Hyperbolic QR factorization for J-spectral factorization of polynomial matrices

Didier Henrion[1,2,3,4] Peter Hippe[5]

July 12, 2004

Abstract

Motivated by a superfluous zero extraction problem arising in the discrete-time J-spectral factorization of polynomial matrices, we propose an algorithmic solution to hyperbolic QR factorization of a rank deficient constant matrix. Application to reduced-order H_∞ filtering is illustrated by a numerical example.

Keywords

Polynomial matrices, J-spectral factorization, H_∞ control, numerical linear algebra, computer-aided control system design.

1 Introduction

Polynomial methods offer an alternative to state-space methods when designing computer-aided control system design algorithms. Amongst other software products, the Fortran SLICOT library [N98] incorporates mostly state-space methods, whereas the Polynomial Toolbox [P98] can be viewed as an alternative, or complementary product including various routines to deal with algebraic entities such as polynomials and polynomial matrices, based on the theory pioneered in [K79].

Keeping with our endeavour of designing efficient and reliable numerical routines for polynomial matrices for the Polynomial Toolbox package, in this paper we describe the

[1]Corresponding author. FAX: +33 5 61 33 69 69. E-mail: henrion@laas.fr

[2]LAAS-CNRS, 7 Avenue du Colonel Roche, 31 077 Toulouse, France.

[3]Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Praha, Czech Republic.

[4]Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 16627 Praha, Czech Republic.

[5]Universität Erlangen-Nürnberg, Lehrstuhl für Regelungstechnik, Cauerstrasse 7, D-91058 Erlangen, Germany.

algorithmic solution of a hyperbolic QR factorization problem arising when pursuing a polynomial approach to discrete-time reduced-order H_∞ filtering [HD02]. The problem concerns extraction of superfluous zeros when performing J-spectral factorization of polynomial matrices [KS94]. Spectral factorization is an operation that consists in extracting a solution of a quadratic polynomial matrix equation, and that can be viewed as a polynomial alternative to the state-space algebraic Riccati equation [K96].

In section 2 we briefly recall the polynomial solution to the discrete-time reduced-order H_∞ filtering problem as described in [HD02]. We show that extraction of superfluous zeros is a necessary step when performing J-spectral factorization of polynomial matrices, and we describe formally in section 3 the corresponding hyperbolic QR factorization problem. An algorithmic solution is proposed in section 4, together with illustrative numerical examples. An application to the original H_∞ filtering problem is also described. Numerical stability issues are mentioned in the conclusion.

2 Reduced-order H_∞ filtering

Consider a time-invariant, discrete-time, linear system of order n with unmeasurable outputs y_z (size p_z), measurements y_m (size p_m) and a vector of disturbances w represented by

$$y(z) = \bar{G}(z)w(z) = [G(s) + E]w(z) \tag{1}$$

where $G(z)$ is the strictly proper part of $\bar{G}(z)$, with the partitioning

$$y = \begin{bmatrix} y_z \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} y_z \\ y_m \end{bmatrix}$$

and

$$E = \begin{bmatrix} 0 \\ E_1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ E_m \end{bmatrix}$$

and the left coprime factorization

$$G(z) = D^{-1}(z)N(z). \tag{2}$$

Output y_m is subdivided such that y_1 contains the $p_m - p$ disturbed measurements and y_2 the p perfect ones. We assume that matrix E_1 has full row rank and $\det D(z)$ does not have zeros at $z = 0$.

The problem considered in H_∞ filtering is the following. Given system (1) and its p_m measurements y_m, find an estimate $\hat{y}_z$ for the p_z unmeasurable linear combinations y_z of the state x in the H_∞-norm minimization sense, i.e. such that the (worst-case) performance measure

$$\sup_w \|y_z - \hat{y}_z\|_2 / \|w\|_2$$

over all non-zero real square summable functions w on the interval $[0, \infty[$ is less than a given upper bound γ.

A state-space (time-domain) solution to this problem was presented in [HD01], whereas a polynomial (frequency-domain) solution was described in [HD02]. In this paper we will focus on the polynomial solution.

For the filter design, it is assumed that the factorization (2) is such that the polynomial part of matrix

$$D^p(z) = D(z) \begin{bmatrix} I & 0 \\ 0 & z^{-1}I_p \end{bmatrix}$$

is row proper, i.e. its highest row degree coefficient matrix $G_r[D^p(z)]$ has full rank. This can always be ensured by appropriate left unimodular operations, see [W74].

The optimal filter has the transfer function

$$\hat{y}(z) = \begin{bmatrix} I_{pz} & 0 \end{bmatrix} \tilde{D}^{-1}(z) \begin{bmatrix} \tilde{D}(z) - D(z) \end{bmatrix} \begin{bmatrix} 0 \\ y_m(z) \end{bmatrix}$$

where $\tilde{D}(z)$ parametrizes the optimal H_∞ filter. As explained in [HD01], in order to obtain the optimal filter matrix $\tilde{D}(z)$ we need as an intermediate result the parametrizing polynomial matrix $\tilde{D}_f(z)$ for the fictive filter (assuming y_z to be measurable) described by

$$\hat{y}_f(z) = \begin{bmatrix} I_{pz} & 0 \end{bmatrix} \tilde{D}_f^{-1}(z) \begin{bmatrix} \tilde{D}_f(z) - D(z) \end{bmatrix} y(z)$$

with $\tilde{D}_f(z)$ and $D^p(z)$ sharing the same highest row degree coefficient matrices, i.e. $G_r[\tilde{D}_f(z)] = G_r[D^p(z)]$.

The polynomial matrix $\tilde{D}_f(z)$ is obtained via the following steps. First solve the J-spectral factorization equation

$$\begin{aligned} D_f(z)JD_f^T(z^{-1}) = D(z) \begin{bmatrix} -\gamma^2 I_{p_z} & 0 \\ 0 & E_m E_m^T \end{bmatrix} D^T(z^{-1}) + N(z)N^T(z^{-1}) + \\ N(z)E^T D^T(z^{-1}) + D(z)EN^T(z^{-1}) \end{aligned} \tag{3}$$

The polynomial matrix $D_f(z)$ resulting from factorization (3) may contain q superfluous zeros at $z = 0$ which must be extracted by column operations

$$D_r(z)V(z) = D_f(z) \tag{4}$$

such that

$$\det V(z) = z^q, \quad V(z)JV^T(z^{-1}) = J.$$

This can be done by taking the zero coefficient matrix

$$A = D_f^T(0)$$

which due to the zeros of $D_f(z)$ at $z = 0$ is rank deficient with $\operatorname{rank} A = n - q$, and finding a so-called hyperbolic QR factorization

$$A = QR \tag{5a}$$

$$Q^T JQ = J. \tag{5b}$$

Indeed, from factorization (5) matrix $D_f(0)Q^T = R^T$ has q zero columns, and polynomial matrix $D_f(z)Q^T$ has a common factor z in all non-zero elements of the corresponding columns that can be extracted by a matrix

$$E(z) = \text{diag} \left[\begin{array}{ccccccc} 1 & \cdots & z^{-1} & \cdots & 1 & \cdots & z^{-1} \end{array} \right]$$

giving

$$V^{-1}(z) = Q^T E(z).$$

With the above $D_r(z)$, the parametrizing polynomial matrix for the fictive filter finally is

$$\tilde{D}_f(z) = D_r(z) G_r^{-1}[D_r(z)] G_r[D^p(z)].$$

To sum up, a key step in obtaining the H_∞ filter is the extraction of superfluous zeros in polynomial matrix $D_f(z)$ via hyperbolic QR factorization (5).

3 Problem statement

The hyperbolic QR factorization problem arising in discrete-time J-spectral factorization of polynomial matrices can be formally stated as follows:

HQR Problem: Given an n-by-n square matrix A of rank m and a signature matrix J, compute a full rank matrix Q and a matrix R of rank m with $n-m$ zeros rows satisyfing relations (5).

Some remarks are in order.

Contrary to the classical QR factorization (corresponding to $J = I$ or $J = -I$), it is not assumed here that matrix R in equation (5a) has an upper-triangular structure. For our J-spectral factorization application, the key property is that R has just as many zero rows as the rank defect of A.

Matrix Q in (5b) is referred to as a J-orthogonal matrix. Properties and generation of J-orthogonal matrices have been surveyed in [KSH00, App. B] and [H02]. Besides the J-spectral factorization problem studied here, J-orthogonal matrices arise in fast array algorithms for estimation and filtering [KSH00], as well as in the numerical solution of various matrix problems including downdating of Cholesky factorization, symmetric definite generalized eigenvalue problems, study of J-contractive matrices, see [H02] and references therein.

4 Algorithm

In order to provide an algorithmic solution to problem HQR, we briefly recall properties and generation of orthogonal and hyperbolic rotation matrices.

4.1 Orthogonal rotations

An orthogonal rotation (also called plane, or Givens rotation) zeroes a single element of a vector or matrix. Suppose we are given a vector

$$a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

and we want to cancel entry a_2 with a row operation. Define the rotation matrix

$$Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1 \tag{6}$$

and notice that

$$Qa = \begin{bmatrix} \sqrt{a_1^2 + a_2^2} \\ 0 \end{bmatrix}$$

for the choice

$$c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, \quad s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}.$$

Due to the constraint $c^2 + s^2 = 1$, matrix Q in equation (6) is orthogonal, i.e.

$$Q^T Q = I.$$

All the entries but one can be cancelled in a vector or matrix column by successive applications of the above 2-by-2 orthogonal rotations. This is the rationale behind QR factorization, where lower-triangular entries in a matrix are cancelled columnwise, producing an upper-triangular matrix R and an orthogonal matrix Q satisfying equation (5a), see [GV96] for details.

4.2 Hyperbolic rotations

Just like orthogonal rotations, hyperbolic rotations are used to cancel an individual entry in a vector or matrix. A hyperbolic rotation has the form

$$Q = \begin{bmatrix} c & -s \\ -s & c \end{bmatrix}, \quad c^2 - s^2 = 1.$$

Its name origins from the fact that $c = \cosh\theta = \frac{1}{2}(e^\theta + e^{-\theta})$ and $s = \sinh\theta = \frac{1}{2}(e^\theta - e^{-\theta})$ for some θ, compare with the plane rotation matrix (6) for which $c = \cos\theta$ and $s = \sin\theta$.

The equation

$$Qa = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

has a real solution only when $|a_1| > |a_2|$, in which case

$$c = \frac{a_1}{\sqrt{a_1^2 - a_2^2}}, \quad s = \frac{a_2}{\sqrt{a_1^2 - a_2^2}}, \quad b = \sqrt{a_1^2 - a_2^2}.$$

It can be checked that matrix Q is J-orthogonal, i.e.

$$Q^T J Q = J$$

for signature matrices

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since a hyperbolic rotation can be applied only when $|a_1| > |a_2|$, it is necessary to carry out a row permutation prior to the transformation when $|a_1| < |a_2|$. As a result, unlike for orthogonal reductions, one cannot choose arbitrarily the locations of elements to be zeroed in a vector or matrix.

Finally, in the case $|a_1| = |a_2|$, no hyperbolic transformation can be applied. It means that, unlike for orthogonal reductions and the QR factorization, in general successive applications of hyperbolic rotations do not result in an upper-triangular matrix R.

4.3 Algorithm description

The algorithmic solution to problem HQR combines row and column orthogonal rotations, row hyperbolic rotations and row permutations. Basically, the algorithm mimics standard QR factorization: it consists in successive columnwise cancellations, until all linearly dependent rows appear at the bottom of the transformed matrix.

The main difference with standard QR factorization, which is also the main difficulty, concerns processing of the case when no row hyperbolic rotation can be applied. Below we will show that under the following assumption on input matrices A and J, a hyperbolic rotation can always be applied.

Assumption: Matrices A and J satisfy

$$\text{rank } A^T J A = \text{rank } A = m. \tag{7}$$

Obviously, the above rank relation necessarily holds since if algorithm HQR successfully returns matrices Q and R satisfying relations (5), then

$$A^T J A = R^T Q^T J Q R = R^T J R.$$

Denoting by U and V the matrices obtained by keeping only the m non-zero rows in matrices R and JR, respectively, relation (7) follows from Sylvester's rank inequality applied on the product $U^T V$.

Conversely, suppose that at some step k of the algorithm we obtain matrices R_k and Q_k such that

$$Q_k A = R_k, \quad Q_k^T J Q_k = J$$

and we are processing column c_1. Let a_1 in this column denote the leading entry in row r_1 in matrix R_k, used to cancel entry a_2 in row r_2 in the same column. Suppose that

these entries correspond to different signs, so that a hyperbolic rotation must be applied. Moreover, suppose that $|a_1| = |a_2|$. Then several cases must be distinguished:

- either some remaining entries in the column have not yet been cancelled, in which case the algorithm proceeds to cancel remaining entries (see example 5.2 to understand why), and then the same column is processed again;
- or all the remaining entries in the column have been cancelled, in which case
  - either rows r_1 and r_2 are linearly independent; then there exists a column c_2 with elements b_i in the same row as a_i for $i = 1, 2$, such that $a_1 b_2 \neq a_2 b_2$. We can then combine columns c_1 and c_2 with the help of a column orthogonal rotation U_k so that the resulting matrix $R_k U_k$ features leading entries a_1 and a_2 such that $|a_1| \neq |a_2|$. For a reason that should be clear with example 5.3, we choose a random 2×2 orthogonal rotation. Then the algorithm proceeds and the same column is processed again;
  - or $r_1 = \pm r_2$, so that rows r_1 and r_2 are linearly dependent. Then when evaluating the product
    $$A^T J A = R_k^T J R_k = S_k + r_1^T r_1 - r_2^T r_2 = S_k$$
    it follows that
    $$\text{rank } A^T J A = \text{rank } S_k = \text{rank } A - 1$$
    which contradicts our rank assumption (7). In other words, this situation never occurs.

The above analysis shows that the HQR problem can always be solved as soon as condition (7) is satisfied.

Based on these observations, a schematic description of Algorithm HQR is as follows.

Algorithm HQR: Proceed columnwise, denoting a_1 the pivot entry in row number i_1 and column number j_1 used to cancel iteratively bottom entries a_2 in rows number $i_2 > i_1$ and column number j_1. If entry a_2 is non-zero and

- $J_{i_1} = J_{i_2}$ then cancel a_2 with an orthogonal rotation;
- $J_{i_1} \neq J_{i_2}$ and $|a_1| \neq |a_2|$ then if
  - $|a_1| > |a_2|$ then cancel a_2 with a hyperbolic rotation;
  - $|a_2| > |a_1|$ then permute rows i_1 and i_2 and cancel a_1 with a hyperbolic rotation;
- $J_{i_1} \neq J_{i_2}$ and $|a_1| = |a_2|$ then
  - try first to cancel all remaining entries in column j_1;
  - if all remaining entries in column j_1 have already been cancelled, then find column index j_2 such that $a_1 b_2 \neq a_2 b_2$ where b_k is the entry in row i_k and column j_2. Combine columns j_1 and j_2 with an orthogonal rotation, and process again column j_1.

The above algorithm has been implemented in a documented, stand-alone Matlab m-file available at

`http://www.laas.fr/~henrion/software/hqr`

This function will be included to the next release 3.0 of the commercial Polynomial Toolbox for Matlab, see [P98].

5 Examples

5.1 No solution to problem HQR

With this basic example we want to show that matrices A and J may exist for which assumption (7) is not satisfied. Just take

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It holds rank $A = 1$ yet rank $A^T JA = 0$. As a result, problem HQR cannot be solved.

5.2 Cancelling all remaining entries first

This example shows that, when no hyperbolic rotation can be applied because $|a_1| = |a_2|$, then it is recommended first to process remaining entries in the column.

Let

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Check that rank $A^T JA = \text{rank}\, A = 2$, so assumption (7) is satisfied, and matrix A can be reduced to the form (5).

When processing the third column with leading entry $a_1 = A_{23} = 1$ with sign $J_2 = -1$, entry $a_2 = A_{33} = 1$ with opposite sign $J_3 = 1$ cannot be cancelled with a hyperbolic rotation. Yet next entry $a_2 = A_{43} = 1$ with sign $J_4 = J_2 = -1$ can be cancelled with an orthogonal rotation:

$$Q_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}, \quad R_1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_1 A = R_1.$$

Processing again the third column with the new leading entry $a_1 = A_{23} = \sqrt{2}$, entry $a_2 = A_{33} = 1$ can be cancelled now with a hyperbolic rotation since $|a_1| \neq |a_2|$:

$$Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2} & -1 & 0 \\ 0 & -1 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_2 Q_1 A = R_2.$$

Matrix R_2 is in the required form, and problem HQR is solved.

5.3 Column orthogonal rotations

In some special situations, all the remaining entries in a column have been cancelled, yet no hyperbolic rotation can be applied, and it may be necessary as a last resort to carry out a column orthogonal rotation. Such operations destroy the upper-triangular structure of matrix R, yet zero rows in R are left unchanged.

Let

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

It holds rank $A^T J A = \text{rank } A = 2$, so assumption (7) is satisfied, and matrix A can be reduced to the form (5).

Processing the first column with leading entry $a_1 = A_{11} = 1$, no hyperbolic rotation can be used to cancel entry $a_2 = A_{31} = 1$, yet remaining entry $A_{21} = 0$ is already zero. Since rows r_1 and r_3 are linearly independent, a column orthogonal rotation must be applied here. A natural choice would then be a permutation of columns 1 and 2:

$$U_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad A U_1 = R_1$$

resulting in a matrix R_1 with $A_{31} = 0$. However, the operation altered remaining entries in the column, and $a_2 = A_{21} = 1$ must now be cancelled with leading entry $a_1 = A_{11} = 1$. A hyperbolic rotation must be applied since $J_1 = -J_2$, yet $|a_1| = |a_2|$. Remaining entry A_{31} is zero, so that once more we must resort to a column orthogonal rotation. A natural choice would then be a permutation of column 1 and 2:

$$U_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A U_1 U_2 = R_2.$$

However note that $R_2 = A$, so that we entered a never-ending cycle.

For this reason, we propose the following way out: instead of applying a column orthogonal rotation to cancel entry a_2, we apply a random column orthogonal rotation, and then the

column is processed again. In the above example, instead of permuting columns 1 and 2, we combine them randomly, e.g.

$$U_1 = \begin{bmatrix} 0.9501 & 0.3119 & 0 \\ -0.3119 & 0.9501 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 0.6383 & 1.2620 & 1 \\ -0.3119 & 0.9501 & 1 \\ 0.9501 & 0.3119 & 0 \end{bmatrix}, \quad AU_1 = R_1.$$

Then we process again the first column with leading entry $a_1 = A_{11} = 0.6383$. Entry $A_{21} = -0.3119$ is first cancelled with a hyperbolic rotation:

$$Q_2 = \begin{bmatrix} 1.1461 & 0.5600 & 0 \\ 0.5600 & 1.1461 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0.5569 & 1.9784 & 1.7061 \\ 0 & 1.7957 & 1.7061 \\ 0.9501 & 0.3119 & 0 \end{bmatrix} \quad Q_2 A U_1 = R_2.$$

Proceeding this way, we eventually solve problem HQR with output matrices

$$Q = \begin{bmatrix} 1.6393 & -0.4051 & -1.2342 \\ -1 & 1 & 1 \\ -0.8291 & -0.4051 & 1.2342 \end{bmatrix}, \quad R = \begin{bmatrix} 0.4051 & 1.2342 & 1.2342 \\ 0 & 0 & 0 \\ 0.4051 & -1.2342 & -1.2342 \end{bmatrix}.$$

5.4 H_∞ filtering

Let system (1) be given by

$$\begin{aligned} y_z(z) &= \frac{1}{z^2 - z + 0.25} \begin{bmatrix} z + 3.25 & -3z + 1.5 & 0 \end{bmatrix} w(z) \\ y_m(z) &= \left(\frac{1}{z^2 - z + 0.25} \begin{bmatrix} 2z + 0.25 & -z + 0.5 & 0 \\ -z + 3 & -2z + 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \right) w(z) \end{aligned}$$

so that $p_z = 1$, $p_m = 2$ and $p = 1$. A left coprime factorization (2) of the strictly proper part

$$\frac{1}{z^2 - z + 0.25} \begin{bmatrix} z + 3.25 & -3z + 1.5 & 0 \\ 2z + 0.25 & -z + 0.5 & 0 \\ -z + 3 & -2z + 1 & 0 \end{bmatrix}$$

is given by

$$N(z) = \begin{bmatrix} 0.7071 & -2.1213 & 0 \\ -1.4142 & -2.8284 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$D(z) = \begin{bmatrix} 0.4714z - 0.4125 & 0.2357z - 1.0017 & 0.2357z + 0.5893 \\ -0.4714 & -0.9428 & 1.4142z + 0.4714 \\ 0.5774 & -0.5774 & -0.5774 \end{bmatrix}.$$

With $\gamma = 2.41523$ the right hand-side of polynomial matrix equation (3) reads

$$\begin{bmatrix} 0.8982z + 3.7703 + 0.8982z^{-1} & 1.0741z + 4.8102 & -1.7237z + 1.9675 \\ 4.8102 + 1.0741z^{-1} & 9.5926 & 2.1320 \\ 1.9675 - 1.7237z^{-1} & 2.1320 & -1.6111 \end{bmatrix}.$$

Applying J-spectral factorization with an updated version of macro `spf` (available upon request, and to be implemented in the next release of the Polynomial Toolbox for Matlab), the factorization result is

$$D_f(z) = \begin{bmatrix} -0.5832z + 0.1117 & -0.9082z - 0.6920 & -1.6746z - 0.1221 \\ -0.3093z + 0.1336 & 0.5696z - 0.8276 & -2.9454z - 0.1460 \\ 2.0619z - 0.2145 & -0.1714z + 1.3281 & -0.9155z + 0.2343 \end{bmatrix}, \quad J = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Polynomial matrix $D_f(z)$ has $n - p = 1$ zero at $z = 0.1429$ and $q = 2$ zeros at $z = 0$, which have to be extracted.

Applying algorithm HQR on the constant coefficient matrix $A = N_f^T(0)$ we obtain the extraction matrix

$$V(z) = \begin{bmatrix} -1.0129z & 0.1586z & 0.0280z \\ -0.1611 & 0.9975 & 0.1760 \\ 0 & 0.1738z & -0.9848z \end{bmatrix}$$

such that the reduced matrix satisfying relation (4) reads

$$D_r(z) = \begin{bmatrix} 0.7817 & -1.2946z - 0.6938 & 1.4913 \\ 0.3054 & -0.8297 & 2.9996 \\ -2.0357 & 1.3315 & 0.8718 \end{bmatrix}.$$

The above matrix has just one zero at $z = 0.1429$, which corresponds to the pole of the fictive filter. The fictive filter is then obtained from polynomial matrix $D_r(z)$ along the lines given in [HD02], and we obtain

$$\tilde{D}_f(z) = \begin{bmatrix} 0.4714z - 0.5135 & 0.2357z - 0.4125 & 0.8250 \\ -0.4714 & -0.9428 & 1.4142 \\ 0.5774 & -0.5774 & 0 \end{bmatrix}.$$

6 Conclusion

Hyperbolic QR factorization has been studied recently in reference [BHP03], from which most of the material found here was extracted. Note however than in [BHP03] the authors are concerned with solving an indefinite least squares problem for which matrix A is rectangular and full column-rank, under the assumption that matrix $A^T JA$ is positive definite. In the development here, because of our polynomial J-spectral factorization background, matrix A is assumed to be square (even though our results can be extended to rectangular matrices) but rank-deficient. Moreover, our algorithm works under the less stringent assumption that matrices $A^T JA$ and A share the same rank. Finally, we are basically interested in zeroing rows in matrix A, without specific requirements on its triangular structure.

In [BHP03] it is mentioned that obtaining useful error bounds for the application of a product of hyperbolic transformations to a vector is much more difficult that when the transformations are orthogonal. It is well-known [GV96] that orthogonal rotations are backward stable, property on which numerically reliable algorithms such as the standard

QR factorization heavily rely. This is in contrast with hyperbolic rotations, and it is presently unclear to numerical analysts whether the hyperbolic QR factorization algorithm is mixed backward-forward stable, or even backward stable. As recalled in [H02], eigenvalues and singular values of J-orthogonal matrices occur in reciprocal pairs and these matrices can be arbitrarily ill-conditioned. Hyperbolic rotations must thus be carried out in a careful way, as recommended in [KSH00, App. B] and [BHP03]. In order to minimize the number of operations, it is also possible to use Householder reflections, allowing to zero several entries at once in a vector. J-orthogonal Householder reflections are described in [KSH00, App. B].

Acknowledgment

This work was supported by the Grant Agency of the Czech Republic under Project No. 102/02/0709. We are grateful to Hans Grabmüller who largely contributed to this work.

References

[BHP03] A. Bojanczyk, N. J. Higham, H. Patel. Solving the indefinite least squares problem by hyperbolic QR factorization. SIAM Journal on Matrix Analysis and Applications, Vol. 24, No. 4, pp. 914–931, 2002.

[GV96] G. H. Golub, C. F. Van Loan. Matrix computations. 3rd edition. The Johns Hopkins University Press, New York, NY, 1996.

[H02] N. J. Higham. J-orthogonal matrices: properties and generation. SIAM Review, Vol. 45, No. 3, pp. 504–519, 2003.

[HD01] P. Hippe, J. Deutscher. Design of reduced order H_∞ filters for discrete-time systems. Proceedings of the IFAC World Congress on Automatic Control, paper T-We-M18-1, Barcelona, Spain, 2002.

[HD02] P. Hippe, J. Deutscher. Frequency domain design of reduced order H_∞ filters for discrete-time systems. Proceedings of the European Control Conference, Cambridge, UK, 2003.

[KSH00] T. Kailath, A. H. Sayed, B. Hassabi. Linear estimation. Prentice Hall, Upper Saddle River, NJ, 2000.

[K79] V. Kučera. Discrete Linear Control: The Polynomial Approach. John Wiley and Sons, Chichester, UK, 1979.

[KS94] H. Kwakernaak, M. Šebek. Polynomial J-spectral factorization. IEEE Transactions on Automatic Control, Vol. 39, No. 2, pp. 315–328, 1994.

[K96] H. Kwakernaak. Frequency domain solution of the standard H_∞ problem. In M. J. Grimble, V. Kučera (Editors). Polynomial methods for control system design. Springer Verlag, Berlin, Germany, 1996.

[N98] NICONET: The Numerics in Control Network. 1998-2002. See `www.win.tue.nl/niconet`

[P98] PolyX, Ltd. The Polynomial Toolbox for Matlab. Prague, Czech Republic. Version 2.5 released in 2000. See `www.polyx.cz`

[W74] W. A. Wolovich. Linear multivariable systems. Springer Verlag, New York, NY, 1974.