Modular shift of a polynomial matrix using Matlab

Zdeněk Hurák, Michael Šebek
Institute of Information Theory and Automation
Czech Academy of Science, Prague
e-mail: {hurak, msebek}@utia.cas.cz

Abstract

Efficient algorithm for modular shift of a polynomial matrix is proposed in this note. The algorithm avoids any iteration that is inherent in standard Euclidean algorithm for division of polynomial matrices. It assumes that the denominator polynomial matrix is row-reduced. If it is not, it can always be transformed into row-reduced form accepting some additional computational cost. Numerical experiments with an implementation of the proposed algorithm in Matlab are reported.

1 Introduction

A practical motivation for developing the algorithm for a modular shift of a polynomial matrix comes from the ℓ_1-optimal control, where the impulse response of an LTI system must often be truncated (for example, when computing the norm), which introduces an error. The first step towards assessing a significance of this error is to find some auxiliary plant with an impulse response identical to the truncated *tail* of the impulse response of the original system. Modular shift of a polynomial matrix is a tool to find such a description.

Definition 1 (Shift of a polynomial matrix). *Let $N(z)$ be a polynomial matrix in complex variable z. A shifted polynomial matrix $\hat{N}(z)$ is defined as $N(z)$ with the powers of the complex variable increased by one*

$$\hat{N}(z) = zN(z) \tag{1}$$

A system-theoretic interpretation of a shift of a polynomial matrix is a forward operator acting on a sequence of matrices. If a forward shift operator is applied k times, we a get a k-step shift.

Definition 2 (k-step shift of a polynomial matrix). *Let $N(z)$ be a polynomial matrix in complex variable z. A shifted polynomial matrix $\hat{N}(z)$ is defined as $N(z)$ with the powers of the complex variable increased by k*

$$\hat{N}(z) = z^k N(z) \tag{2}$$

Occasionally, it is necessary to consider a shift of a polynomial matrix modulo another polynomial matrix. Recall that with polynomial matrices it is necessary to distinguish between left and right division. In this paper, left division of polynomial matrices is considered.

Definition 3 (Left modular shift of a polynomial matrix). *Let $\hat{N}(z) = zN(z)$ be a shifted polynomial matrix and $D(z)$ be another polynomial matrix such that the left polynomial division $D(z)^{-1}\hat{N}(z)$ is well-defined. Shifted polynomial matrix $\bar{N}(z)$ modulo $D(z)$ is a remainder in the left division of the shifted polynomial matrix $\hat{N}(z)$ and $D(z)$*

$$\begin{aligned} \bar{N}(z) &= \hat{N}(z) \mod D(z) && (3) \\ &= zN(z) - D(z)Q(z) && (4) \end{aligned}$$

where $Q(z)$ is a quotient determined uniquely by the polynomial matrices $\hat{N}(z)$ and $D(z)$.

Similarly, a k-step left modular shift of a polynomial matrix can be defined.

Following the previous definition, the first idea about how to compute the modular shift is to use the standard Euclidean algorithm for division of polynomial matrices. This however, turns out computationally ineffective as the standard Euclidean algorithm does not take advantage of the special structure of the problem.

A computationally more efficient algorithm can be devised extending a standard result from (scalar) polynomial modular arithmetics into the polynomial matrix framework. It is shown in the next section that the concepts of row and column reducedness of polynomial matrices play the major role.

2 Algorithm for modular shift of a polynomial matrix

Consider two polynomial matrices $N(z) \in \mathbb{R}[z]^{n\times m}$ and $D(z) \in \mathbb{R}[z]^{n\times n}$ such that the left polynomial matrix division $D(z)^{-1}N(z)$ defines a matrix of strictly proper rational functions. In other words, the polynomial matrix $N(z) = N(s) \mod D(z)$. Assume that $D(z)$ is row-reduced. If it is not, it can be always reduced [?] at some additional computational cost.

Now, write the polynomial matrix $D(z)$ as follows

$$D(z) = \begin{bmatrix} z^{k_1} & 0 & \dots & 0 \\ 0 & z^{k_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n} \end{bmatrix} D_q + \begin{bmatrix} z^{k_1-1} & 0 & \dots & 0 \\ 0 & z^{k_2-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n-1} \end{bmatrix} D_{q-1} + \dots + \begin{bmatrix} * & 0 & \dots & 0 \\ 0 & * & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & * \end{bmatrix} D_0 \quad (5)$$

where $k_i,\ i = 1, \dots, n$ are the row degrees of $D(z)$, $q = \max_{1\leq i\leq n} k_i$ and D_q is the leading row coefficient matrix and the star denotes an element that is either 1 or 0. Under the assumption of row-reducedness of $D(z)$, the constant matrix D_q is nonsingular, i.e., $detD_q \neq 0$.

Perform the same decomposition of the polynomial matrix $N(s)$

$$N(z) = \begin{bmatrix} z^{k_1-1} & 0 & \dots & 0 \\ 0 & z^{k_2-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n-1} \end{bmatrix} N_{q-1} + \begin{bmatrix} z^{k_1-2} & 0 & \dots & 0 \\ 0 & z^{k_2-2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n-2} \end{bmatrix} N_{q-2} + \dots + \begin{bmatrix} * & 0 & \dots & 0 \\ 0 & * & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & * \end{bmatrix} N_0 \quad (6)$$

The major result of this paper now follows the arguments used for scalar polynomials in [?], pp.337, with the role of the leading coefficient of the scalar polynomial taken over by the leading row coefficient matrix that is guaranteed to be nonsingular.

Lemma 1 (k-step left modular shift of a polynomial matrix). *Consider two polynomial matrices $N(z) \in \mathbb{R}[z]^{n\times m}$ and $D(z) \in \mathbb{R}[z]^{n\times n}$ such that the left polynomial matrix division $D(z)^{-1}N(z)$ defines a matrix of strictly proper rational functions, $D(z)$ being row-reduced. Using the* row degree *decomposition of the polynomial matrices as in (5) and (6), the relationship between the constant matrices of $N(z)$ and $\bar{N}(z) = z^k N(z) \mod D(z)$ is given*

$$\begin{bmatrix} \bar{N}_0 \\ \bar{N}_1 \\ \vdots \\ \bar{N}_{q-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & -D_0 D_q^{-1} \\ I & 0 & 0 & \dots & 0 & -D_1 D_q^{-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I & -D_{q_1-1} D_q^{-1} \end{bmatrix}^k \begin{bmatrix} N_0 \\ N_1 \\ \vdots \\ N_{q-1} \end{bmatrix} \quad (7)$$

Proof. First, prove the lemma for one-step shift. Combining the definition (2) of a shifted polynomial matrix $\hat{N}(s)$ and the equation (6)

$$zN(s) = \begin{bmatrix} z^{k_1} & 0 & \dots & 0 \\ 0 & z^{k_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n} \end{bmatrix} N_{q-1} + \begin{bmatrix} z^{k_1-1} & 0 & \dots & 0 \\ 0 & z^{k_2-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & z^{k_n-1} \end{bmatrix} N_{q-2} + \dots + \begin{bmatrix} *z & 0 & \dots & 0 \\ 0 & *z & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & *z \end{bmatrix} N_0 \quad (8)$$

Table 1: Some computation times [s] and relative errors for SHIFTMOD and LDIV algorithms.

Size, degree and step	SHIFTMOD	LDIV	Relative error
m=n=20, d=5, k=10	0.0100	0.2100	2.8681e-014
m=n=20, d=10, k=10	0.0200	0.2100	3.8701e-015
m=n=20, d=20, k=10	0.0100	0.2300	2.2448e-014
m=n=20, d=30, k=10	0.0200	0.2800	7.7952e-016
m=n=20, d=10, k=10	0.0200	0.2100	1.7120e-014
m=n=20, d=10, k=50	0.0100	12.8580	2.1058e-015
m=n=20, d=10, k=100	0.0200	92.5929	1.6125e-014

Substitute for the diagonal matrix in (8) from (5). The remainder $\bar{N}(z) = \hat{N}(z) \mod D(z)$ in the left polynomial matrix division $D^{-1}(z)\hat{N}(z)$ is then

$$\bar{N}(z) = N_0 z + N_1 z^2 + \ldots + N_{k_1-2} z^{q-1} - (D_0 + D_1 z + D_2 z^2 + \ldots + D_{q-1} z^{q-1}) D_q^{-1} N_{q-1} \qquad (9)$$

Comparing terms with equal powers, the lemma (for k=1) follows. The extension to $k > 1$ steps can be devised easily because the resulting constant matrix $\bar{N}$ can be placed at the position of matrix N in (7) to obtain a two-step modular shift. By induction, the lemma follows. □

3 Numerical experiments

Platform: PC, Intel Pentium 4, CPU 1300MHz, 568 kB RAM, Microsoft Windows 2000, Matlab 6.1, Polynomial Toolbox 3.0.

Accuracy and computational speed were compared between the proposed algorithm SHIFTMOD and the standard Euclidean division algorithm LDIV implemented in the Polynomial Toolbox for Matlab. Dependence of computation times for LDIV and SHIFTMOD algorithms on the size and degree of the polynomial matrix is visualized in Figure 1 for $k = 10$. Dependence of computation times on the size of the polynomial matrix and the number of steps of the shift is plotted in Figure 2 for fixed degree $d = 10$.

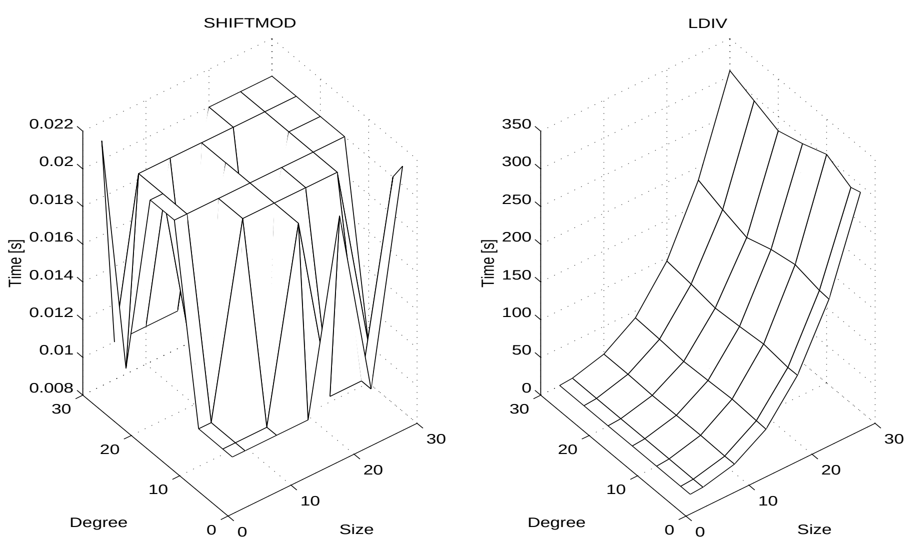

Figure 1: Computation times for SHIFTMOD and LDIV for fixed number of steps k=10.

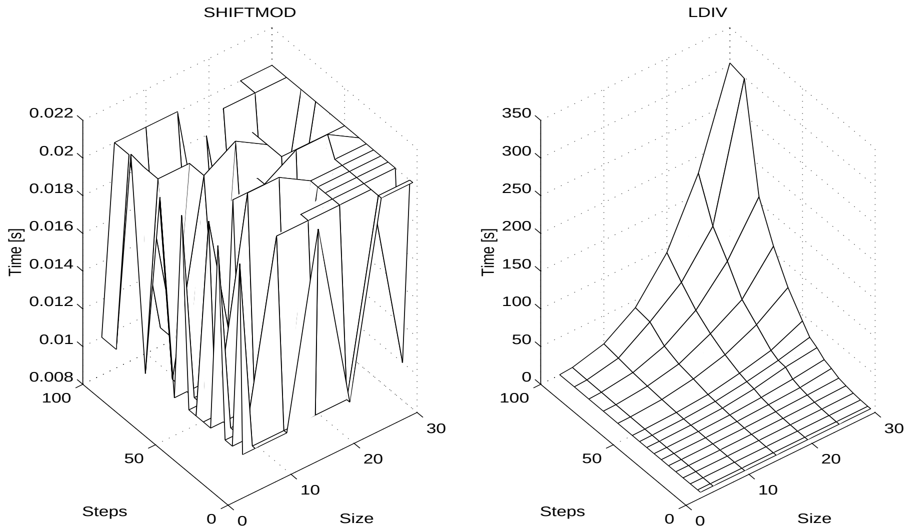

Figure 2: Computation times for SHIFTMOD and LDIV for fixed degree d=10.

4 Conclusions

In this note, an efficient algorithm for modular shift of a polynomial matrix was described.

Numerical experiments confirm that the proposed algorithm is a lot faster then standard Euclidean algorithm for division of polynomial matrices. The reason is that the proposed algorithm takes advantage of the special structure of the problem. The numerical accuracy is preserved.

The difference in computation times grows with the size of the polynomial matrix and the number of steps by which the matrix must be shifted. For a polynomial matrix of size 20×20 of degree 10, computation of the 100-step modular shift using the proposed algorithm SHIFTMOD is nearly one hundred times faster.

From practical point of view, the computation time of SHIFTMOD algorithm is not affected by the size, degree of the number of steps of shift. The standard Euclidean division algorithm LDIV performs poorly for higher sizes of polynomial matrices and larger number of steps of shift.

The algorithm might be useful for fast and reliable computation of a polynomial matrix fraction description of a system with the impulse response coinciding with a truncated *tail* in FIR approximation of LTI systems.

Acknowledgments

The work was supported by the Ministry of Education of the Czech Republic under Project GAČR 102/02/0709.