Tensor Kalman Filter

For Large-Scale Systems

Delft University of Technology

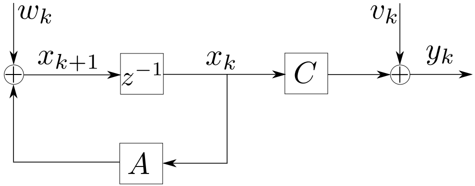
Daniel Gedon

July 5, 2019

Outline

Introduction / Problem Description

Filtering of a system in state-space description



Problem of Kalman filter for large-scale systems

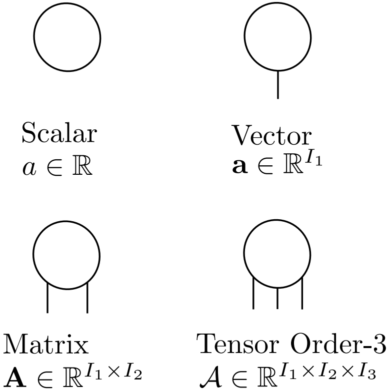
States n and outputs p large

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

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

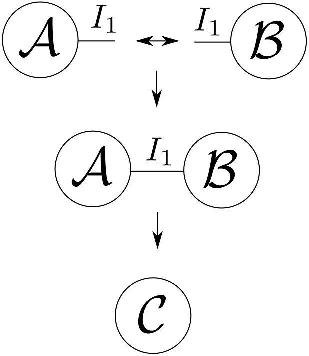
Introduction to Tensors

- 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:



Basic Operations (1)

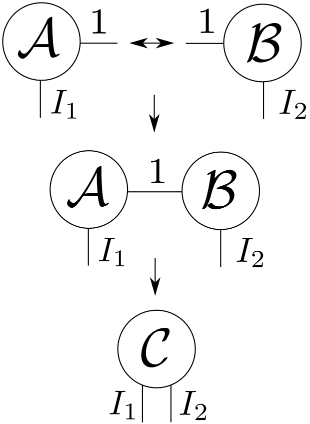
- Define $\mathcal{A} \in \mathbb{R}^{I_1}$, $\mathcal{B} \in \mathbb{R}^{I_1}$
- Elementwise notation:
$$c = \sum_{i_1=1}^{n_1} a_{i_1} b_{i_1} = \mathbf{a} \cdot \mathbf{b}$$
- Tensor notation:
$$\mathcal{C} = \mathcal{B} \times_1^1 \mathcal{A}$$
$\Rightarrow$ Inner Product



Basic Operations (2)

- Define $\mathcal{A} \in \mathbb{R}^{I_1}$, $\mathcal{B} \in \mathbb{R}^{I_2}$
  $\rightarrow \mathcal{A} \in \mathbb{R}^{I_1 \times 1}$, $\mathcal{B} \in \mathbb{R}^{I_2 \times 1}$
- 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:
  $\mathcal{C} = \mathcal{B} \times_2^2 \mathcal{A}$

$\Rightarrow$ Outer Product

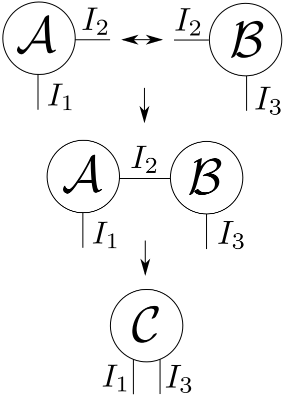


Basic Operations (3)

- Define $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2}$, $\mathcal{B} \in \mathbb{R}^{I_2 \times I_3}$
- 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:
  $\mathcal{C} = \mathcal{B} \times_1^2 \mathcal{A}$

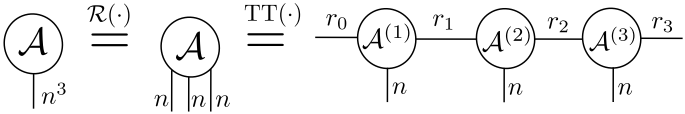
  $\Rightarrow$ Matrix Product

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

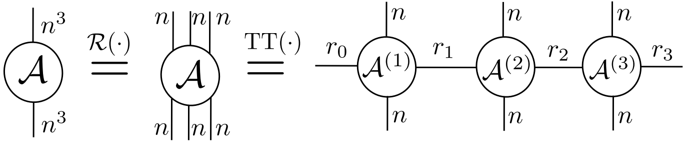


Tensor Train (TT) Decomposition

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



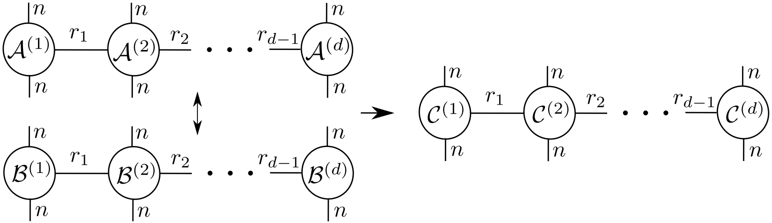
- Matrix, example:



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

Tensor Train (TT) Operations

- Matrix-Format:
  - Define $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$, $\mathbf{B} \in \mathbb{R}^{n^d \times n^d}$
  - $\mathbf{C} = \mathbf{AB}$
  - Complexity: $\mathcal{O}\left(n^{3d}\right)$
- TT-Format:
  - Define $\mathcal{A}^{(i)}, \mathcal{B}^{(i)} \in \mathbb{R}^{r_{i-1} \times n \times n \times r_i}$
  - $\mathcal{C} = \mathcal{B} \times_2^3 \mathcal{A}$

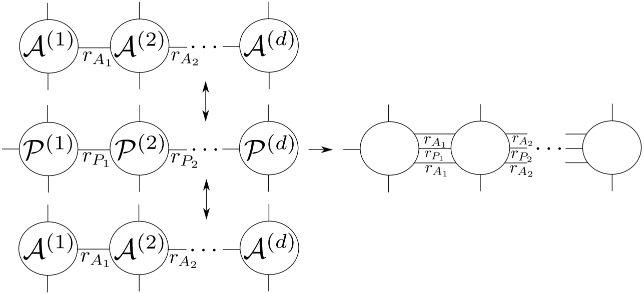


  - Complexity: $\mathcal{O}\left(dn^3 r^2\right)$

Tensor Kalman Filter

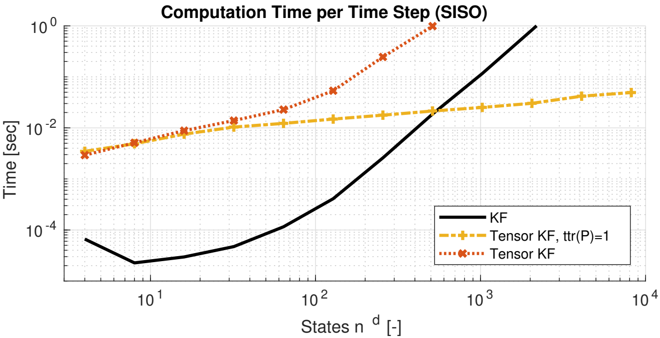
1. Transform system matrices **A**, **C**, (**Q**, **R**) in TT-format.
   $\mathbf{A} \in \mathbb{R}^{n^d \times n^d}$, $\mathbf{C} \in \mathbb{R}^{p \times n^d}$
2. Rewrite all variables in the KF in TT-format
   Covariance, Kalman gain, ...
3. Rewrite KF equations with multilinear operations

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 $\mathrm{ttr}(\mathcal{P}) \rightarrow$ Approximation

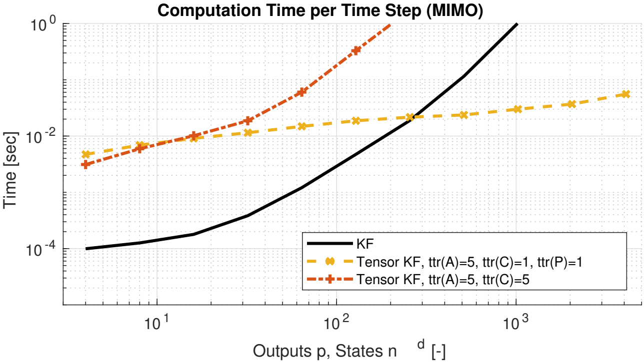


Tensor Kalman Filter - Improvement: MIMO

- Analysis MIMO ($p = n^d$) including TT-rank solution:
  - Inversion $\mathbf{S} = \left(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^{\mathrm{T}} + \mathbf{R}\right)$ with $\mathcal{O}\left(p^3\right)$ same as matrix filter
  - Special case for $\mathrm{ttr}(\mathcal{S}) = 1$

    $\rightarrow$ all cores will be matrices $(1 \times n \times n \times 1)$

    $\rightarrow$ obtained by $\mathrm{ttr}(\mathcal{C}) = 1$ and low measurement noise



Application - General

In General: Systems with low TT-ranks in $\mathcal{A}, \mathcal{C}$

- Recall: TT is outer product

$$\mathcal{A} = \mathcal{A}^{(d)} \times_1^3 \cdots \times_1^3 \mathcal{A}^{(1)}$$

- If all TT-ranks are equal to one, then

$$\mathcal{A} = \mathcal{R}\left(\mathbf{A}^{(1)} \otimes \cdots \otimes \mathbf{A}^{(d)}\right)$$

- Reason: Outer product is reshaped and permuted version of Kronecker product

  $\Rightarrow$ every Kronecker Model has TT-rank of one in TT-format

TU Delft

Application - Specific: MIMO

For MIMO special case system:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{w}_k$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{C}_1 & 0 & \dots & 0 \\ 0 & \mathbf{C}_1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & \mathbf{C}_1 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k$$

- Interconnected systems with distributed sensing
- Applications:
  - (Power) Networks
  - Partial Differential Equations
  - ...

Conclusion / Future Work

Conclusion:

- Solves the problem of exponentially large system matrices **A**, **C**
- Fast SISO filter
- Fast MIMO filter in special case

Limitations:

- General: Necessary small TT-ranks of system matrices
- MIMO filter: $\mathrm{ttr}(\mathcal{C}) = 1$ and low measurement noise
- No square root implementation available

Future Work:

1. Improve filter algorithm
2. Find more applications

Thank you for your attention

Questions?

Appendix - Applications

- Recall: TT is outer product

$$\mathcal{A} = \mathcal{A}^{(d)} \times_1^3 \cdots \times_1^3 \mathcal{A}^{(1)}$$

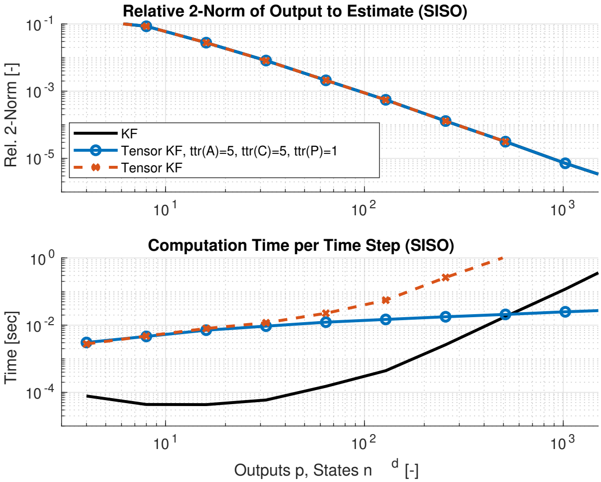
- If all TT-ranks are equal to r, then

$$\mathcal{A} = \mathcal{R}\left(\sum_{k=1}^{r} \mathbf{A}_k^{(1)} \otimes \cdots \otimes \mathbf{A}_k^{(d)}\right)$$

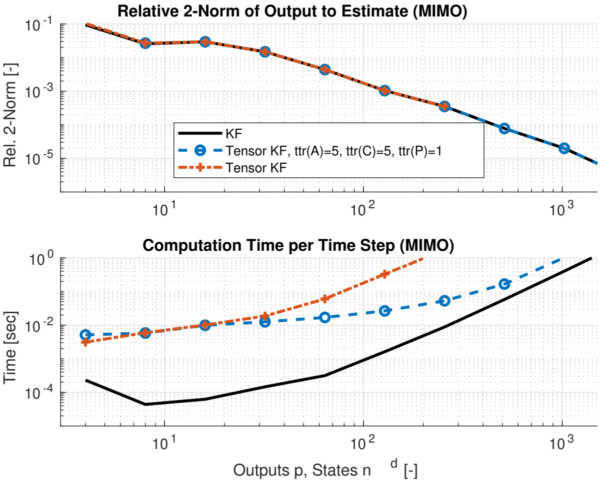
- Reason: Outer product is reshaped and permuted version of Kronecker product

  $\Rightarrow$ every Kronecker Model with low r has low TT-ranks in TT-format

Appendix - SISO Filter



Appendix - MIMO Filter



TU Delft