



# Time-Varying Systems and Computations

## Lecture 3

Klaus Diepold

1. November 2012

### Linear Time-Varying Systems

#### State-Space System Model

We aim to derive the matrix containing the time-varying impulse responses by inspection of a generic causal time-varying system. Recall the state equations

$$\begin{aligned} x_{k+1} &= A_k \cdot x_k + B_k \cdot u_k \\ y_k &= C_k \cdot x_k + D_k \cdot u_k, \end{aligned} \quad (1)$$

or in matrix form,

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \cdot \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \quad (2)$$

These state-space equations are similar to the equations for the time-invariant case except that now the entries  $\{A_k, B_k, C_k, D_k\}$  of the realization matrix are depending on the time index  $k$ , which means that these matrices can change from time step to time step, incorporating the time-variation that we are looking for. A generic causal time-varying system is drawn as shown in Figure 1.

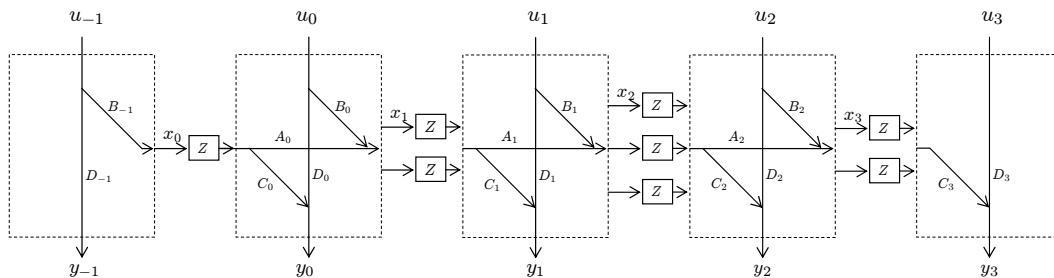


Abbildung 1: A time-varying system

Besides the time-varying system parameters  $\{A_k, B_k, C_k, D_k\}$  we can see that the dimension of the state-space, i.e. the number of delay elements may change from time step to time step. Similarly, the dimension of the input signal vector  $u_k$  and the dimension of the output signal vector  $y_k$  may change with time. However, we will not dwell in more detail on the the latter point.

## Time-Varying Impulse-Response

We seek to determine the I/O operator, i.e. the Toeplitz operator for this time-varying causal system. We proceed in a similar fashion as with the time-invariant case, by applying an impulse at time instant  $k$  and noting the output of the system (only in time-varying case, the response depends on  $k$  as well). Taking the impulse responses as the columns of the matrix  $\mathbf{T}$  we arrive at

$$\mathbf{T} = \begin{bmatrix} & k=0 & k=1 & k=2 & & \\ \ddots & \vdots & \vdots & \vdots & & \\ & \ddots & 0 & \vdots & & \\ & \ddots & \mathbf{D}_0 & 0 & \vdots & \\ & \ddots & \mathbf{C}_1\mathbf{B}_0 & \mathbf{D}_1 & 0 & \\ & \ddots & \mathbf{C}_2\mathbf{A}_1\mathbf{B}_0 & \mathbf{C}_2\mathbf{B}_1 & \mathbf{D}_2 & \ddots \\ & \ddots & \mathbf{C}_3\mathbf{A}_2\mathbf{A}_1\mathbf{B}_0 & \mathbf{C}_3\mathbf{A}_2\mathbf{B}_1 & \mathbf{C}_3\mathbf{B}_2 & \ddots \\ & & \mathbf{C}_4\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1\mathbf{B}_0 & \mathbf{C}_4\mathbf{A}_3\mathbf{A}_2\mathbf{B}_1 & \mathbf{C}_4\mathbf{A}_3\mathbf{B}_2 & \ddots \\ & & \vdots & \mathbf{C}_5\mathbf{A}_4\mathbf{A}_3\mathbf{A}_2\mathbf{B}_1 & \mathbf{C}_5\mathbf{A}_4\mathbf{A}_3\mathbf{B}_2 & \ddots \\ & & \vdots & \vdots & \mathbf{C}_6\mathbf{A}_5\mathbf{A}_4\mathbf{A}_3\mathbf{B}_2 & \ddots \\ & & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (3)$$

We can note from the above that

- $i^{th}$  column represents the impulse response of the system for an impulse input at  $k = i$ , i.e.

$$u_i = \begin{cases} 1 & i = k, \\ 0 & else \end{cases}$$

- $\mathbf{T}$  has the same functionality as Toeplitz matrix of the time-invariant case. In fact, if the system is time-invariant, all the matrices  $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k$  are identical, hence  $\mathbf{T}$  will exhibit Toeplitz structure.
- We can identify the system to be causal by lower-triangular structure of  $\mathbf{T}$ .
- The clear difference is that  $\mathbf{T}$  has no Toeplitz structure.
- Because of its time-varying nature, the dimension of the matrix  $\mathbf{T}$  may be finite.

## Time-Varying Transfer Operator

We discussed in the context of time-invariant systems that the concepts of diagonal expansion and the traditional z-transformation, Fourier-transformation. We discussed that the conventional concept of a transfer function does not translate directly to time-varying systems. However, we aim to derive a purely algebraic representation of the input-output operator  $\mathbf{T}$ , which we will call the *Toeplitz Operator*. To this end we combine the set of all time-varying state-space realization matrices into block diagonal matrices

$$\mathbf{A} = \begin{bmatrix} \ddots & & \\ & \mathbf{A}_k & \\ & & \ddots \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \ddots & & \\ & \mathbf{B}_k & \\ & & \ddots \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \ddots & & \\ & \mathbf{C}_k & \\ & & \ddots \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \ddots & & \\ & \mathbf{D}_k & \\ & & \ddots \end{bmatrix}.$$

Note that the dimension of the individual block matrices may change from time index to time index. Correspondingly, we combine all input-, output- and state-signals to form the vectors

$$\mathbf{x} = \begin{bmatrix} \vdots \\ x_k \\ \vdots \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \vdots \\ u_k \\ \vdots \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \vdots \\ y_k \\ \vdots \end{bmatrix},$$

whose length may also change from time step to time step. Using these block-diagonal matrices, the vectors and the causal shift-down operator  $\mathbf{Z}$  we can rewrite the state space equations (1) as

$$\begin{aligned} \mathbf{Z}^{-1}\mathbf{x} &= \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{y} &= \mathbf{C} \cdot \mathbf{x} + \mathbf{D} \cdot \mathbf{u} \end{aligned} \tag{4}$$

Starting out from equation (4) we can eliminate the state vector  $\mathbf{x}$  to arrive at a description of the output signal  $\mathbf{y}$  only depending on the input signal  $\mathbf{u}$  and the state space realization

$$\begin{aligned} \mathbf{x} &= (\mathbf{1} - \mathbf{Z}\mathbf{A})^{-1} \mathbf{Z}\mathbf{B}\mathbf{u} \\ \mathbf{y} &= \left[ \mathbf{D} + \mathbf{C}(\mathbf{1} - \mathbf{Z}\mathbf{A})^{-1} \mathbf{Z}\mathbf{B} \right] \mathbf{u}. \end{aligned} \tag{5}$$

Looking at this result we can identify a representation of the input-output, or Toeplitz operator in terms of the linear fractional transformation (LFT)

$$\mathbf{T} = \mathbf{D} + \mathbf{C}(\mathbf{1} - \mathbf{Z}\mathbf{A})^{-1} \mathbf{Z}\mathbf{B}, \tag{6}$$

which is given in terms of the state space realization matrices and the shift operator. This representation of the Toeplitz operator has an obvious structural resemblance with the standard transfer function representation for time-invariant state-space systems. However, note that  $\mathbf{Z}$  is the matrix representation of the shift operator and not a complex variable; the state-space matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  and  $\mathbf{D}$  are block-diagonal matrices.

## Construction of Transfer Operator

A look at Equation (6) does not immediately reveal that this linear fractional transformation based on block-diagonal matrices does actually represent the Toeplitz operator as shown in Equation (3). In this section we actually plug in the block-diagonal matrices in (6) to verify the correctness of this representation.

We consider the the block-diagonal matrices

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \ddots & & & & \\ & A_0 & & & \\ & & \boxed{A_1} & & \\ & & & A_2 & \\ & & & & \ddots \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \ddots & & & & \\ & B_0 & & & \\ & & \boxed{B_1} & & \\ & & & B_2 & \\ & & & & \ddots \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \ddots & & & & \\ & C_0 & & & \\ & & \boxed{C_1} & & \\ & & & C_2 & \\ & & & & \ddots \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \ddots & & & & \\ & D_0 & & & \\ & & \boxed{D_1} & & \\ & & & D_2 & \\ & & & & \ddots \end{bmatrix}, \end{aligned}$$

and the shift operator

$$\mathbf{Z} = \begin{bmatrix} \ddots & & & & & \\ & \ddots & 0 & & & \\ & & \mathbf{1} & \boxed{0} & & \\ & & & \mathbf{1} & 0 & \\ & & & & \mathbf{1} & \ddots \\ & & & & & \ddots \end{bmatrix}.$$

We can safely assume that the dimensions of the block-entries of all matrices are compatible to make the matrix multiplications meaningful. We first have a closer look at the middle part of Equation (6), which can be represented by the Neuman expansion, which comparable to geometric series for scalar quantities

$$(\mathbf{1} - \mathbf{Z}\mathbf{A})^{-1} = \mathbf{1} + \mathbf{Z}\mathbf{A} + (\mathbf{Z}\mathbf{A})^2 + (\mathbf{Z}\mathbf{A})^3 + \dots$$

The series expansion is based on powers of the term  $\mathbf{Z}\mathbf{A}$ . Let's investigate how these powers can be calculated by checking the example

$$(\mathbf{Z}\mathbf{A})^3 = \mathbf{Z}\mathbf{A}\mathbf{Z}\mathbf{A}\mathbf{Z}\mathbf{A} = (\mathbf{Z}\mathbf{A}\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}\mathbf{A}\mathbf{Z}^T\mathbf{Z}^T)(\mathbf{Z}\mathbf{Z}\mathbf{Z}\mathbf{A}\mathbf{Z}^T\mathbf{Z}^T\mathbf{Z}^T)\mathbf{Z}\mathbf{Z}\mathbf{Z},$$

where I have inserted various copies of the identity in terms of  $\mathbf{1} = \mathbf{Z}^T\mathbf{Z}$  to arrive at the right hand side of the equation. We will simplify notation for further manipulations by observing the shift of a block-diagonal matrix by one block in the south-east direction is given by

$$\mathbf{A}^{(1)} := \mathbf{Z}\mathbf{A}\mathbf{Z}^T = \begin{bmatrix} \ddots & & & & \\ & A_{-1} & & & \\ & & \boxed{A_0} & & \\ & & & A_1 & \\ & & & & \ddots \end{bmatrix},$$

which allows us to identify the appearance of shifted versions of the block-diagonal matrices

$$(\mathbf{Z}\mathbf{A})^3 = \underbrace{\mathbf{Z}\mathbf{A}\mathbf{Z}^T}_{\mathbf{A}^{(1)}} \underbrace{\mathbf{Z}^2\mathbf{A}\mathbf{Z}^{2T}}_{\mathbf{A}^{(2)}} \underbrace{\mathbf{Z}^3\mathbf{A}\mathbf{Z}^{3T}}_{\mathbf{A}^{(3)}} \mathbf{Z}^3 = \mathbf{A}^{(1)}\mathbf{A}^{(2)}\mathbf{A}^{(3)}\mathbf{Z}^3.$$

This notional simplification leads to another notational simplification

$$\mathbf{A}^{[n]} := \mathbf{A}^{(1)}\mathbf{A}^{(2)}\dots\mathbf{A}^{(n)},$$

such that we can rewrite the expression for  $(\mathbf{Z}\mathbf{A})^n$  as

$$(\mathbf{Z}\mathbf{A})^n = \underbrace{\mathbf{A}^{(1)}\mathbf{A}^{(2)}\dots\mathbf{A}^{(n)}}_{\mathbf{A}^{[n]}} \mathbf{Z}^n = \mathbf{A}^{[n]}\mathbf{Z}^n$$

Finally, after all these notional modifications we can write out the series expansion

$$(\mathbf{1} - \mathbf{Z}\mathbf{A})^{-1} = \mathbf{1} + \mathbf{A}^{[1]}\mathbf{Z}^1 + \mathbf{A}^{[2]}\mathbf{Z}^2 + \mathbf{A}^{[3]}\mathbf{Z}^3 + \dots, \quad (7)$$

which looks structurally very similar to the conventional form of the geometric series, in spite of the elements in the series being block-diagonal matrices.

In the following we will use this modified von Neuman expansion (7) to further move on with producing the Toeplitz operator  $\mathbf{T}$ . The first term in the expansion then looks like

$$\begin{aligned} \mathbf{A}^{[1]} \mathbf{Z}^1 &= \mathbf{A}^{(1)} \mathbf{Z}^1 = \\ &= \begin{bmatrix} \ddots & & & & \\ & A_{-1} & & & \\ & & \boxed{A_0} & & \\ & & & A_1 & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 0 & & & \\ & \mathbf{1} & \boxed{0} & & \\ & & \mathbf{1} & 0 & \\ & & & \mathbf{1} & \ddots \end{bmatrix} = \begin{bmatrix} \ddots & & & & \\ & 0 & & & \\ & A_{-1} & 0 & & \\ & & A_0 & \boxed{0} & \\ & & & A_1 & 0 \\ & & & & \ddots \end{bmatrix}. \end{aligned}$$

Following this recipe we can write up the second term of the series expansion more explicitly as

$$\begin{aligned} \mathbf{A}^{[2]} \mathbf{Z}^2 &= \mathbf{A}^{(1)} \mathbf{A}^{(2)} \mathbf{Z}^2 = \\ &= \begin{bmatrix} \ddots & & & & \\ & A_{-1} & & & \\ & & \boxed{A_0} & & \\ & & & A_1 & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & A_{-2} & & & \\ & & \boxed{A_{-1}} & & \\ & & & A_0 & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 0 & & & \\ & 0 & \boxed{0} & & \\ & \mathbf{1} & 0 & 0 & \\ & & \mathbf{1} & 0 & \ddots \end{bmatrix} \\ &= \begin{bmatrix} \ddots & & & & \\ & A_{-1}A_{-2} & & & \\ & & \boxed{A_0A_{-1}} & & \\ & & & A_1A_0 & \\ & & & & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & & & & \\ & 0 & & & \\ & 0 & \boxed{0} & & \\ & \mathbf{1} & 0 & 0 & \\ & & \mathbf{1} & 0 & \ddots \end{bmatrix} = \\ &= \begin{bmatrix} \ddots & & & & \\ & A_{-1}A_{-2} & 0 & & \\ & & 0 & 0 & \\ & & A_0A_{-1} & 0 & \boxed{0} \\ & & & A_1A_0 & 0 \\ & & & & \ddots \end{bmatrix}. \end{aligned}$$

Adding up the first three terms of the series expansion produces the lower-triangular tri-diagonal matrix

$$\mathbf{1} + \mathbf{A}^{[1]} \mathbf{Z}^1 + \mathbf{A}^{[2]} \mathbf{Z}^2 = \begin{bmatrix} \ddots & & & & & \\ & \ddots & & & & \\ & & \boxed{\mathbf{1}} & & & \\ & & A_1 & \mathbf{1} & & \\ & & A_2 A_1 & A_2 & \mathbf{1} & \\ & & & A_3 A_2 & A_3 & \mathbf{1} \\ & & & & A_4 A_3 & A_4 & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix}.$$

Just to make the construction principle more visible I add the forth term of the series to produce the matrix

$$\mathbf{1} + \mathbf{A}^{[1]} \mathbf{Z}^1 + \mathbf{A}^{[2]} \mathbf{Z}^2 + \mathbf{A}^{[3]} \mathbf{Z}^3 = \begin{bmatrix} \ddots & & & & & \\ & \ddots & & & & \\ & & \boxed{\mathbf{1}} & & & \\ & & A_1 & \mathbf{1} & & \\ & & A_2 A_1 & A_2 & \mathbf{1} & \\ & & A_3 A_2 A_1 & A_3 A_2 & A_3 & \mathbf{1} \\ & & 0 & A_4 A_3 A_2 & A_4 A_3 & A_4 & \mathbf{1} \\ & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Taking this intermediate result we can finally generate the Toeplitz operator according to (4)

$$\begin{bmatrix} \ddots & & & \\ & C_0 & & \\ & & \boxed{C_1} & \\ & & & C_2 & \ddots \end{bmatrix} \begin{bmatrix} \ddots & & & & & \\ & \ddots & & & & \\ & & \boxed{\mathbf{1}} & & & \\ & & A_1 & \mathbf{1} & & \\ & & A_2 A_1 & A_2 & \mathbf{1} & \\ & & A_3 A_2 A_1 & A_3 A_2 & A_3 & \mathbf{1} \\ & & A_4 A_3 A_2 A_1 & A_4 A_3 A_2 & A_4 A_3 & A_4 & \mathbf{1} \\ & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \ddots & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & B_{-1} & \boxed{0} & & \\ & & & B_0 & 0 & \\ & & & & B_1 & 0 & \\ & & & & & \ddots & \ddots \end{bmatrix}.$$

## Example: Finite Matrix

We now want to use a simple example to demonstrate the inner workings of the state-space computations. To this end, let us consider the example of the finite dimensional  $4 \times 4$  (non-Toeplitz) matrix

$$\mathbf{T} = \begin{bmatrix} 1 & & & \\ 1/2 & 1 & & \\ 1/6 & 1/3 & 1 & \\ 1/24 & 1/12 & 1/4 & 1 \end{bmatrix}, \quad (8)$$

which we will interpret as a transfer operator of a simple time-varying system. The matrix corresponds to a time-varying system since the matrix does not exhibit Toeplitz structure and because the matrix is

finite dimensional. Furthermore,  $\mathbf{T}$  is lower-triangular, indicating the system to be causal. Here  $\mathbf{T}$  is a  $4 \times 4$  matrix, however, real systems of interest are significantly larger, typical of the order  $n \sim 10,000$ . In the following discussion the term *signal* is used interchangeably with *vector*, likewise the term *system* is used for *matrix*. Similar to our discussion about time-invariant systems, we strive to represent the matrix  $\mathbf{T}$  in terms of the linear fractional map

$$\mathbf{T} = \mathbf{D} + \mathbf{C}(\mathbf{I} - \mathbf{Z}\mathbf{A})^{-1} \mathbf{Z}\mathbf{B}.$$

The target of state-space realization for these problems is to determine *efficient computations*, where the efficiency of the computations is measured in terms of the number of multiplications (arithmetic cost) and in terms of the number of registers (delays), which denotes the memory cost of a computation. As discussed earlier, truly time-invariant systems are associated with matrices, which

1. are infinite dimensional
2. exhibit Toeplitz structure.

Checking Equation 8 reveals that the matrix  $\mathbf{T}$  neither has Toeplitz structure nor is it infinite dimensional. Hence it must correspond to a linear time-varying system. However, we can still interpret the columns of  $\mathbf{T}$  as time-varying impulse responses.

## Direct Realization

We proceed to draw a direct realization for the given matrix  $\mathbf{T}$  as a time-varying system. As a start we redraw the state-space realization for one time step  $k$  as is shown in Figure 2. The signal flow graph on

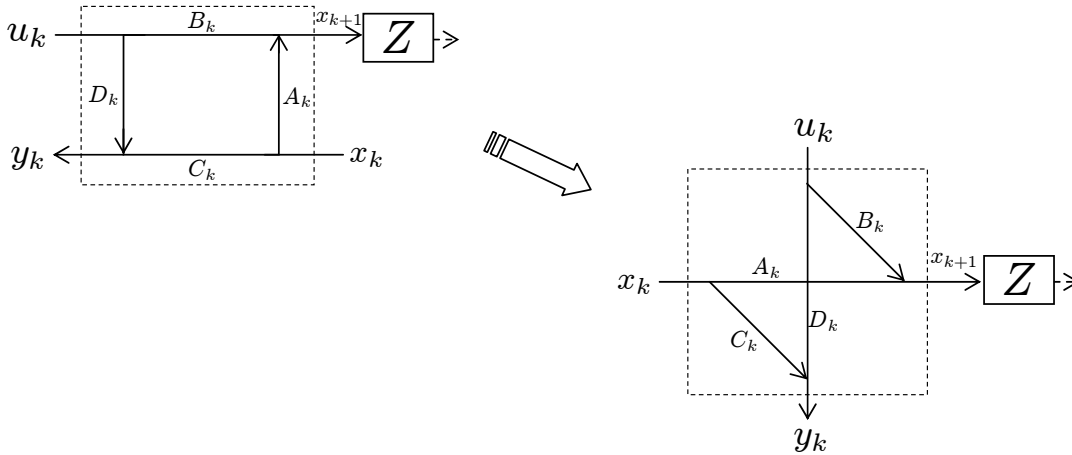
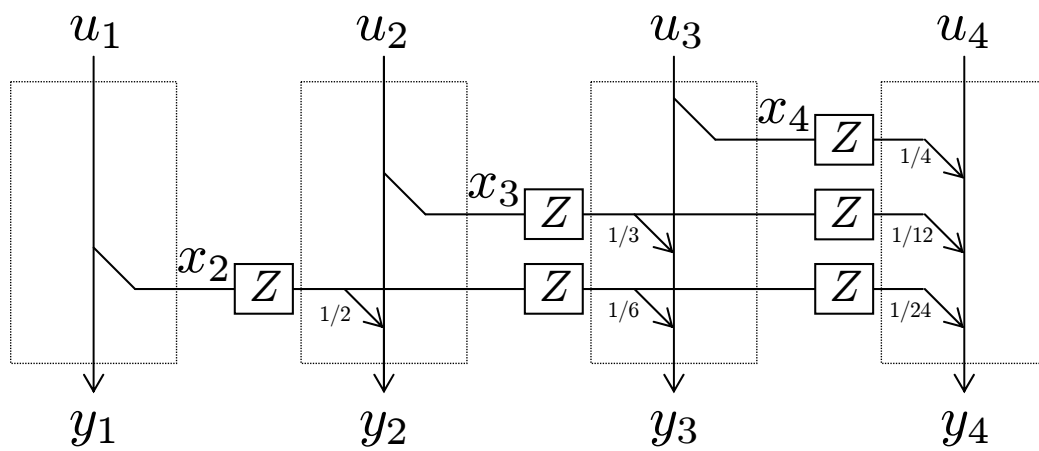


Abbildung 2: Redrawing and simplification

right-hand side of Figure 2 is the basic building block of a time-varying system. Based on this building-block, the system that realizes the transfer function in Equation 8 is shown in Figure 3. The signal flow inherently depicts causality by its unidirectional signal flow, i.e. all the arrows strictly point from top to bottom and from left to right. Other properties that can be observed are that this simple system uses 6 registers and 6 (non-trivial) multipliers for the realization of our given matrix  $\mathbf{T}$  (adders are typically not accounted for such complexity estimates). We denote the realization matrix for an individual block

Abbildung 3: Direct implementation of the matrix  $T$  in Equation 8



at time-index  $k$  by

$$\Sigma_k = \left[ \begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array} \right]. \quad (9)$$

For the direct realization shown in Figure 3 we can write down the individual realization matrices  $\Sigma_k$  by inspection as

$$\Sigma_1 = \left[ \begin{array}{c|c} \cdot & 1 \\ \hline \cdot & 1 \end{array} \right] \quad (10)$$

$$\Sigma_2 = \left[ \begin{array}{c|c} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \hline 1/2 & 1 \end{array} \right] \quad (11)$$

$$\Sigma_3 = \left[ \begin{array}{c|c} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \hline \begin{bmatrix} 1/6 & 1/3 \end{bmatrix} & 1 \end{array} \right] \quad (12)$$

$$\Sigma_4 = \left[ \begin{array}{c|c} \cdot & \cdot \\ \hline \begin{bmatrix} 1/24 & 1/12 & 1/4 \end{bmatrix} & 1 \end{array} \right], \quad (13)$$

which we can combine to specify the time-varying realization matrix

$$\Sigma = \left[ \begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right]$$

where the corresponding block-diagonal matrices are given as

$$\begin{aligned} \mathbf{A} &= \left[ \begin{array}{cccc} [A_1] & & & \\ & [A_2] & & \\ & & [A_3] & \\ & & & [A_4] \end{array} \right] = \left[ \begin{array}{cccc} [\cdot] & & & \\ & \begin{bmatrix} 1 \\ 0 \end{bmatrix} & & \\ & & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} & \\ & & & [\cdot] \end{array} \right] \\ \mathbf{B} &= \left[ \begin{array}{cccc} [B_1] & & & \\ & [B_2] & & \\ & & [B_3] & \\ & & & [B_4] \end{array} \right] = \left[ \begin{array}{cccc} [1] & & & \\ & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & & \\ & & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & \\ & & & [\cdot] \end{array} \right] \\ \mathbf{C} &= \left[ \begin{array}{cccc} [C_1] & & & \\ & [C_2] & & \\ & & [C_3] & \\ & & & [C_4] \end{array} \right] = \left[ \begin{array}{cccc} [\cdot] & & & \\ & [1/2] & & \\ & & \begin{bmatrix} 1/6 & 1/3 \end{bmatrix} & \\ & & & \begin{bmatrix} 1/24 & 1/12 & 1/4 \end{bmatrix} \end{array} \right] \\ \mathbf{D} &= \left[ \begin{array}{cccc} [D_1] & & & \\ & [D_2] & & \\ & & [D_3] & \\ & & & [D_4] \end{array} \right] = \left[ \begin{array}{cccc} [1] & & & \\ & [1] & & \\ & & [1] & \\ & & & [1] \end{array} \right]. \end{aligned}$$

Here and in the following a ' $[\cdot]$ ' as an entry represents a zero-dimensional matrix.

### Alternative Realization

Theoretically speaking an infinite number of realizations are possible for a given transfer operator  $\mathbf{T}$ . In reality only a few alternates will be of interest. For example, we consider the implementation as shown in Figure 4 that realizes the same transfer function as in Equation 8, but enables this with 3 registers (half the number as used by the straight-forward realization of Figure 3) and 5 multiplications. Although the simplification might not look significant at the moment, its worth considering that for large systems with  $n \sim 10,000$  and given that  $r \ll n$ , the complexity of implementation could be reduced to  $\mathcal{O}(8 \cdot r \cdot n)$  compared to  $\mathcal{O}((1/2)n^2)$  for a straight-forward implementation. However, the question is how can we

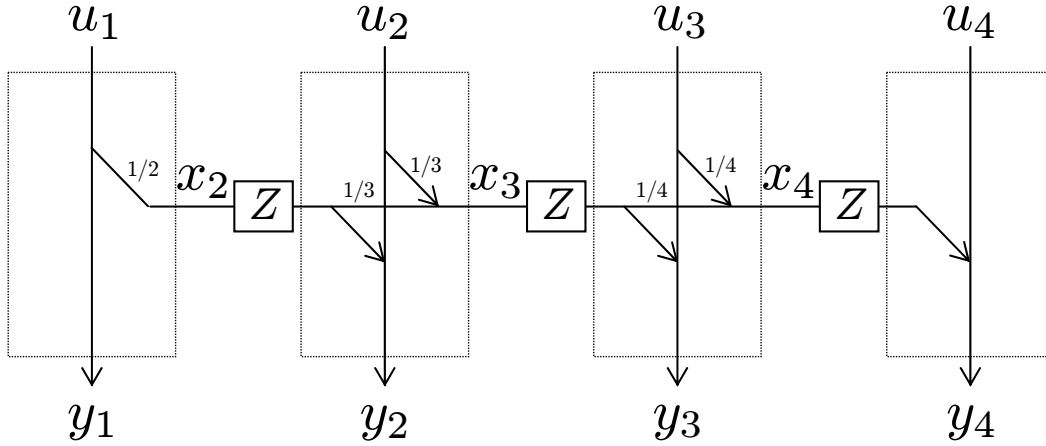


Abbildung 4: Simplified implementation of the matrix  $\mathbf{T}$  in Equation 8

tell if implementation complexity reduction is possible for a given system, and how to determine the minimal-complexity implementation? To answer this, we take a closer look at the implementations. For the system in Figure 4 we can read off the individual realization matrices

$$\Sigma'_1 = \left[ \begin{array}{c|c} \cdot & 1/2 \\ \cdot & 1 \end{array} \right] \quad \Sigma'_2 = \left[ \begin{array}{c|c} 1/3 & 1/3 \\ 1 & 1 \end{array} \right] \quad \Sigma'_3 = \left[ \begin{array}{c|c} 1/4 & 1/4 \\ 1 & 1 \end{array} \right] \quad \Sigma'_4 = \left[ \begin{array}{c|c} \cdot & \cdot \\ 1 & 1 \end{array} \right], \quad (14)$$

which we can decompose and reassemble to form the time-varying realization matrix

$$\Sigma' = \left[ \begin{array}{c|c} \mathbf{A}' & \mathbf{B}' \\ \hline \mathbf{C}' & \mathbf{D}' \end{array} \right]$$

with the corresponding block-diagonal matrices

$$\mathbf{A}' = \left[ \begin{array}{cccc} [A'_1] & & & \\ & [A'_2] & & \\ & & [A'_3] & \\ & & & [A'_4] \end{array} \right] = \left[ \begin{array}{cccc} [\cdot] & & & \\ & [1/3] & & \\ & & [1/4] & \\ & & & [\cdot] \end{array} \right],$$

$$\mathbf{B}' = \left[ \begin{array}{cccc} [B'_1] & & & \\ & [B'_2] & & \\ & & [B'_3] & \\ & & & [B'_4] \end{array} \right] = \left[ \begin{array}{cccc} [1/2] & & & \\ & [1/3] & & \\ & & [1/4] & \\ & & & [\cdot] \end{array} \right],$$

$$\mathbf{C}' = \begin{bmatrix} [C'_1] & & & \\ & [C'_2] & & \\ & & [C'_3] & \\ & & & [C'_4] \end{bmatrix} = \begin{bmatrix} [\cdot] & & & \\ & [1] & & \\ & & [1] & \\ & & & [1] \end{bmatrix},$$

$$\mathbf{D}' = \begin{bmatrix} [D'_1] & & & \\ & [D'_2] & & \\ & & [D'_3] & \\ & & & [D'_4] \end{bmatrix} = \begin{bmatrix} [1] & & & \\ & [1] & & \\ & & [1] & \\ & & & [1] \end{bmatrix}.$$

### Realization of Inverse Operator

Based on the realization given in the previous section, we can determine a realization  $\mathbf{\Gamma}$ , which implements the inverse operator  $\mathbf{T}^{-1}$ . We write the realization of the inverse operator as

$$\mathbf{\Gamma} = \left[ \begin{array}{c|c} \hat{\mathbf{A}} & \hat{\mathbf{B}} \\ \hline \hat{\mathbf{C}} & \hat{\mathbf{D}} \end{array} \right].$$

We can determine the individual realization matrices  $\Gamma_k$  as

$$\Gamma_k = \left[ \begin{array}{c|c} \hat{A}_k & \hat{B}_k \\ \hline \hat{C}_k & \hat{D}_k \end{array} \right] = \left[ \begin{array}{c|c} A_k - B_k \cdot D_k^{-1} \cdot C_k & B_k \cdot D_k^{-1} \\ \hline -D_k^{-1} \cdot C_k & D_k^{-1} \end{array} \right]. \quad (15)$$

For the in system depicted in Equation 14, the inverse realization can be done by using Equation 15, and is given as

$$\Gamma_1 = \left[ \begin{array}{c|c} \cdot & 1/2 \\ \hline \cdot & 1 \end{array} \right] \quad \Gamma_2 = \left[ \begin{array}{c|c} 0 & 1/3 \\ \hline -1 & 1 \end{array} \right] \quad \Gamma_3 = \left[ \begin{array}{c|c} 0 & 1/4 \\ \hline -1 & 1 \end{array} \right] \quad \Gamma_4 = \left[ \begin{array}{c|c} \cdot & \cdot \\ \hline -1 & 1 \end{array} \right] \quad (16)$$

$$\hat{\mathbf{A}} = \begin{bmatrix} [\hat{A}_1] & & & \\ & [\hat{A}_2] & & \\ & & [\hat{A}_3] & \\ & & & [\hat{A}_4] \end{bmatrix} = \begin{bmatrix} [\cdot] & & & \\ & [1/3] & & \\ & & [1/4] & \\ & & & [\cdot] \end{bmatrix}$$

$$\hat{\mathbf{B}} = \begin{bmatrix} [\hat{B}_1] & & & \\ & [\hat{B}_2] & & \\ & & [\hat{B}_3] & \\ & & & [\hat{B}_4] \end{bmatrix} = \begin{bmatrix} [1/2] & & & \\ & [1/3] & & \\ & & [1/4] & \\ & & & [\cdot] \end{bmatrix}$$

$$\hat{\mathbf{C}} = \begin{bmatrix} [\hat{C}_1] & & & \\ & [\hat{C}_2] & & \\ & & [\hat{C}_3] & \\ & & & [\hat{C}_4] \end{bmatrix} = \begin{bmatrix} [\cdot] & & & \\ & [1] & & \\ & & [1] & \\ & & & [1] \end{bmatrix}$$

$$\hat{\mathbf{D}} = \begin{bmatrix} [\hat{D}_1] & & & \\ & [\hat{D}_2] & & \\ & & [\hat{D}_3] & \\ & & & [\hat{D}_4] \end{bmatrix} = \begin{bmatrix} [1] & & & \\ & [1] & & \\ & & [1] & \\ & & & [1] \end{bmatrix}$$

The realization is depicted in Figure 5. The inverse transfer function can be read-off as

$$\mathbf{T}^{-1} = \hat{\mathbf{D}} - \hat{\mathbf{C}} \left( \mathbf{I} - \mathbf{Z} \hat{\mathbf{A}} \right)^{-1} \mathbf{Z} \hat{\mathbf{B}} = \begin{bmatrix} 1 & & \mathbf{0} \\ -1/2 & 1 & \\ 0 & -1/3 & 1 \\ 0 & 0 & -1/4 & 1 \end{bmatrix}. \quad (17)$$

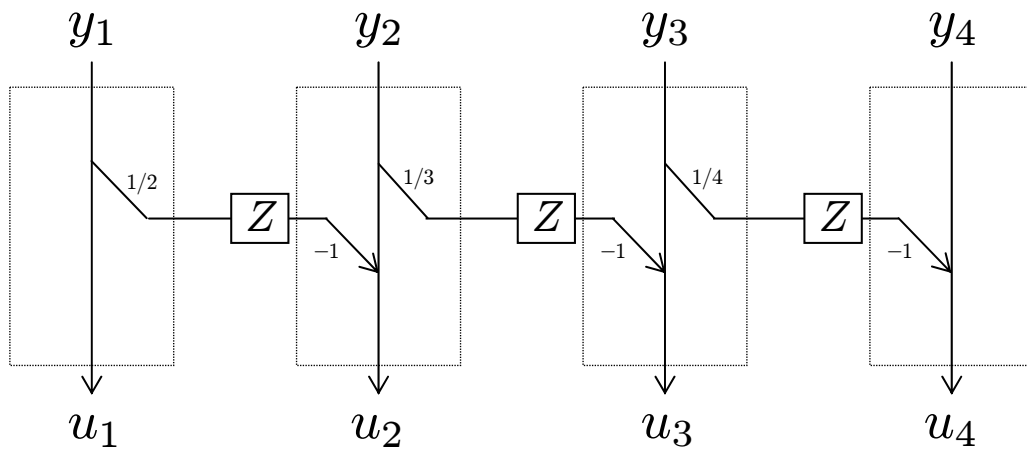


Abbildung 5: Inverse realization of the system

## Efficient Methods for Solving Linear Systems of Equations

In a more general setting we consider the problem of solving the set of linear equations  $T \cdot u = y$ , for the variable  $u$  where here the matrix  $T$  is not assumed to have any special structure. The variable  $u$  takes on the interpretation of an input signal applied to the input of a linear time-variant system, which we describe by means of the matrix  $T$ . The signal  $y$  is the corresponding output signal computed by a linear matrix-vector multiplication. In many technical situations we may need to invert this map. We can solve the inversion problem by determining the inverse of  $T$ , which we may compute using standard linear algebra tools such as Gaussian elimination or similar. However, for a general matrix  $T$  this inversion process costs  $\mathcal{O}(n^3)$  operations, if  $n$  denotes the size of the matrix  $n$ .

Similar to the strategy of using Fast Fourier techniques we now venture on a de-tour to arrive at a more efficient computational scheme. Figure 6 illustrates this de-tour. In a first step, we compute the parameters  $\{A, B, C, D\}$  of a time-varying state-space realization for the matrix  $T$ . This means that we express the matrix  $T$  in terms of those parameters as

$$T = D + C(1 - ZA)^{-1} ZB.$$

Once we have such a state-space realization for  $T$  we can compute the realization of the inverse system, symbolically represented as  $\{A, B, C, D\}^{-1}$  with  $\mathcal{O}(n^2d)$  operations. The parameter  $d$  denotes the dynamical (Smith-McMillan) degree of the system. Filtering the signal  $y$  with the inverse state-space system  $\{A, B, C, D\}^{-1}$  produces the solution  $u$ .

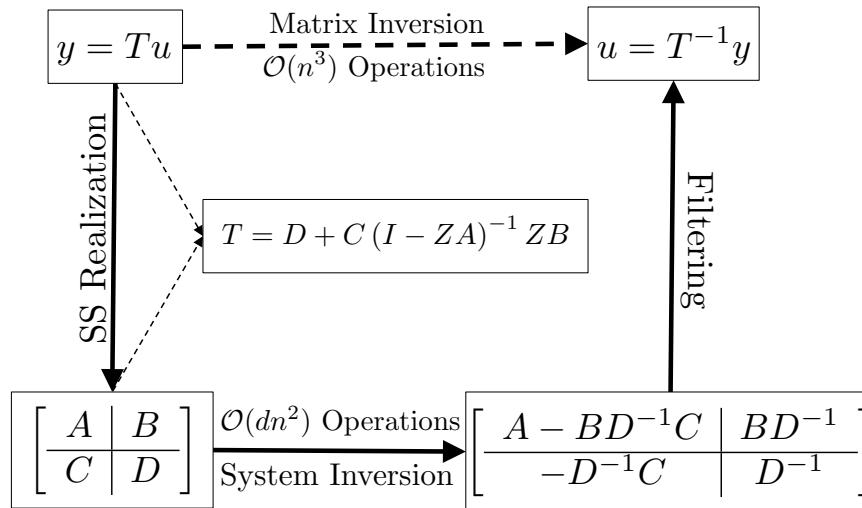


Abbildung 6: Schematic Procedure for Computing the Inverse of an Operator.

If  $d = n$  the detour for solving the linear system does not give us any reduction in computing the inverse. However, whenever  $d \ll n$  holds, then we achieve a reduction of the computational complexity by approximately one order of magnitude. We will study the conditions for a matrix  $T$  to correspond with a system that satisfies  $d \ll n$ . In situations where we get  $d \approx n$  we can employ model order reduction techniques, which are well known in system theory, to determine an approximate inverse of  $T$  or a solution  $\hat{u}$  which is 'close' to the exact solution. Model reduction means that the original system with degree  $d$  is approximated with a system that has degree  $\hat{d} < d$ . The notion of 'Close' is measured in terms of a strong norm, the Hankel norm. This allows us to determine an approximate solution  $\hat{u}$  that lies within a prescribed error bound. In many technical problems it is sufficient to determine an approximate solution that is 'good enough' and can be computed cheaply, while an exact solution does not give much benefit for the application, but requires a substantially more resources to compute. Here, often we work according to the motto that it is better to be approximately right than to be exactly wrong.

### State Transformation

We can apply a linear and non-singular transformation of the state-space in the form of

$$x_k = R_k x'_k, \quad \det R_k \neq 0, \forall k$$

to the system as shown in Figure 7. The resulting system of equations then is

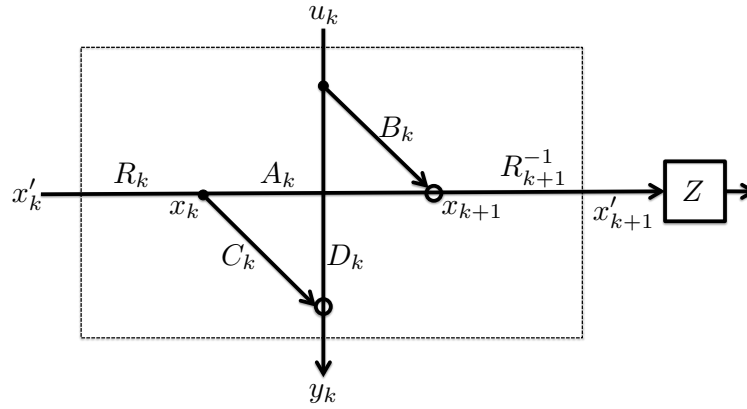


Abbildung 7: State transformation

$$\begin{aligned} R_{k+1} x'_{k+1} &= A_k \cdot (R_k \cdot x'_k) + B_k \cdot u_k \\ y_k &= C_k \cdot (R_k \cdot x'_k) + D_k \cdot u_k, \end{aligned} \tag{18}$$

which we can summarize by the matrix expression

$$\begin{bmatrix} x'_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} R_{k+1}^{-1} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \cdot \begin{bmatrix} R_k & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} x'_k \\ u_k \end{bmatrix}, \tag{19}$$

giving rise to the transformed realization matrix

$$\Sigma' = \begin{bmatrix} R_{k+1}^{-1} A_k R_k & R_{k+1}^{-1} B_k \\ C_k R_k & D_k \end{bmatrix}.$$

Block-diagonal expansion of the sequence of state-transformation matrices produces the block-diagonal matrix

$$\mathbf{R} = \begin{bmatrix} \ddots & & \\ & \boxed{R_k} & \\ & & \ddots \end{bmatrix},$$

which we will use in connection with the block-diagonal expansion of the state-space realization matrices, to achieve a sequence of elementary manipulations of the state-equation

$$\begin{aligned} \mathbf{Z}^{-1} \mathbf{R} \cdot \mathbf{x}' &= \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{x}' + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{Z}^{-1} \mathbf{R} \mathbf{Z} \mathbf{Z}^{-1} \cdot \mathbf{x}' &= \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{x}' + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{R}^{[-1]} \mathbf{Z}^{-1} \cdot \mathbf{x}' &= \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{x}' + \mathbf{B} \cdot \mathbf{u} \\ \mathbf{Z}^{-1} \cdot \mathbf{x}' &= \left( \mathbf{R}^{[-1]} \right)^{-1} \mathbf{A} \cdot \mathbf{R} \cdot \mathbf{x}' + \left( \mathbf{R}^{(-1)} \right)^{-1} \mathbf{B} \cdot \mathbf{u} \end{aligned}$$

while we write the output signal equation as

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{R} \cdot \mathbf{x}' + \mathbf{D} \cdot \mathbf{u}.$$

Here, the symbol  $\mathbf{R}^{[-1]}$  represents the block-diagonal matrix  $\mathbf{R}$  shifted diagonally one block in the north-west direction, i.e. we have

$$\mathbf{R}^{[-1]} = \mathbf{Z}^{-1} \mathbf{R} \mathbf{Z} = \begin{bmatrix} \ddots & & \\ & \boxed{R_{k+1}} & \\ & & \ddots \end{bmatrix}.$$

In matrix form we can summarize this state-transformation as

$$\begin{bmatrix} \mathbf{Z}^{-1} \cdot \mathbf{x}' \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \left( \mathbf{R}^{[-1]} \right)^{-1} \\ \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R} & \\ & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}' \\ \mathbf{u} \end{bmatrix}. \quad (20)$$

The state-transformation modifies the state-space realization matrices according to

$$\begin{bmatrix} \mathbf{A}' & \mathbf{B}' \\ \mathbf{C}' & \mathbf{D}' \end{bmatrix} = \begin{bmatrix} \left( \mathbf{R}^{[-1]} \right)^{-1} \mathbf{A} \mathbf{R} & \left( \mathbf{R}^{[-1]} \right)^{-1} \mathbf{B} \\ \mathbf{C} \mathbf{R} & \mathbf{D} \end{bmatrix}.$$

## Invariance of Transfer Operator

We check the effect of the state transformation on the transfer operator. To this end we determine the transfer operator using the transformed state-space realization according to

$$\begin{aligned}
 \hat{T} &= \hat{D} + \hat{C} \left( \mathbf{1} - \mathbf{Z} \hat{\mathbf{A}} \right)^{-1} \mathbf{Z} \hat{B} \\
 &= D + CR \left( \mathbf{1} - \mathbf{Z} \left( R^{[-1]} \right)^{-1} AR \right)^{-1} \mathbf{Z} \left( R^{[-1]} \right)^{-1} B \\
 &= D + CR \left( \mathbf{1} - \mathbf{Z} (\mathbf{Z}^{-1} R \mathbf{Z})^{-1} AR \right)^{-1} \mathbf{Z} (\mathbf{Z}^{-1} R \mathbf{Z})^{-1} B \\
 &= D + CR \left( \mathbf{1} - R^{-1} Z A R \right)^{-1} R^{-1} Z B \\
 &= D + C \left( \mathbf{1} - Z A \right)^{-1} Z B \\
 &= T.
 \end{aligned} \tag{21}$$

This result indicates that the state-transformation changes the realization matrix, but it leaves the transfer operator invariant. Expressed in other words this means that we have an infinite number of realizations  $\Sigma$  for one given transfer operator  $T$ . The set of all realizations is parameterized in terms of the non-singular state-transformations  $R$ , which gives us the opportunity to search in the space of realizations for a particular one, which optimizes a given cost function. We can use an appropriately chosen cost function to find a realization that minimizes the coefficient sensitivity, or the effect of rounding errors (noise optimization) or also to minimize the number of arithmetic operations or any other technically interesting characteristic. Hence, we can employ numerical optimization techniques to support this search for a particular realization. The mapping of a given transfer operator onto a corresponding realization is a one-to-many mapping. The realization theory for linear state-space system addresses this problem. Notice, that the map of a given state-space realization to the corresponding transfer operator is unique.

## Literatur

- [1] G. Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.
- [2] P.Regalia. *Adaptive IIR Filtering in Signal Processing and Control*. CRC Press, 1995.
- [3] T.Kailath. *Linear Systems*. Prentice Hall, 1980.
- [4] P. Dewilde, A.-J. van der Veen. *Time-Varying Systems and Computations*. Kluwer Academic Publishers, 1998.