# CONTROLLER DESIGN USING POLYNOMIAL MATRIX DESCRIPTION

**Didier Henrion**, Centre National de la Recherche Scientifique, France
**Michael Šebek**, Center for Applied Cybernetics, Faculty of Electrical Eng., Czech Technical University in Prague, Czech Republic

**Keywords:** Polynomial methods, polynomial matrices, dynamics assignment, deadbeat regulation, $H_2$ optimal control, polynomial Diophantine equation, polynomial spectral factorization, Polynomial Toolbox for Matlab

## Contents



## Glossary

**Deadbeat regulation:** A control problem that consists in driving the state of the system to the origin in the shortest possible time.
**Diophantine equation:** A linear polynomial matrix equation.
**Dynamics assignment:** A control problem that consists in assigning the closed-loop similarity invariants.
$H_2$ **optimal control:** A control problem that consists in stabilizing a system while minimizing the $H_2$ norm of some transfer function.
**Interpolation:** An algorithm consisting in working with values taken by polynomials at given points, rather than with polynomial coefficients.
**Matrix fraction description:** Ratio of two polynomial matrices describing a matrix transfer function.
**Polynomial matrix:** A matrix with polynomial entries, or equivalently a polynomial with matrix coefficients.
**Polynomial Toolbox:** The Matlab Toolbox for polynomials, polynomial matrices and their application in systems, signals and control.
**Similarity invariants:** A set of polynomials describing the dynamics of a system.
**Spectral factor:** The solution of a quadratic polynomial matrix equation.
**Spectral factorization:** A quadratic polynomial matrix equation.
**Sylvester matrix:** A structured matrix arising when solving polynomial Diophantine equations.

**Summary**

Polynomial matrix techniques can be used as an alternative to state-space techniques when designing controllers for linear systems. In this article, we show how polynomial techniques can be invoked to solve three classical control problems: dynamics assignment, deadbeat control and $H_2$ optimal control. We assume that the control problems are formulated in the state-space setting, and we show how to solve them in the polynomial setting, thus illustrating the linkage existing between the two approaches. Finally, we mention the numerical methods available to solve problems involving polynomial matrices.

## 1. Introduction

Polynomial matrices arise naturally when modeling physical systems. For example, many dynamical systems in mechanics, acoustics or linear stability of flows in fluid dynamics can be represented by a second order vector differential equation

$$A_0x(t)+A_1\dot{x}(t)+A_2\ddot{x}(t)=0.$$

Upon application of the Laplace transform, studying the above equation amounts to studying the characteristic matrix polynomial

$$A(s)=A_0+sA_1+s^2A_2.$$

The constant coefficient matrices $A_0$, $A_1$ and $A_2$ are known as the stiffness, damping and inertia matrices, respectively, usually having some special structure depending on the type of loads acting on the system. For example, when $A_0$ is symmetric negative definite, $A_1$ is anti-symmetric and $A_2$ is symmetric positive definite, then the equation models the so-called gyroscopic systems. In the same way, third degree polynomial matrices arise in aero-acoustics. In fluid mechanics the study of the spatial stability of the Orr-Sommerfeld equation yields a quartic matrix polynomial.

It is therefore not surprising to learn that most of the control design problems boil down to solving mathematical equations involving polynomial matrices. For historical reasons, prior to the sixties most of the control problems were formulated for scalar plants, and they involved manipulations on scalar polynomials and scalar rational functions. The extension of these methods to the multivariable (multi-input, multi-output) case was not obvious at this time, and it has been achieved only with the newly developed concept of state-space setting. In the seventies, several multivariable results were therefore not available in the polynomial setting, which somehow renewed the interest in this approach. Now most of the results are available both in the state-space and polynomial settings.

In this article we will study in detail three standard control problems and their solution by means of polynomial methods. The first problem is known as the *dynamics assignment* problem, a generalization of eigenvalue placement. The second problem is called the *deadbeat regulation* problem. It consists of finding a control law such that the state of a discrete-time system is stirred to the origin as fast as possible, i.e. in a minimal number of steps. The third and last problem is $H_2$ *optimal control*, where a stabilizing control law is sought that minimizes the $H_2$ norm of some transfer function. All these problems are formulated in the state-space setting, and then solved with the help of polynomial methods, to better illustrate the linkage between the two approaches. After describing these problems, we will then focus more on practical aspects, enumerating the numerical methods that are available to solve the polynomial equations.

## 2. Polynomial approach to three classical control problems

### 2.1. Dynamics assignment

We consider a discrete-time linear system

$$x_{k+1} = Ax_k + Bu_k$$

with $n$ states and $m$ inputs, and we study the effects of the linear static state feedback

$$u_k = -Kx_k$$

on the system dynamics. One of the simplest problem we can think of is that of enforcing closed-loop eigenvalues, i.e. finding a matrix $K$ such that the closed-loop system matrix $A - BK$ has prescribed eigenvalues. In the polynomial setting, assigning the closed-loop eigenvalues amounts to assigning the characteristic polynomial $\det(zI - A + BK)$. This is possible if and only if $(A, B)$ is a reachable pair. (See **"Pole placement control"** for more information on pole assignment, and **"System characteristics: stability, controllability, observability"** for more information on reachability).

A more involved problem is that of enforcing not only eigenvalues, but also the eigenstructure of the closed-loop system matrix. In the polynomial setting, the eigenstructure is captured by the so-called *similarity invariants* of matrix $A - BK$, i.e. the polynomials that appear in the Smith diagonal form of the polynomial matrix $z - A + BK$. Rosenbrock's fundamental theorem captures the degrees of freedom one has in enforcing these invariants.

Let $(A, B)$ be a reachable pair with reachability indices $k_1 \geq \cdots \geq k_m$. Let $c_1(z), \ldots, c_p(z)$ be monic polynomials such that $c_{i+1}(z)$ divides $c_i(z)$ and $\sum_{i=1}^{p} \deg c_i(z) = n$. Then there exists a feedback matrix $K$ such that closed-loop matrix $A - BK$ has similarity invariants $c_i(z)$ if and only if

$$\sum_{i=1}^{k} \deg c_i(z) \geq \sum_{i=1}^{k} k_i, \quad k = 1, \ldots, p.$$

The above theorem basically says that one can place the eigenvalues at arbitrary specified locations but the structure of each multiple eigenvalue is limited: one cannot split it into as many repeated eigenvalues as one might wish. Rosenbrock's result is constructive, and we now describe a procedure to assign a set of invariant polynomials by static state feedback.

First we must find relatively right coprime polynomial matrices $D_R(z)$ and $N_R(z)$ with $D_R(z)$ column-reduced and column-degree ordered, such that

$$(zI-A)^{-1}B = N_R(z)D_R^{-1}(z). \tag{1}$$

See **“Polynomial and matrix fraction description”** for the definition of a column-reduced matrix. A column-reduced matrix can be put into column-degree ordered form by suitable column permutations.

Then we must form a column-reduced polynomial matrix $C(z)$ with invariant polynomials $c_1(z),\dots,c_p(z)$ which has the same column degrees as $D_R(z)$.

Finally, we solve the equation

$$X_L D_R(z) + Y_L N_R(z) = C(z) \tag{2}$$

for constant matrices $X_L$ and $Y_L$, and let

$$K = X_L^{-1}Y_L.$$

The above equation over polynomial matrices is called a *Diophantine polynomial equation*. Under the assumptions of Rosenbrock’s theorem, there always exists a constant solution $X$ and $Y$ to this equation.

As an example, take

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

as a reachable pair. We seek a feedback matrix $K$ such that $A-BK$ has similarity invariants

$$c_1(z) = z^3 - z^2, \quad c_2(z) = z.$$

Following a procedure of conversion from state-space form to matrix fraction description (MFD) form (see **Polynomial and matrix fraction description**) we obtain

$$D_R(z) = \begin{bmatrix} z^2 & z \\ 0 & z^2-z-1 \end{bmatrix}, \quad N_R(z) = \begin{bmatrix} 0 & z \\ 1 & 1 \\ z & z \\ 0 & 1 \end{bmatrix}$$

as a right coprime pair satisfying relation (1). The reachability indices of $(A,B)$ are the column degrees of column-reduced matrix $D_R(z)$, namely $k_1 = 2$ and $k_2 = 2$. Rosenbrock's inequalities are satisfied since

$$\deg c_1(z) \geq 2, \quad \deg c_1(z) + \deg c_2(z) \geq 4.$$

A column-reduced matrix with column degrees $k_1$, $k_2$ and $c_1(z)$, $c_2(z)$ as similarity invariants is found to be

$$C(z) = \begin{bmatrix} z^2 & 0 \\ z & z^2 - z \end{bmatrix}.$$

Then we have to solve the Diophantine equation

$$X_L \begin{bmatrix} z^2 & z \\ 0 & z^2 - z - 1 \end{bmatrix} + Y_L \begin{bmatrix} 0 & z \\ 1 & 1 \\ z & z \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} z^2 & 0 \\ z & z^2 - z \end{bmatrix}.$$

We find the constant solution

$$X_L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Y_L = \begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 \end{bmatrix}$$

corresponding to the feedback matrix

$$K = \begin{bmatrix} -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 \end{bmatrix}$$

yielding the desired dynamics.

### 2.2. Deadbeat regulation

We consider a discrete-time linear system

$$x_{k+1} = Ax_k + Bu_k$$

with $n$ states and $m$ inputs, and our objective is to determine a linear state feedback

$$u_k = -Kx_k$$

such that every initial state $x_0$ is driven to the origin in the shortest possible time. This is the *deadbeat regulation* problem.

Deadbeat regulation is a typical discrete-time control problem. Indeed, in the continuous-time setting, it is not possible to find a linear state feedback that stirs the system state in finite time. The idea of driving the initial state to the origin is closed related to discrete-time controllability.

To solve the above problem formulated in the state-space setting, we will use polynomial techniques. We will describe a procedure in the same spirit than in the previous paragraph. The fundamental constructive result on deadbeat control can be formulated as follows.

There exists a state-feedback deadbeat regulator if and only if the pair $(A, B)$ is controllable (see **"System characteristics: stability, controllability, observability"** for the definition of controllability. Under this assumption, let $X_R(z^{-1})$, $Y_R(z^{-1})$ be a minimum-column-degree polynomial solution of the Diophantine equation

$$(I - Az^{-1})X_R(z^{-1}) + Bz^{-1}Y_R(z) = I. \tag{3}$$

Then $X_R(z^{-1})$ is non-singular and

$$K = Y_R(z^{-1})X_R^{-1}(z^{-1})$$

is a deadbeat gain.

The above equation is a matrix polynomial Diophantine equation in the indeterminate $z^{-1}$. Note that in the previous paragraph, we have shown that solving the dynamics assignment problem amounts to solving a matrix polynomial Diophantine equation in the indeterminate $z$, where the right hand-side matrix contained all the required similarity invariants. It turns out that the deadbeat control problem can be viewed as a special case of dynamics assignment.

To see this, build a right coprime MFD $\bar{N}_R(z)$, $\bar{D}_R(z)$ of the transfer function

$$(zI - A)^{-1}B = \bar{N}_R(z)\bar{D}_R^{-1}(z)$$

where polynomial matrix $\bar{D}_R(z)$ is column-reduced with ordered column degrees $k_1 \geq \cdots \geq k_m$ (see **"Polynomial and matrix fraction description"**).

Then define

$$C^{-1}(z) = \text{diag} \begin{bmatrix} z^{-k_1} & \cdots & z^{-k_m} \end{bmatrix}$$

and

$$N_R(z^{-1}) = \bar{N}_R(z^{-1})C^{-1}(z), \quad D_R(z^{-1}) = \bar{D}_R(z^{-1})C^{-1}(z)$$

as right coprime polynomial matrices in the indeterminate $z^{-1}$. They satisfy the equation

$$(I-Az^{-1})^{-1}Bz^{-1} = N_R(z^{-1})D_R^{-1}(z^{-1})$$

and can be found in the generalized Bézout identity

$$\begin{bmatrix} I-Az^{-1} & -Bz^{-1} \\ Y_L & X_L \end{bmatrix} \begin{bmatrix} X_R(z^{-1}) & N_R(z^{-1}) \\ -Y_R(z^{-1}) & D_R(z^{-1}) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

where constant matrices $X_L$ and $Y_L$ are such that

$$X_L D_R(z^{-1}) + Y_L N_R(z^{-1}) = I.$$

Upon multiplication on the right by the matrix diag $[\, z^{k_1} \quad z^{k_m} \,]$ this later equation is precisely the dynamics assignment Diophantine equation (2) with right hand-side matrix

$$C(z) = \text{diag} \left[ \, z^{k_1} \quad \cdots \quad z^{k_m} \, \right].$$

It means that the deadbeat regulator assigns all the closed-loop eigenvalues at the origin.

As an example, take

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

as a controllable pair for which we seek a deadbeat gain matrix $K$. Diophantine equation (3) reads

$$\begin{bmatrix} 1 & -z^{-1} & 0 & 0 \\ 0 & 1 & -z^{-1} & 0 \\ 0 & 0 & 1 & -z^{-1} \\ 0 & 0 & -z^{-1} & 1 \end{bmatrix} X_R(z^{-1}) + \begin{bmatrix} z^{-1} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & z^{-1} \end{bmatrix} Y_R(z^{-1}) = I.$$

It has a minimum-column-degree solution pair

$$X_R(z^{-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & z^{-1} & z^{-2} \\ 0 & 0 & 1 & z^{-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Y_R(z^{-1}) = \begin{bmatrix} 0 & 1 & z^{-1} & z^{-2} \\ 0 & 0 & 1 & z^{-1} \end{bmatrix}$$

corresponding to the deadbeat feedback gain

$$K = Y_R(z^{-1})X_R^{-1}(z^{-1}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Closed-loop dynamics are given by

$$x(z^{-1}) = \sum_k z^{-k}(A-BK)^k x_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & z^{-1} & z^{-2} \\ 0 & 0 & 1 & z^{-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} x_0$$

so every initial condition $x_0$ is stirred to the origin in three steps at most since $(A-BK)^k = 0$ for $k \geq 3$.

## 2.3. $H_2$ optimal control

The $H_2$ optimal control problem consists of stabilizing a linear system in such a way that its transfer matrix attains a minimum norm in the Hardy space $H_2$. The linear quadratic (LQ) optimal control problem (see **"Optimal linear quadratic control"**) is a special case of a $H_2$ optimal control problem.

The continuous-time plant is modeled by the equations

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 v(t) + B_2 u(t) \\ z(t) &= C_1 x(t) + D_{11} v(t) + D_{12} u(t) \\ y(t) &= C_2 x(t) + D_{21} v(t) + D_{22} u(t) \end{aligned}$$

giving rise to the plant transfer matrix

$$\begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}.$$

Denoting $K(s)$ the transfer matrix of the controller, the control system transfer matrix between $v(t)$ and $z(t)$ becomes

$$G(s) = P_{11}(s) + P_{12}(s)[I - K(s)P_{22}(s)]^{-1}K(s)P_{21}(s).$$

The standard $H_2$ optimal control problem consists in finding a controller $K(s)$ that stabilizes the control system and minimizes the squared $H_2$ norm of $G(s)$, defined as

$$\int \operatorname{trace} G'(-s)G(s)ds$$

where the integral is a contour integral up the imaginary axis and then around an infinite semi-circle in the left half-plane.

We make the following assumptions on the plant:

- $(A, B_1)$ is stabilizable, $(C_1, A)$ is detectable, and $D_{11} = 0$
- $(A, B_2)$ is controllable, $(C_2, A)$ is observable, and $D_{22} = 0$
- $D'_{12}C_1 = 0, \quad D'_{12}D_{12} = I$
- $B_1D'_{21} = 0, \quad D_{21}D'_{21} = I.$

See **"System characteristics: stability, controllability, observability"** for more information on these concepts. The above assumptions greatly simplify the exposition of the results. They can be relaxed, but at the price of additional technicalities.

Then write

$$C_2(sI-A)^{-1} = D_L^{-1}(s)N_L(s)$$

where $N_L(s)$ and $D_L(s)$ are left coprime polynomial matrices such that $D_L(s)$ is row-reduced. Further, write

$$(sI-A)^{-1}B_2 = N_R(s)D_R^{-1}(s)$$

where $N_R(s)$ and $D_R(s)$ are right coprime polynomial matrices such that $D_R(s)$ is column-reduced.

Let $C_L(s)$ be a square polynomial matrix with Hurwitz determinant (i.e. all its roots have strictly negative real parts), such that

$$C_L(s)C'_L(-s) = [N_L(s)B_1 + D_L(s)D_{21}][N_L(-s)B_1 + D_L(-s)D_{21}]'$$

and let $C_R(s)$ be a square polynomial matrix with Hurwitz determinant, such that

$$C'_R(-s)C_R(s) = [C_1N_R(-s) + D_{12}D_R(-s)]'[C_1N_R(s) + D_{12}D_R(s)].$$

The matrices $C_L(s)$ and $C_R(s)$ are called *spectral factors* and are uniquely determined up to right and left orthogonal factors, respectively.

Under the above assumptions, then there exists a unique constant solution $X_R$, $Y_R$ to the *polynomial Diophantine equation*

$$D_L(s)X_R + N_L(s)Y_R = C_L(s) \tag{4}$$

from which we build the feedback matrix

$$F = Y_R X_R^{-1}.$$

Similarly, there exists a unique constant solution $X_L, Y_L$ to the polynomial Diophantine equation

$$X_L D_R(s) + Y_L N_R(s) = C_R(s) \tag{5}$$

from which we build the feedback matrix

$$G = X_L^{-1} Y_L.$$

Then the unique $H_2$ optimal controller is given by

$$K(s) = -G(sI - A + FC_2 + B_2 G)^{-1} F.$$

A matrix fraction description can then be readily determined (see **"Polynomial and matrix fraction description"**).

From equations (4) and (5) we see that $H_2$ optimal control can be interpreted as a two-stage dynamics assignment problem, where the assigned optimal invariant polynomials are obtained by solving two polynomial matrix spectral factorization problems.

Now we treat a very simple example to illustrate the procedure. The plant data are given by

$$\begin{array}{lll} A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} & B_1 = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} & & D_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} & D_{21} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \end{array}$$

and we can check that all the standard assumptions are satisfied.

First we build a left MFD by extracting a minimal polynomial basis of a left null-space

$$\begin{bmatrix} N_L(s) & D_L(s) \end{bmatrix} \begin{bmatrix} sI - A \\ -C_2 \end{bmatrix} = 0.$$

We find

$$N_L(s) = \begin{bmatrix} s & 1 \end{bmatrix}, \quad D_L(s) = s^2.$$

Then we build a right MFD by extracting a minimal polynomial basis of a right null-space

$$\begin{bmatrix} sI-A & -B_2 \end{bmatrix} \begin{bmatrix} N_R(s) \\ D_R(s) \end{bmatrix} = 0.$$

We find

$$N_R(s) = \begin{bmatrix} 1 \\ s \end{bmatrix}, \quad D_R(s) = s^2.$$

Following this, we must perform the spectral factorization

$$\begin{aligned} C_L(s)C_L'(-s) &= [N_L(s)B_1 + D_L(s)D_{21}][N_L(-s)B_1 + D_L(-s)D_{21}]' \\ &= \begin{bmatrix} \sqrt{2}s & 1 & s^2 \end{bmatrix} \begin{bmatrix} -\sqrt{2}s \\ 1 \\ s^2 \end{bmatrix} \\ &= 1 - 2s^2 + s^4 = (s-1)^2(s+1)^2 \end{aligned}$$

and we find the Hurwitz spectral factor

$$C_L(s) = s^2 + 2s + 1 = (s+1)^2.$$

Then we perform the spectral factorization

$$\begin{aligned} C_R'(-s)C_R(s) &= [C_1N_R(-s) + D_{12}D_R(s)]'[C_1N_R(s) + D_{12}D_R(s)] \\ &= \begin{bmatrix} 2 & s^2 \end{bmatrix} \begin{bmatrix} 2 \\ s^2 \end{bmatrix} \\ &= 4 + s^4 = (s^2 - 2s + 2)(s^2 + 2s + 2) \end{aligned}$$

and we find the Hurwitz spectral factor

$$C_R(s) = s^2 + 2s + 2 = (s+1+i)(s+1-i).$$

The next step consists in solving Diophantine equation (4)

$$\begin{bmatrix} s^2 & s & 1 \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \end{bmatrix} = 1 + 2s + s^2.$$

We easily find

$$X_R = 1, \quad Y_R = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

so

$$F = Y_R = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

Then we solve Diophantine equation (5)

$$\begin{bmatrix} X_L & Y_L \end{bmatrix} \begin{bmatrix} s^2 \\ 1 \\ s \end{bmatrix} = s^2 + 2s + 2.$$

We easily find

$$X_L = 1, \quad Y_L = \begin{bmatrix} 2 & 2 \end{bmatrix}$$

so

$$G = Y_L = \begin{bmatrix} 2 & 2 \end{bmatrix}.$$

Finally, we can build a right MFD for the transfer function $(sI - A + FC_2 + B_2G)^{-1}F$ by extracting a minimal polynomial basis of a right null-space

$$\begin{bmatrix} sI - A + FC_2 + B_2G & -F \end{bmatrix} \begin{bmatrix} \bar{N}_K(s) \\ D_K(s) \end{bmatrix} = 0.$$

We find

$$N_K(s) = \begin{bmatrix} 2s+5 \\ s-4 \end{bmatrix}, \quad D_K(s) = s^2 + 4s + 7$$

so that with

$$N_K(s) = -G\bar{N}_K(s) = -6s - 2$$

we obtain the $H_2$ optimal controller with transfer function

$$K(s) = -\frac{N_K(s)}{D_K(s)} = -\frac{6s+2}{s^2+4s+7}.$$

## 3. Numerical methods for polynomial matrices

In the previous section we have shown that the main ingredients when solving control problems with polynomial techniques are

- the polynomial Diophantine equation, or linear system of equations over polynomial matrices
- the spectral factorization equation, or quadratic system of equations over polynomial matrices.

In this section we shortly review the numerical routines that are available to address these problems.

### 3.1. Diophantine equation

A polynomial Diophantine equation has generally the form

$$A_L(s)X_R(s) + B_L(s)Y_R(s) = C_L(s)$$

where polynomial matrices $A_L(s)$, $B_L(s)$ and $C(s)$ are given, and polynomial matrices $X_R(s)$ and $Y_R(s)$ are to be found. Usually, $A_L(s)$ has special row-reducedness properties, and a solution $X_R(s)$ of minimum column degrees is generally sought. We can find the dual equation

$$X_L(s)A_R(s) + Y_L(s)B_R(s) = C_R(s)$$

or even two-sided equations of the type

$$X_L(s)A_R(s) + B_L(s)Y_R(s) = C(s)$$

where the unknowns $X_L(s)$ and $Y_R(s)$ multiply matrices $A_R(s)$ and $B_L(s)$ both from the left and from the right, respectively. With the help of the Kronecker product, we can however always rewrite these equations to the general form

$$A(s)X(s) = B(s) \tag{6}$$

where $A(s)$ and $B(s)$ are given, and $X(s)$ is to be found.

There exist several methods to solve equation (6):

- *Sylvester matrix*. We assume that the expected degree $n$ of the solution $X(s)$ is given, and we identify like powers of indeterminate $s$ to obtain a linear system of equations

$$\begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & A_1 & \ddots & \\ A_m & \vdots & & A_0 \\ & A_m & & A_1 \\ & & \ddots & \vdots \\ & & & A_m \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} B_0 \\ B_1 \\ \vdots \\ B_{n+m} \end{bmatrix}.$$

  The block matrix appearing at the left hand-side is called a Sylvester matrix. The linear system has a special banded Toeplitz structure, so that fast algorithms can be designed for its solution. If there is no solution of degree $n$, then we may try to find a solution of degree $n+1$ and so on. Fast structured algorithms can be designed to use the information obtained at the previous step to improve the performance. When properly implemented, these algorithms are numerically stable.

- *Interpolation*. We assume that polynomial matrices are not known by coefficients but rather by values taken at a certain number of points $s_i$. Then, from the matrix values $A(s_i)$ and $B(s_i)$ we build a linear system of equations. The block matrix appearing in this system can be shown to be the product of a block Vandermonde matrix times the above Sylvester matrix. Basically, this means that the problem will be badly conditioned, unless we properly select the interpolation points. The best choice, corresponding to a perfect conditioning of the Vandermonde matrix, are complex points equidistantly spaced along the unit circle. The interpolated linear system can then be solved by any non-structured algorithm based on the numerically stable QR factorization.

- *Polynomial reduction*. We reduce matrix $A(s)$ to some special form, such as a triangular Hermite form, with the help of successive elementary operations (see **“Polynomial and matrix fraction description”** for more information on the Hermite form and elementary operations). This method is not recommended from the numerical point of view.

Finally, we point out that in the special case that $B(s) = 0$ in equation (6) then solving the Diophantine equation amounts to extracting a polynomial basis $X(s)$ for the right *null space* of polynomial $A(s)$, i.e. we have to solve the equation

$$A(s)X(s) = 0.$$

This problem typically arises when converting from state-space to matrix fraction description (MFD), or from right MFD to left MFD and vice-versa.

There are many bases for the right null-space of $A(s)$, and generally we are interested in the one which is row-reduced with minimum row degrees, the so-called *minimal polynomial basis*. There exist a specific algorithm to extract the minimal polynomial basis of a null space. It is based on numerically stable operations and takes advantage of the special problem structure. The degree of the polynomial basis, usually not known in advance, is found as a byproduct of the algorithm.

### 3.2. Spectral factorization equation

In the continuous-time spectral factorization problem we are given a para-Hermitian polynomial matrix $B(s)$, i.e. such that

$$B(s) = B'(-s),$$

and we try to find a Hurwitz stable matrix $A(s)$ (i.e. all its zeros are located in the open left half-plane) such that

$$A'(-s)A(s) = B(s).$$

Under the assumption that $B(s)$ is positive definite when evaluated along the imaginary axis, such a spectral factor always exists.

In the discrete-time case, the para-Hermitian polynomial matrix satisfies

$$B'(z^{-1}) = B(z)$$

so $B(z)$ features positive and negative powers of $z^{-1}$. When $B(z)$ is positive definite along the unit circle, there always exists a Schur spectral factor $A(z)$ (i.e. all its zeros are located in the open unit disk) such that

$$A'(z^{-1})A(z) = B(z).$$

The problem is relatively easy to solve in the scalar case (when both $A(s)$ and $B(s)$ are scalar polynomials). In the matrix case the problem is more involved. Basically we can distinguish the following approaches:

- *Diagonalization*. It is based on the reduction into a diagonal form with elementary operations. It is not recommended for the numerical point of view.
- *Successive factor extraction*. It requires first to compute the zeros of the polynomial matrix. Typically this is done with the numerically stable Schur decomposition. Then the zeros are extracted successively in a symmetric way.
- *Interpolation*. It requires first to compute the zeros of the polynomial matrix. Then, all the zeros are extracted simultaneously via an interpolation equation.
- *Algebraic Riccati equation*. It is a reformulation of the problem in the state-space setting. There exist several numerically efficient algorithms to solve algebraic Riccati equations.

It must be pointed out that in the scalar discrete-time case there exists still a more efficient FFT-based algorithm to perform polynomial spectral factorization. Unfortunately no nice extension of the results is available for the continuous-time case or the matrix case.

Finally, a more general polynomial spectral factorization problem is encountered in LQG game theory and $H_\infty$ optimal control, where a stable polynomial matrix $A(s)$ and a possibly indefinite constant matrix $J$ are sought such that (in continuous-time)

$$A'(-s)JA(s) = B(s)$$

for a given para-Hermitian matrix $B(s)$. It is called the *J-spectral factorization problem*. There exists also a discrete-time version of the problem. All the algorithms mentioned above can handle $J$-spectral factorization.

## 4. Conclusion

With the help of three simple control problems, namely dynamics assignment, deadbeat regulation and $H_2$ optimal control, we have shown how polynomial methods naturally arise as an alternative to state-space methods. Dynamics assignment amounts to solving a special linear equation over polynomial matrices called the Diophantine equation. Deadbeat control can be seen as a special case of dynamics assignment where all the poles are located at the origin. Minimization of a quadratic cost in $H_2$ optimal control implies solving a special quadratic equation over polynomial matrices called the spectral factorization equation. Numerical methods available to solve these polynomial equations were then reviewed.

Although not mentioned in this article, several other control problems can be solved with polynomial techniques. Most notably the $H_\infty$ problem and most of the robust control problems can fit the approach. When applying polynomial methods to solve these problems, the designer will face the similar linear and quadratic matrix polynomial equations that arised when solving the simpler problems described here.

## Bibliography

Grimble M. J. and Kučera V. (Editors) (1996). *Polynomial methods for control systems design*. London: Springer Verlag. [Collection of papers on the polynomial approach to solving both $H_2$ and $H_\infty$ control and filtering problems.]

Kučera V. (1979). *Discrete linear control: the polynomial equation approach*. London: Wiley. [This is a pioneering textbook that paved the way for the use of polynomials, polynomial matrices and polynomial equations to solve linear control problems.]

Kučera V. (1991). *Analysis and design of discrete linear control systems*. London: Prentice Hall. [Textbook that investigates connections between the state-space approach and the polynomial approach to several standard linear control problems. Most of the material proposed in this article was taken from this reference.]

Kwakernaak H. (1993). *Robust control and $H_\infty$ optimization - Tutorial paper*. Automatica 29(2):255–273. [Tutorial exposition of the polynomial approach to $H_\infty$ optimal regulation theory, with a focus on the mixed sensitivity problem for robust control system design.]

Kwakernaak H. and Šebek M. (1994). *Polynomial J-spectral factorization*. IEEE Trans. Autom. Control 39(2):315–328. [Collection of several algorithms for the $J$-spectral factorization of a polynomial matrix.]

Kwakernaak H. and Šebek M. (1998)). *The Polynomial Toolbox for Matlab*, Prague, CZ: PolyX Ltd., Web site at www.polyx.com or www.polyx.cz.[The Matlab Toolbox for polynomials, polynomial matrices and their application in systems, signals and control. Contains over 200 functions including various solvers for linear and quadratic matrix polynomial equations.]