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

Christof Vermeersch and Bart De Moor, *Fellow*, *IEEE & SIAM*

***Abstract*—We present a *recursive-recursive* and *sparse-recursive* block Macaulay matrix algorithm to solve multiparameter eigenvalue problems (MEPs). In earlier work, we have developed algorithms to find the solutions of an MEP via a multidimensional realization problem in the null space of the block Macaulay matrix constructed from the coefficient matrices of that MEP. However, this null space based approach requires an iterative increase of the degree of the block Macaulay matrix: in order to determine if the null space contains all the (affine) solutions of the MEP, i.e., if the degree is *large enough*, we need to compute a numerical basis matrix of the null space and check its rank structure in every iteration. In this letter, we use recursive and sparse techniques to update the basis matrix and to perform the necessary rank checks. We provide two examples from system identification to show our improvements in both computation time and memory usage: in one example, we notice that the recursive-recursive and sparse-recursive algorithm are 400 and 700 times faster than the standard approach, respectively.**

***Index Terms*—Numerical algorithms, linear systems, identification.**

I. Introduction

When identifying linear time-invariant systems or solving partial differential equations via the method of separation of variables, one can encounter multiparameter eigenvalue problems (MEPs) [1], [2], [3]. In contrast to standard eigenvalue problems (SEPs) and polynomial eigenvalue problems (PEPs), which are well-understood and for which many efficient algorithms exist [4], techniques to solve MEPs are much less abundant in the literature [5].

We have introduced in earlier work the block Macaulay matrix [1], [2], [5], which is an extension of the classical (scalar) Macaulay matrix from polynomial system solving, to solve MEPs via a multidimensional realization problem in its structured null space. Although the block Macaulay matrix approach creates an elegant, unifying framework to solve SEPs, PEPs, and (polynomial) MEPs, it suffers from a computational burden: before constructing a multidimensional realization problem that yields the solutions of the MEP, we need to increase the degree of the block Macaulay matrix iteratively and check in every iteration the structure of its null space in order to determine if it contains all the (affine) solutions, i.e., we need to check if the degree is *large enough*.

Algorithms that recursively[1] construct the block Macaulay matrix and compute a numerical basis matrix of its null space based on previous iterations are, without any doubt, very useful. Thereupon, we use in this letter the recursive algorithms developed in [6] and combine them into two double recursive null space based block Macaulay matrix algorithms: we use a (i) recursive/sparse algorithm to update a numerical basis matrix of the null space of the block Macaulay matrix and we apply a (ii) recursive algorithm to check the rank structure of that basis matrix (which is a block row matrix). By means of two examples from system identification, we develop in this letter a *recursive-recursive* (with both (i) and (ii) recursive) and *sparse-recursive* (with (i) sparse and (ii) recursive) block Macaulay matrix algorithm to find the solutions of an MEP. Although the key ingredients of the two double recursive algorithms have already been discussed in [6], we combine them for the first time in this letter in order to solve MEPs more efficiently. We also show how to deflate positive-dimensional solution sets at infinity (i.e., the number of solutions at infinity is not finite), an additional difficulty that arises, for example, in the system identification problems that we tackle in this letter. We solve both system identification examples to demonstrate our improvements in both computation time and memory usage: in one example, we notice that the recursive-recursive and sparse-recursive algorithm are 400 and 700 times faster than the standard approach, respectively.

The remainder of this letter proceeds as follows: In Section II, we define (rectangular) MEPs and give two examples from system identification. Next, in Section III, we construct the block Macaulay matrix from an MEP and show

Christof Vermeersch (corresponding author) and Bart De Moor are with the KU Leuven, Department of Electrical Engineering (ESAT), Center for Dynamical Systems, Signal Processing, and Data Analytics (STADIUS). Kasteelpark Arenberg 10, 3001 Leuven, Belgium. E-mail addresses: {christof.vermeersch, bart.demoor}@esat.kuleuven.be

This work was supported in part by the KU Leuven: Research Fund (projects C16/15/059, C3/19/053, C24/18/022, C3/20/117, and C3I-21-00316), Industrial Research Fund (fellowships 13-0260, IOFm/16/004, and IOFm/20/002), and several Leuven Research and Development bilateral industrial projects, in part by Flemish Government agencies: FWO (EOS project G0F6718N (SeLMA), SBO project S005319, infrastructure project I013218N, TBM project T001919N, and PhD grant SB/1SA1319N), EWI (Flanders AI Research Program), and VLAIO (Baekeland PhD mandate HBC.2019.2204, and HBC.2021.0076), and in part by the European Commission (EU Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant) under grant 885682). Other funding: Foundation "Kom op tegen Kanker", Christelijke Mutualiteit. The work of Christof Vermeersch was supported by the FWO Strategic Basic Research fellowship under grant SB/1SA1319N.

[1]We do not use the term *recursion* in its strict computer science meaning ("an algorithm that calls itself on smaller input values"), but see it as an algorithm that performs the same steps on different input values ("an algorithm that uses in every iteration the same approach on new input values"), cf., the recursive least-squares algorithm.

how a multidimensional realization problem in its null space yields the solutions of that MEP. Afterwards, we develop in Section IV the double recursive algorithm and its sparse adaptation. Section V illustrates both algorithms by means of the system identification examples. Finally we conclude this letter and point at ideas for future research in Section VI.

II. Multiparameter eigenvalue problems

While a standard eigenvalue problem (SEP) contains only single eigenvalues λ, a multiparameter eigenvalue problem (MEP) has eigentuples $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ of multiple eigenvalues. Several manifestations of MEPs appear in the literature, but in this letter we focus solely on rectangular MEPs (see [5] for a more elaborate overview):

Definition 1: Given coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} \in \mathbb{R}^{k \times l}$ (with $k \geq l + n - 1$), 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 $\boldsymbol{z} \in \mathbb{C}^{l \times 1} \setminus \{\boldsymbol{0}\}$, such that

$$\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\, \boldsymbol{z} = \left(\sum_{\{\boldsymbol{\omega}\}} \boldsymbol{A}_{\boldsymbol{\omega}} \boldsymbol{\lambda}^{\boldsymbol{\omega}} \right) \boldsymbol{z} = \boldsymbol{0}, \tag{1}$$

where the summation runs over all the multi-indices $\boldsymbol{\omega}$ of the monomials $\boldsymbol{\lambda}^{\boldsymbol{\omega}} = \prod_{i=1}^{n} \lambda_i^{\omega_i}$. The n-tuples $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ and (non-zero) vectors $\boldsymbol{z}$ are the eigentuples (with eigenvalues $\lambda_1, \ldots, \lambda_n$) and eigenvectors of the MEP, respectively.

The size condition on the coefficient matrices is a necessary (but not a sufficient) condition to have a zero-dimensional solution set: there are k equations and one non-triviality constraint on $\boldsymbol{z}$ (e.g., $\|\boldsymbol{z}\|_2 = 1$) in $l + n$ unknowns (l elements in the eigenvectors and n eigenvalues), hence $k + 1 \geq l + n$. The matrix $\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)$ is a multivariate polynomial in the eigenvalues $\lambda_1, \ldots, \lambda_n$ with matrix coefficients $\boldsymbol{A}_{\boldsymbol{\omega}}$.

Example 1: Our first example is the MEP that arises from the least-squares realization problem: given a data sequence $y_0, \ldots, y_{N-1}$ ($\boldsymbol{y} \in \mathbb{R}^{N \times 1}$), find the adapted data sequence $\hat{y}_0, \ldots, \hat{y}_{N-1}$ so that the misfit $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$ is minimized and $\hat{\boldsymbol{y}} \in \mathbb{R}^{N \times 1}$ is the output of a model of pre-specified order n [2]:

$$\hat{y}_k = \boldsymbol{C}\boldsymbol{A}^k \boldsymbol{x}_0,$$

where $\boldsymbol{x}_0 \in \mathbb{R}^{n \times 1}$ is the initial state, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the system matrix, and $\boldsymbol{C} \in \mathbb{R}^{1 \times n}$ is the output vector. In [2] has been shown how this identification problem corresponds to a quadratic MEP, with the number of eigenvalues equal to n. When we consider a model of order $n = 2$, we obtain a quadratic two-parameter eigenvalue problem

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\lambda_1, \lambda_2)\, \boldsymbol{z} = \big(&\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\lambda_1 + \boldsymbol{A}_{01}\lambda_2 + \\ &\boldsymbol{A}_{20}\lambda_1^2 + \boldsymbol{A}_{11}\lambda_1\lambda_2 + \boldsymbol{A}_{02}\lambda_2^2\big)\boldsymbol{z} = \boldsymbol{0}, \end{aligned} \tag{2}$$

with coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} \in \mathbb{R}^{(3N-4) \times (3N-5)}$ as described in [2]. The integer multi-index $\boldsymbol{\omega} = (\omega_1, \omega_2) \in \mathbb{N}^2$ labels the powers of the eigenvalues in the monomial $\lambda_1^{\omega_1}\lambda_2^{\omega_2}$ and indexes the associated coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} = \boldsymbol{A}_{(\omega_1, \omega_2)}$. The total degree of a monomial is equal to the sum of its powers, denoted by $|\boldsymbol{\omega}| = \omega_1 + \omega_2$. Hence, an integer multi-index $\boldsymbol{\omega} = (0, 2)$ labels the monomial λ_2^2 (with total degree 2) and indexes the associated coefficient matrix $\boldsymbol{A}_{02}$.

Example 2: Secondly, we consider the least-squares ARMA model identification problem. We have shown in [1] that the stationary points of the related optimization problem correspond to the solutions of an MEP. A first-order ARMA$(1, 1)$ model combines a regression of the observed output variable $y_k \in \mathbb{R}$ on its own lagged value y_{k-1} with a linear combination of unobserved, latent inputs e_k and $e_{k-1} \in \mathbb{R}$ [1]:

$$y_k + \alpha y_{k-1} = e_k + \gamma e_{k-1},$$

where the weighting factors α and γ are the model parameters of this ARMA model. When we consider a given series of N output samples $\boldsymbol{y} \in \mathbb{R}^{N \times 1}$, the quadratic two-parameter eigenvalue problem (see [1] for the exact construction of $\boldsymbol{A}_{\boldsymbol{\omega}}$)

$$\boldsymbol{\mathcal{M}}(\alpha, \gamma)\, \boldsymbol{z} = \left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{10}\gamma + \boldsymbol{A}_{02}\gamma^2\right) \boldsymbol{z} = \boldsymbol{0} \tag{3}$$

yields the stationary points of the related optimization problem. This problem contains $n = 2$ spectral parameters (α, γ) and has four non-zero coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} \in \mathbb{R}^{(3N-1) \times (3N-2)}$ ($\boldsymbol{A}_{20}$ and $\boldsymbol{A}_{11}$ are zero-matrices).

III. Standard approach

In this section, we sketch the different steps of the standard null space based block Macaulay matrix approach to solve MEPs[2]. After an intuitive construction of the block Macaulay matrix (Section III-A), we first assume that all solutions are simple, isolated, and affine, which allows us to show that a multidimensional realization problem in the structured null space yields the solutions of the MEP (Section III-B). Sometimes, MEPs that arrive from system identification problems (like Examples 1 and 2), have a positive-dimensional solution set at infinity. We show how to deflate these solutions at infinity (Section III-C) and close this section with an overview of the different steps of the standard approach (Section III-D).

A. Block Macaulay matrix of an MEP

The MEP $\boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\, \boldsymbol{z} = \boldsymbol{0}$ in (1) constitutes the so-called *seed equation* of its corresponding block Macaulay matrix [5]. Indeed, we can generate "new" matrix equations $\left\{\prod_{i=1}^{n} \lambda_i^{d_i}\right\} \boldsymbol{\mathcal{M}}(\lambda_1, \ldots, \lambda_n)\, \boldsymbol{z} = \boldsymbol{0}$ by multiplying the seed equation (i.e., the MEP) by different monomials $\prod_{i=1}^{n} \lambda_i^{d_i}$ of increasing total degree $d_R = \sum_{i=1}^{n} d_i$, and we stack these "new" matrix equations as the block rows of the block Macaulay matrix $\boldsymbol{M}_d$, where the degree d is equal to the total degree of the highest monomial in all the matrix equations. A rigorous definition of the block Macaulay matrix can be found in [5], while we restrict ourselves in this letter to a more intuitive construction.

Example 3: If we revisit the quadratic MEP in (3),

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

[2]We refer the reader to [5] for a more detailed exposition of the different steps and an alternative column space based block Macaulay matrix approach.

$$\begin{array}{c|c|cc|ccc|cccc|cc}
 & \boldsymbol{z} & \alpha\boldsymbol{z} & \gamma\boldsymbol{z} & \alpha^2\boldsymbol{z} & \alpha\gamma\boldsymbol{z} & \gamma^2\boldsymbol{z} & \alpha^3\boldsymbol{z} & \alpha^2\gamma\boldsymbol{z} & \alpha\gamma^2\boldsymbol{z} & \gamma^3\boldsymbol{z} & \alpha^4\boldsymbol{z} & \\
1 & \boldsymbol{A}_{00} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{02} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\
\alpha & \boldsymbol{0} & \boldsymbol{A}_{00} & \boldsymbol{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{02} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\
\gamma & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{00} & \boldsymbol{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{02} & \boldsymbol{0} & \cdots \\
\alpha^2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{00} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\
\alpha\gamma & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{00} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{10} & \boldsymbol{A}_{01} & \boldsymbol{0} & \boldsymbol{0} & \cdots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

Fig. 1. The block Macaulay matrix $\boldsymbol{M}_d$ of the quadratic two-parameter eigenvalue problem in Example 2. The coefficient matrices of the seed equation, i.e., the generating MEP, are indicated in red. Vertical lines indicate the different degree blocks. Clearly, the block Macaulay matrix exhibits a quasi-Toeplitz structure: the coefficient matrices of the seed equation return in every block row in very sparse and structured pattern.

and we multiply this seed equation by monomials of total degree $d_R = 1$, then we obtain two "new" matrix equations:

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

The corresponding block Macaulay matrix $\boldsymbol{M}_3$ has degree $d = 3$ (highest total degree of MEP is 2 and $d_R = 1$). We can continue this process with monomials of increasing total degree d_R, i.e.,

$$\underbrace{\alpha, \gamma}_{d_R=1}, \underbrace{\alpha^2, \alpha\gamma, \gamma^2}_{d_R=2}, \underbrace{\alpha^3, \alpha^2\gamma, \dots}_{d_R\geq 3}$$

and organize the resulting coefficient matrices in a block Macaulay matrix $\boldsymbol{M}_d$ as in Fig. 1. The actual structure of the block Macaulay matrix depends on its multivariate monomial ordering. We use in this letter the degree negative lexicographic ordering [7], but the remainder remains valid for every graded monomial ordering.

Consequently, we can rewrite the MEP and "new" matrix equations as a matrix-vector product of a block Macaulay matrix $\boldsymbol{M}_d \in \mathbb{R}^{p_d \times q_d}$ and a vector $\boldsymbol{v}_d \in \mathbb{C}^{q_d \times 1}$, i.e.,

$$\underbrace{\left[\left\{\prod_{i=1}^{n} \lambda_i^{d_i}\right\} \boldsymbol{\mathcal{M}}\left(\lambda_1, \dots, \lambda_n\right)\right]}_{\boldsymbol{M}_d} \underbrace{\left[\begin{array}{c} \boldsymbol{z} \\ \hline \boldsymbol{z}\lambda_1 \\ \vdots \\ \boldsymbol{z}\lambda_n \\ \hline \boldsymbol{z}\lambda_1^2 \\ \vdots \end{array}\right]}_{\boldsymbol{v}_d} = \boldsymbol{0}.$$

The vector $\boldsymbol{v}_d$ is a vector in the (right) null space of the block Macaulay matrix $\boldsymbol{M}_d$ and has a special block multivariate Vandermonde structure, because of the monomial ordering of the columns of the block Macaulay matrix[3]. In the structure of the null space of the block Macaulay matrix lies the key to solve its generating MEP. We need to increase the degree d of the block Macaulay matrix $\boldsymbol{M}_d$ until the structure of its null space allows us to retrieve all the (affine) solutions of the MEP: the degree d needs to be *large enough*, a concept on which we elaborate in the next section.

[3]Note that we make a distinction between *blocks* and *degree blocks* in this letter. A block gathers all the rows or columns that correspond to one monomial (e.g., all the rows that belong to λ_1^2), while a degree block contains all the blocks that correspond to monomials of the same total degree (e.g., all the rows that belong to λ_1^2, $\lambda_1\lambda_2$, and λ_2^2). We separate different degree blocks in matrices and vectors with lines.

B. Affine solutions in the structured null space

Initially, we consider an MEP $\boldsymbol{\mathcal{M}}\left(\lambda_1, \dots, \lambda_n\right)\boldsymbol{z} = \boldsymbol{0}$ that only has m_a simple and affine solutions, i.e., an MEP with a simple and affine zero-dimensional solution set. When we iteratively increase the degree d of the block Macaulay matrix $\boldsymbol{M}_d$, we notice that the nullity (i.e., the dimension of the null space) grows until it reaches the number of affine solutions m_a (at $d = d^*$, by definition), and it remains the same for larger degrees ($d \geq d^*$). Every solution of the MEP corresponds to one block multivariate Vandermonde vector in the null space and, together, these basis vectors span the entire null space of $\boldsymbol{M}_d$. They naturally form the block multivariate Vandermonde basis matrix $\boldsymbol{V}_d \in \mathbb{C}^{q_d \times m_a}$ of degree d:

$$\boldsymbol{V}_d = \left[\begin{array}{ccc} \boldsymbol{z}|_{(1)} & \cdots & \boldsymbol{z}|_{(m_a)} \\ \hline (\lambda_1\boldsymbol{z})|_{(1)} & \cdots & (\lambda_1\boldsymbol{z})|_{(m_a)} \\ \vdots & & \vdots \\ (\lambda_n\boldsymbol{z})|_{(1)} & \cdots & (\lambda_n\boldsymbol{z})|_{(m_a)} \\ \hline (\lambda_1^2\boldsymbol{z})|_{(1)} & \cdots & (\lambda_1^2\boldsymbol{z})|_{(m_a)} \\ \vdots & & \vdots \end{array}\right],$$

which has one column $\boldsymbol{v}_d|_{(j)}$, $j = 1, \dots, m_a$, for every affine solution of the MEP. As explained thoroughly in [5], the null space of the block Macaulay matrix has a special structure:

Proposition 1: The (affine) null space of the block Macaulay matrix is **(backward) block multi-shift-invariant**, which means that if we select a block row of a *large enough* basis matrix of the null space and multiply/shift this block row by one of the eigenvalues, then we obtain another block row of that basis matrix [5].

This (backward) block multi-shift-invariance of the null space corresponds mathematically to

$$\underbrace{\boldsymbol{S}_g\boldsymbol{V}_d}_{\text{after shift}} = \underbrace{\boldsymbol{S}_1\boldsymbol{V}_d}_{\text{before shift}}\boldsymbol{D}_g, \tag{4}$$

where the diagonal matrix $\boldsymbol{D}_g \in \mathbb{C}^{m_a \times m_a}$ contains the evaluations of the shift polynomial $g\left(\lambda_1, \dots, \lambda_n\right)$ in the different solutions of the MEP. In order for this expression to contain all m_a affine solutions, the matrix $\boldsymbol{S}_1\boldsymbol{V}_d$ has to be non-singular. This means that the row selection matrix $\boldsymbol{S}_1 \in \mathbb{R}^{s \times q_d}$ has to select block rows that contain together m_a linearly independent rows from the block multivariate Vandermonde basis matrix $\boldsymbol{V}_d$. The matrix $\boldsymbol{S}_g \in \mathbb{R}^{s \times q_d}$, on the other hand, simply selects the block rows obtained after the multiplication (or shift) by the shift polynomial $g(\lambda_1, \ldots, \lambda_n)$. The basis matrix $\boldsymbol{V}_d$ is *large enough* when it can *accommodate* this shift polynomial: when we shift the first r degree blocks of $\boldsymbol{V}_d$ by a shift polynomial of degree d_g, the degree of $\boldsymbol{V}_d$ (and thus also of $\boldsymbol{M}_d$) must be at least $r + d_g > d^*$.

In practice, we do not know the block multivariate Vandermonde basis $\boldsymbol{V}_d$ in advance, since it is constructed from the unknown solutions of the MEP. Therefore, as the (backward) block multi-shift-invariance is a property of the null space and not of its specific basis matrix, we work with a numerical basis matrix $\boldsymbol{Z}_d \in \mathbb{C}^{q_d \times m_a}$ of $\boldsymbol{M}_d$ instead, for example obtained via the singular value decomposition. There exists, of course, a linear transformation between these basis matrices, namely $\boldsymbol{V}_d = \boldsymbol{Z}_d \boldsymbol{T}$, with $\boldsymbol{T} \in \mathbb{C}^{m_a \times m_a}$ a non-singular transformation matrix, which transforms (4) into a solvable rectangular GEP,

$$(\boldsymbol{S}_g \boldsymbol{Z}_d)\, \boldsymbol{T} = (\boldsymbol{S}_1 \boldsymbol{Z}_d)\, \boldsymbol{T} \boldsymbol{D}_g, \tag{5}$$

where $\boldsymbol{T}$ contains the eigenvectors and $\boldsymbol{D}_g$ the eigenvalues of the matrix pencil $(\boldsymbol{S}_g \boldsymbol{Z}_d, \boldsymbol{S}_1 \boldsymbol{Z}_d)$. Alternatively, we can also consider the SEP ($.^\dagger$ denotes the pseudo-inverse)

$$\boldsymbol{T} \boldsymbol{D}_g \boldsymbol{T}^{-1} = (\boldsymbol{S}_1 \boldsymbol{Z}_d)^\dagger (\boldsymbol{S}_g \boldsymbol{Z}_d).$$

We can then use the matrix of eigenvectors $\boldsymbol{T}$ to retrieve $\boldsymbol{V}_d$ via $\boldsymbol{V}_d = \boldsymbol{Z}_d \boldsymbol{T}$. From $\boldsymbol{V}_d$ and/or $\boldsymbol{D}_g$, we find the solutions of the MEP.

Influence of multiplicity larger than one: Affine and isolated solutions can have a multiplicity larger than one. This poses no problems to the above-mentioned approach (see [5] for a more extensive discussion about multiple solutions).

C. *Solutions at infinity and positive-dimensional solution sets*

Due to the singularity of some higher degree coefficient matrices, MEPs can also have solutions at infinity. Moreover, the MEPs that arise in system identification problems (like in [1], [2]) can have a positive-dimensional solution set at infinity, which means that the total number of solutions is infinite (remember that the condition on the size of the coefficient matrices is only a necessary condition). In that case, the nullity n_d of the block Macaulay matrix $\boldsymbol{M}_d$ no longer stabilizes at the number of affine solutions m_a, but keeps increasing when we increase the degree d (see Fig. 2).

However, when the number of affine solutions is finite, we can still apply the above-described affine solution approach. The solutions of an MEP give rise to linearly independent rows in the (numerical) basis matrix $\boldsymbol{Z}_d$ of the null space of $\boldsymbol{M}_d$ (see [5]). When we monitor the linearly independent rows in $\boldsymbol{Z}_d$ (checked row-wise from top to bottom – see Fig. 2), we notice at least one additional linearly independent row per degree block up to degree $d \leq d^*$ The linearly independent rows that correspond to the standard monomials associated to the affine solutions stabilize from a certain degree d^* on at their respective positions. The linearly independent rows that correspond to the standard monomials associated to the solutions at infinity, on the other hand, keep moving to higher degree blocks when we further increase the degree $d > d^*$. A gap zone in the rows of $\boldsymbol{Z}_d$ without any additional

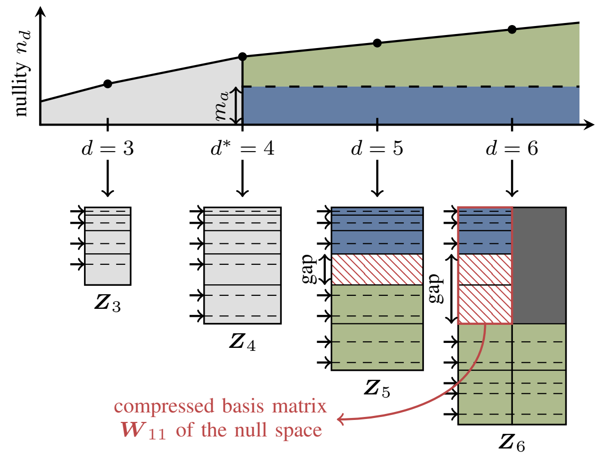

Fig. 2. The nullity of the null space of the block Macaulay matrix $\boldsymbol{M}_d$ grows as its degree d increases. However, at a certain degree $d \geq d^*$ (in this example $d^* = 4$), the affine zone of the basis matrix stabilizes. From that degree on, some linearly independent rows of the basis matrix $\boldsymbol{Z}_d$ of the null space (related to the affine solutions – indicated by dashed lines) stabilize, while the other linearly independent rows (related to the solutions at infinity – also indicated by dashed lines) move to higher degree blocks, and a gap emerges that separates these two types of linearly independent rows. The influence of the solutions at infinity can then be deflated via a column compression [5].

Algorithm 1 Standard null space based approach

```
1: while $d^* + d_g$ is not yet reached do
2:     Construct the block Macaulay matrix $\boldsymbol{M}_d$.
3:     Compute a numerical basis matrix $\boldsymbol{Z}_d$ of $\boldsymbol{M}_d$.
4:     Check the rank structure of $\boldsymbol{Z}_d$ — if there is a
       large enough gap in $\boldsymbol{Z}_d$, then $d^* + d_g$ has been reached.
5: end while
6: Use Algorithm 2 to compute the solutions of the MEP.
```

linearly independent rows emerges. When this gap zone can *accommodate* the shift polynomial (a shift polynomial of degree d_g requires a gap zone of d_g degree blocks), the basis matrix $\boldsymbol{Z}_d$ is *large enough* and we can deflate the solutions at infinity via a column compression (see [5]). This column compression allows us to use the above-described affine null space based approach as if no solutions at infinity were present (we simply replace $\boldsymbol{Z}_d$ in (5) by the compressed basis matrix $\boldsymbol{W}_{11}$ of the null space).

In order to determine if the basis matrix is *large enough*, we need to check the rank structure (i.e., the linearly independent rows) of the basis matrix and search for a gap zone of d_g degree blocks in every iteration.

D. *Standard null space based approach*

Algorithm 1 gives an overview of the different steps to compute the solutions of an MEP via the standard null space based approach, which uses Algorithm 2 to retrieve the solutions from a *large enough* basis matrix $\boldsymbol{Z}_d$.

IV. Two double recursive algorithms

The fact that the required degree $d^* + d_g$ is not known in advance is the main difficulty of the standard null space based approach. We need to increase the degree d of the

Algorithm 2 Solve MEP from a *large enough* basis matrix

```
1: Use a column compression to obtain the compressed numerical basis matrix $\boldsymbol{W}_{11}$ of the null space [5].
2: For a user-defined shift polynomial $g(\lambda_1, \ldots, \lambda_n)$, solve the rectangular GEP
       $(\boldsymbol{S}_g \boldsymbol{W}) \boldsymbol{T} = (\boldsymbol{S}_1 \boldsymbol{W}_{11}) \boldsymbol{T} \boldsymbol{D}_g,$
   where $\boldsymbol{S}_1$, $\boldsymbol{S}_g$, $\boldsymbol{T}$, and $\boldsymbol{D}_g$ are defined as in (5).
3: Retrieve the different components of the solutions from the block multivariate Vandermonde basis $\boldsymbol{V}_d = \boldsymbol{W}_{11}\boldsymbol{T}$.
```

block Macaulay matrix $\boldsymbol{M}_d$ and compute a (numerical) basis matrix $\boldsymbol{Z}_d$ of its null space in every iteration in order to check if d is *large enough* (i.e., to check if $\boldsymbol{Z}_d$ contains a gap zone that can accommodate the shift). Instead of re-computing $\boldsymbol{Z}_d$ in every iteration, we can use a recursive algorithm to construct $\boldsymbol{Z}_d$ based on $\boldsymbol{Z}_{d-1}$ (Section IV-A). Furthermore, we can also check the rank structure via a recursive algorithm (Section IV-B). Afterwards, we combine both recursive techniques in a double recursive block Macaulay matrix algorithm (Section IV-C) and propose a sparse adaptation (Section IV-D).

A. *Recursive computation of a basis matrix of the null space*

Consider a block Macaulay matrix $\boldsymbol{M}_{d-1} \in \mathbb{R}^{p_{d-1} \times q_{d-1}}$ and a numerical basis matrix $\boldsymbol{Z}_{d-1} \in \mathbb{C}^{q_{d-1} \times n_{d-1}}$ of its null space. Obviously, $\boldsymbol{M}_{d-1}\boldsymbol{Z}_{d-1} = \boldsymbol{0}$, and we can append the block Macaulay matrix with t_d zero columns at the end (with $\boldsymbol{I}_{t_d \times t_d} \in \mathbb{N}^{t_d \times t_d}$ the identity matrix):

$$\begin{bmatrix} \boldsymbol{M}_{d-1} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{Z}_{d-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{t_d \times t_d} \end{bmatrix} = \boldsymbol{0}.$$

If we append the new part of the block Macaulay matrix, (we now consider the block Macaulay matrix $\boldsymbol{M}_d \in \mathbb{R}^{p_d \times q_d}$), then we know that there exists an orthonormal matrix $\boldsymbol{V}_d \in \mathbb{C}^{(n_{d-1}+t_d) \times n_d}$, such that we can compute a numerical basis matrix of $\boldsymbol{M}_d$ as

$$\underbrace{\begin{bmatrix} \boldsymbol{M}^1_{d-1} & \boldsymbol{M}^2_{d-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_d & \boldsymbol{Y}_d \end{bmatrix}}_{\boldsymbol{M}_d} \begin{bmatrix} \boldsymbol{Z}^1_{d-1} & \boldsymbol{0} \\ \boldsymbol{Z}^2_{d-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{t_d \times t_d} \end{bmatrix} \boldsymbol{V}_d = \boldsymbol{0}, \qquad (6)$$

where we have split the block Macaulay matrix $\boldsymbol{M}_{d-1}$ into $\boldsymbol{M}^1_{d-1} \in \mathbb{C}^{p_{d-1} \times (q_{d-1}-s_d)}$ (part with only zeros below) and $\boldsymbol{M}^2_{d-1} \in \mathbb{C}^{p_{d-1} \times s_d}$ (part with $\boldsymbol{X}_d$ below). The matrices $\boldsymbol{X}_d \in \mathbb{R}^{m_d \times s_d}$ and $\boldsymbol{Y}_d \in \mathbb{R}^{m_d \times t_d}$ correspond to the "new" block rows of $\boldsymbol{M}_d$. We can rewrite (6), after partitioning $\boldsymbol{V}_d$, as

$$\begin{bmatrix} \boldsymbol{M}_{d-1}\boldsymbol{Z}_{d-1} & \boldsymbol{0} \\ \boldsymbol{X}_d \boldsymbol{Z}^2_{d-1} & \boldsymbol{Y}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^1_d \\ \boldsymbol{V}^2_d \end{bmatrix} = \boldsymbol{0}.$$

Hence, we find the orthonormal matrix $\boldsymbol{V}_d$ as a numerical basis matrix of a null space:

$$\begin{bmatrix} \boldsymbol{X}_d \boldsymbol{Z}^2_{d-1} & \boldsymbol{Y}_d \end{bmatrix} \boldsymbol{V}_d = \boldsymbol{0}.$$

Now, we can update the numerical basis matrix $\boldsymbol{Z}_d \in \mathbb{C}^{q_d \times n_d}$ of the null space as

$$\boldsymbol{Z}_d = \begin{bmatrix} \boldsymbol{Z}_{d-1}\boldsymbol{V}^1_d \\ \boldsymbol{V}^2_d \end{bmatrix}.$$

Algorithm 3 Recursive-recursive null space based approach

```
1: while $d^* + d_g$ is not yet reached do
2:     Update the block Macaulay matrix $\boldsymbol{M}_{d-1}$ to $\boldsymbol{M}_d$.
3:     Construct a numerical basis matrix $\boldsymbol{Z}_d$ from $\boldsymbol{Z}_{d-1}$.
4:     Recursively check the rank structure of $\boldsymbol{Z}_d$ — if there is a large enough gap in $\boldsymbol{Z}_d$, then $d^* + d_g$ has been reached.
5: end while
6: Use Algorithm 2 to compute the solutions of the MEP.
```

For a more detailed explanation about the properties of this recursive computation and extensive numerical experiments, we point the interested reader to [6].

B. *Recursive determination of the rank structure*

The numerical basis matrix $\boldsymbol{Z}_d$ of the null space consists of a series of degree block rows, which we need to consider iteratively in order to reveal the rank structure of the null space (i.e., to find a gap zone that can accommodate the shift). We can interpret $\boldsymbol{Z}_d$ in every iteration as a block row matrix $\boldsymbol{R}_d \in \mathbb{C}^{q_d \times n_d}$ and apply a second recursive algorithm:

$$\boldsymbol{Z}_d := \boldsymbol{R}_d = \begin{bmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{d-1} \\ \boldsymbol{A}_d \end{bmatrix},$$

where for degree d the basis matrix $\boldsymbol{Z}_d$ consists of $d+1$ blocks $\boldsymbol{A}_i \in \mathbb{C}^{v_i \times n_d}$, for $i = 0, \ldots, d$, with a different number of rows v_i for every i.

Consider a block row matrix $\boldsymbol{R}_{i-1} \in \mathbb{C}^{q_{i-1} \times n_d}$ and a numerical basis matrix $\boldsymbol{U}_{i-1} \in \mathbb{C}^{q_{i-1} \times w_{i-1}}$ of its null space, such that

$$\boldsymbol{R}_{i-1}\boldsymbol{U}_{i-1} = \boldsymbol{0}. \qquad (7)$$

When we append a new block $\boldsymbol{A}_i$ to obtain $\boldsymbol{R}_i$, we know that there exists an orthonormal matrix $\boldsymbol{V}_i \in \mathbb{C}^{w_{i-1} \times w_i}$, such that

$$\underbrace{\begin{bmatrix} \boldsymbol{R}_{i-1} \\ \boldsymbol{A}_i \end{bmatrix}}_{\boldsymbol{R}_i} \boldsymbol{U}_{i-1}\boldsymbol{V}_i = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{A}_i \boldsymbol{U}_{i-1} \end{bmatrix} \boldsymbol{V}_i = \boldsymbol{0},$$

because of (7). The matrix $\boldsymbol{U}_i$, on the one hand, corresponds to a numerical basis matrix of the null space of the matrix $\boldsymbol{W}_i = \boldsymbol{A}_i \boldsymbol{U}_{i-1} \in \mathbb{C}^{v_i \times w_{i-1}}$. The matrix product $\boldsymbol{U}_i = \boldsymbol{U}_{i-1}\boldsymbol{V}_i = \prod_{j=0}^{i} \boldsymbol{V}_j \in \mathbb{R}^{n_d \times w_i}$, on the other hand, is a numerical basis matrix of the null space of the block row matrix $\boldsymbol{R}_i$.

By monitoring the change of w_i, for $i = 0, \ldots, d$, we can recursively reveal the rank structure of $\boldsymbol{Z}_d$ for a particular degree d (so we need to apply this recursive technique in every iteration). We refer again to [6] for the details of this recursive computation.

C. *Recursive-recursive null space based approach*

Algorithm 3 combines both recursive techniques into a recursive-recursive approach to solve MEPs.

Algorithm 4 Sparse-recursive null space based approach

```
1: while $d^* + d_g$ is not yet reached do
2:     Construct a numerical basis matrix $\boldsymbol{Z}_d$ from $\boldsymbol{Z}_{d-1}$ (without building $\boldsymbol{M}_d$).
3:     Recursively check the rank structure of $\boldsymbol{Z}_d$ — if there is a large enough gap in $\boldsymbol{Z}_d$, then $d^* + d_g$ has been reached.
4: end while
5: Use Algorithm 2 to compute the solutions of the MEP.
```

D. Sparse-recursive null space based approach

Note that Algorithm 3 stores in every iteration the block Macaulay matrix, while this matrix contains in every block row the same coefficient matrices and many zeros. We can instead construct a numerical basis matrix $\boldsymbol{Z}_d$ in every iteration based only on $\boldsymbol{Z}_{d-1}$ and the coefficient matrices of the MEP. We have proposed in [6] an algorithm to avoid the explicit construction of the block Macaulay matrix, which allows us to develop a sparse algorithm that only stores a (full) basis matrix of the null space (see Algorithm 4).

V. Numerical examples

We revisit in this section Examples 1 and 2, in order to illustrate the recursive-recursive and sparse-recursive algorithm.

A. Least-squares realization problem

We consider the quadratic two-parameter eigenvalue problem from Example 1 constructed from a series of $N = 6$ random data points $\boldsymbol{y}$, which results in 14×13 coefficient matrices $\boldsymbol{A_\omega}$. The MEP in (2) has a positive-dimensional solution set at infinity, meaning that the nullity of its corresponding block Macaulay matrix does not stabilize. Therefore, in every iteration, we need to compute a basis matrix of the null space and check its rank structure. A block Macaulay matrix of degree $d = 18$ suffices to find a *large enough* gap zone (with $d_g = 1$) and to deflate the positive-dimensional solution set at infinity via a column compression. We apply Algorithms 1, 3, and 4 to the identification problem and obtain the results in Table I. The recursive-recursive and sparse-recursive algorithm are much faster than the standard approach, while resulting in more or less the same residual error[4]. Furthermore, Algorithm 4 requires less memory than the other two algorithms.

B. Least-squares ARMA model identification problem

Next, we solve the ARMA model identification problem applied to a series of $N = 8$ random data points $\boldsymbol{y}$ (Example 2). The MEP consists of four 23×22 coefficient matrices $\boldsymbol{A_\omega}$. Table II contains similar results as the previous example[4]: the recursive-recursive and sparse-recursive algorithm are 400 and 700 times faster than standard approach, respectively.

[4]We ran all the computations on Skylake nodes with two Xeon Gold 6140 processors working at 2.3 GHz and calculated the residual error by substituting the computed eigenvalues and eigenvectors in the MEP and computing the norm of the residual vector.

TABLE I
Comparison of the standard, recursive-recursive, and sparse-recursive approach to solve the least-squares realization problem (Example 1 – Section V-A).

Algorithm	Computation time	Memory	Residual error
standard	672.32 s	503.840 MB	5.16×10^{-14}
recr.-recr.	8.89 s	503.840 MB	1.24×10^{-12}
sparse-recr.	6.68 s	34.36 MB	1.48×10^{-13}

TABLE II
Comparison of the standard, recursive-recursive, and sparse-recursive approach to solve the least-squares ARMA model identification problem (Example 2 – Section V-B).

Algorithm	Computation time	Memory	Residual error
standard	30 996.92 s	3.76 GB	2.34×10^{-14}
recr.-recr.	71.66 s	3.80 GB	3.54×10^{-13}
sparse-recr.	44.01 s	0.18 GB	1.12×10^{-13}

VI. Conclusion and future work

In this letter, we developed a double recursive algorithm, and a sparse adaptation, to solve an MEP via the null space of the block Macaulay matrix constructed from the coefficient matrices of that MEP. By exploiting the structure and sparsity, we were able to obtain impressive reductions in computation time and memory usage compared to the standard approach, while keeping more or less the same accuracy.

In future work, we want to translate these recursive and sparse techniques to the column space of the block Macaulay matrix [5]. Furthermore, we currently investigate how to replace the second recursive algorithm by more efficient procedures to reveal the rank structure (e.g., URV-algorithms [8]).

References

[1] C. Vermeersch and B. De Moor, “Globally optimal least-squares ARMA model identification is an eigenvalue problem,” *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 1062–1067, 2019.

[2] B. De Moor, “Least squares realization of LTI models is an eigenvalue problem,” in *Proc. of the 18th European Control Conference (ECC)*, Naples, Italy, 2019, pp. 2270–2275.

[3] B. Plestenjak, C. I. Gheorghiu, and M. E. Hochstenbach, “Spectral collocation for multiparameter eigenvalue problems arising from separable boundary value problems,” *Journal of Computational Physics*, vol. 298, pp. 585–601, 2015.

[4] N. J. Higham, S. D. Mackey, and F. Tisseur, “The conditioning of linearizations of matrix polynomials,” *SIAM Journal of Matrix Analysis and Applications*, vol. 28, no. 4, pp. 1005–1028, 2006.

[5] C. Vermeersch and B. De Moor, “Two complementary block Macaulay matrix algorithms to solve multiparameter eigenvalue problems,” KU Leuven, Leuven, Belgium, Tech. Rep., 2022, submitted to Linear Algebra and Applications (LAA), https://ftp.esat.kuleuven.be/pub/stadius/cvermeer/21-108.pdf.

[6] ——, “Recursive algorithms to update a numerical basis matrix of the null space of the block row, (banded) block Toeplitz, and block Macaulay matrix,” KU Leuven, Leuven, Belgium, Tech. Rep., 2022, submitted to SIAM Journal on Matrix Analysis and Applications (SIMAX), https://ftp.esat.kuleuven.be/pub/stadius/cvermeer/21-111.pdf.

[7] K. Batselier, P. Dreesen, and B. De Moor, “The geometry of multivariate polynomial division and elimination,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 102–125, 2013.

[8] G. B. Adams, M. F. Griffin, and G. W. Stewart, “Direction-of-arrival estimation using the rank-revealing URV decomposition,” in *Proc. of the 1991 International Conference on Acoustics, Speech, and Signal Processing (ICASSP 91)*, Toronto, Canada, 1991, pp. 1385–1388.