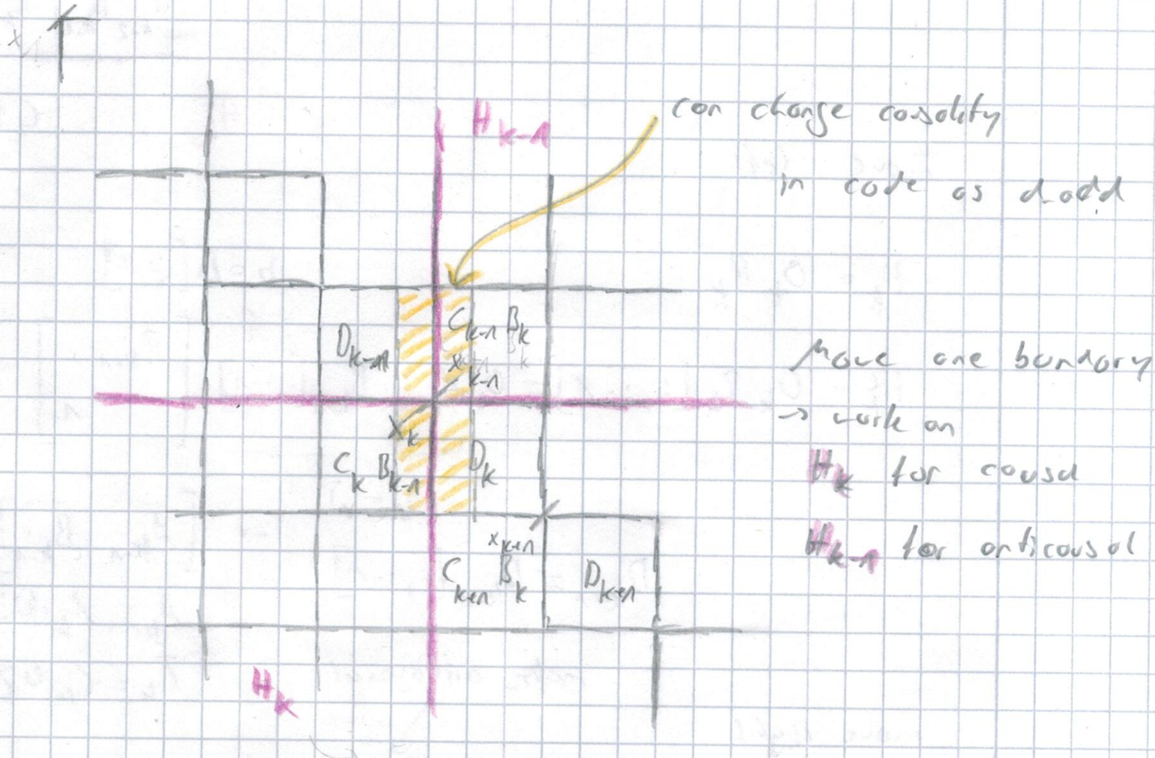

Indeces based on causal system ⇔ correlate with x_{k_S}

loop $k = 2, \ldots, N$ (1, len   range(1, len(stages))

causal system → output normal at start
→ convert to input normal

anticausal system → input normal at start
→ convert to output normal

move right

$$H_k^R = \begin{bmatrix} & 1 \\ O_{k+1} & \end{bmatrix} \begin{bmatrix} A_k & b \\ C_k & d \end{bmatrix} \begin{bmatrix} A_{k-1} & B_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{k-1} & \\ & I \end{bmatrix}$$

$$= \begin{bmatrix} & 1 \\ O_{k+1} & \end{bmatrix} \begin{bmatrix} A_k & b \\ C_k & d \end{bmatrix} \begin{bmatrix} R_k & \\ & 1 \end{bmatrix} \qquad b = B_k[:,0] \quad d = D_k[:,0]$$

$$U \Sigma V^T = \begin{bmatrix} A_k & b \\ C_k & d \end{bmatrix}$$

needed to keep $R_k^T R_k = 1$ and ordered

$$\begin{bmatrix} A_k^R \\ C_k^R \end{bmatrix} = U \Sigma \qquad B_k^R = B_k[:,1:] \quad D_k^R = D_k[:,1:]$$

$$A_{k-1}^R = V^T \begin{bmatrix} A_{k-1} \\ 0 \end{bmatrix} \qquad B_{k-1}^R = V^T \begin{bmatrix} B_{k-1} & 0 \\ 0 & 1 \end{bmatrix}$$

$$C_{k-1}^R = C_{k-1}^R \qquad D_{k-1}^R = \begin{bmatrix} D_{k-1} & d_{\text{onbreak}} \end{bmatrix}$$

Σ is Σ of H_k if $O_{k+1}^T O_{k+1} = 1$ and $R_k R_k^T = 1$

$$\underbrace{\begin{bmatrix} & 1 \\ O_{k+1} & \end{bmatrix} U}_{U'} \Sigma \underbrace{V^T \begin{bmatrix} R_k & \\ & 1 \end{bmatrix}}_{V'^T}$$

$$R_k^R = V R_k \Rightarrow R_k^R R_k^{R^T} = V R_k R_k^T V^T = V V^T = 1$$

move bounds of H_k

move left

$$H_k^L = O_k R_k [:, :-1] = O_k [A_{k-1} \; B_{k-1} [:, :-1]] \begin{bmatrix} R_{k-1} & \\ & 1 \end{bmatrix}$$

$b = B_{k-1}[:, -1]$

$U \Sigma V^T = [A_{k-1} \; B_{k-1} [:, :-1]]$ -> state transform for x_k

$$\left[\begin{array}{l} [A_{k-1}^L \; B_{k-1}^L] = V^T \\ C_{k-1}^{k-1} = C_{k-1}^{k-1} \quad D_{k-1}^L = D_{k-1}[:, :-1] \end{array} \right.$$

$$\left[\begin{array}{l} A_k^L = A_k U \Sigma \quad B_k^L = [A_k b \; B_k] \\ C_{k-1}^L = C_k U \Sigma \quad D_k = [C_k b \; D_k] \end{array} \right.$$

? old here as x_{k+1} is not transform

Σ is Σ of H_k if $O_k O_k^T = 1$ and $R_{k-1}^T R_{k-1} = 1$

then $\underbrace{O_k U}_{U^L} \Sigma \underbrace{V^T \begin{bmatrix} R_{k-1} & \\ & 1 \end{bmatrix}}_{V^{T L}}$

following: $R_k^T R_k = 1$

Move r/c anticausal part

→ due to inheriting consistency we will work on H_{k-1}

Prerequisite: $O_{k-2}^T O_{k-2} = I$ $\Rightarrow$ later $O_{k-1}^T O_{k-1} = I$

$R_{k-1} R_{k-1}^T = I$

Move left → add collumn

added input to added state

$$\tilde{H}_{k-1} = \tilde{O}_{k-1} \tilde{R}_{k-1} = \underbrace{\begin{bmatrix} O_{k-2} & \\ & 1 \end{bmatrix}}_{\perp} \underbrace{\begin{bmatrix} b & A_{k-1} \\ d & C_{k-1} \end{bmatrix}}_{U \Sigma V^T} \begin{bmatrix} 1 & \\ & R_{k-1} \end{bmatrix}$$

old input old state

$$\underbrace{\perp \quad U}_{\perp}$$

$b = B_{i-1}[:,0]$

$d = D_{i-1}[:,0]$

~~$b =$~~

~~No move~~

No move → basically make output normal

$$H_{k-1} = O_{k-1} R_{k-1} = \begin{bmatrix} O_{k-2} & \\ & 1 \end{bmatrix} \underbrace{\begin{bmatrix} A_{k-1} \\ C_{k-1} \end{bmatrix}}_{U \Sigma V^T} \begin{bmatrix} R_{k-1} \end{bmatrix}$$

Move right → base on no move as we need $O_{k-1}^T O_{k-1} = I$

$$\hat{H}_{k-1} = \hat{O}_{k-1} \hat{R}_{k-1} = O_{k-1} R_{k-1}[:,1:] = O_{k-1} \underbrace{\begin{bmatrix} B_{k-1}[:] & A_k \end{bmatrix}}_{U \Sigma V^T} \begin{bmatrix} 1 & \\ & R_k \end{bmatrix}$$

Objective function for matrices

1) Approach A:

$\|A\| = tr(A^T A) = tr(V \Sigma U^T U \Sigma V^T)$

$= tr(\Sigma \Sigma) = \sum_{is} \sigma_i^2$

Decrease the approximation error

→ cost function based on the norm of the error

$\|E\| = \sum_{i \text{ is cut}} \sigma_i^2$ ← question where to cut

To decouple it from the matrix size
→ scale for matrix size

For case where stages are similar this also reduces / keeps cost
⇒ Problem: this is not the case for systems

2) Approach B:

Fit some bound on σ_s cut and optimize for the cost

cut all i with $\sigma_i < \varepsilon$
this gives a new n_i
⇒ Compute cost of system

Gives also a upper bound on approximation error

$$\|E\| = \sum_{i \text{ s cut}} \sigma_i^2 < \sum_{i \text{ s cut}} \varepsilon^2 = \underbrace{(n_i - n_{i\,\text{approx}})}_{\#\text{ of } \sigma_s \text{ cut}} \varepsilon^2$$

depends on the input / output dims
→ adversarial to cost
suggests to use a soft thresholding