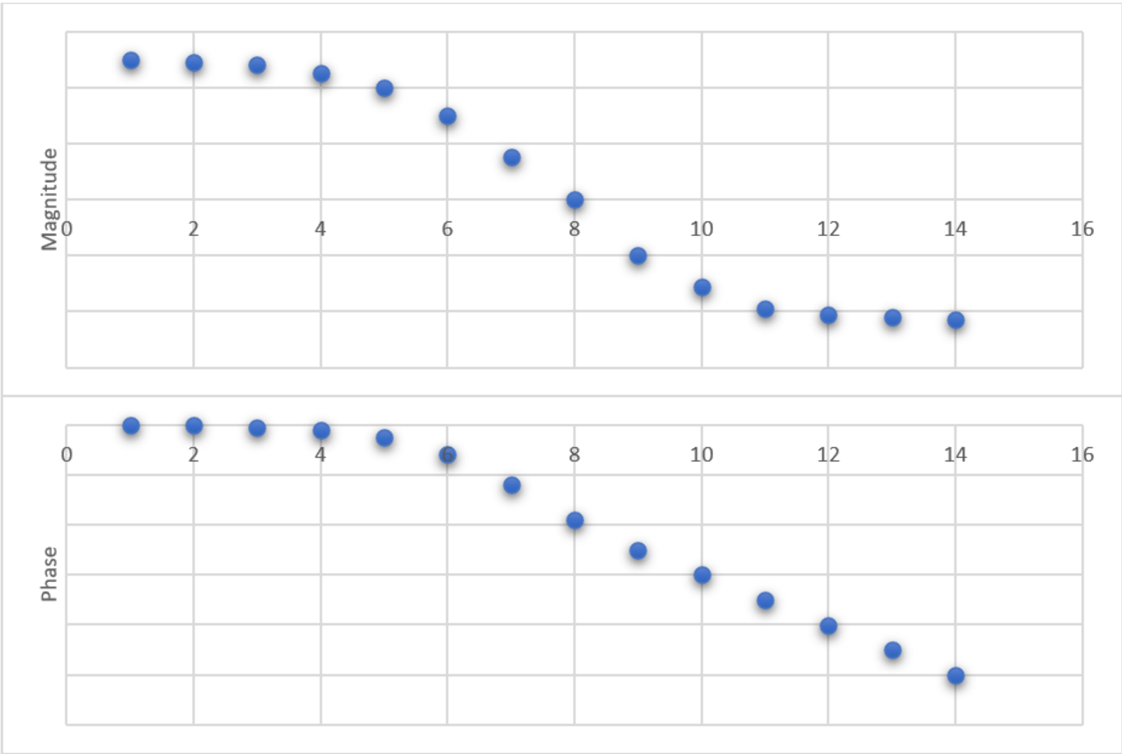
Parametric identification:

$$\{\hat{a}_i, \hat{b}_i\} = \underset{\{a_i, b_i\}}{\operatorname{argmin}} \{J_N(a_i, b_i)\}$$

$$J_N(a_i, b_i) = \overbrace{\frac{1}{N} \sum_{t=1}^{N} (\overbrace{\underbrace{y\_i(t)}_{\text{Measured noisy output}} \underbrace{- a_i \sin(\omega_1 t) - b_i \cos(\omega_i t)}_{\text{Modeled output}}}^{\text{Modeling error}})^2}^{\text{Sample variance of the modeling error}}$$

We obtain H solution: $\hat{a}_i, \hat{b}_i$, and convert them back to the polar form $\hat{B}_i, \hat{\varphi}_i$

We have obtained H complex numbers which are estimated H points of the frequency response of the transfer function W(z)



At the end of step 1 we have a frequency domain data set (H values) representing H estimated points of the freq. response of the system.

2. Selection of parametric model class (T.F.)

$$m(\theta): W(z;\theta) = \frac{b_0 + b_1 z^{-1} + \cdots + b_p z^p}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} z^{-1} \quad \theta = \begin{bmatrix} a_1 \\ \vdots \\ b_p \end{bmatrix}$$

Remark: as usual we have the problem of order selection (n, p) → use **cross validation** approach or **visual inspection of the fitting of Bode plots.**

We need a new performance index, *freq. domain* not *time domain*

$$\underset{\mathbb{R}^{n+p}(\theta)\to\mathbb{R}^+(J)}{J_H(\theta)} = \overbrace{\frac{1}{H}\sum_{t=1}^{H}(\underbrace{W(e^{j\omega_i},\theta)}_{\text{Modeled freq.resp.}} - \underbrace{\frac{\hat{B}_i}{A_i}e^{j\hat{\varphi}_i}}_{\text{Measured freq.resp}})^2}^{\text{Estimated error variance of freq. resp. fitting}}$$

Optimization

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\{J_H(\theta)\}$$

Usually J_H is a non-quadratic non-convex function → **iterative methods are needed.**

Remarks

Remark on Step1: Frequency bandwidth selection? Theoretically the best solution is H points distributed uniformly from 0 to Ω_N (Nyquist freq = $\Omega_s/2$).

In practice, it is better to concentrate the experiment effort in a smaller more focused bandwidth. **Rule of thumb**: 3 x ω_c (the "cut off" frequency of the control system).

Remark2: In some cases, between ω_1 and ω_H, we want to be more accurate in **system identification** on some frequencies ranges (typically the cut-off or the resonance freq.). In this case we can use different **weights** for different frequencies.

$$\tilde{J}_H(\theta) = \frac{1}{H}\sum_{t=1}^{H} \gamma_i (W(e^{j\omega_i}, \theta - \frac{\hat{B}_i}{A_i} e^{j\hat{\varphi}_i})^2$$

We could alternatively have denser ω spacing in the frequency region of interest.

Remark3: Sometimes the set of H independent/separated single-sinusoid experiments can be replaced by a **single experiment**, a long *single sine sweep experiment*: a slowly varying sinusoid with increasing frequency and decreasing amplitude from $\omega_i \rightarrow \omega_H$. We can cut a-posteriori the signal into H piecies back to the standart proedure or directly compute an estimation of $\widehat{W}\left(e^{j\omega}\right)$ as a ratio of the output input spectra:

$$W\left(e^{j\omega}\right) \approx \frac{\hat{\Gamma}_y(e^{j\omega})}{\hat{\Gamma}_u(e^{j\omega})}$$

We can fit the esitmated $\widehat{W}\left(e^{j\omega}\right)$ with the model freq. resp $W\left(e^{j\omega};\theta\right)$ in the performance index. This experiment is **quicker** but it usually has a **lower signal-to-noise ratio**.

Pros and **Cons** of **freq. domain parametric methods** vs **time domain** (ARMAX).

+ robust and reliable (due to the fact that we put a lot of energy in each sinusoid)
+ intuitive
+ consistent with control design methods
– more demanding
– no noise model is estimated

3. Kalman Filter (sw-sensing in feedback)

Kalman Filter is not a system identification technique; thus, we don't need recorder data.

With KF we can address the following problems

1. k-step ahead predictor of output $\hat{y}(t+k|t)$
   this problem was solved in MIDA1 with ARMAX
2. k-step ahead predictor of state $\hat{x}(t+k|t)$
   not solvable with ARMAX models
3. find the **filter** of state $\hat{x}(t|t)$
   $x(t)$ given $y(t), u(t), y(t-1), \ldots$
4. Gray box system identification
   (chapter 5…)

SW-sensing

Let us consider a system with m inputs n states p outputs. The outputs could be seen as states which we can measure through sensors. The states are variable which are either impossible to measure or cost-prohibitive to measure. We want to develop a system which has *inputs* and *output* as **input** and *states* as **output**.

1. **Is sw-sending feasible for a certain state?** We must test the observability of the states.
2. **Quality of estimation error**.

Basic system

$$S: \begin{cases} x(t+1) = Fx(t) + Gu(t) + v_1(t) \leftarrow \textit{state equation} \\ y(t) \qquad = Hx(t) \qquad\quad + v_2(t) \leftarrow \textit{output equation} \end{cases}$$

$$|x| = |v_1| = n$$
$$|u| = m$$
$$|y| = |v_2| = p$$

Model/state noise:

$$\vec{v}_1(t) \sim WN(0, V_1)$$

1. $E[v_1(t)] = \vec{0}$
2. $E[v_1(t) \cdot v_1(t)^T] = V_1 \geq 0$
3. $E[v_1(t) \cdot v_1(t-\tau)] = 0 \ \forall t, \tau \neq 0$

Output/measurement noise:

$$\vec{v}_2(t) \sim WN(0, V_2)$$

1. $E[v_2(t)] = \vec{0}$
2. $E[v_2(t) \cdot v_2(t)^T] = V_2 > 0$
   (additional assumption)
3. $E[v_2(t) \cdot v_2(t-\tau)] = 0 \ \forall t, \tau \neq 0$

$$E[v_2(t) \cdot v_2(t-\tau)] = \underset{nxp}{V_{12}} = \begin{cases} 0 \text{ if } \tau \neq 0 \\ \text{can be} \neq 0 \text{ if } \tau = 0 \end{cases} \quad \text{(cross variance matrix)}$$

Since the system S is dynamic, we need to define the initial conditions:

$$E[x(1)] = x_0$$
$$E[(x(1)-x_0)(x(1)-x_0)^T] = \underset{nxn}{P_0} \geq 0$$

$$x(1) \perp v_1(t)$$
$$x(1) \perp v_2(t)$$

KF for the basic solution of the basic system

Now we present the **basic solution** → 1-step ahead prediction for the **basic system** (Gu(t) = 0).

$\hat{x}(t+1|t) = F\hat{x}(t|t-1) + k(t) \cdot e(t)$ state equation
$\hat{y}(t|t-1) = H\hat{x}(t|t-1)$ output equation

$e(t) = y(t) - \hat{y}(t|t-1)$ output prediction error
$k(t) = (FP(t)H^T + V_{12})(HP(t)H^T + V_2)^{-1}$ eq. of the gain of the KF

$P(t+1) = (F\,P(t)F^T + V_1) - (F\,P(t)H^T + V_{12})(H\,P(t)H^T + V_2)^{-1}(F\,P(t)H^T + V_{12})^T$
Difference Riccati equation D.R.E.

These equations must be completede with 2 initial conditions (since 2 eq. are dynamic):

State equation → $\hat{x}(1|0) = E[x(1)] = x_0$
D.R.E. → $P(1) = Var[x(1)] = P_0$

D.R.E. have a blockset structure

$State \rightarrow F\ P(t)\ F^T + V_1$
$Ouput \rightarrow H\ P(t)\ H^T + V_2$
$Mix \rightarrow F\ P(t)\ H^T + V_{12}$

GAIN: $k(t) = Mix \cdot Output^{-1}$
DRE:$P(t+1) = State - Mix \cdot Ouput^{-1} \cdot Mix^T$

Remark: Riccati equation
Riccati eq. is a special type of **non linear matrix difference equation**. Notice that **D.R.E.** is an **autonomous**, **non linear**, **discrete**, **multivariable system** described by a **nonlinear differene matrix eq.**.

Remark: Existance of D.R.E.
In order to guarantee the existance of D.R.E. $\forall t$ the only critical part is the inversion of the **output block**.

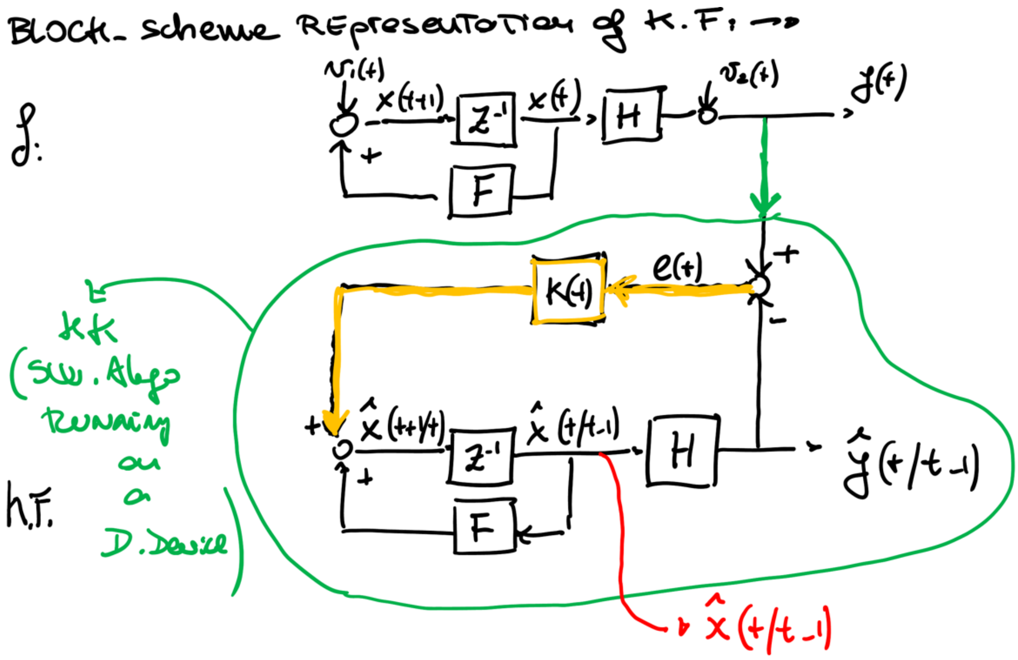
$$(H\ P(t)\ H^T + V_2)^{-1}$$

H ≥ 0 but not guaranteed to be in vertible
V_2 > 0 we previously made this assumption hence the sum is > 0.

Remark: Meaningn of P(t)
It can be proven that P(t) has a importan meaning.

$$P(t) = Var[x(t) - \hat{x}(t|t-1)]$$

P(t) is a sqaure covariance matrix ≥ 0, covarinace of the 1-step ahead prediction error of the state.



- Make a **simulated replica** of the system (without noises since v_1, v_2 are not measureable)
- Compare the **true** (measured) output with the **estimated/simulated** output $\hat{y}(t|t-1)$
- **Make correction on KF** main equation proportional (with k(t)) to the output error to keep KF as close as possible to *S*.
- **Extract** the state estimation $\hat{x}(t|t-1)$

KF is a feedback system where the feedback isn't used for control (as usual) but for estimation.

The structure of state observer existed before KFs. **KF found the optimabl gain k(t).**

k(t) is not a simple scalar gain (it cannot be tuned empirically) but It is a *nxp* maxtrix which cane be very large, thus difficult to be tuned.

Selection of gain matrix **k(t)** is very critical:

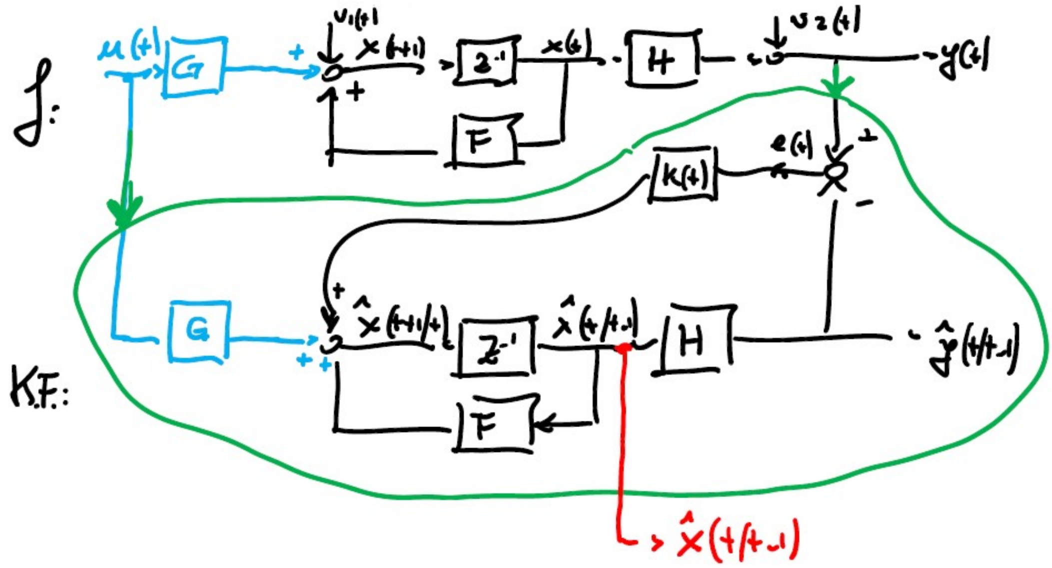
- If k(t) is **too small**: estimation is not optimal because we are **under-exploiting** the information in *y(t)*.
- If k(t) is **too big**: we riskt to be **over-exploiting** *y(t)* → noise aplification, risk of instability

Design of KG does not require a **training dataset** but a complete model of the system, usually obtained with whtie-box physical modeling of *S*.

Whereas, V_2 is easily built from sensors specification, V_1 (**model noise**) is much more difficult to design, it is the **most critical part** of the KF.

Extensions of Basic Problem of Basic System

Extension 1: Exogenous input



Notice that **k(t)** remains the same becasuse **P(t)** is the covariance of the estiamtion (prediction) error of x(t) and remains the same because **G u(t) does not introduce any additional noise or uncertainties** to the system, because **G u(t)** is a totally known deterministic signal.]

Extension 2: Multi-step prediction

Assume that $\hat{x}(t+1|t)$ is known (from basic solution) we can simply obtain a multi-step prediction as:

$$\hat{x}(t+2|t) = F\hat{x}(t+1|t)$$
$$\hat{x}(t+3|t) = F^2\hat{x}(t+1|t)$$
$$\vdots$$
$$\hat{x}(t+k|t) = F^{k-1}\hat{x}(t+1|t)$$
$$\hat{y}(t+k|t) = H\hat{x}(t+k|t)$$

Which is valid also for exgenous systems, *u(t)* is known up to *t*.

Extension 3: Filter

$$\hat{x}(t|t) = F^{-1}\hat{x}(t+1|t)$$

This formula is valid only if F is invertible. If it isn't the filter can be obtained with a specific filter formulation of K.F. **Filter Form**

$\hat{x}(t|t) = F\hat{x}(t|t-1) + k_0(t) \cdot e(t)$ state equation
$\hat{y}(t|t-1) = H\hat{x}(t|t-1)$ output equation

$e(t) = y(t) - \hat{y}(t|t-1)$ output prediction error
$k_0(t) = (\cancel{F}P(t)H^T + \cancel{V_{12}})(HP(t)H^T + V_2)^{-1}$ eq. of the gain of the KF
D.R.E. unchanged

These equations are valid under the restrictive assumption of $V_{12} = 0$

Extension 4: Time varying system

$$S: \begin{cases} x(t+1) = F(t)\, x(t) + G(t)\, u(t) + v_1(t) \\ \quad\;\; y(t) = H(t)\, x(t) \qquad\qquad\quad + v_2(t) \end{cases}$$

K.F. eq remain identical we just replace matrices with time varying matrices.

Asymptotic solution of KF

KF is LTV (linear time varying) since the gain is time varying. This causes 2 problems:

1. **Checking the stability of KF is very difficult.**
   LTI stability check: $x(t+1) = Fx(t) + Gu(t) \rightarrow EIG(F)$ inside uni-circle.
   In LTV even if $EIG(F)$ $\forall t$ inside uni-circle the system is not guaranteed to be asymptotically stable.
2. **Computational problem**
   k(t) must be computed at each sampling time

Because of these problems the asymptotic version of KF is preferred.

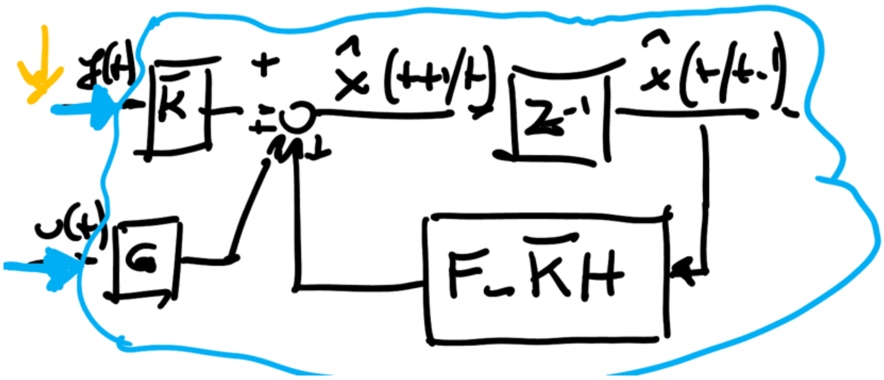
$$P(t) \xrightarrow{converges} \bar{P} \;\Rightarrow k(t) \xrightarrow{conv.} \bar{K}$$

$\bar{P}$: steady state value of *P(t)*
$\bar{K}$: stead state asymptotic value of *k(t)*

First let's analyze the asymptotic stability of KF, when $\bar{K}$ is used (KF is LTI). Let's assume that $\bar{K}$ exists.
Analysis of dynamic state of state equation of K.F.

$$\hat{x}(t+1|t) = F\hat{x}(t|t-1) + Gu(t) + \bar{K}e(t)$$
$$e(t) = y(t) - \hat{y}(t|t-1) = y(t) - H\hat{x}(t|t-1)$$
$$\hat{x}(t+1|t) = (F - \bar{K}H)\, \hat{x}(t|t-1) + Gu(t) + \bar{K}y(t)$$

New state matrix of KF



If $\bar{K}$ does exist the KF is **asymptotically stable** iff all **the $EIG(F - \bar{K}H)$ are strictly inside the uni-circle**.
KF can be asymptotically stable **even if the system is unstable**.

Since $\bar{K}$ exists if $\bar{P}$ exists.
Let's finde the quilibrium of dynamical autonomous system.

Continuous time	**Discrete time**
$\dot{x} = f(x)$	$x(t+1) = f(x(t))$
equilibrium $\rightarrow \dot{x} = 0$	equilibrium $\rightarrow x(t+1) = x(t)$
$\rightarrow f(\bar{x}) = 0$	$\rightarrow f(\bar{x}) = \bar{x}$

D.R.E. is an autnomous discrete time system.

$$\bar{P} = (F\,\bar{P}\,F^T + V_1) - (F\,\bar{P}\,H^T + V_{12})(H\,\bar{P}\,H^T + V_2)^{-1}(F\,\bar{P}\,H^T + V_{12})^T$$

This is a non-linear algebraic eq. called **Algebraic Riccati Eq** (A.R.E.).

3 Questions:

1. **Existance**: does A.R.E. have a semi definitive positive solution?
2. **Convergence**: if (1), does D.R.E. converge to $\bar{P}$?
   in principle $\bar{P}$ can be an equilibrium point for D.R.E. but not an attractor
3. **Stability**: if (1) and (2): is the corresponding $\bar{K}$ such that K.F is asymptotically stable?
   $EIG(F - \bar{K}H)$ are strictly inside the uni-circle

To answer to these questions we need 2 fundamental Theorems.

Asymptotic Th #1

#1-Th:

Assumptions:

- $V_{12} = 0$
- S is asym. stable → all *EIG(F)* strictly inside the uni-circle.

Then:

- A.R.E. has only one solution F ≥ 0
- D.R.E. converges to $\bar{P}$ $\forall P_0 \geq 0$
- The corresponding $\bar{K}$ is such that KF is asym. Stable

Asymptotic Th #2

For the 2nd Th we need **controllability** from noise.

$$x(t+1) = Fx(t) + \cancel{Gu(t)} + v_1(t) \quad v_1 \sim WN(0, V_1)$$

It is always possible to factorize $V_1 = \Gamma \cdot \Gamma^T$.

$$x(t+1) = Fx(t) + \Gamma\omega(t) \quad \omega(t) \sim WN(0, I)$$

Example

$$x(t+1) = \frac{1}{2}x(t) + v_1(t) \quad v \sim WN(0,4)$$
$$x(t+1) = \frac{1}{2}x(t) + 2\omega(t) \quad \omega \sim WN(0,1)$$

We can say that the state *x* is **controllable** from input noise $v_1(t)$ iff:

$$R = [\Gamma \quad F\,\Gamma \quad \cdots \quad F^{n-1}\Gamma] \text{ is of full Rank (n)}$$

#2-Th
Assumption:

- $V_{12} = 0$
- (F, H) is observable
- (F, Γ) is controllable

Then:

- A.R.E. has 1 definitive positive solution $\bar{P} > 0$
- D.R.E. converges to $\bar{P}\ \forall P_0 \geq 0$
- The corresponding $\bar{K}$ is such that KF is asym. stable.

These theorems are useful to avoid the convergence analysis of D.R.E..
Example page 47

Extended System

KF formula assumes v_1 and v_2 to be white noises, which is often a too demanding requirement.

$$S: \begin{cases} x(t+1) = a\,x(t) + \eta(t) \\ y(t) = b\,x(t) + v_2(t) \end{cases} \qquad \eta(t) = \frac{1}{1 - c\,z^{-1}} e(t) \qquad \begin{matrix} e(t) \sim WN(0,1) \\ e \perp v_2 \end{matrix}$$

$$v_2(t) \sim WN(0,1)$$

We cannot apply KF formula, we proceed as follows:

$$\begin{matrix} \eta(t) = c\,\eta(t-1) + e(t) \\ \eta(t+1) = c\,\eta(t) + e(t+1) \\ \eta(t+1) = c\,\eta(t) + v(t) \end{matrix} \qquad \begin{matrix} v(t) = e(t+1) \\ v \sim WN(0,1) \\ v \perp v_2 \end{matrix}$$

Extension of the state vector

$$\begin{matrix} x(t) \to x_1(t) \\ \eta(t) \to x_2(t) \end{matrix} \qquad x = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

$$S: \begin{cases} x_1(t+1) = a\,x_1(t) + x_2(t) \\ x_2(t+1) = c\,x_2(t) + v(2) \\ y(t) = b\,x_1(t) + v_2(t) \end{cases}$$

Now we can apply KF formulas to the extended system

Extension 5: Extension of KF to non-linear systems

$$S: \begin{cases} x(t+1) = f\big(x(t), u(t)\big) + v_1(t) \\ y(t) = h\big(x(t)\big) + v_2(t) \end{cases}$$

$f(\cdot)$, $h(\cdot)$ non-linear functions of $x(t)$ and $u(t) \in C^{(1)}$ or higher.
The block scheme is unchanged (see Extension 1 without G matrix). We have to solve the Gain Block.

We have 2 solutions:

1. **gain is a non-linear function** $k(e(t))$
2. **gain is a linear time-varying (LTV) function**: Extended KF

EKF

We can re-use FK formulas with small adjustments, the idea is to make a time varying linear local approximation of a non-linear time invariant system. So, the only thing we change to the KF formulas are F and H, which are now time varying matrices computed as follow:

$$F(t) = \frac{\partial f(x(t), u(t))}{\partial x(t)}\Big|_{x(t)=\hat{x}(t|t-1)} \qquad H(t) = \frac{\partial h(x(t))}{\partial x(t)}\Big|_{x(t)=\hat{x}(t|t-1)}$$

EKF is the time varying solution of KF where $F(t)$, $H(t)$ are local linear matrices computed (at each sampling time) around the last available prediction $\hat{x}(x|t-1)$.

Remarks on EKF

- it does not have a "steady state" asymptotic solution
- it has the same problem of LTV KF
  - it is almost impossible to have a theorical guarantee of EKF stability
    extensive empirical tests are needed
  - computational load
    $F(t)$, $H(t)$, $k(t)$ and $P(t)$ must be computed at *run-time*.

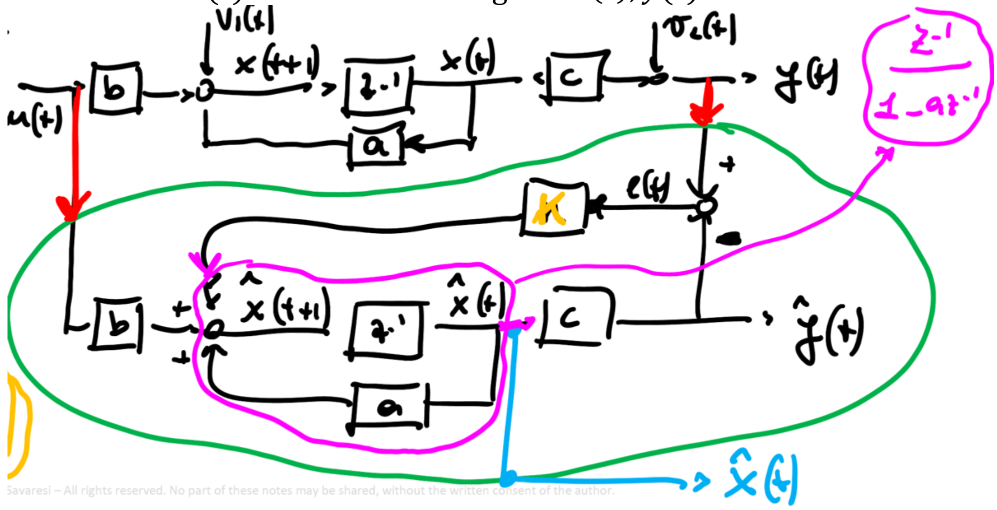
4. SW-sensing with Black Box Models
virtual sensing, variable estimator

In this chapter we will see black-box approaches with learning/training from data for SW-techniques.

Let us start with the case of LTI systems. We will first consider a simplified case: SISO-1 state.

$$S: \begin{cases} x(t+1) = a\,x(t) + b\,u(t) + v_1(t) \\ y(t) = c\,x(t) + v_2(t) \end{cases} \qquad \begin{matrix} v_1 \sim WN \\ v_2 \sim WN \end{matrix}$$

We want to estimate $\hat{x}(t)$ from measured signals $u(t), y(t)$


Savaresi – All rights reserved. No part of these notes may be shared, without the written consent of the author.

Let us find the relationship between $u(t) \rightarrow \hat{x}(t), y(t) \rightarrow \hat{x}(t)$

$$\hat{x}(t) = \frac{b \cdot \frac{z^{-1}}{1 - a\,z^{-1}}}{1 + K\,c\frac{z^{-1}}{1 - a\,z^{-1}}} u(t) + \frac{K \cdot \frac{z^{-1}}{1 - a\,z^{-1}}}{1 + K\,c\frac{z^{-1}}{1 - a\,z^{-1}}} y(t)$$

$$\hat{x}(t) = \boxed{\frac{b}{1 + (K\,c - a)z^{-1}}} u(t-1) + \boxed{\frac{K}{1 + (K\,c - a)z^{-1}}} y(t-1)$$

KF is a sophisticated way to build these transfer functions from a white-box model. Alternatively, we can adopt a black-box approach to estimate these 2 transfer functions from data.

In **supervised training approach** we need measurement of the state to be estimated, so we need a physical sensor on $x(t)$ only for design/training of SW-sensor.

$$\{u(1), \dots, u(n)\}, \{y(1), \dots, y(n)\}, \{x(1), \dots, x(n)\}$$

We now have 2 options:
1. Use **4SID** for direct **non parametric identification** of $u,y \rightarrow x$ dynamics.
2. Use a classic **parametric system identification** approach

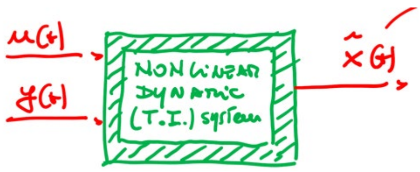
$$\hat{x}(t) = S_{ux}(z;\theta)\,u(t-1) + S_{yx}(z;\theta)\,y(t-1)$$

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\left[x(t) - \left(S_{ux}(z;\theta)\cdot u(t) + S_{yx}(z;\theta)\cdot y(t)\right)\right]^2$$

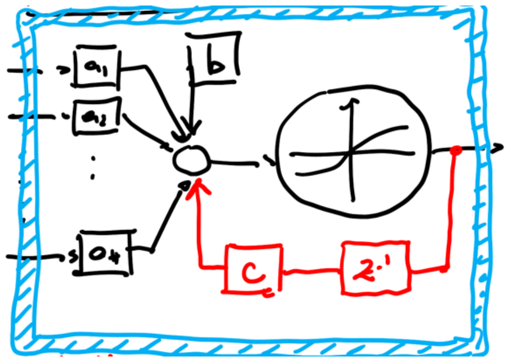
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\{J_N(\theta)\}$$

	KF	BB
Need of W.B. physical model of S	YES	NO
Need of training dataset	(NO)	YES
Interpretability of the result	YES	NO
Easy re-tuning of a similar system	YES	NO
Accuracy of the estimator	Good	Very Good
Can be used in case of un-measurable states	YES	NO

Non-linear system



When the system is non-linear the problems becomes more complicated. EKF uses the trick of time-varying linear gain $k(t)$, however, the obvious natural choice is the non-linear gain (static non-linear function).

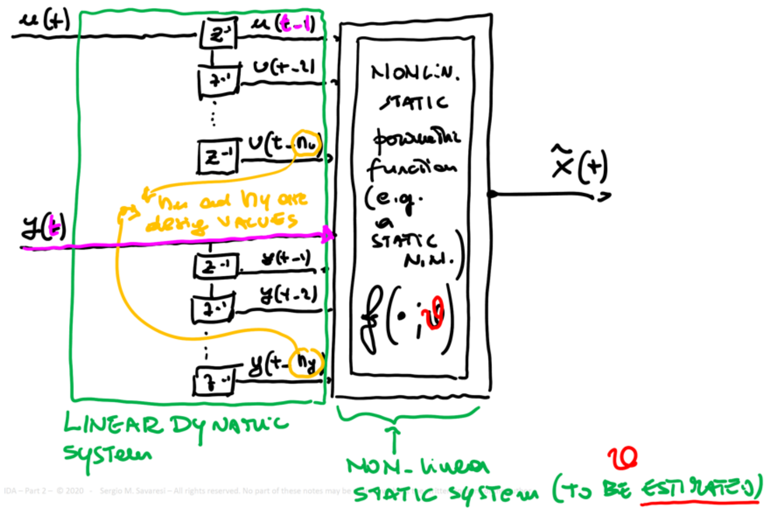


Architecture #1

It uses a **recurrent neural network RNN** with dynamic neurons. It is the most general approach, but it is practically never used → major issues of stability and convergence training.

Architecture #2

Finite Input Response Architecture (FIR): Split the system into a static NL system and linear dynamics.



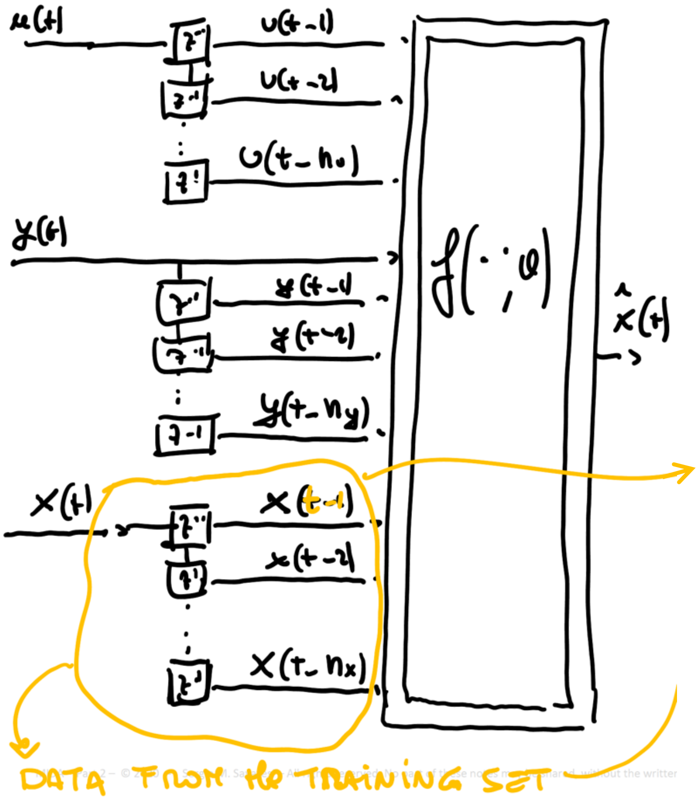
Remark:
In case of a MIMO with m inputs, p outputs, and n states. The estimator is the search of the optimal parameter vector θ for the function:

$$f(\cdot\,;\theta)\colon \mathbb{R}^{m \times n_u + p \times (n_y+1)} \rightarrow \mathbb{R}^n$$

Estimation (training) of this function $f(\cdot\,;\theta)$ is much simpler than the estimation of a Recursive Neural Network. Moreover, stability is guaranteed (the system is **F**inite **I**nput **R**esponse FIR).

Architecture #3

Infinite Input Response IIR: static non-linear function plus linear dynamic but this time I.I.R which means we have recursive input $x(t)$.



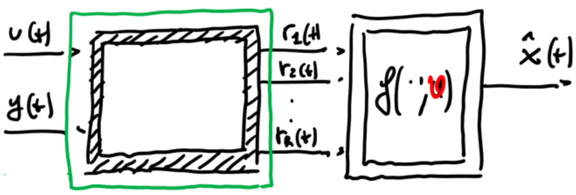
Advantage: n_u, n_y are smaller compared to Arch. #2

Disadvantage: this Arch. is not guaranteed to be stable by construction.

This part to the system is recursive, *in production* we feedback the delayed $\hat{x}(t)$ signal (problem of instability).

Architecture #4

Separation of system dynamics and a static non-linear system using regressor built from physical knowledge.



The system, which can be dynamic and non-linear, builds the regressors $r_1(t)$, ... $r_n(t)$, from physical signals $u(t)$, $y(t)$ using physical knowledge/understating of the system.

Conclusions

In case of black-box SW-sensing with nonlinear system the problem can be quite complex.
Using “brute force” approach (1 Dynamic Neural Network) is usually doomed to failure.
It is best to gain some insight of the system and build some “smart” regressor before black-box mapping.

5. Gray Box System System Identification

Gray Box System Identification using KF

We have a model, typically built as a white-box model using first principles:

$$S: \begin{cases} x(t+1) = f(x(t), u(t); \theta) + v_1(t) & \text{model noise} \\ y(t) = h(x(t); \theta) + v_2(t) & \text{output noise} \end{cases}$$

f and *h* are linear or non-linear functions depending on some unknown parameter θ (which has a physical meaning such as the mass, friction coefficient, etc.) our goal is to estimate θ.

KF solves this problem by transforming the unknown parameters in extended states → KF makes the simultaneous estimation of:

$\hat{x}(t|t)$ (classic KF problem)

$\hat{\theta}(t)$ (parameter identification problem)

State extension *trick*.

$$\begin{cases} x(t+1) = f(x(t), u(t); \theta) + v_1(t) \\ \theta(t+1) = \theta(t) + v_\theta(t) \\ y(t) = h(x(t); \theta) + v_2(t) \end{cases}$$

$$x_E = \begin{bmatrix} x(t) \\ \theta(t) \end{bmatrix}$$

Unknown parameters are transformed into unknown variables

The new equation is not a physical equation but a *fictitious* one.

$$\theta(t+1) = \theta(t) + v_\theta(t)$$

(core dynamics) this is the equation of something constant, in fact $\theta(t)$ is indeed a constant vector of parameters.

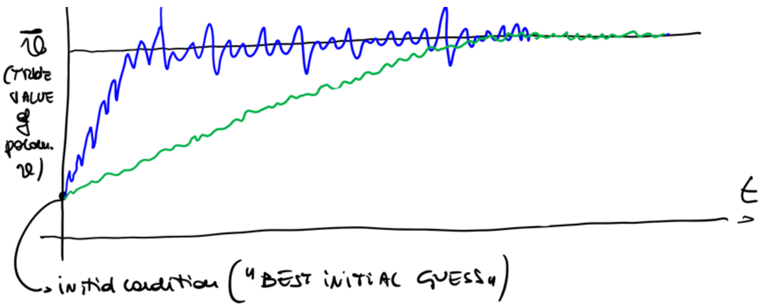
(fictitious noise) we need it to force KF to find the right value of θ. It there were no noise KF might stay fixed on the initial condition.

This equation is not an asymptotically stable system but a **s**imply **s**table **s**ystem.

The choice of V_θ is a design choice. We can usually assume that $v_\theta(t)$ is a set of independent *WN* with the same variance:

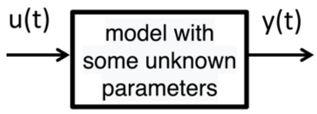
- $v_1 \perp v_\theta; v_2 \perp v_\theta$
- $\lambda_{1,\theta}^2 = \cdots = \lambda_{n_\theta,\theta}^2$

How the choice of λ_θ^2 parameters influences the estimation:



Large values of λ_θ^2 → fast convergence but noisy at steady state. k
Small values of λ_θ^2 → slow convergence, less noise at steady state.

Simulation Error Method (SEM)



We will now see a parametric identification approach based on Simulation Error method.

1. Collect data from an experiment: $\{\tilde{u}(1), \dots, \tilde{u}(N)\}, \{\tilde{y}(1), \dots, \tilde{y}(N)\}$
2. Define model structure: $y(t) = m(u(t); \bar{\theta}; \theta)$
   $\bar{\theta}$ set of known parameters
   θ set of unknown parameters (possibly with bounds)
3. Performance index definition SEM

$$J_N(\theta) = \overbrace{\frac{1}{N}\sum_{t=1}^{H}\Big(\underbrace{\tilde{y}(t)}_{\text{Measured output}} \underbrace{-m(\tilde{u}(t); \bar{\theta}; \theta)}_{\text{Measured freq.resp}} \Big)^2}^{\text{sample variance of the simulation error}}$$

4. Optimization: $\hat{\theta}_N = \underset{\theta}{\operatorname{argmin}}\{J_N(\theta)\}$

   Usually:
   - no analytic expression of $J_N(\theta)$ is available
   - each computation of $J_N(\theta)$ requires an entire **simulation of the model**
   - $J_N(\theta)$ is non quadratic and non-convex function so **iterative** and **randomized** optimization methods must be used

   These conditions make SEM computationally very demanding

PEM for black-box

We want to estimate from data this I/O model

$$y(t) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} u(t-1) \qquad \theta = \begin{bmatrix} a_1 \\ \vdots \\ b_1 \end{bmatrix}$$

Time domain

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_0 u(t-1) + b_1(t-2)$$

Using **P**rediction **E**rror **M**ethod (measured data)

$$\tilde{y}(t|t-1) = -a_1 \tilde{y}(t-1) - a_2 \tilde{y}(t-2) + b_0 \tilde{u}(t-2) + b_1 \tilde{u}(t-2)$$

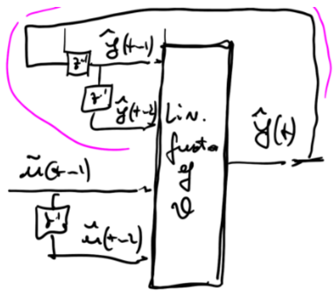
$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{\infty}\big(\tilde{y}(t) - \hat{y}(t|t-1;\theta)\big)^2 =$$

$$= \frac{1}{N}\sum_{t=1}^{\infty}[\tilde{y}(t) + a_1\tilde{y}(t-1) + a_2(t-2) + b_0 u(t-1) + b_1(t-2)]^2$$

This is a quadratic function of θ; thus the minimization is very simple.

PEM vs SEM

P.E.M.



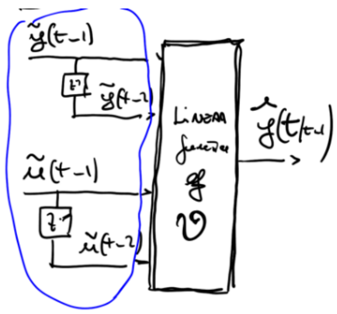
$$\hat{y}(t) = -a_1\hat{y}(t-1) - a_2\hat{y}(t-2) + b_0\tilde{u}(t-1) + b_1\tilde{u}(t-2)$$

Old values of simulated output
Measured values

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[\tilde{y}(t) - \hat{y}(t;\theta)]^2 =$$

$$= \frac{1}{N}\sum_{t=1}^{N}[\tilde{y}(t) + a_1\hat{y}(t-1) + a_2\hat{y}(t-2) - b_0\tilde{u}(t-1) - b_1\tilde{u}(t-2)]^2$$

Which is highly non-linear w.r.t. θ (non-quadratic, non-convex).



P.E.M

Disadvantages of P.E.M.:

1. P.E.M. is **much less robust to noise** and we must include a model of the noise in the **estimated model**.
   We need a ARMAX instead of ARX
   $$y(t) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} u(t-1) + \frac{1 + c_1 z^{-1} + \cdots + c_n z^{-n}}{1 + a_1 z^{-1} + a_2 z^{-2}} e(t)$$
   ARX would be linear in the parameter vector:
   $$y(t) = b_0 u(t-1) + b_1(t-2) - a_1 y(t-1) - a_2 y(t-2)$$
   but ARMAX isn't, therefore $J_N(\theta)$ is highly non-linear → same complexity as S.E.M.
2. P.E.M. is very **sensitive to sampling time choice**.
   If Δ is very small so is the difference between $\tilde{y}(t)$, $\tilde{y}(t-1)$. As a result the P.E.M. tends to provide this solution
   $$\theta = \begin{bmatrix} a_1 \to -1 \\ a_2 \to 0 \\ b_0 \to 0 \\ b_1 \to 0 \end{bmatrix} \to \tilde{y}(t) \approx \tilde{y}(t-1)$$
   This is due to the fact that the recursive part of the model is using past measures of the output instead of past values of the time model outputs.

Summary page 71

6. Minimum Variance Control (MVC)

design and analysis of feedback systems

Setup of the problem

Consider a generic ARMAX model

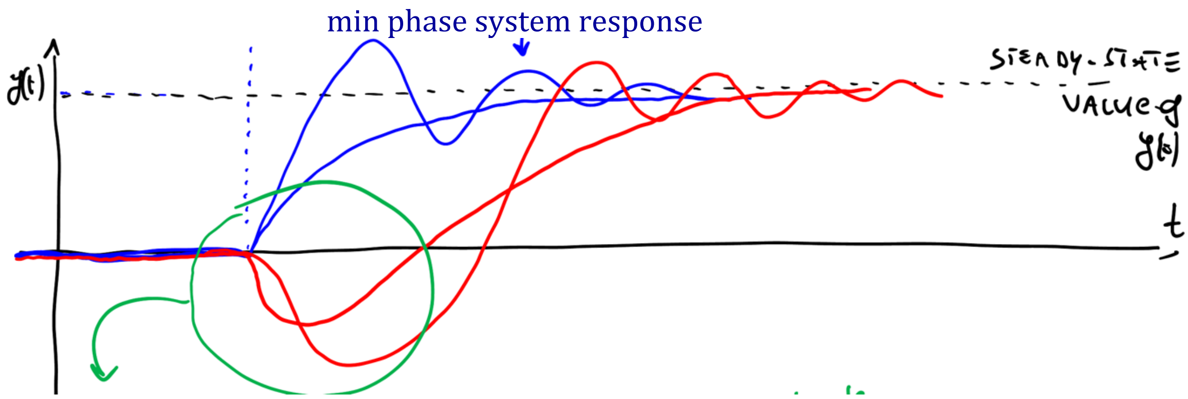
$$y(t) = \frac{B(z)}{A(z)}u(t-k) + \frac{C(z)}{A(z)}e(t)$$
$$e \sim WN(0, \lambda^2)$$

$$B(z) = b_0 + b_1 z^{-1} + \dots + b_p z^{-p}$$
$$A(z) = 1 + a_1 z^{-1} + \dots + a_m z^{-m}$$
$$C(z) = 1 + c_1 z^{-1} + \dots + c_n z^{-n}$$

Assumptions:

- $C(z)/A(z)$ is in canonical form
  - zero relative degree
  - $C(z)$, $A(z)$ are coprime
  - $C(z)$, $A(z)$ are monics (a_0, b_0 = 1)
  - $C(z)$, $A(z)$ are stable
- $b \neq 0$
  thus k is the actual delay of the system
- $B(z)/A(z)$ is "minimum phase"
  roots of $B(z)$ are strictly inside the uni-circle

Control design for non-phase systems is difficult ant it requires a special design technique

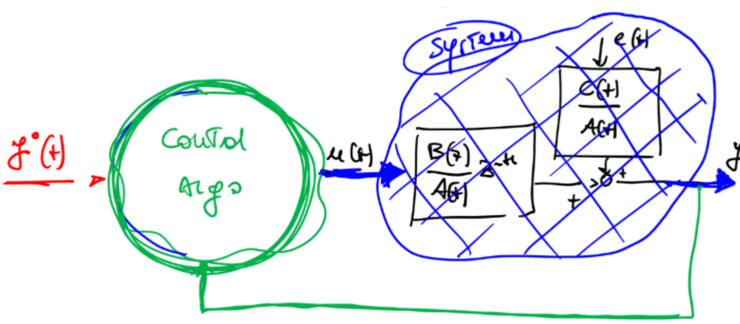


the response of non min phase systems (at the beginning the response moves away from the final value.

We wish to obtain the optimal tracking of the desired behavior of output

In a non-formal way M.V.C. tries to minimize the performance index

$$J = \mathrm{E}[(y(t) - y^o(t))^2]$$



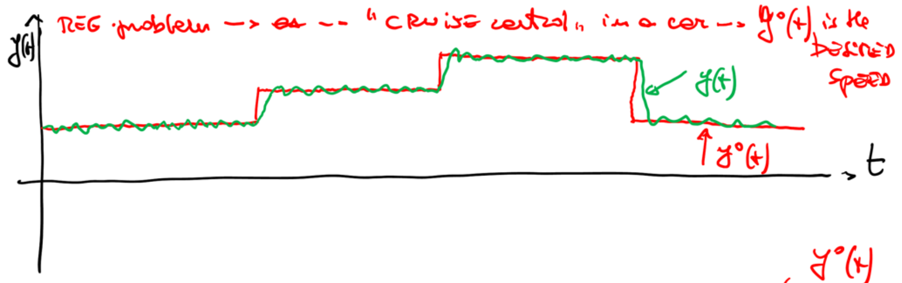
Variance of the tracking error

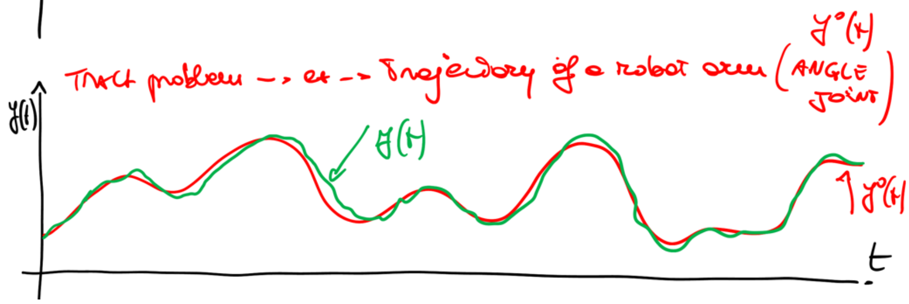
(Small) technical assumptions

- $y^o(t) \perp e(t)$
- $y^o(t)$ is known only up to time t
  being $y^o(t+k|t)$ totally unpredictable, the best prediction is simply $y^o(t)$.

Remark: there are 2 sub-classes of control problems

1) when $y^o(t)$ is **constant** or **stepwise** → *regulation problem*
2) when $y^o(t)$ is **varying** → *tracking problem*





Simplified Problem #1

$$S: y(t) = ay(t-1) + b_0 u(t-1) + b_1 u(t-2)$$
$$y(t) = \frac{b_0 + b_1 z^{-1}}{1 - a\, z^{-1}} u(t-1) + noise\ free$$

We assume $y^o(t) = \bar{y}^o$ (regulation problem)

$b_0 \neq 0$ roots($B(z)$) inside uni-cirlce

To design M.V.Controller we have to minimize

$$J = \mathrm{E}\left[\left(y(t) - y^o(t)\right)^2\right]$$

There is no noise for E[·] to remove, $y^o(t) = \bar{y}^o$, we plug in S expression.

$$J = (a\, y(t-1) + b_0 u(t-1) + b_1 u(t-2) - \bar{y}^o)^2$$

Time shift

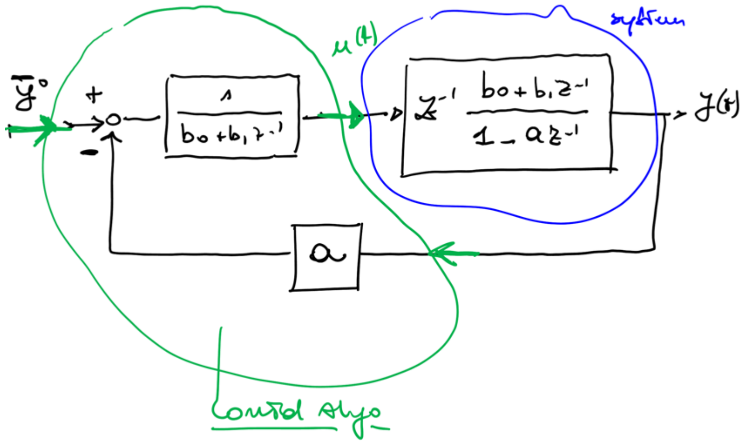
$$J = (a\, y(t) + b_0 u(t) + b_1 u(t) - \bar{y}^o)^2$$

Derivative w.r.t. $u(t)$

$$\frac{\partial J}{\partial u(t)} = 2(a\, y(t) + b_0 u(t) + b_1 u(t-1) - \bar{y}^o) \cdot b_0 = 0$$

(at time t: $y(t)$, $y(t\text{-}1)$, $y(t\text{-}2)$, $u(t\text{-}1)$, $u(t\text{-}2)$ are no longer variables but numbers)

$$u(t) = \left(\bar{y}^o - a\, y(t)\right) \cdot \frac{1}{b_o + b_1 z^{-1}}$$



Simplified Problem #2

$$S\colon y(t) = ay(t-1) + b_0 u(t-1) + b_1 u(t-2) + e(t)$$

Assumptions: $b_0 \neq 0$ roots$(B(z))$ inside uni-cirlce

$$y(t) = \underbrace{\hat{y}(t|t-1)}_{\substack{\text{prediction}\\ \text{of } y(t) \text{ at}\\ \text{time } t-1}} \quad \underbrace{+\varepsilon(t)}_{\substack{\text{corresponding}\\ \text{prediction}\\ \text{error}}}$$

Since $k = 1 \rightarrow \varepsilon(t) = e(t)$

$$J = \mathrm{E}\left[\left(y(t) - y^o(t)\right)^2\right] =$$
$$= \mathrm{E}\left[\left(\hat{y}(t|t-1) + e(t) - y^o(t)\right)^2\right] =$$
$$= \mathrm{E}\left[\left(\hat{y}(t|t-1) - y^o(t)\right)^2\right] + \mathrm{E}[e(t)^2] + \cancel{2\mathrm{E}[e(t) \cdot (\hat{y}(t|t-1) - y^o(t))]}$$

$e(t) \perp \hat{y}(t|t-1)$ by construction

$e(t) \perp y^o(t)$ by assumption

$\mathrm{E}[e(t)^2] = \lambda^2$

$$\underset{u(t)}{\operatorname{argmin}} \left\{\mathrm{E}\left[\left(\hat{y}(t|t-1) - y^o(t)\right)^2\right]\right\}$$

Now we must compute the 1-step predictor for S.

$$S\colon y(t) = \frac{b_0 + b_1 z^{-1}}{1 - a\, z^{-1}} u(t-1) + \frac{1}{1 - a\, z^{-1}} e(t)$$

ARMAX(1,0,1+1) = ARX(1,2)

ARMAX 1-step predictor

$$\hat{y}(t|t-1) = \frac{B(z)}{C(z)}u(t-1) + \frac{C(z)-A(z)}{C(z)}y(t)$$

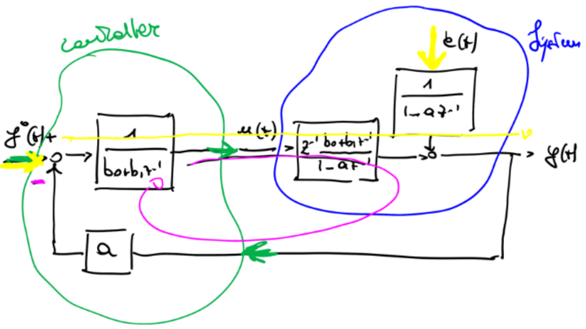
$$\hat{y}(t|t-1) = (b_0 + b_1 z^{-1})\, u(t-1) + a\, y(t-1)$$

Shift and impose: $\hat{y}(t+1|t) = y^o(t+1)$

$$y^o(t+1) = b_0 u(t) + b_1 u(t-1) + a\, y(t)$$

$$u(t) = (y^0(t) - a y(t)) \cdot \frac{1}{b_0 + b_1 z^{-1}} \qquad y^0(t+1) = y^0(t)$$

Let us analyze **stability** and **performance** of the obtained system.



Stability

To check the closed loop stability

1. compute the *loop function*:
   $L(z) = F_1(z) \cdot F_2(z)$
2. build the *characteristic polynomial*:
   $X(z) = L_N(z) + L_D(z)$
3. find the roots of $X(z)$ → closed loop system is asymptotically stable iff all the roots of $X(z)$ are inside the uni-circle.

$$L(z) = \frac{1}{b_0 + b_1 z^{-1}} \cdot \frac{z^{-1}(b_0 + b_1 z^{-1})}{1 - a\, z^{-1}} \cdot a$$

$$X(z) = a\, z^{-1}(b_0 + b_1 z^{-1}) + (1 - a\, z^{-1})(b_0 + b_1 z^{-1}) = b_0 + b_1 z^{-1} = B(z)$$

Thanks to *min-phase assumption* we know that roots($B(z)$) are inside uni-circle.

Performance

Since the system is LTI we can use the super position principle.

$$y(t) = F_{y^o,y}(z) y^o(t) + F_{e,y}(z) e(t)$$

$$F_{y^o,y}(z) = \frac{\frac{1}{b_0 + b_1 z^{-1}} \cdot \frac{z^{-1}(b_0 + b_1 z^{-1})}{1 - a_1 z^{-1}}}{1 + \frac{1}{b_0 + b_1 z^{-1}} z^{-1} \cdot \frac{b_0 + b_1 z^{-1}}{1 - a z^{-1}} a} = \cdots = z^{-1}$$

$$F_{e,y}(z) = \frac{\frac{1}{1 - a\, z^{-1}}}{1 + (loop\ function)} = \cdots = 1$$

Thus, the closed loop system has very simple closed-loop behavior, which is our optimal control, the best possible solution:

$$y(z) = y^o(t-1) + e(t)$$

General Solution

$$S: y(t) = \frac{B(z)}{A(z)} u(t-k) - \frac{C(z)}{A(z)} e(t)$$

Assumptions:

- $b_0 \neq 0$
- $B(z)$ has all roots strictly inside uni-circle (S is *min-phase*)
- $C(z)/A(z)$ is in canonical representation
- $y^o(t)$ is unpredictable → $y^o(t+k|t) = y^o(t)$
- $y^o(t) \perp e(t)$

We need to minimize the performance index

$$y(t) = \hat{y}(t|t-k) + \varepsilon(t)$$

$$J = \mathrm{E}\left[\left(y(t) - y^o(t)\right)^2\right] =$$

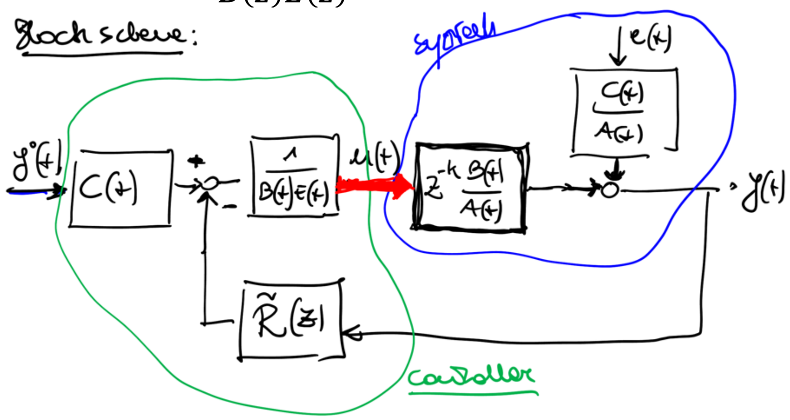
$$= \mathrm{E}\left[\left(\hat{y}(t|t-k) + \varepsilon(t) - y^o(t)\right)^2\right] =$$

$$= \mathrm{E}\left[\left(\hat{y}(t|t-k) - y^o(t)\right)^2\right] + \mathrm{E}[\varepsilon(t)^2] + \cancel{2\mathrm{E}\left[\varepsilon(t) \cdot \left(\hat{y}(t|t-k) - y^o(t)\right)\right]}$$

Since $\mathrm{E}[\varepsilon(t)^2]$ does not depend on $u(t)$ the optimal solution is: $\hat{y}(t|t-k) = y^o(t)$
We now use the ARMAX predictor, shifted k-step ahead.

$$\hat{y}(t+k|t) = \frac{B(z)E(z)}{C(z)} u(t) + \frac{\tilde{R}(z)}{C(z)} y(t) = y^o(t+k) \sim y^o(t)$$

$$\boldsymbol{u(t) = \frac{1}{B(z)E(z)} \left(C(z) y^o(t) - \tilde{R}(z) y(z)\right)}$$



Stability

$$L(z) = \frac{1}{B(z)E(z)} \cdot \frac{z^{-k}B(z)}{A(z)} \tilde{R}(z)$$

$$X(z) = z^{-k}B(z)\tilde{R}(z) + B(z)E(z)A(z)$$

$$= B(z)\left(z^{-k}\tilde{R}(z) + E(z)A(z)\right)$$

$$= B(z)C(z)$$

- roots($C(z)$) are stable since $C(z)$ is in canonical representation
- roots($B(z)$) are stable because of initial assumptions

Performance

$$y(t) = F_{y^o,y}(z) y^o(t) + F_{e,y}(z) e(t)$$

$$F_{y^o,y}(z) = \frac{C(z) \frac{1}{B(z)E(z)} \cdot \frac{z^{-k}B(z)}{A(z)}}{1 + \frac{1}{B(z)E(z)} \cdot \frac{z^{-k}B(z)}{A(z)} \tilde{R}(z)} = \cdots = z^{-k}$$

$$F_{e,y}(z) = \frac{C(z)/A(z)}{1 + (loop\ function)} = \cdots = E(z)$$

Thus, the closed loop system has very simple closed-loop behavior, which is our optimal control, y(t) exactly tracks $y^o(t)$ but with k-step delay and it is disturbed by noise (prediction error k-step ahead):

$$y(z) = y^o(t-k) + E(z) \cdot e(t)$$

Remark:
M.V.Controller pushes all the system poles into the N.O. and/or N.C. parts of the system (by making internal cancellations), this generates no problem since we verified that the system is **internally asymptotically stable**.

G.M.V.C.

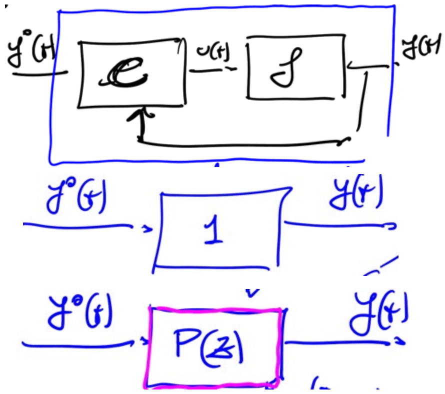
Main limits of M.V.C. are

- can be applied only to *min-phase* systems
- we cannot **moderate** the **control/actuator effort**
- we cannot **design** a specific **behavior** from $y^o(t) \rightarrow y(t)$

To overcome those limits, we need an extension called Generalized M.V.C. in which the **performance index is *extended***.

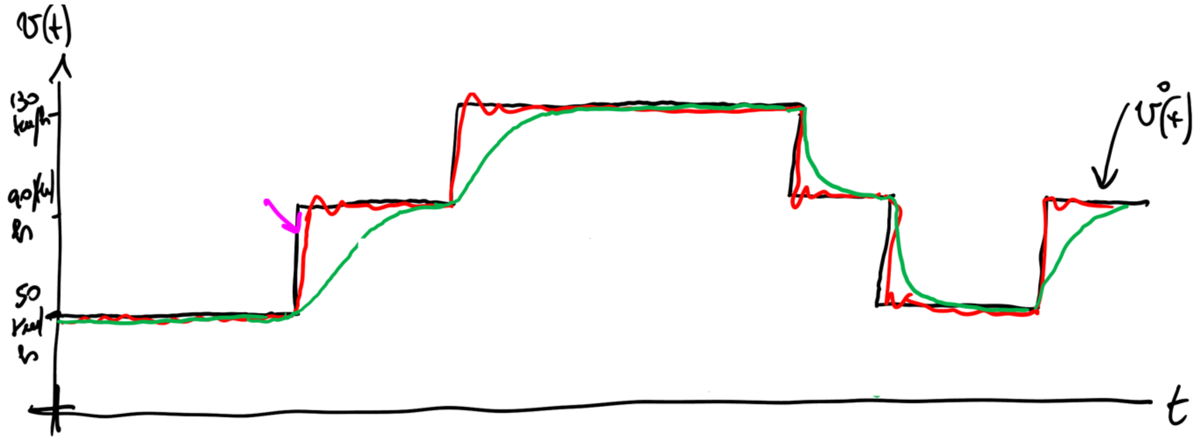
$$J = \mathrm{E}\left[\left(P(z)y(t) - y^o(t) + Q(z)u(t)\right)^2\right]$$

- $P(z)$ is a t.f. called *reference model*
- $Q(z)$ is a t.f. **moderates** the use of $u(t)$, it does so by making a penalty to big values of $u(t)$
- In M.V.C.: $P(z)$=1, $Q(z)$ = 0



Remark on $P(z)$
When we have a feedback system and we want to obtain the best possible tracking the most intuitive solution is to have a closed loop control = 1. However, in most cases perfect tracking is not the best solution, it is better to track on a ***reference model.***

Example
Car's cruise control.



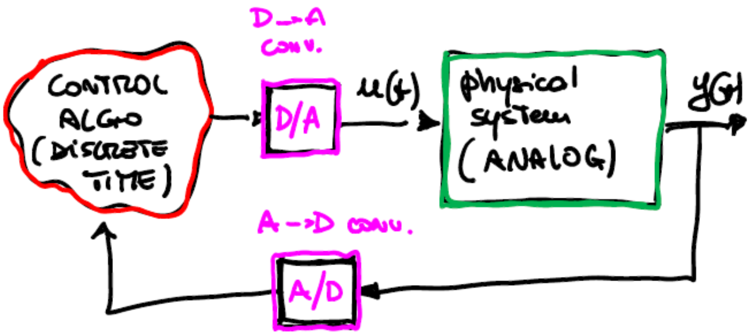
If $P(z)=1$ $v(t)$ should follow $v^o(t)$ as fast as possible.

With $P(z)$ design you can smoothen the behavior

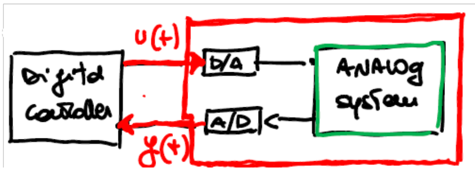
7. Discretization of Analog systems
(valid for identification and control)

We need 2 interfaces, to make the system and the control algorithm communicate correctly.

High quality A/D converters uses a small **sampling time ΔT** and a **amplitude discretization** with a high number of levels.



In case of **black-box** system identification from measured data we **directly estimate** a **discrete Time model**.
If we have a physical **white-box** model, we need to **discretize**.



Discretization

State Space Transformation

Given sampling time ΔT.

Continuous time

$$S: \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Transformation Formulas

$$F = e^{A\,\Delta T}$$
$$G = \int_0^{\Delta T} e^{A\delta} B \, d\delta$$
$$H = C$$
$$D = D$$

Discrete time

$$S: \begin{cases} x(t+1) = Fx(t) + Gu(t) \\ y(t) = Hx(t) + Du(t) \end{cases}$$

It can be proven that the *eigenvalues* (*poles*) follow the "sampling transformation rule".

$$\underbrace{z}_{\substack{z-\text{domain} \\ D.T.}} = \underbrace{e^{s\,\Delta T}}_{\substack{s-\text{domain} \\ \text{C.T.}}} \Rightarrow \lambda_F = e^{\lambda_A \Delta T}$$

The stable region of **continuous time** is the **left plane** ℜe>0.
The stable region of **discrete time** is the **uni-circle**.

The *axis origin* in **c.t.** is mapped in 1+0j.

There is no simple rule for the *zeroes*. We can only say if $G(s)$ is strictly proper ($K > h$)

$$G(s) = \frac{\text{poly. in "s" with } h \text{ zeros}}{\text{poly. in "s" with } k \text{ poles}}$$

We apply discretization rule (s.s.)

$$G(\mathcal{Z}) = \frac{\text{poly. in "}\mathcal{Z}\text{" with } k-1 \text{ zeros}}{\text{poly. in "}\mathcal{Z}\text{" with } k \text{ poles}}$$

So, $G(\mathcal{Z})$ has relative degree = 1
In $\mathcal{Z}$ we have $k - h - 1$ are generated by the discretization, they are called *hidden zeros*. Unfortunately, they are frequently unstable, because of this we need, for instance, G.M.V.C. (which can deal with *non-min-phase* systems) to design control system.

Discretization of time derivative $\dot{x}$

Euler Backward

$$\underbrace{\dot{x}(t)}_{C.T.} \cong \frac{x(t) - x(t-1)}{\Delta T} = \frac{x(t) - z^{-1}x(t)}{\Delta T} = \frac{z-1}{z\Delta T} \cdot x(t)$$

Euler Forward

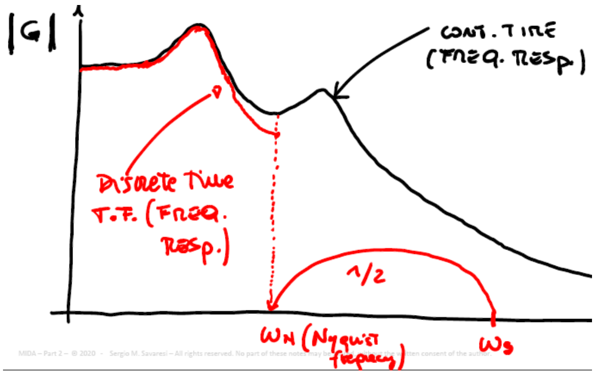
$$\dot{x}(t) = \frac{x(t+1) - x(t)}{\Delta T} = \frac{z-1}{\Delta T} \cdot x(t)$$

General formula for this approach

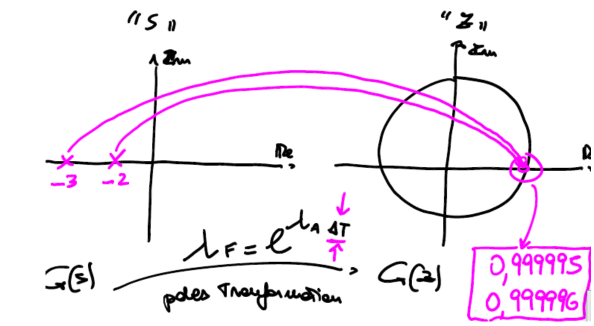
$$\dot{x}(t) = \left[\frac{z-1}{\Delta T} \cdot \frac{1}{\alpha z + (1-\alpha)}\right] x(t)$$

$0 \le \alpha \le 1$ special cases:

- If $\alpha = 0 \rightarrow$ Euler Forward
- If $\alpha = 1 \rightarrow$ Euler Backward
- If $\alpha = ½ \rightarrow$ Tustin Method



Once we decide the sampling frequency ω_s(=$2\pi/\Delta$T) the highest frequency we get is the Nyquist freq. ω_N. however the closer we get to ω_N the worst the approximation is.



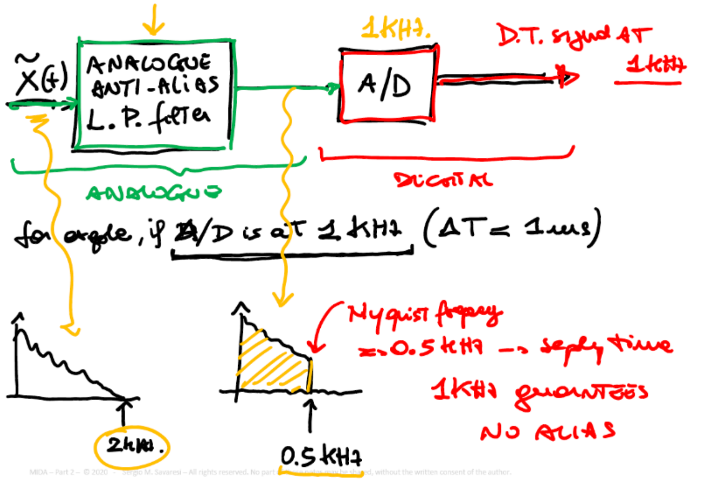
However, choosing an excessive small ΔT might be cost prohibitive

- Sampling devices might be too expensive
- Computational cost
- Cost of memory
- Numerical Precision cost

If we choose a too small ΔT all the poles go to the same region, because we squeeze all poles close to 1+0j. The effect of this is that we need very high numerical precision to avoid instability.

Rule of thumb of control engineers $\rightarrow f_s$ is between 10 and 20 times the system bandwidth we are interested in.

Remark another way of managing the choice of ΔT w.r.t. the *aliasing* problem (signal processing perspective).



If we make the spectrum of the analog signal, we want to convert into digital we obtain the bandwidth of the full spectral content of $\tilde{x}(t)$.

Before we convert it to digital, we might have an *anti-alias low-pass* filter, which cuts it off at the Nyquist frequency.

Fully digital approach: