# TENSOR-BASED FRAMEWORK WITH MODEL ORDER SELECTION, DENOISING AND HIGH ACCURACY MATRIX DECOMPOSITION FOR THIRD GENERATION GLOBAL POSITIONING SYSTEM TIME-DELAY ESTIMATION IN DYNAMIC MULTIPATH SCENARIOS

**MATEUS DA ROSA ZANATTA**

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA**
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA**

**UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA**
**FACULDADE DE TECNOLOGIA**
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

# TENSOR-BASED FRAMEWORK WITH MODEL ORDER SELECTION, DENOISING AND HIGH ACCURACY MATRIX DECOMPOSITION FOR THIRD GENERATION GLOBAL POSITIONING SYSTEM TIME-DELAY ESTIMATION IN DYNAMIC MULTIPATH SCENARIOS

**MATEUS DA ROSA ZANATTA**

**Orientador: Prof. Dr.-Ing. João Paulo Javidi da Costa, ENE/UnB**
**Coorientador: Prof. Dr. Rafael Timóteo de Sousa Júnior, ENE/UnB**

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO PPGENE.TD - 185/22**
**BRASÍLIA-DF, 24 DE FEVEREIRO DE 2022.**

**UNIVERSIDADE DE BRASÍLIA**
**FACULDADE DE TECNOLOGIA**
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

# TENSOR-BASED FRAMEWORK WITH MODEL ORDER SELECTION, DENOISING AND HIGH ACCURACY MATRIX DECOMPOSITION FOR THIRD GENERATION GLOBAL POSITIONING SYSTEM TIME-DELAY ESTIMATION IN DYNAMIC MULTIPATH SCENARIOS

**MATEUS DA ROSA ZANATTA**

TESE DE DOUTORADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

APROVADA POR:

Prof. Dr.-Ing. João Paulo Javidi da Costa, ENE/UnB
Orientador

Prof. Dr. Rafael Timóteo de Sousa Júnior, ENE/UnB
Coorientador

Prof. Dr. Sébastien Roland Marie Joseph Rondineau FGA/UNB
Examinador interno

Prof. Dr.-Ing. Felix Antreich, IEET/ITA
Examinador externo

Prof. Dr. Carlos Alexandre Rolim, DETI/UFC
Examinador externo

**BRASÍLIA, 24 DE FEVEREIRO DE 2022.**

**FICHA CATALOGRÁFICA**
MATEUS DA ROSA ZANATTA
**Tensor-based Framework with Model Order Selection, Denoising and High Accuracy Matrix Decomposition for Third Generation Global Positioning System Time-Delay Estimation in Dynamic Multipath Scenarios**
**2022xv, 164p., 201x297 mm**
(ENE/FT/UnB, Doutor, Engenharia Elétrica, 2022)
Tese de Doutorado - Universidade de Brasília
Faculdade de Tecnologia - Departamento de Engenharia Elétrica

## REFERÊNCIA BIBLIOGRÁFICA

MATEUS DA ROSA ZANATTA (2022) Tensor-based Framework with Model Order Selection, Denoising and High Accuracy Matrix Decomposition for Third Generation Global Positioning System Time-Delay Estimation in Dynamic Multipath Scenarios. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.TD - 185/22, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 164p.

## CESSÃO DE DIREITOS

AUTOR: Mateus da Rosa Zanatta
TÍTULO: Tensor-based Framework with Model Order Selection, Denoising and High Accuracy Matrix Decomposition for Third Generation Global Positioning System Time-Delay Estimation in Dynamic Multipath Scenarios.
GRAU: Doutor ANO: 2022

---

Mateus da Rosa Zanatta
Rua Emília Pomar 1, Cacilhas, Setúbal - Portugal

# ACKNOWLEDGMENTS

I am grateful for the technical and computational support of the Decision-Making Technologies Laboratory - LATITUDE, of the University of Brasilia, which is supported by CNPq - Brazilian National Research Council (Grants 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN, and 465741/2014-2 INCT on Cybersecurity), by CAPES - Brazilian Higher Education Personnel Improvement Coordination (Grants PROAP PPGEE/UnB, 23038.007604/2014-69 FORTE, and 88887.144009/2017-00 PROBRAL), by FAP-DF - Brazilian Federal District Research Support Foundation (Grant 0193.001366/2016 UIoT, and Grant 0193.001365/2016 SSDDC), by the Brazilian Ministry of the Economy (Grant 005/2016 DIPLA, and Grant 083/2016 ENAP), by the Institutional Security Office of the Presidency of Brazil (Grant ABIN 002/2017), by the Administrative Council for Economic Defense (Grant CADE 08700.000047/2019-14), by the General Attorney of the Union (Grant AGU 697.935/2019), by the National Auditing Department of the Brazilian Health System SUS (Grant DENASUS 23106.118410/2020-85), by the General Attorney's Office for the National Treasure (Grant PGFN 23106.148934/2019-67), and by the Decanates of Research and Innovation and Postgraduate Studies of the University of Brasilia (Grant 23106.067186/2021-37).

# Abstract

Global Navigation Satellite Systems (GNSS), such as the American Global Positioning System (GPS), European Galileo, Russian GLONASS, and Chinese BeiDou, are crucial for applications that demand very accurate positioning. Time-delay estimation tensor-based methods, such as Canonical Polyadic Decomposition by Generalized Eigenvalue Decomposition (CPD-GEVD), Direction of Arrival Khatri-Rao Factorization (DoA/KRF), and Canonical Polyadic Decomposition via simultaneous matrix diagonalization (SECSI), can be combined with the third-generation GPS (GPS3) L1C Civil (L1C) signal in order to significantly mitigate the positioning degradation caused by multipath components. However, these schemes assume that the number of multipath components is constant and known, although, in practice, the amount of multipath components is time-varying, requiring the incorporation of a suitable model order selection (MOS) scheme. In this paper, we propose a complete framework including the following steps: estimation of the number of multipath components, separation of the source signals via a tensor-based approach, and estimation of the time-delay of the estimated line of sight (LOS) component. To estimate the model order, we consider the discriminant function based method (DFBM), which is suitable for colored noise scenarios. To estimate the model order in dynamic scenarios, we consider the Exponential Fitting Test (EFT). Moreover, we significantly mitigate the effect of multipath components by incorporating the pre-processing step called Tensor-based Multiple Denoising (MuDe) approach in our proposed framework. Finally, the proposed framework in dynamic scenarios groups the epochs with the same model orders forming sub-tensors with constant model order. Therefore, such grouping allows for increased accuracy.

# Resumo

Os Sistemas Globais de Navegação por Satélite (GNSS), como o Sistema de Posicionamento Global (GPS), o Galileo, o GLONASS e o BeiDou, são cruciais para aplicações que exigem um posicionamento muito preciso. Métodos baseados em tensores de estimativa de atraso de tempo, como *Canonical Polyadic Decomposition by Generalized Eigenvalue Decomposition* (CPD-GEVD), *Direction of Arrival Khatri-Rao Factorization* (DoA/KRF) e *Canonical Polyadic Decomposition via simultaneous matrix diagonalization* (SECSI), podem ser combinado com o sinal L1C Civil (L1C) do GPS de terceira geração (GPS3) para mitigar significativamente a degradação de posicionamento causada por componentes de multicaminho. No entanto, esses métodos assumem que o número de componentes de multicaminho é constante e conhecido, embora, na prática, a quantidade de componentes de multicaminho seja variável no tempo, exigindo a incorporação de um esquema de seleção de ordem de modelo (MOS) adequado. Neste artigo, propomos uma estrutura completa incluindo as seguintes etapas: estimativa do número de componentes de multicaminho, separação dos sinais de origem por meio de uma abordagem baseada em tensores e estimativa do atraso de tempo da linha de visão (LOS) componente. Para estimar a ordem do modelo, consideramos o método baseado em função discriminante (DFBM), que é adequado para cenários de ruído colorido. Para estimar a ordem do modelo em cenários dinâmicos, consideramos o T*Exponential Fitting Test* (EFT). Além disso, mitigamos significativamente o efeito de componentes de multicaminho incorporando a etapa de pré-processamento *Tensor-based Multiple Denoising* (MuDe) na estrutura proposta. Por fim, o framework proposto em cenários dinâmicos agrupa as épocas com as mesmas ordens de modelo formando subtensores com ordem de modelo estimadas. Portanto, tal agrupamento permite maior precisão.

# CONTENTS

# LIST OF FIGURES

# Chapter 1

# Introduction

As Global Navigation Satellite Systems (GNSS), such as GPS, Galileo, GLONASS, and Beidou, become more ubiquitous, this technology shows to be essential to applications such as civilian aviation, autonomous driving, defense, and timing and synchronization of critical networks. The GNSS receivers require the line of sight (LOS) signals from at least four satellites to estimate their position on the Earth's surface. Additionally to the LOS component, non-LOS (NLOS) mutipath components occur due to the reflections on trees, poles, lamp and buildings. The superposition of the LOS and NLOS multipath components degrades the time-delay estimation and, consequently, the positioning estimation.

Thus, state-of-the-art GNSS receivers equipped with a single antenna are extremely sensitive to the effect of multipath components [1, 2, 3]. Therefore, tensor-based multipath mitigation methods combined with antenna arrays have been proposed to significantly mitigate the effect of the multipath components in comparison with the single antenna matrix-based counterpart techniques. In [4], the authors propose the Tensor-based Eigenfilter using Higher Order Singular Vector Decomposition (HOSVD) combined with Forward-Backward Averaging (FBA) [5], Spatial Smoothing (SPS) [6, 7], and a compressed signal bank in order to mitigate multipath and to improve the time-delay estimation. In [8] a three step approach based on the direction of arrival (DoA) estimation, the Khatri-Rao factorization (KRF) and the selection with a compressed signal bank is proposed in order to mitigate multipath in the time-delay estimation (TDE). In [9], the Canonical Polyadic Decomposition by a Generalized Eigenvalue Decomposition (CPD-GEVD) is proposed due to its robustness against multipath components and array imperfections. In [10], the state-of-the-art tensor-based methods [4, 8, 9] are extended to third-generation GPS (GPS3), since GPS3 is robust against multipath components in comparison with the second generation GPS (GPS2) due to its Time Multiplexed Binary Offset Carrier (TMBOC) modulation [11, 12, 13]. Moreover, improved signals based on the Time Multiplexed Binary Offset Carrier (TMBOC) modulation are proposed in the new GPS L1 civilian signal (L1C) for the (GPS3) [11, 12, 13]. Furthermore, in [10] the state-of-the-art tensor-based methods [4, 8, 9] are extended to GPS3 in a straightforward manner, and tensor-based GPS3 methods outperform the tensor-based

GPS2 method. Still in [10], the L1C pilot signal is combined with the algorithms derived in [4], which include the Tensor-based Eigenfilter with Forward-Backward Averaging (FBA) [14, 5] and Expanded Spatial Smoothing (ESPS) [6, 7]. Furthermore, in [10] the L1C pilot signal is combined with the CPD-GEVD based time-delay estimation with the L1C pilot signal. Moreover, in [15], the authors indicate that the SECSI method can be applied to GPS2 and GPS3 signals in a straightforward manner, and antenna array based receivers using GPS3 and SECSI outperform antenna array receivers using GPS2 and SECSI. Finally, in [16] the authors propose a complete tensor-based framework to estimate the tensor model order by rearranging the main tensor into sub-tensors and use the estimated model order to computed the time-delay. In [17], the authors proposed the $(L_r, L_r, 1)$-GEVD approach to perform multi-linear rank-$(L_r, L_r, 1)$ decomposition by clustering NLOS components and obtaining the LOS component. Meanwhile, in [18, 19], the authors proposed a tensor-based subspace tracking framework to keep track and update the tensor signal subspace. However, similarly to previous tensor-based methods, [17] and [18, 19] assume that the model order is constant between data epochs. Additionally, [20] combines an Extended Kalman Filter (EFK) with an antenna array and tensor-based GNSS receivers to improve performance of vehicle localization estimation.

In contrast to the state-of-the-art solutions for antenna array based GNSS receivers that assume a known amount of multipath components, also known as model order, in this thesis, we propose a complete framework including the following steps: estimation of the number of multipath components, mitigation of the multipath effect, separation of the source signals via a tensor-based approach and estimation of the time delay of the estimated line of sight (LOS) component. To estimate the model order in static scenarios, we consider the discriminant function based method (DFBM) [21], which is suitable for colored noise scenarios. To estimate the model order in dynamic scenarios, we consider the Exponential Fitting Test (EFT) [22]. Additionally, we further mitigate the effect of multipath components by incorporating the pre-processing step called Tensor-based Multiple Denoising (MuDe) [23] approach in our proposed framework. Finally, the proposed framework groups the epochs with same model order by forming sub-tensors with constant model order. Therefore, such grouping allows for increased accuracy. Moreover, we explore other tensor modes by rotating the received signal tensor. Therefore, we propose to utilize the Mode 1 HOSVD SECSI with left-hand matrix (Mode 1 HOSVD SECSI) method [24, 25, 26] to perform factor matrix estimation since the Mode 1 HOSVD SECSI provided better factor matrices estimates in dynamic scenarios.

The matrix based Model Order Selection (MOS) literature is quite extensive including the following state-of-the-art approaches: 1-D Akaike's Information Criterion (1-D AIC) [27], 1-D Minimum Description Length (1-D MDL) [27], EFT [22], Modified EFT (M-EFT) [27], RADOI [28], and the subspaced-based ESTER [29]. In addition, tensor-based MOS are also present in the literature, such as R-D AIC [30], R-D MDL [30], and R-D EFT [27]. Therefore, we evaluate the most suitable MOS for time-delay estimation for antenna array based GNSS receivers. Finally, the proposed framework is suitable for both GPS2 and GPS3.

Moreover, we show that we can utilize the matrix-based MOS and the R-D EFT methods in a scenario with a varying model order. Firstly, we describe that matrix-based MOS methods can be used to estimate the model order of each collected epoch. Then, we can use the estimated model orders to group tensor slices and to form sub-tensors with the same model order. Afterward, we employ the tensor-based TDE methods to decompose the sub-tensors. However, in some cases we cannot group the tensor slices; therefore, we have to decompose a single matrix. Thus, we propose a Matrix-based Eigenfilter utilizing the Singular Value Decomposition (SVD) approach to compute the time-delay. Subsequently, we show that we can use the maximum and the minimum estimated model order of each epoch to perform the tensor decomposition. Consequently, by using the maximum and the minimum model order, we no longer have to split the original tensor into several sub-tensors. Hence, we improve the TDE by assuming the tensor model order is constant. Finally, when we have a varying model order, we show that the R-D AIC and R-D MDL methods cannot estimate the tensor model order while the R-D EFT successfully estimates it. Finally, to perform matrix decomposition in an dynamic environment we utilized two variants of the SECSI method. We utilize the SECSI variants that utilize the right-hand side and left-hand side matrix of the first mode, as described in [24, 25, 26].

This thesis is divided into five chapters, including this introduction. Chapter 2 introduces the data model for the antenna array-based GNSS assuming multipath components used throughout this thesis. Moreover, the author introduces the static scenario as an environment where the model order is constant across the collected epochs. Also, Chapter 2 presents the MOS and TDE methods combination and simulation results. In addition, Chapter 3 introduces a dynamic scenario as an environment where the model order may vary across the collected epochs. Also, the authors present the MOS methods along with the proposed TDE method used in dynamic scenarios and the simulation results. Furthermore, in Appendix A, the author introduces the Notation and the Matrix and Tensor calculus used in this thesis.

## 1.1 Author's Publications

M. R. Zanatta, J. P. C. L. Da Costa, F. Antreich, M. Haardt, G. Elger, F. L. Lopes De Mendonça, and R. T. De Sousa, "Tensor-based framework with model order selection and high accuracy factor decomposition for time-delay estimation in dynamic multipath scenarios," IEEE Access, vol. 8, pp. 174 931–174 942, 2020.

D. V. de Lima, M. R. Zanatta, J. P. C. L. da Costa, R. T. de Sousa Jr, and M. Haardt, "Robust tensor-based techniques for antenna array-based GNSS receivers in scenarios with highly correlated multipath components," Digital Signal Processing, p. 102715, Mar 2020.

G. A. Santos, J. P. C. L. da Costa, D. V. de Lima, M. R. Zanatta, B. J. G. Praciano, G. P. M. Pinheiro, F. L. L. de Mendonça, and R. T. de Sousa, "Improved localization framework for autonomous vehicles via tensor and antenna array based gnss receivers," in 2020 Workshop on Communication Networks and Power Systems (WCNPS), 2020, pp. 1–6.

M. R. Zanatta, F. L. L. de Mendonça, F. Antreich, D. V. de Lima, R. K. Miranda, G. D. Galdo, and J. P. C. L. da Costa, "Tensor-based timedelay estimation for second and third generation global positioning system," Digital Signal Processing, vol. 92, pp. 1 – 19, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200418307073.

M. R. Zanatta, D. V. de Lima, J. P. C. L. da Costa, R. K. Miranda, F. Antreich, and R. T. de Souza Jr., "Técnica Tensorial de Estimação de Atraso para GPS de Segunda e Terceira Geração," in XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, September 2018.

M. R. Zanatta, R. K. Miranda, J. P. C. L. da Costa, F. Antreich, and D. V. de Lima, "Antenna array based receivers for third generation global positioning system," in Workshop on Communication Networks and Power Systems, November 2017.

# Chapter 2

# Time-Delay Estimation for Static Scenarios

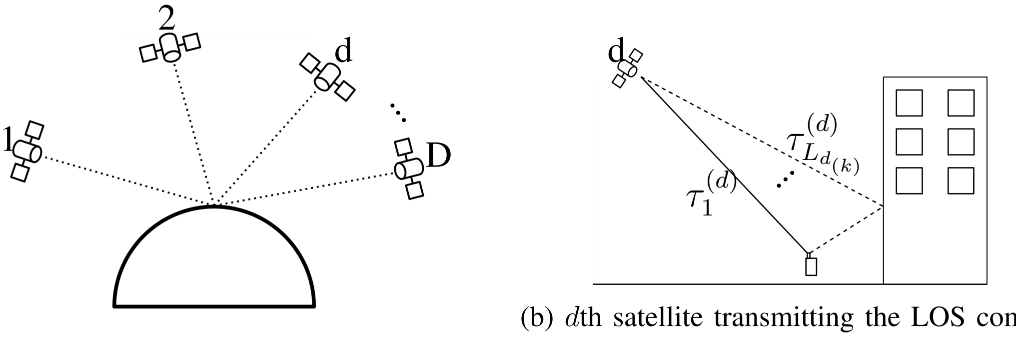
In this chapter we firstly introduce the tensor data model in Section 2.1. Moreover, in Section 2.2 we propose to use the RADOI method as a viable solution for model order selection in static scenarios. Furthermore, three tensor-based time-delay solutions, the state-of-the-art tensor-based time-delay estimation HOSVD, DoA/KRF, and CPD-GEVD from [4] and [9] are overviewed along with the HOSVD SECSI [15] in Section 2.3. In Subsection 2.3.5 we present the LOS selection method while in Subsection 2.5 we present the computational complexity of the state-of-the-art and proposed methods. Finally, in Subsection 2.4 we display the TDE simulation results after combining the RADOI method with the state-of-the-art factorization matrices.

## 2.1 Data Model

This section firstly introduces the scenario considered in this paper in Subsection 2.1.1. Subsection 2.1.2 describes how the signal tensor is constructed. Finally, in Subsection 2.1.3, the post-correlation data model is defined using both the GPS C/A code and the L1C pilot signal [11].

### 2.1.1 Scenario

We consider a GNSS receiver equipped with an antenna array with $M$ elements. We assume that for the received signals of $d = 1, \dots, D$ satellites, the LOS signal of the $d$th satellite is superimposed with $(L_{d_{(k)}} - 1)$ NLOS multipath components. The observations are collected during $K$ periods (or epochs) each with $N$ samples, where $k = 1, \dots, K$ and $n = 1, \dots, N$. Moreover, the total number of received signal components is $L_{(k)} = \sum_{d=1}^{D} L_{d_{(k)}}$, where $\ell_{(k)} = 1, \dots, L_{(k)}$ and $\ell_{d_{(k)}} = 1, \dots, L_{d_{(k)}}$ are the $\ell$th and $\ell_d$th component at the $k$th epoch. Furthermore, we assume that $\tau_1^{(d)}$ is the time-delay of the LOS component of the $d$th satellite, while $\tau_2^{(d)}, \ldots, \tau_{L_{d_{(k)}}}^{(d)}$ are the time-delays of the $(L_{d_{(k)}} - 1)$ NLOS components. Each satellite broadcasts the L1C pilot signal with carrier frequency $f_c = 1575.42$ MHz. The received signals are down-converted to baseband and sampled at a sampling rate of $f_s = 2B$, where $B$ is the one-sided signal bandwidth. As illustrated in Figure 2.1(b), we assume that $\tau_1^{(d)}$ is the time-delay of the LOS component of the $d$th satellite, while $\tau_2^{(d)}, \ldots, \tau_{L_{d_{(k)}}}^{(d)}$ are the time-delays of the $(L_{d_{(k)}} - 1)$ NLOS components. Each satellite broadcasts the L1C pilot signal with carrier frequency $f_c = 1575.42$. The received signals are down-converted to baseband and sampled at a sampling rate $f_s = 2B$, where $B$ is the one-sided signal bandwidth.



(a) $D$ satellites transmitting the LOS component impinging onto a GNSS receiver.

(b) $d$th satellite transmitting the LOS component and the NLOS interference generated by a reflection on a building.

Figure 2.1: Multipath scenario for an antenna array based GNSS receiver.

### 2.1.2 Pre-Correlation Data Model

As shown in [9], the received complex basedband signal at the output of the $M$ antennas of the antenna array can be expressed by a tensor model $K \times N \times M$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \tilde{\mathbf{\Gamma}}^{\mathrm{T}} \times_2 \tilde{\mathbf{C}}^{\mathrm{T}} \times_3 \tilde{\mathbf{A}} + \boldsymbol{\mathcal{N}} \tag{2.1}$$

where $\boldsymbol{\mathcal{I}}_{3,L} \in \mathbb{R}^{L \times L \times L}$ is the identity tensor

$$\tilde{\mathbf{\Gamma}}^{\mathrm{T}} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{L_{(k)}}] \in \mathbb{C}^{K \times L_{(k)}} \tag{2.2}$$

collects the complex amplitudes related to each signal component during $K$ epochs with

$$\boldsymbol{\gamma}_\ell \in \mathbb{C}^{K \times 1} = [\gamma_1, \ldots, \gamma_K] \tag{2.3}$$

gathering the complex amplitudes related to each epoch. The matrix

$$\begin{aligned}\tilde{\mathbf{C}} = \Big[&\mathbf{c}_1[\tau_1^{(1)}], \dots, \mathbf{c}_1[\tau_{\ell_{1_{(k)}}}^{(1)}], \dots, \\ &\mathbf{c}_D[\tau_{\ell_{d_{(k)}}}^{(d)}], \dots, \mathbf{c}_D[\tau_{L_{d_{(k)}}}^{(d)}]\Big] \in \mathbb{R}^{N\times L_{(k)}}\end{aligned} \tag{2.4}$$

comprises the sampled L1C pilot code sequence with $\mathbf{c}_d[\tau_{\ell_{d_{(k)}}}^{(d)}] \in \mathbb{C}^{N\times 1}$ collecting the periodically repeated pseudo random binary sequences (PRS) with time-delay $\tau_{\ell_d(k)}^{(d)}$ for each satellite and the respective multipath components. Then, the matrix

$$\tilde{\mathbf{A}} = [\mathbf{a}(\phi_1), \dots, \mathbf{a}(\phi_{L_{(k)}})] \in \mathbb{C}^{M\times L_{(k)}} \tag{2.5}$$

collects the array responses with $\mathbf{a}(\phi_{\ell_{(k)}}) \in \mathbb{C}^{M\times 1}$ being the steering vector of an antenna array with azimuth angle $\phi_{\ell_{(k)}}$ with $\ell_{(k)}$ components at the $k$th epoch. Moreover, $\boldsymbol{\mathcal{N}}$ is a white Gaussian noise tensor.

The tensor in (2.1) is composed of three dimensions, the first dimensions of size $K$ is related to each epoch, the second dimension of size $N$ is associated to the collected samples in each epoch, and the third dimension of size $M$ corresponds to the spatial diversity of the receive antenna array.

### 2.1.3 Post-Correlation Data Model

To separate the $L_{d_{(k)}}$ LOS signal and NLOS multipath components from the $d$th satellite, the GNSS receiver utilizes a correlator bank related to each satellite. Thus, the GNSS receiver applies $D$ correlator banks on the received signal, obtaining $D$ output signals. The $D$ correlator banks enable the code division multiple access (CDMA) scheme and, therefore, make it possible to estimate synchronization parameters such as time-delay and carrier phase of each satellite in parallel. Furthermore, the correlator bank compresses the signal while it preserves all the relevant information required to estimate the time-delay of the LOS and NLOS components. Moreover, to enable signal tracking with a correlator bank, coarse knowledge of the time-delay of each satellite is necessary, which can be achieved by standard acquisition as performed in any GNSS receiver as an initial step before parameter estimation and tracking. We define the $d$th correlator bank with $Q$ “taps” as

$$\mathbf{Q}_d = \big[\mathbf{c}_d[\tau_1], \dots, \mathbf{c}_d[\tau_Q]\big] \in \mathbb{C}^{N\times Q}, \tag{2.6}$$

with $\tau_1 < \ldots < \tau_Q$ and the $q$th delayed reference sequence $\mathbf{c}_d[\tau_q]$ corresponding to the $q$th tap. The left-hand singular value of the thin SVD of $\mathbf{Q}_d$ with

$$\mathbf{Q}_d = \mathbf{Q}_\omega^{(d)} \mathbf{\Sigma} \mathbf{V}^{\mathrm{H}} \tag{2.7}$$

and provides the correlator bank

$$\mathbf{Q}_{\omega}^{(d)} = \mathbf{Q}_d(\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}})^{-1} \tag{2.8}$$

which performs cross-correlation (compression) preserving the noise statistics [31] at the output of the correlator bank; however, the output, with respect to the signal, is changed. When comparing $\mathbf{Q}_{\omega}^{(d)}$ to the correlator bank $\mathbf{Q}_d$, we observe that $\mathbf{Q}_d$ provide a sampled cross-correlation function of the correlator bank with the received LOS component. Thus, according to [4], the received signal tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{K\times N\times M}$ can be correlated with $\mathbf{Q}_{\omega}^{(d)}$ to separate the $d$th satellite from all other received satellites and we obtain

$$\begin{aligned}
\boldsymbol{\mathcal{Y}} &= \boldsymbol{\mathcal{X}} \times_2 (\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} \\
&= \boldsymbol{\mathcal{I}}_{3,L_{d_{(k)}}} \times_1 \mathbf{\Gamma}^{\mathrm{T}} \times_2 (\mathbf{C}\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} \times_3 \mathbf{A} + \boldsymbol{\mathcal{N}} \times_2 (\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} + \boldsymbol{\mathcal{M}} \\
&= \boldsymbol{\mathcal{I}}_{3,L_{d_{(k)}}} \times_1 \mathbf{\Gamma}^{\mathrm{T}} \times_2 (\mathbf{C}\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} \times_3 \mathbf{A} + \boldsymbol{\mathcal{N}}_{\omega} + \boldsymbol{\mathcal{M}} \\
&\approx \boldsymbol{\mathcal{I}}_{3,L_{d_{(k)}}} \times_1 \mathbf{\Gamma}^{\mathrm{T}} \times_2 (\mathbf{C}\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} \times_3 \mathbf{A} + \boldsymbol{\mathcal{N}}_{\omega},
\end{aligned} \tag{2.9}$$

where $\boldsymbol{\mathcal{I}}_{3,L_{d_{(k)}}} \in \mathbb{R}^{L_{d_{(k)}}\times L_{d_{(k)}}\times L_{d_{(k)}}}$ is the identity tensor, $\mathbf{\Gamma}^{\mathrm{T}} \in \mathbb{C}^{K\times L_{d_{(k)}}}$ collects the complex amplitudes of the $L_{d_{(k)}}$ signal components of satellite $d$ obtained from matrix $\tilde{\mathbf{\Gamma}}^{\mathrm{T}}$, $(\mathbf{C}\mathbf{Q}_{\omega}^{(d)})^{\mathrm{T}} \in \mathbb{R}^{Q\times L_{d_{(k)}}}$, and $\mathbf{A} \in \mathbb{C}^{M\times L_{d_{(k)}}}$ comprizes the $L_{d_{(k)}}$ array responses of the signals of satellite $d$ from $\tilde{\mathbf{A}}$, therefore, $\mathbf{A}$ does not vary throughout the time since the changes of $L_{d_{(k)}}$ is related to the changes in the environment. Moreover, $\boldsymbol{\mathcal{N}}_{\omega} \in \mathbb{C}^{K\times Q\times M}$ is the white Gaussian noise tensor after correlation. The tensor $\boldsymbol{\mathcal{M}}$ is the multiple access interference (cross-correlation) of the other satellites and their respective multipath components. Moreover, $\boldsymbol{\mathcal{M}}$ is neglegible in comparison to other terms, since signals are decorrelated.

## 2.2 Model Order Selection for Static Environments

To decompose the tensor $\boldsymbol{\mathcal{Y}}$ into factor matrices for the time delay estimation, first the number of multipath components of the $d$th satellite, $L_{d_{(k)}}$, should be estimated. Moreover, in realistic scenarios, several satellites are transmitting from earth orbit. Therefore, even though signals are decorrelated, the signals from other satellites may interfere with the LOS component. Consequently, colored noise deteriorates the LOS component. Hence, the RADOI method provides a reliable model order selection under colored noise conditions. Thus, to perform the Model Order Selection using the tensor data model, RADOI utilizes the covariance matrix $\hat{\mathbf{R}}_{yy}$ obtained from the third-mode unfolding of tensor $[\boldsymbol{\mathcal{Y}}]_{(3)}$ from (2.9) to compute the Eigenvalue Decomposition (EVD)

$$\hat{\mathbf{R}}_{yy} = \frac{1}{KQ}[\boldsymbol{\mathcal{Y}}]_{(3)}[\boldsymbol{\mathcal{Y}}]_{(3)}]^{\mathrm{H}} \tag{2.10}$$

$$= \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{H}} + \mathbf{R}_{qq}, \tag{2.11}$$

where $\mathbf{R}_{yy} \in \mathbb{C}^{M \times M}$ is a Hermitian matrix, $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_M] \in \mathbb{C}^{M \times M}$ is a unitary matrix containing the eigenvectors, $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \dots, \lambda_M\} \in \mathbb{C}^{M \times M}$ is a diagonal matrix holding the the sorted eigenvalues $\lambda_i$, such that $\lambda_1 > \lambda_2 > \cdots > \lambda_M$, and the correlator bank covariance matrix $\mathbf{R}_{qq} \in \mathbb{C}^{M \times M}$. Moreover, we define $\mathbf{U}^{(s)} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_P] \in \mathbb{C}^{M \times P}$ as the truncated matrix composed of $P$ eigenvectors of $\mathbf{U}$ corresponding to the $P$ largest eigenvalues of $\boldsymbol{\Lambda}$. Therefore, in case that $P = L_{d_{(k)}}$, the dominant eigenvectors $\mathbf{U}^{(s)} \in \mathbb{C}^{M \times L_{d_{(k)}}}$ and column space of the steering matrix $\mathbf{A}$ have the same subspace.

Since estimating $\hat{L}_{d_{(k)}}$ is a decisive step, the RADOI algorithm solves the following optimization problem

$$\hat{L}_{d_{(k)}} = \arg\min_{P} \mathrm{RADOI}(P) \quad \text{where} \tag{2.12}$$

$$\mathrm{RADOI}(P) = \lambda_{P+1}\left(\sum_{i=2}^{M} \lambda_i\right)^{-1} - \xi_P \left(\sum_{i=1}^{M-1} \xi_i\right)^{-1}, \tag{2.13}$$

with

$$\xi_P = 1 - \frac{\alpha_R(\lambda_P - \hat{\sigma}_P^2)}{\hat{\sigma}_P^2} \tag{2.14}$$

$$\hat{\sigma}_P^2 = \frac{1}{M-P}\sum_{i=P+1}^{M} \lambda_i \tag{2.15}$$

and $\alpha_R$ is defined as

$$\alpha_R = \left[\arg\max_{P} \frac{(\lambda_P - \hat{\sigma}_P^2)}{\hat{\sigma}_P^2}\right]^{-1}, \tag{2.16}$$

where $\alpha_R$ is the inverse of the index $P$ corresponding to the greatest normalize difference between $\lambda_P$ and $\hat{\sigma}_P^2$, as illustrated in Figure 2.2.

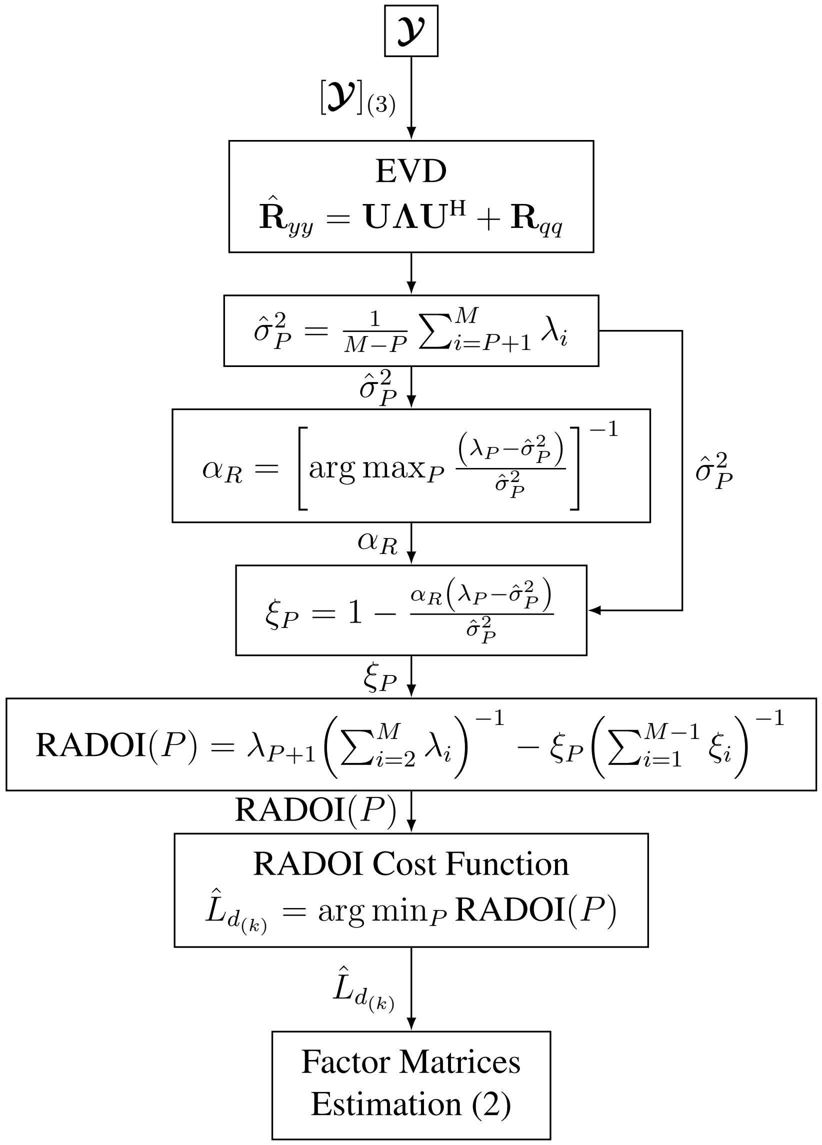


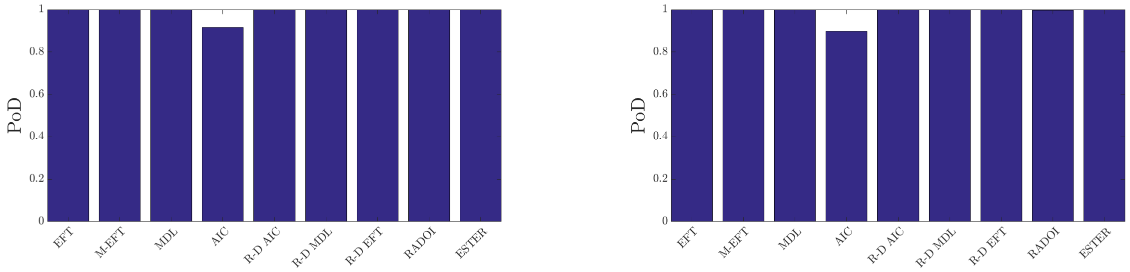
Figure 2.2: Block diagram for the RADOI model order selection method.

### 2.2.1 Simulation Results for Probability of Detection for Static Scenarios

Following [10], we consider a static scenario that consists of a left centro-hermitian ULA with $M = 8$ elements and half-wavelength $\Delta = \lambda/2$ spacing. The L1C pilot channel is transmitted by the satellites with $PRS = 3, 4, 17$ with a carrier frequency $f_s = 1575.42$ MHz. The simulations use the modulated L1C pilot channel with a total period $t_{\text{3rd}} = 10$ ms and with a bandwidth $B = 12.276$ MHz. The L1C pilot code samples are collected every $k$th epoch during $K = 30$ epochs with each epoch having a duration of $10$ ms [10]. Therefore, $N = 245520$ samples were collected for the L1C pilot code per epoch. The carrier-to-noise ratio is $C/N_0 = 48$ dB-Hz, resulting in a pre-correlation signal-to-noise ratio $SNR_{\text{pre}} = C/N_0 - 10\log_{10}(2B) \approx -25.10$ dB for GPS3. Given the processing gain $G = 10\log_{10}(Bt) \approx 50.9$ dB for GPS3. Hence, the post-correlation signal-to-noise ratio

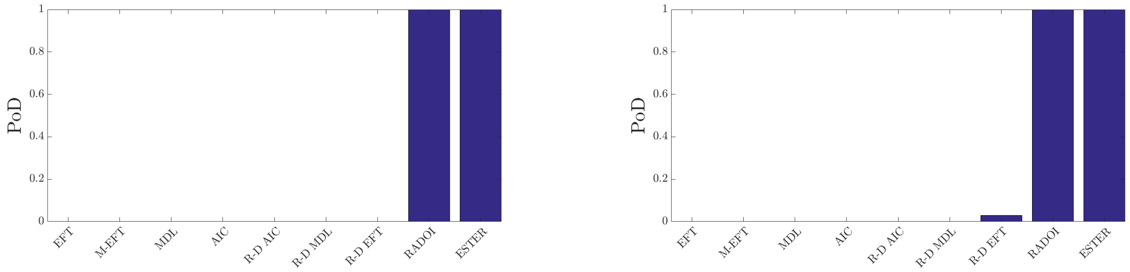
$SNR_{\text{post}} = SNR_{\text{pre}} + G \approx 25$ dB. Moreover, the signal-to-multipath ratio $\text{SMR}_1 = 5$ dB for $L_d = 2$. In case $L_d = 3$ the $\text{SMR}_1 = 5$ dB for the first NLOS signal, and a $\text{SMR}_2 = 10$ dB for the second NLOS signal. Besides the Probability of Detection (PoD) simulation using a perfect array, we added errors in the antenna array geometry to distort the antennas' $x$ and $y$ positions according to a normal distribution $\sim \mathcal{N}(0, \sigma^2)$. The standard deviation is computed in terms of the probability $p = \text{P}(e > \lambda/2)$ that the the error exceeds a half wavelength. We fix the relative delay $\Delta\tau$ at $0.5T_c$ while varying the error probability $p$ from $10^{-6}$ to $10^{-1}$. Additionally, we executed simulations using the matrix- and the tensor-based MOS techniques to perform model order estimation. Therefore, we use a Probability of False Alarm $P_{\text{fa}} = 10^{-6}$.

In this section, we present the Probability of Detection (PoD) as the probability of the estimated model order is equal to the actual model order. The PoD is computed during simulations when considering a perfectly aligned array of antennas. We compute the PoD at relative delay $\Delta\tau$ at $0.5T_c$.



(a) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 1$ and $L_d = 2$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

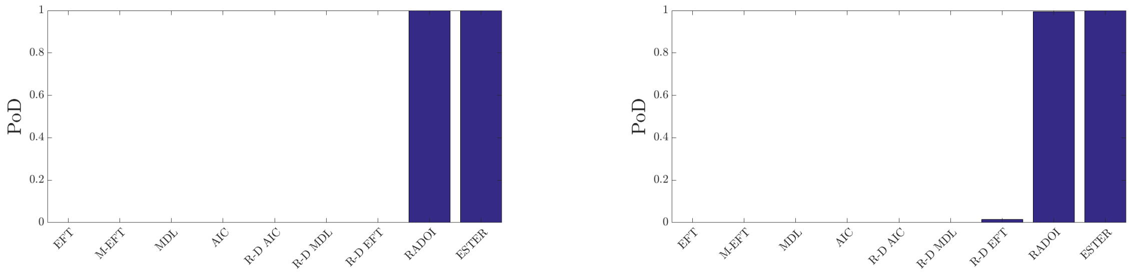
(b) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 1$ and $L_d = 3$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.



(c) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 2$ and $L_d = 2$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

(d) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 2$ and $L_d = 3$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Figure 2.3: PoD results for an Static Scenario.



(e) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 3$ and $L_d = 2$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

(f) Probability of Detection for MOS methods simulation with $M = 8$ antennas and $D = 3$ and $L_d = 3$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Figure 2.3: PoD results for an Static Scenario.

In Figure 2.3 we show the results of PoD for MOS methos. Note that each subfigure varies the number of satellites and the actual model order. Moreover, when $D = 1$, all MOS methods are capable of correctly estimating the model order since, in this scenario, there are no coloured noise. Finally, observe that as we add more satellites to the simulations, EFT, AIC, and MDL-based methods do not correctly estimate the model order. However, note that both RADOI and ESTER methods have a consistent performance achieving the same PoD for all simulations, as consolidated in Table 2.1.

| | PoD $d = 1$ and $L_d = 3$ | PoD $d = 3$ and $L_d = 3$ |
|---|---|---|
| EFT | 99% | 0% |
| M-EFT | 99% | 0% |
| MDL | 99% | 0% |
| AIC | 85% | 0% |
| R-D AIC | 99% | 0% |
| R-D MDL | 99% | 0% |
| R-D EFT | 99% | 1% |
| RADOI | 100% | 99% |
| ESTER | 100% | 99% |

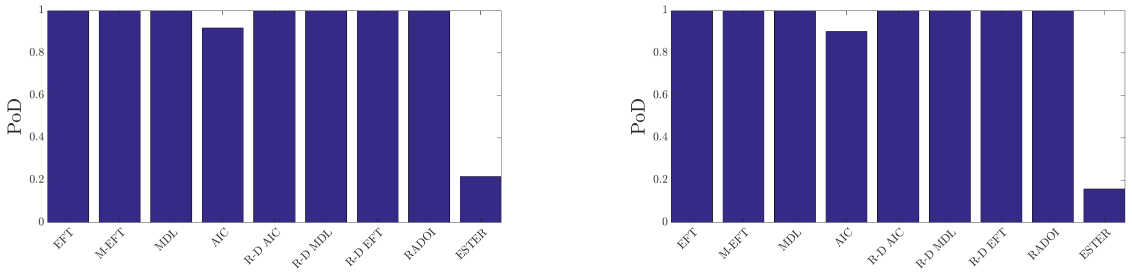
Table 2.1: Probability of Detection for MOS with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and $N = 245520$.

### 2.2.2 Probability of Detection Utilizing an Array with Errors Considering an Static Scenario

In this section we suppose an antenna array with errors with $D = 1$ and $L_d = 2$ and $D = 1$ and $L_d = 3$ impinging signals and a fixed relative delay $\Delta\tau = 0.5T_c$.

In Figure 2.4 we show the PoD for the MOS methods used in the simulation for $D = 1$ and $L_d = 2$. Therefore, note that when we have an imperfect array of antennas, the eigenvalue-based methods EFT, M-EFT, MDL, R-D AIC, R-D MDL, R-D EFET, and

RADOI are insensitive to array positioning errors. However, since the ESTER method assumes the matrix $\mathbf{A}$ has a Vandermonde structure, variations errors in the array reflects in the matrix $\mathbf{A}$ structure. Therefore, the subspace-based MOS ESTER shows to be sensitive to array imperfections. Moreover, we show that RADOI is the most accurate MOS method for GPS3.



(a) Probability of Detection for MOS methods simulation with an imperfect array with $M = 8$ antennas and $D = 1$ and $L_d = 3$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

(b) Probability of Detection for MOS methods simulation with an imperfect array with $M = 8$ antennas and $D = 1$ and $L_d = 3$ impinging signal. In both cases, code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Figure 2.4: Probability of Detection for MOS methods simulation with an imperfect array.

## 2.3 State-Of-The-Art Tensor-Based Time-Delay Estimation For Third Generation GPS

In this section we overview the state-of-the-art tensor-based time-delay estimation approaches. Firstly, in Subsection 2.3.1 we introduce an HOSVD based eigenfilter with Forward Backward Averaging (FBA) and Expanded Spatial Smoothing (ESPS). Then, in Subsection 2.3.2 we describe the Direction of Arrival (DoA) estimation and Khatri-Rao factorization approach. Next, in Subsection 2.3.3 we present the Canonical Polyadic Decomposition by Generalized Eigenvalue Decomposition (CPD-GEVD). Finally, in Subsection 2.3.4 we describe the Semi-algebraic Framework for Approximate Canonical Polyadic Decomposition via Simultaneous Matrix Diagonalization (SECSI)

### 2.3.1 HOSVD based Time-Delay Estimation

As shown in Figure 2.5, in order to perform the Higher-Order Eigenfilter decomposition [4], first the incoming signal $\mathcal{Y}$ is pre-processed to incorporate the Forward-Backward Averaging (FBA) [14], [5] and Expanded Spatial Smoothing (ESPS) [6], [7].

Similarly to [14], the tensor-based FBA uses flipped identity matrices in order to duplicate the number of samples. Thus, the left-hand identity matrix $\mathbf{\Pi}_M \in \mathbb{R}^{M \times M}$ is of size $M$ flipped along its vertical axis. Moreover, the right-hand identity matrix $\mathbf{\Pi}_{KQ} \in \mathbb{R}^{KQ \times KQ}$

is of size $KQ$ flipped along its vertical axis. Then, the identity matrices are applied to the third-mode unfolding of the received signal tensor:

$$\mathbf{Z} = \begin{bmatrix} [\boldsymbol{\mathcal{Y}}]_3 & \mathbf{\Pi}_M [\boldsymbol{\mathcal{Y}}]_3^* \mathbf{\Pi}_{KQ} \end{bmatrix} \in \mathbb{C}^{M \times 2KQ}. \tag{2.17}$$

Then, similarly to [6], the tensor-based ESPS uses selection matrices that separate the antenna array into $L_S$ subarrays with $M_S = M - L_S + 1$ elements. Therefore, we define the selection matrices as follows:

$$\mathbf{J}_{\ell_S} = \begin{bmatrix} \mathbf{0}_{M_S \times \ell_S - 1} & \mathbf{I}_{M_S} & \mathbf{0}_{M_S \times L_S - 1} \end{bmatrix} \in \mathbb{R}^{M_S \times M}, \tag{2.18}$$

where $\ell_S = 1, \ldots, L_S$. Thus, we use the selection matrices to apply the spatial smoothing to the FBA unfolding of the received signal tensor from (2.17)

$$\mathbf{W} = \begin{bmatrix} \mathbf{J}_1 \mathbf{Z} & \cdots & \mathbf{J}_{L_S} \mathbf{Z} \end{bmatrix} \in \mathbb{C}^{M_S \times 2L_S KQ}, \tag{2.19}$$

where $\mathbf{W}$ is folded back using the third-mode unfolding thus resulting in a forward-backward averaged spatially-smoothed fourth-order tensor $\boldsymbol{\mathcal{Z}}_{ESPS} \in \mathbb{C}^{2K \times Q \times M_S \times L_S}$.

Next, the resulting tensor $\boldsymbol{\mathcal{Z}}_{ESPS}$ is used to perform the HOSVD rank-one approximation on the space and epoch dimensions.

$$\boldsymbol{\mathcal{Z}}_{ESPS} = \boldsymbol{\mathcal{R}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \times_4 \mathbf{U}_4, \tag{2.20}$$

where $\boldsymbol{\mathcal{R}} \in \mathbb{C}^{2K \times Q \times M_S \times L_S}$ is the core tensor, $\mathbf{U}_1 \in \mathbb{C}^{2K \times 2K}$,$\mathbf{U}_2 \in \mathbb{C}^{Q \times Q}$,$\mathbf{U}_3 \in \mathbb{C}^{M_S \times M_S}$, and $\mathbf{U}_4 \in \mathbb{C}^{L_S \times L_S}$ are unitary matrices collecting singular vectors of each mode's unfolding [32] from (2.9).

Afterwards, once we assume the LOS component has the greatest power, the dominant singular vectors are multiplied by $\boldsymbol{\mathcal{Z}}_{ESPS}$. Then, the resulting vector is multiplied by the $\mathbf{\Sigma V}^{\mathrm{H}}$ from the thin SVD of $\mathbf{Q}$. Thus, resulting in the $\mathbf{q}_{ESPS}$ vector.

$$\mathbf{q}_{ESPS} = \left[ \boldsymbol{\mathcal{Z}}_{ESPS} \times_1 (\mathbf{u}_1^{(1)})^{\mathrm{H}} \times_3 (\mathbf{u}_1^{(3)})^{\mathrm{H}} \times_4 (\mathbf{u}_1^{(4)})^{\mathrm{H}} \right] \mathbf{\Sigma V}^{\mathrm{H}}, \tag{2.21}$$

where $\mathbf{q}_{ESPS}$ contains the multi-dimensionally filtered cross-correlation values at each tap of the correlator bank. The resulting vector in the correlator dimension is then interpolated using a cubic spline so that higher accuracy can be achieved.

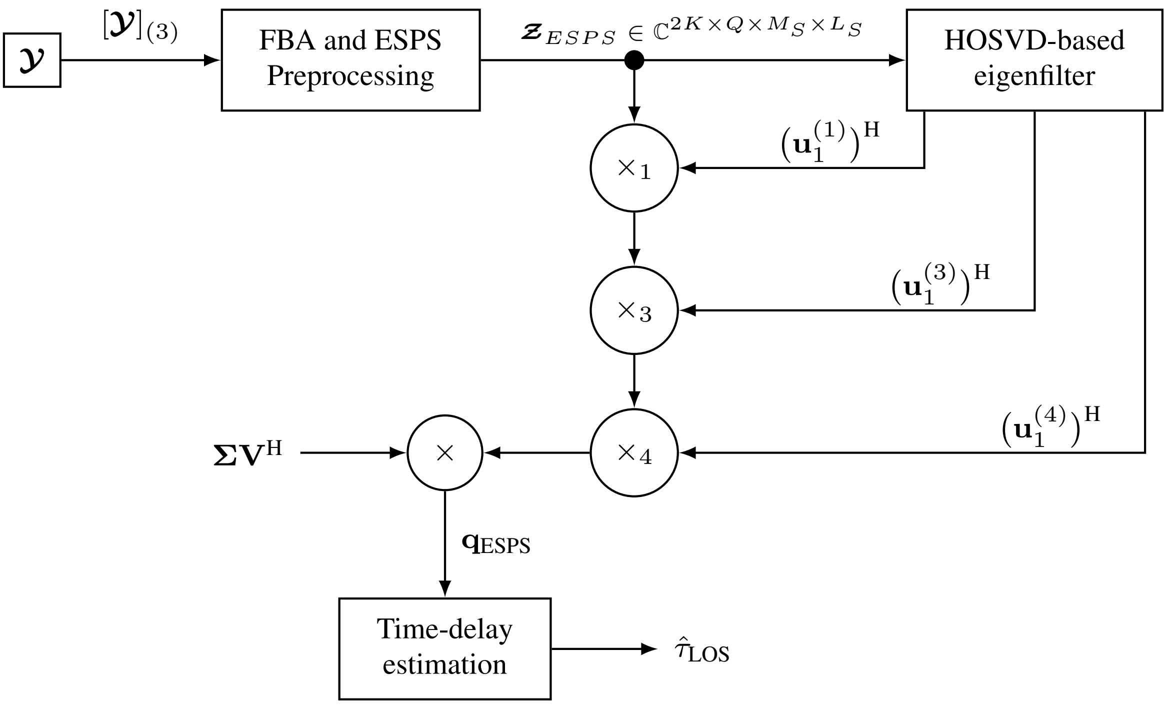


Figure 2.5: HOSVD Time-Delay Estimation block diagram.

### 2.3.2 DoA/KRF based Time-Delay Estimation

A three step approach based on direction of arrival (DoA) estimation, the Khatri-Rao factorization (KRF), and the selection of the estimated LOS component was proposed by [8]. In order to perform DoA estimation, as illustrated in Figure 2.6, firstly the received signal tensor is pre-processed using FBA and ESPS, thus we use the forward-backward averaged spatially smoothed signal matrix $\mathbf{W}$ from (2.19). As shown in Figure 2.6 the Estimation of Signal Parameter via Rotational Invariance Technique (ESPRIT) [33] is applied to $\mathbf{W}$ in order to estimate $\hat{\mathbf{A}}$.

Firstly we rewrite (2.9) to obtain the following equation

$$[\mathcal{Y}]_{(3)} = \mathbf{A}\Big[\boldsymbol{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\Big]^{\mathrm{T}} \in \mathbb{C}^{M \times KQ}. \tag{2.22}$$

Once matrix $\hat{\mathbf{A}}$ is estimated by the ESPRIT technique, its pseudo-inverse can be applied to (2.22) such that

$$\begin{aligned}\hat{\mathbf{A}}^{+}[\mathcal{Y}]_{3} &= \hat{\mathbf{A}}^{+}\mathbf{A}\Big[\boldsymbol{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\Big]^{\mathrm{T}} \\ &\approx \Big[\boldsymbol{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\Big]^{\mathrm{T}} \in \mathbb{C}^{L_{d(k)} \times KQ},\end{aligned} \tag{2.23}$$

where the factor matrices $\boldsymbol{\Gamma}$ and $(\mathbf{C}\mathbf{Q}_{\omega})$ can be estimated by Least Square Khatri-Rao factorization (LSKRF) [34, 35].

Once $(\mathbf{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}})^{\mathrm{T}}$ is given in (2.23), and considering that it's $\ell_{d_{(k)}}$-th column can be computed as the Khatri-Rao product of the $\ell_{d_{(k)}}$-th column of $\mathbf{\Gamma}^{\mathrm{T}}$ and $(\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}$, then

$$\left[\mathbf{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\right](:, \ell_{d_{(k)}}) = (\mathbf{\Gamma}^{\mathrm{T}})(:, \ell_{d_{(k)}}) \diamond (\mathbf{C}\mathbf{Q}_{\omega})(:, \ell_{d_{(k)}}), \tag{2.24}$$

where each column $(\mathbf{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}})(:, \ell_{d_{(k)}}) \in \mathbb{C}^{KQ}$. Thus, in order to solve the estimates of $\mathbf{\Gamma}$ and $\mathbf{C}\mathbf{Q}_{\omega}$, we reshape (2.24) into a matrix of size $Q \times K$ as

$$\underset{Q\times K}{\mathrm{unvec}}\left\{\left[\mathbf{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\right](:, \ell_{d_{(k)}})\right\} = (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}(:, \ell_{d_{(k)}})(\mathbf{\Gamma}^{\mathrm{T}})^{\mathrm{T}}(:, \ell_{d_{(k)}}). \tag{2.25}$$

Once (2.25) is a rank-one matrix, we can perform the SVD-based rank-one approximation

$$\underset{Q\times K}{\mathrm{unvec}}\left\{\left[\mathbf{\Gamma}^{\mathrm{T}} \diamond (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}\right](:, \ell_{d_{(k)}})\right\} = \mathbf{U}_{\ell_{d_{(k)}}} \mathbf{\Sigma}_{\ell_{d_{(k)}}} \mathbf{V}_{\ell_{d_{(k)}}}. \tag{2.26}$$

Therefore, the estimates for $(\mathbf{\Gamma}^{\mathrm{T}})(:, \ell_{d_{(k)}})$ and $((\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}})(:, \ell_{d_{(k)}})$ are $\sqrt{\sigma_{\ell_{d_{(k)}},1}}\mathbf{v}^{*}_{\ell_{d_{(k)}},1}$ and $\sqrt{\sigma_{\ell_{d_{(k)}},1}}\mathbf{u}_{\ell_{d_{(k)}},1}$, respectively, where $\sigma_{\ell_{d_{(k)}},1}$ is the dominant singular value of $\mathbf{\Sigma}_{\ell_{d_{(k)}}}$, $\mathbf{v}^{*}_{\ell_{d_{(k)}},1}$ is the conjugate of the dominant right singular vector of $\mathbf{V}_{\ell_{d_{(k)}}}$, and $\mathbf{u}_{\ell_{d_{(k)}},1}$ is the dominant left singular vector of $\mathbf{U}_{\ell_{d_{(k)}}}$. This is repeated for $\ell_{d_{(k)}} = 1, \ldots, L_{d_{(k)}}$.

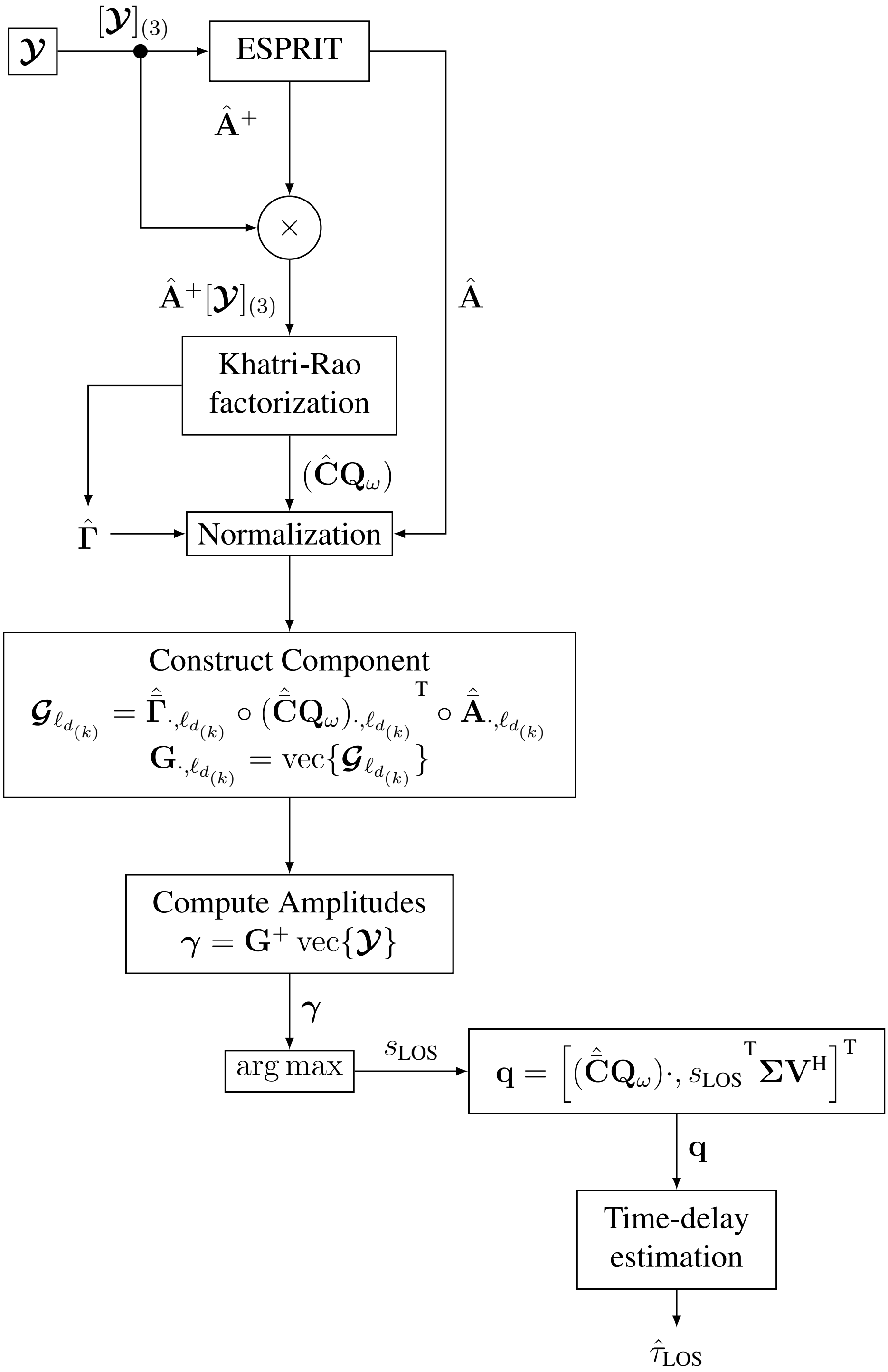


Figure 2.6: DoA/KRF Time-delay estimation block

### 2.3.3 CPD-GEVD based Time-Delay Estimation

A more accurate tensor-based scheme has been proposed for time delay estimation was proposed in [36]. This method computes the Canonical Polyadic Decomposition, also known as Candecomp and PARAFAC, by Generalized Eigenvalue Decomposition. In Figure 2.7 we present a more accurate block diagram yet equivalent to the one presented in [9]. As illustrated in Figure 2.7, the CPD-GEVD firstly computes the HOSVD low-rank approximation of the incoming signal $\boldsymbol{\mathcal{Y}}$.

$$\boldsymbol{\mathcal{Y}} \approx \boldsymbol{\mathcal{S}}^{\mathrm{g}} \times_1 \mathbf{U}_1^{\mathrm{g}} \times_1 \mathbf{U}_2^{\mathrm{g}} \times_3 \mathbf{U}_3^{\mathrm{g}}, \tag{2.27}$$

where $\mathbf{U}_1^{\mathrm{g}} \in \mathbb{C}^{K \times L_{d(k)}}$, $\mathbf{U}_2^{\mathrm{g}} \in \mathbb{C}^{Q \times L_{d(k)}}$, and $\mathbf{U}_3^{\mathrm{g}} \in \mathbb{C}^{M \times L_{d(k)}}$, and $\boldsymbol{\mathcal{S}}^{\mathrm{g}} \in \mathbb{C}^{L_{d(k)} \times L_{d(k)} \times L_{d(k)}}$ is the core tensor and can be expressed as a PARAFAC decomposition:

$$\boldsymbol{\mathcal{S}}^{\mathrm{g}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1^{\mathrm{g}} \times_2 \mathbf{T}_2^{\mathrm{g}} \times_3 \mathbf{T}_3^{\mathrm{g}}, \tag{2.28}$$

where the first two frontal slices of $\boldsymbol{\mathcal{S}}^{\mathrm{g}}$ can be expressed as:

$$\begin{aligned} (\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,1} &= \mathbf{T}_1^{\mathrm{g}} \operatorname{diag}\{(\mathbf{T}_3^{\mathrm{g}})_{\cdot,1}\}(\mathbf{T}_2^{\mathrm{g}})^{\mathrm{T}} \\ (\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,2} &= \mathbf{T}_1^{\mathrm{g}} \operatorname{diag}\{(\mathbf{T}_3^{\mathrm{g}})_{\cdot,2}\}(\mathbf{T}_2^{\mathrm{g}})^{\mathrm{T}}. \end{aligned} \tag{2.29}$$

As shown in (2.30), the CPD-GEVD uses the eigenvectors, $\mathbf{E}$, from the GEVD of the matrix pencil formed by $(\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,1}$ and $(\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,2}$:

$$(\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,1}^{\mathrm{T}}\mathbf{E} = (\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,2}^{\mathrm{T}}\mathbf{E}\mathbf{D}, \tag{2.30}$$

where $\mathbf{D}$ the eigenvalues in its diagonal. Note that (2.30) is equivalent to

$$\begin{aligned} (\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,2}^{-\mathrm{T}}(\boldsymbol{\mathcal{S}}^{\mathrm{g}})_{\cdot,\cdot,1}^{\mathrm{T}} &= \mathbf{E}\mathbf{D}\mathbf{E}^{-1} \\ &= (\mathbf{T}_1^{\mathrm{g}})^{-\mathrm{T}} \operatorname{diag}\{(\mathbf{T}_3^{\mathrm{g}})_{\cdot,2}\}^{-1} \operatorname{diag}\{(\mathbf{T}_3^{\mathrm{g}})_{\cdot,1}\}\mathbf{T}_1^{\mathrm{g}}. \end{aligned} \tag{2.31}$$

Equation (2.31) is a diagonalization problem in which $\mathbf{E}$ provides an estimate of $(\mathbf{T}_1^{\mathrm{g}})^{-\mathrm{T}}$. Therefore, by combining $(\mathbf{T}_1^{\mathrm{g}})^{-\mathrm{T}}$ from (2.31) and $\mathbf{U}_1^{*}$ from (2.27) the factor matrix $\boldsymbol{\Gamma}^{\text{-T}}$ can be computed as

$$\hat{\boldsymbol{\Gamma}}^{+\mathrm{T}} = (\mathbf{U}_1^{\mathrm{g}})^{*}(\mathbf{T}_1^{\mathrm{g}})^{-\mathrm{T}} = (\mathbf{U}_1^{\mathrm{g}})^{*}\mathbf{E}. \tag{2.32}$$

Next, since in the noiseless case, according to (2.9),

$$[\boldsymbol{\mathcal{Y}}]_{(1)}^{\mathrm{T}} = \Big[(\mathbf{C}\mathbf{Q}_{\omega}) \diamond \mathbf{A}\Big]\boldsymbol{\Gamma}^{\mathrm{T}}, \tag{2.33}$$

and since $\hat{\boldsymbol{\Gamma}}^{\text{-T}}$ is given in (2.32), we define $\mathbf{F}^{\mathrm{g}(2,3)}$ as

$$\begin{aligned}\mathbf{F}^{\mathrm{g}(2,3)} &= [\boldsymbol{\mathcal{Y}}]_{(1)}^{\mathrm{T}}(\mathbf{U}_1^{\mathrm{g}})^{*}\mathbf{E} = \Big[(\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}} \diamond \mathbf{A}\Big]\boldsymbol{\Gamma}\hat{\boldsymbol{\Gamma}}^{+} \\ &\approx \Big[(\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}} \diamond \mathbf{A}\Big] \in \mathbb{C}^{QM\times L}.\end{aligned} \tag{2.34}$$

Then by performing the Least Squares Khatri-Rao Factorization (LSKRF)[34, 35], we can estimate the factor matrices $(\hat{\mathbf{C}}\mathbf{Q}_{\omega})^{\mathrm{T}}$ and $\hat{\mathbf{A}}$. Furthermore, we use the estimated factor matrices to perform the time-delay estimation technique described in Subsection 2.3.6

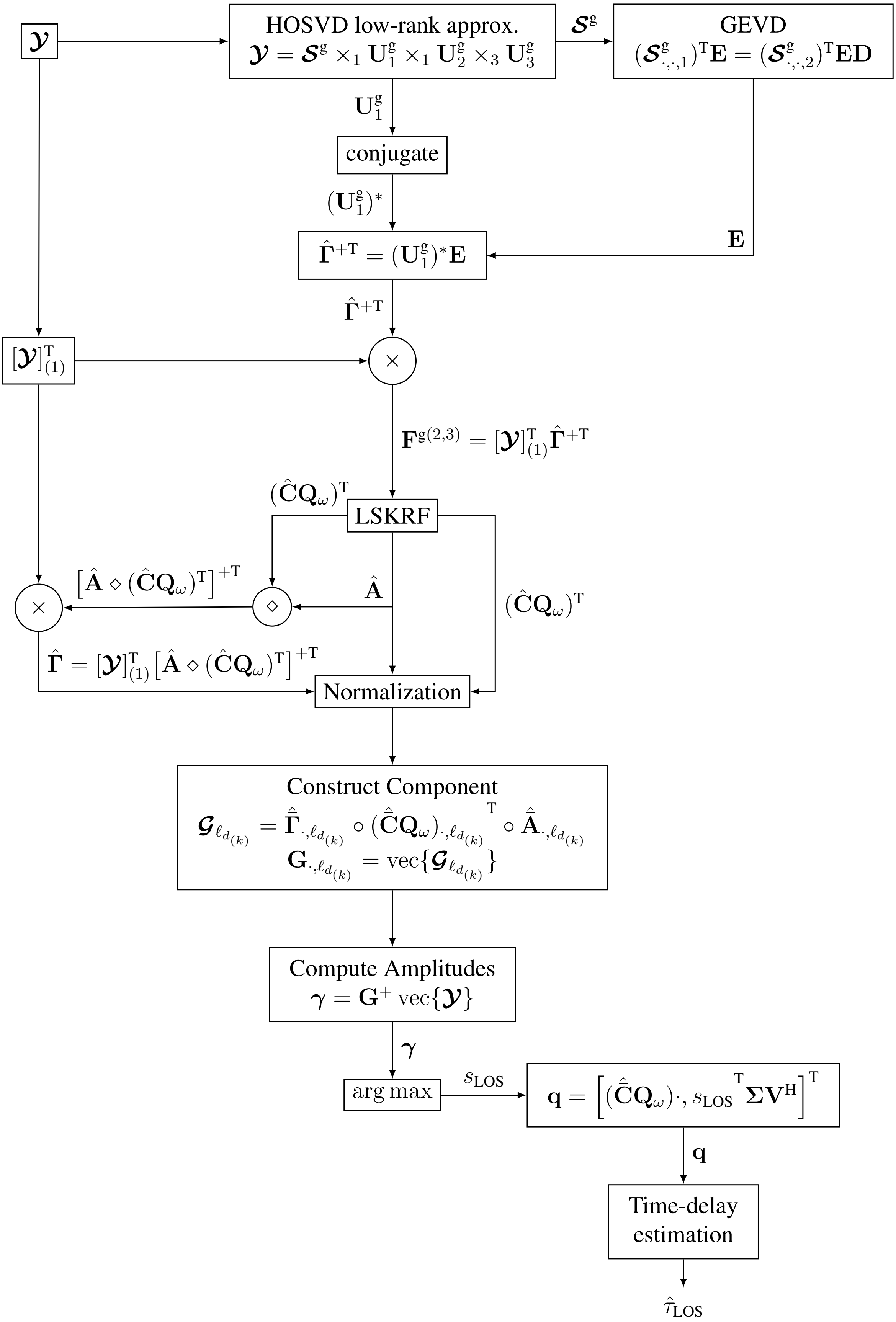


Figure 2.7: CPD-GEVD Time-Delay Estimation block diagram

### 2.3.4 HOSVD SECSI Based Time-Delay Estimation

In this section, we propose to utilize the Semi-algebraic framework for approximate Canonical Polyadic Decomposition via simultaneous matrix diagonalization (SECSI) from [24, 25, 26] to transforming our factor estimation problem into several redundant simultaneous matrix diagonalization problems. Therefore, we propose a SECSI based framework for time-delay estimation. Thus, firstly, as illustrated in Figure 2.8, the SECSI approach computes the Higher Order Singular Value Decomposition (HOSVD) low-rank approximation, which computes the thin-SVD of the incoming signal $\boldsymbol{\mathcal{Y}}$ from Equation (2.9) is:

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{S}}^{\mathrm{c}} \times_1 \mathbf{U}_1^{\mathrm{c}} \times_2 \mathbf{U}_2^{\mathrm{c}} \times_3 \mathbf{U}_3^{\mathrm{c}}, \tag{2.35}$$

where $\boldsymbol{\mathcal{S}}^{\mathrm{c}} \in \mathbb{C}^{L_d \times L_d \times L_d}$ is the compressed core tensor, and $\mathbf{U}_1^{\mathrm{c}} \in \mathbb{C}^{K \times L_d}$, $\mathbf{U}_2^{\mathrm{c}} \in \mathbb{C}^{Q \times L_d}$, $\mathbf{U}_3^{\mathrm{c}} \in \mathbb{C}^{M \times L_d}$ are the singular matrices. Thus, we can represent the tensor $\boldsymbol{\mathcal{S}}$ as follows

$$\boldsymbol{\mathcal{S}}^{\mathrm{c}} = \boldsymbol{\mathcal{I}} \times_1 \mathbf{T}_1^{\mathrm{c}} \times_2 \mathbf{T}_2^{\mathrm{c}} \times_3 \mathbf{T}_3^{\mathrm{c}}, \tag{2.36}$$

where $\mathbf{T}_1^{\mathrm{c}} \in \mathbb{C}^{L_d \times L_d}$, $\mathbf{T}_2^{\mathrm{c}} \in \mathbb{C}^{L_d \times L_d}$, and $\mathbf{T}_3^{\mathrm{c}} \in \mathbb{C}^{L_d \times L_d}$. Thus

$$\mathbf{U}_1^{\mathrm{c}}\mathbf{T}_1^{\mathrm{c}} = \mathbf{\Gamma}^{\mathrm{T}}, \tag{2.37}$$

$$\mathbf{U}_2^{\mathrm{c}}\mathbf{T}_2^{\mathrm{c}} = (\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{T}}, \tag{2.38}$$

$$\mathbf{U}_3^{\mathrm{c}}\mathbf{T}_3^{\mathrm{c}} = \mathbf{A}^{\mathrm{T}}. \tag{2.39}$$

Once we perform joint matrix diagonalization on our tensor $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$, we have several diagonalization problems to be solved. Because we have a third-order tensor, we can have six different estimates which means we have two estimates for each dimension of tensor $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$. Thus, in order to compute the joint matrix diagonalization we firstly compute the first-, second-, and third-mode slice of the $i$th slice of tensor $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$. Thus, we have the third-mode slice of $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$ represented as

$$\begin{aligned} \mathbf{S}_{3,i}^{c} &= \left[\left(\boldsymbol{\mathcal{S}}^{\mathrm{c}} \times_3 \mathbf{U}_3^{\mathrm{c}}\right) \times_3 \mathbf{e}_i^{\mathrm{T}}\right] \\ &= \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{\mathbf{A}^{\mathrm{H}}(:,i)\}(\mathbf{T}_2^{\mathrm{c}})^{\mathrm{T}}, \end{aligned} \tag{2.40}$$

second-mode slice of $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$ as

$$\begin{aligned} \mathbf{S}_{2,i}^{c} &= \left[\left(\boldsymbol{\mathcal{S}}^{\mathrm{c}} \times_2 \mathbf{U}_2^{\mathrm{c}}\right) \times_2 \mathbf{e}_i^{\mathrm{T}}\right] \\ &= \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{(\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{H}}(:,i)\}(\mathbf{T}_3^{\mathrm{c}})^{\mathrm{T}}, \end{aligned} \tag{2.41}$$

and, finally, we have the first-mode slice

$$\begin{aligned}\mathbf{S}_{1,i}^{c} &= \left[\left(\boldsymbol{\mathcal{S}}^{\mathrm{c}} \times_1 \mathbf{U}_1^{\mathrm{c}}\right) \times_1 \mathbf{e}_i^{\mathrm{T}}\right] \\ &= \mathbf{T}_2^{\mathrm{c}} \operatorname{diag}\{\boldsymbol{\Gamma}^{\mathrm{H}}(:,i)\}(\mathbf{T}_3^{\mathrm{c}})^{\mathrm{T}},\end{aligned} \tag{2.42}$$

then

$$\mathbf{S}_{3,p}^{\mathrm{c}} = \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{\mathbf{A}^{\mathrm{H}}(:,p)\}(\mathbf{T}_2^{\mathrm{c}})^{\mathrm{T}}, \tag{2.43}$$

$$\mathbf{S}_{2,p}^{\mathrm{c}} = \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{(\mathbf{C}\mathbf{Q}_{\omega})^{\mathrm{H}}(:,p)\}(\mathbf{T}_3^{\mathrm{c}})^{\mathrm{T}}, \tag{2.44}$$

$$\mathbf{S}_{1,p}^{\mathrm{c}} = \mathbf{T}_2^{\mathrm{c}} \operatorname{diag}\{\boldsymbol{\Gamma}^{\mathrm{H}}(:,p)\}(\mathbf{T}_3^{\mathrm{c}})^{\mathrm{T}}, \tag{2.45}$$

where $\mathbf{e}_i^{\mathrm{T}}$ is vector with zeros in all positions except in the $i$th position, and $p$ is an arbitrary index between one and the $n$th mode slice to be diagonalized:

$$p = \arg\min_{i} \operatorname{cond}\{\mathbf{S}_{n,i}^{\mathrm{c}}\}, \tag{2.46}$$

where $\operatorname{cond}\{\cdot\}$ computes the condition number of a matrix. The smaller the condition number, the more stable is the matrix inversion. Therefore, we select the matrix with the smallest condition number.

Therefore, with each mode slice defined, we can compute the right-hand and left-hand matrices of each mode slice. Note that the right-hand matrix simultaneous diagonalization for the third-mode slice is similar to the state-of-the-art CPD-GEVD described in Subsection 2.3.3. Since $p$ is fixed, we can vary all possible values of $i$, thus obtaining $N-1$ equations for each mode slice, since $i \neq p$.

Thus, firstly, we define the right-hand matrix for the third-mode slice in the following fashion:

$$\begin{aligned}\mathbf{S}_{3,i}^{\mathrm{c,rhs}} &= \mathbf{S}_{3,i}^{\mathrm{c}}(\mathbf{S}_{3,p}^{\mathrm{c}})^{-1} \\ &= \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{\mathbf{A}^{\mathrm{H}}(:,i)\mathbf{A}^{\mathrm{H}}(:,p)\}(\mathbf{T}_1^{\mathrm{c}})^{-1} \\ &= \mathbf{T}_1^{\mathrm{c}}\mathbf{A}^{\mathrm{H}}(\mathbf{T}_1^{\mathrm{c}})^{-1},\end{aligned} \tag{2.47}$$

then, we define the left-hand matrix for the third-mode slice as follows:

$$\begin{aligned}\mathbf{S}_{3,i}^{\mathrm{c,lhs}} &= ((\mathbf{S}_{3,p}^{\mathrm{c}})^{-1}\mathbf{S}_{3,i}^{\mathrm{c}})^{\mathrm{T}} = (\mathbf{S}_{3,i}^{\mathrm{c}})^{\mathrm{T}}(\mathbf{S}_{3,p}^{\mathrm{c}})^{-T} \\ &= \mathbf{T}_1^{\mathrm{c}} \operatorname{diag}\{\mathbf{A}^{\mathrm{H}}(:,i)\mathbf{A}^{\mathrm{H}}(:,p)\}(\mathbf{T}_2^{\mathrm{c}})^{-1} \\ &= \mathbf{T}_2^{\mathrm{c}}\mathbf{A}^{\mathrm{H}}(\mathbf{T}_2^{\mathrm{c}})^{-1}.\end{aligned} \tag{2.48}$$

This way, we transformed the third-mode slice onto two joint matrix diagonalization problems. Thus, we can obtain an estimate of $\mathbf{A}$ from the joint matrix diagonalization of $\mathbf{S}_{3,i}^{\text{c,rhs}}$. Additionally, we can obtain an estimate of $\mathbf{A}$ from the joint diagonalization of $\mathbf{S}_{3,i}^{\text{c,lhs}}$. Furthermore, the matrices that diagonalize $\mathbf{S}_{3,i}^{\text{c,rhs}}$ and $\mathbf{S}_{3,i}^{\text{c,lhs}}$ outputs estimates for $\mathbf{T}_1^{\text{c}}$, and $\mathbf{T}_2^{\text{c}}$. Thus, we can obtain the estimates for $\mathbf{U}_1^{\text{c}}\mathbf{T}_1^{\text{c}} = \hat{\mathbf{\Gamma}}^{\text{T}}$, and $\mathbf{U}_2^{\text{c}}\mathbf{T}_2^{\text{c}} = (\hat{\mathbf{C}}\mathbf{Q}_\omega)^{\text{T}}$.

Similarly, for the second-mode slice, we compute the right-hand matrix in the following fashion:

$$\begin{aligned}\mathbf{S}_{2,i}^{\text{c,rhs}} &= \mathbf{S}_{2,i}^{\text{c}}(\mathbf{S}_{2,p}^{\text{c}})^{-1} \\ &= \mathbf{T}_1^{\text{c}}\,\text{diag}\{(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(:,i)(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(:,p)\}(\mathbf{T}_1^{\text{c}})^{-1} \\ &= \mathbf{T}_1^{\text{c}}(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(\mathbf{T}_1^{\text{c}})^{-1},\end{aligned} \tag{2.49}$$

then we define left-hand matrix of the second-mode slice

$$\begin{aligned}\mathbf{S}_{2,i}^{\text{c,lhs}} &= ((\mathbf{S}_{2,p}^{\text{c}})^{-1}\mathbf{S}_{2,i}^{\text{c}})^{T} = (\mathbf{S}_{2,i}^{\text{c}})^{\text{T}}(\mathbf{S}_{2,p}^{\text{c}})^{-T} \\ &= \mathbf{T}_3^{\text{c}}\,\text{diag}\{(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(:,i)(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(:,p)\}(\mathbf{T}_3^{\text{c}})^{-1} \\ &= \mathbf{T}_3^{\text{c}}(\mathbf{C}\mathbf{Q}_\omega)^{\text{H}}(\mathbf{T}_3^{\text{c}})^{-1}.\end{aligned} \tag{2.50}$$

Again, we created two joint matrix diagonalization problems for the second-mode slice. Therefore, we can acquire two estimates of $(\mathbf{C}\mathbf{Q}_\omega)$ from the joint diagonalization of $\mathbf{S}_{2,i}^{\text{c,rhs}}$ and $\mathbf{S}_{2,i}^{\text{c,lhs}}$. Additionally, these joint diagonalization yield estimates for $\mathbf{T}_1^{\text{c}}$ and $\mathbf{T}_3^{\text{c}}$. Therefore, we can estimate the factor matrices $\mathbf{U}_1^{\text{c}}\mathbf{T}_1^{\text{c}} = \hat{\mathbf{\Gamma}}^{\text{T}}$, and $\mathbf{U}_3^{\text{c}}\mathbf{T}_3^{\text{c}} = \hat{\mathbf{A}}^{\text{T}}$.

Finally, we have to define the right-hand and left-hand matrices for the first-mode slice. Again, we firstly define the right-hand matrix in the following fashion:

$$\begin{aligned}\mathbf{S}_{1,i}^{\text{c,rhs}} &= \mathbf{S}_{1,i}^{\text{c}}(\mathbf{S}_{1,p}^{\text{c}})^{-1} \\ &= \mathbf{T}_2^{\text{c}}\,\text{diag}\{\mathbf{\Gamma}^{\text{H}}(:,i)\mathbf{\Gamma}^{\text{H}}(:,p)\}(\mathbf{T}_2^{\text{c}})^{-1} \\ &= \mathbf{T}_2^{\text{c}}\mathbf{\Gamma}^{\text{H}}(\mathbf{T}_2^{\text{c}})^{-1},\end{aligned} \tag{2.51}$$

then we define the left-hand matrix

$$\begin{aligned}\mathbf{S}_{1,i}^{\text{c,lhs}} &= ((\mathbf{S}_{1,p}^{\text{c}})^{-1}\mathbf{S}_{1,i}^{\text{c}})^{\text{T}} = (\mathbf{S}_{1,i}^{\text{c}})^{\text{T}}(\mathbf{S}_{1,p}^{\text{c}})^{-T} \\ &= \mathbf{T}_3^{\text{c}}\,\text{diag}\{\mathbf{\Gamma}^{\text{H}}(:,i)\mathbf{\Gamma}^{\text{H}}(:,p)\}(\mathbf{T}_3^{\text{c}})^{-1} \\ &= \mathbf{T}_3^{\text{c}}\mathbf{\Gamma}^{\text{H}}(\mathbf{T}_3^{\text{c}})^{-1}.\end{aligned} \tag{2.52}$$

Thus, the joint diagonalization problems $\mathbf{S}_{1,i}^{\mathrm{c,rhs}}$ and $\mathbf{S}_{1,i}^{\mathrm{c,lhs}}$ for the first-mode slice yield two estimates for $\mathbf{\Gamma}$. Furthermore, the joint diagonalization results in estimates for $\mathbf{T}_2^{\mathrm{c}}$ and $\mathbf{T}_3^{\mathrm{c}}$. Thus, we can estimate the factor matrices $\mathbf{U}_2^{\mathrm{c}}\mathbf{T}_2^{\mathrm{c}} = (\hat{\mathbf{C}}\mathbf{Q}_\omega)^{\mathrm{T}}$, and $\mathbf{U}_3^{\mathrm{c}}\mathbf{T}_3^{\mathrm{c}} = \hat{\mathbf{A}}^{\mathrm{T}}$.

Through performing various simulations using all mode slices previously described, we discovered that the right-hand matrices and left-hand matrices of each mode slice yielded similar time-delay estimation to the state-of-the-art CPD-GEVD method for a scenario with one LOS and one NLOS. However, we too discovered that when we have one impinging LOS and two NLOS the right-hand matrix of the third-mode slice have the best performance. Thus, we propose to only use the estimate provided by the joint diagonalization of $\mathbf{S}_{3,i}^{\mathrm{c,rhs}}$. Therefore, our goal is to find $\hat{\mathbf{T}}_1^{\mathrm{c}}$ that simultaneously diagonalizes the $N-1$ equations for the right-hand matrix of the second-mode slice. We refer here to the techniques in [37] and [38]. Then, we use $\mathbf{U}_1^{\mathrm{c}}$ from (2.35) to estimate $\hat{\mathbf{\Gamma}}^{\mathrm{T}}$

$$\mathbf{U}_1^{\mathrm{c}}\hat{\mathbf{T}}_1^{\mathrm{c}} = \hat{\mathbf{\Gamma}}^{\mathrm{T}}, \tag{2.53}$$

Next, since the noiseless case, according to (2.9) the first-mode unfolding of $\boldsymbol{\mathcal{Y}}$ can be described as

$$[\boldsymbol{\mathcal{Y}}]_{(1)}^{\mathrm{T}} = \Big[(\mathbf{C}\mathbf{Q}_\omega) \diamond \mathbf{A}\Big]\mathbf{\Gamma}, \tag{2.54}$$

and once $\hat{\mathbf{\Gamma}}^{\mathrm{T}}$ is given in (2.53), we define $\mathbf{F}^{\mathrm{c}(2,3)}$ as

$$\begin{aligned} \mathbf{F}^{\mathrm{c}(2,3)} = [\boldsymbol{\mathcal{Y}}]_{(1)}^{\mathrm{T}}\hat{\mathbf{\Gamma}}^{+\mathrm{T}} &= \Big[(\mathbf{C}\mathbf{Q}_\omega)^{\mathrm{T}} \diamond \mathbf{A}\Big]\mathbf{\Gamma}\hat{\mathbf{\Gamma}}^{+\mathrm{T}} \\ &\approx \Big[(\mathbf{C}\mathbf{Q}_\omega)^{\mathrm{T}} \diamond \mathbf{A}\Big] \in \mathbb{C}^{QM\times L_d}. \end{aligned} \tag{2.55}$$

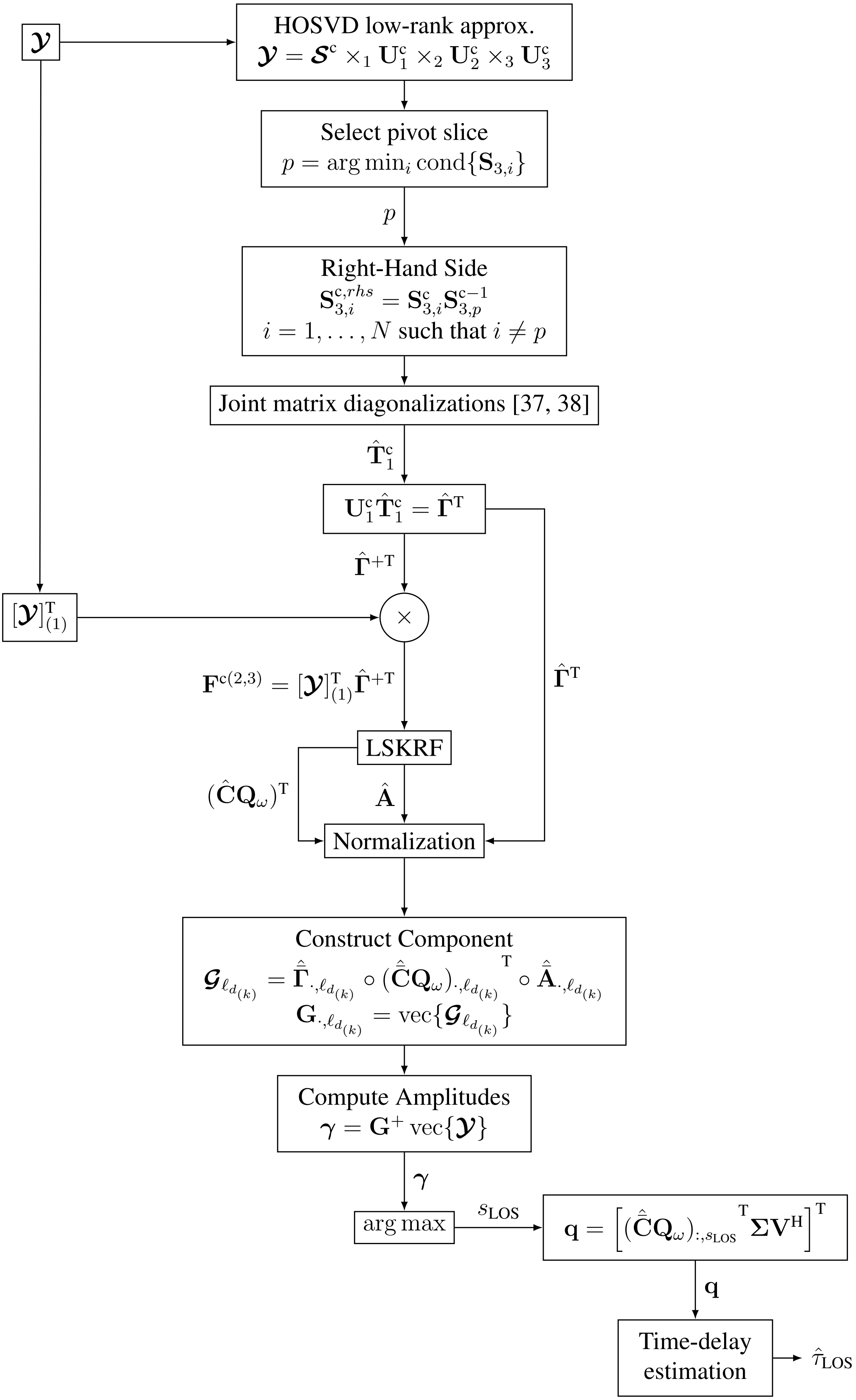


Figure 2.8: Proposed SECSI based time-delay estimation block diagram using the right-hand matrix from the second dimension of $\boldsymbol{\mathcal{S}}^{\mathrm{c}}$

The factor matrices $(\hat{\mathbf{C}}\mathbf{Q}_{\omega})^{\mathrm{T}}$ and $\hat{\mathbf{A}}$ can be estimated from (2.55) by applying the Least Squares Khatri-Rao Factorization (LSKRF) [34, 35]. Furthermore, we use the estimated factor matrices to perform the time-delay estimation technique described in Subsection 2.3.6 Then, this solution uses the pseudo-inverse of $((\hat{\mathbf{C}}\mathbf{Q}_{\omega})^{\mathrm{T}} \diamond \mathbf{A})$ to estimate $\hat{\mathbf{\Gamma}}$.

### 2.3.5 LOS Selection

Subsequently to estimating all the parameters of the received signal, we need to separate the LOS and NLOS signal parameters. In this Subsection we describe the third element of the framework. This element performs the LOS selection utilizing the estimated factor matrices. Therefore, as described in [15], to perform the LOS selection the estimated factor To perform the time-delay estimation the resulting estimated factor matrices $(\hat{\mathbf{C}}\mathbf{Q}_{\omega}^{(d)}))^{(t)^{\mathrm{T}}}$, $\hat{\mathbf{A}}^{(t)}$, and $\hat{\mathbf{\Gamma}}^{(t)}$ are normalized to unit norm for the $\ell_d^{(t)}$th component

$$(\hat{\hat{\mathbf{C}}}\mathbf{Q}_{\omega}^{(d)})^{(t)^{\mathrm{T}}}_{\cdot,\ell_d^{(t)}} = (\hat{\mathbf{C}}\mathbf{Q}_{\omega}^{(d)})^{(t)^{\mathrm{T}}}_{\cdot,\ell_d^{(t)}} / ||(\hat{\mathbf{C}}\mathbf{Q}_{\omega}^{(d)})^{(t)^{\mathrm{T}}}_{\cdot,\ell_d^{(t)}}||_F \tag{2.56}$$

$$\hat{\hat{\mathbf{A}}}^{(t)}_{\cdot,\ell_d^{(t)}} = \hat{\mathbf{A}}^{(t)}_{\cdot,\ell_d^{(t)}} / ||\hat{\mathbf{A}}^{(t)}_{\cdot,\ell_d^{(t)}}||_F \tag{2.57}$$

$$\hat{\hat{\mathbf{\Gamma}}}^{(t)}_{\cdot,\ell_d^{(t)}} = \hat{\mathbf{\Gamma}}^{(t)}_{\cdot,\ell_d^{(t)}} / ||\hat{\mathbf{\Gamma}}^{(t)}_{\cdot,\ell_d^{(t)}}||_F. \tag{2.58}$$

Next, with the normalized factor matrices, we construct the tensor $\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}}$ for the $\ell_d^{(t)}$th normalized component of the estimated factor matrices. using (2.56), (2.57) and (2.58) and we obtain

$$\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}} = \hat{\hat{\mathbf{\Gamma}}}^{(t)}_{\cdot,\ell_d^{(t)}} \circ (\hat{\hat{\mathbf{C}}}\mathbf{Q}_{\omega}^{(d)})^{(t)^{\mathrm{T}}}_{\cdot,\ell_d^{(t)}} \circ \hat{\hat{\mathbf{A}}}^{(t)}_{\cdot,\ell_d^{(t)}}, \tag{2.59}$$

where $\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}} \in \mathbb{C}^{K \times Q \times M}$. Then, we store the tensor $\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}}$ corresponding to the $\ell_d^{(t)}$th component in a matrix

$$\mathbf{G}^{(t)}_{\cdot,\ell_d^{(t)}} = \mathrm{vec}\{\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}}\}, \tag{2.60}$$

where $\mathbf{G}^{(t)} \in \mathbb{C}^{KQM \times L_d}$, and $\mathrm{vec}\{\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}}\}$ vectorize the tensor $\boldsymbol{\mathcal{G}}^{(t)}_{\ell_d^{(t)}}$. Thus, we can compute the tensor amplitudes by multiplying the pseudoinverse of $\mathbf{G}^{(t)}$ by the $\mathrm{vec}\{\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}\}$ and we obtain

$$\gamma^{(t)} = \mathbf{G}^{(t)^{+}} \, \mathrm{vec}\{\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}\} \tag{2.61}$$

Assuming that the received signal component with the largest power corresponds to the LOS signal, we select the respective column of the estimated $(\hat{\hat{\mathbf{C}}}\mathbf{Q}_{\omega}^{(d)})^{(t)^{\mathrm{T}}}$ with

$$\hat{\ell_d}^{(t)} = \arg \max_{\ell_d^{(t)}=1,\ldots,L_d^{(t)}} |\gamma^{(t)}_{\ell_d^{(t)},\cdot}|^2. \tag{2.62}$$

### 2.3.6 Time-Delay Estimation

Since we have selected the LOS component, in this Subsection we describe the fourth element of the framework by describing the Time-Delay Estimation process. Therefore, we use the $s_{\text{LOS}}$ from the third element to select the LOS component from the estimated $(\hat{\bar{\mathbf{C}}}\mathbf{Q}_{\omega})^{(t)^{\text{T}}}$ and multiply it by $\mathbf{\Sigma}\mathbf{V}^{\text{H}}$ from the thin SVD of $\mathbf{Q}_d$

$$\mathbf{q} = \left[(\hat{\bar{\mathbf{C}}}\mathbf{Q}_{\omega})^{(t)^{\text{T}}}_{\cdot,s_{\text{LOS}}}\mathbf{\Sigma}\mathbf{V}^{\text{H}}\right]^{\text{T}}. \tag{2.63}$$

where $\mathbf{q}$ contains the cross-correlation values at each tap of the correlator bank. The use of the correlator bank $\mathbf{Q}_{\omega}^{(d)}$ does preserve the noise statistics but, the output with respect to the signal, changes compared to the correlator bank $\mathbf{Q}_d$ which produces a sampled cross-correlation function with the received LOS signal. Thus, vector $\mathbf{q}$ is computed by spatial filtering (along $M$-dimension) of each tap in the compressed time domain preceded by temporal filtering over the taken snapshots (along $K$-dimension). Then, as shown in [4, 8, 9, 39, 40, 15], the resulting vector $\mathbf{q}$ is interpolated using a simple cubic spline interpolation. Therefore, by using the resulting interpolated vector, we can derive the cost function $F(\kappa)$ which is the cross-correlation function with the received LOS signal. Finally, we use the cost function to estimate the time-delay of the LOS signal by solving

$$\hat{\tau}_{\text{LOS}}^{(t)} = \arg_{\kappa} \max F(\kappa). \tag{2.64}$$

## 2.4 Simulation Results for Time-Delay Estimation Simulations for Static Scenarios

In this section, we present simulation results of the various methods for GPS3 L1C signal. We assess the cases $L_d = 1$, $L_d = 2$and $L_d = 3$. Additionally, preceding the time-delay estimation, we apply the state-of-the-art AIC, MDL, EFT, M-EFT, R-D AIC, R-D MDL, R-D EFT, RADOI, and ESTER MOS methods. Moreover, we present simulations comparing the state-of-the-art MOS methods to the case with a known model order. Moreover, since [15] shows that the simulations potentially have outliers when signals are strongly correlated, we, therefore, performed 1000 iteration Monte Carlo (MC) simulation to compare all approaches in terms of the Root Median-Squared Error (RMDSE) of the time-delay estimation of the LOS components considering the state-of-the-art CPD-GEVD [9] based approaches, the ideal case; i.e. filtering assuming known $\tilde{\mathbf{\Gamma}}$, and $\mathbf{A}$, and known model order. We utilize the median because this method is more precise to provide a good measure of central tendency in the presence of outliers than the mean.

In Figure 2.9 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 1$ and $L_d = 2$. Note that even when AIC fails to properly estimate the model order, the CPD-GEVD is capable to perform time-delay estimation. Since the NLOS components power is lower than the actual LOS signal, the CPD-GEVD can properly estimate the time-delay even in scenarios with the wrong model order.

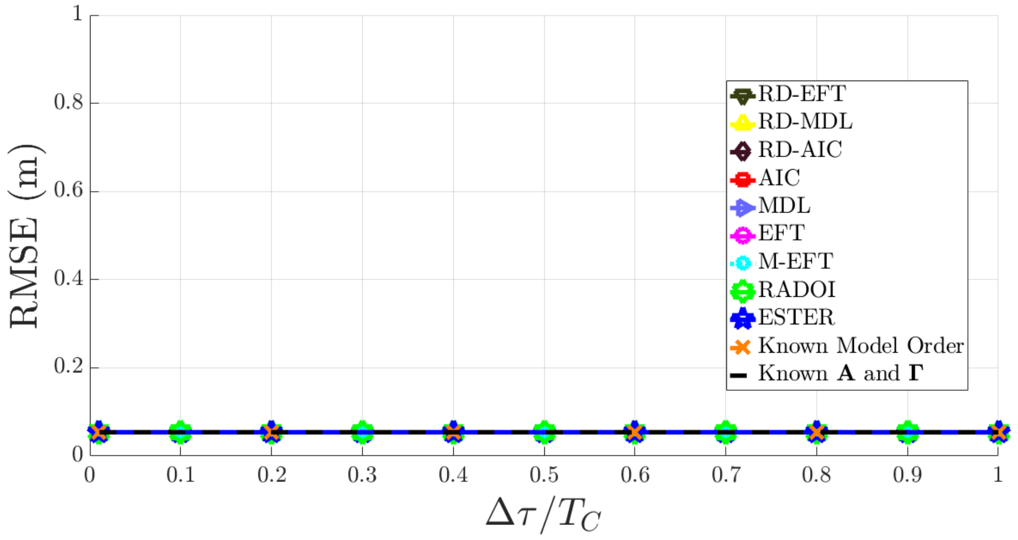


Figure 2.9: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 1$ and $L_d = 2$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 2.10 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 1$ and $L_d = 3$.

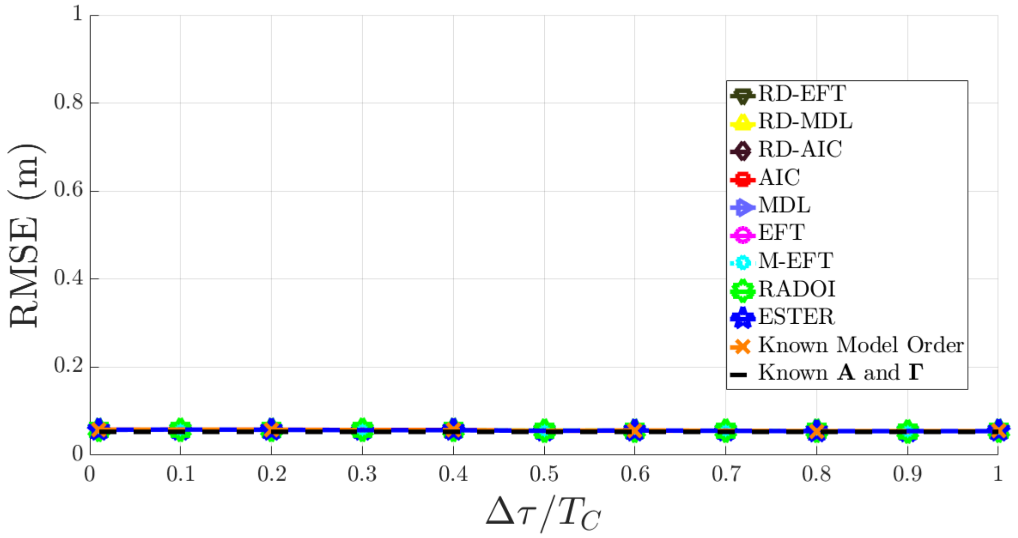


Figure 2.10: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 1$ and $L_d = 3$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 2.11 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 2$ and $L_d = 2$. Note that the CPD-GEVD can successfully computes the time-delay estimation even when the MOS methods fail to estimate the model order. Since the NLOS components power is lower than the actual LOS signal, the CPD-GEVD can properly estimate the time-delay even in scenarios with the wrong model order.

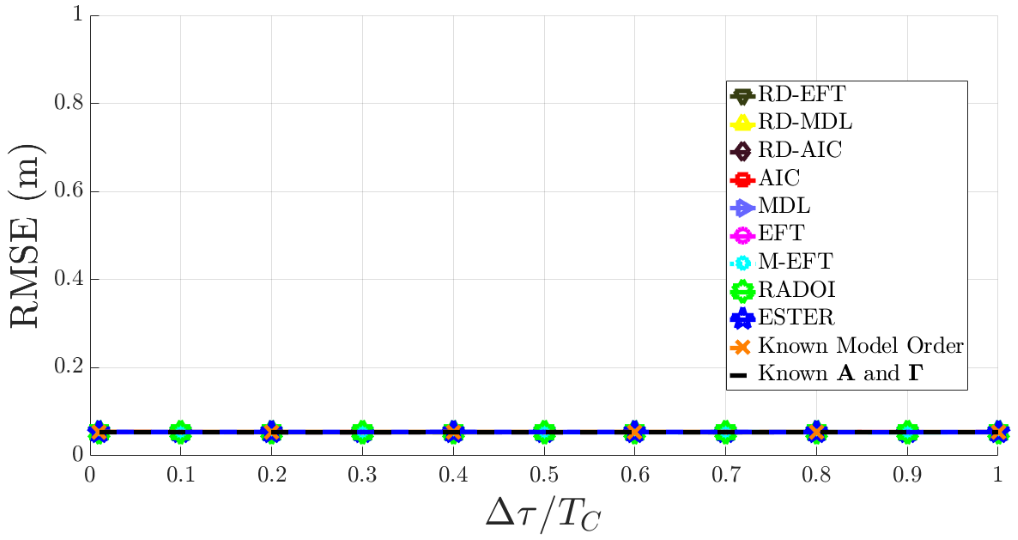


Figure 2.11: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 2$ and $L_d = 2$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 2.12 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 2$ and $L_d = 3$.

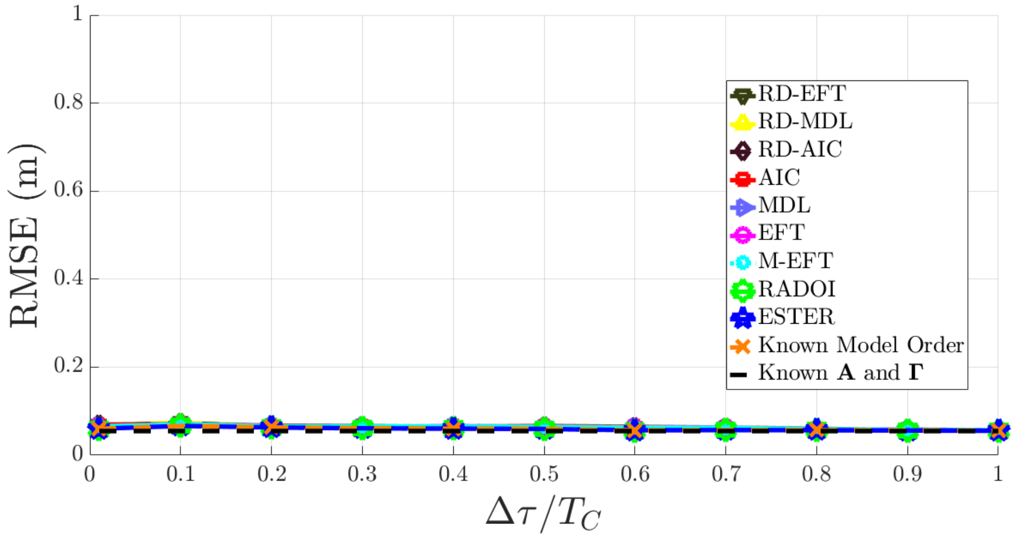


Figure 2.12: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 2$ and $L_d = 3$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 2.13 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 3$ and $L_d = 2$.

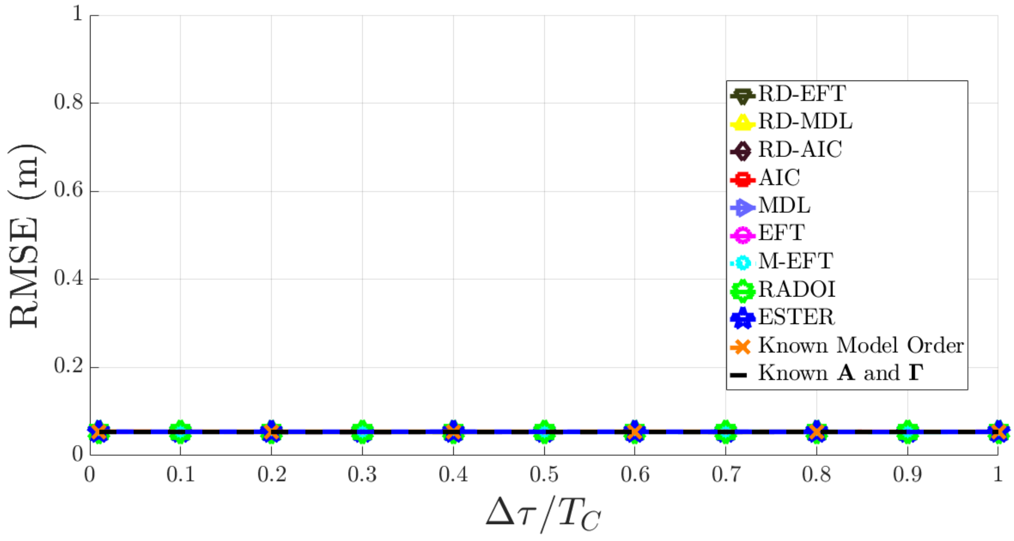


Figure 2.13: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 3$ and $L_d = 2$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 2.14 the MOS methods are compared using state-of-the-art CPD-GEVD method for $d = 3$ and $L_d = 3$.

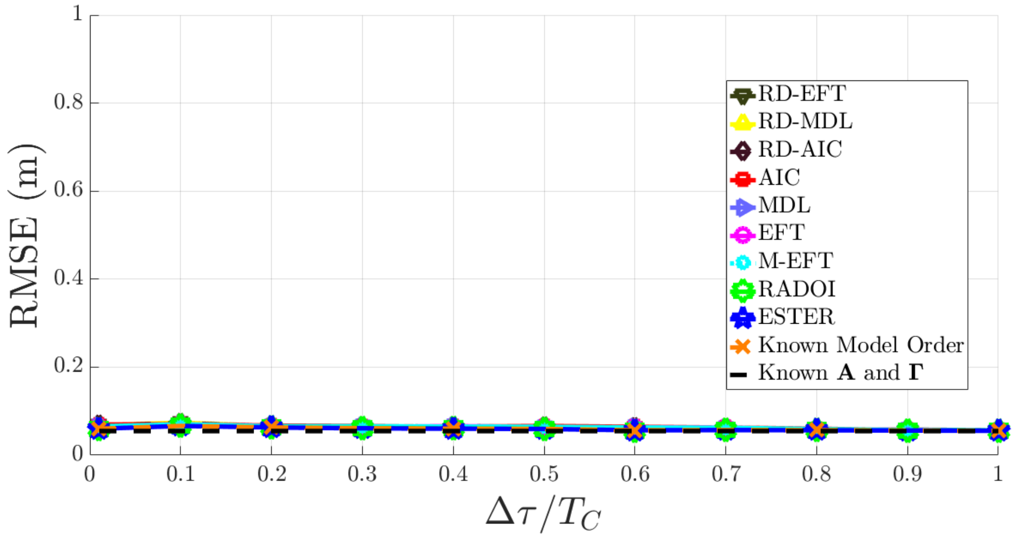


Figure 2.14: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas and $d = 3$ and $L_d = 3$ impinging signal. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Furthermore, even though the EFT-based, AIC-based, MDL-based MOS methods failed to perform model order estimation, the CPD-GEVD successfully performed the time-delay estimation. The model order error for the EFT-based, AIC-based, MDL-based MOS methods varied between $\hat{L}_d - 2$ and $\hat{L}_d + 2$. Therefore, we show that even though the MOS is crucial on factor matrices estimation, the CPD-GEVD tolerates small errors and successfully decomposes the factor matrices.

## 2.5 Computational Complexity

In this section, the state-of-the-art HOSVD+FBA+ESPS, DoA/KRF, and CPD-GEVD as well as the proposed HOSVD SECSI method computational complexity are discussed.

The computational complexity is computed in term of FLoating point OPeration (flop) counts. For instance, the computational complexity of a matrix multiplication between two complex matrices, $\mathbf{A} \in \mathbb{C}^{M \times N}$ and $\mathbf{B} \in \mathbb{C}^{N \times L}$ is denoted as $\mathcal{O}(AB) = 2MNL$ [41]. Since unfolding, inverse-unfolding are functions about data representation, we are not considering this operations in the computational complexity. In Subsection 2.5.1 we show the computational complexity of the state-of-the-art HOSVD+FBA+ESPS. Subsection 2.5.2 show the computational complexity of the state-of-the-art DoA/KRF method. Subsection 2.5.3 show the computational complexity of the state-of-the-art CPD-GEVD method. In Subsection 2.5.4 we show the computational complexity of the proposed HOSVD SECSI.

### 2.5.1 Complexity of HOSVD+FBA+ESPS

Since we want to compute $\mathbf{q}_{\text{ESPS}} \in \mathbb{C}^{Q \times 1}$, firstly we compute the FBA of the received tensor $\boldsymbol{\mathcal{Y}}$. The FBA pre-processing step results in the following complexity:

$$\mathcal{O}(\text{FBA}) = 2M^2KQ + 2(KQ)^2M, \tag{2.65}$$

then, after computing the FBA we compute the ESPS, which yields the following complexity:

$$\mathcal{O}(\text{ESPS}) = \Big[2M(2KQ)\Big]L_s, \tag{2.66}$$

afterwards we use the resulting tensor $\boldsymbol{\mathcal{Z}}_{\text{ESPS}}$ to perform the HOSVD rank-one operation. Therefore, the HOSVD complexity is given by:

$$\begin{aligned} \mathcal{O}(\text{HOSVD}) = &\Big[4K^3 + I_K(8K^2 + 10K)\Big] + \Big[4Q^3 + I_Q(8Q^2 + 10Q)\Big] \\ &+ \Big[4L_S^3 + I_{L_S}(8L_S^2 + 10L_S)\Big] + \Big[4M_S^3 + I_{M_S}(8M_S^2 + 10M_S)\Big], \end{aligned} \tag{2.67}$$

where $I_K$ is the number of SVD power operations performed when using the first-mode unfolding of tensor $\boldsymbol{\mathcal{Z}}_{\text{ESPS}}$, $I_Q$ is the number of SVD power operations performed when using the second-mode unfolding, $I_{M_S}$ is the number of SVD power operations performed when using the third-mode unfolding, $I_{L_S}$ is the number of SVD power operations performed when using the fourth-mode unfolding.

Then, in order to compute the vector $\mathbf{q}_{\text{ESPS}}$, we have the following complexity:

$$\mathcal{O}(\mathbf{q}_{\text{ESPS}}) = 4KQL_SM_S + 4L_SKQM_S + 4L_SKQM_S + 4L_SKQM_S. \tag{2.68}$$

Moreover, we can define the total computational complexity of the state-of-the-art HOSVD+FBA+ESPS methods as:

$$\mathcal{O}(\text{HOSVD+FBA+ESPS} = \mathcal{O}(\text{FBA}) + \mathcal{O}(\text{ESPS}) + \mathcal{O}(\text{HOSVD}) + \mathcal{O}(\mathbf{q}_{\text{ESPS}}). \tag{2.69}$$

### 2.5.2 Complexity of DoA/KRF

Additionally, we compute the computational complexity of the state-of-the-art DoA/KRF. Therefore, once the first step performs the ESPRIT operation, we define the ESPRIT computational complexity as follows:

$$\begin{aligned}\mathcal{O}(\text{ESPRIT}) = 2KQM^2 + 2KQ^2M + I(4KQ^2 + 5KQ + 4M^2 + 5M) \\ + 2\Big[2(M-1)L_d^2\Big] + \frac{5}{3}(M-1)^3 + L_d^3\end{aligned}, \tag{2.70}$$

where $\mathbf{I}$ is the number of SVD power operations. Afterwards, the Khatri-Rao Factorization computational complexity can be described as:

$$\mathcal{O}(\text{KRF}) = 2L_dKQ^2 + 2L_dK^2Q + L_dI(4K^2 + 5K + 4Q^2 + 5Q). \tag{2.71}$$

Then, the DoA/KRF method normalizes the estimated factor matrices, and computes the amplitudes. Therefore, the normalization and amplitude estimation has a computational complexity:

$$\mathcal{O}(\text{NORM+AMP}) = L_d(KQ + KQM) + 2L_d^2KQM + \frac{5}{3}L_d^3 + 2L_d^2KQM. \tag{2.72}$$

Furthermore, we can define the total computational complexity by summing the computed complexities with other minor operations used in the the DoA/KRF method:

$$\begin{aligned}\mathcal{O}(\text{DoA/KRF}) = \mathcal{O}(\text{ESPRIT}) + \mathcal{O}(\text{KRF}) + \mathcal{O}(\text{NORM+AMP}) \\ + 2L_d^2M + \frac{5}{3}L_d^3 + 2L_d^2M + 2L_dMQ + 2Q + L_d^3.\end{aligned} \tag{2.73}$$

### 2.5.3 Complexity of CPD-GEVD

Similarly to the HOSVD+FBA+ESPS, the state-of-the-art CPD-GEVD show the same computational complexity as (2.67) when performing the HOSVD step. Furthermore, we compute the LSKRF computational complexity as follows:

$$\mathcal{O}(\text{LSKRF}) = 2L_dMQ^2 + 2L_dM^2Q + L_dI(4M^2 + 5M + 4Q^2 + 5Q). \tag{2.74}$$

Then, we can define the final computational complexity by summing the computed complexities with the steps that compute the GEVD, factor matrix $\hat{\mathbf{\Gamma}}$, factor matrix $\mathbf{F}^{(2,3)}$, and least square as follows:

$$\begin{aligned}\mathcal{O}(\text{CPD-GEVD}) = \mathcal{O}(\text{HOSVD}) + \mathcal{O}(\text{LSKRF}) + \mathcal{O}(\text{NORM+AMP})\\ + L_d^3 + 2KL_d^2 + 2QMKL_d + 2L_dKQ + 4KMQ.\end{aligned} \tag{2.75}$$

### 2.5.4 Complexity of HOSVD SECSI

Since we decided to use only the right-hand matrix of the third-mode unfolding to perform time-delay estimation, we only calculated the computational complexity of HOSVD SECSI for the first factor estimate. Similarly to the state-of-the-art CPD-GEVD, the HOSVD low-rank approximation step of proposed HOSVD SECSI shows the same computational complexity as Equation (2.67). Moreover, after HOSVD low-rank approximation we compute the computational complexity of constructing the third-mode slice as:

$$\mathcal{O}(\text{3-MODE}) = ML_d^3. \tag{2.76}$$

Then, we define the complexity of the conditional operation as:

$$\mathcal{O}(\text{COND}) = 4L_d^3 + I(8L_d^2 + 10L_d), \tag{2.77}$$

where $I$ is the number of SVD power operations. Then, we define the computational complexity of computing the right-hand matrix as:

$$\mathcal{O}(\text{RIGHT-HAND}) = NL_d^3, \tag{2.78}$$

where $N$ is the number of slices. Furthermore, we define the joint diagonalization computation complexity as:

$$\mathcal{O}(\text{JOINTDIAG}) = J(4L_d^2 + 16L_d^2 + \frac{10}{3}L_d), \tag{2.79}$$

where $J$ is the number of iterations. Moreover, the LSKRF has the same computational complexity as (2.74). Finally, we compute the total computational complexity by combining the computed complexities with inverse, pseudo-inverse, and factor matrices estimation computational complexities as follows:

$$\begin{aligned}\mathcal{O}(\text{SECSI}) = \mathcal{O}(\text{HOSVD}) + \mathcal{O}(\text{3-MODE}) + \mathcal{O}(\text{LSKRF}) + \mathcal{O}(\text{COND})\\ + \mathcal{O}(\text{RIGHT-HAND}) + \mathcal{O}(\text{JOINTDIAG}) + \mathcal{O}(\text{NORM+AMP})\\ + \frac{10}{3}L_d^3 + 6KL_d^2 + 2L_dKQ\end{aligned} \tag{2.80}$$

In Table 2.2 a numerical example is shown in which we consider the simulation scenarios considered in this work. The dimensions of post-correlation signal $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{K \times Q \times M}$ are $K = 30$ epochs, $Q = 11$ taps of the correlator bank, and $M = 8$ antenna array elements. Performing the ESPS step, we divide the antenna array into $L_S = 5$ subarrays with $M_S = 4$ elements each. Moreover, we consider the scenario with $L_d = 2$ impinging signals, one LOS and one NLOS signal. The computational time, in seconds, of each approach is given in the last column of Table 2.2 for a single MC run and considering a noiseless case. We can observe that the proposed HOSVD SECSI-based approach has the highest complexity and computational time. However, the difference in computational complexity and computational time is rather small while the HOSVD SECSI-based approach provides the best performance of all assessed methods, especially in the case of highly correlated signals which is essential for safety-critical applications using GNSS.

| Method | Complexity | Time |
|---|---|---|
| HOSVD+FBA+ESPS | $2057120 + 7500I_K + 250I_{L_s} + 168I_{M_s} + 1078I_Q$ | 0.0087 s |
| DoA/KRF | $25044I + \frac{611545}{3}$ | 0.0084 s |
| CPD-GEVD | $258346 + 7500I_K + 592I_M + 1078I_Q + 2505I$ | 0.0044 s |
| HOSVD SECSI | $233413 + 7500I_K + 592I_M + 1078I_Q + 2607I + 190J + 27N$ | 0.0092 s |

Table 2.2: Numerical example of the computational complexity

# Chapter 3

# Time-Delay Estimation for Dynamic Scenarios

In this chapter, in Section 3.1 we propose the EFT method as a feasible solution for model order selection in dynamic scenarios. Moreover, in Section 3.2 we describe the proposed variation od the HOSVD SECSI for dynamic scenarios. This approach is based on [24, 25, 26] which uses a Semi-algebraic solution for the Canonical Polyadic (CP) model by performing several joint matrix diagonalizations. Thus, this solution simultaneously estimates the complex amplitude factor matrix. Then, we use the Least Square Khatri-Rao Factorization (LSKRF) to estimate the steering and code factor matrices. Finally in Section in 3.3 we display the TDE simulation results after combining the EFT method with the MuDe method, the proposed factorization method and the state-of-the-art factorization matrices in dynamic scenarios.

## 3.1 Model Order Selection for Dynamic Environments

To decompose the tensor $\boldsymbol{\mathcal{Y}}$ into factor matrices for the time-delay estimation, we estimate the $L_d(k)$ multipath components of the $d$th satellite. Therefore, to perform the Model Order Selection of each epoch of tensor $\boldsymbol{\mathcal{Y}}$, we can utilize the Eigenvalue Fitting Test (EFT) [22]. Moreover, to compute the model order using each epoch of the tensor data model, we utilize the covariance matrix $\mathbf{R}\hat{[k]}$ obtained from each epoch of tensor $\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}$ from (2.9). Thus, we firstly compute the EVD

$$\begin{aligned} \hat{\mathbf{R}}[k] &= \frac{1}{Q}\boldsymbol{\mathcal{Y}}[k]\boldsymbol{\mathcal{Y}}[k]^{\mathrm{H}} && (3.1) \\ &= \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{H}} + \mathbf{R}_{qq}[k], && (3.2) \end{aligned}$$

where $\mathbf{R}[k] \in \mathbb{C}^{M\times M}$ is a Hermitian matrix, $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \dots\ \mathbf{u}_M] \in \mathbb{C}^{M\times M}$ is a unitary matrix containing the eigenvectors, $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_M\} \in \mathbb{C}^{M \times M}$ is a diagonal matrix holding the the sorted eigenvalues $\lambda_i$, such that $\lambda_1 > \lambda_2 > \cdots > \lambda_M$, and the correlator bank covariance matrix $\mathbf{R}_{qq} \in \mathbb{C}^{M \times M}$. Moreover, we define $\mathbf{U}^{(s)} = [\mathbf{u}_1 \; \mathbf{u}_2 \; \ldots \; \mathbf{u}_P] \in \mathbb{C}^{M \times P}$ as the truncated matrix composed of $P$ eigenvectors of $\mathbf{U}$ corresponding to the $P$ largest eigenvalues of $\mathbf{\Lambda}$. Therefore, in case that $P = L_d(k)$, the dominant eigenvectors $\mathbf{U}^{(s)} \in \mathbb{C}^{M \times L_d(k)}$ and column space of the steering matrix $\mathbf{A}$ have the same subspace.

Moreover, the EFT utilize an exponential profile do approximate the Wishart profile of the noise eigenvalues, thus, enabling the prediction of the noise eigenvalues. The EFT estimates the model order by computing the distance from $\lambda_{M-P}$, calculated from the measurements to the predicted eigenvalue $\hat{\lambda}_{M-P}$, where $P$ is a possible number of noise eigenvalues. Furthermore, the EFT method computes the threshold coefficients $\eta_P$, then, estimate the model order. Since $M$, $Q$, and the probability of false alarm $P_{\text{fa}}$ do not vary, the $\eta_P$ can be previously computed and stored.

Then, the EFT can be computed by utilizing the following correction equations [42]

$$\hat{\lambda}_{M-P} = (P-1)\frac{1-q(P+1,Q)}{1-q(P+1,Q)^{P+1}}\hat{\sigma^2} \tag{3.3}$$

$$\hat{\sigma^2} = \frac{1}{P+1}\sum_{i=0}^{P}\lambda_{M-i}, \tag{3.4}$$

where the rate of the exponential profile $q(M,Q)$ is defined as

$$q(M,Q) = \exp\left\{ -\sqrt{\frac{30}{M^2+2} - \sqrt{\frac{900}{(M^2+2)^2} - \frac{720M}{Q(M^4+M^2-2)}}} \right\}. \tag{3.5}$$

Observe that since $\lambda_{M-P}$ from (3.4) is utilized in (3.3) to estimate $\hat{\lambda}_{M-P}$, the (3.3) is a correction equation rather than a prediction equation. To estimate the threshold coefficients $\eta_P$, we consider $Q_{\ell(k)}$ realizations of a white Gaussian noise matrix $\mathbf{N} \in \mathbb{C}^{M \times Q}$. Thus, to obtain the $P_{\text{fa}}$ as a function of $\eta_P$, we presume the following hypotheses:

$$\begin{aligned} H_{P+1} &: \lambda_{M+P} \text{ is a noise EV, } \frac{\lambda_{M-P} - \hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}} \leq \eta_P \\ H_{P+1} &: \lambda_{M+P} \text{ is a signal EV, } \frac{\lambda_{M-P} - \hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}} > \eta_P, \end{aligned} \tag{3.6}$$

where EV stands for eigenvalue. Then, we define the $Q_{\text{fa}}$ as the number of times that $H_{P+1}$ is observed for all $Q_{\ell(k)}$ noise realizations. Thus, $P_{\text{fa}} = \frac{Q_{\text{fa}}}{Q_{\ell(k)}}$, and for each value of the predefined range of $\eta_P$, a certain value of $P_{\text{fa}}$ is calculated. Finally, once we have all $\eta_P$ the model order can be estimated by the following cost function

$$\begin{aligned} &\hat{L_d}(k) = M - \min(\mathcal{P}) \text{ where} \\ &P \in \mathcal{P},\ if\ \frac{\lambda_{M-P} - \hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}} > \eta_P, \end{aligned} \tag{3.7}$$

where $\mathcal{P}$ is the set of all values of $P$, such that the relative distance $\frac{\lambda_{M-P}-\hat{\lambda}_{M-P}}{\hat{\lambda}_{M-P}}$ is greater than $\eta_P$.

Then, once we have the estimated model order $\hat{L_d}(k)$ for each epoch, we group the epochs with the same estimated model order. Moreover, we create a vector $\hat{L_d}(k)^{(t)}$ that contains the grouped model order with respect to the epochs, with $t = 1, \ldots, T$ sub-tensors. Therefore, we create $\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}$ sub-tensors utilized to perform the time-delay estimation. For instance, if we estimate three different model orders $\hat{L_{d_1}}(k) = 2$ and $\hat{L_{d_2}}(k) = 3$ we concatenate the epochs with the same model order to create a new tensor, as illustrated in Figure 3.1(a) and 3.1(b).

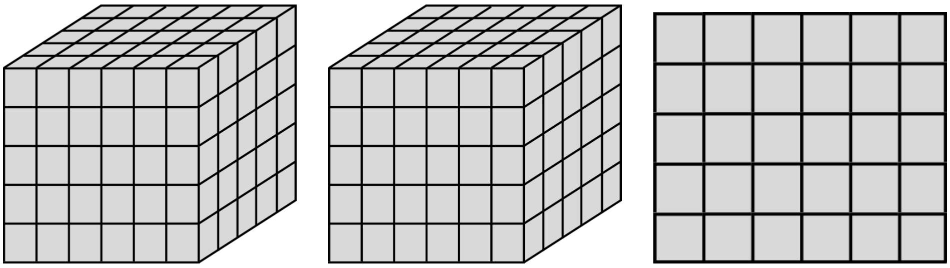

(a) $K = 5, \hat{L_{d_1}}(k) = 2$ (b) $K = 4, \hat{L_{d_2}}(k) = 3$ (c) $K = 1, \hat{L_{d_3}}(k) = 4$

Figure 3.1: Sub-tensors obtained after grouping the epochs with same estimated model order

Therefore, by creating sub-tensors, we attempt to create static scenarios. Thus, we can utilize the sub-tensors to perform the time-delay estimation method, as illustrate in Figure 3.2

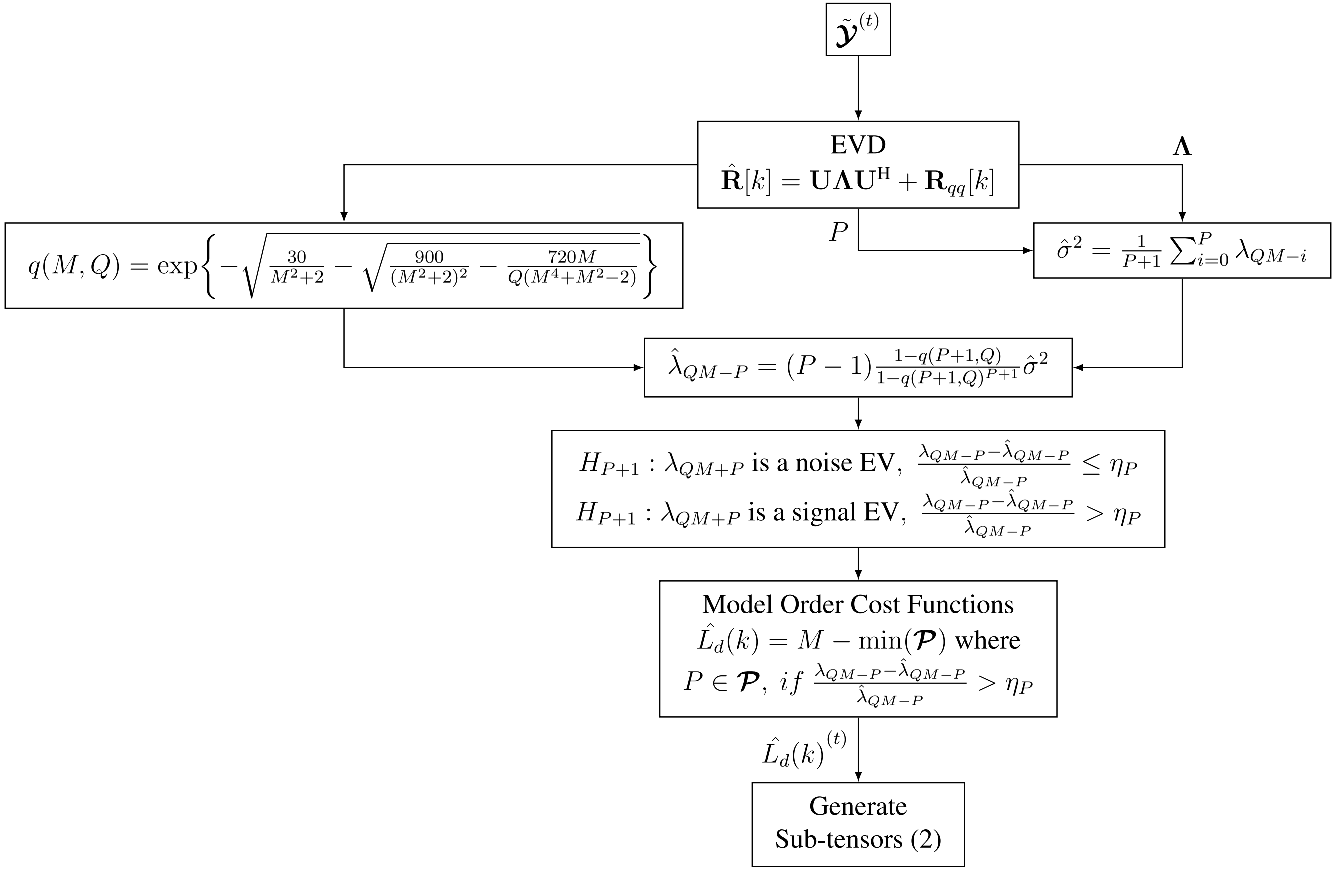


Figure 3.2: Block diagram for the EFT model order selection method.

### 3.1.1 Simulation Results for Probability of Detection Considering Dynamic Scenarios

In this section we present the PoD computed during simulations when considering a dynamic scenario, a perfectly aligned array of antennas, and utilizing one satellites with $PRS = 17$. Since in the dynamic scenario we are varying the number of LOS and NLOS components within the tensor $\boldsymbol{\mathcal{Y}}$, when utilizing matrix-based MOS methods we performed the MOS at each epoch and individually computed the PoD. Therefore, we define a DoA difference between epochs of 2° and established that half of the tensor has $L_d = 2$ while the other half has $L_d = 3$. Moreover, we compute the PoD by taking the maximum and minimum model order estimated by the MOS methods. To compute the maximum and minimum model order we utilize the $\max \cdot$ and $\min \cdot$ operators to select the maximum and minimum estimated model order. Additionally, we utilized tensor-based MOS methods to compute the tensor model order then compared the estimated model order to the maximum and minimum model order, i.e. the minimum model order in $\boldsymbol{\mathcal{Y}}$ is $L_d = 2$ while the maximum model order is $L_d = 3$. Hence, we use the maximum and minimum model order to compute the PoD for the tensor-based MOS methods. Moreover, in the dynamic scenario we utilize the pre-processing methods FBA, SPS, and their combinations to attempt to improve the MOS accuracy.

In Figure 3.3 we show the PoD for the AIC method at $\Delta\tau = 0.1$ and $K = 8$. Note that the PoD is computed for each epoch, the epochs 1 through 5 have an $L_d = 2$, and epochs 6 through 10 have and $L_d = 3$. Observe that when applying the pre-processing methods we improved the AIC accuracy. Overall, the AIC method combined with FBA shows the best results with and PoD of about $50\%$.

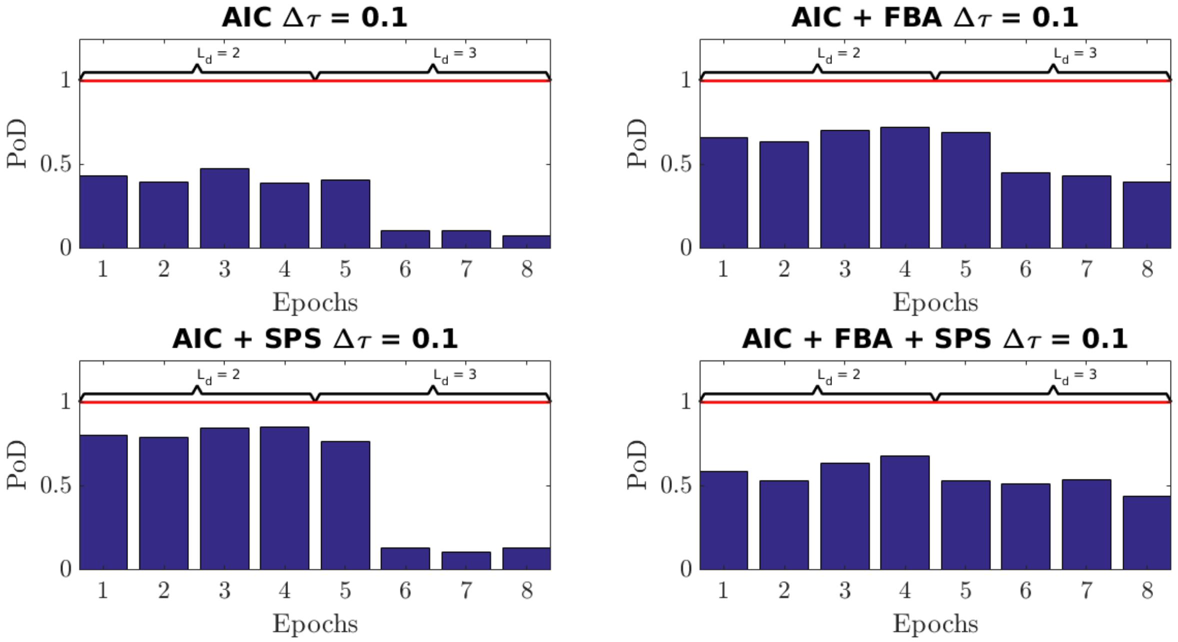


Figure 3.3: Probability of Detection at $\Delta\tau = 0.1$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.4 we show the PoD for the AIC method at $\Delta\tau = 0.1$ and $K = 15$. Again, the PoD is computed for each epoch, the epochs 1 through 7 have an $L_d = 2$, and epochs 6 through 10 have and $L_d = 3$. Observe that when applying the pre-processing methods we improved the AIC accuracy. Overall, the AIC method combined with FBA shows the best results with and PoD of about 50%.

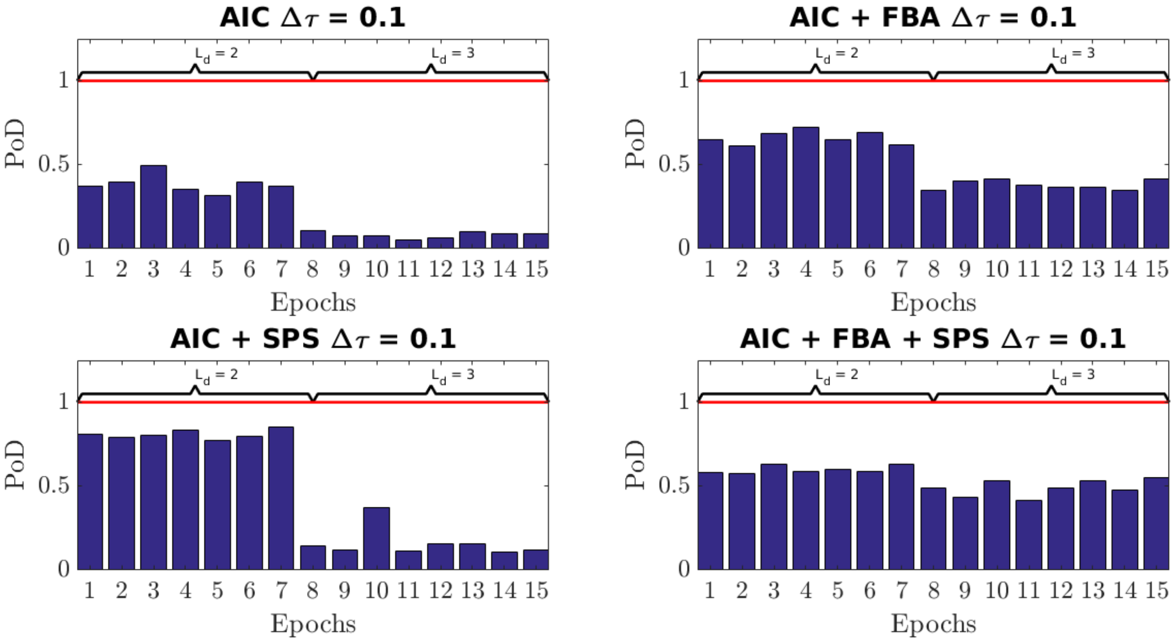


Figure 3.4: Probability of Detection at $\Delta\tau = 0.1$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.5 we show the PoD for the AIC method at $\Delta\tau = 0.1$ and $K = 30$. Again, the PoD is computed for each epoch, the epochs 1 through 7 have an $L_d = 2$, and epochs 6 through 10 have and $L_d = 3$. Observe that when applying the pre-processing methods we improved the AIC accuracy. Overall, the AIC method combined with FBA shows the best results with and PoD of about 50%. Therefore, we see that since we are performing MOS to each slice of the tensor $\boldsymbol{\mathcal{Y}}$, we do not benefit from the increase in the amount of epochs.

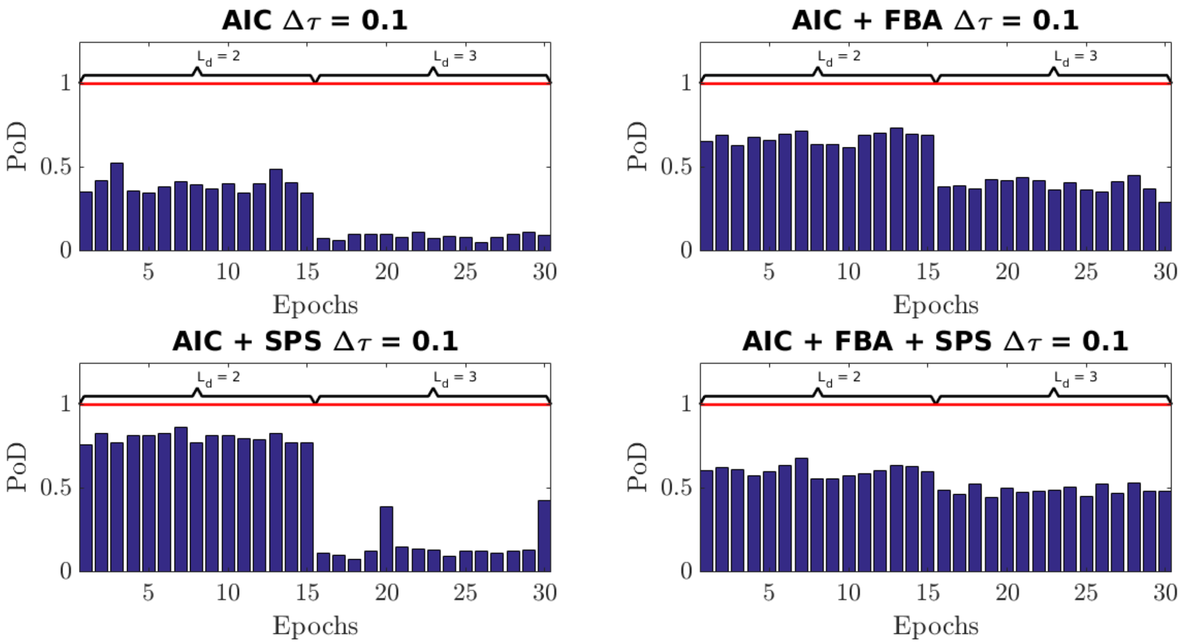


Figure 3.5: Probability of Detection at $\Delta\tau = 0.1$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.6 we show the PoD for the AIC method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 8$. Note that the AIC method shows a better performance when signal components are less correlated. Moreover, observe that the AIC combined with SPS presents the best performance throughout all epochs.

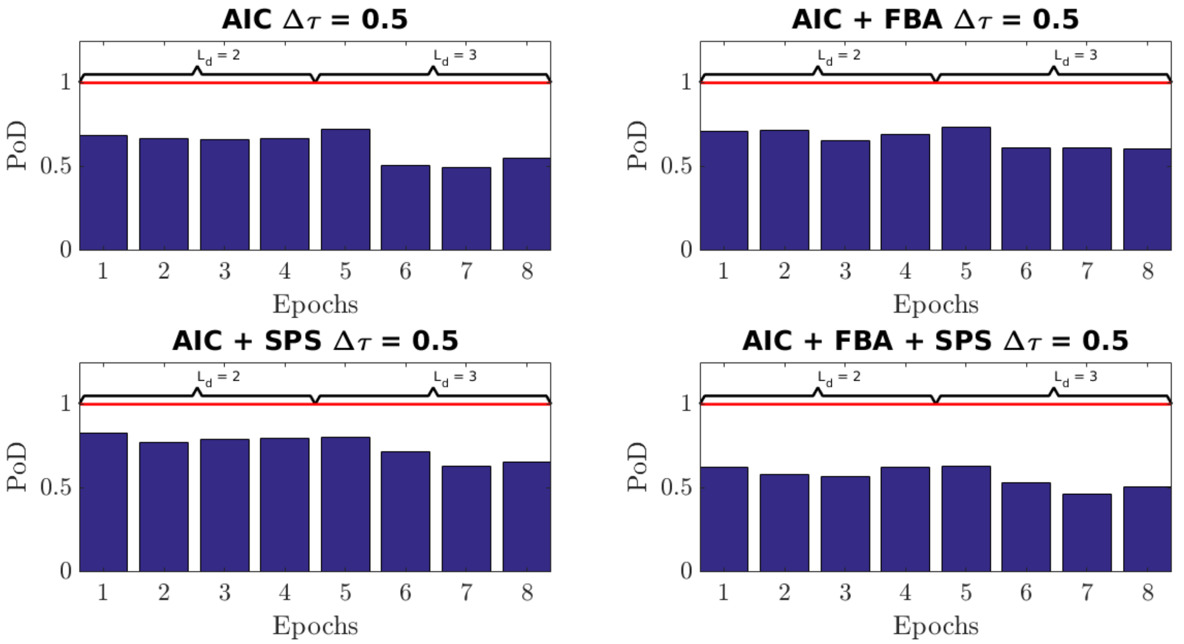


Figure 3.6: Probability of Detection at $\Delta\tau = 0.5$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.7 we show the PoD for the AIC method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 15$. Note that the AIC method shows a better performance when signal components are less correlated. Moreover, note that the AIC combined with SPS keeps the best performance throughout all epochs.

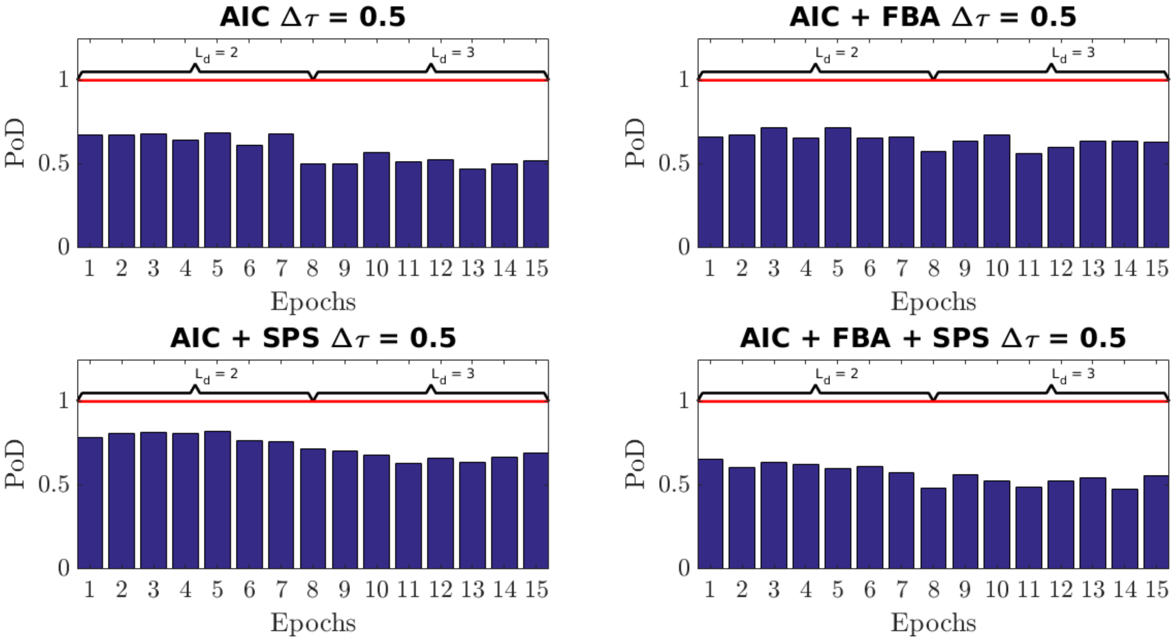


Figure 3.7: Probability of Detection at $\Delta\tau = 0.5$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.8 we show the PoD for the AIC method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 30$. Note that the AIC method shows a better performance when signal components are less correlated. Moreover, note that the AIC does not take any advantage from the higher amount of epochs. Therefore, the AIC combined with SPS keeps the best performance throughout all epochs.

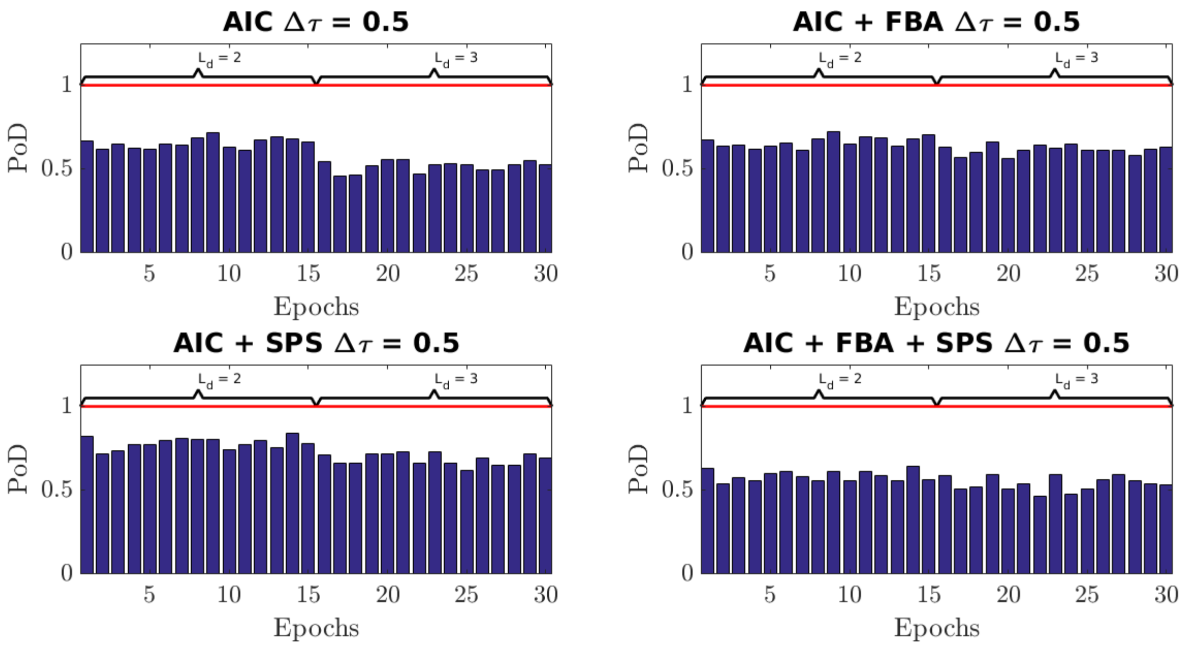


Figure 3.8: Probability of Detection at $\Delta\tau = 0.5$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.9 we show the PoD for the AIC+FBA method at $\Delta\tau = 0.1$ compared to the minimum and maximum model order defined for each epoch. Note that the AIC combined with FBA tends to understimate the model order since that we have a PoD above 90% when comparing the estimated model order with the minimum model order within the received tensor $\boldsymbol{\mathcal{Y}}$.

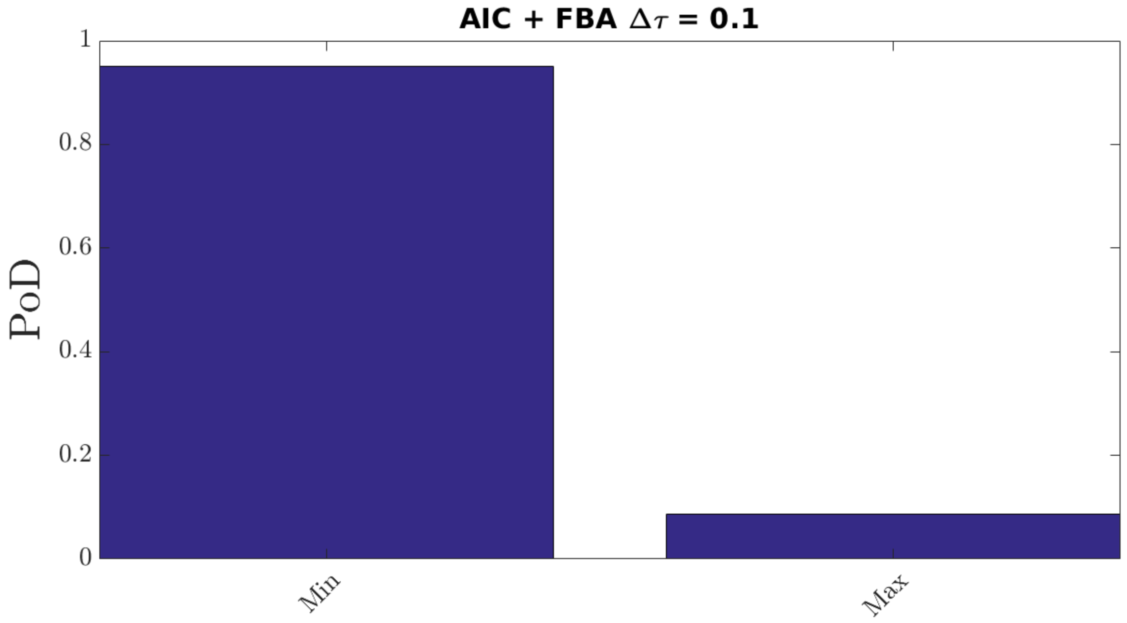


Figure 3.9: Probability of Detection at $\Delta\tau = 0.1$ for AIC method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.10 we show the PoD for the ESTER method at $\Delta\tau = 0.1$ and $K = 8$. Observe that the ESTER method fails to estimate the model order when signals are strongly correlated. However, we can improve ESTER performance by combining it with the pre-processing techniques. Note that although the ESTER+FBA+SPS shows the best results, it has a poor performance when $L_d = 3$.

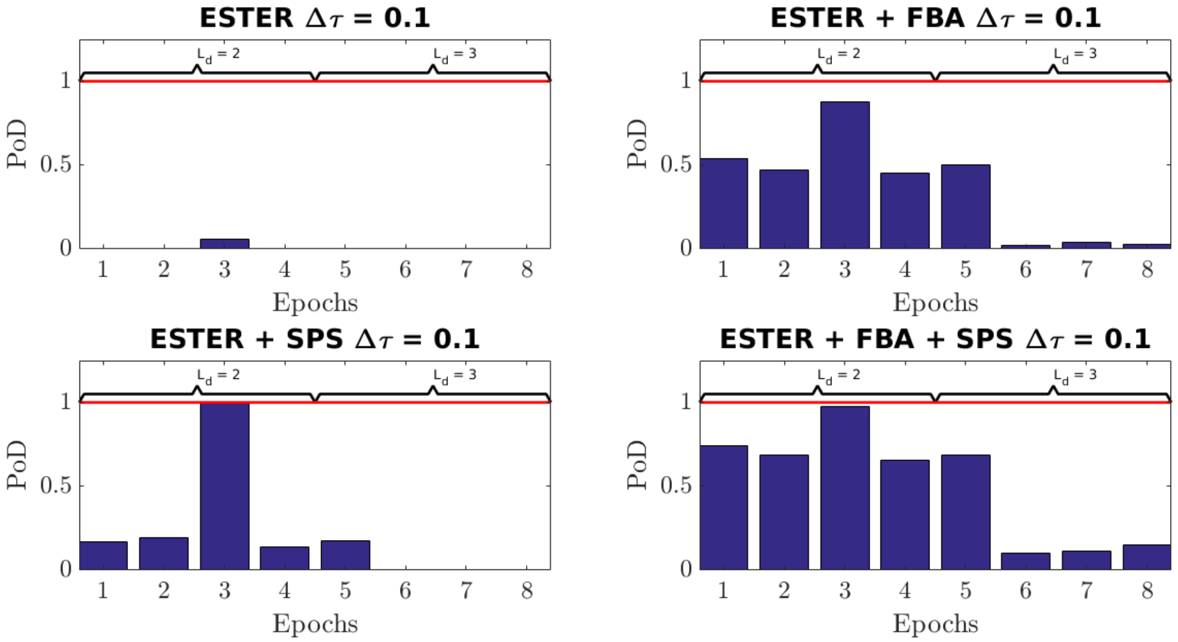


Figure 3.10: Probability of Detection at $\Delta\tau = 0.1$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.11 we show the PoD for the ESTER method at $\Delta\tau = 0.1$ and $K = 15$. Observe that the ESTER method fails to estimate the model order when signals are strongly correlated. Moreover, note that the ESTER methods does not take any advantage of the higher amount of epochs.

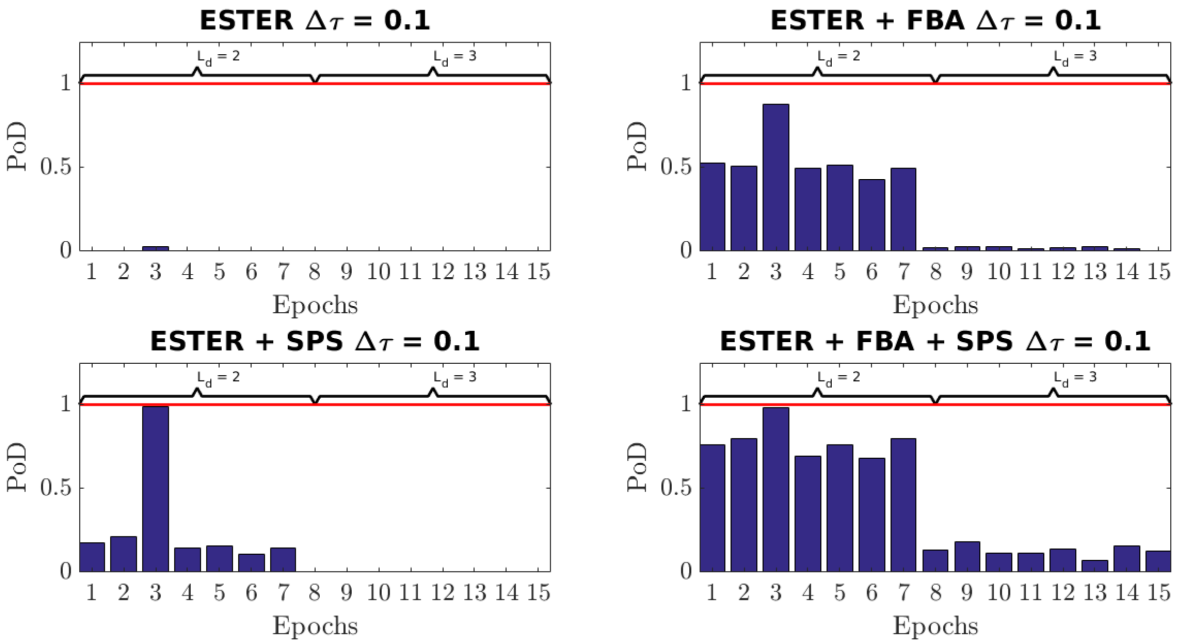


Figure 3.11: Probability of Detection at $\Delta\tau = 0.1$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.12 we show the PoD for the ESTER method at $\Delta\tau = 0.1$ and $K = 30$. Observe that the ESTER method fails to estimate the model order when signals are strongly correlated. Moreover, note that the ESTER methods does not take any advantage of the higher amount of epochs.

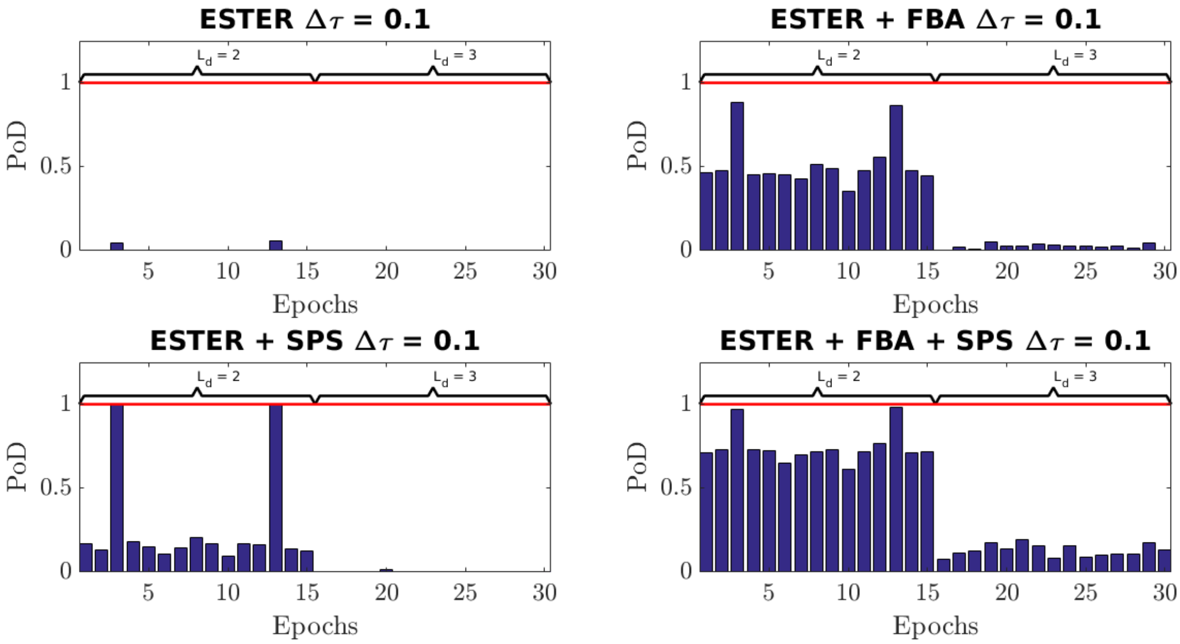


Figure 3.12: Probability of Detection at $\Delta\tau = 0.1$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.13 we show the PoD for the ESTER method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 8$. Note that the ESTER method shows a better performance when signal components are less correlated. Again, observe that the ESTER combined with FBA and SPS presents the best performance however it has a poor performance when $L_d = 3$.

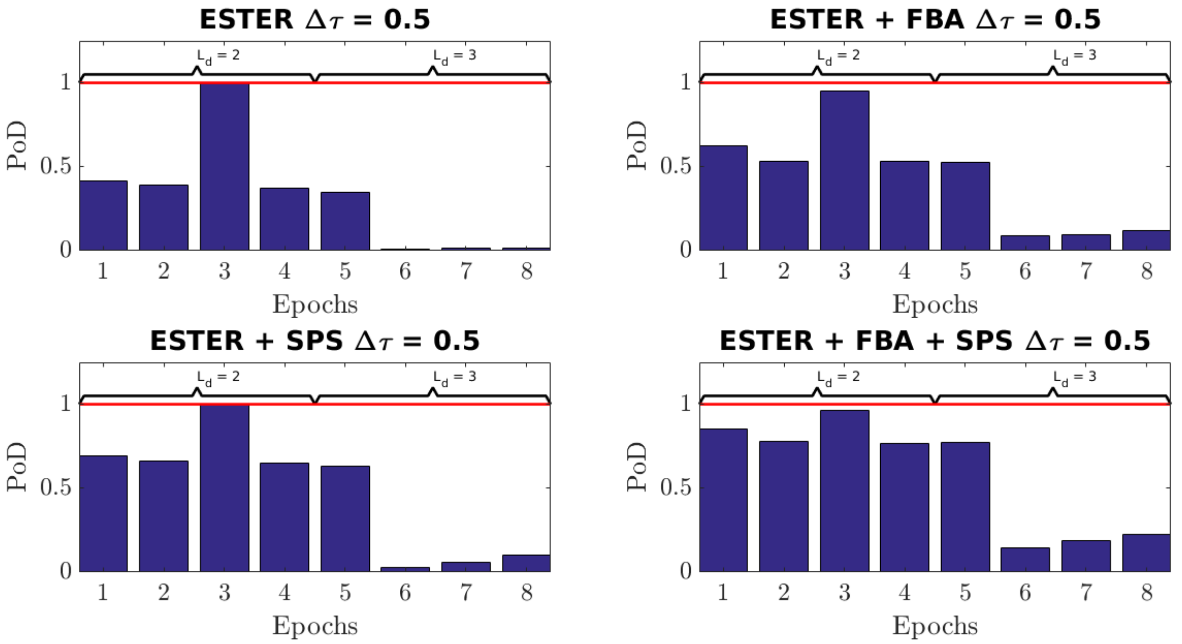


Figure 3.13: Probability of Detection at $\Delta\tau = 0.5$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.14 we show the PoD for the ESTER method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 15$. Again, observe that the ESTER combined with FBA and SPS presents the best performance however it has a poor performance when $L_d = 3$.

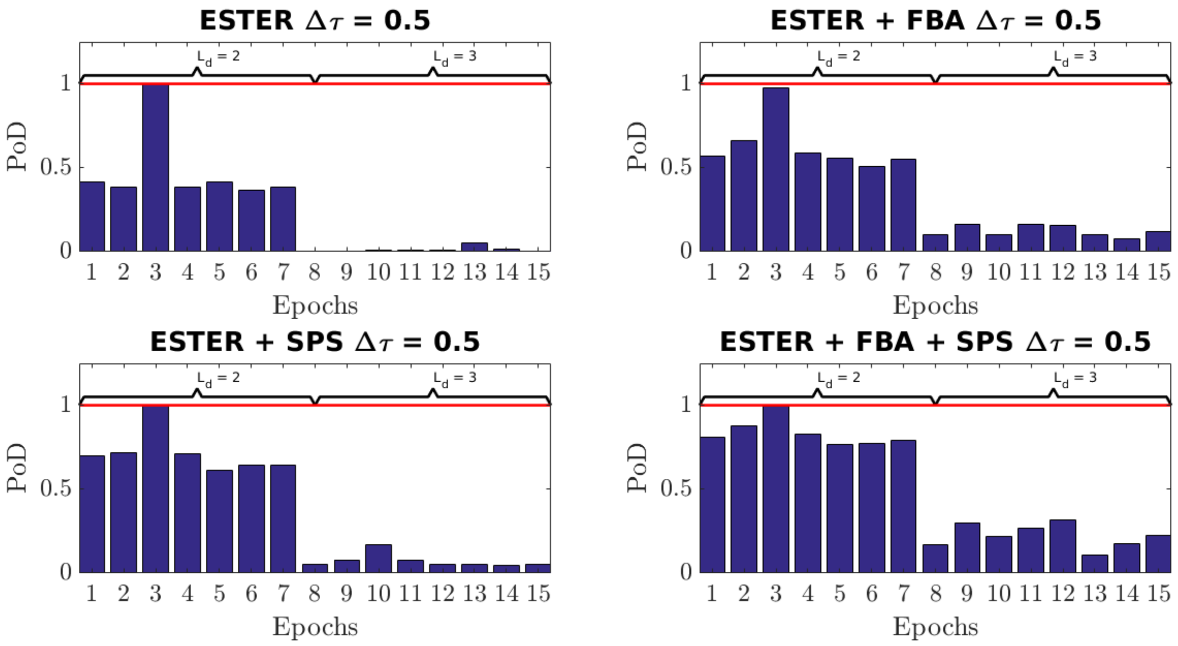


Figure 3.14: Probability of Detection at $\Delta\tau = 0.5$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.15 we show the PoD for the ESTER method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 30$. Once more, observe that the ESTER combined with FBA and SPS presents the best performance however it has a poor performance when $L_d = 3$.

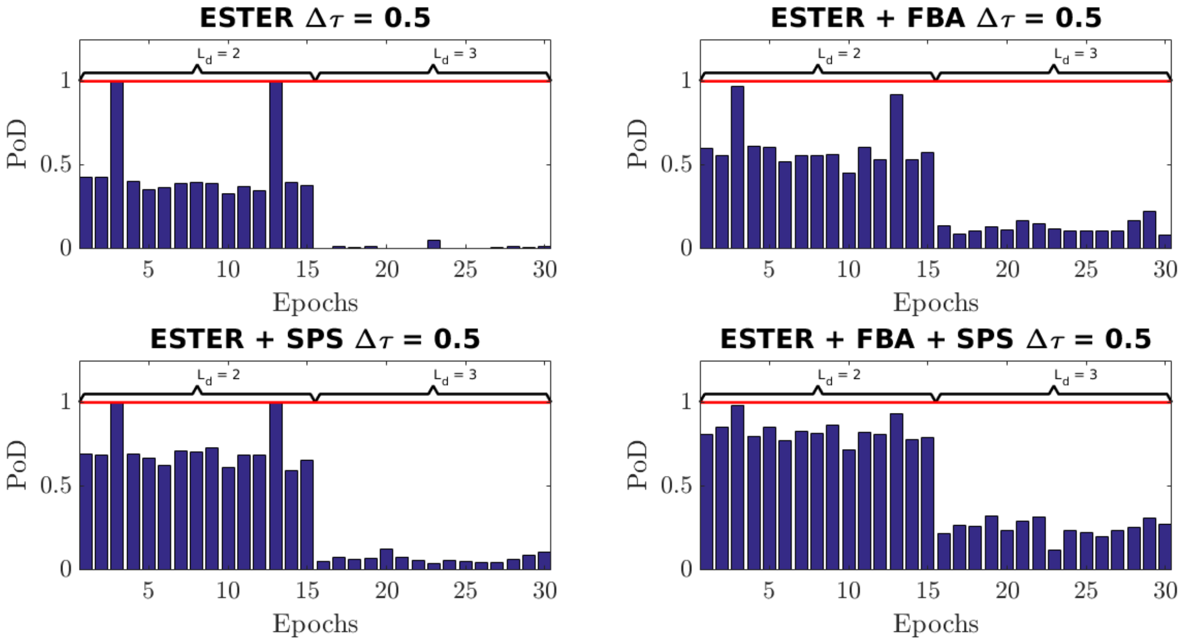


Figure 3.15: Probability of Detection at $\Delta\tau = 0.5$ for ESTER method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.16 we show the PoD for the ESTER+FBA+SPS method at $\Delta\tau = 0.1$ compared to the minimum and maximum model order defined for each epoch. Note that the ESTER combined with FBA and SPS tends to overestimate the model order since that we have a PoD above 40% when comparing the estimated model order to the maximum model order within the received tensor $\boldsymbol{\mathcal{Y}}$.

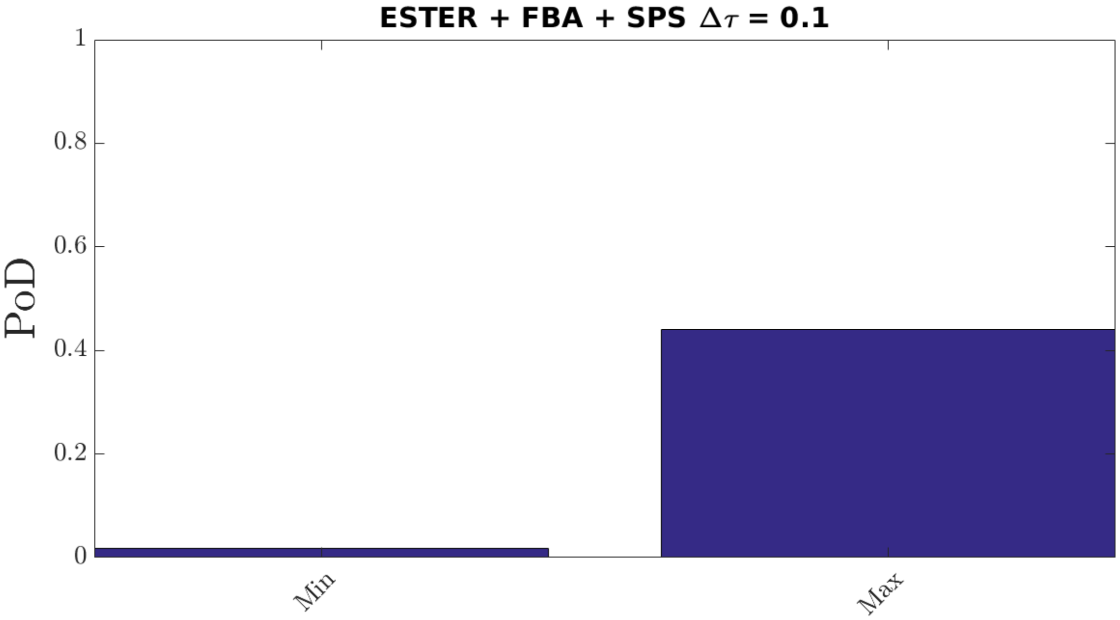


Figure 3.16: Probability of Detection at $\Delta\tau = 0.1$ for ESTER+FBA+SPS method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.17 we show the PoD for the RADOI method at $\Delta\tau = 0.1$ and $K = 8$. Note that the RADOI method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques RADOI shows a PoD above 95%.

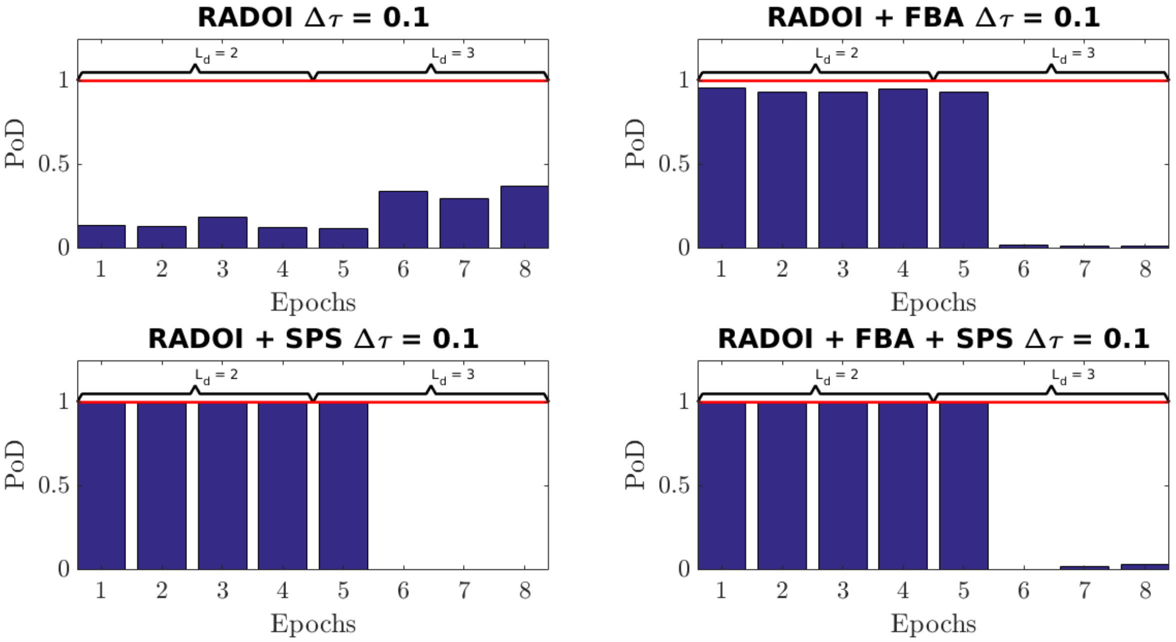


Figure 3.17: Probability of Detection at $\Delta\tau = 0.1$ for RADOI method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.18 we show the PoD for the RADOI method at $\Delta\tau = 0.1$ and $K = 15$. Note that the RADOI method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques RADOI shows a PoD above 95%.

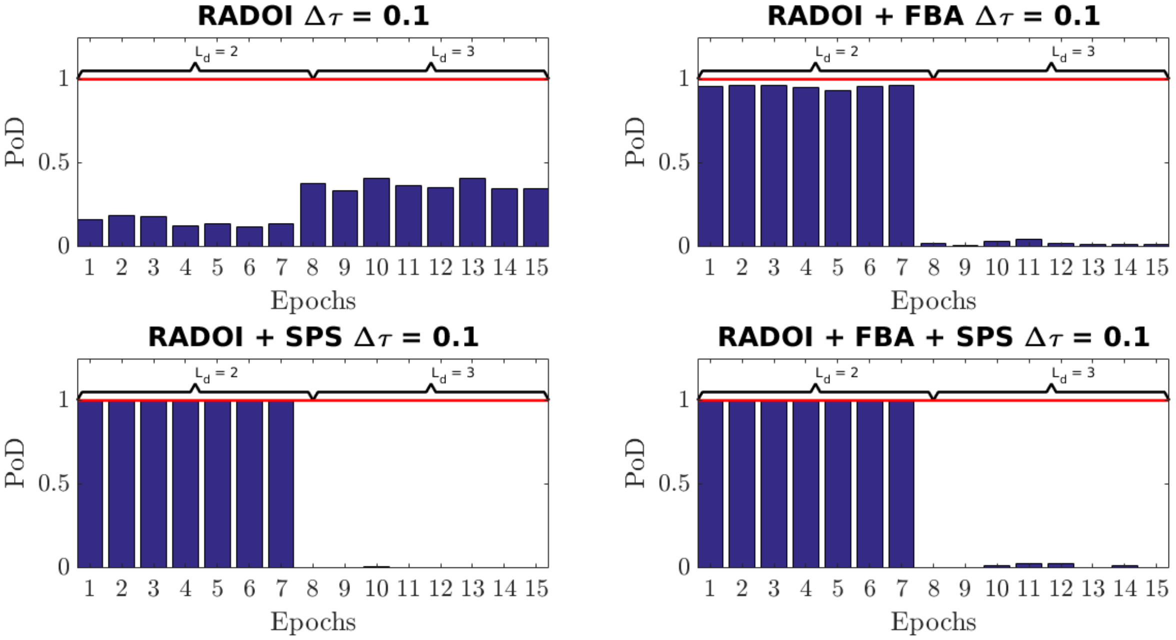


Figure 3.18: Probability of Detection at $\Delta\tau = 0.1$ for RADOI method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.19 we show the PoD for the RADOI method at $\Delta\tau = 0.1$ and $K = 30$. Note that the RADOI method fails to estimate the model order however by applying the preprocessing FBA and SPS techniques RADOI shows a PoD above 95%. Note that even after increasing the amount of epochs, the RADOI methods presents a poor performance when $L_d = 3$

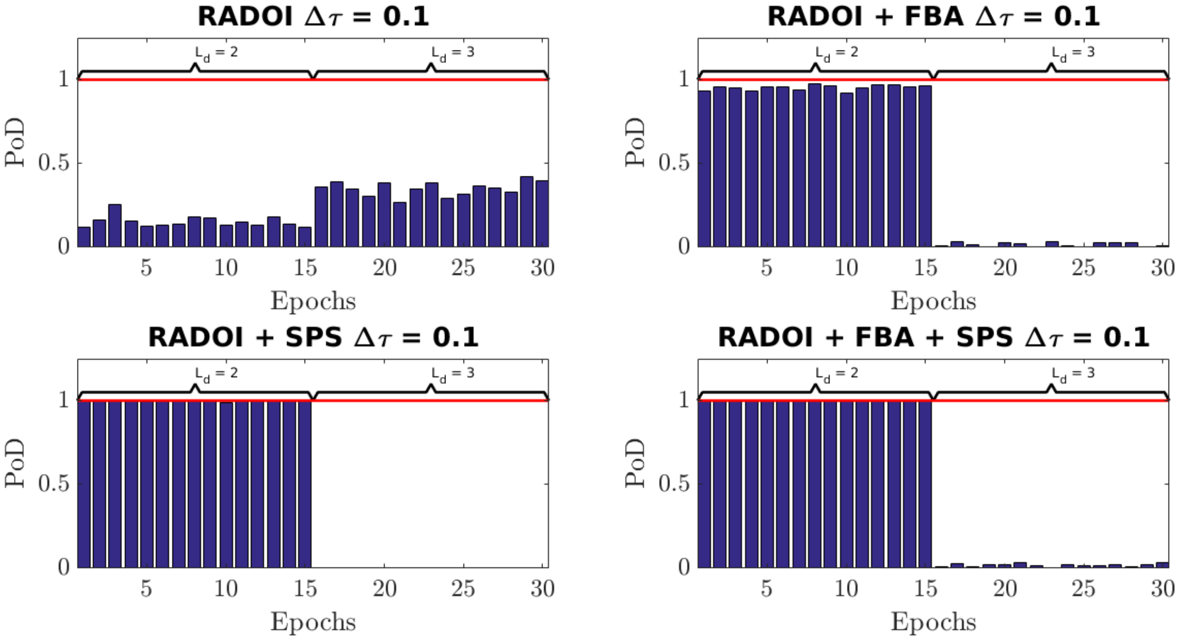


Figure 3.19: Probability of Detection at $\Delta\tau = 0.1$ for RADOI method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.20 we show the PoD for the RADOI method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 15$. Note that even though the RADOI method show a much better performance when signals are weakly correlated, the method has a poor performance when we have an $L_d = 3$.

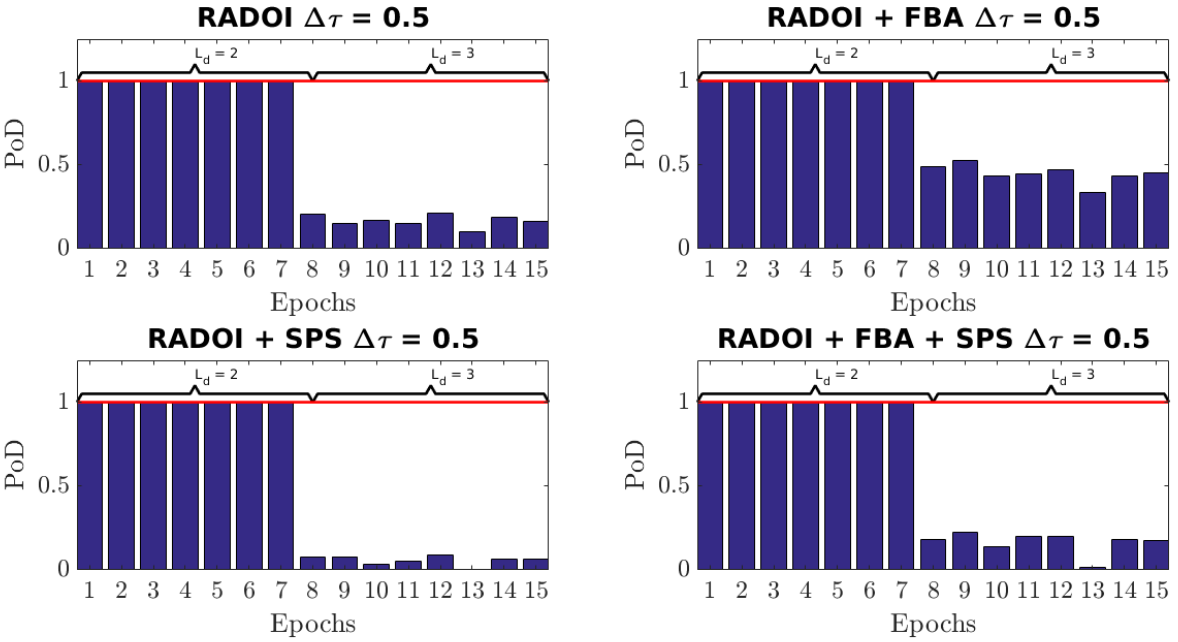


Figure 3.20: Probability of Detection at $\Delta\tau = 0.5$ for RADOI method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.21 we show the PoD for the RADOI method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 30$. Note that even though the RADOI method show a much better performance when signals are weakly correlated, the method has a poor performance when we have an $L_d = 3$.

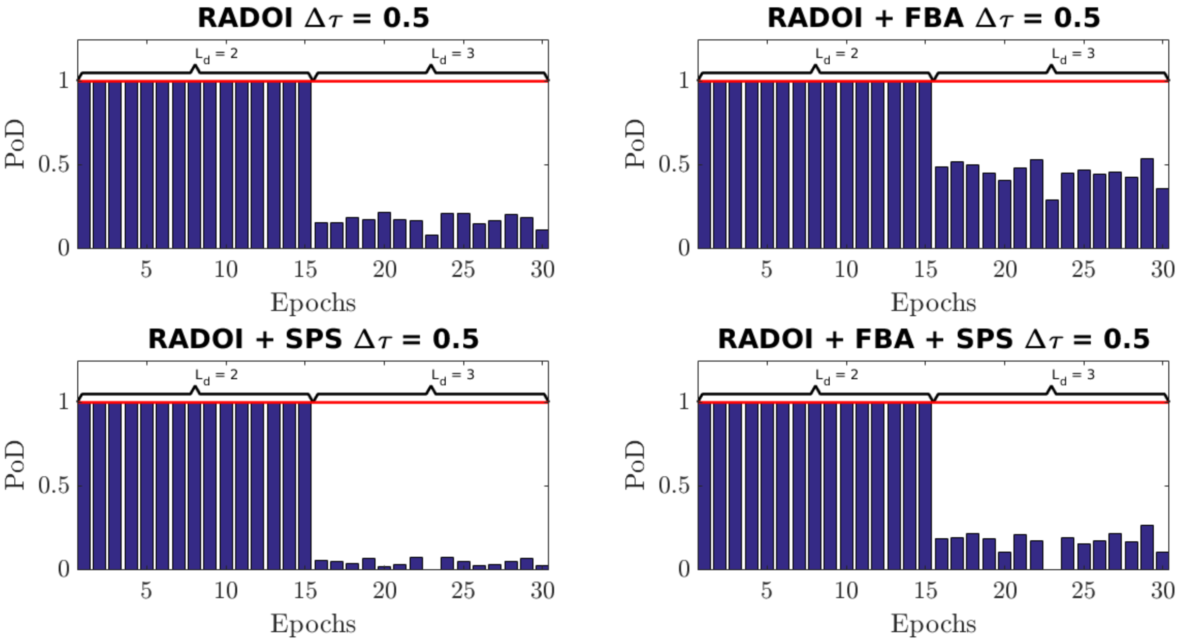


Figure 3.21: Probability of Detection at $\Delta\tau = 0.5$ for RADOI method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.22 we show the PoD for the RADOI+FBA method at $\Delta\tau = 0.1$ compared to the minimum and maximum model order defined for each epoch. Note that the RADOI combined with FBA tends to understimate the model order since that we have a PoD of 100% when comparing the estimated model order to the maximum model order within the received tensor $\boldsymbol{\mathcal{Y}}$.

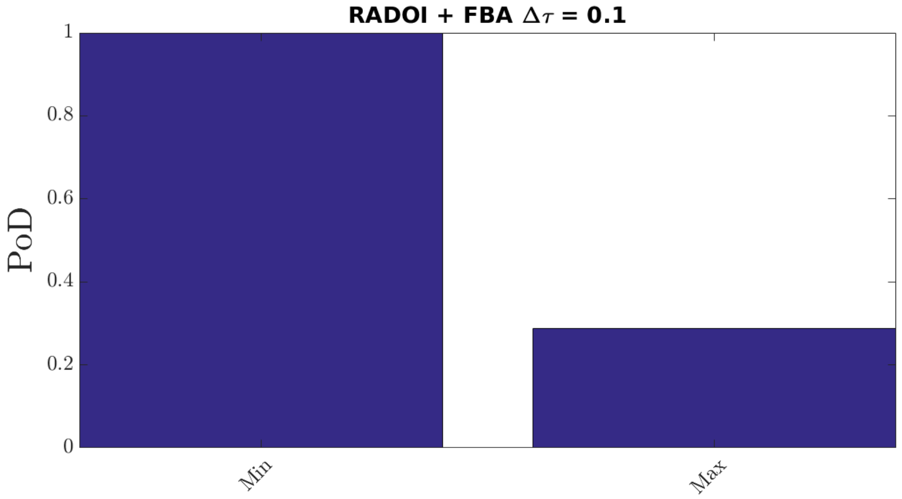


Figure 3.22: Probability of Detection at $\Delta\tau = 0.1$ for RADOI+FBA method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.23 we show the PoD for the MDL method at $\Delta\tau = 0.1$ and $K = 8$. Note that the MDL method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques MDL shows a PoD above 95% when $L_d = 2$ and above 70% when $L_d = 3$.

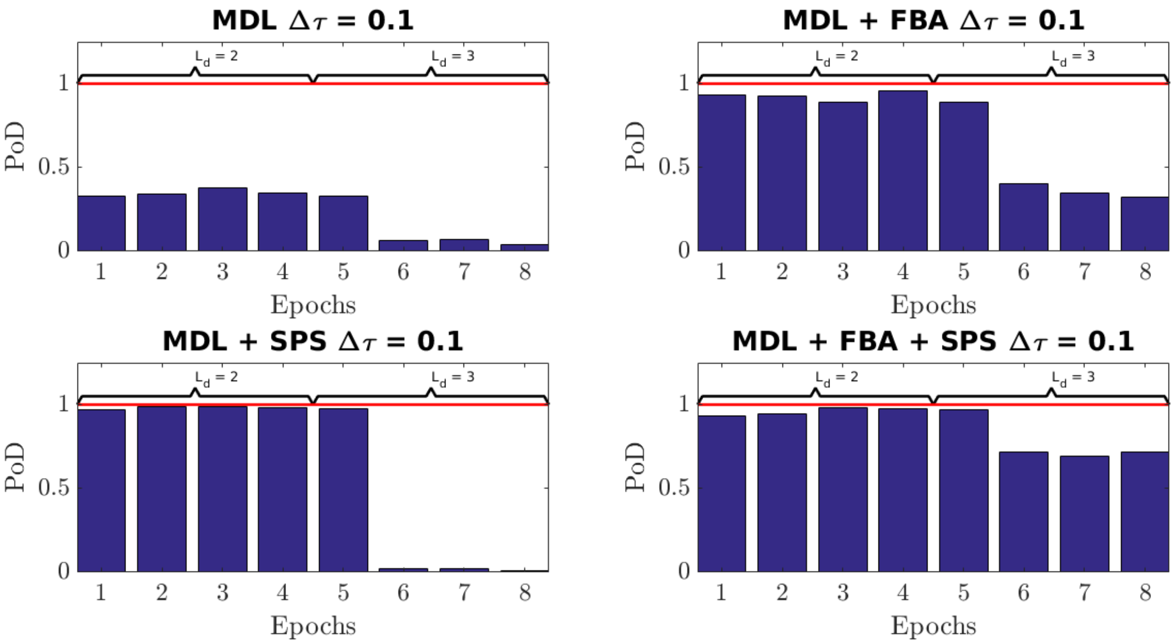


Figure 3.23: Probability of Detection at $\Delta\tau = 0.1$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.24 we show the PoD for the MDL method at $\Delta\tau = 0.1$ and $K = 15$. Note that even though the MDL method performs better than AIC, RADOI and ESTER, the MDL method does not benefit of the higher amount of epochs. Therefore, the MDL method combined with FBA and SPS shows a PoD above 95% when $L_d = 2$ and above 70% when $L_d = 3$.

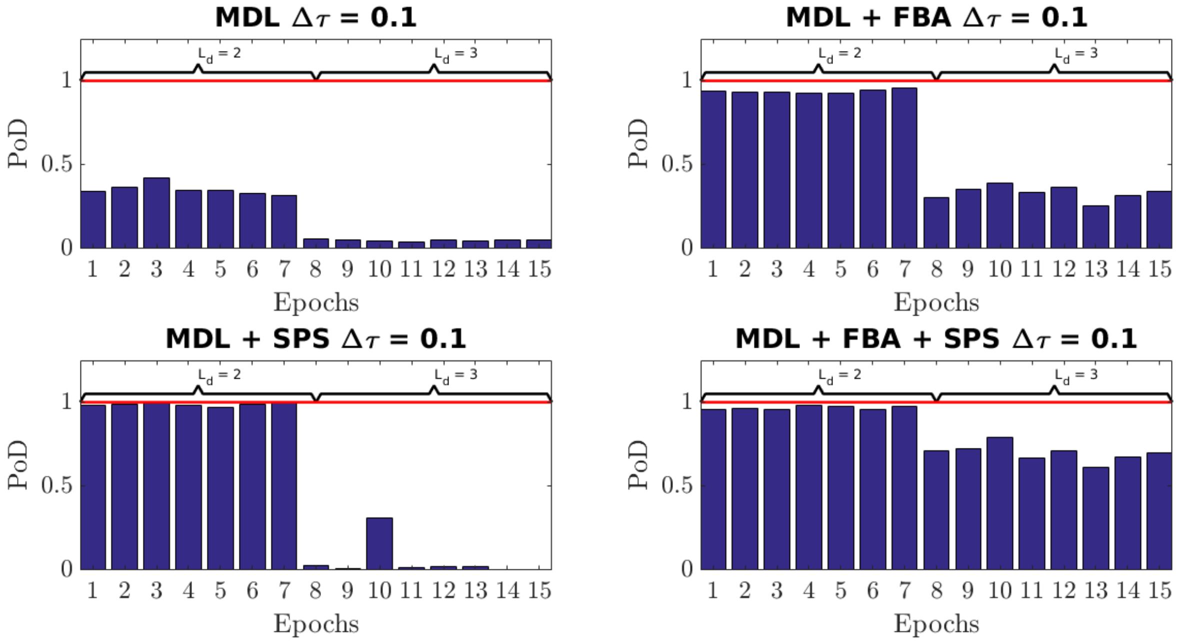


Figure 3.24: Probability of Detection at $\Delta\tau = 0.1$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.25 we show the PoD for the MDL method at $\Delta\tau = 0.1$ and $K = 30$. Note that even though the MDL method performs better than AIC, RADOI and ESTER, the MDL method does not benefit of the higher amount of epochs. Therefore, the MDL method combined with FBA and SPS shows a PoD above 95% when $L_d = 2$ and above 70% when $L_d = 3$.

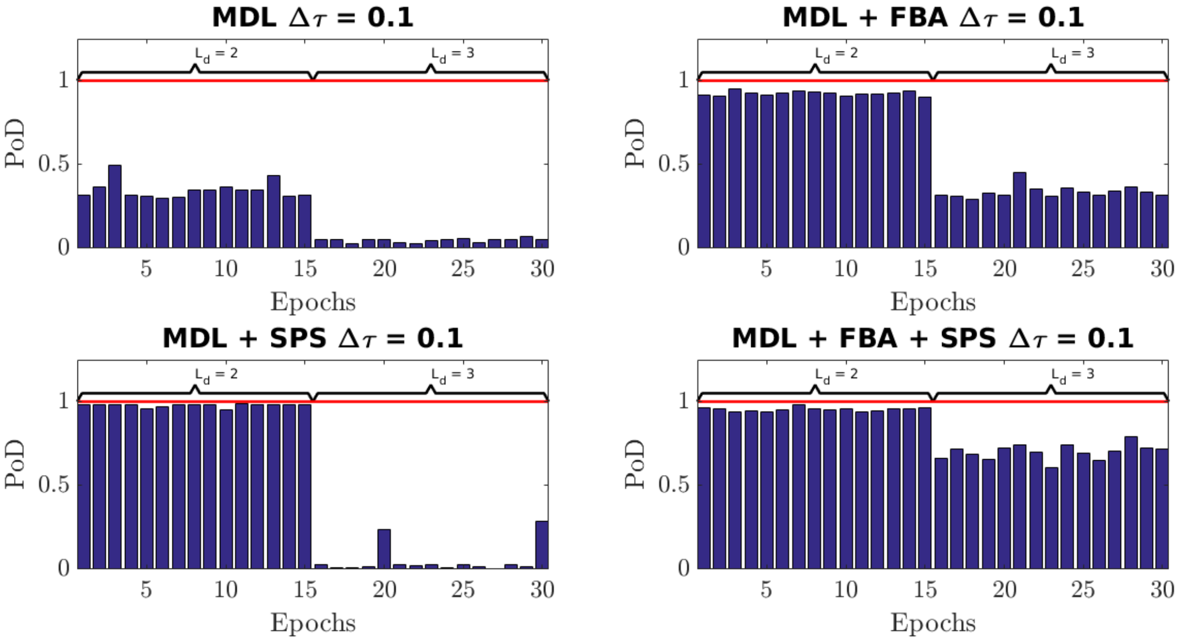


Figure 3.25: Probability of Detection at $\Delta\tau = 0.1$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.26 we show the PoD for the MDL method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 8$. Note that the MDL method shows an improved performance when signals are weakly correlated. Moreover, when applying the pre-processing techniques FBA and SPS we obtain a PoD of approximately 95% for both $L_d = 2$ and $L_d = 3$

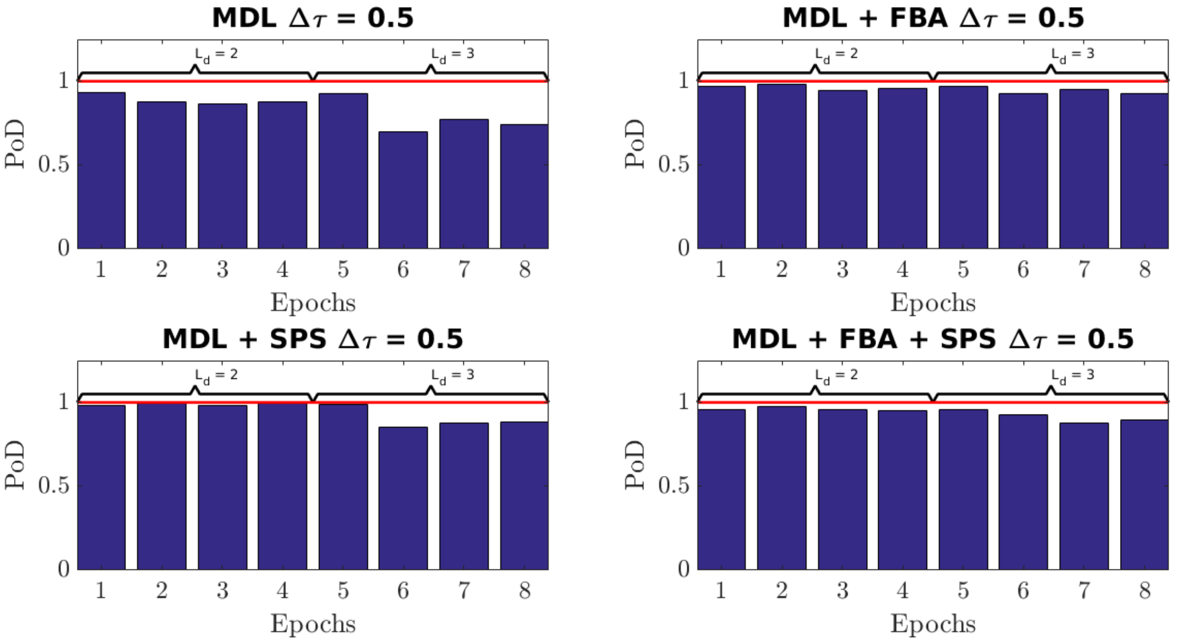


Figure 3.26: Probability of Detection at $\Delta\tau = 0.5$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.27 we show the PoD for the MDL method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 15$. Again, note that when applying the pre-processing techniques FBA and SPS we obtain a PoD of approximately 95% for both $L_d = 2$ and $L_d = 3$

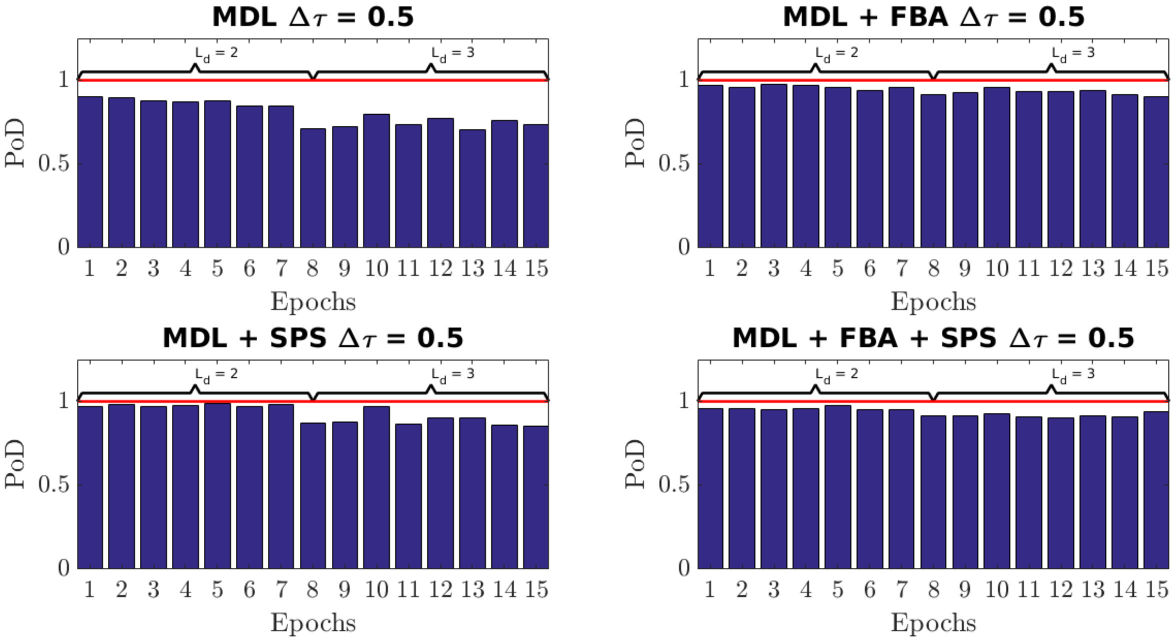


Figure 3.27: Probability of Detection at $\Delta\tau = 0.5$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.28 we show the PoD for the MDL method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 30$. Again, note that when applying the pre-processing techniques FBA and SPS we obtain a PoD of approximately 95% for both $L_d = 2$ and $L_d = 3$

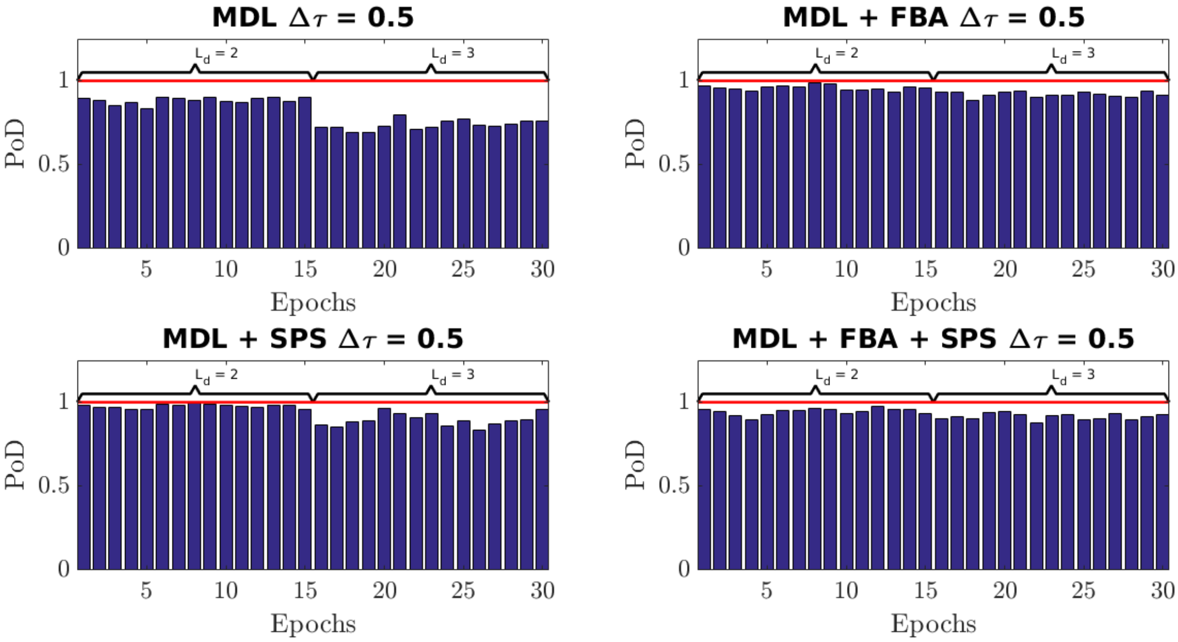


Figure 3.28: Probability of Detection at $\Delta\tau = 0.5$ for MDL method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.29 we show the PoD for the MDL+FBA+SPS method at $\Delta\tau = 0.1$ compared to the minimum and maximum model order defined for each epoch. Note that the MDL combined with FBA and SPS tends to understimate the model order since that we have a PoD of 100% when comparing the estimated model order to the maximum model order within the received tensor $\boldsymbol{\mathcal{Y}}$.

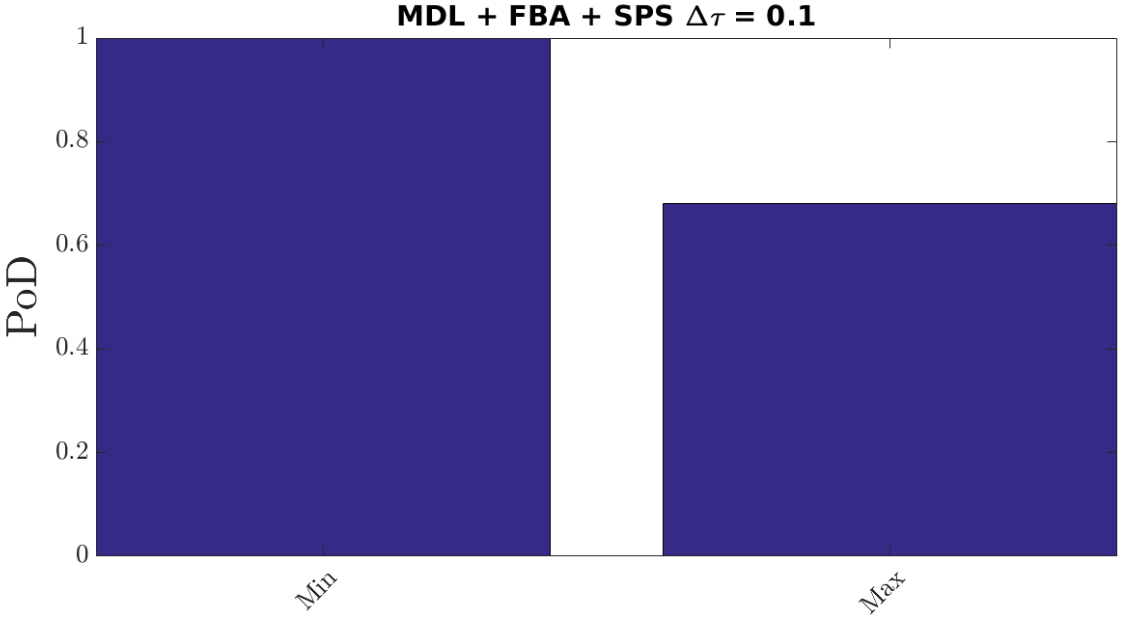


Figure 3.29: Probability of Detection at $\Delta\tau = 0.1$ for MDL+FBA+SPS method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.30 we show the PoD for the EFT method at $\Delta\tau = 0.1$ and $K = 8$. Note that the EFT method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques EFT shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$. Therefore, the EFT combined with FBA and SPS presents similar performance to the MDL+FBA+SPS method.

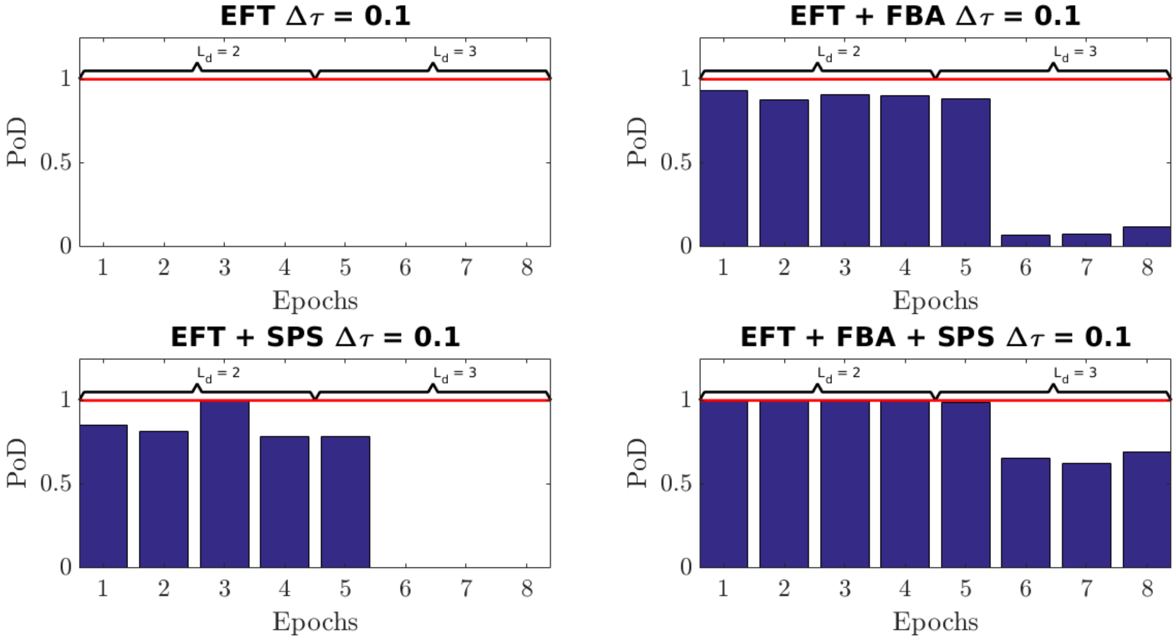


Figure 3.30: Probability of Detection at $\Delta\tau = 0.1$ for EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.31 we show the PoD for the EFT method at $\Delta\tau = 0.1$ and $K = 15$. Note that the EFT method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques EFT shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$. Therefore, the EFT combined with FBA and SPS presents similar performance to the MDL+FBA+SPS method.

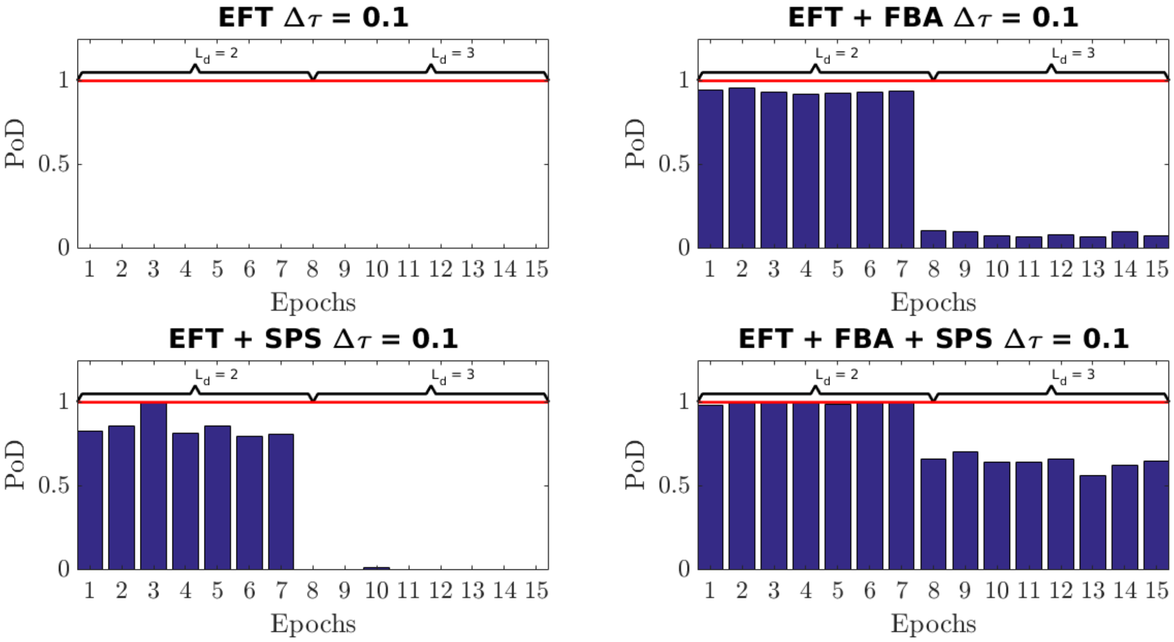


Figure 3.31: Probability of Detection at $\Delta\tau = 0.1$ for EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.32 we show the PoD for the EFT method at $\Delta\tau = 0.1$ and $K = 30$. Note that the EFT method fails to estimate the model order however by applying the pre-processing FBA and SPS techniques EFT shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$. Therefore, the EFT combined with FBA and SPS presents similar performance to the MDL+FBA+SPS method.

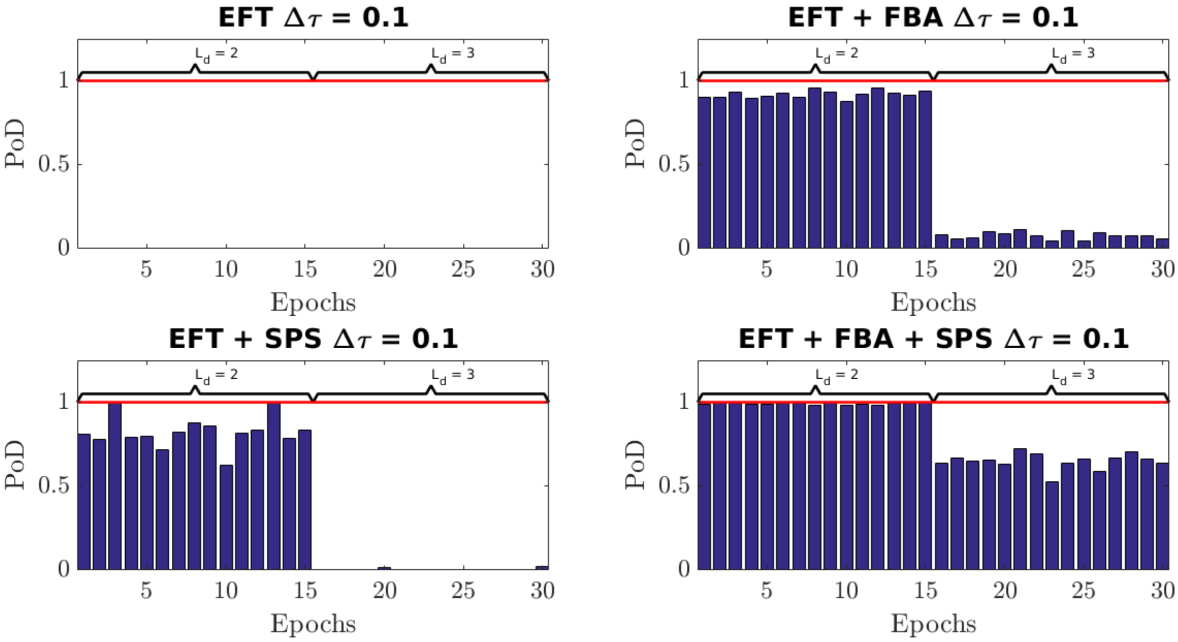


Figure 3.32: Probability of Detection at $\Delta\tau = 0.1$ for EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.33 we show the PoD for the EFT method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 8$. Note that even though the EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD above 95% for both $L_d = 2$ and $L_d = 3$

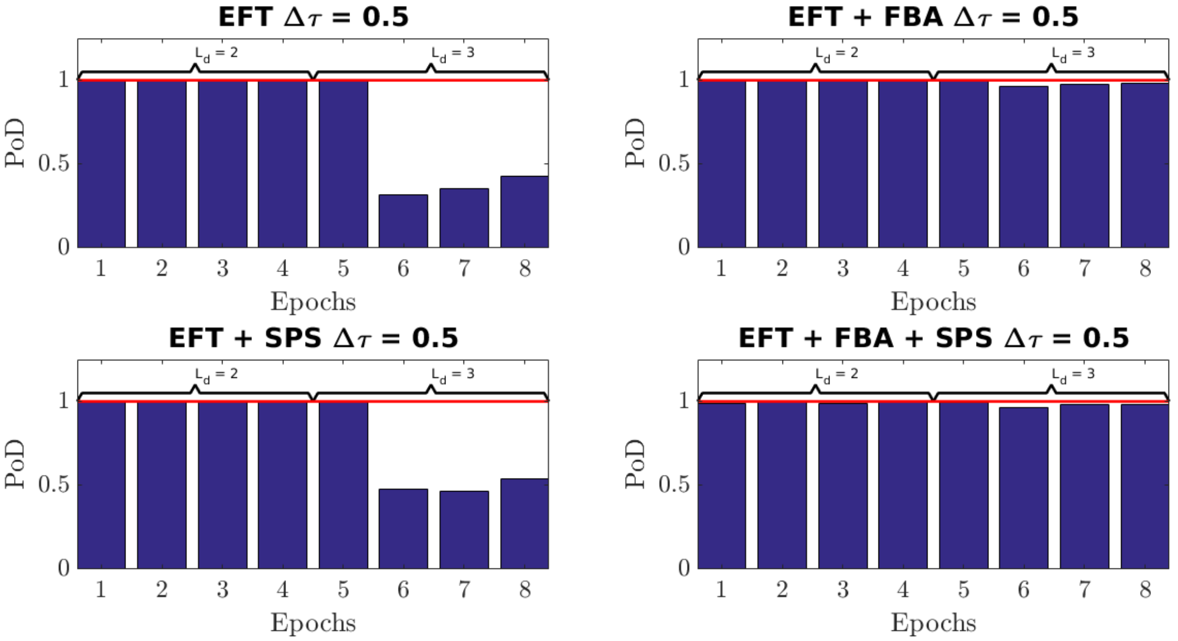


Figure 3.33: Probability of Detection at $\Delta\tau = 0.5$ for EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.34 we show the PoD for the EFT method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 15$. Note that even though the EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD above 95% for both $L_d = 2$ and $L_d = 3$

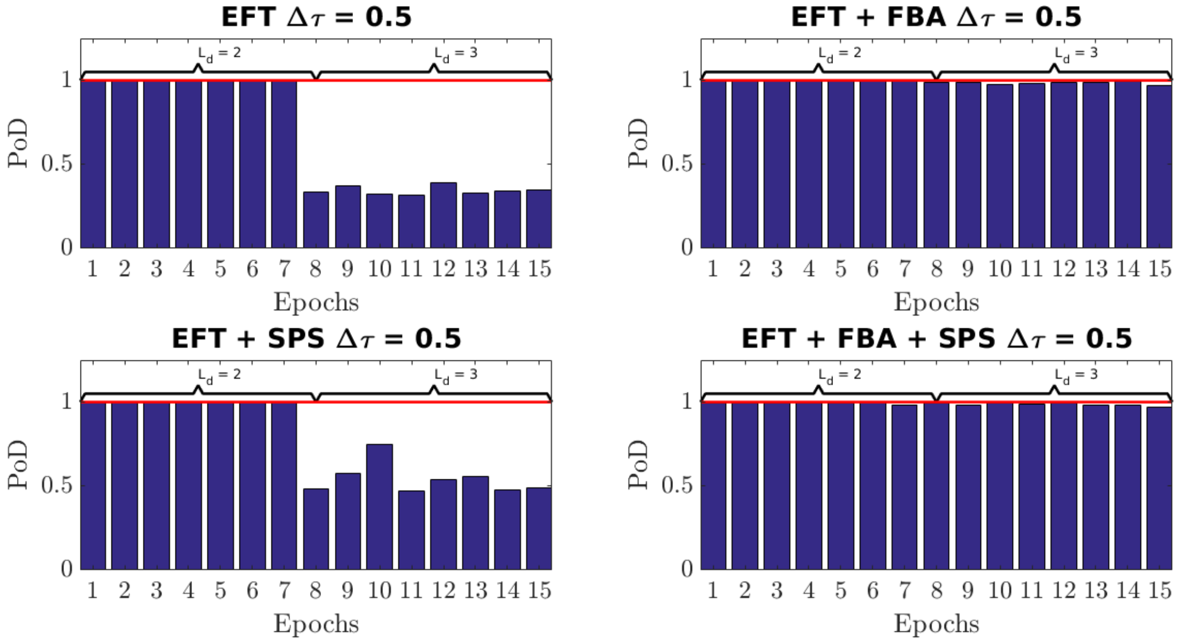


Figure 3.34: Probability of Detection at $\Delta\tau = 0.5$ for EFT method in a Dynamic scenario with an perfect array with $M = 15$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.35 we show the PoD for the EFT method at $\Delta\tau = 0.5$, since above it the PoD remains stable, and $K = 30$. Note that even though the EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD above 95% for both $L_d = 2$ and $L_d = 3$

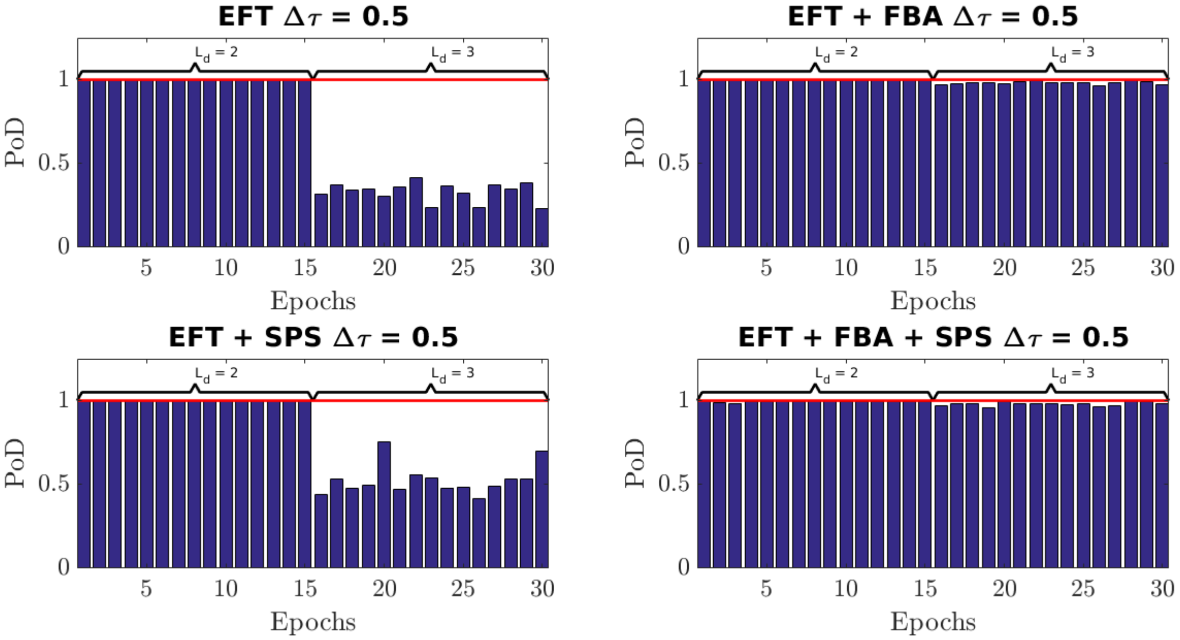


Figure 3.35: Probability of Detection at $\Delta\tau = 0.5$ for EFT method in a Dynamic scenario with an perfect array with $M = 30$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.36 we show the PoD for the EFT+FBA method at $\Delta\tau = 0.1$ and we compare the minimum and maximum model order defined for each epoch. Note that, the EFT combined with FBA and SPS equaly estimate the minimum and maximum model order.

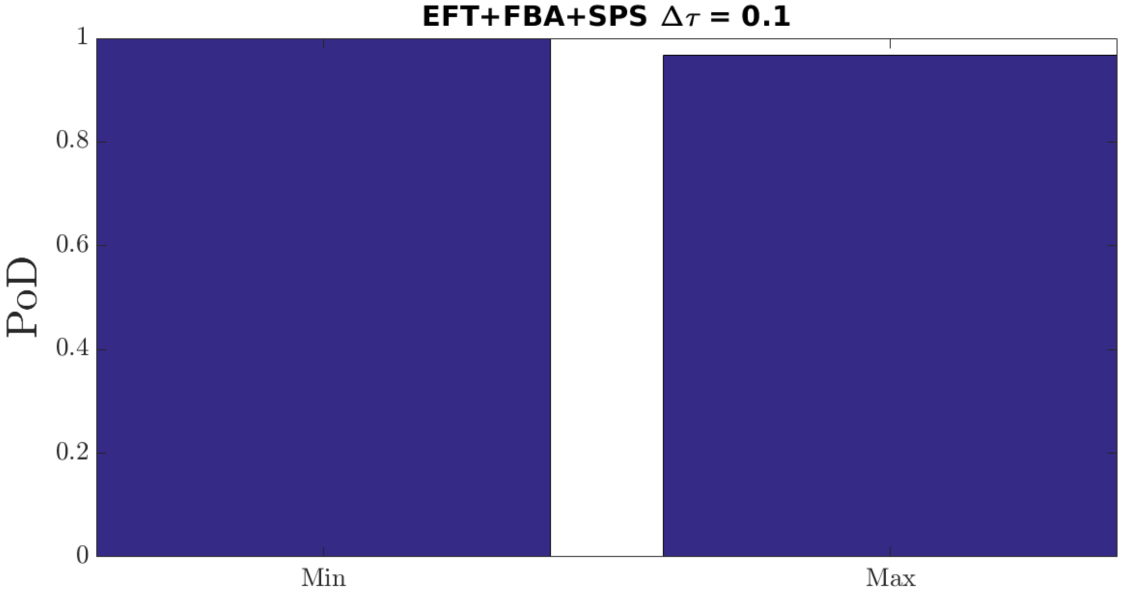


Figure 3.36: Probability of Detection at $\Delta\tau = 0.1$ for EFT+FBA method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.37 we show the PoD for the M-EFT method at $\Delta\tau = 0.1$ and $K = 8$. Note that the M-EFT method presents similar performance to the EFT method. Therefore, the M-EFT method combined with FBA and SPS shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$

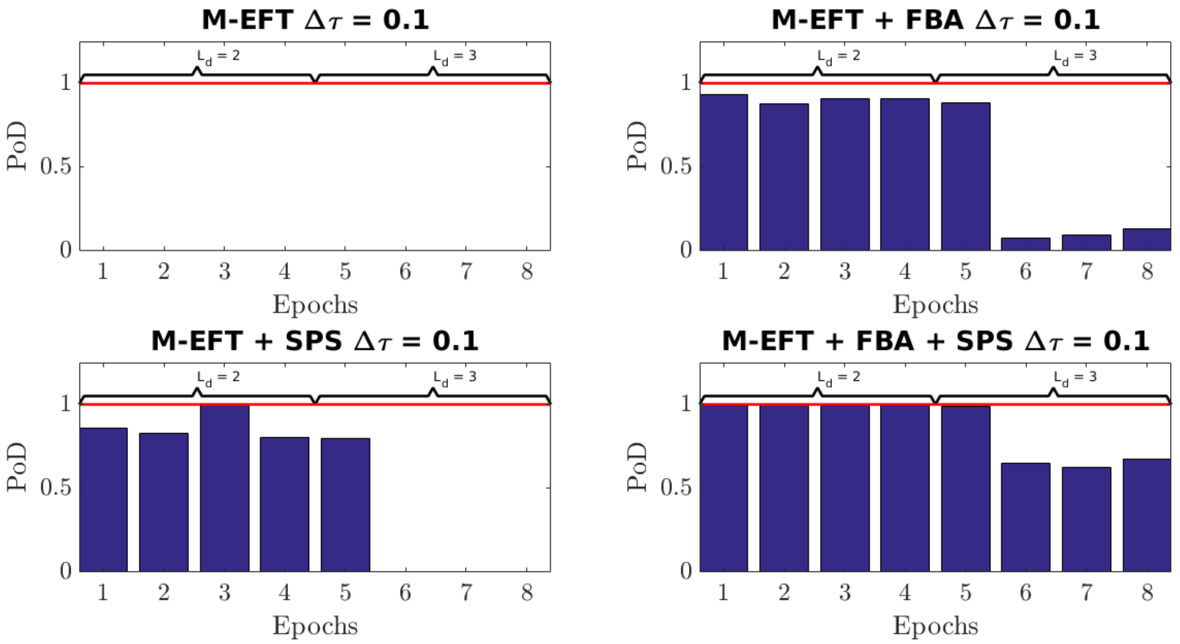


Figure 3.37: Probability of Detection at $\Delta\tau = 0.1$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.38 we show the PoD for the M-EFT method at $\Delta\tau = 0.1$ and $K = 15$. Note that the M-EFT method presents similar performance to the EFT method. Therefore, the M-EFT method combined with FBA and SPS shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$

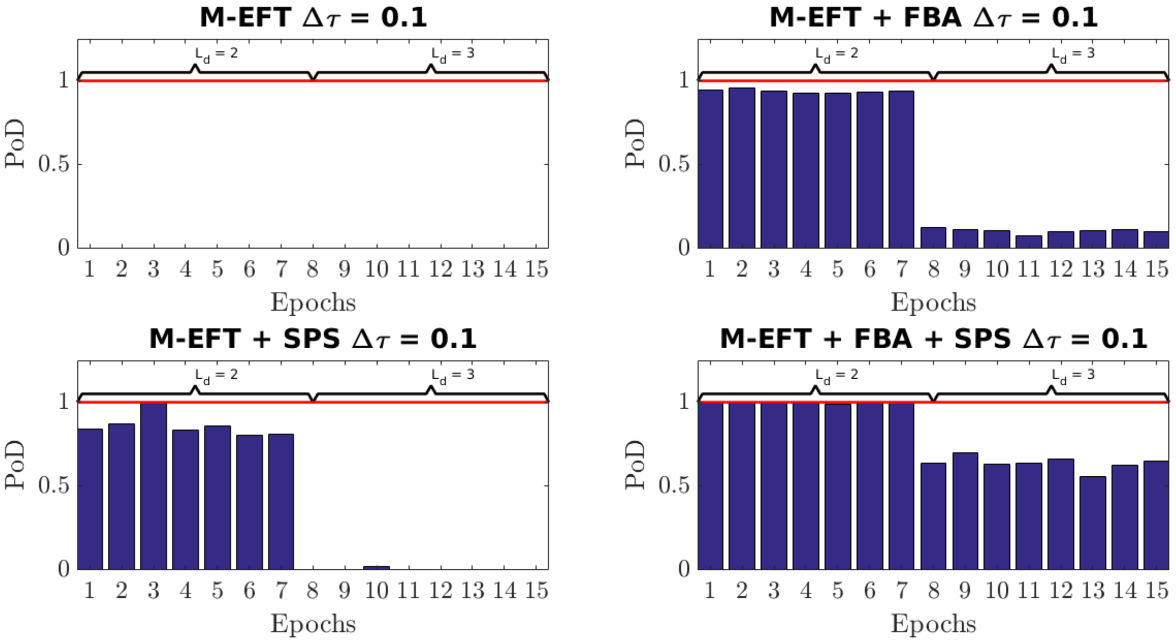


Figure 3.38: Probability of Detection at $\Delta\tau = 0.1$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.39 we show the PoD for the M-EFT method at $\Delta\tau = 0.1$ and $K = 30$. Note that the M-EFT method presents similar performance to the EFT method. Therefore, the M-EFT method combined with FBA and SPS shows a PoD above 95% when $L_d = 2$ and above 65% when $L_d = 3$

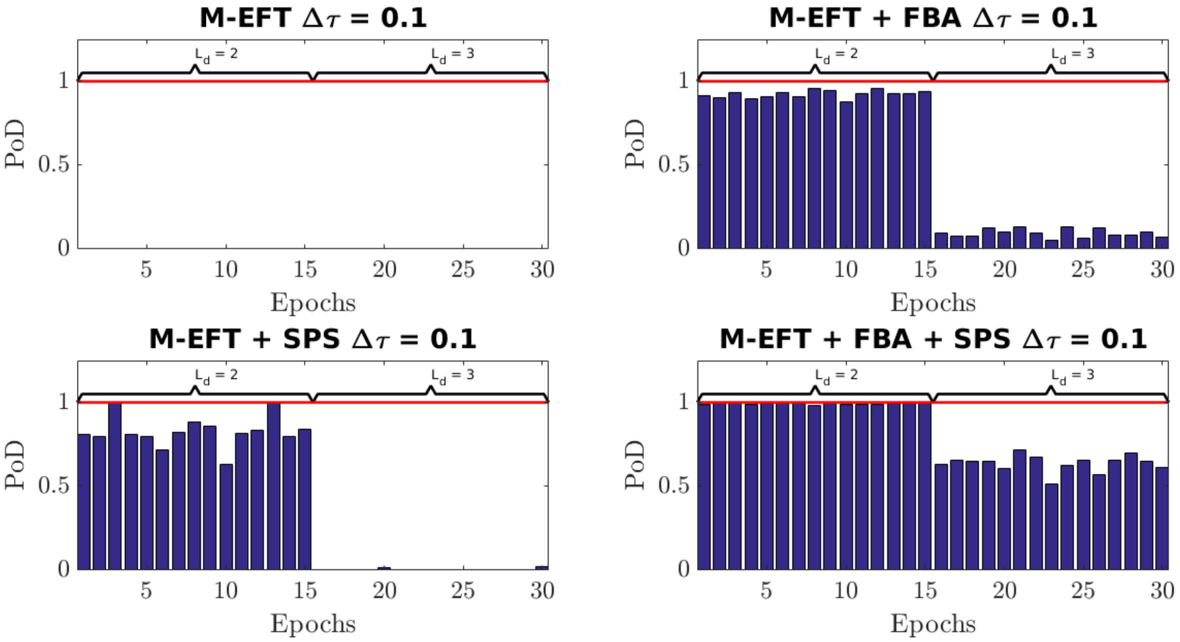


Figure 3.39: Probability of Detection at $\Delta\tau = 0.1$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.40 we show the PoD for the M-EFT method at $\Delta\tau = 0.5$ and $K = 8$ since above it the PoD remains stable. Note that even though the M-EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD of above 98% for both $L_d = 2$ and $L_d = 3$. Observe that, the M-EFT method shows an increased performance when signals are less correlated.

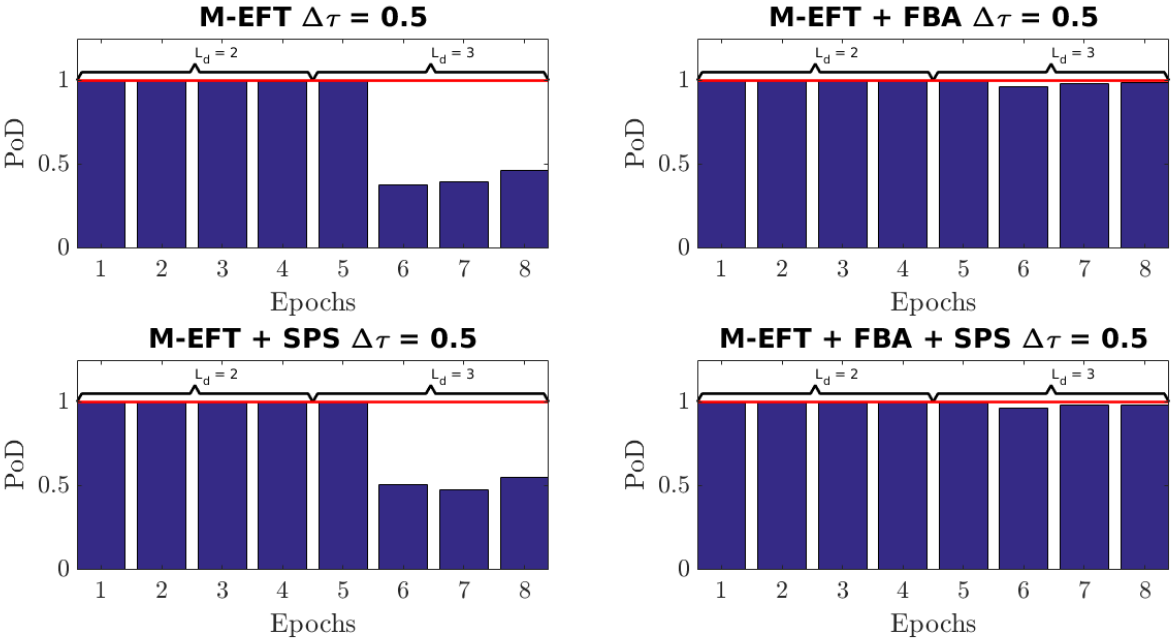


Figure 3.40: Probability of Detection at $\Delta\tau = 0.5$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.41 we show the PoD for the M-EFT method at $\Delta\tau = 0.5$ and $K = 15$ since above it the PoD remains stable. Note that even though the M-EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD of above 98% for both $L_d = 2$ and $L_d = 3$. Observe that, the M-EFT method shows an increased performance when signals are less correlated.

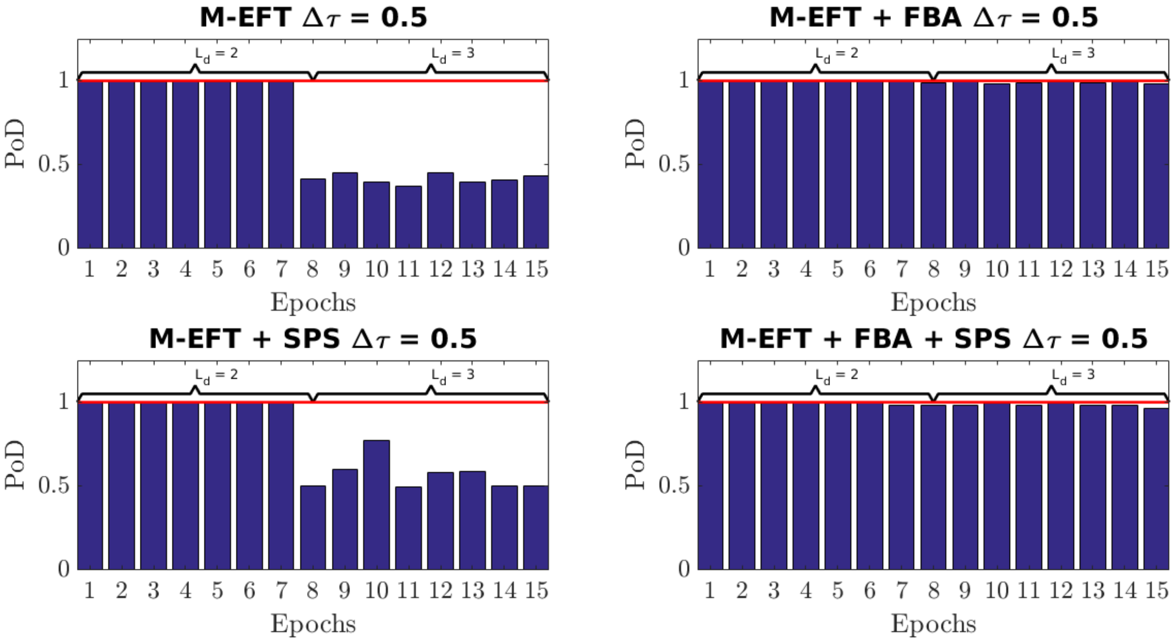


Figure 3.41: Probability of Detection at $\Delta\tau = 0.5$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.42 we show the PoD for the M-EFT method at $\Delta\tau = 0.5$ and $K = 30$ since above it the PoD remains stable. Note that even though the M-EFT method shows an improved performance when signals are weakly correlated, by applying the pre-processing techniques FBA and SPS we obtain a PoD of above 98% for both $L_d = 2$ and $L_d = 3$. Observe that, the M-EFT method shows an increased performance when signals are less correlated.

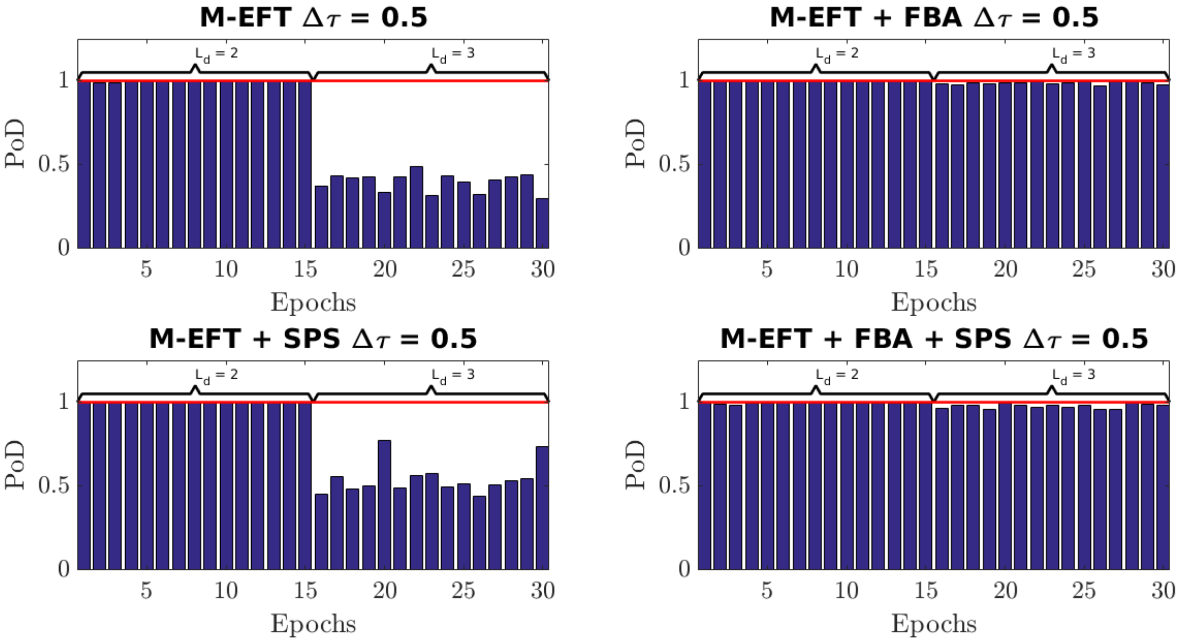


Figure 3.42: Probability of Detection at $\Delta\tau = 0.5$ for M-EFT method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.43 we show the PoD for the M-EFT+FBA+SPS method at $\Delta\tau = 0.1$ and we compare the minimum and maximum model order defined for each epoch. Note that, diferently from previous methods, the M-EFT combined with FBA and SPS tends to correctly select the model order.

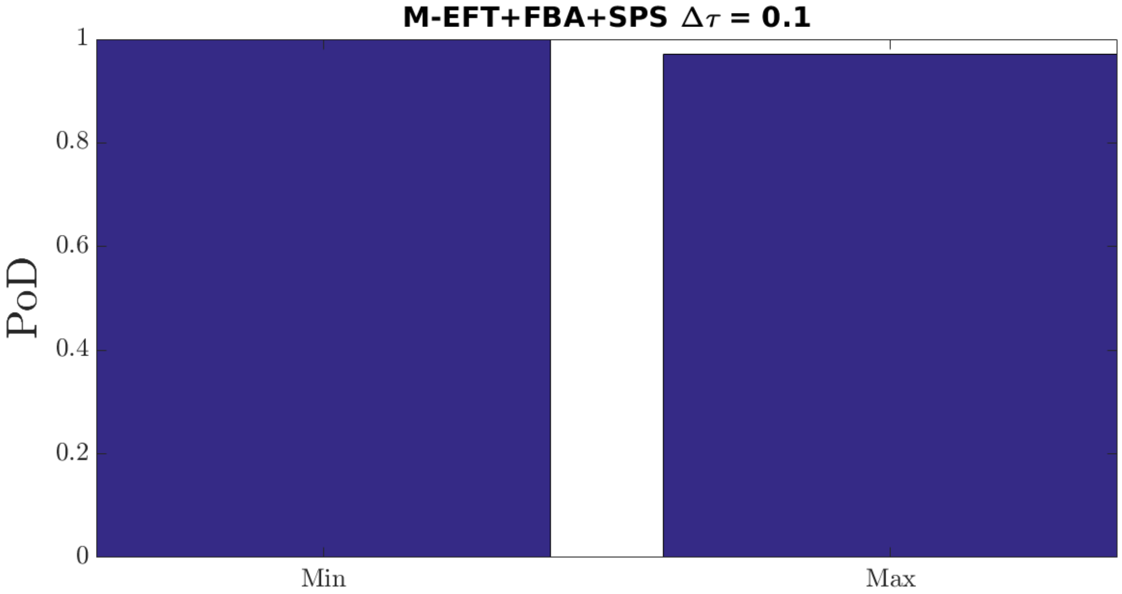
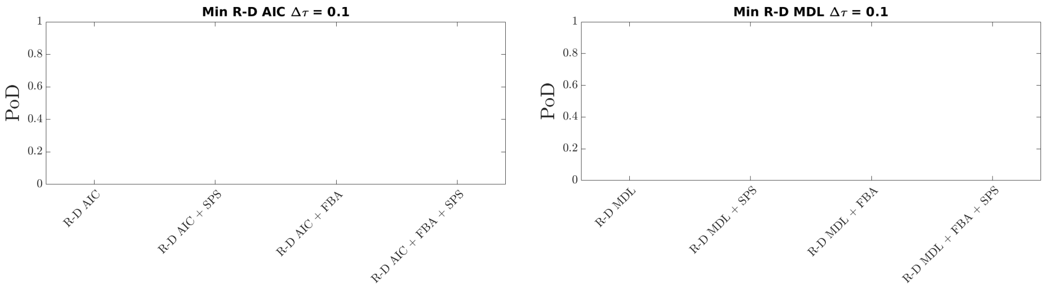


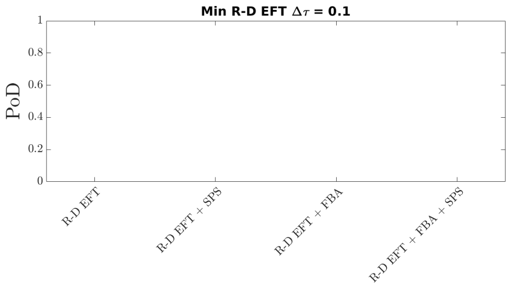
Figure 3.43: Probability of Detection at $\Delta\tau = 0.1$ for M-EFT+FBA method in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 10$ epochs, and have $N = 245520$ samples.

In Figure 3.44 we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 8$. We compared the estimate model orde to the minimum model order. Note that, both the R-D AIC, R-D MDL and R-D EFT have a PoD of 0% therefore the R-D methods failed to detect the tensor model order. Moreover, the pre-processing methods did not help to improve the model order estimation. Furthermore, when verifying the simulation results individually we observed that the R-D methods overestimated the model order. Frequently, the R-D methods estimate a model order above 3.



(a) R-D AIC

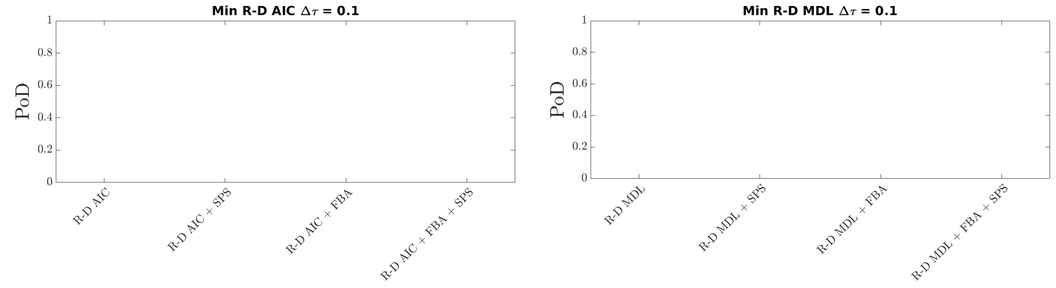
(b) R-D MDL



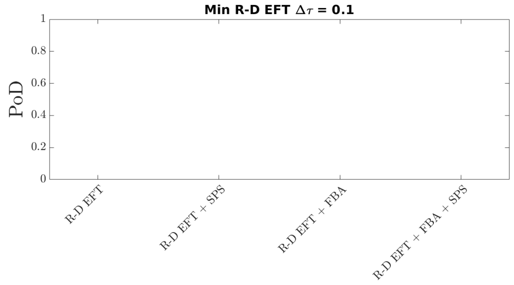
(c) R-D EFT

Figure 3.44: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.45 we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 15$. We compared the estimate model order to the minimum model order. Note that, the R-D methods have a PoD of 0% therefore the R-D methods failed to detect the tensor model order. Moreover, the pre-processing methods did not help to improve the model order estimation. Furthermore, when verifying the simulation results individually we observed that the methods overestimated the model order. Frequently, the R-D methods estimate a model order above 3.



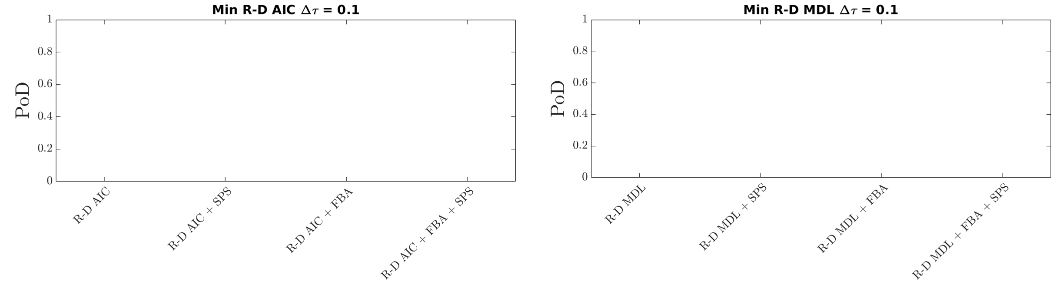
(a) R-D AIC (b) R-D MDL



(c) R-D EFT

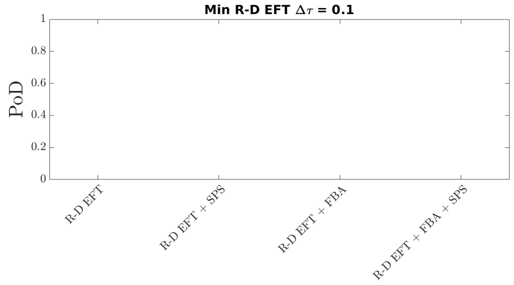
Figure 3.45: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.46 we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 30$. We compared the estimate model ordes to the minimum model order. Note that, the R-D methods have a PoD of 0% therefore these methods failed to detect the tensor model order. Moreover, the pre-processing methods did not help to improve the model order estimation. Furthermore, when verifying the simulation results individually we observed that the R-D methods overestimated the model order. Frequently, estimating a model order above 3.



(a) R-D AIC

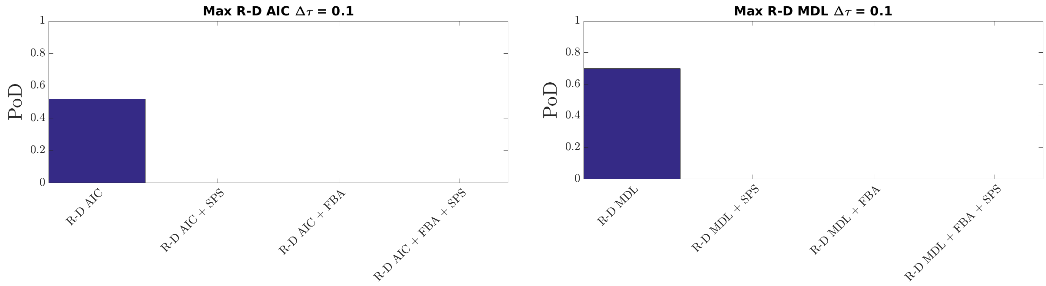
(b) R-D MDL



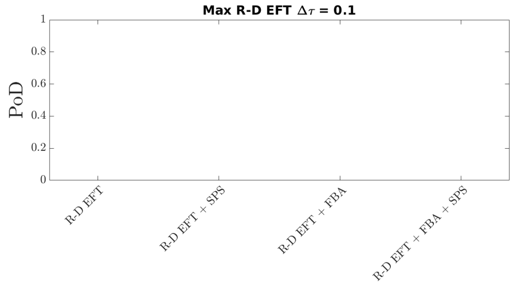
(c) R-D EFT

Figure 3.46: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.47 we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 8$. We compared the estimate model orde to the maximum model order. Note that, the R-D methods without pre-processing have a PoD of about 0.5% therefore these failed to detect the tensor model order. When applying the pre-processing techniques we have a worst performance with a PoD of $0\%$. Moreover, when verifying the simulation results individually we observed that the R-D methods overestimated the model order, frequently, above 3.



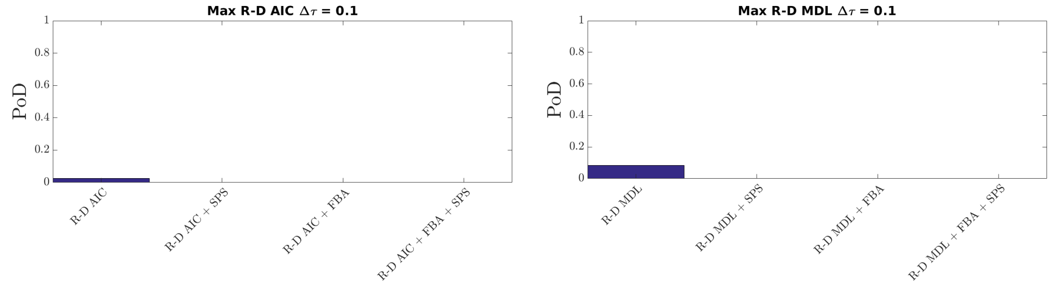
(a) R-D AIC (b) R-D MDL



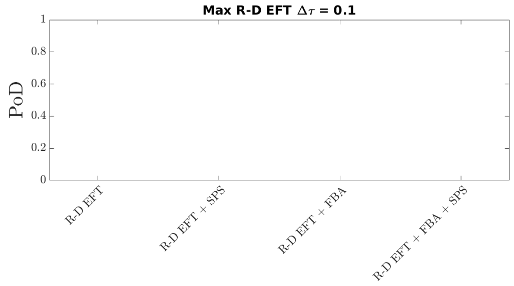
(c) R-D EFT

Figure 3.47: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.48 we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 15$. We compared the estimate model order to the maximum model order. Note that, these methods have a PoD of about 0.02$\%$ therefore the methods failed to detect the tensor model order. When applying the pre-processing techniques we have a worst performance with a PoD of $0\%$. Moreover, when verifying the simulation results individually we observed that the R-D methods overestimated the model order by 1.



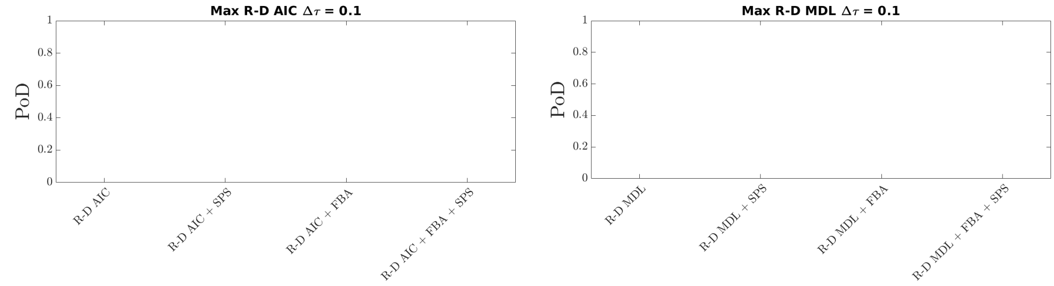
(a) R-D AIC (b) R-D MDL



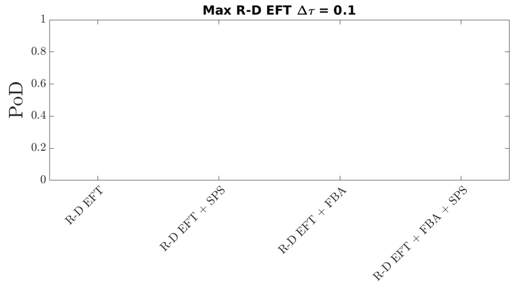
(c) R-D EFT

Figure 3.48: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.49a we show the PoD for the R-D AIC, R-D MDL and R-D EFT methods at $\Delta\tau = 0.1$ and $K = 30$. We compared the estimate model order to the maximum model order. Note that, these shown a PoD of 0% therefore the R-D methods failed to detect the tensor model order. When applying the pre-processing techniques we have a worst performance with a PoD of $0\%$. Moreover, when verifying the simulation results individually we observed that the R-D methods overestimated the model order. Frequently, the R-D methods estimate a model order above 3.



(a) R-D AIC (b) R-D MDL



(c) R-D EFT

Figure 3.49: Probability of Detection at $\Delta\tau = 0.1$ for R-D methods in a Dynamic scenario with an perfect array with $M = 8$ antennas. Code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Therefore, the AIC, ESTER and RADOI methods are not suitable to perform model order selection on GPS3 signals. Moreover, the MDL, EFT, and M-EFT also show poor performance. However, when applying the pre-processing methods FBA and SPS we improve the model order selection performance. Mainly, we acquire better result when combining these pre-processing methods with the MDL, EFT and M-EFT methods. Furthermore, note that the MOS methods have a higher PoD when performing model order selection on epochs with $L_d = 2$. The worst performance when $L_d = 3$ is due to the low power of the second NLOS component thus the second NLOS is identified as noise when performing MOS. Additionally, observe that when increasing the time-delay difference $\Delta\tau$ all matrix-based MOS methods presented better performance. Moreover, the tensor-based R-D AIC, R-D MDL, and R-D EFT show poor performance since these methods do not correctly estimate the tensor model order. Moreover, applying pre-processing methods to R-D methods did not improve the model order selection. Finally, we show that the R-D methods are not suitable to dynamic scenarios where we have varying model order.

## 3.2 Proposed Tensor-based Factor Matrices Estimation

In this subsection, we present the proposed Canonical Polyadic method utilized to estimate the factor matrices in the second element of the framework. In [15] the authors show that the state-of-the-art DoA/Krf, CPD-GEVD, and HOSVD SECSI methods present excellent performance when we have a perfectly aligned array of antennas. Moreover, [15] show that when we have imperfections in the antenna array alignment the performance of the DoA/Krf methods is severely deteriorated while the CPD-GEVD and HOSVD SECSI methods have similar performance to the case with a correctly aligned array of antennas. Moreover, [15] shows that the HOSVD SECSI method presents a higher complexity than the CPD-GEVD method without showing significant overall improvement. Despite the HOSVD higher complexity, this method presents a better performance in scenarios with strongly correlated signals. Consequently, the HOSVD SECSI method is more reliable in highly demanding scenarios. However, throughout simulations, we show that, by utilizing different tensor modes, we can have yet greater performance in dynamic scenarios. Therefore, we decided to implement the Mode 1 HOSVD SECSI with left-hand matrix (Mode 1 HOSVD SECSI) method to perform the factor matrix estimation.

The Mode 1 HOSVD SECSI method was originally proposed in [26] to provide a more accurate tensor-based scheme. The Mode 1 HOSVD SECSI performs Canonical Polyadic Decomposition (CPD), also known as Candecomp and PARAFAC, by rearranging the tensor factor estimation problem into several redundant simultaneous matrix diagonalization problems. As illustrated in Figure 3.50 the Mode 1 HOSVD SECSI method first computes the HOSVD low rank approximation of the sub-tensors $\boldsymbol{\mathcal{Y}}^{(t)}$ obtained from grouping the tensor $\boldsymbol{\mathcal{Y}}$ epochs according to their model order.

$$\boldsymbol{\mathcal{Y}}^{(t)} \approx \boldsymbol{\mathcal{S}}^{(t)} \times_1 \mathbf{U}_1^{(t)} \times_1 \mathbf{U}_2^{(t)} \times_3 \mathbf{U}_3^{(t)}, \tag{3.8}$$

where the superscript ${}^{(t)}$ indicates the $t$th sub-tensor, $\mathbf{U}_1^{(t)} \in \mathbb{C}^{K \times L_d(k)}$, $\mathbf{U}_2^{(t)} \in \mathbb{C}^{Q \times L_d(k)}$, and $\mathbf{U}_3^{(t)} \in \mathbb{C}^{M \times L_d(k)}$, are the truncated singular matrices related to the $t$th sub-tensor. Moreover, to compute the singular matrices and the core tensor, the SVD is applied to the unfolding of the tensor for each dimension. Thus, we construct the following equations

$$[\boldsymbol{\mathcal{Y}}]_{(1)}^{(t)} = \mathbf{U}_1^{(t)} \mathbf{S}_1^{(t)} \mathbf{V}_1^{\mathrm{H}(t)}, \tag{3.9}$$

$$[\boldsymbol{\mathcal{Y}}]_{(2)}^{(t)} = \mathbf{U}_2^{(t)} \mathbf{S}_2^{(t)} \mathbf{V}_2^{\mathrm{H}(t)}, \tag{3.10}$$

$$[\boldsymbol{\mathcal{Y}}]_{(3)}^{(t)} = \mathbf{U}_3^{(t)} \mathbf{S}_3^{(t)} \mathbf{V}_3^{\mathrm{H}(t)}, \tag{3.11}$$

where $\mathbf{U}_i^{(t)}$, $\mathbf{S}_i^{(t)}$, and $\mathbf{V}_i^{\mathrm{H}(t)}$ stand for the left singular vector matrix, singular value matrix and right singular vector matrix for the $i$th dimension, respectively. Consider that the left singular vector matrices can be be sequentially computed as follows:

$$[\boldsymbol{\mathcal{Y}}]_{(1)}^{(t)} = \mathbf{U}_1^{(t)}\mathbf{S}_1^{(t)}\mathbf{V}_1^{\mathrm{H(t)}}, \tag{3.12}$$

$$\left[\boldsymbol{\mathcal{Y}}^{(t)} \times_1 \mathbf{U}_1^{\mathrm{H(t)}}\right]_{(2)} = \mathbf{U}_2^{(t)}\mathbf{S}_2^{(t)}\mathbf{V}_2^{\mathrm{H(t)}}, \tag{3.13}$$

$$\left[\boldsymbol{\mathcal{Y}}^{(t)} \times_1 \mathbf{U}_1^{\mathrm{H(t)}} \times_2 \mathbf{U}_2^{\mathrm{H}(t)}\right]_{(3)} = \mathbf{U}_3^{(t)}\mathbf{S}_3^{(t)}\mathbf{V}_3^{\mathrm{H}(t)}. \tag{3.14}$$

Next, let us consider the third-mode unfolding of $\boldsymbol{\mathcal{Y}}$ and use the representation in (2.9) and (3.8), then, we obtain

$$\mathbf{U}_1^{(t)}[\boldsymbol{\mathcal{S}}]_{(1)}^{(t)}(\mathbf{U}_2^{(t)} \otimes \mathbf{U}_3^{(t)})^{\mathrm{T}} = \mathbf{\Gamma}^{\mathrm{T}(t)}[\boldsymbol{\mathcal{I}}_{3,\mathrm{L}}]_{(1)}^{(t)}((\mathbf{C}\mathbf{Q}_{\omega}^{(d)(t)})^{\mathrm{T}} \otimes \mathbf{A}^{(t)})^{\mathrm{T}}. \tag{3.15}$$

Hence, the subspace spanned by the columns of $\mathbf{U}_1^{(t)}$, $\mathbf{U}_2^{(t)}$, and $\mathbf{U}_3^{(t)}$ and the subspace spanned by the columns of ${\mathbf{\Gamma}^{(\mathbf{t})}}^{\mathrm{T}}$, $(\mathbf{C}\mathbf{Q}_{\omega}^{(d)(t)})^{\mathrm{T}}$, and $\mathbf{A}^{(t)}$ are respectively identical. Consequently, there exist a set of non-singular transform matrices $\mathbf{T}_1^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$, $\mathbf{T}_2^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$, and $\mathbf{T}_3^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$ which represent the loading matrices of the CPD of the core tensor of the HOSVD from (3.8). Thus, the tensor $\boldsymbol{\mathcal{S}}$ can be represented as

$$\boldsymbol{\mathcal{S}}^{(t)} = \boldsymbol{\mathcal{I}}_{3,L_d(k)}^{(t)} \times_1 \mathbf{T}_1^{(t)} \times_2 \mathbf{T}_2^{(t)} \times_3 \mathbf{T}_3^{(t)}. \tag{3.16}$$

The transform matrices $\mathbf{T}_1^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$, $\mathbf{T}_2^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$, and $\mathbf{T}_3^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k)}$ represent the loading matrices of the CPD of the core tensor $\boldsymbol{\mathcal{S}}^{(t)} \in \mathbb{C}^{L_d(k) \times L_d(k) \times L_d(k)}$ in (3.16). In theory, the CPD can be directly applied to $\boldsymbol{\mathcal{Y}}$ in order to extract the factor matrices. However, as demonstrated in [4], by directly computing the factor matrices using Alternating Least Squares (ALS), there are convergence problems, resulting into poor performance in terms of time-delay estimation. Therefore, to perform the CPD, it is sufficient to compute the loading matrices $\mathbf{T}_1^{(t)}$, $\mathbf{T}_2^{(t)}$, and $\mathbf{T}_3^{(t)}$, to obtain the factor matrices

$$\mathbf{U}_1^{(t)}\mathbf{T}_1^{(t)} = {\mathbf{\Gamma}^{(\mathbf{t})}}^{\mathrm{T}}, \tag{3.17}$$

$$\mathbf{U}_2^{(t)}\mathbf{T}_2^{(t)} = \left(\mathbf{C}\mathbf{Q}_{\omega}^{(d)(t)}\right)^{\mathrm{T}}, \tag{3.18}$$

$$\mathbf{U}_3^{(t)}\mathbf{T}_3^{(t)} = \mathbf{A}^{(t)}. \tag{3.19}$$

As a consequence of the symmetry of the SECSI problem, we can build six Simultaneous Matrix Diagonalization (SMD) problems for a three-way model [26]. However, as shown by [15], without loss of generality we can only use one mode of the compressed core tensor $\boldsymbol{\mathcal{S}}^{(t)}$ to compute the right-hand and left-hand matrices utilized in the SMD step described in [26]. In [15] the authors shows that the right-hand matrix of the third-mode a given compressed core tensor yields the best estimation in static scenarios. However, after performing several numerical simulations, we select the left-hand matrix of the first mode of the compressed core tensor $\boldsymbol{\mathcal{S}}^{(t)}$. Therefore, the $i$th slice of the first-mode of tensor $\boldsymbol{\mathcal{S}}^{(t)}$ is selected to compute the left-hand matrix

$$\begin{aligned}\mathbf{S}_{1,i}^{(t)} &= \left[\left(\boldsymbol{\mathcal{S}}^{(t)} \times_1 \mathbf{U}_1^{(t)}\right) \times_1 \mathbf{e}_i^{\mathrm{T}}\right] \\ &= \mathbf{T}_2^{(t)} \operatorname{diag}\{{\boldsymbol{\Gamma}^{(\mathbf{t})}}^{\mathrm{H}}_{\cdot,i}\}{\mathbf{T}_3^{(t)}}^{\mathrm{T}},\end{aligned} \tag{3.20}$$

where $\mathbf{e}_i^{\mathrm{T}}$ is a vector with all zeros except in the $i$th position. Next, we select the slice of tensor $\boldsymbol{\mathcal{S}}^{(t)}$ with the smallest condition number

$$\mathbf{S}_{1,p}^{(t)} = \mathbf{T}_2^{(t)} \operatorname{diag}\{{\boldsymbol{\Gamma}^{(t)}}^{\mathrm{H}}_{\cdot,p}\}(\mathbf{T}_3^{(t)})^{\mathrm{T}}, \tag{3.21}$$

where $p$ is an arbitrary index between one and the total number of slices to be diagonalized and defines the slice of tensor $\boldsymbol{\mathcal{S}}^t$ with the smallest condition number

$$p = \arg\min_i \operatorname{cond}\{\mathbf{S}_{1,i}^{(t)}\}, \tag{3.22}$$

where $\operatorname{cond}\{\cdot\}$ computes the condition number of a matrix. The smaller the condition number of a matrix, the more stable is its inversion. Furthermore, we obtain the left-hand matrices $\mathbf{S}_{1,i}^{(t),\text{lhs}}$ by multiplying $\mathrm{S}_{1,i}$ by $\mathrm{S}_{1,p}$ on the left-hand side

$$\begin{aligned}\mathbf{S}_{1,i}^{(t),\text{lhs}} &= \left({\mathbf{S}_{1,p}^{(t)}}^{-1}\mathbf{S}_{1,i}^{(t)}\right)^{\mathrm{T}} \\ &= \mathbf{T}_3^{(t)} \operatorname{diag}\{\boldsymbol{\Gamma}^{\mathrm{H}}_{\cdot,i}{\boldsymbol{\Gamma}^{(t)}}^{\mathrm{H}}\}_{\cdot,p}{\mathbf{T}_3^{(t)}}^{-1} \\ &= \mathbf{T}_3^{(t)}{\boldsymbol{\Gamma}^{(t)}}^{\mathrm{H}}{\mathbf{T}_3^{(t)}}^{-1}.\end{aligned} \tag{3.23}$$

Since $p$ is fixed, we can vary all possible values of $i$ in (3.23) obtaining $N-1$ equations, since $i \neq p$. Our goal is to find $\hat{\mathbf{T}^{(t)}}_3$ that simultaneously diagonalizes the $N-1$ equations. We refer here to the techniques in [37] and [38].

Since in the noiseless case, according to (2.9), the third-mode unfolding exposes the factor matrix $\mathbf{A}^{(t)}$, we can write

$$[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)}^{\mathrm{T}} = \left[\boldsymbol{\Gamma}^{(t)} \diamond \left(\mathbf{C}\mathbf{Q}_{\omega}^{(d)}\right)^{(t)}\right]\mathbf{A}^{(t)}. \tag{3.24}$$

Using $\mathbf{U}_3^{(t)}$ from (3.8) and $\hat{\mathbf{T}}_3^{(t)}$ from the diagonalization step we obtain

$$\mathbf{U}_3^{(t)}\hat{\mathbf{T}}_3^{(t)} = \hat{\mathbf{A}}^{(t)}. \tag{3.25}$$

Afterwards, we multiply (3.24) by the pseudo-inverse of the estimated $\hat{\mathbf{A}}^{(t)}$ from the left-hand side to obtain

$$\begin{aligned}\mathbf{F}^{(t)} = [\boldsymbol{\mathcal{Y}}^{(t)}]^{\mathrm{T}}_{(3)}(\hat{\mathbf{A}}^{(t)})^{+\mathrm{T}} &= \left[\boldsymbol{\Gamma}^{(t)} \diamond \left(\mathbf{C}\mathbf{Q}^{(d)}_{\omega}\right)^{(t)}\right]\mathbf{A}^{(t)}(\hat{\mathbf{A}}^{(t)})^{+\mathrm{T}} \\ &\approx \left[\boldsymbol{\Gamma}^{(t)} \diamond \left(\mathbf{C}\mathbf{Q}^{(d)}_{\omega}\right)^{(t)}\right] \in \mathbb{C}^{KM \times L_d(k)}.\end{aligned} \tag{3.26}$$

Then, factor matrices $(\mathbf{C}\mathbf{Q}^{(d)^{(t)}}_{\omega})^{\mathrm{T}}$ and $\boldsymbol{\Gamma}^{(t)}$ can be estimated from (3.26) by applying the Least Squares Khatri-Rao Factorization (LSKRF) [35].

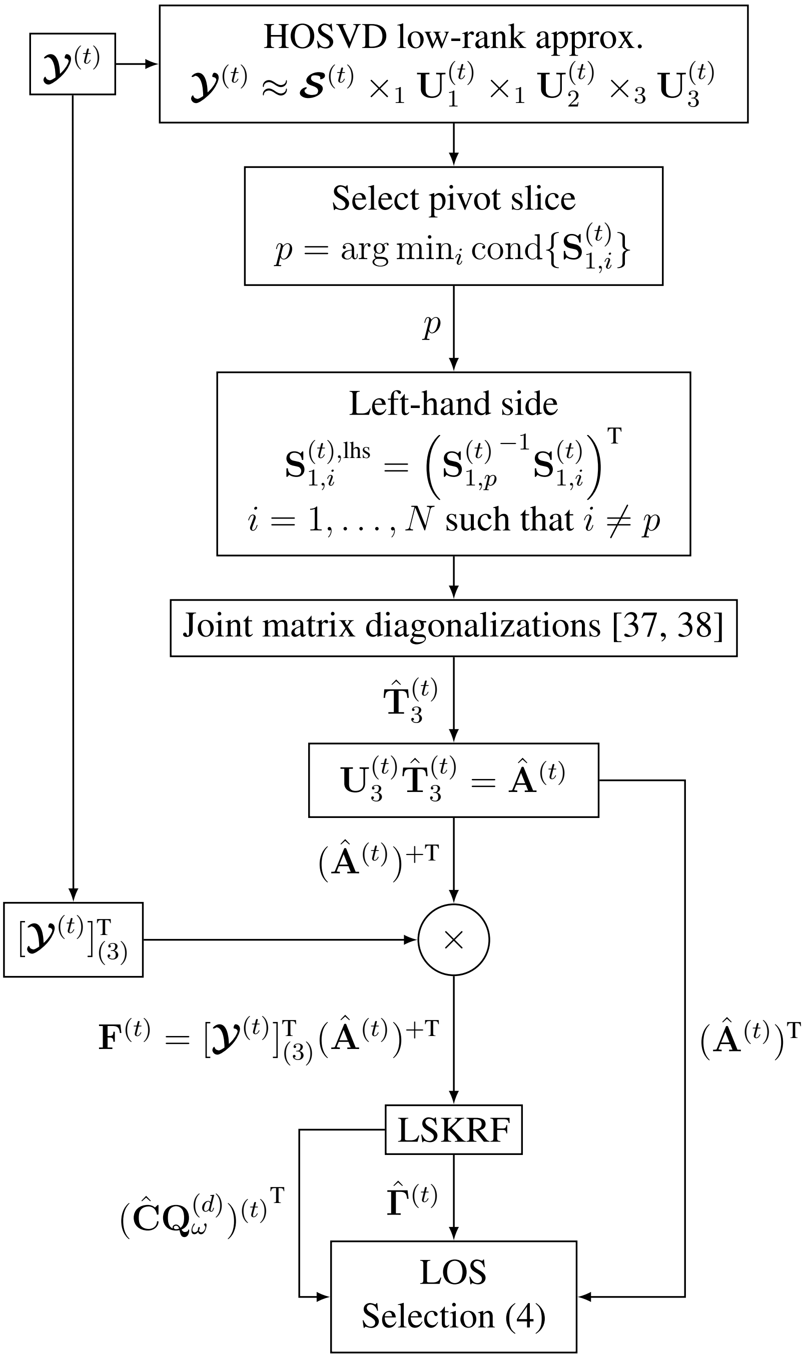


Figure 3.50: Mode 1 HOSVD SECSI Time-Delay Estimation block diagram

## 3.3 Simulation Results for Time-Delay Estimation Simulations for Dynamic Scenario

In this section we present the TDE computed during simulations when considering a dynamic scenario and a perfectly aligned array of antennas. Since in the dynamic scenario we are varying the number of LOS and NLOS components within the tensor $\mathcal{Y}$, when utilizing matrix-based MOS methods we grouped the epochs with the same estimated model order.

Moreover, similarly to computing the PoD, we utilize the maximum and minimum estimated model order, obtained from the matrix-based MOS methods, to perform matrix separation and compute the LOS time-delay. Since the matrix-based AIC, ESTER and RADOI methods showed poor performance, we only utilized the matrix-based MDL+FBA+SPS, EFT+FBA+SPS and EFT+FBA+SPS methods. Additionally, we performed simulations considering we know the model order of each epoch. Thus, we could divide the main tensor into new tensors and them compute the TDE. Furthermore, we utilize the tensor-based R-D EFT MOS method to simultaneously compute the model order and provide an estimate to perform matrix-separation and time-delay estimation. We did not perform simulation using R-D AIC and R-D MDL since these methods extrapolate the estimated model order. Additionally, we present results for the Tensor-based Eigenfilter. Differently from the CPD-based methods, the Tensor-based Eigenfilter does not require any estimate of the model order to perform matrix separation. Therefore, the Tensor-based Eigenfilter might be a suitable alternative in a dynamic scenario. Moreover, the state-of-the-art CPD-GEVD and the state-of-the-art HOSVD SECSI are not suitable to be combined with the approach where we create sub-tensors to perform the TDE. Since both methods utilize the dimension of epochs, e.g. dimension $K$, of the tensors to perform the factor matrix estimation and in some scenarios $\hat{L}_d > K$ the matrix decomposition becomes impossible. Finally, we present simulation results for scenarios with different epochs, e.g. $K = 8$, $K = 15$, and $K = 30$ epochs

In Figure 3.51 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the minimum estimated model order considering we have gathered $K = 8$ epochs. Note that the state-of-the-art CPD-GEVD method has improved performance when we select the minimum estimated model order. Furthermore, this method outperforms the Tensor-based Eigenfilter and Known Model Order methods. Therefore, when utilizing the minimum model order, we no longer need to split the tensor into various sub-tensors thus we can jointly estimate the time-delay. Moreover, by underestimating the model order, we assume that the second NLOS component is considered a noise component. Then, since the second NLOS component is extremely weak when compared to LOS signals, it has a low impact in the time-delay measurement.

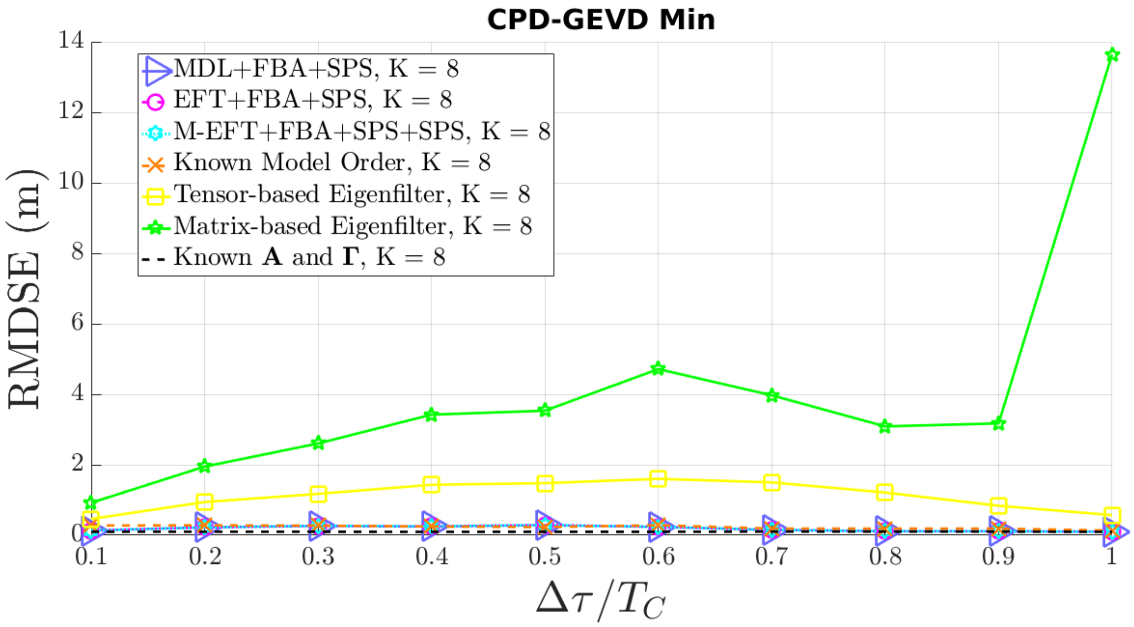


Figure 3.51: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.52 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the minimum estimated model order with $K = 15$. Note that when gathering more epochs, the time-delay estimation error slightly increased when compared to Figure 3.51. Therefore, since we increase the amount of epochs, we also increased the number of slices with different estimated model orders. Then, the state-of-the-art CPD-GEVD presents higher error rate.

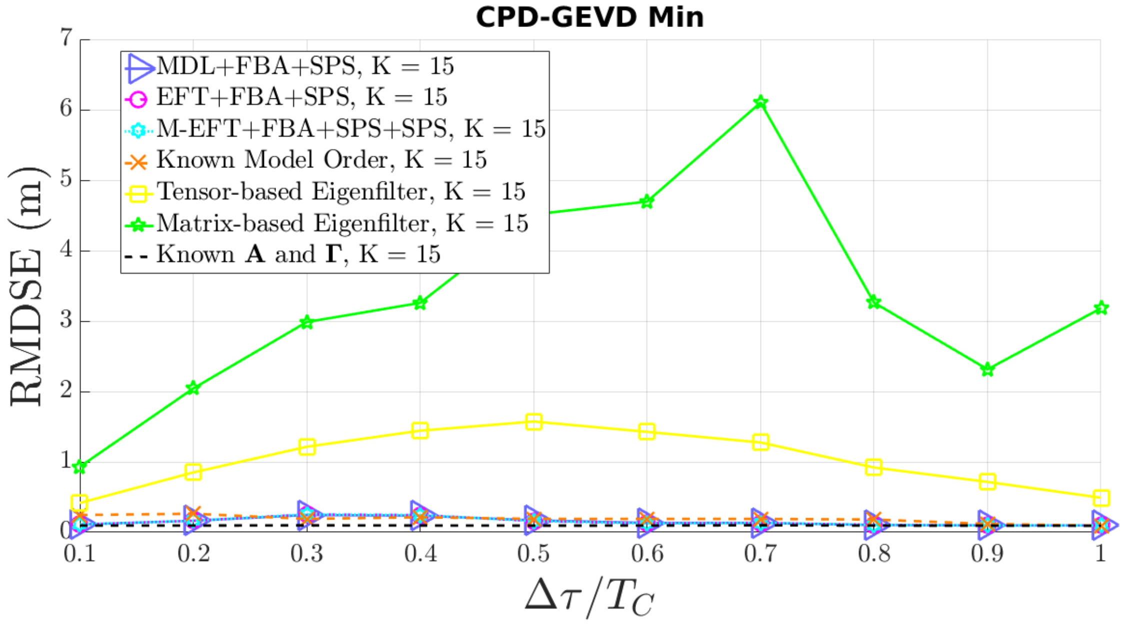


Figure 3.52: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.52 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the minimum estimated model order with $K = 30$. Note that with $K = 30$, the time-delay estimation error slightly increased when compared to Figure 3.51. However, the time-delay estimation error stayed similar to Figure 3.53. Then, we show that the error rate kept approximately constante as we increase the number of epochs.

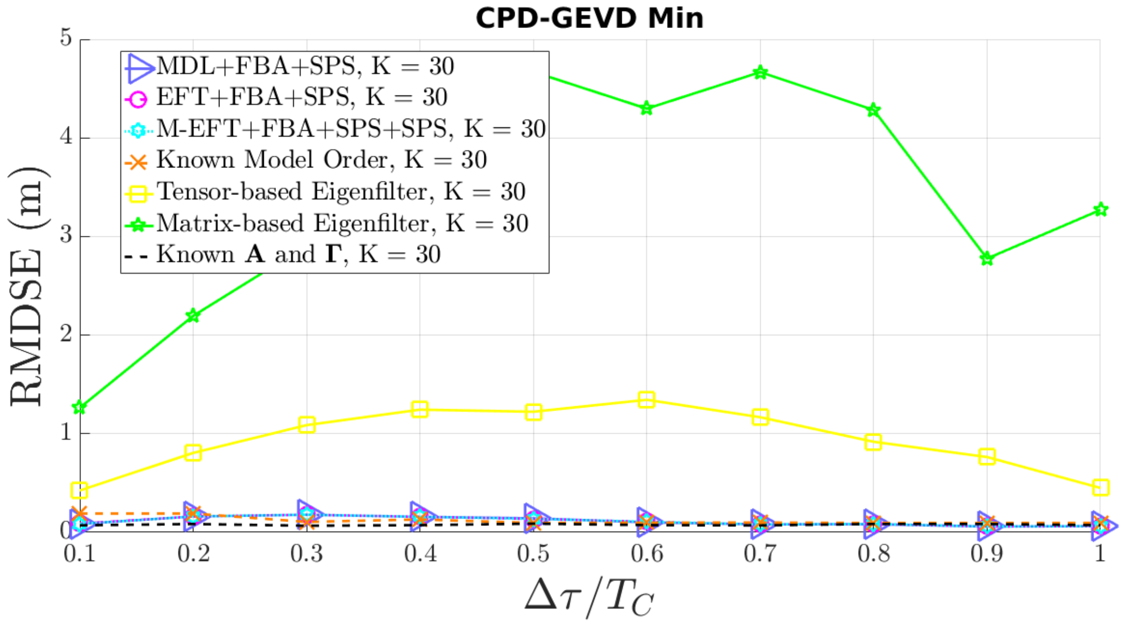


Figure 3.53: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.54 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the maximum estimated model order when $K = 8$. Note that the matrix-based MOS methods have improved performance when we select the maximum estimated model order. Furthermore, these methods outperform the Tensor-based Eigenfilter and Known Model Order methods. Therefore, when utilizing the maximum model order, we no longer need to split the tensor into various sub-tensors thus we can jointly estimate the time-delay. Moreover, selecting the maximum estimated model order does not necessarily mean we have a model order $\hat{L}_d = 3$ as shown in Figures 3.29, 3.36, 3.43. Therefore, by selecting the maximum estimated model order we may model noise components as signals. Additionally, we indicate that an underestimated model order, as in Figure 3.51, is a more suitable solution. Therefore, notice that the second NLOS effect on matrix-separation and TDE might be negligible when using the state-of-the-art CPD-GEVD.

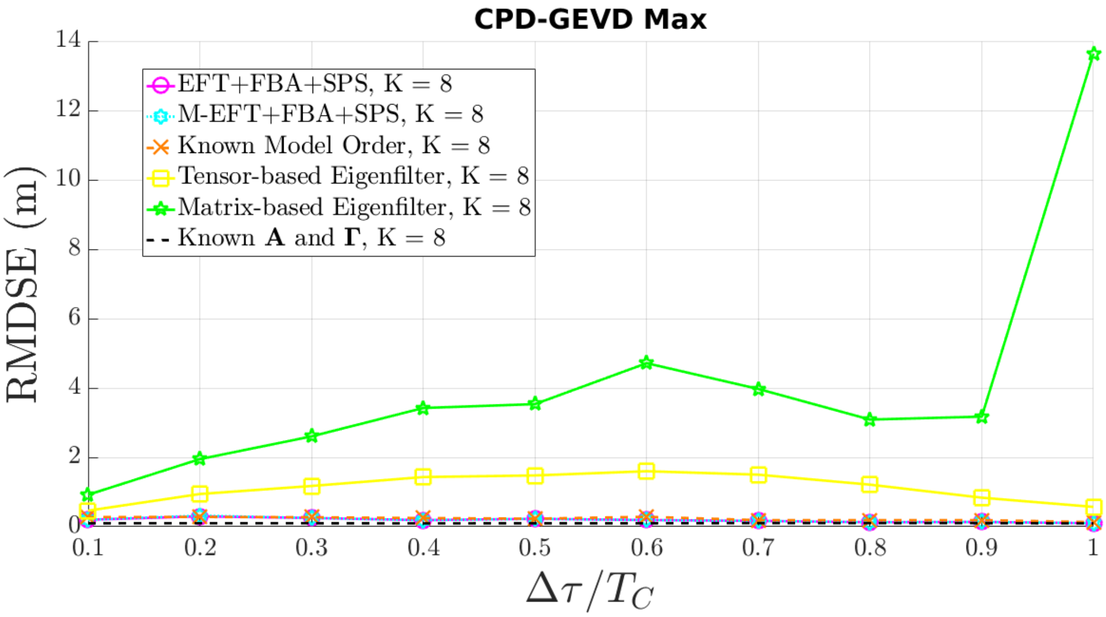


Figure 3.54: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.55 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the maximum estimated model order when $K = 15$. Note that the matrix-based MOS methods have improved performance when we select the maximum estimated model order. However, observe that when using the maximum estimated model order obtained from the MDL+FBA+SPS method we have a higher time-delay estimation error. The higher error is due to the lowest PoD presented by the MDL+FBA+SPS method when estimating the model order in slices with $L_d = 3$. Therefore, when utilizing the MDL+FBA+SPS method we have a higher amount of noise components being modeled as signal. Additionally, we reinforce that an underestimated model order, as in Figure 3.52, is a more suitable solution. Therefore, notice that the second NLOS effect on matrix-separation and TDE might be negligible when using the state-of-the-art CPD-GEVD.

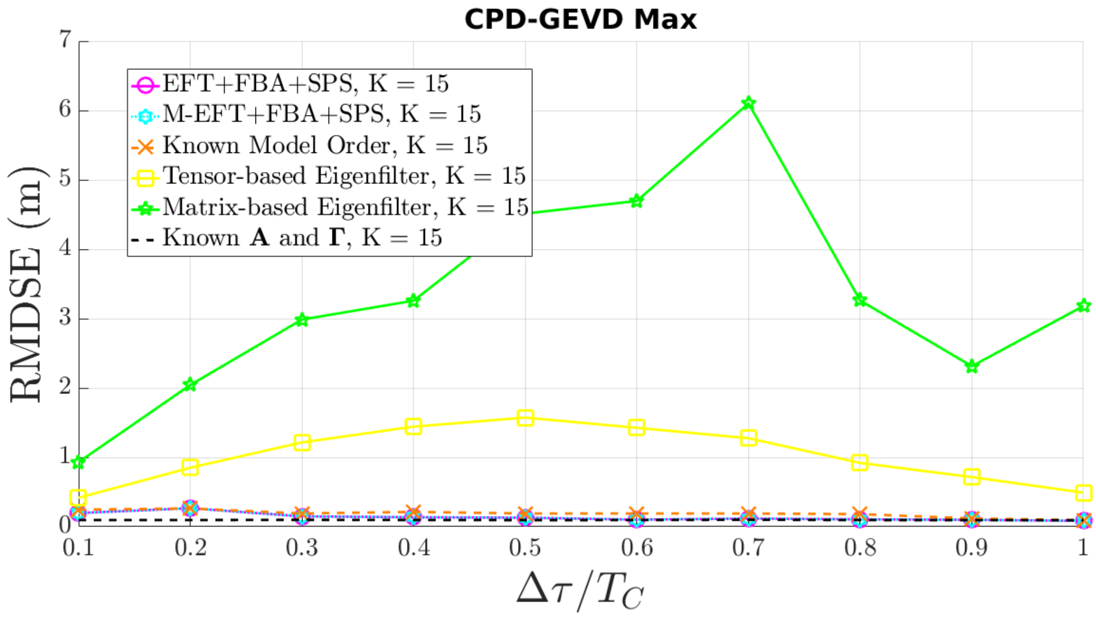


Figure 3.55: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.56 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the maximum estimated model order when $K = 30$. We reinforce the trend in higher error rate when utilizing the maximum estimated model order obtained from the MDL+FBA+SPS method. Moreover, we show that when we have $K = 30$ epochs the maximum estimated model order obtained from EFT+FBA+SPS and M-EFT+FBA+SPS have similar performance to the minimum estimated model order obtained from the same methods.

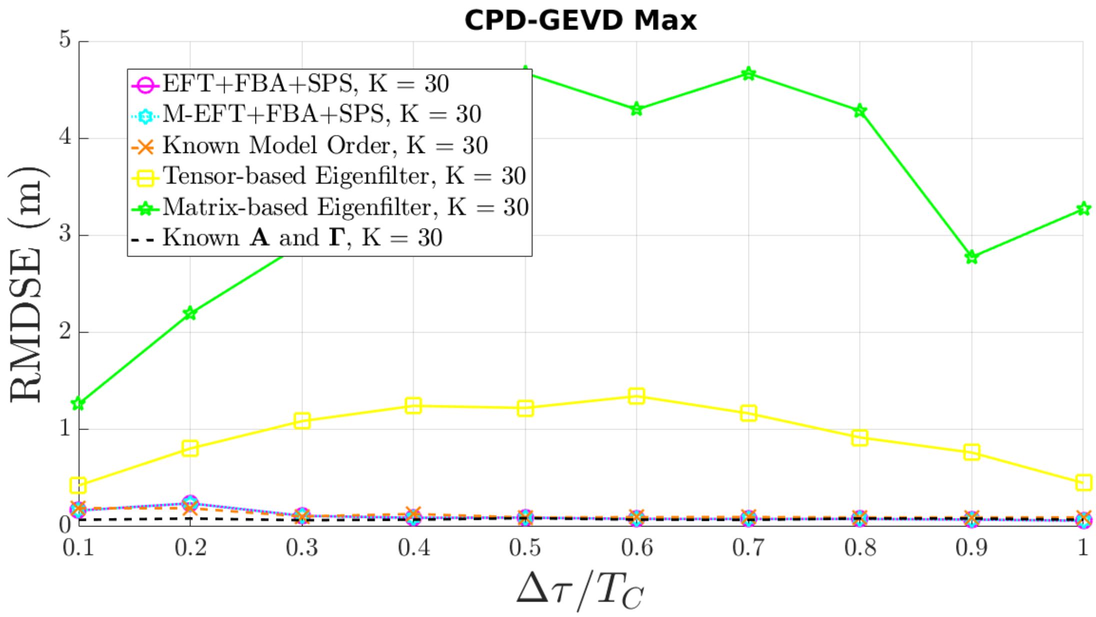


Figure 3.56: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Since the state-of-the-art CPD-GEVD method is not suitable to the sub-tensor approach, we exploited the possible tensor rotations that could be utilized to perform matrix factorization. Therefore, we utilize the first mode CPD-GEVD method. Opposingly to the state-of-the-art CPD-GEVD, the first mode CPD-GEVD rotates the tensor $\boldsymbol{\mathcal{Y}}$ and places the antenna dimension, e.g. tensor $\boldsymbol{\mathcal{Y}}$ third dimension, as the first dimension. Thus, the first mode CPD-GEVD method utilizes the antenna dimension to perform matrix factorization. In Figure 3.57 we performed simulation combining the sub-tensor approach with the first mode CPD-GEVD method. Hence, we show that this CPD-GEVD variant is suitable to perform matrix factorization and TDE when utilizing the sub-tensor approach. Note that, the first mode CPD-GEVD combined with sub-tensors has similar perfomance to the state-of-the-art CPD-GEVD combined with minimum estimated model order, Figure 3.51.

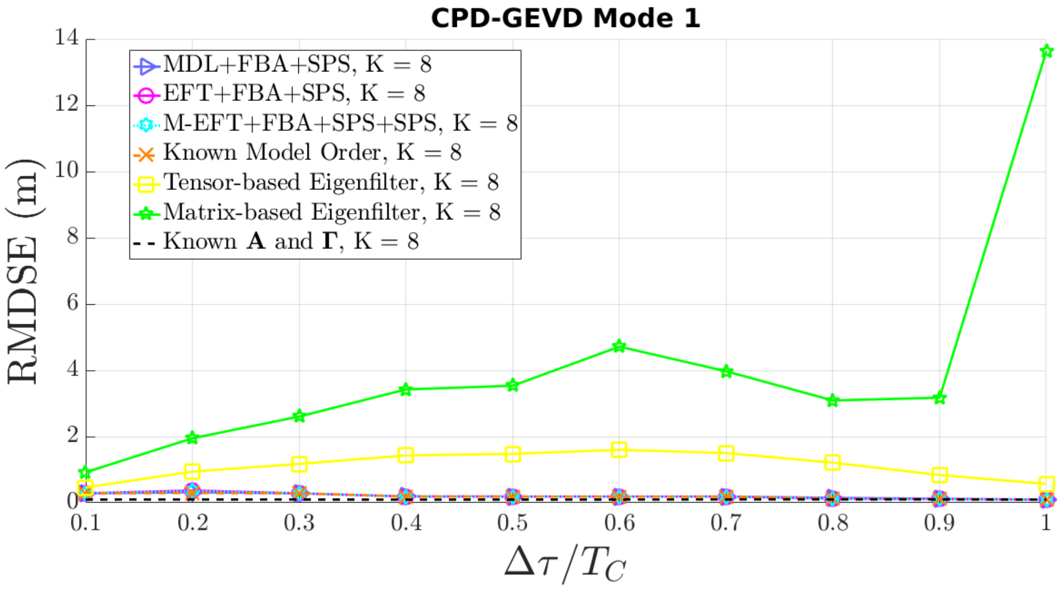


Figure 3.57: MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.58 we performed simulation combining the sub-tensor approach with the first mode CPD-GEVD method. Hence, we show that this CPD-GEVD variant is suitable to perform matrix factorization and TDE when utilizing the sub-tensor approach. Note that, the first mode CPD-GEVD combined with sub-tensors has similar perfomance to the state-of-the-art CPD-GEVD combined with minimum estimated model order, Figure 3.52.

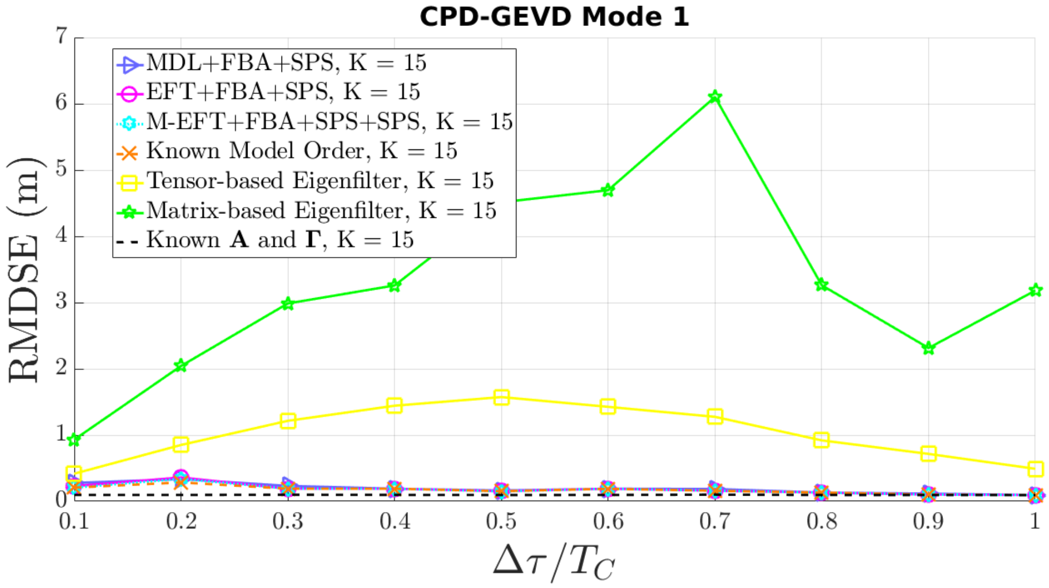


Figure 3.58: MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.59 we performed simulation combining the sub-tensor approach with the first mode CPD-GEVD method. Hence, we show that this CPD-GEVD variant is suitable to perform matrix factorization and TDE when utilizing the sub-tensor approach. Note that, the first mode CPD-GEVD combined with sub-tensors has similar perfomance to the state-of-the-art CPD-GEVD combined with minimum estimated model order, Figure 3.53.

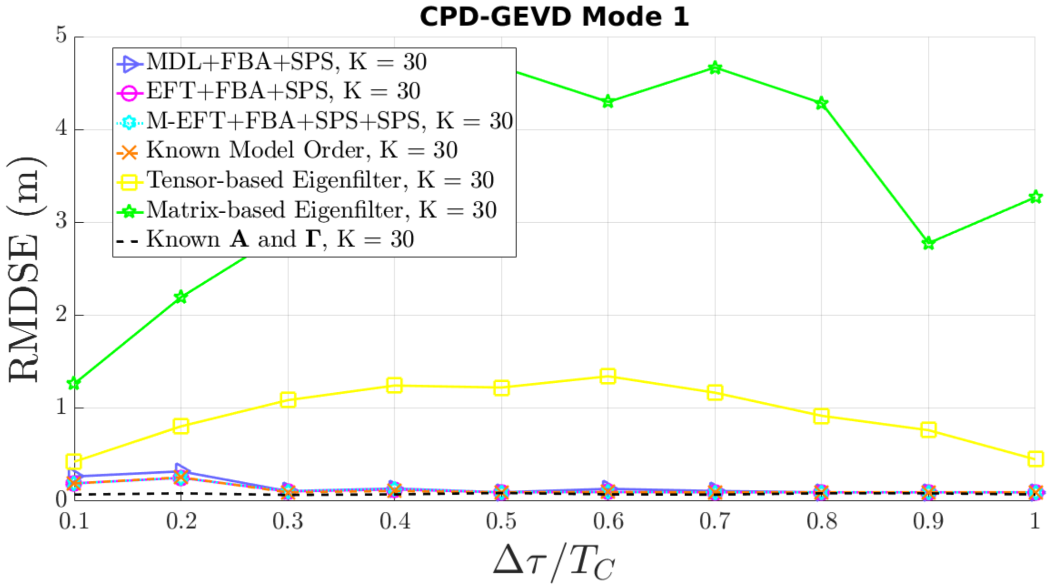


Figure 3.59: MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Additionally to performing simulations with first mode CPD-GEVD method combined with the sub-tensors approach. Therefore, in in Figure 3.60 we show the results of simulations combining the TDE method with the minimum estimated model order. Note that even though the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.57, the first mode CPD-GEVD method combined with the sub-tensors presents a better performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The improved performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Therefore, we can accuratly construct sub-tensors that can better approximate an static scenario. Consequently, we have a lower time-delay error.

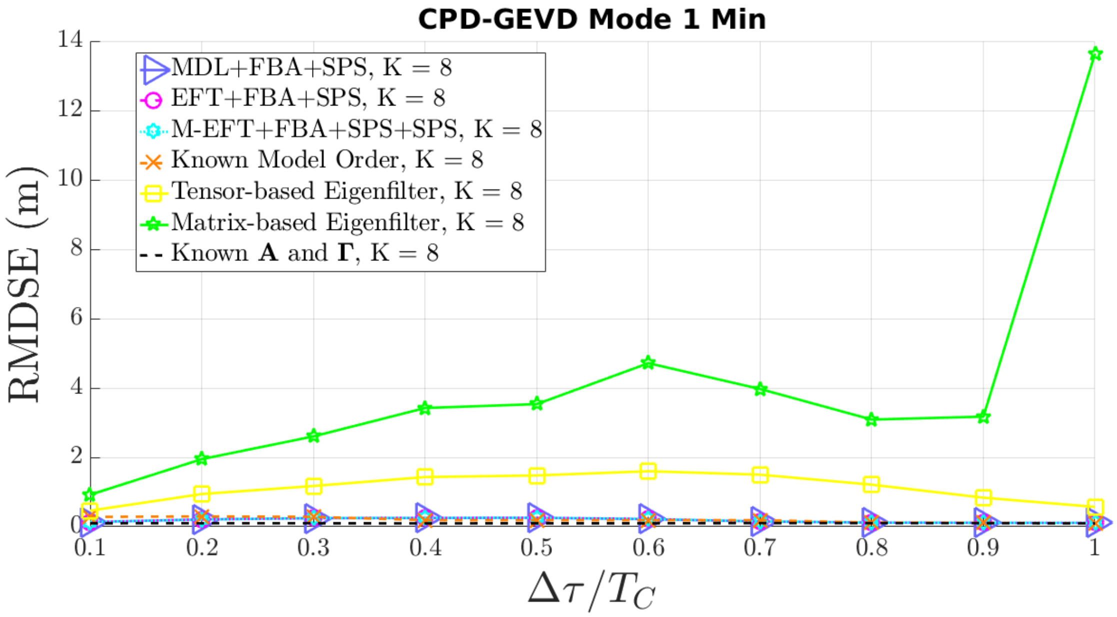


Figure 3.60: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

Moreover, in Figure 3.61 we show the results of simulations combining the TDE method with the minimum estimated model order when $K = 15$. Observe that the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.58. However, the first mode CPD-GEVD method combined with the sub-tensors presented a better performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The improved performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Thus, we can accuratly construct sub-tensors that can better approximate an static scenario. Therefore, we have a lower time-delay error.

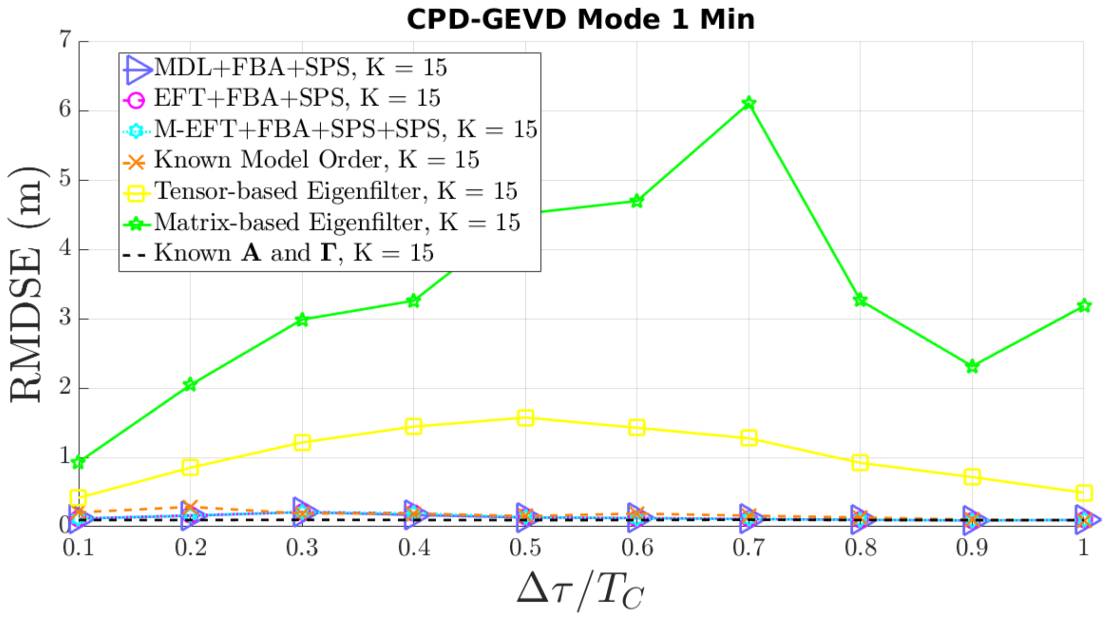


Figure 3.61: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.62 we show the results of simulations combining the TDE method with the minimum estimated model order when $K = 30$. Observe that the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.59. However, the first mode CPD-GEVD method combined with the sub-tensors presented a better performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The improved performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Thus, we can accuratly construct sub-tensors that can better approximate an static scenario. Therefore, we have a lower time-delay error.

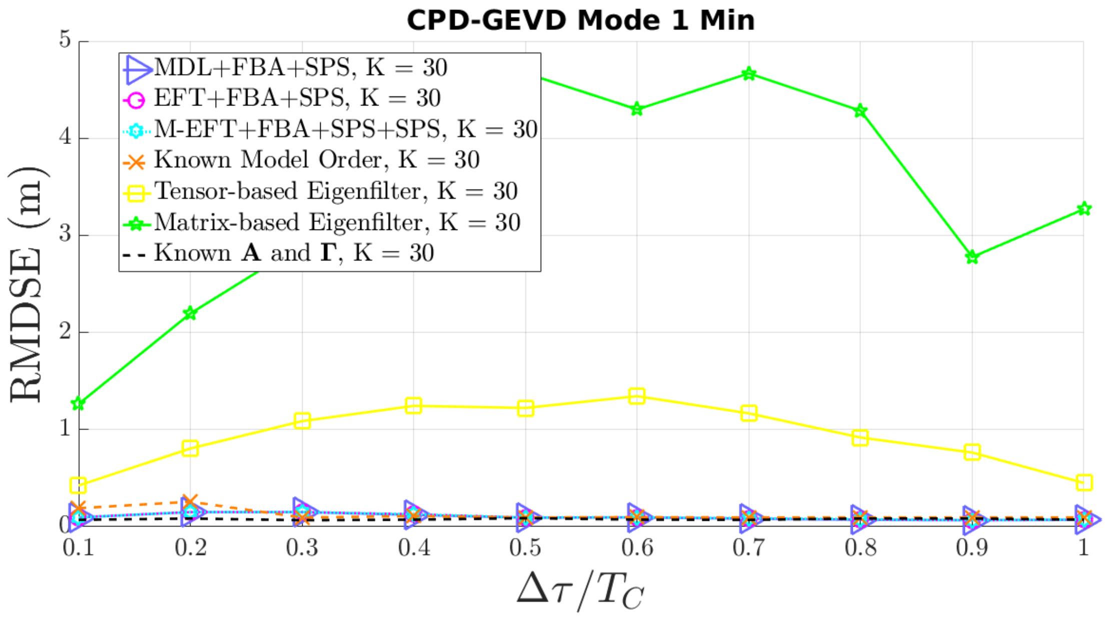


Figure 3.62: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

As previously described, combining matrix factorization with the minimun model order estimation may be a suitable option to successfully perform TDE in dynamic scenarios. Thus, in addition to performing simulations with first mode CPD-GEVD method combined with the sub-tensors and minimum estimated model order approach, in Figure 3.63 we show the results of simulations combining the TDE method with the maximum estimated model order. Note that even though the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.57, the first mode CPD-GEVD method combined with the sub-tensors presents a similar performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The comparable performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Therefore, we can accuratly construct sub-tensors that can better approximate an static scenario. Consequently, we have a lower time-delay error.

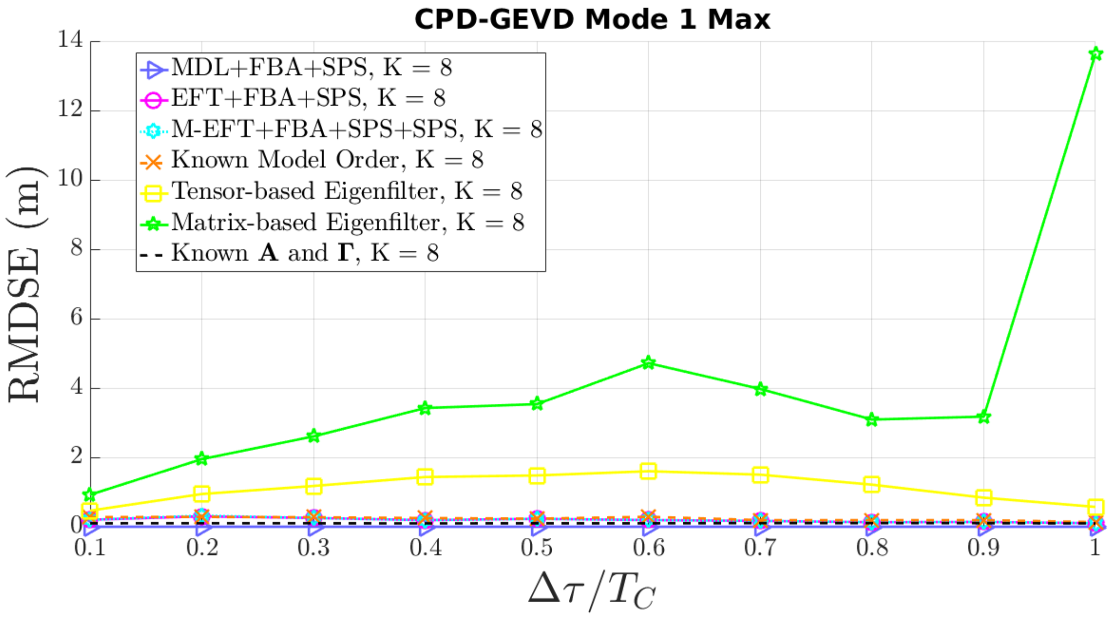


Figure 3.63: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.64 we show the results of simulations combining the TDE method with the maximum estimated model order. Note that the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.58. However, the first mode CPD-GEVD method combined with the sub-tensors presents a similar performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The comparable performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Therefore, we can accuratly construct sub-tensors that can better approximate an static scenario. Consequently, we have a lower time-delay error.

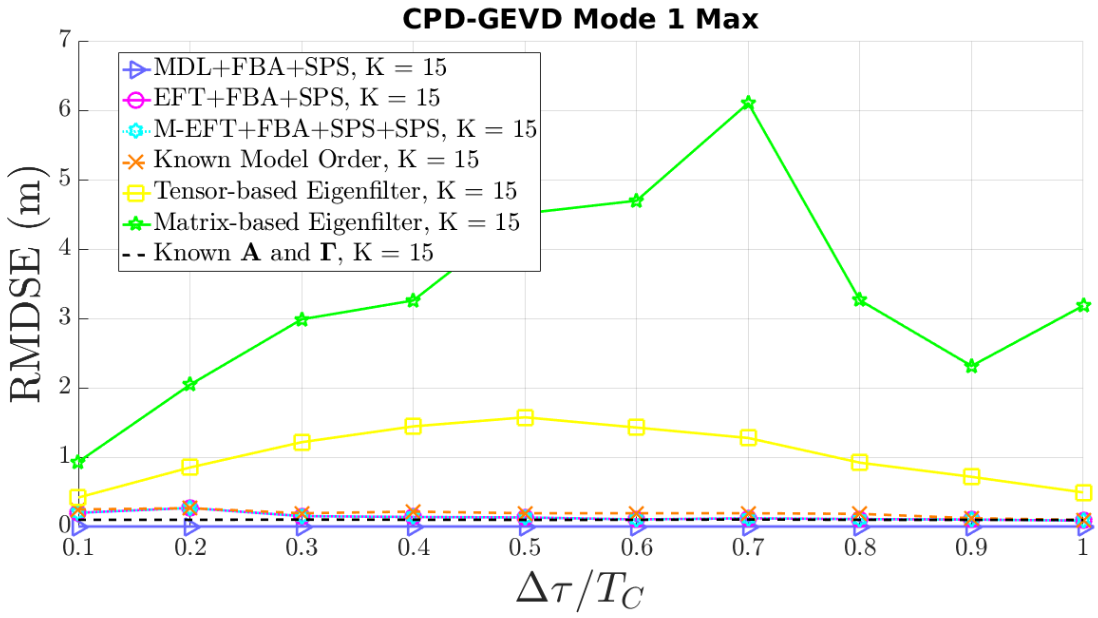


Figure 3.64: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.65 we show the results of simulations combining the TDE method with the maximum estimated model order. Note that the first mode CPD-GEVD method is slightly better when signals are strongly correlated, e.g. $\Delta\tau < 0.3$ than the method shown in Figure 3.58. However, the first mode CPD-GEVD method combined with the sub-tensors presents a similar performance when signals are weakly correlated, e.g. $\Delta\tau \geq 0.3$. The comparable performance is due to the more accurate MOS performed by the MOS methods when $\Delta\tau \geq 0.3$. Therefore, we can accuratly construct sub-tensors that can better approximate an static scenario. Consequently, we have a lower time-delay error.

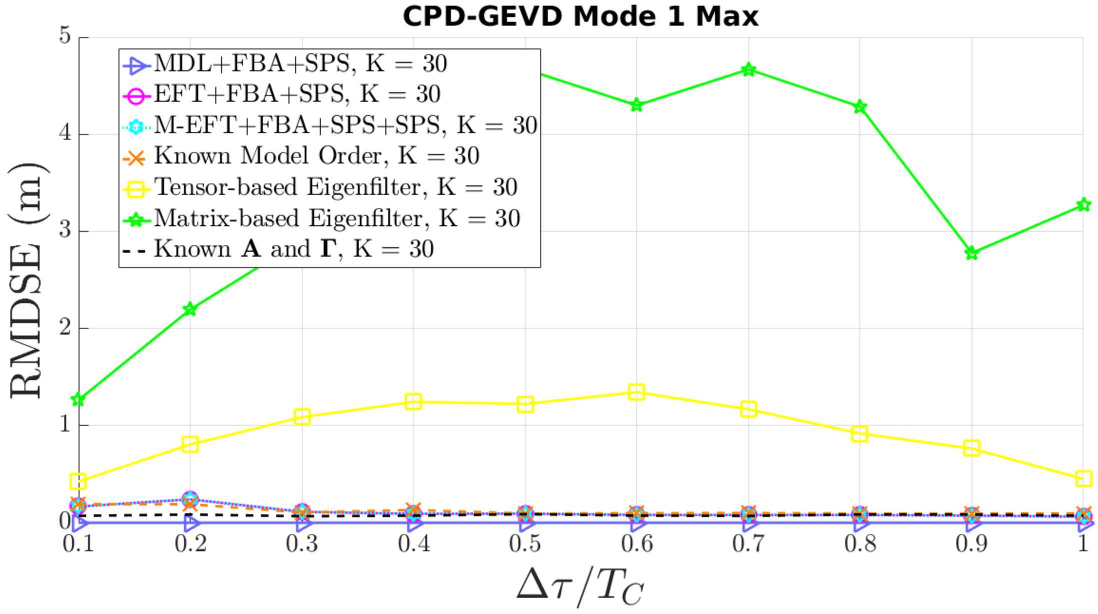


Figure 3.65: MOS techniques and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Similarly to the state-of-the-art CPD-GEVD method, the HOSVD SECSI method utilizes the dimension of epochs to perform matrix factorization therefore this method cannot be combined with the sub-tensor approach. Since, the sub-tensor approach may create sub-tensors with $K < \hat{L}_d^{(k)}$ therefore the state-of-the-art HOSVD SECSI method may not to be able to perform factor matrix estimation. Consequently, in Figure 3.66 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the minimum estimated model order with $K = 8$. Note that even with an underestimated model order, the state-of-the-art HOSVD SECSI method shows improved performance when compared to the CPD-GEVD methods. Furthermore, observe that this TDE method is reliable when signals are strongly correlated, e.g. $\Delta\tau < 0.3$

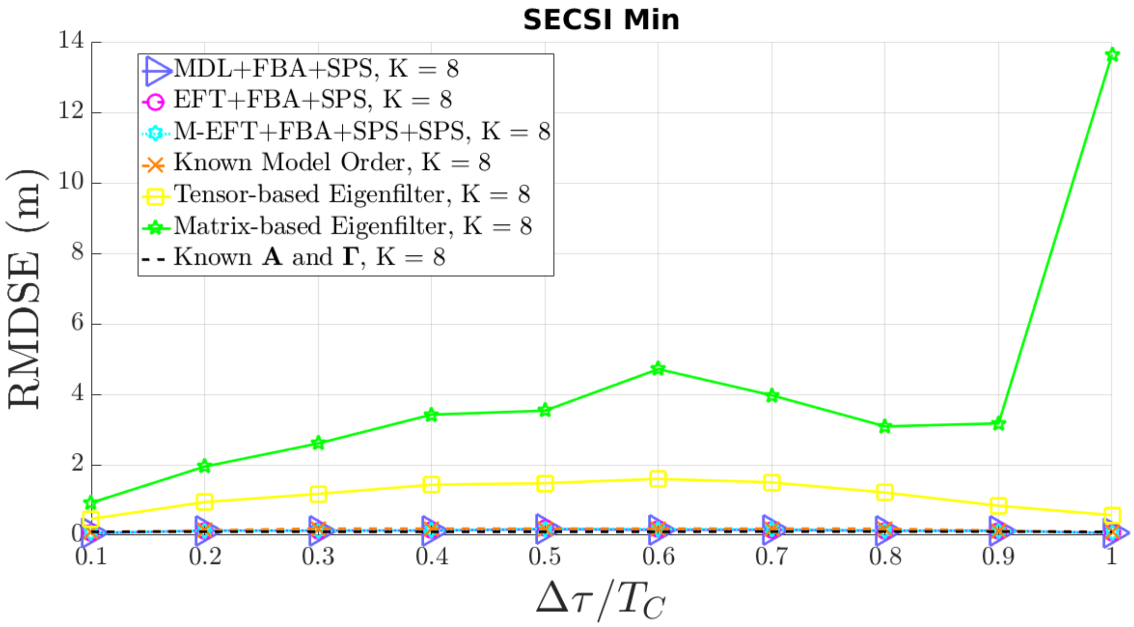


Figure 3.66: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.67 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the minimum estimated model order with $K = 15$. Again, the state-of-the-art HOSVD SECSI method provides a better TDE than the previously described CPD-GEVD methods. However, note that by adding more epochs we decreased the state-of-the-art HOSVD SECSI method. The worsen performance is due to modeling the NLOS compents as noise, therefore, presenting a higher time-delay error.

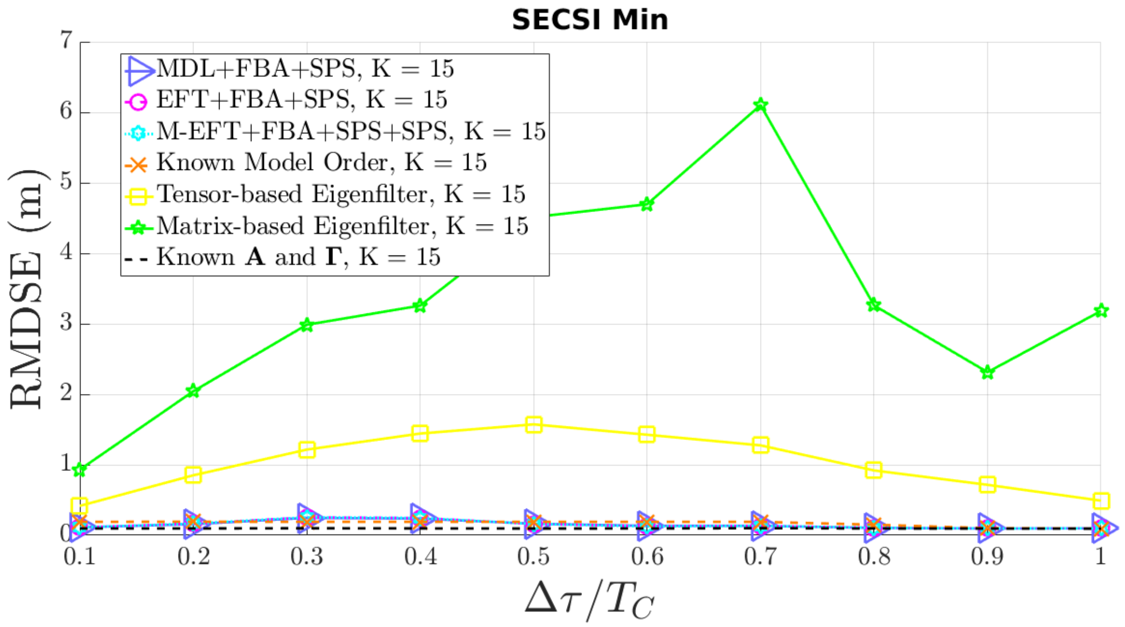


Figure 3.67: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.68 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the minimum estimated model order with $K = 30$. Observe that the state-of-the-art HOSVD SECSI method provides a better TDE than the previously described CPD-GEVD methods. However, note that by adding more epochs we decreased the state-of-the-art HOSVD SECSI method. The worsen performance is due to modeling the NLOS componts as noise, therefore, presenting a higher time-delay error.

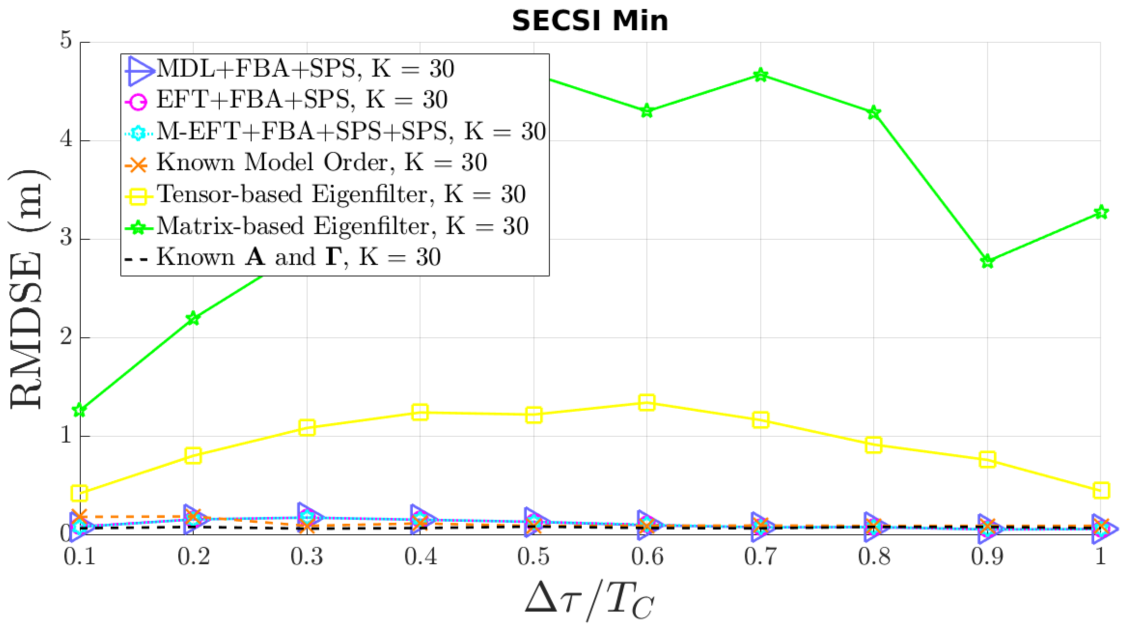


Figure 3.68: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In addition to the minimum estimated model order, we performed simulations combining the state-of-the-art HOSVD SECSI combined with the maximum estimated model order. In Figure 3.69 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the maximum estimated model order with $K = 8$. Note that the maximum estimated model order presents a similar performance to the minimum estimated model order. This is possible since the NLOS components have a much lower SNR than the LOS component.

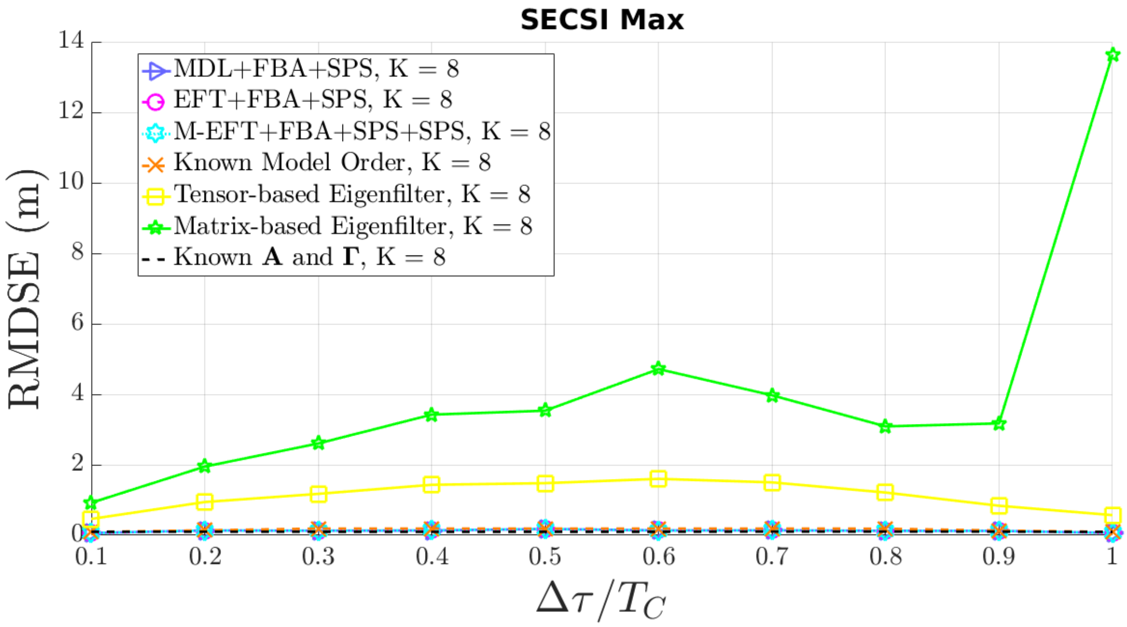


Figure 3.69: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.70 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the maximum estimated model order with $K = 15$. Similarly to the minimum estimated model order, the maximum estimated model order presents worsen performance after adding more epochs to the main tensor. This is due to modeling noise componts as signals components by overestimating the model order. However, observe that the maximum estimated model order presents lower time-delay error when compared to Figure 3.67.

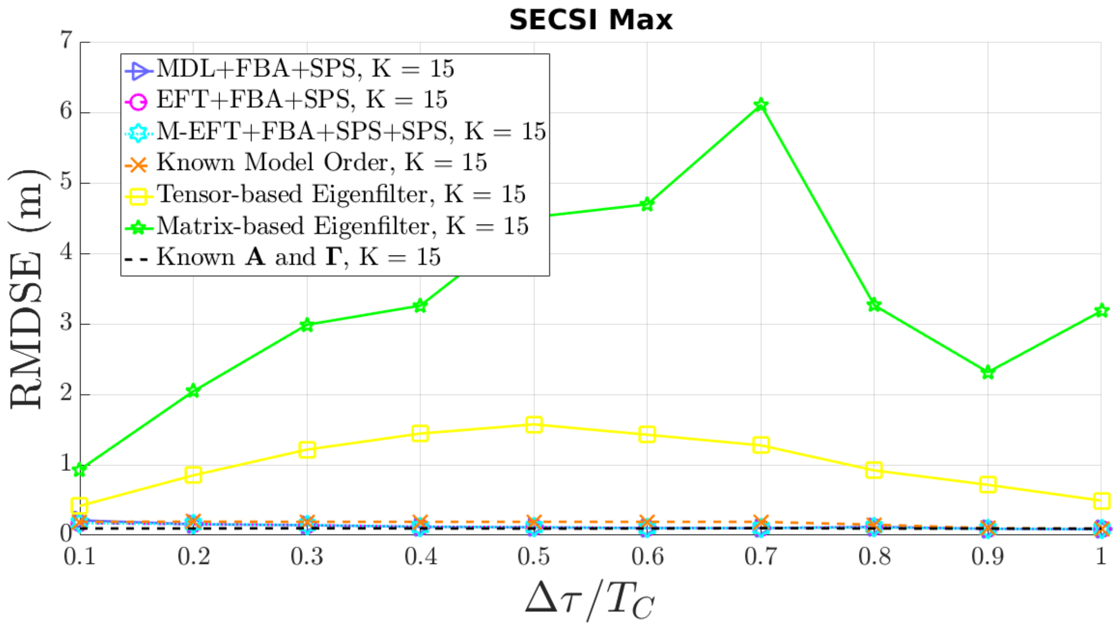


Figure 3.70: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.70 we show the simulation results for the state-of-the-art HOSVD SECSI method utilizing the maximum estimated model order with $K = 30$. Again, we have a worsen performance after adding more epochs to the main tensor. This is due to modeling noise componts as signals components by overestimating the model order. However, observe that the maximum estimated model order presents lower time-delay error when compared to Figure 3.67.

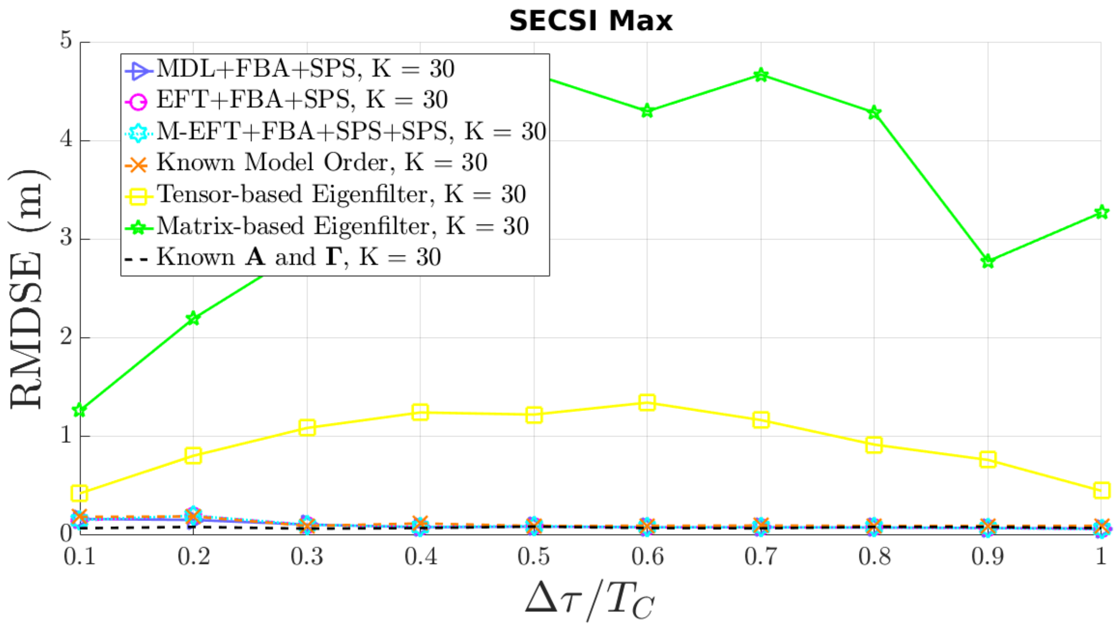


Figure 3.71: MOS techniques and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

Since the state-of-the-art HOSVD SECSI method is not suitable to the sub-tensor approach, we exploit the tensor dimensions that could be utilized to perform matrix factorization. Thus, we utilize the first mode Mode 1 HOSVD SECSI with left hand-hand matrix method. In contrast to the state-of-the-art HOSVD SECSI, the Mode 1 HOSVD SECSI with left hand-hand matrix method utilizes the antenna dimension, e.g. tensor $\boldsymbol{\mathcal{Y}}$ third dimension, to perform matrix factorization. In Figure 3.72 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method with $K = 8$. As previously described, we utilize the matrix-based MOS methods to estimate the model order of each epoch then we group the epochs with same model order and create sub-tensors. Therefore, since the sub-tensor approach is a solution that aims to create static scenarios, we show that the HOSVD SECSI variant is suitable to perform matrix factorization and TDE when utilizing the sub-tensor approach. Note that the MDL+FBA+SPS has a better performance than the Tensor-based Eigenfilter, the M-EFT+FBA+SPS and EFT-FBA+SPS show similar performance to the Known Model Order.

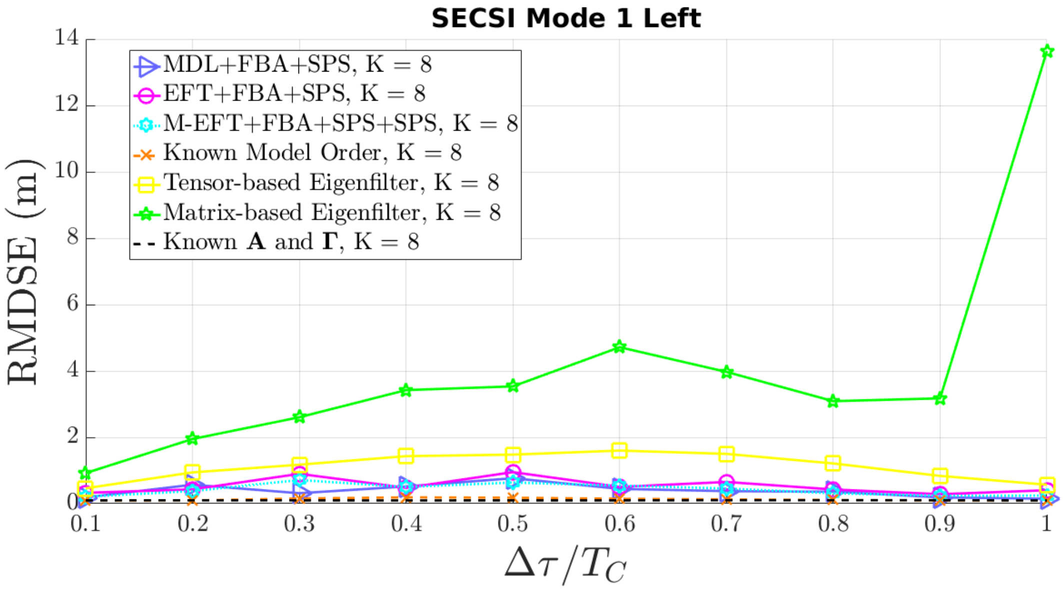


Figure 3.72: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.73 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method with $K = 15$. Note that since we have more epochs, we can construct more accurate sub-tensors. Therefore, the Mode 1 HOSVD SECSI shows similar performance to the state-of-the-art HOSVD SECSI.

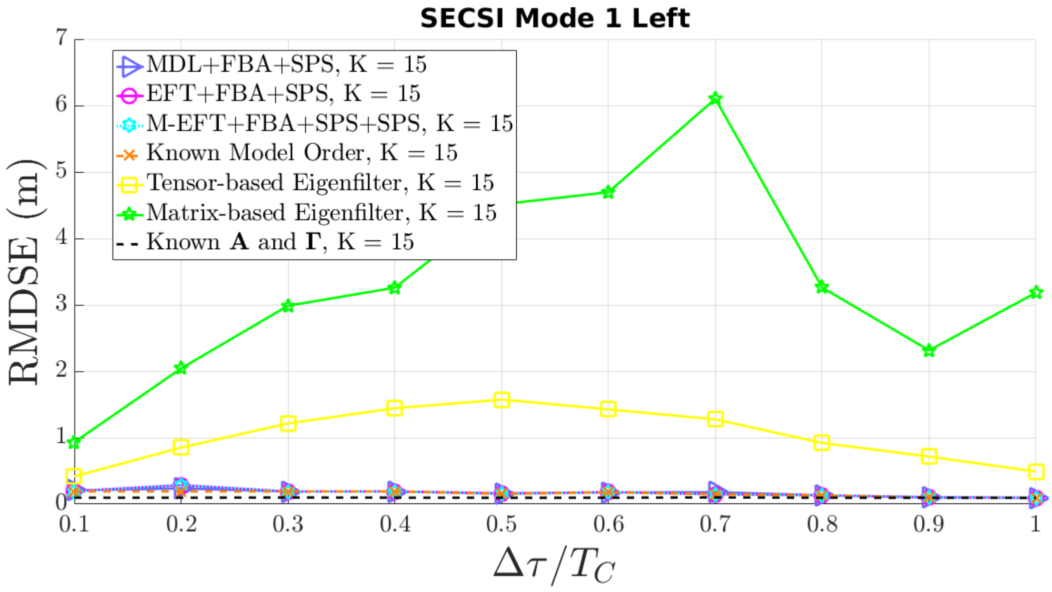


Figure 3.73: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.74 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method with $K = 30$. Observe that since we have more epochs, we can construct more accurate sub-tensors. Therefore, the Mode 1 HOSVD SECSI shows similar performance to the state-of-the-art HOSVD SECSI.

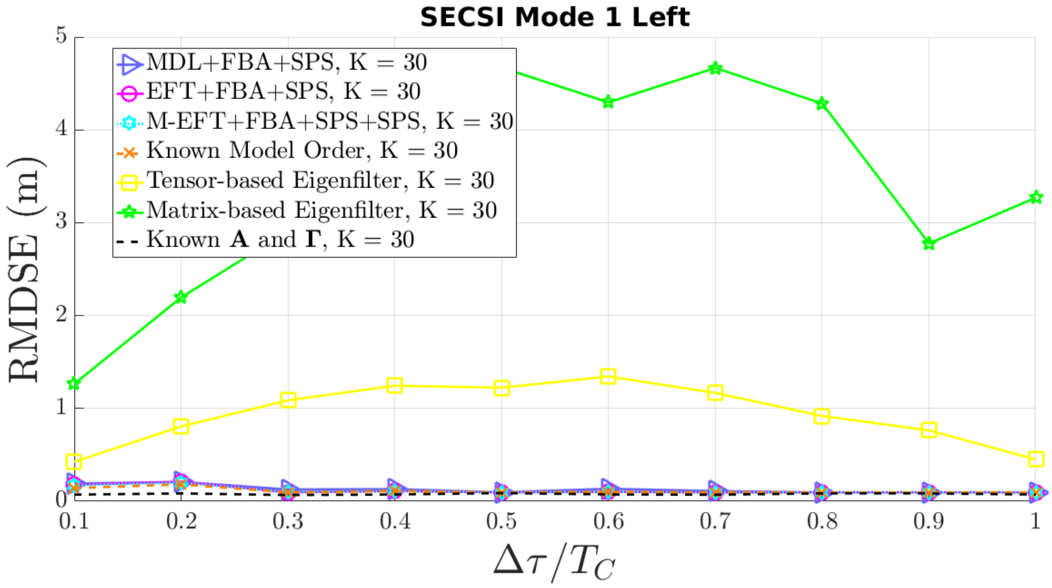


Figure 3.74: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.75 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the minimum estimated model order with $K = 8$. Note that the matrix-based MOS methods have improved performance when we select the minimum estimated model order. Furthermore, these methods outperform the Tensor-based Eigenfilter and Known Model Order methods. Again, when utilizing the minimum model order, we no longer need to split the tensor into various sub-tensors thus we can jointly estimate the time-delay. Moreover, by underestimating the model order, we assume that the second NLOS component is considered a noise component. Since the second NLOS component is extremely weak when compared to LOS signals, it has a low impact in the time-delay measurement.

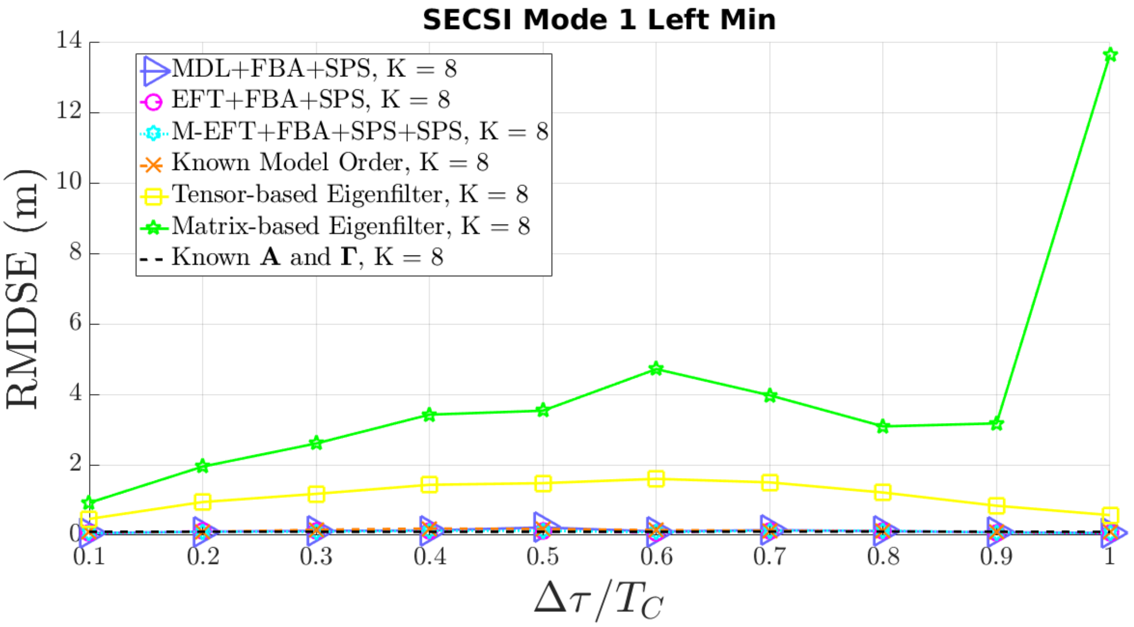


Figure 3.75: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.76 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the minimum estimated model order with $K = 15$. Observe that the increase in the amount of epochs did not affect the time-delay estimation since we obtain similar performance to Figure 3.75.

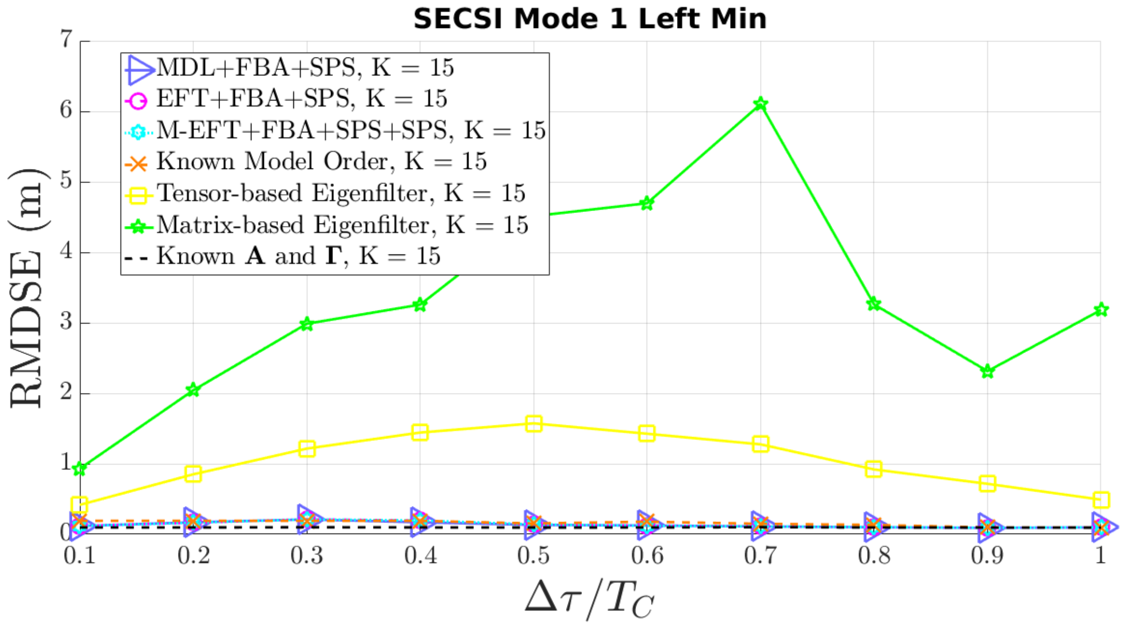


Figure 3.76: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.77 we show the simulation results for the state-of-the-art Mode 1 HOSVD

SECSI with left-hand matrix method utilizing the minimum estimated model order with $K = 30$. Note that the time-delay error remains the same even after increasing further the amount of collected epochs. Therefore, we see that the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method is insensitive to the amount of epochs.

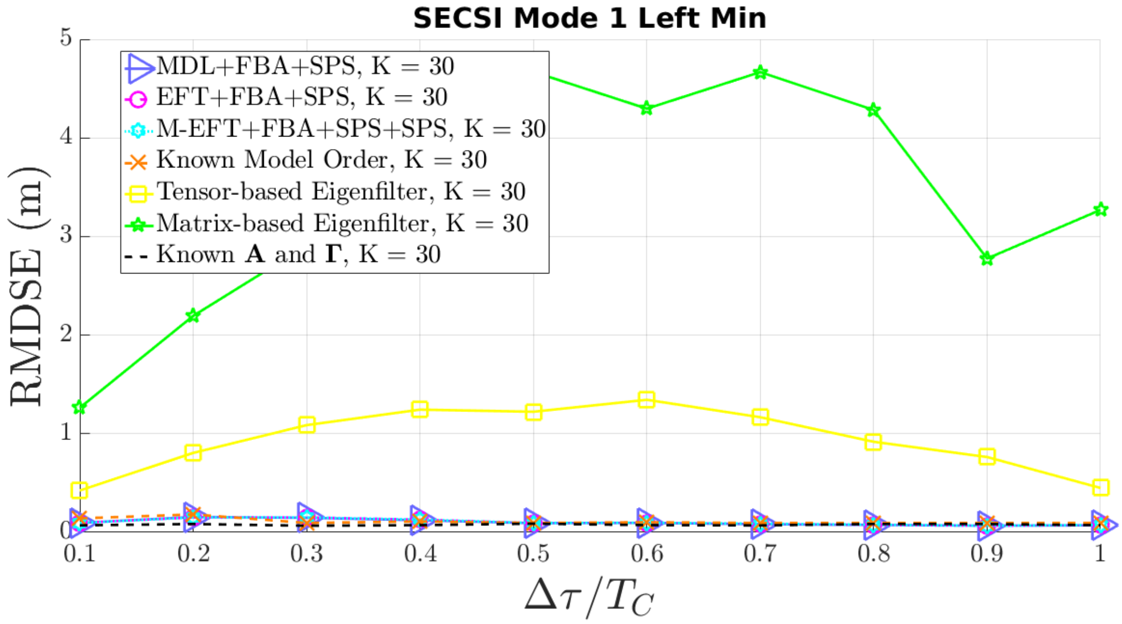


Figure 3.77: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.78 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the maximum estimated model order with $K = 8$. Note that the matrix-based methods have improved performance when we select the maximum estimated model order. Furthermore, these methods outperform the Tensor-based Eigenfilter and Known Model Order methods. Thus, by utilizing the maximum model order, we no longer need to split the tensor into various sub-tensors. Therefore, we can jointly estimate the time-delay. Moreover, selecting the maximum estimated model order does not necessarily mean we have a model order $\hat{L}_d = 3$ as shown in Figures 3.29, 3.36, 3.43.

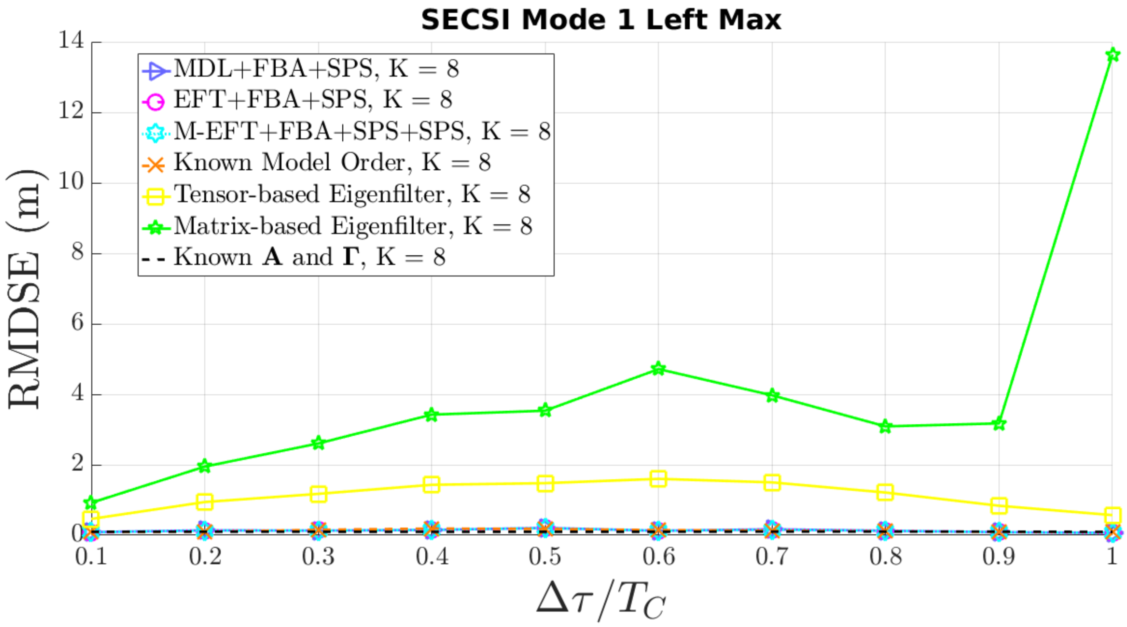


Figure 3.78: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.79 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the maximum estimated model order with $K = 15$. Note that when utilizing the maximum estimated model order we have an increase in the time-delay estimation error. This is due to modeling noise components as signals. Therefore, we have a degraded MOS and consequently a worsen TDE. However, observe that the maximum estimated model order combined with the Mode 1 HOSVD SECSI provides improved performance when compared to the minimum estimated model order in Figure 3.76.

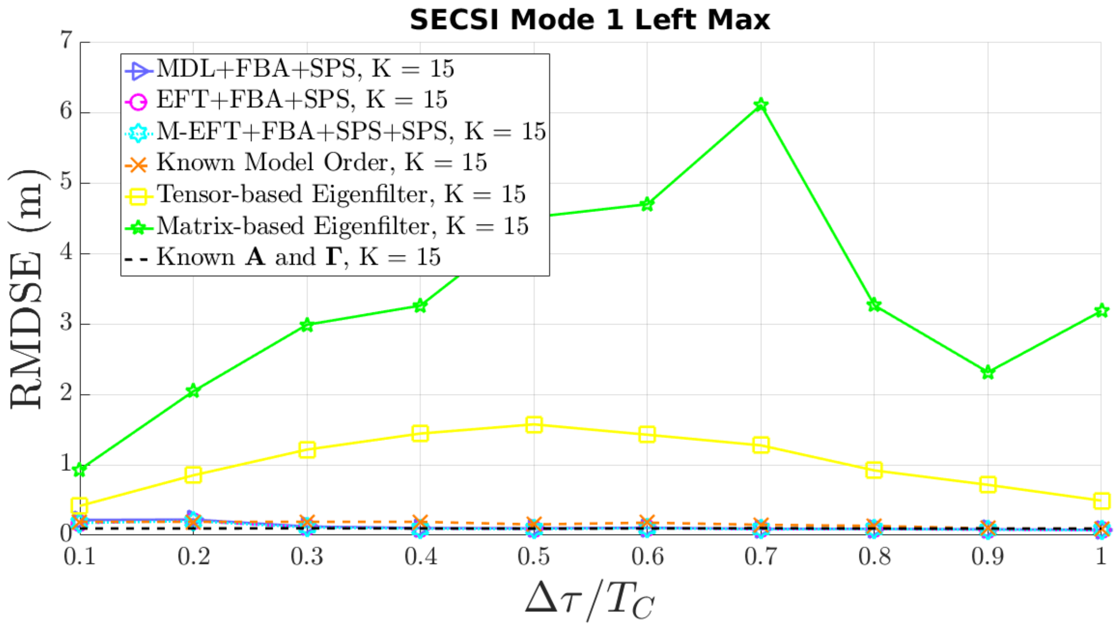


Figure 3.79: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.80 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the maximum estimated model order with $K = 30$. Again, note that when utilizing the maximum estimated model order we have an increase in the time-delay estimation error. Therefore, we have a degraded MOS and consequently a worsen TDE. However, observe that the maximum estimated model order combined with the Mode 1 HOSVD SECSI provides improved performance when compared to the minimum estimated model order in Figure 3.77. In addition, even though the Mode 1 HOSVD SECSI is more suitable to be combined with the sut-tensors approach, note that the Mode 1 HOSVD SECSI combined with the maximum estimated model order method performs displayed better perfomance with a low amount of epochs.

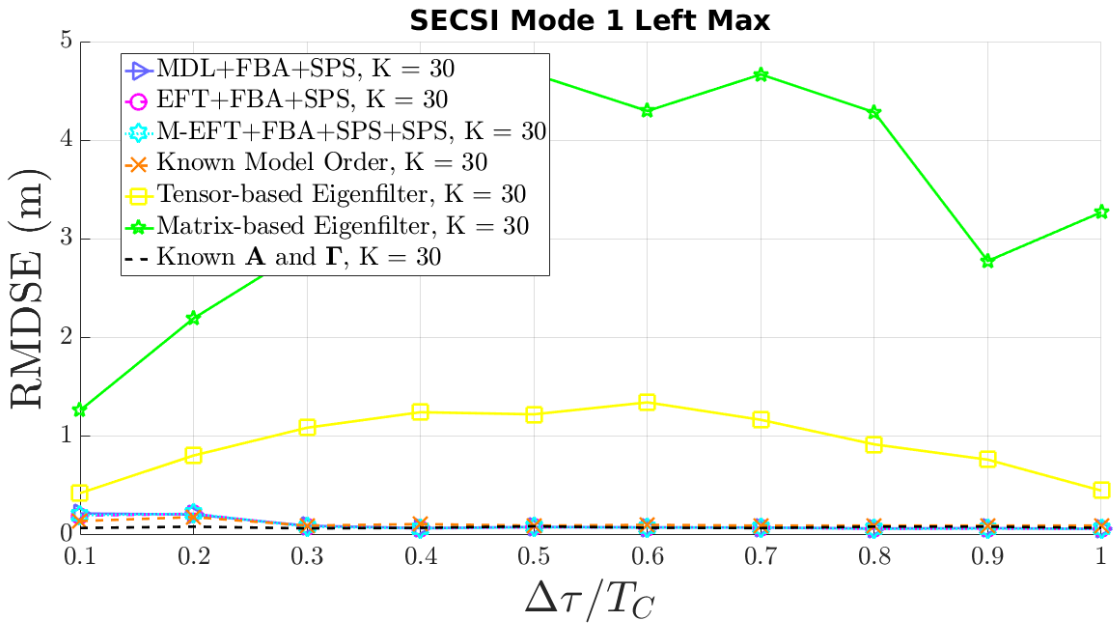


Figure 3.80: MOS techniques and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.81 we show the simulation results for the state-of-the-art DoA/KRF method with $K = 8$. As previously described, we utilize the matrix-based MOS methods to estimate the model order of each epoch then we group the epochs with same model order and create new tensors. Note the MDL+FBA+SPS, EFT+FBA+SPS, M-EFT+FBA+SPS, and Known Model Order show worse performance than the Tensor-based Eigenfilter. Since the DoA of the signal is changing over the epochs and each sub-tensor has fewer information than the original tensor, the DoA/KRF shows the worst performance among the tensor-based matrix factorization methods.

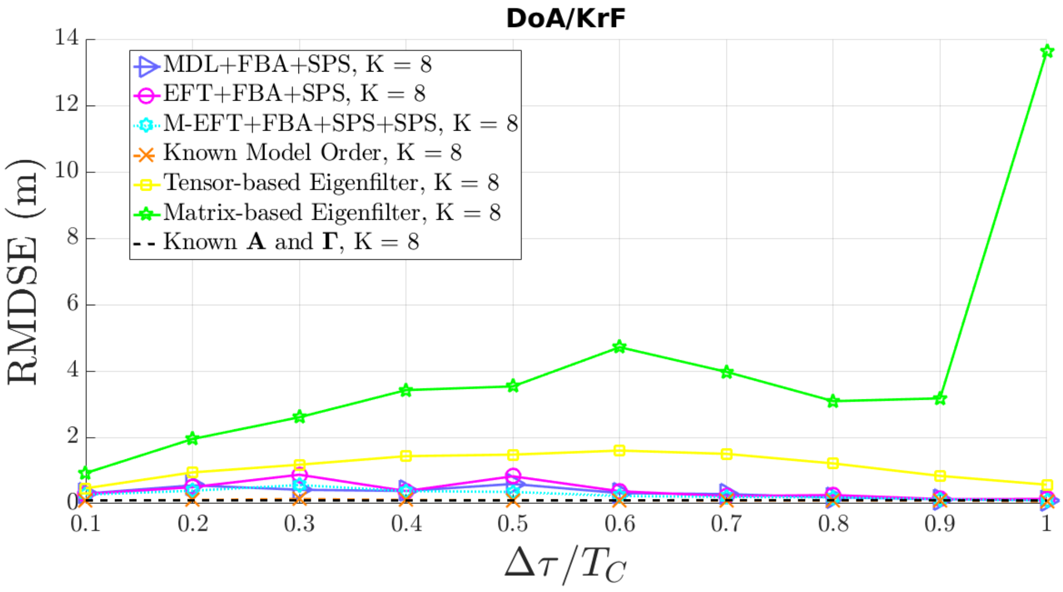


Figure 3.81: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.82 we show the simulation results for the state-of-the-art DoA/KRF method with $K = 15$. We show that when collecting more epochs and, consequently, having more variance on DoA, the DoA/KRF method combined with the matrix-based MOS methods increases the time-delay error.

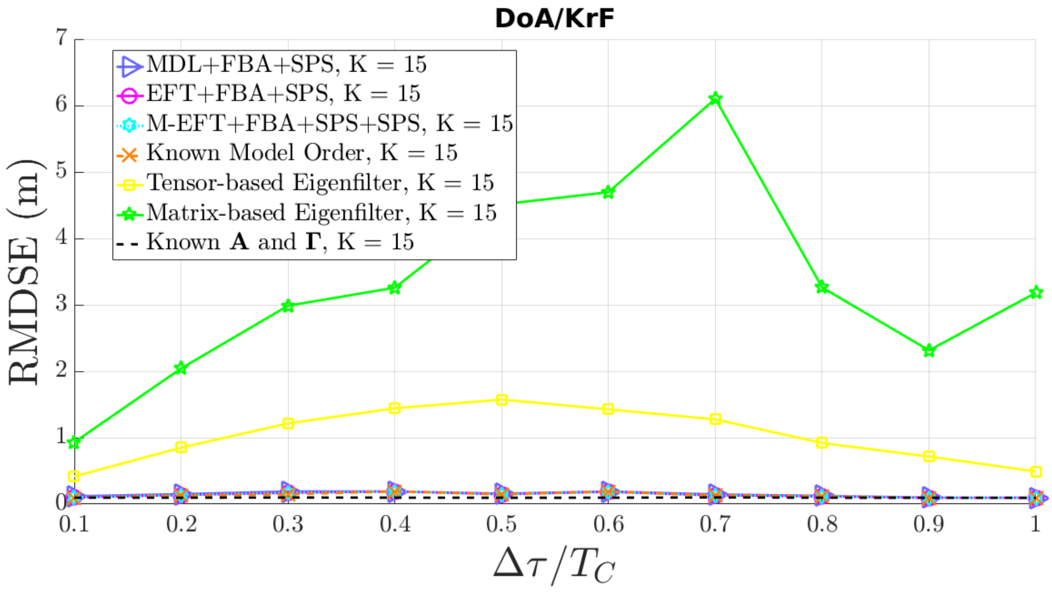


Figure 3.82: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.83 we show the simulation results for the state-of-the-art DoA/KRF method with $K = 30$. When increasasing further the amount of collected epochs, we show that the

DoA/KRF is not suitable to the method applied: we utilize the matrix-based MOS methods to estimate the model order of each epoch then we group the epochs with same model order and create new tensors.

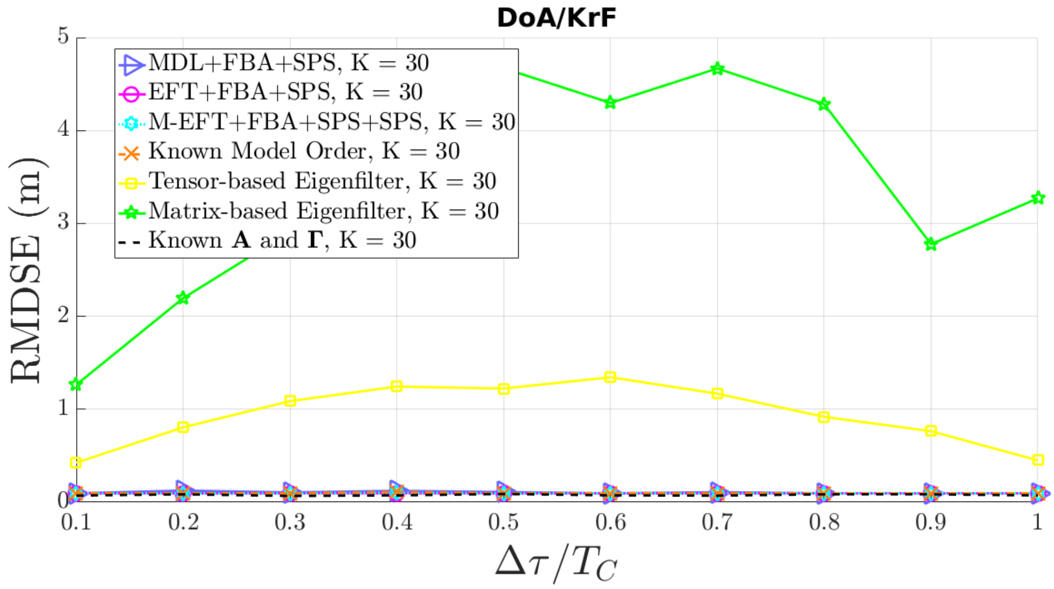


Figure 3.83: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.84 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the minimum estimated model order with $K = 8$. Note that the time-delay estimation error decreases when we select the minimum estimated model order. Moreover, notice that by spliting the tensor $\mathcal{Y}$ into smaller sub-tensors we degrade the the time-delay measurement. This degradation is due to smaller amount of information combined with the varying LOS DoA within the tensor.

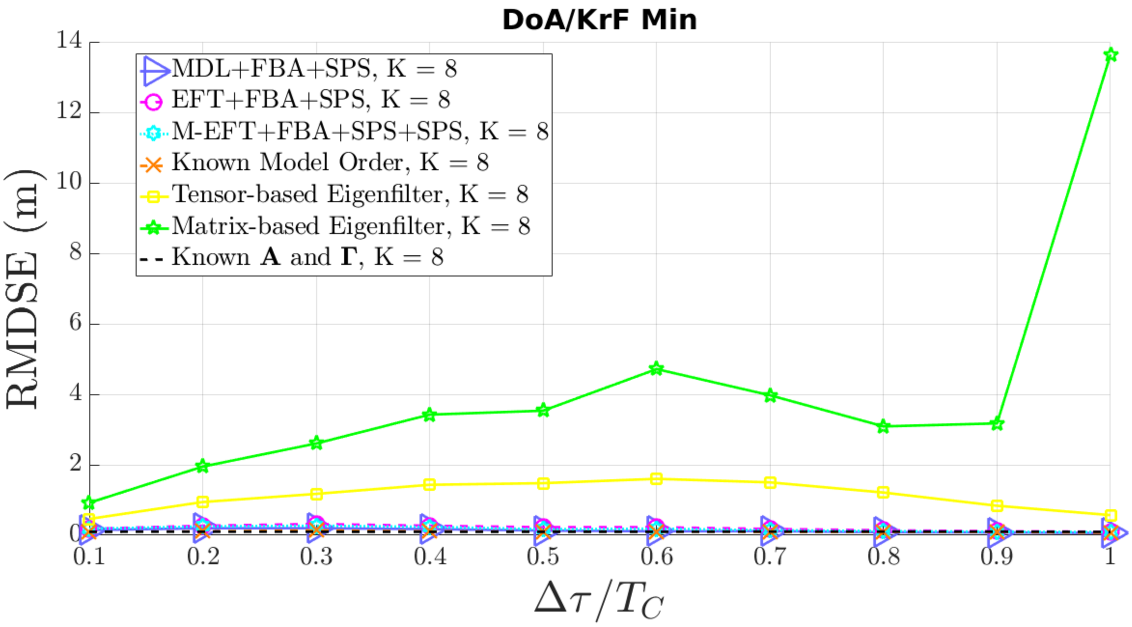


Figure 3.84: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.85 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the minimum estimated model order with $K = 15$. Notice that now, when utilizing minimum estimated model order, even though we increased the amount of collected epochs, the DoA/KRF overcomes its limitation and have similar time-delay estimaition error to Figure 3.84. However, the DoA/KRF combined with the minimum estimated model order does not show a significant improvement on time-delay estimation error.

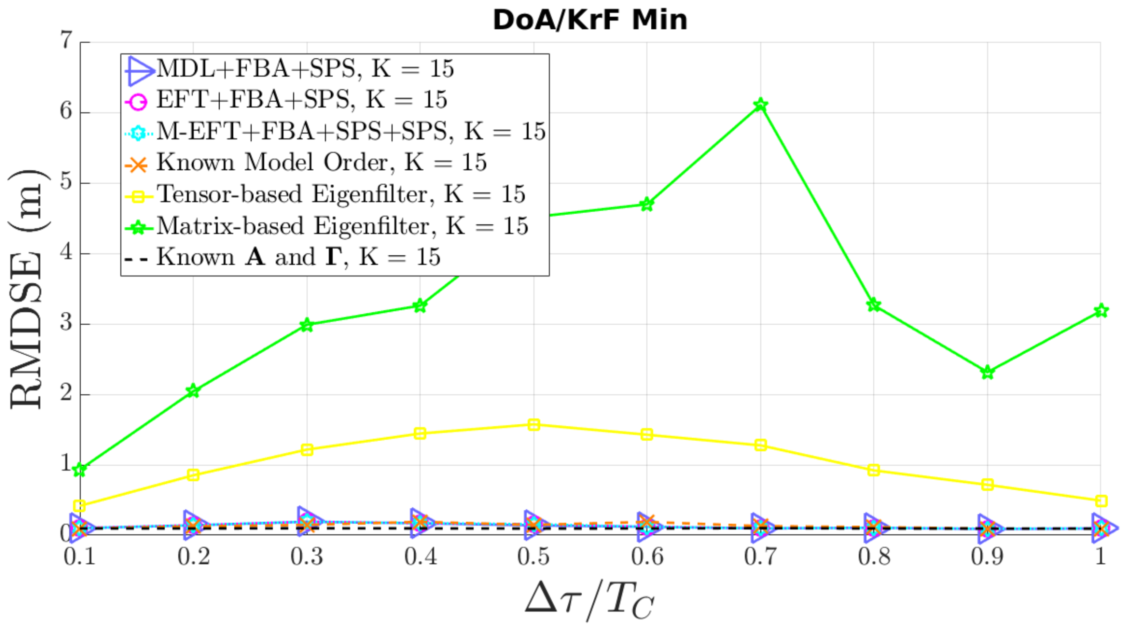


Figure 3.85: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.86 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the minimum estimated model order with $K = 30$. We reinforce that the DoA/KRF works properly when we utilize the estimated minimum model order. However, we perceive that the increase in the amount of epochs does not reflect a lower time-delay estimation.

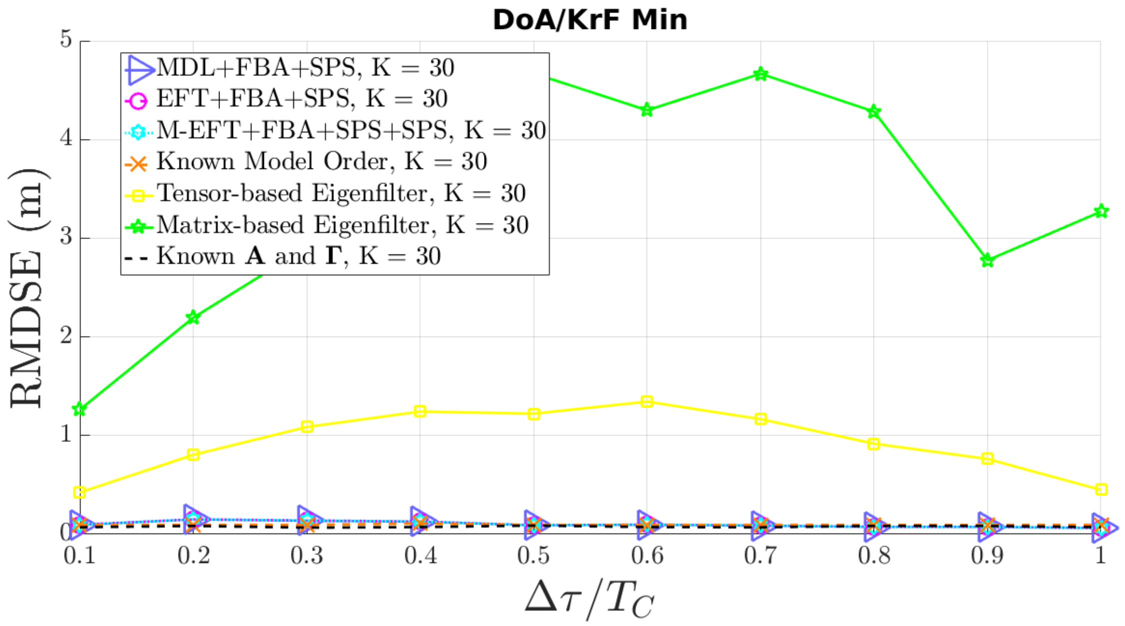


Figure 3.86: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 3.87 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the maximum estimated model order with $K = 8$. Note that the matrix-based MOS methods have improved performance when we select the maximum estimated model order. Moreover, these methods outperform the Tensor-based Eigenfilter and Known Model Order methods. Thus, by utilizing the maximum model order, we no longer need to split the tensor into various sub-tensors thus we can jointly estimate the time-delay. Furthermore, selecting the maximum estimated model order does not necessarily mean we have a model order $\hat{L}_d = 3$ as shown in Figures 3.29, 3.36, 3.43. Though, observe that the maximum estimated model order drawn from the MDL+FBA+SPS, M-EFT+FBA+SPS, and EFT+FBA+SPS outperforms the scenario where we utlize the minimum estimate model order, Figure 3.84.

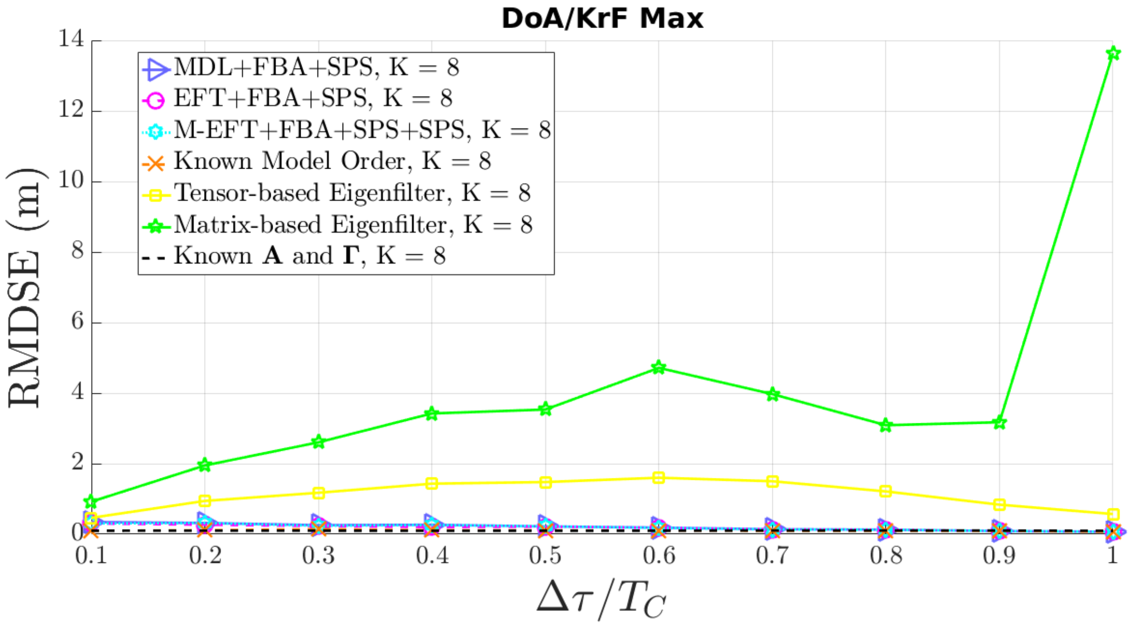


Figure 3.87: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

In Figure 3.88 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the maximum estimated model order with $K = 15$. Observe that when utilizing the maximum estimated model order in a scenario with higher amount of epochs the DoA/KRF shows a worse performance than using only $K = 8$ epochs.

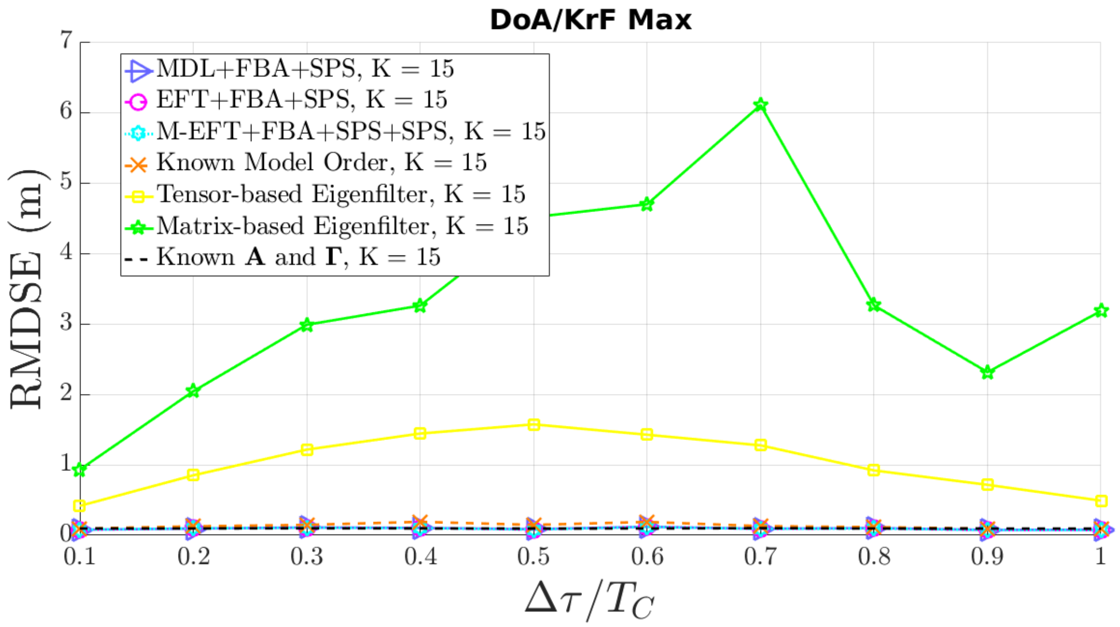


Figure 3.88: MOS techniques and tate-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

In Figure 3.89 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the maximum estimated model order with $K = 30$. We reinforce that the increase in the amount of collected epochs has negative influence on time-delay estimation whe combining the DoA/KRF with the maximum estimated model order. Moreover, we show that the DoA/KRF has better performance with a low amount of signal samples.

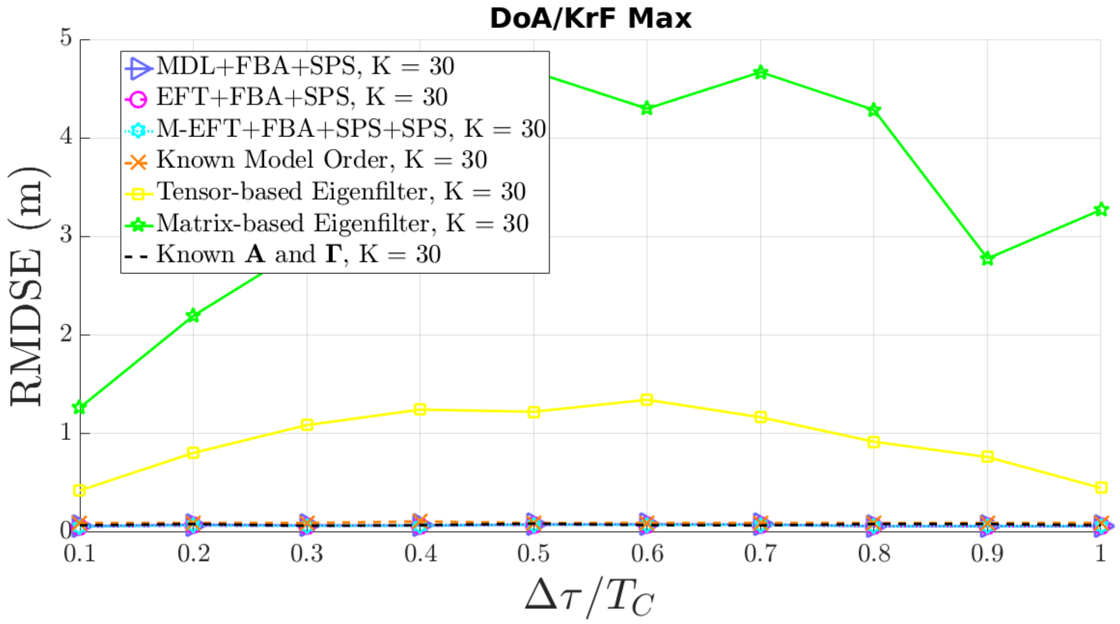


Figure 3.89: MOS techniques and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

# Chapter 4

# Time-Delay Estimation for Dynamic Scenario with MuDe

In this chapter, we present the denoising method utilized to denoise the sub-tensor $\boldsymbol{\mathcal{Y}}^{(t)}$. We included a denoising step since the time-delay estimation performance is sensitive to signal-to-noise ratio (SNR) and degraded in noisy scenarios. Therefore, we propose to use the MuDe approach, which consists of a pre-processing technique to denoise tensor-like data. MuDe combines the Spatial Smoothing (SPS) idea [6] with successive SVD-based low-rank approximations in the output signals for sub-arrays of varying size in each spatial dimension of the obtained signal tensor and, then, rebuilding the sub-arrays into a tensor.

## 4.1 Tensor-based Multiple Denoising

Hence, we firstly briefly review the SPS pre-processing scheme. Thus, consider the 3rd mode of tensor $[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)}$ from (2.9), with $M$ rows is divided into $L_r = 4$ sub-arrays of size $M^{(\text{sub})} = M - L_r + 1$. When applying the SPS pre-processing method independently in the unfolding matrix, we define the 3rd mode spatially smoothed matrix $\mathbf{Y}_{SPS,3}^{(t)^{(L_r)}} \in \mathbb{C}^{M^{(\text{sub})} \times KQ}$ to collect the output signal of the $L_r$ sub-arrays as follows [43]

$$\begin{aligned}\mathbf{Y}_{SPS,3}^{(t)^{(L_r)}} &= \left[[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)}^{(1)},\ \boldsymbol{\mathcal{Y}}]_{(3)}^{(t)^{(2)}},\boldsymbol{\mathcal{Y}}]_{(3)}^{(t)^{(3)}},\ \boldsymbol{\mathcal{Y}}]_{(3)}^{(t)^{(4)}}\right] \\ &= \left[\mathbf{J}_1^{(M_r)}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)},\ \mathbf{J}_2^{(M_r)}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)},\right. \\ &\qquad \left.\mathbf{J}_3^{(M_r)}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)},\ \mathbf{J}_4^{(M_r)}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)}\right],\end{aligned} \tag{4.1}$$

where

$$[\boldsymbol{\mathcal{Y}}^{(t)}]^{(1)}_{(3)} = \mathbf{J}^{(M_r)}_{1}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)} \tag{4.2}$$

$$[\boldsymbol{\mathcal{Y}}^{(t)}]^{(2)}_{(3)} = \mathbf{J}^{(M_r)}_{2}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)} \tag{4.3}$$

$$[\boldsymbol{\mathcal{Y}}^{(t)}]^{(3)}_{(3)} = \mathbf{J}^{(M_r)}_{3}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)} \tag{4.4}$$

$$[\boldsymbol{\mathcal{Y}}^{(t)}]^{(4)}_{(3)} = \mathbf{J}^{(M_r)}_{4}[\boldsymbol{\mathcal{Y}}^{(t)}]_{(3)} \tag{4.5}$$

are the output signal of the $\ell_r$th sub-array in the 3rd dimension. The selection matrices $\mathbf{J}^{(M_r)}_{\ell_r}$ is defined as

$$\mathbf{J}^{(M_r)}_{\ell_r} = \begin{bmatrix} \mathbf{0}_{M_r^{(sub)} \times (\ell_r - 1)} & \mathbf{I}_{M_r^{(sub)}} & \mathbf{0}_{M_r^{(sub)} \times (L_r - \ell_r)} \end{bmatrix}. \tag{4.6}$$

Afterward, the received tensor can be denoised by successively applying low-rank approximations to the set of spatially smoothed unfolding matrices $\mathbf{Y}^{(t)^{(1)}}_{SPS,3}, \ldots, \mathbf{Y}^{(t)^{(L_r)}}_{SPS,3}$ defined in (4.1). Therefore, after performing MOS and obtaining the estimated model order $L_d(k)^{(t)}$ of a given sub-tensor, we compute the low-rank approximation as following

$$\tilde{\mathbf{Y}}^{(t)^{(\ell_r)}}_{SPS,3} = \begin{bmatrix} [\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]^{(1)}_{(3)} & \cdots & \tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]^{(\ell_r)}_{(3)} \end{bmatrix}, \tag{4.7}$$

where

$$\tilde{\mathbf{Y}}^{(t)^{(\ell_r)}}_{SPS,3} = \mathbf{U}^{(t)}_{s} \boldsymbol{\Sigma}^{(t)}_{s} \mathbf{V}^{(t)^{\mathrm{H}}}_{s} \tag{4.8}$$

is obtained by applying the thin SVD to $\mathbf{Y}^{(t)^{(\ell_r)}}_{SPS,3}$. Note that the $M_r^{(\mathrm{sub})} \geq L(k)^{(t)}$.

Then, the denoised tensor $\tilde{\boldsymbol{\mathcal{Y}}} \in \mathbb{C}^{K \times Q \times M}$ can be rebuilt from the low-rank approximation of the sub-arrays $\tilde{\mathbf{Y}}^{(t)^{(L_r)}}_{SPS,3}$ from (4.7). By applying SPS, we produce overlapping sub-arrays that share sensors in the spatial dimension. Therefore, we obtain a multiple denoised unfolding matrix associated with the 3rd dimension as follows

$$[\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]_{(3)} = \begin{bmatrix} [\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]_{(3)_{1,\cdot}} \\ [\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]_{(3)_{2,\cdot}} \\ \vdots \\ [\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]_{(3)_{M_r,\cdot}} \end{bmatrix} \in \mathbb{C}^{M \times KQ}, \tag{4.9}$$

which can be rebuilt from each sub-array utilizing a mean-based method. Thus, the $m$th row of the denoised unfolding matrix $[\tilde{\boldsymbol{\mathcal{Y}}}^{(t)}]_{(3)}$ is defined as

(4.10)

where $\ell\prime$ is the number of times $[\tilde{\mathcal{Y}}^{(t)}]^{i}_{(3)_{m-i+1,\cdot}}$ is valid output in the $\ell_r$th sub-array. Finally, the denoised tensor $\tilde{\mathcal{Y}}^{(t)}$ is obtained by rearranging (4.9) in a tensor fashion. Afterward, we can utilize the pre-processed tensor $\tilde{\mathcal{Y}}^{(t)}$ to perform matrix factorization instead of the raw tensor $\mathcal{Y}^{(t)}$ as illustrated in Figure 4.1.

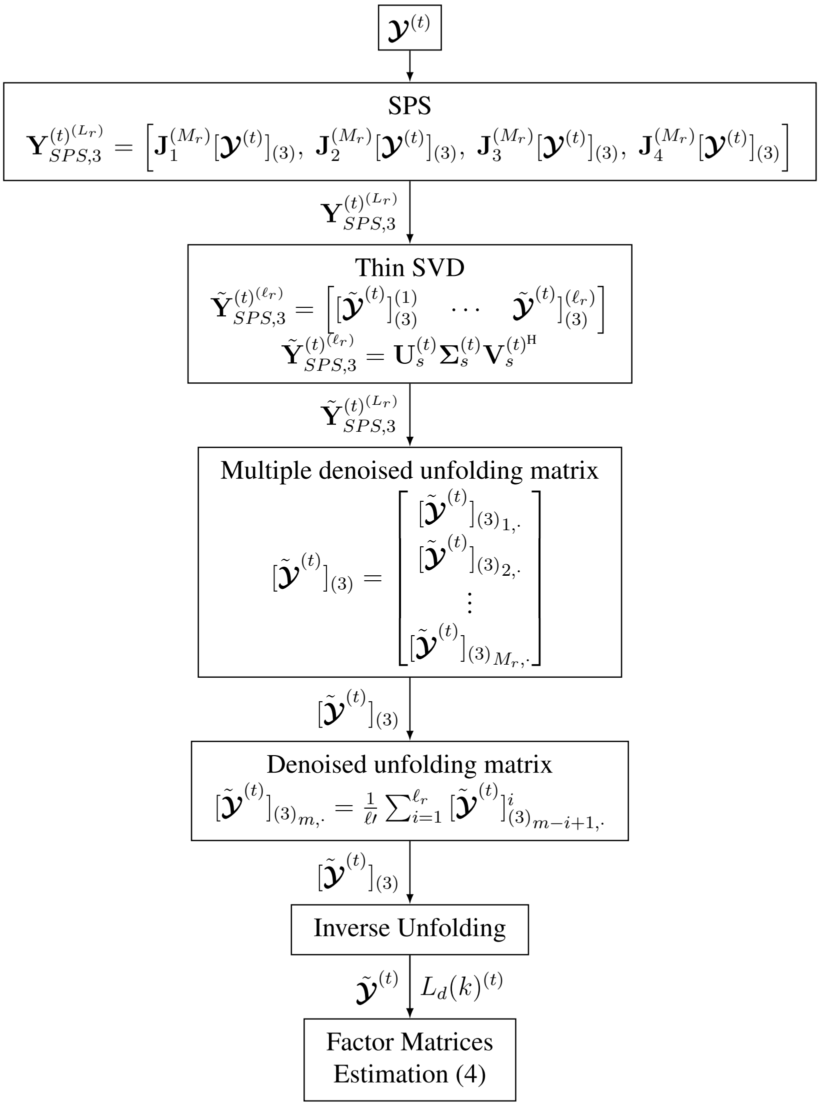


Figure 4.1: Block diagram for the MuDe method.

### 4.1.1 Time-Delay Estimation Simulations Considering Dynamic Scenarios with MuDe

In this section we present the TDE computed during simulations when considering a dynamic scenario, a perfectly aligned array of antennas, and the MuDe technique. Since in the dynamic scenario we are varying the number of LOS and NLOS components within the tensor $\mathcal{Y}$, when utilizing matrix-based MOS methods we grouped the epochs with the same estimated model order. Furthermore, since MuDe shows better performance when the DoA difference between LOS and NLOS is small, we define $\Delta\phi = 1°$.

Moreover, we could not utilize all MOS techniques previously described. During simulations with $\Delta\phi = 1°$ most MOS techniques failed to correctly estimate the $\hat{L_d}^{(t)}$ by either overestimating or not being capable of determining the model order. Similarly, we could not use the maximum estimated model order. However, when $\Delta\phi = 1°$ we could estimate $\hat{L_d}^{(t)}$ utilizing the EFT+FBA, EFT+SPS, and M-EFT+SPS methods. Additionally, we could utilize the minumum estimated model order obtained after performing the EFT+FBA+SPS, M-EFT+FBA+SPS, and MDL+FBA+SPS techniques.Furthermore, we performed simulations considering we know the model order of each epoch. Furthermore, we compare the results obtained from the Tensor-based Eigenfilter approach. Finally, we present simulation results for a $K = 30$ epochs.

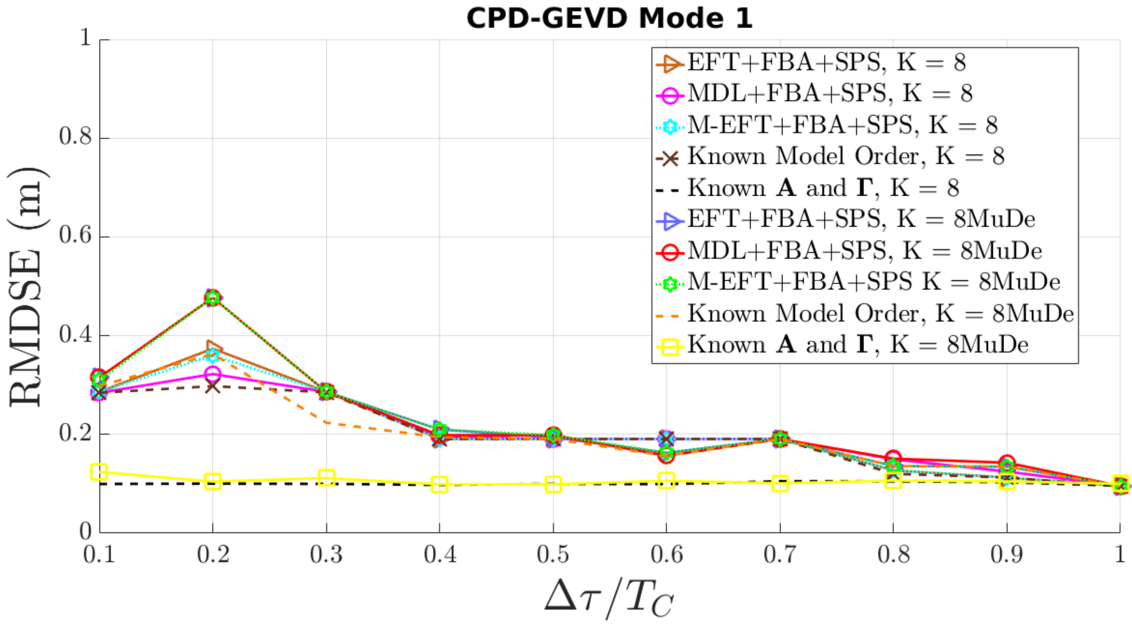


Figure 4.2: MuDe method, MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

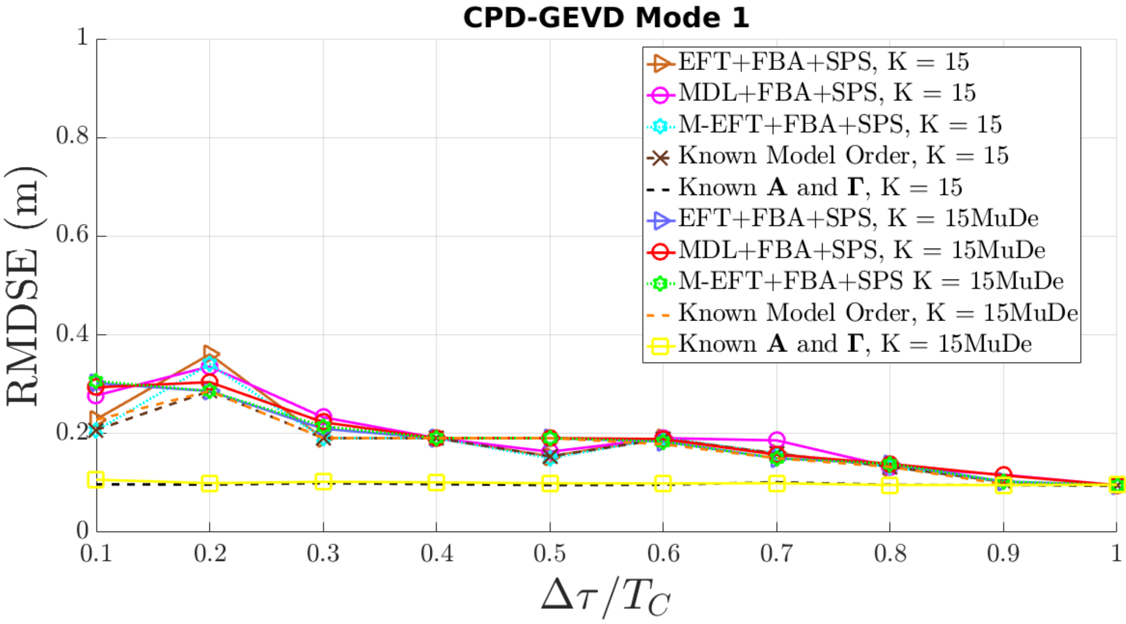


Figure 4.3: MuDe method, MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

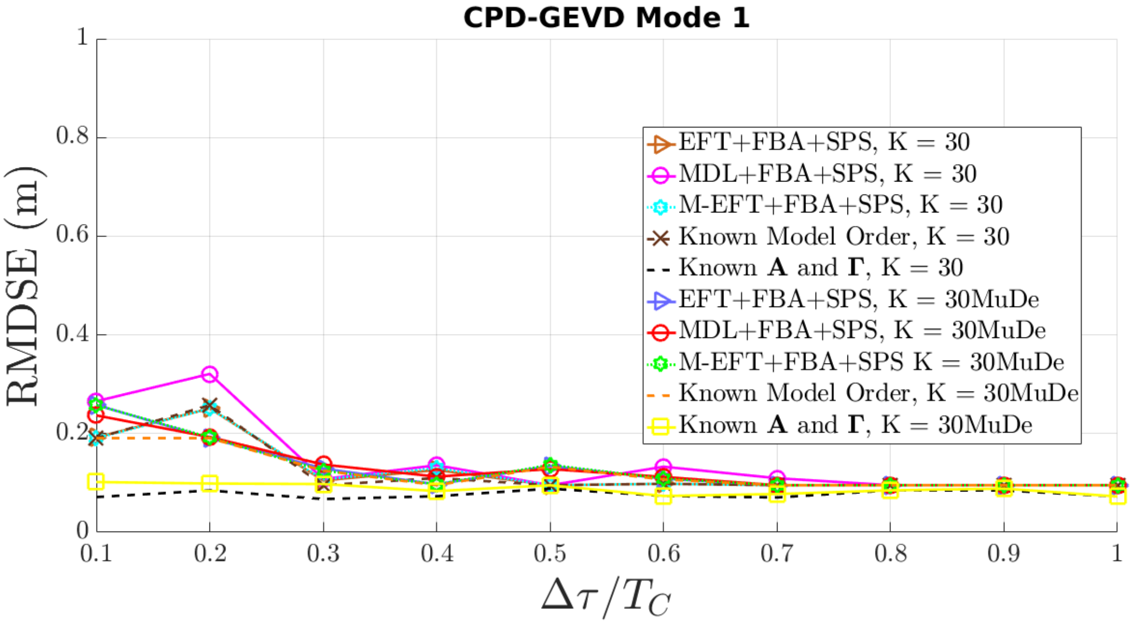


Figure 4.4: MuDe method, MOS techniques and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.5 we show the simulation results for the state-of-the-art CPD-GEVD method utilizing the MuDe method with $K = 30$ and minimum estimated model order. Observe that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is a reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Observe that, by examining the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even greater.

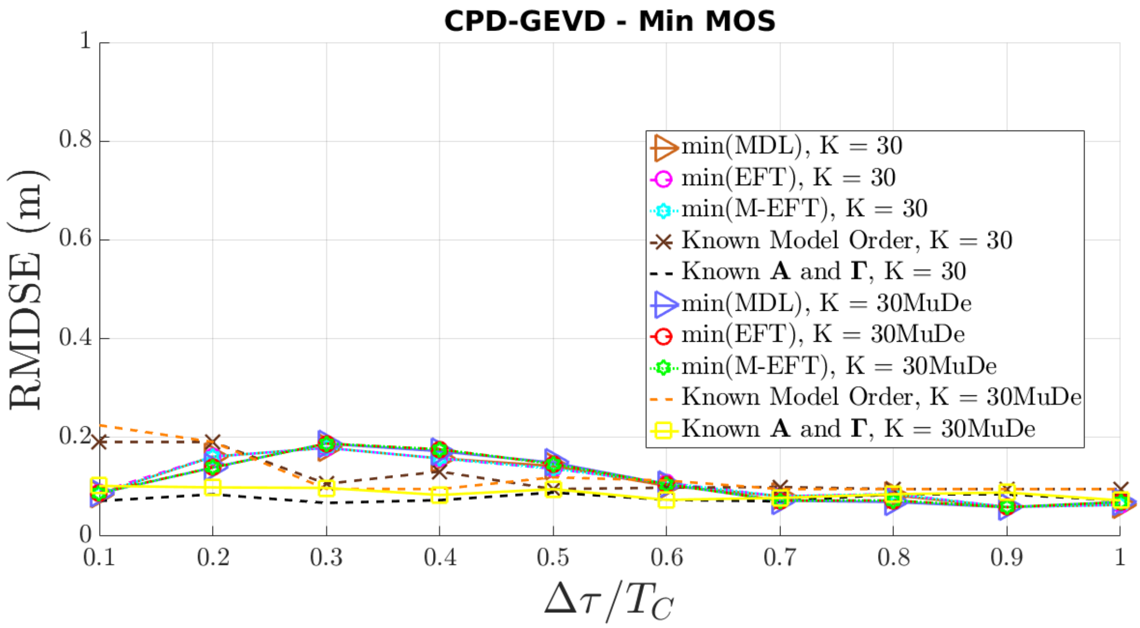


Figure 4.5: MuDe method, minimum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

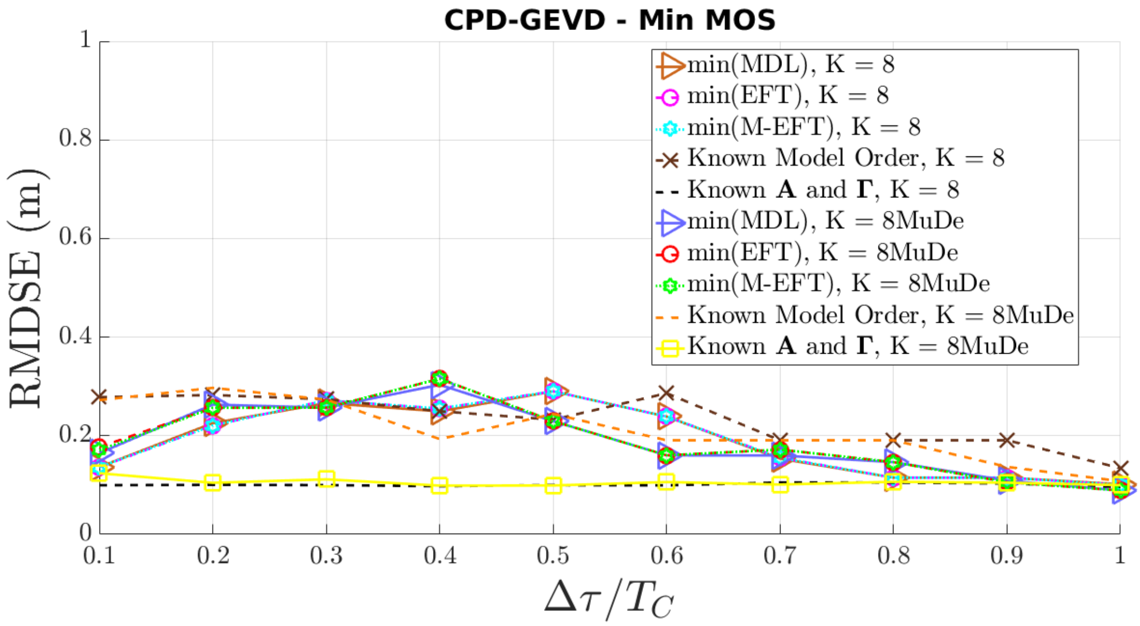


Figure 4.6: MuDe method, minimum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

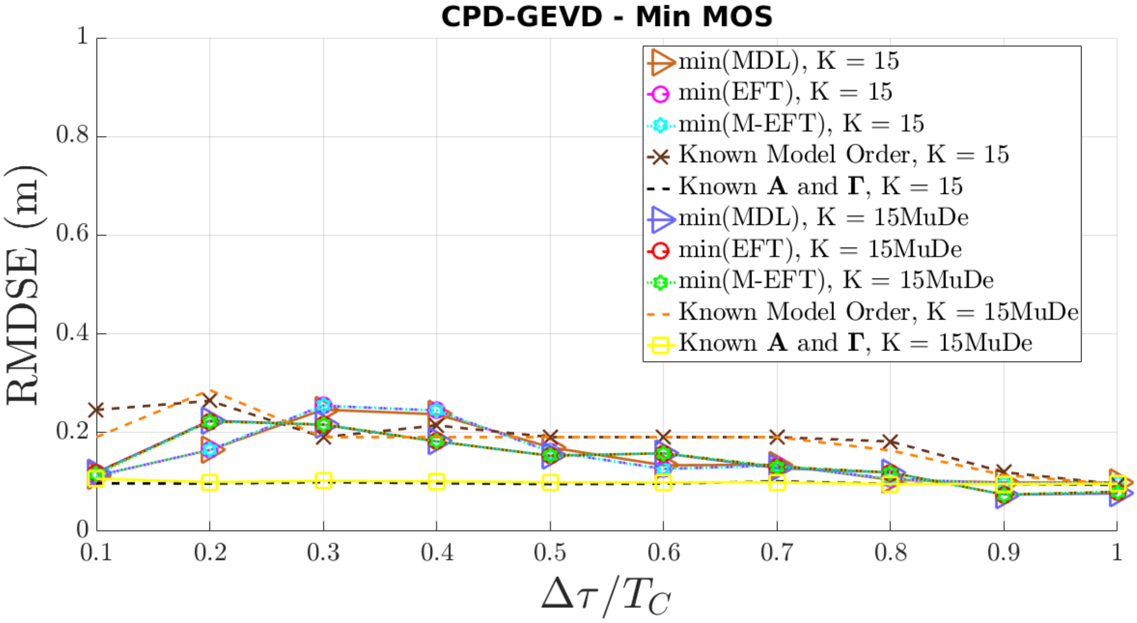


Figure 4.7: MuDe method, minimum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

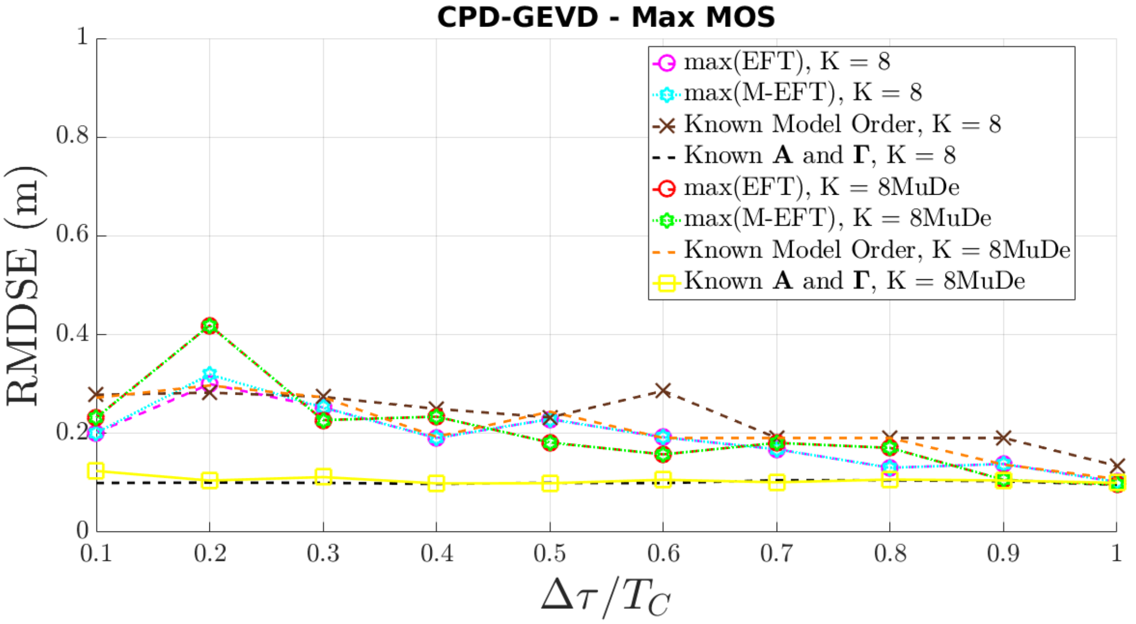


Figure 4.8: MuDe method, maximum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

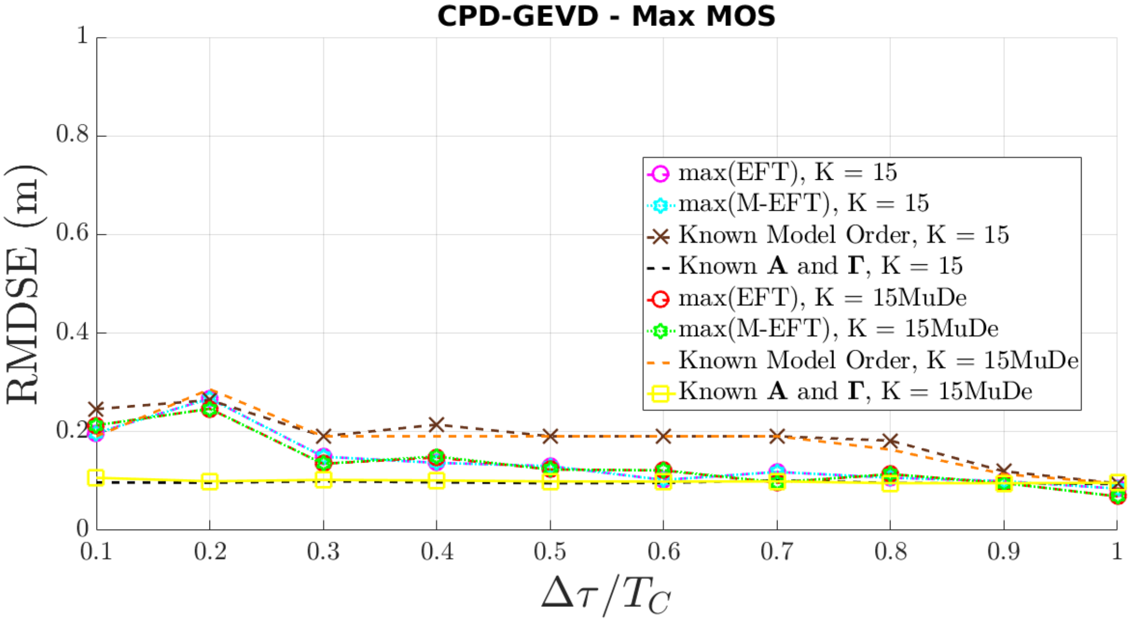


Figure 4.9: MuDe method, maximum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

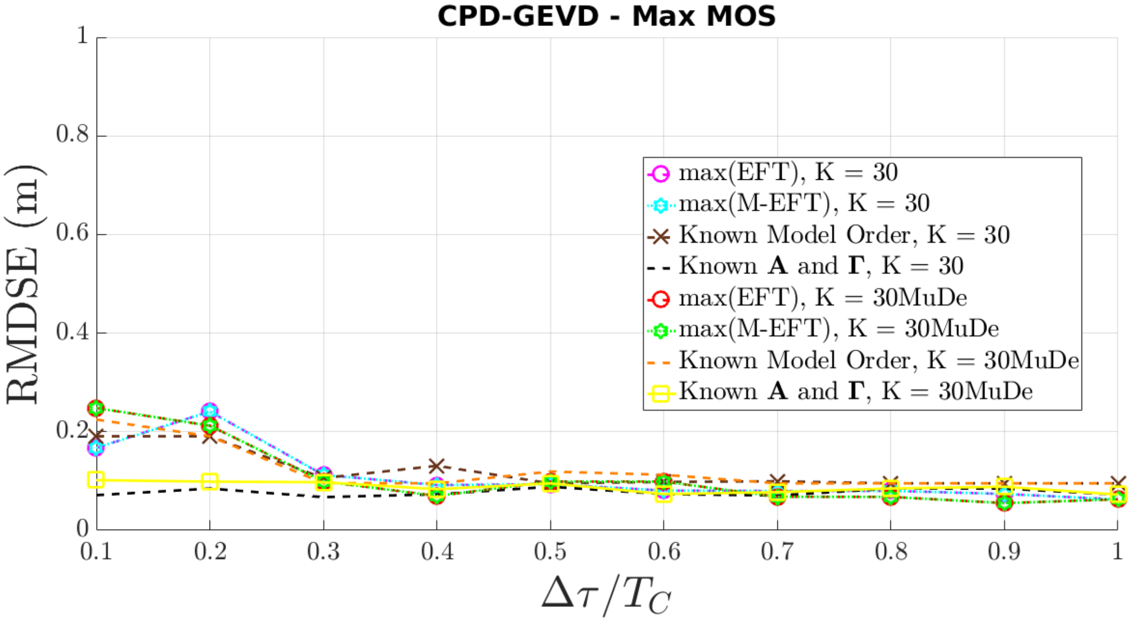


Figure 4.10: MuDe method, maximum estimated model order and state-of-the-art CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

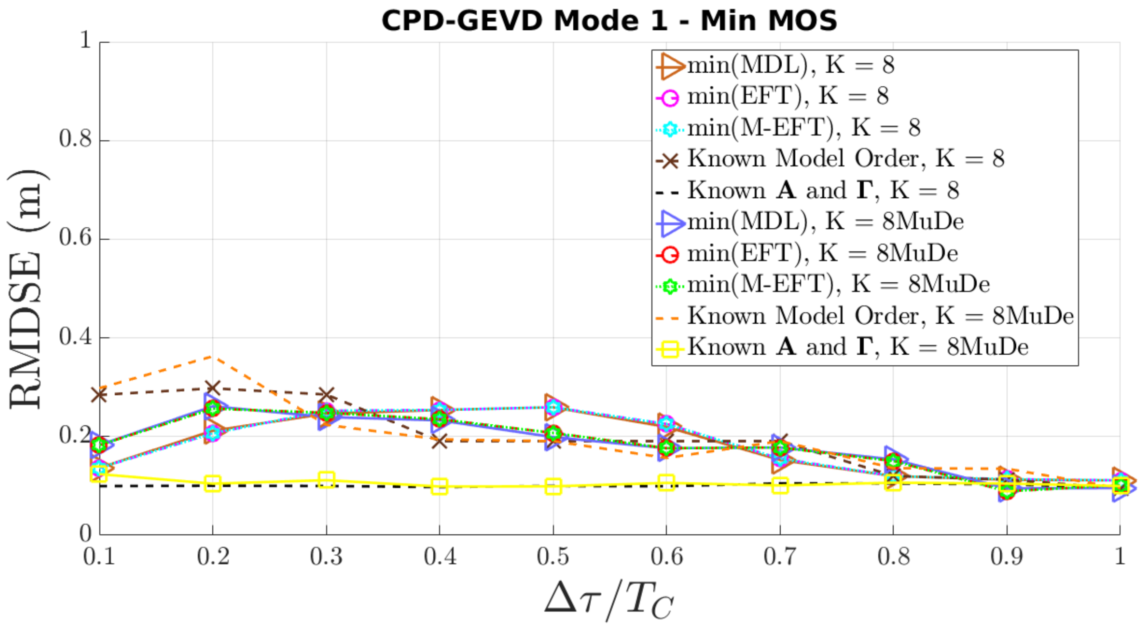


Figure 4.11: MuDe method, minimum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

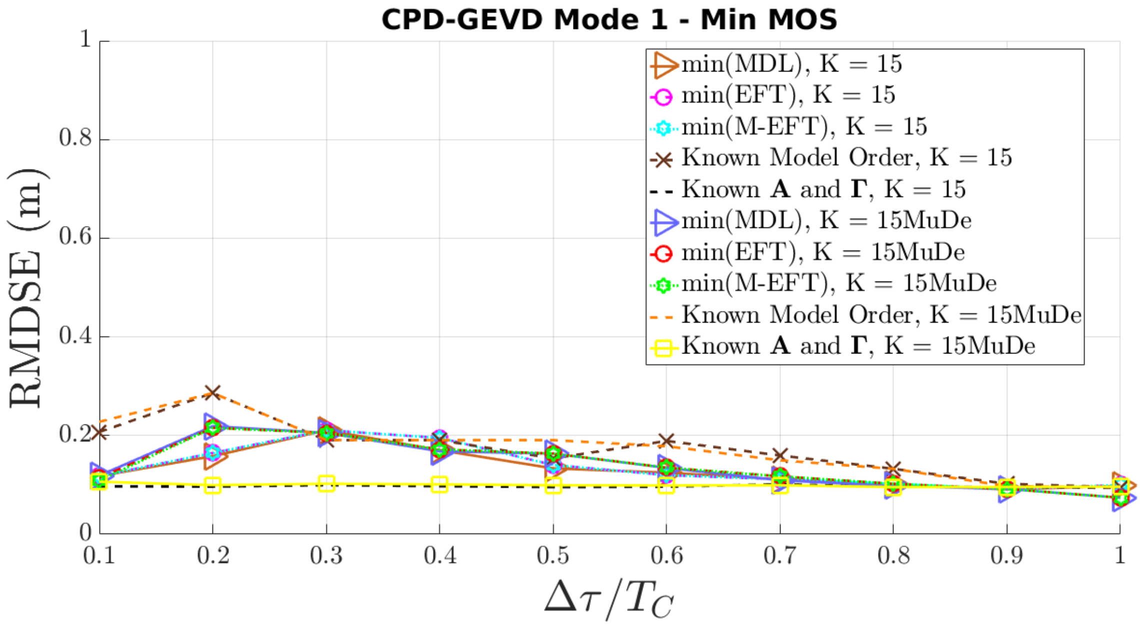


Figure 4.12: MuDe method, minimum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

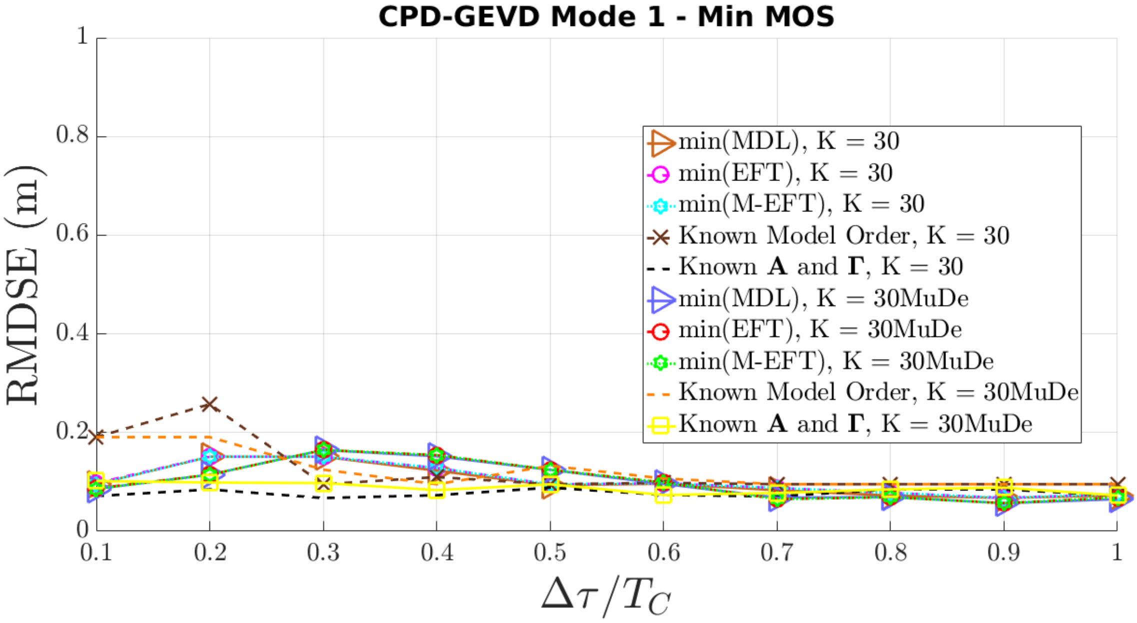


Figure 4.13: MuDe method, minimum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

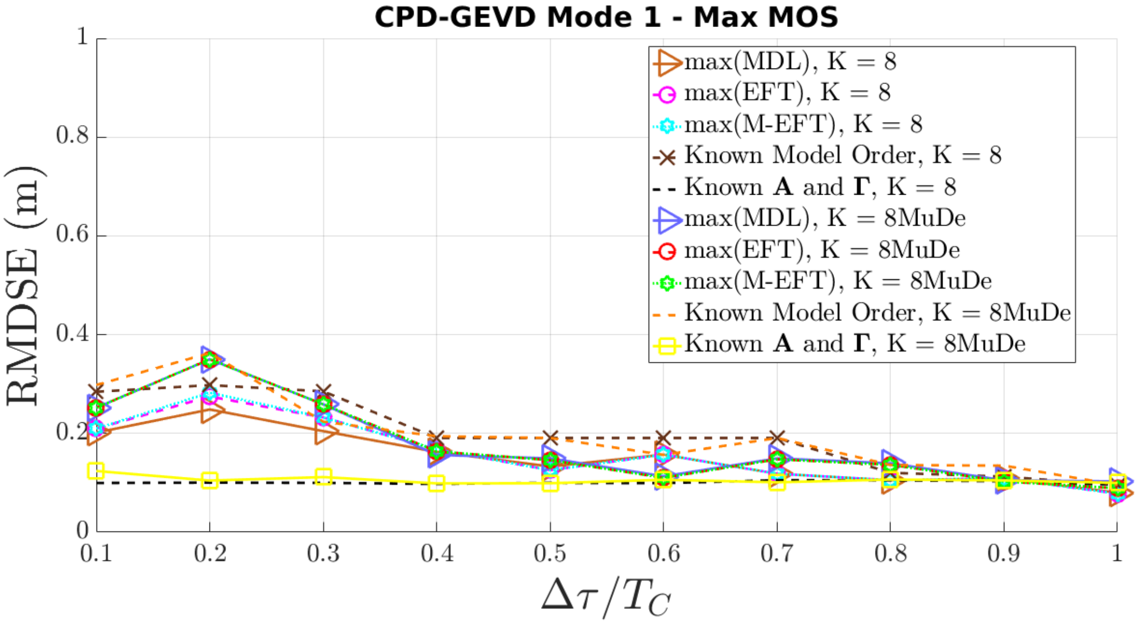


Figure 4.14: MuDe method, maximum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

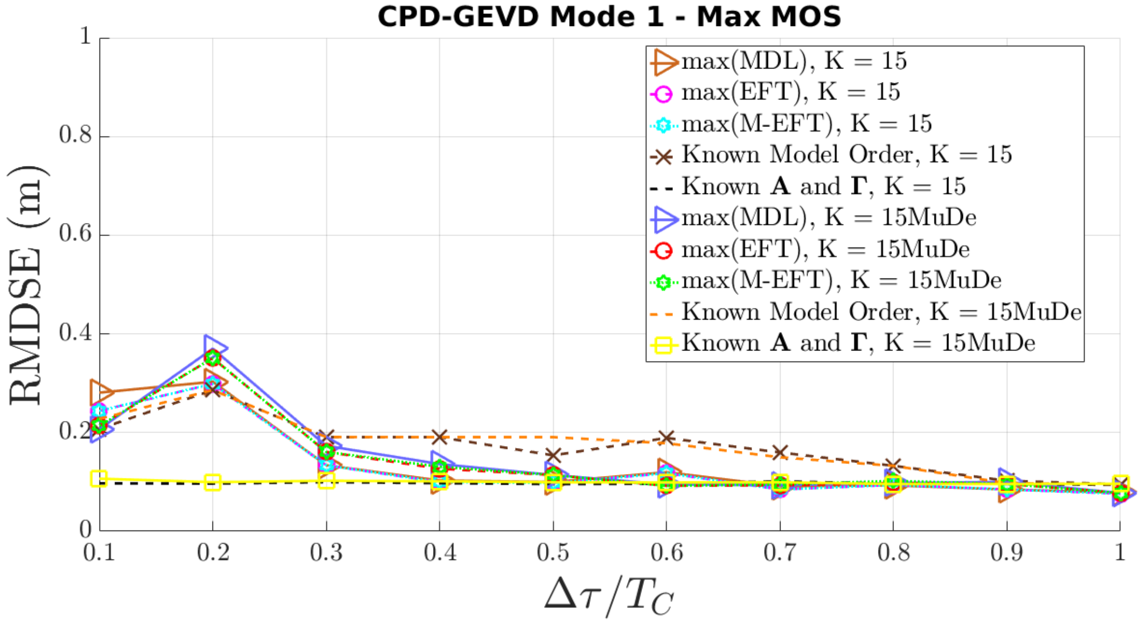


Figure 4.15: MuDe method, maximum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

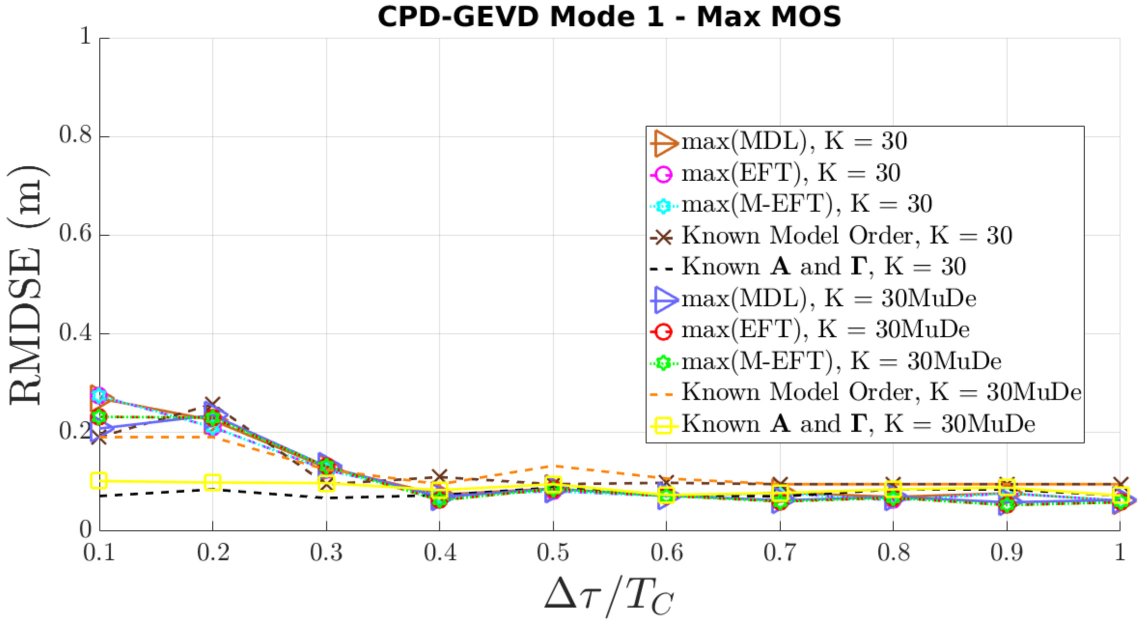


Figure 4.16: MuDe method, maximum estimated model order and state-of-the-art first mode CPD-GEVD method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.17 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix approach utilizing the MuDe method with $K = 30$. Note that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is a reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Observe that, by inspecting the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even higher. Moreover, when we assume a known model order, the MuDe contributes to reduce the time-delay estimation error. Furthermore, it is perceptible the gain, when applying MuDe to the sub-tensors, since the MOS methods EFT+FBA, EFT+SPS, and M-EFT+SPS curves show a lower error when combined with MuDe, which is the outcome of the denoise capability offered by MuDe.

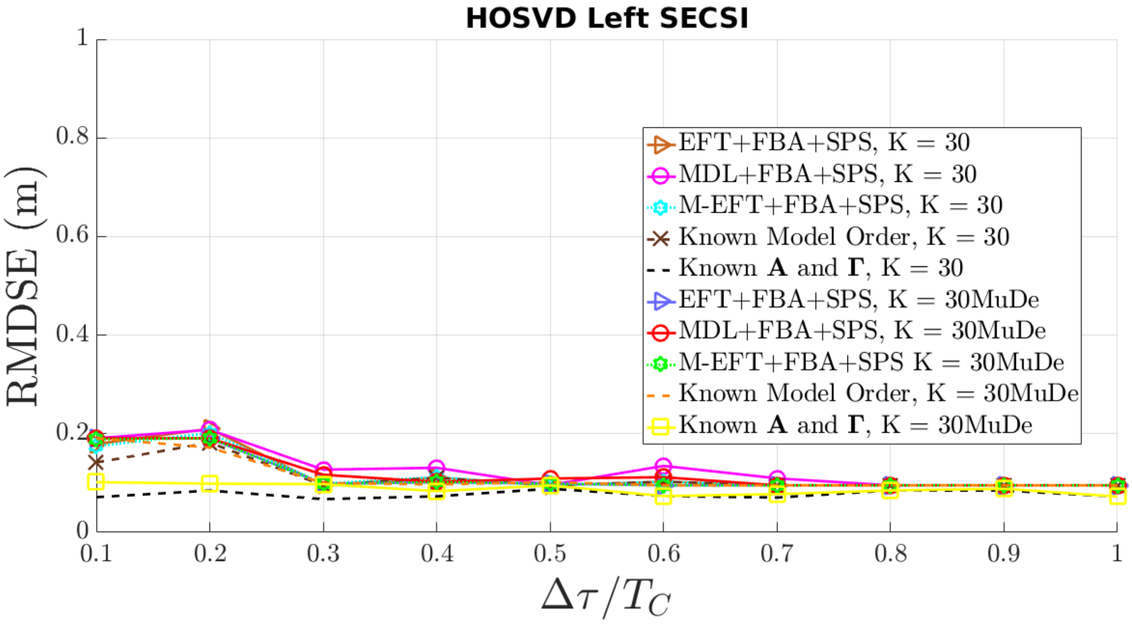


Figure 4.17: MuDe method, MOS techniques and state-of-the-art state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.18 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method utilizing the MuDe method with $K = 30$ and minimum estimated model order. Observe that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is a reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Observe that, by examining the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even greater. Moreover, note that when utilizing the minimum estimated model order, we obtain a slightly smaller time-delay estimation error than in Figure 4.17. However, observe that the EFT+FBA from Figure 4.17 provides a better estimate than the minimum estimated model order, which is due to a more accurate model order to perform the MuDe denoising.

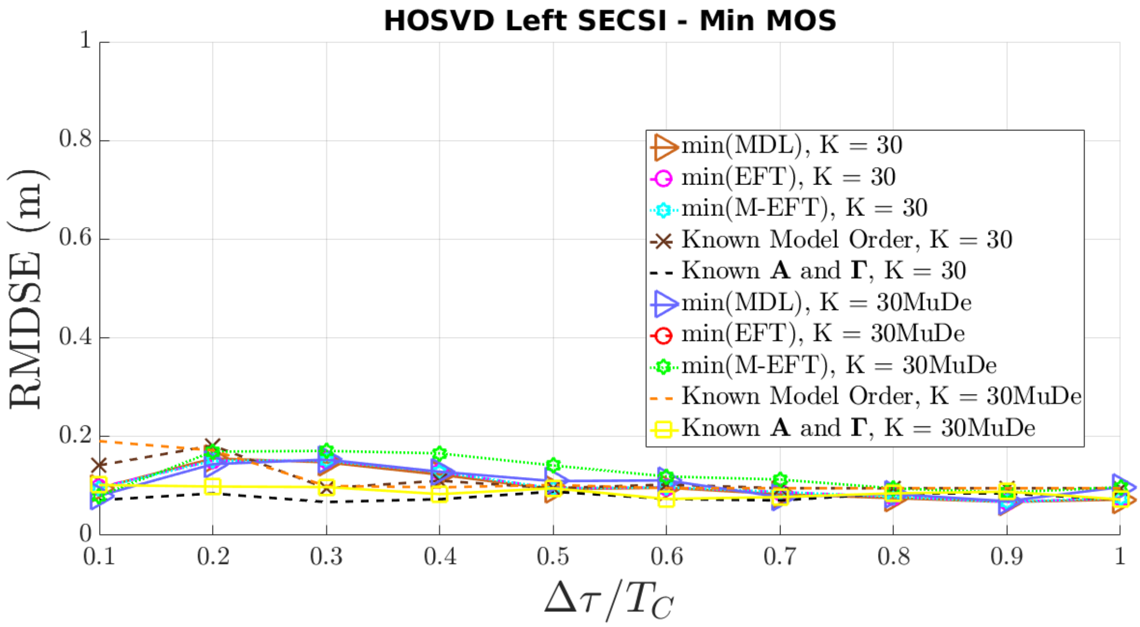


Figure 4.18: MuDe method, minimum estimated model order and state-of-the-art Mode 1 HOSVD SECSI with left-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.19 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with right-hand matrix approach utilizing the MuDe method with $K = 30$. Observa that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is a reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Note that, by inspecting the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even higher. Moreover, when we assume a known model order, the MuDe contributes to reduce the time-delay estimation error. Furthermore, it is perceptible the gain, when applying MuDe to the sub-tensors, since the MOS methods EFT+FBA, EFT+SPS, and M-EFT+SPS curves show a lower error when combined with MuDe. Aforementioned is the outcome of the denoise capability added by MuDe.

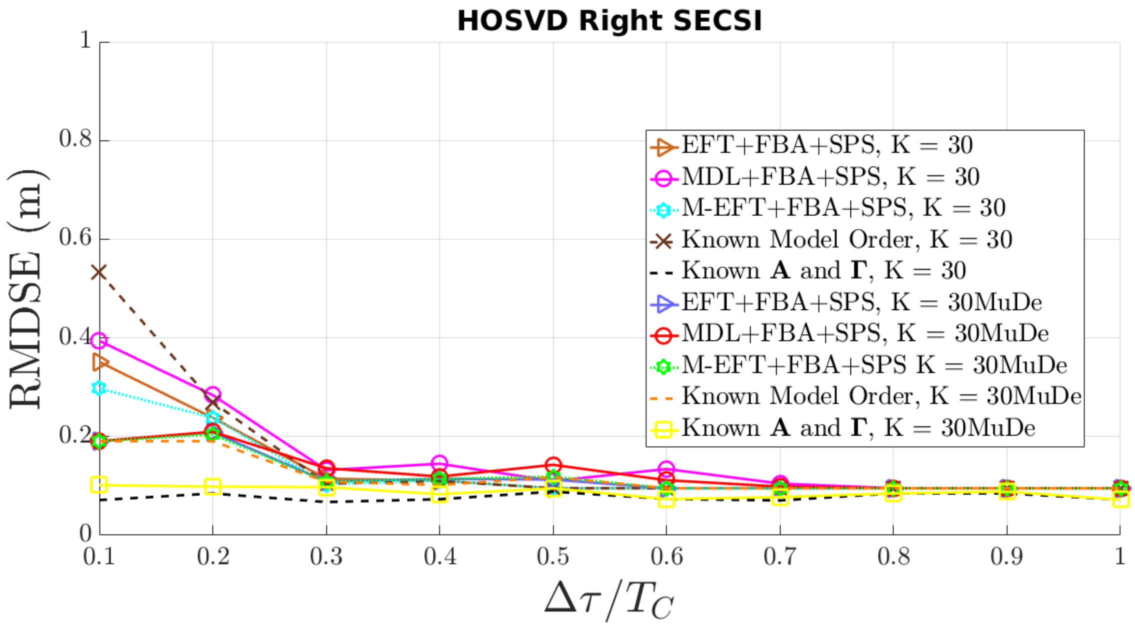


Figure 4.19: MuDe method, MOS techniques and state-of-the-art state-of-the-art Mode 1 HOSVD SECSI with right-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.20 we show the simulation results for the state-of-the-art Mode 1 HOSVD SECSI with right-hand matrix method utilizing the MuDe method with $K = 30$ and minimum estimated model order. Observe that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is utilized as reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Note that, by inspecting the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even greater. Furthermore, note that when utilizing the minimum estimated model order, we obtain a slightly smaller time-delay estimation error than in Figure 4.19. However, observe that the EFT+FBA from Figure 4.19 provides a better estimate than the minimum estimated model order, which is due to providing a more accurate model order to the MuDe denoising step.

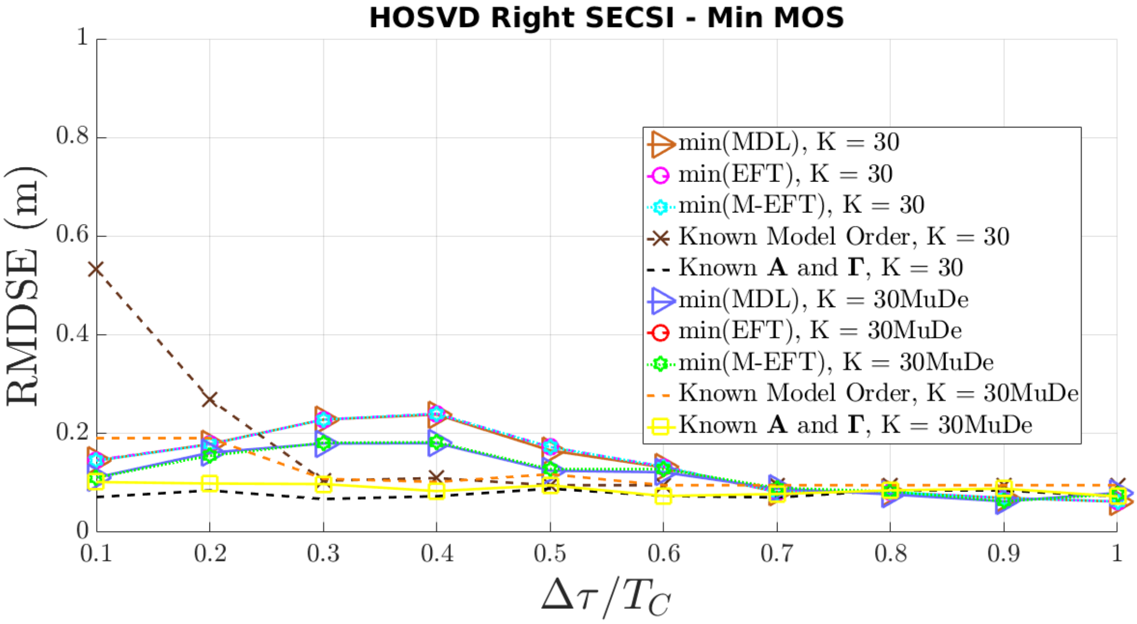


Figure 4.20: MuDe method, minimum estimated model order and state-of-the-art Mode 1 HOSVD SECSI with right-hand matrix method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

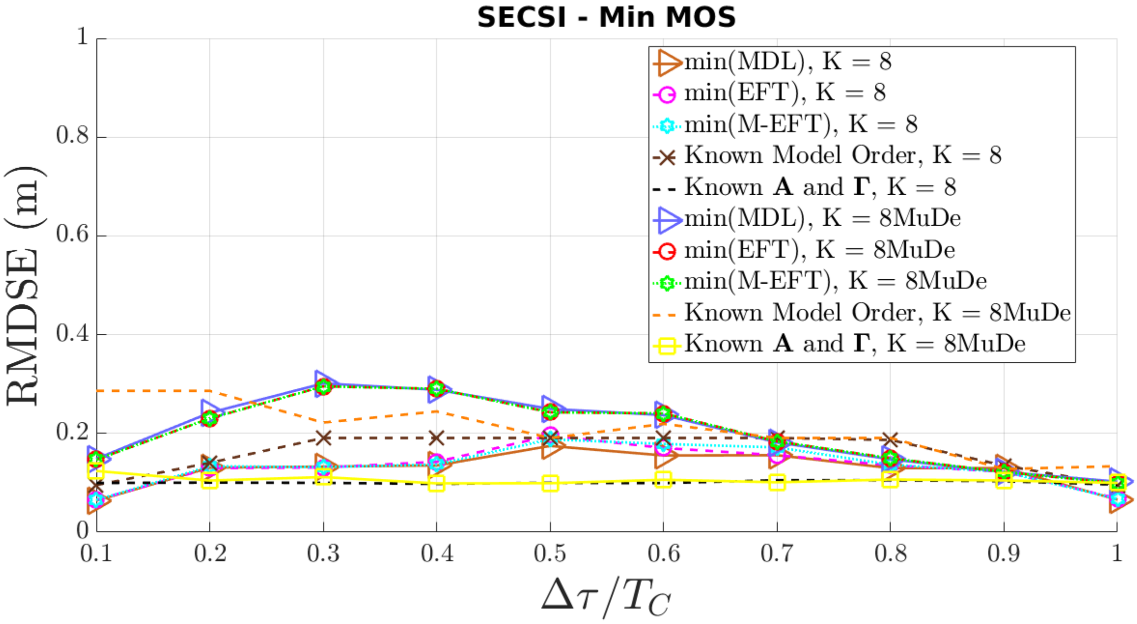


Figure 4.21: MuDe method, minimum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

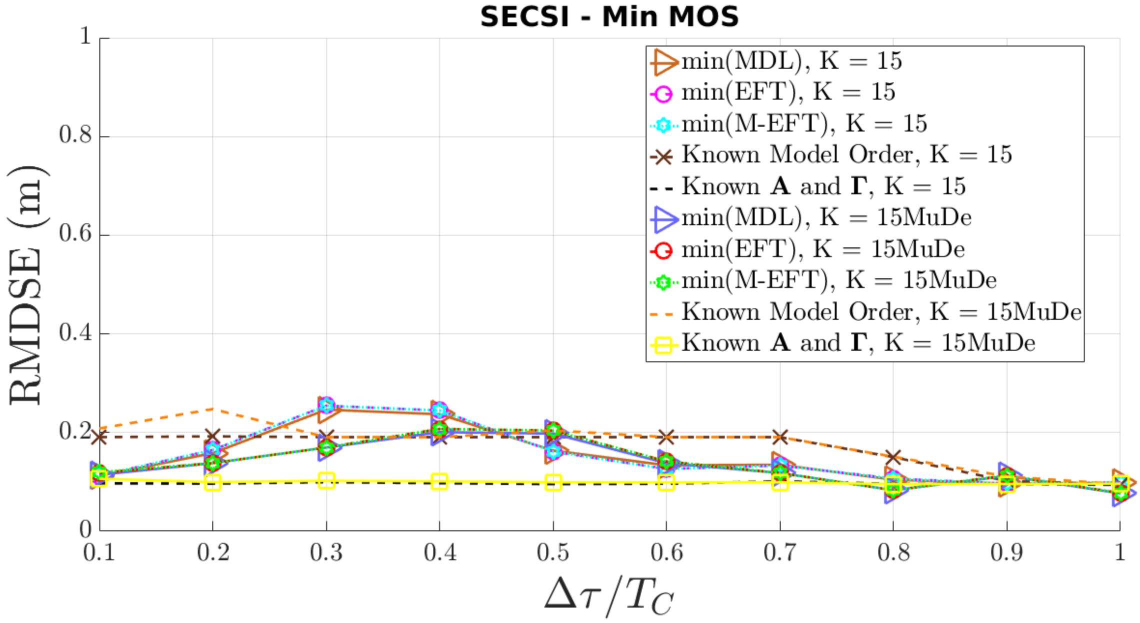


Figure 4.22: MuDe method, minimum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

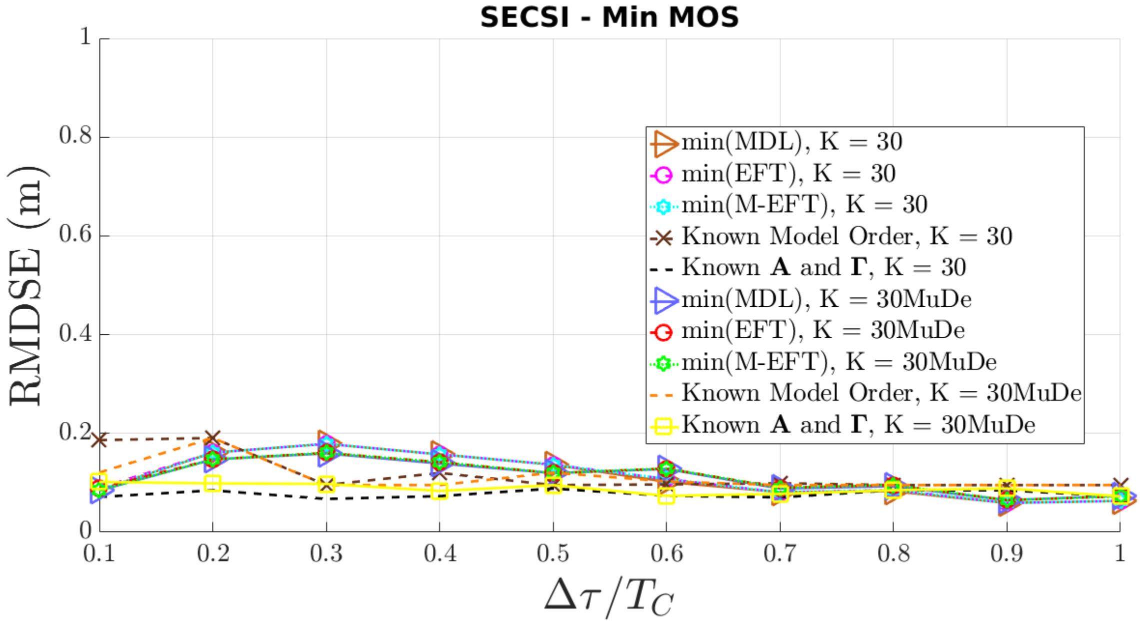


Figure 4.23: MuDe method, minimum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

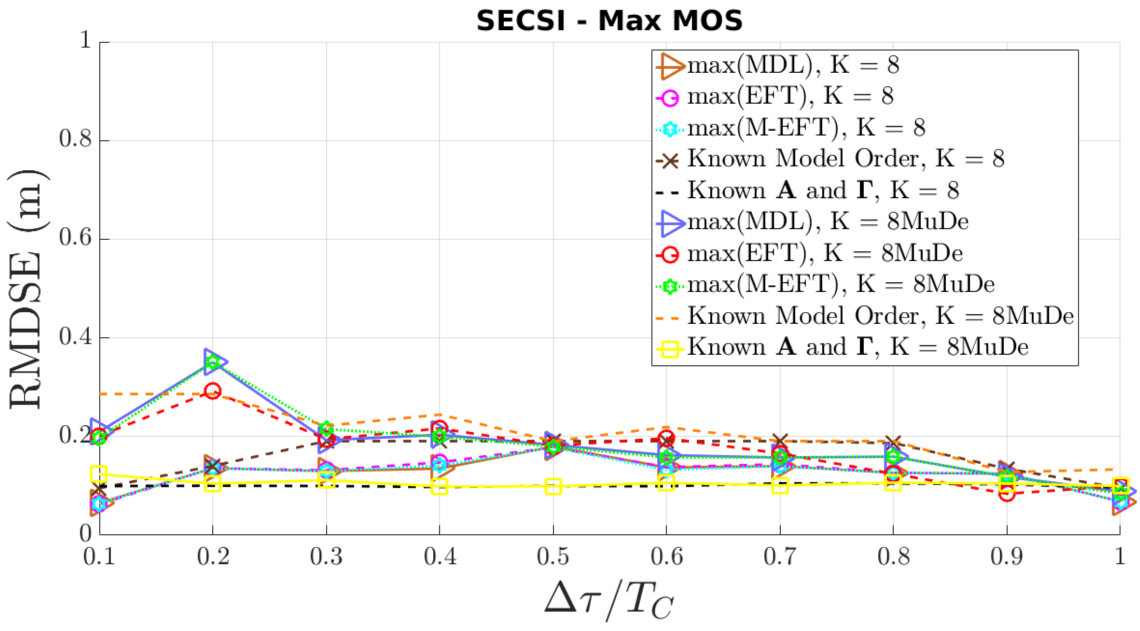


Figure 4.24: MuDe method, maximum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 8$ epochs, and have $N = 245520$ samples.

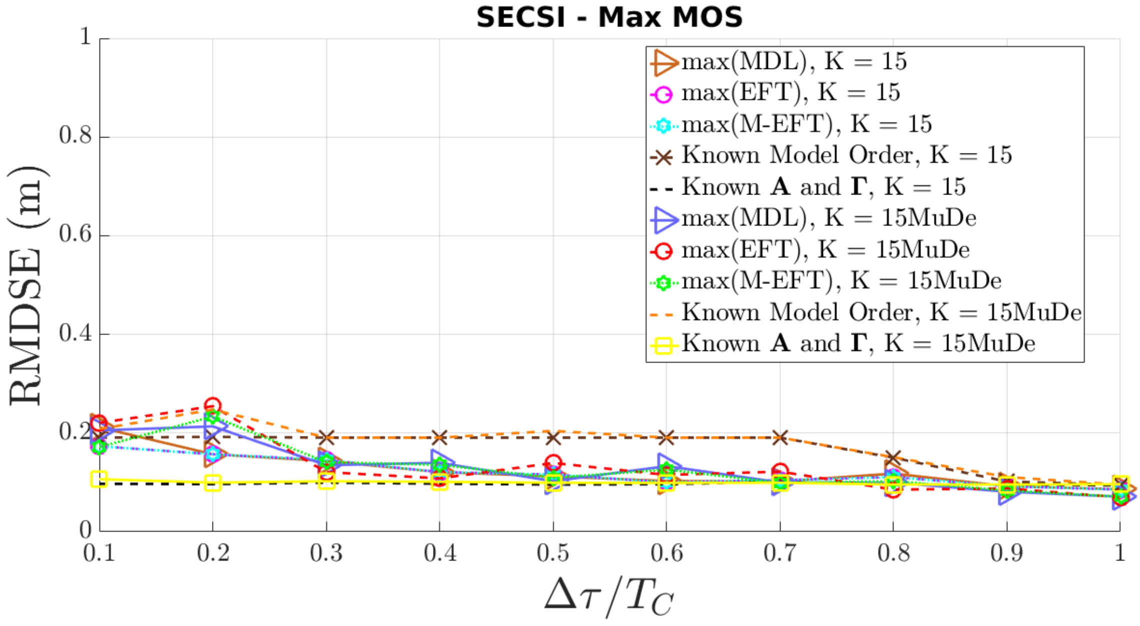


Figure 4.25: MuDe method, maximum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 15$ epochs, and have $N = 245520$ samples.

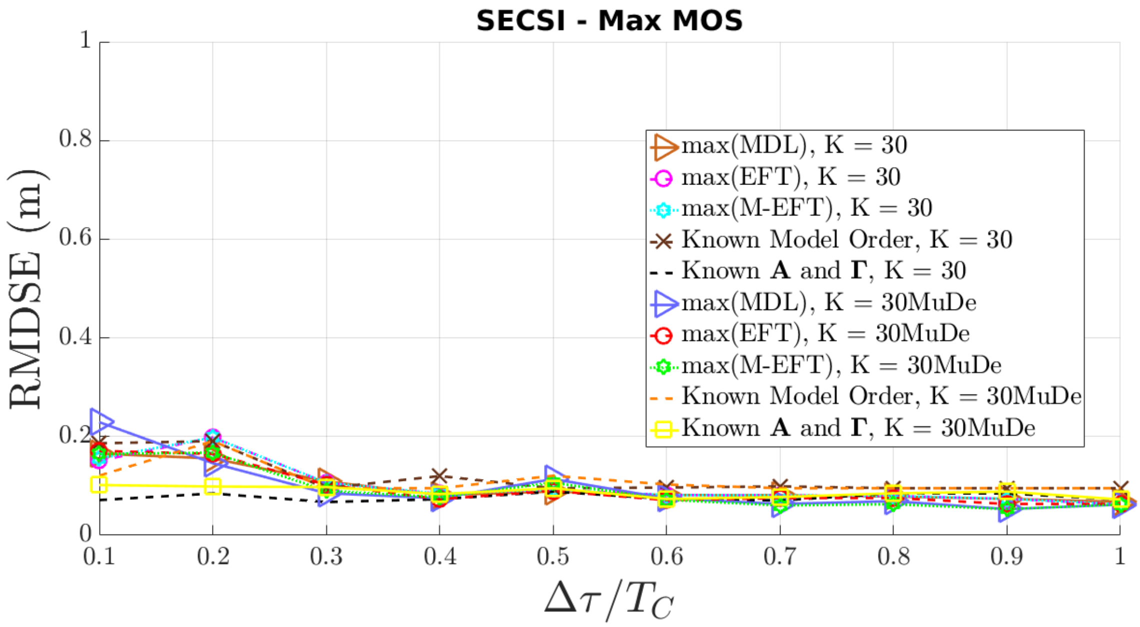


Figure 4.26: MuDe method, maximum estimated model order and state-of-the-art HOSVD SECSI method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.27 we show the simulation results for the state-of-the-art DoA/KRF approach utilizing the MuDe method with $K = 30$. Observa that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is a reference to the smallest error in noisy scenarios. However, in a real-world scenario, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Note that, by inspecting the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even higher. Moreover, when we assume a known model order, the MuDe contributes to reduce the time-delay estimation error. Furthermore, it is perceptible the gain, when applying MuDe to the sub-tensors, since the MOS methods EFT+FBA, EFT+SPS, and M-EFT+SPS curves show a lower error when combined with MuDe. Aforementioned is the outcome of the denoise capability included by MuDe.

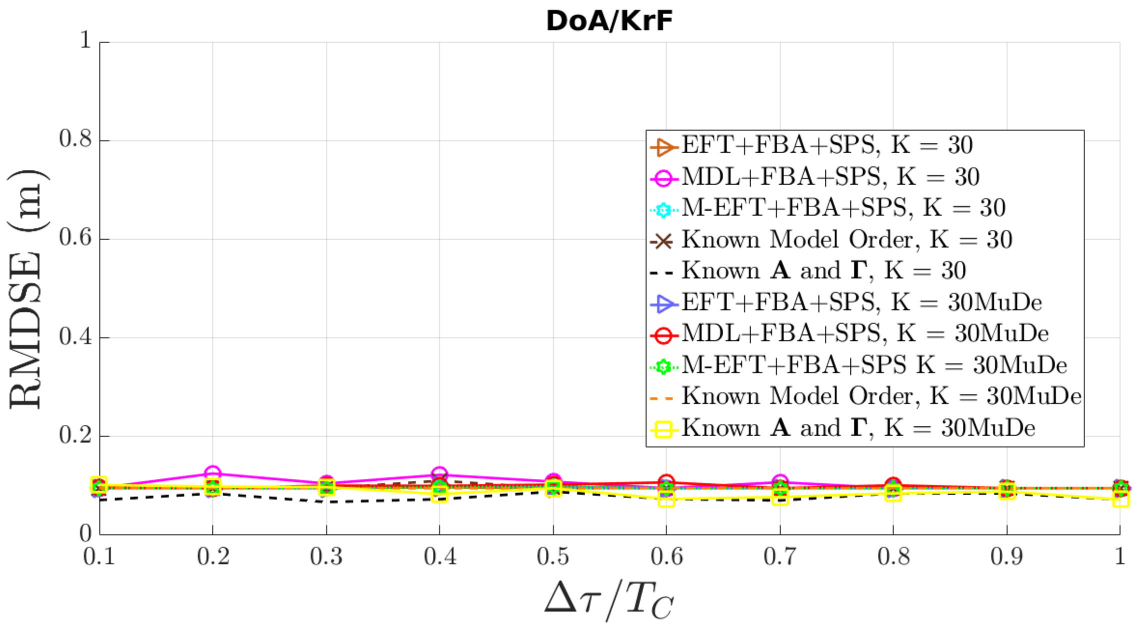


Figure 4.27: MuDe method, MOS techniques and state-of-the-art state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

In Figure 4.28 we show the simulation results for the state-of-the-art DoA/KRF method utilizing the MuDe method with $K = 30$ and minimum estimated model order. Observe that the ideal case with known $\mathbf{A}$ and $\mathbf{\Gamma}$ is utilized as reference to the smallest error in noisy scenarios. However, in practice, the $\mathbf{A}$ and $\mathbf{\Gamma}$ must be estimated. Note that, by inspecting the ideal cases with and without MuDe, if the tensor had higher and bigger dimensions, then the gain would be even greater. Furthermore, note that when utilizing the minimum estimated model order, we obtain a slightly smaller time-delay estimation error than in Figure 4.27.

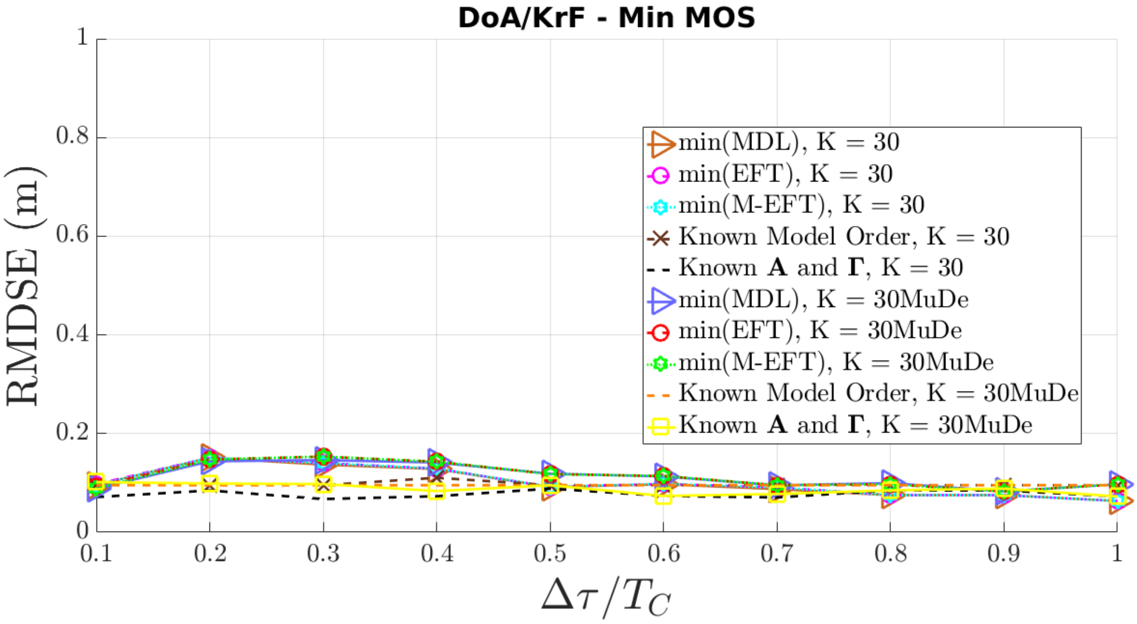


Figure 4.28: MuDe method, minimum estimated model order and state-of-the-art DoA/KRF method simulation with $M = 8$ antennas. In both cases code samples are collected during $K = 30$ epochs, and have $N = 245520$ samples.

# Chapter 5

# Conclusion

Firstly, we propose to utilize the RADOI method to perform the Model Order Selection (MOS) in static scenarios. The RADOI method presented the highest PoD, even when we simulated employing an array with errors in static scenarios. In addition to the RADOI method, we showed that the EFT+FBA+SPS method is suitable for dynamic scenarios. Afterward, we show that we can utilize the EFT+FBA+SPS method to create sub-tensors from tensor $\mathcal{Y}$.

We demonstrate that when we have $d = 1$ satellite, only AIC fails to perform model order estimation. Moreover, when we performed simulations considering $d = 2$ and $d = 3$ the EFT, AIC, and MDL-based methods failed to propoerly estimate the model order. In static scenarios, the RADOI and ESTER methods showed the best performance when considering $d = 2$ and $d = 3$ satellites. However, when assuming an imperfect array of antennas, the ESTER method showed the worst perfomance since it is a subspace-based method.

Furthermore, we simulated a dynamic scenario by varying the DoA of the LOS and NLOS components throughout each epoch and by manipulating the model order of each epoch. Overall, among the matrix-based MOS methods, we show that the EFT+FBA+SPSm M-EFT+FBA+SPS, and MDL+FBA+SPS the highest PoD. Therefore, we selected these three methods to perform TDE. When analysing the PoD, we noticed that even in a dynamic scenario, we can produce reliable measurements when negleting the model order variation.

Next, we defined the Factor Matrices Estimation method. We opted for the Mode 1 HOSVD SECSI method since it is accurate and reliable under demanding scenarios. Then, we remarked that the MuDe successfully denoise the created sub-tensors. Moreover, if we increase the number of dimensions and the size of those dimensions, we would obtain a higher gain. Finally, the proposed framework applied to dynamic scenarios demonstrated similar performance to static scenarios simulations.

# Bibliography

[1] M. S. Braasch and A. Van Dierendonck, "GPS receiver architectures and measurements," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 48–64, 1999.

[2] R. D. Van Nee, "Spread-spectrum code and carrier synchronization errors caused by multipath and interference," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 4, pp. 1359–1365, 1993.

[3] B. W. Parkingson and J. J. Spilker, "Global positioning system: Theory and applications," *American Institute of Aeronautics and Astronautics (AIAA)*, vol. 1, 1996.

[4] B. Hammoud, F. Antreich, J. A. Nossek, J. P. C. L. d. Costa, and A. L. F. d. Almeida, "Tensor-based approach for time-delay estimation," in *WSA 2016; 20th International ITG Workshop on Smart Antennas*, March 2016, pp. 1–7.

[5] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3198–3213, 2008.

[6] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, 1985.

[7] P. R. Gomes, A. L. de Almeida, J. P. C. L. da Costa, J. Mota, D. V. de Lima, and G. Del Galdo, "Tensor-based methods for blind spatial signature estimation in multidimensional sensor arrays," *International Journal of Antennas and Propagation*, vol. 2017, 2017.

[8] D. V. de Lima, J. P. C. L. da Costa, F. Antreich, and G. Del Galdo, "High resolution time-delay estimation via direction of arrival estimation and Khatri-Rao factorization for multipath mitigation," in *Workshop on Smart Antennas (WSA)*, Berlin, 2017.

[9] D. V. de Lima, J. P. C. L. da Costa, F. Antreich, R. K. Miranda, and G. Del Galdo, "Time-Delay estimation via CPD-GEVD applied to tensor-based GNSS arrays with errors," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor*

*Adaptive Processing (CAMSAP) (IEEE CAMSAP 2017)*, Curaçao, Netherlands Antilles, Dec. 2017.

[10] M. R. Zanatta, R. K. Miranda, J. P. C. L. da Costa, F. Antreich, and D. V. de Lima, "Antenna array based receivers for third generation global positioning system," in *Workshop on Communication Networks and Power Systems*, November 2017.

[11] Navstar GPS, "Interface specification IS-GPS-800D," U.S. Air Force, Tech. Rep., 2013. [Online]. Available: http://www.gps.gov/technical/icwg/IS-GPS-800D.pdf

[12] F. D. Côté, I. N. Psaromiligkos, and W. J. Gross, "GNSS modulation: A unified statistical description," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 1814–1836, 2011.

[13] F. Macchi, "Development and testing of an L1 combined GPS-Galileo software receiver," Ph.D. dissertation, University of Calgary, 2010. [Online]. Available: http://www.geomatics.ucalgary.ca/graduatetheses

[14] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, pp. 8–15, 1989.

[15] M. R. Zanatta, F. L. L. de Mendonça, F. Antreich, D. V. de Lima, R. K. Miranda, G. D. Galdo, and J. P. C. L. da Costa, "Tensor-based time-delay estimation for second and third generation global positioning system," *Digital Signal Processing*, vol. 92, pp. 1 – 19, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200418307073

[16] M. R. Zanatta, J. P. C. L. Da Costa, F. Antreich, M. Haardt, G. Elger, F. L. Lopes De Mendonça, and R. T. De Sousa, "Tensor-based framework with model order selection and high accuracy factor decomposition for time-delay estimation in dynamic multipath scenarios," *IEEE Access*, vol. 8, pp. 174 931–174 942, 2020.

[17] D. V. de Lima, M. R. Zanatta, J. P. C. L. da Costa, R. T. de Sousa Jr, and M. Haardt, "Robust tensor-based techniques for antenna array-based GNSS receivers in scenarios with highly correlated multipath components," *Digital Signal Processing*, p. 102715, Mar 2020.

[18] C. C. Garcez, D. V. de Lima, R. K. Miranda, F. Mendonça, J. P. C. L. da Costa, A. L. de Almeida, and R. T. de Sousa Jr, "Tensor-Based Subspace Tracking for Time-Delay Estimation in GNSS Multi-Antenna Receivers," *Sensors*, vol. 19, no. 23, p. 5076, Nov 2019.

[19] C. Garcez, D. Valle, R. Miranda, F. L. Mendonça, J. P. C. L. da Costa, A. de Almeida, and R. de Sousa Junior, "Técnicas Tensoriais Baseadas em Rastreamento de Subespaços

Aplicadas a Receptores GNSS," in *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 01 2019.

[20] G. A. Santos, J. P. C. L. da Costa, D. V. de Lima, M. R. Zanatta, B. J. G. Praciano, G. P. M. Pinheiro, F. L. L. de Mendonça, and R. T. de Sousa, "Improved localization framework for autonomous vehicles via tensor and antenna array based gnss receivers," in *2020 Workshop on Communication Networks and Power Systems (WCNPS)*, 2020, pp. 1–6.

[21] P. A. Lachenbruch and M. Goldstein, "Discriminant analysis," *Biometrics*, vol. 35, no. 1, pp. 69–85, 1979. [Online]. Available: http://www.jstor.org/stable/2529937

[22] J. Grouffaud, P. Larzabal, and H. Clergeot, "Some properties of ordered eigenvalues of a wishart matrix: application in detection test and model order selection," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 5, May 1996, pp. 2463–2466 vol. 5.

[23] P. R. Gomes, J. P. C. da Costa, A. L. de Almeida, and R. T. de Sousa, "Tensor-based multiple denoising via successive spatial smoothing, low-rank approximation and reconstruction for r-d sensor array processing," *Digital Signal Processing*, vol. 89, pp. 1 – 7, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1051200418302847

[24] F. Roemer and M. Haardt, "A closed-form solution for parallel factor (PARAFAC) analysis," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 2365–2368.

[25] ——, "A closed-form solution for multilinear PARAFAC decompositions," in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, July 2008, pp. 487–491.

[26] ——, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)," *Signal Processing*, vol. 93, no. 9, pp. 2722 – 2738, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168413000704

[27] J. P. C. L. da Costa, M. Haardt, F. Romer, and G. D. Galdo, "Enhanced model order estimation using higher-order arrays," in *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, Nov 2007, pp. 412–416.

[28] E. Radoi and A. Quinquis, "A new method for estimating the number of harmonic components in noise with application in high resolution radar," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 1177–1188, Jan. 2004. [Online]. Available: http://dx.doi-org.ez54.periodicos.capes.gov.br/10.1155/S1110865704401097

[29] R. Badeau, B. David, and G. Richard, "Selecting the modeling order for the esprit high resolution method: an alternative approach," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2004, pp. ii–1025.

[30] J. P. C. L. da Costa, *Parameter Estimation Techniques for Multi-Dimensional Array Signal Processing*. Shaker Verlag, 2010.

[31] J. Selva Vera, "Efficient multipath mitigation in navigation systems," Ph.D. dissertation, Universitat Politecnica de Catalunya, 2003.

[32] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000. [Online]. Available: https://doi.org/10.1137/S0895479896305696

[33] R. Roy, A. Paulraj, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques - esprit," in *Military Communications Conference - Communications-Computers: Teamed for the 90's, 1986. MILCOM 1986. IEEE*, vol. 3, Oct 1986, pp. 41.6.1–41.6.5.

[34] F. Roemer and M. Haardt, "Tensor-based channel estimation (TENCE) for two-way relaying with multiple antennas and spatial reuse," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 3641–3644.

[35] J. P. C. L. da Costa, D. Schulz, F. Roemer, M. Haardt, and J. A. Apolinário, "Robust R-D parameter estimation via closed-form PARAFAC in kronecker colored environments," in *2010 7th International Symposium on Wireless Communication Systems*, Sept 2010, pp. 115–119.

[36] I. Domanov and L. D. Lathauwer, "Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 2, pp. 636–660, 2014.

[37] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, May 2006, pp. IV–IV.

[38] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, 1996. [Online]. Available: https://doi.org/10.1137/S0895479893259546

[39] D. V. de Lima, J. P. C. L. da Costa, J. P. A. Maranhão, and R. T. de Souza Jr, "Time-delay estimation via procrustes estimation and Khatri-Rao factorization for GNSS multipath mitigation," in *11th IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Gold Coast, 2017.

[40] M. R. Zanatta, D. V. de Lima, J. P. C. L. da Costa, R. K. Miranda, F. Antreich, and R. T. de Souza Jr., “Técnica Tensorial de Estimação de Atraso para GPS de Segunda e Terceira Geração,” in *XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, Setember 2018.

[41] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[42] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt, “Model order selection for short data: An exponential fitting test (eft),” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 201–201, 2007.

[43] A. Thakre, M. Haardt, F. Roemer, and K. Giridhar, “Tensor-Based Spatial Smoothing (TB-SS) Using Multiple Snapshots,” *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2715–2728, May 2010.

[44] M. Weis, G. Del Galdo, and M. Haardt, “A correlation tensor-based model for time variant frequency selective mimo channels,” in *Proceedings of International ITG/IEEE Workshop on Smart Antennas (WSA’07)*, 2007.

# Appendix A

# Tensor Calculus

## A.1 Notation

Scalar are represented by italic letters ($a$, $b$, $A$, $B$), vector by lowercase bold letters ($\mathbf{a}$, $\mathbf{b}$), matrices by uppercase bold letters ($\mathbf{A}$, $\mathbf{B}$), and tensors by uppercase bold calligraphic letters ($\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$).

The superscripts $^{\mathrm{T}}$, $^{*}$, $^{\mathrm{H}}$, $^{-1}$, and $^{+}$ denote the transpose, conjugate, conjugate transpose (Hermitian), inverse of a matrix, and pseudo-inverse of a matrix, respectively.

For a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$, the element in the $m$-th row and $n$-th column is denoted by $a_{m,n}$, its $m$-th row is denoted by $(\mathbf{A})_{m,\cdot}$, and its $n$-th column is denoted by $(\mathbf{A})_{\cdot,n}$. The 2-norm of a matrix $\mathbf{A}$ is denoted by $||\mathbf{A}||_2$.

For a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ with $M < N$, the $\operatorname{diag}\{\cdot\}$ operator extracts the diagonal:

$$\operatorname{diag}\{\mathbf{A}\} \triangleq \begin{bmatrix} a_{1,1} \\ a_{2,2} \\ \vdots \\ a_{M,M} \end{bmatrix} \tag{A.1}$$

The $n$-th mode unfolding of tensor $\boldsymbol{\mathcal{A}}$ is denoted as $[\boldsymbol{\mathcal{A}}]_(n)$. The $n$-mode product between tensor $\boldsymbol{\mathcal{A}}$ and a matrix $\mathbf{B}$ is represented as $\boldsymbol{\mathcal{A}} \times_n \mathbf{B}$. The $N$-th order identity tensor of size $L \times \cdots \times L$ is denoted by $\boldsymbol{\mathcal{I}}_{N,L}$.

For two $N$-th order tensor $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, both of size $I_1 \times I_2 \times \cdots \times I_N$, composed of individual scalar elementes $a_{i_1,i_2,\ldots,i_N}$ and $b_{i_1,i_2,\ldots,i_N}$, respectively, its inner product is denoted by $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle$, and is defined as

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \triangleq \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} a_{i_1,i_2,\ldots,i_N} b_{i_1,i_2,\ldots,i_N}. \tag{A.2}$$

The norm of a tensor $\boldsymbol{\mathcal{A}}$, denoted by $||\boldsymbol{\mathcal{A}}||_F$, is the Frobenius norm described as

$$||\boldsymbol{\mathcal{A}}||_F \triangleq \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle} \tag{A.3}$$

## A.2 Matrix Calculus

### A.2.1 Kronecker product

Given two matrices $\mathbf{A} \in \mathbb{C}^{I \times J}$ and $\mathbf{B} \in \mathbb{C}^{K \times L}$ their Kronecker product, denote by $\otimes$, is denote as:

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,J}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{I,1}\mathbf{B} & \cdots & a_{I,J}\mathbf{B} \end{bmatrix} \in \mathbb{C}^{IK \times JL}. \tag{A.4}$$

### A.2.2 Khatri-Rao product

Given two matrices $\mathbf{A} \in \mathbb{C}^{I \times R}$ and $\mathbf{B} \in \mathbb{C}^{K \times R}$ their Khatri-Rao product, denote by $\diamond$, is denote as:

$$\mathbf{A} \diamond \mathbf{B} \triangleq \Big[ (\mathbf{A})_{\cdot,1} \otimes (\mathbf{B})_{\cdot,1} \cdots (\mathbf{A})_{\cdot,R} \otimes (\mathbf{B})_{\cdot,R} \Big] \in \mathbb{C}^{IJ \times R}. \tag{A.5}$$

### A.2.3 Outer product

The outer product is a special case of the Kronecker product where the outer product of two vectors $\mathbf{a} \in \mathbb{C}^{I}$ and $\mathbf{b} \in \mathbb{C}^{J}$ results in a matrix $\mathbf{C} \in \mathbb{C}^{I \times J}$

$$\begin{aligned} \mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^{\mathrm{T}} &= \begin{bmatrix} a_1 \\ \vdots \\ a_I \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_J \end{bmatrix} \\ &= \begin{bmatrix} a_1 b_1 & \cdots & a_1 b_J \\ \vdots & \ddots & \vdots \\ a_I b_1 & \cdots & a_I b_J \end{bmatrix} = \mathbf{C} \in \mathbb{C}^{I \times J}, \end{aligned} \tag{A.6}$$

thus the elements of $\mathbf{C}$ satisfy $c_{ij} = a_i b_j, i \in 1, \ldots, I, j \in 1, \ldots, J$.

Once the outer product can be extended into other dimensions, an outer product of three vectors $\mathbf{a} \in \mathbb{C}^I$, $\mathbf{b} \in \mathbb{C}^J$, and $\mathbf{c} \in \mathbb{C}^K$ results in a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I \times J \times K}$

$$\mathbf{a} \circ \mathbf{b} \circ \mathbf{c} = \boldsymbol{\mathcal{X}}, \tag{A.7}$$

and $x_{ijk} = a_i b_j c_k, i \in 1, \ldots, I, j \in 1, \ldots, J, k \in 1, \ldots, K$.

### A.2.4 The $\mathrm{vec}\{\cdot\}$ operator

The $\mathrm{vec}\{\cdot\}$ operator reshapes a matrix into a vector in such a manner that its vectors are stacked. For instance, for a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$

$$\begin{aligned} \mathrm{vec}\{\mathbf{A}\} &= \mathrm{vec}\{[\mathbf{A}_1 \quad \cdots \quad \mathbf{A}_N]\} \\ &= \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \in \mathbb{C}^{MN}. \end{aligned} \tag{A.8}$$

An importante property of $\mathrm{vec}\{\cdot\}$ operator is that for $\mathbf{X} = \mathbf{ABC}$ with $\mathbf{A} \in \mathbb{C}^{I \times J}$, a diagonal matrix $\mathbf{B} \in \mathbb{C}^{J \times J}$, and $\mathbf{C} \in \mathbb{C}^{J \times K}$

$$\begin{aligned} \mathrm{vec}\{\mathbf{X}\} &= \mathrm{vec}\{\mathbf{ABC}\} \\ &= (\mathbf{C}^{\mathrm{T}} \diamond \mathbf{A})\, \mathrm{diag}\{\mathbf{B}\} \in \mathbb{C}^{IK}. \end{aligned} \tag{A.9}$$

### A.2.5 The $\mathrm{unvec}\,\{\cdot\}$ operator

The $\mathrm{unvec}\cdot$ operator reshapes a vector into a matrix of determined size. Thus, for a vector $\mathbf{a} = [\mathbf{a}_1^{\mathrm{T}}, \ldots, \mathbf{a}_N^{\mathrm{T}}] \in \mathbb{C}^{MN}$

$$\begin{aligned} \underset{M \times N}{\mathrm{unvec}}\,\{\mathbf{a}\} &= \underset{M \times N}{\mathrm{unvec}} \left\{ \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \right\} \\ &= \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_N \end{bmatrix} \end{aligned} \tag{A.10}$$

## A.3 Tensor Calculus

### A.3.1 Tensors

Vectors are generalizations of scalars, and matrices are generalizations of vectors, tensors are generalizations of matrices. However, matrices are limited to two dimensions while tensors can have any number of dimensions. Therefore, we apply the terms scalar, vector and matrix to 0-, 1-, and 2-dimensional structures. Thus, we reserve the term tensor to structures with 3 or more dimensions.

In (A.11) we examplify a scalar $I \in \mathbb{C}$, a vector $\mathbf{i} \in \mathbb{C}^3$, and an identity matrix $\mathbf{I} \in \mathbb{C}^{3\times 3}$

$$I = 1, \qquad \mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{I} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \tag{A.11}$$

while in (A.12) a third order tensor $\boldsymbol{\mathcal{I}}_{3,3} \in \mathbb{C}^{3\times 3\times 3}$

$$\boldsymbol{\mathcal{I}}_{3,3} \quad = \quad \begin{array}{ccc} & & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & & \end{array} \tag{A.12}$$

Once higher-order tensors can be created, visualization becomes difficult. A $N$-dimensional tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ can be seen in "slices" by maintaining its first two indexes fixed while varying the remaining $N-2$ indexes. For instance, by changing the third index of the third-order identity tensor from (A.12) while fixing the first and second indexes we have

$$(\boldsymbol{\mathcal{I}}_{3,3})_{\cdot,\cdot,1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, (\boldsymbol{\mathcal{I}}_{3,3})_{\cdot,\cdot,2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, (\boldsymbol{\mathcal{I}}_{3,3})_{\cdot,\cdot,3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.13}$$

### A.3.2 $n$-mode unfolding

Once tensor representation can be complex to understand, the $n$-mode unfolding provides a way to represent a tensor as a matrix. This is done by fixing the $n$-th index while varying the other indexes in reverse order, concatenating these vectors along the $n+1$-th dimension, then permuting the order of the dimensions from the $n$-th to the $n-1$-th dimensions. For instance, for a third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{2\times 2 \times \times 2}$ we can write

$$\boldsymbol{\mathcal{A}} \quad = \quad \begin{array}{cc} & \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & \end{array} \tag{A.14}$$

then $\boldsymbol{\mathcal{A}}$ has the following unfoldings:

$$[\boldsymbol{\mathcal{A}}]_{(1)} = \begin{bmatrix} 1 & 5 & 2 & 6 \\ 3 & 7 & 4 & 8 \end{bmatrix}, \tag{A.15}$$

$$[\boldsymbol{\mathcal{A}}]_{(2)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}, \tag{A.16}$$

$$[\boldsymbol{\mathcal{A}}]_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}. \tag{A.17}$$

For a $N$-dimensional tensor, $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$, its $n$-mode unfolding, $[\boldsymbol{\mathcal{A}}]_{(n)}$, will be of size $I_n \times \Pi_{r \neq n} I_r$.

### A.3.3 $n$-mode product

The $n$-mode product permits the calculation of the product of a matrix and a tensor by utilizing the $n$-mode unfolding. For instance, for an $N$-dimensional tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and a matrix $\mathbb{B} \in \mathbb{C}^{M \times I_n}$, the $n$-mode product is then denoted as $\boldsymbol{\mathcal{A}} \times_n \mathbf{B}$. Therefore, we have the matrix product $\mathbf{B} \cdot [\boldsymbol{\mathcal{A}}]_{(n)}$ folded back into a tensor of size $I_1 \times \cdots \times M \times \cdots \times I_N$.

### A.3.4 PARAFAC model

The PARAFAC model assumes that a given $N$-dimensional tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ can be decomposed into a summation of a minimum number of rank-one tensors $\boldsymbol{\mathcal{X}}^{(i)}, i = 1, \ldots, L$.

$$\boldsymbol{\mathcal{X}} = \sum_{\ell=1}^{L} \boldsymbol{\mathcal{X}}^{(\ell)} = \sum_{\ell=1}^{L} \mathbf{a}_\ell^{(1)} \circ \cdots \circ \mathbf{a}_\ell^{(N)}, \tag{A.18}$$

where $L$ is the model order of the noiseless tensor.

Through defining factor matrices $\mathbf{A}^{(i)} = [\mathbf{a}_1^{(i)}, \ldots, \mathbf{a}_1^{(i)}]$, the Equation (A.18) can be denoted in terms of the $n$-mode product of an $N$-dimensional identity tensor $\boldsymbol{\mathcal{I}}_{N,L} \in \mathbb{R}^{L\times\cdots\times L}$ and loading matrices $\mathbf{A}^{(i)}$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{N,L} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}. \tag{A.19}$$

Moreover, a useful property for a third-order tensor $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{I}}_{N,3} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \in \mathbb{C}^{I_1\times I_2\times I_3}$ with $\mathbf{A}^{(1)} \in \mathbb{C}^{I_1\times d}$, $\mathbf{A}^{(2)} \in \mathbb{C}^{I_2\times d}$, and $\mathbf{A}^{(3)} \in \mathbb{C}^{I_3\times d}$, its unfolding are:

$$[\boldsymbol{\mathcal{A}}]_{(1)} = \mathbf{A}^{(1)}(\mathbf{A}^{(2)} \diamond \mathbf{A}^{(3)})^{\mathrm{T}} \in \mathbb{C}^{I_1\times I_2 I_3}, \tag{A.20}$$

$$[\boldsymbol{\mathcal{A}}]_{(2)} = \mathbf{A}^{(2)}(\mathbf{A}^{(3)} \diamond \mathbf{A}^{(1)})^{\mathrm{T}} \in \mathbb{C}^{I_2\times I_3 I_1}, \tag{A.21}$$

$$[\boldsymbol{\mathcal{A}}]_{(3)} = \mathbf{A}^{(3)}(\mathbf{A}^{(1)} \diamond \mathbf{A}^{(2)})^{\mathrm{T}} \in \mathbb{C}^{I_3\times I_1 I_2}. \tag{A.22}$$

### A.3.5 Higher-Order SVD

The Singular Value Decomposition (SVD) decomposes a given matrix $\mathbf{X} \in \mathbb{C}^{I_1\times I_2}$ into the following fashion:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}}, \tag{A.23}$$

where $\mathbf{U} \in \mathbb{C}^{I_1\times I_1}$ is a unitary matrix holding the left-hand singular vectors and its columns are related to the column space of $\mathbf{X}$, $\boldsymbol{\Sigma} \in \mathbb{C}^{I_1\times I_2}$ is a matrix containing the singular values $\sigma_1, \sigma_s, \ldots, \sigma_{\min(I_1,I_2)}$ in its diagonal, and $\mathbf{V} \in \mathbb{C}^{I_2\times I_2}$ is a unitary matrix holding the right-hand singular vectors and its rows are related to the row space of $\mathbf{X}$.

In order to generalize the SVD to an $N$-th order tensor the $n$-mode product previously described can be applied

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{H}} \\ &= \boldsymbol{\Sigma} \times_1 \mathbf{U} \times_1 \mathbf{V}^* \\ &= \boldsymbol{\Sigma} \times_1 \mathbf{U}^{(1)} \times_1 \mathbf{U}^{(2)}, \end{aligned} \tag{A.24}$$

where $\mathbf{U}^{(1)} = \mathbf{U}$, and $\mathbf{U}^{(2)} = \mathbf{V}^*$. Also, $\boldsymbol{\Sigma}$ have the properties of pseudodiogonality, only its diagonal is non-zero, and ordering, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(I_1,I_2)} \geq 0$. Therefore, we can generalize the SVD to an $N$-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{I_1} \times \mathbb{I_2} \times \cdots \times \mathbb{I_N}$ in terms of an $n$-mode product

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{R}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}, \tag{A.25}$$

where the tensor $\boldsymbol{\mathcal{R}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ is the core tensor and has the property of all-orthogonality, which mean that for two subtensors $\boldsymbol{\mathcal{R}}_{i_n=\alpha}$ and $\boldsymbol{\mathcal{R}}_{i_n=\beta}$ constructed by keeping the index $i_n$ fixed, their inner product $\langle \boldsymbol{\mathcal{R}}_{i_n=\alpha}, \boldsymbol{\mathcal{R}}_{i_n=\beta} \rangle = 0$ for $\alpha \neq \beta$, and ordering.

Finally, the calculation of the HOSVD can be achieved by finding each unitary left singular vector matrix $\mathbf{U}^{(n)}$ from its respective $n$-mode unfolding by applying the SVD to $[\boldsymbol{\mathcal{X}}]_{(n)}$. Then, the core tensor $\boldsymbol{\mathcal{R}}$ can be computed by applying (A.25)

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}. \tag{A.26}$$

### A.3.6 Dual-symmetric tensors

A $2N$-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{I_1} \times \cdots \times \mathbb{I_N} \times \mathbb{I_{N+1}} \times \ldots \mathbb{I_{2N}}$ is dual-symmetric if and only if there can be a permutation of indexes $P$, resulting in a tensor $\boldsymbol{\mathcal{X}}_P$ which follows the particular PARAFAC Decomposition [34]

$$\boldsymbol{\mathcal{X}}_P = \boldsymbol{\mathcal{I}}_{2N,L} \times_1 \mathbf{A}_1^{(1)} \cdots \times_N \mathbf{A}_{N+1}^{(N)} (\mathbf{A}^{(1)})^* \cdots \times_{2N} (\mathbf{A}^{(N)})^*. \tag{A.27}$$

The dual-symmetric decomposition is useful in signal processing once every correlation tensor follows this decomposition [44]. In order to harness the relation in (A.27), a particular unfolding known as Hermitian-symmetric unfolding [5] applied to a dual-simmetric tensor. The Hermitian-simmetric unfolding of $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \cdots \times I_{2N}}$, $\mathbf{X}_\mathrm{H}$ is defined as:

$$\mathbf{X}_\mathrm{H} = \underset{K \times K}{\mathrm{unvec}} \{\mathrm{vec}\{\boldsymbol{\mathcal{X}}\}\} \in \mathbb{C}^{K \times K}, \tag{A.28}$$

with $K = I_1, \ldots, I_N$. Therefore, the unfolding can be represented in terms of its factor matrices as:

$$\mathbf{X}_\mathrm{H} = (\mathbf{A}^{(N)} \diamond \cdots \diamond \mathbf{A}^{(1)})(\mathbf{A}^{(N)} \diamond \cdots \diamond \mathbf{A}^{(N)})^\mathrm{H}. \tag{A.29}$$