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

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

***Abstract*—We present two double recursive block Macaulay matrix algorithms to solve multiparameter eigenvalue problems (MEPs). In earlier work, we have developed a non-recursive approach that finds 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 approach requires an iterative increase of the degree of the block Macaulay matrix: in order to determine whether the null space contains all the (affine) solutions of the MEP, we need to compute a basis matrix of the null space for every degree and check its dimension or rank structure. In this letter, we employ a recursive/sparse technique to compute a basis matrix of the null space of the block Macaulay matrix and a recursive technique to perform the necessary rank checks. We provide two system identification examples to show our improvements in computation time and memory usage.**

***Index Terms*—Numerical algorithms, 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 to solve MEPs via a multidimensional realization problem in its structured null space [1], [2], [5]. 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 have to make sure that the structured null space contains all the (affine) solutions. This requires two alternating steps: we need (i) to iteratively increase the degree of the block Macaulay matrix and construct a numerical basis matrix of its null space and (ii) to check whether we can retrieve all the affine solutions from that basis matrix. When the solution set is zero-dimensional, we can monitor the dimension of the basis matrix (as is the case in [5]). However, when the MEP has a positive-dimensional solution set at infinity, the dimension of the basis matrix does not stabilize and we need to check its rank structure to determine if we can solve the MEP with that basis matrix (otherwise we need to increase the degree again).

In this letter, we employ two different recursive[1] techniques in order to tackle MEPs that appear, for example, in a system identification context. Our *contribution* is three-fold:

- we present two double recursive algorithms to solve MEPs more efficiently,
- we show how to deflate positive-dimensional solution sets at infinity (a difficulty that arises sometimes in system identification problems),
- and we discuss how to exploit the shift polynomials that appear in the multidimensional realization problems.

Although the key ingredients of the two double recursive algorithms have already been discussed in [6], we bring them now together by recognizing that a basis matrix of the null space of the block Macaulay matrix is a block row matrix: we use (i) a recursive/sparse technique to construct a numerical basis matrix and we apply (ii) a recursive technique to check the rank structure of that basis matrix. In particular, the sparse adaptation is very useful, because it avoids the explicit construction of a large block Macaulay matrix. We solve two system identification examples with a positive-dimensional solution set at infinity to demonstrate our contribution.

The remainder of this letter proceeds as follows: In Section II, we define (rectangular) MEPs and give three examples. Next, in Section III, we construct the block Macaulay matrix and show how a multidimensional realization problem in its null space yields the (affine) solutions of that MEP. Afterwards, we develop in Section IV two double recursive algorithms, which are illustrated in Section V. Finally, we conclude this letter and discuss future research ideas in Section VI.

Christof Vermeersch (corresponding author) and Bart De Moor are with the Center for Dynamical Systems, Signal Processing, and Data Analytics (STADIUS), Dept. of Electrical Engineering (ESAT), KU Leuven. 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 LRD bilateral industrial projects, in part by Flemish Government agencies: FWO (EOS project G0F6718N (SeLMA), SBO project S005319, infrastructure project I013218N, and TBM project T001919N), EWI (Flanders AI Research Program), and VLAIO (Baekeland PhD mandate HBC.2019.2204 and HBC.2021.0076), and in part by the European Commission (ERC Advanced Grant under grant 885682). The work of Christof Vermeersch was supported by the FWO Strategic Basic Research fellowship under grant SB/1SA1319N.

[1]We do not use *recursive* in its strict computer science meaning, but to denote an algorithm that performs the same steps on different input values.

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 $\boldsymbol{z} \in \mathbb{C}^{l \times 1} \setminus \{\boldsymbol{0}\}$, so that

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

where the summation runs over all the multi-indices $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ of the monomials $\boldsymbol{\lambda}^{\boldsymbol{\omega}} = \prod_{i=1}^{n} \lambda_i^{\omega_i}$ and coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} = \boldsymbol{A}_{(\omega_1, \ldots, \omega_n)}$. The n-tuples $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ and (non-zero) vectors $\boldsymbol{z}$ are the eigentuples 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]:

$$\hat{y}_k = \boldsymbol{C} \boldsymbol{A}^k \boldsymbol{x}_0, \tag{2}$$

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], it 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

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\lambda_1, \lambda_2)\, \boldsymbol{z} = \big(& \boldsymbol{A}_{00} + \boldsymbol{A}_{10}\lambda_1 + \boldsymbol{A}_{01}\lambda_2 + \\ & \boldsymbol{A}_{20}\lambda_1^2 + \boldsymbol{A}_{11}\lambda_1\lambda_2 + \boldsymbol{A}_{02}\lambda_2^2 \big) \boldsymbol{z} = \boldsymbol{0}, \end{aligned} \tag{3}$$

with the 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 globally optimal least-squares ARMA model identification problem. The stationary points of the related optimization problem are the solutions of an MEP [1]. 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]:

$$y_k + \alpha y_{k-1} = e_k + \gamma e_{k-1}, \tag{4}$$

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

$$\boldsymbol{\mathcal{M}}(\alpha, \gamma)\, \boldsymbol{z} = \left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{01}\gamma + \boldsymbol{A}_{02}\gamma^2 \right) \boldsymbol{z} = \boldsymbol{0}, \tag{5}$$

with the coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}} \in \mathbb{R}^{(3N-1) \times (3N-2)}$ as described in [1], yields the stationary points.

Example 3: The method of separation of variables applied to some PDEs also leads to MEPs [3]. For example, the solution of the three-dimensional Helmholtz equation $\nabla \boldsymbol{u} + k^2 \boldsymbol{u} = \boldsymbol{0}$ in ellipsoidal coordinates leads to three wave equations, which can be translated via spectral collocation into a linear three-parameter eigenvalue problem [3]:

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\lambda, \mu, \eta)\, \boldsymbol{z} = \big(& \boldsymbol{A}_{000} + \boldsymbol{A}_{100}\lambda \\ & + \boldsymbol{A}_{010}\mu + \boldsymbol{A}_{001}\eta \big) \boldsymbol{z} = \boldsymbol{0}. \end{aligned} \tag{6}$$

The eigenvalues λ and μ are separation constants, while the eigenvalues η are related to the wave numbers k. We do not elaborate on the construction of the $2N^3 \times N^3$ coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}}$ (when using N collocation points), but we refer to [3] for the classical square problem setting and to [5] for the translation into a rectangular MEP.

III. Non-recursive approach

In this section, we sketch the non-recursive null space based block Macaulay matrix approach to solve MEPs (see [5] for a more detailed exposition and an alternative column space based approach). After an intuitive construction of the block Macaulay matrix (Section III-A), we show that its structured null space yields the (affine) solutions of the MEP (Section III-B). Moreover, we explain how to select a useful shift polynomial (Section III-C) and how to deflate the solutions (or positive-dimensional solution set) at infinity (Section III-D).

A. Block Macaulay matrix

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]. 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 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 the coefficient matrices of 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 4: If we revisit the quadratic MEP in (5),

$$\left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{01}\gamma + \boldsymbol{A}_{02}\gamma^2 \right) \boldsymbol{z} = \boldsymbol{0}, \tag{7}$$

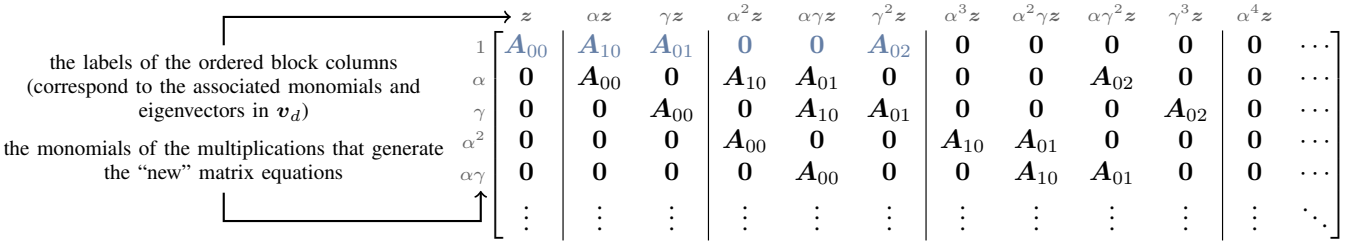

Fig. 1. The block Macaulay matrix $\boldsymbol{M}_d$ of the quadratic two-parameter eigenvalue problem in Example 4. The coefficient matrices of the seed equation, i.e., the generating MEP, are indicated in blue. Vertical lines denote the different degree blocks.

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

$$\begin{aligned} \alpha\left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{01}\gamma + \boldsymbol{A}_{02}\gamma^2\right)\boldsymbol{z} &= \boldsymbol{0} \\ \gamma\left(\boldsymbol{A}_{00} + \boldsymbol{A}_{10}\alpha + \boldsymbol{A}_{01}\gamma + \boldsymbol{A}_{02}\gamma^2\right)\boldsymbol{z} &= \boldsymbol{0}. \end{aligned} \tag{8}$$

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

$$\underbrace{\alpha, \gamma}_{d_R=1}, \underbrace{\alpha^2, \alpha\gamma, \gamma^2}_{d_R=2}, \underbrace{\alpha^3, \alpha^2\gamma, \ldots}_{d_R\geq 3} \tag{9}$$

and organize the coefficient matrices in a block Macaulay matrix $\boldsymbol{M}_d$ as in Fig. 1. The actual structure of the block Macaulay matrix depends on the chosen multivariate monomial ordering [5].

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

$$\boldsymbol{M}_d \underbrace{\begin{bmatrix} \boldsymbol{z} \\ \hline \boldsymbol{z}\lambda_1 \\ \vdots \\ \boldsymbol{z}\lambda_n \\ \hline \boldsymbol{z}\lambda_1^2 \\ \vdots \end{bmatrix}}_{\boldsymbol{v}_d} = \boldsymbol{0}. \tag{10}$$

The vector $\boldsymbol{v}_d$ is a vector in the (right) null space of $\boldsymbol{M}_d$ and has a special block multivariate Vandermonde structure, because of the monomial ordering of the block columns of the block Macaulay matrix[2]. 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* (see below).

B. Affine solutions in the structured null space

Initially, we consider an MEP that only has m_a simple and affine solutions (i.e., all solutions have algebraic multiplicity equal to one and are non-infinite). When we iteratively increase the degree d of the block Macaulay matrix $\boldsymbol{M}_d$, we notice that 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 $\boldsymbol{v}_d|_{(j)}$ ($j = 1, \ldots, m_a$) 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. As explained thoroughly in [5], the (affine) null space of the block Macaulay matrix has a special structure:

Proposition 1: The (affine) null space of the block Macaulay matrix is **block multi-shift-invariant**, which means that if we select a block row of a basis matrix of the null space and shift (or multiply) this block row by a polynomial in the eigenvalues, then we obtain another block row of that basis matrix (when it can *accommodate* the shift polynomial).

The degree d is *large enough* when the basis matrix can *accommodate* the shift polynomial: when we shift the rth degree block of a basis matrix by a shift polynomial of degree d_g, the degree of the basis matrix must be at least $r + d_g > d^*$.

When we consider a shift polynomial $g(\lambda_1, \ldots, \lambda_n)$, this multiplicative shift property corresponds mathematically to

$$\underbrace{\boldsymbol{S}_g\boldsymbol{V}_d}_{\text{after shift}} = \underbrace{\boldsymbol{S}_1\boldsymbol{V}_d}_{\text{before shift}}\,\boldsymbol{D}_g, \tag{11}$$

where the diagonal matrix $\boldsymbol{D}_g \in \mathbb{C}^{m_a \times m_a}$ contains the evaluations of $g(\lambda_1, \ldots, \lambda_n)$ 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 (the shifted block rows need to contain m_a linearly independent rows). The matrix $\boldsymbol{S}_g \in \mathbb{R}^{s \times q_d}$, on the other hand, simply selects the block rows obtained after the multiplicative shift.

Example 5: To clarify, we consider a two-parameter eigenvalue problem and shift (or multiply) the first two degree blocks ($r \leq 1$) with $g(\lambda_1, \lambda_2) = 2\lambda_1 + 3\lambda_2$ (so, $d_g = 1$ and $d \geq 2 = 1 + 1$ to *accommodate* the shift polynomial):

$$\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{z}\lambda_1 \\ \boldsymbol{z}\lambda_2 \end{bmatrix} \xrightarrow{2\lambda_1 + 3\lambda_2} \begin{bmatrix} 2\boldsymbol{z}\lambda_1 + 3\boldsymbol{z}\lambda_2 \\ 2\boldsymbol{z}\lambda_1^2 + 3\boldsymbol{z}\lambda_1\lambda_2 \\ 2\boldsymbol{z}\lambda_1\lambda_2 + 3\boldsymbol{z}\lambda_2^2 \end{bmatrix}.$$

The shift results in three combinations of block rows of $\boldsymbol{V}_d$. The corresponding row selection/combination matrices are

$$\boldsymbol{S}_1 = \begin{bmatrix} \boldsymbol{I}_{l\times l} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{l\times l} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{l\times l} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{12}$$

[2]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).

and

$$\boldsymbol{S}_g = \begin{bmatrix} \boldsymbol{0} & 2\boldsymbol{I}_{l\times l} & 3\boldsymbol{I}_{l\times l} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{I}_{l\times l} & 3\boldsymbol{I}_{l\times l} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 2\boldsymbol{I}_{l\times l} & 3\boldsymbol{I}_{l\times l} \end{bmatrix}, \quad (13)$$

with $\boldsymbol{I}_{l\times l} \in \mathbb{N}^{l\times l}$ the identity matrix.

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, we work with a numerical basis matrix $\boldsymbol{Z}_d \in \mathbb{C}^{q_d\times m_a}$ of $\boldsymbol{M}_d$ instead. A linear transformation exists 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 (11) into a solvable rectangular GEP,

$$(\boldsymbol{S}_g\boldsymbol{Z}_d)\,\boldsymbol{T} = (\boldsymbol{S}_1\boldsymbol{Z}_d)\,\boldsymbol{T}\boldsymbol{D}_g, \quad (14)$$

where $\boldsymbol{T}$ contains the eigenvectors and $\boldsymbol{D}_g$ the eigenvalues of the matrix pencil $(\boldsymbol{S}_g\boldsymbol{Z}_d, \boldsymbol{S}_1\boldsymbol{Z}_d)$. 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 (affine) solutions of the MEP.

Influence of multiplicity larger than one: Affine and isolated solutions can have a multiplicity larger than one. This poses no problem to the above-described approach (see [5]).

C. About special shift polynomials in applications

One question still remains unanswered: "How do we choose the shift polynomial(s)?" Since the block multivariate Vandermonde matrix is ill-conditioned, shifting with each of the eigenvalues λ_i provides a reliable way of obtaining the different eigenvalues of the MEP through the matrices $\boldsymbol{D}_{\lambda_i}$ (see [5]). Furthermore, in some applications, selecting a special shift polynomial may yield an additional benefit. Since the eigenvalues of (14) correspond with the evaluations of the shift polynomial in the different solutions, we can use this GEP to evaluate a polynomial, for example, the objective function of the underlying optimization problem or an additional constraint on the eigenvalues. A shift polynomial of high degree d_g, however, implies using a higher degree d.

Example 6: When solving (6), we search for the solution with the smallest wave number k [3]. If we shift with $g(\lambda, \mu, \eta) = k$, then methods like the inverse power method converge automatically to the smallest evaluation of that shift polynomial; hence, give us the "optimal" solution.

D. Solutions at infinity/positive-dimensional solution set

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 Examples 1 and 2) sometimes even 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).

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 find at least one additional linearly independent row per degree block (as long as degree $d \leq d^*$). The linearly independent rows that correspond to the standard monomials associated with the affine solutions stabilize at their respective positions from a certain degree d^* on. The linearly independent rows that correspond to the standard monomials associated with the solutions at infinity, on the other hand, keep moving to higher degree blocks when we further increase the degree $d > d^*$. When the solution set at infinity is positive-dimensional, more linearly independent rows keep appearing in the higher degree blocks (see Fig. 2). A gap zone in the rows of $\boldsymbol{Z}_d$ without any additional linearly independent rows emerges (at $d = d^*$). Similar to the affine case, the degree d is *large enough* when the basis matrix can *accommodate* the shift polynomial, which means now that the gap zone must be able to *accommodate* the shift polynomial (a shift polynomial of degree d_g requires a gap zone of d_g degree blocks), so that we can deflate the (infinitely many) solutions at infinity via a column compression [5]. We simply replace $\boldsymbol{Z}_d$ in (14) by the compressed basis matrix $\boldsymbol{W}_{11}$.

Algorithm 1 gives an overview of the different steps to compute the (affine) solutions of an MEP via the non-recursive null space based approach, which uses Algorithm 2 to retrieve the solutions from a basis matrix $\boldsymbol{Z}_d$ when the degree d is *large enough*. In order to determine whether d is *large enough* in the case of a positive-dimensional solution set at infinity, we need

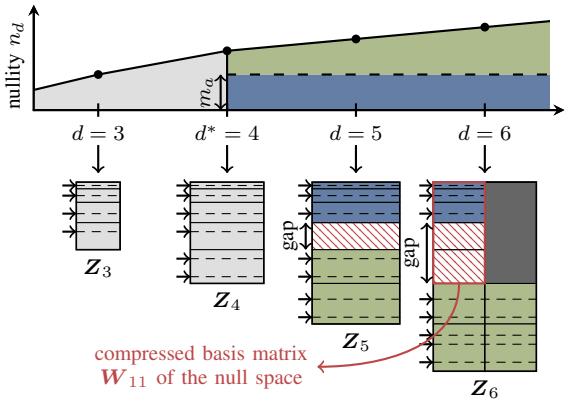

Fig. 2. The nullity of the null space of the block Macaulay matrix $\boldsymbol{M}_d$ grows as its degree d increases. However, at a certain degree $d = d^*$ (in this example $d^* = 4$), the affine zone of the basis matrix stabilizes. From that degree on, some linearly independent rows of the basis matrix $\boldsymbol{Z}_d$ 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 (infinitely many) solutions at infinity can then be deflated via a column compression [5].

Algorithm 1 Non-recursive null space based approach

1: **while** gap is smaller than d_g degree blocks **do**
2: Construct the block Macaulay matrix $\boldsymbol{M}_d$ and compute a numerical basis matrix $\boldsymbol{Z}_d$ of its null space.
3: Check the rank structure of $\boldsymbol{Z}_d$ to determine if a gap exists of d_g degree blocks.
4: **end while**
5: Use Algorithm 2 to compute the solutions of the MEP.

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

$$(\boldsymbol{S}_g \boldsymbol{W}_{11})\boldsymbol{T} = (\boldsymbol{S}_1 \boldsymbol{W}_{11})\boldsymbol{T}\boldsymbol{D}_g,$$

where $\boldsymbol{S}_1$, $\boldsymbol{S}_g$, $\boldsymbol{T}$, and $\boldsymbol{D}_g$ are defined as in (14).
3: Retrieve the solutions from the block multivariate Vandermonde basis $\boldsymbol{V}_d = \boldsymbol{W}_{11}\boldsymbol{T}$ and/or $\boldsymbol{D}_g$.

Algorithm 3 Double recursive null space based approach

1: **while** gap is smaller than d_g degree blocks **do**
2: Update the block Macaulay matrix $\boldsymbol{M}_d$ from $\boldsymbol{M}_{d-1}$ and compute a numerical basis matrix $\boldsymbol{Z}_d$ from $\boldsymbol{Z}_{d-1}$ *(avoid construction of $\boldsymbol{M}_d$ in the sparse adaptation)*.
3: Recursively check the rank structure of $\boldsymbol{Z}_d$ to determine if a gap exists of d_g degree blocks.
4: **end while**
5: Use Algorithm 2 to compute the solutions of the MEP.

to check the rank structure (i.e., the linearly independent rows) of the basis matrix to verify if it contains a gap zone of d_g degree blocks. The fact that the required degree $d^* + d_g$ is not known in advance (and can no longer be verified by monitoring the nullity) is the main difficulty of the non-recursive null space based approach. By means of two recursive techniques, we try to ease this computational burden in Section IV.

IV. Double recursive approach

Instead of re-computing $\boldsymbol{Z}_d$ for every degree, we can use a recursive technique to compute $\boldsymbol{Z}_d$ based on $\boldsymbol{Z}_{d-1}$ (Section IV-A). Furthermore, we can also check the rank structure via a recursive technique (Section IV-B). We combine these two techniques in a double recursive block Macaulay matrix algorithm and propose a sparse adaptation (Section IV-C). For more details, we refer the interested reader to [6].

A. Recursive computation of a null space basis matrix

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):

$$\begin{bmatrix}\boldsymbol{M}_{d-1} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}_{d-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{t_d \times t_d}\end{bmatrix} = \boldsymbol{0}. \tag{15}$$

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

$$\underbrace{\begin{bmatrix}\boldsymbol{M}_{d-1}^1 & \boldsymbol{M}_{d-1}^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}_d & \boldsymbol{Y}_d\end{bmatrix}}_{\boldsymbol{M}_d}\underbrace{\begin{bmatrix}\boldsymbol{Z}_{d-1}^1 & \boldsymbol{0} \\ \boldsymbol{Z}_{d-1}^2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{t_d \times t_d}\end{bmatrix}}_{\boldsymbol{N}_d}\boldsymbol{V}_d = \boldsymbol{0}, \tag{16}$$

where we have split the block Macaulay matrix $\boldsymbol{M}_{d-1}$ into $\boldsymbol{M}_{d-1}^1 \in \mathbb{R}^{p_{d-1} \times (q_{d-1} - s_d)}$ (part with only zeros below) and $\boldsymbol{M}_{d-1}^2 \in \mathbb{R}^{p_{d-1} \times s_d}$ (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 (16), after partitioning $\boldsymbol{V}_d$ according to the columns of $\boldsymbol{N}_d$, as

$$\begin{bmatrix}\boldsymbol{M}_{d-1}\boldsymbol{Z}_{d-1} & \boldsymbol{0} \\ \boldsymbol{X}_d\boldsymbol{Z}_{d-1}^2 & \boldsymbol{Y}_d\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_d^1 \\ \boldsymbol{V}_d^2\end{bmatrix} = \boldsymbol{0}. \tag{17}$$

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

$$\underbrace{\begin{bmatrix}\boldsymbol{X}_d\boldsymbol{Z}_{d-1}^2 & \boldsymbol{Y}_d\end{bmatrix}}_{\boldsymbol{W}_d}\boldsymbol{V}_d = \boldsymbol{0}, \tag{18}$$

and we combine $\boldsymbol{N}_d$ and $\boldsymbol{V}_d$ into $\boldsymbol{Z}_d \in \mathbb{C}^{q_d \times n_d}$:

$$\boldsymbol{Z}_d = \begin{bmatrix}\boldsymbol{Z}_{d-1}\boldsymbol{V}_d^1 \\ \boldsymbol{V}_d^2\end{bmatrix}. \tag{19}$$

B. Recursive check of the rank structure

The numerical basis matrix $\boldsymbol{Z}_d$ of the null space consists of a series of degree blocks, which we need to consider iteratively in order to identify a gap zone of d_g degree blocks. We can interpret $\boldsymbol{Z}_d$ for every degree d as a block row matrix $\boldsymbol{R}_d \in \mathbb{C}^{q_d \times n_d}$ with growing blocks:

$$\boldsymbol{Z}_d := \boldsymbol{R}_d = \begin{bmatrix}\boldsymbol{B}_0 \\ \boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{B}_d\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}_{d-1} \\ \boldsymbol{B}_d\end{bmatrix}, \tag{20}$$

where the basis matrix $\boldsymbol{Z}_d$ consists of $d+1$ blocks $\boldsymbol{B}_i \in \mathbb{C}^{v_i \times n_d}$ $(i = 0, \ldots, d)$ 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 $\boldsymbol{U}_{i-1} \in \mathbb{C}^{n_d \times w_{i-1}}$ of its null space, such that

$$\boldsymbol{R}_{i-1}\boldsymbol{U}_{i-1} = \boldsymbol{0}. \tag{21}$$

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

$$\underbrace{\begin{bmatrix}\boldsymbol{R}_{i-1} \\ \boldsymbol{B}_i\end{bmatrix}}_{\boldsymbol{R}_i}\boldsymbol{U}_{i-1}\boldsymbol{V}_i = \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{B}_i\boldsymbol{U}_{i-1}\end{bmatrix}\boldsymbol{V}_i = \boldsymbol{0}, \tag{22}$$

because of (21). The matrix $\boldsymbol{V}_i$, on the one hand, corresponds to a numerical basis matrix of the null space of the matrix $\boldsymbol{B}_i\boldsymbol{U}_{i-1} \in \mathbb{C}^{v_i \times w_{i-1}}$. The matrix product $\boldsymbol{U}_i = \boldsymbol{U}_{i-1}\boldsymbol{V}_i = \prod_{j=0}^{i}\boldsymbol{V}_j \in \mathbb{C}^{n_d \times w_i}$, 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 subsequent i, we can recursively reveal the rank structure of $\boldsymbol{Z}_d$ (for a particular degree d). When $n_d - w_i$ remains the same for d_g block row matrices, we have a gap zone of d_g degree blocks. We need to apply this recursive technique for every degree d, since the structure of $\boldsymbol{Z}_{d-1}$ does not contain any useful information about the structure of $\boldsymbol{Z}_d$, but we are able to retrieve the rank structure more efficiently than by re-computing the rank for every $\boldsymbol{R}_i$. Note that we compute a basis matrix $\boldsymbol{U}_i$, while we only need to track the rank/nullity of $\boldsymbol{R}_i$.

C. Two double recursive algorithms

Algorithm 3 combines both recursive techniques into a double recursive null space based approach. However, it still stores for every degree the block Macaulay matrix, while this matrix contains in every block row the same coefficient matrices and many zeros. We have proposed in [6] a sparse adaptation to avoid the explicit construction of the block Macaulay matrix in Algorithm 3, which constructs a numerical basis matrix $\boldsymbol{Z}_d$ in every iteration based only on $\boldsymbol{Z}_{d-1}$ and the coefficient matrices of the MEP.

V. Numerical examples

We now revisit Examples 1 and 2 in order to illustrate the two double recursive algorithms[3].

A. Least-squares realization problem

We consider the MEP in Example 1 constructed from a series of $N = 6$ random data points $\boldsymbol{y}$, which results in 14×13 coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}}$. This problem has a positive-dimensional solution set at infinity, so we need to compute a basis matrix of the null space for every degree and check its rank structure. A block Macaulay matrix of degree $d = 24$ has a gap zone that can *accommodate* the shift and allows us to deflate the positive-dimensional solution set at infinity via a column compression. Table I compares the computation time and maximum residual error of the different combinations of standard/recursive/sparse techniques. The recursive-recursive and sparse-recursive algorithm are much faster than the standard-standard algorithm (non-recursive approach), while resulting in more or less the same residual errors[4]. In this example, we shift with $g(\lambda_1, \lambda_2) = \lambda_1^2 + \lambda_2^2$ ($d_g = 2$). Since the eigenvalues of (14) are the evaluations of the shift polynomial in the different solutions, we can use MATLAB's `eigs` to only obtain the eigentuple with the smallest 2-norm[5].

B. ARMA model identification problem

Next, we solve the ARMA model identification problem in Example 2 applied to a series of $N = 8$ random data points $\boldsymbol{y}$. The MEP consists of 23×22 coefficient matrices $\boldsymbol{A}_{\boldsymbol{\omega}}$. Table II contains similar results as the previous example[4]: the recursive-recursive and sparse-recursive algorithm are 435 and 725 times faster than the standard-standard algorithm, respectively. The sparse-recursive algorithm, in particular, requires much less memory than the other algorithms. In this example, the full construction of a 20769×21780 block Macaulay matrix ($d = 43$) takes $3.62\,\mathrm{GB}$, while the sparse adaptation only retains the coefficient matrices ($24.28\,\mathrm{kB}$).

[3]Since Example 3 has a zero-dimensional solution set, we can check whether the degree d is *large enough* by monitoring the dimension of the null space. This example does not require a double recursive approach.

[4]We ran all our computations on a MacBook Pro with M1 CPU (2020) working at $3.2\,\mathrm{GHz}$ and calculated the residual error of a solution by substituting the computed solution in the MEP and computing the norm of the residual vector.

[5]Ideally, we would like to shift with the objective function of the underlying optimization problem. However, due to the high degree of this objective function, the computational cost of constructing a large gap zone is not worth the benefit of a special shift function. Alternatively, we could also incorporate the objective function in the MEP as an additional eigenvalue and shift with this new eigenvalue [2].

TABLE I
Results of the different combinations of techniques to solve the least-squares realization problem (Section V-A).

Combination	Computation time	Maximum residual error
standard-standard	$106.68\,\mathrm{s}$	6.54×10^{-14}
standard-recursive	$94.76\,\mathrm{s}$	6.54×10^{-14}
recursive-standard	$17.21\,\mathrm{s}$	1.41×10^{-14}
recursive-recursive	$2.62\,\mathrm{s}$	1.41×10^{-14}
sparse-recursive	$2.30\,\mathrm{s}$	7.15×10^{-14}

TABLE II
Results of the different combinations of techniques to solve the ARMA model identification problem (Section V-B).

Combination	Computation time	Maximum residual error
standard-standard	$31\,223.95\,\mathrm{s}$	5.16×10^{-14}
standard-recursive	$27\,951.57\,\mathrm{s}$	5.16×10^{-14}
recursive-standard	$323.00\,\mathrm{s}$	1.24×10^{-12}
recursive-recursive	$69.41\,\mathrm{s}$	1.24×10^{-12}
sparse-recursive	$41.74\,\mathrm{s}$	1.48×10^{-13}

VI. Conclusion and future work

In this letter, we combined two existing recursive techniques into a double recursive algorithm (and its sparse adaptation) to solve MEPs via the null space of the block Macaulay matrix. By exploiting structure and sparsity, we obtained impressive reductions in computation time and memory usage compared to the existing non-recursive approach, while keeping more or less the same accuracy. In one of our numerical examples, we observed a factor 725 improvement in computation time compared to the non-recursive approach and noticed that the sparse implementation avoids the construction of a 20769×21780 block Macaulay matrix.

In future work, we want to translate these recursive and sparse techniques to the complementary column space based approach [5]. Furthermore, we currently investigate how to replace the second recursive technique by more efficient procedures to reveal the rank structure (e.g., URV-algorithms [7]).

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, 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, https://ftp.esat.kuleuven.be/pub/stadius/cvermeer/21-111.pdf.

[7] 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.