Now we can eventually present the PROCEDURE (ALGO) to obtain $\{\hat{F}, \hat{G}, \hat{H}\}$ starting from a

NOISE-FREE measured I.R.:

$$\{\omega(0), \omega(1) \ldots \omega(N)\}$$

STEP 1: Build the HANKEL matrix in increasing order; EACH Time check RANK

$$H_1 = [\omega(1)]$$

$$H_2 = \begin{bmatrix} \omega(1) & \omega(2) \\ \omega(2) & \omega(3) \end{bmatrix} \longrightarrow \text{check} \to \text{RANK}(H_2) = 2 \checkmark$$

$$H_3 = \begin{bmatrix} \omega(1) & \omega(2) & \omega(3) \\ \omega(2) & \omega(3) & \omega(4) \\ \omega(3) & \omega(4) & \omega(5) \end{bmatrix} \longrightarrow \text{check} \to \text{RANK}(H_3) = 3 \checkmark$$

$$\vdots$$

$$H_n = \begin{bmatrix} \cdots \end{bmatrix} \longrightarrow \text{check} \to \text{RANK}(H_n) = n \checkmark$$

$$H_{n+1} = \begin{bmatrix} \cdots \end{bmatrix} \longrightarrow \text{check} \to \text{RANK}(H_{n+1}) = n$$

STOP

→ 1ST matrix NOT Full-RANK

STEP 1 → we have ESTIMATED The ORDER n of the system

STEP 2 take H_{n+1} (it is a square $n+1 \times n+1$ matrix of rank $= n$)

and factorize it

into two RECTANGULAR matrices of size $n+1 \times n$ and $n \times n+1$

(Remark → possible because rank of H_{n+1} is n)

$$H_{n+1} = \begin{array}{c} \overset{n}{\longleftrightarrow} \\ \left[\begin{array}{c} \text{we consider this matrix as} \\ \text{EXTENDED OBSERVABILITY matrix } O_{n+1} \end{array} \right] \updownarrow n+1 \end{array} \begin{array}{c} \overset{n+1}{\longleftrightarrow} \\ \left[\begin{array}{c} \text{we consider this matrix as} \\ \text{EXTENDED CONTROLLABILITY matrix } R_{n+1} \end{array} \right] \updownarrow n \end{array}$$

$$O_{n+1} = \left.\underbrace{\begin{bmatrix} \boxed{H} \\ HF \\ HF^2 \\ \vdots \\ HF^{n-1} \\ HF^n \end{bmatrix}}_{n}\right\} n+1 \qquad R_{n+1} = \overbrace{\begin{bmatrix} \boxed{G} & FG & F^2G & \dots & F^{n-1}G & F^nG \end{bmatrix}}^{n+1}\Big\} n$$

<u>STEP 3</u> (H, F, G estimation)

Using O_{n+1} and R_{n+1} we can easily find

$$\hat{G} = R_{n+1}(:;1)$$

$$\hat{H} = O_{n+1}(1;:)$$

What about $\hat{F}$? Consider for example O_{n+1} (but the same can be done also starting from R_{n+1})

$$O_{n+1} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^{n-1} \\ HF^n \end{bmatrix}$$

$$O_1 = O_{n+1}(1:n\,;\,:)$$

$$O_2 = O_{n+1}(2:n+1\,;\,:)$$

$\downarrow$ O_1 and O_2 are square "$n \times n$" matrices

Notice that O_1 and O_2 are linked by the so-called "shift invariance property" ⟶

$$O_2 = O_1 \cdot F$$

↳ since O_1 is square and invertible

$$\boxed{\hat{F} = O_1^{-1} \cdot O_2}$$

~~end of~~ ALGO.

Conclusion: in a simple and CONSTRUCTIVE way we have estimated a S.S. model of the system $\{\hat{F}, \hat{G}, \hat{H}\}$ starting from a MEASURED (noise-free) I.R., using ONLY $2n+1$ samples of the I.R.

Example: $N = 200$, $n = 4$ — we can make system IDENT. using only 9 samples of I.R.

This method was known since '60 (KALMAN–HO method)

It is simple, clean, nice BUT

USELESS!

why practically USELESS? $\rightarrow$ if $w(t)$ is NOISY $\rightarrow$

- STEP 1 NEVER STOPS (stops with probab. $= \emptyset$)
- EVEN if we know A-PRIORI the VALUE of n $\rightarrow$ the estimated $\{\hat{F}, \hat{G}, \hat{H}\}$ would be BADLY WRONG! (Method is NOT ROBUST W.R.T. measurement NOISE)

$\downarrow$

this method went sleeping until late '80 when a new NUMERICAL-ALGEBRA TOOL was fully DEVELOPED!

S. V. D.

Singular VALUES Decomposition

A Technique for
- DATA-compression
- optimal separation of SIGNAL from NOISE

Prof. G. GOLUB AT STANFORD
- NETWORKS ~ early '90
- Google ~ late '90

Now we can consider the REAL problem →

we have noisy measurement of I.R.

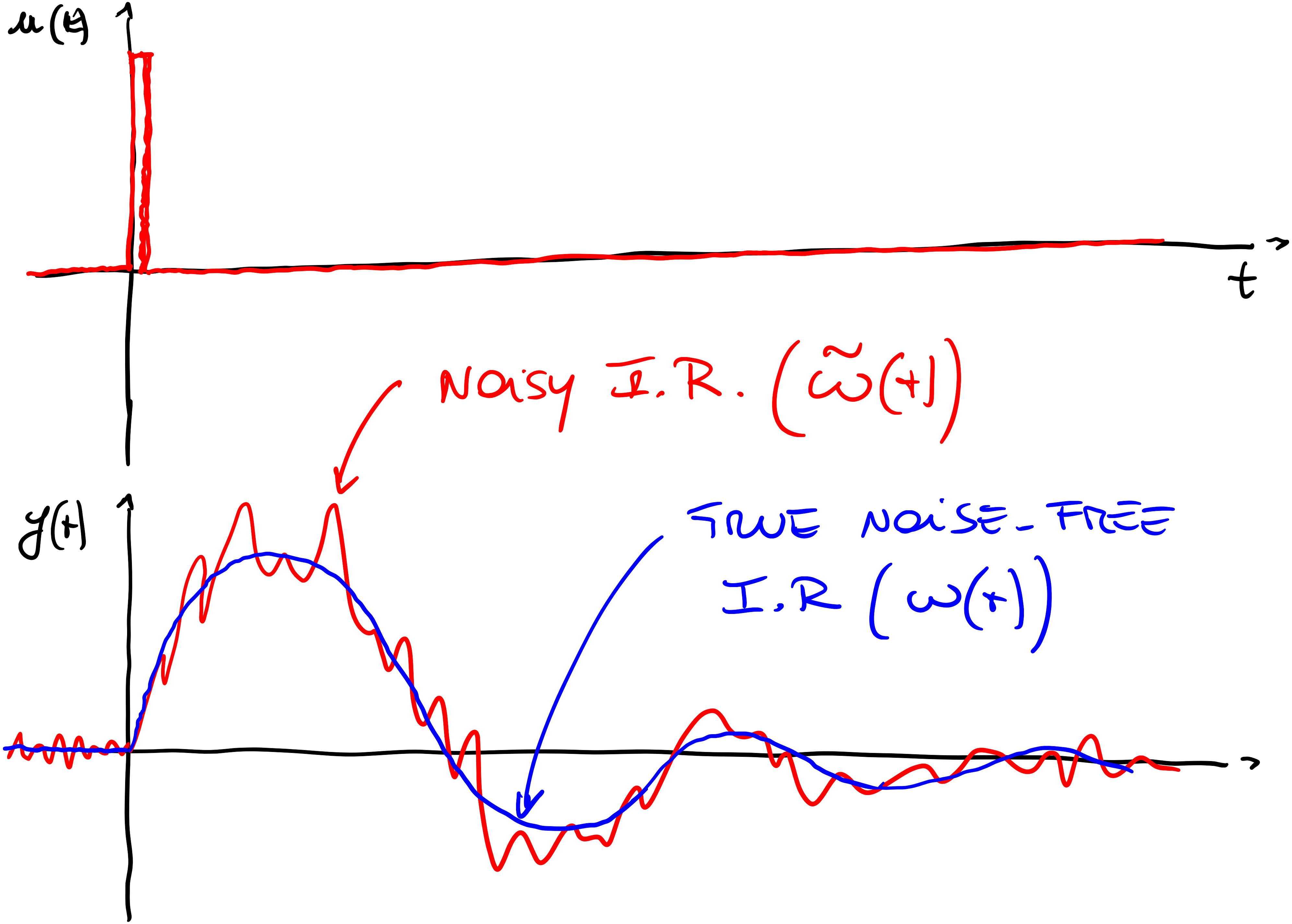
$$\underbrace{\tilde{\omega}(t)}_{} = \underbrace{\omega(t)}_{} + \underbrace{n(t)}_{} \qquad t = 0, 1, 2, 3 \ldots N$$

Measured noisy I.R.

TRUE (NOISE-FREE) I.R.

MEAS. NOISE (BROAD-BAND NOISE)

Typical value of N is something between 100 → 1000


u(t)
t
Noisy I.R. ($\tilde{\omega}(t)$)
y(t)
TRUE NOISE-FREE I.R ($\omega(t)$)

4SID procedure using noisy meas. of I.R

STEP 1: Build Hankel matrix from data using "ONE shot" ALL the N available DATA-points

$$\tilde{\omega}(t) \quad t = 0, 1, 2, \dots N$$

$$\Downarrow$$

$$\tilde{H}_{qd}$$

we indicate that is built with NOISY DATA

NOT a square matrix but a RECTANGULAR MATRIX of size $q \times d$

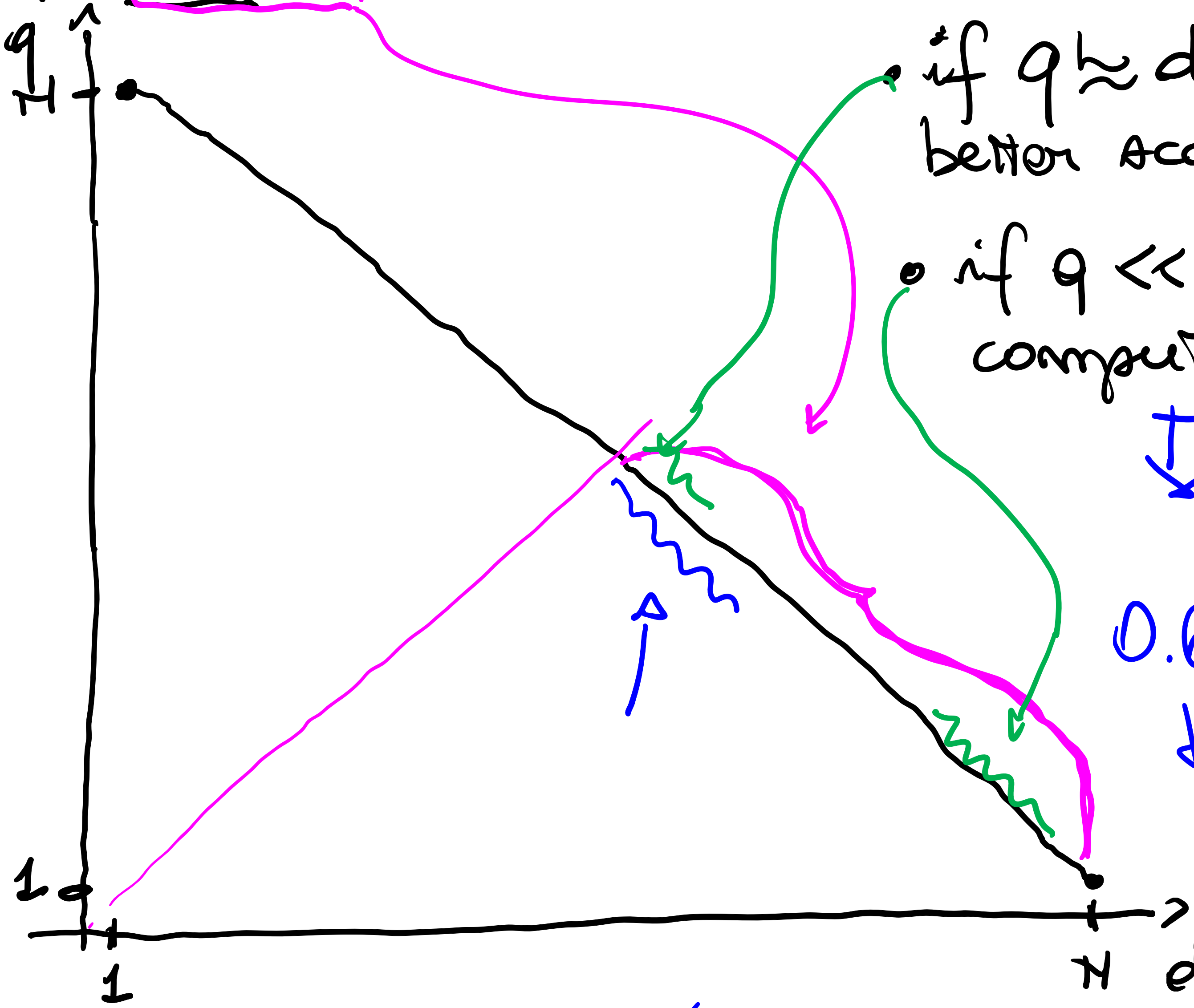
$$
\tilde{H}_{qd} = \begin{array}{c|c}
 & \begin{bmatrix}
\tilde{\omega}(1) & \tilde{\omega}(2) & \tilde{\omega}(3) & \cdots & \tilde{\omega}(d) \\
\tilde{\omega}(2) & \tilde{\omega}(3) & \tilde{\omega}(4) & \cdots & \tilde{\omega}(d+1) \\
\vdots & & & & \\
\vdots & & & & \\
\tilde{\omega}(q) & \tilde{\omega}(q+1) & \cdots & & \tilde{\omega}(q+d-1)
\end{bmatrix} \\
q \updownarrow & \\
 & \xleftrightarrow[\quad d \quad]{}
\end{array}
$$

we want to use the full

DATASET $\Rightarrow \boxed{q+d-1 = N}$

Remark (on the choice of q and d)

hp: $\boxed{q<d}$ $\rightarrow$ $q+d-1=N \iff q=N+1-d$



- if $q \simeq d$ the method has a better accuracy
- if $q \ll d$ the method is computationally less intensive

$\Downarrow$ Rule of thumb

$$0.6d < q < d$$

the sensitivity of the estimation quality to the choice of q and d is very low $\Rightarrow q \approx 0.6d$

Example ~ $N=1000$ / $q=400$ $d=601$

STEP 2: $\boxed{\text{SVD of } \tilde{H}_{qd}}$

$$\tilde{H}_{qd} = \tilde{U}\tilde{S}\tilde{V}^T$$

$$\boxed{q \times d} = \boxed{q \times q}\,\boxed{q \times d}\,\boxed{d \times d}$$

$\tilde{U}$ and $\tilde{V}$ are UNITARY MATRICES →

Def: M (square matrix) is UNITARY if:

$$\begin{cases} \bullet\ \det(M) = 1 \ (\Rightarrow \text{INVERTIBLE}) \\ \bullet\ M^{-1} = M^T \end{cases}$$

$$\tilde{S} = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & \emptyset & \\ & & \sigma_3 & & \\ & \emptyset & & \ddots & \\ & & & & \sigma_q \end{bmatrix}$$

$\sigma_1, \sigma_2, \sigma_3 \ldots \sigma_q$ are the "SINGULAR VALUES" of $\tilde{H}_{qd}$

They are REAL, POSITIVE numbers, SORTED in DECREASING ORDER:

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots \geq \sigma_q$$

Remark: the singular VALUES of a RECTANGULAR matrix are a "SORT of" EIGENVALUES of a square matrix

SVD is a "sort of" DIAGONALIZATION of a RECTANGULAR MATRIX

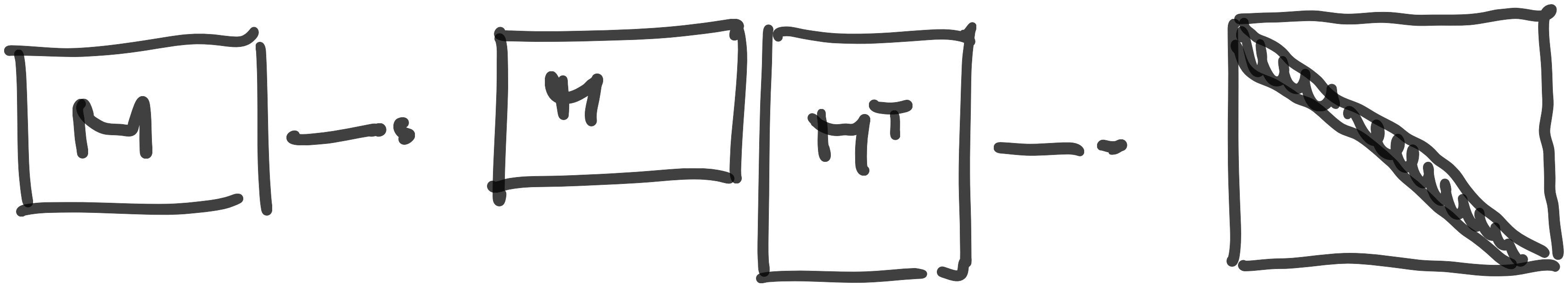
Recall that for a square matrix

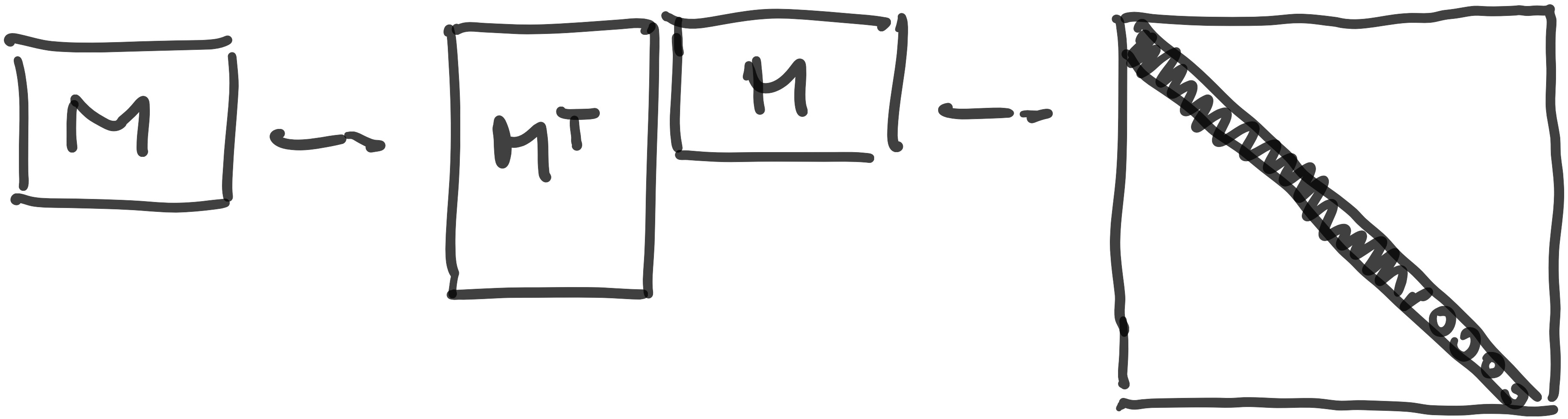
$$eig(A) = roots(det(A - \lambda I))$$

if M is RECTANGULAR:

$$\boxed{SV(M) = \sqrt{EIG(MM^T)} = \sqrt{EIG(M^T M)}}$$

Relationship VALID only for NON-ZERO EIGENVALUES


M
M
M^T


M
M^T
M

How can we compute SVD? → optimal numerical computation is NOT TRIVIAL →

→ USE >> svd(M) in MATLAB

Theoretical method for SVD computation is to make 2 DIAGONALIZATION STEPS:

$$\underbrace{\tilde{H}_{qq}\tilde{H}_{qq}^T}_{\text{square } q\times q \text{ matrix}} = \boxed{\tilde{U}}\,\boxed{\tilde{S}\tilde{S}^T}\,\tilde{U}^T$$

$$\begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_q^2 \end{bmatrix}$$

$$\underbrace{\tilde{H}_{qq}^T \cdot \tilde{H}_{qd}}_{\substack{\text{square matrix} \\ d \times d}} = \tilde{V} \; \boxed{\tilde{S}^T \tilde{S}} \; \boxed{\tilde{V}^T}$$

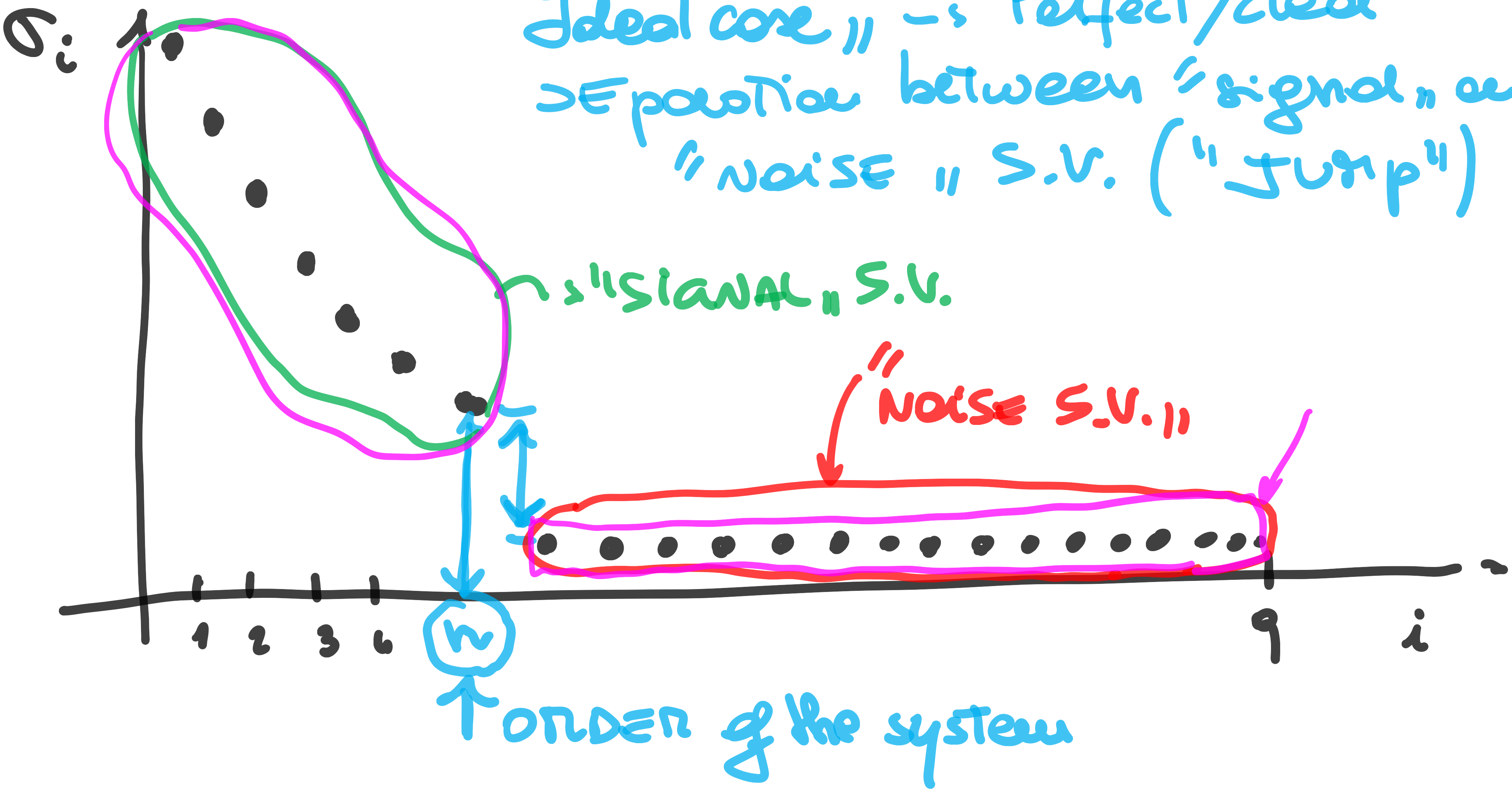
DIAG.

$$\tilde{S}^T \tilde{S} = \begin{bmatrix} \sigma_1^2 & & & & & & \\ & \sigma_2^2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_q^2 & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}$$

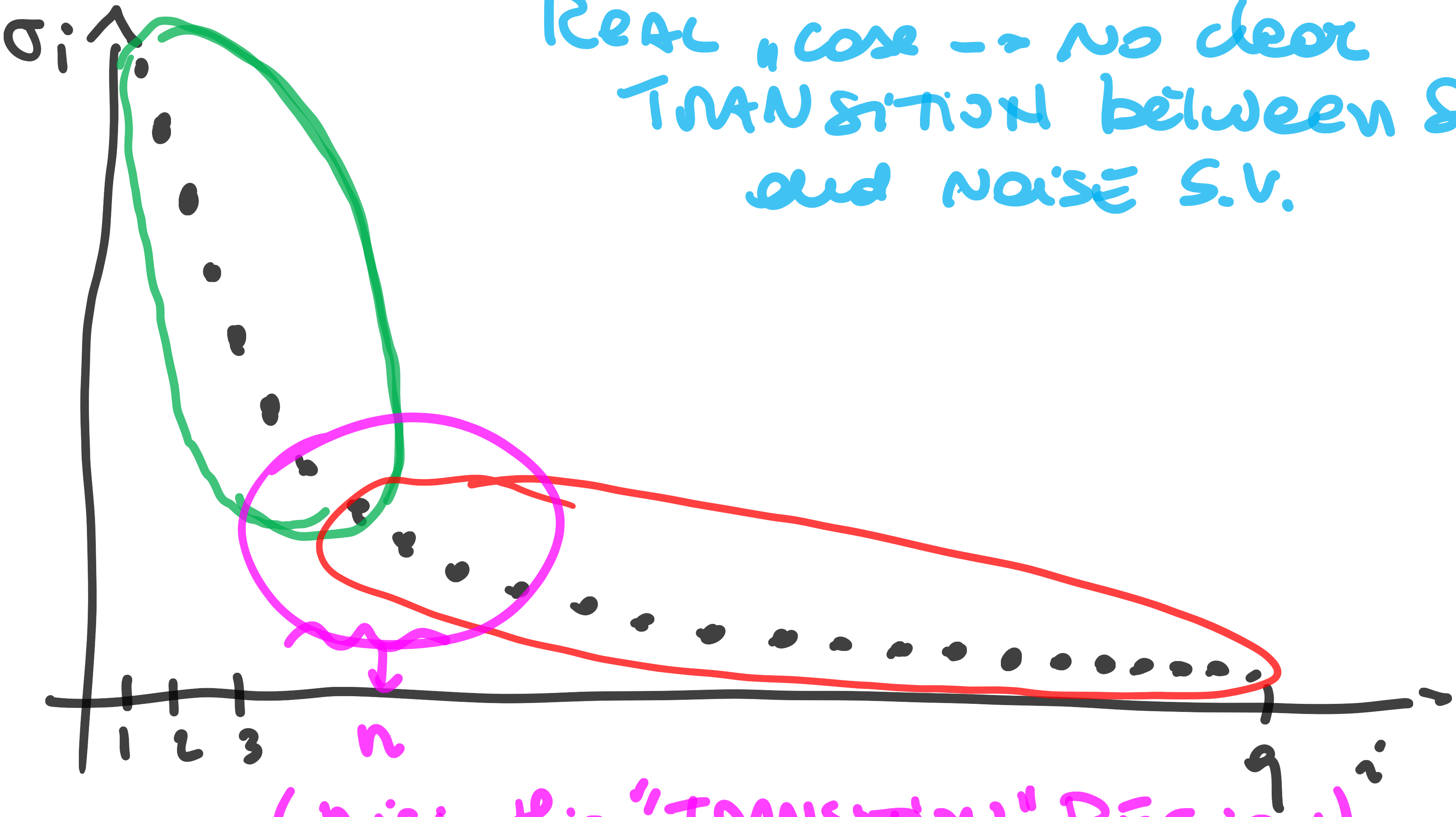
STEP #3: plot the S.values and "cut-off" the 3 MATRICES:

$$\boxed{\sigma_1, \sigma_2 \;.. \qquad \sigma_q}$$

"Ideal case" → Perfect/clear separation between "signal" and "noise" S.V. ("jump")



"Real" case → no clear transition between signal and noise S.V.



(n is in this "transition" region)

→ with some empirical test we can select a good compromise between complexity, precision and overfitting (see cross-validation)

After decision on the value of n
we split $\tilde{U}$ $\tilde{S}$ and $\tilde{V}^T$:

$$\tilde{H}_{qd} = \tilde{U}\,\tilde{S}\,\tilde{V}^T$$

n $n \times n$ n

$$\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \\ & & & \sigma_n \end{bmatrix}$$

$$\underbrace{\tilde{H}_{qd}}_{\text{rank} = q} = \underbrace{\hat{U}\hat{S}\hat{V}^T}_{\hat{H}_{qd}} + \underbrace{H_{RES\,qd}}_{\text{rank} = q}$$

Difference between $\tilde{H}_{qd}$ and $\hat{H}_{qd}$

rank $\hat{H}_{qd} = n$

$\tilde{H}_{qd} \rightarrow \hat{H}_{qd}$ we make a major rank reduction

STEP 4: The procedure can be completed with the estimation of $\{\hat{F}, \hat{G}, \hat{H}\}$, using the "cleaned" matrix $\hat{H}_{qd}$

$$\hat{H}_{qd} = \hat{U}\hat{S}\hat{V}^T = \underbrace{\hat{U}\hat{S}^{1/2}}\underbrace{\hat{S}^{1/2}\hat{V}^T}$$

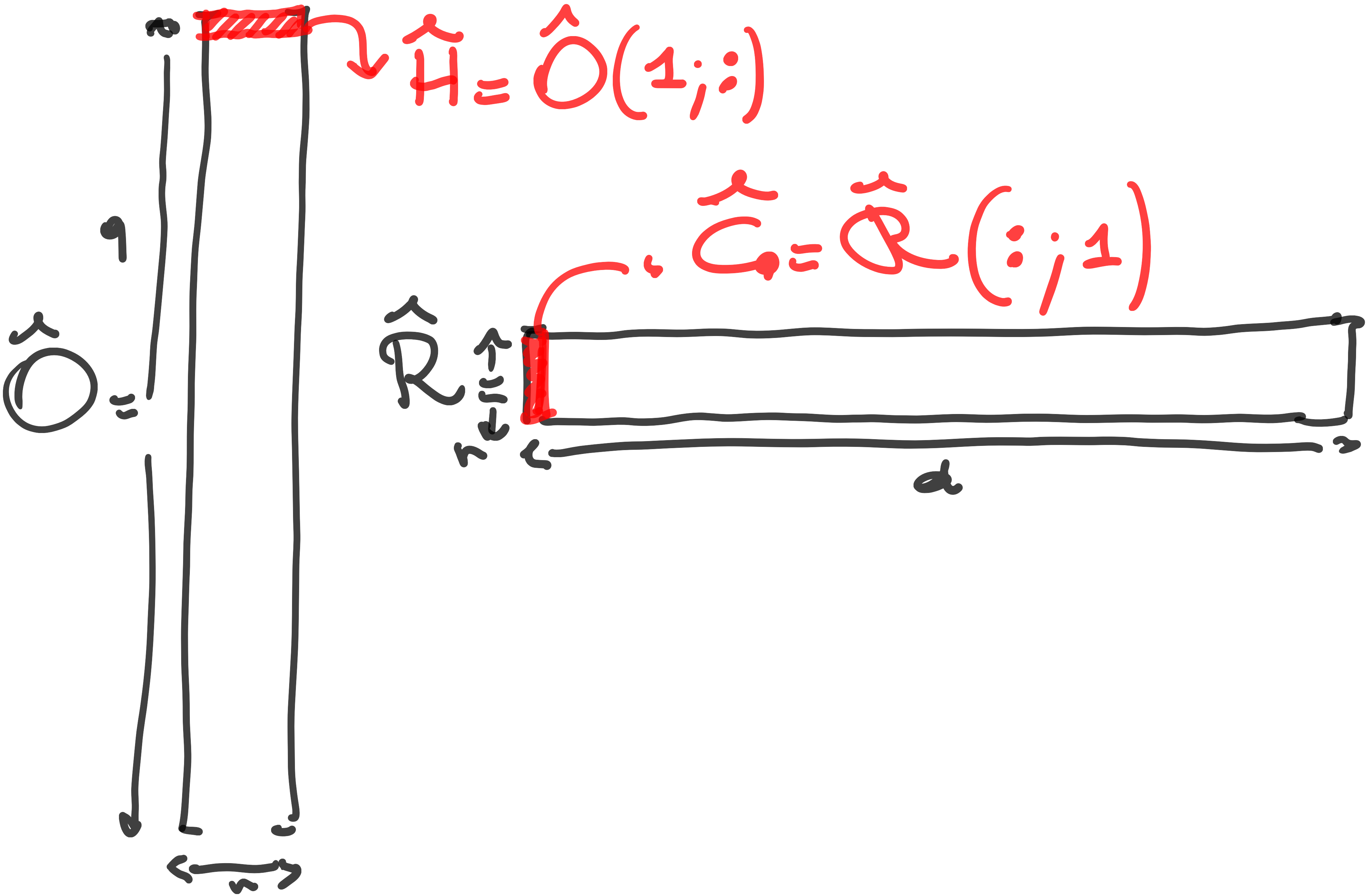
$$\hat{S}^{1/2} = \begin{bmatrix} \sqrt{\sigma_1} & & & \\ & \sqrt{\sigma_2} & & \\ & & \ddots & \\ & & & \sqrt{\sigma_n} \end{bmatrix}$$

DEF:

$$\left.\begin{aligned} \hat{O} &= \hat{U}\hat{S}^{1/2} \\ \hat{Q} &= \hat{S}^{1/2}\hat{V}^T \end{aligned}\right\}$$

$\hat{H}_{qd} = \hat{O} \cdot \hat{R}$ → can consider/view $\hat{O}$ and $\hat{R}$ as the EXTENDED Observability and Reachability matrices of the system

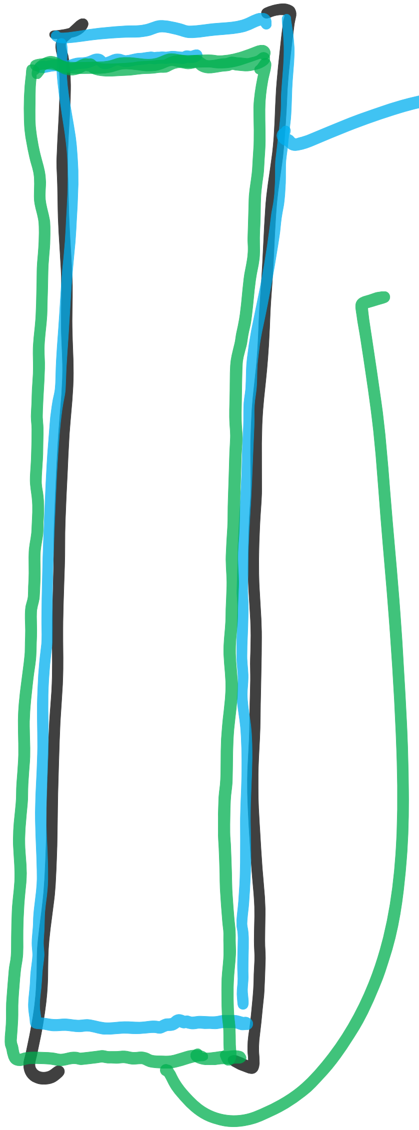
$$\underbrace{\hat{H}_{qd}}_{q\times d} = \underbrace{\hat{U}}_{q\times n}\ \underbrace{\hat{S}}_{n\times n}\ \underbrace{\hat{V}^T}_{n\times d}$$



$$\hat{H} = \hat{O}(1;:)$$

$$\hat{G} = \hat{R}(:;1)$$

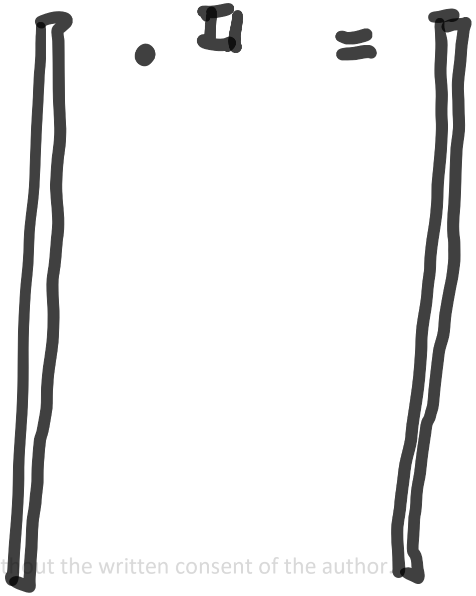
what about estimation of $\hat{F}$? → consider e.g. $\hat{O}$ and use the shift-invariance property:

$\hat{O} =$ 

$\hat{O}_1 = \hat{O}(1:q-1;:)$

$\hat{O}_2 = \hat{O}(2:q;:)$

S.I.P.: $\hat{O}_1 \cdot \hat{F} = \hat{O}_2$

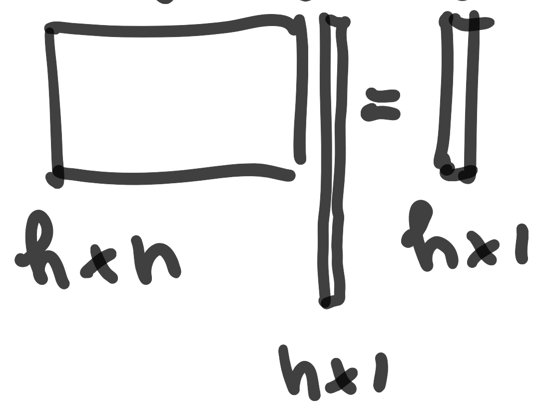


$\Rightarrow$ $\cancel{\hat{F} = \hat{O}_1^{-1} \hat{O}_2}$

$\hat{O}_1$ is NOT a square matrix

In this case we can use the approximate "LEAST SQUARE" solution of this linear system.

$Ax = B$ → 3 ≠ cases

1) 

$h < n$

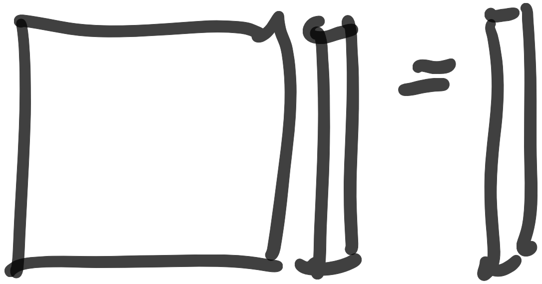
Number of equations

Number of unknowns

we have less equations than variables

system is "UNDER-DETERMINED"

$\Rightarrow \infty$ solutions

2)  $h = n \Rightarrow 1!$ solution (if A is invertible)

3)  $h > n \rightarrow$ more equations than VARIABLES $\Rightarrow$ system is OVER-DETERMINED $\rightarrow$ NO-SOLUTIONS

$\downarrow$ Approx. L.S. solution

$$Ax = B$$

$$\underbrace{A^T A}_{\text{square matrix}} x = A^T B \Rightarrow \hat{x} = \boxed{(A^T A)^{-1} A^T} B$$

$\downarrow$ A^+ called PSEUDO-INVERSE ("SURROGATE" of A^{-1}) when A is RECTANGULAR

Using this pseudo-inverse method $\downarrow$

$$\hat{O}_1 \cdot \hat{F} = \hat{O}_2$$

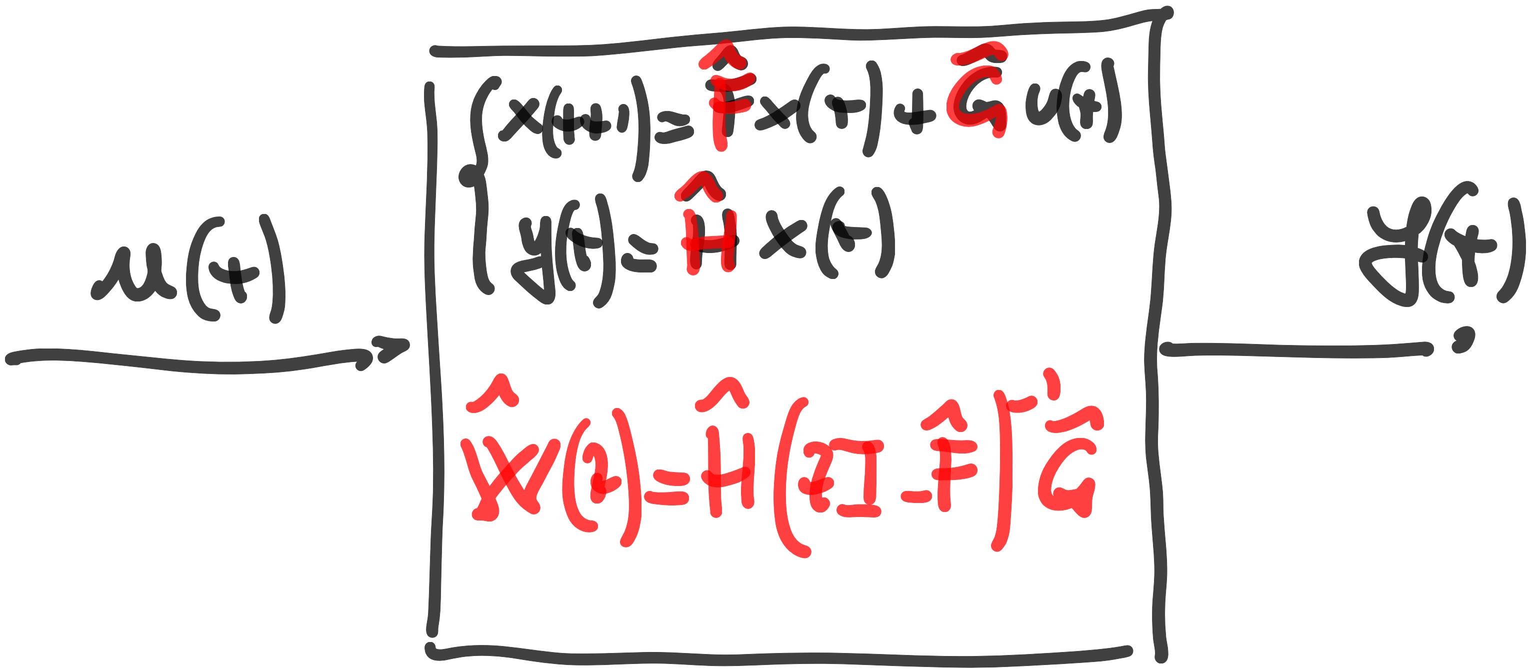
$$\underbrace{\hat{O}_1^T \hat{O}_1}_{\text{square}} \hat{F} = \hat{O}_1^T \hat{O}_2 \Rightarrow$$

$$\hat{F} = \boxed{\left(\hat{O}_1^T \hat{O}_1\right)^{-1} \hat{O}_1^T \hat{O}_2}$$

estimation of $\hat{F}$

// end

Conclusion → starting from a Noisy I.R. → $\{\omega(1), \omega(2) \dots \omega(N)\}$ we have estimated a model $\{\hat{F}, \hat{G}, \hat{H}\}$ in a non parametric / constructive way!



$$\begin{cases} x(t+1) = \hat{F}x(t) + \hat{G}u(t) \\ y(t) = \hat{H}x(t) \end{cases}$$

$$\hat{W}(z) = \hat{H}(zI - \hat{F})^{-1}\hat{G}$$

Remark → can be done something similar also if the measured input is generic (not impulse)

Remark: optimality of 4SID

the method is OPTIMAL in the sense that
it makes the BEST possible rank REDUCTION
of $\tilde{H}_{qd}$

$$\underset{\text{rank}:\ q}{\tilde{H}_{qd}} = \underset{n}{\hat{H}_{qd}} + \underset{q}{\tilde{H}_{res\,qd}}$$

In general there are ∞ ways to make a rank
reduction: ex:

$$\begin{bmatrix} 2 & 5 & 3 & 6 & 5 \\ 5 & 3 & 6 & 5 & 7 \\ 3 & 6 & 5 & 7 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}} + \begin{bmatrix} 1 & 5 & 3 & 6 & 5 \\ 5 & 2 & 6 & 5 & 7 \\ 3 & 6 & 5 & 7 & 1 \end{bmatrix}$$

↳ rank = 3

↑ we want a rank 2 matrix

↑ RESIDUAL MATRIX (RANK = 3)

is this an OPTIMAL rank reduction?
NO!!

GOAL: obtain the DESIRED RANK-REDUCTION by DISCARDING/TRASHING the MINIMUM AMOUNT of information CONTAINED in the ORIGINAL MATRIX

→ SVD makes exactly this → $\boxed{\tilde{H}_{RES_{qd}}}$ is the MINIMUM possible

→ IN the sense of FROBENIUS NORM $\| \cdot \|$

$$\left\| \tilde{H}_{RESqq} \right\|_F = \sqrt{\sum_{ij} \left(\tilde{H}^{(ij)}_{RESqq} \right)^2}$$

ex:

$$\left\| \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \right\|_F = \sqrt{1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2}$$

<u>Remark</u>: 4SID is a CONSTRUCTIVE method that can be implemented in a fully AUTOMATIC WAY, EXCEPT:

- (q and d selection (NOT CRITICAL))
- <u>choice</u> <u>of n</u> (<u>supervised by the designer</u>)
  - ↳ It can made AUTOMATIC using a <u>CROSS-VALIDATION METHOD</u>

SVD was an hystorical TURNING point in MACHINE-LEARNING ALGOS:

ALLOWS:

- VERY efficient COMPRESSION of INFORMATION
- VERY efficient SEPARATION of "IMPORTANT information" from NOISE

EXERCISE (similar to an exam ex.)

Consider the following S.S. model:

$$F = \underbrace{\begin{bmatrix} 1/2 & 0 \\ 1 & 1/4 \end{bmatrix}}_{2\times 2 \text{ matrix}} \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (D=0)$$

→ EIG(F) → 1/2 and 1/4

2 x 2 matrix

=> n = 2

1

INPUT

1

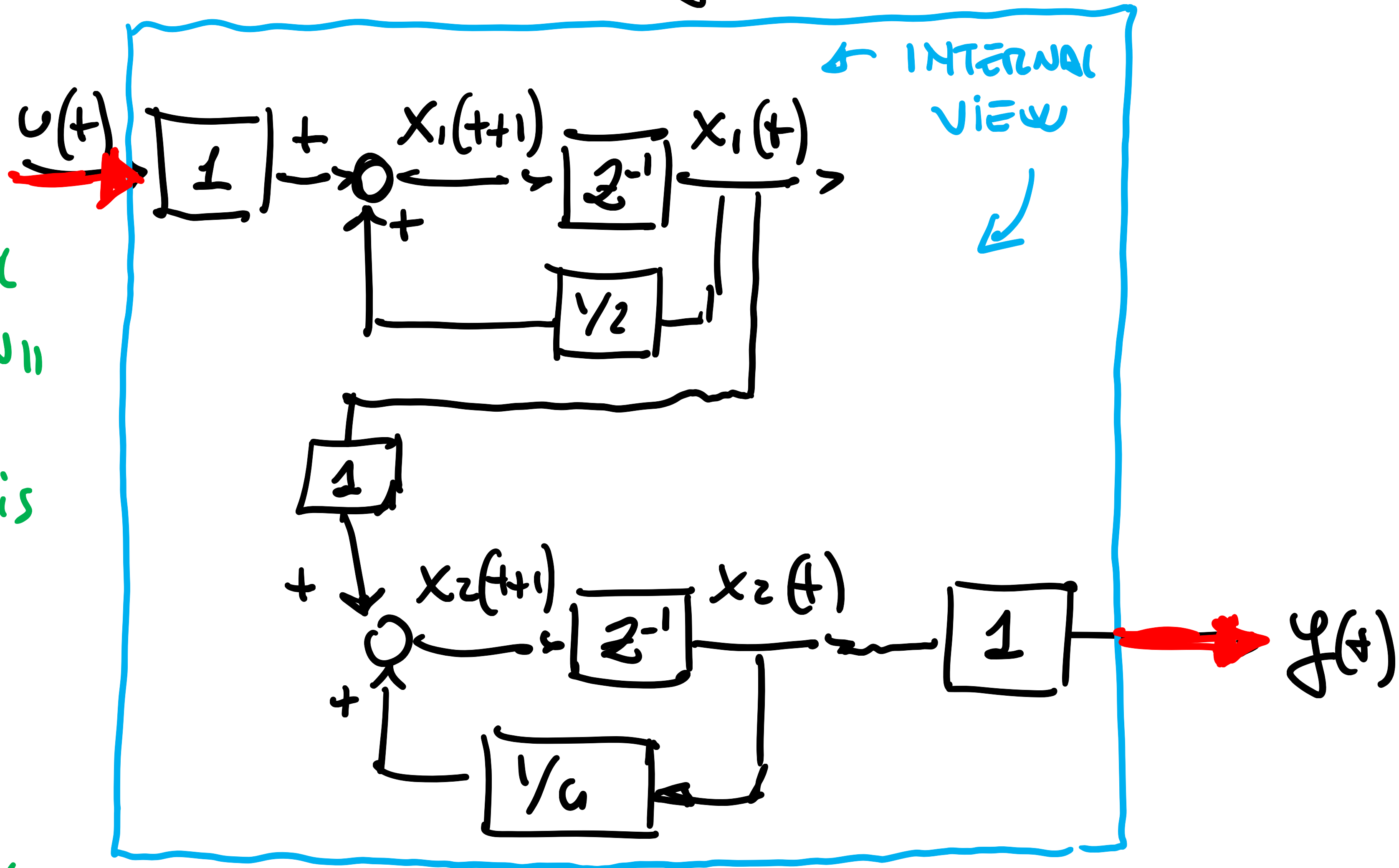
OUTPUT

SISO system / order n=2

Write time-domain equations of the system in S.S. rep.

$$\begin{cases} x_1(t+1) = \frac{1}{2}x_1(t) + \cancel{0}\,x_2(t) + 1\cdot u(t) \\ x_2(t+1) = 1\cdot x_1(t) + \frac{1}{4}x_2(t) + \cancel{0}\,u(t) \\ y(t) = \cancel{0}\,x_1(t) + 1\,x_2(t) \end{cases}$$

Write the Block-scheme of the S.S. Rep. of system



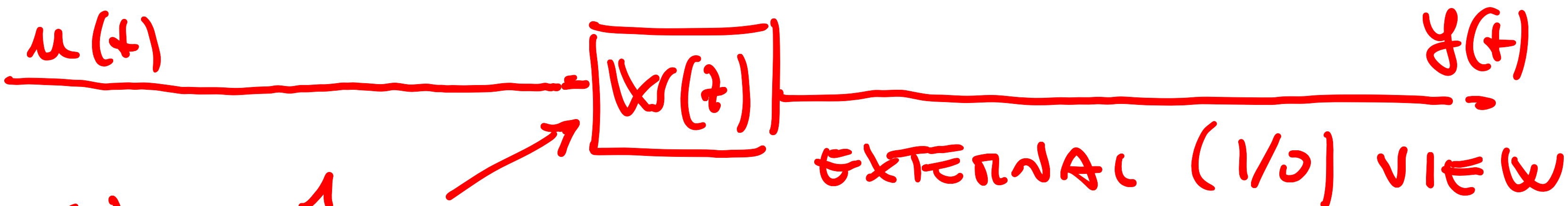
By "visual inspection"

⇓

system is fully obs.

AND

fully controll.



$$W(z) = \frac{1}{(z-\frac{1}{2})(z-\frac{1}{4})}$$

Formal verification that system is fully $\mathcal{O}$ and C.

$$\mathcal{O} = \begin{bmatrix} H \\ HF \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1/4 \end{bmatrix} \rightarrow \text{rank} = 2 = n \Rightarrow \text{fully } \mathcal{O}$$

$$\mathcal{R} = \begin{bmatrix} G & FG \end{bmatrix} = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \rightarrow \text{rank} = 2 = n \Rightarrow \text{fully C.}$$

<u>EXTENDED</u> ("$n+1$") $\mathcal{O}$ and $\mathcal{R}$ matrices:

$$\mathcal{O}_3 = \begin{bmatrix} H \\ HF \\ HF^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1/4 \\ 3/4 & 1/6 \end{bmatrix} \qquad \mathcal{R}_3 = \begin{bmatrix} 1 & 1/2 & 1/4 \\ 0 & 1 & 3/4 \end{bmatrix}$$

Compute the corresponding T.F. $\Rightarrow$ poles.

1st method $\rightarrow$ direct manipulation of S.S. Eq.

$$\begin{cases} \boxed{x_1(t+1) = \frac{1}{2} x_1(t) + u(t)} \\ \boxed{x_2(t+1) = x_1(t) + \frac{1}{4} x_2(t)} \\ \boxed{y(t) = x_2(t)} \end{cases}$$

$\rightarrow z x_1(t) - \frac{1}{2} x_1(t) = u(t) \Rightarrow x_1(t) = \frac{1}{\left(z - \frac{1}{2}\right)} u(t)$

$\rightarrow z x_2(t) - \frac{1}{4} x_2(t) = \frac{1}{z - 1/2} u(t)$

$$x_2(t) = \frac{1}{\left(z - \frac{1}{4}\right)\left(z - 1/2\right)} u(t)$$

$$y(t) = \boxed{\frac{1}{\left(z - \frac{1}{4}\right)\left(z - \frac{1}{2}\right)}} u(t)$$

$W(z)$

$\rightarrow$ 2 poles $z = 1/4$, $z = 1/2$ $\Rightarrow$ SAME AS EIG. of F

2nd method → FORMULA:

$$W(z) = H(zI - F)^{-1}G =$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \left(\begin{bmatrix} z-\frac{1}{2} & 0 \\ -1 & z-\frac{1}{4} \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \dots = \frac{1}{(z-\frac{1}{2})(z-\frac{1}{4})}$$

→ NO ZEROS

→ 2 poles

⇒ REL. DEGREE is 2

WRITE I/O Time-domain representation

$$y(t) = \frac{1}{z^2 - \frac{3}{4}z + \frac{1}{8}} u(t) \rightarrow y(t) = \frac{z^{-2}}{1 - \frac{3}{4}z^{-1} + \frac{1}{8}z^{-2}} u(t)$$

$$\boxed{y(t) = \frac{3}{4}y(t-1) - \frac{1}{8}y(t-2) + u(t-2)}$$

Difference eq. (I/O)

Compute the first 6 VALUES (including $\omega(0)$) of I.R.

$\omega(0), \omega(1), \omega(2), \omega(3), \omega(4), \omega(5)$

we decide to compute from T.F.:

$$\frac{z^{-2}}{1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}}$$

LONG DIVISION:

$$
\begin{array}{l|l}
z^{-2} & 1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2} \\
\hline
-z^{-2}+\frac{3}{4}z^{-3}-\frac{1}{8}z^{-4} & \boxed{z^{-2}+\frac{3}{4}z^{-3}+\frac{7}{16}z^{-4}+\frac{15}{64}z^{-5}} \\
\hline
// \quad \frac{3}{4}z^{-3}-\frac{1}{8}z^{-4} & \\
\vdots &
\end{array}
$$

$\omega(0)=0$ $\omega(1)=0$, $\omega(2)=1$, $\omega(3)=\frac{3}{4}$, $\omega(4)=\frac{7}{16}$, $\omega(5)=\frac{15}{64}$

BUILD Hankel matrix and stop when RANK is not full →

$$H_1 = [0]$$

$$H_2 = \begin{bmatrix} 0 & 1 \\ 1 & 3/4 \end{bmatrix} \rightarrow \text{RANK} = 2 \checkmark$$

$$H_3 = \begin{bmatrix} 0 & 1 & 3/4 \\ 1 & 3/4 & 7/16 \\ 3/4 & 7/16 & 15/64 \end{bmatrix} \rightarrow \text{RANK} = 2 \times$$

NOT full RANK ⇒ ORDER of system $\underline{n = 2}$

<u>Notice that</u>: $\boxed{O_3 \cdot R_3 = H_3}$

END chapter 1