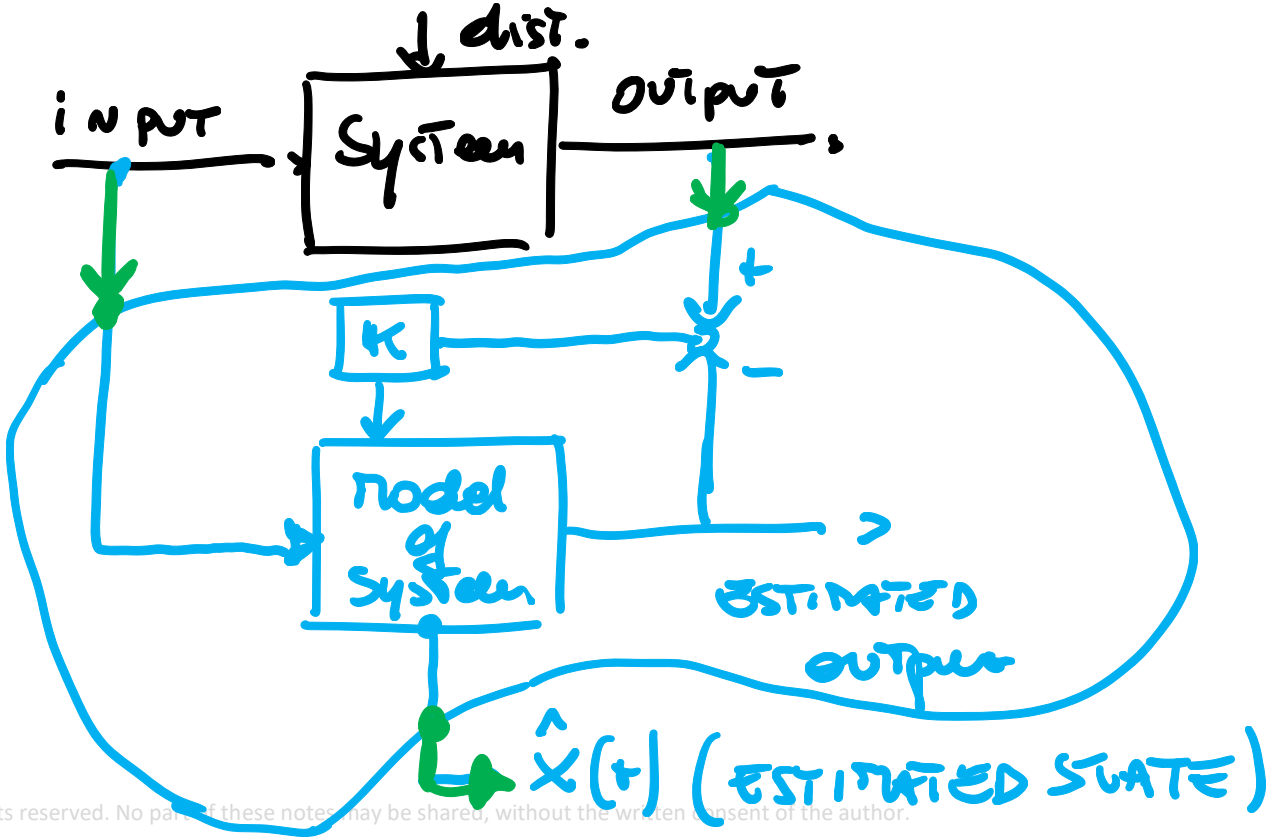
Chapter #6 → SW_SENSING (VIRTUAL_SENSING) (VARIABLE ESTIMATION) with BLACK-BOX methods

In chapter #3 we have SEEN classical Technology of sw-sensing based on K.F:



MAIN FEATURES of this approach:

- A (white-box / physical) model must be AVAILABLE
- NO NEED ("IN principle") of a TRAINING data-set including measurements of the STATE TO-BE-ESTIMATED
- It is a feedback-estimation Algo. (feedback correction of the model, using estimated-output error)
- CONSTRUCTIVE METHOD (NON-parametric / NO optimization INVOLVED)
- CAN BE USED ("IN principle") TO ESTIMATE STATES which are IMpossible to BE MEASURED (ALSO AT prototyping/TRAINING/DESIGN STAGE)

Are there other classes of SOL-seeking techniques?

YES → BLACK-BOX approaches with "LEARNING/TRAINING" from data ("machine-learning" approaches)

In this chapter we are seen, focusing on the ARCHITECTURES (we don't NEED new Algo. / JUST USE something we have already studied)

LET'S START with the case of L.T.I. systems

LET'S consider a simplified case (SISO system with 1 STATE) to UNDERSTAND the Approach

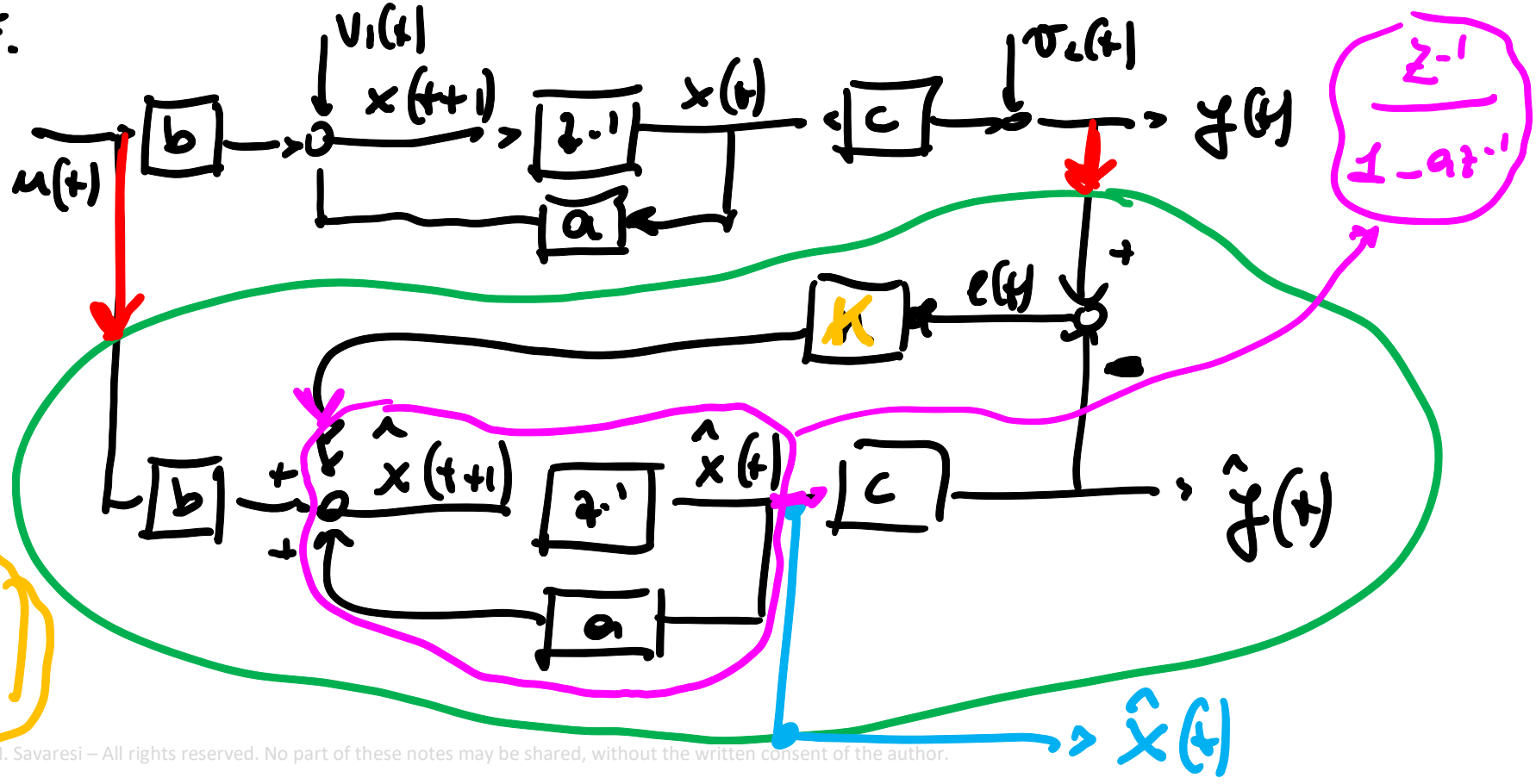
$$\mathcal{S}: \begin{cases} x(t+1) = a\,x(t) + b\,u(t) + v_1(t) \\ y(t) = c\,x(t) + v_2(t) \end{cases}$$

$v_1 \sim$ WN

$v_2 \sim$ WN

problem -> estimation of $\hat{x}(t)$ from measured signals $u(t)$ and $y(t)$

OBSERVER K.F. architecture:



K -> obtained by minimizing

$\mathrm{Var}\left[x(t) - \hat{x}(t)\right]$

Let's find the relationship between $u(t) \rightarrow \hat{x}(t)$
$y(t) \rightarrow \hat{x}(t)$

$$\hat{X}(t) = \frac{b \cdot \frac{z^{-1}}{1-az^{-1}}}{1 + KC\frac{z^{-1}}{1-az^{-1}}} u(t) + \frac{K \cdot \frac{z^{-1}}{1-az^{-1}}}{1 + KC\frac{z^{-1}}{1-az^{-1}}} y(t)$$

$$\vdots$$

$$\tilde{X}(t) = \frac{b}{1 + (KC - a)z^{-1}} u(t-1) + \frac{K}{1 + (KC - a)z^{-1}} y(t-1)$$

we can estimate B.B. these two transfer functions from data! => K.F. is a sophisticated way to BUILD these T. FUNCTIONS from a L.S. model → we can estimate these T.F. directly from data

We can adopt a B.B. approach to estimate these T.F:

DATA-SET (for TRAINING)

$\{u(1), u(2), u(3) \ldots u(N)\}$

$\{y(1), y(2), y(3) \ldots y(N)\}$

$\{x(1), x(2), x(3) \ldots x(N)\}$

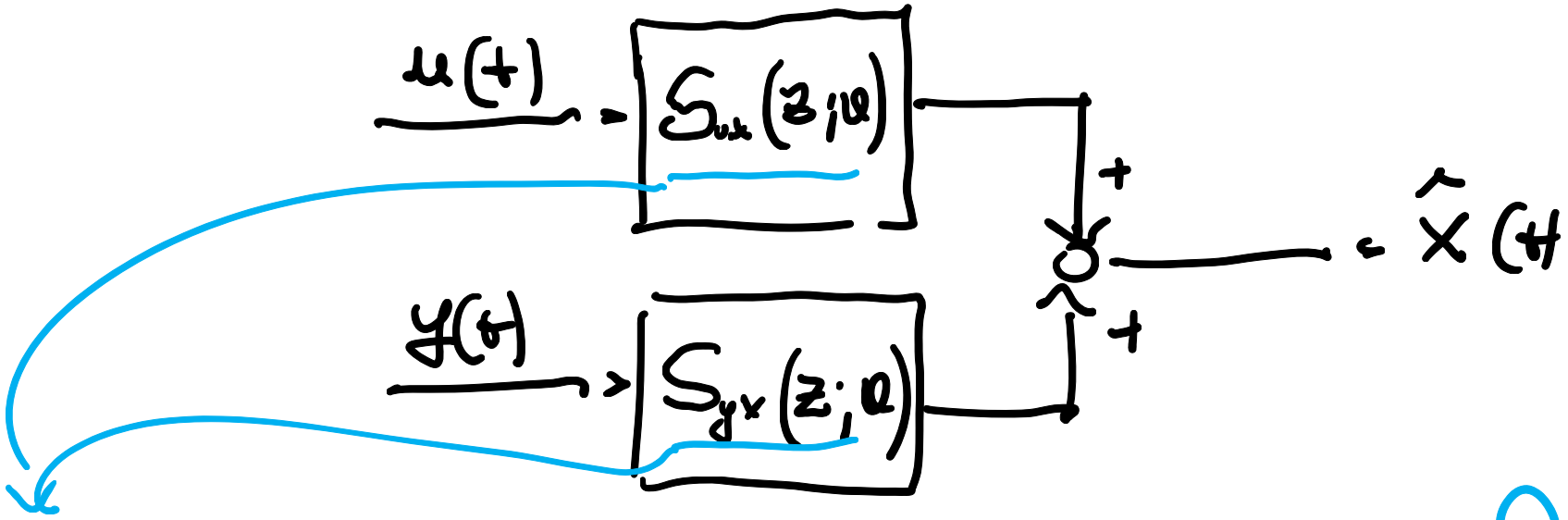
→ In the supervised training approach we NEED measurements of the state to BE ESTIMATED (physical sensor for $x(t)$ ONLY for the DESIGN/TRAINING of the SW SENSOR)

→ we can use 4SID for DIRECT non parametric identification of $u \longrightarrow x$ dynamics
$y \nearrow$

OR

→ we can use a classic parametric SY. ID. Approach

SELECT model class:



parametric T.F. with parameter vector θ

perf. INDEX:

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

$X(t)$ ← MEASURED STATE

$S_{ux}(z,\theta) \cdot u(t) + S_{yx}(z;\theta)\, y(t)$ ESTIMATED STATE from models

variance of the estimation error

optimization → $\hat{\theta}_N = \arg\min_{\theta} \{ J_N(\theta) \}$

we obtain ⇒

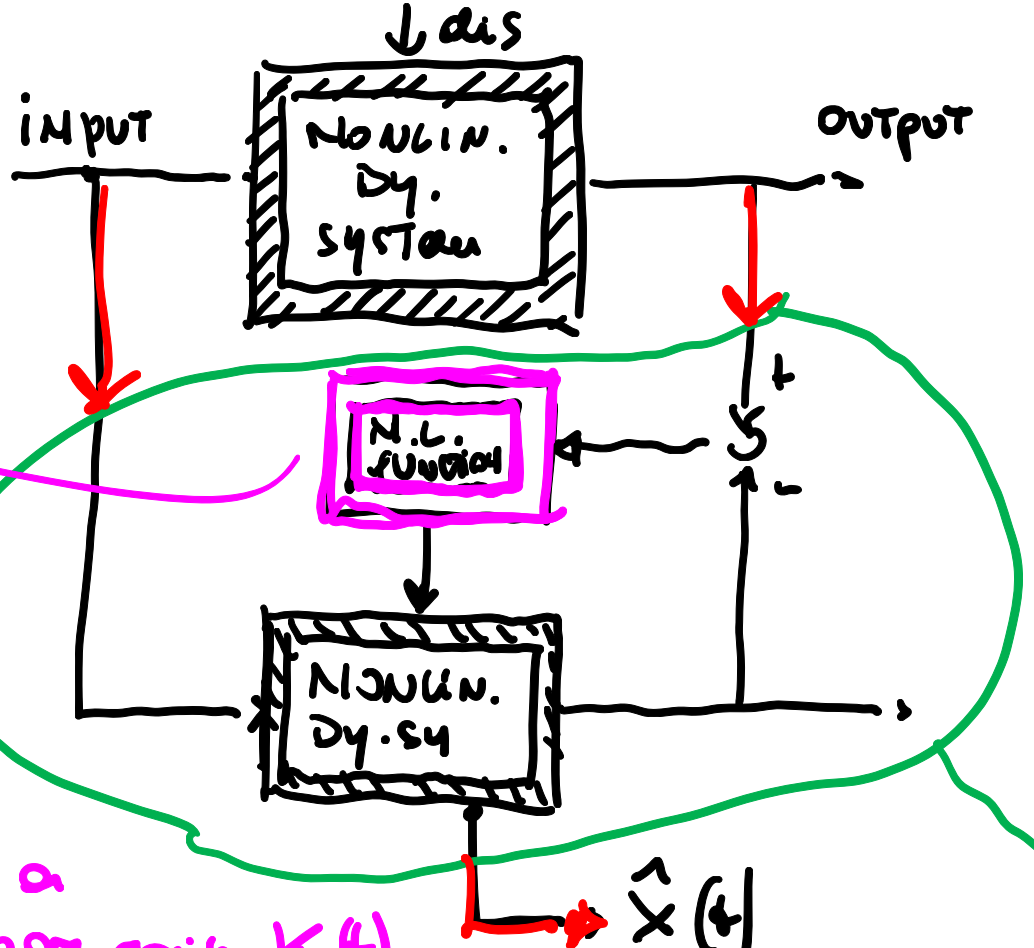
$$S_{ux}(z;\hat{\theta}_N)$$
$$S_{yx}(z;\hat{\theta}_N)$$

→ "BB. VIRTUAL SENSOR"

Comparison table between K.F. and B.B sw. setup:

	K.F.	B.B
NEED of a (W.S) physical model of S.	YES	NO
NEED of a TRAINING DATASET	(NO)	YES ←
Interpretability of the RESULT	YES	NO
EASY RE-TUNING for a SIMILAR (DIFFERENT) SYSTEM	YES	NO
ACCURACY of the ESTIMATION	GOOD (APPROX. MODEL)	VERY GOOD
CAN BE USED ALSO IN CASE OF UN-REASONABLE STATES	YES	NO (WE NEED X(t) MEAS. FOR TRAIN.) ←

When the system is NON-LINEAR the problem becomes more complicate. LET'S START again by TAKING INSPIRATION from KF (EKF)

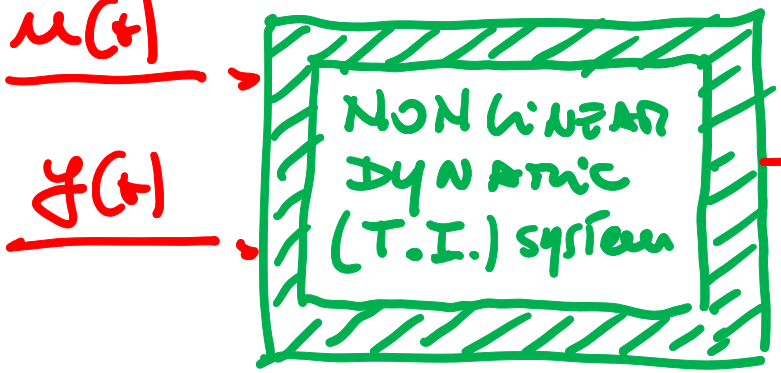


NOTATION:

- LINEAR (STATIC or dynamic)
- STATIC Nonlinear system
- DYNAMIC nonlinear system

Remark: IN K.F the EKF extension uses the "TRICK" of a TIME-VARYING linear gain K(t) BUT the obvious and NATURAL choice is a NONLINEAR gain (STATIC NONLINEAR FUNCTION)

the content of the " " green box is:



→ filter of state:

$$\hat{x}(t) = \hat{x}(t/t)$$

we are allowed, if needed to use data up to present time t

the problem is, again, the BB identification (SUPERVISED LEARNING) of a N.L. dynamic system, starting from a measured "TRAINING DATASET"

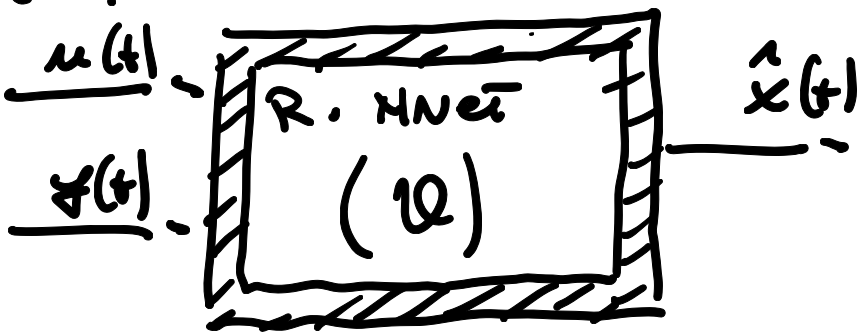
$\{u(1), u(2) \ldots \quad u(N)\}$

$\{y(1), y(2) \ldots \quad y(N)\}$

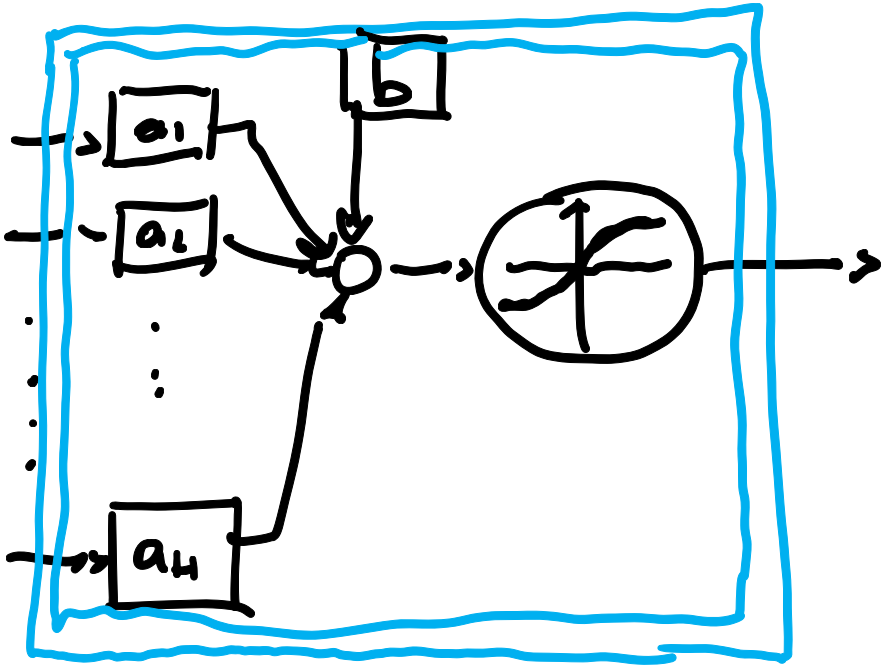
$\{x(1), x(2) \ldots \quad x(N)\}$

physical sensors needed for training

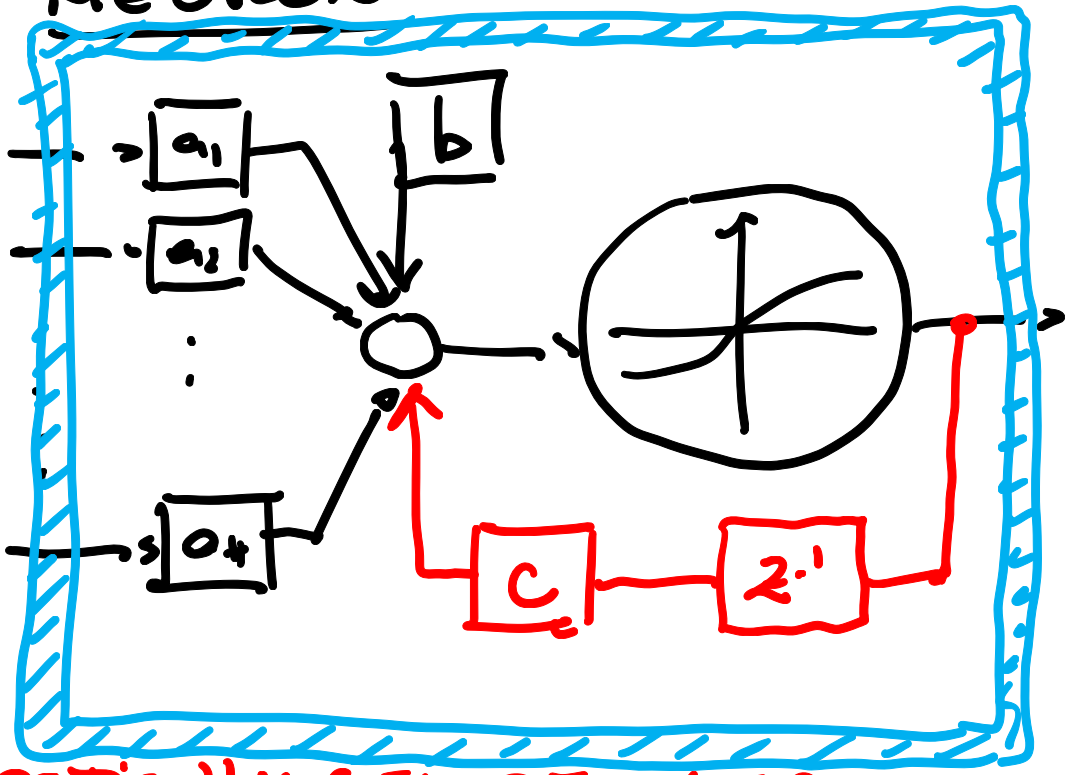
ARCHITECTURE # 1 (most general) → USE a RECURRENT NEURAL NETWORK:



STATIC function of a NEURON:

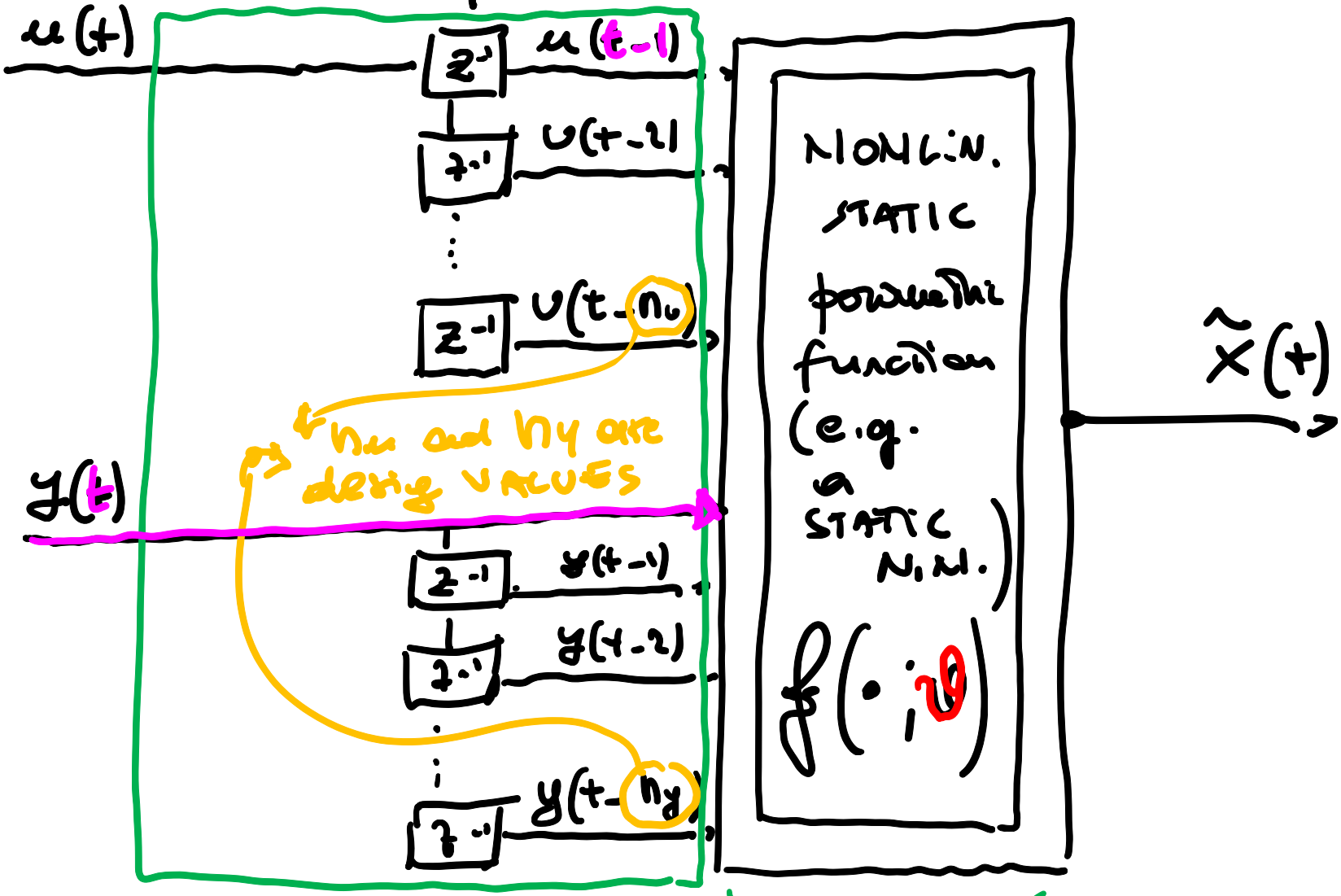


→ UPGRADE TO a dynamic NEURON:



MOST GENERAL APPROACH BUT PRACTICALLY SELDOM USED → MAJOR ISSUES of STABILITY and CONVERGENCE of TRAINING.

Architecture #2 → split the system into a STATIC NL. system and LINEAR dynamics (FIR architecture)



Remark: notice that in principle $\hat{x}(t)$ can depend on $y(t)$, whereas we know that $\bar{x}(t)$ can only depend on $u(t-1)$ (and other past values)

In case of a MIMO system with:

m inputs: $u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix}$ p outputs: $y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_p(t) \end{bmatrix}$

and n states $x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$ the estimation problem is

the search of the optimal parameter vector θ for the function:

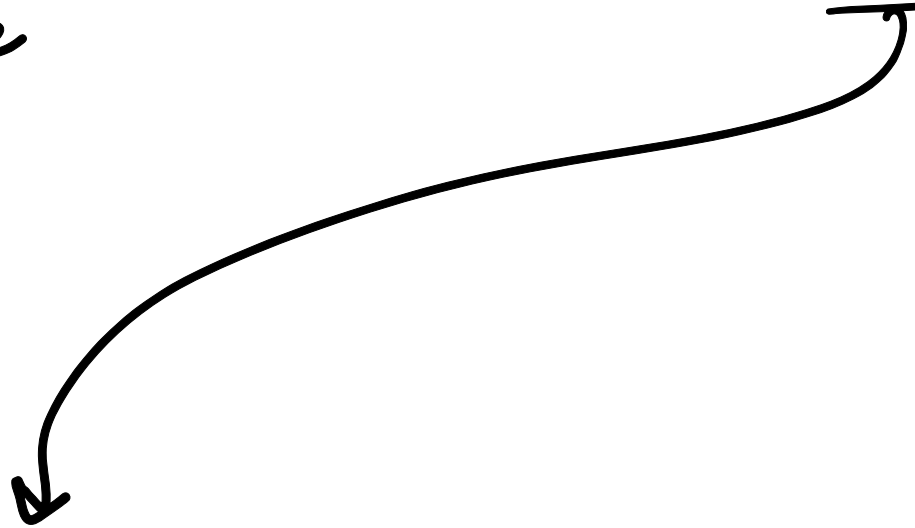
$$f(\cdot;\theta): \mathbb{R}^{m \times n_u + p \times (n_y+1)} \longrightarrow \mathbb{R}^n$$

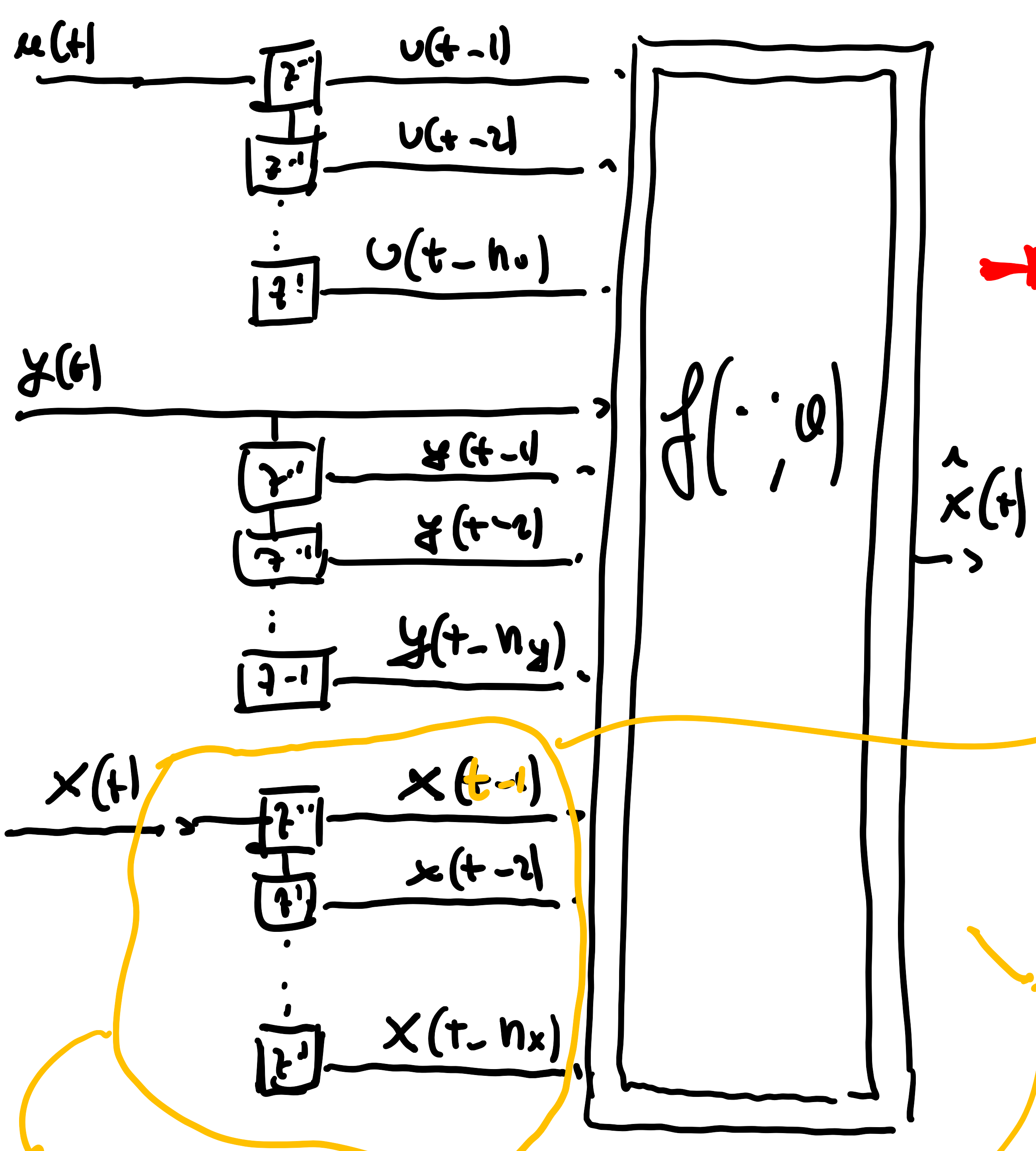
Estimation (TRAINING) of this function $f(\cdot;\theta)$ is much more simple than the estimation of a Rec. NN

MOREOVER: STABILITY is GUARANTEED

(the full system is F.I.R.)

Architecture # 3 → STATIC nonlinear function plus linear dynamic but without IIR scheme





POTENTIAL ADVANTAGE W.R.T. Arch #2 → n_u and n_y are smaller

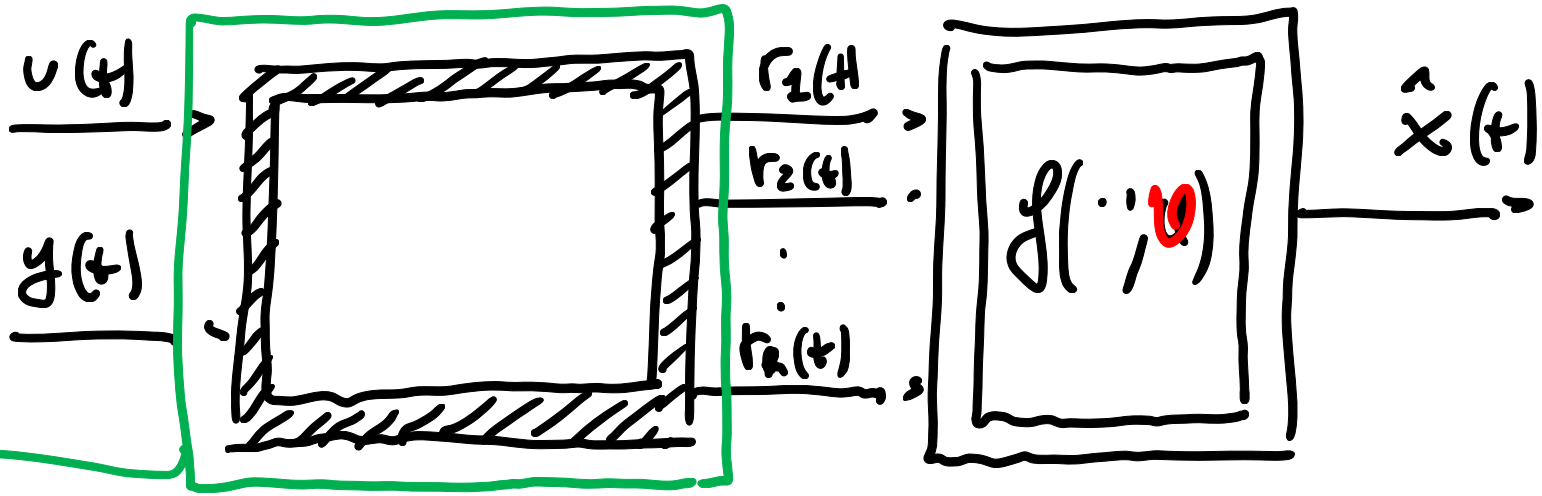
POTENTIAL DISADV. → this ARCHITECTURE is NOT GUARANTEED TO be STABLE by CONSTRUCTION

this part of the system is "RECURSIVE" →

I.I.R ARCHITECTURE

"IN PRODUCTION" → we feedback the delayed $\hat{x}(t)$ signal (→ problem of INSTABILITY)

ARCHITECTURE #4 → separation of system dynamics and a STATIC nonlinear system USING REGRESSORS BUILT from physical knowledge

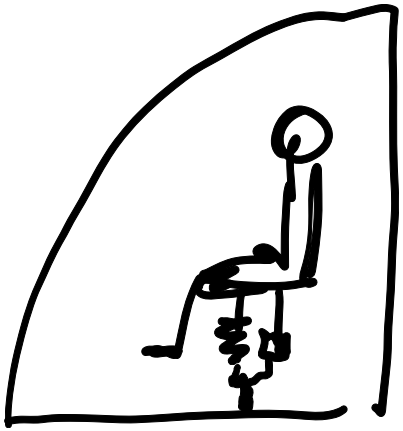


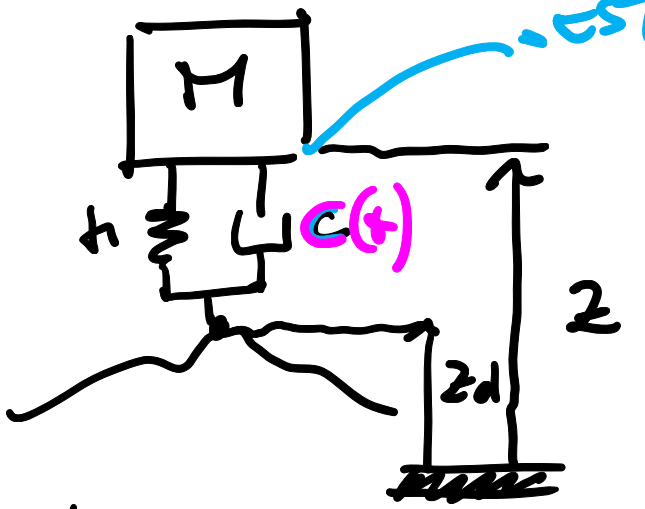
↓ SYSTEM (can be dynamic AND nonlinear) that BUILDS the REGRESSORS (signals $r_1(t)$ .. $r_n(t)$) from physical signals $u(t)$ and $y(t)$, USING physical KNOWLEDGE / UNDERSTANDING of the system

(it uses "human-driven" → "ART" of M.L / SW. eng.)

↳ the idea is to FACILITATE the "JOB" of $f(\cdot;\theta)$ by PRESENTING at its input a SMALLER and more MEANINGful SET of signals (REGRESSORS)

EXAMPLE (CONTINUE example of the end of chapter #3)





ESTIMATION (SW sensor) Prob. $\dot{z} = ?$ (one of the states)

Now we assume a "semi-active" suspension → c(t) can be electronically changed (control variable)

Model (key equations) of the system:

$$M\ddot{z} = -c(t)(\dot{z} - \dot{z}_a) - K(z - z_a)$$

$\ddot{z}_a$ → measurable input (accelerometer)

$y(t) = z - z_d$ → measurable output (ecoug. sensor)

-- we can solve the problem with KF. (see ch. 3.)
OR --> we can make an experiment and collect TRAINING data:

DATASET

$C(t) \rightarrow \{C(1), C(2) \ldots C(N)\}$

$z(t) - z_a(t) \rightarrow \{z(1) - z_a(1) \ldots z(N) - z_a(N)\}$

$\ddot{z}(t) \rightarrow \{\ddot{z}(1)\ \ddot{z}(2) \ldots \ddot{z}(N)\}$

AND FOR DESIGN only:

$\dot{z}(t) \rightarrow \{\dot{z}(1)\ \dot{z}(2) \ldots \dot{z}(N)\}$

↳ MEASURED BUT JUST for TRAINING!

Back to the main equation!

$$M\ddot{z} = -K(z - z_a) - c(t)(\dot{z} - \dot{z}_a)$$

↓ Integrate:

$$\dot{z} = -\frac{K}{M}\int (z - z_a)\,dt - \frac{1}{M}\int c(t)(\dot{z} - \dot{z}_a)\,dt$$

$r_1(t)$ $r_2(t)$

primary regressors → directly linked to $\dot{z}(t)$

$\dot{z}$: TO BE ESTIMATED

D D D sensors

we also consider this equation

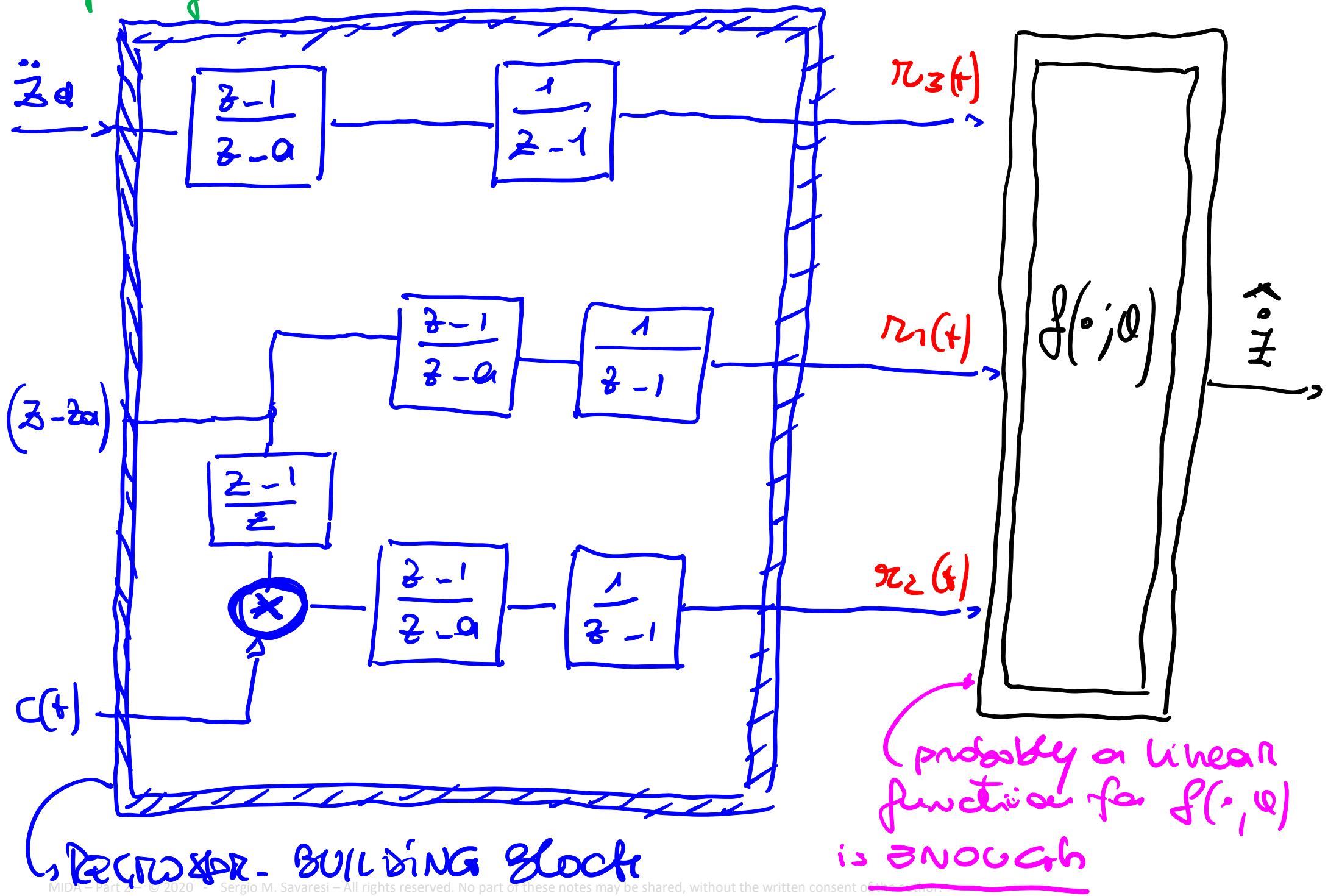
$$\dot{z}_a = \int \ddot{z}_a\,dt$$

$r_3(t)$

, secondary reg. → can help $r_1(t)$

since these regressors are obtained by integration → to avoid drifty (by DC-components of noise integration) → we have to High-pass the inputs with H.P. filters: $\frac{z-t}{z-a}$

Full factory scheme:



Conclusions —>

In case of B.B SW testing with NONLINEAR system —. problem can be quite complex

USING "BRUTE force" approach (1 Dynamic HW) is usually doomed to failure

The best is to gain some insight into the system and build some "SMART" REGRESSIONS BEFORE "B.B. mapping"