TIME-DELAY ESTIMATION UNDER NON-CLUSTERED AND CLUSTERED SCENARIOS FOR GNSS SIGNALS

DANIEL VALLE DE LIMA

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

TIME-DELAY ESTIMATION UNDER NON-CLUSTERED AND CLUSTERED SCENARIOS FOR GNSS SIGNALS

DANIEL VALLE DE LIMA

Orientador: PROF. DR.-ING. JOÃO PAULO CARVALHO LUSTOSA DA COSTA, ENE/UNB E HSHL
Co-orientador: PROF. DR. RAFAEL TIMÓTEO DE SOUSA JR., ENE/UNB

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.TD - 177/21
BRASÍLIA-DF, 16 DE ABRIL DE 2021.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

TIME-DELAY ESTIMATION UNDER NON-CLUSTERED AND CLUSTERED SCENARIOS FOR GNSS SIGNALS

DANIEL VALLE DE LIMA

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. Rafael Timóteo de Sousa Jr., ENE/UnB
Presidente

Prof. Dr. Leonardo Rodrigues Araújo Xavier de Menezes, ENE/UnB
Examinador interno

Prof. Dr. Carlos Alexandre Rolim Fernandes, PPGETI/UFC
Examinador externo

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

Prof. Dr. Adoniran Judson de Barros Braga, ENE/UnB
Suplente

BRASÍLIA, 16 DE ABRIL DE 2021.

FICHA CATALOGRÁFICA

DE LIMA, DANIEL VALLE

Time-Delay Estimation Under Non-clustered and Clustered Scenarios for GNSS Signals [Distrito Federal] 2021.

xvi, 161 p., 201×297 mm (ENE/FT/UnB, Doutor, Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia.)

Departamento de Engenharia Elétrica

1. Sistema Global de Navegação por Satélites 2. Tensores

3. Processamento de Sinais 4. Processamento Adaptativo

I. ENE/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

DE LIMA, D. V. (2021). Time-Delay Estimation Under Non-clustered and Clustered Scenarios for GNSS Signals. Tese de Doutorado em Engenharia Elétrica, Publicação 177/21, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 161p.

CESSÃO DE DIREITOS

AUTOR: Daniel Valle de Lima

TÍTULO: Time-Delay Estimation Under Non-clustered and Clustered Scenarios for GNSS Signals.

GRAU: Doutor ANO: 2021

_____

Daniel Valle de Lima

QL 7 Conj. 3 Casa 2, Setor de Habitações Individuais Norte.

71515-035 Brasília – DF – Brasil.

Agradecimentos

Quero agradecer todas as pessoas me ajudaram a gerar este trabalho.

Agradeço meu orientador, Prof. João Paulo Carvalho Lustosa da Costa, pelas aulas, ajuda, críticas, comentários, sugestões, e orientação, assim como meu coorientador, Prof. Rafael Timóteo de Sousa Jr.

Sou grato também aos professores com quem tive aulas, Francisco Assis, André Noll Barreto, Marcelo Menezes de Carvalho, Geovany Araújo Borges, e Daniel Guerreiro e Silva. Agradeço também os professores Adoniran Judson de Barros Braga, Carlos Alexandre Rolim Fernandes, Felix Dieter Antreich, Leonardo Rodrigues Araújo Xavier de Menezes, e Ricardo Zelenovsky pelos críticas, perguntas, comentários e sugestões que ajudaram a melhorar a primeira versão desta tese. Assim como os professores Lieven de Lathauwer e Martin Haardt pelas suas contribuições técnicas.

Agradeço também a todos os membros da secretaria de engenharia elétrica pelo trabalho deles.

Finalmente, sou grato pelo meus pais, que me apoiaram e sempre acreditaram em mim, inclusive naqueles momentos que eu não acreditei em mim mesmo.

Dedicatória

Para minha mãe.

Resumo

Aplicações que empregam sistemas globais de navegação por satélite, do inglês *Global Navigation Satellite Systems* (GNSS) para prover posicionamento acurado estão sujeitos a degradação drástica não só por intereferências eletromagnéticas, como também componentes de multipercurso causados por reflexões e refrações no ambiente. Aplicações de segurança crítica como veículos autonômos e aviação civil, e aplicações de risco crítico como gestão de pesca, pedágio automático, e agricultura de precisão dependem de posicionamento acurado sob cenários complicados. Tipicamente quanto mais agrupamento ocorre entre o componente de linha de visada, do inglês *line-of-sight* (LOS) e componentes de multipercurso ou não-linha de visada, do inglês *non-line-of-sight* (NLOS), menos acurada é a estimação da posição. Abordagens tensorials estado da arte para receptores GNSS baseado em arranjos de antenas utilizam processamento tensorial de sinais para separar o componente LOS dos componentes NLOS, assim mitigando os efeitos destes, utilizando decomposição em valores singulares multilinear, do inglês *multilinear singular value decomposition* (MLSVD) para gerar um autofiltro de order superior, do inglês *higher-order eigenfilter* (HOE) com pré-processamento por média frente-costas, do inglês *forward-backward averaging* (FBA), e suavização espacial expandida, do inglês *expanded spatial smoothing* (ESPS), estimação de direção de chegada, do inglês *direction of arrival* (DoA) e fatorização Khatri-Rao, do inglês *Khatri-Rao factorization* (KRF), estimação de Procrustes e fatorização Khatri-Rao (ProKRaft), e o sistema semi-algébrico de decomposição poliádica canônica por diagonalização matricial simultânea, do inglês semi-algebraic framework for approximate canonical polyadic decomposition via simultaneous matrix diagonalization (SECSI), respectivamente. Propomos duas abordagens de processamento para estimação de atraso, do inglês *time-delay estimation* (TDE). A primeira é a abordagem em lotes utilizando dados de vários períodos do sinal. Usando estimação em lotes propomos duas abordagens algébricas para TDE, em que diagonalizaçao é efetivada por decomposição generalizada em autovalores, do inglês *generalized eigenvalue decomposition* (GEVD), das primeiras duas fatias frontais do tensor núcleo do tensor de dados, estimado por MLSVD. Esta primeira abordagem, como os métodos citados, na quais simulações foram feitas com 1 componente LOS e 1 componente NLOS, assim os dados observados tem posto cheio em todos seus modos, não faz suposições sobre o posto do tensor de dados. A segunda abordagem supõe cenários nos quais mais de 1 componente NLOS está presente e são agregados (*clustered* em inglês), assim vários vetores de uma das matrizes-fator que formam o tensor de dados são altamente correlaciona-

dos, resultando num tensor de dados que é de posto deficiente em pelo menos um modo. Os esquemas algébricos baseados em tensores propostos utilizam a decomposição poliádica canônica por decomposição generalizada em autovalores, do inglês *canonical polyadic decomposition via generalized eigenvalue decomposition* (CPD-GEVD), e a decomposição em termos de posto-$(L_r, L_r, 1)$ por decomposição generalizada em autovalores, do inglês *decomposition in multilinear rank-*$(L_r, L_r, 1)$ *terms via generalized eigenvalue decomposition* ($(L_r, L_r, 1)$-GEVD) para melhorar a TDE do componente LOS sob cenários desafiadores. A segunda é a abordagem de processamento adaptativo de amostras individuais utilizando rastreamento de subespaço a cada período de código, *epoch* em inglês. Usando processamento adaptativo propomos duas abordagem, uma aplicando FBA expandido (EFBA) e ESPS ao dados e estimando um HOE, e outra usando usa estimação paramétrica para estimar a DoA. Estendendo o modelo para um arranjo retangular uniforme, do inglês *uniform rectangular array* (URA), o fluxo de dados são tensores de terceira ordem. Para este modelo propomos três abordagens para TDE baseado em HOE, CPD-GEVD, e ESPRIT tensorial, respectivamente e empregando uma estratégia de truncamento sequencial para reduzir a quantidade de operações necessárias para cada modo do tensor.

Abstract

Applications employing Global Navigation Satellite Systems (GNSS) to provide accurate positioning are subject to drastic degradation not only due to electromagnetic interference, but also due to multipath components caused by reflections and refractions in the environment. Safety-critical applications such as autonomous vehicles and civil aviation, and liability-critical applications such as fisheries management, automatic tolling, and precision agriculture depend on accurate positioning under such demanding scenarios. Typically, the more clustering occurs between the line-of-sight (LOS) component and multipath or non-line-of-sight (NLOS) components, the more inaccurate is the estimation of the positioning. State-of-the-art tensor based approaches for antenna array-based GNSS receivers apply tensor-based signal processing to separate the LOS components from NLOS components, thus mitigating the effects of the latter, using the multilinear singular value decomposition (MLSVD) to generate a higher-order eigenfilter (HOE) with forward-backward averaging (FBA) and expanded spatial smoothing (ESPS) preprocessing, direction of arrival (DoA) estimation and Khatri-Rao factorization (KRF), Procrustes estimation and Khatri-Rao factorization (ProKRaft), and the semi-algebraic framework for approximate canonical polyadic decomposition via simultaneous matrix diagonalization (SECSI), respectively. These approaches use filtering, parameter estimation and filtering, iterative algebraic factor matrix estimation and filtering, and algebraic factor matrix estimation, respectively. We propose two processing approaches to time-delay estimation (TDE). The first is batch processing taking data from several signal periods. Using batch processing we propose two algebraic approaches to TDE, in which diagonalization is achieved using the generalized eigenvalue decomposition (GEVD) of the first two frontal slices of the measurement tensor's core tensor, estimated via MLSVD. The former approach, like the cited methods, in which simulations were performed with 1 LOS component and 1 NLOS component, and thus the measured data has full-rank tensor in all its modes, makes no assumption about the rank of the measurement tensor. The latter approach assumes scenarios in which more than 1 NLOS component is present and these are clustered, thus several vectors of one of the factor matrices which forms the tensor data are highly correlated, resulting in a rank-deficient measurement tensor in at least one mode. These proposed algebraic tensor-based schemes utilize the canonical polyadic decomposition via generalized eigenvalue decomposition (CPD-GEVD) and the decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via generalized eigenvalue decomposition ($(L_r, L_r, 1)$-GEVD) in order to improve the TDE of the LOS component in challenging sce-

narios. The second approach is adaptive processing of individual samples utilizing subspace tracking to iteratively estimate the subspace at each epoch. Using adaptive processing we propose two approaches, one applying FBA and ESPS to the data and estimating a higher-order eigenfilter, and the other using a parametric approach using DoA estimation. By extending the data model for an uniform rectangular array, we have a data stream of third-order tensors. For this model we propose three approaches to TDE based on HOE, CPD-GEVD, and standard tensor ESPRIT, respectively and employing a sequential truncation strategy to reduce the amount of operations necessary for each tensor mode.

CONTENTS

LIST OF FIGURES

List of Terms

$(L_r, L_r, 1)$**-GEVD** decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via generalized eigenvalue decomposition.

BPSK binary phase-shift keying.

BTD block term decomposition.

C/A coarse acquisition.

CDMA code division multiple access.

CPD canonical polyadic decomposition.

CPD-GEVD canonical polyadic decomposition via generalized eigenvalue decomposition.

CSI cubic spline interpolation.

DoA direction of arrival.

DoA/KRF direction of arrival and Khatri-Rao factorization.

DSSS direct sequence spread spectrum.

EFBA expanded forward-backward averaging.

ESPRIT estimation of parameters via rotational invariance technique.

ESPS expanded spatial smoothing.

FAPI fast approximate power iteration.

FBA forward-backward averaging.

FLOPS floating point operations.

GEVD generalized eigenvalue decomposition.

GLONASS *global´naya navigatsionnaya sputnikova sistema*.

GNSS global navigation satellite system.

GPS global positioning system.

GYAST generalized yet another subspace tracker.

HOE higher-order eigenfilter.

iSVD iterative singular value decomposition.

KRF Khatri-Rao factorization.

LMLRA low multilinear rank approximation.

LOS line-of-sight.

LSKRF least-squares Khatri-Rao factorization.

MAE mean absolute error.

MLSVD multilinear singular value decomposition.

NLOS non-line-of-sight.

OPP orthogonal Procrustes problem.

PARAFAC parallel factors.

PAST projection approximation subspace tracking.

PRN pseudo-random noise.

ProKRaft Procrustes estimation and Khatri-Rao factorization.

PRS pseudo-random sequence.

PSD power spectral density.

rhs right-hand side.

RLS recursive least-squares.

RMS root-means squared.

RMSE root-means squared error.

SECSI semi-algebraic framework for computation of approximate canonical polyadic decompositions via simulataneous matrix diagonalization.

SMR signal-to-multipath ratio.

SNR signal-to-noise ratio.

SPS spatial smoothing.

ST subspace tracking.

SVD singular value decomposition.

TDE time-delay estimation.

ULA uniform linear array.

URA uniform rectangular array.

XOR exclusive or.

Chapter 1

Introduction

In this chapter we present a brief overview of global navigation satellite system (GNSS) and some of the concepts involved in time-delay estimation in Section 1.1. The notation used through and tensor algebra is presented in Section 1.2.

1.1 Global Navigation Satellite Systems

GNSS receivers combine ephemeris data and timing from several satellites to accurately estimate their positions. Accurate TDE is essential for safety-critical applications such as autonomous vehicles [1, 2] and civil aviation [3], and liability-critical applications such as such as fisheries management [4, 5], automatic tolling [6], and precision agriculture [7]. Since errors are proportional to the speed of light, small degradations can be decisive to applications which risk loss of life or property.

Code division multiple access (CDMA) exploiting direct sequence spread spectrum (DSSS) [8] allows long distance data transmission and separation of each satellite using its own pseudo-random sequence (PRS). In this sense, since the same frequency band is shared by the satellites, the time-delay estimation is based on the autocorrelation function of each known PRS [9].

Accordingly, the line-of-sight (LOS) signal from each satellite is separated from the signal from the other satellites using DSSS. However, the NLOS components caused by reflections and refractions in the environment are pernicious, since they have the same PRS of the LOS signal [10–12]. Single-antenna state-of-the-art GNSS receivers suffer significant degradation in time-delay estimation under multipath propagation [13, 14]. One of the motivations of this work is filtering and mitigating the effect of NLOS components using tensor techniques before applying time-delay estimation.

The use of tensor techniques for signal processing is desirable because tensors provide a more "natural" fit to the received data, e.g. signals from a multidimensional array, a tensor has greater identifiability because its (tensor) rank can be greater than its multilinear ranks,

factor matrices can be separated from one another without imposing additional restrictions, and "tensor gain" resulting from exploiting its multidimensional structure.

In scenarios with non-clustered NLOS components, traditional antenna array-based signal processing techniques can significantly mitigate the degradation. Particularly robust solutions can be accomplished by using a tensor-based receiver that combines data from multiple sensors and sampling periods. By applying multilinear algebra to filter the desired LOS components of the multiple satellites, we can facilitate TDE [15, 16].

There are several state-of-the-art tensor based approaches for antenna array-based GNSS receivers considering different scenarios and requirements. One such approach is based on the HOE [16], which creates a filter based on the rank-1 multilinear singular value decomposition (MLSVD) [17, 18]. This approach has low computational complexity and does not require knowledge of the model order but has low precision subject to a bell-shaped performance curve depending on LOS/NLOS correlation. HOE-based filtering can also be augmented by applying pre-processing techniques such as FBA [19] and ESPS [20, 21].

Another approach is based on parameter estimation using estimation of parameters via rotational invariance technique (ESPRIT) [22] to approximate the array response factor matrix, followed by least-squares Khatri-Rao factorization (LSKRF) [23] to estimate the code factor matrix from which the time-delay can be estimated [24]. This approach has consistent performance but requires previous knowledge of the model order [25–27] and can suffer degradation under realistic conditions such as a poorly-calibrated receiver array [28].

Another approach estimates the tensor factor matrices by repeatedly applying a known solution to the orthogonal Procrustes problem (OPP) [29], followed by LSKRF (or another Khatri-Rao factorization approach such as Average Vector and Hadamard Ratio Rank One Approximation [30]) to iteratively refine [31] estimates of the array response and complex amplitude factor matrices [32]. This approach has slightly better performance but one of its assumptions is invalidated if more than one NLOS component is present, degrading its performance.

Other state-of-the-art approaches are based on algebraic estimation of the factor matrices, such as CPD-GEVD [33] and the SECSI [34]. These approaches directly estimate the code factor matrix to perform time-delay estimation [28, 35, 36] and thus are robust against array imperfections but require prior knowledge of the model order, can have high computational cost, and are vulnerable to rank deficiencies in the factor matrices when NLOS component are clustered [37].

Another algebraic approach assumes a particular type of block term decomposition (BTD) [38] known as $(L_r, L_r, 1)$-term decomposition [39], in which multipath components cluster together [40] and share similar characteristics, resulting in a rank deficiency in one of its factor matrices. This is resolved with a more balanced variant of the CPD-GEVD which also determines how these components are clustered [37].

1.1.1 Ranging

Ranging is the act of measuring distance between two points. It can be performed directly by measuring the length between two points as well as indirectly by measuring another parameter from which the range can be derived. A well-known example of this is measuring the difference in time between lightning and thunder to estimate the distance to the observer.

When lightning strikes, we can see the light emitted from it and the time it takes to reach the observer is

$$t_{\text{light}} = \frac{d}{c} + \epsilon,$$

where d is the distance to the observer, c is the speed of light, and ϵ is a measurement error. After a delay, we hear the thunder, and the time it takes to reach the observer is

$$t_{\text{sound}} = \frac{d}{v_{\text{sound}}} + \epsilon$$

where v_{sound} is the speed of sound and ϵ is the measurement error, assumed here to be the same as before.

By taking the difference between t_{light} and t_{sound} we can estimate the time-delay between the observer and the lightning strike as

$$\begin{aligned}\Delta t &= t_{\text{sound}} - t_{\text{light}} \\ &= \frac{d}{v_{\text{sound}}} + \epsilon - \left(\frac{d}{c} + \epsilon\right) \\ &= \frac{d}{v_{\text{sound}}} - \frac{d}{c}.\end{aligned}$$

Since $d \ll c$, $\Delta t \approx t_{\text{sound}}$ and we can range the distance to target

$$\hat{d} = v_{\text{sound}} \cdot \Delta t.$$

This introduces the concept of time-delay estimation and ranging.

1.1.2 Positioning

Positioning consists of determining (relative) position. An excellent example of two-dimensional positioning is presented in [41] and is partially recreated here to introduce the concept of positioning using time-delay estimation.

This example begins by introducing ranging. A seaman is attempting to determine his vessel's position using a foghorn whistle that sounds every minute. It is assumed both the vessel and the foghorn clocks are synchronized. The seaman measures the time-delay

elapsed from the minute until he hears the whistle. With only one foghorn, the seaman can only estimate his range. This is shown in Figure 1.1.

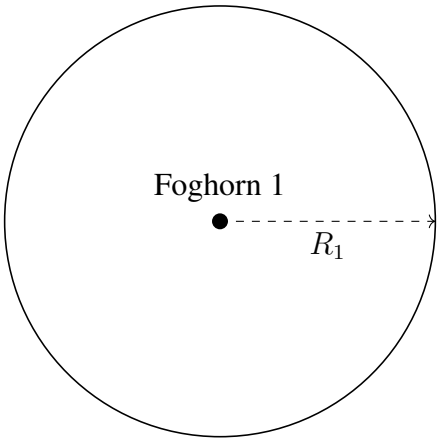

Figure 1.1: Ranging with one source.

So far the vessel's position can only be narrowed down to the circle of R_1 radius with the foghorn in its center. By simultaneously measuring the time-delay of two foghorns, it is possible to estimate two possible positions A and B at the intersections of the circle radii R_1 and R_2 of the two foghorns. This is shown in Figure 1.2.

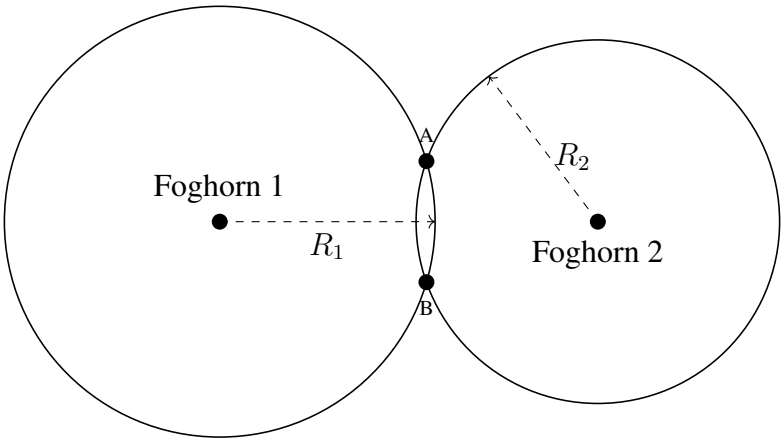

Figure 1.2: Ranging with two sources.

To resolve this ambiguity a third time-delay measurement would be required. This is shown in Figure 1.3.

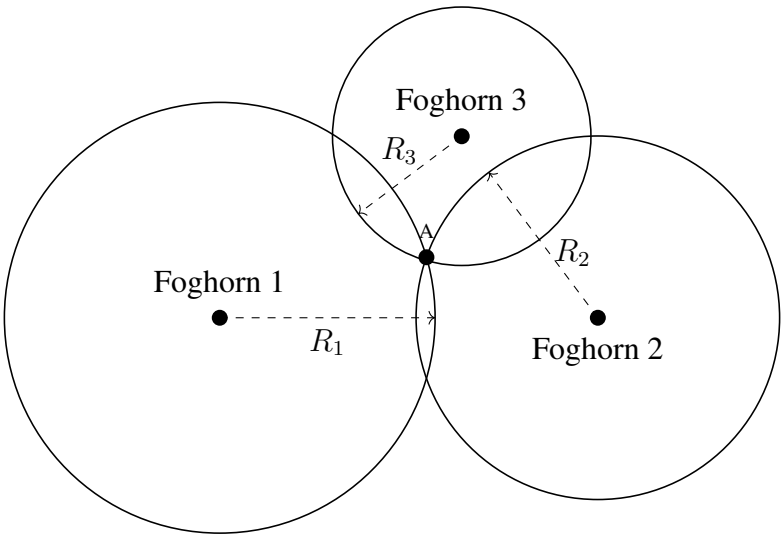

Figure 1.3: Ranging with three sources.

With three time-delay measurements, three circle radii can be estimated, and two-dimensional positioning can be performed without ambiguity. Positioning as performed by a GNSS functions in a similar manner. Satellites are used instead of foghorns and positioning is performed in three dimensions by intersecting spheres rather than circles.

It possible for a GNSS to perform three-dimensional positioning (on Earth) using only three satellites despite there being an intersection ambiguity as this is easily resolvable, unlike in Figure 1.2, since one candidate position will be located on Earth while the other one will be located in outer space. This is, however, not done in practice because GNSS client receivers employ low-cost crystal oscillator clocks which deviate from satellite clocks. Thus measurements from four or more satellites are used in to accurately determine timing and position. Because of this, real-world GNSS have satellite constellations with orbits designed to keep at least 6 satellites in view of the user at all times [42].

1.1.3 GNSS signals

GNSS signals are designed to deliver a data payload, which includes ephemeris data containing information on the position and velocity of each satellite, and separation and identifiction of each satellite's signal using CDMA [43] (and frequency division multiple access in the case of *global ´naya navigatsionnaya sputnikova sistema* (GLONASS) [44]), a DSSS technique.

In Figure 1.4, the data message D is spread using a spreading code C by applying an exclusive or (XOR) operation, denoted by $\oplus$, resulting in $C \oplus D$, which is then modulated by a binary phase-shift keying (BPSK) carrier.

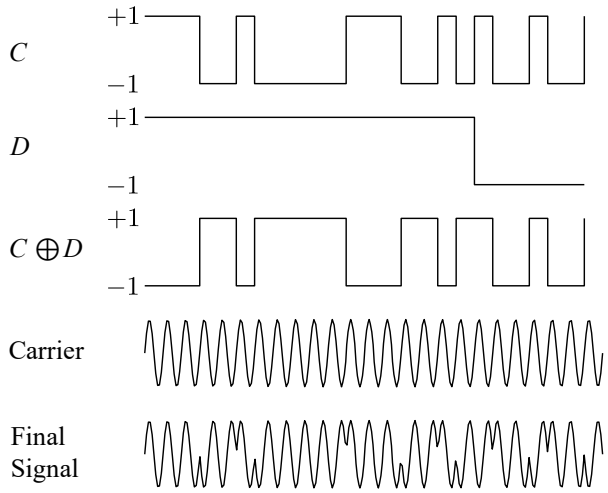

Figure 1.4: GNSS signal example.

CDMA not only allows the satellites to be individually identified, it also spreads power spectrally, as can be seen in the PSD in Figure 1.5 where the message $m(t)$ is spread using code $c(t)$. Spreading also allows correlation-based time-delay estimation to be performed.

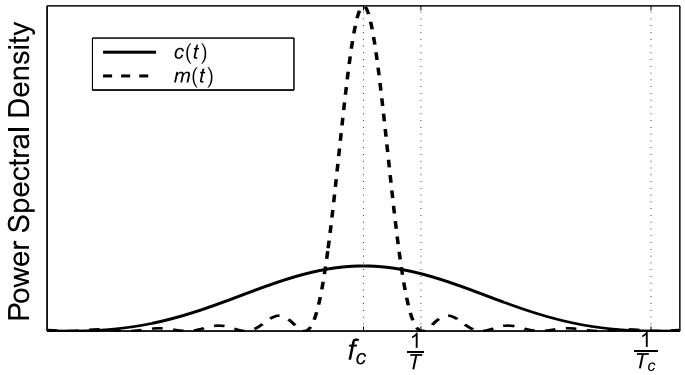

Figure 1.5: PSD of message $m(t)$ and spreading code $c(t)$.

The spreading code is also referred to as PRS, or pseudo-random noise (PRN), as it is intentionally chosen to be highly correlated with itself when it is synchronized (not shifted), thus its auto-correlation peaks when the same PRS is temporally aligned with itself, and lowly correlated with other sequences, thus its cross-correlation is small. An example of this are Gold codes [45] used in second generation global positioning system (GPS) and Weil sequences [46] used in third generation GPS.

1.1.4 Time-delay estimation via correlator bank

A method for time-delay estimation uses a correlator bank to calculate an autocorrelation function [47] which is then interpolated to estimate the time-delay.

Given a PRS vector $\mathbf{c}[\tau] \in \mathbb{R}^N$ of known sequence and delay τ, we can make a correlator bank of matched signal correlators with Q taps by shifting the copies of the known PRS. For

a signal sampled with N samples, the correlator bank is

$$\mathbf{Q} = \begin{bmatrix} \mathbf{c}[\tau_1] & \cdots & \mathbf{c}[\tau_Q] \end{bmatrix} \in \mathbb{R}^{N \times Q}. \tag{1.1}$$

For a PRS with chip time T_c, a typical value for τ_1 and τ_Q is $\mp T_c$ as that contains the range in which the initial signal acquisition time-delay estimate varies [9].

The auto-correlation is calculated by left-multiplying the transpose of the correlator bank $\mathbf{Q}$ with the code $\mathbf{c}[\tau]$,

$$\mathbf{q} = \left| \mathbf{Q}^{\mathrm{T}} \cdot \mathbf{c}[\tau] \right| \in \mathbb{R}^{Q}. \tag{1.2}$$

A cubic spline interpolation [48] is then applied to eq. (1.2), generating a interpolated auto-correlation function $\mathbf{f}(\tau)$. The time-delay is then estimated as

$$\hat{\tau} = \arg\max_{\tau} \mathbf{f}(\tau). \tag{1.3}$$

In Figure 1.6, the auto-correlation $\mathbf{q}$, interpolated auto-correlation function $\mathbf{f}(\tau)$, and estimated time-delay $\hat{\tau}$ is shown for a correlator bank with $Q = 11$ taps and PRS $\mathbf{c}[\tau]$ with delay $\tau = -0.5299 \cdot T_c$ and estimated time-delay $\hat{\tau} = -0.5305 \cdot T_c$, resulting in a ranging error of 0.1903 m.

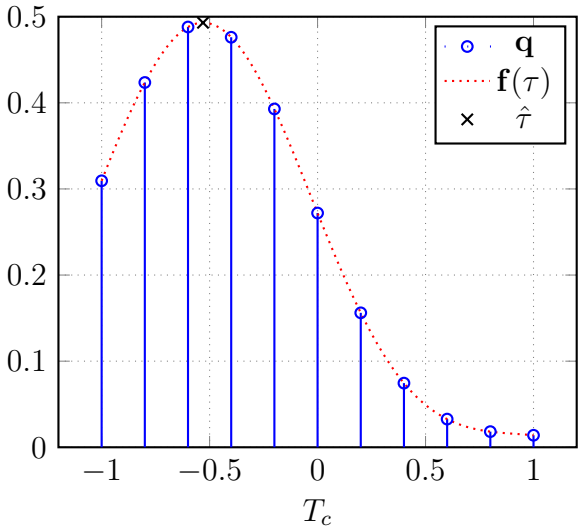

Figure 1.6: Autocorrelation $\mathbf{q}$ (normalized), interpolated autocorrelation function $\mathbf{f}(\tau)$, and estimated $\hat{\tau} = -0.5305 \cdot T_c$.

It is common for reflections and refractions to occurs in urban environments [49, 50], adding non-line-of-sight components to a received GNSS signal. This is shown in Figure 1.7 where a LOS signal with time-delay τ_{LOS} and a NLOS signal with time-delay τ_{NLOS}, resulting from reflection off a building, arrive at the same receiver.

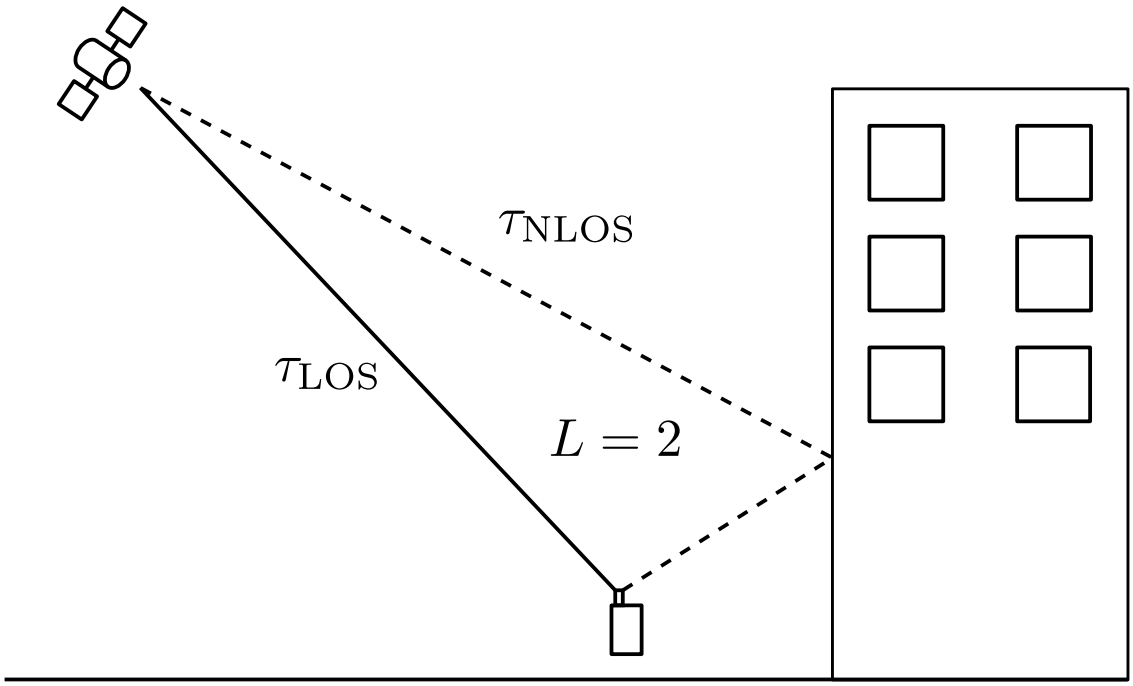

Figure 1.7: NLOS component resulting from reflection on a building.

In such cases, the NLOS signal is delayed and attenuated relative to the LOS signal. This can be seen in Figure 1.8, where the NLOS signal is delayed by $\tau_{\text{NLOS}} - \tau_{\text{LOS}} = 0.5 \cdot T_c$ and attenuated by 5 dB relative to the LOS signal. The correlation of both signals are shown separately.

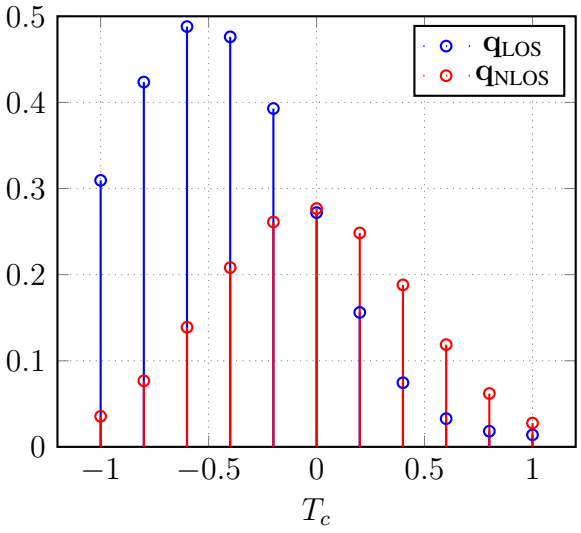

Figure 1.8: Autocorrelations $\mathbf{q}_{\text{LOS}}$ and $\mathbf{q}_{\text{NLOS}}$ (normalized).

Since a receiver will receive the sum of both signals, the result is the distorted autocorrelation function shown in Figure 1.9, with estimated time-delay $\hat{\tau} = -0.4318 \cdot T_c$, resulting in a ranging error of 33.3014 m.

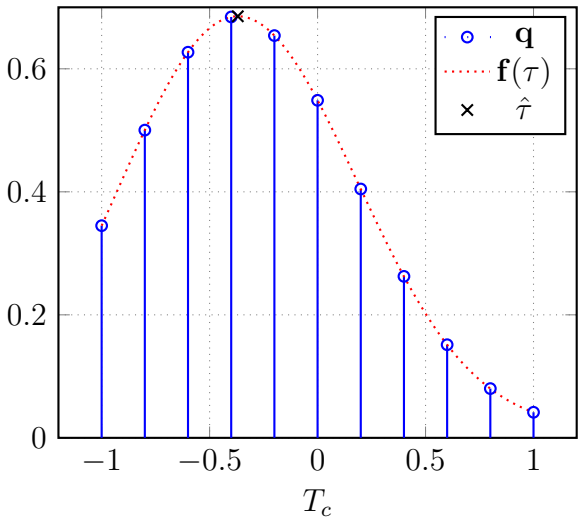

Figure 1.9: Distorted autocorrelation $\mathbf{q}$ (normalized), interpolated autocorrelation function $\mathbf{f}(\tau)$, and estimated $\hat{\tau} = -0.4318 \cdot T_c$.

Applying the correlator bank directly will also correlate the noise, coloring it [15,24,47]. This can be resolved by compressing the correlator bank. Compression is performed by applying a "thin" singular value decomposition (SVD) to the correlator bank, which preserves its information while making it unitary,

$$\mathbf{Q} = \tilde{\mathbf{Q}} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^{\mathrm{T}}, \tag{1.4}$$

and the correlator bank $\mathbf{Q} \in \mathbb{R}^{N \times Q}$ is decomposed into a "compressed" correlator bank (left singular values matrix) $\tilde{\mathbf{Q}} \in \mathbb{R}^{N \times Q}$ (using the full SVD would create an $N \times N$ matrix), singular values matrix $\mathbf{\Sigma} \in \mathbb{R}^{Q \times Q}$, and right singular vectors matrix $\mathbf{V} \in \mathbb{R}^{Q \times Q}$.

$\tilde{\mathbf{Q}}$ is defined as the compressed correlator bank and $\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ is defined as the remainder of the correlator bank. Because $\tilde{\mathbf{Q}}$ is unitary, performing correlation with it will not color the noise.

Observed data can then be correlated using $\tilde{\mathbf{Q}}$, followed by signal processing. After signal processing has been applied the correlation is completed by multiplying the results with the remainder of the correlator bank, $\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$.

1.2 Notation and Tensor Algebra

Scalars are indicated by italics, with uppercase denoting constants, e.g. L, while variables (and iterators) are denoted by lowercase, e.g. ℓ. Vectors are denoted by lowercase bold letters, e.g. $\mathbf{a}$, while matrices are denoted by uppercase bold letters, e.g. $\mathbf{A}$.

The domain and size (of each mode) of a given vector or matrix is denoted by $\mathbf{a} \in \mathbb{C}^M$ and $\mathbf{A} \in \mathbb{R}^{M \times N}$, respectively. The m, n-th element of $\mathbf{A}$ is denoted by $\mathbf{A}(m, n)$, while individual elements are indicated lowercase italics and subscript indexes such as $a_{m,n}$. The

m-th row and n-th column are denoted by $\mathbf{A}(m,\cdot)$ and $\mathbf{A}(\cdot,n)$, respectively. The composition of a matrix in in terms of its vectors is denoted by column concatenation like such $\mathbf{A} = \left[\mathbf{a}_1,\ldots,\mathbf{a}_N\right] \in \mathbb{C}^{M\times N}$, with each vector $\mathbf{a}_n \in \mathbb{C}^M, n = 1,\ldots,N$.

The $M \times M$ identity matrix is denoted by $\mathbf{I}_M$. The $M \times N$ rectangular matrix with ones in its diagonal is denoted by $\mathbf{I}_{M\times N}$. The all-zeros and all-ones vectors of size N are denoted by $\mathbf{0}_N$ and $\mathbf{1}_N$, respectively. The canonical basis vector of size P with 1 in its first position is indicated by $\mathbf{e}_P$ while the canonical basis vector of size P with 1 in its p-th position is indicated by $\mathbf{e}_p$.

The inverse, pseudoinverse, transpose, inverse transpose, conjugation, and conjugate transpose (Hermitian) are denoted by $^{-1}$, †, $^{\mathrm{T}}$, $^{-\mathrm{T}}$, *, and $^{\mathrm{H}}$, respectively. The operator $\mathrm{dediag}\left\{\mathbf{v}\right\}$ creates a (square) diagonal matrix from the vector $\mathbf{v}$ and the operator $\mathrm{diag}\left\{\mathbf{V}\right\}$ extracts the diagonal elements from the matrix $\mathbf{V}$, forming a vector. The operator $\mathcal{D}_m\left\{\mathbf{\Gamma}\right\}$ extracts the k-th row of $\mathbf{\Gamma}$ and creates a diagonal matrix with the extracted row such that

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_{1,1} & \cdots & \gamma_{1,L} \\ \vdots & \vdots & \vdots \\ \gamma_{K,1} & \cdots & \gamma_{K,L} \end{bmatrix} \in \mathbb{C}^{K\times L}, \tag{1.5}$$

$$\mathcal{D}_k\left\{\mathbf{\Gamma}\right\} = \begin{bmatrix} \gamma_{k,1} & 0 & \cdots & 0 \\ 0 & \gamma_{k,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{k,L} \end{bmatrix} \in \mathbb{C}^{L\times L}. \tag{1.6}$$

The outer product, Hadamard product, Kronecker product, and Khatri-Rao (or column-wise Kronecker) product are indicated by $\circ$, $\odot$, $\otimes$, and $\diamond$, respectively.

Tensors are represented by bold calligraphic letters, e.g. $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1\times M_2\times\ldots\times M_N}$ is the N-th order tensor of size M_n in each n-th mode, $n = 1,\ldots,N$. Tensor concatenation along the n-th mode is denoted by $\sqcup_n$.

The third-order tensor $\boldsymbol{\mathcal{T}}$ composed by factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of sizes $I\times R$, $J\times R$, and $K\times R$, respectively is denoted by [51] $\boldsymbol{\mathcal{T}} = \left[\!\left[\mathbf{A},\mathbf{B},\mathbf{C}\right]\!\right]_R \in \mathbb{C}^{I\times J\times K}$. If $R \leq I, J, K$, the tensor $\boldsymbol{\mathcal{T}}$ can be decomposed into a sum of R rank-1 outer products of the columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$,

$$\boldsymbol{\mathcal{T}} = \left[\!\left[\mathbf{A},\mathbf{B},\mathbf{C}\right]\!\right]_R = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{1.7}$$

when R is the minimum amount of rank-1 outer products whose summation compose $\boldsymbol{\mathcal{T}}$. This is known as the canonical polyadic decomposition.

Tensor unfolding, or matricization, is denoted by $\left[\cdot\right]_{(n)}$, with n being the unfolding mode. Unfolding an N-th order tensor is performed by extracting the tensor fiber resultant from

fixing the n-th index while iterating through the indexes of the the remaining $N-1$ modes in reverse order, such that

$$
\begin{aligned}
&\boldsymbol{\mathcal{T}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]_R \in \mathbb{C}^{I \times J \times K}, \\
&[\boldsymbol{\mathcal{T}}]_{(1)} = \begin{bmatrix} \boldsymbol{\mathcal{T}}(\cdot,1,1) & \boldsymbol{\mathcal{T}}(\cdot,1,2) & \cdots & \boldsymbol{\mathcal{T}}(\cdot,1,K) & \boldsymbol{\mathcal{T}}(\cdot,2,1) & \boldsymbol{\mathcal{T}}(\cdot,J,K) \end{bmatrix} \in \mathbb{C}^{I \times JK}, \\
&[\boldsymbol{\mathcal{T}}]_{(2)} = \begin{bmatrix} \boldsymbol{\mathcal{T}}(1,\cdot,1) & \boldsymbol{\mathcal{T}}(1,\cdot,2) & \cdots & \boldsymbol{\mathcal{T}}(1,\cdot,K) & \boldsymbol{\mathcal{T}}(2,\cdot,1) & \boldsymbol{\mathcal{T}}(I,\cdot,K) \end{bmatrix} \in \mathbb{C}^{J \times KI}, \\
&[\boldsymbol{\mathcal{T}}]_{(3)} = \begin{bmatrix} \boldsymbol{\mathcal{T}}(1,1,\cdot) & \boldsymbol{\mathcal{T}}(1,2,\cdot) & \cdots & \boldsymbol{\mathcal{T}}(1,J,\cdot) & \boldsymbol{\mathcal{T}}(2,1,\cdot) & \boldsymbol{\mathcal{T}}(I,J,\cdot) \end{bmatrix} \in \mathbb{C}^{K \times IJ}.
\end{aligned}
$$

The n-mode product between a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_N}$ and a matrix $\mathbf{M} \in \mathbb{C}^{N \times M_n}$ is equivalent to unfolding $\boldsymbol{\mathcal{T}}$ in its n-th mode and left-multiplying by $\mathbf{M}$, such that $[\boldsymbol{\mathcal{T}} \times_n \mathbf{M}]_{(n)} = \mathbf{M} \cdot [\boldsymbol{\mathcal{T}}]_{(n)}$, then refolding back into a tensor of size $M_1 \times M_2 \times \cdots \times M_{n-1} \times N \times M_{n+1} \times \cdots \times M_N$.

The composition of an N-order tensor $\boldsymbol{\mathcal{T}}$ can be represented in terms of n-mode products of an N-th order identity tensor $\boldsymbol{\mathcal{I}}_{N \times R} \in \mathbb{R}^{R \times \cdots \times R}$ core and factor matrices $\mathbf{A}_n$ such that

$$
\begin{aligned}
\boldsymbol{\mathcal{T}} &= [\![\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{N-1}, \mathbf{A}_N]\!]_R \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_{N-1} \times M_N}, \\
&= \boldsymbol{\mathcal{I}}_{N,R} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_{N-1} \mathbf{A}_{N-1} \times_N \mathbf{A}_N.
\end{aligned}
$$

Unfoldings can be also represented in terms of Kronecker products of the factor matrices of a tensor, continuing from above,

$$
\begin{aligned}
[\boldsymbol{\mathcal{T}}]_{(n)} &= \mathbf{A}_n \, [\boldsymbol{\mathcal{I}}_{N,R}]_{(n)} \, (\mathbf{A}_1 \otimes \cdots \otimes \mathbf{A}_{n-1} \otimes \mathbf{A}_{n+1} \otimes \cdots \otimes \mathbf{A}_N)^{\mathrm{T}}, \\
&= \mathbf{A}_n \, [\boldsymbol{\mathcal{I}}_{N,R}]_{(n)} \left(\bigotimes_{\substack{m=1 \\ m \neq n}}^{N} \mathbf{A}_m \right)^{\mathrm{T}} \in \mathbb{C}^{M_n \times \prod_{\substack{m=1 \\ m \neq n}}^{N} M_m}.
\end{aligned}
$$

If the tensor core is an identity tensor, the unfolding can be represented in terms of Khatri-Rao products of the factor matrices,

$$
\begin{aligned}
[\boldsymbol{\mathcal{T}}]_{(n)} &= \mathbf{A}_n \, (\mathbf{A}_1 \diamond \cdots \diamond \mathbf{A}_{n-1} \diamond \mathbf{A}_{n+1} \diamond \cdots \diamond \mathbf{A}_N)^{\mathrm{T}}, \\
&= \mathbf{A}_n \left(\bigdiamond_{\substack{m=1 \\ m \neq n}}^{N} \mathbf{A}_m \right)^{\mathrm{T}}.
\end{aligned}
$$

The n-mode unfolding is a particular case of the generalized unfolding of a tensor in which indexes $\{1, \ldots, N\}$ are partitioned into two ordered subsets of indexes [52]. The unfolding $[\boldsymbol{\mathcal{X}}]_{(\mathbb{S}_1;\mathbb{S}_2)}$ has the general formula

$$
[\boldsymbol{\mathcal{X}}]_{(\mathbb{S}_1;\mathbb{S}_2)} = \left(\bigdiamond_{n \in \mathbb{S}_1} \mathbf{A}_n \right) \left(\bigdiamond_{m \in \mathbb{S}_2} \mathbf{A}_m \right)^{\mathrm{T}} \in \mathbb{C}^{\prod_{n \in \mathbb{S}_1} M_n \times \prod_{m \in \mathbb{S}_2} M_m}. \tag{1.8}
$$

1.3 Work Division

This work is divided into 5 chapters, including this introduction which presents GNSS and a brief theoretical presentation of related concepts, the notation used throughout, and tensor algebra. Of import is time-delay estimation performed using a compressed correlator bank as this is used frequently in this work.

Chapter 2 presents batch tensor-based time-delay estimation in an environment which signals are well separated (non-clustered). Data from several observation periods are collected and processed using parametric and algebraic tensor processing techniques with a comparison of each technique's advantages over the state-of-the-art and each other.

Chapter 3 presents batch tensor-based time-delay estimation in an environment in which signals are clustered. This is due to multipath signal components which are appear in clusters [12, 40] and have similar characteristics, resulting in an observation tensor that is rank-deficient in one of its modes. Data from several observation periods are collected processed using parametric and algebraic tensor processing techniques with a comparison of each technique's advantages over the state-of-the-art and each other.

Chapter 4 presents adaptive tensor-based time-delay estimation in an environment which signals are well separated. Processing is performed over each observation period to iteratively improve subspace estimates used in tensor processing techniques.

Chapter 5 extends adaptive processing to a higher-order model and employs sequential truncation strategies to minimize computation cost when estimating the subspaces from each dimension.

Chapter 2

Tensor-Based Time-Delay Estimation for Batch Processing in Lowly Correlated Scenarios

In this chapter, we introduce the tensor data model for sampled GNSS signals introduced in [15] with some modification, review the state-of-the-art tensor-based schemes for batch time-delay estimation, and present the proposed CPD-GEVD tensor-based schemes for time-delay estimation. Finally, numerical simulations are performed comparing the state-of-the-art performance against the proposed schemes.

2.1 Data Model

The data model is similar to the one found in [16] with the difference that the modes are ordered in such a way as to correspond to the "classical" model [53]

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{\Gamma} \cdot \mathbf{S}^{\mathrm{T}} + \mathbf{N} \in \mathbb{C}^{M \times N}, \tag{2.1}$$

where $\mathbf{A} \in \mathbb{C}^{M \times d}$ is the array response matrix and $\mathbf{S} \in \mathbb{C}^{N \times d}$ is the signal matrix for d signals (model order [54, 55]). Furthermor, a diagonal matrix $\mathbf{\Gamma} = \operatorname{dediag}\{\boldsymbol{\gamma}\} \in \mathbb{C}^{d \times d}$ is the (diagonal) complex amplitude (power) matrix, which scales and adds a complex phase to each column of $\mathbf{A}$ and $\mathbf{S}$, while $\mathbf{N} \in \mathbb{C}^{M \times N}$ is additive noise.

2.1.1 Pre-Correlation Data Model

At each k-th period of the PRS, N samples of GNSS data from L signal components (1 LOS component and $L-1$ NLOS components) are collected in

$$\mathbf{X}[k] = \mathbf{A}\,\mathrm{diag}\left\{\boldsymbol{\gamma}[k]\right\}\mathbf{C}^{\mathrm{T}} + \mathbf{N}[k] \in \mathbb{C}^{M\times N}, \tag{2.2}$$

in which the matrix $\mathbf{A} \in \mathbb{C}^{M\times L}$ collects the array response from the L signals impinging on M sensors in an array, such that

$$\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_L\end{bmatrix} = \begin{bmatrix}\mathbf{a}(\phi_1) & \cdots & \mathbf{a}(\phi_L)\end{bmatrix}, \tag{2.3}$$

and $\mathbf{a}_\ell = \mathbf{a}(\phi_\ell) \in \mathbb{C}^M$ is the array response vector for a given direction of arrival (DoA) $\phi_\ell, \ell = 1,\ldots,L$. The vector

$$\boldsymbol{\gamma}[k] = \begin{bmatrix}\gamma_1[k]\\ \vdots \\ \gamma_L[k]\end{bmatrix} \in \mathbb{C}^L \tag{2.4}$$

collects the complex amplitudes, and

$$\mathbf{C} = \begin{bmatrix}\mathbf{c}_1 & \cdots & \mathbf{c}_L\end{bmatrix} = \begin{bmatrix}\mathbf{c}[\tau_1] & \cdots & \mathbf{c}[\tau_L]\end{bmatrix} \in \mathbb{R}^{N\times L} \tag{2.5}$$

collects all L sampled PRS $\mathbf{c}[\tau_\ell] \in \mathbb{R}^N$, such as GPS coarse acquisition (C/A) code, each with delay $\tau_\ell, \ell = \{1,\ldots,L\}$. $\mathbf{N}[k]$ is white Gaussian noise.

Collecting the complex amplitudes from K periods in a matrix

$$\boldsymbol{\Gamma} = \begin{bmatrix}\boldsymbol{\gamma}[1] & \cdots & \boldsymbol{\gamma}[K]\end{bmatrix}^{\mathrm{T}} = \begin{bmatrix}\boldsymbol{\gamma}_1 & \cdots & \boldsymbol{\gamma}_L\end{bmatrix} \in \mathbb{C}^{K\times L} \tag{2.6}$$

with

$$\boldsymbol{\gamma}_\ell = \begin{bmatrix}\gamma_\ell[1]\\ \vdots \\ \gamma_\ell[K]\end{bmatrix} \in \mathbb{C}^K, \ell = 1,\ldots,L. \tag{2.7}$$

We can rewrite eq. (2.2) as

$$\mathbf{X}[k] = \mathbf{A}\cdot\mathcal{D}_k\left\{\boldsymbol{\Gamma}\right\}\cdot\mathbf{C}^{\mathrm{T}} + \mathbf{N}[k]. \tag{2.8}$$

Collecting all K observations into a matrix $\mathbf{X}$, by vertically stacking these, we have

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}[1] \\ \vdots \\ \mathbf{X}[K] \end{bmatrix} = \begin{bmatrix} \mathbf{A} \cdot \mathcal{D}_1 \{\mathbf{\Gamma}\} \cdot \mathbf{C}^{\mathrm{T}} \\ \vdots \\ \mathbf{A} \cdot \mathcal{D}_K \{\mathbf{\Gamma}\} \cdot \mathbf{C}^{\mathrm{T}} \end{bmatrix} + \begin{bmatrix} \mathbf{N}[1] \\ \vdots \\ \mathbf{N}[K] \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathcal{D}_1 \{\mathbf{\Gamma}\} \\ \vdots \\ \mathbf{A}\mathcal{D}_K \{\mathbf{\Gamma}\} \end{bmatrix} \mathbf{C}^{\mathrm{T}} + \mathbf{N}. \tag{2.9}$$

Since each k-th matrix product $\mathbf{A}\mathcal{D}_k \{\mathbf{\Gamma}\} \in \mathbb{C}^{M \times L}$ scales each column of $\mathbf{A}$, by vertically stacking all K products, we have the Khatri-Rao product $\mathbf{\Gamma} \diamond \mathbf{A} \in \mathbb{C}^{KM \times L}$ and eq. (2.9) can be more compactly written as

$$\mathbf{X} = (\mathbf{\Gamma} \diamond \mathbf{A})\, \mathbf{C}^{\mathrm{T}} + \mathbf{N} \in \mathbb{C}^{MK \times N}, \tag{2.10}$$

which can be interpreted as the $(2, 1; 3)$-mode unfolding of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M \times K \times N}$,

$$[\boldsymbol{\mathcal{X}}]_{(2,1;3)} = (\mathbf{\Gamma} \diamond \mathbf{A})\, \mathbf{C}^{\mathrm{T}} + \mathbf{N} = [\boldsymbol{\mathcal{X}}_0]_{(2,1;3)} + [\boldsymbol{\mathcal{N}}]_{(2,1;3)} \tag{2.11}$$

The noiseless tensor $\boldsymbol{\mathcal{X}}_0$ has the following canonical polyadic decomposition (CPD) [56]:

$$\boldsymbol{\mathcal{X}}_0 = \sum_{\ell=1}^{L} \mathbf{A}(\cdot, \ell) \circ \mathbf{\Gamma}(\cdot, \ell) \circ \mathbf{C}(\cdot, \ell), = \sum_{\ell=1}^{L} \mathbf{a}_\ell \circ \boldsymbol{\gamma}_\ell \circ \mathbf{c}_\ell. \tag{2.12}$$

Equation (2.12) can be rewritten in terms of n-mode products in the following fashion [57–59]:

$$\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \mathbf{\Gamma} \times_3 \mathbf{C} \in \mathbb{C}^{M \times K \times N}. \tag{2.13}$$

2.1.2 Post-Correlation Data Model

The correlator bank $\mathbf{Q} = \big[\mathbf{c}[\tau_1], \ldots, \mathbf{c}[\tau_Q]\big] \in \mathbb{R}^{N \times Q}$ with Q taps can be applied directly to eq. (2.2), resulting in the correlating the sampled PRS with the known PRS taps in the correlator bank

$$\mathbf{Y}[k] = \mathbf{X}[k] \cdot \mathbf{Q} = \mathbf{A} \cdot \mathcal{D}_k \{\Gamma\} \cdot \mathbf{C}^{\mathrm{T}} \cdot \mathbf{Q} + \mathbf{N}[k] \cdot \mathbf{Q} \in \mathbf{C}^{M \times Q}, \tag{2.14}$$

but this will color the noise [32]. To overcome this problem, the correlator bank can be compressed, preserving its information while becoming unitary, using the "thin" SVD [47]

$$\mathbf{Q} = \tilde{\mathbf{Q}} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{N \times Q}, \tag{2.15}$$

where $\tilde{\mathbf{Q}} \in \mathbb{R}^{N \times Q}$ contains the left singular vectors, $\mathbf{\Sigma} \in \mathbb{R}^{Q \times Q}$ is a diagonal matrix containing the singular values, and $\mathbf{V}^{\mathrm{T}} \in \mathbb{R}$ contains the right singular vectors of $\mathbf{Q}$, respectively. $\tilde{\mathbf{Q}}$ is defined as the compressed correlator bank and $\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{Q \times Q}$ is defined as the remainder

of the correlator bank.

To apply the compressed correlator bank to eq. (2.13), we apply its transpose to $\boldsymbol{\mathcal{X}}$ using the n-mode product in the third mode,

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_3 \tilde{\mathbf{Q}}^{\mathrm{T}} = (\boldsymbol{\mathcal{X}}_0 + \boldsymbol{\mathcal{N}}) \times_3 \tilde{\mathbf{Q}}^{\mathrm{T}}, \tag{2.16}$$

$$= (\boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \mathbf{C} + \boldsymbol{\mathcal{N}}) \times_3 \tilde{\mathbf{Q}}^{\mathrm{T}}, \tag{2.17}$$

$$= \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{Q}}^{\mathrm{T}}\mathbf{C} + \tilde{\boldsymbol{\mathcal{N}}}, \tag{2.18}$$

$$= \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}} + \tilde{\boldsymbol{\mathcal{N}}} \in \mathbb{C}^{M \times K \times Q}, \tag{2.19}$$

and the noiseless correlated signal tensor has canonical polyadic decomposition

$$\boldsymbol{\mathcal{Y}}_0 = \sum_{\ell=1}^{L} \mathbf{A}(:,\ell) \circ \boldsymbol{\Gamma}(:,\ell) \circ \tilde{\mathbf{C}}(:,\ell), \tag{2.20}$$

$$= \sum_{\ell=1}^{L} \mathbf{a}_\ell \circ \boldsymbol{\gamma}_\ell \circ \tilde{\mathbf{c}}_\ell. \tag{2.21}$$

An example of a post-correlation data tensor of size $M = 8$, $K = 30$, and $Q = 11$ is visualized as voxels [60] in Figure 2.1. Color intensity is based on absolute (normalized) magnitude of each element of the data.

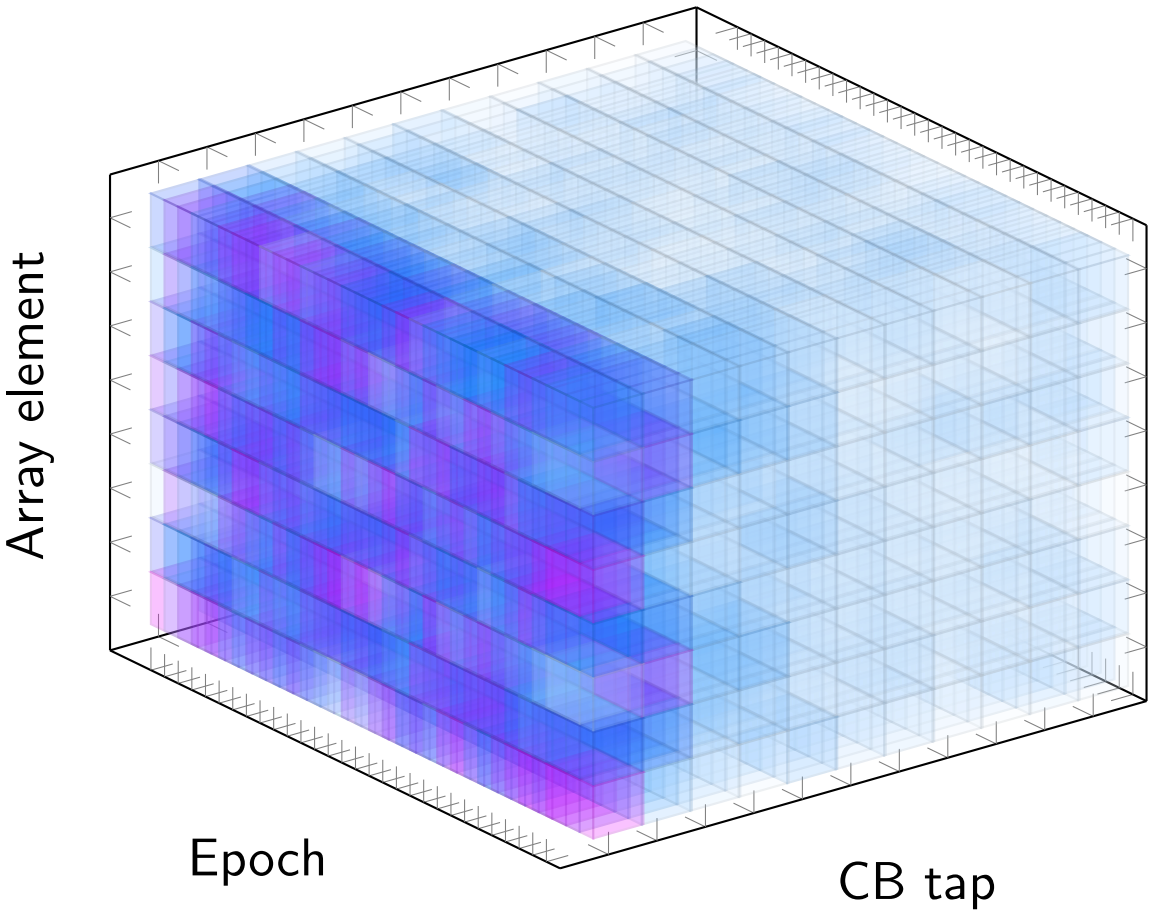

Figure 2.1: Voxel visualization of 3rd-order tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{8\times 30\times 11}$

2.2 State-of-the-Art Tensor-Based Time-Delay Estimation

This sections reviews the state-of-the-art tensor-based schemes for time-delay estimation, namely the HOE [15] with FBA [19] and ESPS [20,61], DoA/KRF [24], ProKRaft [32], and SECSI [35].

2.2.1 Higher-order eigenfilter (HOE)

The higher-order eigenfilter utilizes the rank-1 MLSVD [17] to calculate the primary singular vectors of $\boldsymbol{\mathcal{Y}}$

$$\boldsymbol{\mathcal{Y}} \approx s \cdot \mathbf{u}_1 \circ \mathbf{u}_2 \circ \mathbf{u}_3 \in \mathbb{C}^{M\times K\times Q}, \tag{2.22}$$

$$\tag{2.23}$$

where $\mathbf{u}_1 \in \mathbb{C}^M$, $\mathbf{u}_2 \in \mathbb{C}^K$, and $\mathbf{u}_3 \in \mathbb{C}^Q$ are the rank-1 (left) singular vectors of the SVD in each mode, and $s \in \mathbb{C}$ is a scalar.

The conjugate transpose first- and second-mode singular vectors are used to filter the

LOS signal from the data

$$\hat{\tilde{\mathbf{c}}}_{\text{LOS}} = \left[\boldsymbol{\mathcal{Y}} \times_1 \mathbf{u}_1^{\text{H}} \times_2 \mathbf{u}_2^{\text{H}}\right]_{(3)} \in \mathbb{C}^Q. \tag{2.24}$$

Since $\tilde{\mathbf{c}}_{\text{LOS}} = \tilde{\mathbf{Q}}^{\text{T}}\mathbf{c}_{\text{LOS}}$, the correlation is completed by left-multiplying the remainder of the correlator bank from eq. (2.15) with the estimated LOS vector and taking the absolute value, we have the auto-correlation vector

$$\hat{\mathbf{q}} = \left|\left(\boldsymbol{\Sigma}\mathbf{V}^{\text{T}}\right)^{\text{T}} \cdot \hat{\tilde{\mathbf{c}}}_{\text{LOS}}\right| = \left|\mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{c}}}_{\text{LOS}}\right| \in \mathbb{R}^Q. \tag{2.25}$$

A cubic spline interpolation (CSI) is applied to $\hat{\mathbf{q}}$ to estimate the time delay [16].

The HOE can also be combined with FBA [19] and ESPS [20] preprocessing to increase accuracy. Given the first-mode unfolding of $\boldsymbol{\mathcal{Y}}$

$$\mathbf{Y}_1 = [\boldsymbol{\mathcal{Y}}]_{(1)} = \mathbf{A}\left(\boldsymbol{\Gamma} \diamond \tilde{\mathbf{C}}\right)^{\text{T}} + \left[\tilde{\boldsymbol{\mathcal{N}}}\right]_{(1)} \in \mathbb{C}^{M\times KQ}, \tag{2.26}$$

which is left-multiplied by the array response matrix, and exchange matrices $\boldsymbol{\Pi}_M \in \mathbb{R}^{M\times M}$ and $\boldsymbol{\Pi}_{KQ} \in \mathbb{R}^{KQ\times KQ}$,

$$\boldsymbol{\Pi}_M = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{M\times M}, \tag{2.27}$$

$$\boldsymbol{\Pi}_{KQ} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{KQ\times KQ}, \tag{2.28}$$

with $\boldsymbol{\Pi}_M \cdot \boldsymbol{\Pi}_M = \mathbf{I}_M$ and $\boldsymbol{\Pi}_{KQ} \cdot \boldsymbol{\Pi}_{KQ} = \mathbf{I}_{KQ}$, the forward-backward averaged first-mode unfolding of $\boldsymbol{\mathcal{Y}}$ is

$$\mathbf{Y}_{1,\text{FBA}} = \begin{bmatrix} \mathbf{Y}_1 & \boldsymbol{\Pi}_M \cdot \mathbf{Y}_1^* \cdot \boldsymbol{\Pi}_{KQ} \end{bmatrix} \in \mathbb{C}^{M\times 2KQ}, \tag{2.29}$$

the concatenation of $\mathbf{Y}_1$ and $\boldsymbol{\Pi}_M \cdot \mathbf{Y}_1^* \cdot \boldsymbol{\Pi}_{KQ}$, which is $\mathbf{Y}_1$ conjugated and flipped along is horizontal and vertical axes.

If only FBA were applied, eq. (2.29) would be folded back into a tensor $\boldsymbol{\mathcal{Y}}_{\text{FBA}} \in \mathbb{C}^{M\times 2K\times Q}$. To apply ESPS using L_s subarrays and $M_s = M - L_s + 1$ elements in each

subarray, we utilize selection matrices $\mathbf{J}_{\ell_s}, \ell_s = 1, \ldots, L_s$,

$$\mathbf{J}_{\ell_s} = \begin{bmatrix} \mathbf{0}_{M_s,\ell_s-1} & \mathbf{I}_{M_s} & \mathbf{0}_{M_s,L_s-\ell_s} \end{bmatrix} \in \mathbb{R}^{M_s \times M}, \tag{2.30}$$

to apply spatial smoothing (SPS) to eq. (2.29),

$$\mathbf{Y}_{1,\text{FBA+SPS}} = \begin{bmatrix} \mathbf{J}_1 \cdot \mathbf{Y}_{1,\text{FBA}} & \mathbf{J}_2 \cdot \mathbf{Y}_{1,\text{FBA}} & \cdots & \mathbf{J}_{L_s} \cdot \mathbf{Y}_{1,\text{FBA}} \end{bmatrix} \in \mathbb{C}^{M_s \times 2L_sKQ}. \tag{2.31}$$

If SPS were applied, eq. (2.31) would be folded back into a tensor $\boldsymbol{\mathcal{Y}}_{\text{FBA+SPS}} \in \mathbb{C}^{M_s \times 2L_sK \times Q}$. ESPS takes advantage of the increased information by folding eq. (2.31) into a fourth-order tensor $\boldsymbol{\mathcal{Y}}_{\text{FBA+ESPS}} \in \mathbb{C}^{M_s \times L_s \times 2K \times Q}$. The rank-1 MLSVD is

$$\boldsymbol{\mathcal{Y}}_{\text{FBA+ESPS}} = s\mathbf{u}_1 \circ \mathbf{u}_2 \circ \mathbf{u}_3 \circ \mathbf{u}_4 \in \mathbb{C}^{M_s \times L_s \times 2K \times Q} \tag{2.32}$$

and the HOE with FBA and ESPS preprocessing is

$$\hat{\tilde{\mathbf{c}}} = \left[\boldsymbol{\mathcal{Y}}_{\text{FBA+ESPS}} \times_1 \mathbf{u}_1^{\text{H}} \times_2 \mathbf{u}_2^{\text{H}} \times_3 \mathbf{u}_3^{\text{H}}\right]_{(4)} \in \mathbb{C}^{Q}, \tag{2.33}$$

and we proceed by estimating the cross correlation vector described in eq. (2.25) followed by CSI to estimate the time delay.

The HOE+FBA+ESPS approach is illustrated as a block diagram in Figure 2.2.

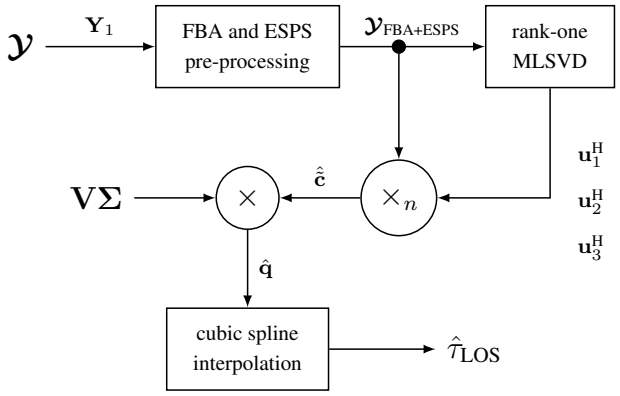

Figure 2.2: Block diagram of the state-of-the-art higher-order eigenfilter with FBA and ESPS preprocessing TDE approach

2.2.2 Direction of Arrival and Khatri-Rao factorization (DoA/KRF)

The DoA/KRF approach also utilizes the first-mode unfolding to expose the array response matrix. A low-rank SVD of rank L is then used to estimate the signal subspace

$$\mathbf{Y}_1 = \mathbf{U}_s \cdot \boldsymbol{\Sigma}_s \cdot \mathbf{V}_s^{\text{H}}, \tag{2.34}$$

with $\mathbf{U}_s \in \mathbb{C}^{M \times L}$, $\boldsymbol{\Sigma}_s \in \mathbb{C}^{L \times L}$, and $\mathbf{V}_s \in \mathbb{C}^{KQ \times L}$.

Since $\mathbf{U}_s$ spans the same column space as $\mathbf{A}$ there exists a nonsingular transform matrix $\mathbf{T} \in \mathbb{C}^{L\times L}$ such that $\mathbf{A} = \mathbf{U}_s \cdot \mathbf{T}$. This can be used to exploit the shift invariance property [22] to estimate the spatial frequencies then the directions of arrival. Considering the relation with the rows of $\mathbf{A}$ with the greatest overlap [62]

$$\mathbf{A}_{\text{first}} = \mathbf{A}(1:(M-1),\cdot) \in \mathbb{C}^{M-1,L} \tag{2.35}$$

$$\mathbf{A}_{\text{last}} = \mathbf{A}(2:M,\cdot) \in \mathbb{C}^{M-1,L} \tag{2.36}$$

the following relation holds

$$\mathbf{A}_{\text{first}} \operatorname{diag}\left\{\begin{bmatrix} e^{j\mu_1} & \cdots & e^{j\mu_L}\end{bmatrix}^{\mathrm{T}}\right\} = \mathbf{A}_{\text{first}}\boldsymbol{\Phi} = \mathbf{A}_{\text{last}}. \tag{2.37}$$

Replacing $\mathbf{A}$ with $\mathbf{U}_s \cdot \mathbf{T}$

$$\mathbf{U}_{s,\text{first}} \cdot \mathbf{T} \cdot \boldsymbol{\Phi} = \mathbf{U}_{s,\text{last}} \cdot \mathbf{T} \tag{2.38}$$

and changing the order of the products

$$\mathbf{T} \cdot \boldsymbol{\Phi} \cdot \mathbf{T}^{-1} = \mathbf{U}^{\dagger}_{s,\text{first}} \cdot \mathbf{U}_{s,\text{last}} = \boldsymbol{\Psi} \in \mathbb{C}^{L\times L} \tag{2.39}$$

which is an eigenvalue problem in which the estimated complex exponential spatial frequencies $e^{j\boldsymbol{\mu}} \in \mathbb{C}^L$ are the eigenvalues of $\boldsymbol{\Psi}$

$$\boldsymbol{\mu} = \arg\left(\operatorname{diag}\left\{\boldsymbol{\Phi}\right\}\right) \in \mathbb{R}^L. \tag{2.40}$$

With the estimated spatial frequencies, assuming a perfectly calibrated uniform linear array (ULA), the DoAs can be calculated

$$\hat{\boldsymbol{\phi}} = \arcsin\left(\frac{\boldsymbol{\mu}}{\pi}\right) = \begin{bmatrix}\hat{\phi}_1 & \cdots & \hat{\phi}_L\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^L. \tag{2.41}$$

With the estimated DoAs, the array response matrix is rebuilt

$$\hat{\mathbf{A}} = \begin{bmatrix}\mathbf{a}(\hat{\phi}_1) & \cdots & \mathbf{a}(\hat{\phi}_L)\end{bmatrix} \in \mathbb{C}^{M\times L} \tag{2.42}$$

and its pseudoinverse is used to estimate the Khatri-Rao product of $\boldsymbol{\Gamma}$ and $\tilde{\mathbf{C}}$

$$\left(\hat{\mathbf{A}}^{\dagger} \cdot \mathbf{Y}_1\right)^{\mathrm{T}} = \hat{\boldsymbol{\Gamma}} \diamond \hat{\tilde{\mathbf{C}}}, \tag{2.43}$$

and apply LSKRF [23], shown in Algorithm 1, to estimate $\boldsymbol{\Gamma}$ and $\tilde{\mathbf{C}}$.

With the estimates of $\mathbf{A}$, $\boldsymbol{\Gamma}$, and $\tilde{\mathbf{C}}$, the LOS component can be selected by selecting the column of $\hat{\boldsymbol{\Gamma}}$ with greatest 2-norm [24] or estimating all delays calculating the auto-

Algorithm 1 Least-Squares Khatri-Rao factorization (LSKRF)

Input: $\mathbf{K} = \mathbf{\Gamma} \diamond \tilde{\mathbf{C}} = \left[\mathbf{k}_1, \ldots, \mathbf{k}_L\right] \in \mathbb{C}^{KQ \times L}$, sizes K and Q.
Output: estimated factor matrices $\mathbf{\Gamma} \in \mathbb{C}^{K \times L}$ and $\tilde{\mathbf{C}} \in \mathbb{C}^{Q \times L}$

1: **for** $\ell = 1$ to L **do**
2: Calculate the rank-1 SVD of

$$\underset{Q \times K}{\text{unvec}}\left\{\mathbf{k}_\ell\right\} = \mathbf{u}_\ell \cdot s \cdot \mathbf{v}_\ell^{\mathrm{H}} = \tilde{\mathbf{c}}_\ell \cdot \boldsymbol{\gamma}_\ell^{\mathrm{T}}$$

3: Estimate $\tilde{\mathbf{c}}_\ell$ and $\boldsymbol{\gamma}_\ell$

$$\hat{\tilde{\mathbf{c}}}_\ell = \sqrt{s} \cdot \mathbf{u}_\ell$$
$$\hat{\tilde{\boldsymbol{\gamma}}}_\ell = \sqrt{s} \cdot \mathbf{v}_\ell^*$$

4: **end for**
5: Rebuild $\hat{\mathbf{\Gamma}}$ and $\hat{\tilde{\mathbf{C}}}$

$$\hat{\mathbf{\Gamma}} = \begin{bmatrix}\hat{\boldsymbol{\gamma}}_1 & \cdots & \hat{\boldsymbol{\gamma}}_L\end{bmatrix} \in \mathbb{C}^{K \times L}$$
$$\hat{\tilde{\mathbf{C}}} = \begin{bmatrix}\hat{\tilde{\mathbf{c}}}_1 & \cdots & \hat{\tilde{\mathbf{c}}}_L\end{bmatrix} \in \mathbb{C}^{Q \times L}$$

correlation matrix

$$\hat{\mathbf{Q}} = \left|\mathbf{V\Sigma} \cdot \hat{\tilde{\mathbf{C}}}\right| \in \mathbb{C}^{Q \times L} \tag{2.44}$$

and applying CSI to all columns of $\hat{\tilde{\mathbf{Q}}}$ to estimate all time-delays and selecting the promptest [24], or using all estimated factor matrices to calculate the CPD component amplitudes [35, 36, 63] and selecting the component with greatest absolute amplitude.

The DoA/KRF approach to time-delay estimation is illustrated as a block diagram in Figure 2.3.

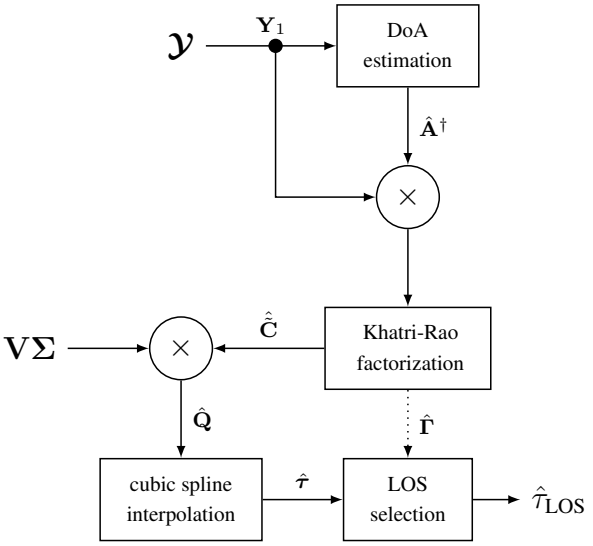

Figure 2.3: Block diagram of the state-of-the-art DoA/KRF TDE approach

2.2.3 Procrustes estimation and Khatri-Rao factorization (ProKRaft)

Using the third-mode unfolding of $\boldsymbol{\mathcal{Y}}$

$$\mathbf{Y}_3 = [\boldsymbol{\mathcal{Y}}]_{(3)} = \tilde{\mathbf{C}} \left(\boldsymbol{\Gamma} \diamond \mathbf{A}\right)^{\mathrm{T}} + \left[\tilde{\boldsymbol{\mathcal{N}}}\right]_{(3)} \in \mathbb{C}^{Q \times MK}, \tag{2.45}$$

to calculate the multimode covariance matrix $\mathbf{R}_{\text{mm}}$ [31, 64]

$$\mathbf{R}_{\text{mm}} = \frac{1}{N} \mathbf{Y}_3^{\mathrm{T}} \cdot \mathbf{Y}_3^{*}, \tag{2.46}$$

$$= \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right) \frac{1}{N} \cdot \tilde{\mathbf{C}}^{\mathrm{T}} \cdot \tilde{\mathbf{C}}^{*} \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right)^{\mathrm{H}}, \tag{2.47}$$

$$= \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right) \mathbf{R}_{\tilde{\mathbf{C}}} \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right)^{\mathrm{H}} \in \mathbb{C}^{MK \times MK}, \tag{2.48}$$

where $\mathbf{R}_{\tilde{\mathbf{C}}} \in \mathbb{R}^{L \times L}$ is the code covariance matrix.

Considering $\mathbf{R}_{\tilde{\mathbf{C}}} \approx \mathbf{I}_L$, the multimode covariance matrix is

$$\mathbf{R}_{\text{mm}} \approx \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right) \left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right)^{\mathrm{H}}, \tag{2.49}$$

which allows a Hermitian-symmetric unfolding

$$\mathbf{R}_{\text{mm}} = \mathbf{R}_{\text{mm}}^{\frac{1}{2}} \left(\mathbf{R}_{\text{mm}}^{\frac{1}{2}}\right)^{\mathrm{H}}, \tag{2.50}$$

defined by root-square factor matrix $\mathbf{R}_{\text{mm}}^{\frac{1}{2}} \in \mathbb{C}^{MK \times L}$.

$\mathbf{R}_{\text{mm}}^{\frac{1}{2}}$ can be estimated using the the left-hand singular vectors and singular values which span the signal subspace of $\mathbf{R}_{\text{mm}}$, and a unitary rotation matrix $\mathbf{W}^{\mathrm{H}}$

$$\mathbf{R}_{\text{mm}} = \mathbf{U}_R \cdot \boldsymbol{\Sigma}_R \cdot \mathbf{V}_R^{\mathrm{H}}, \tag{2.51}$$

$$\hat{\mathbf{R}}_{\text{mm}}^{\frac{1}{2}} = \mathbf{U}_R \cdot \boldsymbol{\Sigma}_R^{\frac{1}{2}} \cdot \mathbf{W}^{\mathrm{H}}. \tag{2.52}$$

Since $\mathbf{R}_{\text{mm}}^{\frac{1}{2}} = \mathbf{A} \diamond \boldsymbol{\Gamma}$, we have

$$\hat{\mathbf{R}}_{\text{mm}}^{\frac{1}{2}} = \mathbf{U}_R \cdot \boldsymbol{\Sigma}_R^{\frac{1}{2}} \cdot \mathbf{W}^{\mathrm{H}} = \mathbf{A} \diamond \boldsymbol{\Gamma}. \tag{2.53}$$

Estimating the orthogonal matrix $\mathbf{W}^{\mathrm{H}}$ which maps $\mathbf{U}_R \boldsymbol{\Sigma}_R^{\frac{1}{2}}$ to $\mathbf{A} \diamond \boldsymbol{\Gamma}$ is known as the OPP and has a known solution [29]

$$\left(\mathbf{A} \diamond \boldsymbol{\Gamma}\right)^{\mathrm{H}} \left(\mathbf{U}_R \cdot \boldsymbol{\Sigma}_R^{\frac{1}{2}}\right) = \mathbf{U}_P \cdot \boldsymbol{\Sigma}_P \cdot \mathbf{V}_P^{\mathrm{H}}, \tag{2.54}$$

$$\hat{\mathbf{W}}^{\mathrm{H}} = \mathbf{U}_P \cdot \mathbf{V}_P^{\mathrm{H}}. \tag{2.55}$$

By initializing $\mathbf{W}_0^{\mathrm{H}} = \mathbf{I}_L$, then alternating between applying Khatri-Rao factorization

(KRF) to eq. (2.53) to update $\hat{\mathbf{A}}$ and $\hat{\mathbf{\Gamma}}$, and updating $\mathbf{W}^{\mathrm{H}}$ using the known OPP solution, we can iteratively update estimates of $\mathbf{A}$ and $\mathbf{\Gamma}$.

With eq. (2.45) and the estimates of $\mathbf{A}$ and $\mathbf{\Gamma}$, we can estimate $\tilde{\mathbf{C}}$

$$\hat{\tilde{\mathbf{C}}} = \mathbf{Y}_3 \left(\left(\hat{\mathbf{A}} \diamond \hat{\mathbf{\Gamma}} \right)^{\mathrm{T}} \right)^{\dagger}. \tag{2.56}$$

LOS selection and time-delay estimation is performed as described previously in Section 2.2.2.

The ProKRaft approach to time-delay estimation is illustrated as a block diagram in Figure 2.4.

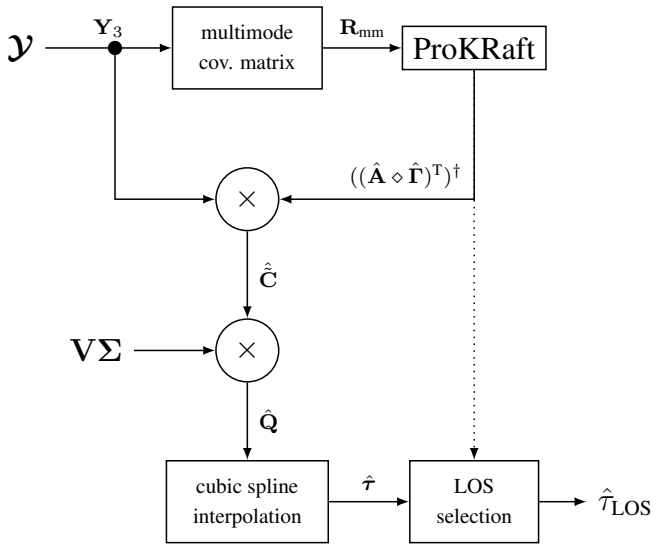

Figure 2.4: Block diagram of the state-of-the-art ProKRaft TDE approach

2.2.4 Semi-algebraic framework for approximate canonical polyadic decomposition via simultaneous matrix diagonalization (SECSI)

It is possible to calculate a low multilinear rank approximation (LMLRA) [65, 66] of the tensor $\mathcal{Y}$ via a truncated MLSVD of rank L

$$\mathcal{Y} \approx \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{2.57}$$

with core tensor $\mathcal{S} \in \mathbb{C}^{L \times L \times L}$, and singular vector matrices $\mathbf{U}_1 \in \mathbb{C}^{M \times L}$, $\mathbf{U}_2 \in \mathbb{C}^{K \times L}$, and $\mathbf{U}_3 \in \mathbb{C}^{Q \times L}$. Equation (2.57) holds exactly in the absence of noise.

Since $\mathbf{A}$, $\mathbf{\Gamma}$, and $\tilde{\mathbf{C}}$ span the same column space as $\mathbf{U}_1$, $\mathbf{U}_2$, and $\mathbf{U}_3$ respectively, there exists nonsingular transformations matrices $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$ such that $\mathbf{A} = \mathbf{U}_1 \cdot \mathbf{T}_1$, $\mathbf{\Gamma} = \mathbf{U}_2 \cdot \mathbf{T}_2$, and $\tilde{\mathbf{C}} = \mathbf{U}_3 \cdot \mathbf{T}_3$.

Placing the above relations into eq. (2.16) and disregarding noise,

$$\boldsymbol{\mathcal{Y}}_0 = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{U}_1 \cdot \mathbf{T}_1 \times_2 \mathbf{U}_2 \cdot \mathbf{T}_2 \times_3 \mathbf{U}_3 \cdot \mathbf{T}_3, \tag{2.58}$$

$$= (\boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{T}_3) \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{2.59}$$

$$= \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{2.60}$$

with equality holding since noise is absent. Moreover, it is clear that the core tensor has the following decomposition

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{T}_3 \tag{2.61}$$

and therefore $\mathbf{T}_1^{-1}$, $\mathbf{T}_2^{-1}$, and $\mathbf{T}_3^{-1}$ diagonalizes the core tensor

$$\boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{T}_3^{-1} = \boldsymbol{\mathcal{I}}_{3,L}. \tag{2.62}$$

The third-mode product of eq. (2.62) and $\tilde{\mathbf{C}}$ is

$$\boldsymbol{\mathcal{I}}_{3,L} \times_3 \tilde{\mathbf{C}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{T}_3^{-1} \times_3 \tilde{\mathbf{C}}, \tag{2.63}$$

$$= \boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \tilde{\mathbf{C}} \cdot \mathbf{T}_3^{-1}, \tag{2.64}$$

$$= \boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{U}_3 \cdot \mathbf{T}_3 \cdot \mathbf{T}_3^{-1}, \tag{2.65}$$

$$= \boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{U}_3 \in \mathbb{C}^{L \times L \times Q}. \tag{2.66}$$

Selecting the q-th slice of the 3rd mode of eq. (2.66)

$$\boldsymbol{\mathcal{I}}_{3,L} \times_3 \tilde{\mathbf{C}} \times_3 \mathbf{e}_q^{\mathrm{T}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{U}_3 \times_3 \mathbf{e}_q^{\mathrm{T}} \in \mathbb{C}^{L \times L}. \tag{2.67}$$

Using the first-mode unfolding to extract the selected slice

$$\left[\boldsymbol{\mathcal{I}}_{3,L} \times_3 \tilde{\mathbf{C}} \times_3 \mathbf{e}_q^{\mathrm{T}}\right]_{(1)} = \left[\boldsymbol{\mathcal{S}} \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_3 \mathbf{U}_3 \times_3 \mathbf{e}_q^{\mathrm{T}}\right]_{(1)}, \tag{2.68}$$

$$\left[\boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{I}_L \times_2 \mathbf{I}_L \times_3 \tilde{\mathbf{C}} \times_3 \mathbf{e}_q^{\mathrm{T}}\right]_{(1)} = \mathbf{T}_1^{-1} \left[\boldsymbol{\mathcal{S}} \times_3 \mathbf{U}_3 \times_3 \mathbf{e}_q^{\mathrm{T}}\right]_{(1)} \mathbf{T}_2^{-\mathrm{T}}, \tag{2.69}$$

$$\mathbf{I}_L \left(\mathbf{I}_L \diamond \tilde{\mathbf{C}}(q, \cdot)\right)^{\mathrm{T}} = \mathbf{T}_1^{-1} \cdot \mathbf{S}_{3,q} \cdot \mathbf{T}_2^{-\mathrm{T}} = \mathcal{D}_q \left\{\tilde{\mathbf{C}}\right\}, \tag{2.70}$$

and therefore $\mathbf{S}_{3,q} = \mathbf{T}_1 \cdot \mathcal{D}_q \left\{\tilde{\mathbf{C}}\right\} \cdot \mathbf{T}_2^{\mathrm{T}}$, which is an asymmetric diagonalization problem.

To make this a symmetric diagonalization problem, we first define a matrix $\mathbf{S}_{3,p} = \mathbf{T}_1 \cdot \mathcal{D}_p \left\{\tilde{\mathbf{C}}\right\} \cdot \mathbf{T}_2^{\mathrm{T}}$, and use its inverse, $\mathbf{S}_{3,p}^{-1} = \mathbf{T}_2^{-\mathrm{T}} \cdot \mathcal{D}_p \left\{\tilde{\mathbf{C}}\right\}^{-1} \cdot \mathbf{T}_1^{-1}$, to make a right-hand side

(rhs) symmetric diagonalization problem

$$\mathbf{S}_{3,q}^{\text{rhs}} = \mathbf{S}_{3,q} \cdot \mathbf{S}_{2,p}^{-1} = \mathbf{T}_1 \cdot \mathcal{D}_q \left\{ \tilde{\mathbf{C}} \right\} \cdot \mathbf{T}_2^{\text{T}} \cdot \mathbf{T}_2^{-\text{T}} \cdot \mathcal{D}_p \left\{ \tilde{\mathbf{C}} \right\}^{-1} \cdot \mathbf{T}_1^{-1}, \tag{2.71}$$

$$= \mathbf{T}_1 \cdot \mathcal{D}_q \left\{ \tilde{\mathbf{C}} \right\} \cdot \mathcal{D}_p \left\{ \tilde{\mathbf{C}} \right\}^{-1} \cdot \mathbf{T}_1^{-1}, \tag{2.72}$$

$$= \mathbf{T}_1 \cdot \tilde{\mathbf{C}}_i^{\text{D},p} \cdot \mathbf{T}_1^{-1}, \; \tilde{\mathbf{C}}_i^{\text{D},p} = \mathcal{D}_q \left\{ \tilde{\mathbf{C}} \right\} \cdot \mathcal{D}_p \left\{ \tilde{\mathbf{C}} \right\}^{-1}. \tag{2.73}$$

The tensor slice p used to make the diagonalization problem in eq. (2.73) symmetric is the called the "pivot" slice [67]. To ensure the pivot slice is well-conditioned [68], the third-mode slice of eq. (2.66) with smallest condition number is selected

$$p = \underset{q}{\arg\min} \operatorname{cond} \left\{ \mathbf{S}_{3,q} \right\}. \tag{2.74}$$

The result is a set of $Q-1$ matrices whose diagonalization yield the estimate of $\mathbf{T}_1$ which simultaneously diagonalizes the entire set, and is also the nonsingular transformation matrix which estimates $\mathbf{A}$,

$$\mathbf{S}_{3,1}^{\text{rhs}} = \mathbf{T}_1 \cdot \mathbf{C}_1^{\text{D},p} \cdot \mathbf{T}_1^{-1}, \tag{2.75}$$

$$\mathbf{S}_{3,1}^{\text{rhs}} = \mathbf{T}_1 \cdot \mathbf{C}_2^{\text{D},p} \cdot \mathbf{T}_1^{-1}, \tag{2.76}$$

$$\vdots \tag{2.77}$$

$$\mathbf{S}_{3,Q}^{\text{rhs}} = \mathbf{T}_1 \cdot \mathbf{C}_Q^{\text{D},p} \cdot \mathbf{T}_1^{-1}. \tag{2.78}$$

With $\hat{\mathbf{T}}_1$, $\mathbf{A}$ can be estimated

$$\hat{\mathbf{A}} = \mathbf{U}_1 \cdot \hat{\mathbf{T}}_1, \tag{2.79}$$

whose pseudoinverse left-multiplying the first-mode unfolding of $\boldsymbol{\mathcal{Y}}$ yields the Khatri-Rao product of $\boldsymbol{\Gamma}$ and $\tilde{\mathbf{C}}$

$$\hat{\boldsymbol{\Gamma}} \diamond \hat{\tilde{\mathbf{C}}} = \left(\hat{\mathbf{A}} \cdot \mathbf{Y}_1 \right)^{\text{T}} \tag{2.80}$$

which are then estimated using least-squares Khatri-Rao factorization described in Algorithm 1.

The tensor-based approach to time-delay estimation using SECSI is illustrated as a block diagram in Figure 2.5.

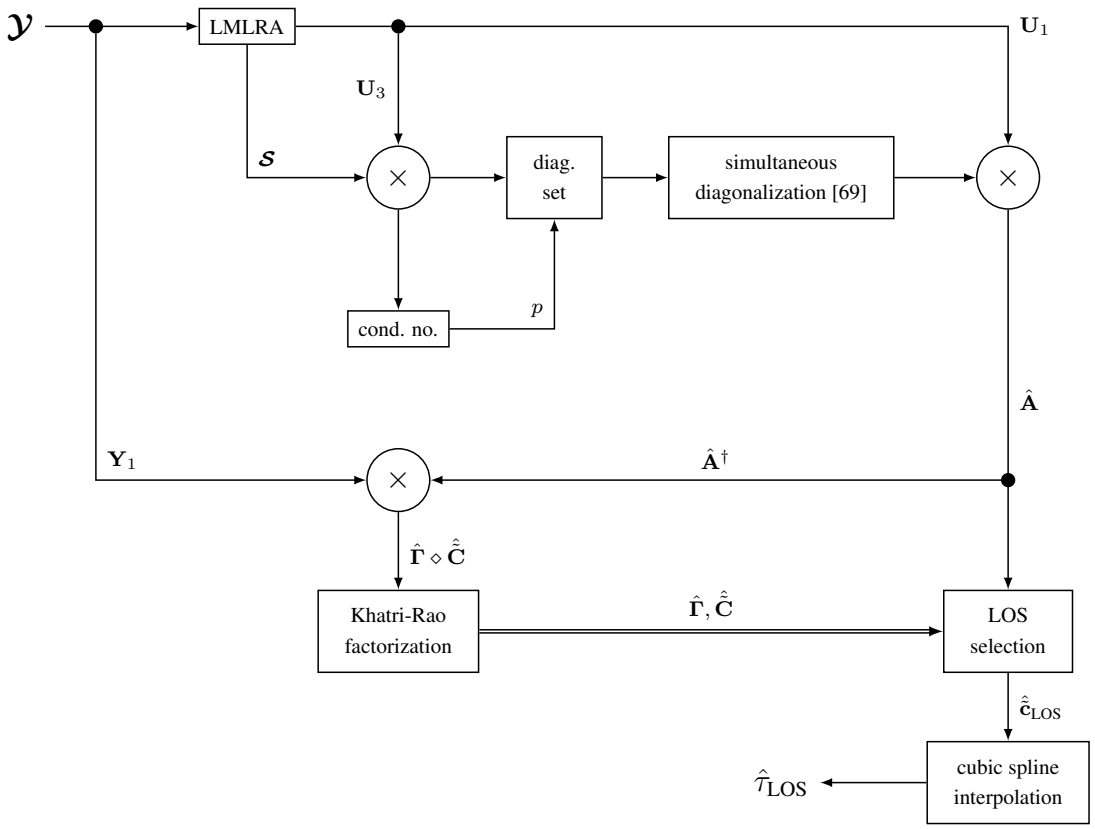

Figure 2.5: Block diagram of the state-of-the-art SECSI TDE approach

2.3 Proposed Tensor-Based Time-Delay Estimation

In this section the proposed tensor-based time-delay estimation scheme is presented, along with a variant that utilizes a more balanced approach.

2.3.1 Canonical polyadic decomposition via generalized eigenvalue decomposition

Similarly to SECSI-based TDE, described in Section 2.2.4, CPD-GEVD-based TDE also utilizes joint diagonalization to estimate factor matrices. Unlike SECSI, it utilizes the generalized eigenvalue decomposition or exact joint diagonalization [69] of the first two slices of the core tensor obtained via the truncated MLSVD and does not perform any n-mode product of the core tensor with any of the singular values matrices.

Continuing from eq. (2.61) and utilizing the notation $\mathbf{S}_\ell$ to represent the ℓ-th third-mode

slice of the core tensor $\boldsymbol{\mathcal{S}}$

$$\begin{aligned}
\mathbf{S}_\ell &= \boldsymbol{\mathcal{S}}(\cdot,\cdot,\ell), && (2.81)\\
&= \left[\boldsymbol{\mathcal{S}} \times_3 \mathbf{e}_\ell^{\mathrm{T}}\right]_{(1)}, && (2.82)\\
&= \left[\boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{e}_\ell^{\mathrm{T}}\mathbf{T}_3\right]_{(1)}, && (2.83)\\
&= \left[\boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{T}_3(\ell,\cdot)\right]_{(1)}, && (2.84)\\
&= \mathbf{T}_1 \left(\mathbf{T}_2 \diamond \mathbf{T}_3(\ell,\cdot)\right)^{\mathrm{T}}, && (2.85)\\
&= \mathbf{T}_1 \left(\mathbf{T}_2 \cdot \mathcal{D}_\ell\left\{\mathbf{T}_3\right\}\right)^{\mathrm{T}}, && (2.86)\\
&= \mathbf{T}_1 \cdot \mathcal{D}_\ell\left\{\mathbf{T}_3\right\} \cdot \mathbf{T}_2^{\mathrm{T}}, && (2.87)
\end{aligned}$$

we calculate the generalized eigenvalue decomposition of the transposed first two slices of the core tensor $\boldsymbol{\mathcal{S}}$, the matrix pencil [70] $(\mathbf{S}_1^{\mathrm{T}}, \mathbf{S}_2^{\mathrm{T}})$

$$\mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{S}_2^{\mathrm{T}} \cdot \mathbf{R} \cdot \mathbf{D} \tag{2.88}$$

where $\mathbf{R} \in \mathbb{C}^{L\times L}$ is the (right) eigenvectors matrix and $\mathbf{D} \in \mathbb{C}^{L\times L}$ is the eigenvalues matrix.

Changing the order of the products in eq. (2.88)

$$\begin{aligned}
\mathbf{S}_2^{-\mathrm{T}} \cdot \mathbf{S}_1^{\mathrm{T}} &= \mathbf{R} \cdot \mathbf{D} \cdot \mathbf{R}^{-1} && (2.89)\\
\mathbf{S}_1 \cdot \mathbf{S}_2^{-1} &= \mathbf{R}^{-\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{R}^{\mathrm{T}} && (2.90)\\
&= \mathbf{T}_1 \cdot \mathcal{D}_1\left\{\mathbf{T}_3\right\} \cdot \mathcal{D}_2\left\{\mathbf{T}_3\right\}^{-1} \cdot \mathbf{T}_1^{-1} && (2.91)
\end{aligned}$$

we have estimates

$$\hat{\mathbf{T}}_1 = \mathbf{R}^{-\mathrm{T}}, \qquad\qquad \hat{\mathbf{T}}_1^{-\mathrm{T}} = \mathbf{R}, \tag{2.92}$$

while allowing trivial scaling ambiguities and $\hat{\mathbf{A}} = \mathbf{U}_1 \cdot \hat{\mathbf{T}}_1 = \mathbf{U}_1 \cdot \mathbf{R}^{-\mathrm{T}}$.

With the estimate $\hat{\mathbf{T}}_1^{-\mathrm{T}} = \mathbf{R}$ we can make an estimate of the transpose pseudoinverse of the array response factor matrix $\mathbf{A}$

$$\begin{aligned}
\left(\hat{\mathbf{A}}^{\dagger}\right)^{\mathrm{T}} &= \left(\left(\mathbf{U}_1 \cdot \hat{\mathbf{T}}_1\right)^{\dagger}\right)^{\mathrm{T}} && (2.93)\\
&= \left(\hat{\mathbf{T}}_1^{-1} \cdot \mathbf{U}_1^{\dagger}\right)^{\mathrm{T}} && (2.94)\\
&= \left(\hat{\mathbf{T}}_1^{-1} \cdot \mathbf{U}_1^{\mathrm{H}}\right)^{\mathrm{T}} && (2.95)\\
&= \mathbf{U}_1^{*} \cdot \hat{\mathbf{T}}_1^{-\mathrm{T}} && (2.96)\\
&= \mathbf{U}_1^{*} \cdot \mathbf{R}. && (2.97)
\end{aligned}$$

Multiplying the transpose of the first-mode unfolding of $\boldsymbol{\mathcal{Y}}$, $\mathbf{Y}_1^{\mathrm{T}}$ by eq. (2.97)

$$\mathbf{Y}_1^{\mathrm{T}} \cdot \mathbf{U}_1^{*} \cdot \mathbf{R} = \left(\boldsymbol{\Gamma} \diamond \tilde{\mathbf{C}}\right) \mathbf{A}^{\mathrm{T}} \left(\hat{\mathbf{A}}^{\mathrm{T}}\right)^{\dagger} \tag{2.98}$$

$$= \hat{\boldsymbol{\Gamma}} \diamond \hat{\tilde{\mathbf{C}}} \in \mathbb{C}^{KQ \times L}, \tag{2.99}$$

with which we can estimate $\hat{\boldsymbol{\Gamma}}$ and $\hat{\tilde{\mathbf{C}}}$ utilizing LSKRF described above in Algorithm 1.

LOS selection and time-delay estimation is performed as described previously in Section 2.2.2.

2.3.2 Dual eigenvector matrix variant CPD-GEVD

While the approach described in Section 2.3.1 only utilizes the right eigenvector matrix of the generalized eigenvalue decomposition (GEVD), it is possible to both left and right eigenvector matrices.

Following the LMLRA of $\boldsymbol{\mathcal{Y}}$

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{2.100}$$

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{T}_3, \tag{2.101}$$

we calculate the GEVD of the matrix pencil of $(\mathbf{S}_1, \mathbf{S}_2)$.

Using the left eigenvector matrix $\mathbf{L}$,

$$\mathbf{L}^{\mathrm{H}} \cdot \mathbf{S}_1 = \mathbf{D} \cdot \mathbf{L}^{\mathrm{H}} \cdot \mathbf{S}_2, \tag{2.102}$$

$$\mathbf{S}_1 \cdot \mathbf{S}_2^{-1} = \mathbf{L}^{-\mathrm{H}} \cdot \mathbf{D} \cdot \mathbf{L}^{\mathrm{H}}, \tag{2.103}$$

$$= \mathbf{T}_1 \cdot \mathcal{D}_1\left\{\mathbf{T}_3\right\} \cdot \mathcal{D}_2\left\{\mathbf{T}_3\right\}^{-1} \cdot \mathbf{T}_1^{-1}, \tag{2.104}$$

and therefore we have an estimate of $\mathbf{T}_1$,

$$\hat{\mathbf{T}}_1 = \mathbf{L}^{-\mathrm{H}}. \tag{2.105}$$

Using the right eigenvector matrix $\mathbf{R}$,

$$\mathbf{S}_1 \cdot \mathbf{R} = \mathbf{S}_2 \cdot \mathbf{R} \cdot \mathbf{D}, \tag{2.106}$$

$$\mathbf{S}_2^{-1} \cdot \mathbf{S}_1 = \mathbf{R} \cdot \mathbf{D} \cdot \mathbf{R}^{-1}, \tag{2.107}$$

$$\mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{S}_2^{-\mathrm{T}} = \mathbf{R}^{-\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{R}^{\mathrm{T}}, \tag{2.108}$$

$$= \mathbf{T}_2 \cdot \mathcal{D}_1\left\{\mathbf{T}_3\right\} \cdot \mathcal{D}_2\left\{\mathbf{T}_3\right\}^{-1} \cdot \mathbf{T}_2^{-1}, \tag{2.109}$$

and therefore we have an estimate of $\mathbf{T}_2$,

$$\hat{\mathbf{T}}_2 = \mathbf{R}^{-\mathrm{T}}. \tag{2.110}$$

The transform matrix $\mathbf{T}_3$ is estimated using the third-mode unfolding of the core tensor $\boldsymbol{\mathcal{S}}$ and the previously estimated transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$,

$$\hat{\mathbf{T}}_3 = [\boldsymbol{\mathcal{S}}]_{(3)} \left(\left(\hat{\mathbf{T}}_1 \diamond \hat{\mathbf{T}}_2 \right)^{\mathrm{T}} \right)^{\dagger}, \tag{2.111}$$

$$= \mathbf{T}_3 \left(\mathbf{T}_1 \diamond \mathbf{T}_2 \right)^{\mathrm{T}} \left(\left(\hat{\mathbf{T}}_1 \diamond \hat{\mathbf{T}}_2 \right)^{\mathrm{T}} \right)^{\dagger}, \tag{2.112}$$

$$\approx \mathbf{T}_3. \tag{2.113}$$

The factor matrices are then estimated by expanding the previously estimated subspaces in eq. (2.57) using each respective transformation matrix

$$\hat{\mathbf{A}} = \mathbf{U}_1 \cdot \hat{\mathbf{T}}_1, \tag{2.114}$$

$$\hat{\mathbf{\Gamma}} = \mathbf{U}_2 \cdot \hat{\mathbf{T}}_2, \tag{2.115}$$

$$\hat{\tilde{\mathbf{C}}} = \mathbf{U}_3 \cdot \hat{\mathbf{T}}_3. \tag{2.116}$$

2.4 Numerical Simulation Results

In this section we perform numerical simulation and corresponding results using the proposed approaches compared with the state-of-the art approaches. The generation of each of the factor matrices which form the data tensor is detailed in Sections 2.4.1 to 2.4.3. The scenario is detailed in Section 2.4.4, with the parameters used for the simulation.

2.4.1 Array response matrix generation

The simulations with a perfectly calibrated array, spacing between array elements are an exact half-wavelength, thus the array response matrix, $\mathbf{A} \in \mathbb{C}^{M \times L}$, is a left $\mathbf{\Pi}$ real symmetric matrix, that is, given an exchange matrix,

$$\mathbf{\Pi} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{M \times M}, \tag{2.117}$$

with $\mathbf{\Pi} \cdot \mathbf{\Pi} = \mathbf{I}_M$, the following relation holds

$$\mathbf{\Pi}\mathbf{A}^* = \mathbf{A}. \tag{2.118}$$

A separation angle $\Delta\phi$ between the L signals can be enforced, thus avoiding the possibility of rank deficiency. This may help avoid the ill-posedness of the CPD [71, 72] without completely ruling it out.

For the simulation with an imperfect array, a random shift is added to the spacing of each array element, thus the array is no longer calibrated. These element placement errors impacts array response.

2.4.2 Complex amplitude matrix generation

The complex amplitude matrix $\mathbf{\Gamma} \in \mathbb{C}^{K \times L}$ collects the complex amplitudes (magnitude and phase shift) of the L signals during the K epoch sampling interval.

The magnitudes are kept constant during sampling for each ℓ-th signal while the phase shifts are drawn from $\sim \mathcal{U}[0, 2\pi)$. Because the phase shifts are iid this ensures $\mathbf{\Gamma}$ is almost surely full-rank.

2.4.3 Code (PRS) matrix generation

Each ℓ-th column of $\mathbf{C} \in \mathbb{R}^{N \times L}$ represents a same PRS temporally shifted by a delay τ_ℓ. The first delay is the LOS delay, which is drawn from $\tau_{\text{LOS}} \sim \mathcal{U}[-T_C, 0]$, in which T_C is the time length of one chip period of the PRS.

For $L = 2$, there is only one NLOS time-delay τ_{NLOS}. For simulation purposes τ_{NLOS} is selected with a delay relative to τ_{LOS} so that

$$\tau_{\text{NLOS}} = \tau_{\text{LOS}} + \Delta\tau. \tag{2.119}$$

Details on the implementation of the signal generator are found in Appendix A.

2.4.4 Simulation scenario

A 2000 realization Monte Carlo simulation is performed to compare performance in time-delay estimation and tensor reconstruction, given the estimated factor matrices.

The array is a Vandermonde left centro-hermitian uniform linear array with $M = 8$ elements. An azimuth separation angle of $\Delta\phi = \pi/6$ between LOS and NLOS signals is enforced to avoid the possibility of rank deficiency in the array response matrix. This can help avoid the ill-posedness of the CPD [71, 72] without completely ruling it out.

The PRS used is GPS C/A code from satellite no. 17 with bandwidth $B = 1.023$ MHz, epoch duration $t = 1$ ms, chip period $T_C = 1/B \approx 978$ ns, sampled at $T_s = 1/2B$ resulting in $N = 2046$ samples/epoch, transmitted with carrier frequency $f_c = 1575.42$ MHz.

The correlator bank is equally spaced between $[-T_C, T_C]$ with $Q = 11$ taps.

Sampling occurs over a $K = 30$ epoch sampling period in which the signal phases $\arg\{\gamma_{k,\ell}\} \sim \mathcal{U}[0, 2\pi)$. The carrier-to-noise ratio is $C/N_0 = 48$ dB-Hz, resulting in a pre-correlation signal-to-noise ratio (SNR) of approximately -15 dB, and a post-correlation SNR of approximately $+15$ dB, and a signal-to-multipath ratio (SMR) of 5 dB.

2.4.5 Perfectly calibrated array

In this section simulations are performed assuming a perfectly calibrated array. The array response matrix is a left $\mathbf{\Pi}$ real symmetric matrix, as explained above.

2.4.5.1 Time-delay estimation performance

In this section, performance is measures in terms of time-delay estimation. The figure of merit is the root-means squared error (RMSE) of the time-delay estimation multiplied by the speed of light $c = 299.792.458$ m/s. For I realizations, the RMSE is

$$\text{RMSE} = c \cdot \sqrt{\frac{1}{I} \sum_{i=1}^{I} \left(\hat{\tau}_i - \tau_i\right)^2} \text{ (m)}. \tag{2.120}$$

LOS selection is performed by choosing the promptest delay [24] for the DoA/KRF, SECSI, and CPD-GEVD approaches. Maximizing CPD amplitudes is for the ProKRaft approach. This is also used for the CPD-GEVD (dual) approach.

Filtering with a priori knowledge of the factor matrices $\mathbf{A}$ and $\mathbf{\Gamma}$ is used to estimate the time delay. The PRS code factor matrix is estimated using

$$\hat{\tilde{\mathbf{C}}} = [\boldsymbol{\mathcal{Y}}]_{(3)} \left((\mathbf{A} \diamond \mathbf{\Gamma})^{\mathrm{T}} \right)^{\dagger}, \tag{2.121}$$

and its first column is used to estimate the time delay. This is used as a baseline comparison.

In Figure 2.6 we see the TDE results for all methods in the presence of 1 LOS and 1 NLOS component, $L = 2$.

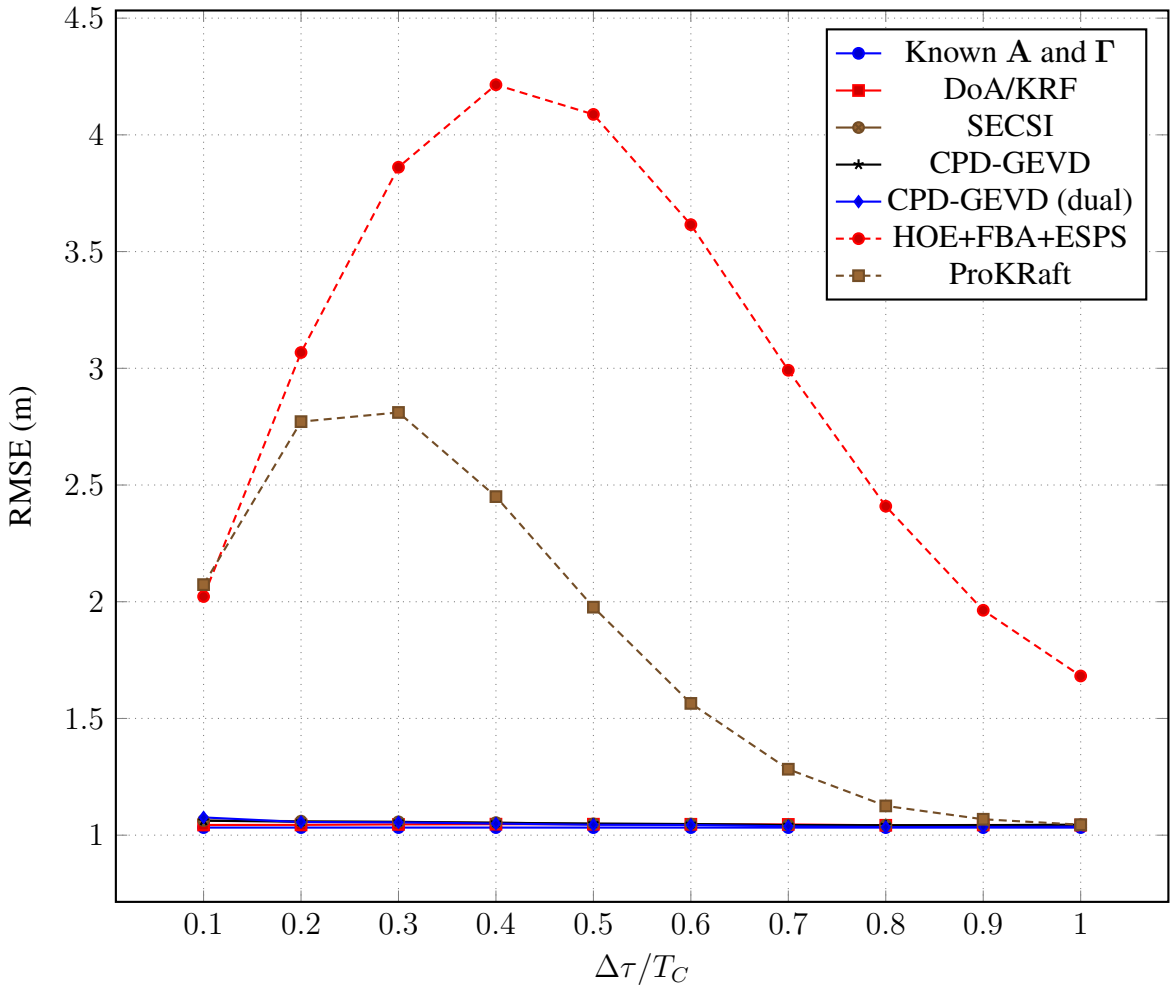

Figure 2.6: Time-delay estimation error in meters for $L = 2$.

In Figure 2.7, the HOE+FBA+ESPS and ProKRaft TDE approaches are omitted to clearly demonstrate the performance differences between the DoA/KRF, SECSI, and CPD-GEVD TDE approaches. Here the performance of the SECSI-based approach to TDE follows the performance of the CPD-GEVD approach closely. The DoA/KRF approach shows consistent performance, with CPD-GEVD matching its performance when $\Delta\tau/T_C \geq 0.6$, and it is outperformed by CPD-GEVD (dual) when $\Delta\tau/T_C \geq 0.5$.

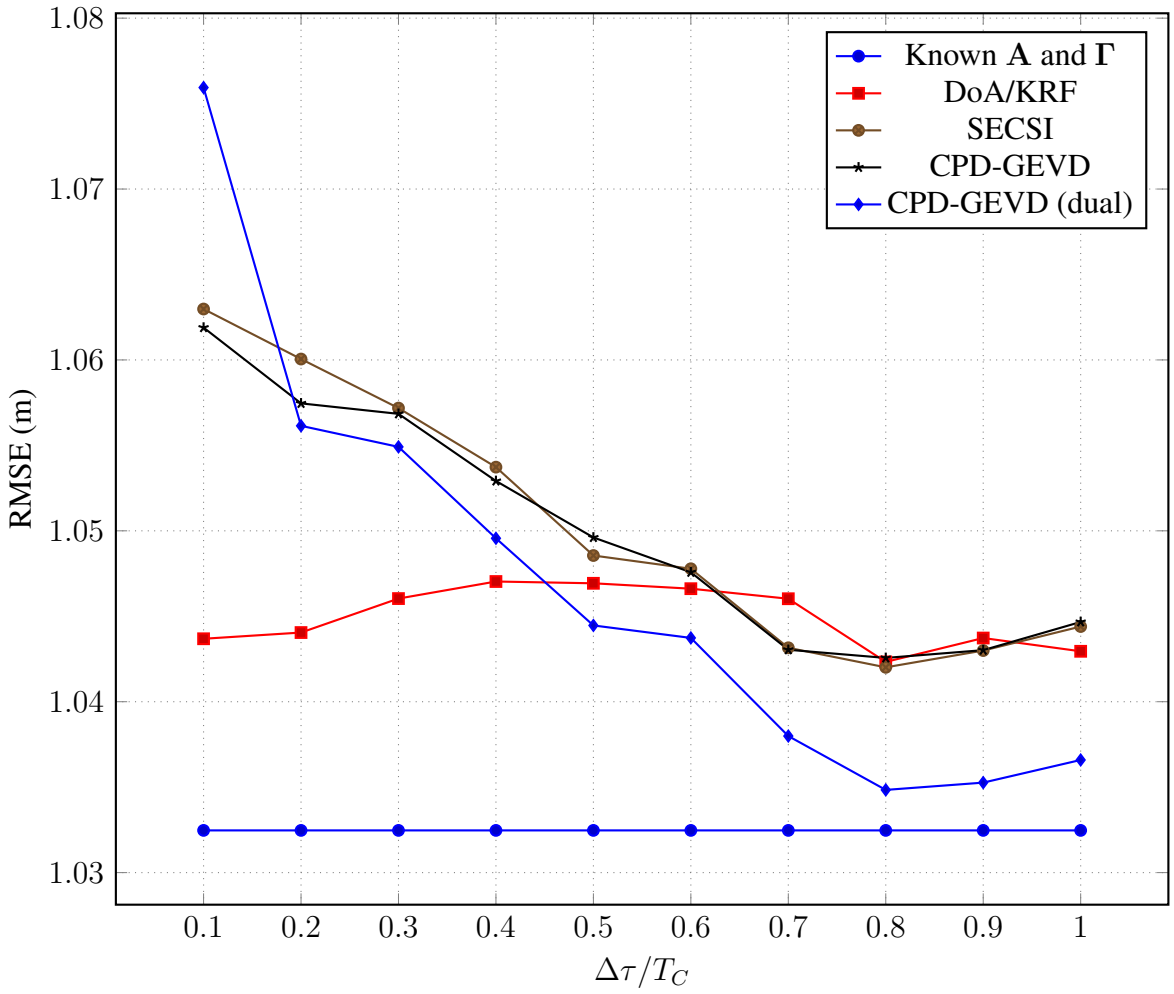

Figure 2.7: Time-delay estimation error in meters for $L = 2$, HOE+FBA+ESPS and ProKRaft omitted.

2.4.5.2 Reconstruction error performance

In this section, performance is measured in terms of root-means squared (RMS) relative reconstruction error. The relative reconstruction error at realization i is

$$e_i = \frac{\left\|\boldsymbol{\mathcal{Y}}_i - \hat{\boldsymbol{\mathcal{Y}}}_i\right\|_{\mathrm{F}}}{\|\boldsymbol{\mathcal{Y}}_i\|_{\mathrm{F}}}. \tag{2.122}$$

and its RMS is

$$\mathrm{RMS} = \sqrt{\frac{1}{I}\sum_{i=1}^{I} e_i^2}. \tag{2.123}$$

In Figure 2.8, we see the relative reconstruction error for all methods in the presence of 1 LOS and 1 NLOS component.

The reconstruction error is greatest for the higher-order eigenfilter. This is expected since

it applies a rank-1 MLSVD to data from 2 sources. The ProKRaft shows better results as the delay increases, consequently decreasing correlation between LOS and NLOS signal.

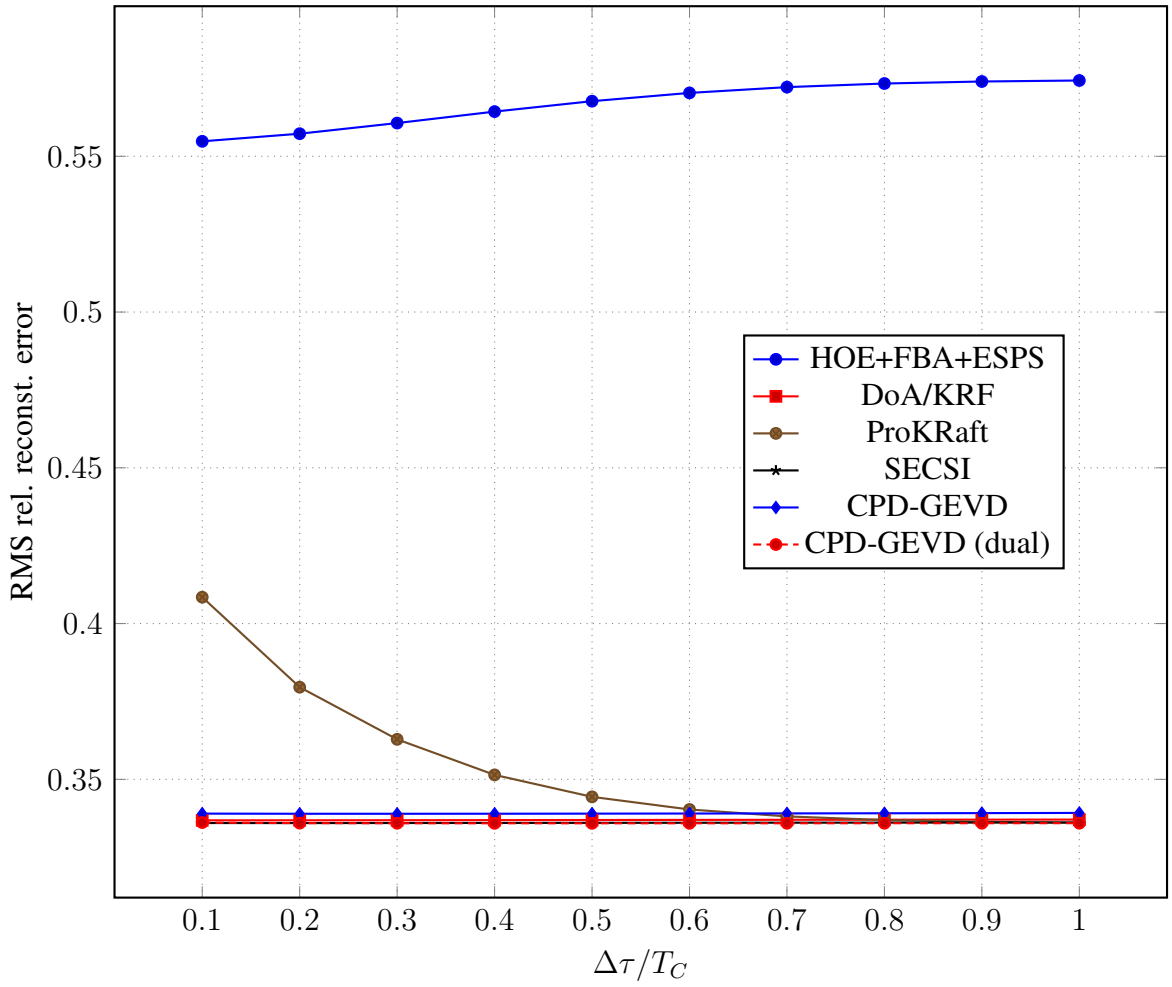

Figure 2.8: Relative reconstruction error for $L = 2$.

In Figure 2.9, the HOE+FBA+ESPS and ProKRaft approaches were omitted to clearly demonstrate the performance differences between the DoA/KRF, SECSI, and CPD-GEVD approaches. Relative reconstruction error does not necessarily track with TDE performance.

Reconstruction error for these approaches are consistent and do not correlate with the increase in TDE performance as separation between LOS and NLOS signals increases, nor are these ranked in the same order as TDE performance.

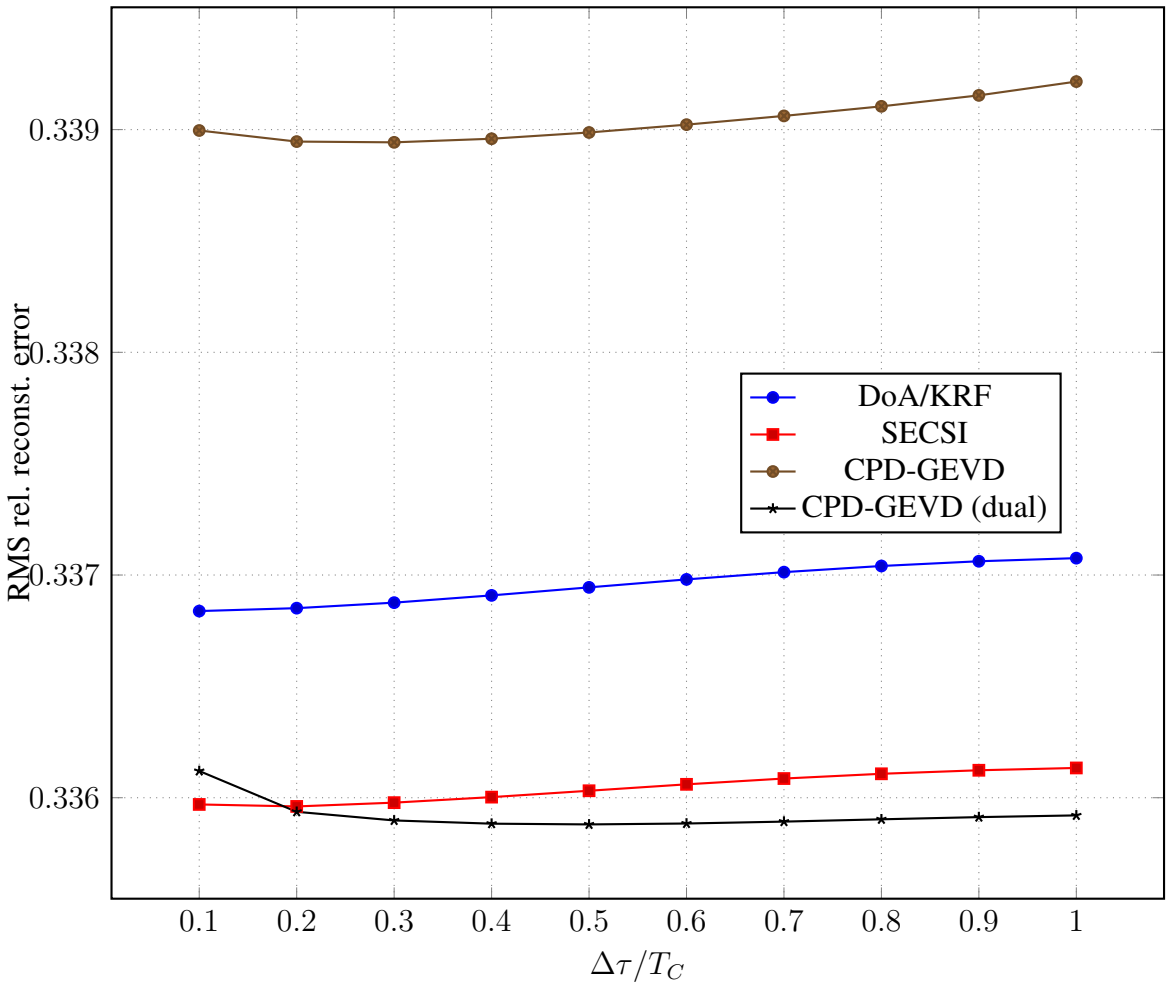

Figure 2.9: Relative reconstruction error for $L = 2$, HOE+FBA+ESPS and ProKRaft omitted.

2.4.6 Imperfect Array

In this section an imperfect array is assumed. This is expected in more realistic scenarios in which array calibration is not ensured [61].

To control for error we define the array error as the absolute difference between a position x from its expected position $\bar{x}$ and assume it has a Gaussian distribution, $e = |x - \bar{x}| \sim \mathcal{N}(\bar{x}, \sigma^2)$, and define a probability p that the array error will exceed a half-wavelength

$$p = \Pr\left(e > \frac{\lambda}{2}\right), \tag{2.124}$$

$$= \Pr\left(|x - \bar{x}| > \frac{\lambda}{2}\right), \tag{2.125}$$

$$= \Pr\left(x - \bar{x} < -\frac{\lambda}{2} \cup x - \bar{x} > \frac{\lambda}{2}\right), \tag{2.126}$$

since $x - \bar{x} < -\lambda/2$ and $x - \bar{x} > \lambda/2$ are disjoint sets

$$p = \Pr\left(x - \bar{x} < -\frac{\lambda}{2}\right) + \Pr\left(x - \bar{x} > \frac{\lambda}{2}\right). \tag{2.127}$$

Expressing the above in terms of a normalized Gaussian rv $Z = (x - \bar{x})/\sigma$

$$p = \Pr\left(\frac{x - \bar{x}}{\sigma} < -\frac{\lambda}{2\sigma}\right) + \Pr\left(\frac{x - \bar{x}}{\sigma} > \frac{\lambda}{2\sigma}\right), \tag{2.128}$$

$$= \Pr\left(Z < -\frac{\lambda}{2\sigma}\right) + \Pr\left(Z > \frac{\lambda}{2\sigma}\right), \tag{2.129}$$

$$= 1 - \Pr\left(Z > -\frac{\lambda}{2\sigma}\right) + \Pr\left(Z > \frac{\lambda}{2\sigma}\right), \tag{2.130}$$

which can be expressed in terms of the Q-function

$$p = 1 - \mathrm{Q}\left(-\frac{\lambda}{2\sigma}\right) + \mathrm{Q}\left(\frac{\lambda}{2\sigma}\right), \tag{2.131}$$

$$= 1 - \left(1 - \mathrm{Q}\left(\frac{\lambda}{2\sigma}\right)\right) + \mathrm{Q}\left(\frac{\lambda}{2\sigma}\right), \tag{2.132}$$

$$= 2\,\mathrm{Q}\left(\frac{\lambda}{2\sigma}\right). \tag{2.133}$$

Resolving for σ

$$\frac{p}{2} = \mathrm{Q}\left(\frac{\lambda}{2\sigma}\right), \tag{2.134}$$

$$\mathrm{Q}^{-1}\left(\frac{p}{2}\right) = \frac{\lambda}{2\sigma}, \tag{2.135}$$

$$\therefore \sigma = \frac{\lambda}{2\,\mathrm{Q}^{-1}\left(\frac{p}{2}\right)}. \tag{2.136}$$

2.4.6.1 Time-delay estimation performance

Simulations were performed with the same parameters as previously with the exception the array response which is now for an imperfect array. Results for all methods is shown in Figure 2.10.

TDE error for HOE+FBA+ESPS and ProKRaft approaches remain unchanged compared to the results where a perfect array is assumed. This is expected as these do not rely on the array response or its parameters to perform TDE.

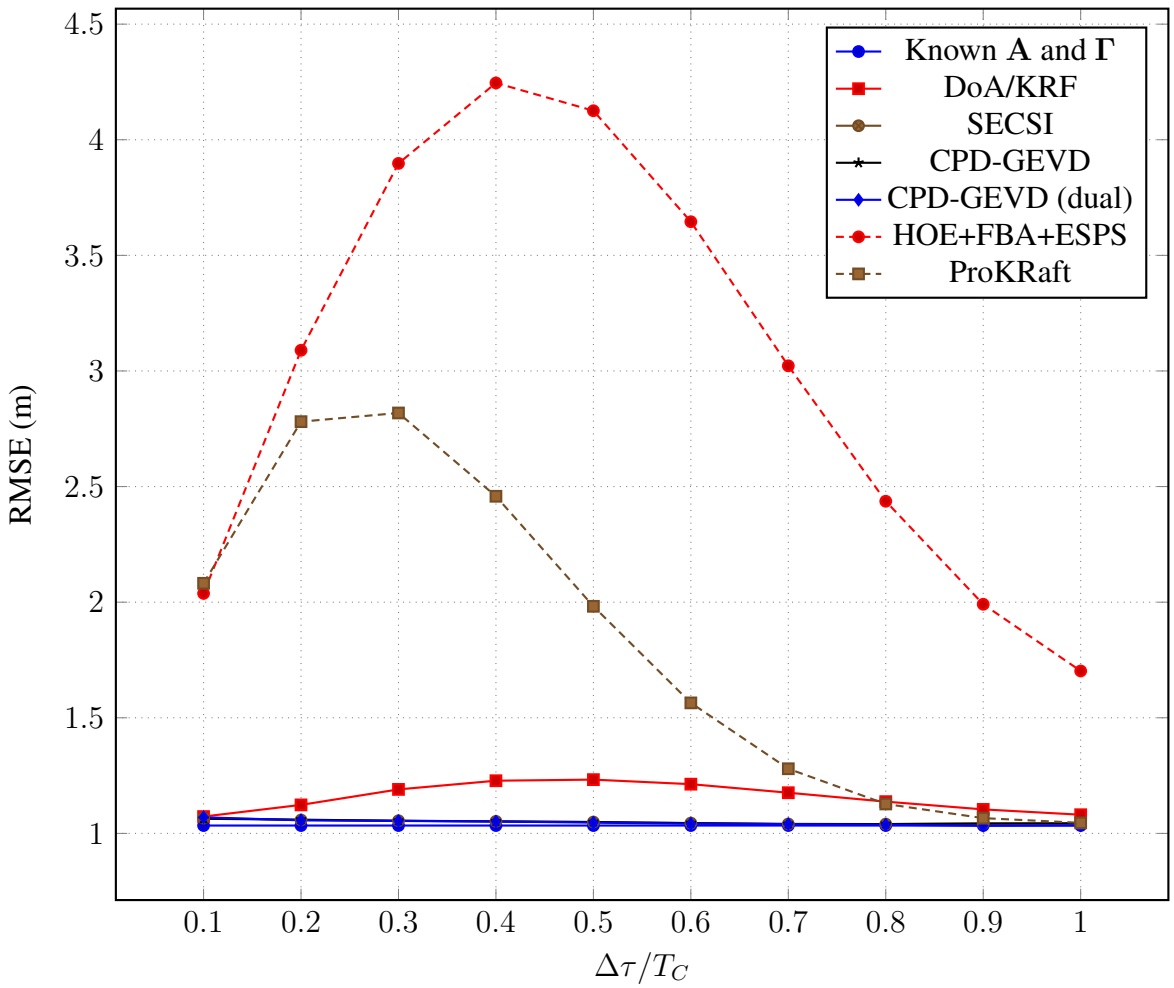

Figure 2.10: Time-delay estimation error in meters for $L = 2$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$.

In **??** the TDE results for HOE+FBA+ESPS and ProKRaft approaches have been omitted for clarity. While the algebraic approaches have yielded similar results as previously, the DoA/KRF approach now displays not only a worse but also inconsistent performance, with a bell-shaped error curve.

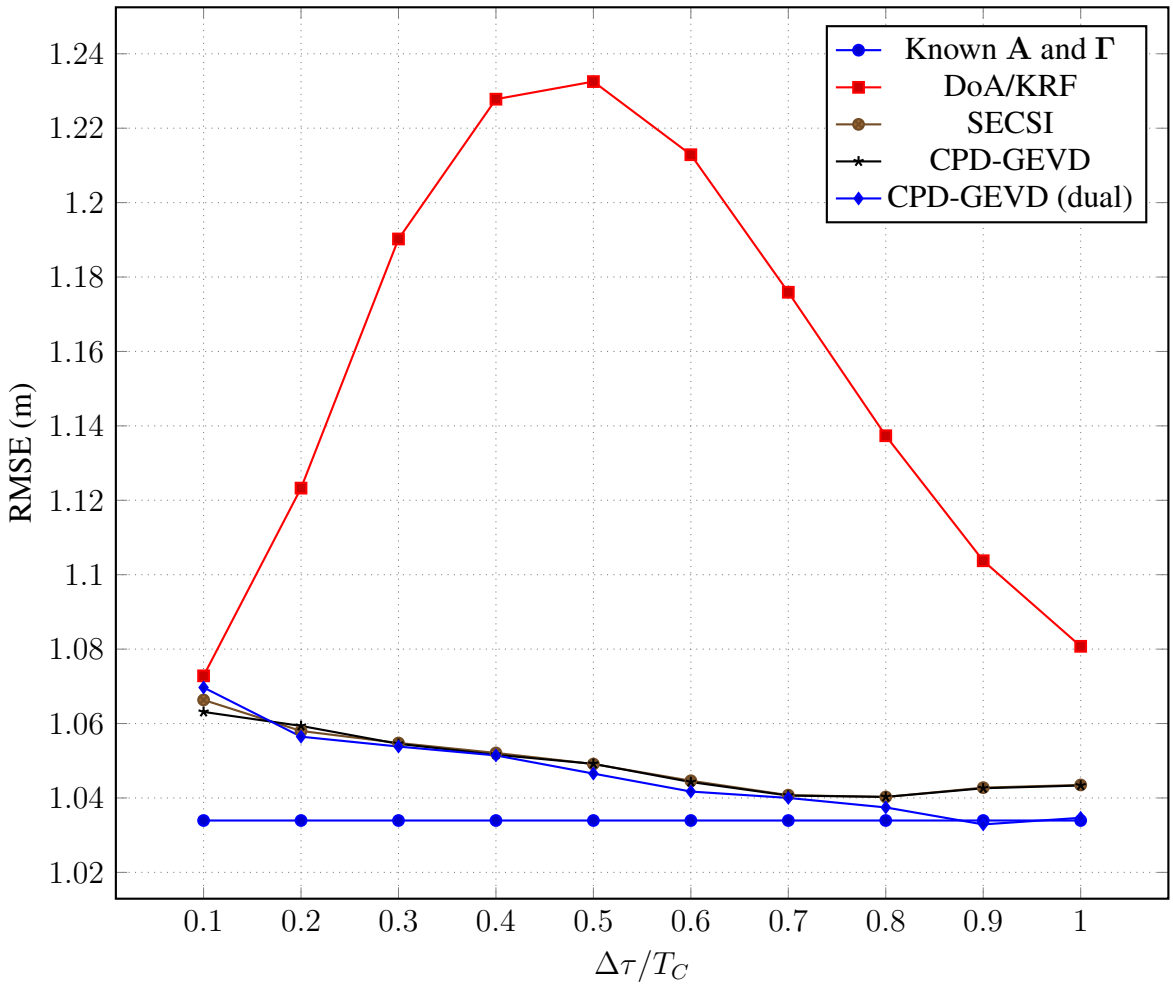

Figure 2.11: Time-delay estimation error in meters for $L = 2$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$, HOE and ProKRaft performance omitted.

2.4.6.2 Reconstruction error performance

In Figure 2.12, we see the relative reconstruction error for all methods in the presence of 1 LOS and 1 NLOS component.

Again relative reconstruction error is greatest for the higher-order eigenfilter and the ProKRaft approach has a decreasing curve, similarly to the results of the simulation using a perfect array.

For the other approaches there is an increase in relative reconstruction error. This is particularly evident for the DoA/KRF approach, which suffers greater errors in its DoA estimation and consequently reconstruction of the array response factor matrix.

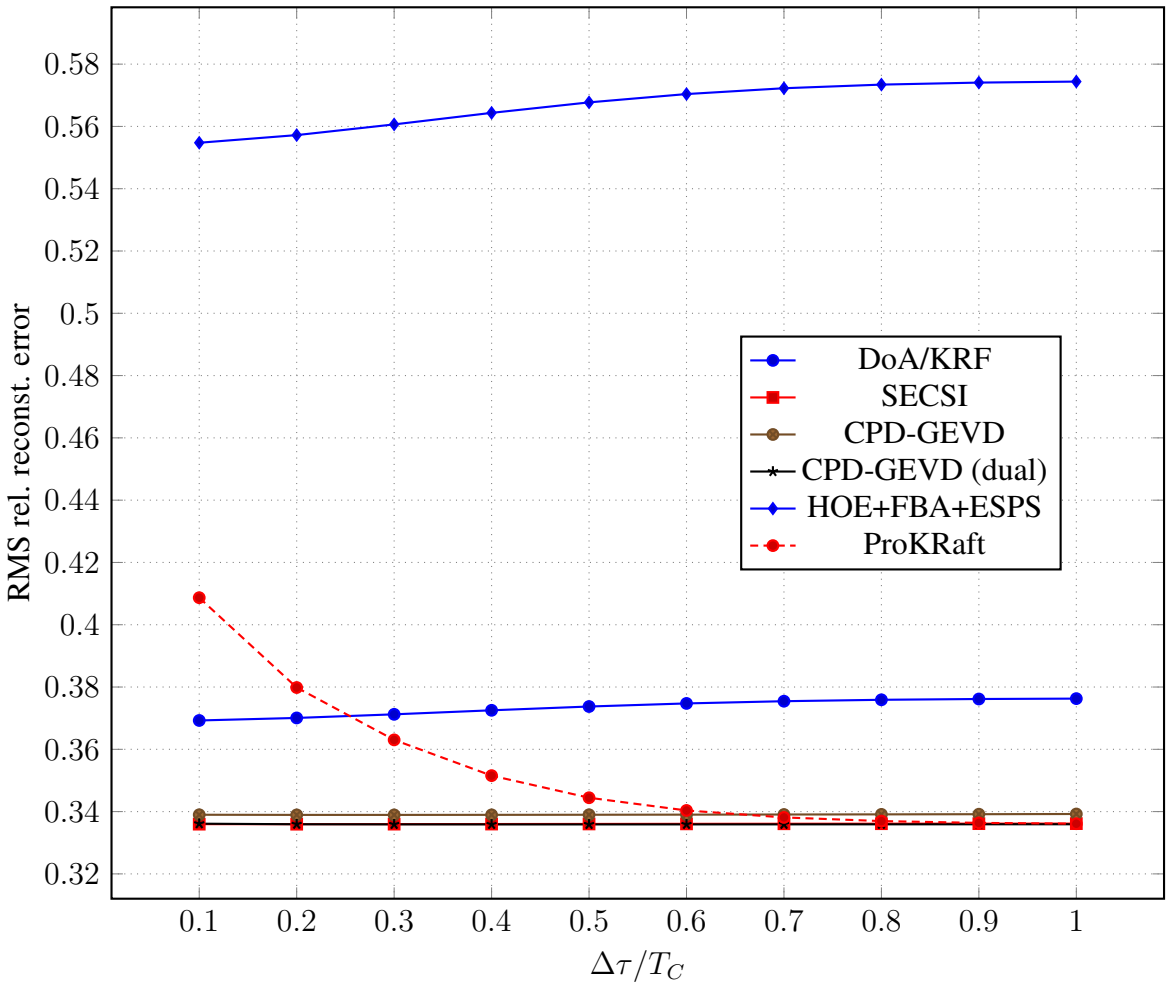

Figure 2.12: Relative reconstruction error for $L = 2$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$.

In Figure 2.13, the HOE+FBA+ESPS and ProKRaft approaches have been omitted for clarity. There was a slight increase in reconstruction error for the algebraic approaches, and a large increase for the DoA/KRF approach. In the previous section, the reconstruction error for the DoA/KRF approach was between that of the CPD-GEVD approaches, now it is greater than both.

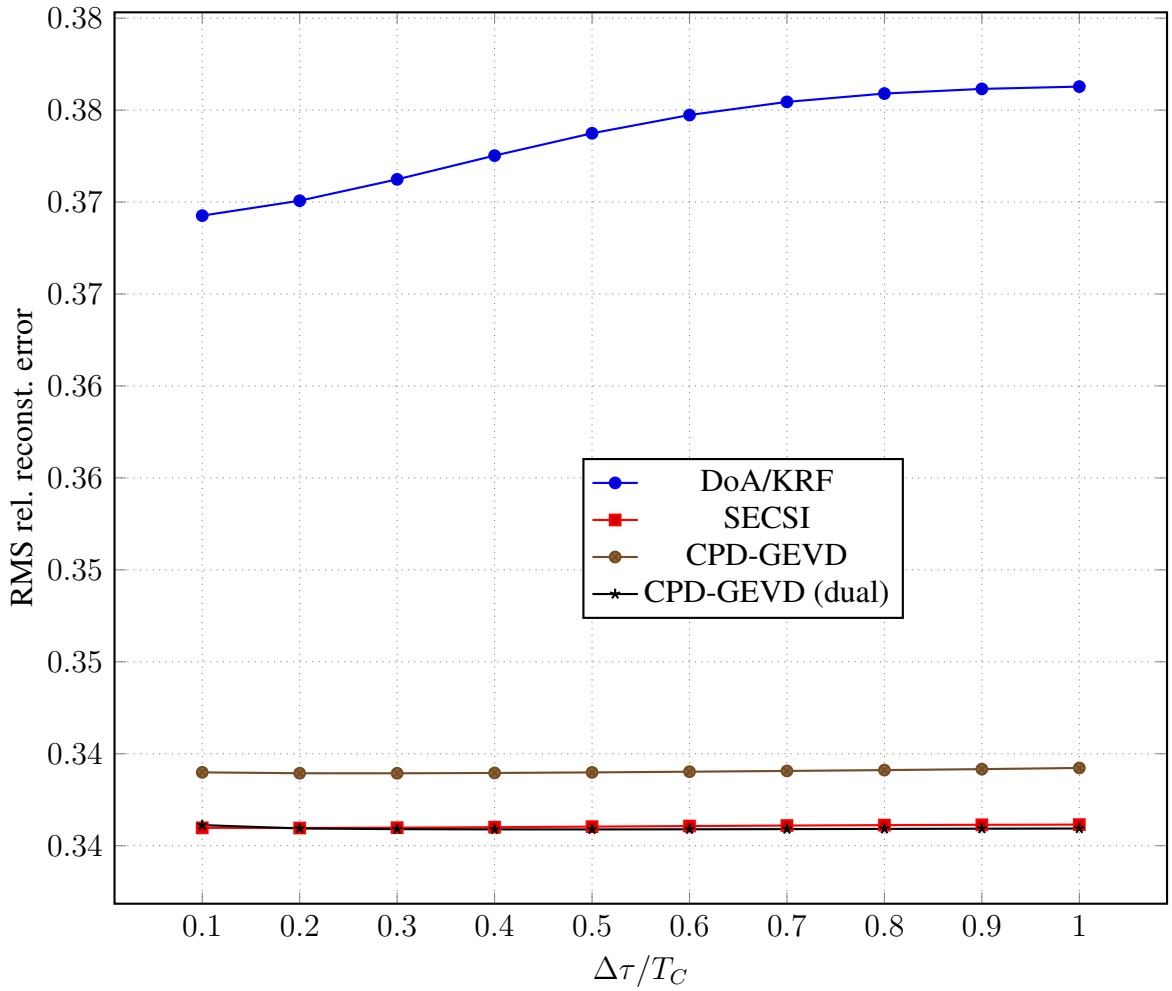

Figure 2.13: Relative reconstruction error for $L = 2$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$, HOE and ProKRaft performance omitted.

Chapter 3

Tensor-Based Time-Delay Estimation for Batch Processing in Clustered Scenarios

When the model order $L > 2$, we have 1 LOS signal component and more than 1 NLOS signal components. Under the assumption these multipath components are clustered [40], have similar timings and are therefore highly correlated, resulting in the pre- and post-correlation PRS matrix, $\mathbf{C}$ and $\tilde{\mathbf{C}}$, being rank-deficient.

In previous publications [28] which followed the data model in [16], the presence of rank deficiencies in the second mode resulted in poor results in the presence of clustered components. This motivated the change in the order in which the factor matrices compose the data tensor.

In this chapter we present a data model which takes into account the rank deficiency of the PRS matrix, compare it to the data model presented previously, and propose a batch tensor-based scheme for clustered scenarios [37]. Finally, numerical simulations are performed to compare performance with the present state of the art.

3.1 Generalized Data Model

For scenarios with more than one multipath component ($L > 2$) present, similar NLOS component timings result in highly correlated columns in the code factor matrix, resulting in a rank-deficient factor matrix.

In this section, a generalized data model for clustered (NLOS) components is presented.

3.1.1 Multilinear Rank-$(L_r, L_r, 1)$ GNSS Data Model for Clustered Multipath Components

Continuing from eq. (2.21), when the NLOS components have similar timings, that is $\tilde{\mathbf{c}}_2 = \tilde{\mathbf{c}}[\tau_{\text{NLOS}}] \approx \tilde{\mathbf{c}}_3 \approx \ldots \approx \tilde{\mathbf{c}}_L$, $\tilde{\mathbf{c}}_\ell$ is approximately equal for $\ell = 2, \ldots, L$,

$$\boldsymbol{\mathcal{Y}}_0 = \mathbf{a}_1 \circ \boldsymbol{\gamma}_1 \circ \tilde{\mathbf{c}}_1 + \sum_{\ell=2}^{L} (\mathbf{a}_\ell \circ \boldsymbol{\gamma}_\ell) \circ \tilde{\mathbf{c}}[\tau_{\text{NLOS}}]. \tag{3.1}$$

This superposition of clustered components impacts time-delay estimation greatly compared to the case where the NLOS components are uncorrelated and can be easily separated.

Continuing from eq. (3.1),

$$\boldsymbol{\mathcal{Y}}_0 = \mathbf{a}_1 \circ \boldsymbol{\gamma}_1 \circ \tilde{\mathbf{c}}_1 + \left(\mathbf{A}_2 \cdot \boldsymbol{\Gamma}_2^{\mathrm{T}}\right) \circ \tilde{\mathbf{c}}_{\text{NLOS}} \tag{3.2}$$

where $\mathbf{A}_2 = \left[\mathbf{a}_2, \ldots, \mathbf{a}_L\right] \in \mathbb{C}^{M \times (L-1)}$ and $\boldsymbol{\Gamma}_2 = \left[\boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_L\right]^{\mathrm{T}} \in \mathbb{C}^{K \times (L-1)}$.

To take into account the rank deficiency of the code factor matrix $\mathbf{C}$, when $L > 2$ a specific type of BTD [38], the decomposition in multilinear rank-$(L_r, L_r, 1)$ terms [73, 74] is proposed for the GNSS data model. The resulting multilinear rank-$(L_r, L_r, 1)$ term representation of eq. (2.21) is

$$\boldsymbol{\mathcal{Y}}_0 = \sum_{r=1}^{2} \left(\mathbf{A}_r \cdot \boldsymbol{\Gamma}_r^{\mathrm{T}}\right) \circ \tilde{\mathbf{c}}_r \in \mathbb{C}^{M \times K \times Q}, \tag{3.3}$$

which is a decomposition in multilinear rank-$(L_r, L_r, 1)$ terms with $R = 2$, and $r = 1$ for the LOS component with $L_1 = 1$ and

$$\mathbf{A}_1 = \mathbf{a}_1 \in \mathbb{C}^M, \tag{3.4}$$
$$\boldsymbol{\Gamma}_1 = \boldsymbol{\gamma}_1 \in \mathbb{C}^K, \tag{3.5}$$
$$\tilde{\mathbf{c}}_1 = \tilde{\mathbf{c}}[\tau_{\text{LOS}}] \in \mathbb{R}^Q, \tag{3.6}$$

and $r = 2$ for the NLOS components with $L_2 = L - 1$, thus

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{a}_2 & \cdots & \mathbf{a}_L \end{bmatrix} \in \mathbb{C}^{M \times (L-1)}, \tag{3.7}$$
$$\boldsymbol{\Gamma}_2 = \begin{bmatrix} \boldsymbol{\gamma}_2 & \cdots & \boldsymbol{\gamma}_L \end{bmatrix} \in \mathbb{C}^{K \times (L-1)}, \tag{3.8}$$
$$\tilde{\mathbf{c}}_2 = \tilde{\mathbf{c}}[\bar{\tau}_{\text{NLOS}}] \in \mathbb{R}^Q. \tag{3.9}$$

3.1.2 PARAFAC and multilinear rank-$(L_r, L_r, 1)$ terms model comparison

The multilinear rank-$(L_r, L_r, 1)$ term representation in eq. (3.3) can be represented as a Tucker-3 model [65]

$$\boldsymbol{\mathcal{Y}}_0 = \boldsymbol{\mathcal{I}}^{(R)} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}}^{(R)} \tag{3.10}$$

in which the core tensor is $\boldsymbol{\mathcal{I}}^{(R)} \in \mathbb{R}^{L \times L \times 2}$, and $\tilde{\mathbf{C}}^{(R)} = \left[\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2\right] \in \mathbb{R}^{Q \times 2}$.

The Tucker-3 model in Equation (3.10) can be expressed as a parallel factors (PARAFAC) model in which a combination matrix $\mathbf{P} \in \mathbb{R}^{2 \times L}$ is applied through the third-mode product to the identity tensor, combining its last $L-1$ third-order slices into the second slice to take into account the contribution of only a single LOS component and a single NLOS component,

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_0 &= (\boldsymbol{\mathcal{I}}_{3,L} \times_3 \mathbf{P}) \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}}^{(R)}, & (3.11)\\ &= \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}}^{(R)} \cdot \mathbf{P}. & (3.12) \end{aligned}$$

The combination matrix $\mathbf{P}$ has structure

$$\mathbf{P} = \begin{bmatrix} 1 & \mathbf{0}_{L-1}^{\mathrm{T}} \\ 0 & \mathbf{1}_{L-1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{2 \times L}, \tag{3.13}$$

and its pseudoinverse

$$\mathbf{P}^{\dagger} = \frac{1}{L-1} \begin{bmatrix} L-1 & 0 \\ \mathbf{0}_{L-1} & \mathbf{1}_{L-1} \end{bmatrix} \in \mathbb{R}^{L \times 2}. \tag{3.14}$$

The connection between the original correlated code factor matrix $\tilde{\mathbf{C}}$ and the 2-component factor matrix $\tilde{\mathbf{C}}^{(R)}$ can be performed by applying $\mathbf{P}$ in the third mode of the identity tensor, and replacing $\tilde{\mathbf{C}}^{(R)} = \tilde{\mathbf{C}}\mathbf{P}^{\dagger}$ in Equation (3.12),

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_0 &= (\boldsymbol{\mathcal{I}}_{3,L} \times_3 \mathbf{P}) \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}} \cdot \mathbf{P}^{\dagger}, & (3.15)\\ &= \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}} \cdot \mathbf{P}^{\dagger} \cdot \mathbf{P}. & (3.16) \end{aligned}$$

Note that the multiplication of $\tilde{\mathbf{C}}$ by $\mathbf{P}^{\dagger}$ does not affect the first column and averages the remaining $L-1$ columns in to a single column, as observed in eqs. (3.6) and (3.9).

The data model in multilinear rank-$(L_r, L_r, 1)$-terms is thus expressed in Equation (3.16) as a PARAFAC model.

While $\mathbf{P}^{\dagger}\mathbf{P}$ does not equal identity unless $L = 2$, its structure is

$$\mathbf{P}^{\dagger} \cdot \mathbf{P} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{L-1} & \cdots & \frac{1}{L-1} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & \frac{1}{L-1} & \cdots & \frac{1}{L-1} \end{bmatrix}. \tag{3.17}$$

By multiplying $\tilde{\mathbf{C}}$ by $\mathbf{P}^{\dagger} \cdot \mathbf{P}$ we have

$$\tilde{\mathbf{C}} \cdot \mathbf{P}^{\dagger} \cdot \mathbf{P} = \begin{bmatrix} \tilde{\mathbf{c}}_1 & \tilde{\mathbf{c}}_2 & \cdots & \tilde{\mathbf{c}}_L \end{bmatrix} \cdot \mathbf{P}^{\dagger} \cdot \mathbf{P}, \tag{3.18}$$

$$= \begin{bmatrix} \tilde{\mathbf{c}}_1 & \tilde{\mathbf{c}}_{\text{avg}} & \cdots & \tilde{\mathbf{c}}_{\text{avg}}, \end{bmatrix} \tag{3.19}$$

where

$$\tilde{\mathbf{c}}_{\text{avg}} = \frac{1}{L-1} \cdot \begin{bmatrix} \tilde{\mathbf{c}}_2 & \cdots & \tilde{\mathbf{c}}_L \end{bmatrix} \mathbf{1}_{L-1} \in \mathbb{R}^{Q} \tag{3.20}$$

being, therefore the average of $\begin{bmatrix} \tilde{\mathbf{c}}_2, \ldots, \tilde{\mathbf{c}}_L \end{bmatrix}$ across the second dimension. For NLOS components with $\tau_2 \approx \ldots \approx \tau_L$, $\tilde{\mathbf{c}}_{\ell} \approx \tilde{\mathbf{c}}_{\text{avg}}$, for $\ell = 2, \ldots, L$.

3.2 Decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via generalized eigenvalue decomposition

The decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via generalized eigenvalue decomposition ($(L_r, L_r, 1)$-GEVD) approach to TDE assumes a data model with clustered components in the code factor matrix. Like the CPD-GEVD approach to TDE described previously in Section 2.3.1, the $(L_r, L_r, 1)$-GEVD approach to TDE begins with a LMLRA of $\boldsymbol{\mathcal{Y}}$,

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}} + \boldsymbol{\mathcal{N}}, \tag{3.21}$$

$$\approx \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{3.22}$$

and the GEVD of the matrix pencil $(\mathbf{S}_1^{\mathrm{T}}, \mathbf{S}_2^{\mathrm{T}})$ to initialize an estimate of transformation matrix $\mathbf{T}_1$,

$$\mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{S}_2^{\mathrm{T}} \cdot \mathbf{R} \cdot \mathbf{D}, \tag{3.23}$$

resulting in $\hat{\mathbf{T}}_1 = \mathbf{R}^{-\mathrm{T}}$ (or equivalently $\hat{\mathbf{T}}_1^{-1} = \mathbf{R}$).

Since estimating $\mathbf{T}_2$ and $\mathbf{T}_3$ via LSKRF is subject to rank deficiencies in the second and third mode [28], the $(L_r, L_r, 1)$-GEVD approach uses a more balanced approach to estimate $\mathbf{T}_2$ and $\mathbf{T}_3$.

To estimate $\mathbf{T}_2$ we utilize the transpose of the first core tensor slice and the previous result of the GEVD, the right generalized eigenvector matrix $\mathbf{R} \in \mathbb{C}^{L \times L}$,

$$\hat{\mathbf{T}}_2 = \mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{R}, \tag{3.24}$$

$$= \mathbf{T}_2 \cdot \mathcal{D}_1 \left\{ \mathbf{T}_3 \right\} \cdot \mathbf{T}_1^{\mathrm{T}} \cdot \hat{\mathbf{T}}_1^{-\mathrm{T}}, \tag{3.25}$$

$$= \mathbf{T}_2 \cdot \mathcal{D}_1 \left\{ \mathbf{T}_3 \right\}, \tag{3.26}$$

is equal to $\mathbf{T}_2$ with trivial scaling ambiguities.

The third-mode transformation matrix $\mathbf{T}_3$ is estimated using the third-mode unfolding of the core tensor $\boldsymbol{\mathcal{S}}$

$$[\boldsymbol{\mathcal{S}}]_{(3)} = \mathbf{T}_3 \left(\mathbf{T}_1 \diamond \mathbf{T}_2 \right)^{\mathrm{T}} \tag{3.27}$$

and the previous estimates of $\mathbf{T}_1$ and $\mathbf{T}_2$

$$\hat{\mathbf{T}}_3 = [\boldsymbol{\mathcal{S}}]_{(3)} \left(\left(\hat{\mathbf{T}}_1 \diamond \hat{\mathbf{T}}_2 \right)^{\mathrm{T}} \right)^{\dagger} \tag{3.28}$$

$$\approx \mathbf{T}_3. \tag{3.29}$$

With the estimates of $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$, we can estimate the factor matrices by expanding the corresponding singular vector matrices

$$\hat{\mathbf{A}} = \mathbf{U}_1 \cdot \hat{\mathbf{T}}_1, \tag{3.30}$$

$$\hat{\mathbf{\Gamma}} = \mathbf{U}_2 \cdot \hat{\mathbf{T}}_2, \tag{3.31}$$

$$\hat{\tilde{\mathbf{C}}} = \mathbf{U}_3 \cdot \hat{\mathbf{T}}_3. \tag{3.32}$$

For clustered NLOS components, we have the relation from eq. (3.12)

$$\tilde{\mathbf{C}} = \tilde{\mathbf{C}}^{(R)} \cdot \mathbf{P}, \tag{3.33}$$

where $\mathbf{P} \in \mathbb{R}^{2 \times L}$ is the combination matrix described in Section 3.1.2.

To estimate $\mathbf{P}$, clustering is applied to eq. (3.32) to select the component of $\hat{\tilde{\mathbf{C}}}$ most likely to be least correlated (and therefore less highly correlated) component of $\tilde{\mathbf{C}}$. We propose a clustering method based on Euclidian distance.

Given an estimated code factor matrix $\hat{\tilde{\mathbf{C}}}$, we calculate the Q distances between each q-th row of $\tilde{\mathbf{C}}$ in a distance matrix $\mathbf{D}_q$ for $q = \{1, \ldots, Q\}$, and square it using the Hadamard product of its conjugate

$$\mathbf{D}_q = \mathbf{1}_L \cdot \hat{\tilde{\mathbf{C}}}(q, \cdot) - \left(\mathbf{1}_L \cdot \hat{\tilde{\mathbf{C}}}(q, \cdot) \right)^{\mathrm{T}} \in \mathbb{C}^{L \times L} \tag{3.34}$$

$$\mathbf{D}_q^2 = \mathbf{D}_q \odot \mathbf{D}_q^* \in \mathbb{R}^{L \times L}. \tag{3.35}$$

We then sum all Q row distances

$$\mathbf{D}^2 = \sum_{q=1}^{Q} \mathbf{D}_q^2 \in \mathbb{R}^{L \times L}, \tag{3.36}$$

then take the average square distance across its rows

$$\mathbf{d}_{\text{avg}}^2 = \frac{1}{L} \mathbf{D}^2 \cdot \mathbf{1}_L \in \mathbb{R}^L. \tag{3.37}$$

Finally, the least correlated component is selected based on the average squared Euclidian distance

$$\ell = \arg\max_{\ell} \mathbf{d}_{\text{avg}}^2(\ell) \tag{3.38}$$

The proposed clustering method and combination matrix reconstruction is summarized in Algorithm 2 is used to select the LOS component and estimate $\mathbf{P}$, after the columns of $\hat{\tilde{\mathbf{C}}}$ are normalized.

Estimating the combination matrix $\mathbf{P}$ implicitly selects the LOS component since estimating $\tilde{\mathbf{C}}^{(R)}$

$$\hat{\tilde{\mathbf{C}}}^{(R)} = \hat{\tilde{\mathbf{C}}} \cdot \hat{\mathbf{P}}^{\dagger} \in \mathbb{C}^{Q \times 2}, \tag{3.39}$$

compresses $\hat{\tilde{\mathbf{C}}}$ and places the column most likely to be the LOS component in the first column of $\hat{\tilde{\mathbf{C}}}^{(R)}$.

The auto-correlation vector can then be estimated

$$\hat{\mathbf{q}} = \left| \mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}^{(R)}(\cdot, 1) \right| \in \mathbb{R}^Q, \tag{3.40}$$

followed by cubic spline interpolation for time-delay estimation.

The $(L_r, L_r, 1)$-GEVD approach is illustrated as a block diagram in Figure 3.1.

Algorithm 2 Proposed single outlier selection clustering

Input: $\hat{\tilde{\mathbf{C}}} \in \mathbb{C}^{Q \times L}$

Output: Estimated combination matrix $\hat{\mathbf{P}}$

1: Calculate absolute squared distances

2: **for** $q = 1$ to Q **do**

$$\mathbf{D}_q = \mathbf{1}_L \cdot \hat{\tilde{\mathbf{C}}}(q, \cdot) - \left(\mathbf{1}_L \cdot \hat{\tilde{\mathbf{C}}}(q, \cdot)\right)^{\mathrm{T}} \in \mathbb{C}^{L \times L}$$
$$\mathbf{D}_q^2 = \mathbf{D}_q \odot \mathbf{D}_q^* \in \mathbb{R}^{L \times L}$$

3: **end for**

4: Sum squared distances

$$\mathbf{D}^2 = \sum_{q=1}^{Q} \mathbf{D}_q^2 \in \mathbb{R}^{L \times L}$$

5: Calculate average squared distance

$$\mathbf{d}_{\text{avg}}^2 = \frac{1}{L} \mathbf{D}^2 \cdot \mathbf{1}_L \in \mathbb{R}^L$$

6: Select greatest average squared distance

$$\ell = \arg\max_{\ell} \mathbf{d}_{\text{avg}}^2(\ell)$$

7: Construct $\hat{\mathbf{P}}$

$$\hat{\mathbf{P}} = \begin{bmatrix} \mathbf{0}_{\ell-1}^{\mathrm{T}} & 1 & \mathbf{0}_{L-\ell+1}^{\mathrm{T}} \\ \mathbf{1}_{\ell-1}^{\mathrm{T}} & 0 & \mathbf{1}_{L-\ell+1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{2 \times L}$$

Algorithm 3 Proposed decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via GEVD TDE approach

Input: tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{M \times K \times Q}$, multilinear ranks $L_{\text{LOS}} = 1$ and $L_{\text{NLOS}} = L - 1$.
Output: estimated time-delay τ_{LOS}, (optionally) factor matrices

1: Estimate LMLRA of $\boldsymbol{\mathcal{Y}}$ using truncated MLSVD

$$\boldsymbol{\mathcal{Y}} \approx \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \in \mathbb{C}^{I \times J \times K}.$$

2: Calculate GEVD of matrix pencil $(\mathbf{S}_1^{\mathrm{T}}, \mathbf{S}_2^{\mathrm{T}})$

$$\mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{R} = \mathbf{S}_2^{\mathrm{T}} \cdot \mathbf{R} \cdot \mathbf{D}.$$

3: Estimate $\mathbf{T}_1$ and $\mathbf{T}_2$

$$\hat{\mathbf{T}}_1 = \mathbf{R}^{-\mathrm{T}}, \qquad \hat{\mathbf{T}}_2 = \mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{R}.$$

4: Estimate $\mathbf{T}_3$ and $\tilde{\mathbf{C}}$

$$\hat{\mathbf{T}}_3 = [\boldsymbol{\mathcal{S}}]_{(3)} \left(\left(\hat{\mathbf{T}}_1 \diamond \hat{\mathbf{T}}_2 \right)^{\mathrm{T}} \right)^{\dagger},$$
$$\hat{\tilde{\mathbf{C}}} = \mathbf{U}_3 \cdot \hat{\mathbf{T}}_3 = \hat{\tilde{\mathbf{C}}}^{(R)} \cdot \mathbf{P}.$$

5: Normalize the columns of $\hat{\tilde{\mathbf{C}}}$

$$\hat{\tilde{\mathbf{C}}}_{\text{norm.}} = \hat{\tilde{\mathbf{C}}} \left\{ \mathbf{1}_R^{\mathrm{T}} \left(\hat{\tilde{\mathbf{C}}} \odot \hat{\tilde{\mathbf{C}}}^{*} \right) \right\}^{-\frac{1}{2}}.$$

6: Use single outlier detection clustering to classify $\hat{\tilde{\mathbf{C}}}_{\text{norm.}}$ into 2 clusters and estimate combination matrix $\mathbf{P}$ (see algorithm 2)

7: Calculate $\hat{\tilde{\mathbf{C}}}^{(R)}$

$$\hat{\tilde{\mathbf{C}}}^{(R)} = \hat{\tilde{\mathbf{C}}} \cdot \hat{\mathbf{P}}^{\dagger}.$$

8: Calculate auto-correlation vector

$$\hat{\mathbf{q}} = \left| \mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}^{(R)}(\cdot, 1) \right| \in \mathbb{R}^{Q}.$$

9: Apply cubic spline interpolation to the auto-correlation vector $\mathbf{q}$ to estimate time-delay [16].

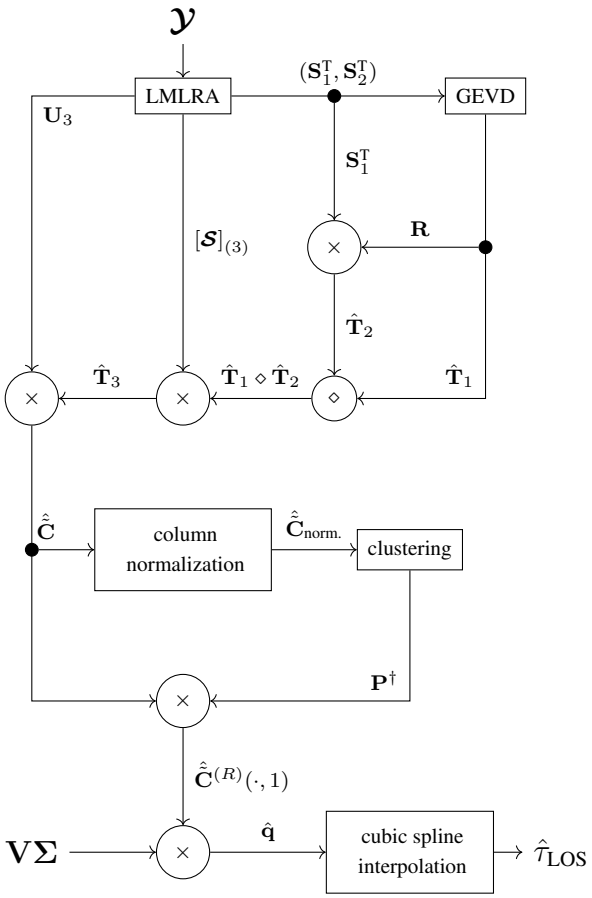

Figure 3.1: Proposed decomposition in multilinear rank-$(L_r, L_r, 1)$ terms via GEVD TDE approach block diagram

3.3 Dual eigenvector matrix variant CPD-GEVD and $(L_r, L_r, 1)$-GEVD

While the approaches described in Section 3.2 only utilize the right eigenvector matrix of the GEVD, it is possible to use both left and right eigenvector matrices. This allows estimating (the inverse of) both transform matrices in a single (GEVD) operation and this more balanced approach allows more accurately estimating the (third mode) code factor matrix.

Following the LMLRA of $\boldsymbol{\mathcal{Y}}$

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{3.41}$$

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{T}_1 \times_2 \mathbf{T}_2 \times_3 \mathbf{T}_3, \tag{3.42}$$

we calculate the GEVD of the matrix pencil of $(\mathbf{S}_1, \mathbf{S}_2)$.

Using the left eigenvector matrix $\mathbf{L}$,

$$\mathbf{L}^{\mathrm{H}} \cdot \mathbf{S}_1 = \mathbf{D} \cdot \mathbf{L}^{\mathrm{H}} \cdot \mathbf{S}_2, \tag{3.43}$$

$$\mathbf{S}_1 \cdot \mathbf{S}_2^{-1} = \mathbf{L}^{-\mathrm{H}} \cdot \mathbf{D} \cdot \mathbf{L}^{\mathrm{H}}, \tag{3.44}$$

$$= \mathbf{T}_1 \cdot \mathcal{D}_1\left\{\mathbf{T}_3\right\} \cdot \mathcal{D}_2\left\{\mathbf{T}_3\right\}^{-1} \cdot \mathbf{T}_1^{-1}, \tag{3.45}$$

and therefore we have an estimate of $\mathbf{T}_1$

$$\hat{\mathbf{T}}_1 = \mathbf{L}^{-\mathrm{H}}. \tag{3.46}$$

Using the right eigenvector matrix $\mathbf{R}$

$$\mathbf{S}_1 \cdot \mathbf{R} = \mathbf{S}_2 \cdot \mathbf{R} \cdot \mathbf{D}, \tag{3.47}$$

$$\mathbf{S}_2^{-1} \cdot \mathbf{S}_1 = \mathbf{R} \cdot \mathbf{D} \cdot \mathbf{R}^{-1}, \tag{3.48}$$

$$\mathbf{S}_1^{\mathrm{T}} \cdot \mathbf{S}_2^{-\mathrm{T}} = \mathbf{R}^{-\mathrm{T}} \cdot \mathbf{D} \cdot \mathbf{R}^{\mathrm{T}}, \tag{3.49}$$

$$= \mathbf{T}_2 \cdot \mathcal{D}_1\left\{\mathbf{T}_3\right\} \cdot \mathcal{D}_2\left\{\mathbf{T}_3\right\}^{-1} \cdot \mathbf{T}_2^{-1}, \tag{3.50}$$

and therefore we have an estimate of $\mathbf{T}_2$

$$\hat{\mathbf{T}}_2 = \mathbf{R}^{-\mathrm{T}}. \tag{3.51}$$

The transform matrix $\mathbf{T}_3$ is estimated the same as in $(L_r, L_r, 1)$-GEVD using eq. (3.28), and the factor matrices estimated by expansion using eqs. (3.30) to (3.32). This is a variant of CPD-GEVD.

The variant of $(L_r, L_r, 1)$-GEVD proceeds from here like the proposed $(L_r, L_r, 1)$-GEVD by applying column normalization to $\hat{\bar{\mathbf{C}}}$ to estimate $\mathbf{P}$, and consequently select the LOS component and $\hat{\bar{\mathbf{C}}}^{(R)}$.

3.4 Computational Cost

For both the state-of-the-art CPD-GEVD and the proposed $(L_r, L_r, 1)$-GEVD approaches the primary defining factor in the computational cost is the model order, L. For a typical simulation scenario with $M = 8$, $K = 30$, and $Q = 11$ [16, 24, 28, 32, 35, 36], the computational cost of the CPD-GEVD approach,in floating point operations (FLOPS), is $2L^3 + 2978L^2 + 6720L$, and the computational cost of the proposed $(L_r, L_r, 1)$-GEVD approach is $2L^4+4L^3+2714L^2+23L$. For these simulation parameters, the proposed approach has a lower cost than the state-of-the art CPD-GEVD up to $L = 17$.

To compare the computational cost we have created three heat maps in which one of the parameters M, K, or Q is fixed, while the other two vary, which shows the $L_{\max}$, given

(M, K, Q), for which the cost of the proposed $(L_r, L_r, 1)$-GEVD approach is greater than that of the CPD-GEVD approach [28].

In Figures 3.2 to 3.4, the value of M, K, and Q is fixed at $M = 8$, $K = 30$, and $Q = 11$, respectively, while the remaining parameters vary between 2 and 30. In all three figures we have marked the point corresponding to $M = 8$, $K = 30$, $Q = 11$ with $\circ$, which are the values of M, K, and Q used in the numerical simulations in Section 2.4.

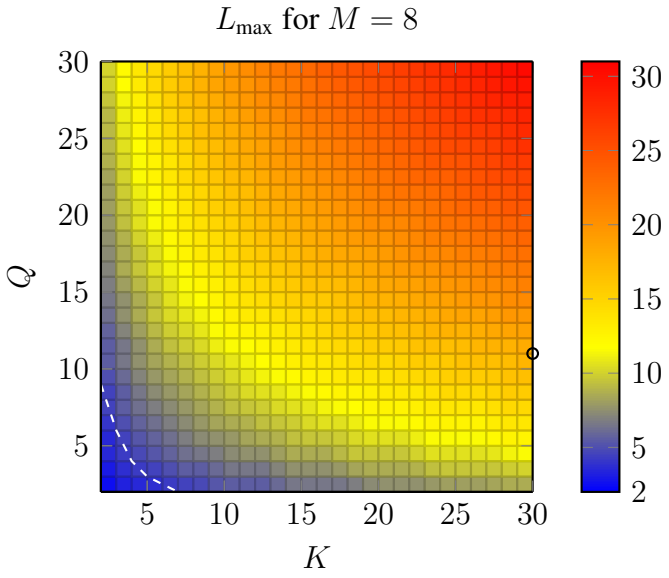

Figure 3.2: Values of L for which the computational complexity of the proposed $(L_r, L_r, 1)$-GEVD approach to TDE is greater than that of the CPD-GEVD approach. Number of array sensors $M = 8$, epochs $K = 2, \ldots, 30$, and correlator bank taps $Q = 2, \ldots, 30$.

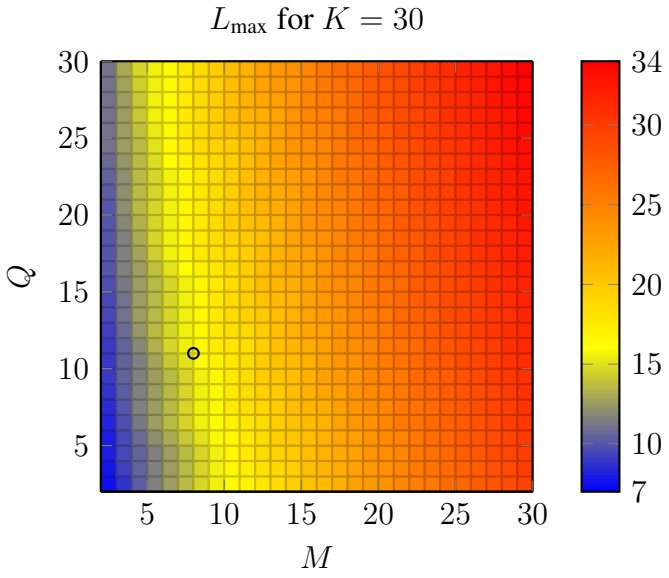

Figure 3.3: Values of L for which the computational complexity of the proposed $(L_r, L_r, 1)$-GEVD approach to TDE is greater than that of the CPD-GEVD approach. Number of array sensors $M = 2, \ldots, 30$, epochs $K = 30$, and correlator bank taps $Q = 2, \ldots, 30$.

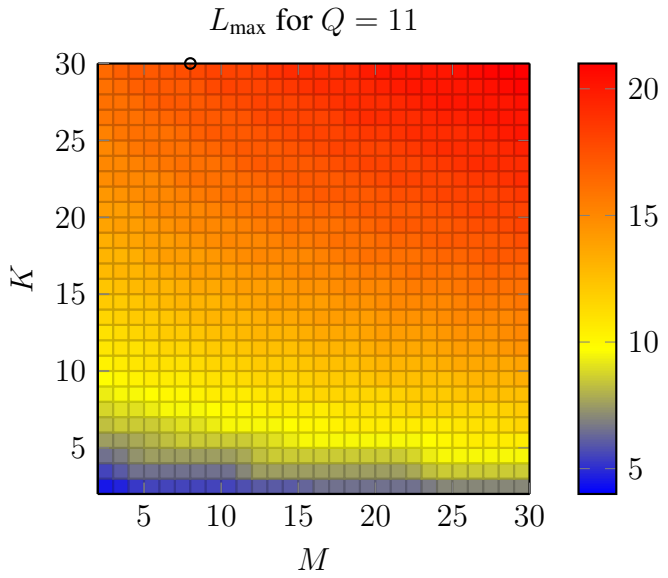

Figure 3.4: Values of L for which the computational complexity of the proposed $(L_r, L_r, 1)$-GEVD approach to TDE is greater than that of the CPD-GEVD approach. Number of array sensors $M = 2, \ldots, 30$, epochs $K = 1, \ldots, 30$, and correlator bank taps $Q = 11$.

In Figure 3.2, both K and Q contribute evenly to $L_{\max}$. For small K and Q, $L_{\max}$ is small. The dashed line delineates the values for which $L_{\max} \leq 4$.

In Figures 3.3 and 3.4, Q and K, respectively, contribute predominantly to $L_{\max}$. In both Figures $L_{\max} \leq 4$.

Details on computational cost of the operations used above are found in Appendix B.

3.5 Numerical Simulation Results

In this section numerical results are presented comparing results with the state-of-the-art and previously presented techniques. The generation of the code factor matrix in the clustered scenario is detailed in Section 3.5.1.

3.5.1 Code (PRS) matrix generation

For $L > 2$, there's more than a single time-delay. For simulation purposes, the average NLOS delay $\bar{\tau}_{\text{NLOS}}$ is selected with a delay relative to τ_{LOS} so that

$$\bar{\tau}_{\text{NLOS}} = \tau_{\text{LOS}} + \Delta\tau. \tag{3.52}$$

For an average NLOS time-delay $\bar{\tau}_{\text{NLOS}}$, each ℓ-th delay can be calculated as

$$\tau_\ell = \bar{\tau}_{\text{NLOS}} + \left(\ell - 1 - \frac{L}{2}\right)\Delta T,\ \ell = \{2, \ldots, L\}, \tag{3.53}$$

resulting in

$$L = 3,\ \tau_2 = \bar{\tau}_{\text{NLOS}} - \frac{1}{2}\Delta T,\ \tau_3 = \bar{\tau}_{\text{NLOS}} + \frac{1}{2}\Delta T, \tag{3.54}$$

$$L = 4,\ \tau_2 = \bar{\tau}_{\text{NLOS}} - \Delta T,\ \tau_3 = \bar{\tau}_{\text{NLOS}},$$
$$\tau_4 = \bar{\tau}_{\text{NLOS}} + \Delta T, \tag{3.55}$$

$$L = 5,\ \tau_2 = \bar{\tau}_{\text{NLOS}} - \frac{3}{2}\Delta T,\ \tau_3 = \bar{\tau}_{\text{NLOS}} - \frac{1}{2}\Delta T,$$
$$\tau_4 = \bar{\tau}_{\text{NLOS}} + \frac{1}{2}\Delta T, \tau_5 = \bar{\tau}_{\text{NLOS}} + \frac{3}{2}\Delta T, \tag{3.56}$$

ensuring an *average* time-delay $\bar{\tau}_{\text{NLOS}}$.

For small values of ΔT, NLOS timings are very similar, resulting in similar time-delays for $\mathbf{c}_\ell$, $\ell = \{2, \ldots, L\}$, consequently both $\mathbf{C}$ and $\tilde{\mathbf{C}}$ are rank-deficient.

3.5.2 Simulation scenario

The simulation scenario is the same as described previously in Section 2.4.4, except the NLOS components are temporally separated by $\Delta T = T_C/(f_c/B) \approx 634.75$ ps, or 19.03 cm, such that the average timing difference between LOS and NLOS components is consistent as described above.

3.5.3 Perfectly calibrated array

In this section simulations are performed assuming a perfectly calibrated array. The array response matrix has perfect Vandermonde structure.

3.5.3.1 Time-delay estimation performance

In Figure 3.5 we see the TDE results for all methods in the presence of 1 LOS and 2 NLOS components, $L = 3$.

TDE performance for the HOE+FBA+ESPS, ProKRaft, and SECSI approaches have degraded considerably with the increase in model order. The considerable degradation in performance for the ProKRaft approach is due to one of its core assumptions, that the core covariance matrix being close to identity no longer being correct [32, 75].

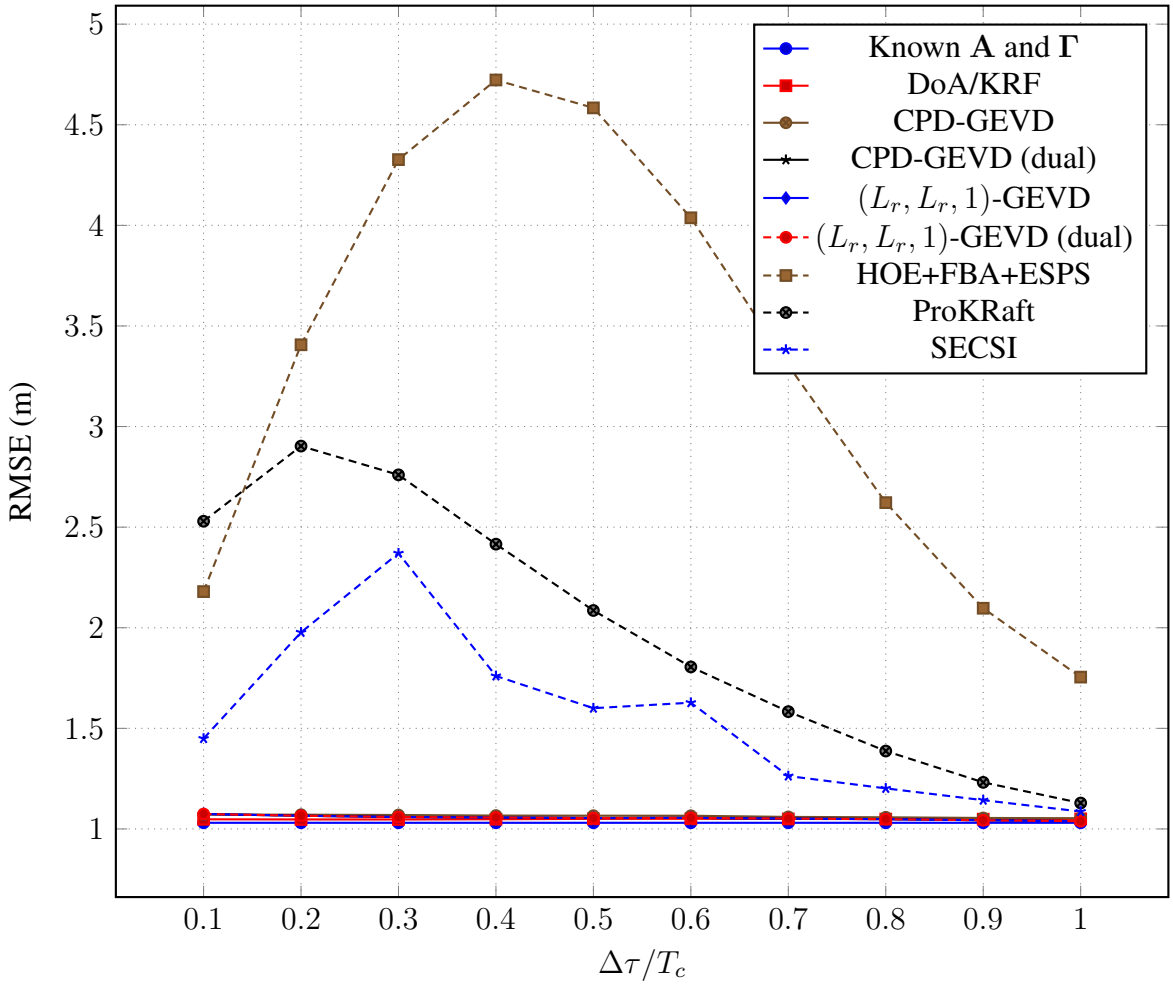

Figure 3.5: Time-delay estimation error in meters for $L = 3$.

In Figure 3.6, the HOE+FBA+ESPS, ProKRaft, and SECSI TDE approaches were omitted to clearly demonstrate the performance differences between the DoA/KRF, CPD-GEVD, and $(L_r, L_r, 1)$-GEVD TDE approaches.

The DoA/KRF approach shows very consistent performance for all $\Delta\tau/T_c$. The performance of the $(L_r, L_r, 1)$-GEVD and both dual eigenvector matrix variants are better than the CPD-GEVD approach and have overlapping performances. This is expected as these three approaches employ essentially equivalent algebraic methods.

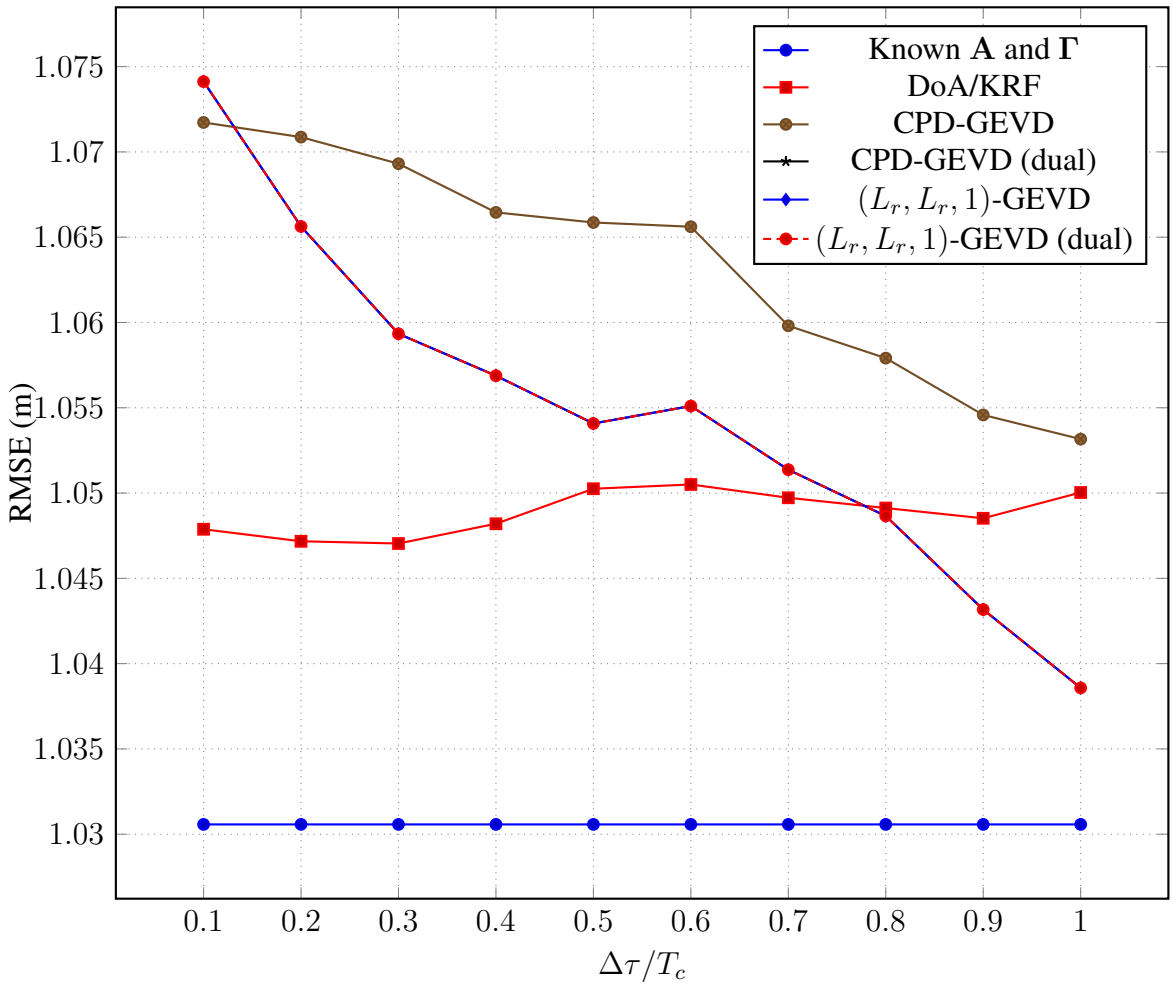

Figure 3.6: Time-delay estimation error in meters for $L = 3$, HOE+FBA+ESPS, ProKRaft, and SECSI omitted.

In Figure 3.7 the model order has been increased to 4, with 1 LOS and 3 NLOS components. TDE performance is shown for all methods.

The ProKRaft approach has failed as expected, and the performance curve for HOE+FBA+ESPS has further increased to nearly 5 m at its peak.

In Figure 3.8, the HOE+FBA+ESPS, ProKRaft, and SECSI TDE approaches were omitted to clearly demonstrate the performance differences between the DoA/KRF, CPD-GEVD, and $(L_r, L_r, 1)$-GEVD TDE approaches. The performance of the $(L_r, L_r, 1)$-GEVD and both dual eigenvector matrix variants have overlapping performance.

Again the DoA/KRF approach has shown consistent performance and the $(L_r, L_r, 1)$-GEVD and both dual eigenvector variants performances overlap and are slightly better than the CPD-GEVD approach.

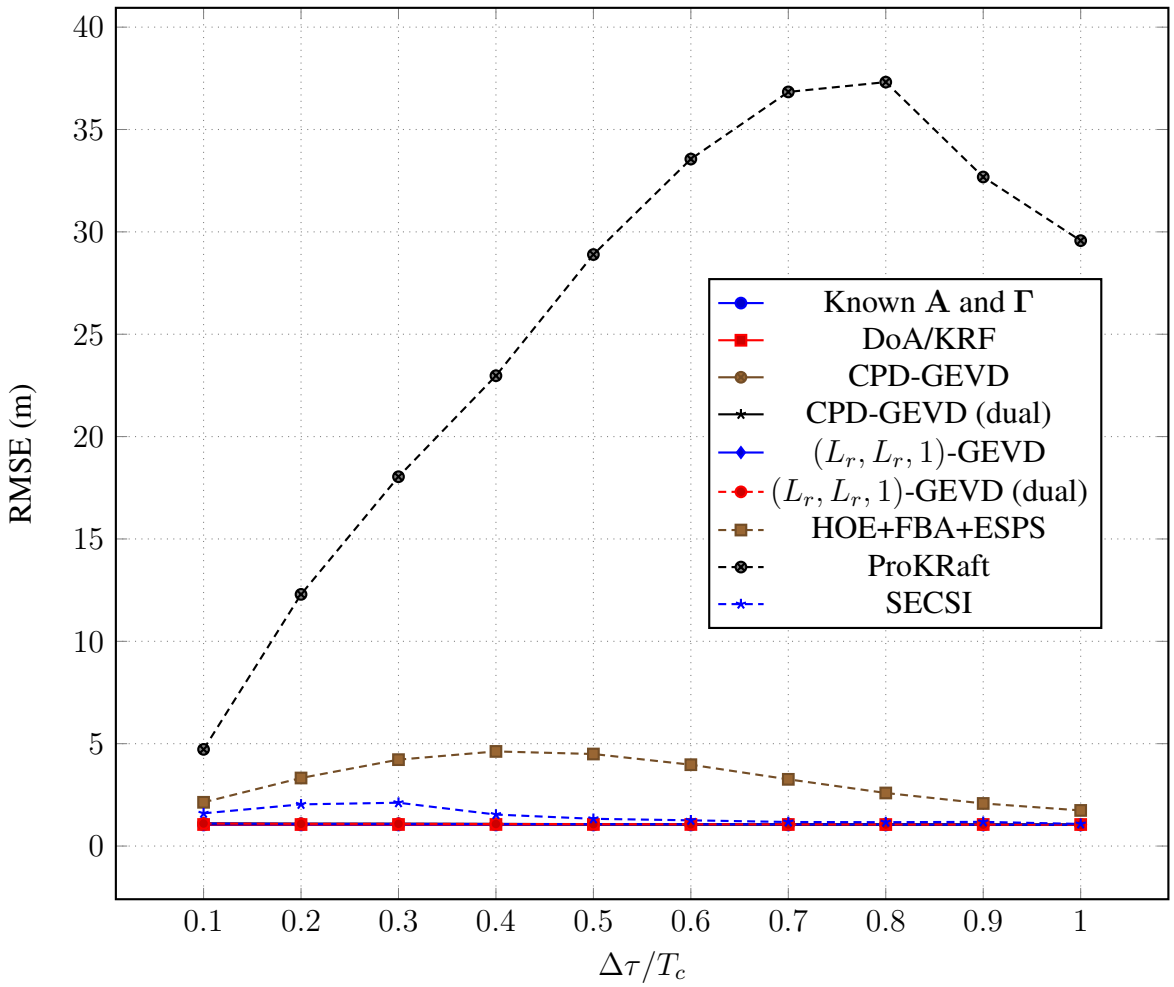

Figure 3.7: Time-delay estimation error in meters for $L = 4$.

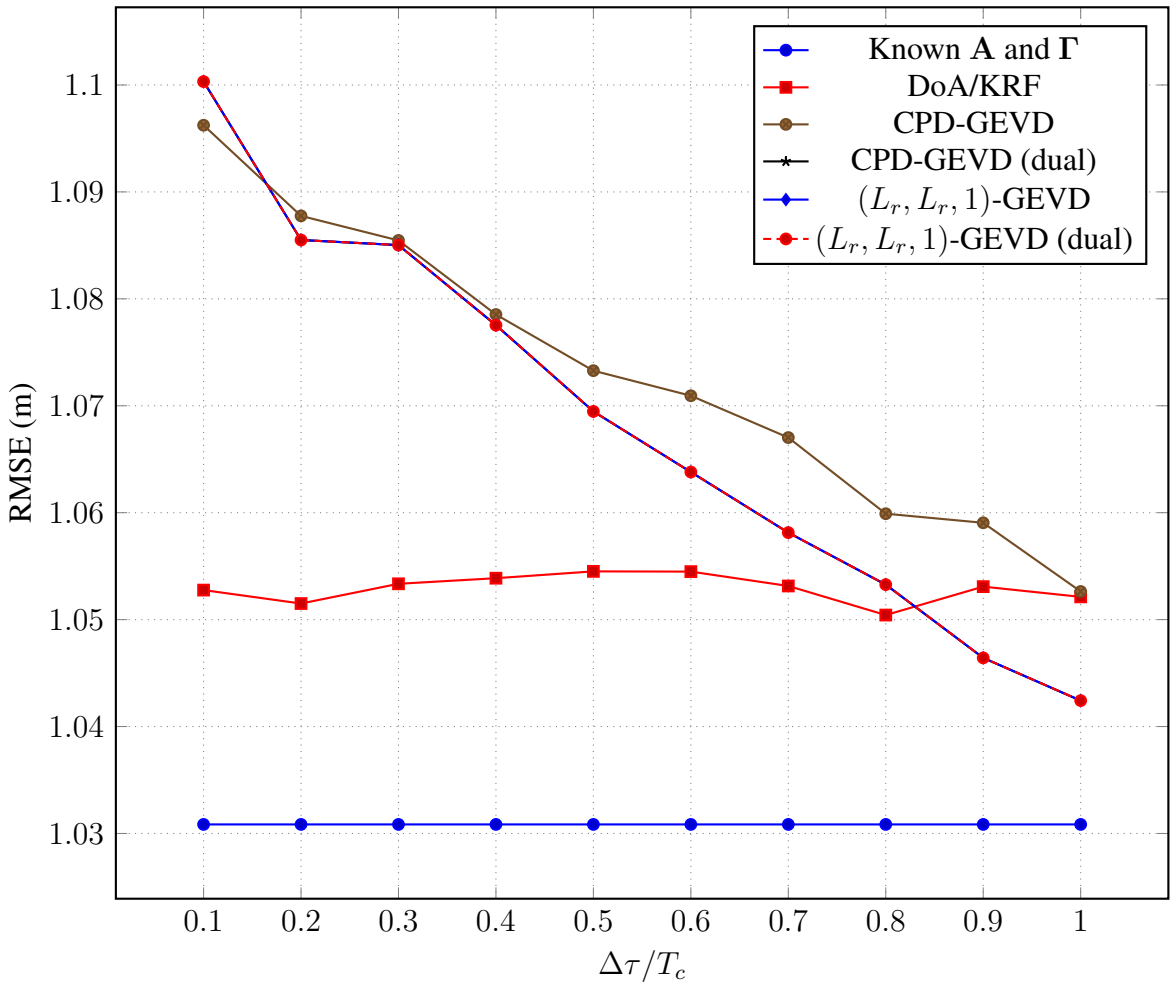

Figure 3.8: Time-delay estimation error in meters for $L = 4$, HOE+FBA+ESPS, ProKRaft, and SECSI omitted.

3.5.3.2 Reconstruction error performance

In Figure 3.9, we see the relative reconstruction error for all methods in the presence of 1 LOS and 2 NLOS components. Construction of $\boldsymbol{\mathcal{Y}}_0$ using eq. (3.16) has been added as a baseline comparison. Its error at iteration i is

$$e_i = \frac{\left\| \boldsymbol{\mathcal{Y}}_{0,i} - \hat{\boldsymbol{\mathcal{Y}}}_{0,i} \right\|_{\mathrm{F}}}{\boldsymbol{\mathcal{Y}}_{0,i}} \tag{3.57}$$

with $\hat{\boldsymbol{\mathcal{Y}}}_{0,i} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A}_i \times_2 \mathbf{\Gamma}_i \times_3 \tilde{\mathbf{C}}_i \cdot \mathbf{P}^{\dagger} \cdot \mathbf{P}$.

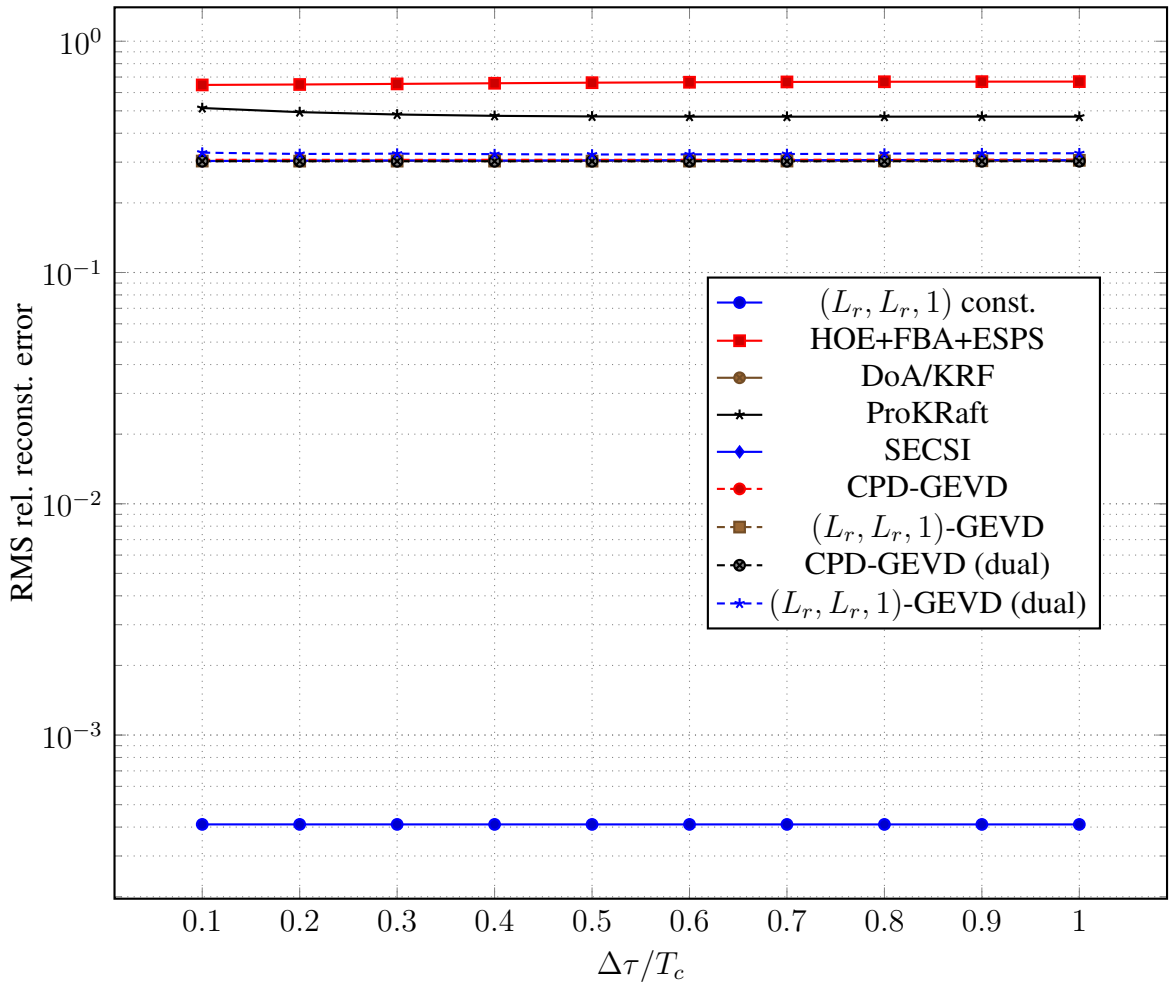

Figure 3.9: Relative reconstruction error for $L = 3$.

The HOE+FBA+ESPS approach has the largest reconstruction error. This is expected as it is a rank-1 reconstruction. The ProKRaft approach has the second largest reconstruction error. This is also expected as with a model order $L = 3$ one of its premises, that the code covariance matrix is similar to an identity matrix, is violated. The baseline has a very low error because it uses a priori information about the tensor.

In Figure 3.10, the $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and $(L_r, L_r, 1)$-GEVD (dual) approaches have been omitted.

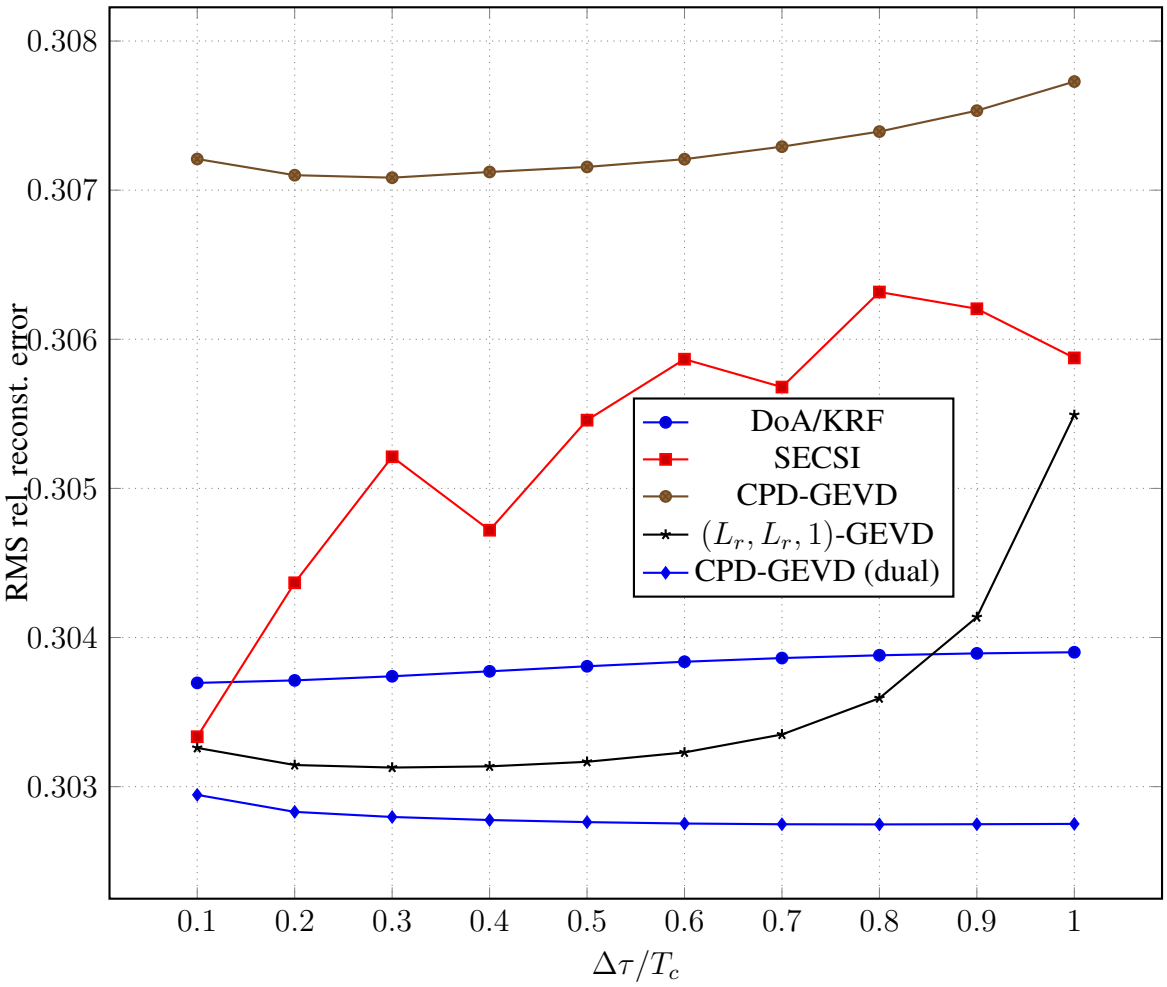

Figure 3.10: Relative reconstruction error for $L = 3$, $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and $(L_r, L_r, 1)$-GEVD (dual) omitted.

Comparing only the approaches above, we see there is little correlation between reconstruction performance and time-delay estimation performance. This is especially notable when comparing the DoA/KRF and CPD-GEVD (dual) approaches, in which the algebraic approach has better reconstruction performance than the parametric approach but the parametric approach has better time-delay performance.

In Figure 3.11, we see the relative reconstruction error for all methods in the presence of 1 LOS and 3 NLOS components.

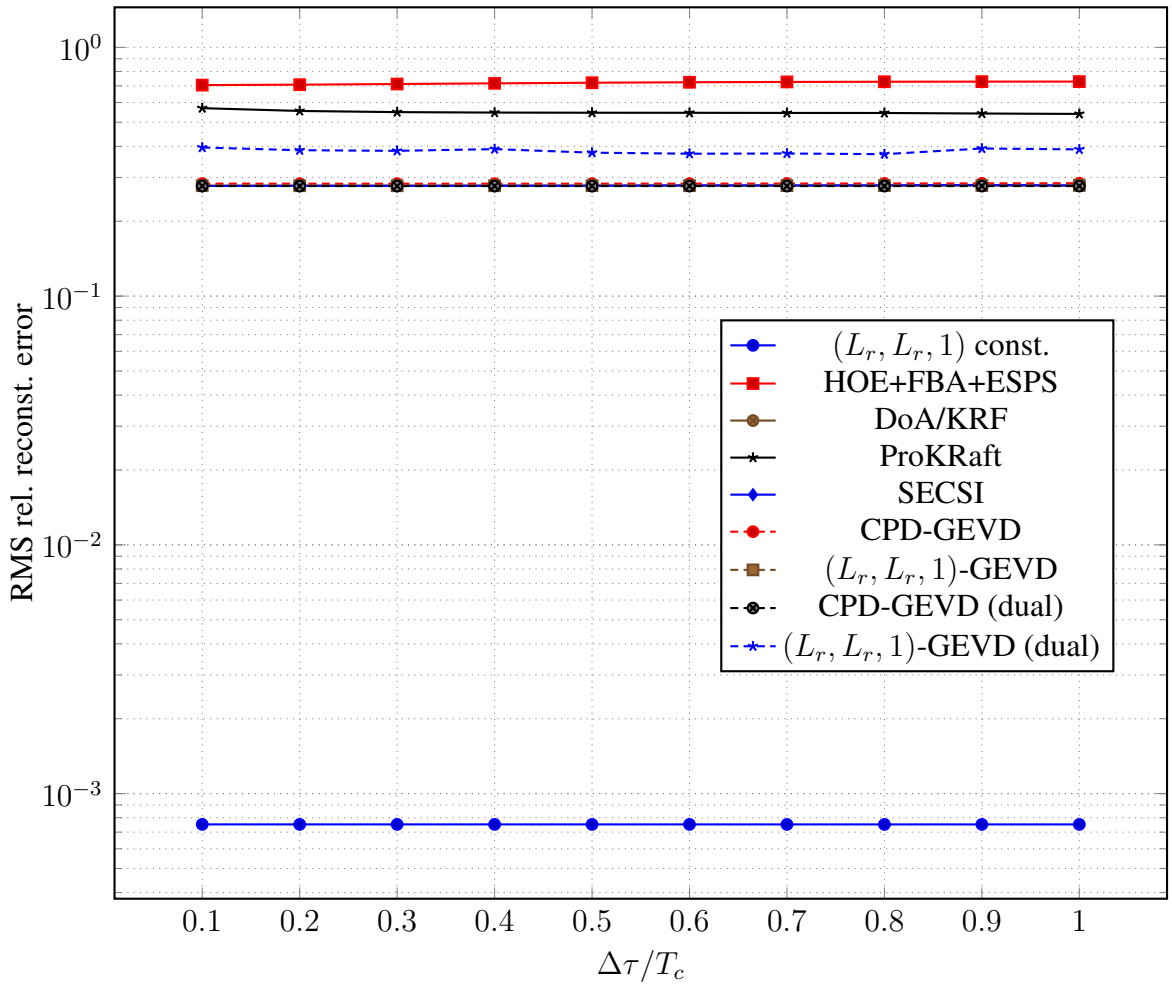

Figure 3.11: Relative reconstruction error for $L = 3$, $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and $(L_r, L_r, 1)$-GEVD (dual) omitted.

Again for the HOE+FBA+ESPS and ProKRaft approaches have the worst reconstruction performance.

In Figure 3.12, the $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and $(L_r, L_r, 1)$-GEVD (dual) approaches have been omitted.

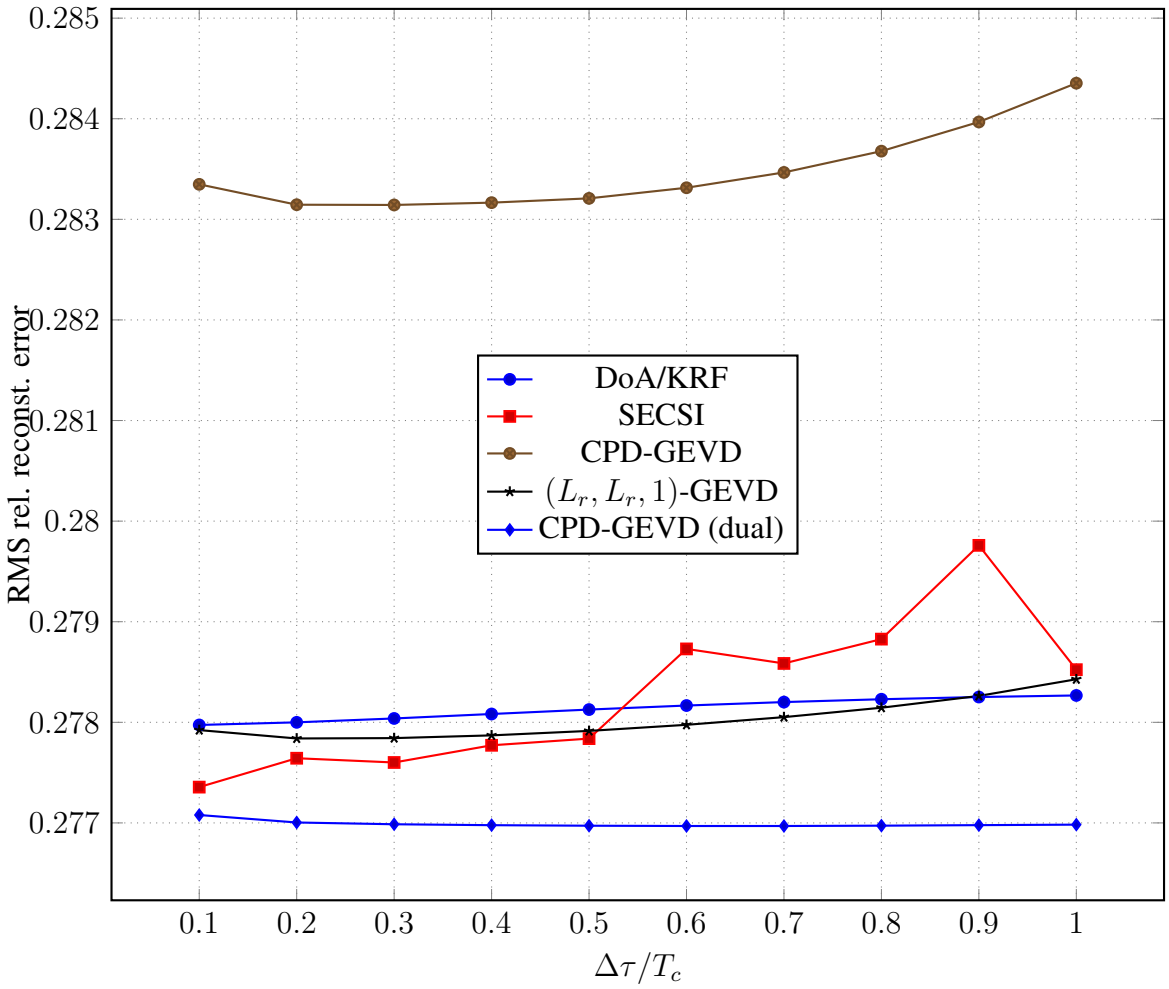

Figure 3.12: Relative reconstruction error for $L = 4$.

The difference in reconstruction performance has become more narrow but there is still little correlation between reconstruction and time-delay estimation performance.

3.5.4 Imperfect Array

In this section, the proposed approaches are tested in the presence of an imperfect array.

3.5.4.1 Time-delay estimation performance

In Figure 3.13 we see the TDE results for all methods in the presence of 1 LOS and 2 NLOS components.

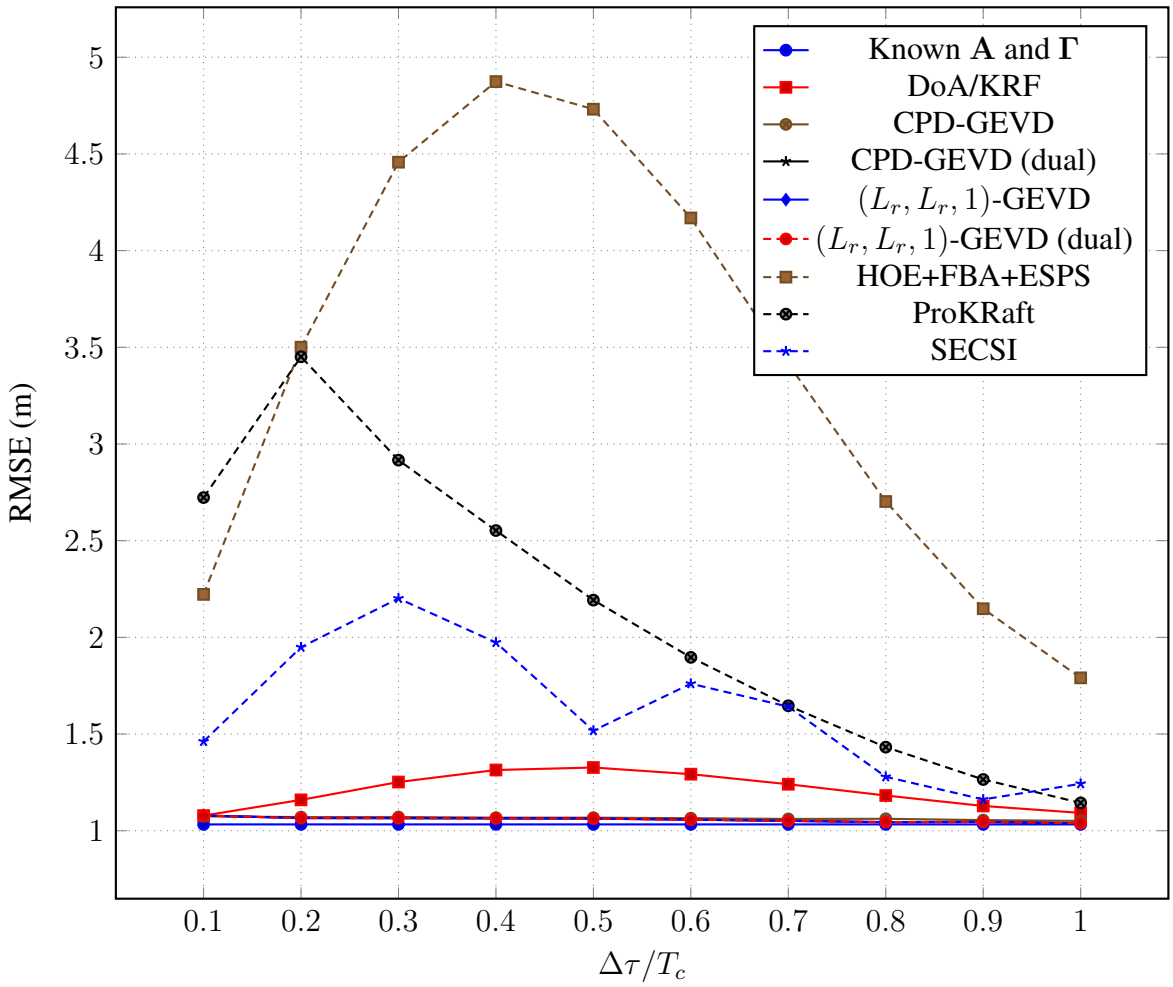

Figure 3.13: Time-delay estimation error in meters for $L = 3$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$.

The HOE+FBA+ESPS, ProKRaft, and SECSI approaches have slightly worse performance when compared to a perfect array. The other algebraic approaches and the DoA/KRF approach have better performance.

In Figure 3.14 these approaches have been omitted for clarity.

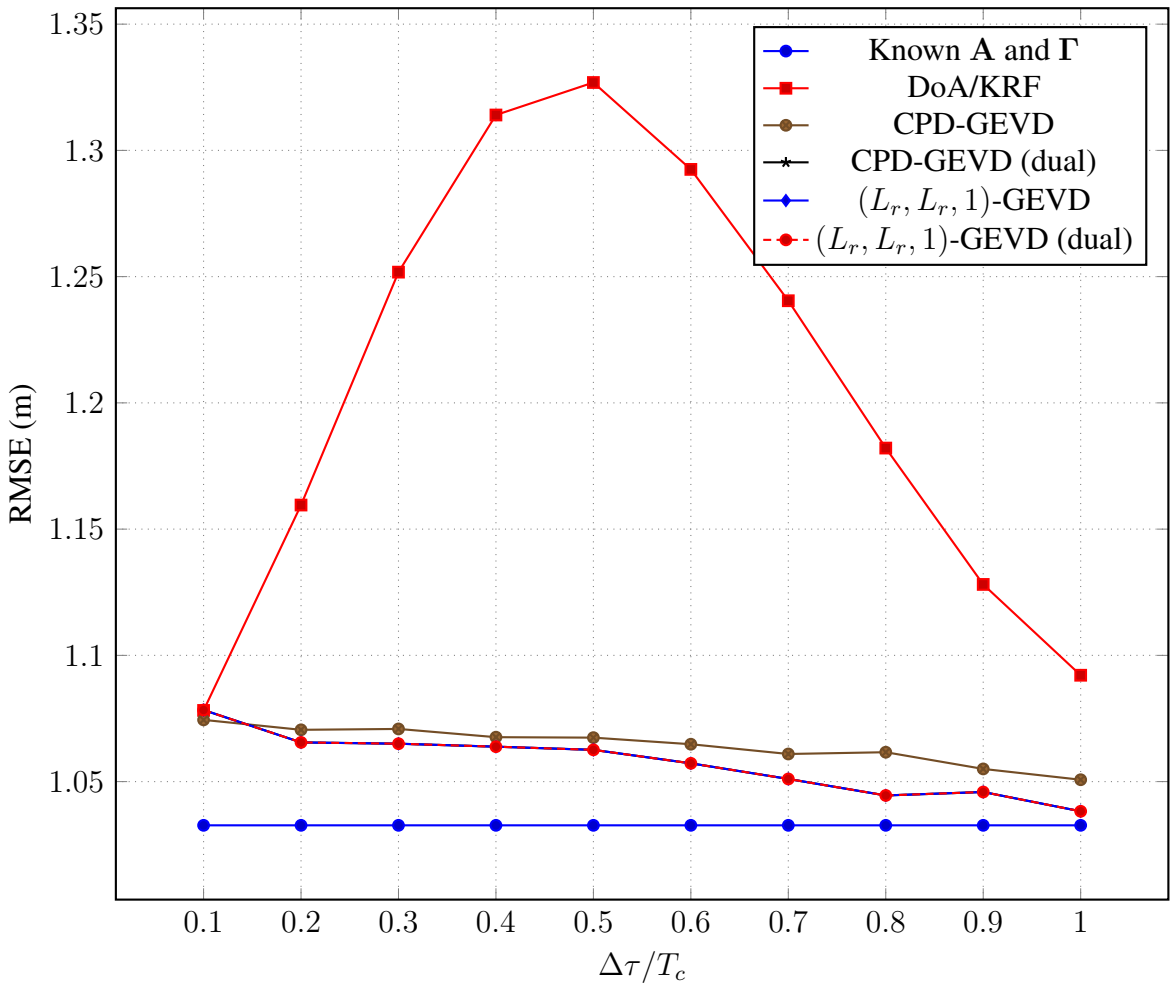

Figure 3.14: Time-delay estimation error in meters for $L = 3$ with an imperfect array with $p = \Pr(e > \lambda/2) = 10^{-30}$. HOE+FBA+ESPS, ProKRaft, and SECSI omitted.

It is clear that the presence of an imperfect array has not only worsened performance for the DoA/KRF approach, it now has a bell shape. The other algebraic approaches have performance consistent with the prior scenario.

In Figure 3.15 we see the TDE results for all methods in the presence of 1 LOS and 3 NLOS components, $L = 4$.

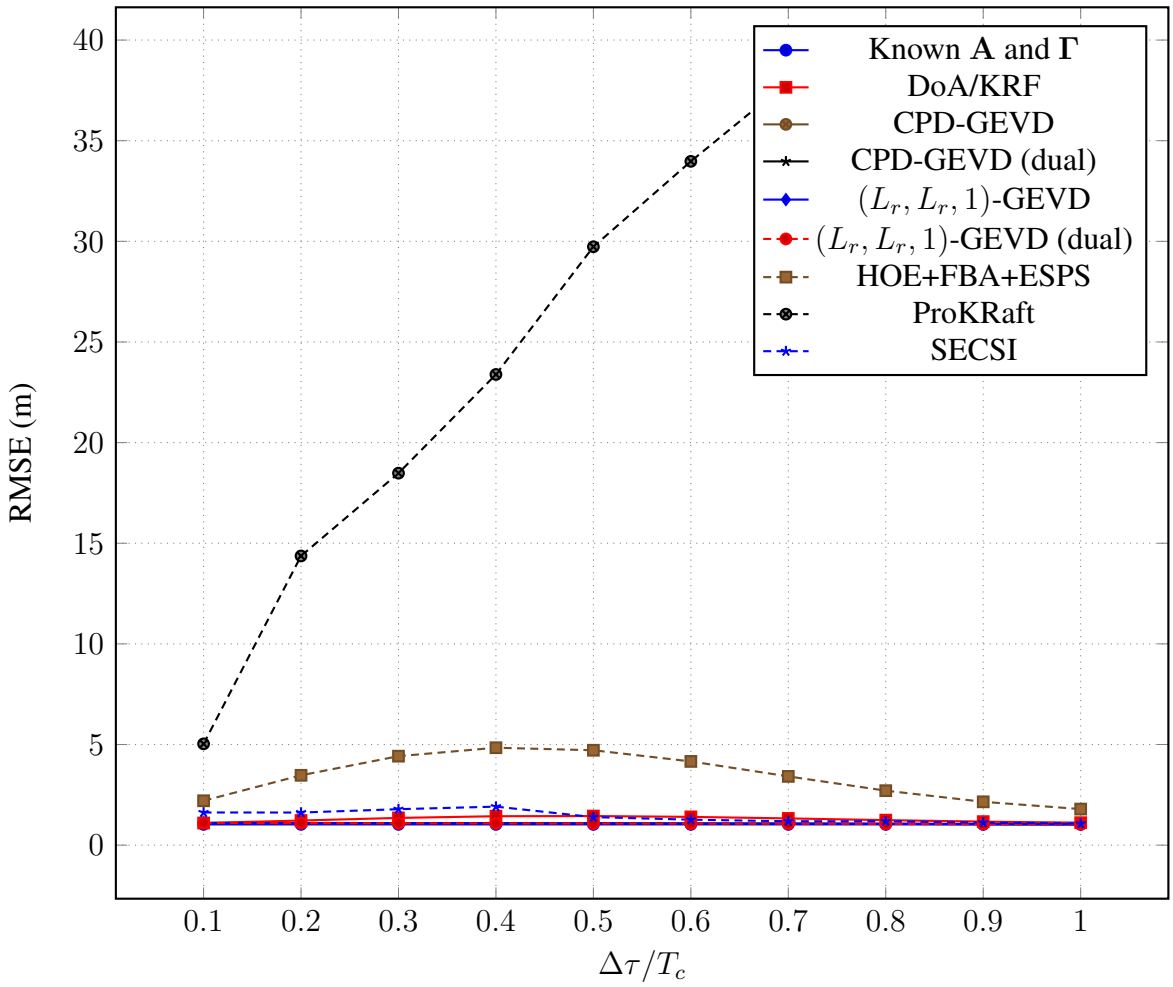

Figure 3.15: Time-delay estimation error in meters for $L = 4$.

The ProKRaft approach has failed and presents performance worst than all other approaches.

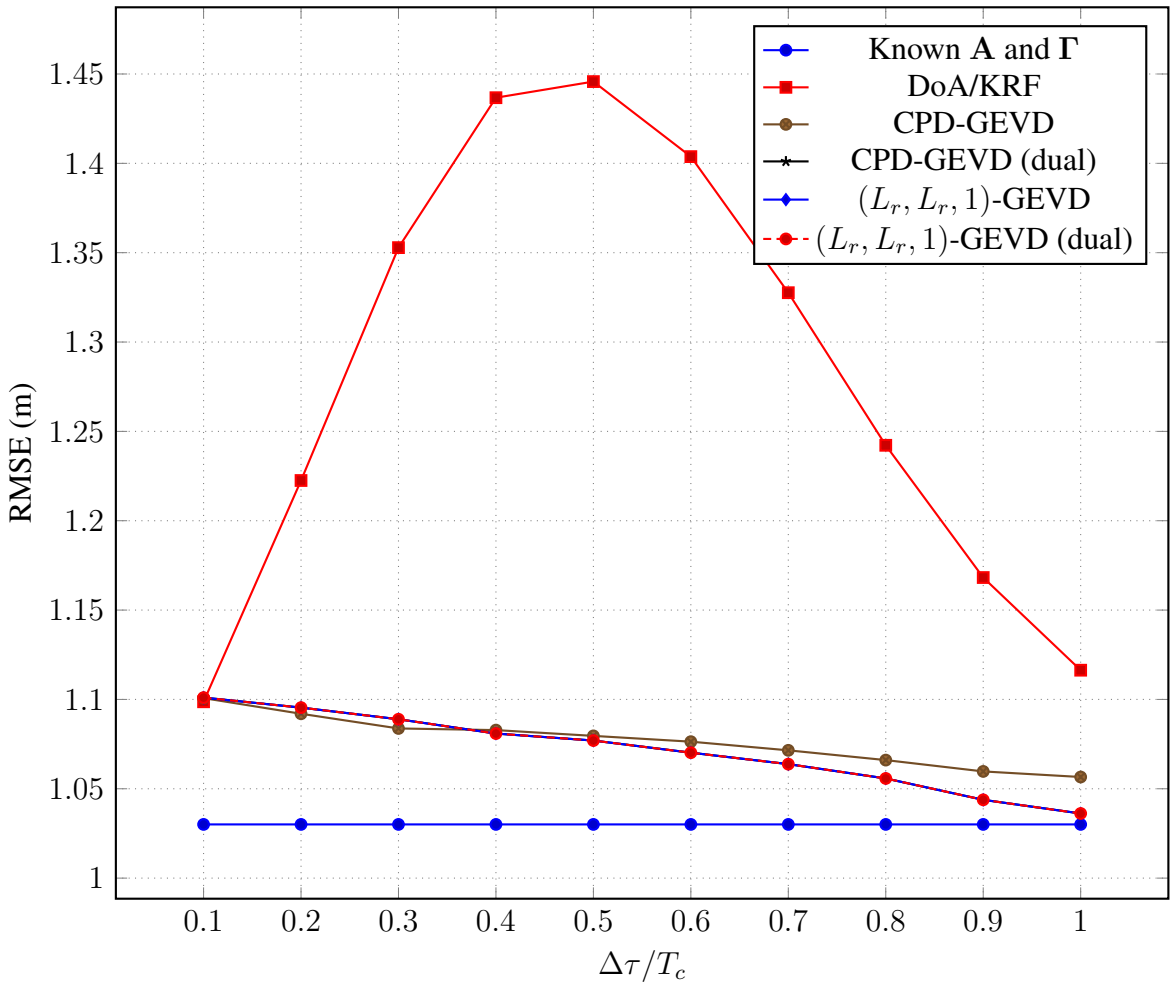

Figure 3.16: Time-delay estimation error in meters for $L = 4$, HOE+FBA+ESPS, ProKRaft, and SECSI omitted.

Again the time-delay estimation performance for the parametric approach has become worse, though it consistently retains a bell-like curve. The other algebraic approaches have maintained a more performance under this scenario.

3.5.4.2 Reconstruction error performance

In Figure 3.9, reconstruction error performance is shown for all approaches, for $L = 3$.

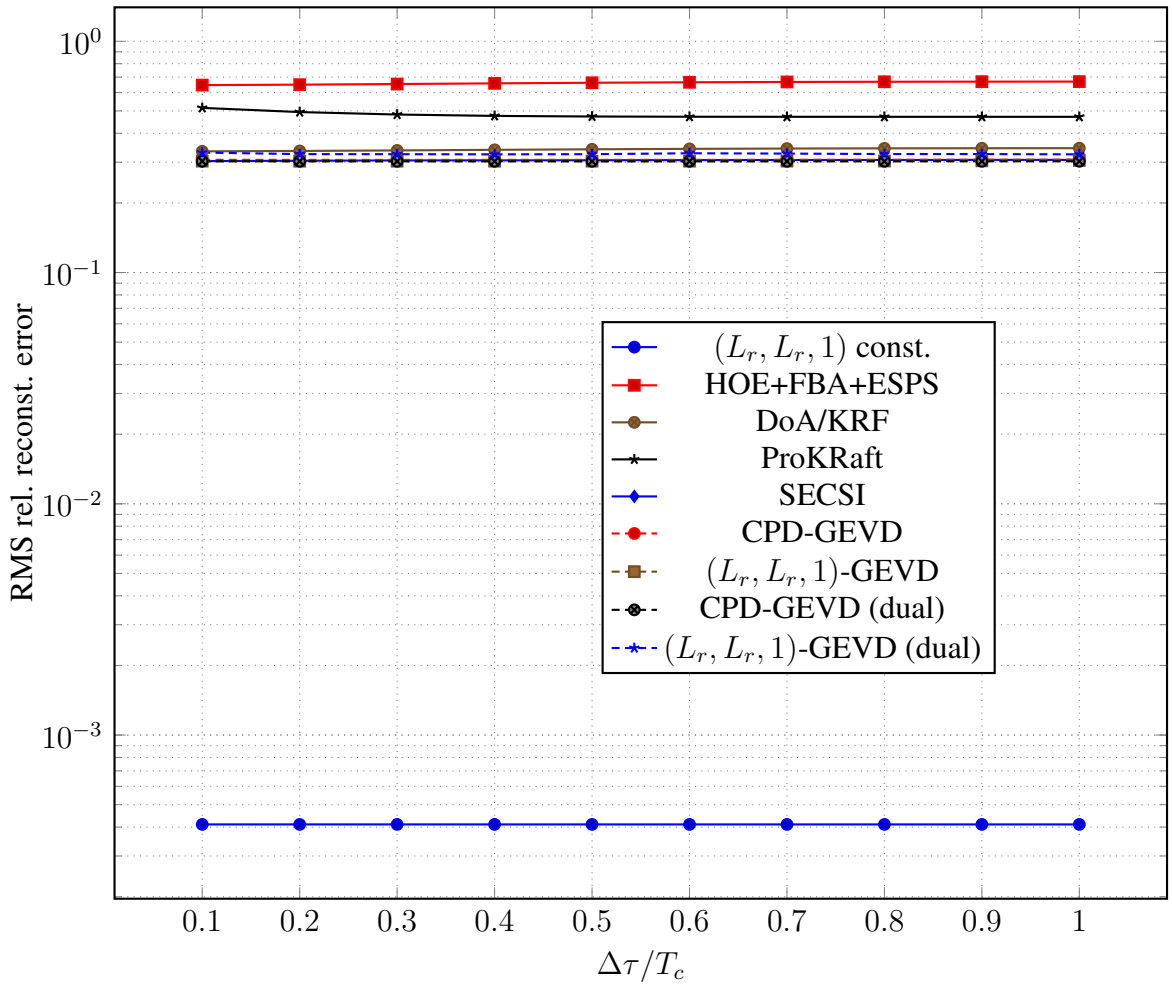

Figure 3.17: Relative reconstruction error for $L = 3$.

The HOE+FBA+ESPS and ProKRaft approaches still retain the worst reconstruction performances.

In Figure 3.10, the $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and SECSI approaches have been omitted.

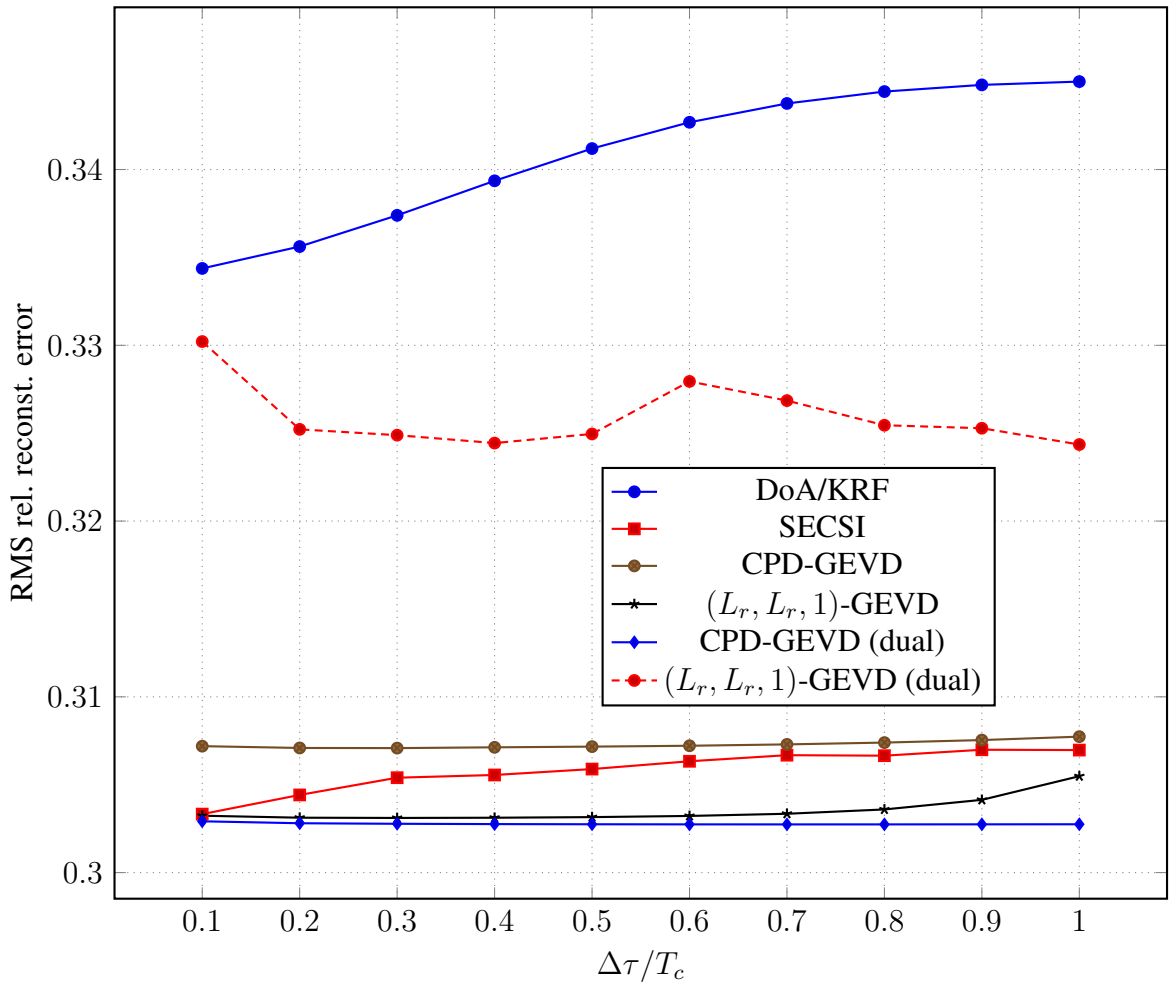

Figure 3.18: Relative reconstruction error for $L = 3$, $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, and ProKRaft omitted.

The lack of correlation between reconstruction and time-delay estimation performance still stands. The reconstruction error has also increased for the $(L_r, L_r, 1)$-GEVD (dual) approach but this did not impact time-delay estimation performance. Notably, the DoA/KRF has the worst performance but its curve is not bell-shaped.

In Figure 3.11, reconstruction error performance is shown for all approaches, for $L = 4$.

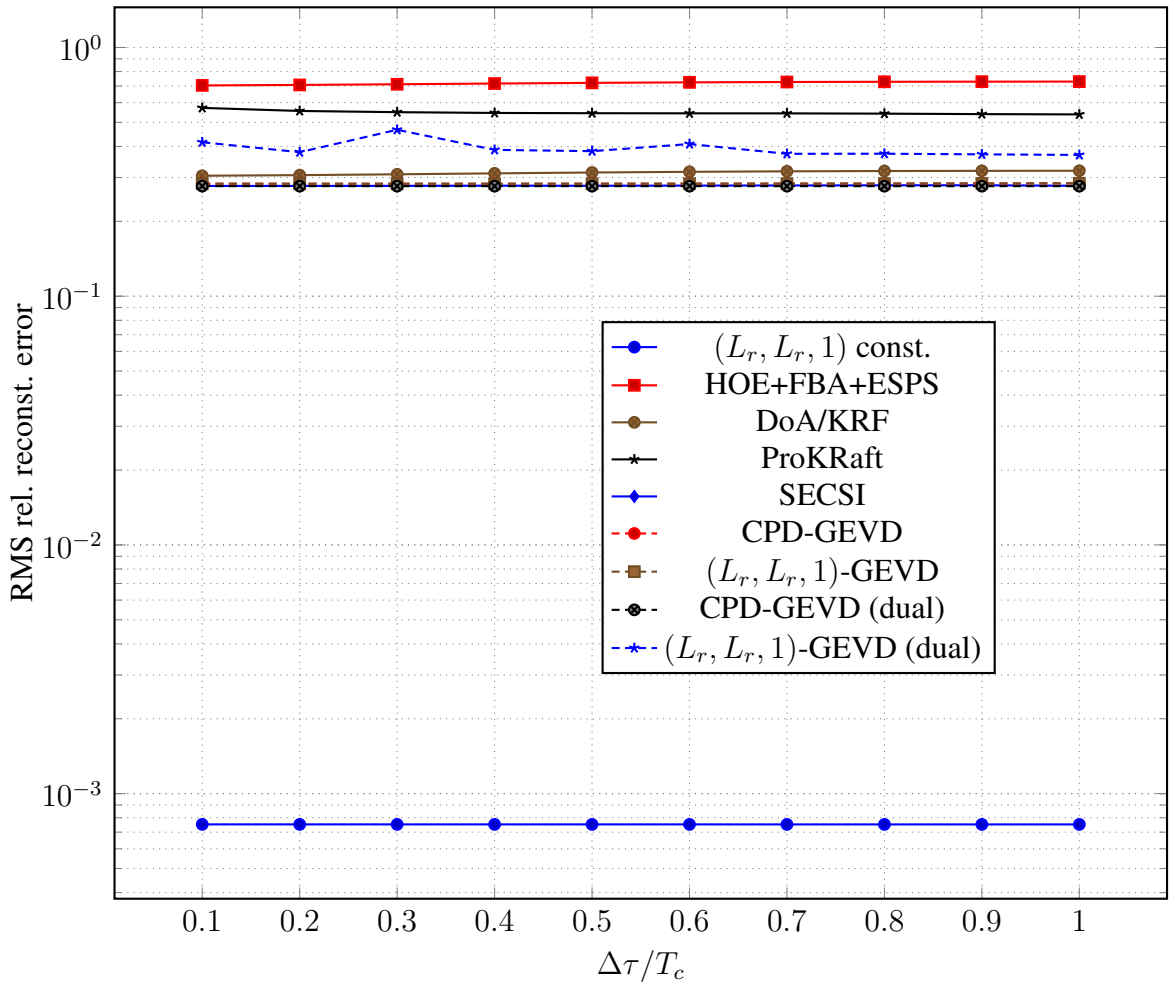

Figure 3.19: Relative reconstruction error for $L = 4$.

Again reconstruction performance for the $(L_r, L_r, 1)$-GEVD (dual) approach has been impacted without affecting time-delay estimation performance.

In Figure 3.12, the $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, ProKRaft, and SECSI approaches have been omitted.

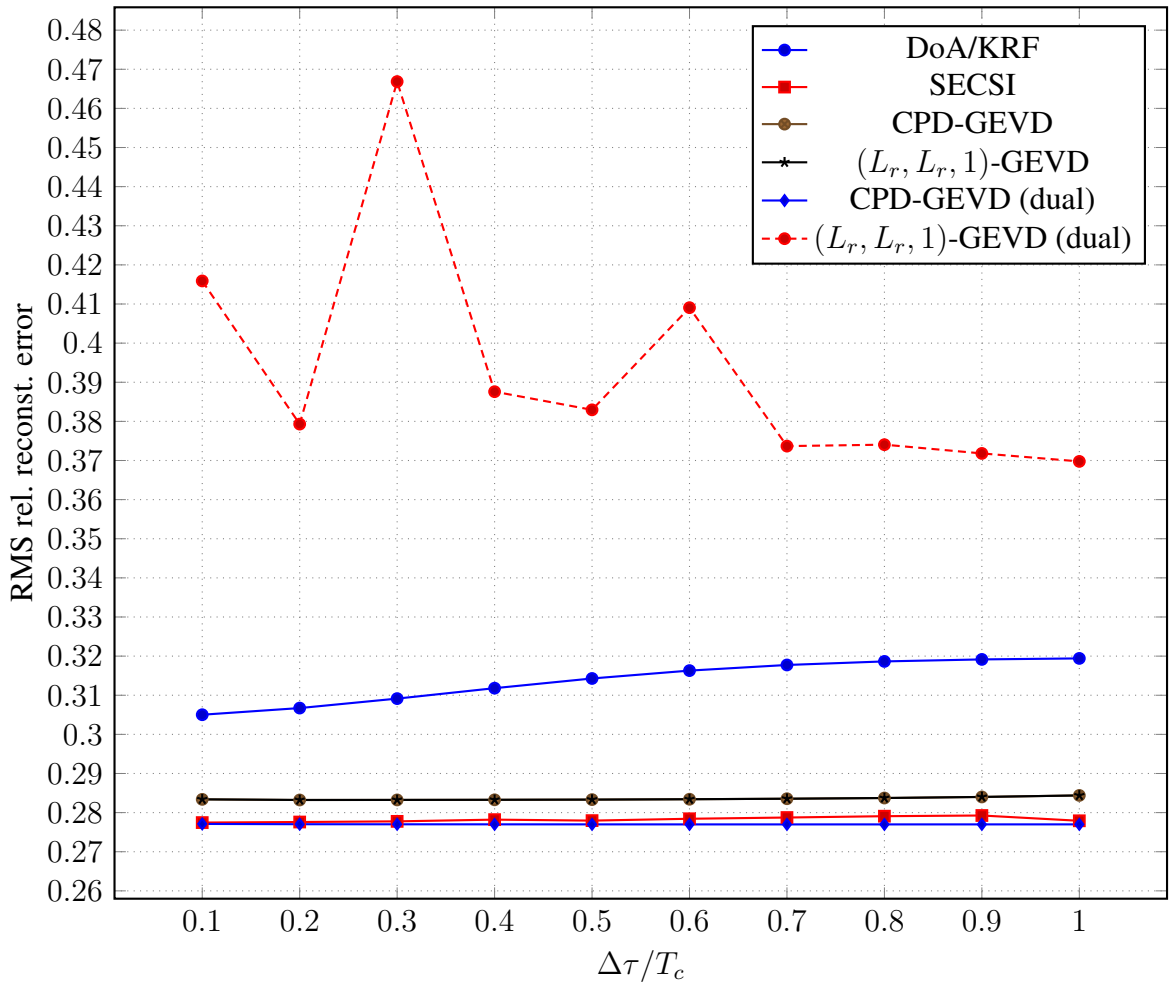

Figure 3.20: Relative reconstruction error for $L = 4$, $(L_r, L_r, 1)$ construction, HOE+FBA+ESPS, and ProKRaft omitted.

Reconstruction error for the $(L_r, L_r, 1)$-GEVD (dual) approach has become even greater than for DoA/KRF, yet is is not impacted as significantly.

Chapter 4

Tensor-Based Time-Delay Estimation with Adaptive Processing

In this chapter, adaptive processing is considered. The data model is reinterpreted looking at a scenario in which the factor matrices can change from an epoch to another and processing occurs per-epoch instead of all at once (batch).

While it is convenient to process a large amount of data in batch mode, it is computationally demanding and is poorly-adapted to scenarios in which data is continuously received, so-called on-line applications. A well-known solution to processing data in an on-line application is to adaptively estimate the subspace(s) of the incoming data.

A class of RLS filters solve this problem by iteratively minimizing the error between the input vector $\mathbf{x}[k]$, at instant k, and the filter's output for an estimated subspace $\hat{\mathbf{U}}[k]$. Figure 4.1 shows the general RLS filter updating estimates of the signal subspace $\hat{\mathbf{U}}[k]$ given an input vector $\mathbf{x}[n]$.

One of the most popular algorithms for subspace estimation is projection approximation subspace tracking (PAST) [76], which treats estimation of the signal subspace as the solution to an unconstrained minimization problem in which the the incoming data and its projection [77] are (assumed to be) approximately equal. This results in a low-complexity ($\mathcal{O}\big(nr\big)$ for an input vector of length n and model order r) algorithm with tracking capability comparable to batch eigenvalue decomposition [78], or iterative singular value decomposition (iSVD), of complexity $\mathcal{O}\big(n^3\big)$.

PAST has been employed for 1-D harmonic retrieval [77], and has also been extended to the tensor case [79, 80] for R-D harmonic retrieval [81]. Other subspace tracking (ST) algorithms include fast approximate power iteration (FAPI) [82] and generalized yet another subspace tracker (GYAST) [83, 84], both have complexity $\mathcal{O}\big(nr\big)$. The pseudocode for all four algorithms, iSVD, PAST, FAPI, and GYAST are found in Appendix C.

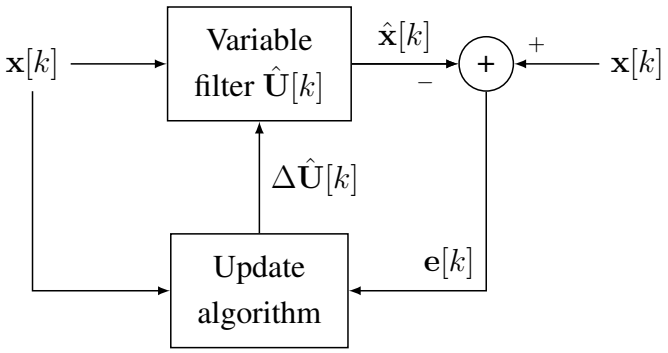

Figure 4.1: General RLS filter with input $\mathbf{x}[k]$.

4.1 Time-Varying Data Model

The previously presented post-correlation data model is

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A} \times_2 \boldsymbol{\Gamma} \times_3 \tilde{\mathbf{C}} + \tilde{\boldsymbol{\mathcal{N}}}, \tag{4.1}$$

with fixed parameters DoA (azimuth) $\boldsymbol{\phi} \in \mathbb{R}^L$ for the array response matrix $\mathbf{A} \in \mathbb{C}^{M \times L}$, absolute magnitude $|\boldsymbol{\gamma}| \in \mathbb{R}^L$ for complex amplitude matrix $\boldsymbol{\Gamma} \in \mathbb{C}^{K \times L}$, and time-delay $\boldsymbol{\tau} \in \mathbb{R}^L$ for PRS matrix $\mathbf{C} \in \mathbb{C}^{N \times L}$ (and $\tilde{\mathbf{C}} \in \mathbb{C}^{Q \times L}$). This assumes a scenario in which the factor matrices do not change, or changes only slightly, during the sampling period.

Reinterpreting this data model as a second mode concatenation [85] of several temporal slices across all K epochs,

$$\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{\mathcal{Y}}[1] \sqcup_2 \boldsymbol{\mathcal{Y}}[2] \sqcup_2 \ldots \sqcup_2 \boldsymbol{\mathcal{Y}}[k] \sqcup_2 \ldots \sqcup_2 \boldsymbol{\mathcal{Y}}[K-1] \sqcup_2 \boldsymbol{\mathcal{Y}}[K]\right], \tag{4.2}$$

where $\boldsymbol{\mathcal{Y}}[k]$ is

$$\boldsymbol{\mathcal{Y}}[k] = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A}[k] \times_2 \boldsymbol{\Gamma}(k,\cdot) \times_3 \tilde{\mathbf{C}}[k] + \tilde{\boldsymbol{\mathcal{N}}}(\cdot,k,\cdot) \in \mathbb{C}^{M \times 1 \times Q}. \tag{4.3}$$

Considering a change in the array response matrix or the PRS matrix at the k-th period and using the more compact notation $\boldsymbol{\Gamma}(k,\cdot) = \boldsymbol{\Gamma}[k] \in \mathbb{C}^{1 \times L}$ and $\tilde{\boldsymbol{\mathcal{N}}}(\cdot,k,\cdot) = \tilde{\boldsymbol{\mathcal{N}}}[k] \in \mathbb{C}^{M \times 1 \times Q}$, the above becomes

$$\boldsymbol{\mathcal{Y}}[k] = \boldsymbol{\mathcal{I}}_{3,L} \times_1 \mathbf{A}[k] \times_2 \boldsymbol{\Gamma}[k] \times_3 \tilde{\mathbf{C}}[k] + \tilde{\boldsymbol{\mathcal{N}}}[k], \tag{4.4}$$

and the parameters representing all L signal components at all K possible epochs are the

matrices

$$\boldsymbol{\Phi} = \Big[\boldsymbol{\phi}[1] \quad \cdots \quad \boldsymbol{\phi}[K]\Big]^{\mathrm{T}} \in \mathbb{R}^{K \times L}, \tag{4.5}$$

$$|\boldsymbol{\Gamma}| = \Big[|\boldsymbol{\gamma}[1]| \quad \cdots \quad |\boldsymbol{\gamma}[K]|\Big]^{\mathrm{T}} \in \mathbb{R}^{K \times L}, \tag{4.6}$$

$$\boldsymbol{\Theta} = \Big[\boldsymbol{\tau}[1] \quad \cdots \quad \boldsymbol{\tau}[K]\Big]^{\mathrm{T}} \in \mathbb{R}^{K \times L}, \tag{4.7}$$

of note is that the absolute amplitude parameter matrix, $|\boldsymbol{\Gamma}|$, is the absolute value of the complex amplitude factor matrix itself. Thus, the complex amplitude factor matrix is its own parameter matrix.

4.2 Proposed Tensor-Based Adaptive Processing Time-Delay Estimation

To adaptively process an input tensor under changing conditions, ST is employed to iteratively estimate the signal subspaces of the data tensor at each sampling epoch k. A HOE-based approach is described in section 4.2.1, while a parametric approach is described in section 4.2.2.

For the sake of convenience, the second and third modes of $\boldsymbol{\mathcal{Y}}[k]$ are permuted, so the singleton dimension remains in the last mode, thus $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M \times Q \times 1}$.

4.2.1 Subspace Tracking HOE+EFBA+ESPS

The Subspace Tracking HOE+EFBA+ESPS (ST-HOE+EFBA+ESPS) approach to adaptive time-delay estimation takes data $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M \times Q}$ at each epoch and extends it dimensionally by applying expanded forward-backward averaging (EFBA) and ESPS to add more dimensions to the data. Similarly to HOE used previously [15], only the dominant singular vectors are tracked, thus eschewing the necessity of having prior knowledge of the model order.

The ST-HOE+EFBA+ESPS TDE update is shown in Algorithm 4, and the whole algorithm is shown in Algorithm 5. The function titled “Subspace Tracking update” represents the application of a subspace tracking algorithm, which can be via iSVD, PAST, FAPI or GYAST.

4.2.2 Subspace Tracking DoA

Performing adaptive time-delay estimation using DoA estimation (ST-DoA) only requires tracking the first-mode subspace of $\boldsymbol{\mathcal{Y}}[k]$.

Algorithm 4 ST-HOE+EFBA+ESPS update

Input: $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M \times Q \times 1}$, number of subarrays ($L_s$), previous subspace vectors $\hat{\mathbf{u}}_1[k-1] \in \mathbb{C}^{M_s}$, $\hat{\mathbf{u}}_2[k-1] \in \mathbb{C}^2$, and $\hat{\mathbf{u}}_3[k-1] \in \mathbb{C}^{L_s}$, previous eigenvalues vector $\hat{\lambda}[k-1] = \big[\hat{\lambda}_1[k-1], \hat{\lambda}_2[k-1], \hat{\lambda}_3[k-1], \big]^{\mathrm{T}} \in \mathbb{R}^3$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delay τ_{LOS}.

1: Apply EFBA and ESPS to $\boldsymbol{\mathcal{Y}}[k]$
2: **for** $m = 1$ to 3 **do**
3: **function** SUBSPACE TRACKING UPDATE($\Big[\tilde{\boldsymbol{\mathcal{Y}}}[k]\Big]_{(n)}$, $\hat{\mathbf{u}}_n[k-1]$, $\hat{\lambda}_n[k-1]$, β)
4: **return** $\hat{\mathbf{u}}_n[k]$, $\hat{\lambda}_n[k]$
5: **end function**
6: **end for**
7: Calculate auto-correlation vector

$$\mathbf{q}[k] = |\boldsymbol{\mathcal{Y}}[k] \times_1 \hat{\mathbf{u}}_1[k] \times_2 \hat{\mathbf{u}}_2[k] \times_3 \hat{\mathbf{u}}_3[k]| \tag{4.8}$$

and apply cubic spline interpolation to estimate the time-delay.

Algorithm 5 ST-HOE+FBA+SPS

Input: $\boldsymbol{\mathcal{Y}} = \big[\boldsymbol{\mathcal{Y}}[1] \sqcup_3 \ldots \sqcup_3 \boldsymbol{\mathcal{Y}}[K]\big]$, number of subarrays L_s, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delays $\boldsymbol{\tau}_{\mathrm{LOS}} = \big[\tau_{\mathrm{LOS}}[1], \ldots, \tau_{\mathrm{LOS}}[K]\big] \in \mathbb{R}^K$.

1: Initialize subspace vectors

$$\hat{\mathbf{u}}_1[0] = \mathbf{e}_{M_s}, \qquad \hat{\mathbf{u}}_2[0] = \mathbf{e}_2, \qquad \hat{\mathbf{u}}_3[0] = \mathbf{e}_{L_s}.$$

2: Initialize eigenvalues

$$\hat{\boldsymbol{\lambda}}[0] = \mathbf{1}_3.$$

3: **for** $k = 1$ to K **do**
4: **function** ST-HOE+FBA+SPS UPD.($\boldsymbol{\mathcal{Y}}[k]$, $\hat{\mathbf{u}}_n[k-1]$ for $n = 1, \ldots, 3$, $\hat{\boldsymbol{\lambda}}[k-1]$)
5: **return** $\tau_{\mathrm{LOS}}[k]$, $\hat{\mathbf{u}}_n[k]$ for $n = 1, \ldots, 3$, $\hat{\boldsymbol{\lambda}}[k]$
6: **end function**
7: **end for**

The ST-DoA/KRF update algorithm is shown in Algorithm 6, while the whole algorithm is shown in Algorithm 7.

Algorithm 6 ST-DoA update

Input: $\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{\mathcal{Y}}[1], \ldots, \boldsymbol{\mathcal{Y}}[K]\right] \in \mathbb{C}^{M \times K \times Q}$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delay τ_{LOS}.

1: Update subspace estimate
2: **function** SUBSPACE TRACKING UPDATE($\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(1)}$, $\hat{\mathbf{U}}[k-1]$, $\hat{\mathbf{P}}[k-1]$, β)
3: **return** $\hat{\mathbf{U}}[k]$, $\hat{\mathbf{P}}_n[k]$
4: **end function**
5: Estimate spatial frequencies

$$\boldsymbol{\Psi} = \left(\mathbf{J}_1 \cdot \hat{\mathbf{U}}[k]\right)^{\dagger} \mathbf{J}_2 \cdot \hat{\mathbf{U}}[k] = \mathbf{T} \cdot \boldsymbol{\Phi} \cdot \mathbf{T}^{-1},$$
$$\hat{\boldsymbol{\mu}} = \angle \operatorname{diag}\left\{\boldsymbol{\Phi}\right\}.$$

6: Rebuild array response matrix using estimated spatial frequencies $\hat{\boldsymbol{\mu}}$.
7: Estimate code factor matrix

$$\hat{\tilde{\mathbf{C}}}[k] = \left[\boldsymbol{\mathcal{Y}}[k]\right]_{(2)} \cdot \left(\hat{\mathbf{A}}[k]^{\mathrm{T}}\right)^{\dagger}$$

8: Finish correlation

$$\mathbf{q}[k] = \mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}[k](\cdot, s_{\text{LOS}}).$$

9: Apply cubic spline interpolation to estimate time-delay.

4.3 Numerical Simulation Results

As in previous sections, a 1000 realization Monte Carlo simulation is performed to measure time-delay estimation performance. Learning rate $\beta_{\text{iSVD}} = 0.95$, $\beta_{\text{PAST}} = 0.75$, $\beta_{\text{FAPI}} = 0.75$, and $\beta_{\text{GYAST}} = 0.75$ were used.

A benchmark method, with known $\mathbf{A}$ has been added for comparison. At each k-th epoch the PRS code factor matrix is estimated

$$\hat{\tilde{\mathbf{C}}}[k]_{\text{Known}} = \left[\boldsymbol{\mathcal{Y}}[k]\right]_{(3)} \left(\mathbf{A}[k]^{\mathrm{T}}\right)^{\dagger}. \tag{4.9}$$

Results are in terms of mean absolute error (MAE) multiplied by the speed of light. For I Monte Carlo realization, the MAE is

$$\text{MAE} = c \cdot \frac{1}{I} \sum_{i=1}^{I} \left|\tau_{\text{LOS}} - \hat{\tau}_{\text{LOS}}\right|. \tag{4.10}$$

Algorithm 7 ST-DoA

Input: $\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{\mathcal{Y}}[1] \sqcup_3 \ldots \sqcup_3 \boldsymbol{\mathcal{Y}}[K]\right]$, number of subarrays L_s, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delays $\boldsymbol{\tau}_{\text{LOS}} = \left[\tau_{\text{LOS}}[1], \ldots, \tau_{\text{LOS}}[K]\right] \in \mathbb{R}^K$.

1: Initialize subspace estimate

$$\hat{\mathbf{U}}[0] = \mathbf{I}_{M \times L}.$$

2: Initialize covariance matrix

$$\hat{\mathbf{P}}[0] = \mathbf{I}_L.$$

3: **for** $k = 1$ to K **do**
4: **function** ST-DoA UPD.($\boldsymbol{\mathcal{Y}}[k]$, $\hat{\mathbf{U}}[k-1]$, $\hat{\mathbf{P}}[k-1]$)
5: **return** $\tau_{\text{LOS}}[k]$, $\hat{\mathbf{U}}[k]$, $\hat{\mathbf{P}}[k]$.
6: **end function**
7: **end for**

4.3.1 Static simulation

Adaptive processing under a static scenario with the same simulation parameters used in previous numerical simulations. Unlike previous simulations, each figure is for a given LOS-NLOS delay difference of $\Delta\tau$ and the x-axis is temporal. Simulations were performed for $L = 2$ and $L = 3$. A forgetting factor of $\beta = 0.97$ was used.

4.3.1.1 Static simulation, $L = 2$

In Figures 4.2 to 4.11 time-delay estimation performance is shown for $\Delta T/T_c$ from 0.1 to 1.0 for $L = 2$ (1 LOS component, 1 NLOS component).

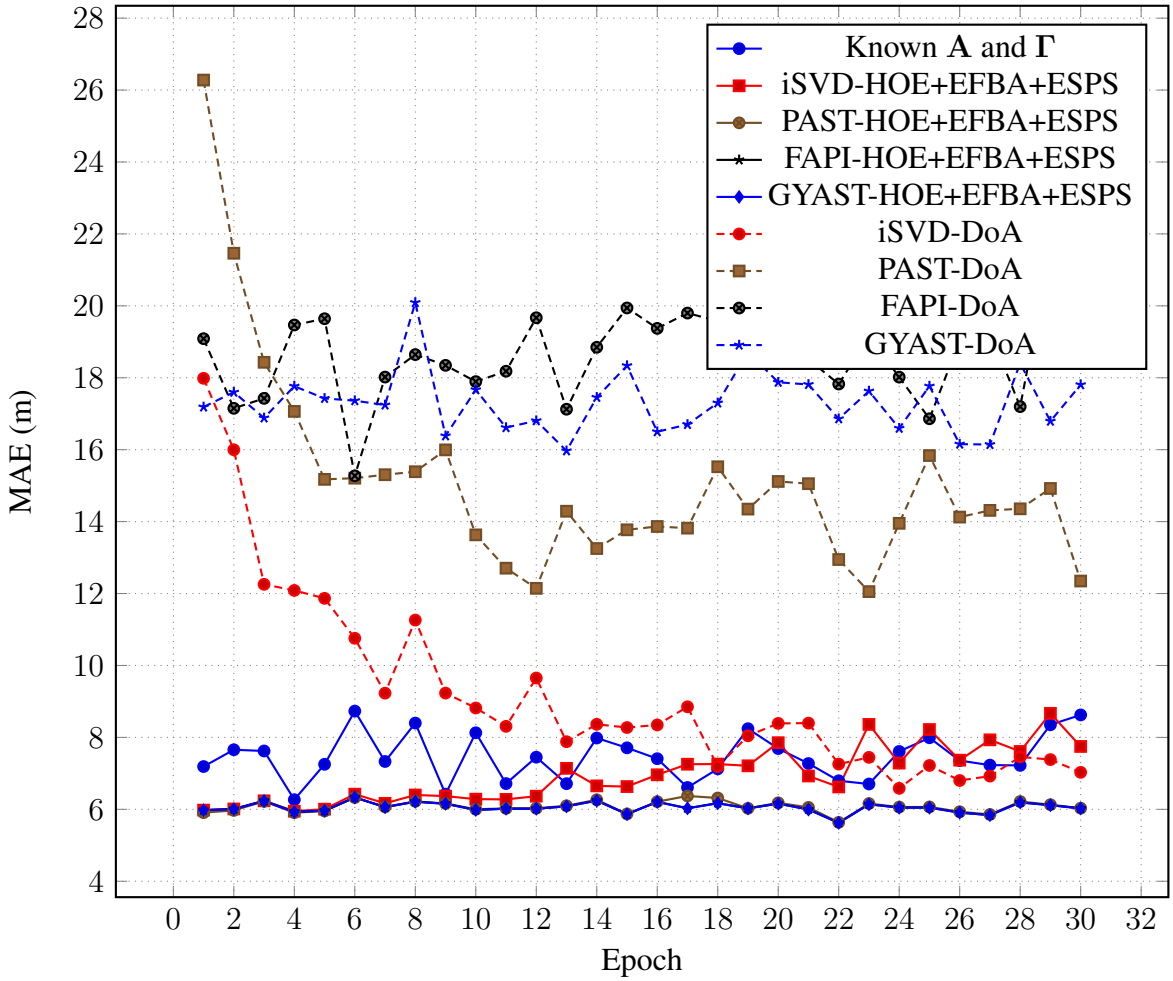

Figure 4.2: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.1T_c$.

In Figure 4.2 the performance of the eigenfiltering approaches not only outperforms filtering with prior knowledge of the factor matrix $\mathbf{A}$, they also outperform the parametric approaches with the exception of iSVD-DoA. Since iSVD directly estimates the covariance matrix and uses this to calculate the subspace via eigendecomposition, subspace estimation is especially precise and this behavior is consistent for other $\Delta\tau$.

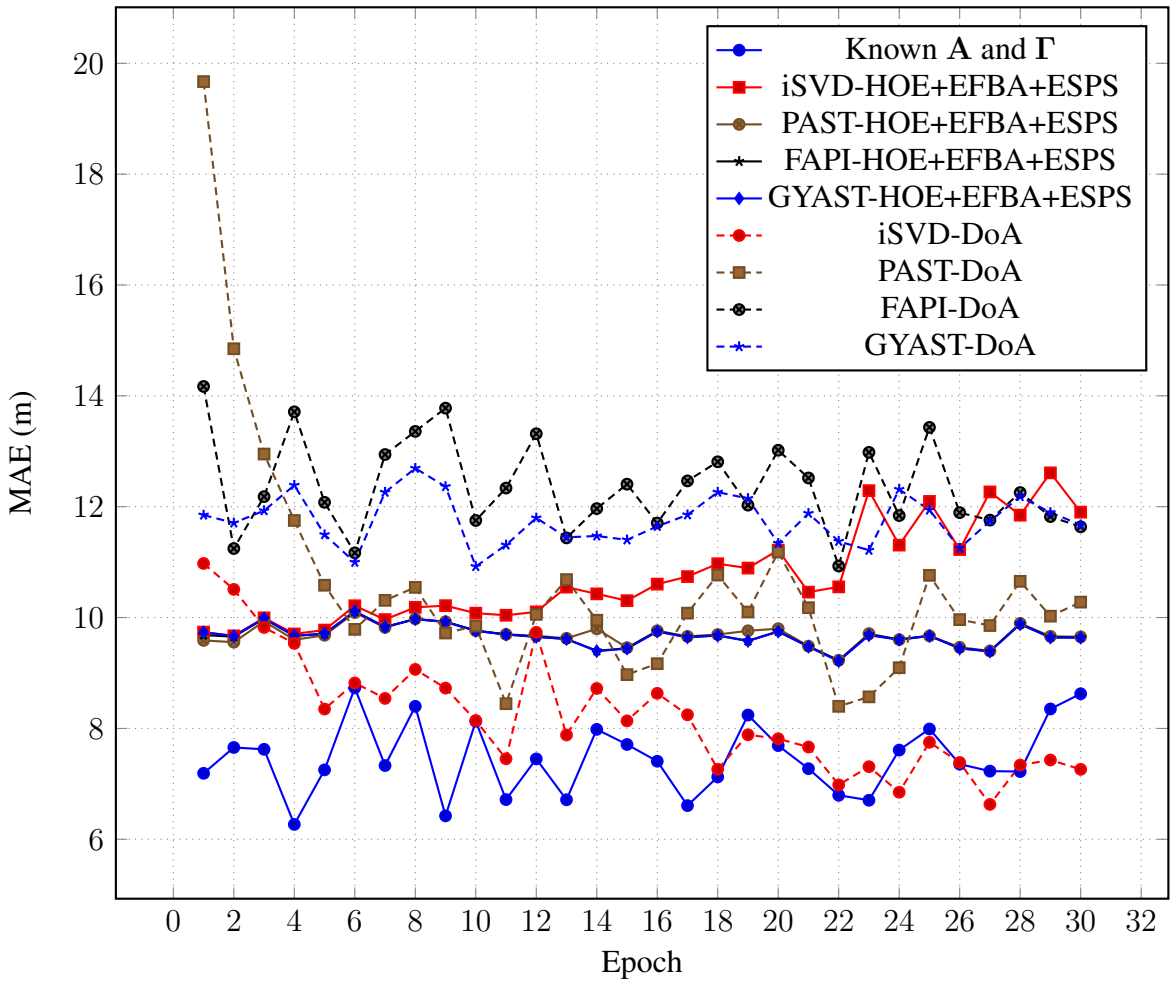

Figure 4.3: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.2T_c$.

In Figure 4.3 we have increased separation between LOS and NLOS components, consequently the performance of the ST-HOE+EFBA+ESPS approaches has started to decrease and the performance of the ST-DoA approaches improved. In the case of iSVD-HOE+EFBA+ESPS, it becomes less consistent with each epoch. This is due to the iSVD only tracking a rank-1 subspace.

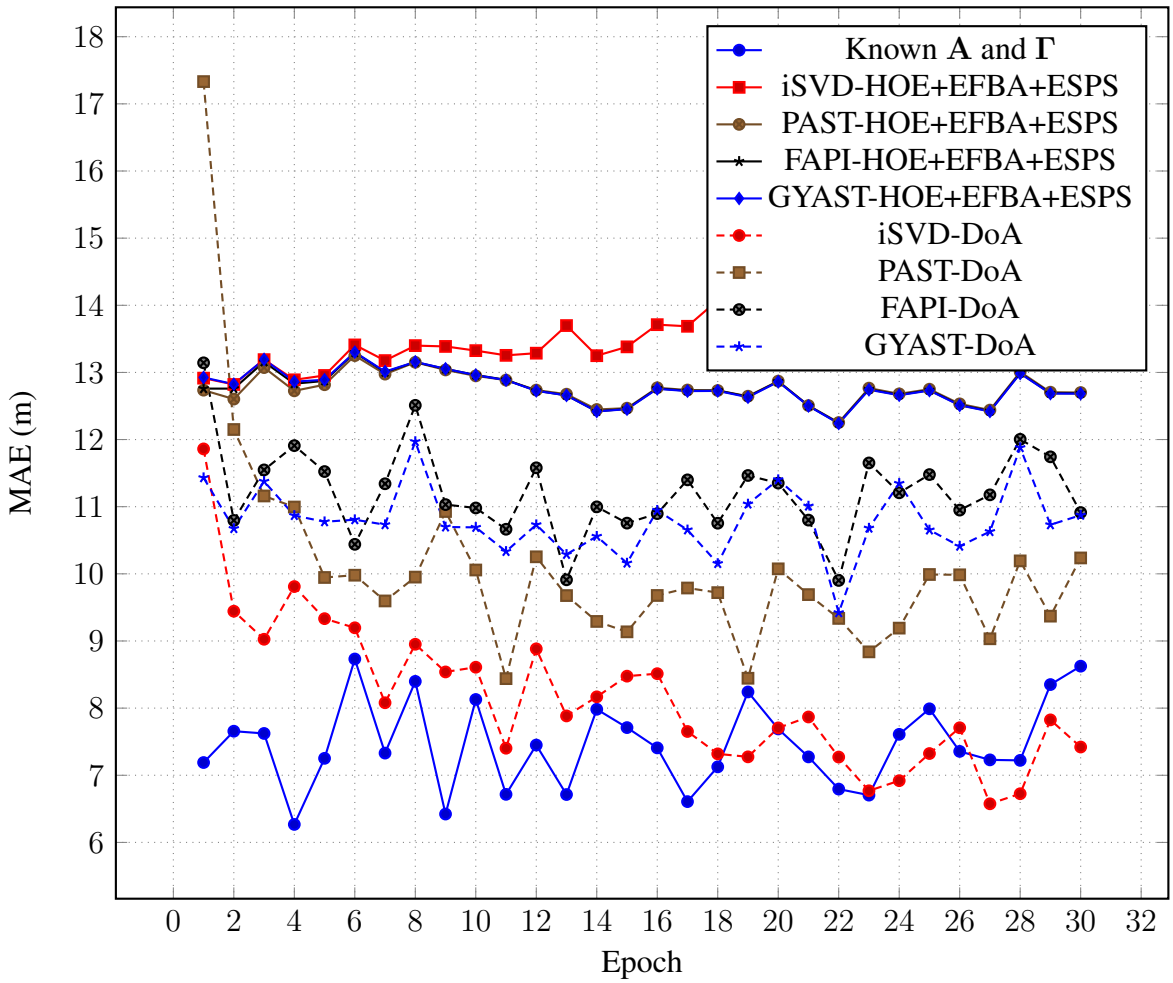

Figure 4.4: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.3T_c$.

In Figure 4.4, after the first epoch, all parametric approaches outperform the eigenfilter-based approaches. The iSVD-DoA approach has performance nearing filtering with a priori knowledge while the other DoA approaches have lesser but consistent performance. The PAST, FAPI, and GYAST eigenfilter approaches have performance consistent and nearly matching performance. The iSVD-HOE+EFBA+ESPS approach again diverges with time.

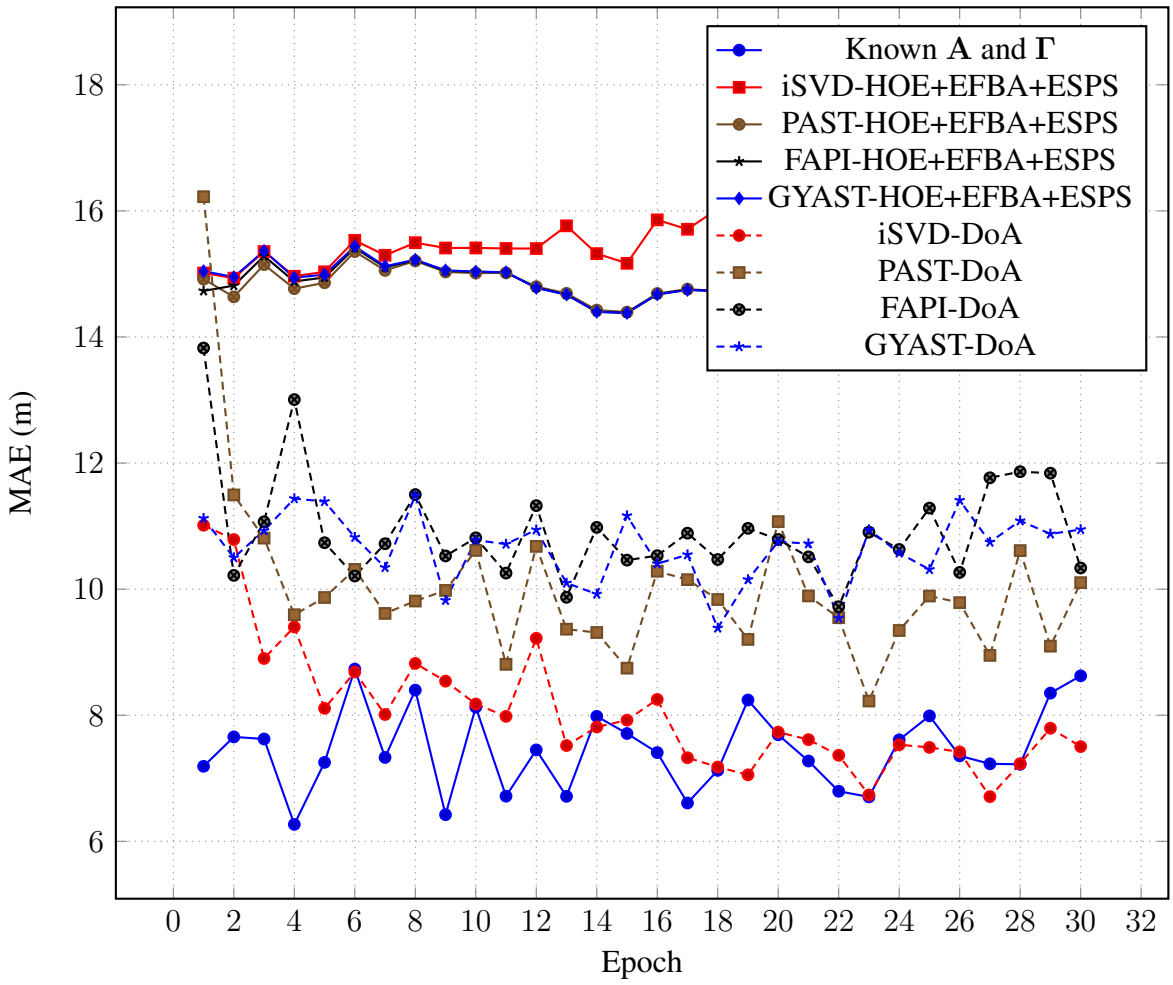

Figure 4.5: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.4T_c$.

In Figure 4.5 performance for the parametric approaches has remained consistent with Figure 4.4 while the eigenfilter-based approaches has deteriorated further.

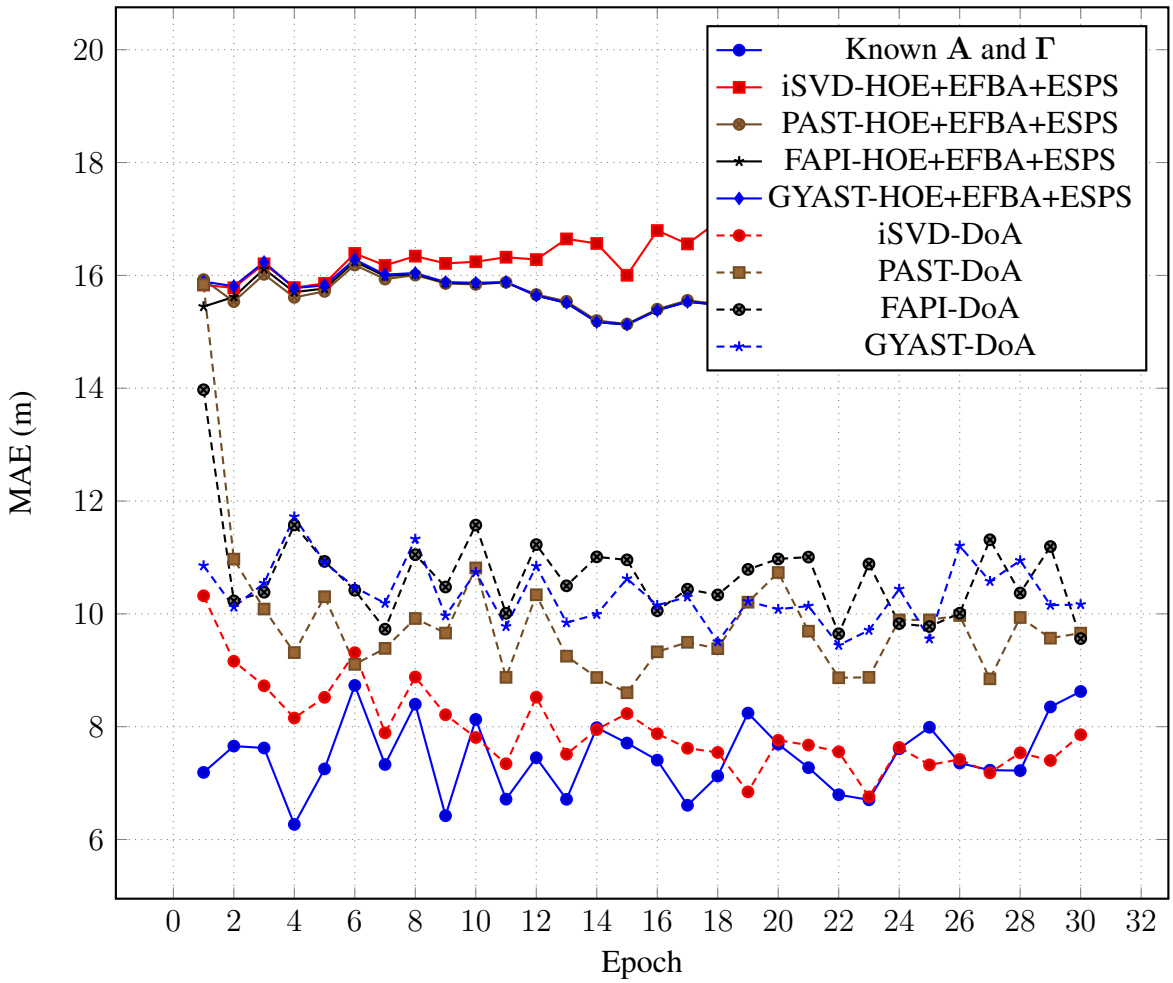

Figure 4.6: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.5T_c$.

In Figures 4.6 to 4.8 performance for all approaches has remained consistent with Figure 4.5 with slight increases in performance for the parametric approaches and an increase for the eigenfiltering approach in Figure 4.8.

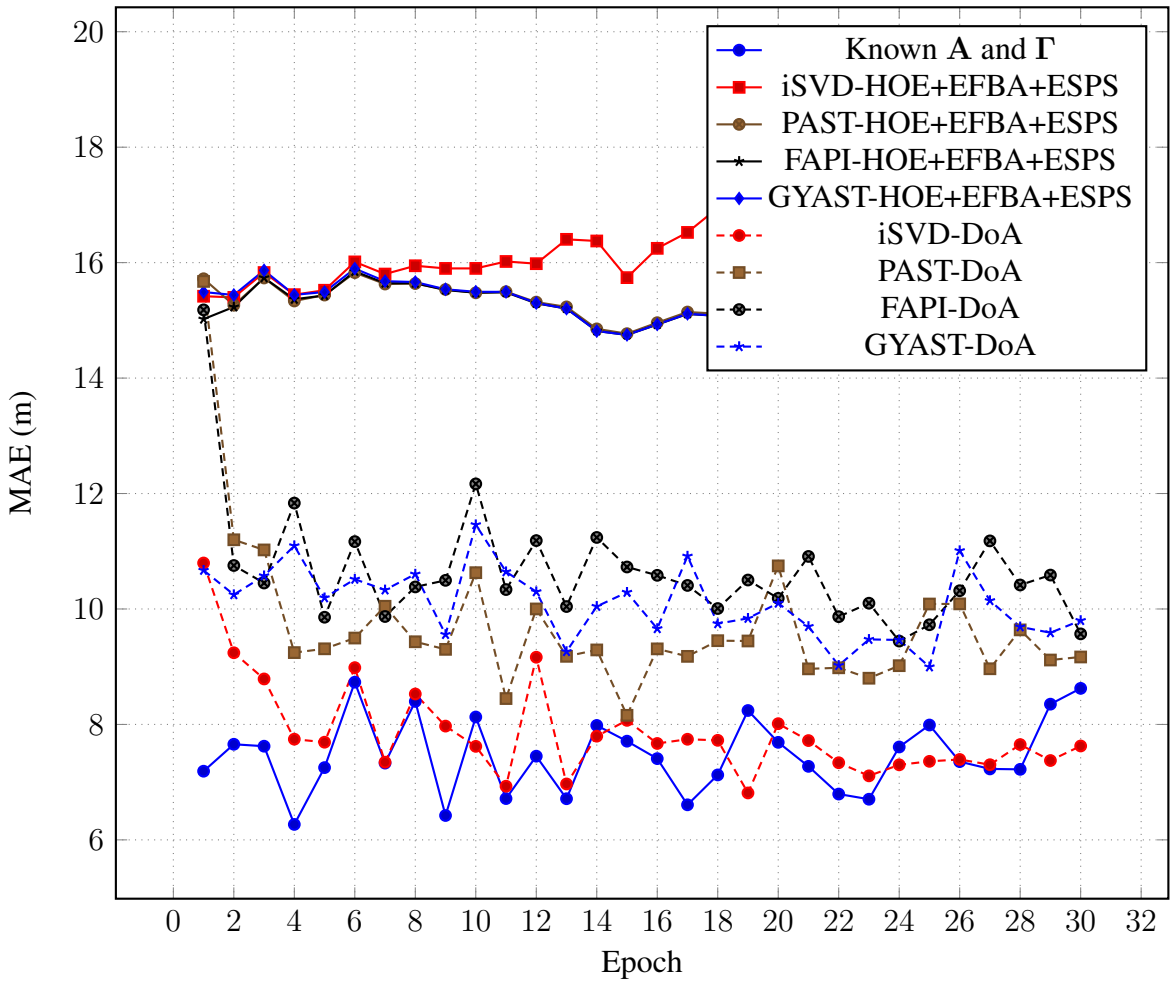

Figure 4.7: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.6T_c$.

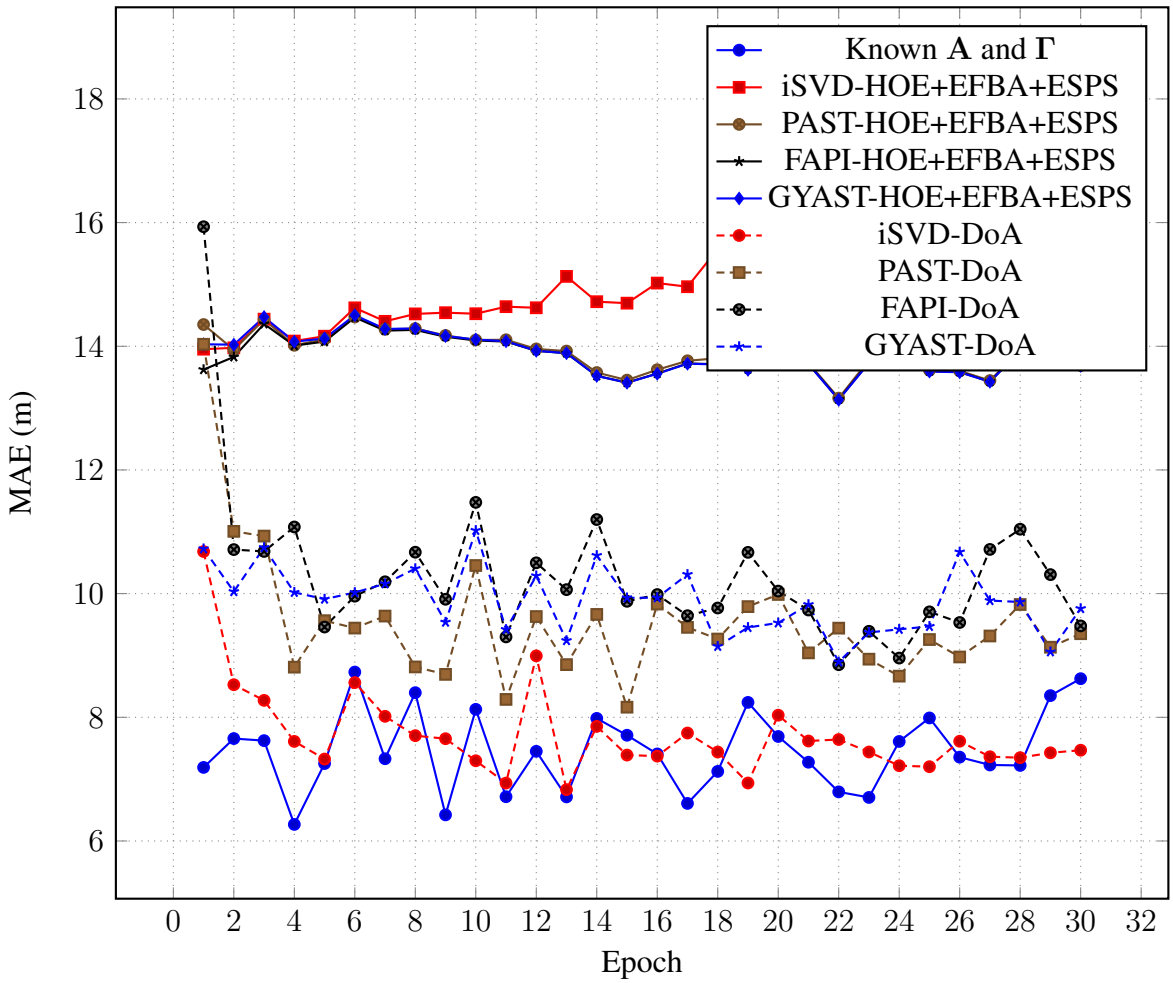

Figure 4.8: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.7T_c$.

In Figures 4.9 to 4.11 performance improves for all approaches with the eigenfiltering approaches nearly matching the performance of the parametric approaches. In all cases, the iSVD-HOE+EFBA+ESPS approach increasingly diverges with the passage of time.

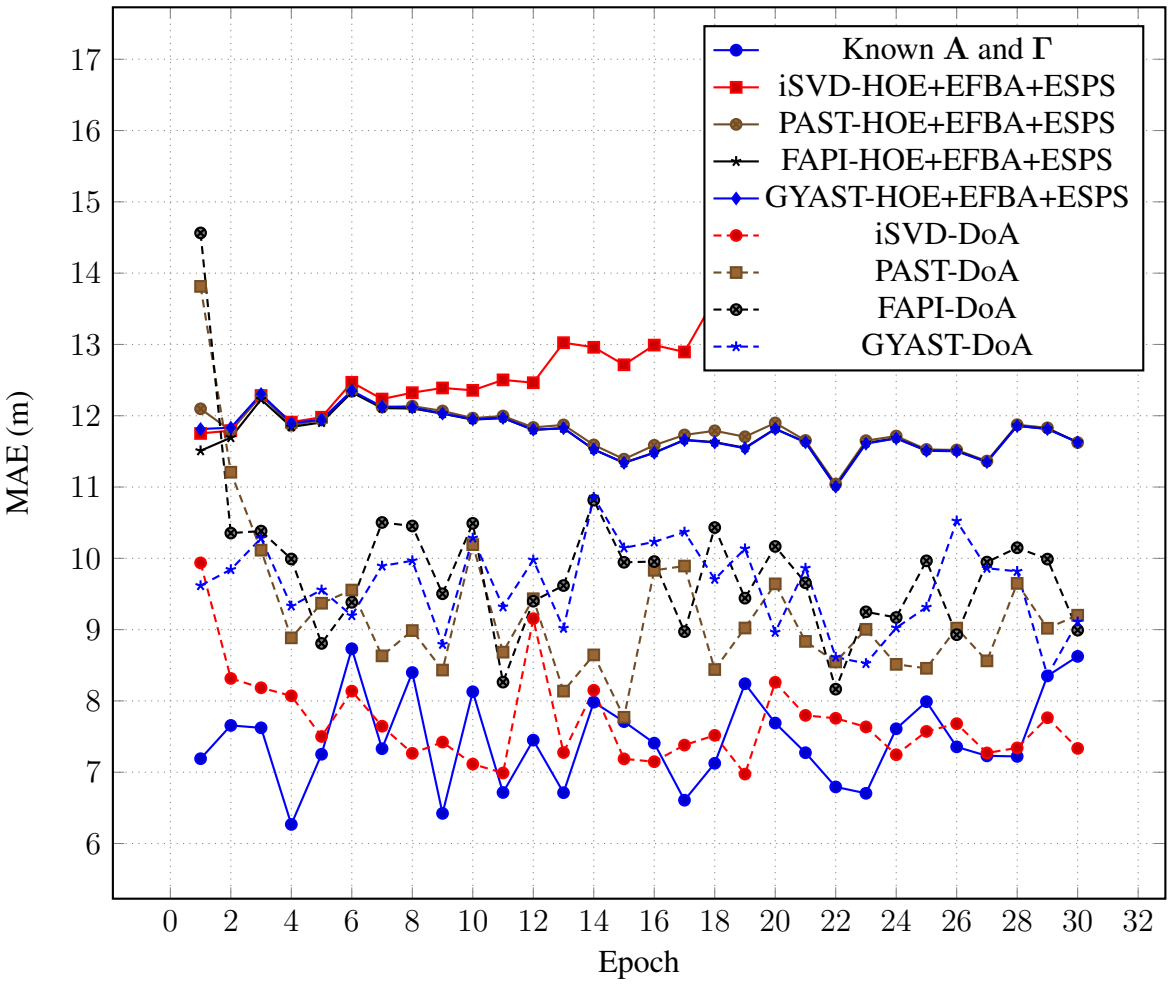

Figure 4.9: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.8T_c$.

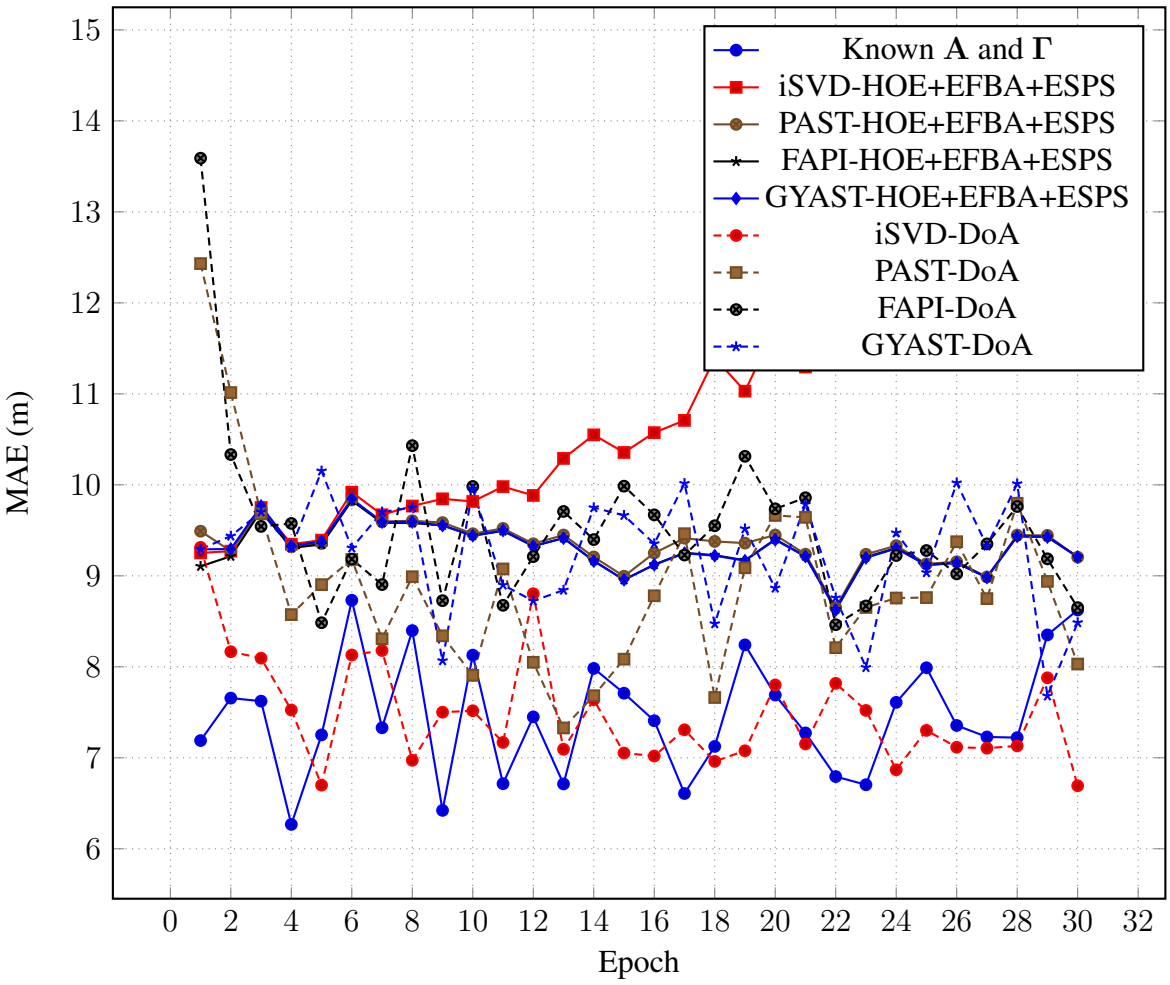

Figure 4.10: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 0.9T_c$.

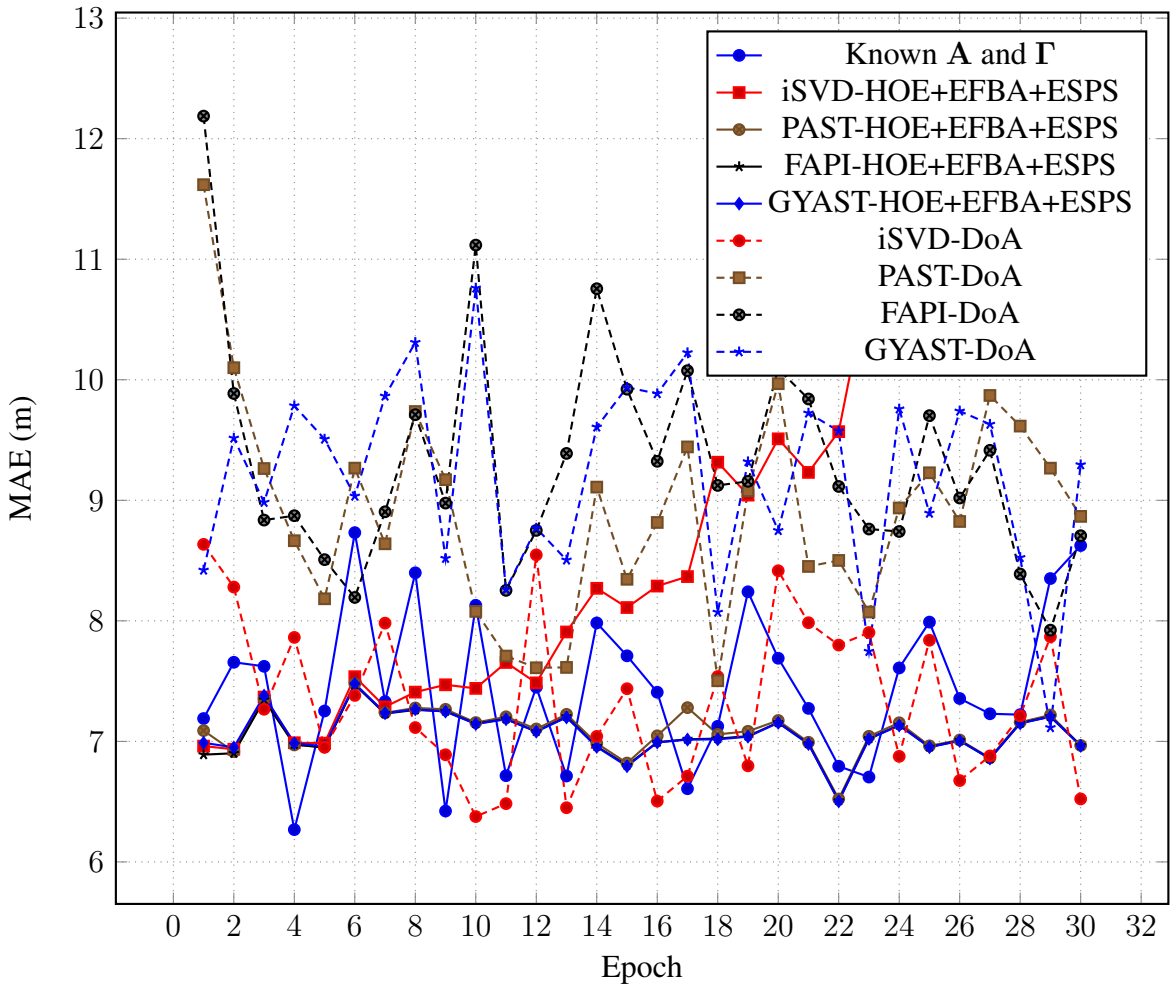

Figure 4.11: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

In all nearly all simulations, the parametric approach outperforms the eigenfiltering approaches and the rank-1 iSVD subspace tracking causes the eigenfiltering approach to decrease in performance with increasing epochs. The exception is in Figure 4.2, where the LOS and NLOS components are clustered, resulting in a rank-deficient factor matrix.

4.3.1.2 Static simulation, $L = 3$

In Figures 4.12 to 4.21 time-delay estimation performance is shown for $\Delta T/T_c$ from 0.1 to 1.0 for $L = 3$ (1 LOS components and 2 NLOS components).

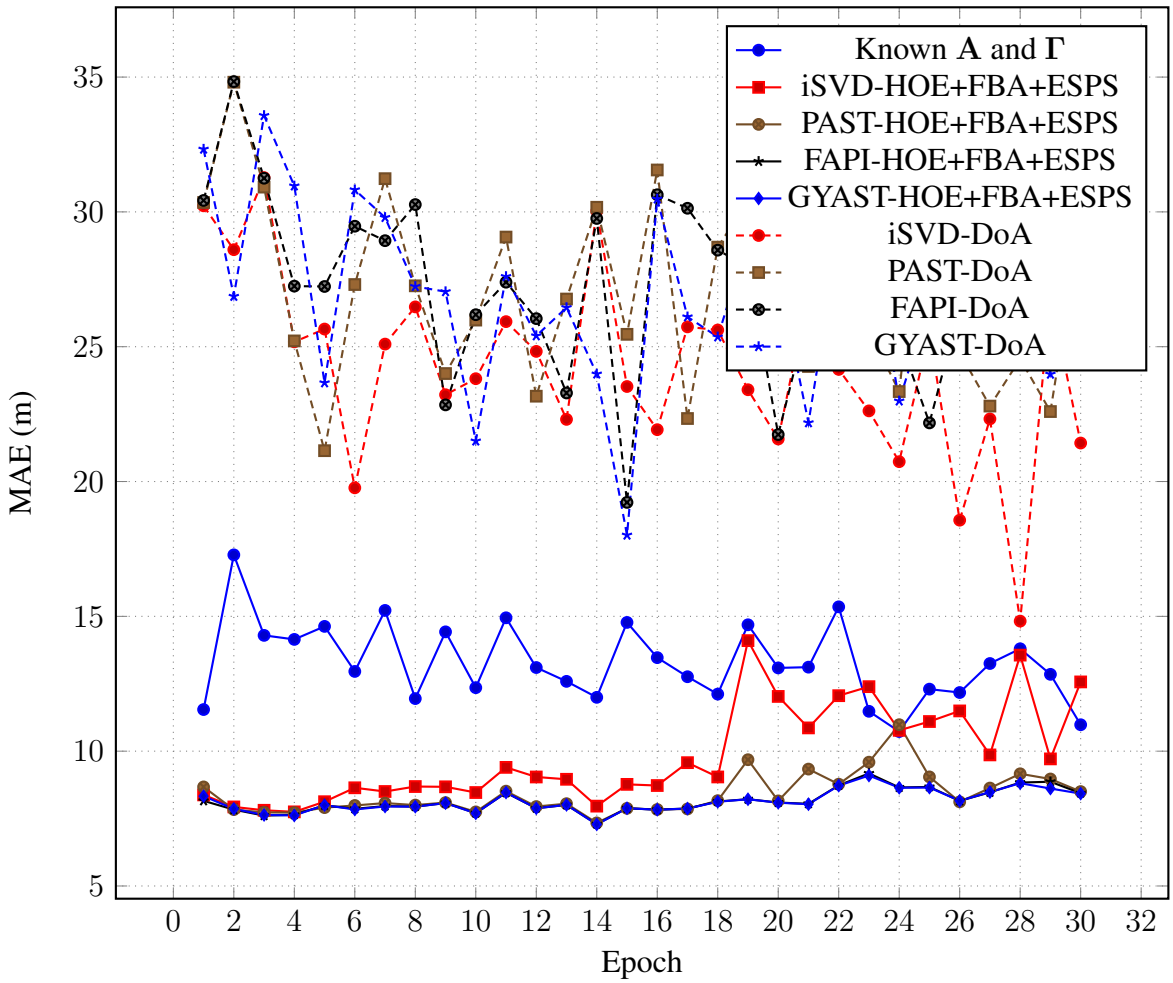

Figure 4.12: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.1T_c$.

In Figure 4.12, where the components are the most clustered, and similarly to the case where $L = 2$, the eigenfiltering approaches outperform the parametric approaches and even filtering with a priori knowledge. The loss in performance due to NLOS clustering is present for all $\Delta\tau$.

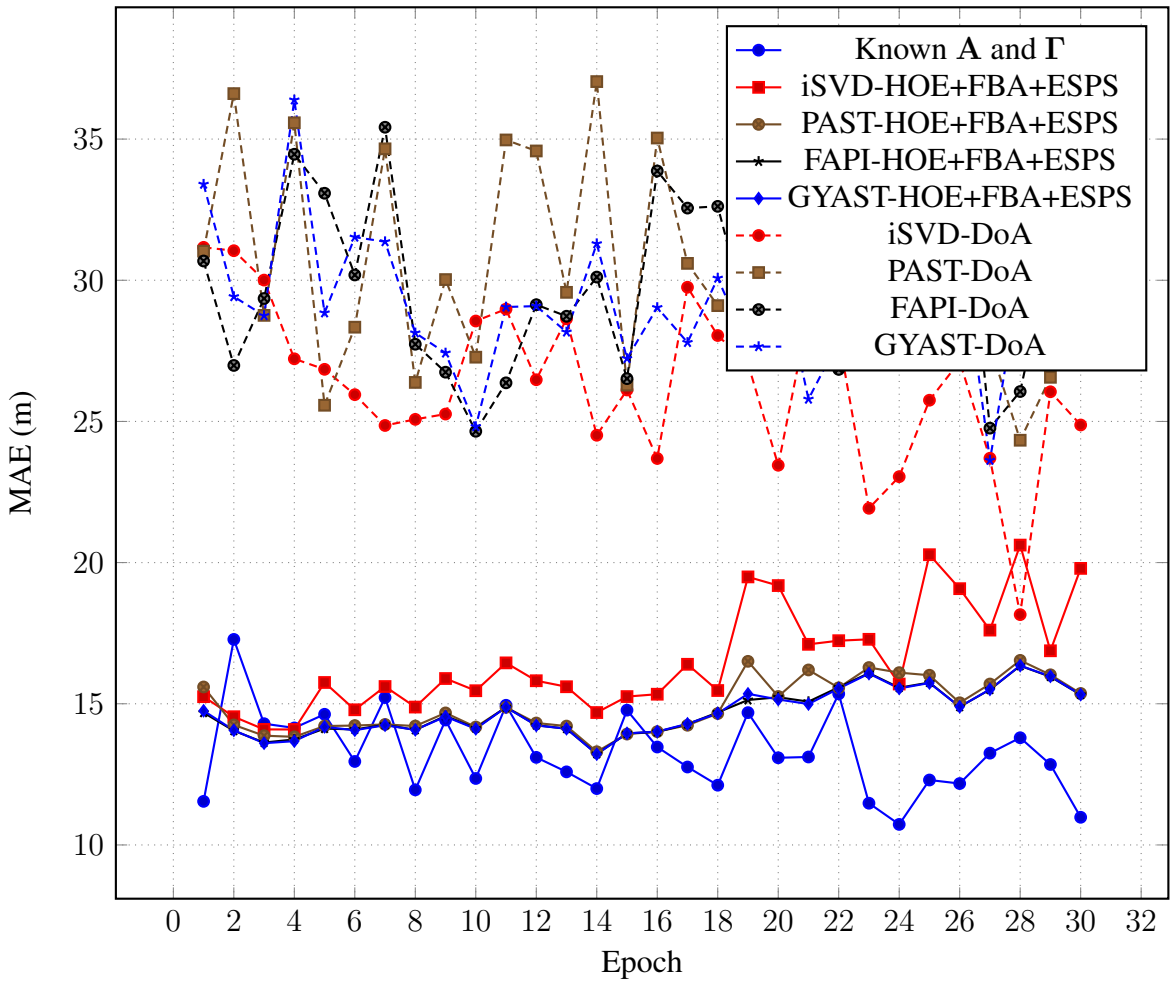

Figure 4.13: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.2T_c$.

In Figures 4.13 to 4.18, similarly to the $L = 2$ case, there's a performance decrease as $\Delta\tau$ increases for the eigenfiltering approaches. The parametric approaches also suffer from a decrease in performance but the impact is less severe. For $\Delta\tau = 0.7T_c$ the iSVD-DoA approach has outperformed the eigenfiltering approaches.

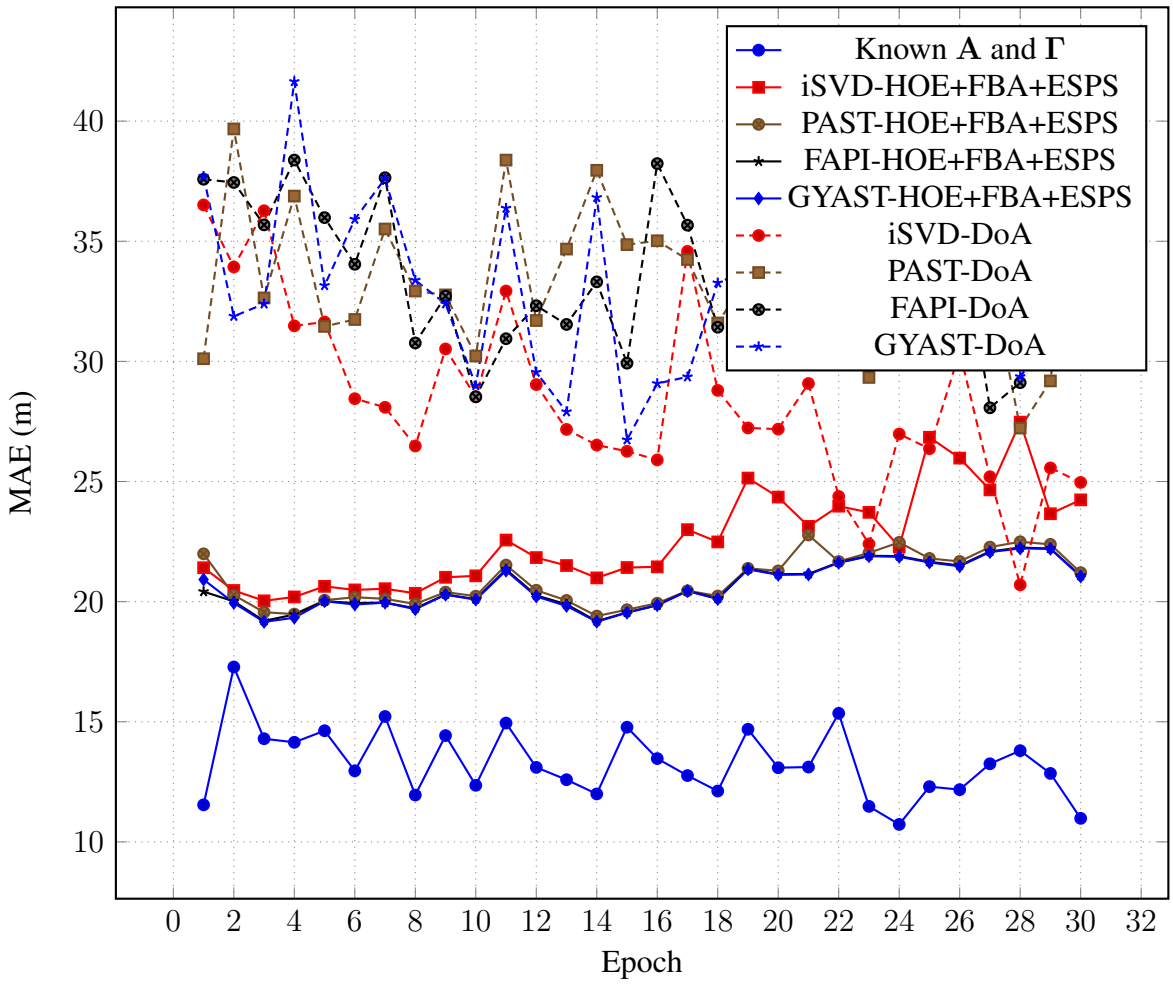

Figure 4.14: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.3T_c$.

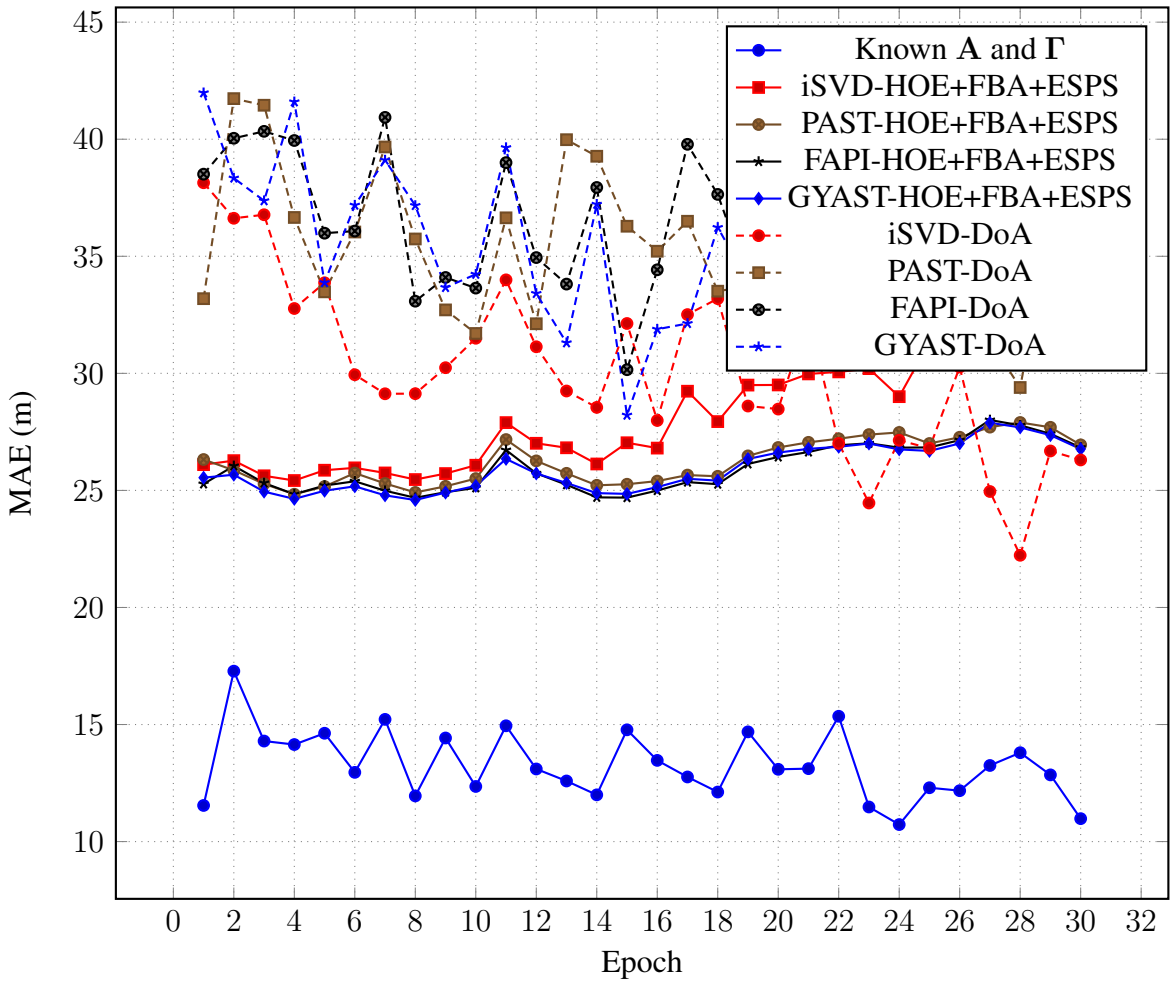

Figure 4.15: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.4T_c$.

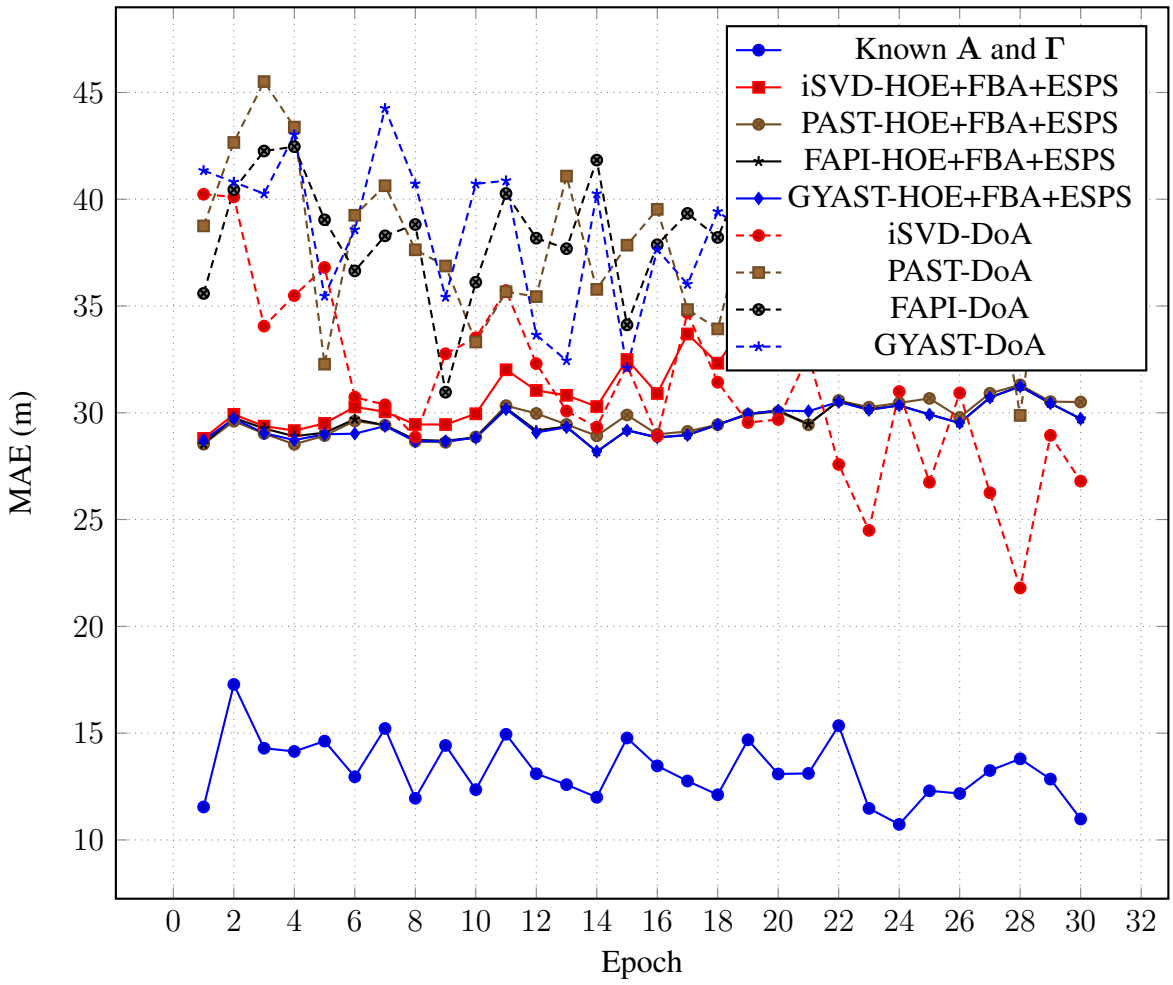

Figure 4.16: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.5T_c$.

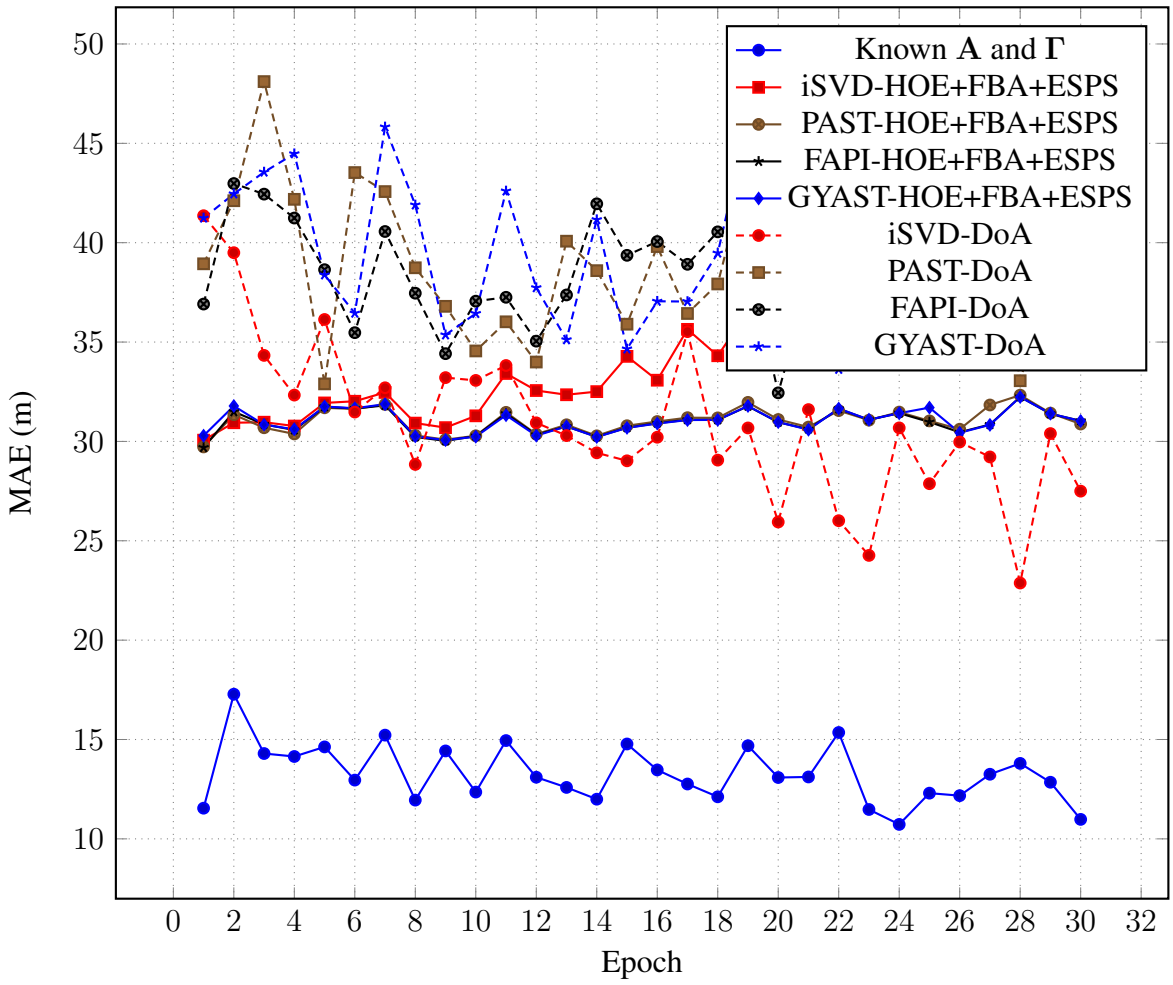

Figure 4.17: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.6T_c$.

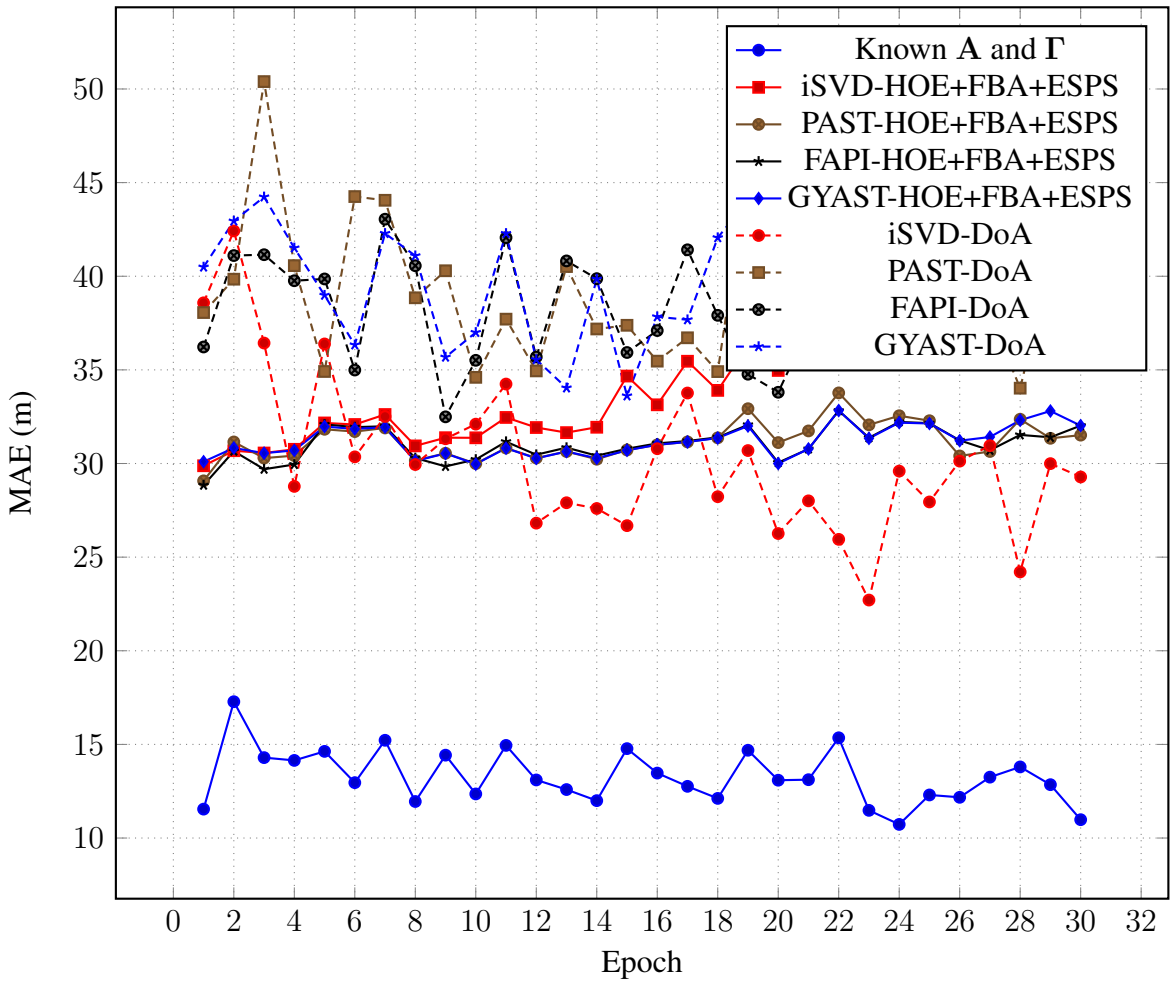

Figure 4.18: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.7T_c$.

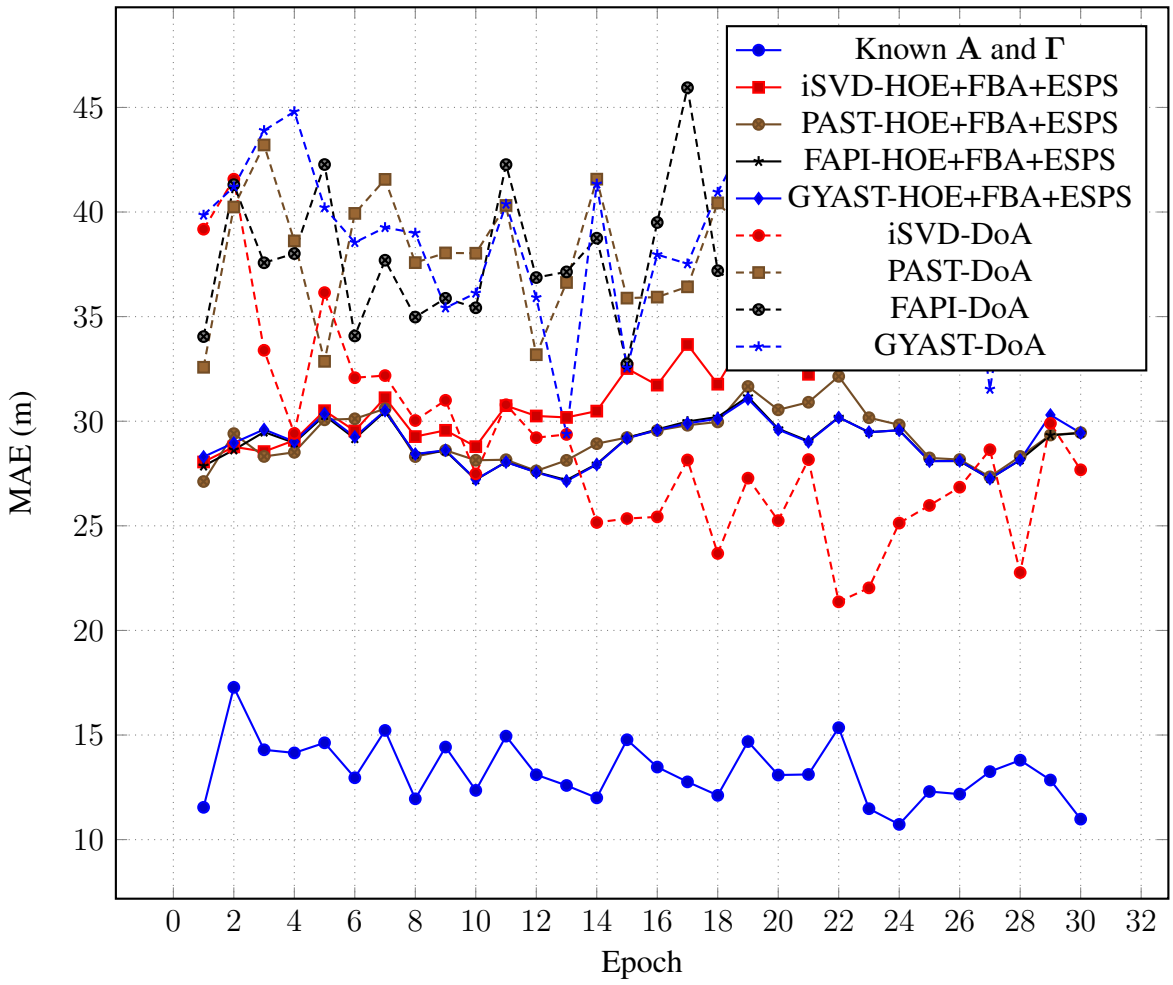

Figure 4.19: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.8T_c$.

In Figures 4.19 to 4.21, performance improves with increasing $\Delta\tau$ for the eigenfiltering approaches, with the iSVD-DoA approach outperforming the eigenfiltering approaches for $\Delta\tau = 0.8T_c$ and $0.9T_c$ but not $\Delta\tau = 1.0T_c$.

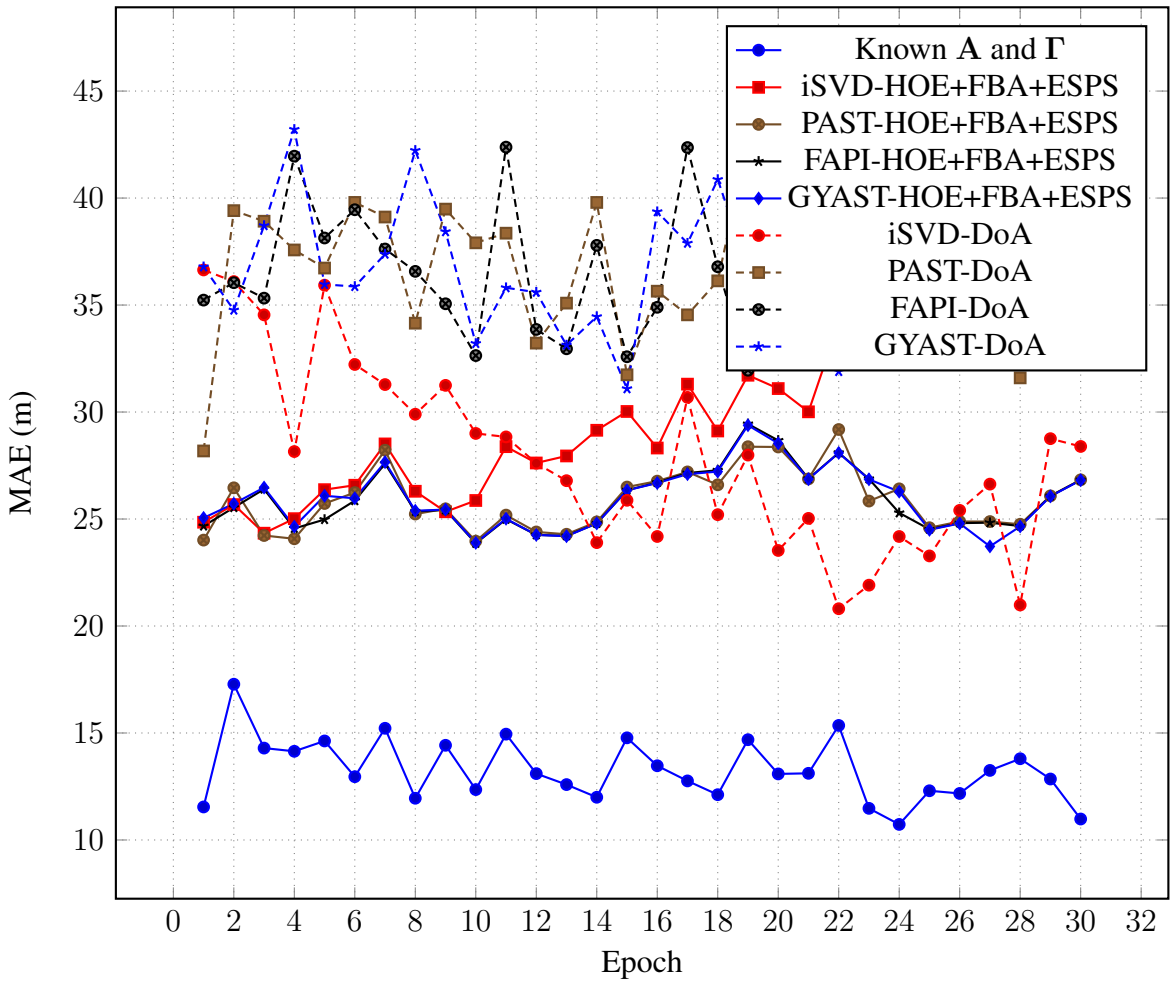

Figure 4.20: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 0.9T_c$.

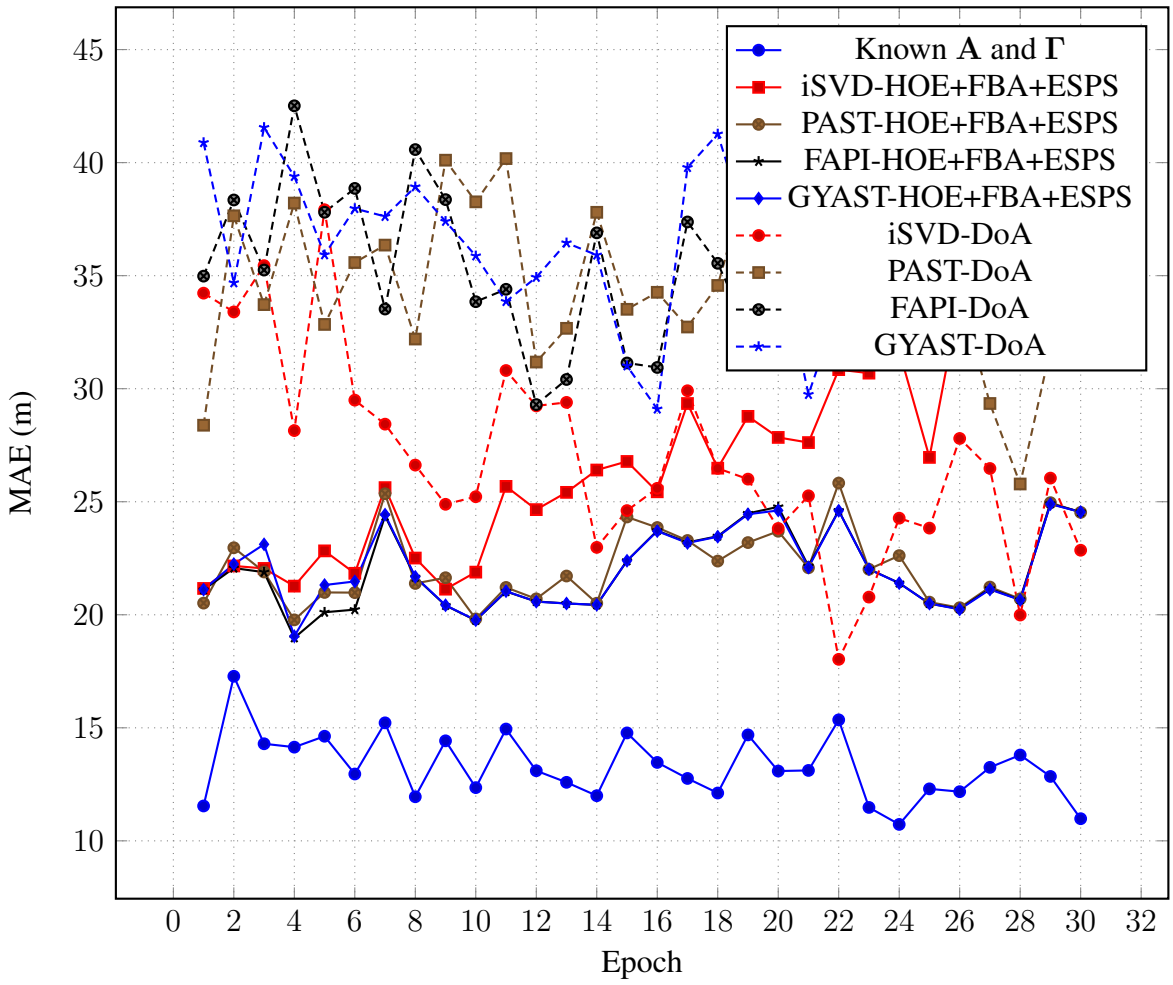

Figure 4.21: Time-delay estimation error in meters for $L = 3$, $\Delta\tau = 1.0T_c$.

Like the case for $L = 2$, the eigenfiltering approaches suffer a decrease then increase in performance as $\Delta\tau$ increases, similarly to the bell-shaped performance observed previously [16]. Again iSVD subspace tracking presents poor performance when tracking a rank-1 subspace.

In all cases the clustered NLOS components cause a rank deficiency that negatively impacts performance for nearly all approaches for all $\Delta\tau$, the exception being when $\Delta\tau = 0.1T_c$, where all components are clustered, thus resembling the rank-1 case.

Chapter 5

Tensor-Based Time-Delay Estimation via Sequentially Truncated Subspace Tracking

In this chapter we propose adaptive tensor-based schemes for time-delay estimation using different subspace tracking algorithms and a sequential truncation strategy. The data model is extended to an uniform rectangular array considering a dynamic scenario.

We propose three tensor-based adaptive processing schemes for GNSS third-order tensor stream. All utilize a truncation strategy [18] applied to subspace tracking to reduce the amount of data that needs to be processed for each mode.

The first is a modification of the HOE with EFBA and ESPS. Rank-1 subspace tracking is performed iteratively by unfolding each mode and applying the n-mode product of the conjugate transpose (Hermitian) of the estimated subspace vector to compress the data before proceeding to the next mode.

The second is based on the variant canonical polyadic decomposition via generalized eigenvalue decomposition using both left and right eigenvector matrices. Subspace tracking is performed iteratively by unfolding each mode n-mode product of the conjugate transpose (Hermitian) of the estimated subspace to compress the data before proceeding to the next mode.

The third is based on subspace tracking via Kronecker-structured projections [79, 81] to estimate the improved signal subspace used in standard tensor ESPRIT [86] with a sequential truncation strategy to compress the amount of data for each mode.

All three proposed schemes are tested using four subspace tracking algorithms, iSVD, PAST, FAPI, and GYAST. The iSVD subspace tracking is used as a baseline.

5.1 Fourth-order Time-Varying Data Model

The data model considered is similar to the data model in Chapter 4 but for a uniform rectangular array (URA) with array response matrices $\mathbf{A}_x[k] \in \mathbb{C}^{M_x \times L}$ and $\mathbf{A}_y[k] \in \mathbb{C}^{M_y \times L}$ and complex amplitude in the last mode.

Both array response matrices have direction of arrival parameters azimuth $\boldsymbol{\theta}[k] \in \mathbb{R}^L$ and elevation $\boldsymbol{\phi}[k] \in \mathbb{R}^L$. Their respective spatial frequency vectors are

$$\boldsymbol{\mu}_x[k] = 2\pi \cdot \frac{\Delta_x}{\lambda} \cos \boldsymbol{\theta}[k] \odot \sin \boldsymbol{\phi}[k] \in \mathbb{R}^L, \tag{5.1}$$

$$\boldsymbol{\mu}_y[k] = 2\pi \cdot \frac{\Delta_y}{\lambda} \sin \boldsymbol{\theta}[k] \odot \sin \boldsymbol{\phi}[k] \in \mathbb{R}^L, \tag{5.2}$$

and assuming a perfectly calibrated array ($\Delta_x = \Delta_y = \lambda/2$) we have

$$\boldsymbol{\mu}_x[k] = \pi \cdot \cos \boldsymbol{\theta}[k] \odot \sin \boldsymbol{\phi}[k] \in \mathbb{R}^L, \tag{5.3}$$

$$\boldsymbol{\mu}_y[k] = \pi \cdot \sin \boldsymbol{\theta}[k] \odot \sin \boldsymbol{\phi}[k] \in \mathbb{R}^L. \tag{5.4}$$

The complex amplitude matrix is placed in the last mode to not leave any singleton dimension in the first three modes

$$\boldsymbol{\mathcal{Y}}[k] = \boldsymbol{\mathcal{I}}_{4,L} \times_1 \mathbf{A}_x[k] \times_2 \mathbf{A}_y[k] \times_3 \tilde{\mathbf{C}}[k] \times_4 \boldsymbol{\Gamma}(k, \cdot) + \tilde{\boldsymbol{\mathcal{N}}}(\cdot, \cdot, \cdot, k), \tag{5.5}$$

$$= \boldsymbol{\mathcal{I}}_{4,L} \times_1 \mathbf{A}_x[k] \times_2 \mathbf{A}_y[k] \times_3 \tilde{\mathbf{C}}[k] \times_4 \boldsymbol{\Gamma}[k] + \tilde{\boldsymbol{\mathcal{N}}}[k] \in \mathbb{C}^{M_x \times M_y \times Q \times 1}. \tag{5.6}$$

5.2 Proposed Sequentially Truncated Subspace Tracked Higher-Order Eigenfilter for Time-delay Estimation (ST-ST-HOE)

The proposed approach begins by applying pre-processing to the received data tensor $\boldsymbol{\mathcal{Y}}[k]$. EFBA is performed by concatenating in the fourth mode the data tensor with the n-mode product of the conjugate of the data tensor and the corresponding exchange matrix in each mode,

$$\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA}} = \left[\boldsymbol{\mathcal{Y}}[k] \sqcup_4 \boldsymbol{\mathcal{Y}}[k]^* \times_1 \boldsymbol{\Pi}_{M_x} \times_2 \boldsymbol{\Pi}_{M_y}\right] \in \mathbb{C}^{M_x \times M_y \times Q \times 2}, \tag{5.7}$$

and permuting the last two dimensions, resulting in $\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA}} \in \mathbb{C}^{M_x \times M_y \times 2 \times Q}$.

For expanded SPS, the number of subarrays, L_s, for both array response matrices has to be the same. Thus, for $\mathbf{M} = \left[M_x, M_y\right]^{\mathrm{T}}$, the number of subarrays, $\mathbf{M}_s = \left[M_x^{(s)}, M_y^{(s)}\right]^{\mathrm{T}}$ is

$$\mathbf{M}_s = \mathbf{M} - L_s + 1. \tag{5.8}$$

Expanded SPS is performed by concatenating in the fifth mode the data tensor with the L_s n-mode products of the data tensor and corresponding exchange matrices in each mode and subarray,

$$\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA+ESPS}} = \Big[\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA}} \sqcup_5 \boldsymbol{\mathcal{Y}}[k]_{\text{EFBA}} \times_1 \mathbf{J}_{M_x^{(s)},1} \times_2 \mathbf{J}_{M_y^{(s)},1} \sqcup_5 \cdots \\ \cdots \sqcup_5 \boldsymbol{\mathcal{Y}}[k]_{\text{EFBA}} \times_1 \mathbf{J}_{M_x^{(s)},L_s} \times_2 \mathbf{J}_{M_y^{(s)},L_s}\Big] \in \mathbb{C}^{M_x^{(s)} \times M_x^{(s)} \times 2 \times Q \times L_s}, \quad (5.9)$$

and again permuting the last two dimensions, resulting in $\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA+ESPS}} \in \mathbb{C}^{M_x \times M_y \times 2 \times L_s \times Q}$.

The higher-order eigenfilter at each k-th iteration is updated by iterating from the first to the fourth mode, estimating the rank-1 subspace vector of the mode unfolding via ST, then performing the corresponding n-mode product of the conjugate transpose of the subspace vector.

We initialize this process by copying the data in $\boldsymbol{\mathcal{Y}}[k]_{\text{EFBA+ESPS}}$ in $\tilde{\boldsymbol{\mathcal{Y}}}[k]$. Starting with the first mode, we estimate the subspace vector using a subspace tracking function[1] $\text{S. T.}(\cdot)$[2]. The estimated subspace vector is used to compress the data in $\tilde{\boldsymbol{\mathcal{Y}}}[k]$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \boldsymbol{\mathcal{Y}}[k]_{\text{EFBA+ESPS}} \in \mathbb{C}^{M_x^{(s)} \times M_y^{(s)} \times 2 \times L_s \times Q}, \quad (5.10)$$

$$\hat{\mathbf{u}}_1[k] = \text{S. T.}\left(\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(1)}\right) \in \mathbb{C}^{M_x^{(s)}}, \quad (5.11)$$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \tilde{\boldsymbol{\mathcal{Y}}}[k] \times_1 \hat{\mathbf{u}}_1[k]^{\text{H}} \in \mathbb{C}^{1 \times M_y^{(s)} \times 2 \times L_s \times Q}, \quad (5.12)$$

we then continue this estimate subspace vector and compress process with the second mode,

$$\hat{\mathbf{u}}_2[k] = \text{S. T.}\left(\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(2)}\right) \in \mathbb{C}^{M_y^{(s)}}, \quad (5.13)$$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \tilde{\boldsymbol{\mathcal{Y}}}[k] \times_2 \hat{\mathbf{u}}_2[k]^{\text{H}} \in \mathbb{C}^{1 \times 1 \times 2 \times L_s \times Q}, \quad (5.14)$$

then the third mode,

$$\hat{\mathbf{u}}_3[k] = \text{S. T.}\left(\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(3)}\right) \in \mathbb{C}^2, \quad (5.15)$$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \tilde{\boldsymbol{\mathcal{Y}}}[k] \times_3 \hat{\mathbf{u}}_3[k]^{\text{H}} \in \mathbb{C}^{1 \times 1 \times 1 \times L_s \times Q}, \quad (5.16)$$

and finally the fourth mode,

$$\hat{\mathbf{u}}_4[k] = \text{S. T.}\left(\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(4)}\right) \in \mathbb{C}^{L_s}, \quad (5.17)$$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \tilde{\boldsymbol{\mathcal{Y}}}[k] \times_4 \hat{\mathbf{u}}_4[k]^{\text{H}} \in \mathbb{C}^{1 \times 1 \times 1 \times 1 \times Q}. \quad (5.18)$$

The auto-correlation vector is calculated by discarding the singleton dimensions of

[1] Each mode's corresponding eigenvalue is also updated but is omitted here for brevity.

[2] Subspace tracking algorithms can be found in Appendix C.

eq. (5.18) (`squeeze` in MATLAB) by applying a vectorization operator, completing the correlation, and taking the absolute value

$$\mathbf{q}[k] = \left| \mathbf{V}\boldsymbol{\Sigma} \cdot \mathrm{vec}\left\{ \tilde{\boldsymbol{\mathcal{Y}}}[k] \right\} \right| \in \mathbb{R}^{Q}, \tag{5.19}$$

followed by cubic spline interpolation to estimate the LOS time-delay.

The sequentially truncated HOE+FBA+SPS update algorithm at each k-th iteration is shown in Algorithm 8 and the whole algorithm is shown in Algorithm 9. The block diagram is shown in Figure 5.1.

Algorithm 8 Sequentially Truncated HOE+FBA+SPS update

Input: $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M_x \times M_y \times Q}$, number of subarrays L_s, previous subspace vectors $\hat{\mathbf{u}}_1[k-1] \in \mathbb{C}^{M_x^{(s)}}$, $\hat{\mathbf{u}}_2[k-1] \in \mathbb{C}^{M_y^{(s)}}$, $\hat{\mathbf{u}}_3[k-1] \in \mathbb{C}^2$, and $\hat{\mathbf{u}}_4[k-1] \in \mathbb{C}^{L_s}$, previous eigenvalues vector $\hat{\boldsymbol{\lambda}}[k-1] = \left[\hat{\lambda}_1[k-1], \hat{\lambda}_2[k-1], \hat{\lambda}_3[k-1], \hat{\lambda}_4[k-1]\right]^{\mathrm{T}} \in \mathbb{R}^4$, and forgetting factor $0 < \beta \leq 1$.

Output: estimated time-delay $\tau_{\mathrm{LOS}}[k]$, updated subspace vectors $\hat{\mathbf{u}}_1[k] \in \mathbb{C}^{M_x^{(s)}}$, $\hat{\mathbf{u}}_2[k] \in \mathbb{C}^{M_y^{(s)}}$, $\hat{\mathbf{u}}_3[k] \in \mathbb{C}^2$, and $\hat{\mathbf{u}}_4[k] \in \mathbb{C}^{L_s}$, and updated eigenvalues vector $\hat{\boldsymbol{\lambda}}[k]$.

1: Apply expanded FBA and SPS to $\boldsymbol{\mathcal{Y}}[k]$
2: Store pre-processed tensor

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \boldsymbol{\mathcal{Y}}[k]_{\text{EFBA+ESPS}}.$$

3: **for** $n = 1$ to 4 **do**
4: **function** SUBSPACE TRACKING UPDATE($\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(n)}$, $\hat{\mathbf{u}}_n[k-1]$, $\hat{\lambda}_n[k-1]$, β)
5: **return** $\hat{\mathbf{u}}_n[k]$, $\hat{\lambda}_n[k]$
6: **end function**
7: Compress tensor data

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] \leftarrow \tilde{\boldsymbol{\mathcal{Y}}}[k] \times_n \hat{\mathbf{u}}_n[k]^{\mathrm{H}}.$$

8: **end for**
9: Calculate auto-correlation vector

$$\mathbf{q}[k] = \left| \mathbf{V}\boldsymbol{\Sigma} \cdot \mathrm{vec}\left\{ \tilde{\boldsymbol{\mathcal{Y}}}[k] \right\} \right| \in \mathbb{R}^{Q},$$

and apply cubic spline interpolation to estimate time-delay.

Algorithm 9 Sequentially Truncated HOE+FBA+SPS

Input: $\boldsymbol{\mathcal{Y}} = \big[\boldsymbol{\mathcal{Y}}[1] \sqcup_4 \ldots \sqcup_4 \boldsymbol{\mathcal{Y}}[K]\big]$, number of subarrays L_s, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delays $\boldsymbol{\tau}_{\text{LOS}} = \big[\tau_{\text{LOS}}[1], \ldots, \tau_{\text{LOS}}[K]\big] \in \mathbb{R}^K$.

1: Initialize subspace vectors

$$\hat{\mathbf{u}}_1[0] = \mathbf{e}_{M_x^{(s)}}, \qquad \hat{\mathbf{u}}_2[0] = \mathbf{e}_{M_y^{(s)}}, \qquad \hat{\mathbf{u}}_3[0] = \mathbf{e}_2, \qquad \hat{\mathbf{u}}_4[0] = \mathbf{e}_{L_s}.$$

2: Initialize eigenvalues

$$\hat{\boldsymbol{\lambda}}[0] = \mathbf{1}_4.$$

3: **for** $k = 1$ to K **do**
4: **function** ST-HOE+FBA+SPS UPD.($\boldsymbol{\mathcal{Y}}[k]$, $\hat{\mathbf{u}}_n[k-1]$ for $n = 1, \ldots, 4$, $\hat{\boldsymbol{\lambda}}[k-1]$)
5: **return** $\tau_{\text{LOS}}[k]$, $\hat{\mathbf{u}}_n[k]$ for $n = 1, \ldots, 4$, $\hat{\boldsymbol{\lambda}}[k]$
6: **end function**
7: **end for**

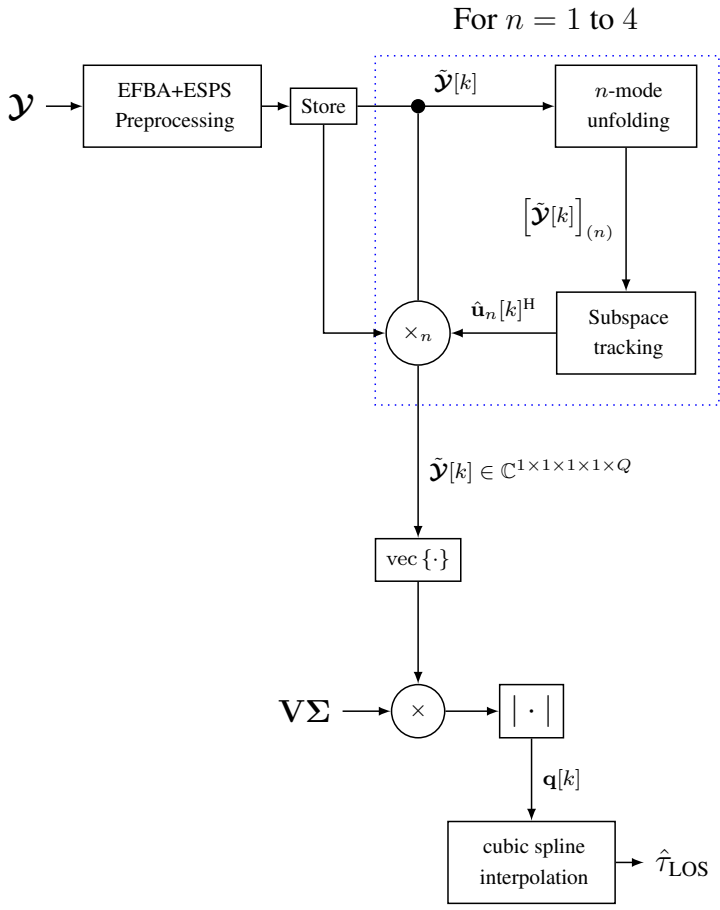

Figure 5.1: ST-ST-HOE block diagram.

5.3 Proposed Sequentially Truncated Core Tensor Estimation Dual Eigenvector Matrix CPD-GEVD for Time-delay Estimation (ST-CPD)

The proposed approach uses a sequential truncation strategy [18] to estimate the core tensor used in dual eigenvector CPD-GEVD at each epoch k.

Instead of applying MLSVD to each unfolding, subspace tracking is used to estimate the subspace, followed by the n-mode product of the tensor data and the conjugate transpose of the estimated subspace matrix.

We initialize this process by copying $\boldsymbol{\mathcal{Y}}[k]$ in $\hat{\boldsymbol{\mathcal{S}}}[k]$. Starting with the third mode, we estimate the subspace using a subspace tracking.[3] The estimated subspace is used to compress the data in $\hat{\boldsymbol{\mathcal{S}}}[k]$

$$\hat{\boldsymbol{\mathcal{S}}}[k] \leftarrow \boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M_x \times M_y \times Q}, \tag{5.20}$$

$$\hat{\mathbf{U}}_3[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(3)}\right) \in \mathbb{C}^{M_x \times L}, \tag{5.21}$$

$$\hat{\boldsymbol{\mathcal{S}}}[k] \leftarrow \hat{\boldsymbol{\mathcal{S}}}[k] \times_3 \hat{\mathbf{U}}_3[k]^{\mathrm{H}} \in \mathbb{C}^{M_x \times M_y \times L}, \tag{5.22}$$

we then continue the process of subspace and core estimation,

$$\hat{\mathbf{U}}_2[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(2)}\right) \in \mathbb{C}^{M_y \times L}, \tag{5.23}$$

$$\hat{\boldsymbol{\mathcal{S}}}[k] \leftarrow \hat{\boldsymbol{\mathcal{S}}}[k] \times_2 \hat{\mathbf{U}}_2[k]^{\mathrm{H}} \in \mathbb{C}^{M_x \times L \times L}, \tag{5.24}$$

and finally the first mode,

$$\hat{\mathbf{U}}_1[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(1)}\right) \in \mathbb{C}^{M_x \times L}, \tag{5.25}$$

$$\hat{\boldsymbol{\mathcal{S}}}[k] \leftarrow \hat{\boldsymbol{\mathcal{S}}}[k] \times_1 \hat{\mathbf{U}}_1[k]^{\mathrm{H}} \in \mathbb{C}^{L \times L \times L}, \tag{5.26}$$

and the core tensor is estimated.

As seen in Section 2.3.2, applying the generalized eigenvalue decomposition to the matrix pencil formed by the first two slices of the estimated core tensor allows estimation of the transformation matrices of the corresponding subspaces.

The correlated code factor matrix is estimated

$$\hat{\tilde{\mathbf{C}}}[k] = \hat{\mathbf{U}}_3[k] \cdot \hat{\mathbf{T}}_3[k], \tag{5.27}$$

[3] Each mode's corresponding covariance matrix is also updated but is omitted here for brevity.

and the auto-correlation vector is estimated

$$\mathbf{q}[k] = \left| \mathbf{V}\mathbf{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}[k](\cdot, s_{\text{LOS}}) \right|, \tag{5.28}$$

followed by cubic spline interpolation to estimate the time-delay.

The sequentially truncated core tensor estimation dual eigenvector matrix CPD-GEVD for time-delay estimation update algorithm is shown in Algorithm 10 and the whole algorithm is shown in Algorithm 11. The block diagram is shown in Figure 5.2.

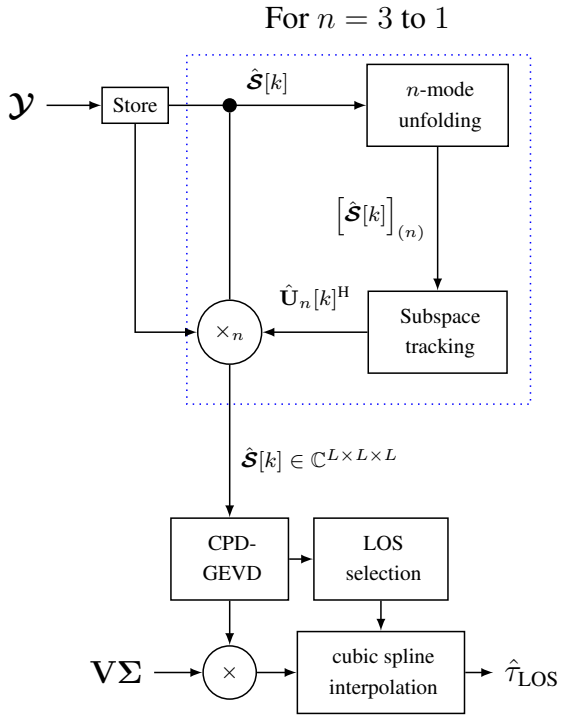

Figure 5.2: ST-CPD block diagram.

5.4 Proposed Sequentially Truncated TeTraKron and Tensor ESPRIT (ST-TTK)

The proposed approach utilizes a truncation strategy to estimate the improved signal subspace [86] without using the core tensor by using the estimated subspaces as projection matrices [79, 81].

5.4.1 Tensor subspace tracking via Kronecker structured projections

Consider a linear mixture of D sources superimposed by additive noise

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S}^{\text{T}} + \mathbf{N} \in \mathbb{C}^{M \times N}. \tag{5.29}$$

Algorithm 10 Sequential Truncated CPD-GEVD2 update

Input: $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M_x \times M_y \times Q}$, previous subspace matrices $\hat{\mathbf{U}}_1[k-1] \in \mathbb{C}^{M_x \times L}$, $\hat{\mathbf{U}}_2[k-1] \in \mathbb{C}^{M_y \times L}$, and $\hat{\mathbf{U}}_3[k-1] \in \mathbb{C}^{Q \times L}$, previous covariance matrices $\hat{\mathbf{R}}_1[k-1] \in \mathbb{C}^{L \times L}$, $\hat{\mathbf{R}}_2[k-1] \in \mathbb{C}^{L \times L}$, and $\hat{\mathbf{R}}_3[k-1] \in \mathbb{C}^{L \times L}$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delay $\tau_{\text{LOS}}[k]$, updated subspace matrices $\hat{\mathbf{U}}_1[k] \in \mathbb{C}^{M_x \times L}$, $\hat{\mathbf{U}}_2[k] \in \mathbb{C}^{M_y \times L}$, and $\hat{\mathbf{U}}_3[k] \in \mathbb{C}^{Q \times L}$, and updated covariance matrices $\hat{\mathbf{R}}_1[k]$, $\hat{\mathbf{R}}_2[k]$, and $\hat{\mathbf{R}}_3[k]$.

1: Store tensor

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \boldsymbol{\mathcal{Y}}[k].$$

2: **for** $n = 3$ to 1 **do**
3: **function** SUBSPACE TRACKING UPDATE($\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(n)}$, $\hat{\mathbf{U}}_n[k-1]$, $\hat{\mathbf{R}}_n[k-1]$, β)
4: **return** $\hat{\mathbf{U}}_n[k]$, $\hat{\mathbf{R}}_n[k]$.
5: **end function**
6: Compress core tensor

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \hat{\boldsymbol{\mathcal{S}}}[k] \times_n \hat{\mathbf{U}}_n[k]^{\text{H}}.$$

7: **end for**
8: Apply GEVD to the matrix pencil formed by the first two slices of $\hat{\boldsymbol{\mathcal{S}}}[k]$, $\mathbf{S}_1 = \hat{\boldsymbol{\mathcal{S}}}[k](\cdot,\cdot,1)$ and $\mathbf{S}_2 = \hat{\boldsymbol{\mathcal{S}}}[k](\cdot,\cdot,2)$, to calculate left- and right-hand generalized eigenvector matrices

$$\mathbf{L}^{\text{H}} \cdot \mathbf{S}_1 = \mathbf{D} \cdot \mathbf{L}^{\text{H}} \cdot \mathbf{S}_2, \qquad \mathbf{S}_1 \cdot \mathbf{R} = \mathbf{S}_2 \cdot \mathbf{R} \cdot \mathbf{D}.$$

9: Estimate transformation matrices

$$\hat{\mathbf{T}}_1 = \mathbf{L}^{-\text{H}}, \qquad \hat{\mathbf{T}}_2 = \mathbf{R}^{-\text{T}},$$
$$\hat{\mathbf{T}}_3 = \left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(3)} \left(\left(\hat{\mathbf{T}}_1 \diamond \hat{\mathbf{T}}_2\right)^{\text{T}}\right)^{\dagger}.$$

10: Estimate code factor matrix

$$\hat{\tilde{\mathbf{C}}}[k] = \hat{\mathbf{U}}_3[k] \cdot \hat{\mathbf{T}}_3.$$

11: Complete the auto-correlation

$$\mathbf{q}[k] = \left|\mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}[k](\cdot, s_{\text{LOS}})\right|,$$

and apply cubic spline interpolation to estimate time-delay.

Algorithm 11 Sequential Truncated CPD-GEVD2

Input: $\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{\mathcal{Y}}[1] \sqcup_4 \ldots \sqcup_4 \boldsymbol{\mathcal{Y}}[K]\right]$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delays $\boldsymbol{\tau}_{\text{LOS}} = \left[\tau_{\text{LOS}}[1], \ldots, \tau_{\text{LOS}}[K]\right] \in \mathbb{R}^K$.

1: Initialize subspace matrices

$$\hat{\mathbf{U}}_1[0] = \mathbf{I}_{M_x \times L}, \qquad \hat{\mathbf{U}}_2[0] = \mathbf{I}_{M_y \times L}, \qquad \hat{\mathbf{U}}_3[0] = \mathbf{I}_{Q \times L}.$$

2: Initialize covariance matrices

$$\hat{\mathbf{R}}_1[0] = \mathbf{I}_L, \qquad \hat{\mathbf{R}}_2[0] = \mathbf{I}_L, \qquad \hat{\mathbf{R}}_3[0] = \mathbf{I}_L.$$

3: **for** $k = 1$ to K **do**
4: **function** ST-CPD-GEVD2 UPD.($\boldsymbol{\mathcal{Y}}[k]$, $\hat{\mathbf{U}}_n[k-1]$ and $\hat{\mathbf{R}}_n[k-1]$ for $n = 1, 2, 3$)
5: **return** $\tau_{\text{LOS}}[k]$, $\hat{\mathbf{U}}_n[k]$ and $\hat{\mathbf{R}}_n[k]$ for $n = 1, 2, 3$.
6: **end function**
7: **end for**

$\mathbf{X} \in \mathbb{C}^{M \times N}$ is the observations matrix from M channels during N time samples. $\mathbf{A} \in \mathbb{C}^{M \times D}$ is the (unknown) mixing matrix, $\mathbf{S} \in \mathbb{C}^{N \times D}$ contains the (unknown) source symbols, and $\mathbf{N} \in \mathbb{C}^{M \times N}$ is additive noise.

$\mathbf{X}$ can be approximated by a low-rank approximation (LRA) via SVD

$$\mathbf{X} \approx \hat{\mathbf{U}}_S \cdot \hat{\boldsymbol{\Sigma}}_S \cdot \hat{\mathbf{V}}_S^{\text{H}} \tag{5.30}$$

with $\hat{\mathbf{U}}_S \in \mathbb{C}^{M \times D}$, $\hat{\boldsymbol{\Sigma}}_S \in \mathbb{C}^{D \times D}$, and $\hat{\mathbf{V}}_S \in \mathbb{C}^{N \times D}$.

Since $\operatorname{span}\left(\hat{\mathbf{U}}_S\right) = \operatorname{span}(\mathbf{A})$ there exists non-singular matrix $\mathbf{T} \in \mathbb{C}^{D \times D}$ such that $\mathbf{A} = \hat{\mathbf{U}}_S \cdot \mathbf{T}$ and $\hat{\mathbf{U}}_S$ can be used as a signal subspace estimate.

Arranging the elements of $\mathbf{X} \in \mathbb{C}^{M \times N}$ into a $R+1$ order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \times \ldots \times M_R \times N}$, with $M = \prod_r M_r$,

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}} \times_{R+1} \mathbf{S} + \boldsymbol{\mathcal{N}}, \tag{5.31}$$

where $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times \ldots \times M_R \times D}$ is the mixing tensor and $\boldsymbol{\mathcal{N}} \in \mathbb{C}^{M_1 \times \ldots \times M_R \times N}$ is the noise tensor. Equation (5.29) and eq. (5.31) are linked by the following relations

$$[\boldsymbol{\mathcal{X}}]_{(R+1)}^{\text{T}} = \mathbf{X}, \tag{5.32}$$

$$[\boldsymbol{\mathcal{A}}]_{(R+1)}^{\text{T}} = \mathbf{A}, \tag{5.33}$$

$$[\boldsymbol{\mathcal{N}}]_{(R+1)}^{\text{T}} = \mathbf{N}. \tag{5.34}$$

An improved signal subspace estimate can be computed by finding $\hat{\boldsymbol{\mathcal{U}}}$ such that $\boldsymbol{\mathcal{A}} =$

$\hat{\boldsymbol{\mathcal{U}}} \cdot \mathbf{T}$. This can be achieved via the LMLRA

$$\boldsymbol{\mathcal{X}} \approx \hat{\boldsymbol{\mathcal{S}}} \times_{r=1}^{R+1} \hat{\mathbf{U}}_r = \hat{\boldsymbol{\mathcal{U}}} \times_{R+1} \hat{\mathbf{U}}_{R+1}, \tag{5.35}$$

thus, $\hat{\boldsymbol{\mathcal{U}}} = \hat{\boldsymbol{\mathcal{S}}} \times_{r=1}^{R} \hat{\mathbf{U}}_r$.

Equation (5.35) suggests the core tensor is necessary to track the improved signal subspace estimate. Consider the definition of the core tensor

$$\hat{\boldsymbol{\mathcal{S}}} = \boldsymbol{\mathcal{X}} \times_{n=1}^{R+1} \hat{\mathbf{U}}_r^{\mathrm{H}}, \tag{5.36}$$

and replacing it in eq. (5.35) we have

$$\begin{aligned} \hat{\boldsymbol{\mathcal{U}}} &= \boldsymbol{\mathcal{X}} \times_{r=1}^{R+1} \hat{\mathbf{U}}_r^{\mathrm{H}} \times_{r=1}^{R} \hat{\mathbf{U}}_r, & (5.37)\\ &= \boldsymbol{\mathcal{X}} \times_{r=1}^{R} \hat{\mathbf{U}}_r \hat{\mathbf{U}}_r^{\mathrm{H}} \times_{R+1} \hat{\mathbf{U}}_{R+1}^{\mathrm{H}}, & (5.38) \end{aligned}$$

We can interpret $\hat{\mathbf{P}}_r = \hat{\mathbf{U}}_r \hat{\mathbf{U}}_r^{\mathrm{H}}$ as a projection matrix which projects onto the space spanned by the r-mode vectors [87] and we rewrite the above as

$$\hat{\boldsymbol{\mathcal{U}}} = \boldsymbol{\mathcal{X}} \times_{r=1}^{R} \hat{\mathbf{P}}_r \times_{R+1} \hat{\mathbf{U}}_{R+1}^{\mathrm{H}}. \tag{5.39}$$

We can exploit the relation $[\boldsymbol{\mathcal{X}}]_{(R+1)}^{\mathrm{T}} = \mathbf{X}$ by considering the transpose of the $R+1$ mode unfolding of $\hat{\boldsymbol{\mathcal{U}}}$

$$\begin{aligned} \left[\hat{\boldsymbol{\mathcal{U}}}\right]_{(R+1)}^{\mathrm{T}} &= \left[\boldsymbol{\mathcal{X}} \times_{r=1}^{R} \hat{\mathbf{P}}_r \times_{R+1} \hat{\mathbf{U}}_{R+1}^{\mathrm{H}}\right]_{(R+1)}^{\mathrm{T}}, & (5.40)\\ &= \left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) [\boldsymbol{\mathcal{X}}]_{(R+1)}^{\mathrm{T}} \left(\hat{\mathbf{U}}_{R+1}^{\mathrm{H}}\right)^{\mathrm{T}}, & (5.41)\\ &= \left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) \mathbf{X} \cdot \hat{\mathbf{U}}_{R+1}^{*}, & (5.42)\\ &= \left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) \hat{\mathbf{U}}_S \cdot \hat{\boldsymbol{\Sigma}}_S \cdot \hat{\mathbf{V}}_S^{\mathrm{H}} \cdot \hat{\mathbf{U}}_{R+1}^{*}, & (5.43) \end{aligned}$$

and since

$$\begin{aligned} [\boldsymbol{\mathcal{X}}]_{(R+1)}^{\mathrm{T}} &= \left[\hat{\boldsymbol{\mathcal{U}}} \times_{R+1} \hat{\mathbf{U}}_{R+1}\right]_{(R+1)}^{\mathrm{T}} = \mathbf{X}, & (5.44)\\ &= \left[\hat{\boldsymbol{\mathcal{U}}}\right]_{(R+1)}^{\mathrm{T}} \hat{\mathbf{U}}_{R+1}^{\mathrm{T}} = \hat{\mathbf{U}}_S \cdot \hat{\boldsymbol{\Sigma}}_S \cdot \hat{\mathbf{V}}_S^{\mathrm{H}}, & (5.45) \end{aligned}$$

therefore $\hat{\mathbf{U}}_{R+1}^{*} = \hat{\mathbf{V}}_S$ and

$$\left[\hat{\boldsymbol{\mathcal{U}}}\right]_{(R+1)}^{\mathrm{T}} = \left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) \hat{\mathbf{U}}_S \cdot \hat{\boldsymbol{\Sigma}}_S. \tag{5.46}$$

Since $\hat{\mathbf{\Sigma}}_S$ is merely a scaling factor, we can forego it [81] and

$$\left[\hat{\boldsymbol{\mathcal{U}}}\right]^{\mathrm{T}}_{(R+1)} = \left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) \hat{\mathbf{U}}_S, \tag{5.47}$$

is the improved signal subspace estimate calculated using Kronecker structured projections instead of estimating the core tensor, reducing computational complexity.

Further reduction in complexity is achieved considering the fact that

$$\left(\bigotimes_{r=1}^{R} \hat{\mathbf{P}}_r\right) = \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r \cdot \hat{\mathbf{U}}_r^{\mathrm{H}}\right), \tag{5.48}$$

$$= \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r\right) \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r^{\mathrm{H}}\right), \tag{5.49}$$

$$= \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r\right) \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r\right)^{\mathrm{H}}, \tag{5.50}$$

thus $(\otimes_{r=1}^{R} \hat{\mathbf{U}}_r)$ is only calculated once and the improved signal subspace estimate is

$$\left[\hat{\boldsymbol{\mathcal{U}}}\right]^{\mathrm{T}}_{(R+1)} = \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r\right) \left(\bigotimes_{r=1}^{R} \hat{\mathbf{U}}_r\right)^{\mathrm{H}} \hat{\mathbf{U}}_S. \tag{5.51}$$

5.4.2 Sequentially truncated TeTraKron

The sequentially truncated TeTraKron algorithm uses the sequential truncation strategy presented previously to reduce computational complexity of subspace tracking by compressing the amount of data used by subspace tracking algorithm and estimating the improved signal subspace estimate.

For the data model in eq. (5.6), we have $R = 2$. We proceed in a similar fashion to sequentially truncated core tensor estimation in Section 5.3.

We begin by storing the original tensor data in $\boldsymbol{\mathcal{S}}[k]$. Starting with the third mode, we estimate the subspace using subspace tracking. The estimate subspace is then used to compress the data in $\hat{\boldsymbol{\mathcal{S}}}[k]$

$$\tilde{\boldsymbol{\mathcal{Y}}}[k] = \boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M_x \times M_y \times Q}, \tag{5.52}$$

$$\hat{\mathbf{U}}_3[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(3)}\right) \in \mathbb{C}^{Q \times L}, \tag{5.53}$$

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \hat{\boldsymbol{\mathcal{S}}}[k] \times_3 \hat{\mathbf{U}}_3[k]^{\mathrm{H}} \in \mathbb{C}^{M_x \times M_y \times L}, \tag{5.54}$$

we continue with the second mode, estimating the subspace and compressing the data,

$$\hat{\mathbf{U}}_2[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(2)}\right) \in \mathbb{C}^{M_y \times L}, \tag{5.55}$$

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \hat{\boldsymbol{\mathcal{S}}}[k] \times_2 \hat{\mathbf{U}}_2[k]^{\mathrm{H}} \in \mathbb{C}^{M_x \times L \times L}, \tag{5.56}$$

and finally estimate the subspace of the first mode,

$$\hat{\mathbf{U}}_1[k] = \mathrm{S.\,T.}\left(\left[\hat{\boldsymbol{\mathcal{S}}}[k]\right]_{(2)}\right) \in \mathbb{C}^{M_x \times L}. \tag{5.57}$$

To calculate the improved signal subspace estimate, we can take advantage of the fact that

$$[\boldsymbol{\mathcal{Y}}[k]]_{(3)}^{\mathrm{T}} = \hat{\mathbf{U}}_S[k] \cdot \hat{\boldsymbol{\Sigma}}_S[k] \cdot \hat{\mathbf{V}}_S[k]^{\mathrm{H}}, \tag{5.58}$$

and $\hat{\mathbf{U}}_3[k]^* = \hat{\mathbf{V}}_S[k]$, thus

$$[\boldsymbol{\mathcal{Y}}[k]]_{(3)}^{\mathrm{T}} \cdot \hat{\mathbf{U}}_3[k]^* = \hat{\mathbf{U}}_S[k] \cdot \hat{\boldsymbol{\Sigma}}_S[k] \cdot \hat{\mathbf{V}}_S[k]^{\mathrm{H}} \cdot \hat{\mathbf{V}}_S[k], \tag{5.59}$$

$$= \hat{\mathbf{U}}_S[k] \cdot \hat{\boldsymbol{\Sigma}}_S[k], \tag{5.60}$$

and we have an improved signal subspace estimate similar to eq. (5.46) without computing $\hat{\mathbf{U}}_S[k]$,

$$\left[\hat{\boldsymbol{\mathcal{U}}}\right]_{(3)}^{\mathrm{T}} = \left(\hat{\mathbf{U}}_1[k] \otimes \hat{\mathbf{U}}_2[k]\right)\left(\hat{\mathbf{U}}_1[k] \otimes \hat{\mathbf{U}}_2[k]\right)^{\mathrm{H}} [\boldsymbol{\mathcal{Y}}[k]]_{(3)}^{\mathrm{T}} \cdot \hat{\mathbf{U}}_3[k]^* \in \mathbb{C}^{M \times L}. \tag{5.61}$$

The improved signal subspace estimate is used by a tensor ESPRIT [86] to estimate the spatial frequencies of $\mathbf{A}_x[k]$ and $\mathbf{A}_y[k]$.

Rebuilding $\mathbf{A}_x[k]$ and $\mathbf{A}_y[k]$, we estimate the correlated code factor matrix

$$\hat{\tilde{\mathbf{C}}}[k] = [\boldsymbol{\mathcal{Y}}[k]]_{(3)} \left(\left(\hat{\mathbf{A}}_x[k] \diamond \hat{\mathbf{A}}_y[k]\right)^{\mathrm{T}}\right)^{\dagger}. \tag{5.62}$$

The auto-correlation vector is then calculated

$$\mathbf{q}[k] = \left|\mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}[k](\cdot, s_{\mathrm{LOS}})\right|, \tag{5.63}$$

followed by cubic spline interpolation to estimate the time-delay.

The sequentially truncated TeTraKron and tensor ESPRIT for time-delay estimation update algorithm is shown in Algorithm 12 and the whole algorithm is shown in Algorithm 13. The block diagram is shown in Figure 5.3.

Algorithm 12 Sequential Truncated TeTraKron Tensor ESPRIT update

Input: $\boldsymbol{\mathcal{Y}}[k] \in \mathbb{C}^{M_x \times M_y \times Q}$, previous subspace matrices $\hat{\mathbf{U}}_1[k-1] \in \mathbb{C}^{M_x \times L}$, $\hat{\mathbf{U}}_2[k-1] \in \mathbb{C}^{M_y \times L}$, and $\hat{\mathbf{U}}_3[k-1] \in \mathbb{C}^{Q \times L}$, previous covariance matrices $\hat{\mathbf{R}}_1[k-1] \in \mathbb{C}^{L \times L}$, $\hat{\mathbf{R}}_2[k-1] \in \mathbb{C}^{L \times L}$, and $\hat{\mathbf{R}}_3[k-1] \in \mathbb{C}^{L \times L}$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delay $\tau_{\text{LOS}}[k]$, updated subspace matrices $\hat{\mathbf{U}}_1[k] \in \mathbb{C}^{M_x \times L}$, $\hat{\mathbf{U}}_2[k] \in \mathbb{C}^{M_y \times L}$, and $\hat{\mathbf{U}}_3[k] \in \mathbb{C}^{Q \times L}$, and updated covariance matrices $\hat{\mathbf{R}}_1[k]$, $\hat{\mathbf{R}}_2[k]$, and $\hat{\mathbf{R}}_3[k]$.

Store original tensor

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \boldsymbol{\mathcal{Y}}[k].$$

for $n = 3$ to 2 **do**
  function SUBSPACE TRACKING UPDATE($\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(n)}$, $\hat{\mathbf{U}}_n[k-1]$, $\hat{\mathbf{R}}_n[k-1]$, β)
    return $\hat{\mathbf{U}}_n[k]$, $\hat{\mathbf{R}}_n[k]$
  end function
  Compress data

$$\hat{\boldsymbol{\mathcal{S}}}[k] = \hat{\boldsymbol{\mathcal{S}}}[k] \times_n \hat{\mathbf{U}}_n^{\mathrm{H}}.$$

end for
function SUBSPACE TRACKING UPDATE($\left[\tilde{\boldsymbol{\mathcal{Y}}}[k]\right]_{(1)}$, $\hat{\mathbf{U}}_1[k-1]$, $\hat{\mathbf{R}}_1[k-1]$, β)
  return $\hat{\mathbf{U}}_1[k]$, $\hat{\mathbf{R}}_1[k]$
end function
Calculate improved signal subspace estimate

$$\left[\hat{\boldsymbol{\mathcal{U}}}\right]_{(3)}^{\mathrm{T}} = \left(\hat{\mathbf{U}}_1[k] \otimes \hat{\mathbf{U}}_2[k]\right) \left(\hat{\mathbf{U}}_1[k] \otimes \hat{\mathbf{U}}_2[k]\right)^{\mathrm{H}} [\boldsymbol{\mathcal{Y}}[k]]_{(3)}^{\mathrm{T}} \cdot \hat{\mathbf{U}}_3[k]^* \in \mathbb{C}^{M \times L}.$$

Apply tensor ESPRIT to estimate the spatial frequencies of the array response matrices.
Rebuild array response matrices $\hat{\mathbf{A}}_x[k]$ and $\hat{\mathbf{A}}_y[k]$.
Estimate code factor matrix

$$\hat{\tilde{\mathbf{C}}}[k] = [\boldsymbol{\mathcal{Y}}[k]]_{(3)} \left(\left(\hat{\mathbf{A}}_x[k] \diamond \hat{\mathbf{A}}_y[k]\right)^{\mathrm{T}}\right)^{\dagger}.$$

Complete the auto-correlation

$$\mathbf{q}[k] = \left|\mathbf{V}\boldsymbol{\Sigma} \cdot \hat{\tilde{\mathbf{C}}}[k](\cdot, s_{\text{LOS}})\right|,$$

and apply cubic spline interpolation to estimate time-delay.

Algorithm 13 Sequential Truncated TeTraKron Tensor ESPRIT

Input: $\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{\mathcal{Y}}[1] \sqcup_4 \ldots \sqcup_4 \boldsymbol{\mathcal{Y}}[K]\right]$, and forgetting factor $0 < \beta \leq 1$.
Output: estimated time-delays $\boldsymbol{\tau}_{\text{LOS}} = \left[\tau_{\text{LOS}}[1], \ldots, \tau_{\text{LOS}}[K]\right] \in \mathbb{R}^K$.

1: Initialize subspace matrices

$$\hat{\mathbf{U}}_1[0] = \mathbf{I}_{M_x \times L}, \qquad \hat{\mathbf{U}}_2[0] = \mathbf{I}_{M_y \times L}, \qquad \hat{\mathbf{U}}_3[0] = \mathbf{I}_{Q \times L}.$$

2: Initialize covariance matrices

$$\hat{\mathbf{R}}_1[0] = \mathbf{I}_L, \qquad \hat{\mathbf{R}}_2[0] = \mathbf{I}_L, \qquad \hat{\mathbf{R}}_3[0] = \mathbf{I}_L.$$

3: **for** $k = 1$ to K **do**
4: **function** ST-TTK UPD.($\boldsymbol{\mathcal{Y}}[k]$, $\hat{\mathbf{U}}_n[k-1]$ and $\hat{\mathbf{R}}_n[k-1]$ for $n = 1, 2, 3$)
5: **return** $\tau_{\text{LOS}}[k]$, $\hat{\mathbf{U}}_n[k]$ and $\hat{\mathbf{R}}_n[k]$ for $n = 1, 2, 3$
6: **end function**
7: **end for**

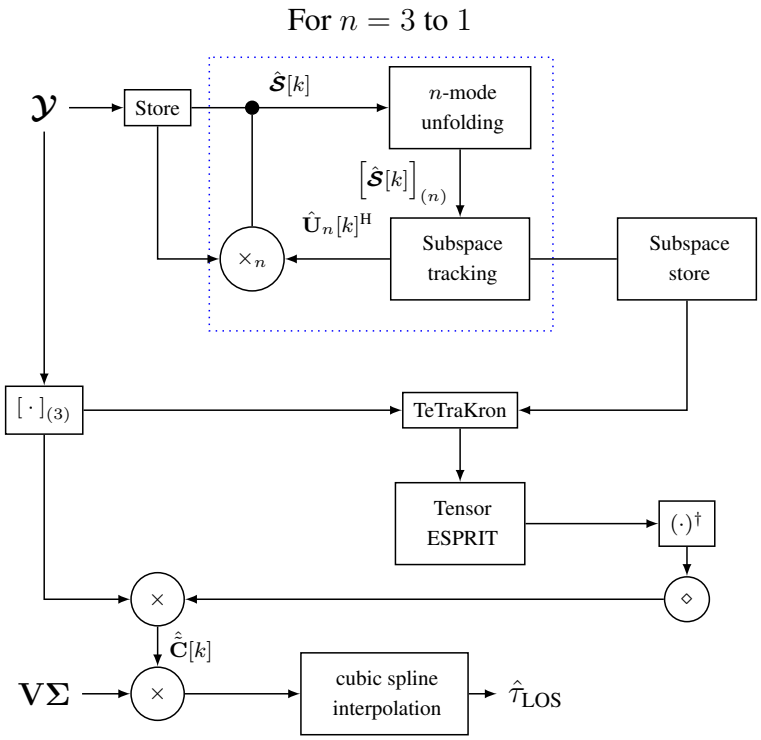

Figure 5.3: ST-TTK block diagram.

5.5 Numerical Simulation Results

In this section Monte Carlo simulation is performed to measure time-delay estimation tracking performance under a static and a dynamic scenario.

A benchmark method which estimates the code factor matrix using assuming known channel state information has been added for comparison. It simply estimates the PRS code

factor matrix directly,

$$\tilde{\mathbf{C}}[k]_{\text{Known}} = [\boldsymbol{\mathcal{Y}}[k]]_{(3)} \left((\mathbf{A}_x[k] \diamond \mathbf{A}_y[k])^{\mathrm{T}} \right)^{\dagger} \in \mathbb{C}^{Q \times L}, \tag{5.64}$$

followed by completing the correlation and applying cubic spline interpolation to estimate the time-delay.

5.5.1 Static scenario

In the static scenario, changes only occur in the phases of the complex amplitude matrix. Direction of arrival, time-delays, and SNR remain unchanged during the K epochs.

Simulation parameters are similar to previous chapters, the most immediate difference is the employment of a URA, thus there are two directions of arrival, an azimuth θ, and an elevation (from the zenith) ϕ. A 4×4 URA is employed and angle separation of $\pi/6$ is enforced between LOS and NLOS signal components.

For the ST-HOE technique expanded FBA and expanded SPS preprocessing was also employed. Expanded FBA is similar to ESPS [61], in which the increase number of samples are accumulated in a new dimension.

1000 Monte Carlo realizations were performed. Learning rate $\beta_{\text{iSVD}} = 0.85$, $\beta_{\text{PAST}} = 0.45$, $\beta_{\text{FAPI}} = 0.65$, and $\beta_{\text{GYAST}} = 0.85$ were used.

Results for ST-HOE are shown in Figure 5.4, for ST-CPD are shown in Figure 5.5, and for ST-TTK are shown in Figure 5.6. In all figures, results for filtering with a priori knowledge of the mixing matrices $\mathbf{A}_x[k]$ and $\mathbf{A}_y[k]$ is included as a baseline.

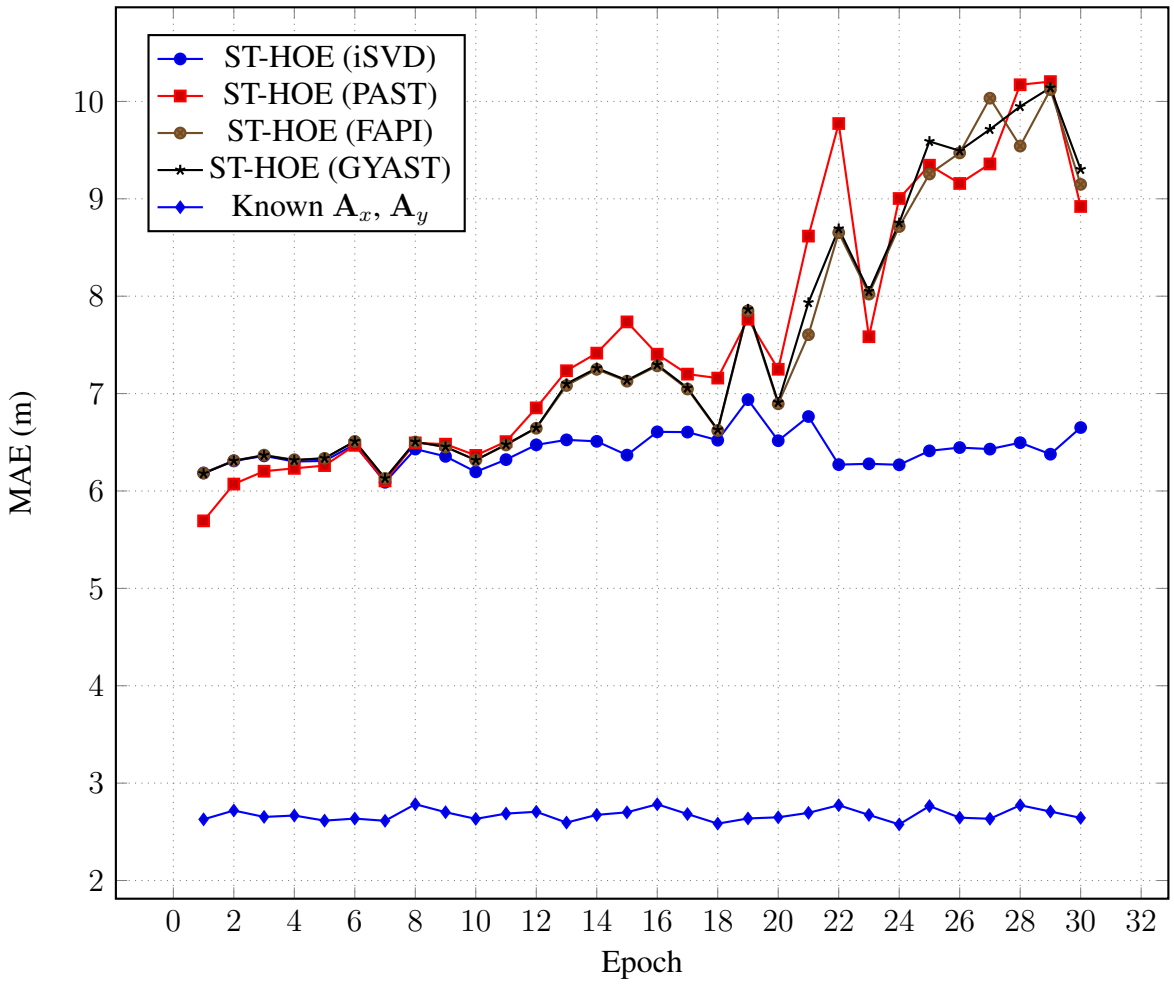

Figure 5.4: ST-HOE+EFBA+SPS time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

Only ST-HOE+EFBA+ESPS using the costly iterative SVD subspace tracking was able to achieve consistent results during all epochs. Using the other subspace tracking algorithms resulted in increased error as the number of epochs increased.

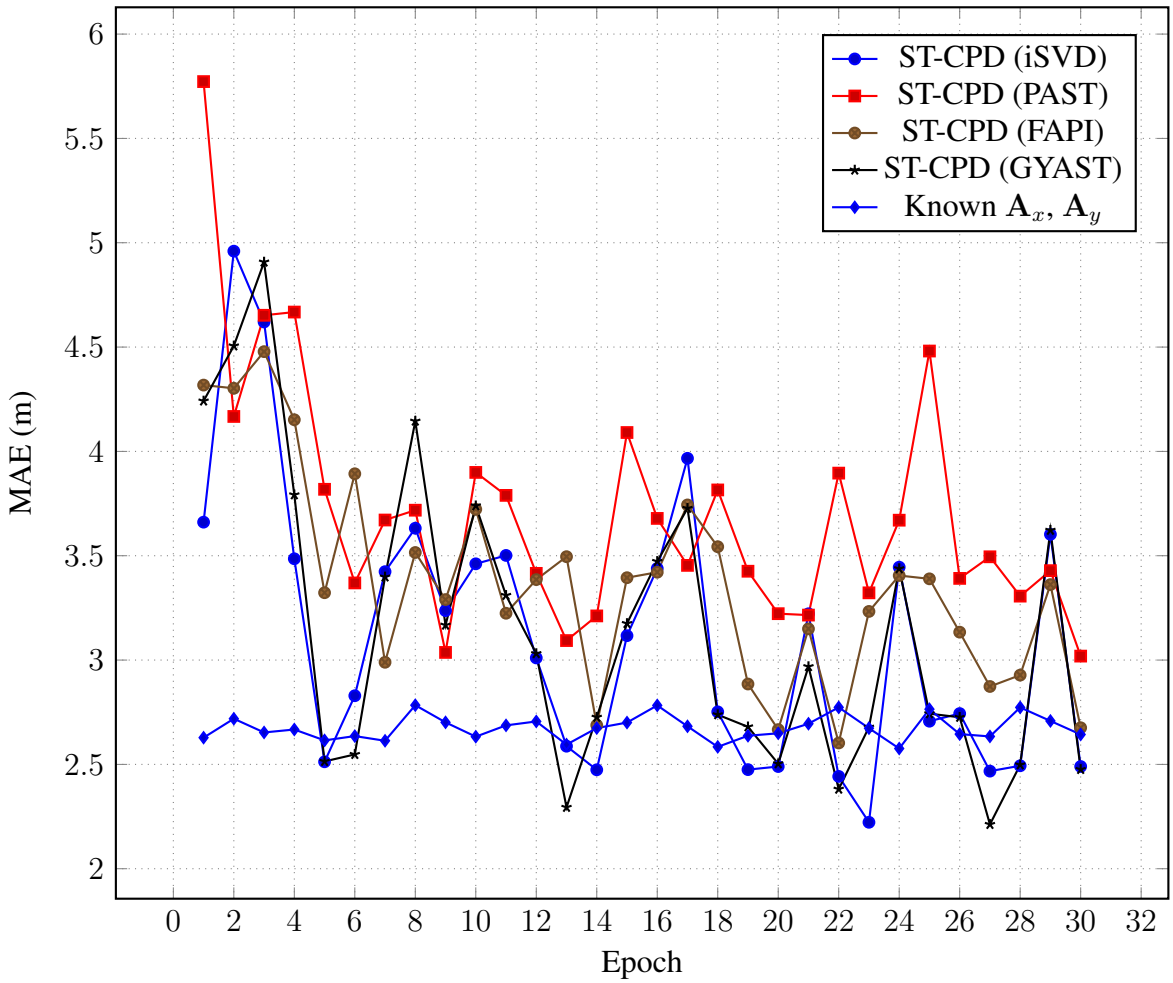

Figure 5.5: ST-CPD time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

ST-CPD results were closer to filtering with a priori knowledge and, in some cases, even surpassing it. Results are similar, with ST-CPD using PAST showing a small loss relative to using FAPI and GYAST. Both which tracked closely with using iterative SVD, with a slight advantage to using GYAST.

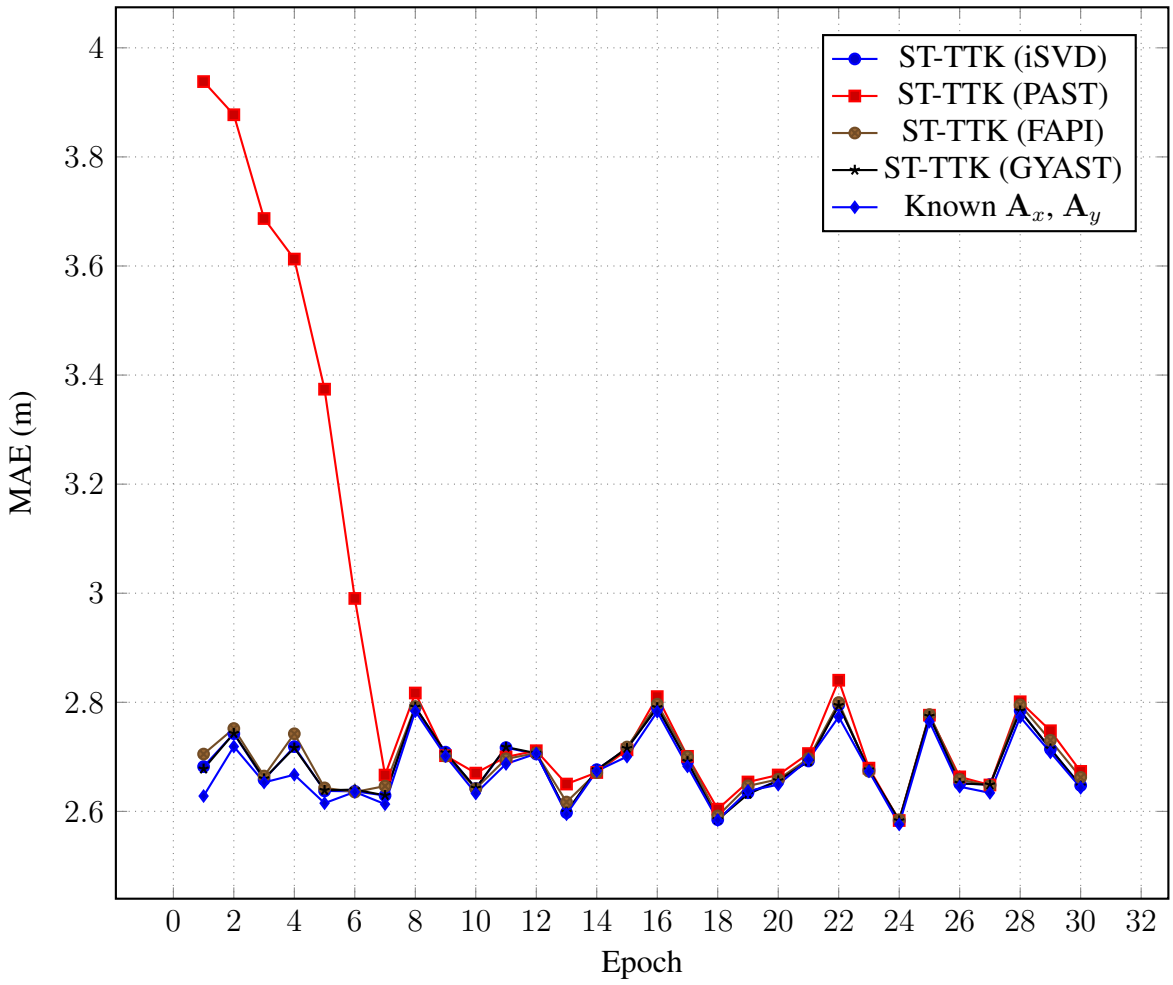

Figure 5.6: ST-TTK time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

All ST-TTK using different subspace tracking methods tracked closely with the baseline. ST-TTK using PAST took 7 epochs to reach the same performance as using the other subspace tracking methods.

These results are somewhat similar to previous results, in which the parametric approach (ST-TTK) has better performance than the algebraic approach (ST-CPD), which has better performance than the eigenfilter approach (ST-HOE).

5.5.2 Dynamic scenario

Under the dynamic scenario, elevation to the satellite decreases (the angle from the horizon increases), and the corresponding changes in directions of arrival, time-delays, and SNR are simulated as well.

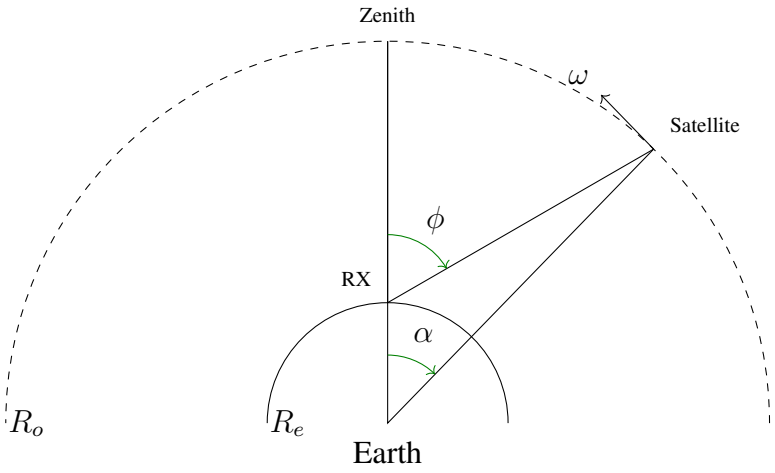

Figure 5.7: Illustration of dynamic simulation scenario.

For GPS, the maximum elevation possible is $\phi_{\text{max}} = \pi/3$ radians [41] or, equivalently, $\pi/6$ radians above the horizon. The angle α considered from the center of the Earth from the zenith has a maximum $\alpha_{\text{max}} = 0.7706$ rad, or $44.1513\,\text{deg}$.

Given a period of half a sidereal day, GPS satellites have an angular velocity of 0.1458×10^{-3} rad/s. For an Earth radius $R_e = 6.37 \times 10^6$ m and orbit radius $R_o = 20.2 \times 10^6$ m, this results in a maximum distance d_{max} to the receiver of 16.247×10^6 m.

The simulation parameters are modeled in the following fashion

$$\omega = -\frac{2\pi}{86164/2} = -0.1458 \times 10^{-3} \text{ rad/s}, \tag{5.65}$$

$$\alpha[k] = \alpha_{\text{max}} + (k-1)\omega \text{ rad}, \tag{5.66}$$

$$d[k] = R_e^2 + R_o^2 - 2 \cdot R_e \cdot R_o \cos\alpha[k] \text{ m}, \tag{5.67}$$

$$\tau_{\text{LOS}}[k] = \text{mod}(d[k]/c, T_c) - T_c, \tag{5.68}$$

$$P_{\text{LOS}}[k] = |\gamma_{\text{LOS}}| \left(\frac{d_{\text{max}}}{d[k]}\right)^2, \tag{5.69}$$

$$P_{\text{NLOS}}[k] = |\gamma_{\text{NLOS}}| \left(\frac{d_{\text{max}}}{d[k]}\right)^2. \tag{5.70}$$

Because the angular velocity is low most parameter change very smoothly, however even the slight changes in timing cause the delay to change swiftly as each entire chip period is shifted. The resulting time-delay is shown in Figure 5.8.

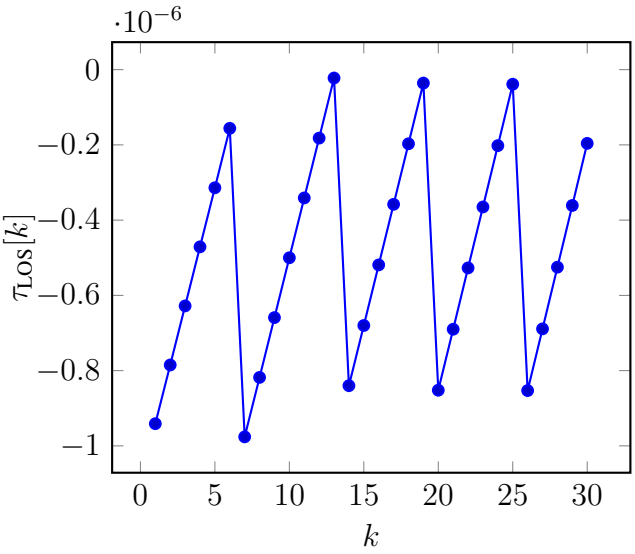

Figure 5.8: Time-delay for dynamic scenario.

For the ST-HOE technique expanded FBA and expanded SPS preprocessing was also employed.

1000 Monte Carlo realizations were performed. A forgetting factor $\beta = 0.05$ was used.

The results of this simulation can be seen in Figures 5.9 to 5.11, which displays the mean absolute error (MAE) in meters at each epoch.

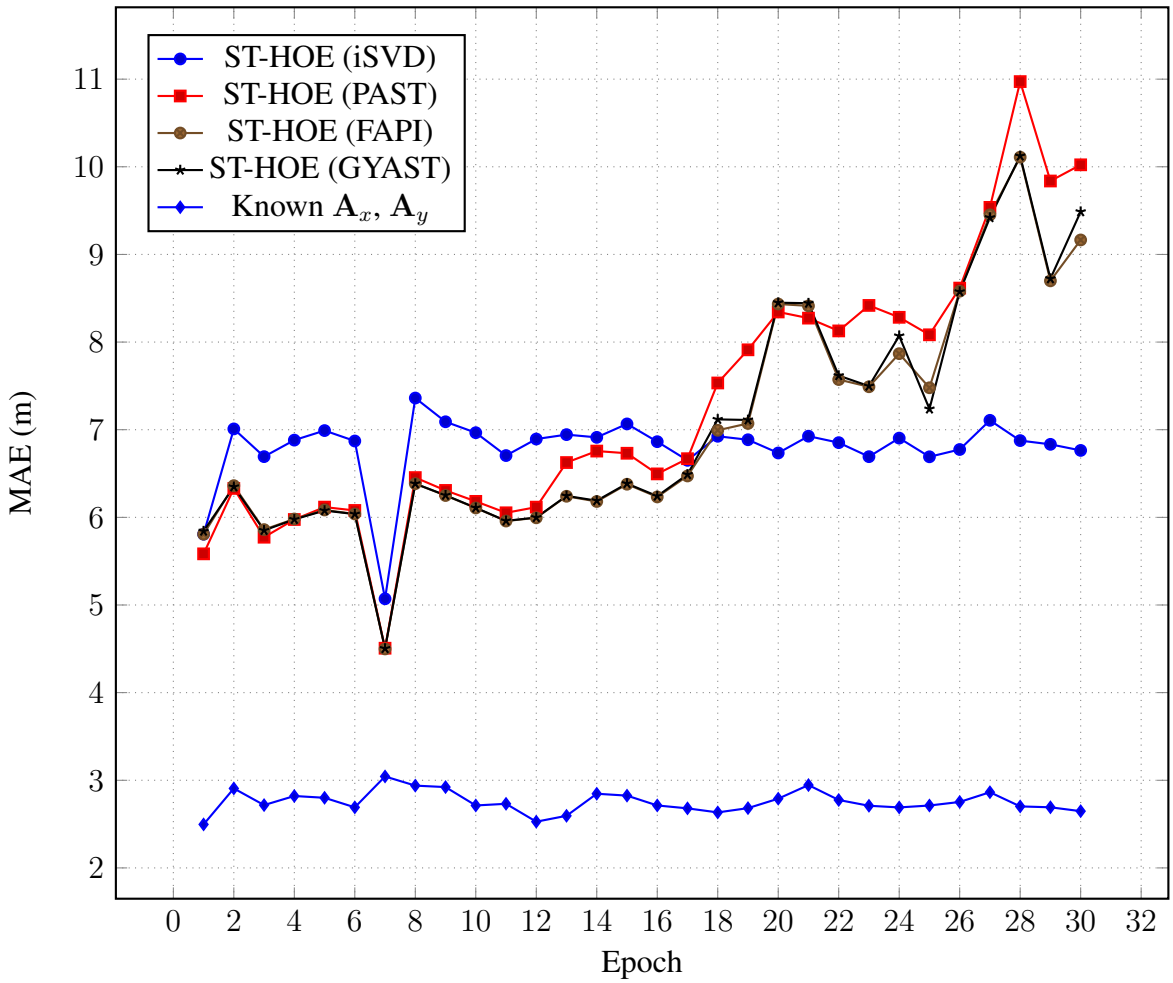

Figure 5.9: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

ST-HOE performance was consistent for over half the observation period then only results were more consistent than in the static scenario, where only iterative SVD susbpace tracking maintains consistent results. These result suggests sequential truncation does not benefit HOE unless a very robust (if costly) subspace tracking algorithm is employed. This could also be mitigated by adaptive learning rate or using a different order in which the estimation then truncation process is performed.

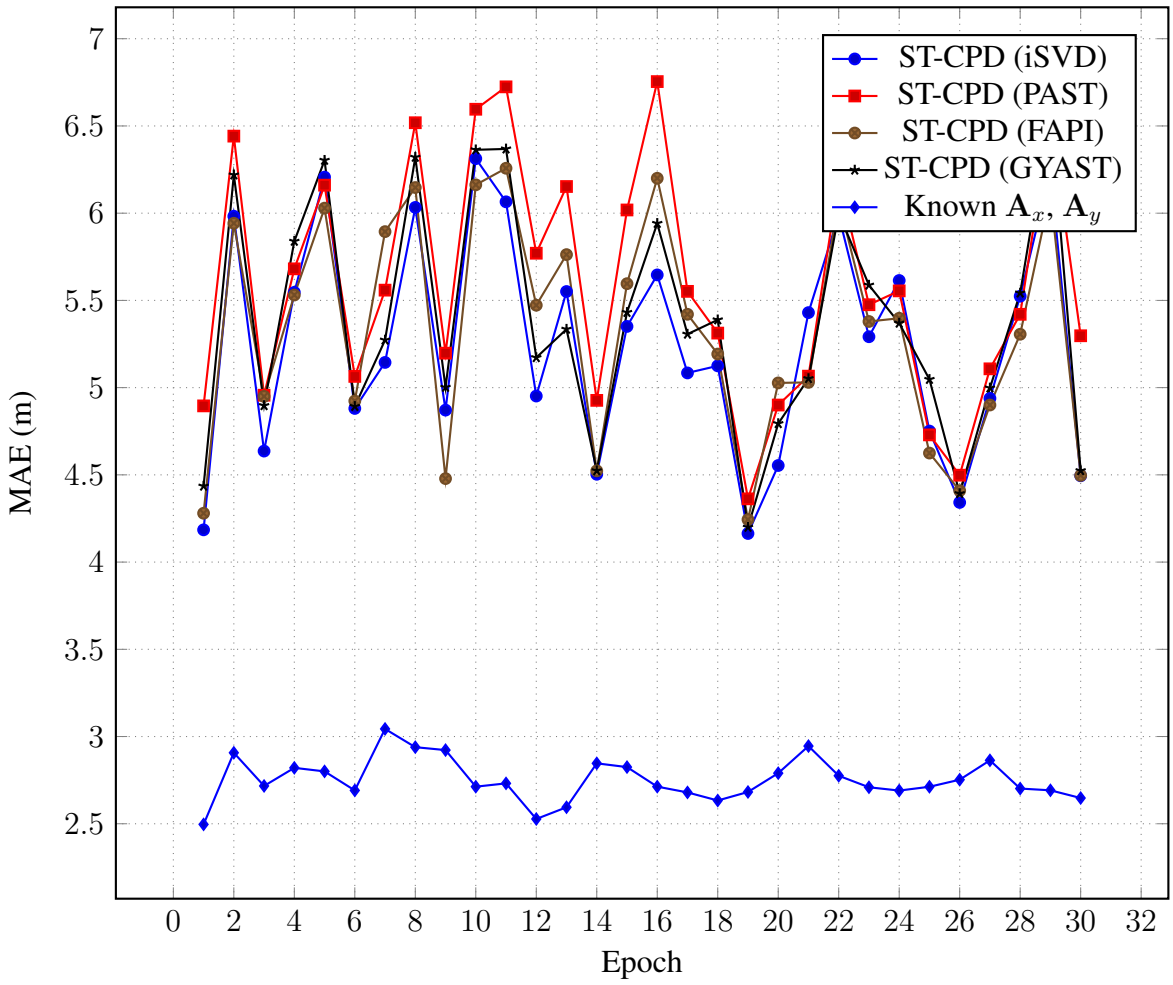

Figure 5.10: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

Unlike the static scenario, results were not close to the baseline, but were still consistent for all subspace tracking methods employed.

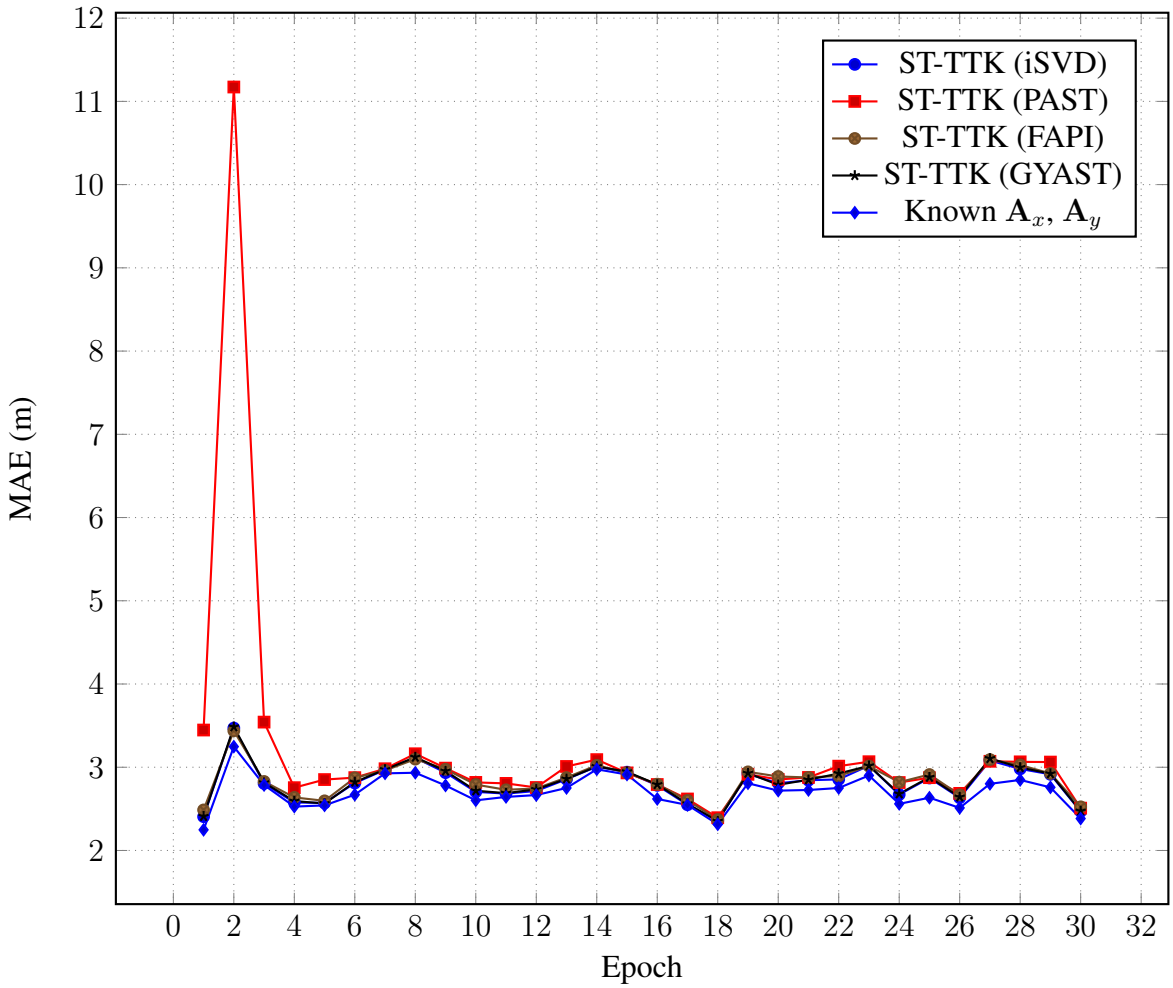

Figure 5.11: Time-delay estimation error in meters for $L = 2$, $\Delta\tau = 1.0T_c$.

The ST-HOE demonstrates the worst performance, and ST-CPD is slightly better. All ST-TTK approaches demonstrated tracking consistent with having a priori channel information with the exception of ST-TTK using PAST, which shows a spike in the very beginning before achieving performance on par with the other subspace tracking algorithms.

Chapter 6

Conclusion

We have reviewed the state-of-the-art tensor-based time-delay estimation batch processing schemes for lowly correlated scenarios, when $L = 2$, and considered the case when more than one multipath component is present, when $L > 2$, and these multipath components are clustered.

We modified the data model such that the PRS code factor matrix is the third mode, thus avoiding rank deficiencies in the first and second modes, and proposed an algebraic tensor-based time-delay estimation scheme which utilizes CPD-GEVD to estimate the factor matrices which are used in LOS selection and time-delay estimation. A variant CPD-GEVD scheme which takes a more balanced approach was also proposed.

Simulations comparing the performance of the state-of-the-art tensor-based time-delay estimation batch processing under lowly correlated scenarios were performed to compare performance in terms of RMS time-delay estimation and reconstruction error. Under this scenario, DoA/KRF showed the most consistent performance in time-delay estimation but it is matched by SECSI and CPD-GEVD and is outperformed by variant CPD-GEVD as LOS/NLOS correlation decreases. The variant CPD-GEVD approach has the lowest reconstruction error performance which suggests it is a good method for algebraic factor matrix estimation of third-order tensors. When a more realistic scenario is induced by simulating the effect an imperfect array, in which case DoA/KRF time-delay estimation performance is impacted even for a very low probability ($p = 10^{-30}$) of array imperfections exceeding half a wavelength, while SECSI and both GEVD methods performance remained fairly consistent and in a few cases showed slight improvement relative to the baseline comparison.

For tensor-based time-delay estimation batch processing schemes for clustered scenarios, when $L > 2$, we proposed a data model based on a specific block term decomposition which allows for rank deficiencies in the PRS factor matrix and proposed a tensor-based time-delay estimation scheme based on (L_r, L_r-1)-GEVD (and a variant) which algebraically estimates the factor matrices and utilizes a proposed Euclidean-distance based clustering scheme that selects the LOS component. The proposed method and its variant has performance on par

with the variant CPD-GEVD, and like the previously proposed methods, is robust against array imperfections.

We have presented adaptive processing as an alternative to batch-mode processing which offers an on-line solution to iteratively process data as it is received at the receiver which has low complexity. Received data processing is done using both classical direction-of-arrival estimation techniques to create a filter and tensor techniques which expand the data into higher dimensions to create a higher-order rank-1 filter. Results for the higher order rank-1 filter were best when both LOS and NLOS components were clustered and thus its factor matrix was rank deficient, otherwise a performance penalty occurs, this is very apparent using the iterative SVD rank-1 subspace tracking, which tends to diverge with increasing number of epochs. The parametric approaches can yield consistent results but requires a few epochs to correctly track the subspace.

For the fourth-order data mode, where a tensor stream is available, we presented three approaches, all three employing sequential truncation to reduce the amount of computations preformed by subspace tracking. While this does not impact the ST-CPD-GEVD2 or ST-TTK approaches, it does affect the higher-order eigenfilter unless a high-cost subspace tracking approach is employed. This can possibly be mitigated by simply changing the order in which subspace estimation and compression occurs. This would require a scheme for selecting the order in which truncation occurs. Another possible approach for mitigation could be an adaptive forgetting factor. This would dynamically change the learning rate but would also require some criterion under which is occurs.

6.1 Future works

Future works would include array interpolation to resolve calibration issues with the array, present under realistic scenarios, and a sliding correlator bank, as is the case with a delay-locked loop (DLL). Another avenue for future research is that of differential GNSS, in which measurements under different baselines are combined to reduce propagation errors and achieve higher accuracy time-delay estimation.

Further studies on different subspace tracking techniques, such as fast data projection method (FDPM) could reveal more robust approaches to on-line time-delay estimation using both parametric as well as algebraic approaches, along with studying the result of applying not only an exponential window as well as a sliding window, and applications using adaptive learning rates to better accommodate incoming data.

Bibliography

[1] C. Kang, S. H. Lee, and C. C. Chung, "On-Road Vehicle Localization with GPS under Long Term Failure of a Vision Sensor," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, Sep. 2015, pp. 1638–1643.

[2] G. A. Santos, J. P. C. L. da Costa, D. V. de Lima, M. da R. Zanatta, B. J. G. Praciano, G. P. M. Pinheiro, F. L. L. de Mendonca, 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)*. IEEE, Nov. 2020.

[3] A. Nelson, "Civil standardization of the Global Positioning System for the aviation community," in *IEEE PLANS 92 Position Location and Navigation Symposium Record*. IEEE, 1992, pp. 27–31.

[4] K. Uchida, N. Arai, K. Moriya, Y. Miyamoto, T. Kakihara, and T. Tokai, "Development of automatic system for monitoring fishing effort in conger-eel tube fishery using radio frequency identification and global positioning system," *Fisheries Science*, vol. 71, no. 5, pp. 992–1002, 2005.

[5] L. V. de Miranda, P. G. Kinas, G. G. Moreira, R. C. Namora, and M. H. Carneiro, "Survey sampling for fisheries monitoring in Brazil: implementation and analysis," *Brazilian Journal of Oceanography*, vol. 64, no. 4, pp. 401–414, Dec. 2016.

[6] M. Antonini, M. Ruggieri, R. Prasad, U. Guida, and G. F. Corini, "Vehicular remote tolling services using EGNOS," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 10, pp. 3–8, Oct. 2005.

[7] T. Esau, Q. Zaman, D. Groulx, K. Corscadden, Y. Chang, A. Schumann, and P. Havard, "Economic analysis for smart sprayer application in wild blueberry fields," *Precision Agriculture*, vol. 17, no. 6, pp. 753–765, Apr. 2016.

[8] S. Haykin, *Digital Communication Systems*. John Wiley & Sons, 2013. [Online]. Available: https://www.ebook.de/de/product/12148635/simon_haykin_digital_communication_systems.html

[9] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2007.

[10] R. D. J. 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, Oct. 1993.

[11] R. D. J. van Nee, J. Siereveld, P. C. Fenton, and B. R. Townsend, "The multipath estimating delay lock loop: approaching theoretical accuracy limits," in *Position Location and Navigation Symposium, 1994., IEEE*, Apr. 1994, pp. 246–251.

[12] J. Soubielle, I. Fijalkow, P. Duvaut, and A. Bibaut, "GPS positioning in a multipath environment," *IEEE Transactions on Signal Processing*, vol. 50, no. 1, pp. 141–150, Jan. 2002.

[13] J. J. Spilker Jr. and B. W. Parkinson, Eds., *Global Positioning System: Theory and Applications*, ser. Progress in Astronautics and Aeronautics. American Institute of Aeronautics and Astronautics (AIAA), 1996, vol. I.

[14] M. S. Braasch and A. J. van Dierendonck, "GPS Receiver Architecture and Measurements," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 48–64, Jan. 1999.

[15] B. Hammoud, "Tensor-Based Approach for Multipath Mitigation for GNSS," mathesis, Institute for Circuit Theory and Signal Processing, Munich University of Technology, 2015.

[16] B. Hammoud, F. Antreich, J. A. Nossek, J. P. C. L. da Costa, and A. L. F. de Almeida, "Tensor-Based Approach for Time-Delay Estimation," in *WSA 2016; 20th International ITG Workshop on Smart Antennas*, Mar. 2016, pp. 1–7.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A Multilinear Singular Value Decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, Jan. 2000.

[18] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A New Truncation Strategy for the Higher-Order Singular Value Decomposition," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A1027–A1052, Jan. 2012.

[19] 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, Jan. 1989.

[20] 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, Aug. 1985.

[21] P. R. B. Gomes, A. L. F. de Almeida, J. P. C. L. da Costa, and G. Del Galdo, "Tensor-based methods for blind spatial signature estimation under arbitrary and unknown source covariance structure," *Digital Signal Processing*, vol. 62, pp. 197–210, Mar. 2017.

[22] A. Paulraj, R. Roy, and T. Kailath, "Estimation Of Signal Parameters Via Rotational Invariance Techniques — ESPRIT," in *Nineteeth Asilomar Conference on Circuits, Systems and Computers, 1985.*, Nov. 1985, pp. 83–89.

[23] 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*, Apr. 2009, pp. 3641–3644.

[24] D. V. de Lima, J. P. C. L. da Costa, F. Antreich, R. K. Miranda, and G. Del Galdo, "High Resolution Time-Delay Estimation via Direction of Arrival Estimation and Khatri-Rao Factorization for Multipath Mitigation," in *WSA 2017; 21th International ITG Workshop on Smart Antennas*, Mar. 2017, pp. 1–8.

[25] J. P. C. L. da Costa, M. Haardt, F. Roemer, and G. Del Galdo, "Enhanced model order estimation using higher-order arrays," in *Proceedings of the 41st Asilomar Conference on Signals, Systems, and Computers*. IEEE, Nov. 2007, pp. 412–416.

[26] J. P. C. L. da Costa, M. Haardt, and F. Roemer, "Robust methods based on the HOSVD for estimating the model order in PARAFAC models," in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2008, pp. 510–514.

[27] J. P. C. L. da Costa, F. Roemer, M. Haardt, and R. T. de Sousa, "Multi-dimensional model order selection," *EURASIP Journal on Advances in Signal Processing.*, vol. 2011, no. 26, pp. 1–13, Jan. 2011.

[28] 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, Dec. 2017, pp. 1–5.

[29] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.

[30] J. P. C. L. da Costa, A. L. F. de Almeida, W. C. Freitas, and D. V. de Lima, "Low Complexity Closed-form Solution to Semi-blind Joint Channel and Symbol Estimation in MIMO-OFDM," in *ICOF 2016; 19th International Conference on OFDM and Frequency Domain Techniques*, Aug. 2016, pp. 1–5.

[31] M. Weis, F. Roemer, M. Haardt, and P. Husar, "Dual-symmetric Parallel Factor analysis using Procrustes estimation and Khatri-Rao factorization," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 270–274.

[32] D. V. de Lima, J. P. C. L. da Costa, J. P. A. Maranhão, and R. T. de Sousa, "Time-delay estimation via Procrustes estimation and Khatri-Rao factorization for GNSS multipath mitigation," in *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, Dec. 2017, pp. 1–7.

[33] 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, Jan. 2014.

[34] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)," *Signal Processing*, vol. 93, no. 9, pp. 2722–2738, Sep. 2013.

[35] M. d. 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 *2017 Workshop on Communication Networks and Power Systems (WCNPS)*. IEEE, Nov. 2017, pp. 1–4.

[36] M. d. R. Zanatta, F. L. L. de Mendonça, F. Antreich, D. V. de Lima, R. K. Miranda, G. Del 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.

[37] D. V. de Lima, M. da Rosa 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*, vol. 101, p. 102715, Jun. 2020.

[38] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part I: Lemmas for Partitioned Matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.

[39] M. Sørensen and L. De Lathauwer, "Coupled Canonical Polyadic Decompositions and (Coupled) Decompositions in Multilinear Rank-$(L_{r,n}, L_{r,n}, 1)$ Terms—Part I: Uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 496–522, Jan. 2015.

[40] R. Heddergott and P. E. Leuthold, "An extension of stochastic radio channel modeling considering propagation environments with clustered multipath components," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 8, pp. 1729–1739, Aug. 2003.

[41] NAVSTAR GPS Joint Program Office (JPO), *GPS NAVSTAR User's Overview, YEE-82-009D*, Mar. 1991.

[42] E. D. Kaplan and C. J. Hegarty, Eds., *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House, 2017. [Online]. Available: https://www.ebook.de/de/product/28653852/understanding_gps_gnss_principles_and_applications_third_edition.html

[43] G. L. Stüber, *Principles of Mobile Communication*. Springer-Verlag GmbH, Jun. 2017. [Online]. Available: https://www.ebook.de/de/product/28548361/gordon_l_stueber_principles_of_mobile_communication.html

[44] V. V. Dvorkin, Y. I. Nosenko, Y. M. Urlichich, and A. M. Finkel'shtein, "The Russian global navigation satellite program," *Herald of the Russian Academy of Sciences*, vol. 79, no. 1, pp. 7–13, feb 2009.

[45] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (Corresp.)," *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619–621, 1967.

[46] J. J. Rushanan, "Weil Sequences: A Family of Binary Sequences with Good Correlation Properties," in *2006 IEEE International Symposium on Information Theory*. IEEE, Jul. 2006.

[47] J. S. Vera, "Efficient Multipath Mitigation in Navigation Systems," Ph.D. dissertation, Universitat Politécnica de Catalunya, Feb. 2004.

[48] C. Loan, *Introduction to scientific computing : a matrix-vector approach using MATLAB*. Upper Saddle River, NJ: Prentice Hall, 2000.

[49] N. Czink, "The Random-Cluster Model - A Stochastic MIMO Channel Model for Broadband Wireless Communication Systems," Ph.D. dissertation, Vienna University of Technology, Dec. 2007.

[50] P. Xie and M. G. Petovello, "Measuring GNSS Multipath Distributions in Urban Canyon Environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 366–377, Feb. 2015.

[51] T. G. Kolda, "Multilinear Operators for Higher-Order Decompositions," Sandia National Laboratories, Albuquerque, NM, Livermore, CA, Tech. Report SAND2006-2081, Apr. 2006.

[52] G. Favier and A. L. F. de Almeida, "Overview of constrained PARAFAC models," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, Sep. 2014.

[53] H. L. van Trees, *Optimum Array Processing*. Wiley-Blackwell, 2002. [Online]. Available: https://www.ebook.de/de/product/3605930/harry_l_van_trees_optimum_array_processing.html

[54] R. L. Kashyap, "Optimal choice of AR and MA parts in autoregressive moving average models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 2, pp. 99–104, Mar. 1982.

[55] F. L. L. de Mendonça, D. V. de Lima, M. R. Zanatta, J. P. C. L. da Costa, R. K. Miranda, A. L. F. de Almeida, and R. T. de Sousa, "Estimação de Componentes de Multipercurso para Receptor Tensorial de GPS de 2ª e 3ª Geração," in *XXXVII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2019)*. Sociedade Brasileira de Telecomunicações, 2019.

[56] F. L. Hitchcock, "The Expression of a Tensor or a Polyadic as a Sum of Products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[57] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[58] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, 1970.

[59] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823, Mar. 2000.

[60] J. Foley, *Computer graphics : principles and practice*. Reading, Mass: Addison-Wesley, 1990.

[61] M. A. M. Marinho, F. Antreich, J. P. C. L. da Costa, and J. A. Nossek, "A signal adaptive array interpolation approach with reduced transformation bias for DOA estimation of highly correlated signals," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2272–2276.

[62] Z. Chen, G. Gokeda, and Y. Yu, *Introduction to Direction-of-Arrival Estimation*. Artech House, 2010.

[63] M. R. Zanatta, D. V. de Lima, J. P. C. L. da Costa, R. K. Miranda, and F. Antreich, "Técnica Tensorial de Estimação de Atraso para GPS de Segunda e Terceira Geração," in *XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2018)*, Campina Grande, PB, Sep. 2018.

[64] P. R. B. Gomes, A. L. F. de Almeida, J. P. C. L. da Costa, J. C. M. 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*, pp. 1–11, 2017.

[65] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.

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

[67] F. Roemer and M. Haardt, "A closed-form solution for multilinear PARAFAC decompositions," in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*. IEEE, Jul. 2008.

[68] C. F. Van Loan, *Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB*. Pearson, 1999.

[69] A. Yeredor, "On using exact joint diagonalization for noniterative approximate joint diagonalization," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 645–648, Sep. 2005.

[70] M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities*. Dover Publications, 1992.

[71] V. de Silva and L.-H. Lim, "Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, Jan. 2008.

[72] A. Stegeman and L. De Lathauwer, "Are diverging CP components always nearly proportional?" *arXiv math.AG*, Oct. 2011.

[73] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.

[74] L. De Lathauwer and D. Nion, "Decompositions of a Higher-Order Tensor in Block Terms—Part III: Alternating Least Squares Algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.

[75] D. V. de Lima and J. P. C. L. da Costa, "Mitigação de Multipercurso por Estimação de Procrustes e Fatorização Khatri-Rao para Estimação de Atraso em GNSS," in *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2017)*, São Pedro, SP, Sep. 2017, pp. 1019–1023.

[76] B. Yang, "Subspace tracking based on the projection approach and the recursive least squares method," in *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, 1993.

[77] ——, "Projection approximation subspace tracking," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.

[78] T. Adali and S. Haykin, *Adaptive Signal Processing*. John Wiley & Sons, Mar. 2010. [Online]. Available: https://www.ebook.de/de/product/9338180/adali_haykin_tuelay_adali_simon_haykin_adaptive_signal_processing.html

[79] F. Roemer, E.-K. Kasnakli, Y. Cheng, and M. Haardt, "Tensor subspace tracking via Kronecker structured projections (TeTraKron)," in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, Dec. 2013.

[80] C. C. R. Garcez, D. V. de Lima, R. K. Miranda, F. Mendonça, J. P. C. L. da Costa, A. L. F. 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.

[81] Y. Cheng, F. Roemer, O. Khatib, and M. Haardt, "Tensor subspace Tracking via Kronecker structured projections (TeTraKron) for time-varying multidimensional harmonic retrieval," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, Aug. 2014.

[82] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2931–2941, Aug. 2005.

[83] R. Badeau, G. Richard, and B. David, "Fast and Stable YAST Algorithm for Principal and Minor Subspace Tracking," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3437–3446, Aug. 2008.

[84] M. Arjomandi-Lari and M. Karimi, "Generalized YAST algorithm for signal subspace tracking," *Signal Processing*, vol. 117, pp. 82–95, Dec. 2015.

[85] D. Nion and N. D. Sidiropoulos, "Adaptive Algorithms to Track the PARAFAC Decomposition of a Third-Order Tensor," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2299–2310, Jun. 2009.

[86] F. Roemer, "Advances in subspace-based parameter estimation: Tensor-ESPRIT-type methods and non-circular sources," mathesis, Ilmenau University of Technology, Oct. 2006.

[87] F. Roemer, M. Haardt, and G. Del Galdo, "Analytical performance assessment of multi-dimensional matrix- and tensor-based ESPRIT-type algorithms," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2611–2625, May 2014.

[88] P. G. H. Golub and P. C. F. V. Loan, *Matrix Computations (Johns Hopkins Studies in the Mathematical Sciences)*. The Johns Hopkins University Press, 1989.

[89] A. J. Stothers, "On the Complexity of Matrix Multiplication," Ph.D. dissertation, University of Edinburgh, 2010.

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

[91] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, pp. 1–37, Oct. 2008.

APPENDICES

Appendix A

Signal generator

A.1 Theory

Given a PRS $c(t)$ with bandwidth B Hz, sequence time T s, with $N_c = B \times T$ chips (or equivalently, chip rate, in chips per epoch), chip period $T_c = 1/B$, and delay τ, its sampled output using N samples, with oversampling factor $F = N/N_c$, can be estimated from the convolution of the pulse function $p(t)$ with the PRS

$$s(t) = p(t) * c(t - \tau). \tag{A.1}$$

Since it is expensive to calculate the convolution in the temporal domain, the Fourier transformation can be applied such that

$$S(f) = P(f)C(f)e^{-j2\pi f\tau}, \tag{A.2}$$

where $P(f) = \mathcal{F}\{p(t)\}$ and $C(f)e^{-j2\pi f\tau} = \mathcal{F}\{c(t-\tau)\}$.

Followed by conversion back into the temporal domain using the inverse Fourier transformation

$$s(t) = \mathcal{F}^{-1}\{S(f)\} = \mathcal{F}^{-1}\left\{P(f)C(f)e^{-j2\pi f\tau}\right\} = \mathcal{F}^{-1}\left\{\mathcal{F}\{p(t) * c(t-\tau)\}\right\}. \tag{A.3}$$

A.2 Implementation

In practice, we'll be working with discretized values are stored in vectors, such that eq. (A.1) is expressed as

$$\mathbf{s} = \mathbf{p} * \mathbf{c}[-\tau] \in \mathbb{R}^N, \tag{A.4}$$

and eq. (A.2) is expressed as

$$\mathbf{s}_{\text{DFT}} = \mathbf{p}_{\text{DFT}} \odot \mathbf{c}_{\text{DFT}} \odot \exp\left(-j2\pi\mathbf{f}\tau\right) = \mathbf{p}_{\text{DFT}} \odot \mathbf{c}_{\text{DFT}} \odot \mathbf{e} \in \mathbb{C}^N, \tag{A.5}$$

where $\mathbf{f}$ is the double-sided frequency sampling vector $\mathbf{f}$ defined as

$$\mathbf{f} \triangleq \begin{bmatrix} -B & -B+f_b & -B+2f_b & \cdots & B-2f_b & B-f_b \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^N, \tag{A.6}$$

where f_b is the basis frequency in Hz, for a sampling factor F, is calculated as

$$f_b = F \cdot \frac{B}{N}. \tag{A.7}$$

Equation (A.5) is then converted back into the temporal domain discarding any complex components

$$\mathbf{s} = \Re\left\{\mathcal{F}^{-1}\left\{\mathbf{s}_{\text{DFT}}\right\}\right\} \in \mathbb{R}^N. \tag{A.8}$$

Pulse shape Fourier transform

The sampling vector is used such that the discretized pulse shape Fourier transform is calculated using the frequency sampling vector f defined previously

$$\mathbf{p}_{\text{DFT}} = P(\mathbf{f}) \in \mathbb{C}^N. \tag{A.9}$$

Pseudorandom sequence Fourier transform

The Fourier transform of the PRS can be estimated by repeating the (known) sequence F times. For $F = 2$,

$$\mathbf{c}_{\text{DFT}} = \mathcal{F}\left\{\text{vec}\left\{\begin{bmatrix}\mathbf{c}[0] & \mathbf{c}[0]\end{bmatrix}^{\mathsf{T}}\right\}\right\} \in \mathbb{C}^N. \tag{A.10}$$

Delay exponential

For a single delay τ the exponential vector is calculated using the frequency sampling vector

$$\mathbf{e} = \exp\left(-j2\pi\mathbf{f}\tau\right) \in \mathbb{C}^N. \tag{A.11}$$

For L delays in a vector $\boldsymbol{\tau} = \begin{bmatrix}\tau_1, \ldots, \tau_L\end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^L$, an exponential delay matrix can be

calculated

$$\mathbf{E} = \exp\left(-j2\pi\mathbf{f}\boldsymbol{\tau}^{\mathrm{T}}\right) \in \mathbb{C}^{N\times L}, \tag{A.12}$$

in which case we calculate a signal matrix $\mathbf{S} \in \mathbb{R}^{N\times L}$. For this, the Fourier transforms of both the pulse shape and the PRS have to be repeated across the second dimension L times such that

$$\mathbf{C}_{\text{DFT}} = \begin{bmatrix} \mathbf{c}_{\text{DFT}} & \cdots & \mathbf{c}_{\text{DFT}} \end{bmatrix} \in \mathbb{C}^{N\times L} \qquad \mathbf{P}_{\text{DFT}} = \begin{bmatrix} \mathbf{p}_{\text{DFT}} & \cdots & \mathbf{p}_{\text{DFT}} \end{bmatrix} \in \mathbb{C}^{N\times L} \tag{A.13}$$

and we calculate

$$\mathbf{S}_{\text{DFT}} = \mathbf{P}_{\text{DFT}} \odot \mathbf{C}_{\text{DFT}} \odot \mathbf{E} \tag{A.14}$$

finally resulting in

$$\mathbf{S} = \Re\left\{\mathcal{F}^{-1}\left\{\mathbf{S}_{\text{DFT}}\right\}\right\} = \Re\left\{\mathcal{F}^{-1}\left\{\mathbf{P}_{\text{DFT}} \odot \mathbf{C}_{\text{DFT}} \odot \mathbf{E}\right\}\right\}. \tag{A.15}$$

A.2.1 GPS C/A Code

Second generation GPS uses C/A codes, also referred to as Gold codes, as spreading sequences. Gold code generation is described in detail in [45].

This PRS is a known sequence with bandwidth $B = 1.023$ MHz, sequence time $T = 1$ ms, and chip rate $N_c = 1.023 \times 10^6 \cdot 1 \times 10^{-3} = 1,023$ chips/epoch. For an oversampling factor $F = 2$, we have $N = F \cdot N_c = 2 \cdot 1,023 = 2,046$ samples/epoch.

Pulse shape The pulse shape is a rectangular pulse shape defined as

$$p(t) = \mathrm{rect}(t) = \frac{1}{\sqrt{T_c}}\left(U\left(t + \frac{T_c}{2}\right) - U\left(t - \frac{T_c}{2}\right)\right), \tag{A.16}$$

where $U(t)$ denotes the unit step function

$$U(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 1 \end{cases}. \tag{A.17}$$

This pulse shape has known Fourier transform

$$P(f) = \frac{\sqrt{T_c}\sin(\pi f T_c)}{\pi f T_c} = \sqrt{T_c}\,\mathrm{sinc}(fT_c), \tag{A.18}$$

and $\mathbf{p}_{\text{DFT}}$ is calculated as

$$\mathbf{p}_{\text{DFT}} = P(\mathbf{f}) = \sqrt{T_c}\,\text{sinc}(T_c\mathbf{f}). \tag{A.19}$$

Correlator bank

The correlator bank is formed by several delayed copies of the (sampled) PRS. Given equally spaced delays $\boldsymbol{\tau}_{\text{CB}} = \left[-T_c, -T_c + \Delta T, \ldots, T_c - \Delta T, T_c\right] \in \mathbb{R}^Q$ with $\Delta T = 2T_c/(Q-1)$, the correlator bank $\mathbf{Q}$ is calculated as

$$\mathbf{Q} = \Re\left\{\mathcal{F}^{-1}\left\{\left(\mathbf{p}_{\text{DFT}} \odot \mathbf{c}_{\text{DFT}}\right)\mathbf{1}_Q^{\text{T}} \odot \mathbf{E}_Q\right\}\right\} \in \mathbb{R}^{N\times Q}, \tag{A.20}$$

with $\mathbf{E}_Q = \exp\left(-j2\pi\mathbf{f}\boldsymbol{\tau}_Q^{\text{T}}\right) \in \mathbb{C}^{N\times Q}$.

To not color the noise when performing correlation, the "thin" SVD is applied to $\mathbf{Q}$ to make it unitary while preserving its information

$$\mathbf{Q} = \tilde{\mathbf{Q}}\boldsymbol{\Sigma}\mathbf{V}^{\text{T}} \in \mathbb{R}^{N\times Q}, \tag{A.21}$$

with $\tilde{\mathbf{Q}} \in \mathbb{R}^{N\times Q}$.

Appendix B

Computational cost

B.1 Linear algebra operations

References in [70, 88–90].

Vectors

Given $\mathbf{a} \in \mathbb{C}^M$ and $\mathbf{b} \in \mathbb{C}^N$

Operation	**Example**	**Operations**
Addition	$\mathbf{a} + \mathbf{a}^*$	M
Subtraction	$\mathbf{b} - \mathbf{b}^*$	N
Multiplication	$\mathbf{a}^{\mathrm{T}} \cdot \mathbf{a}$	M^2
Multiplication	$\mathbf{a} \cdot \mathbf{a}^{\mathrm{T}}$	M^2
Multiplication	$\mathbf{a} \cdot \mathbf{b}^{\mathrm{T}}$	MN
Kronecker prod.	$\mathbf{a} \otimes \mathbf{b}$	MN
Khatri-Rao prod.	$\mathbf{a} \diamond \mathbf{b}$	MN
Outer prod.	$\mathbf{a} \circ \mathbf{b}$	MN

Square matrices

Given square matrices $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{M \times M}$

Operation	Example	Operations
Addition	$\mathbf{A} + \mathbf{B}$	M^2
Subtraction	$\mathbf{A} - \mathbf{B}$	M^2
Hadamard prod.	$\mathbf{A} \odot \mathbf{B}$	M^2
Multiplication	$\mathbf{A} \cdot \mathbf{B}$	M^3
Inversion	$\mathbf{A}^{-1}$	M^3
EVD	$\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^{-1}$	M^3
Generalized EVD (right)	$\mathbf{A}\mathbf{R} = \mathbf{B}\mathbf{D}\mathbf{R}$	M^3
Generalized EVD (left)	$\mathbf{L}^{\mathrm{H}}\mathbf{A} = \mathbf{D}\mathbf{L}^{\mathrm{H}}\mathbf{B}$	M^3
Khatri-Rao prod.	$\mathbf{A} \diamond \mathbf{B}$	M^3
Kronecker prod.	$\mathbf{A} \otimes \mathbf{B}$	M^4
Outer prod.	$\mathbf{A} \circ \mathbf{B}$	M^4

Rectangular matrices

Given rectangular matrices $\mathbf{A} \in \mathbb{C}^{M \times R}$ and $\mathbf{B} \in \mathbb{C}^{N \times R}$, and $M > N > R$

Operation	Example	Operations
Addition	$\mathbf{A} + \mathbf{A}^*$	MR
Subtraction	$\mathbf{B} - \mathbf{B}^*$	NR
Multiplication	$\mathbf{A} \cdot \mathbf{B}^{\mathrm{T}}$	MNR
SVD (Full)	$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$	$MR^2 + M^2R$
SVD ("thin")	$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$	MR^2
SVD ("thin")	$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$	MR^2
SVD (rank-1)	$\mathbf{A} = \mathbf{u}s\mathbf{v}^{\mathrm{H}}$	MR
Pseudoinverse	$\mathbf{A}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}$	$R^3 + 2MR^2 + R$
Khatri-Rao prod.	$\mathbf{A} \diamond \mathbf{B}$	MNR
Kronecker prod.	$\mathbf{A} \otimes \mathbf{B}$	MNR^2
Outer prod.	$\mathbf{A} \circ \mathbf{B}$	MNR^2
LSKRF of $\mathbf{A} \diamond \mathbf{B}$ (via R rank-1 SVDs)	$\mathbf{b}_r\mathbf{a}_r^{\mathrm{T}} = \mathbf{u}s\mathbf{v}^{\mathrm{H}}$	$(MN + M + N)R$

Given rectangular matrices $\mathbf{A} \in \mathbb{C}^{M \times R}$ and $\mathbf{B} \in \mathbb{C}^{N \times S}$

Operation	Example	Operations
Kronecker prod.	$\mathbf{A} \otimes \mathbf{B}$	$MNRS$
Outer prod.	$\mathbf{A} \circ \mathbf{B}$	$MNRS$

B.2 Multilinear algebra operations

Third-order tensors

Given $\boldsymbol{\mathcal{T}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]_R = \boldsymbol{\mathcal{I}}_{3,R} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \in \mathbb{C}^{I \times J \times K}$

Operation	**Example**	**Operations**
1st-mode prod.	$\boldsymbol{\mathcal{I}}_{3,R} \times_1 \mathbf{A}$	IR^3
1, 2nd-mode prod.	$\boldsymbol{\mathcal{I}}_{3,R} \times_1 \mathbf{A} \times_2 \mathbf{B}$	$IR^3 + IJR^2$
$1, 2,$ 3rd-mode prod.	$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{I}}_{3,R} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$	$IR^3 + IJR^2 + IJKR$
MLSVD ("full", no core estimation)[1]	$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$	$(I + J + K)IJK$
Core estimation	$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{T}} \times_1 \mathbf{U}_1^{\mathrm{H}} \times_2 \mathbf{U}_2^{\mathrm{H}} \times_3 \mathbf{U}_3^{\mathrm{H}}$	$(I + J + K)IJK$
MLSVD (trunc., no core estimation)	$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$	$3IJKR^2$
Core estimation (rank-R)[2]	$\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{T}} \times_1 \mathbf{U}_1^{\mathrm{H}} \times_2 \mathbf{U}_2^{\mathrm{H}} \times_3 \mathbf{U}_3^{\mathrm{H}}$	$KR^3 + JKR^2 + IJKR$
MLSVD (rank-1)	$\boldsymbol{\mathcal{T}} \approx s(\mathbf{u}_1 \circ \mathbf{u}_2 \circ \mathbf{u}_3)$	$3IJK$
Inner product	$\boldsymbol{\mathcal{T}} \bullet_3 \boldsymbol{\mathcal{T}}^*$	I^2J^2K
Outer product	$\boldsymbol{\mathcal{T}} \circ \boldsymbol{\mathcal{T}}^*$	$(IJK)^2$

[1] It is assumed $I < JK, J < IK, K < IJ$ generally, thus the "thin" SVD is used.

[2] Assuming the n-mode product is calculate sequentially from 1 to 3. Note that it is possible to reduce the number of operations by permuting the largest dimensions to the front of the tensor, or performing the n-mode product from the largest to smallest mode [18].

Appendix C

Subspace Tracking Algorithms

This appendix presents the update step of the subspace tracking algorithms used in this work. They are iterative SVD [91], projection approximation subspace tracking [76, 77], fast approximate power iteration [82], and generalized yet another subspace tracking [84].

C.1 Iterative SVD

Algorithm 14 Iterative SVD update

Input: Input matrix $\mathbf{X}[k] \in \mathbb{C}^{M \times N}$, previous subspace estimate $\hat{\mathbf{U}}[k-1] \in \mathbb{C}^{M \times L}$, previous eigenvalue estimates $\hat{\boldsymbol{\lambda}}[k-1] \in \mathbb{C}^{L}$ and forgetting factor $0 < \beta \leq 1$.
Output: Updated subspace estimate $\hat{\mathbf{U}}[k]$, updated eigenvalue estimates $\hat{\boldsymbol{\lambda}}[k]$.

1: Reconstruct covariance matrix

$$\hat{\mathbf{C}}[k-1] = \hat{\mathbf{U}}[k-1] \cdot \operatorname{dediag}\left\{\hat{\boldsymbol{\lambda}}[k-1]\right\} \cdot \hat{\mathbf{U}}[k-1]^{\mathrm{H}}.$$

2: Update covariance matrix

$$\hat{\mathbf{C}}[k] = \beta \cdot \hat{\mathbf{C}}[k-1] + \mathbf{X}[k] \cdot \mathbf{X}[k]^{\mathrm{H}}$$

3: Perform eigendecomposition of the covariance matrix

$$\hat{\mathbf{C}}[k] = \mathbf{E} \cdot \boldsymbol{\Lambda} \cdot \mathbf{E}^{\mathrm{H}}$$

4: Truncate the leading eigenvectors and corresponding eigenvalues

$$\hat{\mathbf{U}}[k] = \hat{\mathbf{E}}(1:L, \cdot),$$
$$\hat{\boldsymbol{\lambda}}[k] = \operatorname{diag}\left\{\hat{\boldsymbol{\Lambda}}\right\}.$$

C.2 Projection Approximation Subspace Tracking (PAST)

Algorithm 15 Projection Approximation Subspace Tracking update

Input: Input matrix $\mathbf{X}[k] \in \mathbb{C}^{M \times N}$, previous subspace estimate $\hat{\mathbf{U}}[k-1] \in \mathbb{C}^{M \times L}$, previous covariance matrix $\hat{\mathbf{C}}[k-1] \in \mathbb{C}^{L \times L}$ and forgetting factor $0 < \beta \leq 1$.
Output: Updated subspace estimate $\hat{\mathbf{U}}[k]$, updated covariance matrix $\hat{\mathbf{C}}[k] \in \mathbb{C}^{L \times L}$ $\hat{\boldsymbol{\lambda}}[k]$.

1: Project the input data using the previous subspace estimate

$$\mathbf{Y} = \hat{\mathbf{U}}[k-1]^{\mathrm{H}} \cdot \mathbf{X}[k] \in \mathbb{C}^{D}.$$

2: Update covariance matrix

$$\hat{\mathbf{C}}[k] = \beta \cdot \hat{\mathbf{C}}[k-1] + \mathbf{Y} \cdot \mathbf{Y}^{\mathrm{H}}.$$

3: Calculate gain matrix.

$$\mathbf{G} = \hat{\mathbf{C}}[k]^{-1} \cdot \mathbf{Y}.$$

4: Calculate projection error

$$\mathbf{E} = \mathbf{X}[k] - \hat{\mathbf{U}}[k-1] \cdot \mathbf{Y}.$$

5: Update subspace estimate

$$\hat{\mathbf{U}}[k] = \hat{\mathbf{U}}[k-1] + \mathbf{E} \cdot \mathbf{G}.$$

C.3 Fast Approximate Power Iteration (FAPI)

Algorithm 16 Fast Approximate Power Iteration update

Input: Input matrix $\mathbf{X}[k] \in \mathbb{C}^{M \times N}$, previous subspace estimate $\hat{\mathbf{U}}[k-1] \in \mathbb{C}^{M \times L}$, previous precision matrix $\hat{\mathbf{P}}[k-1] \in \mathbb{C}^{L \times L}$ and forgetting factor $0 < \beta \leq 1$.
Output: Updated subspace estimate $\hat{\mathbf{U}}[k]$, updated precision matrix $\hat{\mathbf{P}}[k] \in \mathbb{C}^{L \times L}$.

1: **for** $n = 1$ to N **do**
2: **if** $n = 1$ **then** $\beta' = \beta$
3: **else** $\beta' = 1$
4: **end if**

$$
\begin{aligned}
&\mathbf{x}(n) = \mathbf{X}(\cdot, n) \in \mathbb{C}^{M}, \\
&\mathbf{y}(n) = \hat{\mathbf{U}}[k-1]^{\mathrm{H}} \cdot \mathbf{x}(n) \in \mathbb{C}^{D}, \\
&\mathbf{h}(n) = \mathbf{P} \cdot \mathbf{y}(n) \in \mathbb{C}^{D}, \\
&\mathbf{g}(n) = \frac{\mathbf{h}(n)}{(\beta' + \mathbf{y}(n)^{\mathrm{H}} \cdot \mathbf{h})} \in \mathbb{C}^{D}, \\
&\epsilon(n) = \sqrt{\|\mathbf{x}(n)\|^2 + \|\mathbf{y}(n)\|^2}, \\
&\tau(n) = \frac{\epsilon(n)^2}{\left(1 + \epsilon(n)^2 \cdot \|\mathbf{g}(n)\| + \sqrt{1 + \epsilon(n)^2 \cdot \|\mathbf{g}(n)\|^2}\right)}, \\
&\eta(n) = 1 - \tau(n) \cdot \|\mathbf{h}(n)\|, \\
&\mathbf{y}'(n) = \eta(n) \cdot \mathbf{y}(n) + \tau \cdot \mathbf{g}(n) \in \mathbb{C}^{D}, \\
&\mathbf{h}'(n) = \hat{\mathbf{P}}[k-1]^{\mathrm{H}} \cdot \mathbf{y}'(n) \in \mathbb{C}^{D}, \\
&\boldsymbol{\epsilon}(n) = \frac{\tau(n)}{\eta(n)} \cdot \left(\hat{\mathbf{P}}[k-1] \cdot \mathbf{g}(n) - (\mathbf{h}'(n) \cdot \mathbf{g}(n)) \cdot \mathbf{g}^{\mathrm{H}}\right) \in \mathbb{C}^{D}, \\
&\hat{\mathbf{P}}[k] = \frac{1}{\beta'} \left(\hat{\mathbf{P}}[k-1] - \mathbf{g}(n) \cdot \mathbf{h}'(n)^{\mathrm{H}} + \boldsymbol{\epsilon}(n) \cdot \mathbf{g}(n)^{\mathrm{H}}\right) \in \mathbb{C}^{D \times D}, \\
&\mathbf{e}(n) = \eta \cdot \mathbf{x}(n) - \hat{\mathbf{U}}[k-1] \cdot \mathbf{y}'(n) \in \mathbb{C}^{M}, \\
&\hat{\mathbf{U}}[k] = \hat{\mathbf{U}}[k-1] + \mathbf{e}(n) \cdot \mathbf{g}(n)^{\mathrm{H}} \in \mathbb{C}^{M \times D}
\end{aligned}
$$

5: **end for**

C.4 Generalized Yet Another Subspace Tracker (GYAST)

Algorithm 17 Generalized Yet Another Subspace Tracker update

Input: Input matrix $\mathbf{X}[k] \in \mathbb{C}^{M \times N}$, previous subspace estimate $\hat{\mathbf{U}}[k-1] \in \mathbb{C}^{M \times L}$, previous precision matrix $\hat{\mathbf{P}}[k-1] \in \mathbb{C}^{L \times L}$, previous noise power estimate $\sigma_n[k-1]$, and forgetting factor $0 < \beta \leq 1$.

Output: Updated subspace estimate $\hat{\mathbf{U}}[k]$, updated precision matrix $\hat{\mathbf{P}}[k] \in \mathbb{C}^{L \times L}$, updated noise power estimate $\sigma_n[k]$.

1: **for** $n = 1$ to N **do**

2: **if** $n = 1$ **then** $\beta' = \beta$

3: **else** $\beta' = 1$

4: **end if**

$$
\begin{aligned}
&\mathbf{x}(n) = \mathbf{X}(\cdot, n) \in \mathbb{C}^{M}, \\
&\mathbf{y}(n) = \hat{\mathbf{U}}[k-1]^{\mathrm{H}} \cdot \mathbf{x}(n) \in \mathbb{C}^{D}, \\
&\mathbf{C}'_{yy}(n) = \beta' \cdot \hat{\mathbf{P}}[k-1] + \mathbf{y}(n) \cdot \mathbf{y}^{\mathrm{H}}(n) \in \mathbb{C}^{D \times D}, \\
&\sigma(n) = \sqrt{\mathbf{x}(n)^{\mathrm{H}} \cdot \mathbf{x}(n) - \mathbf{y}(n)^{\mathrm{H}} \cdot \mathbf{y}(n)}, \\
&\mathbf{z}(n) = \sigma(n) \cdot \mathbf{y}(n) \in \mathbb{C}^{D}, \\
&\gamma(n) = \beta' \cdot \sigma_n(n)^2 + \sigma(n)^2, \\
&\bar{\mathbf{C}}_{yy}(n) = \begin{bmatrix} \mathbf{C}'_{yy}(n) & \mathbf{z}(n) \\ \mathbf{z}(n)^{\mathrm{H}} & \gamma(n) \end{bmatrix} \in \mathbb{C}^{D+1, D+1}, \\
&\mathbf{q}_n(n) = \begin{bmatrix} \mathbf{q}(n) \\ r(n) \end{bmatrix} = \text{minor eigenvector of } \bar{\mathbf{C}}_{yy}(n) \in \mathbb{C}^{D+1}, \\
&\ell(n) = 1/(1 + |r(n)|), \\
&k(n) = r(n)/(|r(n)| \cdot \sigma(n)), \\
&\mathbf{e}'(n) = \hat{\mathbf{U}}[k-1] \cdot (\ell(n) \cdot \mathbf{q}(n) - k(n) \cdot \mathbf{y}(n)) + k(n) \cdot \mathbf{x}(n) \in \mathbb{C}^{M}, \\
&\hat{\mathbf{U}}[k] = \hat{\mathbf{U}}[k-1] - \mathbf{e}'(n) \cdot \mathbf{q}(n)^{\mathrm{H}} \in \mathbb{C}^{M \times D}, \\
&\mathbf{q}'(n) = \mathbf{C}'_{yy} \cdot \mathbf{q}(n) \in \mathbb{C}^{D}, \\
&\mathbf{C}(n) = \left(\frac{r(n)}{|r(n)|} \mathbf{z}(n) + \ell(n) \mathbf{q}'(n) \right) \cdot \mathbf{q}(n)^{\mathrm{H}} \in \mathbb{C}^{D \times D}, \\
&c(n) = \ell(n)^2 \cdot \mathbf{q}(n)^{\mathrm{H}} \cdot \mathbf{q}'(n) + \gamma(n) + 2 \frac{\ell(n)}{|r(n)|} \cdot \Re \left\{ r(n) \cdot \mathbf{q}(n)^{\mathrm{H}} \cdot \mathbf{z}(n) \right\}, \\
&\mathbf{C}_{yy}(n) = \mathbf{C}'_{yy}(n) + c(n) \mathbf{q}(n) \cdot \mathbf{q}(n)^{\mathrm{H}} - \mathbf{C}(n) - \mathbf{C}(n)^{\mathrm{H}} \in \mathbb{C}^{D \times D}, \\
&\lambda_m(n) = \mathrm{Tr}\left\{ \bar{\mathbf{C}}_{yy}(n) \right\} - \mathrm{Tr}\left\{ \mathbf{C}_{yy}(n) \right\}, \\
&\sigma_n[k] = \min\left(\sqrt{\lambda_m(n)}, \sigma_n[k-1] \right).
\end{aligned}
$$

5: **end for**