Selected applications in Telecommunications, system modeling, control

Patrick Dewilde

TUM Inst. of Advanced Study

Content

- WCDMA Decorrelation
- 3D-Modelling
- Flexible control

WCDMA Decorrelation

Ref.: "Blind decorrelating Rake receivers for long code WCDMA", Lang Tong, Alle-Jan van der Veen, P.D. and Youngchul Sung, IEEE Trans. on SP, April 2002, pp. 1-11
to be found on PDPublic/tong-ea.pdf

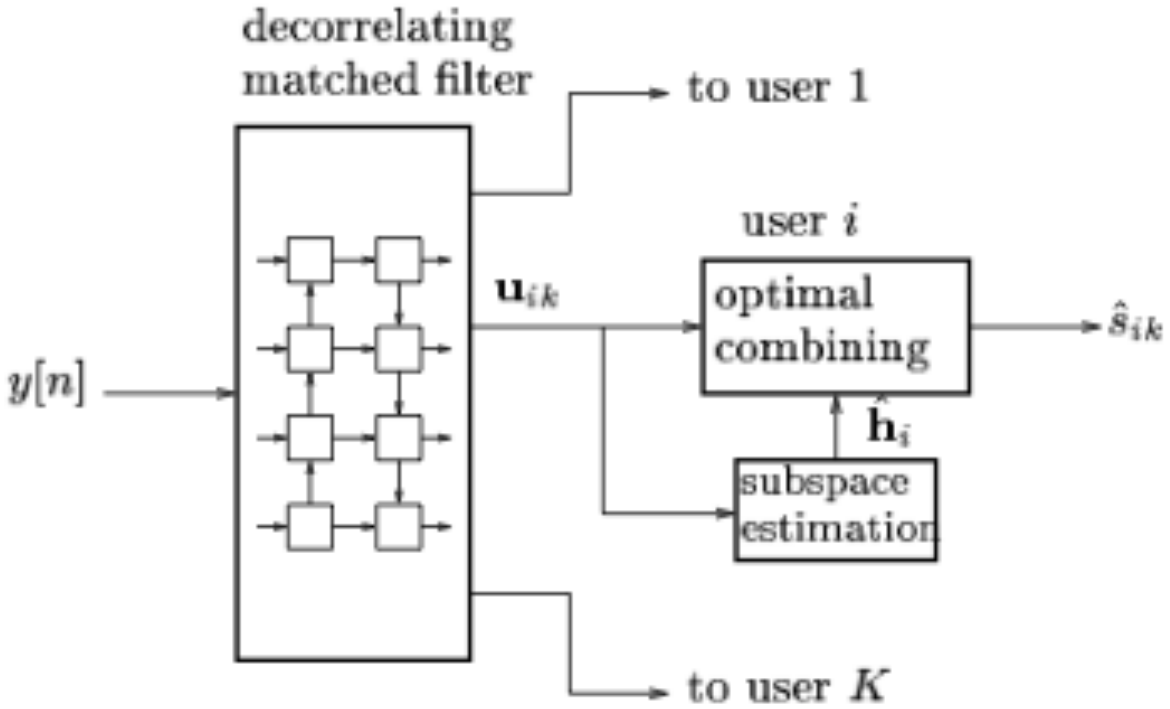


Fig. 1. Receiver structure. The decorrelating matched filter is implemented by an efficient state-space realization.

Code matrix structure

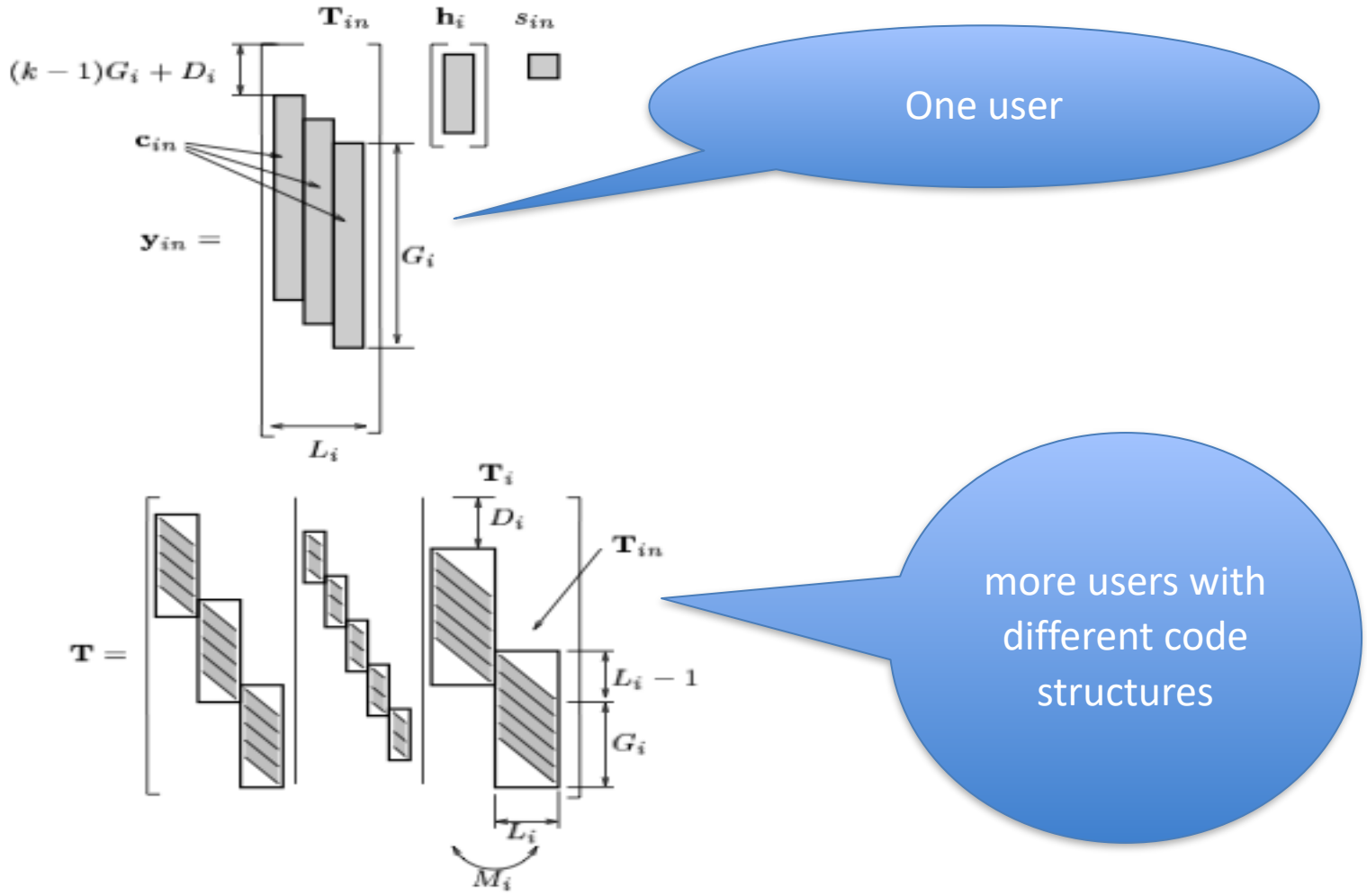


Fig. 2. (a) Structure of $\mathbf{T}_{in}$; (b) Structure of the code matrix $\mathbf{T}$.

Code inversion

Matched code inversion: $\widehat{y} = T'u$

efficient because *T* is extremely sparse, but only correct when $T'T = I$!

But what if the "fingers" are not orthogonal? Is there an efficient way to compute the Moore-Penrose inverse:

$$\widehat{y} = T^{\dagger}u$$

(one may also have to take care of colored noise, but that would not affect the algorithm much – Moore-Penrose minimizes the estimation error, like in the Kalman filter)?

Assume *T* has full column rank, then a QR factorization should do:

$$T = QR \longrightarrow T^{\dagger} = R^{-1}Q'$$

the issue is to compute Q and R efficiently

Building the state space model

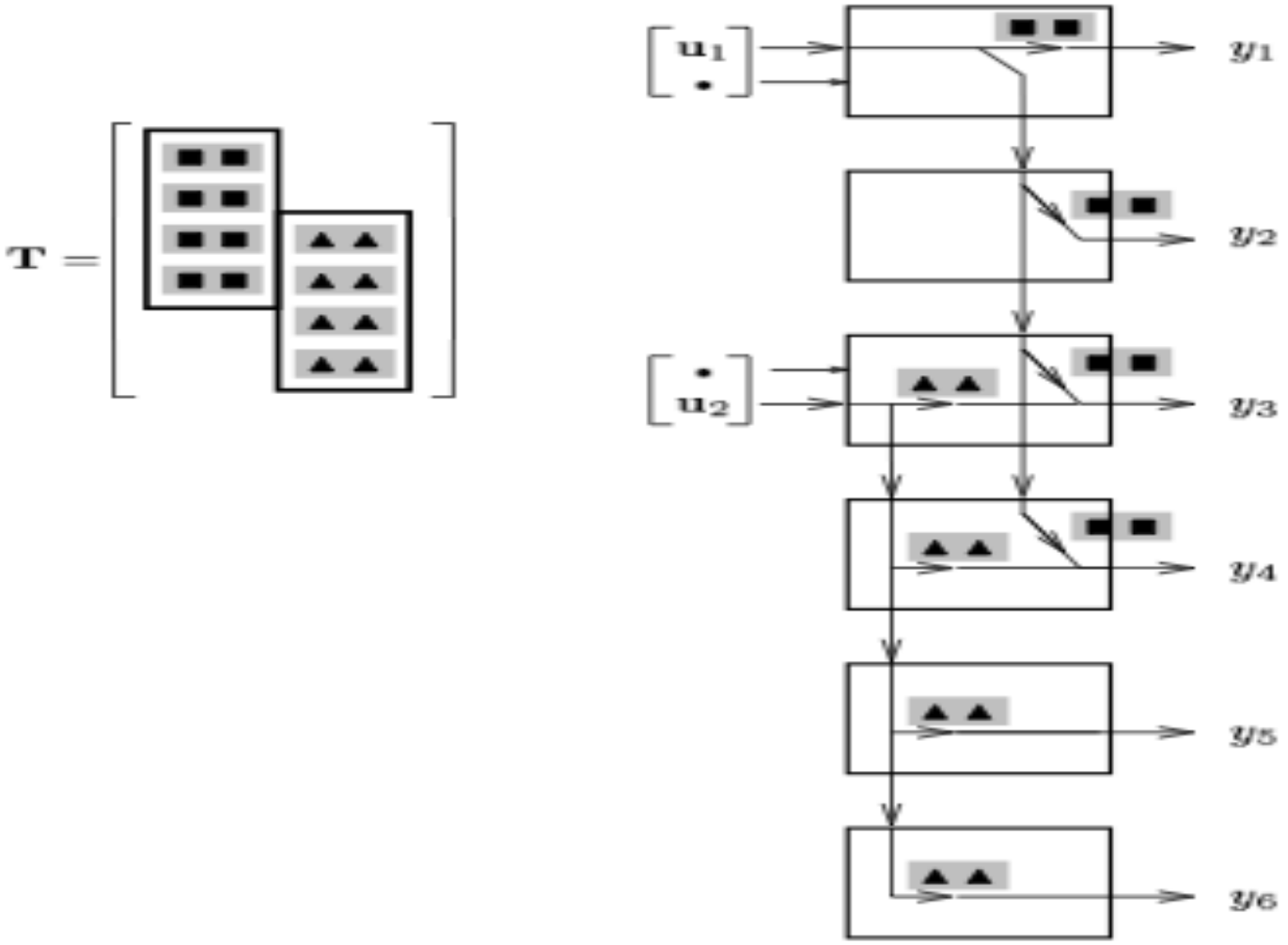


Fig. 5. Computational network for $\mathbf{T} = [\mathbf{T}^{(1)} \ \mathbf{T}^{(2)}]$.

QR = Inner-Outer!

$$\mathbf{T} = \tilde{\mathbf{Q}}_N \cdots \tilde{\mathbf{Q}}_1 \tilde{\mathbf{R}}_N \cdots \tilde{\mathbf{R}}_1$$

$$\mathbf{T}^\dagger = \tilde{\mathbf{S}}_N \cdots \tilde{\mathbf{S}}_1 \tilde{\mathbf{Q}}_1^H \cdots \tilde{\mathbf{Q}}_N^H$$

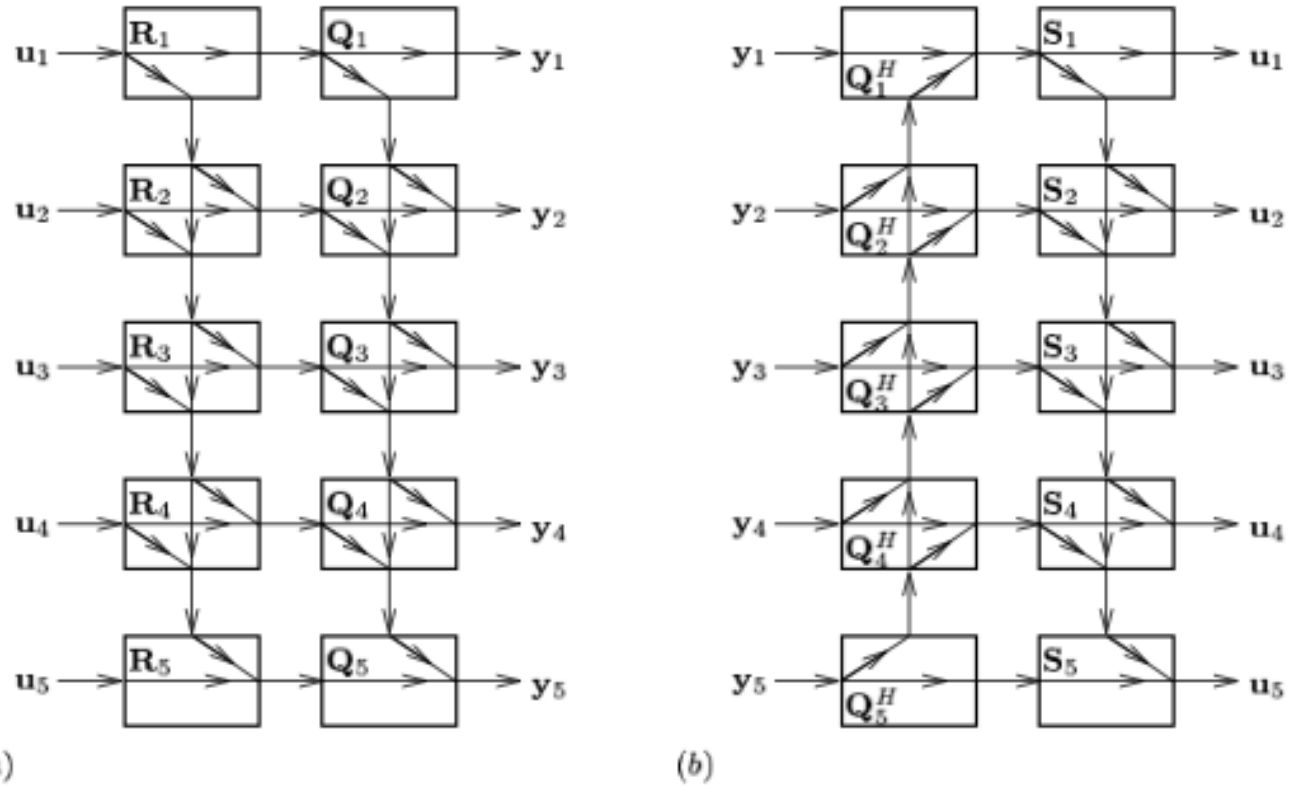


Fig. 6. Inversion. (a) Structure of the QR factorization, (b) structure of the inverse. Note that the inverse is not causal.

$$\begin{bmatrix} \mathbf{Y}_{n+1}\mathbf{A}_n & \mathbf{Y}_{n+1}\mathbf{B}_n \\ \mathbf{C}_n & \mathbf{D}_n \end{bmatrix} \triangleq \mathbf{Q}_n \mathbf{L}_n \triangleq \begin{bmatrix} \mathbf{A}_n^Q & \mathbf{B}_n^Q \\ \mathbf{C}_n^Q & \mathbf{D}_n^Q \end{bmatrix} \begin{bmatrix} \mathbf{Y}_n & 0 \\ \mathbf{C}_n^R & \mathbf{D}_n^R \end{bmatrix} \qquad \mathbf{S}_n = \begin{bmatrix} \mathbf{A}_n^R - \mathbf{B}_n^R \mathbf{D}_n^{R-1} \mathbf{C}_n^R & \mathbf{B}_n^R \mathbf{D}_n^{R-1} \\ -\mathbf{D}_n^{R-1} \mathbf{C}_n^R & \mathbf{D}_n^{R-1} \end{bmatrix} \tag{11}$$

Result

- The optimal decorrelator has the same computational complexity as the original (sparse) matrix would have, and same as the matched filter;
- As expected, it is non-causal, but the outer factor has a causal inverse, and the adjoint of the inner factor does the mixing;
- The system can be used in a variety of circumstances (including the use of pilot information i.e. learning).

Control: spatial semi-separable

Ref.: J.K.Rice and M. Verhaegen, "Distributed Control: a Sequentially Semi-Separable Approach for Spatially Heterogeneous Linear Systems", IEEE Trans. on AC, Vol. 54, 2009, pp. 1270-1283
to be found on PDPublic/RiceVerhaegen

$$\underbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_1 & B_1^pC_2^p & B_1^pW_2^pC_3^p & B_1^pW_2^pW_3^pC_4^p & B_1^pW_2^pW_3^pW_4^pC_5^p \\ B_2^mC_1^m & A_2 & B_2^pC_3^p & B_2^pW_3^pC_4^p & B_2^pW_3^pW_4^pC_5^p \\ B_3^mW_2^mC_1^m & B_3^mC_2^m & A_3 & B_3^pC_4^p & B_3^pW_4^pC_5^p \\ B_4^mW_3^mW_2^mC_1^m & B_4^mW_3^mC_2^m & B_4^mC_3^m & A_4 & B_4^pC_5^p \\ B_5^mW_4^mW_3^mW_2^mC_1^m & B_5^mW_4^mW_3^mC_2^m & B_5^mW_4^mC_3^m & B_5^mC_4^m & A_5 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix}}_{x} \quad (1)$$

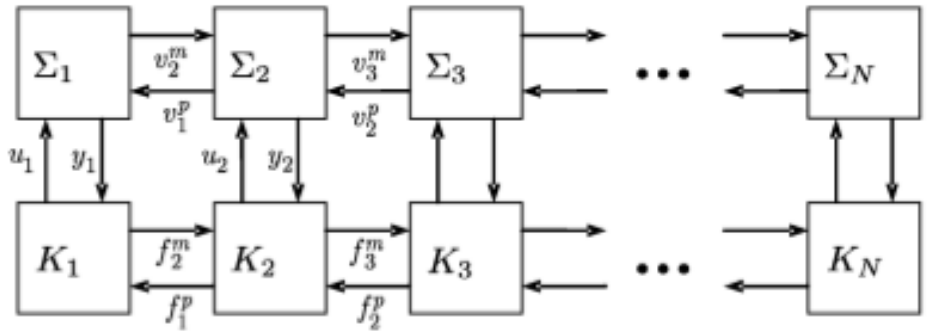


Fig. 2. Distributed controller implementation.

the Rice-Verhaegen solution

Apply to an appropriate "Hamiltonian matrix":

Algorithm 1: Sign Iteration [41]

$$
\begin{aligned}
Z_0 &= X \\
&\text{for k} = 0, 1, 2, \ldots \\
Z_{k+1} &= \frac{1}{2}\left(Z_k + Z_k^{-1}\right) \\
&\text{end} \\
sign(X) &= \lim_{k \to \infty} Z_k
\end{aligned}
$$

in which the entries are quasi-separable and use efficient inverses and additions (and approximations)

to obtain a distributed controller as promised.

The next step...

- Things get out of hand when more than one hierarchical level is involved ("symbolically semi-separable systems).
- I illustrate this with a 2D and a 3D finite difference example, and give some indication about what to do in those cases.

Level 3 example: Horn-Schunck image flow analysis

$$T = \begin{bmatrix} C & R \\ L & C \end{bmatrix} \qquad C = \begin{bmatrix} \boxed{M} & \boxed{-U} & & & & \\ \boxed{-U} & \boxed{M} & \boxed{-U} & & & \\ & \boxed{-U} & \boxed{M} & \boxed{-U} & & \\ & & \boxed{-U} & \ddots & \ddots & \\ & & & \ddots & \ddots & \boxed{-U} \\ & & & & \boxed{-U} & \boxed{M} \end{bmatrix}$$

- R and L purely diagonal (data dependent)
- size $n^2 \times n^2$, $n \approx 1000$
- *C* is the 2D Laplacian (stencil)

$$M = \begin{bmatrix} 1 & -\frac{1}{6} & & \\ -\frac{1}{6} & 1 & -\frac{1}{6} & \\ & -\frac{1}{6} & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}, \; U = \frac{1}{12} \begin{bmatrix} 2 & 1 & & \\ 1 & 2 & 1 & \\ & 1 & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

Optic Flow: Horn-Schunck algorithm

$$\left(\alpha^2 \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} + \begin{bmatrix} \mathbf{I}_x \\ \mathbf{I}_y \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix}\right) \cdot \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \mathbf{I}_x \\ \mathbf{I}_y \end{bmatrix} \cdot \mathbf{I}_t$$

in which the **I**'s are pixel intensities (diagonal matrices), the **v**'s are the vector field to be computed, and α is a relaxation parameter - C is a Laplacian.

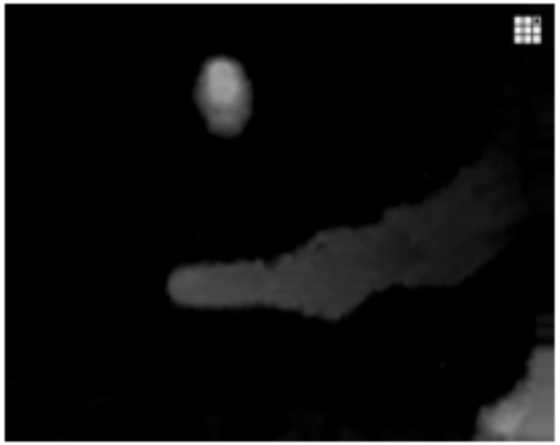


Image flow analysis: tridiagonal block tridiagonal matrices

prototype: $$T = Z^* L + M + UZ$$

with L,M,U themselves tridiagonal matrices

naïve Cholesky factorization: $$T = (Z^* l + \delta)(\varepsilon + uZ)$$

$$\begin{cases} L = l\varepsilon \\ M = \delta\varepsilon + (lu)^{(1)} \\ U = \delta u \end{cases}$$

or

$$\text{STEP 1: } M_0 = \delta_0\varepsilon_0,\ l_0 = L_0\varepsilon_0^{-1},\ u_0 = \delta_0^{-1}U_0.$$

$$\text{GENERIC STEP: } \begin{cases} \delta_k\varepsilon_k = M_k - l_{k-1}u_{k-1} \\ l_k = L_k\varepsilon_k^{-1} \\ u_k = \delta_k^{-1}U_k \end{cases}$$

the recursion kills the lower level semi-separable structure!
solution: model reduction.

Horn-Schunck inversion of C

sizes: n by n blocks of size n by n, n = 1000 e.g.

We can write $C=LL^*$, with L given by

$$L = \begin{bmatrix} m_1^* & & & \\ -Um_1^{-1} & m_2^* & & \\ & -Um_2^{-1} & m_3^* & \\ & & \ddots & \ddots \end{bmatrix}$$

in which

$$\begin{cases} m_i^* m_i &= \mu_i \\ \mu_{i+1} &= M - U\mu_i^{-1}U \end{cases}$$

Riccati recursion!

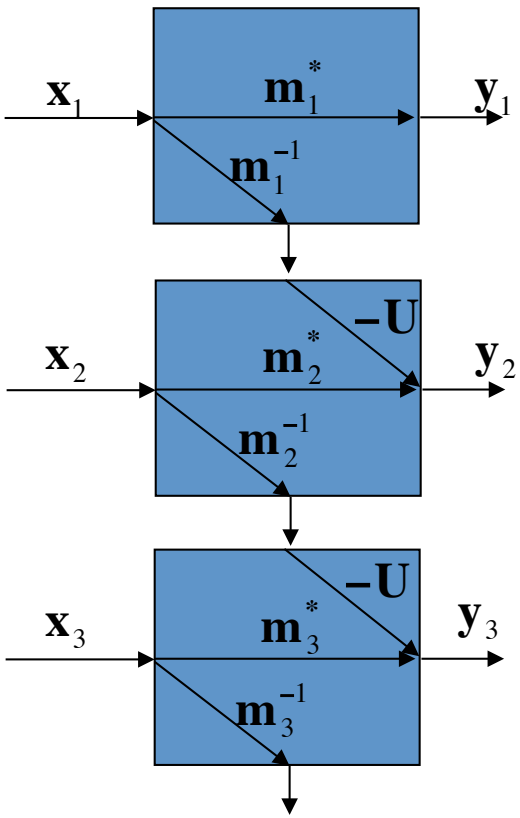
Inverse in the top hierarchy

$$L^{-*} = \begin{bmatrix} m_1^{-1} & \mu_1^{-1} U m_2^{-1} & \mu_1^{-1} U \mu_2^{-1} U m_3^{-1} & \cdots \\ & m_2^{-1} & \mu_2^{-1} U m_3^{-1} & \cdots \\ & & m_3^{-1} & \cdots \\ & & & \ddots \end{bmatrix}$$

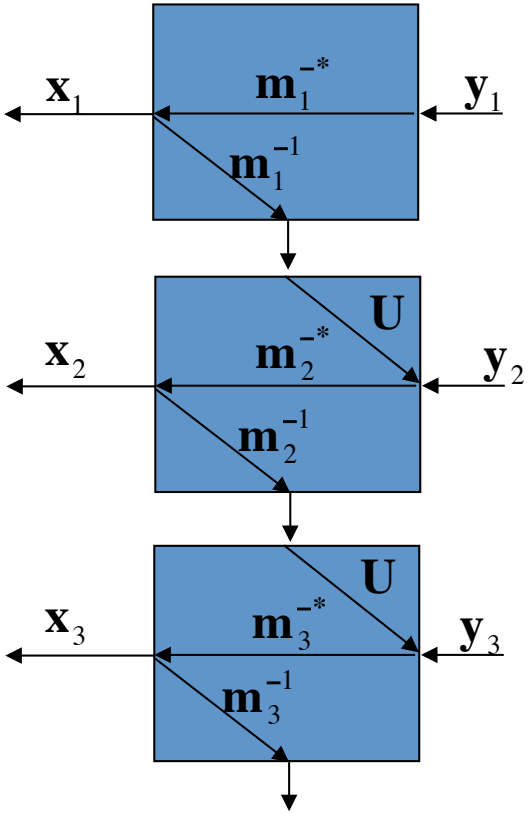
Hierarchically semi-separable = NOT SS, but almost?

$$u_1 = x_1;$$
$$\text{for } i = 2 \text{ step } 1 \text{ to } n \text{ do } u_i = x_i + U \mu_{i-1}^{-1} u_{i-1};$$
$$y_n = \mu_n^{-1} u_n;$$
$$\text{for } i = n-1 \text{ step } -1 \text{ to } 1 \text{ do } y_i = \mu_i^{-1}(u_i + U y_{i+1}).$$

nonetheless efficient at this level!



$$\mathbf{y = Lx}$$



$$\mathbf{x = L^{-1}y}$$

the problem is the next level

A number of issues arise:

- the inversion of the "pivot" should be avoided in favor a numerically stable method;
- the computational complexity has to be reduced at the next level. This can be done either by adopting an iterative algorithm (as in the paper of Rice and Verhaegen), or by doing model reduction.

In the next set of lectures we shall move to these issues, and show how our fundamental insights provides for new ways of tackling them.

Concluding remark on the first set of lectures

My goal has been to show to you how "fundamental thinking" on the central objects in Dynamical System Theory

- Past, Present, Future (Nerode equivalence and Hankel operators)
- the State
- Reachability (Controllability) and Observability

leads to the most powerful results,

when combined with orthodox methods in Matrix Algebra.

this is the highway from insight to realization!