Two Double Recursive Block Macaulay Matrix Algorithms to Solve Multiparameter Eigenvalue Problems

61st Conference on Decision and Control

Christof Vermeersch and Bart De Moor

christof.vermeersch@esat.kuleuven.be

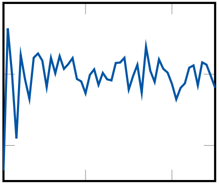
STADIUS
Center for Dynamical Systems, Signal Processing, and Data Analytics

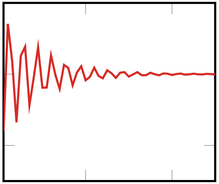
December 8, 2022

Solving the least-squares realization problem

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\text{known}} \longrightarrow \underbrace{\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix}}_{\text{unknown}}$$

such that $\hat{\boldsymbol{y}}_k = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{x}_0$ is the output of an nth-order autonomous system





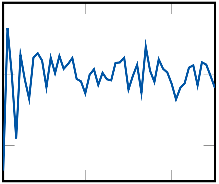
$$\min_{\hat{\boldsymbol{y}},\boldsymbol{\alpha}} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$$

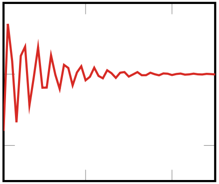
subject to $\boldsymbol{T}_\alpha \hat{\boldsymbol{y}} = \boldsymbol{0}$

(De Moor, 2019, 2020)

Solving the least-squares realization problem

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\text{known}} \longrightarrow \underbrace{\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{bmatrix}}_{\text{unknown}}$$





for example

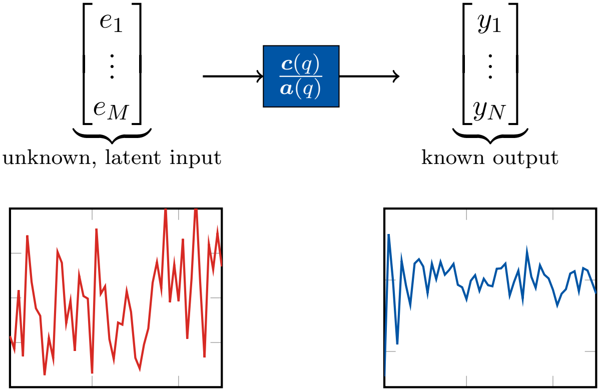
$$\boldsymbol{T}_\alpha = \begin{bmatrix} \alpha_2 & \alpha_1 & 1 & 0 & 0 & 0 \\ 0 & \alpha_2 & \alpha_1 & 1 & 0 & 0 \\ 0 & 0 & \alpha_2 & \alpha_1 & 1 & 0 \\ 0 & 0 & 0 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}$$

$$\min_{\hat{\boldsymbol{y}},\boldsymbol{\alpha}} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$$

$$\text{subject to } \boldsymbol{T}_\alpha \hat{\boldsymbol{y}} = \boldsymbol{0}$$

(De Moor, 2019, 2020)

Identifying the parameters of an ARMA model



such that $\sum_{i=0}^{n_a} \alpha_i y_{k-i} = \sum_{i=0}^{n_c} \gamma_i e_{k-i}$

$$\min_{\boldsymbol{e},\boldsymbol{\alpha},\boldsymbol{\gamma}} \|\boldsymbol{e}\|_2^2$$

subject to $\boldsymbol{T}_\alpha \boldsymbol{y} = \boldsymbol{T}_\gamma \boldsymbol{e}$

(Vermeersch and De Moor, 2019)

Common problem!

These two system identification problems have one thing in common: they are **multiparameter eigenvalue problems**

$$\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\, \boldsymbol{z} = \mathbf{0}$$

But there are also other problems that fit into this framework (solving partial differential equations, vibration analysis, prediction error methods, etc.)

(Plestenjak et al., 2015; Batselier et al., 2012; Tisseur and Meerbergen, 2001)

Multiparameter eigenvalue problem

The **multiparameter eigenvalue problem** $\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\,\boldsymbol{z} = \boldsymbol{0}$ consists in finding all n-tuples $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \mathbb{C}^n$ and corresponding vectors $\boldsymbol{z} \in \mathbb{C}^{l \times 1} \setminus \{\boldsymbol{0}\}$, so that

$$\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\,\boldsymbol{z} = \left(\sum_{\{\boldsymbol{\omega}\}} \boldsymbol{A}_{\boldsymbol{\omega}} \boldsymbol{\lambda}^{\boldsymbol{\omega}} \right) \boldsymbol{z} = \boldsymbol{0},$$

with $\|\boldsymbol{z}\|_2 = 1$.

- $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ indexes the monomials $\boldsymbol{\lambda}^{\boldsymbol{\omega}}$ and coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}}$
- rectangular coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} = \boldsymbol{A}_{(\omega_1, \ldots, \omega_n)} \in \mathbb{R}^{k \times l}$ with $k \geq l + n - 1$
- example: $\left(\boldsymbol{A}_{000} + \boldsymbol{A}_{200}\lambda_1^2 + \boldsymbol{A}_{013}\lambda_2\lambda_3^3\right) \boldsymbol{z} = \boldsymbol{0}$

(Vermeersch and De Moor, 2019, 2022)

Outline

Outline

Block Macaulay matrix

$$\boldsymbol{\mathcal{M}}(\lambda, \mu)\, \boldsymbol{z} = (\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\lambda + \boldsymbol{A}_{01}\mu)\, \boldsymbol{z} = \mathbf{0}$$

$$\begin{matrix} \boldsymbol{\mathcal{M}}(\lambda) \\ \lambda\boldsymbol{\mathcal{M}}(\lambda) \\ \mu\boldsymbol{\mathcal{M}}(\lambda) \\ \lambda^2\boldsymbol{\mathcal{M}}(\lambda) \\ \lambda\mu\boldsymbol{\mathcal{M}}(\lambda) \\ \mu^2\boldsymbol{\mathcal{M}}(\lambda) \end{matrix} \begin{bmatrix} \boldsymbol{A}_{00} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}_{00} & \mathbf{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{00} & \mathbf{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{00} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{00} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{00} & \mathbf{0} & \mathbf{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} \end{bmatrix} \begin{bmatrix} \boldsymbol{z} \\ \lambda\boldsymbol{z} \\ \mu\boldsymbol{z} \\ \lambda^2\boldsymbol{z} \\ \lambda\mu\boldsymbol{z} \\ \lambda^3\boldsymbol{z} \\ \lambda^2\mu\boldsymbol{z} \\ \lambda\mu^2\boldsymbol{z} \\ \mu^3\boldsymbol{z} \end{bmatrix} = \mathbf{0}$$

use **block forward multi-shift recursions (block FmSRs)** to generate the block Macaulay matrix $\boldsymbol{M}$ from $\boldsymbol{\mathcal{M}}(\lambda)\, \boldsymbol{z}$

Multidimensional realization problem

Assume only simple and affine solutions

- Solutions generate vectors in the null space of block Macaulay matrix $\boldsymbol{M}$

$$\boldsymbol{MV} = \mathbf{0}$$

- Nullity corresponds to the number of solutions m_a
- Null space has a **block multi-shift-invariant** structure

block multivariate Vandermonde basis matrix

$$\boldsymbol{V} = \left[\begin{array}{ccc} \boldsymbol{z}|_{(1)} & \cdots & \boldsymbol{z}|_{(m_a)} \\ \hline (\lambda\boldsymbol{z})|_{(1)} & \cdots & (\lambda\boldsymbol{z})|_{(m_a)} \\ (\mu\boldsymbol{z})|_{(1)} & \cdots & (\mu\boldsymbol{z})|_{(m_a)} \\ \hline (\lambda^2\boldsymbol{z})|_{(1)} & \cdots & (\lambda^2\boldsymbol{z})|_{(m_a)} \\ (\lambda\mu\boldsymbol{z})|_{(1)} & \cdots & (\lambda\mu\boldsymbol{z})|_{(m_a)} \\ (\mu^2\boldsymbol{z})|_{(1)} & \cdots & (\mu^2\boldsymbol{z})|_{(m_a)} \\ \hline (\lambda^3\boldsymbol{z})|_{(1)} & \cdots & (\lambda^3\boldsymbol{z})|_{(m_a)} \\ \vdots & & \vdots \end{array}\right]$$

Multidimensional realization theory

$$\left[\begin{array}{ccc} \boldsymbol{z}|_{(1)} & \cdots & \boldsymbol{z}|_{(m_a)} \\ \hline (\lambda\boldsymbol{z})|_{(1)} & \cdots & (\lambda\boldsymbol{z})|_{(m_a)} \\ (\mu\boldsymbol{z})|_{(1)} & \cdots & (\mu\boldsymbol{z})|_{(m_a)} \\ \hline \left(\lambda^2\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\lambda^2\boldsymbol{z}\right)\Big|_{(m_a)} \\ (\lambda\mu\boldsymbol{z})|_{(1)} & \cdots & (\lambda\mu\boldsymbol{z})|_{(m_a)} \\ \left(\mu^2\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\mu^2\boldsymbol{z}\right)\Big|_{(m_a)} \\ \hline \left(\lambda^3\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\lambda^3\boldsymbol{z}\right)\Big|_{(m_a)} \\ \vdots & & \vdots \end{array}\right] \xrightarrow{\lambda} \left[\begin{array}{ccc} \boldsymbol{z}|_{(1)} & \cdots & \boldsymbol{z}|_{(m_a)} \\ \hline (\lambda\boldsymbol{z})|_{(1)} & \cdots & (\lambda\boldsymbol{z})|_{(m_a)} \\ (\mu\boldsymbol{z})|_{(1)} & \cdots & (\mu\boldsymbol{z})|_{(m_a)} \\ \hline \left(\lambda^2\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\lambda^2\boldsymbol{z}\right)\Big|_{(m_a)} \\ (\lambda\mu\boldsymbol{z})|_{(1)} & \cdots & (\lambda\mu\boldsymbol{z})|_{(m_a)} \\ \left(\mu^2\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\mu^2\boldsymbol{z}\right)\Big|_{(m_a)} \\ \hline \left(\lambda^3\boldsymbol{z}\right)\Big|_{(1)} & \cdots & \left(\lambda^3\boldsymbol{z}\right)\Big|_{(m_a)} \\ \vdots & & \vdots \end{array}\right]$$

$$\boldsymbol{S}_1\boldsymbol{V} \underbrace{\begin{bmatrix} \lambda|_{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda|_{(m_a)} \end{bmatrix}}_{\boldsymbol{D}_\lambda} = \boldsymbol{S}_\lambda\boldsymbol{V}$$

Numerical basis matrix for the null space

$$\boldsymbol{S}_1 \boldsymbol{V} \boldsymbol{D}_\lambda = \boldsymbol{S}_\lambda \boldsymbol{V}$$

for example, calculated via the SVD

- Solutions are not known in advance
- Consider a **numerical basis for the null space** $\boldsymbol{Z}$

$$\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{T}$$

non-singular matrix $\boldsymbol{T}$

- This results in a GEP in the shift λ,

$$(\boldsymbol{S}_1 \boldsymbol{Z}) \boldsymbol{T} \boldsymbol{D}_\lambda = (\boldsymbol{S}_\lambda \boldsymbol{Z}) \boldsymbol{T},$$

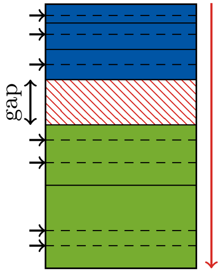
with the matrix of eigenvectors $\boldsymbol{T}$ and the diagonal matrix of eigenvalues $\boldsymbol{D}_\lambda$

Multiplicity and solutions at infinity

- **Multiple solutions** lead to a confluent block multivariate Vandermonde basis matrix and the Jordan normal form, but we can avoid this via multiple Schur decompositions
- The **(infinitely many) solutions at infinity** can be deflated from the numerical basis matrix via a column compression

Multiplicity and solutions at infinity

- **Multiple solutions** lead to a confluent block multivariate Vandermonde basis matrix and the Jordan normal form, but we can avoid this via multiple Schur decompositions
- The **(infinitely many) solutions at infinity** can be deflated from the numerical basis matrix via a column compression → **requires several rank checks**



degree blocks	rank
0	1
0 – 1	2
0 – 2	3
0 – 3	3
0 – 4	5
0 – 5	7

Double recursive algorithms

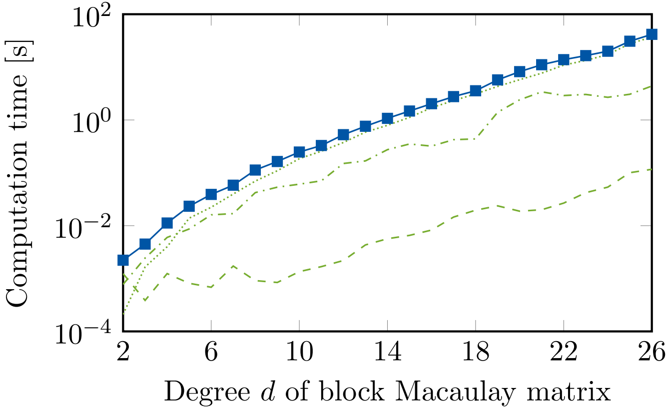
Non-recursive null space based algorithm

```
while gap zone is not yet large enough do
    Construct the block Macaulay matrix and compute a numerical basis matrix
    of its null space.
    Check the nullity or rank structure of the basis matrix to determine if a
    large enough gap zone exists.
end while
Perform column compression and solve the generalized eigenvalue problem
```

We need **double recursive algorithms** to tackle these problems more efficiently:

1. A recursive (or sparse) technique to construct a basis matrix of the null space of the block Macaulay matrix
2. A recursive technique to determine the rank structure of that basis matrix

Computational bottleneck

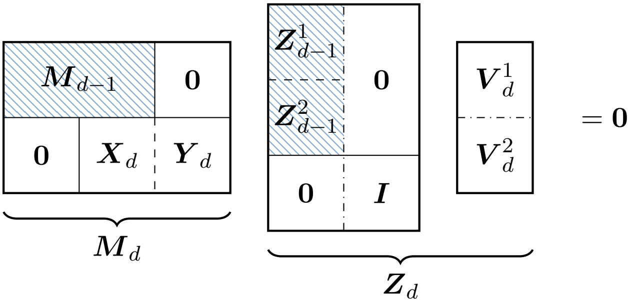


Computation time (—■—) per degree for a second-order least-squares realization problem: constructing block Macaulay matrix (- - -), computing a basis matrix of the null space (·······), and checking the rank structure (-·-·-) of that basis matrix.

Outline

Outline

Recursive block Macaulay orthogonalization



Algorithm

$$W_d = \begin{bmatrix} X_d Z_{d-1}^2 & Y_d \end{bmatrix}$$

$$V_d = \text{NULL}\left(W_d\right)$$

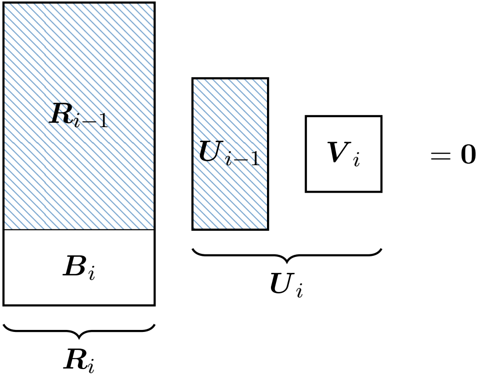
$$Z_d = \begin{bmatrix} Z_{d-1} V_d^1 \\ V_d^2 \end{bmatrix}$$

$\mathcal{O}\left(d^{3n-2}\right)$ instead of $\mathcal{O}\left(d^{3n}\right)$

(Vermeersch and De Moor, 2023)

Outline

Recursive block row orthogonalization



Algorithm

$$\boldsymbol{W}_i = \boldsymbol{B}_i \boldsymbol{U}_{i-1}$$

$$\boldsymbol{V}_i = \text{NULL}\left(\boldsymbol{W}_i\right)$$

$$\boldsymbol{U}_i = \boldsymbol{U}_{i-1} \boldsymbol{V}_i$$

$\mathcal{O}(1)$ instead of $\mathcal{O}(d)$

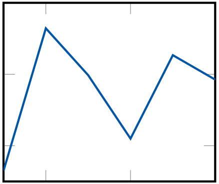
(Vermeersch and De Moor, 2023)

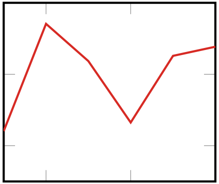
Outline

First example: Least-squares realization problem

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_6 \end{bmatrix}}_{\text{known}} \longrightarrow \underbrace{\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_6 \end{bmatrix}}_{\text{unknown}}$$

such that $\hat{\boldsymbol{y}}_k = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{x}_0$ is the output of a **second-order** autonomous system

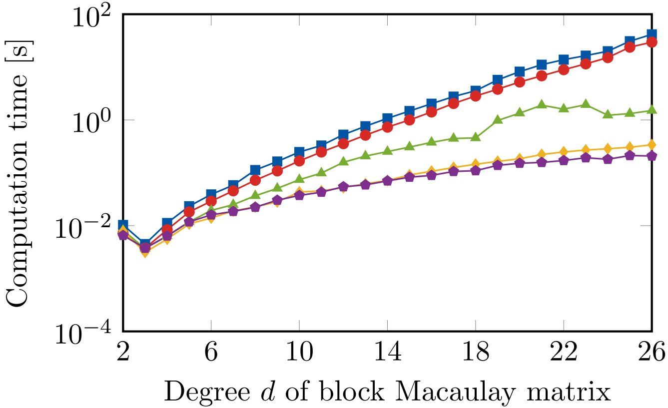




$$\min \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$$
$$\text{subject to } \boldsymbol{T}_\alpha \hat{\boldsymbol{y}} = \boldsymbol{0}$$

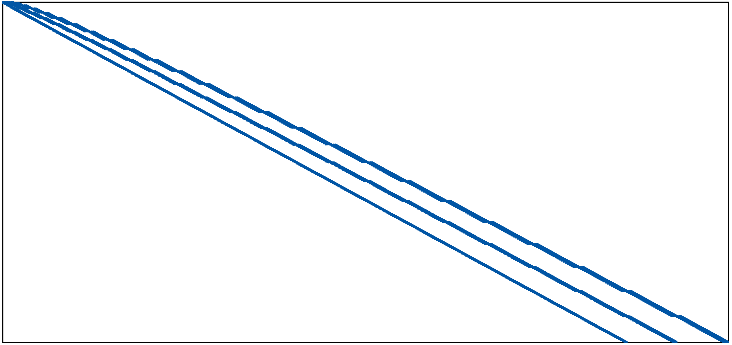
(De Moor, 2019, 2020)

First example: Least-squares realization problem



Different combinations of techniques to solve a second-order realization problem: via STANDARD-STANDARD (—■—), STANDARD-RECURSIVE (—●—), RECURSIVE-STANDARD (—▲—), RECURSIVE-RECURSIVE (—◆—), and SPARSE-RECURSIVE (—⬟—) approach.

First example: Least-squares realization problem



The block Macaulay matrix of degree $d = 26$ for the second-order realization problem is a 4550×4914 matrix. The SPARSE-RECURSIVE approach avoids the construction of this large and sparse matrix (only 0.24% non-zero elements).

Second example: ARMA model

$$\boldsymbol{y} \in \mathbb{R}^8$$

$$\downarrow$$

$$\min \|\boldsymbol{e}\|_2^2$$

$$\text{subject to } \boldsymbol{T}_\alpha \boldsymbol{y} = \boldsymbol{T}_\gamma \boldsymbol{e}$$

$$\downarrow$$

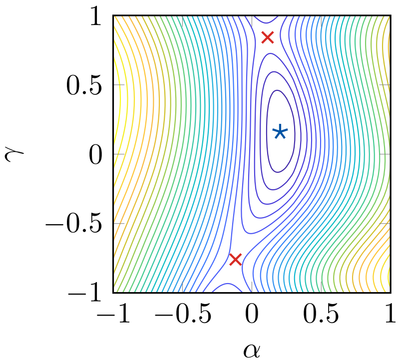
$$\left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{01}\gamma + \boldsymbol{A}_{02}\gamma^2\right)\boldsymbol{z} = \boldsymbol{0}$$

$$\downarrow$$

block Macaulay matrix and shift-invariance

$$\downarrow$$

parameters α and γ



Contour plot of the cost function with one minimum ($\star$) and two saddle points ($\times$)

(Vermeersch and De Moor, 2019, 2022)

Second example: ARMA model

Different combinations of recursive techniques to solve a first-order ARMA$(1,1)$ model identification problem with $N = 8$ data points

Combination	Computation time	Maximum residual error
STANDARD-STANDARD	31 223.95 s	5.16×10^{-14}
STANDARD-RECURSIVE	27 951.57 s	5.16×10^{-14}
RECURSIVE-STANDARD	323.00 s	1.24×10^{-12}
RECURSIVE-RECURSIVE	69.41 s	1.24×10^{-12}
SPARSE-RECURSIVE*	41.74 s	1.48×10^{-13}

* Notice that SPARSE-RECURSIVE implementation avoids the construction of a 20769×21780 block Macaulay matrix.

Outline

Conclusion and future work

- The **double recursive algorithms*** leads to impressive results in computation time
  - Example 1: factor 80 improvement
  - Example 2: factor 725 improvement
- A **sparse adaptation** avoids the construction of the large and sparse block Macaulay matrix
  - Example 1: 178.87 MB → 9.36 kB
  - Example 2: 3.62 GB → 24.28 kB
- Can we do something similar with the QR-decomposition and the column space based approach?

* All algorithms are implemented in MATLAB and available at `www.macaulaylab.net`.

Any questions?

Christof Vermeersch[†,‡]
Bart De Moor[†]

[†] KU Leuven, Dept. of Electrical Engineering (ESAT), Center for Dynamical Systems, Signal Processing, and Data Analytics (STADIUS), Kasteelpark Arenberg 10, 3001 Leuven, Belgium (christof.vermeersch@esat.kuleuven.be, bart.demoor@esat.kuleuven.be)

[‡] Supported by FWO Strategic Basic Research fellowship under grant SB/1SA1319N

References

Kim Batselier, Philippe Dreesen, and Bart De Moor. Prediction error method identification is an eigenvalue problem. In *Proc. of the 16th IFAC Symposium on System Identification*, pages 221–226, Brussels, Belgium, 2012.

Bart De Moor. Least squares realization of LTI models is an eigenvalue problem. In *Proc. of the 18th European Control Conference (ECC)*, pages 2270–2275, Naples, Italy, 2019.

Bart De Moor. Least squares optimal realisation of autonomous LTI systems is an eigenvalue problem. *Communications in Information and Systems*, 20(2):163–207, 2020.

Bor Plestenjak, Călin I. Gheorghiu, and Michiel E. Hochstenbach. Spectral collocation for multiparameter eigenvalue problems arising from separable boundary value problems. *Journal of Computational Physics*, 298:585–601, 2015.

Françoise Tisseur and Karl Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2): 235–286, 2001.

Christof Vermeersch and Bart De Moor. Globally optimal least-squares ARMA model identification is an eigenvalue problem. *IEEE Control Systems Letters*, 3(4):1062–1067, 2019.

Christof Vermeersch and Bart De Moor. Two complementary block Macaulay matrix algorithms to solve multiparameter eigenvalue problems. *Linear Algebra and its Applications*, 654:177–209, 2022.

Christof Vermeersch and Bart De Moor. Two double recursive block Macaulay matrix algorithms to solve multiparameter eigenvalue problems. *IEEE Control Systems Letters*, 7:319–324, 2023.

Other shift functions

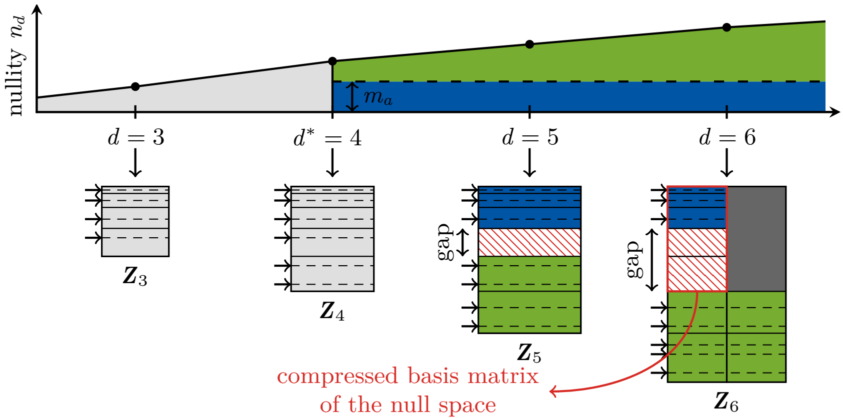
- It is **possible to shift with any polynomial** in the eigenvalues – for example with $g\left(\lambda,\mu\right) = 3\lambda + 2\mu^3$

$$\left(\boldsymbol{S}_1\boldsymbol{Z}\right)\boldsymbol{T}\underbrace{\begin{bmatrix} g\left(\lambda,\mu\right)\big|_{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g\left(\lambda,\mu\right)\big|_{(m_a)} \end{bmatrix}}_{\boldsymbol{D}_g} = \left(\boldsymbol{S}_g\boldsymbol{Z}\right)\boldsymbol{T}$$

- This can be useful in applications with a polynomial cost function

(Vermeersch and De Moor, 2023)

Positive-dimensional solution sets at infinity



Some multiparameter eigenvalue problems have a positive-dimensional solution set at infinity, so the nullity of the block Macaulay matrix does not stabilize. This behavior makes the problem even harder!