Lossless Systems Storage Function

Kothyari, Ashish; Praagman, Cornelis; Belur, Madhu N.

Published in:
IEEE Transactions on Circuits and Systems I - Regular papers

DOI:
10.1109/TCSI.2018.2835528

Publication date:
2018

Lossless systems storage function: new results and numerically-stable and non-iterative computational methods

Ashish Kothyari, Cornelis Praagman and Madhu N. Belur

***Abstract*—In this paper we formulate and prove new results in the context of storage functions for lossless systems: we use these results to propose new algorithms to compute the storage function. The computation of the storage function for the lossless case is not possible using conventional ARE based algorithms, though the storage function itself is well-defined. This is because a certain 'regularity condition' on the feedthrough term in the i/s/o representation of the lossless system does not hold. We formulate new results about the storage function matrix for the lossless case and use them to propose non-iterative and stable algorithms to compute the storage function directly from different representations of the given system, namely: a kernel representation, transfer function and an i/s/o representation of the system. Across the methods, for randomly generated transfer functions, we compare (a) the computational effort (in flops), (b) the computation time using numerical experiments, and (c) the computational error.**

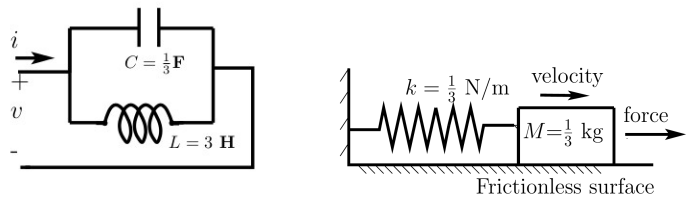
Keywords: Algebraic Riccati Equation (ARE), subspace intersection algorithms, Zassenhaus algorithm, lossless positive real systems.

1. Introduction

For many physical systems, the energy that can be extracted from the system is atmost the energy supplied to the system. A system like this is called a *dissipative system* in the literature. In [26], a dissipative system is defined using a so-called *storage function*. Loosely speaking, the storage system quantifies the available amount of internally stored energy which may be recovered from the system. The central question is how to calculate the stored energy within the system at a specific moment. It is well-known that for linear systems with quadratic supply rates, the storage function is a quadratic function of the states of the system. For strictly dissipative systems this function can be calculated by solving an Algebraic Riccati Equation (ARE). But for a special class of dissipative systems, namely lossless (or energy conservative systems), this is not possible as the ARE does not exist for such systems.

Lossless systems are those where for every system trajectory, the energy that can be extracted out of the system is exactly the amount supplied to the system. In the literature, the notion of 'conservative systems' has also been linked to so-called 'path-independence' of work done along any system trajectory: see [30], [27]. For this paper, we distinguish between "conservative" and "lossless" systems as follows. When the extracted and supplied energies are equal with respect to a general notion of power, then we use the term 'conservative', while we use 'lossless' when dealing with the specific notion of power defined by the so-called 'passivity/positive real supply rate': $2u^T y$, where u is the input and y is the output. Electrical circuits consisting of ideal inductors and/or capacitors have lossless behavior. Generally, lossless systems are also often a good approximation for many systems with very low resistance. A mechanical analogue for a lossless electrical system is a system consisting of only springs and masses. For example, consider the lossless system with transfer function $G(s) = \frac{3s}{s^2+1}$. This corresponds to, for example, an LC tank circuit with $C = \frac{1}{3}$F and $L = 3$H or a spring-mass system with mass $M = \frac{1}{3}$ kg and spring constant $k = \frac{1}{3}$ N/m (see Figure 1). The notion of 'losslessness' for the examples in Figure 1 is defined with respect to the following definitions of power: voltage×current for the LC-tank circuit and force×velocity for the spring-mass system.

A. Kothyari and M.N. Belur are in the Department of Electrical Engineering, Indian Institute of Technology Bombay, India. C. Praagman is with the Faculty of Economics and Business, University of Groningen, the Netherlands. Email: {ashishkothyari, belur}@iitb.ac.in, c.praagman@rug.nl.



(a) An LC circuit (b) A friction-less spring-mass system

Fig. 1: Lossless systems corresponding to the transfer function $G(s) = \frac{3s}{s^2+1}$

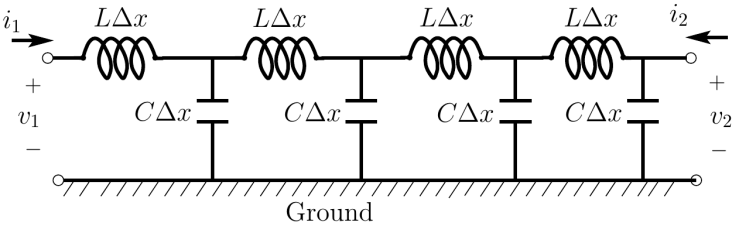


Fig. 2: A section of a transmission line

Lossless systems have been widely studied in the literature (see [28], [25]). LC tank circuits like the one described in Figure 1 are used for carrier frequency generation (see [18],[19],[10]). Moreover, transmission lines having very low resistance are analysed as an LC ladder circuit (see [29]). A section of a lossless 2-line transmission line (shown in Figure 2) resembles a two port LC network where L is the inductance per unit length and C is the capacitance per unit length of the line.

As mentioned earlier, for dissipative systems, the solutions to the Algebraic Riccati Equation (ARE) can be interpreted as storage functions. The ARE has widespread applications in both network theory (see [1, Page 259]) and in optimal control problems (see

[15], [2]). For a dissipative system with a i/s/o representation of the form $\dot{x} = Ax + Bu$, $y = Cx + Du$, the existence of the ARE depends on the nonsingularity of the term $(D + D^T)$, which we call a "regularity condition". Lossless systems admit a storage function, but, as mentioned earlier, calculation by conventional ARE solvers is not possible as the regularity condition (nonsingularity of $D + D^T$) is not satisfied. On the other hand, computing the storage function by solving the corresponding Linear Matrix Equality (LME, see Proposition 2.4 for its definition), where no condition on the feedthrough term is required, is not practical either, for again, conventional methods like interior point methods fail in solving the LME for the lossless case due to lack of *interior points* to work with (for more details, see [5]). Hence, in this paper we propose new results and numerically stable algorithms to compute the storage function for a lossless system.

Algorithms for the computation of the storage function for the lossless case involves computing the storage function either directly from a state space representation or from a transfer function representation of the system (see [1, Page 287] and [5]). These algorithms involve steps which are computationally intensive. In this paper, we first obtain results which allow us to compare two different first order representations and compute the storage function for the given lossless system. Then, using these results, we propose an algorithm to compute the storage function directly from what is called a 'kernel representation' (see Definition 2.1). We then provide stable and numerically improved algorithms for the cases when the computation of the storage function is done starting from a transfer function or an i/s/o representation.

The paper is organized as follows. Section 2 summarizes the preliminaries required in the paper. In Section 3 we propose an algorithm to compute the storage function starting from a given kernel representation. In Section 4, we propose an algorithm for computing the storage function from a given transfer function. In Section 5, we formulate algorithms that are an improvement over the dual/adjoint method proposed in [5]. Section 6 contains a comparison of the algorithms proposed in the paper and in the literature, based on their computational time and numerical accuracy. Some concluding remarks are presented in Section 7. Appendices A and B contain a summary of results and proofs used in Algorithm 1. Appendix C contains numerical examples to illustrate the algorithms presented in the paper. The rest of this section is devoted to notation.

The notation used in the paper is standard. The sets $\mathbb{R}$ and $\mathbb{C}$ denote the fields of real and complex numbers respectively. The set $\mathbb{R}[s]$ denotes the ring of polynomials in s with real coefficients. The set $\mathbb{R}^{\mathtt{w}\times\mathtt{p}}[s]$ denotes all $\mathtt{w}\times\mathtt{p}$ matrices with entries from $\mathbb{R}[s]$. We use $\bullet$ when a dimension need not be specified: for example, $\mathbb{R}^{\mathtt{w}\times\bullet}$ denotes the set of real constant matrices having $\mathtt{w}$ rows. The space $\mathfrak{C}^\infty(\mathbb{R},\mathbb{R}^{\mathtt{w}})$ stands for the space of all infinitely often differentiable functions from $\mathbb{R}$ to $\mathbb{R}^{\mathtt{w}}$, and $\mathfrak{D}(\mathbb{R},\mathbb{R}^{\mathtt{w}})$ stands for the subspace of all compactly supported functions in $\mathfrak{C}^\infty(\mathbb{R},\mathbb{R}^{\mathtt{w}})$.

2. Preliminaries

In this section we give a brief introduction to various concepts that are required to solve the problem addressed in the paper.

A. The behavioral approach

We begin with some essentials of the behavioral approach in control systems. A more detailed explanation can be found in [16].

Definition 2.1. *A behavior $\mathfrak{B}$ is defined as the subspace of $\mathfrak{C}^\infty(\mathbb{R},\mathbb{R}^{\mathtt{w}})$ consisting of all solutions to a set of linear ordinary differential equations with constant coefficients, i.e., for $R(s)\in\mathbb{R}^{\bullet\times\mathtt{w}}[s]$*

$$\mathfrak{B} := \left\{ w \in \mathfrak{C}^\infty(\mathbb{R},\mathbb{R}^{\mathtt{w}}) \mid R\left(\frac{d}{dt}\right) w = 0 \right\}. \tag{1}$$

The variable w in equation (1) is called the *manifest variable* and the set of linear differential behaviors with $\mathtt{w}$ manifest variables is denoted as $\mathfrak{L}^{\mathtt{w}}$. Equation (1) is called a *kernel representation* of the behavior $\mathfrak{B}\in\mathfrak{L}^{\mathtt{w}}$ and written as $\mathfrak{B} = \ker R(\frac{d}{dt})$. The polynomial matrix $R(s)$ is assumed to have full row rank (without loss of generality, see [16, Chapter 6]) which guarantees the existence of a nonsingular block $P(s)$ such that $R(s)T = \begin{bmatrix} P(s) & Q(s) \end{bmatrix}$, where $T\in\mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ is a permutation matrix. Conforming to this partition of $R(s)T$, w is partitioned into input and output as $T^T w = \begin{bmatrix} y \\ u \end{bmatrix}$, where u is input and y is output. Such a partition is called an *input-output* partition of the behavior. Note that this partition is not unique. An input-output partition is called *proper* if $P^{-1}Q$ is a matrix of *proper* rational functions, i.e. for each entry, the numerator degree is atmost its denominator degree. The number of components of the input depends only on $\mathfrak{B}$ and not on the input/output partition. The number of input components of $\mathfrak{B}$ is denoted as $\mathtt{m}(\mathfrak{B})$, and is called the *input cardinality*. The number of components in the output is called the *output cardinality* and is denoted as $\mathtt{p}(\mathfrak{B})$. For a behavior $\mathfrak{B}\in\mathfrak{L}^{\mathtt{w}}$, with a kernel representation $R(\frac{d}{dt})w = 0$, where $R(s)\in\mathbb{R}^{\mathtt{p}\times\mathtt{w}}[s]$ is a full row rank polynomial matrix, we have $\mathtt{p}(\mathfrak{B}) = \text{rank } R(s) = \mathtt{p}$ and $\mathtt{m}(\mathfrak{B}) = \mathtt{w}-\mathtt{p}$: see [16, Definition 3.3.1].

In the behavioral approach, a system is nothing but its behavior and thus the terms behavior/system are used interchangeably in this paper. We now define another important concept required in the paper: controllability of the system.

Definition 2.2. *A behavior $\mathfrak{B}$ is said to be controllable if for every w_1, $w_2\in\mathfrak{B}$ there exists $w_3\in\mathfrak{B}$ and $\tau > 0$ such that*

$$w_3(t) = \begin{cases} w_1(t) & \text{for} \quad t \leqslant 0, \\ w_2(t) & \text{for} \quad t \geqslant \tau. \end{cases}$$

In the paper we represent the set of all controllable behaviors with $\mathtt{w}$ variables as $\mathfrak{L}^{\mathtt{w}}_{\text{cont}}$. Analogous to the PBH test, a behavior $\mathfrak{B}$ with a minimal kernel representation $\mathfrak{B} = \ker R(\frac{d}{dt})$ is controllable if and only if $R(\lambda)$ has constant rank for all $\lambda\in\mathbb{C}$. One of the ways by which a behavior $\mathfrak{B}$ can be represented if $\mathfrak{B}$ is controllable is the so-called 'image representation'. For $M(s)\in\mathbb{R}^{\mathtt{w}\times\bullet}[s]$:

$$\mathfrak{B} := \left\{ w \in \mathfrak{C}^\infty(\mathbb{R},\mathbb{R}^{\mathtt{w}}) \mid \exists \ell \in \mathfrak{C}^\infty(\mathbb{R},\bullet) \text{ s.t. } w = M(\frac{d}{dt})\ell \right\}. \tag{2}$$

If $M(\lambda)$ has full column rank for all $\lambda\in\mathbb{C}$, then the image representation is said to be an *observable* image representation

For a behavior $\mathfrak{B}\in\mathfrak{L}^{\mathtt{w}}$, we define the supply rate as: $Q_\Sigma(w) = w^T\Sigma w$, $\Sigma\in\mathbb{R}^{\mathtt{w}\times\mathtt{w}}$. The supply rate is the rate of supply of energy to the system. Throughout the paper, we assume that for the given

behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, $\mathtt{w} = 2\mathtt{m}$, where $\mathtt{m}$ is both the input and output cardinality (see Footnote 1 for a justification) of the system. Also, for behaviors $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, we deal with supply rates induced by real symmetric constant nonsingular matrices $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$ only ($I_{\mathtt{m}}$ is the identity matrix of dimension $\mathtt{m}$), though many of the results can be generalized for other supply rates as well. For a behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, where $\mathtt{w} = 2\mathtt{m}$ and with an input/output partition as $w = \begin{bmatrix} y \\ u \end{bmatrix}$, the supply rate induced by $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$ corresponds to the passivity/positive real supply rate $2u^T y$. Thus, a lossless system is defined as follows:

Definition 2.3. *A system $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$, with[1] $\mathtt{w} = 2\mathtt{m}$, is called lossless with respect to the supply rate $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$ if:*

$$\int_{\mathbb{R}} w^T \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix} w \, dt = \int_{\mathbb{R}} 2u^T y \, dt = 0 \text{ for all } w \in \mathfrak{B} \cap \mathfrak{D}(\mathbb{R}, \mathbb{R}^{\mathtt{w}}).$$

B. The algebraic Riccati equation (ARE)

Consider a proper input-output partition (u, y) for a controllable dissipative behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}$ ($\mathtt{w} = 2\mathtt{m}$), with the following minimal i/s/o representation:

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du, \tag{3}$$

where $A \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}, B, C^T \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}$ and $D \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}$. If the system $\mathfrak{B}$ is lossless w.r.t. the supply rate $w^T \Sigma w$, then there exists a matrix $K = K^T \in \mathbb{R}^{n\times n}$ such that the following equation holds:

$$w^T \Sigma w = \frac{d}{dt} x^T K x, \text{ for all } w = \begin{bmatrix} y \\ u \\ x \end{bmatrix} \text{ satisfying equation (3).}$$

Here $x^T K x$ is defined as the storage function of the system $\mathfrak{B}$. One of the results relating the storage function of a controllable behavior and the Linear Matrix Inequality (LMI) is the *Kalman-Yakubovich-Popov* (KYP) lemma. For easy reference we present the KYP lemma in the next proposition.

Proposition 2.4. *[26] A behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\text{cont}}$ ($\mathtt{w} = 2\mathtt{m}$), with a minimal i/s/o representation as in equation (3) is dissipative w.r.t. the supply rate $w^T \Sigma w$ ($\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$) if and only if there exists a solution $K = K^T \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ to the LMI.*

$$\begin{bmatrix} A^T K + KA & KB - C^T \\ B^T K - C & -(D + D^T) \end{bmatrix} \leqslant 0. \tag{4}$$

Further, $x^T K x$ is the stored energy corresponding to $K = K^T \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$.

For systems with $D + D^T > 0$, the Schur complement with respect to $D + D^T$ in LMI (4) provides us with the algebraic Riccati inequality: $A^T K + KA + (KB - C^T)(D + D^T)^{-1}(B^T K - C) \leqslant 0$.

[1]The positive real property (Proposition 2.4) only applies to square transfer functions, i.e for systems with equal number of inputs and outputs: hence the assumption $\mathtt{w} = 2\mathtt{m}$.

C. Static relations and storage function

In [5], an algorithm was proposed to compute the storage function by extracting the 'static relations' existing between the state vectors of the system $\mathfrak{B}$ and the state vectors of the Σ-orthogonal complement of $\mathfrak{B}$ (see [27] for definition of a Σ-orthogonal complement). The following result helps in computation of the storage function starting from an i/s/o representation of $\mathfrak{B}$.

Proposition 2.5. *[5, Proposition 6.1] Consider a controllable and lossless (w.r.t. the positive real supply rate) behavior $\mathfrak{B}$ with minimal state space representation as in equation (3). Assuming the McMillan degree of $\mathfrak{B}$ is $\mathtt{n}$, and let $R(s) := sE - H$ where $E := \begin{bmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $H := \begin{bmatrix} A & 0 & B \\ 0 & -A^T & C^T \\ C & -B^T & 0 \end{bmatrix}$, where $E, H \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}$. Then the following statements hold:*

1) *The polynomial matrix $R(s)$ satisfies: $\det R(s) = 0$.*
2) *There exists a matrix $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ such that K satisfies*

$$\text{rank} \begin{bmatrix} & R(s) & \\ -K & I_{\mathtt{n}} & 0 \end{bmatrix} = \text{rank } R(s). \tag{5}$$

   where $I_{\mathtt{n}}$ is the identity matrix of size $\mathtt{n}$ and the matrix K is unique and symmetric.
3) *The matrix $K = K^T \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ of statement 2 satisfies:*

$$\frac{d}{dt} x^T K x = 2u^T y \qquad \text{for all} \qquad \begin{bmatrix} y \\ u \end{bmatrix} \in \mathfrak{B}. \tag{6}$$

Some of the algorithms for the computation of the matrix K given in [5] require the calculation of a minimal polynomial basis[2] (MPB) *twice*: once for an MPB, $M(s)$ of the matrix $R(s)$ (equation (5)) and then an MPB for an appropriate submatrix of $M(s)$. In this paper, we also focus on improving the algorithms given in [5] by avoiding the computation of the MPB and thus making the computation of K much faster and more accurate. Using equation (5), $[-K \quad I \quad 0]$ is in the row-span of the polynomial matrix $R(s)$. This fact is used to develop algorithms in Sections 5-A and 5-B to compute the storage function $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ for lossless systems. We next cover some results that are used to develop such algorithms.

D. The row-reduced-echelon form and LU factorization

A matrix $A \in \mathbb{R}^{n\times n}$ is said to be in the *row-reduced-echelon* form if the following two conditions are satisfied:

1) If row r is zero, then all rows below r are also zero.
2) If $a_{i,j}$ in A is the leading[3] row-element (also called the pivot) of the i^{th} row, then the leading row-element $a_{i+1,k}$ of the $(i+1)^{th}$ row satisfies $k > j$.

A matrix can be brought to a row-reduced-echelon form by pre-multiplication by unit lower triangular matrices, i.e. lower triangular matrices with diagonal entries equal to one [8, Chapter 3]. Due to

[2] For a polynomial matrix $R(s) \in \mathbb{R}^{\mathtt{p}\times\mathtt{w}}[s]$, with rank $R(s) = \mathtt{p}$, let $R(s)Z(s) = 0$, where $Z(s) \in \mathbb{R}^{\mathtt{w}\times(\mathtt{w}-\mathtt{p})}[s]$ be a right nullspace basis of $R(s)$ and the let the column degrees of $Z(s)$ be $d_1, d_2, \ldots, d_{\mathtt{w}-\mathtt{p}}$. If the sum of all the degrees d_i, $i \in \{1, 2, \ldots, \mathtt{w} - \mathtt{p}\}$ is minimal over the choice of all right nullspace basis, then $Z(s)$ is called a minimal polynomial basis (see [11, Section 6.5.4]).

[3] $a_{i,j}$ in A is called the leading row-element if $a_{i,j} \neq 0$, and $a_{i,\ell} = 0$ for all $\ell < j$.

the possibility of unacceptably large growth of entries, we pursue the LU factorization with so-called *partial pivoting* and we have:

$$PA = LU$$

where P is a permutation matrix, U is an upper triangular matrix and L is an unit lower triangular with entries $|\ell_{i,j}| \leqslant 1$ (see: [8, Page 115]).

E. Zassenhaus sum-intersection algorithm for subspace intersection

The Zassenhaus sum-intersection algorithm is used to calculate a basis for the intersection of two subspaces. Consider two full row rank matrices S and T with $S,T \in \mathbb{R}^{\bullet\times n}$, the algorithm (implemented using LU factorization) for computing the intersection of the row spans of S and T ($\langle S\rangle_R \cap \langle T\rangle_R$) involves the following steps summarized into a result for easy reference:

Proposition 2.6. *Let the row spans of $S,T \in \mathbb{R}^{\bullet\times n}$ be denoted by $\langle S\rangle_R$ and $\langle T\rangle_R$ respectively. Also let the dimension of the subspace $\langle S\rangle_R + \langle T\rangle_R$ be n_1 and the dimension of the intersection of the row spaces i.e. $\langle S\rangle_R \cap \langle T\rangle_R$ be n_2, then a basis for $\langle S\rangle_R \cap \langle T\rangle_R$ can be computed as follows:*

1) *Define $W := \begin{bmatrix} S & S \\ T & 0 \end{bmatrix}$ using matrices S and T.*
2) *Compute the LU factorization of W with partial pivoting.*

$$PW = LU$$

3) *The rows of the submatrix $U(n_1+1 : n_1+n_2, n+1 : 2n)$ form a basis for $\langle S\rangle_R \cap \langle T\rangle_R$.*

In Section 5-A, we propose an algorithm applying the Zassenhaus algorithm for the computation of the storage function. For more details about the Zassenhaus algorithm, see [14].

F. Computation of subspace intersection using QR factorization

In this section, we describe how the basis for intersection of two subspaces can be calculated using QR factorization. We use the following proposition in order to find the basis for intersection of two subspaces. The proof is skipped since it is straightforward.

Proposition 2.7. *Let the column spans of two matrices $X,Y \in \mathbb{R}^{m\times\bullet}$ be denoted by $\langle X\rangle_C$ and $\langle Y\rangle_C$ respectively. Let X and Y be full column rank matrices and $\langle X\rangle_C \cap \langle Y\rangle_C \neq \{0\}$, then a basis for $\langle X\rangle_C \cap \langle Y\rangle_C$ can be computed using the following steps:*

- *Define $W := \begin{bmatrix} X & Y \end{bmatrix}$.*
- *Use QR factorization of W^T to find a full column rank matrix N such[4] that $NW^T = 0$ and rank $N = \dim\langle X\rangle_C \cap \langle Y\rangle_C$.*
- *Partition N^T in accordance with W as $N^T := \begin{bmatrix} \hat{X} \\ \hat{Y} \end{bmatrix}$.*
- *Columns of $Z := X\hat{X}$ form a basis for $\langle X\rangle_C \cap \langle Y\rangle_C$.*

In Section 5-B, we propose an algorithm which incorporates the content of this section to compute the storage function.

[4]Let $A =: QR$ where $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$, with R_1 full row rank and upper triangular, and Q an orthogonal matrix of appropriate dimensions. Let the columns of Q corresponding to the zero rows in R be N. Then N satisfies $N^TA = 0$.

3. Two variable polynomial matrix factorization and computation of storage function

In this section we provide an algorithm for computing the storage function of a lossless behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\text{cont}}$ ($\mathtt{w} = 2\mathtt{m}$), directly from its kernel representation $R(\frac{d}{dt})w = 0$, where $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$ is full row rank. Row-reducedness[5]. of $R(s)$ helps in the procedure, hence we assume this without loss of generality[6] In order to compute the storage function, we compare two first order representations of the given system $\mathfrak{B}$, one given by Proposition A.1 which is formulated directly from a given kernel representation, and the second given by Proposition A.2, which contains information of the storage function of the system $\mathfrak{B}$. See Appendix A for Propositions A.1 and A.2. In the following subsection, we describe the construction and properties of certain matrices which are required for the computation of the storage function. Since these results are of independent interest, we present them here.

A. Polynomial matrix factorization and input/output partition for lossless systems

We first describe steps for obtaining the polynomial matrix $Y(s)$ in the minimal factorization of the two variable polynomial $\Pi(\zeta,\eta) =: \frac{R(-\zeta)-R(\eta)}{\zeta+\eta}$ where $\Pi(\zeta,\eta) = Y(\zeta)^TX(\eta)$ (see Appendix A, Proposition A.1) without any numerical computation. The matrix $Y(s)$ is required for the computing the storage function (see Step 4 of Algorithm 1). The initial steps are based on [20, Remark 2.7]. Note that the coefficient matrix $\tilde{\Pi} \in \mathbb{R}^{\mathtt{m}\mathtt{N}\times\mathtt{w}\mathtt{N}}$ for the two variable polynomial $\Pi(\zeta,\eta)$ is equal to:

$$\tilde{\Pi} = \begin{bmatrix} -R_1 & -R_2 & \dots & -R_{N-1} & -R_N \\ R_2 & R_3 & \dots & R_N & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (-1)^{N-1}R_{N-1} & (-1)^{N-1}R_N & 0 & \dots & 0 \\ (-1)^NR_N & 0 & 0 & \dots & 0 \end{bmatrix}$$

Thus we have:

$$\Pi(\zeta,\eta) = \begin{bmatrix} I_{\mathtt{m}} & I_{\mathtt{m}}\zeta & \dots & I_{\mathtt{m}}\zeta^{N-1} \end{bmatrix} \tilde{\Pi} \begin{bmatrix} I_{\mathtt{w}} \\ I_{\mathtt{w}}\eta \\ \vdots \\ I_{\mathtt{w}}\eta^{N-1}. \end{bmatrix} \tag{7}$$

Now, rewriting equation (7), we obtain $\Pi(\zeta,\eta) = \tilde{Y}(\zeta)^T\tilde{X}(\eta)$, where the matrix $\tilde{Y}(s)^T := \begin{bmatrix} -I_{\mathtt{m}} & I_{\mathtt{m}}s & \dots & (-1)^NI_{\mathtt{m}}s^{N-1} \end{bmatrix}$ and $\tilde{X}(s) = \begin{bmatrix} R_1+R_2s+\cdots+R_Ns^{N-1} \\ R_2+R_3s+\cdots+R_Ns^{N-2} \\ \vdots \\ R_{N-1}+R_Ns \\ R_N \end{bmatrix}$. Notice that

[5] Let the matrix $R_{hr} \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}$ be defined as the matrix whose j^{th} row contains the highest degree coefficients of the j^{th} row of $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$, $j = 1,\cdots,\mathtt{m}$. The matrix $R(s)$ is called *row-reduced* if R_{hr} is full row rank. A polynomial matrix $P(s)$ is called column-reduced if $P(s)^T$ is row-reduced.

[6]If $R(s) \in \mathbb{R}^{\mathtt{m}\times\bullet}[s]$ is a full row rank polynomial matrix, there exists a unimodular matrix $U(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ such that the matrix $U(s)R(s) =: \hat{R}(s)$ is row-reduced.

$\tilde{X}(s) = \begin{bmatrix} \sigma_+(R(s)) \\ \sigma_+^2(R(s)) \\ \vdots \\ \sigma_+^N(R(s)) \end{bmatrix}$ where $\sigma_+ : \mathbb{R}[s] \to \mathbb{R}[s]$ is the shift-and-cut operator (see [20]). The factorization $\Pi(\zeta,\eta) = \tilde{Y}(\zeta)^T \tilde{X}(\eta)$ may not be minimal in general as there may be redundant rows in $\tilde{X}(s)$. In our case, since $R(s)$ is assumed to be row-reduced, redundancies of rows in $\tilde{X}(s)$, if any, are only due to zero rows. The construction of a minimal factorization of $\Pi(\zeta,\eta)$ is given by the following lemma. Proof for the lemma follows from the steps described above.

Lemma 3.1. *Construct $X(s)$ from $\tilde{X}(s)$ (equation (7)) by removing its zero rows and construct $Y(s)$ from $\tilde{Y}(s)$ by removing the rows of $\tilde{Y}(s)$ corresponding to the zero rows of $\tilde{X}(s)$. Then, for the two variable polynomial $\Pi(\zeta,\eta)$, a minimal factorization is $\Pi(\zeta,\eta) = Y^T(\zeta)X(\eta)$.*

In order to compute the storage function, we also require an input/output partition for $\mathfrak{B}$; this can be determined with the help of a minimal[7] output-nulling representation of $\mathfrak{B}$. In Proposition A.1, the output-nulling representation obtained in equation (14) is minimal if R_N is full row rank, but this is not true for the general case. Hence we first describe the steps for obtaining a minimal output-nulling representation from the representation given in equation (14). First assume $R(s)$ has rows permuted appropriately to have row degrees d_i, $i \in \{1,2,\ldots,\mathtt{m}\}$ arranged as $d_1 \leqslant d_2 \leqslant \cdots \leqslant d_{\mathtt{m}}$ where $d_{\mathtt{m}} = N$. From Lemma 3.1, we observe that:

- The first set of zero row(s) of $\tilde{X}(s)$ are contained in the submatrix $\sigma_+^{d_1+1}(R(s))$ of $\tilde{X}(s)$.
- The columns corresponding these zero row(s) are the columns of the submatrix $(-1)^{d_1+1} I_{\mathtt{m}} s^{d_1}$ of $\tilde{Y}(s)^T$ which are removed for the construction of $Y(s)$.

Now consider the first order representation (E,F,G) in equation (13) and the submatrices R_{d_1}, R_{d_2}, and so on of G and $Y_{d_1}^T$, $Y_{d_2}^T$, and so on of E. Note that

- Corresponding to the row(s) having highest degree d_1 in R_{d_1}, we have zero row(s) in $Y_{d_1}^T$.
- Corresponding to the row(s) having the highest degree d_2 in R_{d_2}, we have zero row(s) in $Y_{d_2}^T$ and so on.

Hence, using a suitable permutation matrix $P \in \mathbb{R}^{\mathtt{m}(\mathtt{N}+1)\times\mathtt{m}(\mathtt{N}+1)}$, (E,F,G) obtained in equation (13) is transformed as:

$$\begin{bmatrix} Y' \\ 0 \end{bmatrix} \frac{d}{dt}x + \begin{bmatrix} F_1' \\ F_2' \end{bmatrix} x + \begin{bmatrix} R' \\ R_{hr} \end{bmatrix} w = 0 \tag{8}$$

where R_{hr} is the highest row degree coefficient matrix upto sign and $PE = \begin{bmatrix} Y' \\ 0 \end{bmatrix}$, $PF = \begin{bmatrix} F_1' \\ F_2' \end{bmatrix}$ and $PG = \begin{bmatrix} R' \\ R_{hr} \end{bmatrix}$. Since the matrix E is full column rank (as $Y(\zeta)^T X(\eta)$ is a minimal factorization), multiplying equation (8) by the matrix $\begin{bmatrix} L_1 & 0 \\ 0 & I_{\mathtt{m}} \end{bmatrix}$ where L_1 is any left inverse of Y', we obtain:

$$\frac{d}{dt}x = -L_1F_1'x - L_1R'w \text{ and } 0 = -F_2'x + -R_{hr}w \tag{9}$$

Note that Lemma 3.1 and the steps described above work even if the given system $\mathfrak{B}$ is not lossless. In the lossless case, one can use the minimal output-nulling representation in equation (9) to define a proper input/output partition for a lossless system $\mathfrak{B}$ which is required for the computation of the storage function. Further note that the permutation matrix corresponding to an input/output partition for the lossless system $\mathfrak{B}$ may not be unique, but some interesting properties are hold for all such permutation matrices. Since these properties are of independent interest, we formulate and prove them in the following lemma.

Lemma 3.2. *Consider a behavior $\mathfrak{B} \in \mathfrak{L}_{\text{cont}}^{\mathtt{w}}$, with $\mathtt{w} = 2\mathtt{m}$, lossless with respect to the supply rate $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$ and having a minimal output-nulling representation:*

$$\dot{x} = Ax + Bw \ , \ 0 = Cx + Dw.$$

where $A \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$, $B, C^T \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}$ and $D \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}$. Let the permutation matrix $Tw = \begin{bmatrix} y \\ u \end{bmatrix}$, $T \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$, be such that (u,y) is a proper input/output partition for $\mathfrak{B}$. Partition T into $\begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$ such that $T_{ij} \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}$ for $i,j \in \{1,2\}$. Then T_{ij} satisfy:

1) *$T_{ij} = T_{ij}^T$ for $i,j \in \{1,2\}$.*
2) *$T_{11} = T_{22}$ and $T_{12} = T_{21}$.*

Proof. For proving the above lemma, we crucially use a result from [24, Lemma 14] after the following manipulation on C and D which shows that $w^T\Sigma w = 0$ for all $w \in ker(D)$. To see this, suppose $Tw = \begin{bmatrix} y \\ u \end{bmatrix}$, $T \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$, is a proper input/output partition for $\mathfrak{B}$. Also let $\dot{x} = A'x + B'u$, $y = C'x + D'u$ be a corresponding i/s/o representation. The matrix C' can be constructed from C and D as follows. Since D is full row rank, we have $DT = \begin{bmatrix} D_{11} & D_{12} \end{bmatrix}$ where $D_{11}, D_{12} \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}$ and D_{12} is nonsingular. Thus $C' := -D_{12}^{-1}C$. Since the system is lossless with respect $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$, we have $w^T\Sigma w = 2u^Ty = \frac{d}{dt}x^TKx$, where $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ is the storage function. Hence, we have $w^T\Sigma w = (A'x + B'u)^TKx + x^TK(A'x + B'u)$. Using the LME (equation (4)), we obtain $w^T\Sigma w = 2u^TC'x$. Corresponding to the set of w's that lie in the nullspace of the matrix D, we have $Cx = 0$. Thus, $w^T\Sigma w = 0$ for all $w \in ker(D)$. This allows us to apply [24, Lemma 14] to our case. Let D_i be the i^{th} column of the matrix D. Then there exists a selection of $\mathtt{m}$ linearly independent columns $\{c_1, c_2, \ldots, c_{\mathtt{m}}\}$ of the matrix D, such that for all $1 \leqslant i \leqslant \mathtt{m}$, either D_i or $D_{i+\mathtt{m}}$ belongs to $\{c_1, c_2, \ldots, c_{\mathtt{m}}\}$, but not both D_i and $D_{i+\mathtt{m}}$. Thus the permutation matrix T swaps the i^{th} and $(i+\mathtt{m})^{th}$ row of the vector w depending upon whether the $(i+\mathtt{m})^{th}$ is taken as input or not. Thus every matrix T has the following structure:

$$T = \begin{cases} T_{i,i} = 1 \text{ and } T_{(i+\mathtt{m}),(i+\mathtt{m})} = 1, & \text{if } i^{th} \text{ and } (i+\mathtt{m})^{th} \text{ rows are not swapped} \\ T_{i,(i+\mathtt{m})} = 1 \text{ and } T_{(i+\mathtt{m}),i} = 1, & \text{if } i^{th} \text{ and } (i+\mathtt{m})^{th} \text{ rows are swapped} \end{cases}$$

[7] For a controllable behavior $\mathfrak{B}$, an output-nulling representation (A,B,C,D) is called *minimal* if (C,A) is observable and the matrix D is full row rank [22, Proposition 8.5].

Thus, $T_{11} = T_{22}$, $T_{12} = T_{21}$ and $T_{ij} = T_{ij}^T$ for $i, j \in \{1,2\}$. □

B. Computation of the storage function

We focus on the computation of the storage function, and hence in this section, we give only a gist of the results used to formulate Algorithm 1: the relevant results and proofs are in Appendices A and B. We obtain a first order representation for the given behavior $\mathfrak{B}$ using Proposition A.1 (from [20]). While Proposition A.1 provides a first order representation for any behavior, for lossless systems, there exists another first order representation given by Proposition A.2 (from [17]) containing information about the storage function. By comparing both these representations (see Theorem 3.4 below), we obtain the storage function. See Appendix A for Proposition A.1 and Proposition A.2. We first state the following result which helps later in the extraction of the storage function using Theorem 3.4. See Appendix A for the definition of certain terms used in the results below.

Lemma 3.3. *For a behavior $\mathfrak{B} \in \mathfrak{L}_{\text{cont}}^{\mathtt{w}}$, with a row-reduced minimal kernel representation $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$ of degree N, let $R(s)T = \begin{bmatrix} P(s) & -Q(s) \end{bmatrix}$, $P(s), Q(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ be a proper input-output partition. Let $x := X_{\mathtt{w}}(\frac{d}{dt})w$, $X_{\mathtt{w}}(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{w}}[s]$ be the minimal state map constructed using the shift and cut map and let a minimal factorization of the two variable polynomial $\Pi(\zeta,\eta)$ be $\Pi(\zeta,\eta) =: Y(\zeta)^T X_{\mathtt{w}}(\eta)$ (Lemma 3.1). Define $M(s) := T \begin{bmatrix} Q(-s)^T \\ P(-s)^T \end{bmatrix}$ and $X_\ell(s) := X_{\mathtt{w}}(s)M(s)$ and expand $Y(s) = Y_0 + Y_1 s + \cdots + Y_{N-1}s^{N-1}$. Suppose, $\mathfrak{B}$ is lossless w.r.t. the supply rate induced by $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$, then the following hold.*

1) *The degree of $X_\ell(s)$ is $N-1$.*
2) *Let $X_\ell(s) = X_0^\ell + X_1^\ell s + X_2^\ell s^2 + \cdots + X_{N-1}^\ell s^{N-1}$, then the matrices $\hat{X}, \hat{Y} \in \mathbb{R}^{\mathtt{m}N\times\mathtt{n}}$ defined as:*

$$\hat{X} := \begin{bmatrix} (X_0^\ell)^T \\ (X_1^\ell)^T \\ \vdots \\ (X_{N-1}^\ell)^T \end{bmatrix}, \ \hat{Y} := \begin{bmatrix} Y_0^T \\ \vdots \\ Y_{N-1}^T \end{bmatrix}$$

have the same left nullspace, i.e. for any $v \in \mathbb{R}^{\mathtt{m}N}$,

$$v^T\hat{X} = 0 \Leftrightarrow v^T\hat{Y} = 0.$$

Proof. See Appendix B for the proof of Lemma 3.3. □

We now use the above lemma to prove the following result which helps to determine the unique storage function for a given lossless system. The following result is one of the main results of this paper.

Theorem 3.4. *Consider a behavior $\mathfrak{B} \in \mathfrak{L}_{\text{cont}}^{\mathtt{w}}$, with a kernel representation $R(\frac{d}{dt})w = 0$, $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$, with $R(s)$ row-reduced and with degree equal to N. Let $x := X_{\mathtt{w}}(\frac{d}{dt})w$, $X_{\mathtt{w}}(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{w}}[s]$ be the minimal state map constructed using the shift and cut map and let a minimal factorization of $\Pi(\zeta,\eta)$ (see Appendix A) be $\Pi(\zeta,\eta) =: Y(\zeta)^T X_{\mathtt{w}}(\eta)$. Assume, $R(s)T = \begin{bmatrix} P(s) & -Q(s) \end{bmatrix}$, $P(s), Q(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ be a proper input/output partition for the behavior $\mathfrak{B}$, where $T \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ is a permutation matrix. Define $M(s)$, $X_\ell(s)$, $\hat{X}$ and $\hat{Y}$ as in Lemma 3.3 and define $K := \hat{X}^\dagger\hat{Y}$ with $\hat{X}^\dagger$ any left inverse of $\hat{X}$. Suppose $\mathfrak{B}$ is lossless w.r.t. the supply rate $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix}$ then,*

1) *$K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ defined above is symmetric, i.e. $K = K^T$, and*
2) *x^TKx is the unique storage function, i.e. K satisfies $\frac{d}{dt}x^TKx = 2u^Ty$ for all $\begin{bmatrix} y \\ u \end{bmatrix} \in \mathfrak{B}$.*

Proof. From the kernel representation $R(s)$ and using Proposition A.1, we formulate a minimal first order representation for $\mathfrak{B}$: $E_1\dot{x} + F_1x + G_1w = 0$, $E_1, F_1 \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{n}}$, $G_1 \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{w}}$, where x is the state variable. The polynomial matrix $R(s)$ can be written as $R(s) = \begin{bmatrix} P(s) & -Q(s) \end{bmatrix} T^T$. Since $M(s)$ is an observable image representation, we use $M(s)$ and Proposition A.2 to formulate another first order representation for $\mathfrak{B}$: $E_2\dot{z} + F_2z + G_2w = 0$, $E_2, F_2 \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{n}}$, $G_2 \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{w}}$, where z is the state variable. The state variable x is obtained as $x = X_{\mathtt{w}}(\frac{d}{dt})w$ and is minimal. Also, the state variable z is obtained as $z = X_\ell(\frac{d}{dt})\ell$ and is minimal. We also know that $X_\ell(s) = X_{\mathtt{w}}(s)M(s)$, hence $x = z$. Since first order representation $(E_1.F_1, G_1)$ and (E_2, F_2, G_2) are both minimal and $ker(\begin{bmatrix} \frac{d}{dt}E_1 - F_1 & G_1 \end{bmatrix})$ and $ker(\begin{bmatrix} \frac{d}{dt}E_2 - F_2 & G_2 \end{bmatrix})$ are equal, the matrices (E_1, F_1, G_1) and (E_2, F_2, G_2) are related as:

$$E_1 = LE_2, \ F_1 = LF_2 \text{ and } G_1 = LG_2$$

for some nonsingular $L \in \mathbb{R}^{(N+1)\mathtt{m}\times(N+1)\mathtt{m}}$. Now we focus on the matrices G_1 and G_2. The matrix G_1 equals:

$$G_1 = \begin{bmatrix} -R_0 \\ R_1 \\ -R_2 \\ \vdots \\ (-1)^{N+1}R_N \end{bmatrix} = \begin{bmatrix} [-P_0 & Q_0]T^T \\ [P_1 & -Q_1]T^T \\ [-P_2 & Q_2]T^T \\ \vdots & \vdots \\ [(-1)^{N+1}P_N & (-1)^NQ_N]T^T \end{bmatrix}.$$

Similarly, the matrix G_2 equals:

$$G_2 = -\begin{bmatrix} M_0^T \\ M_1^T \\ M_2^T \\ \vdots \\ M_N^T \end{bmatrix}\Sigma = \begin{bmatrix} [-P_0 & Q_0]T^T \\ [P_1 & -Q_1]T^T \\ [-P_2 & Q_2]T^T \\ \vdots & \vdots \\ [(-1)^{N+1}P_N & (-1)^NQ_N]T^T \end{bmatrix}.$$

Since $G_1 = G_2$, we conclude that matrix L is the identity matrix and hence, we have $E_1 = E_2$. Since the construction E_2 involves the storage function K corresponding to the state map $X_\ell(s)$ (see Proposition A.2), we have:

$$K = Z\begin{bmatrix} Y_0^T \\ \vdots \\ Y_{N-1}^T \end{bmatrix}$$

where Z is some left inverse of the matrix $\hat{X} = [X_0^\ell \ X_1^\ell \ \cdots \ X_{N-1}^\ell]^T$. Note that the matrix $\hat{X}$ is full column rank because the state map $X_\ell(s)$ is minimal. Since the left nullspaces of matrix $\hat{X}$ and $\hat{Y}$ ($\hat{Y} = [Y_0 \ \cdots \ Y_{N-1}]^T$) are the same (Lemma 3.3), the matrix K is unique and $K = \hat{X}^\dagger\hat{Y}$ and $\hat{X}^\dagger$ can be any left inverse of $\hat{X}$. Since, in the matrix K is required to be symmetric in the formulation of Proposition A.2 and also of [17], hence K obtained here is also symmetric. □

We propose an algorithm to compute the storage function for the given lossless system $\mathfrak{B}$ starting from a kernel representation of the given system and a proper input/output partition.

Algorithm 1 : Two variable polynomial matrix factorization based computation of storage function.

Input: Lossless behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\text{cont}}$, ($\mathtt{w} = 2\mathtt{m}$) with a minimal kernel representation $R(\frac{d}{dt})w = 0$, $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$, $R(s)$ is row-reduced and the permutation matrix T corresponding to some proper input/output partition $Tw = \begin{bmatrix} y \\ u \end{bmatrix}$.

Output: Storage function $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ such that $w^T\Sigma w = 2u^T y = \frac{d}{dt}x^T K x$.

1: Compute a first order representation for $\mathfrak{B}$ using Proposition A.1 and Lemma 3.1. Let N be the degree of $R(s)$.

2: Define $M(s) := T\begin{bmatrix} -Q(-s)^T \\ P(-s)^T \end{bmatrix}$ where $R(s)T =: \begin{bmatrix} P(s) & -Q(s) \end{bmatrix}$ is a proper input/output partition of the system.

3: Construct the minimal state map $x := X_{\mathtt{w}}(\frac{d}{dt})w$, $X_{\mathtt{w}}(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{w}}[s]$, by applying the shift-and-cut operation to $R(s)$ (Lemma 3.1) and $X_\ell(s) := X_{\mathtt{w}}(s)M(s)$, $X_\ell(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$.

4: Let $\Pi(\zeta,\eta) := \frac{R(-\zeta)-R(\eta)}{\zeta+\eta}$ and factorize $\Pi(\zeta,\eta) =: Y(\zeta)^T X_{\mathtt{w}}(\eta)$ where $Y(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$.

5: Expand $X_\ell(s) = X_0^\ell + X_1^\ell s + X_2^\ell s^2 + \cdots + X_{N-1}^\ell s^{N-1}$ and $Y(s)^T = Y_0^T + Y_1^T s + Y_2^T s^2 + \cdots + Y_{N-1}^T s^{N-1}$

6: $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ is $K := \begin{bmatrix} (X_0^\ell)^T \\ (X_1^\ell)^T \\ \vdots \\ (X_{N-1}^\ell)^T \end{bmatrix}^\dagger \begin{bmatrix} Y_0^T \\ \vdots \\ Y_{N-1}^T \end{bmatrix}$ where $\bullet^\dagger$ denotes the pseudo-inverse.

4. Matrix fraction description and two variable polynomial matrix factorization based computation of the storage function

In this section, we introduce an algorithm to compute the storage function for a lossless system from the given transfer function matrix of the system. We consider only lossless positive real transfer functions. A transfer matrix $G(s) \in \mathbb{R}(s)^{\mathtt{m}\times\mathtt{m}}$ is called *Lossless Positive Real* if $G(s)$ is positive real and $G(s)+G(-s)^T = 0$. A lossless positive real transfer matrix $G(s) := \frac{n_{i,j}(s)}{d_{i,j}(s)}$, $i,j \in \{1,2,\cdots,\mathtt{m}\}$ is strictly proper[8] and from the definition of a positive real transfer function (see [1, Page 51]), we deduce that a necessary condition for $G(s)$ to be a lossless positive real transfer matrix is:

$$\text{roots of } d_{i,j} \subseteq \text{ roots of } d_{i,i} \cap d_{j,j} \text{ (counted with multiplicity).} \qquad (10)$$

First we briefly go through the steps involved in computing the storage function for a given lossless system $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\text{cont}}$ where $\mathtt{w} = 2\mathtt{m}$ from its transfer function representation. Suppose $G(s) = N(s)D(s)^{-1}$ where $N(s), D(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ is a right co-prime matrix fraction description (MFD) of $G(s)$. Then, $w = \begin{bmatrix} N(\frac{d}{dt}) \\ D(\frac{d}{dt}) \end{bmatrix}\ell$ becomes an observable image representation for the given system. Since $G(s)+G(-s)^T = 0$, we note that $\begin{bmatrix} D(-\frac{d}{dt})^T & N(-\frac{d}{dt})^T \end{bmatrix} w = 0$ is a controllable kernel representation for the given system. Assuming that $M(s) := \begin{bmatrix} N(s) \\ D(s) \end{bmatrix}$ (and hence[9] $D(s)$) is column-reduced (see definition in Footnote 5), by constructing two first order representations using the kernel and image representations obtained from the right co-prime MFD and then by applying Theorem 3.4, we obtain the storage function for the system.

For computing the right co-prime MFD

- Construct $\hat{R}(s) := \begin{bmatrix} P(s) & -Q(s) \end{bmatrix}$, $\hat{R}(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s], P(s), Q(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ where $P(s) := \text{diag}\ \{d_{1,1}(s), d_{2,2}(s), \cdots, d_{\mathtt{m},\mathtt{m}}(s)\}$ and $Q(s) = P(s)G(s)$, where $G(s) \in \mathbb{R}(s)^{\mathtt{m}\times\mathtt{m}}$ is the given transfer function and $d_{i,i}(s)$ is the denominator of the $(i,i)^{th}$ element in $G(s)$, $i = 1,\cdots,\mathtt{m}$. Equation (10) ensures that $Q(s)$ is a polynomial matrix.
- Compute a minimal polynomial basis for $\hat{R}(s)$ using the algorithm in [31].
- If $\hat{M}(s) := \begin{bmatrix} M_1(s) \\ M_2(s) \end{bmatrix}$, $\hat{M}(s) \in \mathbb{R}^{\mathtt{w}\times\mathtt{m}}[s]$, $M_1(s), M_2(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ is a minimal nullspace basis of $\hat{R}(s)$, then, $M_1(s)M_2(s)^{-1}$ is the desired right co-prime MFD.

Another advantage of using the algorithm in [31] is that $\hat{M}(s)$ (and hence $M_2(s)$) obtained is column-reduced since the nullspace basis is a minimal polynomial basis. In case of SISO systems, a kernel representation is constructed as $R(s) = \begin{bmatrix} d(s) & -n(s) \end{bmatrix}$ where $G(s) = \frac{n(s)}{d(s)}$ is the SISO transfer function as $n(s)$ and $d(s)$ are co-prime (see [9, Section 5]).

Algorithm 2 : Two variable polynomial matrix factorization based computation of storage function (MFD based)

Input: Behavior $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\text{cont}}$, ($\mathtt{w} = 2\mathtt{m}$), with a lossless positive real transfer matrix $G(s) \in \mathbb{R}(s)^{\mathtt{m}\times\mathtt{m}}$ and $G(s) = N(s)D(s)^{-1}$, $N(s), D(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ and are right co-prime, $D(s)$ is column-reduced.

Output: The storage function $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ such that $2u^T y = \frac{d}{dt}x^T K x$.

1: $R(s) := \begin{bmatrix} D(-s)^T & N(-s)^T \end{bmatrix}$ and $M(s) := \begin{bmatrix} N(s) \\ D(s) \end{bmatrix}$. Let N be the highest degree in both $R(s)$ and $M(s)$.

2: Construct the state map $x := X_{\mathtt{w}}(\frac{d}{dt})w$, $X_{\mathtt{w}}(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{w}}[s]$, by applying the shift-and-cut operation to $R(s)$ (Lemma 3.1) and $X_\ell(s) := X_{\mathtt{w}}(s)M(s)$, $X_\ell(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$.

3: Expand $\Pi(\zeta,\eta) := \frac{R(-\zeta)-R(\eta)}{\zeta+\eta}$ and factorize $\Pi(\zeta,\eta) =: Y(\zeta)^T X_{\mathtt{w}}(\eta)$ where $Y(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$.

4: Let $X_\ell(s) = X_0^\ell + X_1^\ell s + X_2^\ell s^2 + \cdots + X_{N-1}^\ell s^{N-1}$ and $Y(s)^T = Y_0^T + Y_1^T s + Y_2^T s^2 + \cdots + Y_{N-1}^T s^{N-1}$.

5: $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ is $K := \begin{bmatrix} (X_0^\ell)^T \\ (X_1^\ell)^T \\ \vdots \\ (X_{N-1}^\ell)^T \end{bmatrix}^\dagger \begin{bmatrix} Y_0^T \\ \vdots \\ Y_{N-1}^T \end{bmatrix}$ where $\bullet^\dagger$ denotes the pseudo-inverse.

[8] The KYP lemma (see Proposition 2.4) requires the transfer matrix to be proper. The lossless positive real transfer matrix $G(s)$ is in fact strictly proper as it has an LC realization and can be written as [1, Page 216]:

$$G(s) = \Sigma\left(\frac{Z}{s-j\omega} + \frac{Z^*}{s+j\omega}\right) + s^{-1}C$$

where Z and $C \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}$ are all Hermitian and positive semidefinite.

[9] Since $G(s)$ is strictly proper, for each column of M, the highest degree coefficients exist only in the block $D(s)$.

5. Algorithms for Static relations extraction

As noted in Section 2-C, the algorithms proposed in [5] for the computation of the storage function requires computing the minimal polynomial basis of $R(s)$ (see Proposition 2.5). In this section we propose faster algorithms to compute the storage function. We find the intersection of the row spaces of matrices $R(\lambda_1)$ and $R(\lambda_2)$, $\lambda_1, \lambda_2 \in \mathbb{C}$ in order to compute the storage function. As noted in Proposition 2.5, $\begin{bmatrix} -K & I_{\mathtt{n}} & 0 \end{bmatrix}$ lies in the row space of the polynomial matrix $R(s)$. In order to extract K, we evaluate $R(s)$ at various $\lambda \in \mathbb{C}$ and compute the intersection of the row spaces of $R(\lambda)$'s. A notion of interpolation frequencies has been studied in [23], where spectral zeros are candidates for interpolation of a suitable two-variable polynomial matrix. For the case of lossless systems, the spectral zeros are, in fact, the whole complex plane. The precise link between the λ_i where we evaluate $R(s)$ and the interpolation frequencies in [23] needs further investigation.

A. LU Zassenhaus algorithm implementation

In this section, we propose an algorithm for computing the storage function based on LU factorization. We first evaluate $R(\lambda_i)$ where λ_i's are the roots of the polynomial $s^k - 1$ for some value of $k \in \mathbb{Z}^+$ and then extract the storage storage function from the intersection of the row spans of all $R(\lambda_i)$'s, $i = 1, \cdots, k$. In order to extract $\begin{bmatrix} -K & I & 0 \end{bmatrix}$ from the row span of $R(s) \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}[s]$ (Proposition 2.5) we propose the following algorithm.

Algorithm 3 : LU based computation of K

Input: $R(s) := sE - H \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}[s]$, a rank $2\mathtt{n}$ polynomial matrix and tolerance $\varepsilon > 0$.

Output: $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ with $x^T K x$ the storage function.

Require: Evaluate $R(\lambda_i) \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}$ at

$$\lambda_i \in \mathbb{C}, i = 1, 2, .., k \text{ which are the roots of } s^k - 1$$

for k suitably chosen depending on the accuracy ε.

1: $W := \begin{bmatrix} R(\lambda_1) & R(\lambda_1) \\ R(\lambda_2) & 0 \end{bmatrix}$, $[L, U, P] := lu(W)$
2: $D := U(2\mathtt{n}+\mathtt{m}+1 : 4\mathtt{n}+2\mathtt{m}-\ell, 2\mathtt{n}+\mathtt{m}+1 : 4\mathtt{n}+2\mathtt{m})$
3: where ℓ as the largest integer such that

$$\|U(4\mathtt{n}+2\mathtt{m}-\ell+1 : 4\mathtt{n}+2\mathtt{m}, :)\|_2 < \varepsilon.$$

4: Let c be the number of rows of D. Note that $c \geqslant n$.
5: **while** $c > n$ **do**
6: $\quad W := \begin{bmatrix} R(\lambda_d) & R(\lambda_d) \\ D & 0 \end{bmatrix}$, $[L, U, P] := LU(W)$
7: $\quad D := U(2\mathtt{n}+\mathtt{m}+1 : 2\mathtt{n}+\mathtt{m}+c-\ell, 2\mathtt{n}+\mathtt{m}+1 : 4\mathtt{n}+2\mathtt{m})$
8: $\quad \ell$ is the largest integer such that:

$$\|U(2\mathtt{n}+\mathtt{m}+c-\ell+1 : 2\mathtt{n}+\mathtt{m}+c, :)\|_2 < \varepsilon.$$

9: $\quad$ Let c be the number of rows of D and $d = 3, 4, ..., k$.
10: **end while**
11: $X := D(1 : \mathtt{n}, \mathtt{n}+1 : 2\mathtt{n})$, $Y := D(1 : \mathtt{n}, 1 : \mathtt{n})$.
12: $K := -X^{-1}Y$.

B. QR based subspace intersection method

In this section, we propose an algorithm for computing the storage function based on a QR factorization. We first evaluate $R(\lambda_i)$ where λ_i's are the roots of the polynomial $s^k - 1$ for some value of $k \in \mathbb{Z}^+$ and then extract the storage function from the intersection of the row spans of all $R(\lambda_i)$'s, $i = 1, \cdots, k$. In order to extract $\begin{bmatrix} -K & I & 0 \end{bmatrix}$ from the row span of $R(s) \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}[s]$ (Proposition 2.5) we propose the following algorithm:

Algorithm 4 : QR based computation of K

Input: $R(s) := sE - H \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}[s]$, a rank $2\mathtt{n}$ polynomial matrix and tolerance $\varepsilon > 0$.

Output: $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$ with $x^T K x$ the storage function.

Require: Evaluate $R(\lambda_i) \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times(2\mathtt{n}+\mathtt{m})}$, at

$$\lambda_i \in \mathbb{C}, i = 1, 2, .., k \text{ which are the roots of } s^k - 1$$

1: $W := \begin{bmatrix} R(\lambda_1)^T & -R(\lambda_2)^T \end{bmatrix}$, $[Q, R, P] := qr(W^T)$
2: Remove the last ℓ columns of Q where ℓ is the largest integer such that:

$$\|R(4\mathtt{n}+2\mathtt{m}-\ell+1 : 4\mathtt{n}+2\mathtt{m}, :)\|_2 < \varepsilon$$

Let the removed columns form the set $U \in \mathbb{R}^{(4\mathtt{n}+2\mathtt{m})\times\ell}$.
3: Select the first $(2\mathtt{n}+\mathtt{m})$ rows of U (call them U_1) and define $V := W(:, 1 : 2\mathtt{n}+\mathtt{m})U_1$, $V \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times\ell}$.
4: $[Q, R, P] := qr(V)$. Let z be the largest integer such that:

$$\|R(2\mathtt{n}+\mathtt{m}-z+1 : 2\mathtt{n}+\mathtt{m}, :)\|_2 < \varepsilon$$

Let the first $\ell - z$ columns of Q be denoted by $\hat{Q}$ and $c := \ell - z$ and let $d = 3$. Note that $c \geqslant \mathtt{n}$.
5: **while** $c > \mathtt{n}$ **do**
6: $\quad W := \begin{bmatrix} R(\lambda_d)^T & -\hat{Q} \end{bmatrix}$, $[Q, R, P] := qr(W^T)$
7: Remove the last ℓ columns of Q where ℓ is the largest integer such that:

$$\|R(2\mathtt{n}+\mathtt{m}+c-\ell+1 : 2\mathtt{n}+\mathtt{m}+c, :)\|_2 < \varepsilon$$

Let the removed columns form the set $U \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m}+c)\times\ell}$.
8: $\quad$ Select the first $(2\mathtt{n}+\mathtt{m})$ rows of U (call them U_1) and define $V := W(:, 1 : 2\mathtt{n}+\mathtt{m})\mathrm{U}_1$, $V \in \mathbb{R}^{(2\mathtt{n}+\mathtt{m})\times\bullet}$.
9: $\quad [Q, R, P] := qr(V)$. Let z be the largest integer such that:

$$\|R(2\mathtt{n}+\mathtt{m}-z+1 : 2\mathtt{n}+\mathtt{m}, :)\|_2 < \varepsilon$$

Let the first $\ell - z$ columns of Q be denoted by $\hat{Q}$ and $c := \ell - z$.
10: $\quad d = d + 1$.
11: **end while**
12: $D := \hat{Q}^T$
13: $X := D(1 : \mathtt{n}, \mathtt{n}+1 : 2\mathtt{n})$, $Y := D(1 : \mathtt{n}, 1 : \mathtt{n})$.
14: $K := -X^{-1}Y$.

6. Comparison of Algorithms

In this section, we compare the algorithms presented in the paper with the existing algorithms in the literature. We compare the steps taken to compute the storage function in the algorithms presented in this paper and the existing algorithms and argue that the algorithms presented in the paper are computationally

less intensive and are numerical more stable. Using numerical experiments we next compare the time taken and the numerical accuracy of the algorithms proposed in the paper with the existing algorithms and show that the algorithms presented in this paper are faster than the algorithms in the literature and compute the storage function with relatively same accuracy. To summarize, the rest of this section is a comparison of proposed algorithms and the algorithms in the literature with respect to the following criteria.

(a) Computational effort, in terms of flop count estimate: Section 6-A
(b) Computational time (for randomly generated transfer functions of various orders): Section 6-B
(c) Computational error, in particular, $\text{Err}_{\text{LMI}}(K)$ and $\text{Err}_{\text{Sym}}(K)$, formulated in equations (11) and (12): Section 6-C

A. Computational intensity comparison

The existing algorithms in the literature to compute the storage function for the lossless case are described in [1, Page 287] and in [5], where four different algorithms, namely: partial fraction expansion based method, Bezoutian based method, balancing realization and the dual/adjoint method are proposed. Below is a summary of key differences.

1) Balanced realization involves converting the given i/s/o realization to a balanced realization (controllability and observability Gramians are equal) whose storage function is $I_{\mathtt{n}}$ ($\mathtt{n}$ is the number of states).
2) The dual adjoint method involves computing a minimal polynomial basis (MPB) twice in order to compute the storage function. Computing MPB twice is computationally intensive and hence we proposed the LU and QR based algorithms (Algorithms 3 and 4) which not only avoid computing the MPB, but are also more numerically stable.
3) In case of MIMO transfer functions, the partial fraction based method involves computationally intensive steps like computing the partial fraction expansion of the transfer function. The partial fraction based method also involves inverting a matrix of order $\mathtt{n}^2$, hence the flop count of the partial fraction based method is relatively large (see Remark 6.1 for more on this).
4) The Bezoutian based method is only proposed for the SISO case. This involves computing the storage function by performing Euclidean long division or by computing the 2-D Fourier transform.
5) To compute the storage function from a given transfer function using the algorithm given in [1, Page 287], first a controllable i/s/o representation is obtained and then the storage function K is computed by inverting the controllability matrix. Using Algorithm 2, we simultaneously compute both the storage function as well as the corresponding i/s/o realization.
6) In the co-prime MFD based method (Algorithm 2), we compute a right co-prime MFD of the given transfer function. This is done by computing the minimal nullspace basis for the matrix $R(s) := \begin{bmatrix} P(s) - Q(s) \end{bmatrix}$, $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s], P(s), Q(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{m}}[s]$ where $P(s) := \text{diag}\ \{d_{1,1}(s), d_{2,2}(s), \cdots, d_{\mathtt{m},\mathtt{m}}(s)\}$ and $Q(s) = P(s)G(s)$, where $G(s) \in \mathbb{R}(s)^{\mathtt{m}\times\mathtt{m}}$ is the given transfer function and $d_{i,i}(s)$ is the denominator of the $(i,i)^{th}$ element in $G(s)$, $i = 1, \ldots, \mathtt{m}$. For computing the nullspace basis of $R(s)$, stable algorithms like the one given in [31] are employed. Also, the algorithm in [31] computes a nullspace basis which is also column-reduced and hence the state map (w.r.t. the manifest variables) and other required polynomial matrices (Step 3 of Algorithm 2) are computed without any effort.
7) Since only lossless positive real transfer matrices are considered, the storage function K is positive definite (see [1, Page 221]). Note that computing K^{-1} requires hardly any computation since, as described in Lemma 3.1, the structure of the matrix $\begin{bmatrix} Y_0 & Y_1 \cdots & Y_{N-1} & 0 \end{bmatrix}^T$, consisting of just 0's and 1's, allows finding its pseudo-inverse easily. If K is the storage function corresponding to the i/s/o representation (A,B,C), then K^{-1} is the storage function corresponding to the realization[10] $(-A^T, C^T, B^T)$.
8) For the SISO case, the computationally intensive steps in the co-prime MFD based method are: a polynomial matrix multiplication (see Step 2 of Algorithm 2) and computing K (Step 6 of Algorithm 2) which again can be done away with if one computes K^{-1}.

Method	Flop count	Computationally most intensive step	2nd most intensive step
Co-prime MFD (Algorithm 2)	$O(\mathtt{n}^3)$	Polynomial matrix multiplication (Step 2 of Algorithm 2)	Inverting a matrix of size $\mathtt{n}\times\mathtt{n}$
LU/QR factorization (Algorithms 3, 4)	$O(\mathtt{n}^3\mathtt{d})$	LU/QR factorization	Inverting a matrix of size $\mathtt{n}\times\mathtt{n}$
Partial fraction [5]	$O(\mathtt{n}log^2\mathtt{n})$ [13]	Partial fraction expansion	Inverting a diagonal matrix of size $\mathtt{n}^2$
Minimal polynomial basis [5]	$O(\mathtt{n}^3\mathtt{f}^3)$ [31, Theorem 2]	Computing minimal polynomial basis	Inverting a matrix of size $\mathtt{n}\times\mathtt{n}$
Controllability matrix [1]	$O(\mathtt{n}^3)$	Minimal realization	Inverting a matrix of size $\mathtt{n}\times\mathtt{n}$

TABLE I: Comparison of flop count

A comparison of the flop count (for the SISO case) for algorithms discussed in this paper and those in the literature is done in Table I. The terms $\mathtt{n}$, $\mathtt{d}$ and $\mathtt{f}$ in the table are defined as follows.

1) $\mathtt{n}$ degree of the denominator of the SISO transfer function and also the number of states in the system.
2) $\mathtt{d}$ is the number times the loop (steps 2 and 4 in Algorithms 3 and 4 respectively) runs.
3) $\mathtt{f}$ is the highest degree in the nullspace of $R(s)$ (equation (5)).

B. Time comparison

The plot in Figures 3 and 4 show the time taken by both the static relations extraction methods elaborated in Section 5 (Algorithms 3 and 4) and by the co-prime MFD based method (Algorithm 2) to calculate the storage function. Their time is compared with the time taken by the partial fraction based algorithm given in [5, Algorithm 7.1] for SISO systems of different orders. The experiments were carried out on an Intel Core i3 computer at 3.40 GHz with 4 GB

[10] If the matrix K satisfies the LME (4) corresponding to the realization (A,B,C), then K^{-1} satisfies the LME corresponding to the realization $(-A^T, C^T, B^T)$. Since $G(s)$ is lossless, $C(sI_{\mathtt{n}} - A)^{-1}B = -B^T(-sI_{\mathtt{n}} - A^T)^{-1}C^T = B^T(sI_{\mathtt{n}} + A^T)^{-1}C^T$.

RAM using Ubuntu 16.04 LTS operating system. The relative machine error precision is $\varepsilon \approx 2.22 \times 10^{-16}$. Open source numerical computational package Scilab 5.5 has been used to implement the algorithms. The numerical experiments for both time and computational error was computed as follows. For each system order, 10 transfer functions were generated randomly. For each of these transfer functions, the time was averaged over 20 runs to minimize effects due to other system operations. The average time/error over the 10 randomly generated transfer functions for the five algorithms (three proposed in this paper, and two from the literature) are plotted in Figures 3 and 4 for time and Figures 5-8 for computational error. It can be seen from the plot that the LU and QR based methods take approximately the same time for computing K while co-prime MFD based method requires the least amount of time to compute the storage function and is about 8 times faster than the partial fraction based method. It has been elaborated in [12] that the adjoint method in [5, Algorithm 7.4] performs slower than the LU based method and is less suitable for systems of higher orders.

We observe that the time by the partial fractions method is more than the time taken by the co-prime factorization, though the flop count (for the SISO case) partial fraction method is less. The partial fraction method involves the inversion of a matrix of size $\mathtt{n}^2$. But in the SISO case, the matrix of size $\mathtt{n}^2$ is diagonal and this significantly reduces the number of flops. The MIMO case would need further system parameters for an accurate analysis and this is described in the remark below where we compare the flop counts for the co-prime factorization and partial fraction based methods.

Remark 6.1. *If we consider a system with a MIMO transfer function $G(s)$, computation of the storage function using the partial fractions would require $O(\mathtt{n}^6)$ flop, where $\mathtt{n}$ is the number of states in the given system. This is because the partial fraction method involves inverting a matrix of size $\mathtt{n}^2$. Computing a co-prime factorization $G(s) = N(s)D(s)^{-1}$ of the transfer function would require $O(\mathtt{w}^3\mathtt{f}^3\mathtt{d})$ flop counts, where $\mathtt{w}$ is the size of the square transfer function (and is also the number of input/outputs), $\mathtt{f}$ is the degree the matrix $D(s)$ and $\mathtt{d}$ is the highest degree amongst all the denominator terms of $G(s)$. The next computationally intensive step in the co-prime factorization based method is multiplication of polynomial matrix (Step 2 of Algorithm 2). Computing $X_\ell(s) = X(s)M(s)$ (see Algorithm 2) and computing $\tilde{X}_\ell = \tilde{X}(s)M(s)$, where $\tilde{X}(s)$ is constructed using shift and cut map (see Lemma 3.1), would requires the same number of operations. The flop count for computing $\tilde{X}_\ell(s)$ is $\mathtt{w}^3\mathtt{f}^3$. Thus, for the MIMO case, the flop count required for the co-prime Factorization based method would be less than that of the partial fractions based method. The flop counts of both the algorithms are comparable only when the $\mathtt{w}$ and $\mathtt{f}$ are equal to $\mathtt{n}$, which does not happen for most of the cases as examples of large scale systems where the number to states is equal the number of inputs/outputs are rare.*

C. Comparison of the computational error

We compare the computational error of the algorithms presented in the paper and the algorithms in the literature. We compare

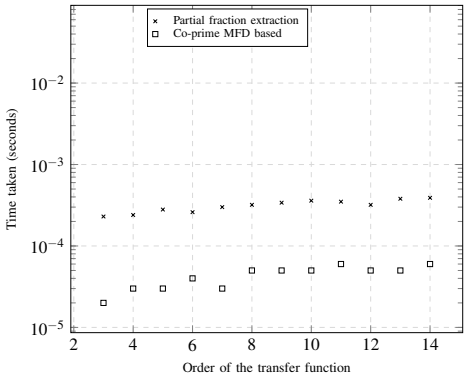


Fig. 3: Plot for time taken by algorithms versus system's order

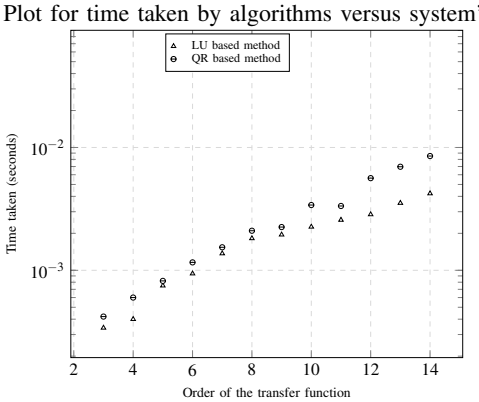


Fig. 4: Plot for time taken by algorithms versus system's order

how accurately the storage function K is computed using the co-prime MFD based method (Algorithm 2), the LU and QR based methods, the minimal polynomial basis-based method and the partial fraction based method. As discussed in Section 2-C, the symmetric matrix K calculated for the lossless case satisfies the LMI given in equation (4) with equality. Thus, the symmetric K satisfies the following equation:

$$\begin{bmatrix} A^TK + KA & KB - C^T \\ B^TK - C & 0 \end{bmatrix} = 0.$$

Thus we consider the following errors associated with computation of K.

$$\mathrm{Err}_{\mathrm{LMI}}(K) := \left\| \begin{bmatrix} A^TK + KA & KB - C^T \\ B^TK - C & 0 \end{bmatrix} \right\|_2. \quad (11)$$

$$\mathrm{Err}_{\mathrm{Sym}}(K) := \|K - K^T\|_2. \quad (12)$$

We calculate the errors $\mathrm{Err}_{\mathrm{LMI}}(K)$ and $\mathrm{Err}_{\mathrm{Sym}}(K)$ for randomly generated lossless systems. From Figures 5, 7, 6 and 8, we see that for all the five algorithms (i.e. three proposed, and two from the literature), the computational error as measured by equations (12) and (11) are comparable. The advantage remains in the computational effort as reflected by the time plots.

7. Concluding remarks

We noted that computing the storage function by finding the solutions to the LME using conventional LMI solvers is not possible in the case of lossless systems. In this paper we formulated four non-iterative and numerically stable algorithms for the computation of the storage function for lossless (and in general, energy conservative) systems. The algorithms proposed in this paper perform faster

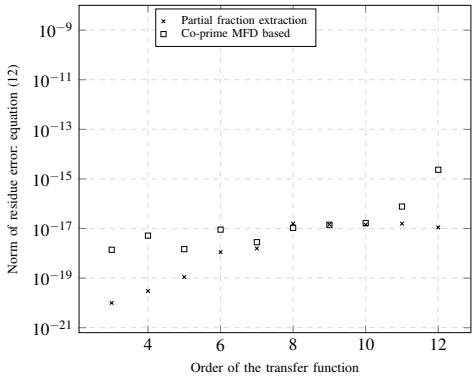


Fig. 5: Plot of $\mathrm{Err}_{\mathrm{LMI}}(K)$ For Partial Fraction based method and Co-prime based method (see equation (11)) versus system order

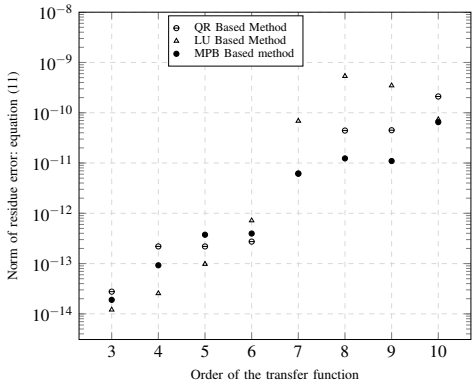


Fig. 6: Plot of $\mathrm{Err}_{\mathrm{LMI}}(K)$ (see equation (11)) versus system order

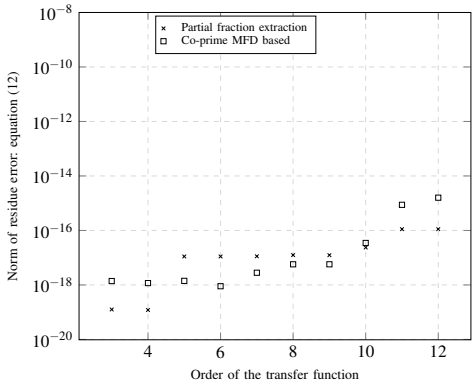


Fig. 7: Plot of $\mathrm{Err}_{\mathrm{Sym}}(K)$ For Partial Fraction based method and Co-prime based method (see equation (12)) versus system order

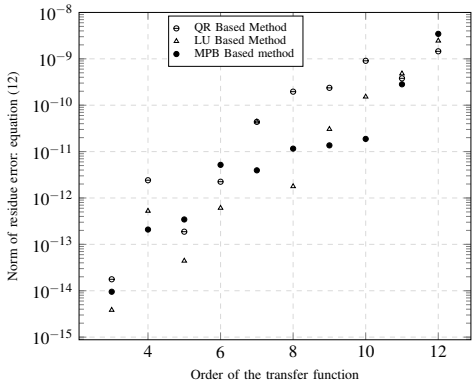


Fig. 8: Plot of $\mathrm{Err}_{\mathrm{Sym}}(K)$ (see equation (12)) versus system order

than similar algorithms available in the literature. We formulated algorithms using different starting points, Algorithm 1 uses a row-reduced minimal kernel representation of the given lossless system, Algorithm 2, uses a transfer function representation of the lossless system and Algorithms 3 and 4 start with an i/s/o representation of the given lossless system. The algorithms presented in the paper are stable, perform well in speed and accuracy and have a lower (or comparable) flop count when compared with similar algorithms present in the literature (see Table I and Remark 6.1). Our algorithms use standard numerical techniques like LU and QR (with permutation for proven numerical stability) which have established LAPACK/LINPACK routines, unlike polynomial/rational matrix algorithms whose numerical stability guarantees are a matter of current research.

Of independent interest are the results formulated and proved in Section 3-A, i.e. the procedure to obtain a minimal output-nulling representation for any system $\mathfrak{B}$ from a given row-reduced minimal kernel representation of $\mathfrak{B}$ and the properties of all permutation matrices that yield a proper input/output partition for a given lossless system (Lemma 3.2). Some numerical examples for Algorithms 2, 3 and 4 are listed in Appendix C.

A direction of further investigation is how these algorithms perform when a system is close to uncontrollable. Further, when the system is uncontrollable, the algorithms would need significant modification. These are areas of future work.

APPENDICES

APPENDIX A

In this appendix, we describe the two first order representations used in the algorithm proposed in Section 3 for the computation of the storage function. Consider a lossless system $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\mathrm{cont}}$, ($\mathtt{w} = 2\mathtt{m}$), lossless with respect to the supply rate induced by $\Sigma = \begin{bmatrix} 0 & I_{\mathtt{m}} \\ I_{\mathtt{m}} & 0 \end{bmatrix} \in \mathbb{R}^{\mathtt{w}\times\mathtt{w}}$ and with a minimal kernel representation $R(\frac{d}{dt}) = 0$, $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$. Let the degree of $R(s)$ be N. Define a two variable polynomial $\Pi(\zeta,\eta) := \frac{R(-\zeta)-R(\eta)}{\zeta+\eta}$ and suppose $\Pi(\zeta,\eta) = Y(\zeta)^T X(\eta)$ is a minimal factorization[11] where $X(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{w}}[s]$ is a minimal state map $x := X(\frac{d}{dt})w$ and $Y(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$. Following result gives us one of the representation of $\mathfrak{B}$ required in Algorithm 1.

Proposition A.1. *[20, Section 2.5] For a system $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\mathrm{cont}}$, ($\mathtt{w} = 2\mathtt{m}$), with a minimal kernel representation $R(\frac{d}{dt}) = 0$, $R(s) \in \mathbb{R}^{\mathtt{m}\times\mathtt{w}}[s]$, let $\Pi(\zeta,\eta) = Y(\zeta)^T X(\eta)$ is a minimal factorization of $\Pi(\zeta,\eta)$. Then a state space representation $E\dot{x} + Fx + Gw = 0$ of $\mathfrak{B}$ is given by:*

$$\begin{bmatrix} Y_0^T \\ \vdots \\ Y_{N-1}^T \\ 0 \end{bmatrix} \dot{x} + \begin{bmatrix} 0 \\ Y_0^T \\ \vdots \\ Y_{N-1}^T \end{bmatrix} x + \begin{bmatrix} -R_0 \\ R_1 \\ \vdots \\ (-1)^{N+1} R_N \end{bmatrix} w = 0 \qquad (13)$$

where $Y(s) = Y_0 + Y_1 s + \cdots + Y_{N-1} s^{N-1}$ and $R(s) = R_0 + R_1 s + \cdots + R_N s^N$. Also, premultiplying equation (13) *by the matrix*

[11] Let $Y(s) = \tilde{Y} \begin{bmatrix} I_{\mathtt{m}} \\ I_{\mathtt{m}} s \\ \vdots \end{bmatrix}$ and $X(s) = \tilde{X} \begin{bmatrix} I_{\mathtt{w}} \\ I_{\mathtt{w}} s \\ \vdots \end{bmatrix}$ where $\tilde{Y}$ and $\tilde{X}$ are the coefficient matrices of $Y(s)$ and $X(s)$. The factorization $\Pi(\zeta,\eta) =: Y(\zeta)^T X(\eta)$ is called a *minimal* factorization if $\tilde{Y}$ and $\tilde{X}$ are each of full row rank [20, Section 2.3].

$T := \begin{bmatrix} L & 0_{\mathtt{n}\times\mathtt{m}} \\ 0_{\mathtt{m}\times\mathtt{m}N} & I_{\mathtt{m}} \end{bmatrix}$, $T \in \mathbb{R}^{(\mathtt{n}+\mathtt{m})\times\mathtt{m}(N+1)}$, *where* $L \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}N}$ *is a left inverse of the matrix* $\begin{bmatrix} Y_0 & Y_1 & \cdots & Y_{N-1} \end{bmatrix}^T$, *one obtains an output-nulling representation:*

$$\begin{aligned} \dot{x} &= -L \begin{bmatrix} 0 \\ Y_0^T \\ \vdots \\ Y_{N-2}^T \end{bmatrix} x - L \begin{bmatrix} -R_0 \\ R_1 \\ \vdots \\ (-1)^N R_{N-1} \end{bmatrix} w \qquad (14) \\ 0 &= -Y_{N-1}^T x + (-1)^N R_N w. \end{aligned}$$

Now let $M(\frac{d}{dt})\ell = 0$, $M(s) \in \mathbb{R}^{\mathtt{w}\times\mathtt{m}}[s]$ be an observable image representation of $\mathfrak{B}$ and let the unique symmetric storage function matrix corresponding to the minimal state map $x := X_\ell(\frac{d}{dt})\ell$, $X_\ell(s) \in \mathbb{R}^{\mathtt{n}\times\mathtt{m}}[s]$ be $K \in \mathbb{R}^{\mathtt{n}\times\mathtt{n}}$. Following result provides us with the second first order representation required in Algorithm 1.

Proposition A.2. *[17] Consider a lossless behavior* $\mathfrak{B} \in \mathfrak{L}^{\mathtt{w}}_{\mathrm{cont}}$, *with an observable image representation* $M(\frac{d}{dt})\ell = 0$, $M(s) \in \mathbb{R}^{\mathtt{w}\times\mathtt{m}}[s]$ *and lossless with respect to the supply rate* Σ. *Suppose, the degree of the matrix polynomial* $M(s)$ *is equal to* N *and let* $X_\ell(s) = X_0^\ell + X_1^\ell s + X_2^\ell s^2 + \cdots + X_{N-1}^\ell s^{N-1}$ *and* $M(s) = M_0 + M_1 s + \cdots M_N s^N$. *Then matrices* $E, F \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{n}}$, $G \in \mathbb{R}^{(N+1)\mathtt{m}\times\mathtt{w}}$ *defined below form a state space representation* $E\dot{x} + Fx + Gw = 0$*:*

$$E := \begin{bmatrix} (X_0^\ell)^T \\ (X_1^\ell)^T \\ \vdots \\ (X_{N-1}^\ell)^T \\ 0 \end{bmatrix} K \ , \ F := \begin{bmatrix} 0 \\ (X_0^\ell)^T \\ (X_1^\ell)^T \\ \vdots \\ (X_{N-1}^\ell)^T \end{bmatrix} K \text{ and } G := -\begin{bmatrix} M_0^T \\ M_1^T \\ \vdots \\ M_N^T \end{bmatrix} \Sigma. \quad (15)$$

APPENDIX B

In this appendix we provide a proof of Lemma 3.3

Proof. **(Proof of Lemma 3.3)** First we note that $R(s)M(s) = 0$. This is due to $G(s) = P(s)^{-1}Q(s)$ and $G(s) = -G(-s)^T$.

Statement 1: Let the highest degree in the polynomial matrix $X_\ell(s) = X_{\mathtt{w}}(s)M(s)$ be $\tilde{N}$. Since, $X_\ell(s) = X_{\mathtt{w}}(s)M(s)$, using degree arguments about $X_{\mathtt{w}}(s)$ and $M(s)$, we conclude that $\tilde{N} \leqslant 2N-1$. Now we show that in fact $\tilde{N} = N-1$. Consider the polynomial matrix $\tilde{X}_\ell(s) = \tilde{X}_{\mathtt{w}}(s)M(s)$, where the matrix $\tilde{X}_{\mathtt{w}}(s) \in \mathbb{R}^{N\mathtt{m}\times\mathtt{w}}[s]$ is:

$$\tilde{X}_{\mathtt{w}}(s) = \begin{bmatrix} R_1 + R_2 s + \ldots + R_N s^{N-1} \\ R_2 + R_3 s + \ldots + R_N s^{N-2} \\ \vdots \\ R_{N-1} + R_N s \\ R_N \end{bmatrix}. \qquad (16)$$

The matrix $X_\ell(s)$ can be constructed by removing the zero rows from $\tilde{X}_\ell(s)$. We first concentrate on the multiplication of the first $\mathtt{m}$ rows of $\tilde{X}_\ell(s)$ and $M(s)$. Define

$$X_1(s) := \left[R_1 + R_2 s + \ldots + R_N s^{N-1}\right] M(s).$$

The coefficient matrices corresponding to degrees greater than $N-1$ of $X_1(s)$ are the same as the coefficient matrices corresponding to degrees greater than N for the matrix polynomial matrix $R(s)M(s)$. Next consider the following polynomial matrix multiplication:

$$X_2(s) := \left[R_2 + R_3 s + \ldots + R_N s^{N-2}\right] M(s).$$

Here again, the coefficient matrices corresponding to degrees greater than $N-1$ of $X_2(s)$ are the same as the coefficient matrices corresponding to degrees greater than $N-1$ for the matrix polynomial matrix $R(s)M(s)$. In a similar manner we show that for polynomial matrices $X_3(s) := \left[R_3 + R_4 s + \ldots + R_N s^{N-3}\right] M(s), \ldots, X_{N-1}(s) := \left[R_{N-1} + R_N s\right] M(s)$ and $X_N(s) := R_N M(s)$ coefficient matrices for degrees greater than $N-1$ are same as the coefficients corresponding to degree greater than N in the polynomial matrix $R(s)M(s)$. Since $R(s)M(s) = 0$, for the polynomial matrix $\tilde{X}_\ell(s)$ (and also for $X_\ell(s)$) all coefficient matrices of terms with degree greater than $N-1$ are zero. This proves Statement 1.

Statement 2: Referring to the construction of $Y(s)$ in Lemma 3.1, we conclude that the left nullspace of $\hat{Y}$ consists of certain rows of the identity matrix $I_{\mathtt{m}N}$. Also, from the construction of $Y(s)$, we conclude that the left nullspace of $\hat{Y}$ is also the left nullspace of $\tilde{X}_{\mathtt{w}}(s)$ (equation (16)) because the zeros rows of $\tilde{X}_{\mathtt{w}}(s)$ correspond to the rows of $\hat{Y}(s)$ (see Lemma 3.1) that are removed in order to construct $Y(s)$. Also note that since $\mathfrak{B}$ is lossless, the row degree structure of $R(s)$ and the column degree structure of $M(s)$ are the same. Expand $\tilde{X}_\ell(s) := \tilde{X}_{\mathtt{w}}(s)M(s)$ as $\tilde{X}_\ell(s) = \tilde{X}_0^\ell + \tilde{X}_1^\ell s + \cdots + \tilde{X}_{N-1}^\ell s^{N-1}$. $\hat{X}$ can be constructed by removing the zero columns of $\hat{X}_\ell := \begin{bmatrix} \tilde{X}_0^\ell & \tilde{X}_1^\ell & \ldots & \tilde{X}_{N-1}^\ell \end{bmatrix}^T$. Also, by rewriting certain elements in matrix $\hat{X}_\ell$ in terms of higher degree coefficients of the matrix $M(s)$, one can show that $\hat{X}_\ell$, $\hat{X}$ and $\hat{Y}$ all have the same left nullspace. This requires a lot of book keeping, hence due to the paucity of space, we prove Statement 2 only for a specific case: $R(s) \in \mathbb{R}^{3\times 6}[s]$ and $w = \begin{bmatrix} y \\ u \end{bmatrix}$ is a proper input/output partition. Further assume that the degree of row 1 of $R(s)$ has degree equal to 1, row 2 of $R(s)$ has degree equal to 2 and the third row has degree equal to 4. Consider the polynomial matrix $\tilde{X}_\ell(s) = \tilde{X}_{\mathtt{w}}(s)M(s)$, where the matrix $\tilde{X}_\ell(s) \in \mathbb{R}^{12\times 3}[s]$ and the $\tilde{X}_{\mathtt{w}}(s) \in \mathbb{R}^{12\times 6}[s]$ equals:

$$\tilde{X}_{\mathtt{w}}(s) = \begin{bmatrix} R_1 + R_2 s + R_3 s^2 + R_4 s^3 \\ R_2 + R_3 s + R_4 s^2 \\ R_3 + R_4 s \\ R_4 \end{bmatrix}.$$

The right nullspace basis of $\hat{Y}^T$ (constructed in accordance with the procedure described in Lemma 3.1) is $\begin{bmatrix} e_4 & e_7 & e_8 & e_{10} & e_{11} \end{bmatrix}$ where e_i is the i^{th} column of I_{12}. Now, consider the matrix $\tilde{X}_\ell(s)$ and let $\tilde{X}_\ell(s) = \tilde{X}_0^\ell + \tilde{X}_1^\ell s + \tilde{X}_2^\ell s^2 + \tilde{X}_3^\ell s^3$. The matrix $\hat{X}$ can be obtained by removing the zero columns from the matrix $\begin{bmatrix} \tilde{X}_0^\ell & \tilde{X}_1^\ell & \tilde{X}_2^\ell & \tilde{X}_3^\ell \end{bmatrix}^T$. We now show that the fourth, seventh, eighth, tenth and the eleventh row of the matrix $\tilde{L} := \begin{bmatrix} \tilde{X}_0^\ell & \tilde{X}_1^\ell & \tilde{X}_2^\ell & \tilde{X}_3^\ell \end{bmatrix}^T$ are zero.

The fourth, fifth and the sixth rows of $\tilde{L}$ are:

$$\begin{bmatrix} M_0^T R_2^T + M_1^T R_1^T & M_0^T R_3^T + M_1^T R_2^T & M_0^T R_4^T + M_1^T R_3^T & M_1^T R_4^T \end{bmatrix}$$

Note that for this example (and for the general case also), the structure of R_i and M_i^T are the same upto signs, $i = 0,1,2,3,4$ ($i = 0,1,\cdots,N$ for the general case). To show that the fourth row is zero, we rewrite the expressions for the fourth, fifth and the sixth rows of $\tilde{L}$ in terms of M_2,M_3 and M_4 as for these matrices, the first row is zero (as the first row of $R(s)$ has the degree equal to 1). Hence, $M_0^T R_2^T + M_1^T R_1^T = -M_2^T R_0^T$ (as $M_0^T R_2^T + M_1^T R_1^T + M_2^T R_0^T$

is a coefficient matrix of $R(s)M(s)$). Similarly, $M_0^T R_3^T + M_1^T R_3^T = -M_2^T R_2^T - M_3^T R_0^T$, $M_0^T R_4^T + M_1^T R_3^T = -M_2^T R_2^T - M_3^T R_1^T - M_4^T R_0^T$ and $M_1^T R_4^T = -M_2^T R_3^T - M_3^T R_2^T - M_4^T R_1^T$. Since, the fourth fifth and the sixth rows of $\tilde{L}$ can be expressed in terms of M_2^T, M_3^T and M_4^T, the fourth row becomes equal to zero.

In a similar manner, we show that the seventh and eighth row of $\tilde{L}$ are also zero. The seventh, eighth and the ninth row of $\tilde{L}$ are:

$$\begin{bmatrix} M_0^T R_3^T + M_1^T R_2^T + M_2^T R_1^T & M_0^T R_4^T + M_1^T R_3^T + M_2^T R_2^T & M_1^T R_4^T + M_2^T R_3^T & M_2^T R_4^T \end{bmatrix}$$

Now, rewriting the seventh, eighth and the ninth row of $\tilde{L}$ in terms of M_3 and M_4 as: $M_0^T R_3^T + M_1^T R_2^T + M_2^T R_1^T = -M_3^T R_3^T$, $M_0^T R_4^T + M_1^T R_3^T + M_2^T R_2^T = -M_3^T R_1^T - M_4^T R_0^T$, $M_1^T R_4^T + M_2^T R_3^T = -M_3^T R_2^T - M_4^T R_1^T$ and $M_2^T R_4^T = -M_3^T R_3^T - M_4^T R_1^T$. Since, the first two rows of M_3^T and M_4^T are zero, the seventh and eight row of $\tilde{L}$ are zero. Again, we can show that the tenth and eleventh row of $\tilde{L}$ are also zero by writing the tenth, eleventh and the twelfth row of $\tilde{L}$ in terms of M_4^T. Hence the fourth, seventh, eighth, tenth and the eleventh row of $\tilde{L}$ and hence $\hat{X}$ are zero. Hence the left nullspaces of $\hat{X}$ and $\hat{Y}$ are the same. This proves Statement 2 and thus completes the proof of Lemma 3.3. □

APPENDIX C

In this appendix we consider two systems, one with a 3×3 transfer matrix, and the other being SISO, and obtain the storage function matrix using Algorithms of this paper.

Example A.3. *This is an example for the **two variable polynomial matrix factorization based method** (Algorithm 2). For the system having the following transfer matrix:*

$$G = \begin{bmatrix} \frac{s}{100+s^2} & \frac{-4+s}{100+s^2} & \frac{-8+3s}{100+s^2} \\ \frac{4+s}{100+s^2} & \frac{2s}{100+s^2} & \frac{4+5s}{100+s^2} \\ \frac{8+3s}{100+s^2} & \frac{-4+5s}{100+s^2} & \frac{13s}{100+s^2} \end{bmatrix}$$

Co-prime factorization for the transfer function matrix[12] *of the form $G(s) = N(s)D(s)^{-1}$ is below.*

$$N(s) := \begin{bmatrix} 0 & 22.4923 + 7.2779s + 4.5685s^2 & 19.5237 + 5.7745s - 0.0829s^2 \\ 0 & -14.0438 + 19.9290s + 0.4655s^2 & 34.6521 + 0.5622s - 0.5810s^2 \\ 0 & -5.5953 + 47.1359s + 5.4996s^2 & 88.8279 + 6.8989s - 1.2448s^2 \end{bmatrix} \times 10^{-3}$$

and

$$D(s) := \begin{bmatrix} d_1(s) & d_2(s) & d_3(s) \end{bmatrix} \text{ with}$$

[12] An obvious but non-co-prime factorization of the transfer matrix G is $G(s) = N(s)D(s)^{-1} = D(s)^{-1}N(s)$ with $D(s) = (s^2+100)I_3$ and $N(s)$ consists of all the numerator terms of G. The non-co-primness is due to $\pm 10i$ being a root of $\det N(s) = 0$. In order to compute a co-prime factorization of G, we find a MPB for the right nullspace of $R(s) := \begin{bmatrix} D(s) & -N(s) \end{bmatrix}$. This fact causes the less-simple co-prime factorization given. Further, due to the column-reducedness of $\mathrm{col}(N,D)$, the first column of the obtained $D(s)$ being zero-degree causes the first column of the obtained $N(s)$ to be identically zero. Further, this is a situation where results of [20] are not directly applicable: the modification proposed in Section 3-A is needed.

$d_1(s)$, $d_2(s)$ and $d_3(s)$ all column vectors with polynomial entries defined as

$$d_1(s) := \begin{bmatrix} -0.4082483 \\ -0.8164966 \\ 0.4082483 \end{bmatrix}$$

$$d_2(s) := \begin{bmatrix} -105.1437 + 859.3938s - 1.0514s^2 + 8.5939s^3 \\ -70.4042 - 425.8174s - 0.704s^2 - 4.2582s^3 \\ -245.9521 + 7.7590s - 2.4595s^2 + 0.0776s^3 \end{bmatrix} \times 10^{-3}$$

$$d_3(s) := \begin{bmatrix} 884.6167 + 51.2074s + 8.8462s^2 + 0.5121s^3 \\ -451.4649 - 30.4451s - 4.5146s^2 - 0.3045s^3 \\ -18.3132 - 9.6827s - 0.1831s^2 - 0.0968s^3 \end{bmatrix} \times 10^{-3}.$$

Consider the state space representation for G:

$$A = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 156.13439 & -87.488013 & 100 & 0 & 1.5613439 & -0.8748801 \\ -2415.0119 & -259.24 & 0 & 100 & -24.150119 & -2.5924 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -190.4385 & -158.76031 & -507.95913 \\ 1568.6849 & 3315.3579 & 8199.4006 \end{bmatrix}$$

$$C = \begin{bmatrix} -22.4923 & -19.5237 & 7.2779 & 5.7745 & -4.5685 & 0.0829 \\ 14.0438 & -34.6521 & 19.929 & 0.5622 & -0.4655 & 0.5810 \\ 5.5953 & -88.8279 & 47.1359 & 6.8989 & -5.4996 & 1.2448 \end{bmatrix} \times 10^{-3}$$

and $D = 0$, with respect to which we obtain the storage function as:

$$K = \begin{bmatrix} 115.050 & -7.100 & 18.658 & -6.192 & 2.527 & 0.163 \\ -7.100 & 38.128 & -34.671 & -2.118 & 0.271 & -0.092 \\ 18.658 & -34.671 & 33.803 & 1.160 & 0.187 & 0.069 \\ -6.192 & -2.118 & 1.160 & 0.819 & -0.478 & -0.021 \\ 2.527 & 0.271 & 0.187 & -0.478 & 0.377 & 0.017 \\ 0.163 & -0.092 & 0.069 & -0.021 & 0.017 & 0.003 \end{bmatrix} \times 10^{-4}.$$

As mentioned in Footnote 12, this is an example where results of Section 3-A are relevant.

Example A.4. *This is an example for the **static relation extraction method** (Algorithms 3 and 4). For the system with the state space matrices:*

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.0011262 & 0 & -0.0878753 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 14.074 & 0 & 479.695 \end{bmatrix} \times 10^{-4} \text{ and } D = 0.$$

The storage function using LU factorization:

$$K_{LU} = \begin{bmatrix} 0.016 & 0 & 0.540 & 0 \\ 0 & 0.697 & 0 & 14.074 \\ 0.540 & 0 & 28.079 & 0 \\ 0 & 14.074 & 0 & 479.695 \end{bmatrix} \times 10^{-4}.$$

The storage function using QR:

$$K_{QR} = \begin{bmatrix} 0.016 & 0 & 0.540 & 0 \\ 0 & 0.697 & 0 & 14.074 \\ 0.540 & 0 & 28.079 & 0 \\ 0 & 14.074 & 0 & 479.695 \end{bmatrix} \times 10^{-4}$$

This example demonstrates the working of the algorithms based on LU/QR with partial pivoting and the errors (computed using the error metrics proposed in Section 6-C) are of the order shown in Figures 6 and 8.

REFERENCES

[1] B.D.O. Anderson, and S. Vongpanitlerd, *Network Analysis and Synthesis: A Modern Systems Theory Approach.*, Dover Publications, New York, 2006.

[2] A.C. Antoulas, *Approximation of Large-scale Dynamical Systems*, SIAM, Philadelphia, 2005.

[3] M.N. Belur, H.K. Pillai and H.L. Trentelman, Synthesis of dissipative systems: a linear algebraic approach, *Linear Algebra and its Applications*, vol. 425, no. 2-3, pages 739-756, 2007.

[4] P. Benner, V. Sima and M. Slowik, Evaluation of the linear matrix equation solvers in SLICOT, *Journal of Numerical Analysis, Industrial and Applied Mathematics*, vol. 2, no. 1-2, pages 11-34, 2007.

[5] C. Bhawal, D. Pal, S. Kumar and M.N. Belur, New results and techniques for computation of stored energy in lossless/all-pass systems, *IEEE Transactions on Circuits and Systems I*, vol. 64, no. 1, pages 72-85, 2017

[6] D.A. Bini, B. Iannazzo and B. Meini, *Numerical Solution of Algebraic Riccati Equations*, SIAM, Philadelphia, 2012.

[7] S.P. Boyd, L.E. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System & Control Theory*, SIAM, Philadelphia, vol. 15, 1994.

[8] G.H. Golub, and C.F. Van Loan, *Matrix Computations*, edition 3, John Hopkins University Press, 1996.

[9] T. Hughes, Controllability of passive single-input single-output systems, *IEEE European Control Conference*, Aalborg, Denmark, 2016, pages 1117-1122.

[10] M. Hershenson, A. Hajimiri , S.S. Mohan , S.P. Boyd and T.H. Lee, Design and optimization of LC oscillators, *In: Proceedings of the 2000 IEEE/ACM international conference on computer-aided design (ICCAD)*, San Jose, USA, November 7-11, pages 6569, 1999.

[11] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[12] A. Kothyari, C. Praagman, M. N. Belur and D. Pal, A subspace intersection based method for faster computation of the storage function for the lossless and all-pass cases, *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, Minnesota, USA, July 12-15, 2016.

[13] H.T. Kung and D.M. Tong, Fast algorithms for partial fraction decomposition, *SIAM Journal on Computing*, vol. 6, no. 2, pages 582-593, 1977.

[14] M. Künzer, A. Martin, M. Tentler and M. Wahrheit, Zassenhaus-Algorithms, `http://mo.mathematik.uni-stuttgart.de/inhalt/beispiel/beispiel1105`

[15] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford University Press, 1995.

[16] J.W. Polderman and J.C. Willems, *Introduction to Mathematical Systems Theory: a Behavioral Approach*, Springer-Verlag, 1998.

[17] P. Rapisarda and S. Rao, Realization of lossless systems via constant matrix factorizations, *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pages 2632-2636, 2013.

[18] A. Rofougaran, J. Rael, M. Rofougaran and A. Abidi, A 900 MHz CMOS LC-oscillator with quadrature outputs, *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, USA, November 10, pages 392-393, 1996.

[19] L. Romano, S. Levantino, C. Samori and A.L. Lacaita, Multiphase LC oscillators, *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 7, pages 1579-1588, 2006

[20] A.J. van der Schaft and P. Rapisarda, State maps from integration by parts, *SIAM Journal on Control and Optimization*, vol. 49, no. 6, pages 2415-2439, 2011.

[21] V. Simoncini, *Computational Methods for Linear Matrix Equations*, Technical Report, Università di Bologna, Italy, 2013.

[22] H.L. Trentelman, H.B. Minh and P. Rapisarda, Dissipativity preserving model reduction by retention of trajectories of minimal dissipation, *Mathematics of Control, Signals, and Systems*, vol. 21, no. 3, pages 171-201, 2009.

[23] H.L. Trentelman, P. Rapisarda, Pick matrix conditions for sign-definite solutions of the algebraic Riccati equation, *SIAM Journal on Control and Optimization*, vol. 40, no. 3, pages 969-991, 2001

[24] H.L. Trentelman and J.C. Willems, Synthesis of dissipative systems using quadratic differential forms: Part II, *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pages 70-86, 2002.

[25] M. Tucsnack and G. Weiss, How to get a conservative well-posed linear system out of thin air, Part I: Well-posedness and energy balance, *ESAIM Control, Optimisation and Calculus of Variations*, vol. 9, pages 247-273, 2003.

[26] J.C. Willems, Dissipative dynamical systems, Part II: Linear systems with quadratic supply rates, *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pages 352-393, 1972.

[27] J.C. Willems and H.L. Trentelman, On quadratic differential forms, *SIAM Journal on Control and Optimization*, vol. 36, no. 5, pages 1703-1749, 1998.

[28] J.L Wyatt, L.O. Chua, J.W. Gannett, I.C. Göknar and D.N. Green, Energy concepts in the state-space theory of nonlinear n-ports: part II- Losslessness, *IEEE Transactions on Circuits and Systems*, vol. 29, no. 7, pages 417-430, 1982.

[29] K.S. Yee, Explicit solutions for the initial boundary value problem of a system of lossless transmission lines, *SIAM Journal on Applied Mathematics*, vol. 24, no. 1, pages 62-80, 1973.

[30] H.D. Young, R.A. Freedman, T.R. Sandin and A.L. Ford, *Sears and Zemansky's University Physics*, Addison Wesley, 1999.

[31] J.C. Zúniga Anaya and D. Henrion, An improved Toeplitz algorithm for polynomial matrix null-space computation, *Applied Mathematics and Computation*, vol. 207, no. 1, pages 256–272, 2009.

Ashish Kothyari obtained his B.Tech. degree in Electrical Engineering from National Institute of Technology Kurukshetra, India in 2012. He is currently finishing Ph.D. in the Dept. of Electrical Engineering, Indian Institute of Technology Bombay. His research areas include singular control systems, numerical linear algebra and behavioral theory of control systems.

Kees Praagman received his Bachelor's degree in Mathematics in 1975, his Bachelors in Law in 1980, his Master's degree in Mathematics in 1981 and his PhD degree in Mathematics and Natural Sciences in 1985, all from the University of Groningen. The topic of his thesis was "Meromorphic Linear Difference Equations". In 1985 he moved to Eindhoven University of Technology, joining the group of M.L.J. Hautus in systems theory. Presently he is working in the Faculty of Economics and Business, University of Groningen. His current research is on the properties of matrix polynomials in behavioral linear systems.

Madhu N. Belur obtained the Ph.D. degree in 2003 from University of Groningen, Netherlands in Applied Mathematics. He currently is in the Department of Electrical Engineering, Indian Institute of Technology Bombay. His interests include dissipative dynamical systems, application of graph theory in control, implementation of various allocation problems, railway timetable analysis/construction and open-source implementation of various applications.