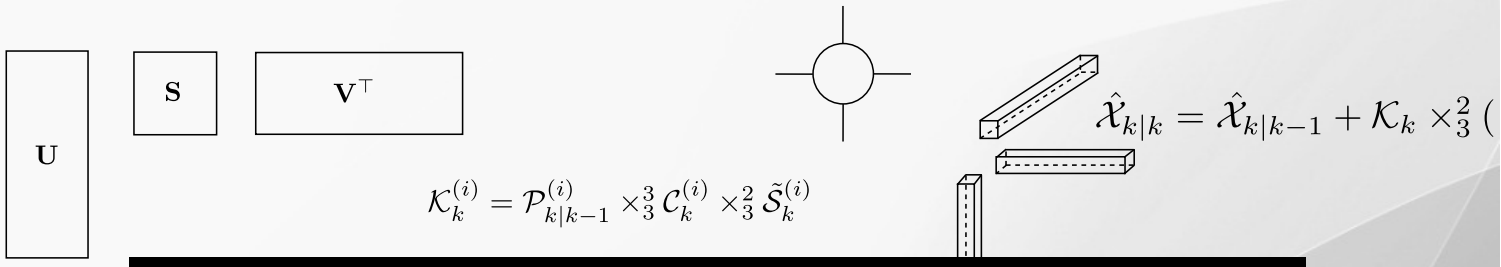
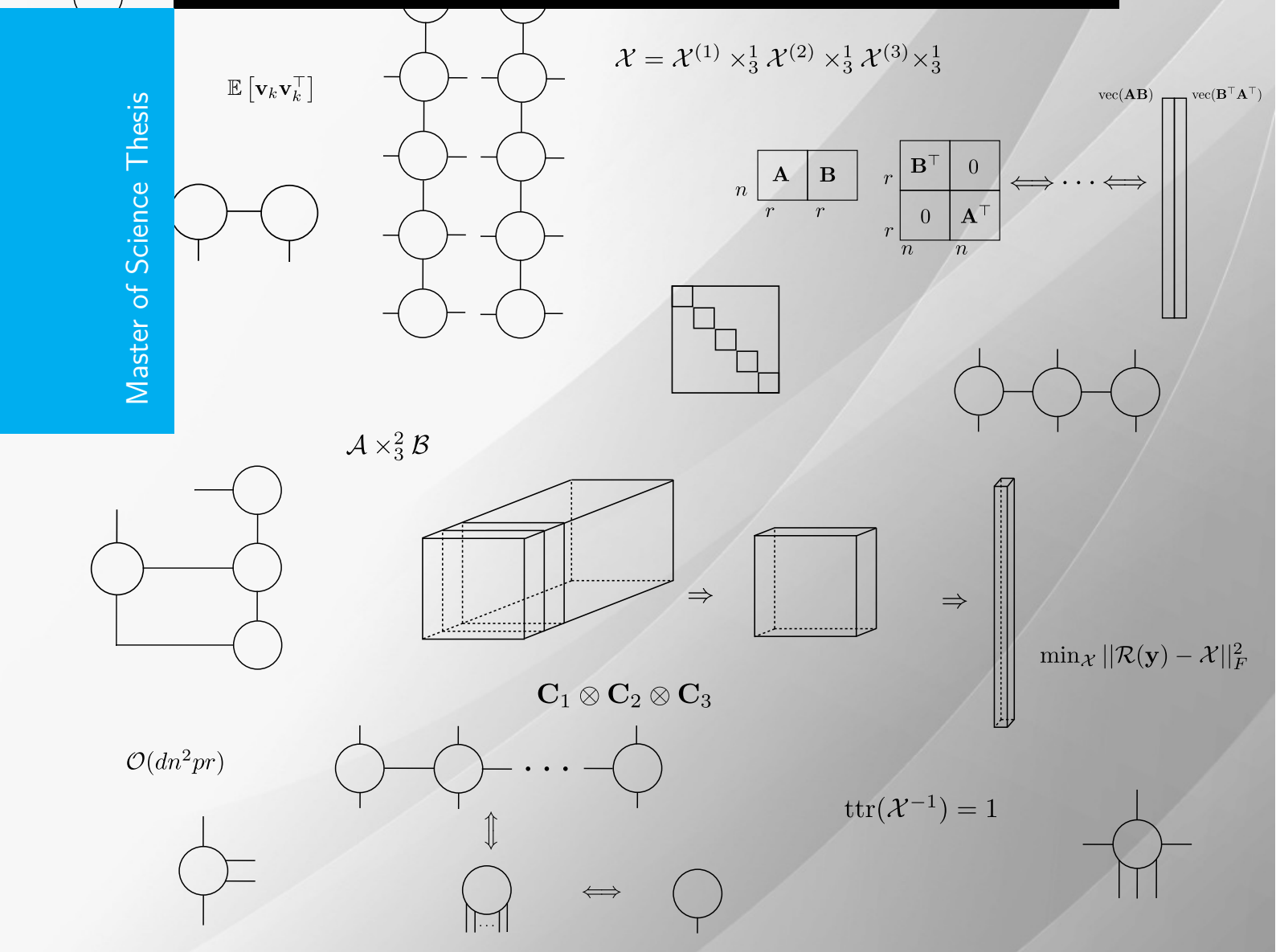


Tensor Network Kalman Filter for Large-Scale MIMO Systems

With Application to Adaptive Optics

Daniel Gedon

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Tensor Network Kalman Filter for Large-Scale MIMO Systems

With Application to Adaptive Optics

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Daniel Gedon

June 21, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

The recursive Kalman filter as unbiased and minimum variance stochastic linear estimator has been applied to a huge range of real-time applications and engineering domains since its development in 1960. However, in practice the theory is mostly limited to small-scale problems. The limitation is due to the necessary and computationally demanding solution of the associated discrete algebraic Riccati equation. Many engineering problems nowadays have a large-scale character with many states and outputs in a representative state-space model. Challenges for example in the field of adaptive optics, (power) networks or traffic systems could benefit from large-scale real-time Kalman filtering to improve the performance of control algorithms.

A possible mathematical framework to lift the curse of dimensionality is to lift the problem in higher dimensions with the use of tensors and then decompose it. The tensor-train decomposition is an often chosen tool which can yield in combination with multilinear tensor algebra a great decrease in computational complexity. An example for models which can be used computationally effective in this framework are models consisting of sums of Kronecker products.

A tensor Kalman filter for systems with exponentially large state vectors has been developed in literature and been successfully applied to the system identification of nonlinear Volterra systems. All variables within this filter are in tensor-train format and use a higher order generalization of the matrix Kalman filter equation by means of multilinear algebra on single tensor cores. However, this theory has two main limitations. First, if the system dynamics in tensor-train format have the inherent property of high tensor-train ranks, the algorithmic operations and therefore the complete filter becomes considerably slow. Second, the developed theory does not provide a framework for Kalman filtering of Multiple-Input Multiple-Output (MIMO) systems with exponentially large output vectors.

This thesis tries to cover both issues independently. For the first limitation a method is developed to truncate the tensor-train rank of the covariances within the filter to lower values, which yields fast and still highly accurate estimations. Furthermore, a possible framework for a MIMO tensor Kalman filter based on specific properties of the system output equation and a low measurement noise assumption is presented.

Both methods combined have been tested on the application of adaptive optics. The next generation of telescopes requires adaptive optics control systems and real-time estimation algorithm for several tens of thousands of outputs and states at high sampling rates. The developed theory has been shown to hold a new design paradigm between a high gain in terms of computational time and a loss of accuracy.

With these methods the thesis provides a complete overview on tensor-train operations and a rigorous introduction for future research in this direction. Many unsolved problems remain such as a general fast inverse in tensor-train format or a square root tensor Kalman filter implementation as numerically required for many practical applications.

Table of Contents

Acknowledgements

This master thesis is the result of two years of work during the MSc. studies at TU Delft with many ups and downs, which could not have been successfully accomplished without the help and contribution of many people.

First of all, I would like to thank my supervisor prof.dr.ir. M. Verhaegen for the opportunity to tackle challenging problems in the field of large-scale systems and Kalman filters which led to this thesis. To question all of my approaches, leading me to new interesting directions and your enthusiasm about results made this thesis journey highly enjoyable.

Furthermore, I would like to thank my co-supervisor Dr.ir. K. Batselier. Not only for plenty of interesting and insightful discussions on tensor calculus and multilinear algebra, but also on all the small life-lessons and advices that you provided every time. Your feedback and constructive criticism led my work to a higher level. Also, Dr.ir. C. Smith provided many helpful thoughts and suggestions, always with an applications of the theory in mind, which I was often too enthusiastic about. Thank you for leading my focus in the right direction. Especially, I want to thank ir. P. Piscaer for each and every discussion and the insight on adaptive optics. Even if our discussions sometimes drifted apart from work, it would not have been possible to finish this thesis without your help. Finally, my gratitude goes to Dr.ir. B. Sinquin on whose work this thesis is based. Your interest in my work and your hints and ideas pushed this work significantly forward.

A huge role in this MSc. adventure played all of my friends - if here in Delft or far away. The time were not nearly as amazing and as enjoyable without all those laughter, discussions, long nights and stupid ideas that we shared together. I hope we keep these memories and experience more adventures in the future even if our ways drift apart.

Natürlich geht mein größter Dank an meine Familie, meine Mama, meinen Papa und meine Schwester. Ohne eure Unterstützung wäre es mir nicht möglich gewesen diese Zeit hier erfolgreich zu meistern. Danke für euer offenes Ohr in allen Angelegenheiten und die Freude, die ihr mir bereitet habt, wann immer ich es nach Hause geschafft habe.

Delft, University of Technology — Daniel Gedon
June 21, 2019

Chapter 1

Introduction

Filtering, state estimation and state propagation are crucial steps in the analysis and design of complex control systems. Engineering solutions ask for optimal estimation algorithms. The Kalman filter [18] yields an unbiased, minimum variance solution to this problem. The filter requires a linear model of the system. Within this thesis autonomous discrete-time state-space models are considered as visualized in Fig. 1-1 and given by

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{1-1}$$

with exponentially large state and output vector as $\mathbf{x} \in \mathbb{R}^{n^d}$, $\mathbf{y} \in \mathbb{R}^{p^d}$ and with Gaussian process and measurement noise $\mathbf{w}_k \sim (0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim (0, \mathbf{R}_k)$, respectively. The theory can straightforwardly be extended for non-autonomous system by including the input dynamics. Note that the general time-varying system description is chosen.

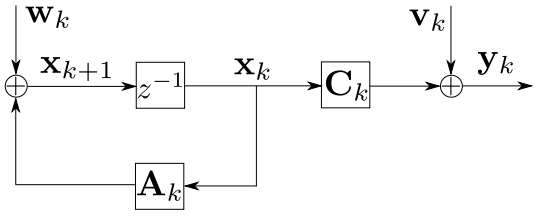


Figure 1-1: Block diagram of autonomous time-varying discrete-time state-space description as used for Kalman filtering

Nowadays the Kalman filter is the method of choice in multiple engineering problems and domains. It is successfully applied in the field of signal processing, robotics, instrumentation, sensorless control, i.e. in general industrial applications [1, 8] but also in the field of aerospace [13] or finance [2]. One feature that all these applications have in common is the small-scale property of the problem. This property is the inherent limitation of the Kalman filter. The basis of the filter builds the associated Discrete Algebraic Riccati Equation (DARE) [17]

which is recursively solved. Computationally effective solutions are only obtained for small-scale problems. Meaning a small number of states and outputs. There are two main problems for large-scale systems:

1. Storage of system matrices: If the system is exponentially large, the storage of the exponentially large system matrices may become prohibitive in the conventional matrix format.
2. Computational complexity: The recursive computation of the covariance matrices (filtered and predicted) requires an order of $\mathcal{O}\left(n^{3d}\right)$ algebraic operations and the inverse for the Kalman gain requires $\mathcal{O}\left(p^{3d}\right)$ for the exponentially large system. Even for small n, d this becomes easily infeasible to compute on-line in real-time.

The goal of this thesis is to design a Kalman filter for large-scale real-time problems. The method of choice are tensor decompositions to lift the curse of dimensionality. These decomposition allow to reduce the storage requirement and computational complexity for a certain type of system descriptions such that computational complexity becomes linear in d instead of exponential. The developed algorithm will be verified on the application of Adaptive Optics (AO). For the next generation of telescopes, like the European Extremely Large Telescope (E-ELT) with a primary mirror of about 40 [m] the number of measurements and states for estimation will lie in the order of several tens of thousands [12]. Hence, large-scale Kalman filters which are capable to estimate the states in real-time are required.

The main part of the thesis consists of two papers in chapter 4 and chapter 5, respectively. The papers are printed in this thesis as accepted to the EUSIPCO conference and as a first draft for the IEEE transactions on control systems technology journal. Both can be read standalone. Therefore, note that the numbering of the references within the papers does not necessarily match the ones within this thesis. In front of the printout of each paper, the main points are summarized and some editorial information is provided in the dedicated chapters. In order to simplify the understanding of the paper due to an extensive use of tensor calculus, supplementary material for the introduction to tensors in presented in chapter 2. This includes the mathematical background for tensor calculus, the used Tensor-Train (TT) decomposition and several important algorithms. The tensor introduction may go in some points beyond the used theory of this thesis but is beneficial for a reader to get familiar with the concepts and to understand tensor calculus clearer. An introduction to AO that is more elaborate and extensive than in the paper is given in chapter 3 such that the verification application within the second paper can be understood in great detail. The thesis is concluded in chapter 6. The main results are summarized and limitations are highlighted. This draws the way for future developments and improvements of the developed theory. Supplementary material of ideas to overcome the limitations are attached in Appendix A.

Chapter 2

Introduction to Tensors

2-1 General

2-1-1 Definition

In this thesis a tensor is considered to be a multidimensional array as an extension to scalars, vectors or matrices for higher orders. In a vector each entry has one index, in a matrix there are two indices and in an order-d tensor each entry has d indices. Scalars are denoted as Roman letters $x \in \mathbb{R}$, vectors as lower-case boldface letters $\mathbf{x} \in \mathbb{R}^{I_1}$, matrices as capital bold letters $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. Tensors of order-$d$ are given as capital calligraphic letters $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$. Note that also vectors or matrices can be written as order-1 or order-2 tensors. The size of the dimension indices goes $I_n = i_1, \ldots, i_N$. In a sequence of tensors, the ith element is denoted with a superscript in round letters, e.g. $\mathcal{X}^{(3)}$ is the third tensor in the sequence. Note that the word index and mode of a tensor are interchangeable. Elaborate and highly useful introductions to tensors are given in [9, 19]. In the following more necessary definitions are given.

Kronecker Product [38] Take two matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{B} \in \mathbb{R}^{J_1 \times J_2}$. Their Kronecker product is given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,i_2}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{i_1,1}\mathbf{B} & \cdots & a_{i_1,i_2}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2} \tag{2-1}$$

The Kronecker product is in multiple ways connected to tensor calculus. This is exploited in more detail in the following subsections.

Fiber [9] A fiber is the higher order equivalent to column or row vectors within a matrix. By fixing all indices but the nth index, the mode-n fiber is generated.

Vectorization [9] A rearrangement of all mode-1 fibers of an order-d tensor in a column vector is performed. The operation is denoted by $\text{vec}(\mathcal{X})$.

Matricification [9] A rearrangement of all mode-n fibers as column vectors of a matrix is performed to obtain a mode-n matricification. This operation is also called unfolding. The operation is denoted with subscripts in round brackets, for example the mode-2 matricification of a tensor $\mathcal{X}$ is given by $\mathbf{X}_{(2)}$. A visualization for the matricification with use of mode-n fibers is visualized for an order-3 tensor in Fig. 2-1.

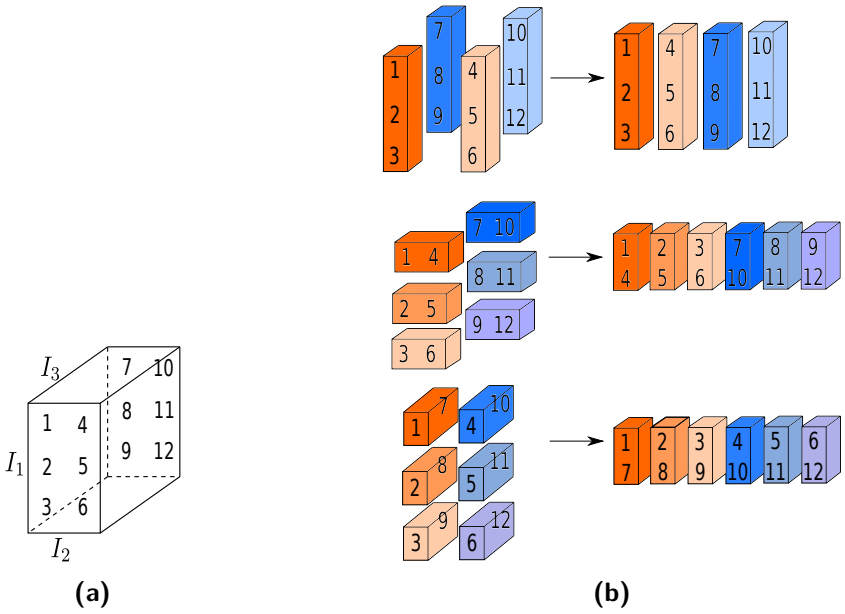


Figure 2-1: Tensor with fibers and different modes of matricization. (a) Representation of order-3 tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with $I_1 = 3$, $I_2 = 2$, $I_3 = 2$ in a cubical representation. (b) Top panel: mode-1 fibers and matricization $\mathbf{X}_{(1)}$. Middle panel: mode-2 fibers and matricization $\mathbf{X}_{(2)}$. Bottom panel: mode-3 fibers and matricization $\mathbf{X}_{(3)}$.

Tensorization [9] This is the reverse operation of the vectorization or matricification. It uses reshuffling and reshaping of the values and is given by Matlab notation $\text{reshape}(\mathbf{X}, [I_1, \ldots, I_d])$. Tensorization is not unique and specified for each operation.

Rank-1 Tensor [19] An order-d tensor $\mathcal{X}$ is said to be rank-1 if it can exactly be described by an outer product of d vectors, mathematically given by

$$\mathcal{X} = \mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(d)} \tag{2-2}$$

Tensor Rank [19] The tensor rank R is a generalization of the matrix rank to higher orders. It is the minimum number of summations R of rank-1 tensors such that the order-d tensor $\mathcal{X}$ is described exactly. This can be expressed as

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{x}_r^{(1)} \circ \cdots \circ \mathbf{x}_r^{(d)} \tag{2-3}$$

TN Diagram A useful way to visualize and intuitively understand higher order tensors is by means of Tensor Network (TN) diagrams. An introduction to TN theory and the visualization in TN diagrams is given in [25] but also in [9]. These diagrams visualize a single tensor core as circle with free lines connected to the core. The number of free lines indicates the order of the tensor. Each line is indexed, representing the size in that dimension of the tensor. An example for tensors of order-0 to order-3 is given in Fig. 2-2.

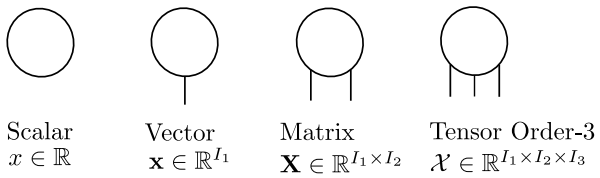


Figure 2-2: Example of a scalar x, a vector $\mathbf{x}$, a matrix $\mathbf{X}$ and an order-3 tensor $\mathcal{X}$ as TN diagram representation.

2-1-2 Basic Multilinear Tensor Operations

Multilinear operations have to be defined. They depict higher order equivalents to vector and matrix operations and are necessary in order to use higher order tensors mathematically.

Mode-n matrix product (contraction) [9, 19] This operation describes the higher order multiplication of an order-d tensor with a matrix. The matrix can be multiplied in any mode with the tensor, therefore the operator indicates which mode of the tensor is used. Take an order-d tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and a matrix $\mathbf{B} \in \mathbb{R}^{J_1 \times J_2}$ with common mode $I_n = J_2$. The mode-n product will yield an order-d tensor which is formally given by

$$\mathcal{C} = \mathcal{A} \times_n \mathbf{B} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_1 \times I_{n+1} \times \cdots \times I_d} \tag{2-4}$$

The operator denotes in its subscript the index or mode of contraction. Sizes in this mode and the one of the associated matrix have to match. The entries of the new tensor are then computed by element-wise summations as

$$c_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_d} = \sum_{i_n}^{I_n} a_{i_1,\ldots,i_n,\ldots,i_d} b_{j,i_n} \tag{2-5}$$

This equation yields that there are $\mathcal{O}(I_1 I_2 \ldots I_d)$ algebraic operations necessary for the computation of the tensor-matrix contraction.

Remark: The mode-n product can equivalently be written with mode-n matricifications as

$$\mathbf{C}_{(n)} = \mathbf{B}\mathbf{A}_{(n)} \tag{2-6}$$

Mode-m,n tensor product (tensor contraction) [9] This is a generalization of the mode-n matrix product for two tensors. Take two tensors of order-d $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and of order-k

$\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_k}$ with common mode $I_n = J_m$. The contraction yields an order-$(d + k - 2)$ tensor, formally given by

$$\mathcal{C} = \mathcal{A} \times_n^m \mathcal{B} \tag{2-7}$$

The operator denotes that the nth mode of tensor $\mathcal{A}$ is contracted with the mth mode of tensor $\mathcal{B}$. The entries of the new tensor $\mathcal{C}$ are computed element-wise by

$$c_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_d,j_1,\ldots,j_{m-1},j_{m+1}\ldots,j_k} = \sum_{i_n=1}^{I_n} a_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_d} b_{j_1,\ldots,j_{m-1},j_n,j_{m+1},\ldots,j_k} \tag{2-8}$$

With this the computational complexity can be written as

$$\mathcal{O}\left(I_n \cdot I_1 \ldots I_{n-1} I_{n+1} \ldots I_d \cdot J_1 \ldots J_{m-1} J_{m+1} \ldots J_k\right) \tag{2-9}$$

Note that the mode-m,n product is in general not associative or commutative [9]. In the following several application of the mode-m,n tensor product which are distinct matrix-vector operations for lower order are presented. This should yield a more intuitive understanding of the multilinear operations.

Example: Inner product Take two vectors $\mathbf{a} \in \mathbb{R}^{I_1}$ and $\mathbf{b} \in \mathbb{R}^{I_1}$ of the same size. The example is visualized in Fig. 2-3. The free indices of each vector can be visually connected since they are of the same size. After connection, the resulting tensor with two cores can be contracted to one single tensor core. This resulting core has zero free indices, i.e. a scalar. Mathematically the operation is given with multilinear operations by

$$c = \mathbf{a} \times_1^1 \mathbf{b} \tag{2-10}$$

yielding in matrix notation

$$c = \sum_{i_1=1}^{n_1} a_{i_1} b_{i_1} = \mathbf{a} \cdot \mathbf{b} \tag{2-11}$$

This operation requires a total of $\mathcal{O}\left(I_1\right)$ algebraic operations.

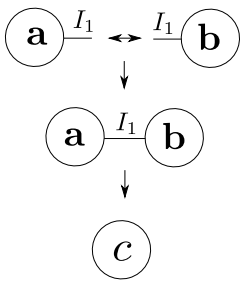


Figure 2-3: TN diagram of inner product of two vectors resulting in a scalar

Example: Outer product Take two vectors $\mathbf{a} \in \mathbb{R}^{I_1}$ and $\mathbf{b} \in \mathbb{R}^{I_2}$ as visualized in Fig. 2-4. However, in contrast to the inner product example the two cores cannot be connected since the indices have different sizes. A simple trick is to extend the tensor cores with singular additional indices resulting in $\mathbf{a} \in \mathbb{R}^{I_1 \times 1}, \mathbf{b} \in \mathbb{R}^{I_2 \times 1}$. These singular indices can be connected and in the next step contracted to a resulting matrix. In tensor notation this yields

$$\mathbf{C} = \mathbf{a} \times_2^2 \mathbf{b} \tag{2-12}$$

or in matrix notation

$$\mathbf{C} = \mathbf{a}\mathbf{b}^\top = \mathbf{a} \circ \mathbf{b} \tag{2-13}$$

or element-wise

$$c_{i_1,i_2} = \sum_{i=1}^{1} a_{i_1} b_{i_1} \tag{2-14}$$

The latter yields the number of algebraic operations as $\mathcal{O}(I_1 I_2)$ for the outer product of two vectors.

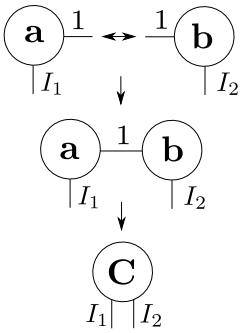


Figure 2-4: TN diagram of outer product of two vectors resulting in a matrix

Example: Tensor outer product Take two tensors of order-d $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and of order-k $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_k}$ without common mode. An extension with a singular index is possible. The contraction in this singular index is then given by

$$\mathcal{C} = \mathcal{A} \times_{d+1}^{k+1} \mathcal{B} \tag{2-15}$$

which yields an order-$(d+k)$ tensor. Note the similarity to the mode-m,n product in eq. (2-7). Hence, the tensor outer product and the tensor contraction operation are closely related.

Example: Matrix product Take two matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{B} \in \mathbb{R}^{I_2 \times I_3}$. Their common mode with size I_2 can be connected and contracted to a single core. This is visualized in Fig. 2-5 and mathematically given by

$$\mathbf{C} = \mathbf{A} \times_2^1 \mathbf{B} \tag{2-16}$$

or in matrix notation

$$\mathbf{C} = \mathbf{AB} = \mathbf{A} \circ \mathbf{B}^{\top} \tag{2-17}$$

or element-wise

$$c_{i_1,i_3} = \sum_{i_2=1}^{I_2} a_{i_1,i_2} b_{i_2,i_3} \tag{2-18}$$

This yields the number of algebraic operations as $\mathcal{O}(I_1 I_2 I_3)$ for the matrix product.

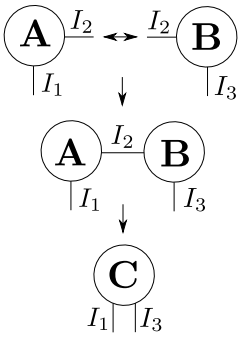


Figure 2-5: TN diagram of matrix product

As highlighted in all these examples the computational complexity of the operations by using the mode-m,n product is equal to the one in matrix notation. No computational advantage is obtained by simply using tensor notation and operations. However, special decompositions of the tensors can yield an advantage and are introduced in the following.

2-2 TT Decomposition

2-2-1 Definition

TT-Decomposition The Tensor-Train (TT) decomposition is introduced in [26] after previous work in [28, 31, 32]. A Matlab toolbox is available in [30]. The decomposition takes one order-d tensor and decomposes it in d order-3 tensor cores. These cores are connected in a line network. An example is given in Fig. 2-6. It shows the reshaping of a vector $\mathbf{a} \in \mathbb{R}^{n^3}$ in an order-3 tensor $\mathcal{A} \in \mathbb{R}^{n \times n \times n}$ and its decomposition in a TT where all TT-cores are of size $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$.

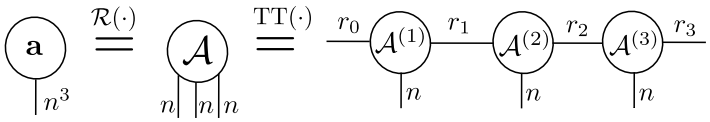


Figure 2-6: TN diagram of a vector on the left, reshaping to an order-d tensor in the middle and the result of its TT-decomposition on the right

The interconnecting indices r_i in mode-1 and mode-3 of each core are called TT-ranks, which play a crucial role in the computational complexity. The maximum TT-rank over all r_k

is defined as $r = \text{ttr}(\mathcal{X})$. Note that the boundary conditions for the TT-decomposition is $r_0 = r_d = 1$. Mathematically the TT-decomposition of a tensor is given with multilinear operations as

$$\mathcal{A} = \mathcal{A}^{(1)} \times_3^1 \cdots \times_3^1 \mathcal{A}^{(d)} \tag{2-19}$$

This means as demonstrated in Fig. 2-6 that mode-3 of tensor core i is connected and contracted with mode-1 of tensor core $i+1$.

TT-Matrix [32] The TT-decomposition can be extended to matrices. Take for example a matrix $\mathbf{A} \in \mathbb{R}^{n^3 \times n^3}$ which can be reshaped in an order-6 tensor $\mathcal{A} \in \mathbb{R}^{n \times \cdots \times n}$. This tensor can then be decomposed in a TT-matrix where each core has size $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times n_i \times r_i}$. Hence, a TT-matrix is a decomposition of an order-$2d$ tensor in d order-4 tensors. The described example is visualized in Fig. 2-7. The decomposition is mathematically then given by

$$\mathcal{A} = \mathcal{A}^{(1)} \times_4^1 \cdots \times_4^1 \mathcal{A}^{(d)} \tag{2-20}$$

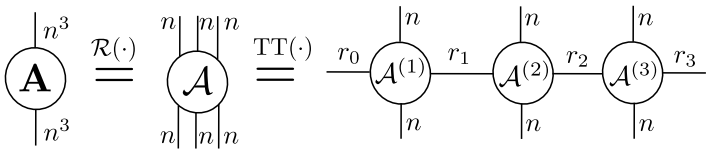


Figure 2-7: TN diagram of the reshaping of a matrix in a tensor and its decomposition in a TT-matrix.

Relation with Kronecker Models [37] A close relation exists between a TT-model and a model of Kronecker products, mathematically given by

$$\mathbf{A} = \sum_{r=1}^{R} \mathbf{A}_r^{(d)} \otimes \cdots \otimes \mathbf{A}_r^{(1)} \tag{2-21}$$

where R is the Kronecker rank. This is equal to a TT-matrix if the Kronecker rank R is equal to the TT-ranks r_i and if the TT-ranks are equal for all tensor cores. Note the reversed order of the subcores in the Kronecker model.

To make this connection clear, take a Kronecker model with Kronecker rank one and an equivalent TT with TT-rank of one. If all tensor cores in the TT-matrix have TT-rank one, then they can be squeezed to matrices. Hence, the mode-m,n product turns into its linear equivalent, the outer product. The two models are equivalent. The latter results since for two matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{B} \in \mathbb{R}^{J_1 \times J_2}$ the (matrix) outer product over a singular index $\mathbf{A} \circ \mathbf{B} \in \mathbb{R}^{I_1 \times I_2 \times J_1 \times J_2}$ and the Kronecker product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2}$ are interrelated by reshuffling and permutations of the indices.

For higher Kronecker ranks and TT-ranks, this results in a concatenation of the single Kronecker submatrices in the tensor cores with the additional degree of freedom by the TT-rank, see e.g. [26]. The demonstration of the Kronecker and TT-model connection also highlights that the TT-model is more general than the Kronecker model since the latter is bounded to uniform ranks over all cores, while the TT-model can have different TT-ranks.

Orthogonality [16] The concept of orthogonality is important for the numeric stability of multiple TT-algorithms; for numeric benefits see e.g. [22]. It is defined for each core of a TT as left- or right-orthogonality. Take one TT core as $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$. It is defined to be left-orthogonal if the transpose of its mode-3 matricification into $\mathbf{X} \in \mathbb{R}^{r_{i-1} n_i \times r_i}$ yields

$$\mathbf{X}^\top \mathbf{X} = \mathbf{I}_{r_i} \tag{2-22}$$

Furthermore it is said to be right orthogonal if its mode-1 matricification into $\mathbf{X} \in \mathbb{R}^{r_{i-1} \times n_i r_i}$ yields

$$\mathbf{X}\mathbf{X}^\top = \mathbf{I}_{r_{i-1}} \tag{2-23}$$

In practice orthogonalization of tensor cores is done core-wise by a QR-decomposition. Therefore, core i is matricified; the orthogonal part is kept for core i and the non-orthogonal part is merged with core $i+1$ or $i-1$ depending on the direction of orthogonalization (left-to-right or right-to-left).

2-2-2 Multilinear Operations on TTs

Multilinear operations can be executed on TTs. This is demonstrated in the following with two examples.

Multidimensional contraction Take two TTs with tensor cores defined as $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{i-1} \times m \times n \times r_i}$ and $\mathcal{B}^{(i)} \in \mathbb{R}^{r_{i-1} \times n \times m \times r_i}$ for $i = 1, \ldots, d$. The contraction is executed core-wise as

$$\mathcal{C}^{(i)} = \mathcal{A}^{(i)} \times_3^2 \mathcal{B}^{(i)}, \quad i = 1, \ldots, d \tag{2-24}$$

The example is visualized in Fig.2-8 where the common indices of $\mathcal{A}^{(i)}$ and $\mathcal{B}^{(i)}$ are connected and contracted in the next step. The computational complexity of each core is given by $\mathcal{O}\left(r^2 n m^2\right)$ assuming that all TT-ranks are equal to r. The complete complexity is given by $\mathcal{O}\left(d r^2 n m^2\right)$. Note that the TT-ranks multiply after contraction as indicated in Fig. 2-8.

Compare the TT contraction with an equivalent representation of matrix multiplication. Take $\mathbf{A} \in \mathbb{R}^{m^d \times n^d}$ and $\mathbf{B} \in \mathbb{R}^{n^d \times m^d}$. The multiplication $\mathbf{C} = \mathbf{A}\mathbf{B}$ requires a total of $\mathcal{O}\left(n^d m^{2d}\right)$ algebraic operations. Hence, the operation in TT-format is more effective for low TT-ranks since it is only linear in d instead of exponential. Low TT-rank models are obtained e.g. by Kronecker models with low Kronecker rank, are separable systems and functions.

Addition [26] The addition of two TTs with the same number of tensor cores is a simple concatenation of each tensor core. The TT-ranks are added. This additional size in the TT-ranks is used to concatenate the two TTs. Take for example two TTs with cores of equal size as $\mathcal{A}^{(i)}, \mathcal{B}^{(i)} \in \mathbb{R}^{r_{i-1} \times n \times r_i}$. Then the addition will yield

$$\mathcal{C}^{(i)} = \begin{bmatrix} \mathcal{A}^{(i)} & 0 \\ 0 & \mathcal{B}^{(i)} \end{bmatrix} \in \mathbb{R}^{2r_{i-1} \times n \times 2r_i}, \quad \text{for } i = 2, \ldots, d-1 \tag{2-25}$$

and for the border cores

$$\mathcal{C}^{(1)} = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{B}^{(1)} \end{bmatrix} \in \mathbb{R}^{1 \times n \times 2r_i}; \quad \mathcal{C}^{(d)} = \begin{bmatrix} \mathcal{A}^{(d)} \\ \mathcal{B}^{(d)} \end{bmatrix} \in \mathbb{R}^{2r_{i-1} \times n \times 1} \tag{2-26}$$

Due to the concatenation no algebraic operations are needed.

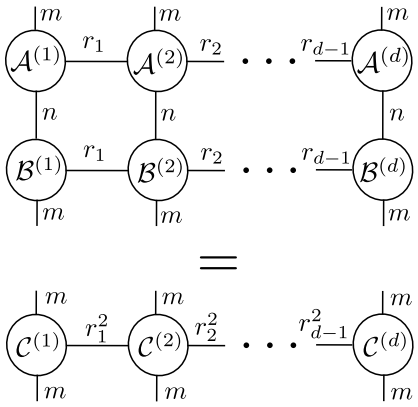


Figure 2-8: Contraction of two TTs over their common index.

2-3 Basic algorithms

In this section several algorithms which are given in literature for TTs are presented. They are used in the development of the MIMO tensor Kalman filter. The section should give an introduction to make the reader familiar with core-wise operations in TT-algorithms.

2-3-1 TT-Rounding

In the previous section it is shown that after contraction of two TTs their TT-ranks multiply and after addition their TT-ranks add. Hence, in the result of multilinear operations the TT-ranks can be suboptimal meaning that an equivalent representation of the TT regarding a given accuracy can be achieved with lower TT-ranks. This truncation of the TT-ranks is accomplished by the TT-rounding algorithm as presented in [26]. It is especially important in recursive algorithms with recursive use of multilinear operations. If no TT-rounding is performed, the TT-ranks grow in each operations and will slow done further multilinear operations due to the dependency on the TT-ranks.

The TT-rounding algorithm is based on truncated SVDs of matricified tensor cores. All cores have to be orthogonalized except the one, which is operated on. The algorithm consists of two main parts

1. Right-to-left orthogonalization of cores:
   Starting from core d until core 2 an economy-size QR-decomposition of the tensor core $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ mode-1 matricified to $\mathbf{X} \in \mathbb{R}^{r_{i-1} \times n_i r_i}$ is performed as $\mathbf{QR} = \text{qr}(\mathbf{X})$. The orthogonal part $\mathbf{Q}$ is reshaped to $\mathcal{Q} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ for the new core $\mathcal{X}^{(i)}$. The non-orthogonal part $\mathbf{R}$ is combined with the tensor core i.

2. $\delta-$ truncated SVD compression of orthogonalized cores:
   The non-orthogonal core $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ is matrizified with the transpose of the mode-3 matricification to a new core $\mathbf{X} \in \mathbb{R}^{r_{i-1} n_i \times r_i}$. A δ-truncated SVD of this core is computed as $\mathbf{USV}^\top = \text{svd}_\delta(\mathbf{X})$. The left-orthogonal matrix $\mathbf{U}$ is reshaped into $\mathcal{U} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ to be the new core $\mathcal{X}^{(i)}$. The remaining part $\mathbf{SV}^\top$ is merged with

the $i+1$th core. Then, the truncated SVD is computed for this next core. This process is repeated from core 1 until core d until all cores are truncated.

The threshold for the truncation in the δ-SVD is computed by the Frobenius norm of the complete TT by

$$\delta = \frac{\epsilon}{\sqrt{d-1}} ||\mathbf{X}||_F \tag{2-27}$$

where ϵ is a prescribed accuracy of the TT.

2-3-2 Vector/Matrix Conversion to TT

Often data is given in matrix or vector format. In order to convert it to TT-format a conversion algorithm is necessary. Two main algorithms are used. The TT-cross approximation [31] and the TT-SVD algorithm [26]. The latter is explained in the following.

Similar to the second step of the TT-rounding algorithm the TT-SVD is based on a sequence of δ-truncated SVDs. The truncation threshold is computed in the same way as for the TT-rounding with Eq. (2-27). As input to the algorithm an order-d tensor is required which can be obtained by reshaping of an exponentially large matrix or vector.

The algorithm initializes the first TT-rank with the boundary condition $r_0 = 1$. The input tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is first matricified into $\mathbf{X} \in \mathbb{R}^{r_{i-1}n_i \times \frac{\text{numel}(\mathcal{X})}{r_{i-1}n_i}}$. A δ-truncated SVD of this matricification is computed as $\mathbf{USV}^\top = \text{svd}_\delta(\mathbf{X})$. The TT-rank r_i is computed as δ-truncated rank of $\mathbf{S}$. The left-orthogonal part $\mathbf{U}$ is reshaped to $\mathcal{U} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$ as core $\mathcal{X}^{(i)}$. The residual part $\mathbf{SV}^\top$ is then used for the next δ-truncation in an SVD after appropriate reshaping into $\mathbf{SV}^\top \in \mathbb{R}^{r_{i-1}n_i \times \frac{\text{numel}(\mathbf{SV}^\top)}{r_{i-1}n_i}}$. This operations are repeated for cores $i = 1, \ldots, d-1$. For the last core the TT-rank is given by the boundary condition $r_d = 1$. Hence, the remaining part of the last $\delta-$truncated SVD $\mathbf{SV}^\top$ is reshaped into $\mathcal{X}^{(d)} \in \mathbb{R}^{r_{i-1} \times n_i \times 1}$.

The algorithm yields an optimal conversion with respect to the given accuracy ϵ due to the fact that the conversion uses δ-truncated SVDs in the same way as the TT-rounding algorithm. All cores except from core d are inherently orthogonal, with the norm being completely in core d.

2-3-3 Alternating Linear Scheme (ALS)

The TT-ALS scheme is presented in [16] and is an iterative and alternating optimization algorithm in TT-format. One tensor core is optimized at a time. Hence, smaller subproblems are solved iteratively instead of one large problem at once. The algorithm is based on orthogonality of all cores except the one which is optimized. This core does contain the complete norm of the TT.

The defined optimization problem is solved for the non-orthogonal core i. The solution of the optimization is a tensor of the same size as the optimized core i. Next, an orthogonalization step is performed to orthogonalize core i and move its norm to core $i+1$. This step is done in the same way as the first part of the TT-rounding algorithm by means of QR-decompositions.

The ALS algorithm performs these core-wise optimizations for core $i = 1, \ldots, d-1$ as so called forward sweep and for $i = d, \ldots, 2$ as backwards sweep. A combination of forward and backward sweep is called a full sweep. Multiple sweeps are executed until a performance criterion for the convergence is met. Often the core-wise optimization step can be executed directly by TT-contractions without the use of extensive optimization algorithms.

The algorithm is suited for general optimization problems. It is successfully applied for multiple applications. Among these are the solution of linear systems in general and the solution of eigenvalue equations [16], the matrix-by-vector product directly in TT-format [29], the solution of linear problems for inversion [27] and for the pseudoinverse [23] and the identification of MIMO Volterra systems [4].

The described ALS algorithm is based on fixed TT-ranks. This means that the TT-ranks after the optimization are a priori determined and are not changed dynamically. A modification is presented in [16] to tackle this issue. This adapted algorithm is called modified ALS (MALS). In a similar form it is previously presented in the area of quantum physics as Density Matrix Renormalization Group (DMRG) [40]. The MALS can vary the TT-ranks after the optimization for optimal TT-ranks according to a given accuracy. This is achieved by optimizing a so called supercore consisting of two single cores at each iteration step. After optimization this supercore is split up by an SVD, where the TT-ranks can be changed dynamically according to a required accuracy ϵ.

Limitations of the TT-ALS and TT-MALS are often given by the TT-ranks. The solution TT-ranks or the ones to compute the solution of the optimization problem can in general be high. Hence, contractions are slow and special optimization algorithms in matrix format may be faster. These limitations are highly dependent on the problem.

2-3-4 TT-randomized SVD (TTrSVD)

The SVD is a powerful and often used tool. An equivalent in TT-format is presented as randomized SVD (rSVD) in [6]. An overview on different rSVD algorithms in matrix format is given in [14]. The rSVD is an approximate matrix factorization of the matrix $\mathbf{X}$ for large-scale problems. The matrix $\mathbf{X}$ is multiplied by a random matrix (often taken as Gaussian) which reduces the size of the decomposition problem. The intuitive idea is to obtain a random sample of the range of $\mathbf{X}$ with this multiplication. Hence, a standard SVD of a smaller sized problem can be performed.

The extension of this scheme to TT-format in [6] works in the same way. A prescribed number of singular values is necessary. The result of the TTrSVD algorithm $[\mathcal{U}, \mathbf{S}, \mathcal{V}] = \text{TTrSVD}(\mathcal{X})$ yields orthogonal TTs $\mathcal{U}, \mathcal{V}$ and a matrix containing the singular values $\mathbf{S}$. The tensor network connects the first core of $\mathcal{U}$ with the matrix $\mathbf{S}$ which is also connected to the first core of $\mathcal{V}^\top$.

Limitations of the algorithms are that the number of singular values has to be known a priori, which is often not the case. Furthermore a contraction of the first i cores of $\mathcal{X}$ is necessary such that the mode-size is larger than the desired number of singular values. This is necessary for the thin TT-QR decomposition within the TTrSVD algorithm.

2-3-5 Tensor Network Kalman Filter

A generalization of the recursive Kalman filter equations [18] for systems with exponentially large state vectors with the help of TTs is presented in [3, 5]. These references demonstrate the Kalman filter for the system identification of MIMO Volterra systems. All variables in the tensor networks Kalman filter are in TT-format and are operated on with multilinear operations.

The basic idea to lift the curse of dimensionality is to distribute the exponentially large state vector as modes over all TT-cores. Suitable prime-factorizations of the state vector size can be used. Within the presented framework, all outputs are on one tensor core. Hence, the theory is not suited for MIMO systems with exponentially large output vectors. Furthermore, large TT-ranks in the system matrices in TT-format can slow down the computations significantly. Approximations with lower TT-ranks are often not possible without loss of important system information.

A solution for the latter problem is presented in chapter 4 and the MIMO tensor Kalman filter problem is addressed in chapter 5. Therefore no elaborate description of the tensor Kalman filter is given here since the following chapters include a more detailed description of the theory, its limitations and possible remedies.

Chapter 3

Introduction to Adaptive Optics

3-1 General

Scientists in the field of optics rely on flat wavefronts in order to obtain a sharp image of point sources that are studied. Take a collection of light rays over a 2D field. The wavefront is considered to be flat if at any measurement plane rectangular to the direction of propagation all light rays have the same phase information. For distant point sources this is in general the case. However, disturbances cause aberrations in the wavefront, resulting in a non-flat smooth wavefront. For example atmospheric turbulence causes local differences in the refraction index. Therefore, the path length for light rays varies, meaning that the phase information of all light rays is not aligned and a non-flat, smooth wavefront is obtained. This distortion results in a blurred image, which is scientifically less valuable. [39]

An AO system is a compensation method to improve the image quality when disturbances are blurring the image dynamically. The shape of the wavefront is measured with a wavefront sensor, which is often a linear Shack-Hartmann sensor [34]. A deformable mirror consisting of an array of lenses is used to actively add a phase correction to compensate for the measured distortions. In order to obtain a flat wavefront, the inverse of the measured wavefront is applied to the deformable mirror [15]. A schematic of the AO system is drawn in Fig. 3-1. This method is equally applicable for astronomy [35], microscopy [7] or lithography [41].

Between measurement of the wavefront in the Shack-Hartmann sensor and the change of the lenses in the deformable mirror there is a time-delay. This occurs mainly due to sensor readout times, computation of the control law and the time for changing the deformable mirror. However, within this time, the disturbance may have changed significantly and the control signal is not of use any more. Therefore, a prediction of the wavefront is necessary. The Kalman filter is a natural choice since it is an optimal predictor, which can minimize the blurring of the image.

The problem that researchers and engineers are facing with this kind of system is the following. For example in astronomy a higher accuracy in the images is demanded to research and study even more faint and distant objects in greater detail. The diffraction limit is a measure

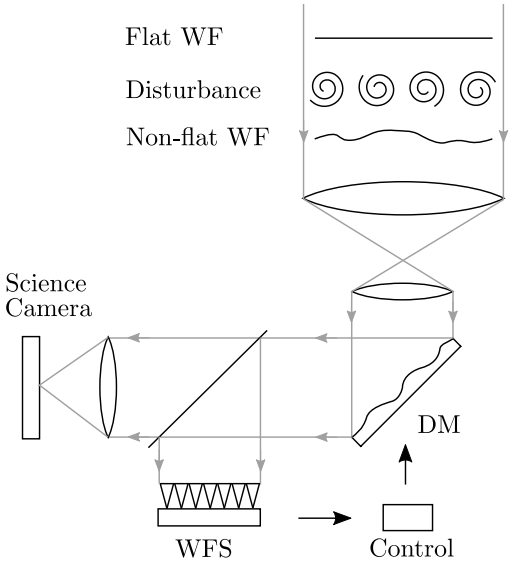


Figure 3-1: Schematic of an AO system including disturbance of the flat wavefront (WF), the deformable mirror (DM), the wavefront sensor (WFS) with the control loop.

for the accuracy of the optical system and changes reciprocal to the lens size. Hence, larger telescopes are build and also larger AO systems are necessary with more sensors and actuators [39]. Therefore, the prediction problem becomes large-scale and a scalable and an accurate large-scale Kalman filter is required to compensate in real-time for the distortions. For this type of filter a linear model of the turbulence with its spatial and temporal dynamics but also of the wavefront sensor and measurement process is necessary.

3-2 Turbulence Model

A 2D grid of the wavefront that reaches the optical system from the observed object is considered. The turbulence changes this wavefront in two ways. In its spatial distribution and temporally by dynamic changes. Both of these processes are analysed individually in the following in order to obtain one combined turbulence model.

3-2-1 Spatial Turbulence

The spatial turbulence describes statistically the changes in the wavefront over the spatial indices parallel to the measurement plane. It is based on first-principles with underlying physical equations. In general the Kolmogorov theory [20, 21] or Von Kármán theory [10] is used. In AO, the latter is often preferred [33]. The theory describes the spatial correlation between points separated by a distance r dependent on two main weather dependent parameters. The Fried parameter r_0, which is an intuitive measure of the turbulence strength [11]. It defines the effective size of a telescope lens under the effect of this turbulence. For example a Fried parameter of $r_0 = 0.5$ [m] means, that with this severity of turbulence the telescope is as effective as a telescope of this lens diameter without any disturbances. The second parameter

is the outer scale L_0, giving a measure for the size of the turbulent structures [24]. The spatial correlation can be given by

$$\mathbf{C}_\phi(r) = \frac{\Gamma(11/6)}{2^{5/6}\pi^{8/3}} \left(\frac{48\pi r \Gamma(6/5)}{5L_0}\right)^{5/6} \left(\frac{r_0}{L_0}\right)^{-5/3} \mathcal{K}_{5/6}\left(2\pi \frac{r}{L_0}\right) \tag{3-1}$$

where $\Gamma(\cdot)$ is the gamma function and $\mathcal{K}(\cdot)$ is the modified Bessel function of second kind. This covariance function yields an analytic expression for the 2D distribution of the vectorized wavefront field ϕ by $\mathbf{C}_{\phi,k} = \mathbb{E}\left[\phi_k \phi_k^\top\right]$.

3-2-2 Temporal Turbulence

Contrary to the spatial dynamics, the temporal dynamics of turbulence are an open research topic. Multiple choices of dynamic models are possible. For the linear Kalman predictor, a linear model is necessary. A frequently chosen type of model is the discrete-time autoregressive (AR) model. A first and second order model approach is presented hereafter [24].

AR-1 Model: This model considers one past value of the state. For the turbulence the vectorized wavefront ϕ_k is chosen as state. Mathematically the model is given by

$$\phi_{k+1} = \mathbf{A}_k \phi_k + \mathbf{v}_k \tag{3-2}$$

with Gaussian noise $\mathbf{v}_k \sim (0, \mathbf{Q}_k)$. Note that the general time-varying case is chosen. This is necessary since in AO, the system matrices can be updated when new information is available. The state transition matrix $\mathbf{A}_k$ is in practice often chosen as diagonal with $\mathbf{A}_k = a_k \mathbf{I}$ where $|a_k| < 1$ for stability. The noise covariance $\mathbf{Q}_k$ is to be computed consistently with the derived spatial covariance $\mathbf{C}_\phi$ as

$$\mathbf{Q}_k = \mathbf{C}_\phi - \mathbf{A}_k \mathbf{C}_\phi \mathbf{A}_k^\top \tag{3-3}$$

This yields the process equation of a state-space system description.

AR-2 Model: For this model two past values of the state ϕ_k are considered. Mathematically this yields

$$\phi_{k+1} = \mathbf{A}_{1,k}\phi_k + \mathbf{A}_{2,k}\phi_{k-1} + \mathbf{v}_k \tag{3-4}$$

this can be translated in the following state equation

$$\begin{bmatrix} \phi_{k+1} \\ \phi_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,k} & \mathbf{A}_{2,k} \\ \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \phi_k \\ \phi_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \mathbf{v}_k \tag{3-5}$$

Similarly to the AR-1 model the noise covariance has to be in line with the wavefront covariance. Hence

$$\begin{bmatrix} \mathbf{Q}_k & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{\phi,k} & \mathbf{C}_{\bar{\phi},k} \\ \mathbf{C}_{\bar{\phi},k}^\top & \mathbf{C}_{\phi,k} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{1,k} & \mathbf{A}_{2,k} \\ \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{C}_{\phi,k} & \mathbf{C}_{\bar{\phi},k} \\ \mathbf{C}_{\bar{\phi},k}^\top & \mathbf{C}_{\phi,k} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{1,k}^\top & \mathbf{I} \\ \mathbf{A}_{2,k}^\top & 0 \end{bmatrix} \tag{3-6}$$

with $\mathbf{C}_{\phi,k} = \mathbb{E}\left[\phi_k \phi_k^\top\right] = \mathbb{E}\left[\phi_{k-1}\phi_{k-1}^\top\right]$ and $\mathbf{C}_{\bar{\phi},k} = \mathbb{E}\left[\phi_k \phi_{k-1}^\top\right]$.

3-3 Wavefront Sensor

The wavefront as a spatially distributed wave changes over time which needs to be measured on a 2D grid at each sampling instance. Therefore, consider a regular grid of wavefront samples $\varphi_{i,j}$ collected in the following matrix obeying the spatial arrangement

$$\Phi = \begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,c} \\ \vdots & \ddots & \vdots \\ \varphi_{c,1} & \cdots & \varphi_{c,c} \end{bmatrix} \tag{3-7}$$

From this matrix the state vector is obtain by vectorization $\text{vec}(\Phi_k^\top) = \phi_k$. The Shack-Hartmann sensor [34] measures these samples by use of an $c \times c$ array of small lenses (or lenslets) which concentrate the light of a bright reference source on a photon sensor. The location on the sensor depends on the local slope of the wavefront. A flat wavefront results in a central measurement; non-flat wavefronts have deviations from the centre on the photon sensor. In fact slopes of the wavefront in $x-$direction s_x and in $y-$direction s_y are measured. In a Fried geometry, four wavefront samples φ are used to generate one slope measurement output. This is illustrated for a small, square example in Fig. 3-2.

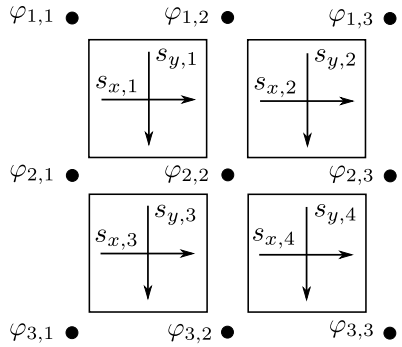


Figure 3-2: Illustration of a Shack-Hartmann sensor for $c = 3$. The black dots represent the wavefront sampling points $\varphi_{i,j}$ which are used to compute the slopes $s_{x,l}, s_{y,m}$.

To illustrate the computation of the slopes with wavefront samples take an example of $c = 2$ lenslets. The following holds

$$\begin{aligned} \begin{bmatrix} s_{x,1} \\ s_{y,1} \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} (\phi_{2,2} + \phi_{2,1}) - (\phi_{1,1} + \phi_{1,2}) \\ (\phi_{2,2} + \phi_{1,2}) - (\phi_{1,1} + \phi_{2,1}) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \end{aligned} \tag{3-8}$$

For large systems, the slopes s are stacked in the measurement matrix $\mathbf{G} \in \mathbb{R}^{2(c-1)^2 \times c^2}$. This directly yields the number of outputs of the system with $p = 2(c-1)^2$ and the number of states with $n = c^2$. As an example a visualization of a measurement matrix with $c = 5$ is given in Fig. 3-3. Note the size of matrix with $p = 32$ outputs and $n = 25$ states. Note also the special block structure of the matrix. This yields the output equation of a state-space system as

$$\mathbf{y}_k = \mathbf{G}\phi_k + \mathbf{w}_k \tag{3-9}$$

with Gaussian output noise $\mathbf{w}_k \sim (0, \mathbf{R}_k)$. The sensor noise can considered to be independent for each channel, by $\mathbf{R} = \sigma_{w,k}^2 \mathbf{I}$. Note that the Shack-Hartmann sensor cannot distinguish two modes, which may result as unobservable modes in the system. First, the piston mode with $\varphi_{i,i} = \varphi_{j,j}$, which is an offset. This mode is often not problematic in control problems since it yields a flat wavefront. Second, the waffle mode with $\varphi_{i,i} = -\varphi_{i+1,i} = -\varphi_{i,i+1} = \varphi_{i+1,i+1}$, which depicts a chessboard pattern. Special measures have to be taken to only consider the observable modes of the system.

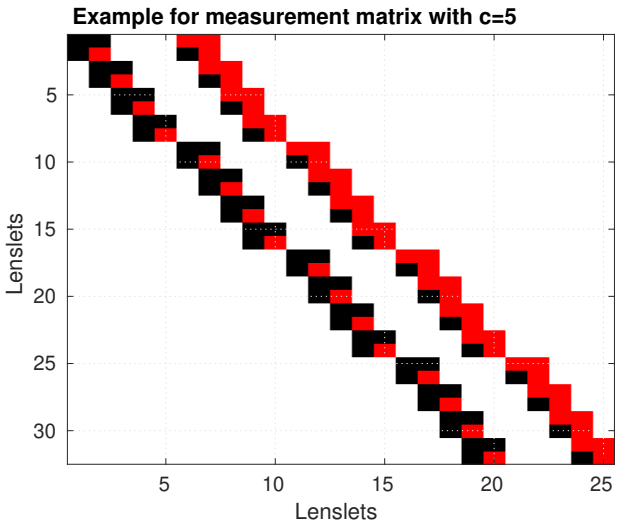


Figure 3-3: Example for stacking of slopes in the measurement matrix $\mathbf{G}$ with $c = 5$. Black fields yield values with -0.5 and red fields with $+0.5$.

3-4 Wavefront Estimation

In the AO system a wavefront predictor for at least one time step is necessary to compensate for the time-delay between the wavefront measurement and the correction with the deformable mirror. Due to the derived state-space model one could use the process equation with its result ϕ_{k+1} for predictions. However, the spatial modelling is done on first principles and the temporal model is based on an assumption about the temporal dynamics. Therefore, the model may not be highly accurate. An optimal estimation that makes use of the current wavefront measurements with statistical updates is necessary. The Kalman filter [18] does have these properties as optimal propagator.

For telescopes nowadays like the ESO Very Large Telescope (VLT) [36] or of the next generation like the European Extremely Large Telescope (E-ELT) [12] the size of the AO systems rise. The number of states to be estimates on-line reaches several tens of thousands with a similar number of outputs. This puts a highly demanding requirement on the computational complexity of the required wavefront real-time prediction algorithm. A second requirement is given by the time-variance of the model. In general the system model can assumed to be time-invariant. However, after some time the turbulence covariance can change significantly such that updates of the model are necessary, making the system inherently time-variant.

3-5 General Issues

Besides the large-scale and time-varying requirement, the AO system has more issues to face with. This has to be considered in the design of the algorithm, especially since the choice is to use tensor calculus for the large-scale predictor. These are in particular

1. The aperture of the AO system is round due to the system of lenses. In contrary the TN Kalman filter is considered to be suited for Kronecker models which are square. Special padding in the output equation is necessary for real-life simulations which will inherently reduce the accuracy of the estimation especially at the edges of the round aperture.
2. The MIMO TN Kalman filter as developed and presented in chapter 5 puts several requirements on a special form for the output equation. The system equations as derived here are not in this form. The derivation is given in dedicated chapter.

Chapter 4

EUSIPCO Paper

This chapter contains in the following a printout of the final submission and accepted paper for the 27th European Signal Processing Conference (EUSIPCO) 2019. This is the flagship conference of the European Association for Signal Processing (EURASIP). The conference is held in A Coruña, Spain from the 2nd to the 6th of September 2019.

The main contribution of the submitted paper is twofold. First, an extension and generalization of the TN Kalman filter [3, 5] to LTI state-space systems is presented. Second, a method based on TT-rank truncation of the covariance tensors $\mathcal{P}_{k|k}$ and $\mathcal{P}_{k+1|k}$ is developed to speed up the computational times for systems with TT-ranks higher than one in the system matrices $\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}$. This method is incorporated in the TT-rounding sequence [26] with a fixed value. Hence, a combination of existing methods to enable the TN Kalman filter for more general systems is presented. Simulation results for randomly generated systems underline and support the chosen approach.

The first draft of the paper was handed in on the 18th of February 2019. Notification of acceptance was received on the 4th of June 2019. Three reviews are obtained and their comments are included in the final submission. Quantitative feedback was given as the following

- Novelty and originality: $\varnothing 3/5$
- Technical content and correctness: $\varnothing 3.67/5$
- Relevance and Timeliness: $\varnothing 4/5$
- Quality of presentation: $\varnothing 3.33/5$

Qualitative and highly useful feedback regarded the accuracy in the simulation results, the randomness of the Monte Carlo approach and to evaluate if the covariance tensors remain symmetric, positive-definite after the truncation method. The final paper was submitted on the 18th of June 2019.

Tensor Network Kalman Filter for LTI Systems

Daniel Gedon†, Pieter Piscaer†, Kim Batselier†, Carlas Smith†, Michel Verhaegen†

***Abstract*—An extension of the Tensor Network (TN) Kalman filter [2], [3] for large scale LTI systems is presented in this paper. The TN Kalman filter can handle exponentially large state vectors without constructing them explicitly. In order to have efficient algebraic operations, a low TN rank is required. We exploit the possibility to approximate the covariance matrix as a TN with a low TN rank. This reduces the computational complexity for general SISO and MIMO LTI systems with TN rank greater than one significantly while obtaining an accurate estimation. Improvements of this method in terms of computational complexity compared to the conventional Kalman filter are demonstrated in numerical simulations for large scale systems.**

***Index Terms*—Kalman filter, LTI systems, tensors, tensor train, large scale systems, SISO, MIMO, curse of dimensionality.**

I. Introduction

The Kalman filter [9] is a stochastic optimal filter for dynamic linear systems. Since its introduction, it is successfully applied to a variety of different applications, see e.g. [1], [7]. For systems with exponentially large state size n^d and output size p assuming $p \leq n^d$, the conventional Kalman filter is infeasible. First, because the computational complexity scales with order $\mathcal{O}\left(n^{3d}\right)$. Second, the storage of exponentially large system dynamics is in matrix form prohibitive. A more suitable filter framework has to be developed.

One possibility for a large scale Kalman filter is the Tensor Network (TN) Kalman filter as developed in [2], [3]. Both concern the system identification of Multiple-Input Multiple-Output (MIMO) Volterra systems using Kalman filter where the latter one uses a batch of multiple measurements. The Volterra systems are rewritten in LTI system form and therefore the described implementation can equally be used for dynamic linear systems. Hence, in the remainder of the paper the method in [2] will be denoted as Single-Input Single-Output (SISO) and the method in [3] as MIMO TN Kalman filter. This filter makes use of special TNs [12] in Tensor Train (TT) format [13] without explicitly constructing the underlying exponentially large matrices and vectors. Hence, a reduction of computational complexity from $\mathcal{O}\left(n^d\right)$ to $\mathcal{O}\left(dn\right)$ is achieved. Fig. 1 highlights the computational advantage of the TN Kalman filter for large LTI systems while still obtaining accurate estimation results (see full and dashed line). This holds for the special case where the system dynamics in TT-format have the property that all TN ranks are equal to one, denoted as $\mathrm{ttr}(\cdot) = 1$ and formally introduced in the preliminaries.

† Affiliated with Delft Center for Systems and Control, Delft University of Technology, The Netherlands. E-mail addresses of corresponding authors: d.gedon@student.tudelft.nl (Daniel Gedon), p.j.piscaer@tudelft.nl (Pieter Piscaer).

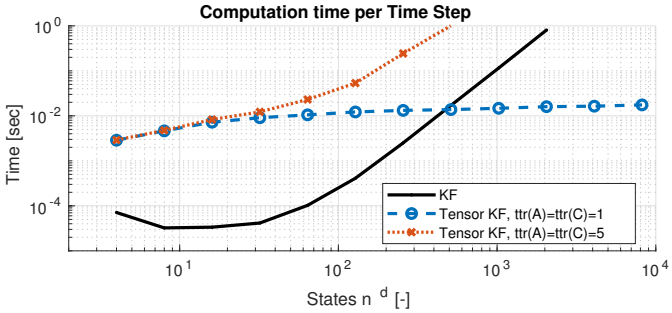


Fig. 1. Computation time per time step of random stable LTI SISO system for conventional and TN Kalman filter using rounding tolerance $\epsilon = 10^{-6}$ and mean out of 15 Monte Carlo simulations (using varying initial conditions and noise) over 25 time steps.

In general, the TN ranks can be high for large unstructured systems and are only bounded by the canonical rank [16] of an exact canonical decomposition of the underlying tensor [4], [11]. The computational complexity of the TN Kalman filter is polynomial dependent on the TN rank of the system dynamics matrices $\mathbf{A}$ and $\mathbf{C}$ in TT-format and the TN rank of the internal variables of the filter (covariance, Kalman gain, state estimate). This results in longer computation times for larger TN ranks. Additionally, the implementation of the MIMO TN Kalman filter in [3] has a complexity of $\mathcal{O}\left(p^3\right)$.

In this paper, a solution is proposed to tackle the problem of large TN ranks in order to reduce computation time while obtaining accurate results. This is especially of interest since general system matrices do not have an optimal TN rank of one. Fig. 1 shows the computation times in a comparison for systems with TN rank at one and higher. The main contributions of this paper can be summarized as:

- A possibility to reduce the effect of the TN ranks on the computational complexity is derived. Therefore, the covariance tensor in the TN Kalman filter is approximated with low TN ranks, yielding a fast and accurate estimation.
- Numeric simulation results are presented as comparison between the conventional and TN Kalman filter. The latter is shown with and without low TN rank approximation of the covariance tensor to demonstrate the power of the novel approach.

The paper is structured as follows. Section 2 introduces the notion of tensors throughout the paper. It gives an overview on TTs with multilinear operations and introduces the use of TTs in the generalized TN Kalman filter. In section 3, the method to identify the computational bottlenecks of the TN Kalman filter is highlighted. Based on this analysis, the approach to

approximate the covariance tensor is derived. Using these insights, numerical simulations of the novel approach are presented in section 4. Finally, section 5 concludes with final remarks and open research directions.

II. Preliminaries

In this paper, a tensor is a multidimensional array as a generalization of matrices to higher order. An order-d tensor has d indices and is denoted with capital calligraphic letters $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. Matrices will be denoted by capital bold letters $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ and vectors by bold letters $\mathbf{x} \in \mathbb{R}^{n_1}$. Scalars are given as Roman letters $x \in \mathbb{R}$. The ith tensor element of a set of tensors is indicated by superscript in round brackets $\mathcal{X}^{(i)}$. Introductions to tensors and their decompositions are given in [6], [11], [12].

A. Tensor Train Theory

The Tensor Train (TT) decomposition is mathematically introduced by [13] after previous work in [15], [17]. It decomposes an order-d tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in a series of d order-3 tensors $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(d)}$ with $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{X_{i-1}} \times n_i \times r_{X_i}} \ \forall i$, called TN cores. The parameter r_{X_i} are the TN ranks connecting the single TN cores in a line network. The border TN cores are defined as $r_{X_0} = r_{X_d} = 1$. Throughout this paper, the maximum TN rank over all TN ranks of a TT $\mathcal{X}$ will be denoted as $\mathrm{ttr}(\mathcal{X}) = \max(r_{X_i}) = r_X \ \forall i$.

An example for a graphical visualization of the TT decomposition using TN diagrams [6], [12] is given for an order-6 tensor $\mathcal{X}$ in Fig. 2. Each circle depicts a TN core; the number of free lines is the order of the tensor; the size of the interconnecting lines represent the TN ranks r_{X_i}. The example decomposes the order-6 tensor $\mathcal{X}$ in six order-3 tensors $\mathcal{X}^{(i)}$, $i = 1, \ldots, 6$.

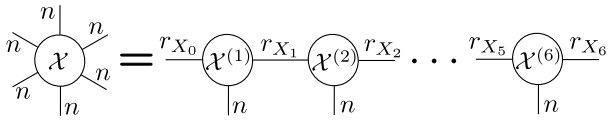


Fig. 2. Graphical visualization of the TT decomposition of an order-6 tensor $\mathcal{X}$ using TN diagrams.

The power of the TT decomposition is twofold. Storing an exponentially large matrix $\mathbf{X} \in \mathbb{R}^{n^d \times n^d}$ requires storage of $\mathcal{O}\left(n^{2d}\right)$ entries. In TT-format storage of $\mathcal{X}$ requires only $\mathcal{O}\left(dn^2 r_X^2\right)$ entries, i.e. linear in the exponent d, which is a reduction if the TN ranks r_X are small. Secondly, multilinear algebra as a generalization of linear algebra to higher spaces can be used to effectively apply basic mathematical operations [8]. This is computationally effective since the operations work on each single TN core, which are small compared to the full matrix. An example of multilinear algebra is the so called mode-n product as a higher order equivalent of matrix multiplications for tensors [6], [11]. Often, the term contraction along the mode n is used for this operation.

A problem with multilinear operations in TT-format is the increase of the TN rank. For example, the contraction along the second mode of the tensor $\mathcal{A}$ with TN cores $\mathcal{A}^{(i)} \in \mathbb{R}^{r_A \times n \times r_A}$ with the tensor $\mathcal{B}$ with TN cores $\mathcal{B}^{(i)} \in \mathbb{R}^{r_B \times n \times r_B}$ has resulting TN ranks $r_A r_B$. In [13], a procedure called rounding is described to decrease the TN ranks towards a given rounding tolerance ϵ. Two main steps are needed. First, an orthogonalization of all TN cores using QR decompositions and second, a δ-truncated SVD.

Consider for example a tensor $\mathcal{X}$ in TT-format with orthogonal TN cores. The procedure computes the δ-truncated SVD of each matricified TN core $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{X_{i-1}} \times n \times r_{X_i}}$. The matricification is done such that the number of columns is equal to the right TN rank $\mathbf{X}^{(i)}_{(r_{X_i})} \in \mathbb{R}^{r_{X_{i-1}} n \times r_{X_i}}$. The truncation threshold is given by

$$\delta = ||\mathbf{X}||_F \frac{\epsilon}{\sqrt{d-1}}, \tag{1}$$

using the rounding tolerance ϵ and the number of TN cores d.

B. Tensor Kalman Filter

The TT theory can be used to define a Kalman filter and lift the curse of dimensionality for large scale systems. Therefore, the following LTI system is considered

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{2}$$

with state $\mathbf{x}_k \in \mathbb{R}^{n^d}$, measurement $\mathbf{y}_k \in \mathbb{R}^p$, $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$, $\mathbf{C} \in \mathbb{R}^{p \times n^d}$ and covariance matrices $\mathbf{Q} \in \mathbb{R}^{n^d \times n^d}$ and $\mathbf{R} \in \mathbb{R}^{p \times p}$ for the process noise $\mathbf{w}_k$ and measurement noise $\mathbf{v}_k$, respectively. The output is $p = 1$ for SISO and $p = n^d$ for the MIMO case. The algebraic equations of the conventional Kalman filter are split in two parts. The measurement update:

$$\mathbf{S} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^\top + \mathbf{R} \tag{3}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^\top \mathbf{S}^{-1} \tag{4}$$

$$\mathbf{v} = \mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1} \tag{5}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v} \tag{6}$$

$$\mathbf{P}_{k|k} = \left(\mathbf{I}_n - \mathbf{K}_k\mathbf{C}\right)\mathbf{P}_{k|k-1} \tag{7}$$

and the time update:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_{k|k} \tag{8}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^\top + \mathbf{Q} \tag{9}$$

When the system matrices $\mathbf{A}$, $\mathbf{C}$ are large scale matrices, the computation time rises exponentially as seen in Fig. 1. This curse of dimensionality can be lifted by using the TN Kalman filter. This is a generalization of the conventional Kalman filter for higher order dimensions in TT format. It adapts the Kalman filter equations to the required multilinear algebra. The TN Kalman filter is introduced in [2] and [3].

In order to apply the TN Kalman filter to general LTI systems, the system dynamics in Eq. (2) have to be transformed in TTs. Consequently, also the variables in the TN Kalman filter are TTs. Use is made of the computational effective multilinear algebra for TTs with this transformation. As an example, the transformation of the state vector estimate $\hat{\mathbf{x}}$ to TT-format is

explained intuitively. First, the vector $\hat{\mathbf{x}} \in \mathbb{R}^{n^d}$ is reshaped into an order-d tensor $\hat{\mathcal{X}} \in \mathbb{R}^{n \times \cdots \times n}$. Second, the order-$d$ tensor is decomposed using the TT decomposition in d order-3 tensors $\hat{\mathcal{X}}^{(i)} \in \mathbb{R}^{r_{X_{i-1}} \times n \times r_{X_i}}$. The set counting index i goes from 1 to d and the border TN ranks are $r_{X_0} = r_{X_d} = 1$. This decomposition step is visualized for $d = 6$ in Fig. 2 using a TN diagram. The TN rank r_{X_i} connects the TN core $\hat{\mathcal{X}}^{(i)}$ with $\hat{\mathcal{X}}^{(i+1)}$. For the state transition matrix $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$ in TT-format, there is one more mode of size n for each TN core compared to the state vector $\hat{\mathbf{x}}$. These examples illustrate how the following variables for the TN Kalman filter are defined in TT-format

- $\mathcal{A}$ with $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{A_{i-1}} \times n \times n \times r_{A_i}}$
- $\mathcal{C}$ with $\mathcal{C}^{(1)} \in \mathbb{R}^{r_{C_0} \times p \times n \times r_{C_1}}$ and $\mathcal{C}^{(i)} \in \mathbb{R}^{r_{C_{i-1}} \times n \times r_{C_i}}$
- $\mathcal{P}_{(\cdot|\cdot)}$ with $\mathcal{P}^{(i)}_{(\cdot|\cdot)} \in \mathbb{R}^{r_{P_{i-1}} \times n \times n \times r_{P_i}}$
- $\hat{\mathcal{X}}_{(\cdot|\cdot)}$ with $\hat{\mathcal{X}}^{(i)}_{(\cdot|\cdot)} \in \mathbb{R}^{r_{X_{i-1}} \times n \times r_{X_i}}$
- $\mathcal{K}_k$ with $\mathcal{K}_k^{(1)} \in \mathbb{R}^{r_{K_0} \times p \times n \times r_{K_1}}$ and $\mathcal{K}_k^{(i)} \in \mathbb{R}^{r_{K_{i-1}} \times n \times r_{K_i}}$
- $\mathcal{Q}$ with $\mathcal{Q}^{(i)} \in \mathbb{R}^{r_{Q_{i-1}} \times n \times n \times r_{Q_i}}$

For completeness, the definition of the remaining matrices and vectors are given as $\mathbf{S} \in \mathbb{R}^{p \times p}$, $\mathbf{R} \in \mathbb{R}^{p \times p}$ and $\mathbf{v} \in \mathbb{R}^{p}$. The specific implementation for the SISO and MIMO filter is elaborated in [2] and [3] respectively.

III. Problem Analysis and Improvment

In this part, the problem of increasing computation time for high TN ranks due to their polynomial complexity is tackled. This is done for the SISO and MIMO case since both struggle with this phenomenon. Stable LTI systems without specific structure of the matrices are considered. Converting such system dynamics in TT-format yields in general $\mathrm{ttr}(\mathcal{A}) > 1$, $\mathrm{ttr}(\mathcal{C}) > 1$. This general case is taken into account for the following analysis. Hence, this will reduce the computational speed of the TN Kalman filter significantly as presented in Fig. 1 for the SISO case. Previous results in [2] have shown that a rounding tolerance greater than the machine precision is sufficient for accurate results and can speed up the computation time. This result is applied in the following by choosing $\epsilon = 10^{-6}$.

A. Analysis of State of the Art

To get insights on possible bottlenecks of the algorithm, the computational complexity of the state of the art is analysed. For the SISO tensor Kalman filter this can be found in Table 2 of [2] and summarized with $\mathcal{O}(dn^3 r^x_{(\cdot)})$ where the last factor indicates the polynomial complexity in several TN ranks. This compares to $\mathcal{O}(n^{3d})$ for a comparable conventional SISO Kalman filter. For the state of the art MIMO tensor KF, the complexity analysis is given in Table I.

It can be seen that there are two computational bottlenecks. (1) The computation of the Kalman gain with cubic complexity in the outputs p, since the inverse of the result of the Riccati equation $\mathbf{S}$ is necessary. (2) The polynomial dependency on all TN ranks. This paper addresses point two, while the first point is a topic for future research.

TABLE I
Computational Complexity of MIMO TN Kalman Filter.

Step	Kalman filter	TN Kalman filter
$\mathbf{S}$	$\mathcal{O}\left(pn^{2d} + p^2 n^d\right)$	$\mathcal{O}((d-1)(n^2 r_P^2 r_C^2 + n r_P^2 r_C^4) + +n^2 p r_P r_C + n p^2 r_P r_C^2)$
$\mathcal{K}$	$\mathcal{O}(pn^{2d} + p^2 n^d + +p^3)$	$\mathcal{O}((d-1)(n^2 r_P^2 r_C^2) + n^2 p r_P r_C + +n p^2 r_P r_C + p^3)$
$\mathbf{v}$	$\mathcal{O}\left(pn^d\right)$	$\mathcal{O}\left((d-1) n r_X^2 r_C^2 + n p r_X r_C\right)$
$\hat{\mathcal{X}}_{k\|k}$	$\mathcal{O}\left(pn^d\right)$	$\mathcal{O}\left(n p r_{K_1}\right)$
$\mathcal{P}_{k\|k}$	$\mathcal{O}\left(pn^{2d} + p^2 n^d\right)$	$\mathcal{O}\left((d-1) n^2 r_K^4 + n^2 p r_K^2 + n p^2 r_K\right)$
$\hat{\mathcal{X}}_{k+1\|k}$	$\mathcal{O}\left(n^{2d}\right)$	$\mathcal{O}\left(d n^2 r_X^2 r_A^2\right)$
$\mathcal{P}_{k+1\|k}$	$\mathcal{O}\left(n^{3d}\right)$	$\mathcal{O}\left(d n^3 r_P^2 r_A^4\right)$

The TN ranks of the system dynamics r_A, r_C are inherently given and determine subsequently the TN ranks of the variables in the filter. Fig. 3 shows the converged maximum TN rank of TT variables in the MIMO TN Kalman filter using $p = n^d$ for increasing sizes of random systems with $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 5$. The TN rank choice is made to ensure a higher TN rank than unity for the system dynamics while keeping it low enough to obtain feasible computation times. The TN cores are generated randomly using the `tt_rand(·)` command from [14]. Rounding tolerance is set to $\epsilon = 10^{-6}$ using 5 Monte Carlo simulations with 50 time steps to ensure convergence. The figure highlights that the TN ranks of the measurement and time update covariance tensor are driving since they grow exponentially with the system size. This yields a tremendous increase in computation time for large systems if no counter measure is done.

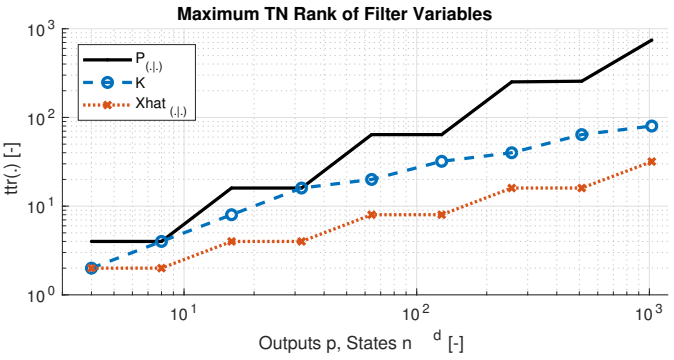


Fig. 3. Maximum TN rank of variables in MIMO TN Kalman filter for increasing system size with $n^d = p$ using $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 5$.

B. Extension for Reduction of Complexity

The variables in the analysis of Fig. 3 do have optimal TN ranks according to the selected rounding tolerance ϵ. The main idea of this paper is to obtain low TN rank approximations of the driving factors, the covariance tensors, by choosing lower TN ranks. This decreases the effect of polynomial computational complexity.

The truncation of TN ranks in the covariance tensor to lower values is based on the idea of the rounding procedure [13]. After the orthogonalization step, a δ-truncated SVD is executed which determines the TN rank. The truncation

threshold δ is computed with Eq. (1) using the rounding tolerance ϵ. However, the choice for the truncation threshold is dependent on the distribution of the singular values. In practice, it is often difficult to choose the tolerance ϵ for the SVD truncation [5]. One approach is to use a fixed truncation value, see e.g. [10]. If the TN rank is truncated with a fixed $r \leq \delta$, only the r most dominant singular values of each TN core are taken into account. Therefore, the TN rounding function is adapted such that the SVD truncation threshold is chosen as

$$\min(r, \delta)$$

in order to obtain the lowest possible truncation threshold. The consequences of a low TN rank approximation in the TN Kalman filter are in general:

- Reduction in computation time. This holds since a lower TN rank of the covariance tensors will yield lower TN ranks of subsequently computed variables and decreases their influence on the computation time.
- Lower accuracy of the resulting estimation since a low rank approximation is used.

Similar analysis as presented in Sec. IV has shown that in case of truncating the TN rank of the second most driving parameter, the Kalman gain, that the filter even tends to diverge. Also, it is not useful to truncate the TN rank of the estimated state since this is an output of the filter and should be kept with the desired accuracy as chosen by the rounding tolerance. Moreover, its influence is according to Fig. 3 less driving than the covariance or Kalman gain. Note that truncation of the covariances $\mathcal{P}_{k|k}$ and $\mathcal{P}_{k+1|k}$ is done online at each time step.

IV. Simulation Results and Discussion

Both the SISO and the MIMO filter can easily be extended with the proposed approach. The power of the truncation method is shown in simulation. Different random stable LTI systems are generated with the state vector size n^d and output size $p = 1$ for SISO and $p = n^d$ for MIMO systems. For simplicity, a mode size of $n = 2$ is chosen as smallest prime factor equally to [17]. The system dynamics are in TT-format described with $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 5$, which is a generalization compared to TN rank unity. The TN rank of the process noise covariance tensor is given by $\mathrm{ttr}(\mathcal{Q}) = 1$. For the assumption of the matrix $\mathbf{Q}$ being diagonal, this holds. The entries of the diagonal process noise covariance are set to 0.1. The measurement noise covariance matrix $\mathbf{R}$ is diagonal with all entries at 0.5. The rounding tolerance is $\epsilon = 10^{-6}$, and a total of 15 Monte Carlo simulations are run over 25 time steps.

The simulation is run on Matlab version 9.3.0.713579 (R2017b) installed on Linux Ubuntu 16.04 LTS making use of the TT toolbox [14], Tensorlab v3.0 [18] and functions provided with the code of [2], [3]. The hardware consists of an Intel core i5-7200 quad-core CPU running at 2.5 GHz with 7.7 GB RAM.

A. SISO Kalman Filter

A comparison between the conventional Kalman filter and the SISO TN Kalman filter with the mentioned properties is presented in Fig. 4. For the TN Kalman filter, three settings are simulated: without truncation of the covariance tensor TN ranks as well as with truncation at $\mathrm{ttr}(\mathcal{P}_{(\cdot|\cdot)}) = \{1, 5\}$. The figure shows the computation time per time step of each filter; the mean computation time over 15 Monte Carlo simulations is depicted. The variance of the computation time is negligible with maximum magnitude of 10^{-1}.

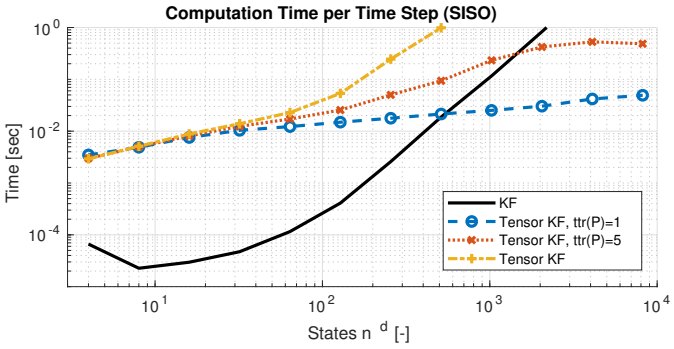


Fig. 4. Computation time per time step of random stable LTI SISO system for conventional and TN Kalman filter with TN rank $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 5$ using covariance TN rank truncation.

The results for the SISO filter yield two main points. First, a comparison of the output estimation of the TN Kalman filter with and without covariance TN truncation and the conventional Kalman filter is done. The relative 2-Norm squared of the outputs is used for this comparison defined as

$$\frac{||\mathrm{vec}(\mathbf{y}) - \mathrm{vec}(\hat{\mathbf{y}})||_2^2}{||\mathrm{vec}(\mathbf{y})||_2^2},$$

where $\mathrm{vec}(\mathbf{y})$ is the vectorized output of the simulated LTI system and $\mathrm{vec}(\hat{\mathbf{y}})$ is the vectorized estimated output of the respective Kalman filter. The difference between the relative 2-Norm squared of the conventional Kalman filter and the TN Kalman filter is negligibly small and, therefore, the estimation is accurate even with covariance TN rank truncation. This yields that the approximation of the covariance tensor with lower TN rank is still sufficiently accurate such that the overall error stays small.

Second, the dash-dotted line with '+' marker in Fig. 4 shows the computation time of the state of the art for systems with $\mathrm{ttr}(\mathcal{A}) > 1$, $\mathrm{ttr}(\mathcal{C}) > 1$. The computation time grows exponentially and is even more than two magnitudes higher than for the conventional Kalman filter. Using the developed approach to truncate the TN rank of the covariance tensor yields a great improvement. With truncation at $\mathrm{ttr}(\mathcal{P}) = 1$, the computation time becomes linear - similar to the results for $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 1$ in Fig. 1. For large scale systems with $n^d > 600$ states, this approach yields accurate and significantly faster estimation results.

B. MIMO Kalman Filter

For the MIMO Kalman filter, simulations are run with the same setting using $p = n^d$, meaning the number of outputs is equal to the number of states. The resulting computation times per time step are depicted in Fig. 5 using the mean of 15 Monte Carlo simulations.

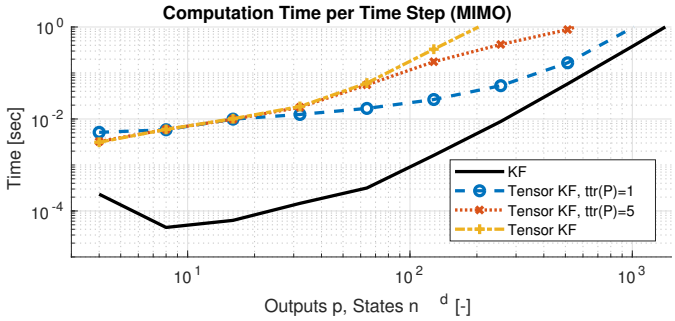


Fig. 5. Computation time per time step of random stable LTI MIMO system for conventional and TN Kalman filter with TN rank $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 5$ using covariance TN rank truncation.

Similar to the results of the SISO filter, the MIMO TN Kalman filter has the same relative 2-Norm of the output with and without covariance TN rank truncation and even with the conventional Kalman filter. This confirms the optimality of the TN Kalman filter. Regarding the computation time, the state of the art TN Kalman filter for $\mathrm{ttr}(\mathcal{A}) > 1$, $\mathrm{ttr}(\mathcal{C}) > 1$ is given by the dash-dotted line in Fig. 5. This depicts an exponential increase of computation time being at least two magnitudes slower than the conventional Kalman filter. The TN Kalman filter with $\mathrm{ttr}(\mathcal{P}) = 1$ truncation converges for large n^d to the one of the conventional Kalman filter. This is reasonable since both filters have a complexity of $\mathcal{O}\left(p^3\right)$ which is not tackled in this paper. Hence, for the MIMO case the approach of covariance TN rank truncation yields an improvement in computation time over the TN Kalman filter without it.

Remark: Note that in all simulations the covariance matrix remained symmetric with $\mathcal{P}_{\cdot|\cdot} - \mathcal{P}_{\cdot|\cdot}^{\top}$ at machine precision level. Moreover, $\mathcal{P}_{k+1|k}$ remained postive definite, while $\mathcal{P}_{k|k}$ only had the smallest eigenvalue larger than $-\mathrm{eps}$.

V. Conclusion

The paper discusses the need for improvement of TN Kalman filter for general LTI systems with TN rank $\mathrm{ttr}(\mathcal{A}) > 1$, $\mathrm{ttr}(\mathcal{C}) > 1$ due to computation time issues. Therefore, the concept of TNs in TTs and the TN Kalman filter for the SISO and MIMO case is explained in detail. An analysis of the state of the art shows a polynomial dependency on the computational complexity of the TN ranks decreasing the computational time for large TN ranks. The driving variable with the highest TN ranks within the Kalman filter is identified to be the covariance matrix. Truncating its TN ranks by considering only the most dominant singular values is verified in simulation to yield accurate and computationally fast results. This approach is applied to both the SISO and MIMO case, improving the computation time for general LTI systems.

The complexity analysis and results for the MIMO TN Kalman filter highlights the output dependency of $\mathcal{O}\left(p^3\right)$. This limits the MIMO filter to the same speed as the conventional Kalman filter. Future work concentrates on the reduction of complexity in the number of outputs. Moreover, a square-root implementation of the TN Kalman filter is desired for practical and numerical problems. This requires the computation of a Cholesky or QR factorization in TT format, which is to the knowledge of the authors not yet efficiently solved.

References

[1] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial applications of the Kalman filter: A review," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5458–5471, Dec 2013.

[2] K. Batselier, Z. Chen, and N. Wong, "A Tensor Network Kalman filter with an application in recursive MIMO Volterra system identification," *Automatica*, vol. 84, 10 2016.

[3] K. Batselier and N. Wong, "Matrix output extension of the Tensor Network Kalman filter with an application in MIMO Volterra system identification," *Automatica*, vol. 95, pp. 413–418, 2018.

[4] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, Sep 1970. [Online]. Available: https://doi.org/10.1007/BF02310791

[5] D. Chicco and M. Masseroli, "A discrete optimization approach for SVD best truncation choice based on ROC curves," in *13th IEEE International Conference on BioInformatics and BioEngineering*, Nov 2013, pp. 1–4.

[6] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor Networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," *Foundations and Trends in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016. [Online]. Available: http://dx.doi.org/10.1561/2200000059

[7] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, Jun 2010.

[8] S. Holtz, T. Rohwedder, and R. Schneider, "The alternating linear scheme for tensor optimization in the Tensor Train format," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. 683–713, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1137/100818893

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] P. Khatri, B. Done, A. Rao, A. Done, and S. Draghici, "A semantic analysis of the annotations of the human genome," *Bioinformatics*, vol. 21 16, pp. 3416–21, 2005.

[11] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. [Online]. Available: https://doi.org/10.1137/07070111X

[12] R. Orus, "A practical introduction to Tensor Networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117 – 158, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003491614001596

[13] I. Oseledets, "Tensor-Train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011. [Online]. Available: https://doi.org/10.1137/090752286

[14] I. Oseledets, S. Dolgov, V. Kazeev, O. Savostyanov, D. Lebedeva, P. Zhlobich, T. Mach, and L. Song, "TT-toolbox," *Available online*, 2012. [Online]. Available: https://github.com/oseledets/TT-Toolbox

[15] I. Oseledets and E. Tyrtyshnikov, "Breaking the curse of dimensionality, or how to use SVD in many dimensions," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3744–3759, 2009. [Online]. Available: https://doi.org/10.1137/090748330

[16] ——, "TT-cross approximation for multidimensional arrays," *Linear Algebra Appl.*, vol. 432, no. 1, pp. 70–88, 2010.

[17] I. Oseledets, "Approximation of 2d*2d matrices using tensor decomposition," *SIAM J. Matrix Analysis Applications*, vol. 31, pp. 2130–2145, 2010.

[18] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v3.0," *Available online*, Mar 2016. [Online]. Available: http://www.tensorlab.net

Chapter 5

IEEE Paper

The paper for EUSIPCO in chapter 4 mentions in the conclusion several open points for future research. In this chapter a second paper is presented which addresses one of the main issues for use of the tensor Kalman filter in applications. Namely to design a fast MIMO tensor Kalman filter. This is specifically visible in the results of Fig. 5 of the EUSIPCO paper: The method of covariance truncation does only yield a MIMO tensor Kalman filter which is for extremely large systems as fast as the conventional Kalman filter and for small-scale systems significantly slower.

The main contributions of this paper is to solve the MIMO tensor Kalman filter problem for a specific class of large-scale systems. This class uses uncorrelated, low measurement noise and has a TT-rank of one in the measurement matrix. If this holds and the method of covariance truncation is applied, the inverse in the DARE can be solved extremely fast. Furthermore, a method is developed for the conversion of vectors to TTs based on the TT-ALS scheme, which outperforms the state-of-the-art TT-SVD conversion while obtaining machine precision conversion. This is used for the recursive conversion of the measurement vector in TT-format as it is used in the tensor Kalman filter. Finally, the novel MIMO tensor Kalman filter is verified on the application of AOs. The system equations are rewritten for the required special class of systems in the MIMO tensor Kalman filter. A derivation is given such that the Shack-Hartmann sensor can be written as a TT with TT-rank of one for a square aperture. The results show that the novel filter method outperforms multiple state-of-the-art algorithms by several orders in means of computational time. This comes at the cost of a lower accuracy due to the low TT-rank approximations.

The printout in the following is a first draft including first internal feedback as it will be submitted to the IEEE transactions on control technology journal. This is one of top journals of the IEEE control systems society which is published bimonthly with the goal to close the gap between theory and applications in the domain of control engineering, which makes it the top choice for the presented contribution.

MIMO Tensor Network Kalman Filter With Application to Adaptive Optics

Daniel Gedon, Pieter Piscaer, Kim Batselier, Michel Verhaegen

***Abstract*—The Kalman filter is an optimal observer for linear dynamic systems which is in practice for many large-scale real-time application infeasible due to its computational complexity. This paper presents a Multiple-Input Multiple-Output (MIMO) Tensor Network (TN) Kalman filter for large-scale linear dynamic systems exploiting TNs to obtain a fast solution for state-space system with a specific structure in the output equation. The main point is to rewrite the Kalman filter equations in tensor form using the Tensor Train (TT) decomposition. Approximation methods are employed to obtain a computational complexity that has a linear trend with increasing system size. The power of the approach is demonstrated for the real-life example of Adaptive Optics (AO), where the next generation of extremely large telescopes have tens of thousands of measurements and require real-time estimation of the same order of states.**

***Index Terms*—Kalman filter, dynamic systems, tensors, tensor train, large-scale systems, MIMO, curse of dimensionality, adaptive optics.**

I. Introduction

KALMAN filtering yields an unbiased, minimum variance estimation for linear dynamic systems [20]. This property makes it the method of choice for various applications, e.g. [1], [2], [8], [15]. In practice the theory is limited to small-scale problems due to computational reasons. The solution of the Kalman filtering problem requires to solve the associated Discrete Algebraic Riccati Equation (DARE) [19]. Consider an exponentially large system with state vector size n^d. The solution of the DARE and the subsequent computation of the Kalman gain require $81n^{3d}$ algebraic operations [24]. For time-invariant systems it is necessary to solve the DARE once for the steady-state Kalman filter gain. In this paper, the more general time-varying case is considered. Therefore, in each time-step the DARE needs to be solved. For real-time applications this becomes already for small n, d infeasible. However, many large-scale problems [26] arise such as in the field of AO [34], power grids [18] and traffic systems [22], which can benefit from large-scale optimal estimation.

Different methods have been developed to address this issue. Distributed Kalman filters [3], [25], sparse methods [7], [12], Kalman filter based on an approximation approach of the DARE [24] or ensemble Kalman filters [14], [35] are just a few special algorithms. All these methods have the drawback of having an exponential computational complexity and hence suffer from the curse of dimensionality.

The authors are with the Delft Center for Systems and Control, Technische Universiteit Delft, Delft 2628, The Netherlands (e-mails: daniel.gedon@gmx.de; p.j.piscaer@tudelft.nl; k.batselier@tudelft.nl; m.verhaegen@tudelft.nl).

Manuscript received XX 2019; revised XX 201X.

Despite that limitation, the latter two methods are of special interest. First, the Riccati approximation method, which is a first-order Taylor approximation, has been shown to outperform several other methods [24] regarding its computational complexity. Theoretically, compared to the full DARE a reduction of a factor of about 10 is achieved. However, it is limited to a special block structure with invertability condition. For the used AO example in [24] it is restricted to a diagonal $\mathbf{A}-$matrix and cannot work with more general identified systems. The second method, the ensemble Kalman filter does not use the covariance matrix but a random sample of an extended state vector to m ensemble realizations. This method has been effectively applied to AO and shown to have a theoretical computational complexity in the order of $\mathcal{O}\left(mn^{2d}\right)$ algebraic operations [14]. This is a reduction by one exponential factor, however the number of ensembles has to be chosen rather large, often in order of $n^d/2$ for a reasonable accuracy.

A mathematical method to lift the curse of dimensionality for storage of the matrices and computational operations is by use of tensors with the TT decomposition [29]. This can be exploited for large-scale systems. A TN Kalman filter is developed in [4], [5] for single output systems for the system identification of MIMO Volterra systems. In [13] the tensor Kalman filter is presented for LTI state-space systems. This method is not free of limitations. It is only fast for a specific class of systems, namely such with the property of low TT-ranks. An approximation solution for a more general approach than unity TT-rank is presented in [13].

This paper uses the methods of [13] and extends it to provide a solution for the unsolved MIMO tensor Kalman filter problem. The contribution of this paper can be separated in two main points.

1) Extension of the framework of the tensor Kalman filter for MIMO linear estimation. Methods are developed for a super fast inversion in the Riccati equation under some assumptions and for a fast transformation of measurement vector to TT-format, outperforming the state-of-the-art TT-SVD [29] algorithm.
2) The developed tensor Kalman filter is verified in a real-life examples at the application of AO. Regarding the computational complexity it exceeds the conventional filter methods by about four orders for this specific system.

The remainder of the paper is structured as follows. Section II provides preliminary mathematical concepts of tensors and the TT-decompositions. Focus lies on the visualization

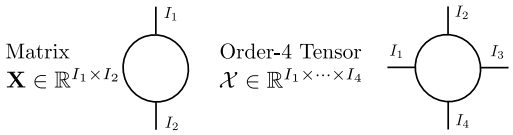


Fig. 1. TN diagram of a matrix and and order-4 tensor with TN core as circle and free lines as indices.

in TN diagrams for intuitive understanding of tensor algorithms. In section III the novel MIMO tensor Kalman filter method is derived and demonstrated. An introduction to AO with the derived linear dynamic model is given in section IV for the comparison with a real-life application. The results of the comparison are evaluated in section V and concluding remarks about future research and open points is given in section VI. An open-source Matlab implementation of the algorithm is available from https://github.com/dgedon/Tensor-Kalman-Filter.

II. Preliminaries

Tensors are multidimensional arrays as a generalization of vectors and matrices to higher order. Each entry of an order-d tensor is given by d indices. Within this paper scalars are denoted by Roman letters $x \in \mathbb{R}$, vectors as bold letters $\mathbf{x} \in \mathbb{R}^{I_1}$ and matrices as capital bold letters $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. Tensors of order-$d$ are given as capital calligraphic letters $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$. Referring in the following to the ith mode of a tensor is equal to referring to the ith index. For example, mode one of the latter defined tensor $\mathcal{X}$ has size I_1. Useful introductions to tensors, tensor operations and their decompositions can be found in [9], [21].

A. Tensor Basics

A simple and intuitive visualization for higher order tensors is by means of Tensor Network (TN) diagrams. An introduction to TNs and their visualization is given in [27]. These diagrams consist of TN cores, depicted as circles and free lines connected to the core. The number of free lines indicates the order of the tensor; each line is indexed with its size in that dimension. An example is given in Fig. 1 for a matrix $\mathbf{X}$ and an order-4 tensor $\mathcal{X}$. TN diagrams are specifically useful to visualize multilinear operations. One of these is the higher order equivalent to the multiplication, namely the mode-m,n product.

Definition 2.1: Mode-m,n product (tensor contraction) [9]. Take two tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$, $\mathcal{B} \in \mathbb{R}^{J_1 \times \cdots \times J_k}$ of order-d and order-k with common mode $I_n = J_m$. Contracting them in their common mode will yield an order-$(d+k-2)$ tensor $\mathcal{C}$. The contraction is given by

$$\mathcal{C} = \mathcal{A} \times_n^m \mathcal{B} \tag{1}$$

The operator indicates the contraction of mode-n of $\mathcal{A}$ with mode-m of $\mathcal{B}$. The entries of $\mathcal{C}$ are computed by element-wise summations in the following way

$$c_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_d,j_1,\ldots,j_{m-1},j_{m+1}\ldots,j_k} = \\ = \sum_{i_n=1}^{I_n} a_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_d} b_{j_1,\ldots,j_{m-1},j_n,j_{m+1},\ldots,j_k} \tag{2}$$

In a TN diagram this is visualized by connection of two free lines indexed with the same size. Note that the mode-m,n product is in general not associative or commutative [9]. Within this paper in all cases of multiple tensor contractions, the operation is associative. The tensor contraction alone does not lift the curse of dimensionality. Specific tensor decompositions have to be used in combination with multilinear algebra.

B. Tensor Train Decomposition

The Tensor Train (TT) decomposition [29] decomposes an order-d tensor $\mathcal{X}$ in d order-3 TN cores $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(d)}$ which are interconnected in a line network. Each TN core has dimensions $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$. The interconnecting indices between the cores r_i are called TT-ranks and play a crucial rule in the computational complexity. The maximum TT-rank over all r_i is defined as $r = \operatorname{ttr}(\mathcal{X})$. Boundary conditions are defined as $r_0 = r_d = 1$. Using the tensor contraction, the TT-decomposition is mathematically described as

$$\mathcal{X} = \mathcal{X}^{(1)} \times_3^1 \cdots \times_3^1 \mathcal{X}^{(d)} \tag{3}$$

This equation indicates the connection and possible contraction of mode-3 of TN core $\mathcal{X}^{(i)}$ with mode-1 of TN core $\mathcal{X}^{(i+1)}$. These modes are the TT-ranks. Based from nomenclature in quantum physics a TT is often called Matrix Product State (MPS). After full contraction of all TN cores the order-d tensor $\mathcal{X}$ can be reshaped in an exponentially large vector $\mathbf{x} \in \mathbb{R}^{n^d}$. Similarly, an exponentially large matrix $\mathbf{Y} \in \mathbb{R}^{m^d \times n^d}$ can be depicted in TT-format as TT-matrix with d order-4 TN cores of size $\mathcal{Y}^{(i)} \in \mathbb{R}^{r_{i-1} \times m \times n \times r_i}$ yielding

$$\mathcal{Y} = \mathcal{Y}^{(1)} \times_4^1 \cdots \times_4^1 \mathcal{Y}^{(d)} \tag{4}$$

A TT-matrix is also called Matrix Product Operator (MPO). The collection of TN cores is denoted by

$$\begin{aligned} \operatorname{TT}(\mathcal{X}) &= \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(d)}\} \\ \operatorname{TT}(\mathcal{Y}) &= \{\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(d)}\} \end{aligned} \tag{5}$$

In the following it will be clear by the order of the TN cores if it is a TT or a TT-matrix. Both, the TT and TT-matrix are visualized as TN diagram in Fig. 2.

An important property for the numeric stability of numerous TT algorithms is orthogonality which is defined in this framework as left- or right-orthogonality of single TN cores.

Definition 2.2: Left- and Right-Orthogonal TN cores [17]. Take a TN core $\mathcal{X}^{(i)} \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$. It is said to be left-orthogonal if it can be reshaped in a matrix $\mathbf{X} \in \mathbb{R}^{r_{i-1} n_i \times r_i}$ and it holds

$$\mathbf{X}^\top \mathbf{X} = \mathbf{I}_{r_i} \tag{6}$$

Similarly, the TN core is said to be right-orthogonal if it can be reshaped in a matrix $\mathbf{X} \in \mathbb{R}^{r_{i-1} \times n_i r_i}$ and it holds

$$\mathbf{X}\mathbf{X}^\top = \mathbf{I}_{r_{i-1}} \tag{7}$$

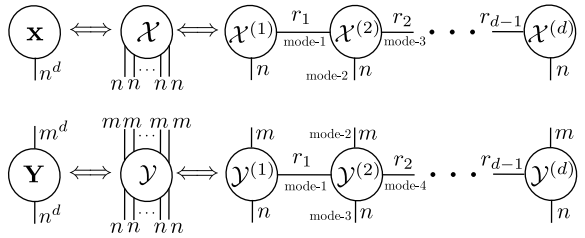


Fig. 2. TN diagrams of vector to TT (top) and of matrix to TT-matrix (bottom). Both with indication of mode numbering.

It is important to note that a TT-model is closely related to models of Kronecker products.

Definition 2.3: Kronecker product [37]. Take two matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{B} \in \mathbb{R}^{J_1 \times J_2}$. The Kronecker product is defined as

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,i_2}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{i_1,1}\mathbf{B} & \dots & a_{i_1,i_2}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{I_1 J_1 \times I_2 J_2} \quad (8)$$

If all TT-ranks are equal to one then the TT-representation in Eq. (3) is equal to a Kronecker model defined as

$$\mathbf{X} = \mathbf{X}^{(d)} \otimes \cdots \otimes \mathbf{X}^{(1)} \quad (9)$$

This results because first, for a TT with TT-rank one, all TN cores are of size $\mathcal{X}^{(i)} \in \mathbb{R}^{1 \times n_i \times n_i \times 1}$, which can equally be squeezed to matrices. Hence, the mode-m,n product becomes its lower order equivalent, namely the outer product. Second, The Kronecker product and the outer product are interrelated by reshaping and permutations. Similar results hold for higher TT-ranks and summation of Kronecker models. Note the reversed order of the TN cores in Eq. (3) compared to the Kronecker model. Note also, this only holds if all TT-ranks are equal. Hence, the TT-format is more general than Kronecker models. An extensive overview on Kronecker models for large-scale systems is given in [36].

An example for the contraction of two TT-matrices will highlight the advantage of the TT-decomposition. See Fig. 3 for a visualization of this contraction.

Example 2.1: Take two TT-matrices with TN cores as $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{i-1} \times m \times n \times r_i}$ and $\mathcal{B}^{(i)} \in \mathbb{R}^{r_{i-1} \times n \times m \times r_i}$. The multilinear contraction is done core-wise according to

$$\mathcal{C}^{(i)} = \mathcal{A}^{(i)} \times_3^2 \mathcal{B}^{(i)} \quad (10)$$

for $i = 1, \dots, d$. The computational complexity of this operation is given by $\mathcal{O}\left(dnm^2r^4\right)$, meaning linear in d. See Eq. (2) for each TN core to obtain the complexity.
These TT-matrices are equivalent representations of exponentially large matrices $\mathbf{A} \in \mathbb{R}^{m^d \times n^d}$,$\mathbf{B} \in \mathbb{R}^{n^d \times m^d}$. Hence, a comparison with the matrix multiplication is drawn. The multiplication $\mathbf{C} = \mathbf{AB}$ requires $\mathcal{O}\left(n^d m^{2d}\right)$ algebraic operations. This highlights that there is a tremendous computational advantage with TTs in case the TT-ranks are small, e.g. for Kronecker models.

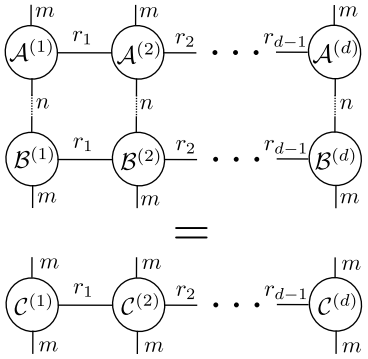


Fig. 3. Contraction of two TTs $\mathcal{A}, \mathcal{B}$ in their common modes to a new TT $\mathcal{C}$

III. MIMO Tensor Kalman Filter Algorithm

Consider a linear time-varying discrete-time state-space system of the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k \mathbf{x}_k + \mathbf{v}_k \\ \mathbf{y}_k &= \mathbf{C}_k \mathbf{x}_k + \mathbf{w}_k \end{aligned} \quad (11)$$

with Gaussian noise covariances $\mathbf{v}_k \sim (0, \mathbf{Q}_k)$, $\mathbf{w}_k \sim (0, \mathbf{R}_k)$. All variables are exponentially large with state vector size $\mathbf{x} \in \mathbb{R}^{n^d}$ and output size $\mathbf{y} \in \mathbb{R}^{p^d}$. For the following algorithm all variables are considered to be in TT-format. This yields the sizes of each TN cores as

- $\mathcal{X}_k$ with $\mathcal{X}_k^{(i)} \in \mathbb{R}^{r_{X_{i-1}} \times n \times r_{X_i}}$
- $\mathcal{A}_k$ with $\mathcal{A}_k^{(i)} \in \mathbb{R}^{r_{A_{i-1}} \times n \times n \times r_{A_i}}$
- $\mathcal{Q}_k$ with $\mathcal{Q}_k^{(i)} \in \mathbb{R}^{r_{Q_{i-1}} \times n \times n \times r_{Q_i}}$
- $\mathcal{Y}_k$ with $\mathcal{Y}_k^{(i)} \in \mathbb{R}^{r_{Y_{i-1}} \times p \times r_{Y_i}}$
- $\mathcal{C}_k$ with $\mathcal{C}_k^{(i)} \in \mathbb{R}^{r_{C_{i-1}} \times p \times n \times r_{C_i}}$
- $\mathcal{R}_k$ with $\mathcal{R}_k^{(i)} \in \mathbb{R}^{r_{R_{i-1}} \times p \times p \times r_{R_i}}$

for $i = 1, \dots, d$.

A. General Algorithm

The derived algorithm in the following is a generalization and extension of [4], [5] to MIMO state-space systems by distributing the output dimension over all TN cores to lift the curse of dimensionality. Three main steps are necessary to design a tensor Kalman filter algorithm

1) Convert the system matrices $\mathbf{A}_k, \mathbf{C}_k, \mathbf{Q}_k, \mathbf{R}_k$ in TT-format.
2) Rewrite all variables in the Kalman filter in TT-format, i.e. $\hat{\mathbf{x}}_{(\cdot|\cdot)}, \mathbf{P}_{(\cdot|\cdot)}, \mathbf{K}_k, \mathbf{y}_k$.
3) Rewrite the conventional Kalman filter equations with multilinear operations for TTs.

For the first step the TT-SVD [29] or TT-cross algorithm [31] can be used. Step two is done for the state estimate and the covariance by initialization with best guesses as zero vector and diagonal matrix. The Kalman gain results by computations as TT. The following sizes of the TTs are considered

- $\mathcal{P}_{(\cdot|\cdot)}$ with $\mathcal{P}_{(\cdot|\cdot)}^{(i)} \in \mathbb{R}^{r_{P_{i-1}} \times n \times n \times r_{P_i}}$
- $\mathcal{K}_k$ with $\mathcal{K}_k^{(i)} \in \mathbb{R}^{r_{K_{i-1}} \times n \times p \times r_{K_i}}$

The recursive conversion of the measurement vector $\mathbf{y}_k$ to TT in each time step is elaborated in subsection III-D. The third step is split up for the measurement update

$$\mathcal{S}_k^{(i)} = \mathcal{C}_k^{(i)} \times_3^2 \mathcal{P}_{k|k-1}^{(i)} \times_3^3 \mathcal{C}_k^{(i)} + \mathcal{R}_k^{(i)} \tag{12}$$

$$\mathrm{TT}\left(\tilde{\mathcal{S}}_k\right) = \mathrm{TT}\left(\mathcal{S}_k\right)^{-1} \tag{13}$$

$$\mathcal{K}_k^{(i)} = \mathcal{P}_{k|k-1}^{(i)} \times_3^3 \mathcal{C}_k^{(i)} \times_3^2 \tilde{\mathcal{S}}_k^{(i)} \tag{14}$$

$$\hat{\mathcal{X}}_{k|k}^{(i)} = \hat{\mathcal{X}}_{k|k-1}^{(i)} + \mathcal{K}_k^{(i)} \times_3^2 \left(\mathcal{Y}_k^{(i)} - \mathcal{C}_k^{(i)} \times_3^2 \hat{\mathcal{X}}_{k|k-1}^{(i)}\right) \tag{15}$$

$$\mathcal{P}_{k|k}^{(i)} = \mathcal{P}_{k|k-1}^{(i)} - \mathcal{K}_k^{(i)} \times_3^2 \mathcal{C}_k^{(i)} \times_3^2 \mathcal{P}_{k|k-1}^{(i)} \tag{16}$$

and similarly for the time update

$$\hat{\mathcal{X}}_{k|k}^{(i)} = \mathcal{A}_k^{(i)} \times_3^2 \hat{\mathcal{X}}_{k|k}^{(i)} \tag{17}$$

$$\mathcal{P}_{k+1|k}^{(i)} = \mathcal{A}_k^{(i)} \times_3^2 \mathcal{P}_{k|k}^{(i)} \times_3^3 \mathcal{A}_k^{(i)} + \mathcal{Q}_k^{(i)} \tag{18}$$

All operations are core-wise for $i = 1, \ldots, d$. This set of equation is a simple generalization of the Kalman filter equations [20] to higher order spaces with the mode-m,n product. An example will highlight this remark.

Example 3.1: Consider the first summand of Eq. (12) with all TTs with TT-rank one. Then, the contractions reduce to $\mathbf{C}_k^{(i)} \mathbf{P}_{k|k-1}^{(i)} (\mathbf{C}_k^{(i)})^\top$ for all TN cores. This is equal to the conventional Kalman filter when using Kronecker models of the form of Eq. (9).

However, there are three main problems with the tensor algorithm which prohibit a fast computation.

Problem 3.1: High TT-ranks: The computational complexity of the tensor contraction is polynomially dependent on the TT-ranks. These are an inherent property of the system matrices and can easily grow high and slow down the computations.

Problem 3.2: Inversion of $\mathcal{S}_k$ in Eq. (13): For the TT-format there is no general fast inversion algorithm available.

Problem 3.3: Recursive conversion of the measurement vector to TT-format: The TT-SVD algorithm can be slow for large vectors due to the repeated SVD. Using this in every time step can be a driving factor.

Solutions to these problems are presented in following subsections independently.

B. Method: Truncation of Covariance TT-Rank

This method gives a solution to problem 3.1 to reduce the effect of the TT-ranks on the computational complexity. The method is originally presented in [13] and hereafter summarized.

The system matrices in TT-format can only be approximated reliably up to certain degree with lower TT-ranks. Hence, the variables with the highest TT-ranks within the Kalman filter have to be identified as driving factors. This is determined to be the covariance tensors. A low TT-rank approximation is desired. The basis of the method is the TT-rounding algorithm [29] where a truncation of singular values for each TN core above a certain threshold is performed. Therefore, the TN cores are matricified to $\mathbf{X}^{(i)} \in \mathbb{R}^{r_{i-1} n^2 \times r_i}$. Similarly, in the covariance truncation method only the first, most dominant singular value is considered yielding a TT with TT-rank one. It is shown that this approximation method reduces the effect of large TT-ranks on the computational complexity and can speed up more general systems than for system matrices with TT-rank of one.

C. Method: Fast Inversion in Riccati Equation

This method yields a solution for a special case of Problem 3.2. For the efficiency of following contractions after the inversion (e.g. to compute the Kalman gain) it is important to have an idea about the TT-ranks of an inverted TT.

Theorem 3.1: Consider a nonsingular TT $\mathcal{X}$ with $\mathrm{ttr}\left(\mathcal{X}\right) = 1$. Then [32]

$$\mathrm{ttr}\left(\mathcal{X}^{-1}\right) = 1 \tag{19}$$

This holds due to the relation of a TT with Kronecker models and by [37]

$$\left(\mathbf{X}_1 \otimes \cdots \otimes \mathbf{X}_d\right)^{-1} = \mathbf{X}_1^{-1} \otimes \cdots \otimes \mathbf{X}_d^{-1} \tag{20}$$

For unstructured TTs with $\mathrm{ttr}\left(\mathcal{X}\right) > 1$ there is no such result, meaning that the TT-rank of the inverse can grow large [32].

Multiple possibilities exist for the inversion of a TT. For example one can solve of a linear system of the form

$$\mathbf{A}\mathbf{X} = \mathbf{I} \tag{21}$$

with $\mathbf{X} = \mathbf{A}^{-1}$ as solution of the system. Rewriting it in a regularized minimization problem yields

$$\begin{aligned} &\min_{\mathbf{X}} \left|\left|\mathbf{I} - \mathbf{A}\mathbf{X}\right|\right|_F^2 + \lambda \left|\left|\mathbf{X}\right|\right|_F^2 \\ &\text{s.t.} \quad \lambda \geq 0 \end{aligned} \tag{22}$$

This optimization can be solved using the Alternating Linear Scheme (ALS) for tensor optimization [17]. A modified version of the ALS is originally developed in quantum physics as Density Matrix Renormalization Group (DMRG) [38]. This is an iterative optimization scheme where one TN core is optimized at a time. Hence, smaller subproblems are solved instead of one large problem. For numeric stability all cores are orthogonalized except the one to be optimized. One sweep is defined as going forward and backward over all TN cores. The scheme applies multiple sweeps until convergence.

The ALS approach has been chosen in [28], [30]. In [23] it is generalized for the pseudo-inverse of non-square matrices. However, the ALS requires to know the TT-ranks of the solution beforehand. According to Theorem 3.1 the solution ranks are only known for $\mathrm{ttr}\left(\mathcal{A}\right) = 1$. The solution TT-ranks could be initialized with high values and then truncated to the optimal ones using the TT-rounding procedure [29]. However, then contractions within the optimization are computationally inefficient.

A second possibility is the use of randomized SVDs. The idea is to use a multiplication with a random matrix to reduce the size of the problem while obtaining a random sample of the range of the matrix. An overview on different algorithms is given in [16]. A TT implementation is proposed in [6]. Inversion is simply achieved by transposing the single matrices and re-contracting them. However, this approach has two main drawbacks. First, for a good approximation the distribution

of the singular values has to be known which is in general not the case. Second, The TT-ranks of the decomposed TTs can be high which lead to inefficient re-contraction after decomposition.

Hence, a general fast solution for the inversion of a TT is not yet available. For the special case of $\mathrm{ttr}\left(\mathcal{C}\right)=1$ and uncorrelated, low measurement noise, there exists a fast inversion. Considering the result of [13] for covariance truncation to $\mathrm{ttr}(\mathcal{P}_{(\cdot|\cdot)})=1$ yields $\mathrm{ttr}\left(\mathcal{S}\right)=2$ due to the summation of two TT-rank one TTs [29] in Eq. (12). If the measurement noise is sufficiently low, then the TT-rounding algorithm will only consider the most dominant singular value, hence $\mathrm{ttr}\left(\mathcal{S}\right)=1$. Then, Eq. (20) states that each TN core can be inverted independently.

Example 3.2: Consider a TT $\mathcal{X}$ with TT-rank one and $\mathcal{X}^{(i)}\in\mathbb{R}^{1\times n\times n\times 1}$. The complete inversion requires $\mathcal{O}\left(dn^3\right)$ operations. For an equal matrix representation with $\mathbf{X}\in\mathbb{R}^{n^d\times n^d}$ there are $\mathcal{O}\left(n^{3d}\right)$ algebraic operations necessary. This shows the high computational advantage of this special case.

D. Method: Recursive Conversion of Measurements in TT-Format

This method yields a solution for problem 3.3. In general one can use the reliable TT-SVD algorithm [29]. This algorithm requires for each TN core to compute a SVD of the vector $\mathbf{y}_k$ reshaped to a $r_{i-1}n_i\times\frac{\mathrm{numel}(\mathbf{y}_k)}{r_{i-1}n_i}$ matrix. For exponentially large systems with increasing TT-ranks this can become limiting.

A novel method is developed. The idea is that the new measurement $\mathbf{y}_k$ is not completely independent of the last one $\mathbf{y}_{k-1}$ due to their connection with past states $\mathbf{x}_{k-1}$. The ALS optimization scheme can be used for the vector to TT conversion with the TT of the previous measurement as initialization. Take

$$\min_{\mathcal{X}}||\mathcal{R}\left(\mathbf{y}_k\right)-\mathcal{X}||_F^2 \tag{23}$$

with $\mathcal{X}_0=\mathcal{Y}_{k-1}$ and the reshape operator $\mathcal{R}(\cdot)$ to reshape the argument in the required tensor. The algorithm follows the general ALS scheme and is visualized in Fig. 4 for an example with $\mathbf{y}_k\in\mathbb{R}^{p_1p_2p_3}$ and $d=3$ TN cores in TT-format for the forward sweep. There are four main steps for each iteration (independent of forward of backward sweep):

1) Reshaping of the vector $\mathbf{y}_k$ to a tensor of order-d with the same mode sizes as the desired TT using the reshape operator $\mathcal{R}(\cdot)$. Visualized in Fig. 4 as the left tensor core.
2) Contraction with all TN cores except the one to be optimized. See that the resulting TN core after contraction has the same size as the one to be optimized. Visualized in Fig. 4 by the dotted lines for the connected indices, indicating contractions.
3) Replacing of the optimized TN core with the new updated one. Visualized in Fig. 4 by the TN core on the right which is of the same form as the dashed one. That is optimized and replaced.

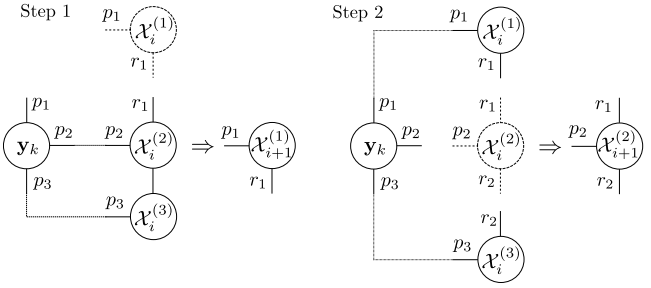


Fig. 4. ALS scheme for conversion of vector $\mathbf{y}_k$ to TT-format. Example with $d=3$ TN cores for the forward sweep. The core which is optimized at step i (dashed) is replaced in the next step with the optimized one (on the right after equal sign).

4) Orthogonalization of the new core. A QR-decomposition of the matricified core is performed and the non-orthogonal part is merged with the next core. Not visualized in Fig.4.

These steps are iteratively executed for all cores in the forward and backward sweep until convergence. A numeric comparison of the TT-SVD and TT-ALS conversion algorithm is attached in appendix A.

An analysis for the computational complexity of the complete algorithm with the presented methods compared to the conventional matrix Kalman filter is attached in appendix B. It can be summarized that the developed MIMO tensor Kalman filter works under the following two assumptions.

Assumption 3.1: The TT-rank of the measurement matrix in TT-format is or can reliably be approximated with $\mathrm{ttr}\left(\mathcal{C}\right)=1$.

Assumption 3.2: (1) The measurement noise of the system is sufficiently low such that its largest singular value of each matricified TN core is smaller than the one of the expression $\mathcal{C}_k\times_3^2\mathcal{P}_{k|k-1}\times_3^3\mathcal{C}_k$. (2) The TT-rank of the measurement noise covariance matrix in TT-format is $\mathrm{ttr}\left(\mathcal{R}\right)=1$, meaning uncorrelated noise.

IV. Introduction to Adaptive Optics

In order to improve the research impact e.g. in astronomy, scientific communities asks for higher image resolution. The limit for the angular resolution is given by the diffraction limit which decreases with increasing lens diameter. Therefore the size of telescopes nowadays increase. However, disturbances such as atmospheric turbulence give this trend a limit. As a countermeasure an AO system is employed.

Atmospheric turbulences distort the originally flat wavefront to a non-flat smooth wavefront. This occurs due to spatially and temporally varying refraction indices of the atmosphere which the light passes. To measure this distortion a Shack-Hartmann wavefront sensor (WFS) can be used [33]. The distortion can then be compensated with the use of deformable mirrors (DM), see Fig. 5.

Since there is a time delay between the wavefront measurement and the change of the DM a predictor is necessary. This can optimally be achieved by a Kalman filter. With the increasing size of telescopes also the AO systems grow and ask for large-scale real-time Kalman filtering.

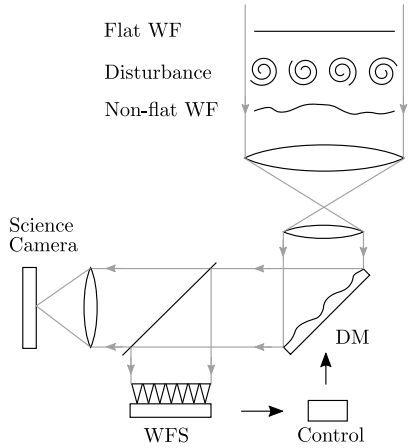


Fig. 5. Schematically drawing of an AO system including turbulence, WFS and DM

A. Turbulence Model

For the Kalman filter a linear model of the turbulence is necessary. A well established choice to describe the spatial correlation of the wavefront on a 2D grid is the Von Kármán model [11]. The correlation of two points can be derived depending on the distance between them and weather dependent parameters for the severity of the turbulence. With this an analytic equation for the covariance $\mathbf{C}_{\phi,k}$ of the vectorized wavefront ϕ_k at time step k is given.

Contrary to the spatial correlation, the temporal relation for the evolution of the wavefront in time is an open subject to research. A simple, linear model that is often chosen is the discrete-time AR-1 model, given as

$$\phi_{k+1} = \mathbf{A}_k \phi_k + \mathbf{v}_k \tag{24}$$

with $\mathbf{v}_k \sim (0, \mathbf{Q}_k)$ as Gaussian noise. This depicts the state process equation of a state-space model. The transition matrix $\mathbf{A}_k$ can be described by a simplified model using $\mathbf{A}_k = a_k \mathbf{I}$, where $|a_k| < 1$ in order to be stable. For a more general model various system identification methods can be used in a data-driven approach to model the temporal behaviour more precisely. The combined knowledge of spatial and temporal correlation can be used to derive an expression for the process noise covariance as

$$\mathbf{Q}_k = \mathbb{E}\left[\mathbf{v}_k \mathbf{v}_k^\top\right] = \mathbf{C}_{\phi,\mathbf{k}} - \mathbf{A}_k \mathbf{C}_{\phi,\mathbf{k}} \mathbf{A}_k^\top \tag{25}$$

B. Wavefront Sensing

For the output equation of the state-space system a model of the Shack-Hartmann WFS is derived. This sensor consists of a grid of small lenses (lenslets) which deflect the incoming light of a bright guide star to a photon sensor. The position of the star on the sensor depends on the local gradient of the wavefront. Using a Fried geometry means that the output of the WFS is the slope at the centre of a square of four sampled

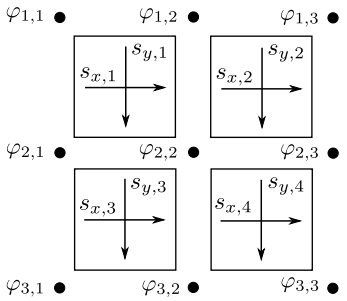


Fig. 6. Representation of Shack-Hartmann WFS with Fried geometry for $c = 3$ lenslets. The dots represent the centre of each lenslet for the wavefront samples. The slopes of the wavefront are denoted by s_x, s_y in the specific direction.

wavefront points, see Fig. 6. Define the sampled wavefront as regular 2D grid

$$\Phi = \begin{bmatrix} \varphi_{1,1} & \cdots & \varphi_{1,c} \\ \vdots & \ddots & \vdots \\ \varphi_{c,1} & \cdots & \varphi_{c,c} \end{bmatrix} \in \mathbb{R}^{c \times c} \tag{26}$$

with c as the number of lenslets in each row and column. Vectorizing this matrix by $\phi_k = \mathrm{vec}(\Phi_k^\top) \in \mathbb{R}^{c^2}$ yields the state vector of Eq. (24).

Example 4.1: For $c = 2$ the slopes in $x-$direction (s_x) and in $y-$ direction (s_y) are given with the notation as in Fig. 6 by

$$\begin{aligned} \begin{bmatrix} s_{x,1} \\ s_{y,1} \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} (\phi_{2,2} + \phi_{2,1}) - (\phi_{1,1} + \phi_{1,2}) \\ (\phi_{2,2} + \phi_{1,2}) - (\phi_{1,1} + \phi_{2,1}) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \phi \end{aligned} \tag{27}$$

For the general case the slopes s are stacked in the measurement matrix $\mathbf{G}_k \in \mathbb{R}^{2(c-1)^2 \times c^2}$ to obtain the measurement equation of the state-space model

$$\mathbf{y}_k = \mathbf{G} \phi_k + \mathbf{w}_k \tag{28}$$

with Gaussian measurement noise $\mathbf{w}_k \sim (0, \mathbf{R}_k)$ which can considered to be uncorrelated for each lenslet, yielding $\mathbf{R} = \sigma_w^2 \mathbf{I}$. From this model one can infer that the number of states is $n = c^2$ and the number of outputs is $p = 2(c-1)^2$.

Note that the system has two unobservable modes. The piston mode for $\varphi_{i,i} = \varphi_{j,j}\ \forall i, j$ which is an offset and the waffle mode with $\varphi_{i,i} = -\varphi_{i+1,i} = -\varphi_{i,i+1} = \varphi_{i+1,i+1}\ \forall i$ that is a chessboard pattern.

C. Linear Model for Tensor Filter

The derived state-space model in Eq. (24) and (28) can not directly be used in the tensor Kalman filter. It is necessary to fulfil assumption 3.1 and assumption 3.2. The latter is justified in [24]. In fact, the AO system can be assumed to be time-invariant. Yet, updates of the system matrices can happen after some time when new data is available, making the system time-varying. In the following the time-invariant notation is used.

The process equation does not need adaptation for the tensor filter. The matrices $\mathbf{A}, \mathbf{Q}$ can a priori be transferred to TT-format with the TT-SVD algorithm [29]. An approximation with lower TT-ranks is possible.

For the measurement equation one can exploit the structure in $\mathbf{G}$ to fulfil assumption 3.1. First, the slopes in $x-$direction (s_x) and in $y-$direction (s_y) are separated by reshuffling. Then, one can derive one output equations for each direction independently. In the following it is shown for the $x-$direction; the $y-$direction follows the same approach. Write

$$\begin{bmatrix} s_{x,1} \\ \vdots \\ s_{x,c-1} \end{bmatrix} = \mathbf{C}_1 \phi \tag{29}$$

with

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{G}_1 & -\mathbf{G}_1 & & \\ & \mathbf{G}_1 & -\mathbf{G}_1 & \\ & & \ddots & \ddots \end{bmatrix}; \quad \mathbf{G}_1 = \frac{1}{2} \begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ & & \ddots & \ddots \end{bmatrix}$$

where $\mathbf{C}_1 \in \mathbb{R}^{(c-1)^2 \times c^2}$ and $\mathbf{G}_1 \in \mathbb{R}^{(c-1)\times c}$. This yields

$$\mathbf{C}_1 = \mathbf{E}_1 \otimes \mathbf{G}_1 \tag{30}$$

with

$$\mathbf{E}_1 = \begin{bmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & \ddots & \ddots \end{bmatrix} \in \mathbb{R}^{(c-1)\times c}$$

to capture the structure and signs of $\mathbf{G}_1$ in $\mathbf{C}_1$. Finally, two distinct output equations can be written with this formulation as

$$\mathbf{y}_{1,k} = \mathbf{C}_1 \phi_k + \mathbf{w}_{1,k} \tag{31}$$

$$\mathbf{y}_{2,k} = \mathbf{C}_2 \phi_k + \mathbf{w}_{2,k} \tag{32}$$

Due to the Kronecker product in Eq. (30) this can be written as a TT with TT-rank of one. Consequently, the tensor Kalman filter has a sequential update with two measurement updates for both output equations. Note that for the conventional Kalman filter there is no requirement on the structure. Hence, one can combine the output equation as

$$\begin{bmatrix} \mathbf{y}_{1,k} \\ \mathbf{y}_{2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \phi_k + \begin{bmatrix} \mathbf{w}_{1,k} \\ \mathbf{w}_{2,k} \end{bmatrix} \tag{33}$$

V. Results and Performance Evaluation

A. Evaluation method

This section presents the performance of the developed tensor Kalman filter for the AO application. The filtered WF $\hat{\phi}_{k|k}$ is compared to the real WF ϕ_k. The evaluation concentrates on two main points. First, the computation times of different state-of-the-art algorithm are compared. Second, as a measure of accuracy the relative 2-Norm squared is considered

$$\frac{\left|\left|\mathrm{vec}(\phi) - \mathrm{vec}(\hat{\phi})\right|\right|_2^2}{||\mathrm{vec}(\phi)||_2^2} \tag{34}$$

Where ϕ is a collection of ϕ_k for all times and $\hat{\phi}$ is a collection of $\hat{\phi}_{k|k}$ for all times. This measure indicates how many digits are correctly estimated.

Since updates of the system matrices are possible, the general, time-varying case is considered. No steady-state assumptions hold and the Kalman gain has to be recomputed in every time step.

In the following a comparisons is drawn based on a basic AO simulator. This is used to compare the presented tensor Kalman filter algorithm with three state-of-the-art algorithms. First, the conventional Kalman filter [20], second, a Kalman filter based on first-order Tayler approximation of the Riccati equation [24] and third a transformation ensemble Kalman filter [14]. All filter are run over varying telescope sizes, i.e. varying state and output vector size. The computational advantage of the novel method and the relative accuracy are demonstrated.

B. Basic AO Simulator

For this simulator a WFS using a Shack-Hartmann sensor with Fried geometry as derived in section IV-B is used. The system size is varied for telescope sizes of $D = 2$ [m] to $D = 35$ [m] with a phase sampling distance of 0.5 [m]. Meaning for a telescope of diameter D there will be a grid of $2D \times 2D$ lenslets in the WFS. This corresponds to $p = 2(2D-1)^2$ outputs and $n = 4D^2$ states. The measurement noise is set to $\mathbf{R} = \sigma_v^2 \mathbf{I}_p$ with $\sigma_v = 3.33 \cdot 10^{-3}$ [rad].

For the turbulence model the Von Kármán model is used with a Fried parameter of $r_0 = 0.5$ [m] and an outer scale of $L_0 = 25$ [m]. A single layer of turbulence with wind speed of $v = 1$ [lenslet/time step] is chosen. This yields that the turbulence conditions are equal for all sizes of telescopes. For the temporal correlation the derived AR-1 model of Eq. (24) is used. The comparison approach [24] is restricted to $\mathbf{A} = a\mathbf{I}$, which is therefore used for all approaches. The process covariance matrix is identified using Eq. (25).

Due to uninformative $\mathbf{A}-$matrix, the two unobservable modes of the WFS (waffle, piston) will remain unobservable. A nonsingular state transformation matrix $\mathbf{T}$ is applied such that

$$\begin{bmatrix} \widetilde{\phi}_k \\ \phi_{unobs.} \end{bmatrix} = \mathbf{T}^\top \phi \tag{35}$$

where $\mathbf{T}^\top$ is given by an SVD of the measurement matrix by $\mathbf{USV}^\top = \mathbf{C}$ with the first $n-2$ columns of $\mathbf{V}$ as transformation matrix $\mathbf{T}$. Only the observable part of the wavefront $\widetilde{\phi}_k$ is compared.

A total simulation time of $k_{max} = 150$ [steps] is simulated in a Monte Carlo approach of $N = 8$ runs. The initial conditions are set to $\phi_0 \sim (0, \mathbf{I})$. An implementation of the ensemble transformation Kalman filter without localization methods is chosen. The number of ensemble members is chosen in line with results in [14] as $m = p/2$. Choosing more ensemble realizations will yield a more accurate result at the expense of higher computational times. For the tensor Kalman filter the TT-rounding tolerance is set to $\epsilon = 10^{-4}$. The process noise covariance is approximated with $\mathrm{ttr}\,(\mathcal{Q}) = 10$. This

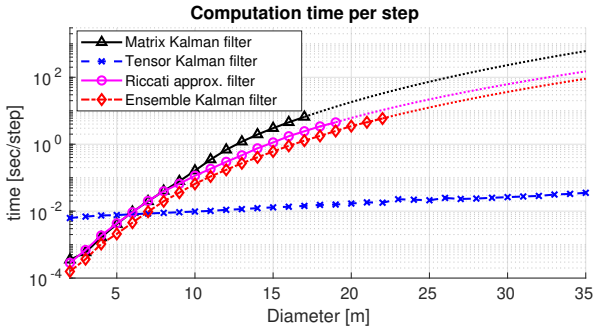


Fig. 7. Computational time for each time step of different Kalman filter methods over varying telescope sizes. Mean value over $N = 8$ Monte Carlo simulations. The dotted lines show extrapolations with power function approximation $(ax^b - c)$.

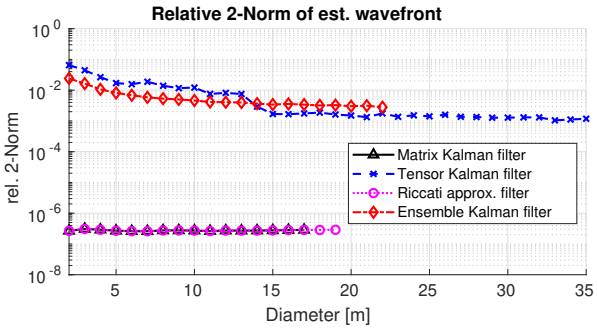


Fig. 8. Relative 2-Norm of the estimated wavefront $\hat{\phi}_{k|k}$ showing the accuracy of the tensor Kalman filter and the comparison methods.

approximation yields for a telescope with $D = 35$ [m] a relative accuracy of $2.82 \cdot 10^{-3}$ and is considered as sufficient. This is seen as worst-case scenario since it is the largest simulated system and therefore approximated with the relative least amount of singular values. The $\mathbf{A}-$matrix in TT-format will yield $\mathrm{ttr}\,(\mathcal{A}) = 1$ due to its diagonal structure.

The simulations are run on a Matlab implementation with version 9.3.0.713579 (R2017b) installed on Linux Ubuntu 16.04 LTS. The hardware consists of an Intel core i5-7200 quad-core CPU running at 2.5 GHz with 7.7 GB RAM.

The computational time for each time step of the four Kalman filter algorithms is depicted in Fig. 7 as mean value over the Monte Carlo runs. The variance is for all points at least two magnitudes lower than the mean value and not visualized. The difference of the Riccati approximation method with the plots given in [24] lies in the fact that here time-varying systems are assumed. Hence, the Kalman gain computation is run in every time step and no precomputations are possible.

The computational complexity of the matrix filter and the Riccati approximation filter are in the order of $\mathcal{O}\left(p^3\right)$ and for the Ensemble filter of $\mathcal{O}\left(1/4p^3\right)$. Since $p > n$ for the AO system, this results for all comparison methods in about $\mathcal{O}\left(D^6\right)$. For the tensor Kalman filter a complexity of $\mathcal{O}\left(D^3\right)$ is obtained because the largest mode size is $2D - 1$ in the output. The figure confirms this statement and shows empirically the high computational advantage of the tensor Kalman filter over the exponential computational complexity of the conventional filter, the Riccati approximation approach and the ensemble filter. All three comparison methods are extrapolated with power function of the form $y = ax^b + c$. The exponents are $b_{mat} = 6.3$ for the matrix filter, $b_{ric} = 5.7$ for the Riccati approximation filter and $b_{ens} = 5.9$ for the ensemble transformation filter. The tensor Kalman filter can reliably be approximated with a linear regression, were the slope is $8.2 \cdot 10^{-4}$. Approximating it with a power function yields an exponent of $b_{ten} = 1.7$. This underlines quantitatively that the tensor Kalman filter is of about four orders faster than the matrix Kalman filter for this specific system.

As a measure of accuracy the relative 2-Norm squared of the estimated, filtered wavefront $\hat{\phi}_{k|k}$ and the real one ϕ_k as given in Eq. (34) is considered. Results are given in Fig. 8 as mean value over the Monte Carlo iterations. Again, the variance of the points is at least two magnitudes lower and not depicted. The matrix filter and Riccati approximation filter reach about the same accuracy which is due to the extremely low measurement noise assumption. For the ensemble and tensor Kalman filter a loss of accuracy of up to four digits is obtained. For the ensemble Kalman filter this is due to the number of ensembles as trade-off between accuracy and computational time. For the tensor Kalman filter this comparably lower accuracy is mainly due to the approximation in the algorithm - namely the covariance truncation. Without these approximations the same accuracy as the matrix Kalman filter can be reached for the cost of about the same computational complexity. Therefore, a new filter design paradigm between an arguable loss of accuracy and a tremendous increase of computational speed is generated by the use of the tensor Kalman filter.

VI. Conclusion

This paper generalizes and extends the theory of tensor Kalman filter for large-scale time-varying MIMO system in state-space form. When applying the TT decomposition and multilinear algebra to the Kalman filter, the curse of dimensionality for a specific class of exponentially large systems is lifted. Methods are developed to tackle the main problems of the tensor Kalman filter for MIMO systems: (1) a specific class of subsystems, namely with uncorrelated, low measurement noise and with a TT-rank of one in the measurement matrix is defined for which the inversion in the Riccati equation can be computed extremely fast in TT-format. (2) a conversion method from vector to TT-format based on the ALS algorithm is presented that outperforms the state-of-the-art TT-SVD algorithm.

The algorithm is verified for AO telescope systems of the next generation. It is presented that the computational time for the tensor Kalman filter rises approximately linear with increasing system sizes while obtaining reasonably accurate results. This outperforms the conventional Kalman filter by several orders. Hence, in future AO systems the computationally driving factor is not the wavefront estimation and propagation but rather factors like the sensor readout and the transient times of the deformable mirrors will play a more crucial role.

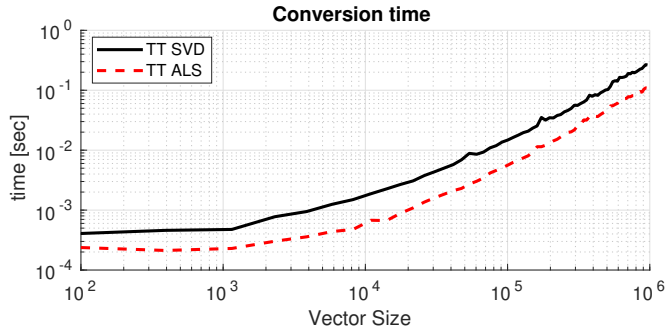


Fig. 9. Computational times for conversion of a vector to a TT with TT-SVD and novel TT-ALS algorithm. Mean time for 30 Monte Carlo simulations.

Points, which are not considered in this work are e.g. a batch-wise measurement update and the identification of the underlying system dynamics directly in TT-format. An improvement in terms of numeric accuracy of the developed filter can be obtained by a square root tensor Kalman filter implementation. This is of particular interest for future practical applications. Furthermore, the class of applicable systems for the fast MIMO filter can be extended if a more general fast inversion method in TT-format is developed.

APPENDIX A
EVALUATION OF TT-ALS FOR VECTOR CONVERSION

A comparison of the conversion of vectors to TTs with the TT-SVD and TT-ALS algorithm is presented. Take two randomly generated vectors with $\mathbf{y}_k \sim (0,1)$ for $k = 0, 1$, where $\mathbf{y}_0$ is in TT-format. Note that the vectors are not correlated by an underlying state-space system. Even for these independent vectors a single forward sweep is sufficient to achieve a conversion with machine precision.

The conversion time of the TT-SVD algorithm compared to the TT-ALS with one forward sweep are depicted in Fig. 9. In order to keep the connection to the AO application, the converted vector has size $\mathbf{y} \in \mathbb{R}^{2p^2}$ yielding three TN cores with mode sizes $(p, p, 2)$. The comparison shows the advantage of the TT-ALS method for the vector conversion.

APPENDIX B
COMPUTATIONAL COMPLEXITY OF THE ALGORITHM

An analysis for the computational complexity of the conventional Kalman filter and the presented tensor Kalman filter under the special condition such that all developed methods apply is presented in Table I. The state vector is taken to be $\mathbf{x} \in \mathbb{R}^{n^d}$ and the output vector as $\mathbf{y} \in \mathbb{R}^{p^d}$ with all modes sizes equal. Note that here the TT-ranks are $r_P = r_C = r_S = r_{S^{-1}} = 1$ is applied. Note also that the TT conversion of the measurement vector is still exponential in d since each optimization step requires a contraction with the complete measurement vector.

TABLE I
COMPUTATIONAL COMPLEXITY OF MIMO TN KALMAN FILTER.

Step	Kalman filter	Tensor Kalman filter
$\mathcal{Y}_k$	$\mathcal{O}(0)$	$\mathcal{O}\left((d-1)p^d r_Y^2\right)$
$\mathcal{S}_k$	$\mathcal{O}\left(n^{2d}p^d + n^d p^{2d}\right)$	$\mathcal{O}\left(d(n^2 p + np^2)\right)$
$\mathcal{S}_k^{-1}$	$\mathcal{O}\left(p^{3d}\right)$	$\mathcal{O}\left(dp^3\right)$
$\mathcal{K}_k$	$\mathcal{O}\left(n^{2d}p^d + n^d p^{2d}\right)$	$\mathcal{O}\left(d(n^2 p + np^2)\right)$
$\hat{\mathcal{X}}_{k\|k}$	$\mathcal{O}\left(2n^d p^d\right)$	$\mathcal{O}\left(dnpr_K^2 r_X^2\right)$
$\mathcal{P}_{k\|k}$	$\mathcal{O}\left(n^{2d}p^d + n^{3d}\right)$	$\mathcal{O}\left(dn^2 r_K^2 (n+p)\right)$
$\hat{\mathcal{X}}_{k+1\|k}$	$\mathcal{O}\left(n^{2d}\right)$	$\mathcal{O}\left(dn^2 r_A^2 r_X^2\right)$
$\mathcal{P}_{k+1\|k}$	$\mathcal{O}\left(2n^{3d}\right)$	$\mathcal{O}\left(dn^3 r_A^4\right)$

ACKNOWLEDGEMENT

The authors would like to thank Paolo Massioni for providing his code of [24] for the numeric comparison of the AO application. Special thanks also goes to Carlas Smith for many insightful discussions and remarks on the work.

REFERENCES

[1] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial applications of the kalman filter: A review," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5458–5471, Dec 2013.

[2] S. H. Babbs and K. B. Nowman, "Kalman filtering of generalized vasicek term structure models," *Journal of Financial and Quantitative Analysis*, vol. 34, no. 1, pp. 115–130, 1999.

[3] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1091–1107, Jul 2002.

[4] K. Batselier, Z. Chen, and N. Wong, "A Tensor Network Kalman filter with an application in recursive MIMO Volterra system identification," *Automatica*, vol. 84, 10 2016.

[5] K. Batselier and N. Wong, "Matrix output extension of the Tensor Network Kalman filter with an application in MIMO Volterra system identification," *Automatica*, vol. 95, pp. 413–418, 2018.

[6] K. Batselier, W. Yu, L. Daniel, and N. Wong, "Computing low-rank approximations of large-scale matrices with the tensor network randomized svd," *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 3, pp. 1221–1244, 2018.

[7] P. Benner, J.-R. Li, and T. Penzl, "Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems," *Numerical Linear Algebra with Applications*, vol. 15, no. 9, pp. 755–777, 2008.

[8] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*. New York, NY: Wiley, 1997, vol. 3.

[9] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions," *Foundations and Trends in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[10] R. Conan and C. Correia, "Object-oriented matlab adaptive optics toolbox," 2014.

[11] R. Conan, "Mean-square residual error of a wavefront after propagation through atmospheric turbulence and after correction with zernike polynomials," *J. Opt. Soc. Am. A*, vol. 25, no. 2, pp. 526–536, Feb. 2008.

[12] W. Ferng, W.-W. Lin, and C.-S. Wang, "The shift-inverted J-Lanczos algorithm for the numerical solutions of large sparse algebraic Riccati equations," *Computers & Mathematics with Applications*, vol. 33, no. 10, pp. 23 – 40, 1997.

[13] D. Gedon, P. Piscaer, K. Batselier, C. Smith, and M. Verhaegen, "Tensor Network Kalmen Filter for LTI Systems," in *2019 27th European Signal Processing Conference (EUSIPCO) (EUSIPCO 2019)*, A Corua, Spain, Sep. 2019.

[14] M. Gray, C. Petit, S. Rodionov, M. Bocquet, L. Bertino, M. Ferrari, and T. Fusco, "Local ensemble transform Kalman filter, a fast non-stationary control law for adaptive optics on ELTs: theoretical aspects and first simulation results," *Opt. Express*, vol. 22, no. 17, pp. 20 894–20 913, Aug 2014.

[15] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, Jun. 2010.

[16] N. Halko, P. Martinsson, and J. Tropp, "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[17] S. Holtz, T. Rohwedder, and R. Schneider, "The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. 683–713, Mar. 2012.
[18] Y.-F. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State Estimation in Electric Power Grids: Meeting New Challenges Presented by the Requirements of the Future Grid," *Signal Processing Magazine, IEEE*, vol. 29, pp. 33–43, Sep 2012.
[19] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall, 2000.
[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
[21] T. Kolda and B. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
[22] Q. Kong, Q. Zhao, C. Wei, and Y. Liu, "Efficient Traffic State Estimation for Large-Scale Urban Road Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 398–407, Mar 2013.
[23] N. Lee and A. Cichocki, "Regularized Computation of Approximate Pseudoinverse of Large Matrices Using Low-Rank Tensor Train Decompositions," *arXiv e-prints*, p. arXiv:1506.01959, Jun. 2015.
[24] P. Massioni, H. Raynaud, C. Kulcsr, and J. Conan, "An Approximation of the Riccati Equation in Large-Scale Systems With Application to Adaptive Optics," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 479–487, Mar. 2015.
[25] P. Massioni and M. Verhaegen, "Distributed Control for Identical Dynamically Coupled Systems: A Decomposition Approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 124–135, Jan 2009.
[26] J. Mohammadpour and K. M. Grigoriadis, *Efficient modeling and control of large-scale systems*. Springer Science & Business Media, 2010.
[27] R. Orus, "A practical introduction to Tensor Networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117 – 158, 2014.
[28] I. Oseledets, "DMRG approach to fast linear algebra in the TT–format," *Comput. Meth. Appl. Math.*, vol. 11, no. 3, pp. 382–393, 2011.
[29] ——, "Tensor-Train Decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
[30] I. Oseledets and S. Dolgov, "Solution of linear systems and matrix inversion in the tt-format," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. A2718–A2739, 2012.
[31] I. Oseledets and E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra Appl.*, vol. 432, no. 1, pp. 70–88, 2010.
[32] I. Oseledets, E. Tyrtyshnikov, and N. Zamarashkin, "Tensor-Train ranks for matrices and their inverses," *Computational Methods in Applied Mathematics*, vol. 11, 01 2011.
[33] B. C. Platt and R. B. Shack, "History and principles of Shack-Hartmann wavefront sensing." *Journal of refractive surgery*, vol. 17, no. 5, pp. S573–S577, Sep./Oct. 2001.
[34] F. Roddier, *Adaptive optics in astronomy*. Cambridge university press, 1999.
[35] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The Ensemble Kalman filter: a signal processing perspective," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 56, Aug 2017.
[36] B. Sinquin, "Structured matrices for predictive control of large and multi-dimensional systems," PhD dissertation, TU Delft, May 2019.
[37] C. F. Van Loan, "The ubiquitous Kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 85 – 100, 2000, numerical Analysis 2000. Vol. III: Linear Algebra.
[38] S. R. White, "Density-matrix algorithms for quantum renormalization groups," *Physical review. B, Condensed matter*, vol. 48, pp. 10 345–10 356, Nov 1993.

Chapter 6

Conclusion

6-1 Summary of work

The study of tensors and the TT-decomposition for the application in a large-scale Kalman filter is motivated by two main factors: The compression rate for systems with inherently low TT-rank like Kronecker models and more importantly, its capability to lift the curse of dimensionality. This mathematical framework can decrease the computational complexity of exponentially large systems to enable optimal real-time estimation to a huge variety of engineering problems.

This thesis provides an extensive overview and rigorous introduction to tensor calculus as necessary tool for the understanding and development of a tensor Kalman filter. The existing theory of the tensor Kalman filter for the system identification of MIMO Volterra systems in [3, 5] is first rewritten for general time-varying state space systems. The scheme is analysed in detail to identify limitations and computational bottlenecks. It has been shown that for system dynamics with TT-ranks greater than one in the representative TT-format of the $\mathbf{A}-$ and $\mathbf{C}$-matrix, the tensor Kalman filter is significantly slower than the conventional matrix Kalman filter. This decrease is theoretically due to the polynomially dependence of the TT-ranks on the computational complexity. The effect of this issue is solved by truncating the TT-ranks of the a priori and a posteriori covariance tensor to low TT-ranks, specifically to TT-rank of one. Hence, the effect of large TT-ranks in the system dynamics on the computational complexity could be decreased significantly. An initial exponential trend of the computation time for larger systems is decreased to an almost linear trend for SISO systems.

As a second bottleneck the computational complexity for MIMO systems is identified. The tensor Kalman filter equations from [5] are rewritten to distribute possible exponentially large output vectors over all tensor cores and therefore lift the curse of dimensionality for MIMO systems. Following, the main problem for a MIMO tensor Kalman filter lies in the inversion for computing the Kalman gain. This inversion has to be computed in TT-format. In case the TT-ranks are unequal to unity, the inversion is in TT-format computationally not effective solved. A special class of systems is defined for which a fast MIMO tensor Kalman filter

can be generated: The TT-ranks of the measurement matrix and the measurement noise are equal to one and the measurement noise is sufficiently low.

An application which falls in this class of systems is found to be AO systems. The system dynamics for the AO system are rewritten to fit in this framework. A comparison with other state-of-the-art methods is performed. These methods are (1) the conventional matrix Kalman filter, (2) an ensemble Kalman filter and (3) a Riccati approximation methods. It is demonstrated that the MIMO tensor Kalman filter outperforms all methods in terms of computational complexity by about four orders for this specific systems. This comes at a cost of an approximation. Hence, a design paradigm as a trade-off between computational complexity and accuracy is generated with this new type of Kalman filter.

The work of this thesis is presented and resulted in two papers. The first one is accepted for the EUSIPCO conference in A Coruña, Spain. The second one is a first draft, which is intended to be submitted in for the IEEE transactions on control systems technology journal. However, limitations of the developed approach have been identified as addressed in the following, which need to be solved for a second publication.

6-2 Limitation of Developed Method

The application of the designed filter algorithm for the defined class of system to AO has demonstrated the following main drawbacks:

- The truncation of the TT-rank of the covariance tensor in the measurement and time update has been shown in the AO example to be a rough approximation. A loss of relative accuracy from about 10^{-7} in the conventional Kalman filter to about 10^{-3} for the tensor Kalman filter is obtained in this specific example. A reason for this loss may be the small number of used tensor cores. Truncating the covariance at TT-rank of one preserves only the most dominant singular value of each matricified tensor core. Using more tensor cores provides theoretically more information about the system. However, the used AO example only uses two cores due to the specific modelling.

- The assumption of low measurement noise for the MIMO tensor Kalman filter corresponds in the application of AO to an accuracy in the optical measurement components of about $\lambda/200$. This is an unreachable accuracy in optical components nowadays. Hence, the theory is in its current status not applicable to real physical AO systems.

- The results for the accuracy of the AO MIMO tensor Kalman filter indicate a change in the Kalman filter between systems sizes of $D = 13$ [m] and $D = 14$ [m], see Fig. 8 of the IEEE paper in chapter 5. Analysis showed that from this system size and larger, the prior in the time update is barely considered. Hence, a breakdown of the Kalman filter equations happens. A more rigorous analysis and explanation is given in Appendix A.

Especially the last listed limitation is a mayor constraint of the developed algorithm. Several ideas to solve this issues are presented in Appendix A.

6-3 Future Work

All presented limitations of the MIMO tensor Kalman filter are the result of the absence of a fast inversion method for the TT-format. Therefore, the goal to solve the MIMO tensor Kalman filter in a more general manner is to develop a fast TT-inversion method for the application in Kalman filters. This inversion method can open the door for MIMO tensor Kalman filter theory of a more general class of systems than the one defined within this thesis. A promising approach is to use a TT-randomized SVD as presented in this thesis [6]. However, the approach needs a generalization with a full TT-QR decomposition meaning that the resulting $\mathcal{Q}$ and $\mathcal{R}$ are both TTs. This is currently not yet mathematically solved.

For practical implementations of the tensor Kalman filter it is of great interest to develop a square root tensor Kalman filter. This ensures numeric stability for the symmetry and positive definiteness of the covariance tensor. Since the square root Kalman filter method is based on a factorization of the covariance matrix, e.g. a full TT-QR decomposition is necessary.

Finally, for the development of new algorithms there should always a specific application been kept in mind. During the work of this thesis the focus often lay on the development of fundamental algorithms for any kind of system. However, this is often an ambitious challenge. More suitable and tailored algorithms can be designed for the class of systems of a specific application. E.g. in the case of this thesis the AO application can be written with TT-ranks of one in the system matrices of the output equation. This property can be exploited to design fast algorithms.

Appendix A

Analysis and Ideas for MIMO Tensor Kalman Filter Limitations

A-1 Analysis of Limitations

As highlighted in the conclusions, chapter 6, one of the main limitations of the MIMO tensor Kalman filter is the change in accuracy for the specific AO example between system size with $D = 13$ [m] and $D = 14$ [m] as shown in chapter 5, Fig. 8 of the IEEE paper draft. The reasons for this change are analysed in the following.

The AO system is modelled in the state process equation with $\mathbf{A} = a\mathbf{I}$. The temporal dynamics which are not covered by this simple model have to be modelled in the process noise covariance by $\mathbf{Q} = \mathbf{C}_\phi - \mathbf{A}\mathbf{C}_\phi\mathbf{A}^\top$, see section 3-2-2. With larger system sizes, the modelling error of $\mathbf{A}$ increases and the norm of $\mathbf{Q}$ increases since the process noise covariance has to capture this modelling error. For simplicity the time-varying index k is omitted here.

In the tensor Kalman filter the a priori and a posteriori covariance tensor are truncated with TT-rank of one. This means that only the most dominant singular value of each matricified tensor core is considered. At some point in the covariance time-update $\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^\top + \mathbf{Q}$, the largest singular value of the process noise covariance $\mathbf{Q}$ becomes larger than the one of the product $\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^\top$. This has two main reasons:

1. As a consequence of the modelling error of the $\mathbf{A}$-matrix the norm of the noise covariance $\mathbf{Q}$ rises. Therefore, the most dominant singular value of $\mathbf{Q}$ outnumbers the one of $\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^\top$.
2. In case the filter does not diverge, the confidence in the state estimations rises with longer estimation times. The values of the state error covariance matrix $\mathbf{P}_{k|k}$ decrease and therefore the most dominant singular value of $\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^\top$ decreases, as well.

As a consequence of these influences the prior $\mathbf{P}_{k|k}$ is barely considered in the time update due to the truncation to TT-rank one. This means that the update of the filter is only based

on noise covariances. Hence, from this point on, the designed filter is not a recursive Kalman filter. This statement is supported by the plots in Fig. A-1. The plot on the left shows a working tensor Kalman filter. The covariance tensor $\mathcal{P}_{k+1|k}$ underapproximates but converges to the covariance of the conventional Kalman filter independent of the noise covariance. In the right plot the prior $\mathcal{P}_{k|k}$ is barely considered and the resulting norm of the covariance tensor $\mathcal{P}_{k+1|k}$ is almost equal to the norm of $\mathbf{Q}$ confirming the hypothesis above.

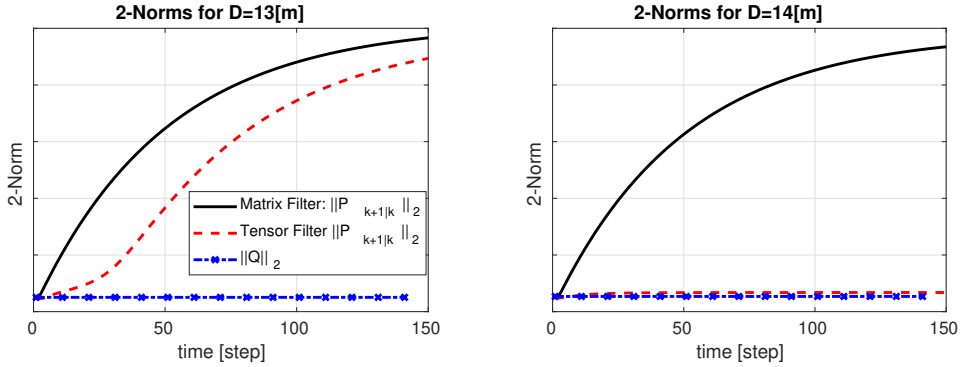


Figure A-1: 2-Norm of Matrix filter covariance $\mathbf{P}_{k+1|k}$, tensor filter covariance $\mathcal{P}_{k+1|k}$ and of noise covariance $\mathbf{Q}$ for the specific AO example with sizes $D = 13$ [m] (left figure) and $D = 14$ [m] (right figure).

A-2 Solution Proposals

The analysis above demonstrates that the truncation of the covariance tensor in the time update causes the limitations of the MIMO tensor Kalman filter algorithm. However, this truncation is necessary for the inversion in the computation of the Kalman gain. For TT-ranks higher than one, the inversion slows the algorithms significantly down. Several ideas have been tried to solve this issue, which are presented in the following.

Parallel Computing of Contractions An independent solution to improve the computational speed is to use parallel computing for TT contractions. Each core is contracted independently from all other cores. This independence enables parallel computations. In Matlab the parallel computing toolbox with the parallel for-loop function `parfor` can be used. Simulations of contractions of two randomly generated TTs with the parallel computing contraction and the conventional sequential contraction are performed. For different number of tensor cores with all modes of size $n = 10$ the results for the timing as a ratio between the parallel computing function and the conventional function are demonstrated in Fig. A-2. The mean value of $N = 250$ Monte Carlo simulations is visualized.

The results show as expected a decreasing tend for larger number of tensor cores. More tensor cores emphasize the advantages of parallel computing over sequential computing. However, the figure clearly indicates that the ratio is always larger than one. This result means that the overhead of the Matlab parallel computing function to copy data to and from the single workers is too large to make this function useful. A possible explanation can be that on the used hardware, only two parallel worker could be used in parallel. More workers might be

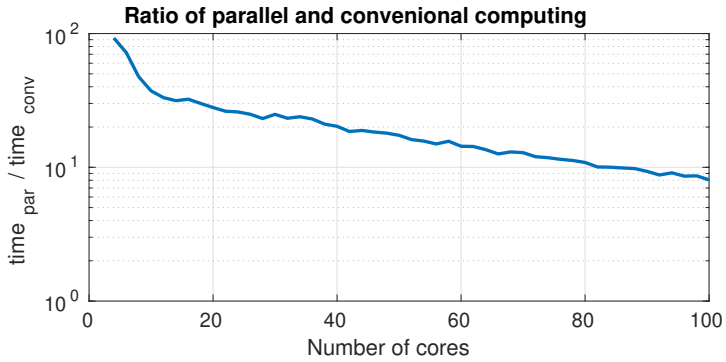


Figure A-2: Ratio of contraction time of d tensor cores between parallel computing and sequential computing over increasing number of tensor cores with mode size $n = 10$ and for $N = 250$ Monte Carlo simulations.

advantageous. Using a mode size less than $n = 10$ or more results in a similar curve as in Fig. A-2. Note that the results are obtained when the Matlab parallel computing pool is pre-activated.

Imposing Structure on the Kalman Gain Using covariance truncation to TT-rank of one $\text{ttr}(\mathcal{P}_{(\cdot|\cdot)}) = 1$ and $\text{ttr}(\mathcal{C}) = 1$ yields a fast inversion of the expression $\mathbf{S} = \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^\top + \mathbf{R}$ due to $\text{ttr}(\mathcal{S}) = 1$. In this case the inverse of $\mathcal{S}$ also has TT-rank of one. Hence, computing the Kalman gain $\mathbf{K} = \mathbf{P}_{k|k-1}\mathbf{C}^\top\mathbf{S}^{-1}$ in TT-format will yield a TT-rank of one $\text{ttr}(\mathcal{K}) = 1$.

Since the filter with covariance TT-rank truncation is stable, one could impose structure on the Kalman gain itself. Covariance TT-rank truncation is not necessary. However, simulations with Kalman gain TT-rank truncation only result in an unstable filter. The reason for this divergence is that the covariance tensors in the measurement and in the time update do not remain symmetric, positive-definite with TT-rank truncation of $\text{ttr}(\mathcal{K}) = 1$. The theoretical background for this phenomenon is unclear and needs further research.

Approximating the Covariance Tensor with Higher TT-Ranks The limitation of the designed tensor Kalman filter originates in the covariance TT-rank truncation. Higher TT-ranks for the covariance truncation can counteract this limitation. However, $\text{ttr}(\mathcal{P}_{(\cdot|\cdot)}) = 1$ is necessary for the inversion of $\mathcal{S}$. Two possibilities are explored to combine both necessities:

1. Increase the truncation TT-rank and only truncate $\mathcal{P}_{k+1|k}$ for the use in $\mathcal{S}$. In the remaining measurement and time update $\text{ttr}(\mathcal{P}_{(\cdot|\cdot)}) > 1$ can be used.
2. Increase $\text{ttr}(\mathcal{P}_{k|k})$ in the measurement update and only use $\text{ttr}(\mathcal{P}_{k+1|k}) = 1$ in the time update. The latter is necessary for the inversion with TT-rank of one. Since two batch measurement updates are performed (for $y-$ and $x-$direction of the measurement slopes), then only for the second measurement update, the covariance TT-rank can be increased.

Both approaches have been tested in simulation. Approach (1) yields similar results as presented in the original tensor Kalman filter, see chapter 5. A stable Kalman filter is obtained.

However, also the same problem arises. Namely, that at a specific point, the prior is not considered in the time update and the update is only based on the noise covariance. Approach (2) directly yields unstable filter results, blowing up the state estimates.

Batch-Wise Measurement Update In the current tensor filter design there are two measurement update batches computed in sequence for the $x-$direction slopes and the $y-$direction slopes. Increasing the number of batches decreases the individual batch output size p_i. The idea is to have a total number of p measurement updates with $p_i = 1$. Hence, the inverse for the Kalman gain becomes a scalar problem. Each batch measurement update is therefore a SISO update with an exponentially large state-vector and one output. This approach does not necessarily require $\mathrm{ttr}(\mathcal{P}_{(\cdot|\cdot)}) = 1$. Increasing the truncation TT-rank of the covariances, then the limitation in the time update that the prior $\mathcal{P}_{k|k}$ is not considered should not occur.

Simulations with this idea are performed. Each row of the measurement matrix from the Shack-Hartmann sensor is considered for one individual batch measurement update. Two tensor cores are chosen in order to be compliant with the time-update. The simulation is performed with the same simulation constraints and system settings as in chapter 5. The resulting time and relative accuracy of the filter are presented for a Monte Carlo iteration of $N = 4$ in Fig. A-3.

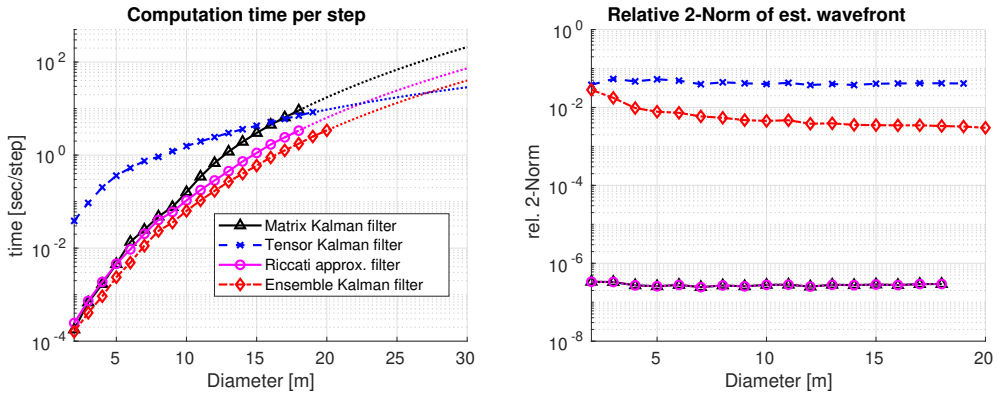


Figure A-3: Results of different Kalman filter algorithms in simulation for varying AO system sizes with SISO batch measurement update. Left: computational times, right: relative accuracy.

Regarding the computational complexity two statements can be made. First, the initial computational demand increases for these SISO batch measurement updates significantly. Second, again a power function of the form $y = ax^b + c$ for the approximation of the computational times yields the factors as given in Tab. A-1. This highlights quantitatively that the tensor Kalman filter has a computational advantage in the exponent of about three orders for this specific system.

For the relative accuracy of the filter also two statements are made. First, it is visible that the relative accuracy decreases to a value of about 0.04, meaning that roughly two digits are estimated correctly. Second, an analysis showed that the the prior is always considered in the time update. Hence, the problem as identified in the previous section for the filter of chapter 5 does not exist. Therefore, with this method a valid Kalman filter is designed which outperforms computationally the state-of-the-art methods for super large systems.

Table A-1: Factor values for approximation of computational times with power function of the form $y = ax^b + c$.

Method	a	b	c
Tensor filter	$2.391 \cdot 10^{-3}$	2.761	$1.330 \cdot 10^{-1}$
Matrix filter	$1.824 \cdot 10^{-7}$	6.135	$-2.804 \cdot 10^{-2}$
Riccati approx. filter	$9.047 \cdot 10^{-8}$	6.032	$2.893 \cdot 10^{-3}$
Ensemble filter	$3.986 \cdot 10^{-8}$	6.095	$8.077 \cdot 10^{-3}$

It is important to note that the results are obtained with $\mathrm{ttr}(\mathcal{P}_{(\cdot|\cdot)}) = 1$. Increasing this truncation TT-rank should give a more accurate estimation with a constant increase of the relative accuracy in Fig. A-3 for the tensor Kalman filter. However, first simulations with higher truncation TT-ranks resulted in unstable estimations. Investigations in this direction are necessary.

A second possibility which is not yet explored is to increase the number of tensor cores. In the simulation two tensor cores are used, equally to the simulations in chapter 5. This number of cores is necessary to obtain $\mathrm{ttr}\,(\mathcal{C}) = 1$. However, this condition is not necessary when using SISO batch measurement updates. Prime factorizations of the modes for a larger number of tensor cores are possible. The increase in tensor cores can possibly decrease the computational time and increase the accuracy since with TT-rank truncation more information is preserved. Research is this direction is necessary.

Bibliography

[1] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial applications of the kalman filter: A review," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5458–5471, Dec 2013.

[2] S. H. Babbs and K. B. Nowman, "Kalman filtering of generalized vasicek term structure models," *Journal of Financial and Quantitative Analysis*, vol. 34, no. 1, pp. 115–130, 1999.

[3] K. Batselier, Z. Chen, and N. Wong, "A Tensor Network Kalman filter with an application in recursive MIMO Volterra system identification," *Automatica*, vol. 84, Oct 2016.

[4] ——, "Tensor Network alternating linear scheme for MIMO Volterra system identification," *Automatica*, vol. 84, pp. 26 – 35, 2017.

[5] K. Batselier and N. Wong, "Matrix output extension of the Tensor Network Kalman filter with an application in MIMO Volterra system identification," *Automatica*, vol. 95, pp. 413–418, 2018.

[6] K. Batselier, W. Yu, L. Daniel, and N. Wong, "Computing low-rank approximations of large-scale matrices with the tensor network randomized SVD," *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 3, pp. 1221–1244, 2018.

[7] M. J. Booth, "Adaptive optics in microscopy," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1861, pp. 2829–2843, 2007.

[8] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering.* New York, NY: Wiley, 1997, vol. 3.

[9] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions," *Foundations and TrendsÂő in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[10] R. Conan, "Mean-square residual error of a wavefront after propagation through atmospheric turbulence and after correction with zernike polynomials," *J. Opt. Soc. Am. A*, vol. 25, no. 2, pp. 526–536, Feb 2008.

[11] D. L. Fried, "Statistics of a geometric representation of wavefront distortion," *J. Opt. Soc. Am.*, vol. 55, no. 11, pp. 1427–1435, Nov 1965.

[12] R. Gilmozzi and J. Spyromilio, "The European Extremely Large Telescope (E-ELT)," *The Messenger*, vol. 127, Mar 2007.

[13] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, Jun 2010.

[14] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[15] K. J. G. Hinnen, "Data-driven optimal control for adaptive optics," PhD dissertation, TU Delft, 2007.

[16] S. Holtz, T. Rohwedder, and R. Schneider, "The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. 683–713, Mar 2012.

[17] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall, 2000.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[19] T. Kolda and B. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[20] A. Kolmogorov, "The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers," *Akademiia Nauk SSSR Doklady*, vol. 30, pp. 301–305, 1941.

[21] A. N. Kolmogorov, "Dissipation of Energy in Locally Isotropic Turbulence," *Akademiia Nauk SSSR Doklady*, vol. 32, p. 16, Apr. 1941.

[22] D. Lee, "Numerically efficient methods for solving least squares problems," 2012.

[23] N. Lee and A. Cichocki, "Regularized Computation of Approximate Pseudoinverse of Large Matrices Using Low-Rank Tensor Train Decompositions," *arXiv e-prints*, p. arXiv:1506.01959, Jun. 2015.

[24] P. Massioni, H. Raynaud, C. KulcsÃąr, and J. Conan, "An Approximation of the Riccati Equation in Large-Scale Systems With Application to Adaptive Optics," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 479–487, Mar 2015.

[25] R. Orus, "A practical introduction to Tensor Networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117 – 158, 2014.

[26] I. Oseledets, "Tensor-Train Decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[27] I. Oseledets and S. Dolgov, "Solution of linear systems and matrix inversion in the tt-format," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, pp. A2718–A2739, 2012.

[28] I. Oseledets and E. Tyrtyshnikov, "Breaking the curse of dimensionality, or how to use SVD in many dimensions," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3744–3759, 2009.

[29] I. V. Oseledets, "DMRG approach to fast linear algebra in the TT–format," *Comput. Meth. Appl. Math.*, vol. 11, no. 3, pp. 382–393, 2011.

[30] I. V. Oseledets, S. V. Dolgov, V. A. Kazeev, O. Savostyanov, D. Lebedeva, P. Zhlobich, T. Mach, and L. Song, "TT-toolbox," *Available online*, 2012.

[31] I. V. Oseledets and E. E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra Appl.*, vol. 432, no. 1, pp. 70–88, 2010.

[32] I. V. Oseledets, "Approximation of 2dx2d matrices using tensor decomposition," *SIAM J. Matrix Analysis Applications*, vol. 31, pp. 2130–2145, 2010.

[33] P. Piscaer, "Sparse VARX model identification for large-scale adaptive optics," July 2016, master Thesis, TU Delft.

[34] B. C. Platt and R. B. Shack, "History and principles of Shack-Hartmann wavefront sensing." *Journal of refractive surgery*, vol. 17, no. 5, pp. S573–S577, Sep./Oct. 2001.

[35] F. Roddier, *Adaptive optics in astronomy*. Cambridge university press, 1999.

[36] G. Rousset, F. Lacombe, P. Puget, N. N. Hubin, E. Gendron, J.-M. Conan, P. Y. Kern, P.-Y. Madec, D. Rabaud, D. Mouillet, A.-M. Lagrange, and F. J. Rigaut, "Design of the Nasmyth adaptive optics system (NAOS) of the VLT," 1998.

[37] B. Sinquin, "Structured matrices for predictive control of large and multi-dimensional systems," PhD dissertation, TU Delft, May 2019.

[38] C. F. Van Loan, "The ubiquitous Kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 85 – 100, 2000, numerical Analysis 2000. Vol. III: Linear Algebra.

[39] M. Verhaegen, P. Pozzi, O. Soloviev, G. Vdovin, and D. Wilding, "Lecture notes on control for high resolution imaging," Delft Center for Systems and Control, Delft University of Technology, Lecture Notes, Apr 2017.

[40] S. R. White, "Density-matrix algorithms for quantum renormalization groups," *Physical review. B, Condensed matter*, vol. 48, pp. 10 345–10 356, Nov 1993.

[41] R. J. Wojnarowski, H. S. Cole, and J. L. Henkes, "Optical light pipe and microwave waveguide interconnects in multichip modules formed using adaptive lithography," Apr. 7 1998, US Patent 5,737,458.