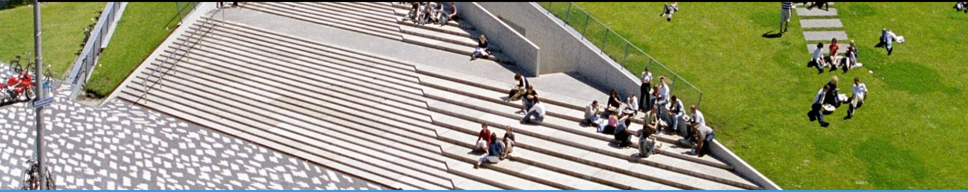
Tensor Network Kalman Filter for LTI Systems

Delft University of Technology

Daniel Gedon

September 4, 2019

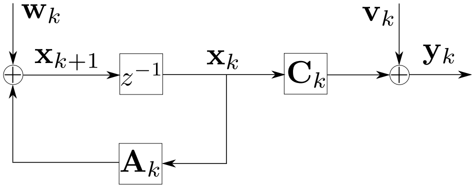


Outline

1. Introduction / Problem Description
2. Tensors
3. Tensor Kalman Filter for LTI Systems
4. Conclusion

Introduction / Problem Description

Filtering of an LTI system in state-space description



Problem of Kalman filter for large-scale systems

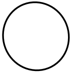
States n and outputs p large

- Computational complexity of order $\mathcal{O}\left(n^3\right)$
  for covariance update $\mathbf{P}_{k|k}$ and $\mathbf{P}_{k+1|k}$, Kalman gain $\mathbf{K}_k$
- Storage of system matrices with $\mathbf{A} \in \mathbb{R}^{n \times n},\ \mathbf{C} \in \mathbb{R}^{p \times n}$

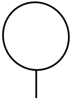
$\Rightarrow$ *Design a Kalman filter for large-scale real-time problems*

Introduction to Tensors

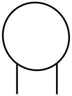
- Tensor as multidimensional array $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$, $I_k = 1, \ldots, n_k$
- Each entry $a_{i_1 i_2 \ldots i_d}$ determined by d indices $\rightarrow$ Order-d tensor
- Visualization in Tensor Network (TN) Diagrams:



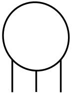
Scalar
$a \in \mathbb{R}$



Vector
$\mathbf{a} \in \mathbb{R}^{I_1}$



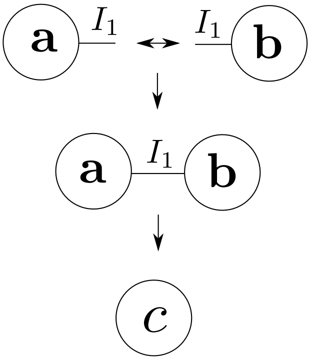
Matrix
$\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$



Tensor Order-3
$\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$

Basic Operations (1)

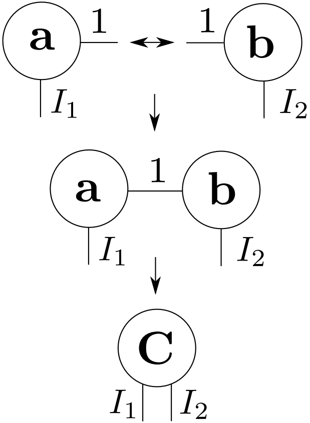
- Define $\mathbf{a} \in \mathbb{R}^{I_1}$, $\mathbf{b} \in \mathbb{R}^{I_1}$
- Elementwise notation:
  $$c = \sum_{i_1=1}^{n_1} a_{i_1} b_{i_1}$$
  $$= \mathbf{a} \cdot \mathbf{b}$$
- Tensor notation:
  $$c = \mathbf{a} \times_1^1 \mathbf{b}$$
  $\Rightarrow$ Inner Product



Basic Operations (2)

- Define $\mathbf{a} \in \mathbb{R}^{I_1}$, $\mathbf{b} \in \mathbb{R}^{I_2}$
  $\rightarrow \mathbf{A} \in \mathbb{R}^{I_1 \times 1}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times 1}$
- Elementwise notation:
  $c_{i_1 i_2} = \sum_{i=1}^{1} a_{i_1} b_{i_2}$
- Vector notation:
  $\mathbf{C} = \mathbf{a}\mathbf{b}^{\mathrm{T}} = \mathbf{a} \circ \mathbf{b}$
- Tensor notation:
  $\mathbf{C} = \mathbf{a} \times_2^2 \mathbf{b}$

$\Rightarrow$ Outer Product

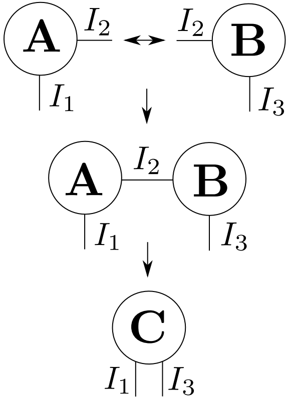


Basic Operations (3)

- Define $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times I_3}$
- Elementwise Notation:
  $c_{i_1 i_3} = \sum_{i_2=1}^{n_2} a_{i_1 i_2} b_{i_2 i_3}$
- Matrix Notation:
  $\mathbf{C} = \mathbf{AB} = \mathbf{A} \circ \mathbf{B}^{\mathrm{T}}$
- Tensor Notation:
  $\mathbf{C} = \mathbf{A} \times_2^1 \mathbf{B}$

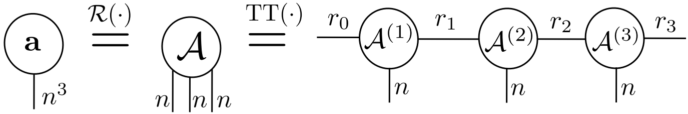
  $\Rightarrow$ Matrix Product

- Complexity: $\mathcal{O}(n_1 n_2 n_3)$ Flops

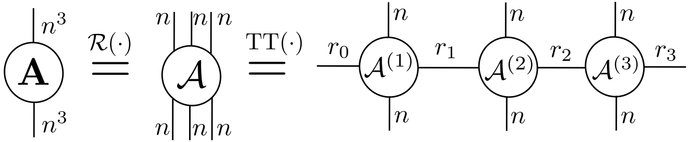


Tensor Train (TT) Decomposition

- TT: One order-d tensor $\Rightarrow$ d order-3 TN cores.
- Vector, example:



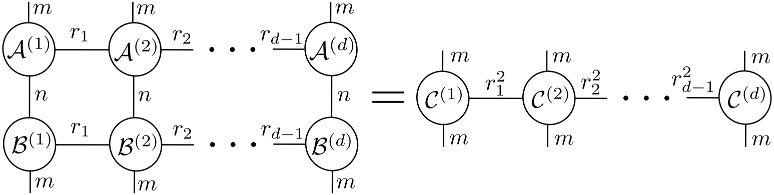
- TT-matrix: One order-$2d$ tensor $\Rightarrow$ d oder-4 TN cores
- Matrix, example:



- TT-rank boundary conditions: $r_0 = r_d = 1$

Tensor Train (TT) Operations

- Matrix-Format:
  - Define $\mathbf{A} \in \mathbb{R}^{m^d \times n^d}$, $\mathbf{B} \in \mathbb{R}^{n^d \times m^d}$
  - $\mathbf{C} = \mathbf{AB}$
  - Complexity: $\mathcal{O}\left(n^d m^{2d}\right)$
- TT-Format:
  - Define $\mathcal{A}^{(i)} \in \mathbb{R}^{r_{i-1} \times m \times n \times r_i}$, $\mathcal{B}^{(i)} \in \mathbb{R}^{r_{i-1} \times n \times m \times r_i}$
  - $\mathcal{C}^{(i)} = \mathcal{A}^{(i)} \times_3^2 \mathcal{B}^{(i)}$



  - Complexity: $\mathcal{O}\left(dr^4 nm^2\right)$

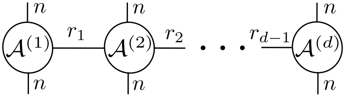
Relation of TT and Kronecker Model

- Kronecker-Model

$$\mathbf{A} = \mathbf{A}^{(d)} \otimes \cdots \otimes \mathbf{A}^{(1)}$$

- TT-model

$$\mathcal{A} = \mathcal{A}^{(1)} \times_4^1 \cdots \times_4^1 \mathcal{A}^{(d)}$$

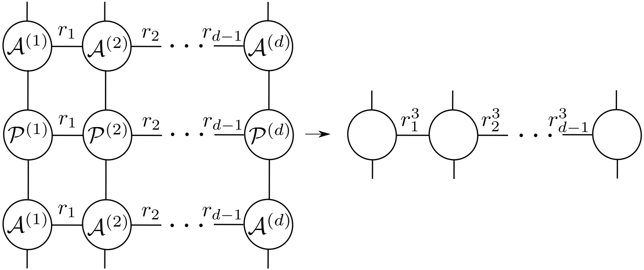


- Both models are equal if: $\mathrm{ttr}(\mathcal{A}) = 1$
- TN core size: $\mathcal{A}^{(i)} \in \mathbb{R}^{1\times n\times n\times 1}$
  - $\mathbf{A}^{(i)} = \mathcal{R}(\mathcal{A}^{(i)})$
  - Tensor contraction $\mathcal{A} \times_4^1 \mathcal{B} \Rightarrow$ outer product $\mathbf{A} \circ \mathbf{B}$
- Outer product/Kronecker product:
  related by reshuffling and permutation

Tensor Kalman Filter - for LTI Systems

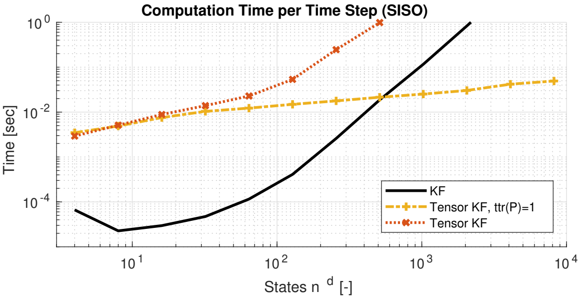
1. Transform system matrices **A**, **C**, (**Q**, **R**) in TT-format.
   $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$, $\mathbf{C} \in \mathbb{R}^{p \times n^d}$
2. Rewrite all variables in the KF in TT-format
   Covariance, Kalman gain, ...
3. Rewrite KF equations with multilinear operations

Example: Time update of covariance (Matrix form: $\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}$)



Tensor Kalman Filter - Improvement: TT-ranks

- Problem:
  - TT-ranks dominate the algorithmic complexity
  - No influence on TT-ranks $\mathrm{ttr}(\mathcal{A})$ and $\mathrm{ttr}(\mathcal{C})$
- Analysis SISO:
  - Effective only for systems with $\mathrm{ttr}(\mathcal{A}) = \mathrm{ttr}(\mathcal{C}) = 1$
  - Solution: Truncation of $\mathrm{ttr}(\mathcal{P}) \rightarrow$ low TT-rank approximation



Conclusion

Summary of work:

- Extends and generalizes existing Tensor Network Kalman filter
- Proposed solutions for bottleneck:
  Effect of large TT-rank $\Rightarrow$ covariance TT-rank truncation
- Opened large-scale Kalman filtering to more general systems

Follow-up work since paper submission:

- Extension to MIMO systems
- Application: adaptive optics for wavefront estimation
- Challenges with accuracy:
  Paradigm: accuracy $\Longleftrightarrow$ computational gain

Future Work:

- Solve challenges with MIMO tensor filtering
- Development of a square root Tensor Network Kalman filter

Thank you for your attention

Questions?

Appendix - SISO Filter