POLYNOMIAL TOOLBOX™ FOR MATLAB™ AND COMPLEX POLYNOMIALS IN COMMUNICATIONS *

Martin Hromčík[‡], **Michael Šebek**[‡], **Jan Ježek**[†]

[‡] Centre for Applied Cybernetics
Czech Technical University, Prague, Czech Republic
and

[†] Institute of Information Theory and Automation
Prague, Czech Republic

Abstract

Quite recently the polynomial design methods found a new great field of application outside the control area: the algebraic approach have been used successfully in signal processing and mobile communications. In contrast to the control systems synthesis, polynomials and polynomial matrices with *complex* coefficients are often required when designing filters, equalizers, decouplers and other components of mobile phones for instance.

Polynomial Toolbox for MATLAB admits complex polynomials in most computations, including Diophantine equations and spectral factorizations. As a result, the toolbox appears a suitable tool for rapid prototyping whenever polynomial design routines with complex coefficients are required.

The objective of this report is twofold. First we would like to explain in a clear and popular manner how the complex coefficients arise in technical practice. Based on this motivation, we will present important numerical algorithms for complex polynomials and polynomial matrices and their implementation in the Polynomial Toolbox for MATLAB. The power of the Toolbox will be illustrated by selected numerical examples involving complex coefficients finally.

1 Introduction

The polynomial or algebraic methods constitute the third basic class of control systems design procedures, next to the classical frequency routines and state-space techniques.

The origin of the polynomial concept is dated back to the early 70s. Systems are described by input-output relations, however, the transfer functions are not regarded as functions of complex variable but as algebraic objects. Design procedures are then reduced to algebraic operations with polynomials and polynomial matrices, typically to solving algebraic polynomial equations and polynomial spectral factorizations. This approach not only enables to resolve many existing control problems in a more elegant and unifying way but also provides further insight into the structure of control systems and shows new relationships between various control tasks [13]. Moreover, relying on input-output relations only, polynomial methods often appear more transparent and straightforward compared to state-space routines.

Within the control community there is no need to take polynomial matrices with *complex* coefficients into account in fact. That is also why the great majority of research results on polynomial methods, including numerical algorithms, concern just the real case as a rule.

However, the ideas of the algebraic approach to control systems synthesis have recently been successfully applied also in problems out of the control area, namely in signals and communications [7, 8], and many of such problems naturally call for polynomials and polynomial matrices with complex coefficients. Speaking in broad terms, complex entries are able to carry twofold information about particular signal module and phase which is often desirable. For example, several mobile radio communication filtering

[*] The work of M. Hromčík and M. Šebek has been supported by the Ministry of Education of the Czech Republic under contract No. LN00B096. The work of Jan Ježek has been supported by the Grant Agency of the Czech Republic under contract No. 102/99/1368.

algorithms rely upon complex polynomials and solutions of related equations [2]. Vibrational systems and filters are additional examples of systems whose models involve complex coefficients [1]. When applying the algebraic design methods in these cases, solutions to linear and quadratic polynomial equations are sought.

The report is organized as follows. In the first section, relevance of complex polynomials in real-world problems is illustrated by an example of the mobile radio communication channel. Description of algorithms for Diophantine equations and spectral factorizations over complex polynomials and polynomial matrices which are implemented in the Polynomial Toolbox then follows. Finally, illustrative numerical examples involving complex coefficients are resolved using the Polynomial Toolbox functions.

2 Complex Coefficients in Communications

In this section the structure of the mobile radio communication channel is presented along with the way the complex valued polynomials are introduced. The facts presented here are mainly adopted from [2].

Basically, there are two reasons for going into complex computations. First, it often appears convenient to code the bitstream to be transmitted into complex numbers and send their real and imaginary parts using the same frequency range. Such a way, the bandwidth reserved for the channel is more efficiently exploited. However, even if the transmitted symbols are real, the amplitude response of the overall communication channel can become asymmetrical around the carrier frequency if interference or multipath propagation occurs, giving rise to a complex transfer function in the baseband representation. These effects will be thoroughly described further.

2.1 The Radio Transmission System

The radio communication channel typically consists of three parts: the transmitter, radio frequency (RF) channel and a receiver. The transmitter accepts data - time series of bits - from a source and transforms it in a form suitable for transmitting via the RF channel, typically to a band-limited continuous-time signal. The channel adds noise to the input signal and acts as a filter on the transmitted data. Therefore, the receiver cannot be just a pure ”reverse” of the transmitter but in addition it has to process the received signal to remove distortion introduced by the RF channel.

In the transmitter, the bit stream to be sent is divided into groups of bits which form digital *symbols* - real or complex numbers. By forming symbols, several bits can be transmitted at the same time.

As the signal for the radio channel must be continuous both in time and amplitude, the digital symbols are pulse-shaped first. This pulse-shaping is performed by a lowpass filter with a spectrum efficient impulse response $p(t)$. Such a way, the sequence of symbols $\{u_k\}$ is transformed in a sequence of continuous-time pulses

$$s_b(t) = \sum_{k=1}^{N} u_k p(t - kT_s)$$

where $1/T_s$ is the symbol rate. This signal is real valued whenever the symbols accepted are real and complex otherwise.

After pulse shaping, the modulation follows: the spectrum of $s_b(t)$ (the *baseband signal*) is shifted to a high frequency band suitable for transmission. For both real and complex valued symbols, the modulation can be accomplished by multiplying the baseband signal $s_b(t)$ by a complex-valued carrier $e^{j\omega_c t}$ and transmitting the real part of corresponding signal:

$$s_p(t) = Re\{s_b(t)e^{j\omega_c t}\} = Re\{s_b(t)\}\cos\omega_c t - Im\{s_b(t)\}\sin\omega_c t \tag{1}$$

At this point, the ineffectiveness of real-valued symbols for transmitting information which was mentioned earlier can be clarified: Since the spectrum $S_b(\omega)$ of real-valued baseband signal $s_b(t)$ resulting from real symbols u_k is symmetric with respect to $\omega = 0$, related passband signal $S_p(\omega)$ is symmetric around ω_c. Hence all information carried by $s_p(t)$ is contained only in one half of the devoted bandwidth, see Figure 1a. On the contrary, once we allow complex symbols, the symmetries of spectra $S_b(\omega)$ and $S_p(\omega)$ disappear and the transmitted information is spread over all frequencies allocated in the reserved bandwidth (Figure 1b). Technically, this is accomplished by employing two orthogonal carriers $\cos\omega_c t$ and $-\sin\omega_c t$ for transmitting real and imaginary parts of the symbols, see formula (1).

The RF channel is a link between the transmitter and the receiver. The passband signal is carried by electromagnetic waves. However, the space between the transmitter and receiver station often contains obstacles such as buildings, mountains, etc. In addition, the waves are reflected and scattered when touching grounds or other objects, namely if the the carrier frequency is high. As a result, signals which traverse different paths reach the receiver at different times. This property is called the delay spread. These effects cause that the transfer function describing the radio channel will in general not be flat within the spectrum of the transmitted passband signal $s_p(t)$ and will distort it accordingly, see Figure 2. Hence the baseband representation of the received signal becomes nonsymmetric around $\omega = 0$ even if the transmitted sequence is real-valued. This implies that the transfer function of the baseband representation of the channel features *complex coefficients*.

It is clear now that the role of the receiver is twofold. First of all, reversed versions of all operations performed by the transmitter are to be applied. In case the channel were ideally perfect, the desired submitted symbols would be exactly recovered.

However, as we tried to explain above, in reality the received and transmitted symbols differ. Hence some additional manipulations with the set of received symbols have to follow to eliminate the imperfections of radio transmission. This problem can be mathematically formulated as an optimization task. For computational reasons, the cost criterion is often chosen as a quadratic form of involved signals. In that case the whole theory of Wiener and Kalman filtering can be directly applied. In addition the overall communication channel is usually described by its transfer function as a ratio of two polynomials. Then the polynomial methods for control systems design can be easily adopted. The main difference compared to the control theory is that *complex* polynomials are often addressed.

2.2 Designing Filters for Mobile Communications

Many results on applying polynomial design methods in filters and equalizers for mobile communications have been achieved by the Signals and Systems Group at the University of Uppsala which also cooperates with the PolyX Ltd. closely. Their algorithms based on polynomial approach for LQ optimal feedforward filters and LQ optimal decision feedback equalizers have been applied by the Ericsson company in their phones for instance. An interested reader is referred to [3], [4], [5]
(http://www.signal.uu.se/Publications/pbookch.html) for detailed description of particular procedures.

As one can check, the crucial computational parts of all the papers cited above are the Diophantine equations, often two-sided and symmetric, and polynomial spectral factorizations. And not only scalar complex polynomials but also complex polynomial matrices are of interest to tackle more complex problems of mobile communications involving multiple-antenna arrays for instance [6].

In the next sections we will demonstrate that the Polynomial Toolbox is capable to resolve such advanced tasks.

3 Polynomial Toolbox and Complex Coefficients

In the previous sections, the importance of polynomials with complex coefficients in communications was enlightened and numerical algorithms for most advanced computations were presented. Now some particular linear equations and spectral factorization problems involving complex polynomials and polynomial matrices of various degree and size will be resolved using the Polynomial Toolbox for MATLAB.

3.1 Diophantine Equations

The Polynomial Toolbox provides nine solvers for various kinds of linear (matrix) polynomial equations. We will concentrate on two in practice most important types: the symmetric equation $A^*X + X^*A = B$ and one-sided Diophantine equation $AX + BY = C$.

First let us create a 3-by-3 polynomial matrix A in variable 'z' of degree 4 with complex coefficients using the Polynomial Toolbox `prand` command:

```
>> A = prand(5,5,'z') + prand(5,5,'z')*j
Polynomial matrix in z: 5-by-5,  degree: 5
A =
Column 1
 1.6-0.046i + (0.28-0.3i)z + (1.5+0.42i)z^2 + (-0.74-0.56i)z^3 + (0.68+0.64i)z^4 + (0.5+0.86i)z^5
 0.45-2.5i + (1.3-0.19i)z + (-0.57+0.068i)z^2 + (-1.5+0.85i)z^3 + (0.85+0.46i)z^4 + (-0.75-2.1i)z^5
 0.21+0.22i + (-0.082-0.23i)z + (-0.15+0.77i)z^2 + (-0.29+1.1i)z^3 + (-0.38-0.83i)z^4 + (0.57+2.1i)z^5
 1-0.39i + (-0.1-1i)z + (0.22+0.085i)z^2 + (0.53-0.33i)z^3 + (3.1-1.1i)z^4 + (1.2-0.1i)z^5
 0.27-0.24i + (-0.88-0.63i)z + (-0.12-0.7i)z^2 + (0.82-0.39i)z^3 + (-0.89-0.45i)z^4 + (-0.43+0.95i)z^5
Column 2
-0.43-0.22i + (2-0.69i)z + (-1.1+0.27i)z^2 + (1.1-0.12i)z^3 + (0.43-0.56i)z^4 + (-0.57-0.99i)z^5
```

```
 0.83+0.88i + (-0.058-0.81i)z + (0.2-0.98i)z^2 + (0.66-1i)z^3 + (-0.63+1.2i)z^4 + (0.88-0.95i)z^5
-0.45-0.24i + (0.66+0.25i)z + (0.6+1.1i)z^2 + (1.1+1.4i)z^3 + (0.49-1.2i)z^4 + (1.8+0.044i)z^5
 0.76-0.081i + (-0.001+0.36i)z + (0.53-0.023i)z^2 + (-0.19-0.12i)z^3 + (1.4+3.7i)z^4 + (0.74+1.1i)z^5
 0.29+0.72i + (-0.66+0.23i)z + (1.8+0.71i)z^2 + (-0.69+0.72i)z^3 + (0.17-0.36i)z^4 + (-0.19+1.2i)z^5
Column 3
-1.5+0.36i + (0.84-1.4i)z + (-2+2.4i)z^2 + (0.26-1.2i)z^3 + (-0.029-0.021i)z^4 + (0.81-1.4i)z^5
 2.4+0.77i + (0.17-2.5i)z + (-0.18-1.5i)z^2 + (-0.47-1.3i)z^3 + (-0.93-0.12i)z^4 + (-2.7-1.2i)z^5
-0.026-1.5i + (0.96+1.6i)z + (-1.4-0.36i)z^2 + (-0.097+1.6i)z^3 + (-1.5-0.013i)z^4 + (-0.076-1.5i)z^5
 0.34-0.85i + (0.066+0.98i)z + (0.52+0.13i)z^2 + (0.059-1.3i)z^3 + (-1-0.49i)z^4 + (0.049-0.23i)z^5
-1.9-0.37i + (0.61-0.069i)z + (0.4+1.1i)z^2 + (0.61-1.1i)z^3 + (-1.6+0.6i)z^4 + (-0.38-0.35i)z^5
Column 4
-0.011-0.9i + (-1.1-2.1i)z + (0.12+0.77i)z^2 + (-0.98+1.6i)z^3 + (1.4-0.18i)z^4 + (-0.14+0.021i)z^5
 0.1+1.6i + (0.65+0.38i)z + (-0.026+0.39i)z^2 + (0.35-0.053i)z^3 + (-1.8+0.24i)z^4 + (-1.6-1.2i)z^5
 0.2+0.19i + (0.41-2.9i)z + (-0.12+0.28i)z^2 + (2.2-0.28i)z^3 + (0.011+1i)z^4 + (0.58-1.8i)z^5
-2.5+0.88i + (-0.29-0.63i)z + (-0.5+1i)z^2 + (0.58-0.54i)z^3 + (-0.77+0.0014i)z^4 + (0.21-0.52i)z^5
 1.4-1.5i + (0.53+2.3i)z + (0.16+0.15i)z^2 + (-2.2-0.76i)z^3 + (1.1-1.5i)z^4 + (-0.23+0.1i)z^5
Column 5
-0.81-0.14i + (0.32-0.98i)z + (0.00072+0.22i)z^2 + (-1.2-0.25i)z^3 + (0.79+1.2i)z^4 + (1.8-0.3i)z^5
 0.39+1.4i + (-2-1.3i)z + (1.2-0.16i)z^2 + (0.65+0.11i)z^3 + (-0.06-1.1i)z^4 + (-1.2-1.7i)z^5
 1+0.059i + (1.5+0.67i)z + (-1.1-0.92i)z^2 + (-0.63-0.66i)z^3 + (-1.3-0.25i)z^4 + (-1.5+0.56i)z^5
-0.68-0.0068i + (0.085+0.37i)z + (0.36-0.14i)z^2 + (-0.48-0.25i)z^3 + (-1.5+0.88i)z^4 + (1.3-0.75i)z^5
 0.5+1.5i + (0.37+0.74i)z + (0.41-1.2i)z^2 + (-0.52+0.32i)z^3 + (-0.46-0.81i)z^4 + (0.59+0.46i)z^5
```

Similarly, a discrete-time symmetric 3-by-3 matrix B of degree 8 is created:

```
>> B = prand(5,5,'z') + prand(5,5,'z')*j;
>> B = B*B';
```

Solution of related discrete-time symmetric polynomial equation $A^*X + X^*A = B$ can be achieved by calling the Polynomial Toolbox `axxab` command:

```
>> X = axxab(A,B)
Polynomial matrix in z: 5-by-5,  degree: 5
X =
Column 1
 2.8e+002+0i + (-38-1.5e+002i)z + (-74-3.4e+002i)z^2 + (-31-3e+002i)z^3 + ...
 (2.5e+002+51i)z + (-50-42i)z^2 + (2.4e+002-1.7e+002i)z^3 + ...
 (7.6e+002+3.1e+002i)z + (79+7.3e+002i)z^2 + (-40+93i)z^3 + ...
 (-3.2e+002+70i)z + (-2.3e+002-3.5e+002i)z^2 + (-80+1.2e+002i)z^3 + ...
 (-1.8e+002+2.1e+002i)z + (-4.5e+002+2.3e+002i)z^2 + ...
Column 2
 77+1.6i + (-67+1.9e+002i)z + (-72+2.7e+002i)z^2 + (-6.1e+002+2.6e+002i)z^3 + ...
 80+0i + (-7.6+4.3i)z + (-3e+002-1.9e+002i)z^2 + (-1.1e+002-4.3e+002i)z^3 + ...
 (5.6e+002+3.7i)z + (-4.2e+002-8.3e+002i)z^2 + (9.6e+002-5.4e+002i)z^3 + ...
 (-1.7e+002+1.2e+002i)z + (2.3e+002+2.3e+002i)z^2 + (-1.2e+002+3.6e+002i)z^3 + ...
 (1.5e+002+3.2e+002i)z + (4.8e+002-2.7e+002i)z^2 + (4.8e+002+5.7e+002i)z^3 + ...
Column 3
 2.1-94i + (-2.1e+002+1.3e+002i)z + (2.6e+002-1.3e+002i)z^2 + ...
 1.4e+002-65i + (-2.1e+002-2.5e+002i)z + (-1.1e+002-13i)z^2 + ...
 4e+002+0i + (5e+002-7.3e+002i)z + (-1.2e+003-4.1e+002i)z^2 + ...
 (-1.2e+002+3.1e+002i)z + (4.3e+002-90i)z^2 + (-3.8e+002+1.9e+002i)z^3 + ...
 (5.5e+002+3.2e+002i)z + (-2e+002-5.9e+002i)z^2 + (12+6.2e+002i)z^3 + ...
Column 4
-14+2.2e+002i + (4.8e+002+5.1e+002i)z + (-1.3e+002+1.3e+002i)z^2 + ...
-4.5+1.9e+002i + (-3.4e+002+12i)z + (1e+002+1.9e+002i)z^2 + (1.9e+002+13i)z^4 + ...
 1.3e+002+1.7e+002i + (-4.9e+002-1.3e+003i)z + (-22-3.8i)z^2 + (-95-2.6e+002i)z^3 + ...
 9.5+0i + (5.4e+002+84i)z + (-1.4e+002+1.8e+002i)z^2 + (2.2e+002-46i)z^3 + ...
 (7.3e+002-6.7e+002i)z + (-2e+002-20i)z^2 + (4.4e+002-29i)z^3 + ...
Column 5
-4.1e+002+5.1e+002i + (-1.8e+002+2.8e+002i)z + (6.5e+002-1.7e+002i)z^2 + ...
-2.9e+002-1.6e+002i + (-4.1e+002-2e+002i)z + (4.2e+002+3.1e+002i)z^2 + ...
 1.5e+002-8.1e+002i + (-2.1e+002-8.5e+002i)z + (-2.6e+002+5.8e+002i)z^2 + ...
 1.4e+002+2.8e+002i + (2.7e+002+2.3e+002i)z + (25-1.1e+002i)z^2 + ...
 6.2e+002+0i + (6.9e+002-46i)z + (-5.1e+002-2e+002i)z^2 + (62-1.8e+002i)z^3 + ...
```

The following check proves accuracy of obtained result:

```
 A'*X + X'*A - B
Zero polynomial matrix: 5-by-5,  degree: -Inf
ans =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

Besides the symmetric equations, also all one-sided Diophantine equation solvers can be addressed by complex polynomial matrices as well. For inastance, having defined complex polynomial matrices A, B, C of degree 5 and size 5-by-5, the solution X, Y of the equation $AX + BY = C$ is reached easily by the following Polynomial Toolbox command (along with a standard check):

```
>> A = prand(5,5,'z') + prand(5,5,'z')*j;
>> B = prand(5,5,'z') + prand(5,5,'z')*j;
>> C = prand(5,5,'z') + prand(5,5,'z')*j;
>> [X,Y] = axbyc(A,B, C);
>> A*X + B*Y - C
Zero polynomial matrix: 5-by-5,  degree: -Inf
ans =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

All linear polynomial equation solvers of the Polynomial Toolbox are based on powerful linear *constant* matrix solvers built into MATLAB. As a result, the Polynomial Toolbox functions for polynomial equations are not only very accurate as it was shown above, but also pretty fast. For instance, each of presented examples did not consume more than 1 second on a PC notebook with Celeron 500 MHz, 64 MB RAM and MATLAB 6.0.

3.2 Polynomial Spectral Factorization

Scalar complex polynomials are considered first. In this case, the FFT based algorithm can be highly recommended for its speed and reliability.

A symmetric polynomial of degree 500 to be factored, positive on the unit circle, is created by the following commands:

```
>> p = prand(250,'z') + j*prand(250,'z');
>> p = p*p';
```

Applying the FFT based routine with 2^{12} Fourier points the spectral factor is received in a fraction of second:

```
>> tic, f = fftspf(p,2^14); toc;
elapsed_time =
    0.2200
```

Correctness checks follow now. Note that some additional Polynomial Toolbox functions and operators are applied to complex polynomials (actually, all operators and standard functions work for complex polynomials and polynomial matrices automatically):

```
>> isstable(f)
ans =
     1
>> norm(f*f'-p)/norm(p)
ans =
  1.2460e-005
```

As one can see, both the stability requirement is met and the equation $ff^{\star} = p$ is fulfilled with good accuracy. Should the residue be too high for a specific application, there is no problem to take more Fourier points to improve the result:

```
>>  f = fftspf(p,2^17);
>> norm(f*f'-p)/norm(p)
ans =
  3.6538e-009

>>  f = fftspf(p,2^19);
>> norm(f*f'-p)/norm(p) ans =
     0
```

In the latter case, an exact (up to working precision) solution was found.

Let us switch to complex polynomial *matrices* now. We proceed similarly to the scalar case, starting with construction of a 5-by-5 symmetric complex matrix of degree 5:

```
>> P = prand(5,5,'z') + j*prand(5,5,'z');
>> P = P*P';
```

For the spectral factorization the spf function of the Polynomial Toolbox is used, implementing the Newton-Raphson iterative procedure.

```
>> tic, F = spf(P); toc;
elapsed_time =
    9.8300
>> F
Polynomial matrix in z: 5-by-5,  degree: 5
F =
Column 1
-0.75-0.25i + (0.27-0.38i)z + (0.23-1.1i)z^2 + (-1.4-1.5i)z^3 + (0.097-1.8i)z^4 + (4.8+0i)z^5
-0.55-0.0062i + (0.3+0.38i)z + (-0.76-0.37i)z^2 + (-0.1-0.84i)z^3 + (-0.13+0.46i)z^4
-0.41+0.93i + (0.86+0.3i)z + (-0.45-0.31i)z^2 + (-1.6+1.6i)z^3 + (0.34+0.34i)z^4
-0.31+0.048i + (0.34-0.78i)z + (-0.38-0.28i)z^2 + (-0.69+0.25i)z^3 + (-0.31-0.68i)z^4
-0.81+0.11i + (0.14-0.54i)z + (0.19+0.61i)z^2 + (-0.25+0.22i)z^3 + (-0.35+0.9i)z^4
Column 2
 1.6+0.6i + (0.84+1.4i)z + (0.83-0.71i)z^2 + (-1.6+0.5i)z^3 + (0.26+1.1i)z^4 + (0.015+0.92i)z^5
-0.74-0.73i + (0.31+1.6i)z + (-0.81-0.46i)z^2 + (0.75+0.13i)z^3 + (0.15-0.45i)z^4 + (4.8+0i)z^5
 0.05+0.29i + (-0.99-0.59i)z + (0.45+1.9i)z^2 + (-0.74+0.45i)z^3 + (-0.17-1.5i)z^4
-0.75-0.17i + (-0.96-0.98i)z + (-1.1+0.035i)z^2 + (-0.28+1i)z^3 + (0.36-0.64i)z^4
-0.25-0.076i + (0.34+0.36i)z + (0.64-0.31i)z^2 + (0.54+1i)z^3 + (-0.75+1i)z^4
Column 3
-0.52+0.83i + (-0.95+0.71i)z + (1.1+0.83i)z^2 + (-0.2-0.8i)z^3 + (-0.44+0.68i)z^4(-2.1+0.2i)z^5
-0.37+0.3i + (-0.9-0.87i)z + (-0.63-0.33i)z^2 + (-0.72-0.7i)z^3 + (0.12+1.1i)z^4 + (-0.89+0.9i)z^5
-0.6+0.036i + (-0.69-0.55i)z + (0.67+0.35i)z^2 + (0.29+0.24i)z^3 + (-1.8-0.69i)z^4 + (5.1+0i)z^5
 0.57-0.66i + (0.57-0.27i)z + (-0.82+0.19i)z^2 + (-1.3+0.34i)z^3 + (0.68+0.14i)z^4
 0.084-0.33i + (0.5-0.59i)z + (-0.92+0.91i)z^2 + (-0.12+1.3i)z^3 + (-0.086+1i)z^4
Column 4
 0.55+0.31i + (1.3-0.63i)z + (-2-0.63i)z^2 + (-1.6+0.23i)z^3 + (-0.71-0.84i)z^4 + (-2.2+0.0018i)z^5
-0.15-0.8i + (-0.25-1.1i)z + (-1.2-0.51i)z^2 + (-0.81+0.028i)z^3 + (-0.49+0.25i)z^4 + (-0.5+0.7i)z^5
-0.26+0.13i + (-0.78-0.54i)z + (-0.49+0.79i)z^2 + (1.4+0.92i)z^3 + (0.24-0.083i)z^4 + (0.24+1.3i)z^5
-0.32-0.31i + (-1.3-0.23i)z + (0.32+2.2i)z^2 + (-0.68+0.79i)z^3 + (-0.4+2.2i)z^4 + (5.1+0i)z^5
-0.15-0.27i + (-0.79-0.044i)z + (0.36+1.5i)z^2 + (1.4-0.088i)z^3 + (-0.35-0.13i)z^4
Column 5
 1.7+0.73i + (0.22+0.025i)z + (-1.4+0.033i)z^2 + (-0.71+1.3i)z^3 + (0.46-0.2i)z^4 + (1.3+0.7i)z^5
-0.018+0.89i + (-0.35+0.31i)z + (-0.64-1.9i)z^2 + (1.4+0.65i)z^3 + (0.71+1.8i)z^4 + (-0.87+0.76i)z^5
 0.69-0.17i + (0.63-1i)z + (-1.4-1.5i)z^2 + (0.76+1.6i)z^3 + (0.94+0.027i)z^4 + (0.98-0.61i)z^5
 0.083-0.37i + (-0.72+0.62i)z + (-0.97+0.23i)z^2 + (-1.6-0.58i)z^3 + (0.45+1.8i)z^4 + (-1.4+0.39i)z^5
-0.53+0.032i + (0.24+1.3i)z + (-1.6-0.34i)z^2 + (-2.3+0.042i)z^3 + (1.2-0.21i)z^4 + (5.7+0i)z^5
```

Standard checks follow:

```
>> isstable(F)
ans =
     1
>> norm(F'*F-P)/norm(P)
ans =
  3.6995e-007
```

Also in this case the solution can be refined. By prescribing a higher desired accuracy (default is 10^{-8}), a more precise solutions are found:

```
>> F = spf(P,1e-10);
>> norm(F'*F-P)/norm(P)
ans =
  1.3141e-009

>> F = spf(P,1e-12);
>> norm(F'*F-P)/norm(P)
ans =
     0
```

Although the spectral factorization as a quadratic problem is much more involving than linear polynomial equations presented in the previous subsection, we can conclude that the Polynomial Toolbox succeeded in factoring relatively large complex polynomials and polynomial matrices at acceptable computational time.

4 Acknowledgements

The authors would like to thank Mikael Sternad from the University of Uppsala, Signals and Systems Department, for most valuable discussions concerning namely the role of complex polynomials, complex polynomial matrices and algebraic design methods in communications.

5 Conclusion

Performance of polynomial equation and spectral factorization solvers included in the Polynomial Toolbox for MATLAB was manifested in this report for complex valued coefficients. Motivation for developing such tools for complex (matrix) polynomials comes from the field of signals and communications as it is shown in the report by an example of a radio communication channel and its optimization. In addition to particular numerical examples, the algorithms used by respective solvers are also described.

Figures

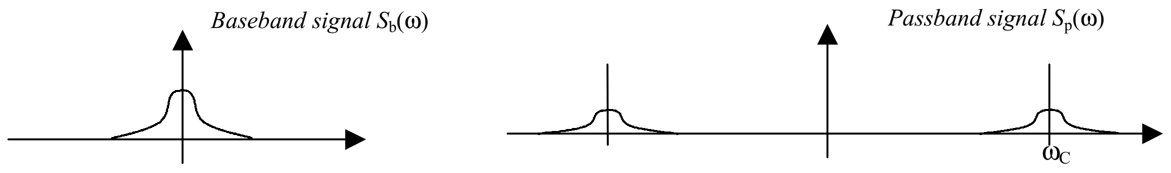

Figure 1a: Modulation for a real-valued baseband signal (symmetric spectra)

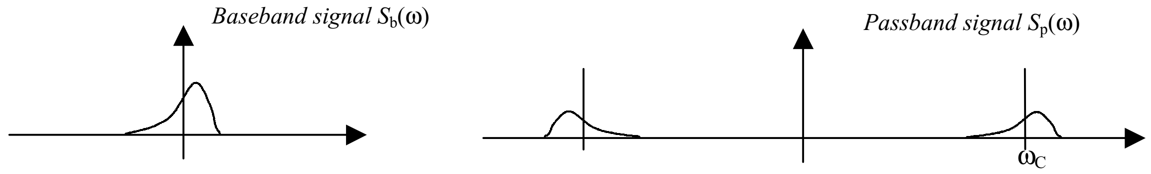

Figure 1b: Modulation for a complex-valued baseband signal

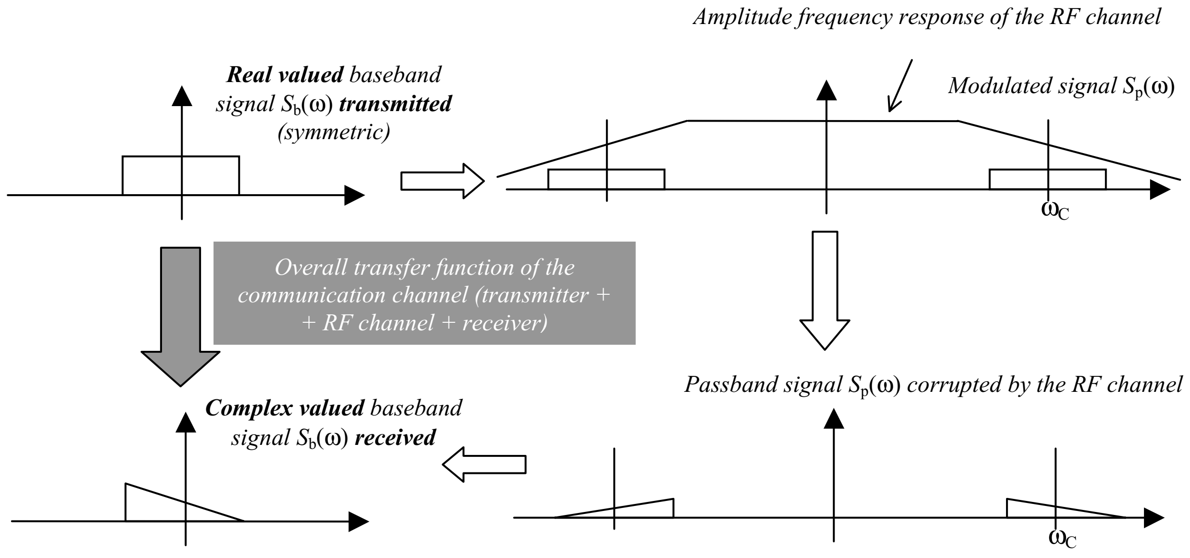
Amplitude frequency response of the RF channel
Real valued baseband signal $S_b(\omega)$ transmitted (symmetric)
Modulated signal $S_p(\omega)$
ω_C
Overall transfer function of the communication channel (transmitter + + RF channel + receiver)
Passband signal $S_p(\omega)$ corrupted by the RF channel
Complex valued baseband signal $S_b(\omega)$ received
ω_C

Figure 2: Scheme of the complete communication channel. Since the overall transfer function of the communication channel maps real-valued signals onto complex signals, it necessarily features complex coefficients.

References

[1] Henrion D., Ježek J. and Šebek M., *Efficient Algorithms for Discrete-Time Symmetric Polynomial Equations with Complex Coefficients*, Proceedings of the IFAC World Congress, Beijing, China, Vol. D, p. 159-164, July 1999.

[2] Lindbom L., *A Wiener Filtering Approach to the Design of Tracking Algorithms with Applications to Mobile Radio Communication*, PhD. Thesis, Signal Processing Group, Department of Technology, Uppsala University, Sweden, 1995.

[3] M. Sternad and A. Ahlen, *H-2 Design of Model-Based Nominal and Robust Discrete Time Filters.* Chapter 5 in M Grimble and V Kucera, eds: A Polynomial Approach to H-2 and H-infinity Robust Control Design. pp 171-222, Springer-Verlag, London, 1996.

[4] A. Ahlen and M. Sternad, *Derivation and Design of Wiener Filters using Polynomial Equations.* In C T Leondes, ed: Control and Dynamic Systems, Vol 64:Stochastic Techniques in Digital Signal Processing Systems. pp 353-418, Academic Press, New York, NY, 1994.

[5] A. Ahlen and M. Sternad, *Optimal Filtering Problems.* Chapter 5 in K Hunt, ed: Polynomial Methods in Optimal Control and Filtering, pp120-161, Control Engineering Series, Peter Peregrinus, London, 1993.

[6] Claes Tidestav, *The Multivariable Decision Feedback Equalizer, Multiuser Detection and Interference Rejection.* PhD Thesis, Uppsala University, 197p, ISBN 91-506-1371-5, December 1999. http://www.signal.uu.se/Publications/ptheses.html

[7] MacChi O., Adaptive Processing: The Least Mean Squares Approach and Applications in Transmission, John Willey and Sons,Chichester, Great Britain 1995.

[8] Proakis J.G. and Salehi M., *Communication Systems Engineering*, Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[9] Youla D.C., On the Fctorization of Rational Matrices, IEEE Transactions on Information Theory, IT-17, 172-189, 1961.

[10] Ježek J., Hromčík and Šebek M., *Spectral Factorization by Means of Discrete Fourier Transform*, submitted for the 8th IEEE Mediterranean Conference on Control and Education MED 2000, July 17-19, Patras, Greece.

[11] Henrion D., *Reliable Algorithms for Polynomial Matrices*, Ph.D. Thesis, Institute of Information Theory and Automation, Prague, 1998.

[12] Kailath T., *Linear Systems*, Prentice Hall, New Jersey (1980).

[13] Kučera V., *Analysis and Design of Discrete Linear Control Systems*, Academia Prague (1991).

[14] Vostrý Z., *New Algorithm for Polynomial Spectral Factorization with Quadratic Convergence I*, Kybernetika, 11, pp. 415-422, 1975.

[15] Vostrý Z., *New Algorithm for Polynomial Spectral Factorization with Quadratic Convergence II*, Kybernetika, 12, pp. 248-259, 1976.

[16] Ježek J. and Kučera V., *Efficient Algorithm for Matrix Spectral Factorization*,Automatica, vol. 29, pp. 663-669, 1985.

[17] Wilson G.T., *Factorization of the Covariance Generation Function of a Pure Moving Average Process*, SIAM Journal on Numerical Analysis, 6, pp. 1-7, 1969.

[18] Kwakernaak H., Šebek M., *PolyX Home Page*,
`http://www.polyx.cz/, http://www.polyx.com/`.

[19] Hromčík M, Ježek J and Šebek M., *New Algorithm for Spectral Factorization Based on FFT and Its Practical Applications*, Proceedings of the European Control Conference 2001, Porto, Portugal, September 4-7, 2001.