

Moving Object Localization Using Frequency Measurements

A Dissertation
presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
Musaab M. Ahmed
Dr. Dominic K. C. Ho, Dissertation Supervisor
December 2021

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

MOVING OBJECT LOCALIZATION USING FREQUENCY MEASUREMENTS

presented by Musaab Mohammed Ahmed,
a candidate for the degree of Doctor of Philosophy, and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Dominic Ho

Professor Naz Islam

Professor Justin Legarsky

Professor Giovanna Guidoboni

DEDICATION

To my beloved father,
Mohammed Ahmed Abdullah Al-Obaidi

ACKNOWLEDGMENTS

First of all, I would like to thank my Lord, the Sustainer, the All-Knowing, the Wise and the Generous for all the blessings that I cannot count. "And if you would count the favours of Allah, never could you be able to count them. Truly! Allah is Oft-Forgiving, Most Merciful". Quran(16:18)

" O mankind! If you are in doubt about the Resurrection, then verily We have created you (i.e. Adam) from dust, then from a Nutfah (mixed drops of male and female sexual discharge i.e. the offspring of Adam), then from a clot (a piece of thick coagulated blood) then from a little lump of flesh - some formed and some unformed (as in the case of miscarriage) - that We may make (it) clear to you (i.e. to show you Our Power and Ability to do what We will). And We cause whom We will to remain in the wombs for an appointed term, then We bring you out as infants,[1] then (give you growth) that you may reach your age of full strength. And among you there is he who dies (young), and among you there is he who is brought back to the miserable old age, so that he knows nothing after having known. And you see the earth barren, but when We send down water (rain) on it, it is stirred (to life), and it swells and puts forth every lovely kind". Quran(22:5)

Throughout the work of my PhD degree I have received a great deal of support and assistance.

I would like to acknowledge with gratefulness the Higher Committee for Education Development in Iraq (HCED) for supporting me during my master degree and giving me the permission to stay in U.S. and complete the PhD degree. Thanks a lot HCED

for your offer, it truly changed my life!

I would like to thank my supervisor, Professor Dominic Ho, whose expertise was invaluable in formulating the research directions and methodology. Prof. Ho has helped me to develop ideas and to become an outstanding researcher. He has always been available to extend his hands and help me to overcome my academic and personal issues. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

Also, I would like to thank Prof. Naz Islam, Prof. Justing Legarsky, and Prof. Giovanna Guidoboni for serving in my PhD committee. Your efforts are highly appreciated. Special thanks go to my teachers at MU who made my PhD studying successful and fruitful.

In addition, I would like to thank my parents for their wise counsel and sympathetic ear. Your love and care are priceless. Finally, I could not have completed this dissertation without the support of my wife, Maryam Al-Sabbagh, who provided me with love and compassion as well as happy distractions to rest my mind outside of my research.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	ix
LIST OF FIGURES	xii
ABSTRACT	xiv
CHAPTER	
1 Introduction	1
1.1 Background and Motivation	1
1.2 Doppler Effect	4
1.3 Applications	8
1.3.1 Acoustic Sensor Networks	8
1.3.2 Airborne Target Positioning	10
1.4 Literature Review	11
1.5 Contribution of the Research	13
1.6 Content Organization	16
2 Background	17
2.1 Localization Scenario	17
2.2 Chan Method	20
2.2.1 Carrier Frequency Unknown	21
2.2.2 Carrier Frequency Known	22

2.3	Shames Method	23
2.4	Semidefinite Programming Method	25
2.5	Maximum Likelihood Estimator(MLE)	27
2.6	CRLB	31
2.6.1	Summary	33
3	2-D Object Localization	34
3.1	Formulation	34
3.1.1	Carrier Frequency Available	35
3.1.2	Carrier Frequency Unavailable	43
3.2	Solution: Carrier Frequency Available	44
3.2.1	Algebraic Solution	45
3.2.2	SDR Solution	52
3.3	Solution: Carrier Frequency Unavailable	57
3.4	Analysis	60
3.4.1	Carrier Frequency Available	62
3.4.2	Carrier Frequency Unavailable	64
3.4.3	Complexity	66
3.5	Simulations	67
3.5.1	Single-Time Observation	68
3.5.2	Multiple-Time Observations	72
3.5.3	Number of Sensors	75
3.6	Summary	77

4	3-D Object Localization	83
4.1	Formulation	84
4.1.1	Carrier Frequency Available	85
4.1.2	Carrier Frequency Unavailable	87
4.2	Single-Time Observation	88
4.2.1	Algebraic Solution	91
4.2.2	SDP Solution	94
4.3	Multiple-Time Observations	96
4.3.1	Algebraic Solution	99
4.3.2	SDP Solution	101
4.4	carrier frequency unavailable solution	103
4.5	Sequential Multiple-Time Solution	105
4.6	Analysis	107
4.6.1	Carrier Frequency Available	108
4.6.2	Carrier Frequency Unavailable	109
4.6.3	Sequential Algorithm	111
4.6.4	Complexity	112
4.7	Simulations	114
4.7.1	Performance w.r.t. Measurement Noise	115
4.7.2	Performance w.r.t. Sensor Position and Carrier Frequency Errors	119
4.7.3	Sequential Estimation for Multiple-Time as Time k Increasing	122
4.7.4	Sensor Number	126

4.8	Summary	126
5	Moving Sensors Scenario	129
5.1	Formulation	130
5.1.1	Single-Time Measurement	134
5.1.2	Multiple-Time Measurements	138
5.2	Algebraic Solution	144
5.3	SDP Solution	157
5.4	CRLB	162
5.5	Analysis	163
5.6	Simulations	166
5.6.1	2-D Single-Time	167
5.6.2	2-D Multiple-Time	167
5.6.3	3-D Single-Time	171
5.6.4	3-D Multiple-Time	172
5.7	Summary	175
6	Conclusion and Future Work	178
6.1	Conclusion	178
6.2	Future Work	181
APPENDIX		
A		182
B		184
C		186

D	188
D.1	Algebraic Solution	188
D.2	SDP Solution	189
D.3	Sequential Estimation	190
E	192
BIBLIOGRAPHY	194
VITA	210

LIST OF TABLES

Table	Page
1.1 Doppler Effect Applications	7
3.1 Relations Among the Elements of φ for the Single-Time Measurement Case	39
3.2 Relations Among the Elements of φ for the Multiple-Time Measure- ments Case	41
3.3 Constraints Among the Elements of φ and Φ for the SDR Solution of the Single-Time Measurement Case	54
3.4 Constraints on φ and Φ for the SDR Solution of the Multiple-Time Measurements Case	56
3.5 Relative Processing Times of Different Algorithms	70
4.1 Relations Among the Elements of φ for the Single-Time Case	90
4.2 Constraints of the SDP Solution for the Single-Time Case	95
4.3 Constraints on φ and Φ for the SDP solution of the multiple-time Case	103
4.4 Complexity of the Algebraic Solution CFS and the SDP Solution in terms of the number of multiplications.	113
4.5 Parameter values for Table IV.	113

4.6	Processing Times at different values of σ relative to that of CFS at $\sigma = 0.01$ Hz.	117
4.7	Processing Times for the sequential and batch algorithms at different k values relative to that of S-CFS at $k=2$	117
5.1	The Elements of φ and The 15 Constraints for The 2-D Single-Time Measurement Case.	137
5.2	The Elements of φ and The 28 Constraints for The 3-D Single-Time Measurement Case.	139
5.3	The Elements of φ and The 21 Constraints for The 2-D Multiple-Time Measurements Case.	142
5.4	The Elements of φ and The 36 Constraints for The 3-D Multiple-Time Measurements Case.	145
5.5	The indices of the data vector for the four different cases corresponding to each code index	152
5.6	The entries of $\tilde{\mathbf{A}}_{19 \times 4}$ and $\tilde{\mathbf{A}}_{34 \times 6}$ matrices for the 2-D Single-Time and 3-D Single-Time cases respectively.	153
5.7	The entries of $\tilde{\mathbf{A}}_{25 \times 4}$ and $\tilde{\mathbf{A}}_{42 \times 6}$ matrices for the 2-D Multiple-Time and 3-D Multiple-Time cases respectively.	154
5.8	The non-zero entries of $\mathbf{C}_{19 \times 19}$ and $\mathbf{C}_{25 \times 25}$ matrices for the 2-D Single-Time and 2-D Multiple-Time cases respectively.	155
5.9	The non-zero entries of $\mathbf{C}_{34 \times 34}$ and $\mathbf{C}_{42 \times 42}$ matrices for the 3-D Single-Time and 3-D Multiple-Time cases respectively.	156
5.10	Constraints of the SDP Solution for each Localization Case	158
5.11	Constraints on φ and Φ for the SDP solution of the 2-D localization	159

5.12	Constraints on Φ for the SDP solution of the 3-D Single-Time and 3-D Multiple-Time localization	160
5.13	Constraints on φ and Φ for the SDP solution of the 3-D localization .	161
5.14	Small Noise Conditions For $k = 0, 1, \dots, N - 1$ and $i = 1, 2, \dots, M$. .	165
5.15	Sensor Positions in (m) and Velocities in (m/s) for 2-D Single-Time Localization	168
5.16	Sensor Positions in (m) and Velocities in (m/s) for 2-D Multiple-Time Localization	170
5.17	Sensor Positions in (m) and Velocities in (m/s) for 3-D Single-Time Localization	173
5.18	Sensor Positions in (m) and Velocities in (m/s) for 3-D Multiple-Time Localization	174
D.1	Computational Costs of the Algebraic Solution CFS	191
D.2	Extra Computational Costs for the Solution Update of Sequential Process	191

LIST OF FIGURES

Figure	Page
1.1 Doppler Effect Illustration.	5
1.2 Acoustic positioning system.	9
1.3 Illustration of MIMO radar system.	11
2.1 Localization scenario of a moving object.	20
3.1 Performance of the proposed methods at different σ levels for single-time measurement when f_o is available.	71
3.2 Performance of the proposed methods at different σ levels for single-time measurement when f_o is unavailable.	72
3.3 Performance of the proposed methods at different σ_s levels for single-time measurement when f_o is available.	73
3.4 Performance of the proposed methods at different σ_{f_o} levels for single-time measurement.	74
3.5 Performance of the proposed methods at different σ levels for multiple-time measurements when f_o is available.	79
3.6 Performance of the proposed methods at different σ levels for multiple-time measurements when f_o is unavailable.	80

3.7	Performance of the proposed methods using different number of sensors for single-time measurement when f_o is available.	81
3.8	Performance of the proposed methods using different number of sensors for multiple-time measurements when f_o is available.	82
4.1	3-D localization scenario of a moving object.	84
4.2	Box figure	118
4.3	Box figure	119
4.4	Box figure	120
4.5	Box figure	121
4.6	Box figure	122
4.7	Box figure	123
4.8	Box figure	124
4.9	Box figure	125
4.10	Box figure	127
4.11	Box figure	128
5.1	Performance of the proposed methods at different σ levels for 2-D single-time case.	169
5.2	Performance of the proposed methods at different σ levels for 2-D multiple-time case.	171
5.3	Performance of the proposed methods at different σ levels for 3-D single-time case.	175
5.4	Performance of the proposed methods at different σ levels for 3-D multiple-time case.	176

ABSTRACT

This research investigates the ability of locating a moving object using the Doppler shifts of a carrier frequency signal sent or reflected by the object and observed by several fixed or moving sensors spatially distributed in the 2-D or 3-D space. The idea was previously studied and several solutions are proposed based on exhaustive grid search or numerical polynomial optimization. We shall formulate the problem as a constrained optimization and propose two efficient solutions. The first is by using linear optimization method to reach a closed-form solution and the second is through semi-definite relaxation technique to achieve a noise resilient estimate. The solutions are derived first for the single-time measurement and then developed to multiple-time observations collected during a short time interval in which the object motion is linear. Several scenarios are considered including 2-D and 3-D localization geometry, the sensors are fixed or moving along nonlinear trajectory with random speed, the presence of errors in the carrier frequency and the sensor positions, and the non-cooperative object scenario where the frequency of the carrier signal is completely not known. Analysis validates the algebraic closed-form solution in reaching the Cramer-Rao Lower Bound accuracy under Gaussian noise within the small error region. The simulations show good performance for the proposed algorithms and support the theoretical analysis.

Chapter 1

Introduction

1.1 Background and Motivation

Localizing a moving object in a network of sensors finds interest in a wide area of applications including airborne surveillance, navigation, search and rescue, air traffic control, unmanned aerial vehicle and many others [1, 2, 3, 4, 5, 6, 7, 8]. The objective of a positioning system is to determine the location of one or multiple objects in the two-dimensional (2-D) or three-dimensional (3-D) space. A positioning system can appear in various forms having different configurations, usages, reliability, performance, complexity and other attributes. The location information of a stationary or moving object can be extracted from the interaction between the object and the time, frequency, energy or bearing signals collected by spatially distributed sensors during single or multiple time instants. In all these measurements, the object location parameters are non-linearly related to the available observations [9, 10, 11]. In the past

few years, huge amount of researches have been done making use of the technology advancement to develop new solutions with better accuracy and lower computational complexity [12, 13, 14, 15, 16].

In general, the most common strategy is to measure the time information of the signal. Time measurements such as time of arrival (TOA) or time difference of arrival (TDOA) are used to form a set of circle or hyperbolic equations and solutions to these nonlinear equations have been proposed over the years that are computationally efficient and able to achieve the optimal accuracy. Some of these solutions are based on iterative methods [17, 18, 19, 20], algebraic closed-form solution (CFS) [21, 22, 23, 24], numerical search [25, 26, 27] and semi-definite relaxation SDR [28, 29, 30, 31]. Similar approaches were proposed based on bearing measurements such as angle of arrival (AOA) [32, 33, 34, 35, 36], energy measurements such as received signal strength (RSS) [37, 38, 39], or a combination of these measurements [40, 41, 42]. Frequency measurements like Doppler shifted frequency (DSF) or frequency difference of arrival (FDOA) can also be exploited to improve the localization performance when there is relative motion between the object and sensors. Including frequency measurements in addition to time, several algorithms have been developed in closed-form [9, 10, 43, 44]. Localizing an object using frequency only measurements has attracted considerable interests over the past few years. It has been driven by the technology advancement capable of acquiring high precision frequency observations [45, 46, 47, 48, 49], as well as the increasing demand of innovative approaches to cover a wide range of localization applications that have been largely expanded to indoor positioning, dynamic routing, self-driving vehicle and others [50, 51, 52].

Frequency measurement contains important information regarding the velocities of

both the transmitter and the receiver. Such property is quite unique and is in demand to be studied in depth. However, because the frequency measurement depends on the radial velocity between the object and sensor, it is required at least one of the two nodes (object or sensor) to be in a moving status. It is also important to know that the position and velocity are related in a highly nonlinear fashion with the observed frequency. This made the problem difficult to be addressed by many researchers and the usual assessment was to join the frequency with another type of measurement to simplify the model and make it solvable through various solution approaches [9, 10].

Under some conditions, it might be difficult, costly, unreliable, or even impossible to have another sort of measurement besides the frequency. For example, TOA measurements require an accurate timestamp of the transmitted signal and all the sensors in the network need to be synchronized [53]. The timestamp information is dropped when using TDOA measurements, but one of the observations should be used as a reference, which degrades the overall performance [54]. Depending on the nature of the emitted signal, frequency measurement can have a much higher resolution than time observation when it has a long pulse duration and narrow bandwidth [55, 56]. Acquiring both time and frequency measurements could be computationally demanding due to the evaluation of the ambiguity function [57] or the signal time scaling [58]. Furthermore, when the emitted signal is very narrow band that is close to a pure tone, time measurements can be highly inaccurate, which renders their usefulness for localization. In the underwater environment, the accuracy of frequency-based localization algorithms can be increased at a very low cost and their transmit-receive devices are less complicated compared to those used with time measurements [59]. DOA measurements need the object to be static and their performance is highly de-

graded in shallow water because of the signal interaction with the surface and bottom of the sea [60]. RSS based localization algorithms are affected by the inaccuracies of the theoretical, roughly calibrated or imperfect channel models used to compute the location [61]. It may be more direct to use frequency measurements only for localization in such circumstances. Thereby, It is the subject of this research to investigate the positioning of a moving object by utilizing the DSF measurements observed at a number of spatially distributed sensors.

From the algorithm perspective, positioning a moving object by frequency observations only can be more challenging than using time, angle or energy measurements because frequency observation depends not only on the object position as in other observations but also on the object velocity, and the two unknowns are coupled together. It is also more complicated than using the frequency observation together with the time measurement where the model can be simplified by exploiting the time measurement [1]. To limit the scope of this study, we shall focus on single object positioning using Doppler shifts observed at a number of distributed sensors. The moving object will be considered as a point-source since it is relatively small in size compared to the localization region.

1.2 Doppler Effect

The Doppler effect first proposed in 1842 by the Austrian mathematician, Christian Doppler, it is the change in the frequency of a wave when the transmitter and the receiver are in relative motion [62]. Doppler specifically raised the idea that the change in the color of stars is due to their relative motion with respect to earth not

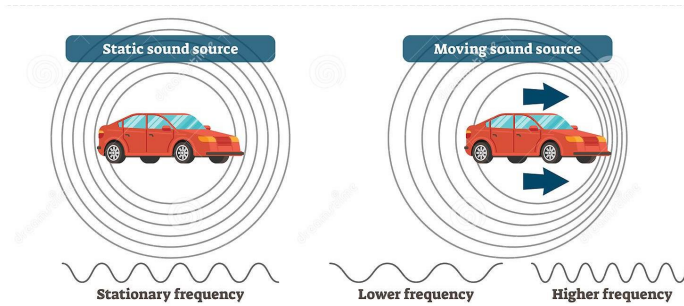


Figure 1.1: Doppler Effect Illustration.

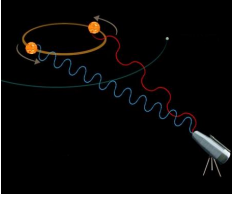


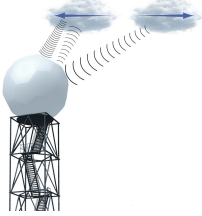

because of the change in their temperature [63]; he also assumed that this exploration is applicable to all kinds of waves. Fig. 1.1 illustrates Doppler effect on sound waves. In 1845, Buys Ballot proved Doppler's theory on sound waves by using two sets of trumpeters that have the same note with perfect pitch match, one set on a moving train and the other kept stationary at the train station [64]. Ballot noticed that the frequency of the two notes was not the same when the train passed the station. In 1868, Doppler effect began to have enormous significance to astronomy when William Huggins showed that the dark spectral lines of the brighter stars are slightly shifted from their normal position in the spectrum of the sunlight to the red or blue color due to the star's motion towards or away from the earth. Because the wavelengths of the spectral lines can be measured precisely, this technique helped later to measure the velocities of the solar prominence, double stars and rings of Saturn [65]. In 1958, The Applied Physics Laboratory of Johns Hopkins University (APL) developed the first satellite positioning system that uses Doppler effect to localize a receiver on the earth. The system uses the received carrier frequencies to determine the velocity of a moving receiver [66].

Nowadays, Doppler effect is used in many different fields ranging from medical applications, such as ultrasound imaging, to weather forecasting, traffic control and

military usages, such as target tracking. Table 1.1 lists several outstanding Doppler applications. Due to the technology advancement of digital processors, Doppler shift can be measured very precisely with relatively low cost for both microwave signals in radars and acoustic waves in sonars [67, 59]. This helped to expand the frequency-based usages to other applications like wireless sensor networks (WSN). The accuracy of estimating Doppler shift depends on many factors including the operating carrier frequency and the bandwidth of the signal. In general, we can achieve better Doppler shift resolution by using higher carrier frequencies; however, each frequency has its own advantages and weaknesses. As an example, every application is limited by a specific frequency band for operation; also, higher frequencies usually require more complex devices which increase the total system cost.

Another way to enhance the accuracy of Doppler measurement is by using narrowband signals with the developed discrete Fourier transform algorithms. Narrowband radars/sonars can be implemented with much lower complexity and hence, lower cost. On the other hand, the high power of single-frequency carriers creates significant interference to the environment in addition to increasing transmitter detectability by the adversary intelligent systems. Nevertheless, new techniques like the narrowband frequency hopping (NB-FH) can help to combat the peak power spectral density and still maintaining the narrowband advantage [68].

Table 1.1: Doppler Effect Applications

<p>Stars motion</p>	<p>Astronomers use Doppler effect to measure the movement of stars relative to Earth. Approaching stars shift the light spectrum towards shorter wavelengths whereas receding stars shift it towards longer wavelengths.</p>	
<p>Doppler echocardiogram</p>	<p>Physicians and medical technicians apply Doppler effect in the medical field to measure the direction and speed of blood cells inside the artery of the patient using ultrasound waves.</p>	
<p>Radar gun</p>	<p>Police officers use radar gun that rely on Doppler effect to check for speeding cars.</p>	
<p>Weather forecasting</p>	<p>In storm systems, meteorologists use Doppler technology to detect the direction and velocity of raindrops. These measures are used then to predict the weather patterns for the next minutes or hours.</p>	
<p>Target tracking</p>	<p>Continuous wave (CW) and pulsed Doppler radars use Doppler effect to discriminate between a moving and stationary targets. In addition, these radars provide accurate range and angle tracking of a moving target in the presence of ground clutter.</p>	

1.3 Applications

1.3.1 Acoustic Sensor Networks

Underwater systems are growing fast to support the applications of both civilian and military needs. Underwater communications depend mainly on acoustic signals because of the physical constraints that limit the propagation ability of electromagnetic waves [69]. The acoustic signals have better transmission characteristics in water than in air. They can travel over larger distances because they undergo less attenuation, their speed is four to five times higher, and they can reach much higher intensity levels [70]. Underwater acoustics is essential and indispensable in mastering most of the human activities at sea. However, these advantages are mitigated by the high ambient noise level and undesirable echoes that perturb the useful signals. Compared to electromagnetic waves used in free space, the acoustic signal has a much higher latency characteristic and thus, it can only support limited underwater distances. The frequency-dependent propagation loss limits the bandwidth of the underwater channel to a few kilohertz [71]. Furthermore, underwater devices are expensive and energy consumers [69]. These challenges in addition to the medium variability in both time and space, lowered the performance of underwater communication systems.

New technologies helped to lessen the above challenges and facilitate the difficulties to produce more efficient communication systems. These include the small size and high-speed processing unit, which increases the accuracy of the sensor measurements, the design of the autonomous underwater vehicle (AUV) that brought mobility property into network designs to enhance the accuracy [72] and the implementation of multiple sensor nodes which raises the information rates and enlarge the coverage

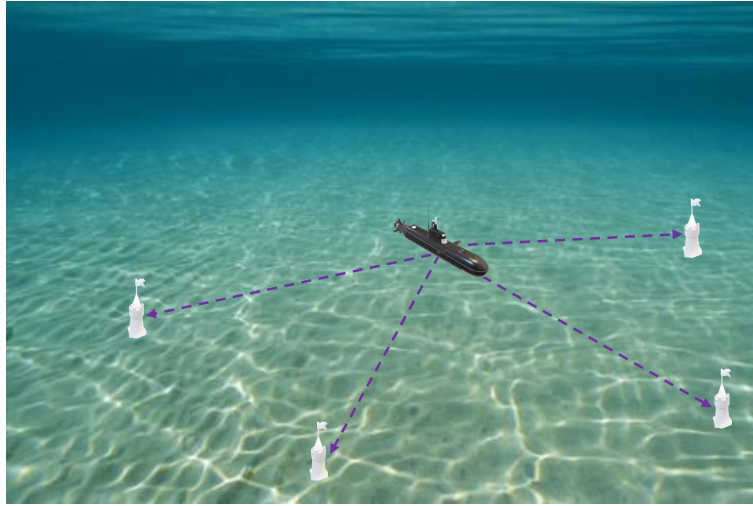


Figure 1.2: Acoustic positioning system.

area [73].

Target detection and location systems using acoustic signals are commonly called sonars. These systems use either echo of the transmitted signals (active sonar) or the direct signal received from a target (passive sonar). In the underwater environment, the accuracy of frequency-based localization algorithms can be increased at a very low cost compared to those use time measurements. Also, the time measurements transmit-receive devices are more complicated than those used to produce frequency measurements [59]. Furthermore, DOA measurements need the target to be static and their performance is highly degraded in shallow water because of the signal interaction with the surface and bottom of the sea [60]. Under some circumstances and when there is a relative motion between an object and a network of sensors, the DSF measurements might be preferred to be used for the object positioning service. Fig. 1.2 illustrates typical acoustic positioning system.

1.3.2 Airborne Target Positioning

Detecting and Tracking an airborne target has been continuously developing over the last 80 years since the radar was first introduced in the late of 1930s. Radar uses electromagnetic (EM) waves for detection and ranging. EM waves travel very fast, around 300,000 kilometers (186,000 mi) per second through the atmosphere. Unlike the acoustic signals that need molecules of solid, liquid or gas for their propagation, EM waves do not require molecules and can travel not only through a material medium but also through a vacuum of space. Because EM waves travel much faster than acoustic signals, radars can cover a much larger geometrical region compared with that covered by sonars when considering both are working on their intended environments. Regarding the frequency of operation, radar systems support a wide range of frequencies that extend to few gigahertz, while sonar systems can only work within tens of kilohertz.

In the last two decades, multiple-input multiple-output (MIMO) radar has been introduced as a newly emerging technology in the design of radar systems. The essence of the MIMO radar is to employ multiple, spatially distributed transmitters and receivers for emitting several orthogonal waveforms and capturing their echoes reflected by a moving target. One of the main problems of the conventional radar system is the target fading resulted from the low coherent gain obtained from the processing of the echos received by the array elements. In the MIMO radar system, the independent signals received at the widely separated antennas are exploited to improve the radar performance [74]. The spatial variations of the target's radar cross section (RCS) associated with the noncoherent processing of the signals can be used to obtain the diversity gain for the estimation of several parameters like the direction

of arrival and Doppler [75]. The Doppler shifts included in the reflected echos have information about both the position and velocity of the target. Fig. 1.3 illustrate the operation of MIMO radar system.



Figure 1.3: Illustration of MIMO radar system.

1.4 Literature Review

Localizing an object in the 2-D or 3-D space based on frequency only measurements has been widely discussed in the literature with different scenarios and different ways of intercepting the signals sent by or reflected from the object. A less complex scenario for frequency-based localization is that the source is stationary and the sensors are moving with known positions and velocities. Such a scenario has found many applications related to radar and indoor localizations [76, 77, 78]. Some considered using only one moving receiver that intercepts the transmitted signal at multiple short intervals along the receiver's trajectory and others investigate several moving receivers

with each of them intercepts the transmitted signal only once for a one-time solution. Many solutions of different complexity and performance have been developed over the years and encouraging results have been reported [79, 80, 56, 81, 82, 11]. Numerical search solutions are proposed with different levels of computational complexities [79, 80, 56, 76, 78], some of these argued that their approach can work better than two-step algebraic solutions under high level noise situations [80, 76]. A low cost CFS is proposed by [59] for a proactive scenario to localize an underwater vehicle using only one moving sensor. In [81], another CFS estimator is proposed with special treatment to the bias problem associated with the pseudo linearization method. [82] derives two solutions based on CFS and SDR and provides treatment to the sensor locations uncertainty.

Another scenario is that the source is moving and the sensors are stationary. The source velocity is an unknown parameter in addition to the position. 2-D localization by DSF under this situation was considered in [83, 84, 85] by using computationally demanding numerical grid search to reach a solution. To reduce the computational complexity, [83] took the rate of change in DSF as additional measurements such that the search dimension is three. Besides the low computational efficiency of this method, it uses the DSF rate, which is not a very reliable quantity and rarely to be used. [85] presented a different formulation by introducing some intermediate variables, which are products of some of the unknowns without using the rate of frequency change and the search dimension is two. The method can be easily developed to 3-D localization with an increment of one in the search dimension. The method suffers from ghost solutions when a limited number of measurements are used. [84] addressed the problem by using the extra measurement of DSF derivative in addition

to DSF, where one dimensional grid search is sufficient for localization in 2-D by three sensors. [86] extended the application to a MIMO scenario for the tracking of a moving object through the signal reflections associated with different transmitter and receiver combination pairs. [87] discusses the ability to locate multiple moving targets using several fixed receivers in the 2-D space. The approach was to do an extensive four dimensional grid search to find all the points that regenerate the received frequency. This procedure is repeated for each one of the frequencies measured at the different receivers. The locations of targets can then be estimated from the intersection points of the surfaces generated by each receiver with the help of pattern recognition techniques.

Recently, [88] expressed the DSF measurement model in a polynomial optimization form and applied a software optimization package to obtain a solution. This method assumes the source frequency is known and the performance is suboptimal as it did not include proper weightings for the error and the algorithm will need to convert the polynomial model to a sum of square and then to a semi-definite programming (SDP) formulation.

1.5 Contribution of the Research

This research advances further the method of locating a moving source by DSF measurements. Different from the previous methods where exhaustive grid search or numerical optimization is necessary, we propose a novel formulation to the problem that enables the development of a computationally attractive algebraic closed-form solution or a noise tolerant convex optimization solution. Furthermore, we provide

comprehensive treatment to the problem where the source frequency has errors or is unknown, sensor position errors can be present, successive-time measurements are available to improve localization, and sensors are moving along nonlinear trajectory with random speed.

At the beginning, we consider the less complex scenario where the localization geometry is 2-D. The development starts assuming the source frequency is known but subject to random errors. We propose a novel nonlinear transformation of the frequency measurement model to a pseudo-linear form by introducing some nuisance variables. The localization problem is then converted to a quadratic optimization under a set of quadratic constraints relating the independent unknowns and the auxiliary variables. Solutions to the constrained optimization are derived using the linear optimization method that leads to an algebraic closed-form solution or the semi-definite relaxation (SDR) technique that yields a noise-resilient solution. The algebraic solution is analyzed and shown to reach the optimum accuracy level described by the Cramer-Rao Lower Bound (CRLB) over the small error region under Gaussian noise. Both solutions are next extended to the situation that the frequency measurements at successive times are available to improve performance, where the source is in linear motion. After that, we consider the non-cooperative scenario in which the source frequency is completely not known. Taking advantage of the solution for the case of known but erroneous source frequency, a new approach is proposed for joint estimation of the source frequency, position and velocity that is computationally attractive and able to achieve the CRLB performance.

Then, we consider the development to 3-D positioning scenario where the measurement model is more complex as the number of unknown variables increased by two

compared with the 2-D case. Assuming the carrier frequency is inaccurately known, we transformed the nonlinear DSF equation to a pseudo linear one by introducing some nuisance variables and then, we built the quadratic cost function with several quadratic constraints that relate the independent unknowns and their polynomials. Two solutions are proposed to minimize the constrained cost function and deliver the object location estimate. The first one is by using a linear algebraic method that yields a CFS, and the second one is by transforming the cost function to an SDP problem and solve it using the SDR technique. Then, we extended the methods to the multiple time measurements model in which each sensor collects several measurements during the non-maneuvering status of the object. Also, we discussed the case when the carrier frequency is completely unknown and derive a new estimator that jointly estimates the frequency and the object location by making use of the two solutions to the case of the inaccurately known carrier frequency.

Later, we investigate the moving sensors scenario where each sensor moves along nonlinear trajectory with random speed and collects frequency measurement from the object. Based on the same derivation approach, the CFS and SDP solutions are derived for 2-D single-time, 2-D multiple-time, 3-D single-time and 3-D multiple-time localization cases. Analysis of the algebraic solution is done to obtain the additional conditions, due to the sensor movement, that are needed to be fulfilled for achieving optimum accuracy.

The CFS is proved theoretically and by simulation to reach the CRLB accuracy under a low level of Gaussian noise. The SDP can tolerate higher noise levels and work with a fewer number of measurements. Although the 2-D and 3-D solutions idea is the same, the derivations for the 3-D case have a lot of differences with the 2-D one.

These differences came from the fact that the CFS and the SDR derivations highly depend on the number of unknown polynomial terms and the relations among. The details of 2-D and 3-D localization problems and the proposed solutions are presented on Chapter 3 and Chapter 4 respectively. In Chapter 5, we presented the solutions when the sensors are moving.

1.6 Content Organization

The dissertation is organized as follows. Chapter 2 introduces the problem scenario and several existed solutions from the literature. It also includes the derivation of the iterative maximum likelihood and the Cramer-Rao lower bound (CRLB) for the problem. Chapter 3 explains the localization problem in 2-D case and the proposed solutions. Chapter 4 extends the investigations and the proposed solutions to the 3-D case. Chapter 5 presents the scenario of moving sensors. Chapter 6 gives the conclusion of this research.

Chapter 2

Background

This chapter introduces the localization problem that we are going to investigate in this research and summarizes other researchers works that have been done on closely related problems. In addition, we shall derive a generic maximum likelihood estimator that can be used for different scenarios of the problem, and also derive the Cramer-Rao lower bound that will be used to evaluate the performance of different solution methods.

2.1 Localization Scenario

Fig. 2.1 illustrates the localization scenario. An object starting at unknown position $\mathbf{u}^o \in \mathbb{R}^d$ in the d -dimensional space is traveling at an unknown velocity $\dot{\mathbf{u}}^o \in \mathbb{R}^d$. It emits a tonal at frequency f_o^o during the linear trajectory. The emitted signal is received by M spatially distributed sensors at known positions $\mathbf{s}_i^o \in \mathbb{R}^d$, $i = 1, 2, \dots, M$, in each of N consecutive time steps. The observation period is not long, so that the

velocity remains the same over the duration. Each sensor is capable of determining the received signal frequency, giving a total of MN frequency observations. We shall consider the time step between every two successive observations as unity for simplicity where the actual time step can be absorbed into $\dot{\mathbf{u}}^o$ as a multiplication factor. For 2-D localization, we define $\mathbf{u}^o = [x^o, y^o]^T$, $\dot{\mathbf{u}}^o = [\dot{x}^o, \dot{y}^o]^T$ and $\mathbf{s}_i^o = [x_i^o, y_i^o]^T$ whereas for 3-D localization, we use $\mathbf{u}^o = [x^o, y^o, z^o]^T$, $\dot{\mathbf{u}}^o = [\dot{x}^o, \dot{y}^o, \dot{z}^o]^T$ and $\mathbf{s}_i^o = [x_i^o, y_i^o, z_i^o]^T$. In each observation time k , the object is at the unknown position

$$\mathbf{u}_k^o = \mathbf{u}^o + k\dot{\mathbf{u}}^o, \quad k = 0, 1, \dots, N-1. \quad (2.1)$$

The observed frequency in sensor i at instant k is [80]

$$f_{k,i} = f_o^o - \frac{f_o^o(\mathbf{u}_k^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|} + n_{k,i}, \quad (2.2)$$

where c is the speed of signal propagation and $n_{k,i}$ is the observation noise. Putting together the frequency measurements from all M sensors at time k gives the vector

$$\mathbf{f}_k = [f_{k,1}, f_{k,2}, \dots, f_{k,M}]^T = \mathbf{f}_k^o + \mathbf{n}_k, \quad (2.3)$$

and \mathbf{f}_k^o is the true value of \mathbf{f}_k without noise. \mathbf{n}_k is modeled by a zero-mean Gaussian vector with covariance matrix \mathbf{Q}_k . Over the N observation times, we have

$$\mathbf{f} = [\mathbf{f}_0^T, \mathbf{f}_1^T, \dots, \mathbf{f}_{N-1}^T]^T = \mathbf{f}^o + \mathbf{n}. \quad (2.4)$$

The covariance matrix of \mathbf{n} is denoted by \mathbf{Q}_n , which is block diagonal with diagonal blocks \mathbf{Q}_k , $k = 0, 1, \dots, N-1$, assuming the noise is uncorrelated at different times.

The carrier frequency f_o is not fully known. There is some knowledge from the past so that the known value f_o is modeled by

$$f_o = f_o^o + \Delta f_o, \quad (2.5)$$

and the error Δf_o follows a zero-mean Gaussian distribution with variance $\sigma_{f_o}^2$.

The available sensor positions are also not accurate and they are corrupted by the additive noise $\Delta \mathbf{s}_i$,

$$\mathbf{s}_i = \mathbf{s}_i^o + \Delta \mathbf{s}_i. \quad (2.6)$$

The sensor position vector is

$$\mathbf{s} = [\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_M^T]^T = \mathbf{s}^o + \Delta \mathbf{s}. \quad (2.7)$$

$\Delta \mathbf{s}$ is zero-mean Gaussian distributed with the covariance matrix equal to \mathbf{Q}_s . For simplicity, the measurement noise, carrier frequency error and sensor position errors are uncorrelated and the covariance matrices \mathbf{Q}_n and \mathbf{Q}_s and the variance $\sigma_{f_o}^2$ are all assumed known.

The localization scenario described is for the general case. In the particular situation that the measurements are from a single time instant only, N is equal to 1. In addition, if the carrier frequency is completely not known, $\sigma_{f_o}^2$ will correspond to ∞ . \mathbf{Q}_s will be zero if the sensor positions are accurate.

The objective is that given the MN frequency measurements together with the available carrier frequency f_o and sensor positions \mathbf{s} , we would like to estimate the object initial position \mathbf{u}^o and its velocity $\dot{\mathbf{u}}^o$. In the situation where multiple frequency observations over a time period are used, the problem is indeed more appropriate to

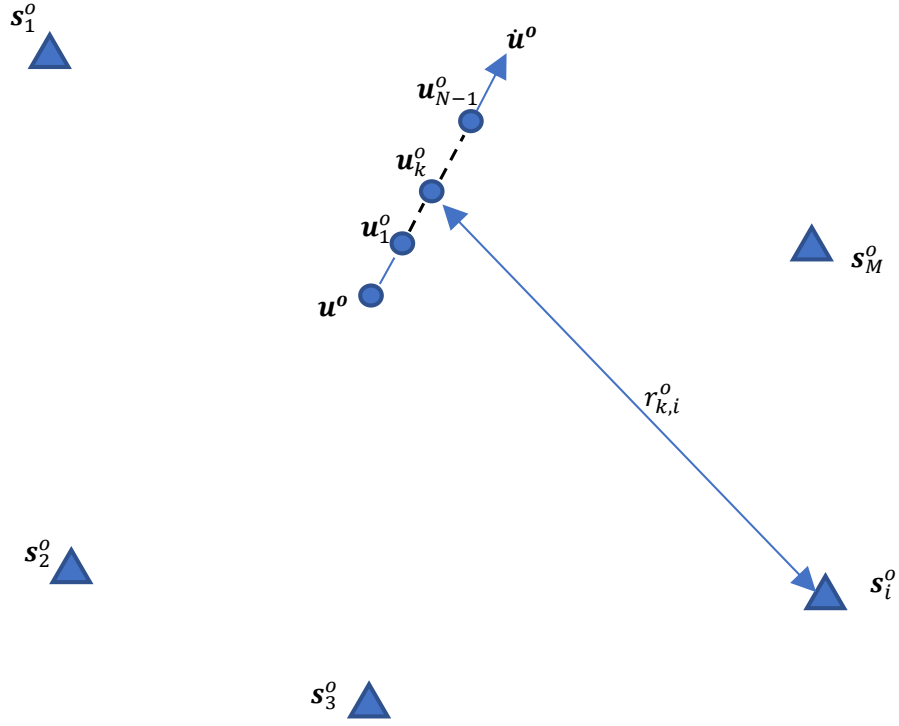


Figure 2.1: Localization scenario of a moving object.

be referred to as target motion analysis (TMA).

2.2 Chan Method

In [85], Chan assumes the carrier frequency is unknown, the geometry of localization is 2-D and the scenario is single-time measurement. Since the observation model described in (2.2) is highly nonlinear, the numerical grid search over some of the unknowns is considered. Usually, It would require five dimensional search to solve the problem, which is computationally expensive. Instead, Chan proposed introducing

intermediate variables that are products of some of the unknowns to reduce the search dimension to only two, over the position coordinates. The resultant intermediate model becomes linear and solvable through the ordinary least square (LS). For each grid point, the corresponding intermediate variables can be computed and then, the measurement vector is reconstructed and compared with the actual one obtained from the sensors. Among the trial grid points, the one that gives the best fit is selected for the object location estimate.

We shall present Chan's solution for the unknown carrier frequency case, followed by the modified solution for the known carrier frequency case. The solutions can be applied for both 2-D and 3-D localization when the variables are defined accordingly.

2.2.1 Carrier Frequency Unknown

The measurement model in (2.2) can be expressed in vector form as

$$\mathbf{H}^o \begin{bmatrix} f_o^o & (\dot{\mathbf{u}}^o f_o^o)^T \end{bmatrix}^T = \mathbf{f} - \mathbf{n}, \quad (2.8)$$

where \mathbf{H}^o is $M \times (d + 1)$ matrix constructed as

$$\mathbf{H}^o = \begin{bmatrix} 1 & -\frac{(\mathbf{u}^o - \mathbf{s}_1^o)^T}{c \|\mathbf{u}^o - \mathbf{s}_1^o\|} \\ 1 & -\frac{(\mathbf{u}^o - \mathbf{s}_2^o)^T}{c \|\mathbf{u}^o - \mathbf{s}_2^o\|} \\ \vdots & \vdots \\ 1 & -\frac{(\mathbf{u}^o - \mathbf{s}_M^o)^T}{c \|\mathbf{u}^o - \mathbf{s}_M^o\|} \end{bmatrix}. \quad (2.9)$$

The LS solution of f_o° and $\dot{\mathbf{u}}^\circ f_o^\circ$ is

$$\begin{bmatrix} f_o \\ \dot{\mathbf{u}} f_o \end{bmatrix} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{f}, \quad (2.10)$$

where \mathbf{H} is evaluated at the trial object position \mathbf{u} and the available sensor positions \mathbf{s} . The reconstructed measurement vector is

$$\hat{\mathbf{f}} = \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{f}. \quad (2.11)$$

Subtracting \mathbf{f} from (2.11) will give the error vector associated with that particular grid point and the cost function of it is

$$j(\mathbf{u}) = \left\| \hat{\mathbf{f}} - \mathbf{f} \right\|^2 = \mathbf{f}^T (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) \mathbf{f}. \quad (2.12)$$

Minimizing (2.12) gives the solution to \mathbf{u}° and the corresponding object velocity $\dot{\mathbf{u}}^\circ$ is found from (2.10) with \mathbf{H} constructed at the solution of \mathbf{u}° .

2.2.2 Carrier Frequency Known

When the carrier frequency is known, the model is slightly changed to account only for the location parameters. Let d_i be the range rate between the object and sensor i expressed as

$$d_i = c(f_o - f_i)/f_o = \frac{(\mathbf{u}^\circ - \mathbf{s}_i^\circ)^T \dot{\mathbf{u}}^\circ}{\|\mathbf{u}^\circ - \mathbf{s}_i^\circ\|} - \frac{c}{f_o} n_i, \quad (2.13)$$

and collecting the rang rates of all sensors in one vector to get

$$\mathbf{d} = [d_1, d_2, \dots, d_M]^T = \tilde{\mathbf{H}}^o \dot{\mathbf{u}}^o - \frac{c}{f_o} \mathbf{n}, \quad (2.14)$$

where $\tilde{\mathbf{H}}^o$ is $M \times d$ matrix constructed as

$$\tilde{\mathbf{H}}^o = \left[\frac{(\mathbf{u}^o - \mathbf{s}_1^o)}{\|\mathbf{u}^o - \mathbf{s}_1^o\|}, \frac{(\mathbf{u}^o - \mathbf{s}_2^o)}{\|\mathbf{u}^o - \mathbf{s}_2^o\|}, \dots, \frac{(\mathbf{u}^o - \mathbf{s}_M^o)}{\|\mathbf{u}^o - \mathbf{s}_M^o\|} \right]^T. \quad (2.15)$$

The cost function at any object position \mathbf{u} in this case will be

$$\tilde{\mathcal{J}}(\mathbf{u}) = \mathbf{d}^T \left(\mathbf{I} - \tilde{\mathbf{H}}(\tilde{\mathbf{H}}^T \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \right) \mathbf{d}. \quad (2.16)$$

Thus, the estimation of object position is found from the grid search process that minimizes (2.16) and the velocity of object is determined by the following LS operation,

$$\dot{\mathbf{u}} = (\tilde{\mathbf{H}}^T \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{d}. \quad (2.17)$$

2.3 Shames Method

Shames explained the scenario for locating a moving target using several fixed sensors working on the active mode based on the frequency shift of the received signals [88]. Each sensor transmits a signal with a distinct carrier frequency $f_{c,i}$, $i = 1, 2, \dots, M$, and measure the Doppler shift of the reflected one. The localization here is to be achieved instantaneously without envisioning collecting information from sensors at

a number of successive time instants. The Doppler shift measured by sensor i is

$$w_i = 2 \frac{f_{c,i} (\mathbf{u}^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o}{c \|\mathbf{u}^o - \mathbf{s}_i^o\|} - e_i, \quad (2.18)$$

where the constant 2 comes from the two-way signal propagation and e_i is the measurement noise. Since c and $f_{c,i}$ are constants and assumed known; it is more convenient to deal with the range rate quantity similar to the one described in (2.13),

$$d_i = \frac{(\mathbf{u}^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o}{\|\mathbf{u}^o - \mathbf{s}_i^o\|} - \frac{c}{2 f_{c,i}} e_i, \quad (2.19)$$

and d_i here is $c w_i / (2 f_{c,i})$, $i = 1, 2, \dots, M$. To solve for the unknowns, Shames converted (2.19) to a polynomial equation (excluding the noisy terms) by multiplying its both sides by $\|\mathbf{u}^o - \mathbf{s}_i^o\|$ and then squaring it to get

$$(\mathbf{u}^o - \mathbf{s}_i^o)^T (\mathbf{u}^o - \mathbf{s}_i^o) d_i^2 = ((\mathbf{u}^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o)^2 + \frac{c \|\mathbf{u}^o - \mathbf{s}_i^o\|}{f_{c,i}} \left(\frac{c \|\mathbf{u}^o - \mathbf{s}_i^o\|}{f_{c,i}} e_i - (\mathbf{u}^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o \right) e_i. \quad (2.20)$$

The second term on the right side of (2.20) is only noise and the idea is to find \mathbf{u} and $\dot{\mathbf{u}}$ that make this noise as small as possible. Thus, the solution to (2.20) can be found by minimizing the following cost function

$$j(\mathbf{u}^o, \dot{\mathbf{u}}^o) = \sum_{i=1}^M \left((\mathbf{u}^o - \mathbf{s}_i^o)^T (\mathbf{u}^o - \mathbf{s}_i^o) d_i^2 - ((\mathbf{u}^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o)^2 \right)^2. \quad (2.21)$$

(2.21) is polynomial in the unknowns and can be minimized using some modern polynomial optimization packages such as GloptiPoly 3 [89]. The typical approach followed in the polynomial optimization packages is to first transform the sum of

squared polynomial functions to a hierarchy of semidefinite programming (SDP) relaxations whose associated optimal values converge to the global optimum and then solve them using the available SDP solvers like SeDuMi [90] and SDPT3 [91]. The size of the hierarchical SDP-relaxations grows fast with the number of unknowns and the order of the polynomials, which limits the applicability of the approach to only small or medium size problems [92].

The target location estimate proposed by Shames is

$$[\mathbf{u}^T, \dot{\mathbf{u}}^T]^T = \min_{\mathbf{u}, \dot{\mathbf{u}}} j(\mathbf{u}^o, \dot{\mathbf{u}}^o). \quad (2.22)$$

Any solution of (2.19) is also a solution of (2.22) but not vice versa as both $\dot{\mathbf{u}}^o$ and $-\dot{\mathbf{u}}^o$ give the same cost value in (2.21). To remove the sign ambiguity of the velocity parameters, Shames used the same LS formula described in (2.17) with \mathbf{d} constructed from the d_i 's defined below (2.19) and $\tilde{\mathbf{H}}$ is found using (2.15) with \mathbf{u} obtained from the solution of (2.22).

The solution presented by Shames method can be used for both 2-D and 3-D localization; however, for 3-D, the number of polynomial terms in (2.21) will be excessive and GloptiPoly 3 will take much longer time to converge, as we shall see in chapter 4.

2.4 Semidefinite Programming Method

Semidefinite programming (SDP) is an extension of linear programming (LP) and share many features and characteristics of the corresponding LP algorithms, such as the duality theory and solvability in polynomial time. Typically, it intends to mini-

mize a linear objective function subject to nonlinear, but convex, matrix inequality constraint. SDP can also be viewed as a special case of cone programming, which has useful applications in combinatorial optimization, control theory, statistics and others. The matrix inequality constraint is written in the form that all eigenvalues of the matrix are non-negative; thus the matrix is positive semidefinite. Mathematically, we can express the SDP problem as [93]

$$\begin{aligned}
& \text{minimize} && \mathbf{A} \bullet \Phi, \\
& \text{s.t.} && \mathbf{L}_j \bullet \Phi = h_j, \quad j = 1, 2, \dots, q, \\
& && \Phi \succeq \mathbf{0},
\end{aligned} \tag{2.23}$$

where Φ is the variable matrix that must lie in the cone of positive semidefinite symmetric matrices, $\mathbf{A} \bullet \Phi = \sum_{a=1}^{\nu} \sum_{b=1}^{\nu} A(a, b) \Phi(a, b)$ is the linear objective function that need to be minimized and $\mathbf{L}_j \bullet \Phi = h_j, \quad j = 1, 2, \dots, q$ are linear equations of Φ that must be satisfied. Note that $\mathbf{A}, \mathbf{L}_1, \dots, \mathbf{L}_q$, and h_1, \dots, h_q are the data for the SDP problem and they are assumed known and constant.

The idea is to convert the localization model of (2.2) into SDP problem similar to (2.23) and solve it efficiently using some available SDP packages like CVX [94]. This approach will be further elaborated and explained in chapters 3, 4 and 5. Similar to this approach for other localization problems has been reached, to name but few [95, 96, 97, 43].

SDP solutions are robust and can achieve optimum accuracy under harsh localization environments in which the measurements are limited in number and/or highly disturbed by noise. However, on the other side, the computational complexity of SDP

limits its usages to only problems that are not highly sensitive to time. The famous interior-point method [98] that are commonly used to solve the SDP problems is an iterative method that depends mainly on solving a least-squares problem of the same size as the original problem at each iteration. Theoretically, the interior-point method according to the worst-case analysis, can solve the SDP problems to a given accuracy within time grows no faster than a polynomial of the problem size [99]. In practice, the observations to the behavior of the interior-point method are very optimistic as it shows much better computational efficiency than predicted by theoretical analysis [100].

2.5 Maximum Likelihood Estimator(MLE)

The idea of the MLE estimator is to choose a value in the parameter space that maximizes the likelihood for a data set as the estimate of the unknown parameter. Given a set of data \mathbf{g} , the likelihood function for an unknown parameter vector ϕ^o is denoted by $\mathcal{L}(\phi^o; \mathbf{g})$. The maximum likelihood estimation is to choose the estimate of ϕ^o at which the value of \mathbf{g} is most likely to happen.

The method of maximum likelihood is very well-known in statistics. In general, for a fixed set of data with a specific statistical model, the method of maximum likelihood finds a set of values for the model parameters that maximize the probability of observed data under its statistical distribution. For our localization problem, we define the data vector \mathbf{g} as

$$\mathbf{g} = [\mathbf{f}^T, f_o, \mathbf{s}^T]^T, \quad (2.24)$$

and the unknown vector $\boldsymbol{\phi}^o$ as

$$\boldsymbol{\phi}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}, f_o^o, \mathbf{s}^{oT}]^T, \quad (2.25)$$

We assume that the measurement noise vector \mathbf{n} , the carrier frequency error Δf_o and the sensor position error vector $\Delta \mathbf{s}$ are independent and identically distributed such that the probability density function of \mathbf{g} can be given by

$$p(\mathbf{g}|\boldsymbol{\phi}^o) = p(\mathbf{f}|\boldsymbol{\phi}^o) \times p(f_o|f_o^o) \times p(\mathbf{s}|\mathbf{s}^o). \quad (2.26)$$

We also assume that \mathbf{f} , f_o and \mathbf{s} are normally distributed and their density functions are respectively given by

$$p(\mathbf{f}|\boldsymbol{\phi}^o) = \frac{1}{(2\pi|\mathbf{Q}_n|)^{\frac{MN}{2}}} \exp\left(-\frac{1}{2}(\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))^T \mathbf{Q}_n^{-1}(\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))\right), \quad (2.27)$$

$$p(f_o|f_o^o) = \frac{1}{\sqrt{2\pi}\sigma_{f_o}} \exp\left(-\frac{(f_o - f_o^o)^2}{2\sigma_{f_o}^2}\right), \quad (2.28)$$

$$p(\mathbf{s}|\mathbf{s}^o) = \frac{1}{(2\pi|\mathbf{Q}_s|)^{\frac{M}{2}}} \exp\left(-\frac{1}{2}(\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1}(\mathbf{s} - \mathbf{s}^o)\right). \quad (2.29)$$

Inserting (2.27), (2.28) and (2.29) in (2.26) and looking at it from different perspective by considering \mathbf{f} is fixed and $\boldsymbol{\phi}^o$ is variable, this equation is called the likelihood function,

$$\mathcal{L}(\boldsymbol{\phi}^o; \mathbf{g}) = (2\pi)^{\frac{-M(N+1)-1}{2}} |\mathbf{Q}_n|^{\frac{-MN}{2}} |\mathbf{Q}_s|^{\frac{-M}{2}} \sigma_{f_o}^{-1}$$

$$\times \exp \left(-\frac{(\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))}{2} - \frac{(f_o - f_o^o)^2}{2\sigma_{f_o}^2} - \frac{(\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o)}{2} \right). \quad (2.30)$$

To continue derivation, it is more convenient to deal with the log-likelihood function which is obtained by taking the natural logarithm of both sides of (2.30) as shown below,

$$\ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{g}) = \kappa - \frac{1}{2} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o)) - \frac{1}{2} \sigma_{f_o}^{-2} (f_o - f_o^o)^2 - \frac{1}{2} (\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o), \quad (2.31)$$

where κ is constant and does not depend on $\boldsymbol{\phi}^o$. Maximizing (2.31) is equivalent to minimizing the following cost function

$$\mathbb{C}(\boldsymbol{\phi}^o) = (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o))^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}^o)) + \sigma_{f_o}^{-2} (f_o - f_o^o)^2 + (\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o). \quad (2.32)$$

The idea is to find $\boldsymbol{\phi}^o$ that gives minimum cost function. The minimum point of the cost function is equivalent to the point where the slope of (2.32) is zero. Since $\mathbf{f}^o(\boldsymbol{\phi}^o)$ is nonlinearly related to $\boldsymbol{\phi}^o$, the MLE will not have direct closed-form solution. However, the MLE is still possible if there is an initial guess $\boldsymbol{\phi}_o$ sufficiently close to $\boldsymbol{\phi}^o$. The nonlinear function $\mathbf{f}^o(\boldsymbol{\phi}^o)$ can be linearized and then iterative method can be used to find the MLE. Consider $\boldsymbol{\phi}_o$ is a point sufficiently close to the point that gives minimum cost function value; the frequency measurement vector can be expanded

using Taylor series as

$$\mathbf{f}^o(\boldsymbol{\phi}^o) = \mathbf{f}^o(\boldsymbol{\phi}_o) + \mathbf{G}(\boldsymbol{\phi}_o)(\boldsymbol{\phi}^o - \boldsymbol{\phi}_o) + o(\|\boldsymbol{\phi}^o - \boldsymbol{\phi}_o\|), \quad (2.33)$$

where $\mathbf{G}(\boldsymbol{\phi}^o)$ is referred to the Jacobean matrix of $\mathbf{f}^o(\boldsymbol{\phi}^o)$ given by

$$\mathbf{G}(\boldsymbol{\phi}^o) = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\phi}^{oT}} = \left[\frac{\partial \mathbf{f}^o}{\partial \mathbf{u}^{oT}}, \frac{\partial \mathbf{f}^o}{\partial \dot{\mathbf{u}}^{oT}}, \frac{\partial \mathbf{f}^o}{\partial f_o}, \frac{\partial \mathbf{f}^o}{\partial \mathbf{s}^{oT}} \right]. \quad (2.34)$$

The partial derivatives in (2.34) can be found in Appendix A. Substituting (2.33) in (2.32) will result in

$$\begin{aligned} \mathbb{C}(\boldsymbol{\phi}^o) &= (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}_o) - \mathbf{G}(\boldsymbol{\phi}_o)(\boldsymbol{\phi}^o - \boldsymbol{\phi}_o))^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}_o) - \mathbf{G}(\boldsymbol{\phi}_o)(\boldsymbol{\phi}^o - \boldsymbol{\phi}_o)) \\ &\quad + \sigma_{f_o}^{-2} (f_o - f_o^o)^2 + (\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o) + o(\|\boldsymbol{\phi}^o - \boldsymbol{\phi}_o\|). \end{aligned} \quad (2.35)$$

Taking the derivative of the right side of (2.35) with respect to $\boldsymbol{\phi}^o$ and equating it to zero give

$$\begin{aligned} \mathbf{0} &= -2\mathbf{G}(\boldsymbol{\phi}_o)^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}_o) + \mathbf{G}(\boldsymbol{\phi}_o)\boldsymbol{\phi}_o - \mathbf{G}(\boldsymbol{\phi}_o)\boldsymbol{\phi}^o) - 2\sigma_{f_o}^{-2} \mathbf{b}_{f_o}^T (f_o - \mathbf{b}_{f_o}\boldsymbol{\phi}^o) \\ &\quad - 2\mathbf{B}_s^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{B}_s\boldsymbol{\phi}^o) + o(\|\boldsymbol{\phi}^o - \boldsymbol{\phi}_o\|), \end{aligned} \quad (2.36)$$

where \mathbf{b}_{f_o} and \mathbf{B}_s are the partial derivatives of f_o^o and \mathbf{s}^o with respect to $\boldsymbol{\phi}^o$ respectively and both are detailed in Appendix A. Neglecting the nonlinear terms of $\|\boldsymbol{\phi}^o - \boldsymbol{\phi}_o\|$ and solving (2.36) for $\boldsymbol{\phi}^o$ will give the iterative MLE solution.

$$\boldsymbol{\phi}_{l+1} = (\mathbf{G}(\boldsymbol{\phi}_l)^T \mathbf{Q}_n^{-1} \mathbf{G}(\boldsymbol{\phi}_l) + \sigma_{f_o}^{-2} \mathbf{b}_{f_o}^T \mathbf{b}_{f_o} + \mathbf{B}_s^T \mathbf{Q}_s^{-1} \mathbf{B}_s)^{-1}$$

$$\times (\mathbf{G}(\boldsymbol{\phi}_l)^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o(\boldsymbol{\phi}_l) + \mathbf{G}(\boldsymbol{\phi}_l) \boldsymbol{\phi}_l) + \sigma_{f_o}^{-2} \mathbf{b}_{f_o}^T f_o + \mathbf{B}_s^T \mathbf{Q}_s^{-1} \mathbf{s}) . \quad (2.37)$$

The MLE in (2.37) is for the general case in which the localization geometry can be 2-D or 3-D, the scenario is for single-time or multiple-time measurements, the carrier frequency is available or unavailable and the sensor positions are accurate or have error. We set $\sigma_{f_o}^{-2}$ to zero when the carrier frequency is unavailable. When Δf_o or $\Delta \mathbf{s}$ are absent, we set their corresponding variances to very small values to avoid the computation of zero inverse.

2.6 CRLB

We shall establish the CRLB for the localization problem and use it as a benchmark to examine the performance of different estimators. The object location vector for estimation is

$$\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T. \quad (2.38)$$

The nuisance variable vector is $\boldsymbol{\alpha}^o = [f_o^o, \mathbf{s}^{oT}]^T$. The parameter vector for evaluating the CRLB is the two together, $\boldsymbol{\phi}^o = [\boldsymbol{\theta}^{oT}, \boldsymbol{\alpha}^{oT}]^T$. Considering the observation vector is $\mathbf{g} = [\mathbf{f}^T, f_o, \mathbf{s}^T]^T$, the logarithm of the probability density function under the Gaussian data model is

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{g}) &= \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{f}) + \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{s}) + \ln \mathcal{L}(\boldsymbol{\phi}^o; f_o) \\ &= \kappa - \frac{1}{2} (\mathbf{f} - \mathbf{f}^o)^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o) - \frac{1}{2} \sigma_{f_o}^{-2} (f_o - f_o^o)^2 - \frac{1}{2} (\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o), \end{aligned} \quad (2.39)$$

where κ is a constant not dependent on $\boldsymbol{\phi}^o$, and \mathbf{f}^o is a function of $\boldsymbol{\phi}^o$ implicitly. Applying matrix inversion on the expectation after taking derivatives with respect to $\boldsymbol{\phi}^o$ twice, the CRLB can be partitioned into a 2×2 block matrix with the blocks corresponding to the estimation performance for $\boldsymbol{\theta}^o$ and $\boldsymbol{\alpha}^o$,

$$\text{CRLB}(\boldsymbol{\phi}^o) = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\phi}^o \partial \boldsymbol{\phi}^{oT}} \right]^{-1} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y}^T & \mathbf{Z} \end{bmatrix}^{-1}. \quad (2.40)$$

The blocks are

$$\mathbf{X} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\theta}^o \partial \boldsymbol{\theta}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\theta}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (2.41a)$$

$$\mathbf{Y} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\theta}^o \partial \boldsymbol{\alpha}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\theta}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}}, \quad (2.41b)$$

$$\mathbf{Z} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\alpha}^o \partial \boldsymbol{\alpha}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\alpha}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}} + \mathbf{Q}_\alpha^{-1}, \quad (2.41c)$$

and $\mathbf{Q}_\alpha = \text{diag}(\sigma_{f_o}^2, \mathbf{Q}_s)$. Appendix A gives the expressions of the partial derivatives in (2.41). Invoking the partitioned matrix inversion formula [101] yields from the upper left block

$$\text{CRLB}(\boldsymbol{\theta}^o) = (\mathbf{X} - \mathbf{Y}\mathbf{Z}^{-1}\mathbf{Y}^T)^{-1} = \mathbf{X}^{-1} + \mathbf{X}^{-1}\mathbf{Y}(\mathbf{Z} - \mathbf{Y}^T\mathbf{X}^{-1}\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}^{-1}. \quad (2.42)$$

Recognizing \mathbf{X}^{-1} is the CRLB in the absence of carrier frequency and sensor position errors, the second term is the performance loss resulted from the presence of Δf_o and $\Delta \mathbf{s}$.

2.6.1 Summary

This chapter explains the scenario of the localization problem that we are going to discuss in this research and define the mathematical model with all the variables and constants associated with it. The chapter also presents a summary of several solution methods that have been proposed by other researchers, along with some adjustments for the scenarios to match the model of our problem. After that, we derived the maximum likelihood estimator that can iteratively solve the problem when an initial start point close to the actual object location is available. We end the chapter by deriving the Cramér-Rao lower bound, which is an important benchmark for the variance of any unbiased estimator.

Chapter 3

2-D Object Localization

3.1 Formulation

The unknowns \mathbf{u}^o and $\dot{\mathbf{u}}^o$ are coupled together and related to the frequency observations in a highly nonlinear and complicated fashion in the measurement model (2.2). In order to solve for the unknowns, we shall derive a formulation of the localization problem under the assumption that the errors are not significant in which the second and higher order error terms are negligible. The proposed formulation can yield an object location estimate by algebraic evaluation for a closed-form solution or by SDR for a convex optimization solution. The cases of available and unavailable carrier frequency, and single-time and multiple-time measurements will be addressed separately.

3.1.1 Carrier Frequency Available

The available carrier frequency is related to the actual by (2.5). We shall express the frequency measurement model (2.2) in terms of the inaccurate carrier frequency and the available sensor positions. Let $r_{k,i} = \|\mathbf{u}_k^o - \mathbf{s}_i\|$ be the Euclidean distance between the object at instant k and the i -th sensor at noisy position \mathbf{s}_i , and $\boldsymbol{\rho}_{k,i} = (\mathbf{u}_k^o - \mathbf{s}_i)/r_{k,i}$ be the unit length vector pointing from \mathbf{s}_i to \mathbf{u}_k^o . Applying the Taylor-series expansion with respect to $\Delta\mathbf{s}_i$ gives

$$\frac{1}{\|\mathbf{u}_k^o - (\mathbf{s}_i - \Delta\mathbf{s}_i)\|} = \frac{1}{r_{k,i}} - \frac{1}{r_{k,i}^3}(\mathbf{u}_k^o - \mathbf{s}_i)^T \Delta\mathbf{s}_i + o(\|\Delta\mathbf{s}_i\|). \quad (3.1)$$

Using (2.5)-(2.6) and (3.1), (2.2) becomes

$$f_{k,i} = f_o - \frac{f_o(\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o}{c r_{k,i}} + \varepsilon_{k,i} + \Delta f_o o(1) + o(\|\Delta\mathbf{s}_i\|). \quad (3.2)$$

The composite noise term $\varepsilon_{k,i}$ is

$$\varepsilon_{k,i} = \left(\frac{\boldsymbol{\rho}_{k,i}^{oT} \dot{\mathbf{u}}^o}{c} - 1 \right) \Delta f_o + \frac{-f_o}{c r_{k,i}^o} \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o} \Delta\mathbf{s}_i + n_{k,i}, \quad (3.3)$$

where $\mathbf{P}_{k,i}^{\perp o} = \mathbf{I} - \boldsymbol{\rho}_{k,i}^o \boldsymbol{\rho}_{k,i}^{oT}$ is the orthogonal projection matrix of $\boldsymbol{\rho}_{k,i}^o$. We shall define $\mathbf{d}_{f,k}$ as the $M \times 1$ vector whose i -th element is $-1 + \boldsymbol{\rho}_{k,i}^{oT} \dot{\mathbf{u}}^o / c$, and $\mathbf{D}_{s,k}$ as the $M \times 2M$ matrix whose i -th row is zero except the elements $\mathbf{D}_{s,k}(i, 2(i-1) + 1 : 2i) = -\dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o} f_o / (c r_{k,i}^o)$. Collecting the composite errors from different sensors at time k gives

$$\boldsymbol{\varepsilon}_k = \mathbf{d}_{f,k} \Delta f_o + \mathbf{D}_{s,k} \Delta\mathbf{s} + \mathbf{n}_k. \quad (3.4)$$

Defining the vector $\mathbf{d}_f = [\mathbf{d}_{f,0}^T, \mathbf{d}_{f,1}^T, \dots, \mathbf{d}_{f,N-1}^T]^T$ and the matrix $\mathbf{D}_s = [\mathbf{D}_{s,0}^T, \mathbf{D}_{s,1}^T, \dots, \mathbf{D}_{s,N-1}^T]^T$, the error vector of all MN measurements $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_0^T, \dots, \boldsymbol{\varepsilon}_{N-1}^T]^T$ is

$$\boldsymbol{\varepsilon} = \mathbf{d}_f \Delta f_o + \mathbf{D}_s \Delta \mathbf{s} + \mathbf{n}. \quad (3.5)$$

$\boldsymbol{\varepsilon}$ remains Gaussian and has the covariance matrix equal to

$$\mathbf{Q}_\varepsilon = E[\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T] = \sigma_{f_o}^2 \mathbf{d}_f \mathbf{d}_f^T + \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{Q}_n. \quad (3.6)$$

To proceed further, it is more convenient to work with a scaled version of the normalized Doppler shift

$$d_{k,i} = c(f_{k,i} - f_o)/f_o. \quad (3.7)$$

It has the meaning of the object range rate at time k observed by sensor i . Rearranging (3.2) gives

$$(d_{k,i} - \varepsilon_{k,i} c/f_o) r_{k,i} = -(\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|). \quad (3.8)$$

Recall that $r_{k,i} = \|\mathbf{u}_k^o - \mathbf{s}_i\|$ which is defined before (3.1), squaring both sides yields

$$\begin{aligned} -2(c/f_o) d_{k,i} r_{k,i}^2 \varepsilon_{k,i} + d_{k,i}^2 (\|\mathbf{s}_i\|^2 - 2\mathbf{s}_i^T \mathbf{u}_k^o + \|\mathbf{u}_k^o\|^2) &= (\mathbf{s}_i^T \dot{\mathbf{u}}^o)^2 - 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}_k^o \\ &+ (\mathbf{u}_k^{oT} \dot{\mathbf{u}}^o)^2 + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|). \end{aligned} \quad (3.9)$$

In terms of coordinate components, we have

$$(\mathbf{s}_i^T \dot{\mathbf{u}}^o)^2 = x_i^2 \dot{x}^{o2} + y_i^2 \dot{y}^{o2} + 2x_i y_i \dot{x}^o \dot{y}^o, \quad (3.10)$$

and (3.9) turns into

$$2(c/f_o)d_{k,i}r_{k,i}^2\varepsilon_{k,i} + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|) = d_{k,i}^2 \|\mathbf{s}_i\|^2 - 2d_{k,i}^2 \mathbf{s}_i^T \mathbf{u}_k^o + d_{k,i}^2 \|\mathbf{u}_k^o\|^2 - x_i^2 \dot{x}^{o2} - y_i^2 \dot{y}^{o2} - 2x_i y_i \dot{x}^o \dot{y}^o + 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}_k^o - (\mathbf{u}_k^{oT} \dot{\mathbf{u}}^o)^2. \quad (3.11)$$

\mathbf{u}_k^o is the object position at the time k . It is dependent on \mathbf{u}^o and $\dot{\mathbf{u}}^o$ only. Expressing it by (2.1) leads (3.11) to

$$2(c/f_o)d_{k,i}r_{k,i}^2\varepsilon_{k,i} + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|) = d_{k,i}^2 \|\mathbf{s}_i\|^2 - 2d_{k,i}^2 \mathbf{s}_i^T \mathbf{u}^o - 2nd_{k,i}^2 \mathbf{s}_i^T \dot{\mathbf{u}}^o + d_{k,i}^2 \|\mathbf{u}^o\|^2 + 2nd_{k,i}^2 \mathbf{u}^{oT} \dot{\mathbf{u}}^o + (k^2 d_{k,i}^2 - x_i^2) \dot{x}^{o2} + (k^2 d_{k,i}^2 - y_i^2) \dot{y}^{o2} - 2x_i y_i \dot{x}^o \dot{y}^o + 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}^o + 2n\mathbf{s}_i^T \dot{\mathbf{u}}^o \|\dot{\mathbf{u}}^o\|^2 - (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 - k^2 \|\dot{\mathbf{u}}^o\|^4 - 2n\mathbf{u}^{oT} \dot{\mathbf{u}}^o \|\dot{\mathbf{u}}^o\|^2. \quad (3.12)$$

(3.12) is a rather involved nonlinear equation for the unknowns. The following considers the observations from single-time snapshot first and continues for multiple instants next.

Single-Time Observation

Each sensor has only one measurement. Setting $k = 0$ and dropping the time zero index for simplicity, (3.12) reduces to the simpler expression

$$2(c/f_o)d_i r_i^2 \varepsilon_i + o(\varepsilon_i) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|) = d_i^2 \|\mathbf{s}_i\|^2 - 2d_i^2 \mathbf{s}_i^T \mathbf{u}^o + d_i^2 \|\mathbf{u}^o\|^2 - x_i^2 \dot{x}^{o2} - y_i^2 \dot{y}^{o2} - 2x_i y_i \dot{x}^o \dot{y}^o + 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}^o - (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2. \quad (3.13)$$

(3.13) remains to be a highly nonlinear equation with respect to \mathbf{u}^o and $\dot{\mathbf{u}}^o$. We

shall formulate the localization problem as a constrained optimization. Let the unknown vector be

$$\boldsymbol{\varphi}^o = \left[\mathbf{u}^{oT}, \|\mathbf{u}^o\|^2, \dot{x}^{o2}, \dot{y}^{o2}, \dot{x}^o \dot{y}^o, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT}, (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 \right]^T. \quad (3.14)$$

Also, define the $M \times 9$ matrix

$$\mathbf{A} = \begin{bmatrix} 2d_1^2 \mathbf{s}_1^T & -d_1^2 & x_1^2 & y_1^2 & 2x_1 y_1 & -2\mathbf{s}_1^T & 1 \\ 2d_2^2 \mathbf{s}_2^T & -d_2^2 & x_2^2 & y_2^2 & 2x_2 y_2 & -2\mathbf{s}_2^T & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2d_M^2 \mathbf{s}_M^T & -d_M^2 & x_M^2 & y_M^2 & 2x_M y_M & -2\mathbf{s}_M^T & 1 \end{bmatrix}, \quad (3.15)$$

the $M \times 1$ vector

$$\mathbf{h} = [d_1^2 \|\mathbf{s}_1\|^2, d_2^2 \|\mathbf{s}_2\|^2, \dots, d_M^2 \|\mathbf{s}_M\|^2]^T, \quad (3.16)$$

and the $M \times M$ matrix

$$\mathbf{B} = 2 \frac{c}{f_o} \text{diag} \{ [d_1 r_1^2, d_2 r_2^2, \dots, d_M r_M^2] \}. \quad (3.17)$$

Over $i = 1, 2, \dots, M$, (3.13) forms the matrix equation after dropping the second and higher order error terms,

$$\mathbf{B} \boldsymbol{\varepsilon} \simeq \mathbf{h} - \mathbf{A} \boldsymbol{\varphi}^o. \quad (3.18)$$

The approximation is valid when the error is small. $\boldsymbol{\varphi}^o$ has 9 elements but the number of independent variables is only 4. Five constraints are necessary to relate the elements of the variable $\boldsymbol{\varphi}$ for the estimation of $\boldsymbol{\varphi}^o$. Based on $\boldsymbol{\varphi}^o$ defined in (3.14), Table 3.1

Table 3.1: Relations Among the Elements of $\boldsymbol{\varphi}$ for the Single-Time Measurement Case

Elements	Relations
$\varphi(1) = x$	$\varphi(3) = \varphi(1)^2 + \varphi(2)^2$
$\varphi(2) = y$	$\varphi(6)^2 = \varphi(4)\varphi(5)$
$\varphi(3) = x^2 + y^2$	$\varphi(7) = \varphi(1)\varphi(4) + \varphi(2)\varphi(6)$
$\varphi(4) = \dot{x}^2$	$\varphi(8) = \varphi(1)\varphi(6) + \varphi(2)\varphi(5)$
$\varphi(5) = \dot{y}^2$	$\varphi(9) = \varphi(1)\varphi(7) + \varphi(2)\varphi(8)$
$\varphi(6) = \dot{x}\dot{y}$	
$\varphi(7) = \dot{x}(x\dot{x} + y\dot{y})$	$\varphi(4)\varphi(9) = \varphi(7)\varphi(7)$
$\varphi(8) = \dot{y}(x\dot{x} + y\dot{y})$	$\varphi(5)\varphi(9) = \varphi(8)\varphi(8)$
$\varphi(9) = (x\dot{x} + y\dot{y})^2$	$\varphi(6)\varphi(9) = \varphi(7)\varphi(8)$

shows the individual components of $\boldsymbol{\varphi}$ and lists the relations among the elements.

Let the weighting matrix \mathbf{W} be an approximation of the covariance matrix inverse for the equation error of (3.18) that is equal to

$$\mathbf{W} = \mathbf{B}^{-T} E [\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]^{-1} \mathbf{B}^{-1} = \mathbf{B}^{-T} \mathbf{Q}_{\boldsymbol{\varepsilon}}^{-1} \mathbf{B}^{-1}, \quad (3.19)$$

where $\mathbf{Q}_{\boldsymbol{\varepsilon}}$ is given by (3.6) with $N = 1$. The localization problem can be cast as a weighted least-squares (WLS) optimization under a set of constraints as follows:

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (3.20a)$$

$$\text{s.t. } \varphi(3) = \varphi(1)^2 + \varphi(2)^2, \quad (3.20b)$$

$$\varphi(6)^2 = \varphi(4)\varphi(5), \quad (3.20c)$$

$$\varphi(7) = \varphi(1)\varphi(4) + \varphi(2)\varphi(6), \quad (3.20d)$$

$$\varphi(8) = \varphi(1)\varphi(6) + \varphi(2)\varphi(5), \quad (3.20e)$$

$$\varphi(9) = \varphi(1)\varphi(7) + \varphi(2)\varphi(8). \quad (3.20f)$$

The 5 quadratic constraints (3.20b)-(3.20f) come from the first five entries in the right column of Table 3.1. The remaining three relations in the Table are redundant in the formulation (3.20). They can be exploited to improve the tightness of the optimization when it is approximated with SDR.

The constrained optimization problem (3.20) will be solved using unconstrained minimization or convex optimization that will be described in Section 3.2.

Multiple-Time Observations

The transformed observation equation at a certain time instant k is (3.12). By collecting in each term the lumped variable involving the object position and velocity, we define the unknown vector as

$$\begin{aligned}\boldsymbol{\varphi}^o(1 : 13) &= [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}, \|\mathbf{u}^o\|^2, \mathbf{u}^{oT}\dot{\mathbf{u}}^o, \dot{x}^{o2}, \dot{y}^{o2}, \dot{x}^o\dot{y}^o, \mathbf{u}^{oT}\dot{\mathbf{u}}^o\dot{\mathbf{u}}^{oT}, \|\dot{\mathbf{u}}^o\|^2\dot{\mathbf{u}}^{oT}]^T, \\ \boldsymbol{\varphi}^o(14 : 16) &= [(\mathbf{u}^{oT}\dot{\mathbf{u}}^o)^2, \|\dot{\mathbf{u}}^o\|^4, \mathbf{u}^{oT}\dot{\mathbf{u}}^o\|\dot{\mathbf{u}}^o\|^2]^T.\end{aligned}\tag{3.21}$$

$\boldsymbol{\varphi}^o$ in this case has 16 elements and the individual variables are shown in Table 3.2. The number of actual unknowns is only 4 and the elements are related to each other. Table 3.2 tabulates all possible first and second order relations for the elements of the estimation variable $\boldsymbol{\varphi}$.

Let \mathbf{A}_k be the $M \times 16$ matrix having the i -th row

$$\begin{aligned}\mathbf{A}_k(i, 1 : 9) &= [2d_{k,i}^2\mathbf{s}_i^T, 2kd_{k,i}^2\mathbf{s}_i^T, -d_{k,i}^2, -2kd_{k,i}^2, x_i^2 - k^2d_{k,i}^2, y_i^2 - k^2d_{k,i}^2, 2x_iy_i], \\ \mathbf{A}_k(i, 10 : 16) &= [-2\mathbf{s}_i^T, -2k\mathbf{s}_i^T, 1, k^2, 2k],\end{aligned}\tag{3.22}$$

Table 3.2: Relations Among the Elements of φ for the Multiple-Time Measurements Case

Elements	Relations	
$\varphi(1) = x$	$\varphi(7) = \varphi(3)^2$	$\varphi(7)\varphi(15) = \varphi(12)\varphi(12)$
$\varphi(2) = y$	$\varphi(8) = \varphi(4)^2$	$\varphi(7)\varphi(16) = \varphi(10)\varphi(12)$
$\varphi(3) = \dot{x}$	$\varphi(9) = \varphi(3)\varphi(4)$	$\varphi(8)\varphi(14) = \varphi(11)\varphi(11)$
$\varphi(4) = \dot{y}$	$\varphi(10) = \varphi(3)\varphi(6)$	$\varphi(8)\varphi(15) = \varphi(13)\varphi(13)$
$\varphi(5) = x^2 + y^2$	$\varphi(11) = \varphi(4)\varphi(6)$	$\varphi(8)\varphi(16) = \varphi(11)\varphi(13)$
$\varphi(6) = x\dot{x} + y\dot{y}$	$\varphi(14) = \varphi(6)\varphi(6)$	$\varphi(9)\varphi(14) = \varphi(10)\varphi(11)$
$\varphi(7) = \dot{x}^2$		$\varphi(9)\varphi(15) = \varphi(12)\varphi(13)$
$\varphi(8) = \dot{y}^2$	$\varphi(5) = \varphi(1)^2 + \varphi(2)^2$	$\varphi(9)\varphi(16) = \varphi(10)\varphi(13)$
$\varphi(9) = \dot{x}\dot{y}$	$\varphi(6) = \varphi(1)\varphi(3) + \varphi(2)\varphi(4)$	$\varphi(9)\varphi(16) = \varphi(11)\varphi(12)$
$\varphi(10) = \dot{x}(x\dot{x} + y\dot{y})$	$\varphi(10) = \varphi(1)\varphi(7) + \varphi(2)\varphi(9)$	$\varphi(10)\varphi(15) = \varphi(12)\varphi(16)$
$\varphi(11) = \dot{y}(x\dot{x} + y\dot{y})$	$\varphi(11) = \varphi(1)\varphi(9) + \varphi(2)\varphi(8)$	$\varphi(10)\varphi(16) = \varphi(12)\varphi(14)$
$\varphi(12) = \dot{x}(\dot{x}^2 + \dot{y}^2)$	$\varphi(12) = \varphi(3)\varphi(7) + \varphi(4)\varphi(9)$	$\varphi(11)\varphi(15) = \varphi(13)\varphi(16)$
$\varphi(13) = \dot{y}(\dot{x}^2 + \dot{y}^2)$	$\varphi(13) = \varphi(3)\varphi(9) + \varphi(4)\varphi(8)$	$\varphi(11)\varphi(16) = \varphi(13)\varphi(14)$
$\varphi(14) = (x\dot{x} + y\dot{y})^2$	$\varphi(14) = \varphi(1)\varphi(10) + \varphi(2)\varphi(11)$	$\varphi(14)\varphi(15) = \varphi(16)\varphi(16)$
$\varphi(15) = (\dot{x}^2 + \dot{y}^2)^2$	$\varphi(15) = \varphi(3)\varphi(12) + \varphi(4)\varphi(13)$	
$\varphi(16) = (x\dot{x} + y\dot{y})(\dot{x}^2 + \dot{y}^2)$	$\varphi(16) = \varphi(1)\varphi(12) + \varphi(2)\varphi(13)$	$\varphi(3)\varphi(12) = \varphi(7)(\varphi(7) + \varphi(8))$
	$\varphi(16) = \varphi(3)\varphi(10) + \varphi(4)\varphi(11)$	$\varphi(3)\varphi(15) = \varphi(12)(\varphi(7) + \varphi(8))$
		$\varphi(3)\varphi(16) = \varphi(10)(\varphi(7) + \varphi(8))$
	$\varphi(3)\varphi(8) = \varphi(4)\varphi(9)$	$\varphi(4)\varphi(13) = \varphi(8)(\varphi(7) + \varphi(8))$
	$\varphi(3)\varphi(9) = \varphi(4)\varphi(7)$	$\varphi(4)\varphi(15) = \varphi(13)(\varphi(7) + \varphi(8))$
	$\varphi(3)\varphi(10) = \varphi(6)\varphi(7)$	$\varphi(4)\varphi(16) = \varphi(11)(\varphi(7) + \varphi(8))$
	$\varphi(4)\varphi(11) = \varphi(6)\varphi(8)$	$\varphi(6)\varphi(15) = \varphi(10)\varphi(12) + \varphi(11)\varphi(13)$
	$\varphi(7)\varphi(8) = \varphi(9)\varphi(9)$	$\varphi(6)\varphi(16) = \varphi(14)(\varphi(7) + \varphi(8))$
	$\varphi(7)\varphi(14) = \varphi(10)\varphi(10)$	

the length M vector \mathbf{h}_k be

$$\mathbf{h}_k = [d_{k,1}^2 \|\mathbf{s}_1\|^2, d_{k,2}^2 \|\mathbf{s}_2\|^2, \dots, d_{k,M}^2 \|\mathbf{s}_M\|^2]^T, \quad (3.23)$$

and the size M matrix \mathbf{B}_k be

$$\mathbf{B}_k = 2 \frac{c}{f_o} \text{diag} \{ [d_{k,1} r_{k,1}^2, d_{k,2} r_{k,2}^2, \dots, d_{k,M} r_{k,M}^2] \}. \quad (3.24)$$

Putting together \mathbf{A}_k , \mathbf{h}_k and \mathbf{B}_k for $k = 0, 1, \dots, N-1$ separately such that

$$\mathbf{A} = [\mathbf{A}_0^T, \mathbf{A}_1^T, \dots, \mathbf{A}_{N-1}^T]^T, \quad (3.25)$$

$$\mathbf{h} = [\mathbf{h}_0^T, \mathbf{h}_1^T, \dots, \mathbf{h}_{N-1}^T]^T, \quad (3.26)$$

$$\mathbf{B} = \text{diag} \{ \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{N-1} \}, \quad (3.27)$$

we can represent all MN equations over $i = 1, \dots, M$ and $k = 0, \dots, N-1$ of (3.12)

in a matrix form as

$$\mathbf{B}\boldsymbol{\varepsilon} \simeq \mathbf{h} - \mathbf{A}\boldsymbol{\varphi}^o, \quad (3.28)$$

where the second and higher order errors terms have been ignored and the approximation is reasonable when the errors are small. The optimization for the multiple-time measurements case, using the weighting matrix in (3.19) with N larger than one, is

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (3.29a)$$

$$\text{s.t. } \varphi(5) = \varphi(1)^2 + \varphi(2)^2, \quad (3.29b)$$

$$\varphi(6) = \varphi(1)\varphi(3) + \varphi(2)\varphi(4), \quad (3.29c)$$

$$\varphi(7) = \varphi(3)^2, \quad (3.29d)$$

$$\varphi(8) = \varphi(4)^2, \quad (3.29e)$$

$$\varphi(9) = \varphi(3)\varphi(4), \quad (3.29f)$$

$$\varphi(10) = \varphi(1)\varphi(7) + \varphi(2)\varphi(9), \quad (3.29g)$$

$$\varphi(11) = \varphi(1)\varphi(9) + \varphi(2)\varphi(8), \quad (3.29h)$$

$$\varphi(12) = \varphi(3)\varphi(7) + \varphi(4)\varphi(9), \quad (3.29i)$$

$$\varphi(13) = \varphi(3)\varphi(9) + \varphi(4)\varphi(8), \quad (3.29j)$$

$$\varphi(14) = \varphi(1)\varphi(10) + \varphi(2)\varphi(11), \quad (3.29k)$$

$$\varphi(15) = \varphi(3)\varphi(12) + \varphi(4)\varphi(13), \quad (3.29l)$$

$$\varphi(16) = \varphi(1)\varphi(12) + \varphi(2)\varphi(13). \quad (3.29m)$$

We have taken the first 12 relations in Table 3.2 to impose constraints among the 16 elements of $\boldsymbol{\varphi}$, for the purpose to fix the number of independent variables to 4. The remaining relations in Table 3.2 are not needed in the formulation. They will be used to improve the optimization when (3.29) is approximated with SDR.

3.1.2 Carrier Frequency Unavailable

When the carrier frequency is not known, f_o^o is an additional unknown. In terms of the available noisy sensor positions, (2.2) becomes after using (2.6) and (3.1),

$$f_{k,i} = f_o^o - \frac{f_o^o(\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o}{c r_{k,i}} + \varepsilon_{k,i} + o(\|\Delta \mathbf{s}_i\|). \quad (3.30)$$

The composite noise term $\varepsilon_{k,i}$ is

$$\varepsilon_{k,i} = \frac{-f_o^o}{c r_{k,i}^o} \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o} \Delta \mathbf{s}_i + n_{k,i}, \quad (3.31)$$

where $\mathbf{P}_{k,i}^{\perp o}$ is given below (3.3). The covariance matrix of the composite noise vector from all measurements is

$$\mathbf{Q}_\varepsilon = \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{Q}_n, \quad (3.32)$$

and \mathbf{D}_s is defined below (3.3) and (3.4).

Rather than constructing a pseudo-linear formulation that would result in a large number of auxiliary variables and constraints, we shall utilize the formulation for the known but erroneous carrier frequency case for joint estimation of the object location and emitted frequency. The methodology will become clear in Section 3.3 for finding the solution.

3.2 Solution: Carrier Frequency Available

We shall present two methods to solve the constrained WLS problems for localization. One is based on the unconstrained successive minimization that results in an algebraic closed-form solution. The other is the convex optimization method using SDR. The former is computationally attractive and is suitable in the small noise environment. The other is more computationally demanding but can yield better results when the noise level is high. Both methods will be able to achieve the CRLB performance in their intended operating environment. This section presents the solutions for the single-time and multiple-time observations when the carrier frequency is available.

The solutions when the carrier frequency is unavailable will be described in the next section.

3.2.1 Algebraic Solution

This solution assumes the elements of $\boldsymbol{\varphi}$ are independent variables to obtain the WLS solution, and the constraints are exploited next through nonlinear transformation to refine the estimate [12]. Albeit the matrix and vector variables \mathbf{A} , \mathbf{B} , \mathbf{h} and \mathbf{W} are defined differently, both the single-time and multiple-time observation cases share the common form that the solution to (3.20a) or (3.29a) when ignoring the constraints is

$$\boldsymbol{\varphi} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{h}. \quad (3.33)$$

The covariance matrix of the WLS solution $\boldsymbol{\varphi}$ can reasonably be approximated by [12],

$$\text{cov}(\boldsymbol{\varphi}) \simeq (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1}, \quad (3.34)$$

when the noise in \mathbf{A} is not significant to be neglected.

The weighting matrix \mathbf{W} is given by (3.19) and it is unavailable since \mathbf{B} and \mathbf{Q}_ε involve the true object location. Nevertheless, we can construct \mathbf{W} through approximating the true values needed with the least-squares solution of $\boldsymbol{\varphi}$ by using $\mathbf{W} = \mathbf{I}$ in (3.33). The resulting error for the solution of $\boldsymbol{\varphi}$ is often negligible as the WLS optimization is not sensitive to the error in the weighting matrix [9, 10].

The refinement steps for the single-time and multiple-time observation scenarios are not the same, due to the differences in the definitions of $\boldsymbol{\varphi}$ and the constraints. They are elaborated separately below.

Single-Time Observation

We shall utilize the relations (3.20b)-(3.20f) among the elements of $\boldsymbol{\varphi}$ to improve the estimation accuracy. Let us introduce the separate unknown vector

$$\tilde{\boldsymbol{\varphi}}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT} \otimes \dot{\mathbf{u}}^{oT}]^T, \quad (3.35)$$

that has independent variables. It consists of the first, second, fourth and fifth elements of $\boldsymbol{\varphi}^o$ in (3.14). Also, let the pseudo data vector constructed from the WLS solution (3.33) be

$$\tilde{\mathbf{h}} = [\boldsymbol{\varphi}^T(1:5), \varphi^2(6), \boldsymbol{\varphi}^T(7:9)]^T. \quad (3.36)$$

Expressing $\boldsymbol{\varphi} = \boldsymbol{\varphi}^o + \Delta\boldsymbol{\varphi}$ where $\Delta\boldsymbol{\varphi}$ is the estimation error of the WLS solution, every element in $\tilde{\mathbf{h}}$ can be expressed in terms of the elements of $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}^o$. In particular,

$$\tilde{\mathbf{h}}(1:2) = \tilde{\boldsymbol{\varphi}}^o(1:2) + \Delta\boldsymbol{\varphi}(1:2), \quad (3.37a)$$

$$\tilde{\mathbf{h}}(4:5) = \tilde{\boldsymbol{\varphi}}^o(3:4) + \Delta\boldsymbol{\varphi}(4:5). \quad (3.37b)$$

Moreover, from the five constraints in (3.20),

$$\tilde{h}(3) = \varphi(1)\tilde{\varphi}^o(1) + \varphi(2)\tilde{\varphi}^o(2) - \varphi(1)\Delta\varphi(1) - \varphi(2)\Delta\varphi(2) + \Delta\varphi(3) + \Delta\varphi(1)^2 + \Delta\varphi(2)^2, \quad (3.38a)$$

$$\begin{aligned} \tilde{h}(6) &= \frac{1}{2}\varphi(5)\tilde{\varphi}^o(3) + \frac{1}{2}\varphi(4)\tilde{\varphi}^o(4) - \frac{1}{2}\varphi(5)\Delta\varphi(4) - \frac{1}{2}\varphi(4)\Delta\varphi(5) + 2\varphi(6)\Delta\varphi(6) \\ &\quad + \Delta\varphi(4)\Delta\varphi(5) - \Delta\varphi(6)^2, \end{aligned} \quad (3.38b)$$

$$\begin{aligned} \tilde{h}(7) &= \varphi(4)\tilde{\varphi}^o(1) + \varphi(6)\tilde{\varphi}^o(2) - \varphi(1)\Delta\varphi(4) - \varphi(2)\Delta\varphi(6) + \Delta\varphi(7) + \Delta\varphi(1)\Delta\varphi(4) \\ &\quad + \Delta\varphi(2)\Delta\varphi(6), \end{aligned} \quad (3.38c)$$

$$\begin{aligned}\tilde{h}(8) &= \varphi(6)\tilde{\varphi}^o(1) + \varphi(5)\tilde{\varphi}^o(2) - \varphi(2)\Delta\varphi(5) - \varphi(1)\Delta\varphi(6) + \Delta\varphi(8) + \Delta\varphi(1)\Delta\varphi(6) \\ &\quad + \Delta\varphi(2)\Delta\varphi(5),\end{aligned}\tag{3.38d}$$

$$\begin{aligned}\tilde{h}(9) &= \varphi(7)\tilde{\varphi}^o(1) + \varphi(8)\tilde{\varphi}^o(2) - \varphi(1)\Delta\varphi(7) - \varphi(2)\Delta\varphi(8) + \Delta\varphi(9) + \Delta\varphi(1)\Delta\varphi(7) \\ &\quad + \Delta\varphi(2)\Delta\varphi(8).\end{aligned}\tag{3.38e}$$

In (3.38), we have avoided the true values appearing in the error terms by associating the elements of $\tilde{\varphi}^o$ with those of φ^o . For instance, (3.38a) has applied $\tilde{\varphi}^o(1)\Delta\varphi(1) = \varphi^o(1)\Delta\varphi(1) = \varphi(1)\Delta\varphi(1) - \Delta\varphi(1)^2$ and $\tilde{\varphi}^o(2)\Delta\varphi(2) = \varphi^o(2)\Delta\varphi(2) = \varphi(2)\Delta\varphi(2) - \Delta\varphi(2)^2$; (3.38b) has used $\tilde{\varphi}^o(4)\Delta\varphi(4) = \varphi^o(5)\Delta\varphi(4) = \varphi(5)\Delta\varphi(4) - \Delta\varphi(4)\Delta\varphi(5)$, $\tilde{\varphi}^o(3)\Delta\varphi(5) = \varphi^o(4)\Delta\varphi(5) = \varphi(4)\Delta\varphi(5) - \Delta\varphi(4)\Delta\varphi(5)$ and $\varphi^o(6)\Delta\varphi(6) = \varphi(6)\Delta\varphi(6) - \Delta\varphi(6)^2$, etc. Accordingly, we construct the 9×4 matrix

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \varphi(1) & \varphi(2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \varphi(5)/2 & \varphi(4)/2 \\ \varphi(4) & \varphi(6) & 0 & 0 \\ \varphi(6) & \varphi(5) & 0 & 0 \\ \varphi(7) & \varphi(8) & 0 & 0 \end{bmatrix},\tag{3.39}$$

and the 9×9 matrix

$$\tilde{\mathbf{B}} = \text{diag}\{[\mathbf{1}_5^T, 2\varphi(6), \mathbf{1}_3^T]^T\} - \mathbf{C}. \quad (3.40)$$

\mathbf{C} is a sparse matrix with the non-zero elements given by

$$\begin{aligned} \mathbf{C}(3, 1:2) &= \boldsymbol{\varphi}^T(1:2), \quad \mathbf{C}(9, 7:8) = \boldsymbol{\varphi}^T(1:2), \\ \mathbf{C}(6:8, 4:6) &= \begin{bmatrix} \varphi(5)/2 & \varphi(4)/2 & 0 \\ \varphi(1) & 0 & \varphi(2) \\ 0 & \varphi(2) & \varphi(1) \end{bmatrix}. \end{aligned} \quad (3.41)$$

(3.38) in matrix form, after dropping the second and higher order error terms, is

$$\tilde{\mathbf{B}}\Delta\boldsymbol{\varphi} \simeq \tilde{\mathbf{h}} - \tilde{\mathbf{A}}\tilde{\boldsymbol{\varphi}}^o. \quad (3.42)$$

The WLS solution for $\tilde{\boldsymbol{\varphi}}^o$ is

$$\tilde{\boldsymbol{\varphi}} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{h}}. \quad (3.43)$$

$\tilde{\mathbf{W}}$ is set as

$$\tilde{\mathbf{W}} = \tilde{\mathbf{B}}^{-T} (\mathbf{A}^T \mathbf{W} \mathbf{A}) \tilde{\mathbf{B}}^{-1}, \quad (3.44)$$

which is an approximation of $E[\tilde{\mathbf{B}}\Delta\boldsymbol{\varphi}\Delta\boldsymbol{\varphi}^T\tilde{\mathbf{B}}^T]^{-1}$ where (3.34) has been used.

At last, we obtain the object position and velocity estimates from $\tilde{\boldsymbol{\varphi}}$ using (3.35) by

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\varphi}}(1:2) \\ \mathbf{P} \sqrt{\tilde{\boldsymbol{\varphi}}(3:4)} \end{bmatrix}. \quad (3.45)$$

The matrix \mathbf{P} has four possible choices:

$$\mathbf{P} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}, \quad (3.46)$$

whose purpose is to resolve the sign ambiguity after the square-root operation. The correct choice of \mathbf{P} is determined by trying all four possibilities of $\boldsymbol{\theta}$ and selecting the one that gives the smallest approximate Maximum Likelihood (ML) cost function deduced from (3.2), which is defined by (3.58)-(3.59) with \mathbf{Q}_ε equal to (3.6).

Multiple-Time Observations

The refinement step in this case uses the same unknown vector in (2.38). It is the first four elements of $\boldsymbol{\varphi}^o$ defined in (3.21). From the constraints (3.29b)-(3.29m), following similar procedure as in the single-time observation case by expressing $\boldsymbol{\varphi}$ as $\boldsymbol{\varphi}^o + \Delta\boldsymbol{\varphi}$ and dropping the second and higher order error terms, the associated set of linear equations is

$$\tilde{\mathbf{B}}\Delta\boldsymbol{\varphi} \simeq \boldsymbol{\varphi} - \tilde{\mathbf{A}}\boldsymbol{\theta}^o. \quad (3.47)$$

The matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \varphi(1) & \varphi(2) & 0 & 0 \\ \varphi(3) & \varphi(4) & 0 & 0 \\ 0 & 0 & \varphi(3) & 0 \\ 0 & 0 & 0 & \varphi(4) \\ 0 & 0 & \varphi(4)/2 & \varphi(3)/2 \\ \varphi(7) & \varphi(9) & 0 & 0 \\ \varphi(9) & \varphi(8) & 0 & 0 \\ 0 & 0 & \varphi(7) & \varphi(9) \\ 0 & 0 & \varphi(9) & \varphi(8) \\ \varphi(10) & \varphi(11) & 0 & 0 \\ 0 & 0 & \varphi(12) & \varphi(13) \\ \varphi(12) & \varphi(13) & 0 & 0 \end{bmatrix}, \quad (3.48)$$

and

$$\tilde{\mathbf{B}} = \mathbf{I}_{16 \times 16} - \mathbf{C}. \quad (3.49)$$

The elements of \mathbf{C} are zeros except for those in (3.50),

$$\begin{aligned}
\mathbf{C}(5, 1 : 2) &= \boldsymbol{\varphi}^T(1 : 2) & , & \quad \mathbf{C}(14, 10 : 11) = \boldsymbol{\varphi}^T(1 : 2), \\
\mathbf{C}(6 : 9, 3 : 4) &= \begin{bmatrix} \varphi(1) & \varphi(2) \\ \varphi(3) & 0 \\ 0 & \varphi(4) \\ \varphi(4)/2 & \varphi(3)/2 \end{bmatrix} & , & \quad \mathbf{C}(10 : 13, 7 : 9) = \begin{bmatrix} \varphi(1) & 0 & \varphi(2) \\ 0 & \varphi(2) & \varphi(1) \\ \varphi(3) & 0 & \varphi(4) \\ 0 & \varphi(4) & \varphi(3) \end{bmatrix}, \\
\mathbf{C}(15 : 16, 12 : 13) &= \begin{bmatrix} \varphi(3) & \varphi(4) \\ \varphi(1) & \varphi(2) \end{bmatrix}.
\end{aligned} \tag{3.50}$$

The WLS solution for $\boldsymbol{\theta}^o$ is

$$\boldsymbol{\theta} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \boldsymbol{\varphi}, \tag{3.51}$$

and the weighting matrix is set to (3.44) where $\tilde{\mathbf{B}}$ is now given by (3.49)-(3.50). The covariance matrix of the estimate over the small error region can be approximated by [12]

$$\text{cov}(\boldsymbol{\theta}) \simeq (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1}, \tag{3.52}$$

where the noise in $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$ are small enough to be neglected.

The Algorithm 1 Table summarizes the proposed algebraic solution for the single-time observation case and Algorithm 2 Table for the multiple-time observation scenario. In Algorithm 2, we have repeated steps 8 and 9 several times to improve the weighting matrix $\tilde{\mathbf{W}}$ as we approximate the true object location it requires by an estimate. Repeating such processing is not needed for Algorithm 1 unless the object

Algorithm 1: Algebraic Solution for Single-Time Observation

Input: \mathbf{f} - frequency measurement vector.

\mathbf{s} - sensor position vector.

$\mathbf{Q}_n, \mathbf{Q}_s, \sigma_{f_o}^2$ - noise covariance matrices.

Output: $\boldsymbol{\theta}$ - object location estimate.

Implementation:

1. Construct \mathbf{A} and \mathbf{h} according to (3.15)-(3.16).
 2. Set $\mathbf{W} = \mathbf{I}$.
 3. Solve for $\boldsymbol{\varphi}$ using (3.33).
 4. Obtain an initial estimate of $\boldsymbol{\theta}$ using (3.55).
 5. Find \mathbf{P} using (3.46) and the procedure below it.
 6. Update \mathbf{W} using (3.6), (3.17) and (3.19) by approximating the true values needed for r_i, \mathbf{d}_f and \mathbf{D}_s from the initial estimate of $\boldsymbol{\theta}$.
 7. Repeat steps 3-6.
 8. Construct $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{h}}$ according to (3.39) and (3.36).
 9. Form $\tilde{\mathbf{W}}$ using (3.40)-(3.41) and (3.44).
 10. Solve for $\tilde{\boldsymbol{\varphi}}$ using (3.43).
 11. Obtain the final estimate $\boldsymbol{\theta}$ using (3.45)-(3.46) and the procedure below them.
-

is found located near the sensors.

3.2.2 SDR Solution

This solution transforms the constrained optimization to a semi-definite programming (SDP) problem [102, 103] through SDR, so that it can be solved efficiently by some convex optimization package such as CVX [94].

Algorithm 2: Algebraic Solution for Multiple-Time Observations

Input: \mathbf{f} - frequency measurement vector.

\mathbf{s} - sensor position vector.

$\mathbf{Q}_n, \mathbf{Q}_s, \sigma_{f_o}^2$ - noise covariance matrices.

Output: $\boldsymbol{\theta}$ - object location estimate.

Implementation:

1. Construct \mathbf{A} and \mathbf{h} according to (3.22)-(3.23) and (3.25)-(3.26).
 2. Set $\mathbf{W} = \mathbf{I}$.
 3. Solve for $\boldsymbol{\varphi}$ using (3.33).
 4. Set an initial estimate of $\boldsymbol{\theta}$ as $\boldsymbol{\varphi}(1 : 4)$.
 5. Update \mathbf{W} using (3.6), (3.19), (3.24) and (3.27) by approximating the true values with those computed from the initial estimate of $\boldsymbol{\theta}$.
 6. Repeat steps 3-5.
 7. Construct $\widetilde{\mathbf{A}}$ according to (3.48).
 8. Form $\widetilde{\mathbf{W}}$ by (3.44), (3.49)-(3.50) using $\boldsymbol{\theta}$ to obtain $\boldsymbol{\varphi}$ from (3.21).
 9. Generate the final estimate $\boldsymbol{\theta}$ using (3.51).
 10. Repeat steps 8-9 a few times.
-

Single-Time Observation

The optimization problem is (3.20). Expanding the cost function J and dropping the constant term $\mathbf{h}^T \mathbf{W} \mathbf{h}$ that does not depend on the unknown $\boldsymbol{\varphi}$, it becomes

$$\bar{J}(\boldsymbol{\varphi}, \boldsymbol{\Phi}) = \text{tr}(\mathbf{A}^T \mathbf{W} \mathbf{A} \boldsymbol{\Phi}) - 2\mathbf{h}^T \mathbf{W} \mathbf{A} \boldsymbol{\varphi}, \quad (3.53)$$

where $\boldsymbol{\Phi} = \boldsymbol{\varphi} \boldsymbol{\varphi}^T$ and has rank one.

We form the SDP by considering both $\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$ are variables and relaxing the rank of $\boldsymbol{\Phi}$ to be larger than one, resulting in

$$\min_{\boldsymbol{\varphi}, \boldsymbol{\Phi}} \bar{J}(\boldsymbol{\varphi}, \boldsymbol{\Phi}), \quad (3.54a)$$

Table 3.3: Constraints Among the Elements of $\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$ for the SDR Solution of the Single-Time Measurement Case

$\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$	$\boldsymbol{\Phi}$
$\varphi(3) = \Phi(1, 1) + \Phi(2, 2)$	$\Phi(4, 9) = \Phi(7, 7)$
$\varphi(7) = \Phi(1, 4) + \Phi(2, 6)$	$\Phi(5, 9) = \Phi(8, 8)$
$\varphi(8) = \Phi(1, 6) + \Phi(2, 5)$	$\Phi(6, 6) = \Phi(4, 5)$
$\varphi(9) = \Phi(1, 7) + \Phi(2, 8)$	$\Phi(6, 9) = \Phi(7, 8)$
$\varphi(3) \geq 0$	
$\varphi(4) \geq 0$	
$\varphi(5) \geq 0$	
$\varphi(9) \geq 0$	

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\varphi} \\ \boldsymbol{\varphi}^T & 1 \end{bmatrix} \succeq 0, \quad (3.54b)$$

$$\text{All constraints specified in Table 3.3.} \quad (3.54c)$$

In the SDP, (3.54b) comes from relaxing the rank of $\boldsymbol{\Phi}$. The constraints in Table 3.3, which includes those of (3.20b)-(3.20f), are directly deduced based on Table 3.1, by realizing from the definition of $\boldsymbol{\Phi}$ below (3.53) that its (i, j) -th element is simply related to those of $\boldsymbol{\varphi}$ by $\Phi(i, j) = \varphi(i)\varphi(j)$. All constraints are linear, except the four non-negative constraints from the definition of $\boldsymbol{\varphi}^o$. In addition to the five constraints (3.20b)-(3.20f), including the remaining constraints from Table 3.1 improves the solution, due to the rank-1 relaxation of $\boldsymbol{\Phi}$ in the SDP.

The relaxed optimization problem can be solved by a convex optimization package. The object localization estimate is

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}(1:2) \\ \mathbf{P}\sqrt{\boldsymbol{\varphi}(4:5)} \end{bmatrix}, \quad (3.55)$$

where \mathbf{P} is given by (3.46) whose correct choice is elaborated below (3.46).

Multiple-Time Observations

The equivalent objective function of (3.29) has the same form as (3.53). After SDR, the relaxed SDP is

$$\min_{\varphi, \Phi} \bar{J}(\varphi, \Phi), \quad (3.56a)$$

$$\text{s.t.} \quad \begin{bmatrix} \Phi & \varphi \\ \varphi^T & 1 \end{bmatrix} \succeq 0, \quad (3.56b)$$

$$\text{All constraints specified in Table 3.4.} \quad (3.56c)$$

The constraints listed in Table 3.4 is deduced directly from Table 3.2. Due to the expansion of the number of variables from the original 4 unknowns to the 16 in φ , using only the 12 constraints (3.29b)-(3.29m) is found not sufficient after relaxation. We include all constraints from the first and second order relations among the elements of φ . Exploiting all of the constraints not only improves the tightness of the SDP optimization to the original, but also increases the noise tolerance.

After solving (3.56), the object location estimate is the first four elements of the φ solution from the relaxed SDP,

$$\boldsymbol{\theta} = \varphi(1 : 4). \quad (3.57)$$

The Algorithm 3 Table summarizes the proposed SDR solution for the single-time observation case, and the Algorithm 4 Table for the multiple-time observation

Table 3.4: Constraints on φ and Φ for the SDR Solution of the Multiple-Time Measurements Case

φ and Φ	Φ
$\varphi(5) = \Phi(1, 1) + \Phi(2, 2)$	$\Phi(3, 8) = \Phi(4, 9)$
$\varphi(6) = \Phi(1, 3) + \Phi(2, 4)$	$\Phi(3, 9) = \Phi(4, 7)$
$\varphi(7) = \Phi(3, 3)$	$\Phi(3, 10) = \Phi(6, 7)$
$\varphi(8) = \Phi(4, 4)$	$\Phi(4, 11) = \Phi(6, 8)$
$\varphi(9) = \Phi(3, 4)$	$\Phi(7, 8) = \Phi(9, 9)$
$\varphi(10) = \Phi(3, 6)$	$\Phi(7, 14) = \Phi(10, 10)$
$\varphi(10) = \Phi(1, 7) + \Phi(2, 9)$	$\Phi(7, 15) = \Phi(12, 12)$
$\varphi(11) = \Phi(4, 6)$	$\Phi(7, 16) = \Phi(10, 12)$
$\varphi(11) = \Phi(1, 9) + \Phi(2, 8)$	$\Phi(8, 14) = \Phi(11, 11)$
$\varphi(12) = \Phi(3, 7) + \Phi(4, 9)$	$\Phi(8, 15) = \Phi(13, 13)$
$\varphi(13) = \Phi(3, 9) + \Phi(4, 8)$	$\Phi(8, 16) = \Phi(11, 13)$
$\varphi(14) = \Phi(6, 6)$	$\Phi(9, 14) = \Phi(10, 11)$
$\varphi(14) = \Phi(1, 10) + \Phi(2, 11)$	$\Phi(9, 15) = \Phi(12, 13)$
$\varphi(15) = \Phi(3, 12) + \Phi(4, 13)$	$\Phi(9, 16) = \Phi(10, 13)$
$\varphi(16) = \Phi(1, 12) + \Phi(2, 13)$	$\Phi(9, 16) = \Phi(11, 12)$
$\varphi(16) = \Phi(3, 10) + \Phi(4, 11)$	$\Phi(10, 15) = \Phi(12, 16)$
	$\Phi(10, 16) = \Phi(12, 14)$
$\varphi(5) \geq 0$	$\Phi(11, 15) = \Phi(13, 16)$
$\varphi(7) \geq 0$	$\Phi(11, 16) = \Phi(13, 14)$
$\varphi(8) \geq 0$	$\Phi(14, 15) = \Phi(16, 16)$
$\varphi(14) \geq 0$	$\Phi(3, 12) = \Phi(7, 7) + \Phi(7, 8)$
$\varphi(15) \geq 0$	$\Phi(3, 15) = \Phi(7, 12) + \Phi(8, 12)$
	$\Phi(3, 16) = \Phi(7, 10) + \Phi(8, 10)$
	$\Phi(4, 13) = \Phi(7, 8) + \Phi(8, 8)$
	$\Phi(4, 15) = \Phi(7, 13) + \Phi(8, 13)$
	$\Phi(4, 16) = \Phi(7, 11) + \Phi(8, 11)$
	$\Phi(6, 15) = \Phi(7, 16) + \Phi(8, 16)$
	$\Phi(6, 16) = \Phi(7, 14) + \Phi(8, 14)$

Algorithm 3: SDR Solution for Single-Time Observation

Input: \mathbf{f} - frequency measurement vector.

\mathbf{s} - sensor position vector.

$\mathbf{Q}_n, \mathbf{Q}_s, \sigma_{f_o}^2$ - noise covariance matrices.

Output: $\boldsymbol{\theta}$ - object location estimate.

Implementation:

1. Create \mathbf{A} and \mathbf{h} according to (3.15)-(3.16).
 2. Set $\mathbf{W} = \mathbf{I}$.
 3. Construct the SDP problem as follows:
 - a. Define $\boldsymbol{\varphi}$ as a 9×1 vector and $\boldsymbol{\Phi}$ as a 9×9 symmetric matrix.
 - b. Set the objective function as (3.53).
 - c. Set the constraints according to (3.54b) and Table 3.3.
 4. Use an SDP solver to find $\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$.
 5. Form an estimate of $\boldsymbol{\theta}$ using (3.55).
 6. Determine \mathbf{P} using (3.46) and the procedure below it.
 7. Update \mathbf{W} using (3.6), (3.17) and (3.19) by approximating the true values with those computed from the estimate of $\boldsymbol{\theta}$.
 8. Repeat Steps 3-6.
-

scenario.

3.3 Solution: Carrier Frequency Unavailable

When the carrier frequency is not available or the error Δf_o is not zero mean, f_o becomes an unknown in addition to $\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T$. While it is possible to create a formulation for constrained optimization in this case as what we did when it is available, the number of elements in $\boldsymbol{\varphi}^o$ and the number of constraints would be excessive. We shall propose an approach for obtaining a location estimate in this case, by making use of the solution when the carrier frequency is available.

When accounting for sensor position errors, the measurement model for the unknown carrier frequency case is (3.30), where the covariance matrix of the composite

Algorithm 4: SDR Solution for Multiple-Time Observations

Input: \mathbf{f} - frequency measurement vector.

\mathbf{s} - sensor position vector.

$\mathbf{Q}_n, \mathbf{Q}_s, \sigma_{f_o}^2$ - noise covariance matrices.

Output: $\boldsymbol{\theta}$ - object location estimate.

Implementation:

1. Create \mathbf{A} and \mathbf{h} according to (3.22)-(3.23) and (3.25)-(3.26).
 2. Set $\mathbf{W} = \mathbf{I}$.
 3. Construct the SDP problem as follows:
 - a. Define $\boldsymbol{\varphi}$ as a 16×1 vector and $\boldsymbol{\Phi}$ as a 16×16 symmetric matrix.
 - b. Set the objective function as (3.53).
 - c. Set the constraints according to (3.56b) and Table 3.4.
 4. Use an SDP solver to find $\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$.
 5. Form an estimate of $\boldsymbol{\theta}$ using (3.57).
 6. Update \mathbf{W} using (3.6), (3.19), (3.24) and (3.27) by approximating the true values with those computed from the estimate of $\boldsymbol{\theta}$.
 7. Repeat steps 3-5.
-

error resulting from the frequency measurement noise and sensor position errors is (3.32). The composite error is Gaussian distributed since the two components are and the ML cost function of (3.30) can be approximated by the following after dropping $o(\|\Delta \mathbf{s}\|)$,

$$J(f_o, \boldsymbol{\theta}) = \left(\mathbf{f} - \hat{\mathbf{f}}(f_o, \boldsymbol{\theta}) \right)^T \mathbf{Q}_\varepsilon^{-1} \left(\mathbf{f} - \hat{\mathbf{f}}(f_o, \boldsymbol{\theta}) \right). \quad (3.58)$$

$\hat{\mathbf{f}}(f_o, \boldsymbol{\theta})$ is the reconstructed frequency vector using the model (2.2) with f_o° replaced by f_o and \mathbf{s}_i° by \mathbf{s}_i , i.e.,

$$\hat{f}_{k,i}(f_o, \boldsymbol{\theta}) = f_o - \frac{f_o(\boldsymbol{\theta}(1:2) + k\boldsymbol{\theta}(3:4) - \mathbf{s}_i)^T \boldsymbol{\theta}(3:4)}{c \|\boldsymbol{\theta}(1:2) + k\boldsymbol{\theta}(3:4) - \mathbf{s}_i\|}. \quad (3.59)$$

$J(f_o, \boldsymbol{\theta})$ is a highly nonlinear complicated function of the unknowns. For a given f_o , a solution of $\boldsymbol{\theta}$ has been derived in Section 3.2, where $\sigma_{f_o}^2$ is set to zero in (3.6) when forming the weighting matrix \mathbf{W} . Hence we can interpret (3.58) as a function of

single unknown f_o ,

$$J(f_o) = \left(\mathbf{f} - \hat{\mathbf{f}}(f_o) \right)^T \mathbf{Q}_\varepsilon^{-1} \left(\mathbf{f} - \hat{\mathbf{f}}(f_o) \right). \quad (3.60)$$

The f_o that minimizes (3.60) can be found by a simple sequential search within an interval. Essentially, with a certain f_o , we (i) find the corresponding $\boldsymbol{\theta}$ using a solution method in Section 3.2, (ii) obtain the elements of $\hat{\mathbf{f}}(f_o)$ by $\hat{f}_{k,i}(f_o) = \hat{f}_{k,i}(f_o, \boldsymbol{\theta})$ from (3.59), and (iii) evaluate $J(f_o)$ by (3.60). Among the trial f_o values, the one that yields the smallest $J(f_o)$ is the solution for f_o^o . The associated $\boldsymbol{\theta}$ is the solution for $\boldsymbol{\theta}^o$.

In practice, the range of possible values of f_o is known, based on some prior knowledge or derived from the observed frequency values [84]. The sequential search can be accomplished through a coarse grid search followed by the Newton-Raphson iteration for better efficiency and higher accuracy. The difference between the case where f_o is completely not known and the case where f_o is available but Δf_o has a non-zero mean will be only the size of the grid search. The number of grid points can be reduced to account only for the small region around the available f_o taking into consideration that σ_{f_o} plays an important role in controlling the size of this region.

The Newton-Raphson iteration is [104]

$$f_o^{(l+1)} = f_o^{(l)} - \left(\nabla^2 J(f_o^{(l)}) \right)^{-1} \nabla J(f_o^{(l)}), \quad (3.61)$$

where $l = 0, 1, \dots, l_{max} - 1$ represents the iteration count and $f_o^{(0)}$ is the carrier frequency value obtained from the coarse grid search process. $\nabla J(\bullet)$ and $\nabla^2 J(\bullet)$ are the first and second derivatives of $J(\bullet)$ with respect to f_o . Obtaining the explicit

expressions of $\nabla J(\bullet)$ and $\nabla^2 J(\bullet)$ is prohibitive. We resort to the numerical approximations

$$\nabla J(f_o^{(l)}) \simeq \frac{J(f_o^{(l)} + \delta) - J(f_o^{(l)})}{\delta}, \quad (3.62)$$

$$\nabla^2 J(f_o^{(l)}) \simeq \frac{\nabla J(f_o^{(l)} + \delta) - \nabla J(f_o^{(l)})}{\delta}, \quad (3.63)$$

and δ is a small value. The maximum number of iterations l_{max} is chosen to satisfy the required accuracy. Algorithm 5 summarizes the processing steps for localizing the object when the carrier frequency is not available.

Before closing this section, we would like to comment on the number of sensors needed for the proposed solutions. The closed-form algebraic solution considers all 9 variables in (3.14) are independent in the first WLS optimization step. As a result it requires at least 9 sensors to operate. The SDR solution, on the other hand, imposes constraints among the elements of $\boldsymbol{\varphi}$ during the optimization process and thereby requiring less sensors. Simulations show that 6 sensors are sufficient for SDR to produce a reasonable estimate and 7 sensors to yield the CRLB performance. The sensor requirements are the same regardless the carrier frequency f_o is available or not. For multiple-time observations, the number of sensors required for both the algebraic and SDR solutions are less. The last subsection in Section 3.5 examines further the number of sensors needed of the proposed algorithms.

3.4 Analysis

Under the first order analysis where the second and higher order noise terms are negligible, we shall derive the theoretical covariance matrix of the proposed algebraic

Algorithm 5: Algorithm for Localization when Carrier Frequency Unavailable

Input: \mathbf{f} - frequency measurement vector.

\mathbf{s} - sensor position vector.

$\mathbf{Q}_n, \mathbf{Q}_s, \sigma_{f_o}^2$ - noise covariance matrices.

$f_{o,min}, f_{o,max}$ - minimum and maximum values of f_o .

μ - step-size.

δ - small value for obtaining gradient.

l_{max} - maximum num. of Newton-Raphson iterations.

Output: $\boldsymbol{\theta}$ - object location estimate.

Implementation:

1. For $f_{o,tst} = f_{o,min}$ to $f_{o,max}$ stepping by μ ,
 - 1.1. Find $\boldsymbol{\theta}$ by Algorithm (*) using $f_{o,tst}$ as the carrier frequency.
 - 1.2. Obtain $\hat{\mathbf{f}}(f_{o,tst})$ from (3.59).
 - 1.3. Determine \mathbf{Q}_ε using (3.32) by approximating the true values with $f_{o,tst}$ and those computed from $\boldsymbol{\theta}$.
 - 1.4. Obtain $J(f_{o,tst})$ using (3.60).
- end For
2. Set $f_o^{(0)}$ to the trial $f_{o,tst}$ value having $J(f_{o,tst})$ the smallest.
3. Set $l = 0$.
4. Find $\nabla J(f_o^{(l)})$ and $\nabla^2 J(f_o^{(l)})$ using (3.60), (3.62)-(3.63).
5. Determine $f_o^{(l+1)}$ using (3.61).
6. Repeat steps 4-5 for $l = 1, 2, \dots, l_{max} - 1$ or reaching required accuracy.
7. Obtain the final solution of $\boldsymbol{\theta}$ using $f_o^{(l_{max})}$ and Algorithm (*).

Algorithm (*) in steps 1.1 and 7 is Algorithm 1, 2, 3 or 4 according to the scenario (single-time or multiple-time measurements) and the solution method.

solution and examine the conditions needed for achieving the CRLB performance. The analysis presented is for the more general case of multiple-time observations. The analytical result and conclusion are valid for the single-time observation scenario as well when the variables are defined accordingly.

3.4.1 Carrier Frequency Available

The covariance matrix of the proposed solution for the multiple-times observation case having f_o available is (3.52). Using (3.19) and (3.44), its inverse over the small error region is

$$\text{cov}(\boldsymbol{\theta})^{-1} \simeq \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \text{cov}(\boldsymbol{\varepsilon})^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \quad (3.64)$$

Let $\mathbf{D} = [\mathbf{d}_f, \mathbf{D}_s]$ for convenience so that (3.6) can be written as $\text{cov}(\boldsymbol{\varepsilon}) = \mathbf{Q}_n + \mathbf{D} \mathbf{Q}_\alpha \mathbf{D}^T$, where \mathbf{Q}_α is defined below (2.41). Applying the matrix inversion lemma [104] gives

$$\text{cov}(\boldsymbol{\varepsilon})^{-1} = \mathbf{Q}_n^{-1} - \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1}. \quad (3.65)$$

Inserting it to (3.64) yields

$$\begin{aligned} \text{cov}(\boldsymbol{\theta})^{-1} \simeq & \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}} - \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} \\ & + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \end{aligned} \quad (3.66)$$

(3.66) has the same structural form as (2.42) with (2.41) inserted. \mathbf{B} in (3.27) is diagonal and $\tilde{\mathbf{B}}$ in (3.49) is sparse, their inverses can be evaluated analytically. Using the true values in \mathbf{A} , \mathbf{B} , $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$, direct algebraic evaluation with the gradients

shown in Appendix A gives

$$\mathbf{B}^{o-1} \mathbf{A}^o \tilde{\mathbf{B}}^{o-1} \tilde{\mathbf{A}}^o = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (3.67a)$$

$$\mathbf{D} = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}}. \quad (3.67b)$$

Let us introduce a few small noise conditions:

$$\frac{n_{k,i}}{f_{k,i}^o - f_o^o} \simeq 0, \quad \frac{\Delta f_o}{f_{k,i}^o - f_o^o} \simeq 0, \quad i = 1, 2, \dots, M, \quad (3.68a)$$

$$\Delta x_i \simeq 0 \quad \text{or} \quad \frac{\Delta x_i}{x_i^o} \simeq 0, \quad i = 1, 2, \dots, M,$$

$$\Delta y_i \simeq 0 \quad \text{or} \quad \frac{\Delta y_i}{y_i^o} \simeq 0, \quad i = 1, 2, \dots, M, \quad (3.68b)$$

$$\Delta x_i \simeq 0 \quad \text{or} \quad \frac{\Delta x_i}{r_{k,i}^o} \simeq 0, \quad i = 1, 2, \dots, M, \quad k = 1, 2, \dots, N,$$

$$\Delta y_i \simeq 0 \quad \text{or} \quad \frac{\Delta y_i}{r_{k,i}^o} \simeq 0, \quad i = 1, 2, \dots, M, \quad k = 1, 2, \dots, N, \quad (3.68c)$$

$$\frac{\Delta \varphi(j)}{\varphi^o(j)} \simeq 0 \quad \text{for} \quad j = 1, 2, \dots, 4, 7, 8, \dots, 13, \quad (3.68d)$$

$$\frac{\Delta \varphi(j)}{r_{k,i}^o} \simeq 0 \quad \text{for} \quad j = 3, 4, \quad k = 1, 2, \dots, N. \quad (3.68e)$$

(3.68a) simply means that the frequency measurement noise and carrier frequency error are small relative to the Doppler shift, which is expected to be the case in order for localization using frequency observation possible. (3.68b)-(3.68c) require the position coordinate errors of sensor i be small compared to the true coordinate values and relative to its distance from the object at time k , $i = 1, 2, \dots, M$ and $k = 1, 2, \dots, N$. (3.68d)-(3.68e) demand the errors in the first stage solution sufficiently small compared to the true values and to the object-sensor distances for two of its values. (3.68d)-(3.68e) should be satisfied when (3.68a)-(3.68c) are fulfilled unless for

some unfavorable nearly degenerated localization geometry.

When the small noise conditions (3.68) are satisfied, we can validate that $\mathbf{A} \simeq \mathbf{A}^\circ$, $\mathbf{B} \simeq \mathbf{B}^\circ$, $\tilde{\mathbf{A}} \simeq \tilde{\mathbf{A}}^\circ$ and $\tilde{\mathbf{B}} \simeq \tilde{\mathbf{B}}^\circ$. Hence

$$\mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}} \simeq \frac{\partial \mathbf{f}^\circ}{\partial \boldsymbol{\theta}^{\circ T}}. \quad (3.69)$$

As a result, when (3.68) is fulfilled,

$$\text{cov}(\boldsymbol{\theta}) \simeq \text{CRLB}(\boldsymbol{\theta}). \quad (3.70)$$

The proposed algorithms utilize the Taylor-series expansion and contain some approximations. Nevertheless, the Taylor-series expansions are applied with respect to the measurement noise or sensor position errors, and the approximations come from maintaining in the expansions up to the first order error terms that are valid under the small noise conditions specified in (3.68). Albeit the expansions and approximations, the proposed algorithms are able to attain the lower bound performance over the small error region where the small noise conditions are satisfied.

3.4.2 Carrier Frequency Unavailable

The cost function for minimization in this case is (3.58). Expanding the function $\hat{f}_{k,i}(f_o, \boldsymbol{\theta})$ through the Taylor-series at the true value $(f_o^\circ, \boldsymbol{\theta}^\circ)$ gives

$$\begin{aligned} \hat{f}_{k,i}(f_o, \boldsymbol{\theta}) = \hat{f}_{k,i}^\circ + \frac{\partial \hat{f}_{k,i}}{\partial f_o} \Bigg|_{f_o^\circ, \boldsymbol{\theta}^\circ} (f_o - f_o^\circ) + \frac{\partial \hat{f}_{k,i}}{\partial \boldsymbol{\theta}^T} \Bigg|_{f_o^\circ, \boldsymbol{\theta}^\circ} (\boldsymbol{\theta} - \boldsymbol{\theta}^\circ) + o(f_o - f_o^\circ) \\ + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^\circ\|), \end{aligned} \quad (3.71)$$

where $\hat{f}_{k,i}^o = \hat{f}_{k,i}(f_o^o, \boldsymbol{\theta}^o)$. Up to the first order error, putting (3.71) in (3.58) and setting to zero the gradients with respect to f_o and $\boldsymbol{\theta}$, the deviation of the solution from the true value is

$$\begin{bmatrix} f_o - f_o^o \\ \boldsymbol{\theta} - \boldsymbol{\theta}^o \end{bmatrix} = \left(\begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o}, & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\epsilon^{-1} \begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o}, & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix} \right)^{-1} \begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o}, & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\epsilon^{-1} (\mathbf{f} - \hat{\mathbf{f}}^o). \quad (3.72)$$

Under (3.68a)-(3.68c), $\mathbf{f} - \hat{\mathbf{f}}^o$, $\partial \hat{\mathbf{f}}^o / \partial f_o^o$ and $\partial \hat{\mathbf{f}}^o / \partial \boldsymbol{\theta}^o$ can reasonably be approximated by $\boldsymbol{\epsilon}$, $\partial \mathbf{f}^o / \partial f_o^o$ and $\partial \mathbf{f}^o / \partial \boldsymbol{\theta}^o$. Thus, multiplying (3.72) by its transpose and taking expectation yield, up to the first order error,

$$\text{cov}(f_o, \boldsymbol{\theta}) = \left(\begin{bmatrix} \frac{\partial \mathbf{f}^o}{\partial f_o^o}, & \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\epsilon^{-1} \begin{bmatrix} \frac{\partial \mathbf{f}^o}{\partial f_o^o}, & \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix} \right)^{-1}. \quad (3.73)$$

After applying the block matrix inversion formula [104], substituting (3.6) with $\sigma_{f_o}^2 = 0$ and simplifying, the lower right block is the same as the right side of (2.42). Thus, the minimization of the cost function (3.58) will give an estimate having the CRLB accuracy, under the first order analysis where the small noise conditions (3.68a)-(3.68c) hold.

The proposed method finds the minimizer of (3.58) efficiently as follows. First, for a certain trial f_o , it minimizes (3.58) over $\boldsymbol{\theta}$. The corresponding solution has been derived in Section 3.2, with the setting that $\sigma_{f_o}^2 = 0$ for \mathbf{Q}_ϵ when forming \mathbf{W} . The resulting cost is obtained by (3.60). Second, a simple sequential search on f_o is able to yield the final solution that gives the smallest value of (3.60). This process essentially reaches the global minimizer of (3.58). As the global minimizer of (3.58) achieves the CRLB, the proposed solution is able to yield the CRLB accuracy. This conclusion

is based on the first order analysis under all small noise conditions in (3.68) when obtaining $\boldsymbol{\theta}$ by the algebraic solution, or under the conditions (3.68a)-(3.68c) when finding it by the SDR solution.

The proposed method handles the unknown carrier frequency scenario by a coarse one-dimensional grid search to locate the region where the global minimum of the cost function (3.58) lies and followed with the Newton-Raphson (NR) iteration. The NR method, perhaps, one of the most common iterative methods used in optimization. It converges at quadratic rate which is much faster than other methods with linear convergence [105]. The NR iteration will reach a minimum and achieve global convergence if the coarse grid search locates the global minimum region correctly. Using sufficiently small step-size in the grid search can take care of the global convergence issue. Having a too fine step-size, however, may result in unnecessary computation and increase processing time. Tradeoff between maintaining global convergence and reducing processing time should be considered when choosing the step-size for the grid search of the proposed method. In our simulation study, 300 grid points are used for the 1-D search to obtain the solution for the unavailable carrier frequency case with a strong convergence behavior.

3.4.3 Complexity

While both the algebraic closed-form and SDR estimators are based on the same WLS formulation, the computational complexity of the closed-form estimator is much lower than that of the SDR. The closed-form estimator applies unconstrained optimization in two stages to solve the constrained problem where explicit solutions of both stages exist. Albeit matrix inversion is needed, efficient implementation is commonly avail-

able. The SDR estimator solves the constrained optimization problem directly by SDR. The relaxed problem is convex and guarantees the global optimum solution, which, nevertheless, does not have an explicit solution. Finding the solution requires iterative numerical optimization procedure typically based on gradient or subgradient with the constraints imposed. Scaling of the values and parameters may also be needed to improve numerical accuracy. Consequently, the SDR estimator demands much larger amount of computation than the closed-form estimator. Using the generic convex optimization toolbox CVX, obtaining the SDR solution is at least 1300 times slower than using the closed-form estimator as shown in Table V from the simulation. Having said, the SDR estimator has the attractive aspect on performance where it outperforms the closed-form estimator considerably when the noise level becomes large, which will be demonstrated in the simulations.

3.5 Simulations

This section supports performance of the proposed methods by simulations. The scenario is for underwater acoustic application in which the object is moving at a constant speed of $v^o = \|\dot{\mathbf{u}}^o\| = 10$ m/s and radiates a single tone at $f_o^o = 15$ kHz [1, 81]. The signal propagation speed c is 1500 m/s [1, 83, 81]. The covariance matrices of frequency measurements are $\mathbf{Q}_k = \sigma^2 \mathbf{I}_M$, $k = 0, 1, \dots, N - 1$, and the unit of σ is Hz.

We shall use 10 different localization configurations. Each configuration places the sensors within a square area of 1.5×1.5 km² and the object inside 4×4 km² [1, 85, 84], both are centered at the origin, and their Cartesian coordinates are chosen randomly by following uniform distribution. The velocity direction is also created randomly

and it is different in each configuration. The performance measure is mean-square error (MSE) of the object location estimate that is computed by averaging over a number of ensemble runs and the 10 randomly created geometries. The number of ensemble runs is 1,000 for the carrier frequency available case and 100 for the carrier frequency unavailable situation, unless specified otherwise.

The proposed closed-form and SDR solutions will be denoted by CFS and SDP in the figures shown. The SDR solution is obtained using the CVX toolbox [94], where two different scaling factors, s_1 and s_2 , are applied to the sensor position vector \mathbf{s} defined in (2.6) and d_i given by (3.7), to handle the numerical aspect of CVX for achieving better accuracy.

3.5.1 Single-Time Observation

We use $M = 10$ sensors. For each random geometry generated, their positions satisfy $\|\mathbf{s}_i - \mathbf{s}_j\| \geq 150$ m for $i, j = 1, 2, \dots, M$ and $i < j$ to avoid near degenerated geometries. The scaling factors s_1 and s_2 for using CVX are 0.0018 and 0.15. When the carrier frequency is unavailable, a sequential search of 300 frequency points followed by the Newton-Raphson iteration with $\delta = 0.03$ is used. For comparison, the figures include the solutions from [88] that is denoted by Shames and [85] that is represented by Chan. Both are slightly modified to match the our problem scenario. The Shames method assumes the carrier frequency is known and accurate, and it is changed slightly to include some conditions from the measurement equations to improve performance of the polynomial optimization package used. The Chan method assumes the carrier frequency is unknown and it is modified to work with the case when the carrier frequency is known as well. The grid size of Chan method is adjusted

so that it takes comparable computation time as the proposed SDR solution. The CRLB derived in Section 2.6 is presented as a performance reference.

Fig. 3.1 shows the estimation accuracy of the proposed methods as the frequency measurement noise level σ increases when f_o is known exactly and the sensor position noise is absent. The MSE of the proposed CFS and SDP methods follow the CRLB performance well for both position and velocity estimation when σ is at a small to moderate level. CFS leaves the CRLB early and SDP is able to stay further with the CRLB as the noise level increases. Shames is unable to produce solutions for some ensemble runs as the noise level increases and its performance shown is by excluding those runs. Even so, it has suboptimal performance and cannot reach the CRLB regardless of the measurement noise level. Beyond the noise level of $\sigma = 10^{-1.25}$ Hz, it fails to provide a solution of a configuration in all ensemble runs. Chan has poor performance due to insufficient grid resolution for maintaining close computational complexity with SDP. In term of the relative processing times among the algorithms for this simulation that are obtained by Matlab implementation, Table 3.5 indicates that they do not vary much with respect to the noise level. Most importantly, it shows that CFS outperforms all the other methods significantly by a big margin. SDP and Chan follow next and they require almost half the time needed for Shames. The relative computation time of SDP in Table 3.5 is obtained from the universal solver, i.e., the CVX implementation. When the code for the SDP problem is specially designed, the computation time could be considerably reduced.

Fig. 3.2 illustrates the performance when the carrier frequency f_o is completely not known, where the other settings remain the same as in Fig. 3.1. The absolute performance drops compared to Fig. 3.1 as we have an additional unknown of the

Table 3.5: Relative Processing Times of Different Algorithms

σ (Hz)	Relative Processing Time			
	CFS	SDP	Shames	Chan
0.001	1	1317	2665	1460
0.01	0.98	1414	3238	1541
0.05	1.05	1416	3169	1508
0.1	1.05	1314	3215	1545

carrier frequency for estimation. Nevertheless, the relative performance among the algorithms remains similar.

We next examine in Fig. 3.3 the estimation performance with respect to the amount of sensor position error, where the measurement noise and carrier frequency error are absent to highlight its effect on localization. The covariance matrix of sensor positions is diagonal with the elements randomly set as

$$\mathbf{Q}_s = \sigma_s^2 \text{diag}\{[0.638, 0.675, 0.927, 1.028, 0.949, 0.624, 0.359, 0.401, 0.559, 0.933, 0.641, 0.971, 0.872, 0.383, 0.818, 1.082, 0.694, 0.339, 1.081, 0.622]\}, \quad (3.74)$$

and the unit of σ_s is m.

Without accounting for the sensor position error by setting $\sigma_s = 0$ in the CFS algorithm, the MSE is nearly 4 dB higher than the CRLB before the accuracy deviates significantly from the bound. When accounting for the sensor position error, both CFS and SDP yield the CRLB performance and the latter remains close to the bound even when the sensor position error is large.

Fig. 3.4 evaluates the performance as the accuracy of the available carrier frequency decreases, where the measurement noise is at a low level of $\sigma = 0.001$ Hz and

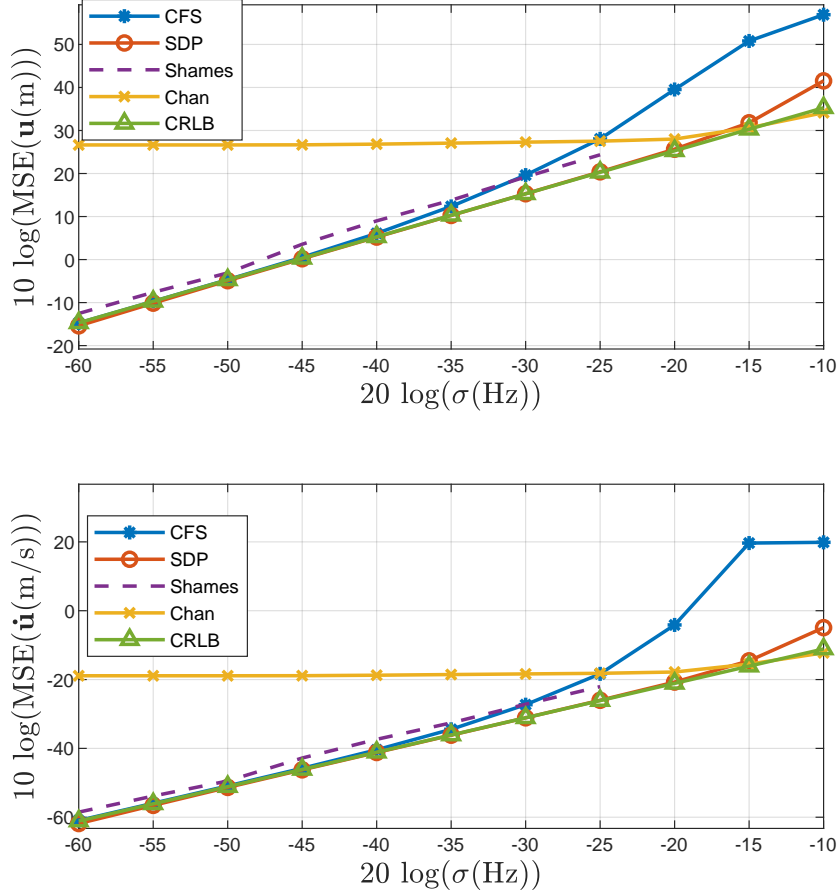


Figure 3.1: Performance of the proposed methods at different σ levels for single-time measurement when f_o is available. (a) position estimation, (b) velocity estimation.

the sensor position noise is absent to emphasize the effect of the carrier frequency error. Both CFS and SDP achieve the CRLB accuracy over the small error region and have close performance. If we always assume the available carrier frequency is accurate by setting $\sigma_{f_o} = 0$ in CFS, the performance is much worse and the degradation is more significant for velocity estimation, unless the frequency error is very small so that the accuracy is dominated by the measurement noise. The results indicate the importance of taking the error of the carrier frequency into consideration when

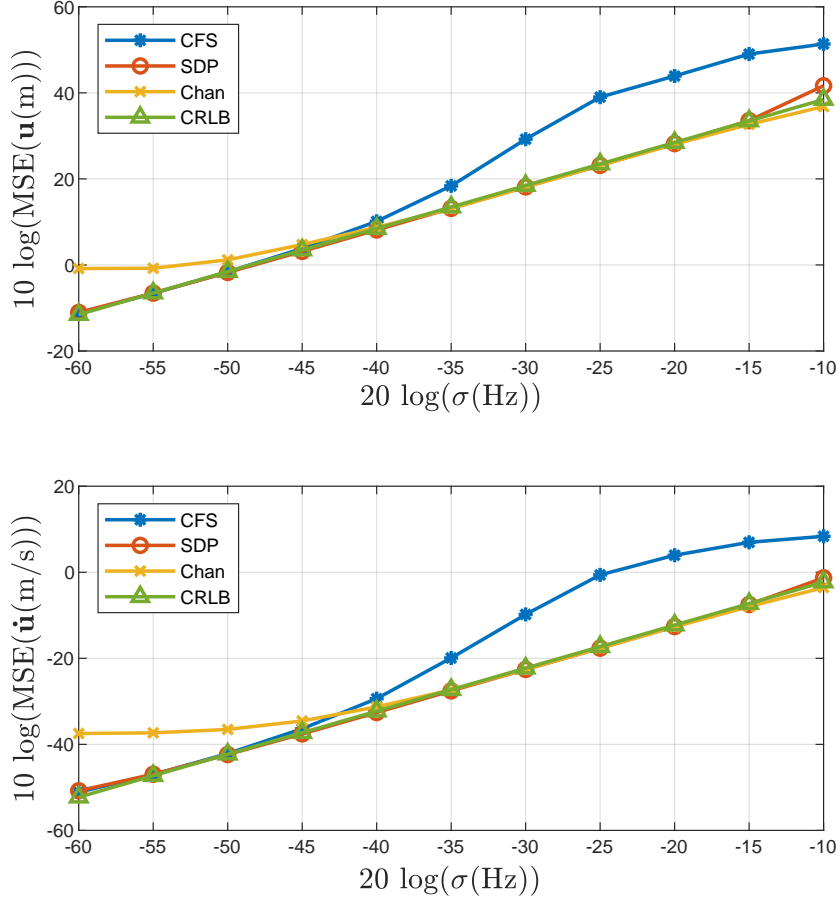


Figure 3.2: Performance of the proposed methods at different σ levels for single-time measurement when f_o is unavailable. (a) position estimation, (b) velocity estimation.

designing a localization algorithm.

3.5.2 Multiple-Time Observations

We use $M = 7$ sensors and the minimum distance among the sensors in each randomly generated geometry is 350 m. The number of successive measurements for each sensor is $N = 35$. The processing steps 8 and 9 in Algorithm 2 of CFS are repeated 5 times. The scaling factors s_1 and s_2 for use with CVX in obtaining the SDR solution are

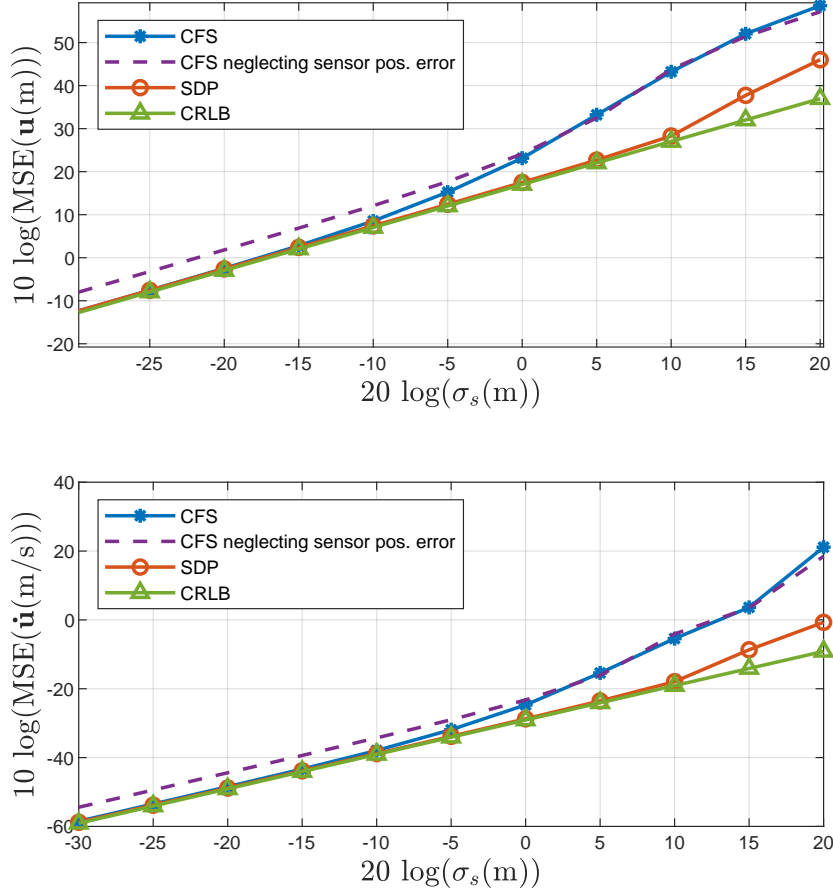


Figure 3.3: Performance of the proposed methods at different σ_s levels for single-time measurement when f_o is available. (a) position estimation, (b) velocity estimation.

0.0175 and 0.001. In the case of unavailable carrier frequency, the grid search has about 300 frequency points and the Newton-Raphson iteration uses $\delta = 0.001$.

Fig. 3.5 gives the estimation performance as the noise level σ increases when the carrier frequency is exactly known and the sensor position error is absent. SDP appears to deviate from the CRLB performance in the velocity estimate when the measurement noise is very small, possibly due to the relaxation. As σ increases, CFS leaves the CRLB while SDP remains to follow the bound much further for larger σ .

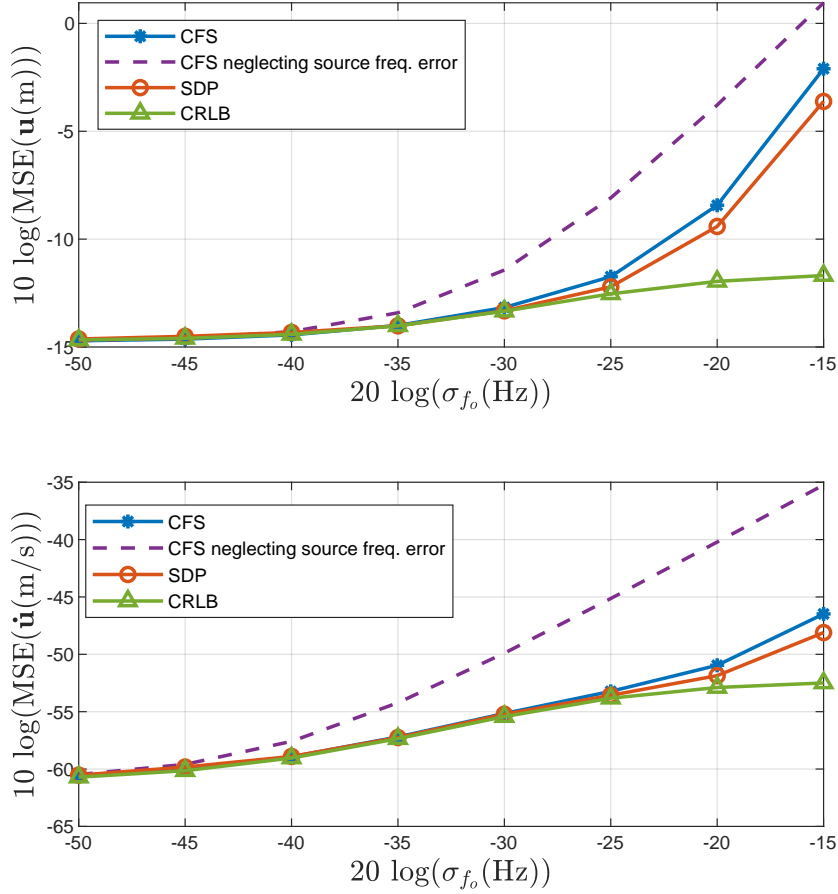


Figure 3.4: Performance of the proposed methods at different σ_{f_o} levels for single-time measurement. (a) position estimation, (b) velocity estimation.

The results for the unknown carrier frequency case are shown in Fig. 3.6, where the sensor positions are exact. The observations have close agreement with Fig. 3.5. As the absolute localization performance reduces when the carrier frequency is not known, SDP can reach the CRLB in velocity even when the carrier frequency error is small in this simulation.

3.5.3 Number of Sensors

We examine the number of sensors needed for the proposed CFS and SDR algorithms in reaching the optimal accuracy. The number of ensemble runs for this simulation is 10,000. Fig. 3.7 illustrates the performance for the single-time measurement case as the number of sensors increases. The setting follows that of Fig. 3.1, except the object is placed in the same area as the sensors [88] for generating the 10 randomly formed geometries. The measurement noise level is kept at $\sigma = 0.1$ Hz. When applying Algorithm 1 for CFS, steps 9-11 are repeated 5 times to improve $\widetilde{\mathbf{W}}$ using $\boldsymbol{\theta}$ from step 11 to form $\boldsymbol{\varphi}$ by (3.14), as the object can be near the sensors in some of the 10 geometries. CFS requires at least 9 sensors to operate properly as expected by theory. SDP attains the CRLB performance starting at 7 sensors. The Shames algorithm is able to give reasonable results if we have 6 sensors. Even so, it is not able to produce a solution 131 times, 1 time, 1 time and 2 times when using 6, 7, 8 and 9 sensors, and those are excluded when obtaining its MSE performance. Furthermore, it cannot reach the CRLB when the number of sensors increases and has twice the complexity of SDP (see Table 3.5). Performance of the Chan method is limited by the insufficiently small grid size for the solution search, although it is expected to function with 5 sensors [85].

Fig. 3.8 shows the results for the case of multiple-time observations with setting that corresponds to Fig. 3.5, except the object is placed in the same area as for the sensors. The measurement noise level is fixed at $\sigma = 0.1$ Hz. Steps 8 and 9 in Algorithm 2 for CFS is repeated 5 times. CFS now only needs 6 sensors to operate. SDP also reduces the number of sensors needed for functioning well to 4.

A minimum of two sensors can be sufficient for localization in the multiple-time measurements case when analyzing the CRLB. It appears the proposed closed-form and SDR estimators require more sensors than needed for localization. It is our next research investigation to improve the algorithms by reducing the number of sensors they need to yield an accurate solution.

Both the closed form solution CFS and the semi-definite relaxation SDP algorithm are based on the same WLS formulation with a set of constraints. In terms of complexity, the CFS estimator is far more attractive than the SDR, as discussed in Section 3.4.3 and validated by Table 3.5. It works better with more regular localization geometry such as those having near optimum sensor placement [106]. CFS has the limitation that it requires additional sensors and high SNR (small frequency observation noise) to function well and it can only reach the CRLB performance under the small noise conditions specified in (3.68). In terms of performance, the SDR estimator is much better in handling poorer localization geometry, operating with fewer sensors and working at lower SNR. While the SDR solution yields better performance, careful scaling of the input values and parameters may be needed to improve the numerical accuracy in which the scaling adjustment could be dependent on the SDP solver used. Incorporating an SDP solver may require certain hardware and software that could be prohibitive in some practical applications. Generally speaking, the CFS algorithm should be used under an environment where the localization geometry is typical, the number of sensors is sufficient and the SNR is high, or when the computation complexity is an important factor for consideration. In the situation where the number of sensors is limited or the SNR is low while the complexity is not a crucial factor, the SDR solution is recommended.

The proposed algorithms appear to require small frequency measurement errors to operate well, especially for CFS. CFS can serve as an effective initialization to the iterative implementation of the ML Estimator, especially in the single-time measurement case where the deviation from the CRLB seems gradual. The low noise requirement, to some extent, comes from the frequency only localization problem itself, which has been illustrated in the prior research studies [85, 84, 88]. The recent publication [59] has shown that frequency estimation error σ in the order of 10 mHz is achievable for the sonar scenario with a carrier frequency of 15 kHz, a data length of 0.2 sec and an SNR of 0 dB. We shall continue to improve the proposed algorithms, and evaluate their applicability in practical environments where high SNR for having low frequency error may not be guaranteed.

3.6 Summary

This chapter develops several algorithms and investigates their performance for passive localization of a moving object using the Doppler shifted frequency measurements in the 2-D space. We take a comprehensive treatment to the problem by considering the carrier frequency is known but having errors or is completely not known, and the sensors may have position errors. Moreover, the cases of single-time measurements and multiple-time observations are studied. We propose a novel transformation for the localization problem to become a quadratic optimization with a set of quadratic constraints. Two methods are derived to solve the constrained optimization problem, one is the computationally efficient closed-form solution and the other is the noise resilient SDR solution. The closed-form solution is shown theoretically to reach the CRLB performance under small Gaussian noise while the SDR solution can include

many additional constraints to increase the robustness against noise. Simulations support the proposed solutions in reaching the optimal performance, and the better accuracy and computational efficiency than the existing methods from the literature.

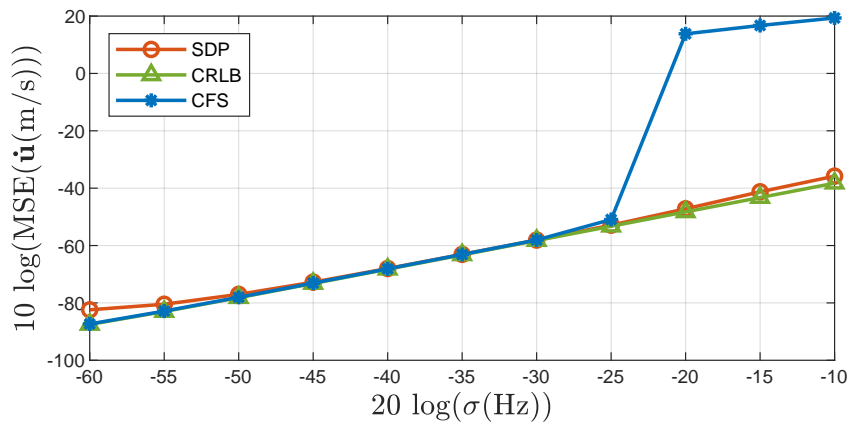
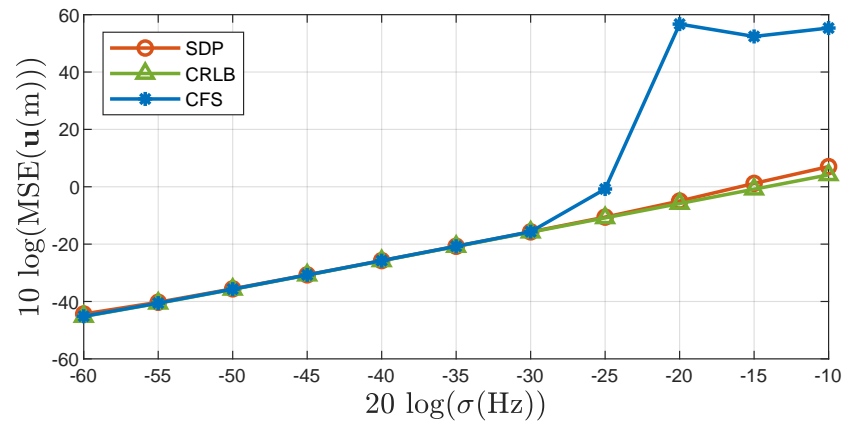


Figure 3.5: Performance of the proposed methods at different σ levels for multiple-time measurements when f_o is available. (a) position estimation, (b) velocity estimation.

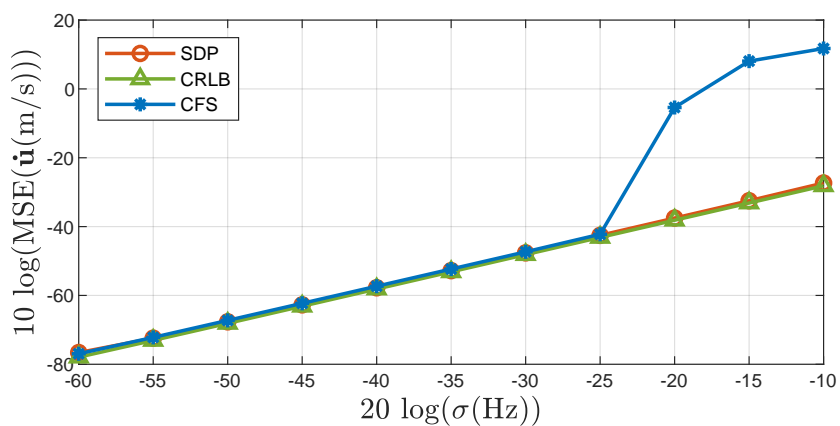
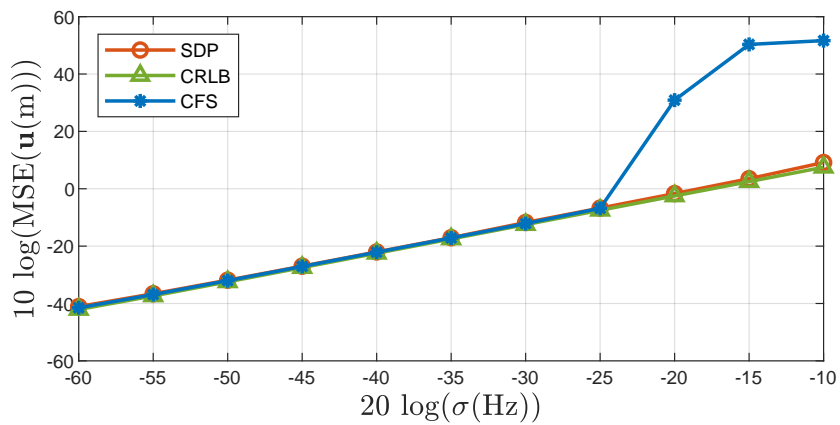


Figure 3.6: Performance of the proposed methods at different σ levels for multiple-time measurements when f_o is unavailable. (a) position estimation, (b) velocity estimation.

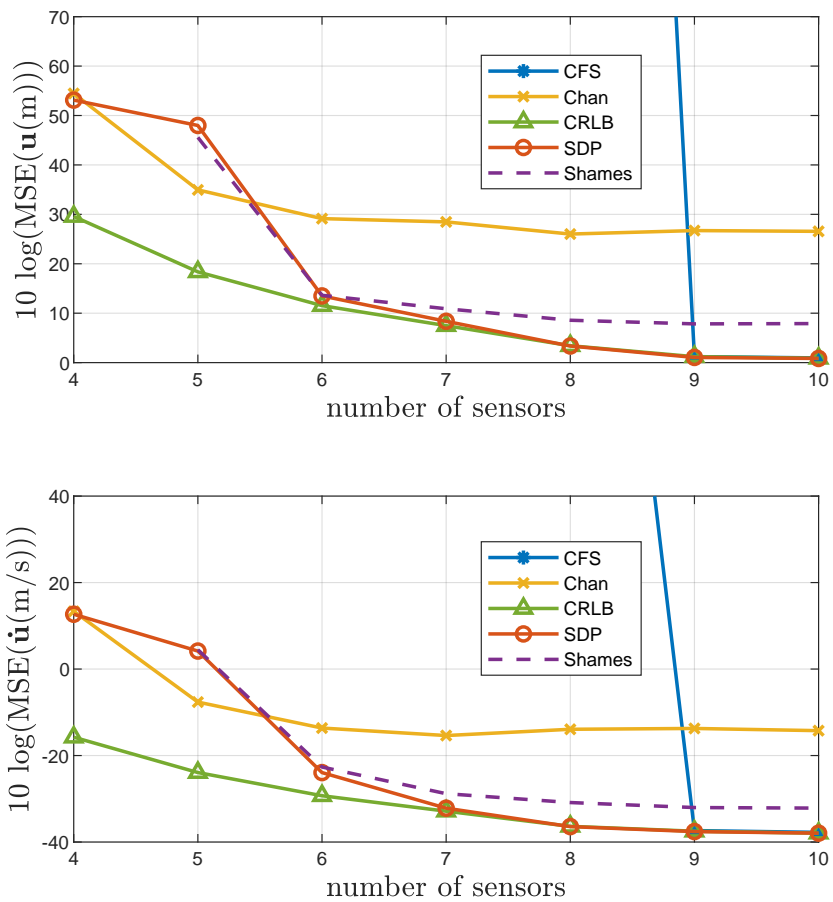


Figure 3.7: Performance of the proposed methods using different number of sensors for single-time measurement when f_o is available. (a) position estimation, (b) velocity estimation.

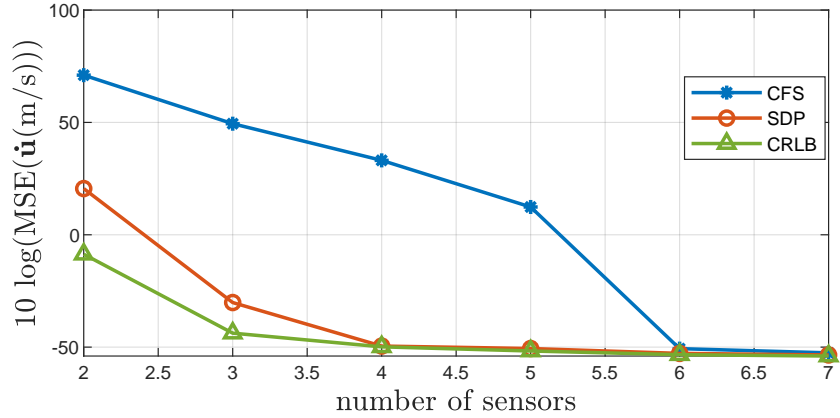
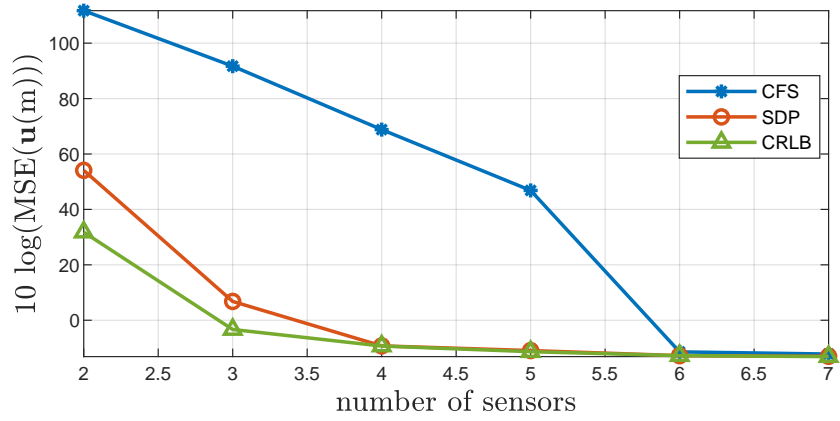


Figure 3.8: Performance of the proposed methods using different number of sensors for multiple-time measurements when f_o is available. (a) position estimation, (b) velocity estimation.

Chapter 4

3-D Object Localization

This Chapter study the problem of moving object localization in the 3-D space. 2-D and 3-D localization problems are different in many aspects. First, 2-D scenario is much simpler and has fewer unknowns than the 3-D situation. Second, the number of auxiliary variables associated with the solution formulation in 3-D is increased by five for the localization case and by seven for the multiple-time case compared to the 2-D situation. The consequence is a considerable increase in the number of relations among the independent unknowns and the nuisance variables, resulting in a significant jump in the number of constraints that are far more complex to handle, especially in the SDP solution. We find all the constraints, select those that are relevant for improving performance and provide them in this Chapter. Third, the investigation on sequential analysis, the proposed sequential estimator and its analysis for the multiple-time case are new and they were not considered in Chapter 3 at all. Additionally, this Chapter examines the effectiveness of the proposed CFS solution as an initialization to the iterative Maximum Likelihood Estimator (MLE). Fig. 4.1

illustrates the localization scenario in 3-D space.

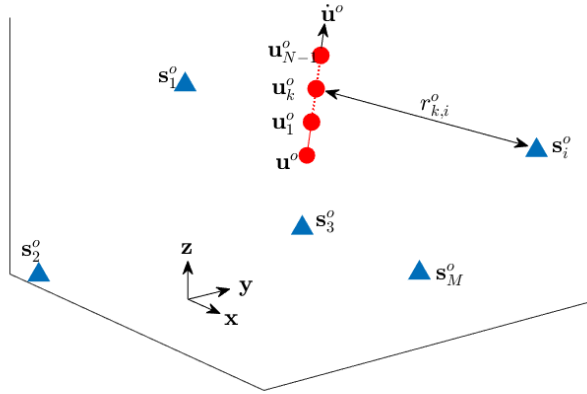


Figure 4.1: 3-D localization scenario of a moving object.

4.1 Formulation

The frequency measurement model (2.2) is nonlinear with respect to the unknowns. Direct evaluation of (2.2) to estimate the location parameters of the object is difficult. While it is possible to use numerical grid search to obtain the solution based on (2.2), it will be time consuming and the accuracy is limited by the grid resolution.

We shall transform the measurement model to a pseudo linear form under the assumption that the noise is not significant in which the second and higher order noise terms are negligible. The new model enables the formulation of a weighted least-squares (WLS) optimization problem with several quadratic constraints relating the unknowns. Two solutions will be proposed to solve the WLS problem. The first is

by algebraic evaluation to reach a CFS and the second is through convex optimization by SDR. Each of them shall be derived separately for the single-time and multiple-time cases. Also, a separate approach and solutions for the scenarios of available and unavailable carrier frequency shall be considered.

4.1.1 Carrier Frequency Available

We shall first express the measurement model (2.2) in terms of the available carrier frequency and sensor positions. Let

$$r_{k,i} = \|\mathbf{u}_k^o - \mathbf{s}_i\|, \quad (4.1)$$

be the Euclidean distance between the object at time k and the inaccurate sensor position \mathbf{s}_i , and

$$\boldsymbol{\rho}_{k,i}^o = (\mathbf{u}_k^o - \mathbf{s}_i^o) / \|\mathbf{u}_k^o - \mathbf{s}_i^o\|, \quad (4.2)$$

be a unit vector pointing from \mathbf{s}_i^o to \mathbf{u}_k^o . Using (2.5) and (2.6), the Taylor-series expansion of the frequency shift term in (2.2) at (f_o, \mathbf{s}_i) is, up to the first order error terms

$$\frac{f_o^o (\mathbf{u}_k^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|} \simeq \frac{f_o (\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o}{c r_{k,i}} - \frac{\boldsymbol{\rho}_{k,i}^{oT} \dot{\mathbf{u}}^o}{c} \Delta f_o + \frac{f_o \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o}}{c r_{k,i}^o} \Delta \mathbf{s}_i, \quad (4.3)$$

where $r_{k,i}^o = \|\mathbf{u}_k^o - \mathbf{s}_i^o\|$ is the true value of $r_{k,i}$ and $\mathbf{P}_{k,i}^{\perp o} = \mathbf{I} - \boldsymbol{\rho}_{k,i}^o \boldsymbol{\rho}_{k,i}^{oT}$ is the orthogonal projection matrix of $\boldsymbol{\rho}_{k,i}^o$.

In terms of the range rate $d_{k,i}$ rather than the observed frequency $f_{k,i}$, it is after using (2.5) and (4.3),

$$d_{k,i} = \frac{c(f_{k,i} - f_o)}{f_o} = -\frac{(\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o}{r_{k,i}} + \frac{c}{f_o} \varepsilon_{k,i}, \quad (4.4)$$

where the equality is used instead of approximation by assuming the higher order error terms are too small to be considered further. $\varepsilon_{k,i}$ is the total error equal to

$$\varepsilon_{k,i} = \left(\frac{\boldsymbol{\rho}_{k,i}^{oT} \dot{\mathbf{u}}^o - c}{c} \right) \Delta f_o + \frac{-f_o \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o}}{c r_{k,i}^o} \Delta \mathbf{s}_i + n_{k,i}. \quad (4.5)$$

For all measurements, we have

$$\mathbf{d} = [\mathbf{d}_1^T, \mathbf{d}_2^T \dots, \mathbf{d}_N^T]^T, \quad (4.6)$$

in which $\mathbf{d}_k = [d_{k,1}, \dots, d_{k,M}]^T$ and $\boldsymbol{\varepsilon}_k = [\varepsilon_{k,1}, \dots, \varepsilon_{k,M}]^T$, $k = 0, 1, \dots, N - 1$. From (4.5), $\boldsymbol{\varepsilon}_k$ is

$$\boldsymbol{\varepsilon}_k = \mathbf{d}_{f,k} \Delta f_o + \mathbf{D}_{s,k} \Delta \mathbf{s} + \mathbf{n}_k, \quad (4.7)$$

where $\mathbf{d}_{f,k}$ is the $M \times 1$ vector whose i -th element is $(\boldsymbol{\rho}_{k,i}^{oT} \dot{\mathbf{u}}^o - c)/c$, and $\mathbf{D}_{s,k}$ is the $M \times 3M$ matrix whose i -th row is zero except the elements $\mathbf{D}_{s,k}(i, 3(i-1) + 1 : 3i) = -f_o \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o} / (c r_{k,i}^o)$. Hence

$$\boldsymbol{\varepsilon} = \mathbf{d}_f \Delta f_o + \mathbf{D}_s \Delta \mathbf{s} + \mathbf{n}, \quad (4.8)$$

in which the vector $\mathbf{d}_f = [\mathbf{d}_{f,0}^T, \mathbf{d}_{f,1}^T, \dots, \mathbf{d}_{f,N-1}^T]^T$ and the matrix $\mathbf{D}_s = [\mathbf{D}_{s,0}^T, \mathbf{D}_{s,1}^T, \dots, \mathbf{D}_{s,N-1}^T]^T$. $\boldsymbol{\varepsilon}$ is zero-mean Gaussian distributed with the covariance matrix equal to

$$\mathbf{Q}_\varepsilon = E[\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T] = \sigma_{f_o}^2 \mathbf{d}_f \mathbf{d}_f^T + \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{Q}_n. \quad (4.9)$$

Rearranging (4.4) gives

$$(d_{k,i} - c \varepsilon_{k,i} / f_o) r_{k,i} = -(\mathbf{u}_k^o - \mathbf{s}_i)^T \dot{\mathbf{u}}^o. \quad (4.10)$$

If we square (4.10), apply (4.1) and neglect the second order error term,

$$-2cd_{k,i}r_{k,i}^2\varepsilon_{k,i}/f_o+d_{k,i}^2(\|\mathbf{s}_i\|^2-2\mathbf{s}_i^T\mathbf{u}_k^o+\|\mathbf{u}_k^o\|^2)=(\mathbf{s}_i^T\dot{\mathbf{u}}^o)^2-2\mathbf{s}_i^T\dot{\mathbf{u}}^o\dot{\mathbf{u}}^{oT}\mathbf{u}_k^o+(\mathbf{u}_k^{oT}\dot{\mathbf{u}}^o)^2. \quad (4.11)$$

Let us define

$$\mathbf{s}_{iQ}=[x_i^2, y_i^2, z_i^2, 2x_iy_i, 2x_iz_i, 2y_iz_i]^T, \quad (4.12)$$

$$\mathbf{v}_Q=[\dot{x}^{o2}, \dot{y}^{o2}, \dot{z}^{o2}, \dot{x}^o\dot{y}^o, \dot{x}^o\dot{z}^o, \dot{y}^o\dot{z}^o]^T, \quad (4.13)$$

such that

$$(\mathbf{s}_i^T\dot{\mathbf{u}}^o)^2=\mathbf{s}_{iQ}^T\mathbf{v}_Q. \quad (4.14)$$

Thus, (4.11) can be expressed as

$$2cd_{k,i}r_{k,i}^2\varepsilon_{k,i}/f_o=d_{k,i}^2\|\mathbf{s}_i\|^2-2d_{k,i}^2\mathbf{s}_i^T\mathbf{u}_k^o+d_{k,i}^2\|\mathbf{u}_k^o\|^2-\mathbf{s}_{iQ}^T\mathbf{v}_Q+2\mathbf{s}_i^T\dot{\mathbf{u}}^o\dot{\mathbf{u}}^{oT}\mathbf{u}_k^o-(\mathbf{u}_k^{oT}\dot{\mathbf{u}}^o)^2. \quad (4.15)$$

Section 4.2 proposes two methods to solve (4.15) for the localization case and Section 4.3 for the multiple-time situation.

4.1.2 Carrier Frequency Unavailable

The carrier frequency f_o is regarded as an additional unknown. The Taylor-series expansion of (2.2) at \mathbf{s}_i is, after keeping only the first order error term,

$$f_{k,i}=f_o-\frac{f_o(\mathbf{u}_k^o-\mathbf{s}_i)^T\dot{\mathbf{u}}^o}{cr_{k,i}}+\varepsilon_{k,i}, \quad (4.16)$$

where the composite error term $\varepsilon_{k,i}$ is now

$$\varepsilon_{k,i} = -\frac{f_o \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o}}{c r_{k,i}^o} \Delta \mathbf{s}_i + n_{k,i}, \quad (4.17)$$

and $\mathbf{P}_{k,i}^{\perp o}$ is given below (4.3). The covariance matrix of the zero-mean error vector $\boldsymbol{\varepsilon}$ from all MN measurements is

$$\mathbf{Q}_\varepsilon = \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{Q}_n, \quad (4.18)$$

and \mathbf{D}_s is shown under (4.7) and (4.8). Section 4.4 presents the solution in this case.

4.2 Single-Time Observation

Having one set of measurement at a given time, setting $k = 0$ and dropping this subscript in (4.15) give

$$2cd_i r_i^2 \varepsilon_i / f_o = d_i^2 \|\mathbf{s}_i\|^2 - 2d_i^2 \mathbf{s}_i^T \mathbf{u}^o + d_i^2 \|\mathbf{u}^o\|^2 - \mathbf{s}_{iQ}^T \mathbf{v}_Q + 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}^o - (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2. \quad (4.19)$$

Let $\boldsymbol{\varphi}^o$ be the unknown vector constructed as

$$\boldsymbol{\varphi}^o = \left[\mathbf{u}^{oT}, \|\mathbf{u}^o\|^2, \mathbf{v}_Q^T, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT}, (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 \right]^T. \quad (4.20)$$

Also, define

$$\mathbf{B} = 2 \frac{c}{f_o} \text{diag} \{ [d_1 r_1^2, d_2 r_2^2, \dots, d_M r_M^2] \}, \quad (4.21)$$

$$\mathbf{A} = \begin{bmatrix} 2d_1^2 \mathbf{s}_1^T & -d_1^2 & \mathbf{s}_{1Q}^T & -2\mathbf{s}_1^T & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2d_M^2 \mathbf{s}_M^T & -d_M^2 & \mathbf{s}_{MQ}^T & -2\mathbf{s}_M^T & 1 \end{bmatrix}, \quad (4.22)$$

$$\mathbf{h} = [d_1^2 \|\mathbf{s}_1\|^2, d_2^2 \|\mathbf{s}_2\|^2, \dots, d_M^2 \|\mathbf{s}_M\|^2]^T, \quad (4.23)$$

where \mathbf{B} is $M \times M$, \mathbf{A} is $M \times 14$ and \mathbf{h} is $M \times 1$. (4.19) can be expressed in a matrix form over $i = 1, 2, \dots, M$ as

$$\mathbf{B}\boldsymbol{\varepsilon} = \mathbf{h} - \mathbf{A}\boldsymbol{\varphi}^\circ. \quad (4.24)$$

The weighting matrix of (4.24) is set as the covariance matrix inverse of $\mathbf{B}\boldsymbol{\varepsilon}$,

$$\mathbf{W} = \mathbf{B}^{-T} E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]^{-1} \mathbf{B}^{-1} = \mathbf{B}^{-T} \mathbf{Q}_\varepsilon^{-1} \mathbf{B}^{-1}, \quad (4.25)$$

where \mathbf{Q}_ε is given by (4.9) with $N = 1$.

Although the number of elements in $\boldsymbol{\varphi}^\circ$ is 14, it has only 6 independent variables. To build an estimator that solves for the actual unknowns \mathbf{u} and $\dot{\mathbf{u}}$ and maintains the same degrees of freedom in finding them, we shall have eight constraints to relate the elements of $\boldsymbol{\varphi}^\circ$. The constraints can be used along with (4.24) and (4.25) to form a WLS optimization problem given by:

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (4.26a)$$

$$\text{s.t. } \varphi(4) = \varphi(1)^2 + \varphi(2)^2 + \varphi(3)^2, \quad (4.26b)$$

$$\varphi(8)^2 = \varphi(5)\varphi(6), \quad (4.26c)$$

$$\varphi(9)^2 = \varphi(5)\varphi(7), \quad (4.26d)$$

Table 4.1: Relations Among the Elements of $\boldsymbol{\varphi}$ for the Single-Time Case

Elements	Relations
$\varphi(1) = x$	$\varphi(4) = \varphi(1)^2 + \varphi(2)^2 + \varphi(3)^2$
$\varphi(2) = y$	$\varphi(8)^2 = \varphi(5)\varphi(6)$
$\varphi(3) = z$	$\varphi(9)^2 = \varphi(5)\varphi(7)$
$\varphi(4) = x^2 + y^2 + z^2$	$\varphi(10)^2 = \varphi(6)\varphi(7)$
$\varphi(5) = \dot{x}^2$	$\varphi(11) = \varphi(1)\varphi(5) + \varphi(2)\varphi(8) + \varphi(3)\varphi(9)$
$\varphi(6) = \dot{y}^2$	$\varphi(12) = \varphi(1)\varphi(8) + \varphi(2)\varphi(6) + \varphi(3)\varphi(10)$
$\varphi(7) = \dot{z}^2$	$\varphi(13) = \varphi(1)\varphi(9) + \varphi(2)\varphi(10) + \varphi(3)\varphi(7)$
$\varphi(8) = \dot{x}\dot{y}$	$\varphi(14) = \varphi(1)\varphi(11) + \varphi(2)\varphi(12) + \varphi(3)\varphi(13)$
$\varphi(9) = \dot{x}\dot{z}$	
$\varphi(10) = \dot{y}\dot{z}$	$\varphi(5)\varphi(10) = \varphi(8)\varphi(9)$
$\varphi(11) = \dot{x}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(5)\varphi(12) = \varphi(8)\varphi(11)$
$\varphi(12) = \dot{y}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(5)\varphi(13) = \varphi(9)\varphi(11)$
$\varphi(13) = \dot{z}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(5)\varphi(14) = \varphi(11)\varphi(11)$
$\varphi(14) = (x\dot{x} + y\dot{y} + z\dot{z})^2$	$\varphi(6)\varphi(9) = \varphi(8)\varphi(10)$
	$\varphi(6)\varphi(11) = \varphi(8)\varphi(12)$
	$\varphi(6)\varphi(13) = \varphi(10)\varphi(12)$
	$\varphi(6)\varphi(14) = \varphi(12)\varphi(12)$
	$\varphi(7)\varphi(8) = \varphi(9)\varphi(10)$
	$\varphi(7)\varphi(11) = \varphi(9)\varphi(13)$
	$\varphi(7)\varphi(12) = \varphi(10)\varphi(13)$
	$\varphi(7)\varphi(14) = \varphi(13)\varphi(13)$
	$\varphi(8)\varphi(13) = \varphi(9)\varphi(12)$
	$\varphi(8)\varphi(14) = \varphi(11)\varphi(12)$
	$\varphi(9)\varphi(12) = \varphi(10)\varphi(11)$
	$\varphi(9)\varphi(14) = \varphi(11)\varphi(13)$
	$\varphi(10)\varphi(14) = \varphi(12)\varphi(13)$

$$\varphi(10)^2 = \varphi(6)\varphi(7), \quad (4.26e)$$

$$\varphi(11) = \varphi(1)\varphi(5) + \varphi(2)\varphi(8) + \varphi(3)\varphi(9), \quad (4.26f)$$

$$\varphi(12) = \varphi(1)\varphi(8) + \varphi(2)\varphi(6) + \varphi(3)\varphi(10), \quad (4.26g)$$

$$\varphi(13) = \varphi(1)\varphi(9) + \varphi(2)\varphi(10) + \varphi(3)\varphi(7), \quad (4.26h)$$

$$\varphi(14) = \varphi(1)\varphi(11) + \varphi(2)\varphi(12) + \varphi(3)\varphi(13). \quad (4.26i)$$

Table 4.1 shows the individual components of $\boldsymbol{\varphi}$ and lists all linear and quadratic relations among them. The first eight constraints on the right of Table 4.1 are taken for

the optimization problem (4.26) and the rest are redundant for the WLS formulation. They will be used when we obtain the SDP solution later in this section. We continue and develop two solutions to solve the constrained optimization problem (4.26).

4.2.1 Algebraic Solution

In this solution method, we assume the elements of φ^o are independent and find its estimate using the direct WLS solution first. Then, the eight quadratic constraints (4.26b)-(4.26i) are used to build another WLS problem through nonlinear transformation to refine the first solution estimate [12].

The WLS solution to (4.26) when ignoring the constraints is

$$\varphi = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{h}. \quad (4.27)$$

The weighting matrix \mathbf{W} is given by (4.25). It requires the true object location to evaluate \mathbf{B} and \mathbf{Q}_ϵ . WLS optimization is not sensitive to small error in the weighting matrix [9, 10] and we can approximate \mathbf{W} using the object location obtained from the least-squares solution by setting $\mathbf{W} = \mathbf{I}$ in (4.27).

The estimation error $\Delta\varphi = \varphi - \varphi^o$ of (4.27) has the covariance matrix that can be approximated by [12, 104]

$$\text{cov}(\Delta\varphi) = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1}, \quad (4.28)$$

where the noise in \mathbf{A} is assumed too small and can be neglected.

We next incorporate (4.28) and the constraints (4.26b)-(4.26i) to perform the refinement process of φ through the second WLS optimization. Let $\tilde{\varphi}^o$ be the unknown

vector that contains the six independent object location parameters,

$$\tilde{\varphi}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT} \otimes \dot{\mathbf{u}}^{oT}]^T, \quad (4.29)$$

and $\tilde{\mathbf{h}}$ be the pseudo data vector constructed from $\boldsymbol{\varphi}$,

$$\tilde{\mathbf{h}} = [\boldsymbol{\varphi}^T(1:7), \varphi^2(8), \varphi^2(9), \varphi^2(10), \boldsymbol{\varphi}^T(11:14)]^T. \quad (4.30)$$

The nonlinear transformation is done through relating the elements of $\tilde{\mathbf{h}}$ with those of $\tilde{\varphi}^o$. It is recognized that

$$\begin{aligned} \tilde{\mathbf{h}}(1:3) &= \tilde{\varphi}^o(1:3) + \Delta\boldsymbol{\varphi}(1:3), \\ \tilde{\mathbf{h}}(5:7) &= \tilde{\varphi}^o(4:6) + \Delta\boldsymbol{\varphi}(5:7). \end{aligned} \quad (4.31)$$

From (4.26b)-(4.26i) and dropping the second error terms, we are able relate the rest of the elements of $\tilde{\mathbf{h}}$ to $\tilde{\varphi}^o$ that are shown in Appendix B. Putting (4.31) and (B.1) in matrix form yields

$$\tilde{\mathbf{h}} \simeq \tilde{\mathbf{A}}\tilde{\varphi}^o + \tilde{\mathbf{B}}\Delta\boldsymbol{\varphi}, \quad (4.32)$$

in which $\widetilde{\mathbf{W}}$ approximates $E[\widetilde{\mathbf{B}}\Delta\boldsymbol{\varphi}\Delta\boldsymbol{\varphi}^T\widetilde{\mathbf{B}}^T]^{-1}$ by using (4.28),

$$\widetilde{\mathbf{W}} = \widetilde{\mathbf{B}}^{-T}(\mathbf{A}^T\mathbf{W}\mathbf{A})\widetilde{\mathbf{B}}^{-1}. \quad (4.37)$$

The covariance matrix, after neglecting the noise in $\widetilde{\mathbf{A}}$ under small noise assumption, is

$$\text{cov}(\widetilde{\boldsymbol{\varphi}}) = (\widetilde{\mathbf{A}}^T\widetilde{\mathbf{W}}\widetilde{\mathbf{A}})^{-1}. \quad (4.38)$$

The matrix $\widetilde{\mathbf{B}}$ needs the true object location for \mathbf{C} in (4.35) and it is not known. Using the solution of (4.27) can approximate $\widetilde{\mathbf{B}}$ to obtain $\widetilde{\boldsymbol{\varphi}}$. To increase the accuracy, we can iterate (4.36) a few times by using the last update of $\widetilde{\boldsymbol{\varphi}}$ for \mathbf{C} in (4.35).

Since $\widetilde{\boldsymbol{\varphi}}$ contains only the squared velocity parameters, further processing is needed to fix the sign of $\dot{\mathbf{u}}$. This is done by trying all eight possible sign permutations and select the one that gives the smallest approximate ML cost function having f_o^o and \mathbf{s}_i^o replaced by f_o and \mathbf{s}_i . The final solution is

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{\varphi}}(1:3) \\ \mathbf{P}\sqrt{\widetilde{\boldsymbol{\varphi}}(4:6)} \end{bmatrix}, \quad (4.39)$$

where \mathbf{P} is a diagonal matrix that has the chosen signs.

4.2.2 SDP Solution

The solution is obtained by transforming the constrained optimization in (4.26) to a semi-definite program [102] by the SDR. This solution is computationally demanding but it is robust against noise and requires less number of sensors. The expanded form

Table 4.2: Constraints of the SDP Solution for the Single-Time Case

φ and Φ	Φ
$\varphi(4) = \Phi(1, 1) + \Phi(2, 2) + \Phi(3, 3)$	$\Phi(8, 8) = \Phi(5, 6)$
$\varphi(11) = \Phi(1, 5) + \Phi(2, 8) + \Phi(3, 9)$	$\Phi(9, 9) = \Phi(5, 7)$
$\varphi(12) = \Phi(1, 8) + \Phi(2, 6) + \Phi(3, 10)$	$\Phi(10, 10) = \Phi(6, 7)$
$\varphi(13) = \Phi(1, 9) + \Phi(2, 10) + \Phi(3, 7)$	$\Phi(5, 10) = \Phi(8, 9)$
$\varphi(14) = \Phi(1, 11) + \Phi(2, 12) + \Phi(3, 13)$	$\Phi(5, 12) = \Phi(8, 11)$
	$\Phi(5, 13) = \Phi(9, 11)$
	$\Phi(5, 14) = \Phi(11, 11)$
	$\Phi(6, 9) = \Phi(8, 10)$
$\varphi(4) \geq 0$	$\Phi(6, 11) = \Phi(8, 12)$
$\varphi(5) \geq 0$	$\Phi(6, 13) = \Phi(10, 12)$
$\varphi(6) \geq 0$	$\Phi(6, 14) = \Phi(12, 12)$
$\varphi(7) \geq 0$	$\Phi(7, 8) = \Phi(9, 10)$
$\varphi(14) \geq 0$	$\Phi(7, 11) = \Phi(9, 13)$
	$\Phi(7, 12) = \Phi(10, 13)$
	$\Phi(7, 14) = \Phi(13, 13)$
	$\Phi(8, 13) = \Phi(9, 12)$
	$\Phi(8, 14) = \Phi(11, 12)$
	$\Phi(9, 12) = \Phi(10, 11)$
	$\Phi(9, 14) = \Phi(11, 13)$
	$\Phi(10, 14) = \Phi(12, 13)$

of the objective function (4.26a) is

$$\bar{J}(\Phi, \varphi) = \text{tr}(\mathbf{A}^T \mathbf{W} \mathbf{A} \Phi) - 2\mathbf{h}^T \mathbf{W} \mathbf{A} \varphi, \quad (4.40)$$

where $\mathbf{h}^T \mathbf{W} \mathbf{h}$ is dropped since it does not depend on the unknown φ . $\Phi = \varphi \varphi^T$ is a matrix of rank one. The SDP problem is formed by considering both Φ and φ are variables and relaxing the rank of $[[\Phi \ \varphi]^T, [\varphi^T \ 1]^T]$ for the minimization of (4.40).

The relaxed SDP problem is

$$\min_{\Phi, \varphi} \bar{J}(\Phi, \varphi), \quad (4.41a)$$

$$\text{s.t.} \quad \begin{bmatrix} \Phi & \varphi \\ \varphi^T & 1 \end{bmatrix} \succeq 0, \quad (4.41b)$$

All constraints listed in Table 4.2. (4.41c)

It is crucial to impose all the constraints in Table II that are derived from the necessary and extra relations from Table I when solving the SDP. The SDP in (4.41) has been relaxed by SDR and the extra optimization variable Φ has been added. The constraints enable the tightening of the relaxed SDP for reaching a solution that will be close to that of the original WLS optimization problem.

(4.41a) is a convex function and can be minimized efficiently using some optimization packages such as CVX [94]. The linear constraints in Table 4.2 are constructed according to the relations listed on the right of Table 4.1 and by exploiting the definition of Φ in which $\Phi(i, k) = \varphi(i)\varphi(k)$. The other five non-negative constraints are directly from the definition of φ^o . The object location estimate is

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}(1:3) \\ \mathbf{P}\sqrt{\boldsymbol{\varphi}(5:7)} \end{bmatrix}, \quad (4.42)$$

where \mathbf{P} is found by the procedure at the end of Section 4.2.1. \mathbf{W} in (4.40) requires the true object location. It is set to the identity matrix for solving an initial solution to approximate \mathbf{W} before obtaining the final solution by repeating the SDP.

4.3 Multiple-Time Observations

This section presents solutions to the scenario in which multiple snapshots are collected by each sensor during a period of time when the object motion is linear. As the object position changes over the period, more auxiliary variables will appear to

relate the initial object position \mathbf{u} at $k = 0$ with the other position \mathbf{u}_k . The algebraic and SDP solutions for this case will need a different development from the single-time case.

Inserting (2.1) that expresses \mathbf{u}_k^o in terms of \mathbf{u}^o and $\dot{\mathbf{u}}^o$ to (4.15) gives

$$\begin{aligned}
2cd_{k,i}r_{k,i}^2\varepsilon_{k,i}/f_o &= d_{k,i}^2\|\mathbf{s}_i\|^2 - 2d_{k,i}^2\mathbf{s}_i^T\mathbf{u}^o - 2kd_{k,i}^2\mathbf{s}_i^T\dot{\mathbf{u}}^o + d_{k,i}^2\|\mathbf{u}^o\|^2 + 2kd_{k,i}^2\mathbf{u}^{oT}\dot{\mathbf{u}}^o \\
&+ (k^2d_{k,i}^2\tilde{\mathbf{1}} - \mathbf{s}_iQ)^T\mathbf{v}_Q + 2\mathbf{s}_i^T\dot{\mathbf{u}}^o\dot{\mathbf{u}}^{oT}\mathbf{u}^o + 2k\mathbf{s}_i^T\dot{\mathbf{u}}^o\|\dot{\mathbf{u}}^o\|^2 - (\mathbf{u}^{oT}\dot{\mathbf{u}}^o)^2 - k^2\|\dot{\mathbf{u}}^o\|^4 - 2k\mathbf{u}^{oT}\dot{\mathbf{u}}^o\|\dot{\mathbf{u}}^o\|^2,
\end{aligned} \tag{4.43}$$

where $\tilde{\mathbf{1}} = [\mathbf{1}_3^T, \mathbf{0}_3^T]^T$. Although (4.43) remains highly non-linear with the unknowns \mathbf{u}^o and $\dot{\mathbf{u}}^o$, it is pseudo linear with respect to the unknowns and their product terms.

We shall set the unknown vector as

$$\begin{aligned}
\boldsymbol{\varphi}^o(1 : 20) &= [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}, \|\mathbf{u}^o\|^2, \mathbf{u}^{oT}\dot{\mathbf{u}}^o, \mathbf{v}_Q^T, \mathbf{u}^{oT}\dot{\mathbf{u}}^o\dot{\mathbf{u}}^{oT}, \|\dot{\mathbf{u}}^o\|^2\dot{\mathbf{u}}^{oT}]^T, \\
\boldsymbol{\varphi}^o(21 : 23) &= [\|\dot{\mathbf{u}}^o\|^4, \mathbf{u}^{oT}\dot{\mathbf{u}}^o\|\dot{\mathbf{u}}^o\|^2, (\mathbf{u}^{oT}\dot{\mathbf{u}}^o)^2]^T.
\end{aligned} \tag{4.44}$$

The independent object location variables are $\boldsymbol{\varphi}^o(1 : 6)$, and the other 17 elements $\boldsymbol{\varphi}^o(7 : 23)$ are auxiliary. Let the $M \times M$ diagonal matrix \mathbf{B}_k , the $M \times 23$ matrix \mathbf{A}_k and the $M \times 1$ vector \mathbf{h}_k be

$$\mathbf{B}_k = 2\frac{c}{f_o}\text{diag}\{[d_{k,1}r_{k,1}^2, r_{k,2}r_{k,2}^2, \dots, r_{k,M}r_{k,M}^2]\}, \tag{4.45}$$

$$\begin{aligned}
\mathbf{A}_k(i, 1 : 14) &= [2r_{k,i}^2\mathbf{s}_i^T, 2kr_{k,i}^2\mathbf{s}_i^T, -r_{k,i}^2, -2kr_{k,i}^2, - (k^2r_{k,i}^2\tilde{\mathbf{1}} - \mathbf{s}_iQ)^T], \\
\mathbf{A}_k(i, 15 : 23) &= [-2\mathbf{s}_i^T, -2k\mathbf{s}_i^T, k^2, 2k, 1],
\end{aligned} \tag{4.46}$$

$$\mathbf{h}_k = [r_{k,1}^2\|\mathbf{s}_1\|^2, r_{k,2}^2\|\mathbf{s}_2\|^2, \dots, r_{k,M}^2\|\mathbf{s}_M\|^2]^T. \tag{4.47}$$

Collecting \mathbf{A}_k , \mathbf{h}_k and \mathbf{B}_k for $k = 0, 1, \dots, N - 1$ give

$$\mathbf{A} = [\mathbf{A}_0^T, \mathbf{A}_1^T, \dots, \mathbf{A}_{N-1}^T]^T, \quad (4.48)$$

$$\mathbf{h} = [\mathbf{h}_0^T, \mathbf{h}_1^T, \dots, \mathbf{h}_{N-1}^T]^T, \quad (4.49)$$

$$\mathbf{B} = \text{diag}\{\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{N-1}\}. \quad (4.50)$$

In matrix form, (4.43) for $i = 1, \dots, M$ and $k = 0, \dots, N - 1$, after neglecting the second and higher error terms under small noise assumption, is

$$\mathbf{B}\boldsymbol{\varepsilon} = \mathbf{h} - \mathbf{A}\boldsymbol{\varphi}^o. \quad (4.51)$$

The constrained optimization problem for the multiple-time case is

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (4.52a)$$

$$\text{s.t. (C.1)}, \quad (4.52b)$$

where \mathbf{W} is given by (4.25). To fix the number of independent variables in (4.52) to only 6, we have imposed 17 constraints (C.1a)-(C.1q) listed in Appendix C, by using the relations among the auxiliary variables and the independent unknowns. They can be verified directly by substituting the elements of $\boldsymbol{\varphi}^o$ in (4.44). Interestingly enough, each constraint can be expressed in a quadratic form in terms of the elements of $\boldsymbol{\varphi}^o$. We next derive the algebraic and SDP solutions from (4.52).

4.3.1 Algebraic Solution

The problems (4.26) and (4.52) have similar form when ignoring the constraints. Pretending the elements of $\boldsymbol{\varphi}^o$ are independent, the WLS solution and its covariance matrix have expressions (4.27) and (4.28), with the matrices \mathbf{A} , \mathbf{B} and the vector \mathbf{h} defined differently in (4.45)-(4.50) and \mathbf{W} given by (4.25).

Similar to the procedure for single-time scenario, the solution of $\boldsymbol{\varphi}$ from (4.27) is nonlinearly transformed to obtain the final object location $\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T$ by making use of the relations among the elements of $\boldsymbol{\varphi}$ described in Appendix C. Let the data vector $\tilde{\mathbf{h}}$ of the second WLS be

$$\tilde{\mathbf{h}} = [\boldsymbol{\varphi}^T(1 : 7), 2\varphi(8), \boldsymbol{\varphi}^T(9 : 11), 2\boldsymbol{\varphi}^T(12 : 17), \boldsymbol{\varphi}^T(18 : 21), 2\varphi(22), \varphi(23)]^T, \quad (4.53)$$

We can write $\tilde{\mathbf{h}}$ in terms of $\Delta\boldsymbol{\varphi}$ and $\boldsymbol{\varphi}^o(1 : 6)$ or equivalently $\boldsymbol{\theta}^o$ by using similar manipulation for obtaining (B.1). Neglecting the second and higher order error terms, we reach

$$\tilde{\mathbf{h}} = \tilde{\mathbf{A}}\boldsymbol{\theta}^o + \tilde{\mathbf{B}}\Delta\boldsymbol{\varphi}, \quad (4.54)$$

$$\begin{aligned}
\mathbf{C}(7, 1 : 3) &= \mathbf{C}(8, 4 : 6) = \mathbf{C}(22, 18 : 20) = \mathbf{C}(23, 15 : 17) = \boldsymbol{\varphi}^{oT}(1 : 3) , \\
\mathbf{C}(8, 1 : 3) &= \mathbf{C}(21, 18 : 20) = \mathbf{C}(22, 15 : 17) = \boldsymbol{\varphi}^{oT}(4 : 6) , \\
\mathbf{C}(9 : 11, 4 : 6) &= \mathbf{C}(18 : 20, 9 : 11) = \text{diag}\{\boldsymbol{\varphi}^o(4 : 6)\} , \\
\mathbf{C}(12 : 14, 4 : 6) &= [\mathbf{C}(18 : 20, 12 : 14)]^T , \\
\mathbf{C}(15 : 17, 9 : 11) &= \text{diag}\{\boldsymbol{\varphi}^o(1 : 3)\} , \mathbf{C}(15 : 17, 8) = \boldsymbol{\varphi}^o(4 : 6) , \\
\mathbf{C}(15 : 17, 12 : 14) &= [\mathbf{C}(18 : 20, 12 : 14)]^T \\
&= \begin{bmatrix} \varphi^o(2) & \varphi^o(3) & 0 \\ \varphi^o(1) & 0 & \varphi^o(3) \\ 0 & \varphi^o(1) & \varphi^o(2) \end{bmatrix} , \quad = \begin{bmatrix} \varphi^o(5) & \varphi^o(4) & 0 \\ \varphi^o(6) & 0 & \varphi^o(4) \\ 0 & \varphi^o(6) & \varphi^o(5) \end{bmatrix} ,
\end{aligned} \tag{4.57}$$

and the other elements of \mathbf{C} are zero.

The WLS solution to (4.54) after setting the weighting matrix to (4.37) with $\tilde{\mathbf{B}}$ given by (4.56)-(4.57) is

$$\boldsymbol{\theta} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{h}} , \tag{4.58}$$

It has a covariance matrix approximately equal to

$$\text{cov}(\boldsymbol{\theta}) \simeq (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} . \tag{4.59}$$

4.3.2 SDP Solution

Expanding the objective function J in (4.52) gives (4.40) after dropping the term $\mathbf{h}^T \mathbf{W} \mathbf{h}$ that is independent of the unknown vector $\boldsymbol{\varphi}$. Using the same methodology

as in the single-time case, the relaxed SDP is

$$\min_{\Phi, \varphi} \bar{J}(\Phi, \varphi), \quad (4.60a)$$

$$\text{s.t.} \quad \begin{bmatrix} \Phi & \varphi \\ \varphi^T & 1 \end{bmatrix} \succeq 0, \quad (4.60b)$$

$$\text{All constraints listed in Table 4.3.} \quad (4.60c)$$

The constraint (4.60b) relaxes the rank of $\Phi = \varphi\varphi^T$. The other constraints listed in Table 4.3 are directly deduced from the relations among the elements of φ and also from the relation between φ and Φ . The object location estimate is extracted from the first 6 elements of the solution φ after solving the SDP,

$$\theta = \varphi(1 : 6). \quad (4.61)$$

Table 4.3 comes with considerable amount of optimization effort. The first and second order relations are over several hundreds. We have carefully investigated each and eliminated those that are not doing much to improve the tightness of SDP. The relations in Table III are those that will make a difference to improve the performance when included as constraints for solving (4.60). It is a big distinction with the 2-D case SDP solution [107] in which the number of relations are not as many and optimization is not needed to reduce the number of constraints.

Table 4.3: Constraints on φ and Φ for the SDP solution of the multiple-time Case

φ and Φ	$\varphi(9) = \Phi(4, 4)$ $\varphi(10) = \Phi(5, 5)$	$\varphi(11) = \Phi(6, 6)$ $\varphi(12) = \Phi(4, 5)$	$\varphi(13) = \Phi(4, 6)$ $\varphi(14) = \Phi(5, 6)$	$\varphi(15) = \Phi(4, 8)$ $\varphi(16) = \Phi(5, 8)$	$\varphi(17) = \Phi(6, 8)$ $\varphi(23) = \Phi(8, 8)$	
	$\varphi(7) = \Phi(1, 1)$ $+ \Phi(2, 2) + \Phi(3, 3)$ $\varphi(8) = \Phi(1, 4)$ $+ \Phi(2, 5) + \Phi(3, 6)$ $\varphi(15) = \Phi(1, 9)$ $+ \Phi(2, 12) + \Phi(3, 13)$	$\varphi(16) = \Phi(1, 12)$ $+ \Phi(2, 10) + \Phi(3, 14)$ $\varphi(17) = \Phi(1, 13)$ $+ \Phi(2, 14) + \Phi(3, 11)$ $\varphi(18) = \Phi(4, 9)$ $+ \Phi(5, 12) + \Phi(6, 13)$	$\varphi(19) = \Phi(4, 12)$ $+ \Phi(5, 10) + \Phi(5, 11)$ $\varphi(20) = \Phi(4, 13)$ $+ \Phi(5, 14) + \Phi(6, 11)$ $\varphi(21) = \Phi(4, 18)$ $+ \Phi(5, 19) + \Phi(6, 20)$	$\varphi(22) = \Phi(1, 18)$ $+ \Phi(2, 19) + \Phi(3, 20)$ $\varphi(22) = \Phi(4, 15)$ $+ \Phi(5, 16) + \Phi(6, 17)$ $\varphi(23) = \Phi(1, 15)$ $+ \Phi(2, 16) + \Phi(3, 17)$		
Φ	$\Phi(4, 10) = \Phi(5, 12)$ $\Phi(4, 11) = \Phi(6, 13)$ $\Phi(4, 12) = \Phi(5, 9)$ $\Phi(4, 13) = \Phi(6, 9)$ $\Phi(4, 14) = \Phi(5, 13)$ $\Phi(4, 15) = \Phi(8, 9)$ $\Phi(4, 16) = \Phi(5, 15)$ $\Phi(4, 17) = \Phi(6, 15)$ $\Phi(4, 19) = \Phi(5, 18)$ $\Phi(4, 20) = \Phi(6, 18)$ $\Phi(4, 22) = \Phi(8, 18)$ $\Phi(4, 23) = \Phi(8, 15)$ $\Phi(5, 11) = \Phi(6, 14)$ $\Phi(5, 13) = \Phi(6, 12)$ $\Phi(5, 14) = \Phi(6, 10)$ $\Phi(5, 15) = \Phi(8, 12)$	$\Phi(5, 16) = \Phi(8, 10)$ $\Phi(5, 17) = \Phi(6, 16)$ $\Phi(5, 20) = \Phi(6, 19)$ $\Phi(5, 22) = \Phi(8, 19)$ $\Phi(5, 23) = \Phi(8, 16)$ $\Phi(6, 15) = \Phi(8, 13)$ $\Phi(6, 16) = \Phi(8, 14)$ $\Phi(6, 17) = \Phi(8, 11)$ $\Phi(6, 22) = \Phi(8, 20)$ $\Phi(6, 23) = \Phi(8, 17)$ $\Phi(9, 10) = \Phi(12, 12)$ $\Phi(9, 11) = \Phi(13, 13)$ $\Phi(9, 14) = \Phi(12, 13)$ $\Phi(9, 16) = \Phi(12, 15)$ $\Phi(9, 17) = \Phi(13, 15)$ $\Phi(9, 19) = \Phi(12, 18)$	$\Phi(9, 20) = \Phi(13, 18)$ $\Phi(9, 21) = \Phi(18, 18)$ $\Phi(9, 22) = \Phi(15, 18)$ $\Phi(9, 23) = \Phi(15, 15)$ $\Phi(10, 11) = \Phi(14, 14)$ $\Phi(10, 13) = \Phi(12, 14)$ $\Phi(10, 15) = \Phi(12, 16)$ $\Phi(10, 17) = \Phi(14, 16)$ $\Phi(10, 18) = \Phi(12, 19)$ $\Phi(10, 20) = \Phi(14, 19)$ $\Phi(10, 21) = \Phi(19, 19)$ $\Phi(10, 22) = \Phi(16, 19)$ $\Phi(10, 23) = \Phi(16, 16)$ $\Phi(11, 12) = \Phi(13, 14)$ $\Phi(11, 15) = \Phi(13, 17)$ $\Phi(11, 16) = \Phi(14, 17)$	$\Phi(11, 18) = \Phi(13, 20)$ $\Phi(11, 19) = \Phi(14, 20)$ $\Phi(11, 21) = \Phi(20, 20)$ $\Phi(11, 22) = \Phi(17, 20)$ $\Phi(11, 23) = \Phi(17, 17)$ $\Phi(12, 17) = \Phi(13, 16)$ $\Phi(12, 20) = \Phi(13, 19)$ $\Phi(12, 21) = \Phi(18, 19)$ $\Phi(12, 22) = \Phi(15, 19)$ $\Phi(12, 23) = \Phi(15, 16)$ $\Phi(13, 16) = \Phi(14, 15)$ $\Phi(13, 19) = \Phi(14, 18)$ $\Phi(13, 21) = \Phi(18, 20)$ $\Phi(13, 22) = \Phi(15, 20)$ $\Phi(13, 23) = \Phi(15, 17)$ $\Phi(14, 21) = \Phi(19, 20)$	$\Phi(14, 22) = \Phi(16, 20)$ $\Phi(14, 23) = \Phi(16, 17)$ $\Phi(15, 19) = \Phi(16, 18)$ $\Phi(15, 20) = \Phi(17, 18)$ $\Phi(15, 21) = \Phi(18, 22)$ $\Phi(15, 22) = \Phi(18, 23)$ $\Phi(16, 20) = \Phi(17, 19)$ $\Phi(16, 21) = \Phi(19, 22)$ $\Phi(16, 22) = \Phi(19, 23)$ $\Phi(17, 21) = \Phi(20, 22)$ $\Phi(17, 22) = \Phi(20, 23)$ $\Phi(21, 23) = \Phi(22, 22)$	

4.4 carrier frequency unavailable solution

The proposed method estimates the unknowns $\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T$ and f_o^o together by taking advantage of the solution when the carrier frequency is available.

The measurement model when f_o^o is an unknown is (4.16). The error term has Gaussian distribution since it is a linear combination of Gaussian random variables. Let $\hat{\mathbf{f}}(f_o, \boldsymbol{\theta})$ be the parametric form of the frequency vector in f_o and $\boldsymbol{\theta}$, having \mathbf{s}_i^o replaced by \mathbf{s}_i . From (2.2), the elements of $\hat{\mathbf{f}}(f_o, \boldsymbol{\theta})$ are

$$\hat{f}_{k,i}(f_o, \boldsymbol{\theta}) = f_o - \frac{f_o(\boldsymbol{\theta}(1:3) + k\boldsymbol{\theta}(4:6) - \mathbf{s}_i)^T \boldsymbol{\theta}(4:6)}{c \|\boldsymbol{\theta}(1:3) + k\boldsymbol{\theta}(4:6) - \mathbf{s}_i\|}. \quad (4.62)$$

The approximated maximum likelihood (ML) cost function is

$$\bar{J}(f_o, \boldsymbol{\theta}) = \left(\mathbf{f} - \hat{\mathbf{f}}(f_o, \boldsymbol{\theta}) \right)^T \mathbf{Q}_\varepsilon^{-1} \left(\mathbf{f} - \hat{\mathbf{f}}(f_o, \boldsymbol{\theta}) \right), \quad (4.63)$$

where \mathbf{Q}_ε is defined in (4.18). (4.63) is nonlinear with respect to the unknowns. Obtaining the solution by minimizing (4.63) directly is not trivial. The gradient-based iterative solution requires a good initial guess, in which pseudo linearization can be a possible approach to obtain an initial guess [108, 109] to start the iteration. The grid search solution will be very time consuming since there are 7 unknowns. We shall resort to a hybrid solution that uses a search over f_o^o and the proposed solution for the available carrier frequency case.

For a given f_o , $\boldsymbol{\theta}$ is found by the algebraic or SDP solution derived in Section 4.2 for single-time case or Section 4.3 for multiple-time scenario, where $\sigma_{f_o}^2$ is set to zero in (4.9) to obtain \mathbf{Q}_ε for the weighting matrix \mathbf{W} . Inserting this f_o and the resulting $\boldsymbol{\theta}$ solution to (4.63) gives the cost for this f_o . We repeat this process for several f_o trial values and identify the one that has the minimum cost. The corresponding $\boldsymbol{\theta}$ will be the location estimate of the object.

The range of possible f_o values is often known in practice [79]. We can improve the performance and efficiency by searching f_o with a coarse grid size, followed by the Newton-Raphson iteration to refine the value. Interpreting (4.63) as a function of single unknown f_o , that is

$$\bar{J}(f_o) = \left(\mathbf{f} - \hat{\mathbf{f}}(f_o) \right)^T \mathbf{Q}_\varepsilon^{-1} \left(\mathbf{f} - \hat{\mathbf{f}}(f_o) \right), \quad (4.64)$$

the Newton-Raphson iteration is [104]

$$f_o^{(l+1)} = f_o^{(l)} - \xi (\nabla^2 \bar{J}(f_o^{(l)}))^{-1} \nabla \bar{J}(f_o^{(l)}). \quad (4.65)$$

$l = 0, 1, \dots, l_{max} - 1$ represents the iteration index with l_{max} set to achieve the required accuracy level, $f_c^{(0)}$ is the estimate from the coarse grid search and ξ is the step-size constant between 0 and 1 to stabilize the iterative process. $\nabla \bar{J}(\bullet)$ and $\nabla^2 \bar{J}(\bullet)$ are the first and second derivatives of $\bar{J}(\bullet)$ with respect to f_o and they can be obtained using the finite-difference approximations

$$\nabla \bar{J}(f_o^{(l)}) \simeq \frac{\bar{J}(f_o^{(l)} + \delta) - \bar{J}(f_o^{(l)})}{\delta}, \quad (4.66)$$

$$\nabla^2 \bar{J}(f_o^{(l)}) \simeq \frac{\nabla \bar{J}(f_o^{(l)} + \delta) - \nabla \bar{J}(f_o^{(l)})}{\delta}, \quad (4.67)$$

where δ is a small constant.

4.5 Sequential Multiple-Time Solution

The CFS and SDP methods presented in Section 4.3 for multiple-time scenario are suitable for batch processing where N observations are collected before estimation can take place. It would be desirable to have a sequential solution for the multiple-time scenario that provides the timely improved object motion parameters upon a new observation arrives. It could also reduce complexity without redoing the estimation entirely when we have a new snapshot observation.

The proposed multiple-time sequential algorithm first uses the single-time case

solution, either CFS or SDP, to obtain the object location parameters for the newly arrived observation. It next combines this single snapshot solution with the estimate from all the past observations.

Let $\boldsymbol{\theta}_k$ be the single-time case solution from the newly received k -th observation and $\boldsymbol{\theta}_{\{k-1\}}$ be the estimate from all the past observations up to time $k-1$. They are related to the true value of the updated estimate $\boldsymbol{\theta}_{\{k\}}^o$ by, after using (2.1),

$$\begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\theta}_{\{k-1\}} \end{bmatrix} = \mathbf{H}_{\{k\}} \boldsymbol{\theta}_{\{k\}}^o + \begin{bmatrix} \Delta\boldsymbol{\theta}_k \\ \Delta\boldsymbol{\theta}_{\{k-1\}} \end{bmatrix}, \quad (4.68a)$$

$$\mathbf{H}_{\{k\}} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{I}_6 \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} \mathbf{I}_3 & k\mathbf{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}. \quad (4.68b)$$

The WLS solution by minimizing the weighted L_2 -norm of $[\Delta\boldsymbol{\theta}_k^T, \Delta\boldsymbol{\theta}_{\{k-1\}}^T]^T$ is

$$\boldsymbol{\theta}_{\{k\}} = (\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}} \mathbf{H}_{\{k\}})^{-1} \mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}} [\boldsymbol{\theta}_k^T, \boldsymbol{\theta}_{\{k-1\}}^T]^T, \quad (4.69a)$$

$$\mathbf{W}_{\{k\}} = \text{diag} \left\{ \text{cov}(\boldsymbol{\theta}_k)^{-1} \Big|_{\mathbf{H}_k \boldsymbol{\theta}_{\{k-1\}}}, \text{cov}(\boldsymbol{\theta}_{\{k-1\}})^{-1} \Big|_{\boldsymbol{\theta}_{\{k-1\}}} \right\}. \quad (4.69b)$$

In $\mathbf{W}_{\{k\}}$, $\text{cov}(\boldsymbol{\theta}_k)$ is given by (4.59) with $\boldsymbol{\theta}$ replaced with $\boldsymbol{\theta}_k$ and $\text{cov}(\boldsymbol{\theta}_{\{k-1\}})$ is given in (4.70) with $k-1$ instead of k . $\text{cov}(\bullet)^{-1}|_*$ stands for obtaining $\text{cov}(\bullet)^{-1}$ with \bullet replaced by $*$. The accuracy of $\boldsymbol{\theta}_{\{k\}}$, $k = 1, 2, \dots$, from the WLS optimization is

$$\text{cov}(\boldsymbol{\theta}_{\{k\}}) \simeq (\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}} \mathbf{H}_{\{k\}})^{-1}, \quad (4.70)$$

where the approximation comes from the randomness of $\boldsymbol{\theta}_{\{k\}}$ in $\mathbf{W}_{\{k\}}$.

In summary, the sequential algorithm operates as follows. (i) Set $\boldsymbol{\theta}_{\{0\}} = \boldsymbol{\theta}_0$ at the

start; (ii) Apply the single-time algorithm, either CFS or SDP, to obtain $\boldsymbol{\theta}_k$ using the latest received snapshot observation \mathbf{f}_k ; (ii) Use (4.69) to obtain $\boldsymbol{\theta}_{\{k\}}$ from $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{\{k-1\}}$.

The multiple-time sequential algorithm is much more efficient in computation than the batch multiple-time solution presented in Section 4.3. The computation complexity in the sequential estimation is dominated by the single-time case estimate that involves 14 unknown variables, whereas the batch processing algorithm, regardless of CFS or SDP, has 23 unknown variables. Having said, the sequential algorithm may deviate from the CRLB earlier than the batch algorithm in suffering from the thresholding effect, which is dependent on the latest single snapshot observation only instead of $k + 1$ snapshot measurements.

4.6 Analysis

We shall prove that under first order analysis in which the second and higher order noise terms are negligible, the covariance matrix of the proposed algebraic solution can reach the CRLB accuracy when some conditions are met. The CRLB has been derived in Section 2.6 which has a generic expression applicable to the 3-D case as well. We only present the analysis for the solution of the multiple-time case. The same analysis applies to single-time case and we can reach the same conclusion. The analysis begins with the batch algorithm for the known carrier frequency case first and the unavailable carrier frequency scenario next. It then continues for the study of the sequential algorithm.

4.6.1 Carrier Frequency Available

Let us start from (4.59). Inserting (4.25) and (4.37) and taking inverse give

$$\text{cov}(\boldsymbol{\theta})^{-1} = \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_\varepsilon^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \quad (4.71)$$

(4.9) can be expressed as $\mathbf{Q}_\varepsilon = \mathbf{Q}_n + \mathbf{D} \mathbf{Q}_\alpha \mathbf{D}^T$, where $\mathbf{D} = [\mathbf{d}_f, \mathbf{D}_s]$ and $\mathbf{Q}_\alpha = \text{diag}\{\sigma_{f_o}^2, \mathbf{Q}_s\}$. Using the matrix inversion lemma [104], the inverse of \mathbf{Q}_ε is

$$\mathbf{Q}_\varepsilon^{-1} = \mathbf{Q}_n^{-1} - \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1}. \quad (4.72)$$

Inserting it to (4.71) gives

$$\begin{aligned} \text{cov}(\boldsymbol{\theta})^{-1} = & \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}} - \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} \\ & + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \end{aligned} \quad (4.73)$$

\mathbf{B}^{-1} and $\tilde{\mathbf{B}}^{-1}$ can be evaluated analytically since \mathbf{B} in (4.50) is diagonal and $\tilde{\mathbf{B}}$ in (4.56) is sparse.

We list below several small noise conditions such that the approximations of \mathbf{A} , \mathbf{B} , $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ by their true values are valid. For $k = 0, 1, \dots, N-1$ and $i = 1, 2, \dots, M$, they are:

$$\frac{n_{k,i}}{f_{k,i}^o - f_o^o} \simeq 0, \quad (4.74a)$$

$$\frac{\Delta f_o}{f_{k,i}^o - f_o^o} \simeq 0, \quad (4.74b)$$

$$\Delta x_i \simeq 0, \Delta y_i \simeq 0, \Delta z_i \simeq 0, \quad (4.74c)$$

$$\frac{\Delta x_i}{x_i^o} \simeq 0, \quad \frac{\Delta y_i}{y_i^o} \simeq 0, \quad \frac{\Delta z_i}{z_i^o} \simeq 0, \quad (4.74d)$$

$$\frac{\Delta x_i}{r_{k,i}^o} \simeq 0, \quad \frac{\Delta y_i}{r_{k,i}^o} \simeq 0, \quad \frac{\Delta z_i}{r_{k,i}^o} \simeq 0, \quad (4.74e)$$

$$\frac{\Delta \varphi(j)}{\varphi^o(j)} \simeq 0 \text{ or } \Delta \varphi(j) \simeq 0, \quad j = 1, 2, \dots, 6, 8, 9, \dots, 20, \quad (4.74f)$$

$$\frac{\Delta \varphi(j)}{r_{k,i}^o} \simeq 0, \quad j = 1, 2, \dots, 6. \quad (4.74g)$$

The conditions are essentially requiring the noise of a certain measured or estimated quantity is small compared to the true value. Such a noise requirement is expected, otherwise the localization will not yield meaningful results.

When (4.74) is fulfilled, direct evaluation gives

$$\mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}} \simeq \mathbf{B}^{o-1} \mathbf{A}^o \tilde{\mathbf{B}}^{o-1} \tilde{\mathbf{A}}^o = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (4.75a)$$

$$\mathbf{D} \simeq \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\varphi}^{oT}}. \quad (4.75b)$$

Using (4.75) in (4.73), we reach the conclusion that

$$\text{cov}(\boldsymbol{\theta}) \simeq \text{CRLB}(\boldsymbol{\theta}). \quad (4.76)$$

4.6.2 Carrier Frequency Unavailable

We shall first show that minimizing the cost function (4.63) can give a solution that can reach the CRLB performance under the conditions in (4.74). We then prove that the proposed method can efficiently finds the minimizer of (4.63).

The expansion of $\hat{f}_{k,i}(f_o, \boldsymbol{\theta})$ around the true values up to the first order error is

$$\hat{f}_{k,i}(f_o, \boldsymbol{\theta}) = \hat{f}_{k,i}^o + \left. \frac{\partial \hat{f}_{k,i}}{\partial f_o} \right|_{f_o^o, \boldsymbol{\theta}^o} (f_o - f_o^o) + \left. \frac{\partial \hat{f}_{k,i}}{\partial \boldsymbol{\theta}^T} \right|_{f_o^o, \boldsymbol{\theta}^o} (\boldsymbol{\theta} - \boldsymbol{\theta}^o), \quad (4.77)$$

where $\hat{f}_{k,i}^o = \hat{f}_{k,i}(f_o^o, \boldsymbol{\theta}^o)$. Using (4.77) in (4.63) and setting the gradients with respect to f_o and $\boldsymbol{\theta}$ to zero yield

$$\begin{bmatrix} f_o \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} f_o^o \\ \boldsymbol{\theta}^o \end{bmatrix} + \left(\begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o} & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\varepsilon^{-1} \begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o} & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix} \right)^{-1} \begin{bmatrix} \frac{\partial \hat{\mathbf{f}}^o}{\partial f_o^o} & \frac{\partial \hat{\mathbf{f}}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\varepsilon^{-1} (\mathbf{f} - \hat{\mathbf{f}}^o), \quad (4.78)$$

where $\hat{\mathbf{f}}^o$ is the vector with elements $\hat{f}_{k,i}(f_o, \boldsymbol{\theta})$, $k = 0, 1, \dots, N-1$ and $i = 1, 2, \dots, M$. (4.78) is the expected solution of the proposed method. It is reasonable to approximate $\mathbf{f} - \hat{\mathbf{f}}^o$, $\partial \hat{\mathbf{f}}^o / \partial f_o^o$ and $\partial \hat{\mathbf{f}}^o / \partial \boldsymbol{\theta}^o$ by $\boldsymbol{\varepsilon}$, $\partial \mathbf{f}^o / \partial f_o^o$ and $\partial \mathbf{f}^o / \partial \boldsymbol{\theta}^o$ under the small noise conditions in (4.74a)-(4.74e). Subtracting the true solution from both sides of (4.78), taking outer product and applying expectation give

$$\text{cov}(f_o, \boldsymbol{\theta}) = \left(\begin{bmatrix} \frac{\partial \mathbf{f}^o}{\partial f_o^o} & \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix}^T \mathbf{Q}_\varepsilon^{-1} \begin{bmatrix} \frac{\partial \mathbf{f}^o}{\partial f_o^o} & \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} \end{bmatrix} \right)^{-1}. \quad (4.79)$$

When we apply the block matrix inversion formula [104] to (4.79), keep only the lower right block, insert (4.18) and simplify, it becomes the expression of the CRLB. Thus, under the small noise conditions (4.74a)-(4.74e), the solution that minimizes (4.63) will achieve the CRLB accuracy.

The procedure in the proposed algorithm of using a one variable search over f_o together with the frequency available solution will eventually reach the smallest value of (4.64). This process is basically finding the global minimizer of (4.63). We can

therefore conclude that the proposed method gives a solution that reaches the CRLB accuracy, under the small noise conditions in (4.74).

4.6.3 Sequential Algorithm

We shall analyze the theoretical performance of the sequential algorithm and show that it can achieve close to the CRLB accuracy, i.e.

$$\text{cov}(\boldsymbol{\theta}_{\{k\}}) \simeq \text{CRLB}(\boldsymbol{\theta}_{\{k\}}^o), \quad (4.80)$$

providing with the small noise conditions in (4.74), through the mathematical induction.

It is clear from the start of the sequential algorithm that

$$\text{cov}(\boldsymbol{\theta}_{\{0\}}) \simeq \text{CRLB}_{\{0\}}(\boldsymbol{\theta}^o), \quad (4.81)$$

by the analysis in Sections 4.6.1 or 4.6.2. Assuming (4.80) is true up to time $k - 1$ in which $\text{cov}(\boldsymbol{\theta}_{\{k-1\}}) \simeq \text{CRLB}(\boldsymbol{\theta}_{\{k-1\}}^o)$, we next show that (4.80) is true at time k under (4.74).

In (4.69b), it is reasonable that $\boldsymbol{\theta}_{\{k-1\}}$ closely approximates $\boldsymbol{\theta}^o$ given (4.74). The inverse of the Fisher Information Matrix (FIM) is the CRLB [104]. Using the analysis results in Sections 4.6.1 or 4.6.2, putting in (4.68b) and (4.69b) and multiplying out, we have

$$\text{cov}(\boldsymbol{\theta}_{\{k\}}) \simeq \left(\text{FIM}(\boldsymbol{\theta}_{\{k-1\}}^o) + \mathbf{H}_k^T \text{FIM}(\boldsymbol{\theta}_k^o) \mathbf{H}_k \right)^{-1}. \quad (4.82)$$

According to the Gaussian data model and the uncorrelated behavior of the ob-

servations at different time instants, it is straightforward to validate that

$$\text{CRLB}(\boldsymbol{\theta}_{\{k\}}^o) = \left(\text{FIM}(\boldsymbol{\theta}_{\{k-1\}}^o) + \frac{\partial \boldsymbol{\theta}_k^{oT}}{\partial \boldsymbol{\theta}_{\{k\}}^o} \text{FIM}(\boldsymbol{\theta}_k^o) \frac{\partial \boldsymbol{\theta}_k^o}{\partial \boldsymbol{\theta}_{\{k\}}^{oT}} \right)^{-1}. \quad (4.83)$$

We have from (2.1)

$$\frac{\partial \boldsymbol{\theta}_k^o}{\partial \boldsymbol{\theta}_{\{k\}}^{oT}} = \mathbf{H}_k, \quad (4.84)$$

in which $\boldsymbol{\theta}_{\{k\}}^o = \boldsymbol{\theta}^o$ has been used. Using (4.84) in (4.83) yields immediately (4.80). We therefore conclude that the sequential solution reaches closely the CRLB performance under the small noise conditions listed in (4.74).

4.6.4 Complexity

The CFS is algebraic and it is more computationally efficient than the SDP solution. We shall perform complexity analysis of the two solutions for the available carrier frequency case. The unavailable carrier frequency case is basically repeating the processing of the available carrier frequency case several times.

Going through the steps of obtaining the CFS, Appendix D shows that the complexity of CFS is given by (D.1). The complexity of the SDP solution involves forming the objective function (4.40) that takes the amount of (D.2) and solving an SDP that has the worst-case complexity given by (D.3) [110]. Simplifying (D.1) and (D.2)–(D.3) further by keeping the dominant components, Table 4.4 summarizes the complexity of the solutions we proposed, in terms of the number of multiplications.

Regardless of the specific solution for single-time, multiple-time(batch) or multiple-time(sequential), the complexity has the common expressions in Table 4.4 but taking different values of the parameters. The parameter values for the three specific cases

Table 4.4: Complexity of the Algebraic Solution CFS and the SDP Solution in terms of the number of multiplications.

Sol.	Order of Complexity
CFS	$O((N^3 + 3N^2 + 18N)M^3 + (2LN^2)M^2 + (2L^2N)M + 4\eta L^3 + X)$
SDP	$O((N^3 + 3N^2 + 18N)M^3 + (2LN^2)M^2 + (2L^2N)M + 2\sqrt{L}(C^3 + C^2L^2 + CL^3)\ln(1/\epsilon) + X)$
	η : (number of repetitions + 1) in the second stage of CFS M : number of sensors N : number of consecutive time measurements L : length of the pseudo unknown vector φ C : number of constraints X : additional computation from sequential algorithm

Table 4.5: Parameter values for Table IV.

Cases	N	L	C	X
CFS: Single-Time	1	14		0
CFS(batch): Multiple-Time	N	23		0
CFS(sequential): Multiple-Time	1	14		$4L^3$
SDP: Single-Time	1	14	30	0
SDP(batch): Multiple-Time	N	23	98	0
SDP(sequential): Multiple-Time	1	14	30	$4L^3$

are listed in Table 4.5.

Roughly speaking, the complexity difference of the two algorithms amounts to the use of the second stage processing in the CFS and an SDP solver in the SDP solution. The former has a complexity on the order of L^3 and the latter $\sqrt{L}C^3 \ln(1/\epsilon)$, where $L < C$ from Table 4.5 and $\epsilon < 1$. It becomes clear that CFS has lower computational complexity.

4.7 Simulations

The performance of the proposed solutions is compared with the CRLB and other solutions from the literature for shallow water acoustic application where the signal propagation speed c is about 1500 m/s [1, 81, 111]. We use shallow water localization for illustration in this section and the proposed algorithm can be applied to other environments and applications for localization such as by a radar network or an airborne platform. The scenario assumes the object is moving within a 3-D cuboid space of dimensions $3 \times 3 \times 1$ km³ centered 500 m below the sea surface at $(0, 0, 500)$ m at a constant speed $\|\dot{\mathbf{u}}^o\| = 10$ m/s and is radiating a tonal signal at frequency $f_o^o = 15$ kHz [1, 81]. The fixed sensors are placed at two different regions with equal size of $2 \times 2 \times 0.1$ km³ centered at $(0, 0, 50)$ m and $(0, 0, 950)$ m. The power of the frequency measurement noise is denoted by σ^2 . It is inversely proportional to the SNR and to third power of the data length [104]. Having higher SNR or longer acquisition time can reduce frequency measurement error σ .

We shall use 10 randomly generated configurations, where the Cartesian coordinates of the object initial position, velocity direction and sensor locations in each are sampled from a uniform distribution within the limits. The sensor positions satisfy $\|\mathbf{s}_i - \mathbf{s}_j\| \geq 400$ m for $i, j = 1, 2, \dots, M$ and $i < j$ to avoid poor geometry and support the conditions in (4.74). For the unavailable carrier frequency case, we used 150 grid points for the coarse search process, δ for the Newton-Raphson iterations is set to 0.01 at $l = 0$ and then changes to $f_o^{(l+1)} - f_o^{(l)}$ for $l = 1, 2, \dots, l_{max} - 2$, and the step-size ξ is 0.25. Algorithm performance is measured using the mean square error (MSE) of the object location estimate computed by averaging the squared error over 10 randomly generated configurations having 1000 ensemble runs in each except the

tests that include the Shames method [88] or the grid search process where we used only 100 ensembles to reduce the processing time.

The proposed closed-form and SDP solutions are denoted by CFS and SDP. In addition, the CFS is used to initialize the MLE by the Gauss-Newton iterative implementation and the resulting solution is denoted by CFS-ML. The Newton-Raphson iterations in the unknown frequency case is dropped from the CFS-ML solution as the refinement process is achieved by the MLE. The SDP solution is obtained using the CVX toolbox [94] where three scaling adjustments are used to reduce the round off errors and achieve better accuracy. The sensor position vector \mathbf{s} in (2.7) is scaled down by the maximum distance between any two sensors, the range rate vector \mathbf{d} in (4.6) is scaled down by its maximum element.

For comparison, we include the solutions from [85] and [88] indicated by Chan and Shames, respectively, after some modifications to match the 3-D scenario. We change the model of Chan method to find the solution for the available frequency case as the original model assumed it is unavailable. We also change the step-size of the grid search so that its computational time is comparable to the SDP solution.

4.7.1 Performance w.r.t. Measurement Noise

For this test, the covariance matrices \mathbf{Q}_k , $k = 0, 1, \dots, N - 1$ of the measurements are set to $\sigma^2 \mathbf{I}_M$ where σ is in Hz. The sensor positions covariance matrix \mathbf{Q}_s is $\mathbf{0}$.

Single-Time

The number of sensors M is 16. Fig. 4.2 shows the performance as the noise level σ increases when f_o is exactly known. When σ is within the small to moderate level,

the MSEs of CFS and SDP attain the CRLB accuracy for both position and velocity estimation. As σ increases to a large level, the SDP shows steady performance while the CFS leaves the CRLB. The CFS-ML improves the result of CFS by extending its optimal accuracy to slightly larger noise level. Regardless of the measurement noise level, Shames could not reach the CRLB performance. Chan shows poor estimation accuracy due to the limited grid resolution when maintaining its computational complexity close to that of SDP.

Table 4.6 examines the relative average processing times of the simulation in Fig. 4.2 obtained from Matlab implementation at two different levels of σ . CFS-ML is 3 times slower than CFS and both of them show great benefit over the other methods by consuming much less processing time. Shames has a very long processing time, while SDP and Chan have moderate computational complexity. It is possible that the complexity of SDP can be reduced when the code for the SDP problem is specially designed, the one used in this simulation is from a universal solver that may have unnecessary procedures added to the original problem. The measurement noise level does not affect much the relative average processing times of these methods.

Fig. 4.3 illustrates the performance when the carrier frequency f_o is unavailable, where the other settings remain the same as for Fig. 4.2. The absolute performance drops from that in Fig. 4.2 as the number of unknowns for this case is increased by one. CFS-ML appear to handle larger σ level better, and it can extend the CRLB performance of CFS significantly by 20 dB.

Table 4.6: Processing Times at different values of σ relative to that of CFS at $\sigma = 0.01$ Hz.

σ (Hz)	CFS	CFS+ML	SDP	Shames	Chan
0.01	1.00	2.89	993.47	113312.40	723.15
0.1	1.15	3.15	985.59	112809.24	720.97

Table 4.7: Processing Times for the sequential and batch algorithms at different k values relative to that of S-CFS at $k=2$.

k	S-CFS	B-CFS	S-SDP	B-SDP	EKF
2	1.00	4.07	419.57	579.8	0.13
4	1.00	4.10	421.56	573.06	0.12
8	0.99	5.16	416.63	562.87	0.12
16	0.99	7.81	440.23	571.80	0.12
32	1.37	19.18	449.36	582.76	0.12

Multiple-Time

The number of sensors is $M = 10$ and that of successive measurements is $N = 35$. Fig. 4.4 gives the estimation MSE versus σ when the carrier frequency is exactly known. At low noise level of $\sigma = 0.001$ Hz, SDP could not reach the CRLB, possibly due to the relaxation accuracy limit. As σ increases, CFS leaves the CRLB first and followed by the CFS-ML, while SDP remains to follow the bound much further for large σ .

In Fig. 4.5, we show the results for unavailable carrier frequency situation where the observations are similar to those from Fig. 4.4. The accuracy of SDP at low noise level can reach the CRLB, since it is limited by noise instead of relaxation.

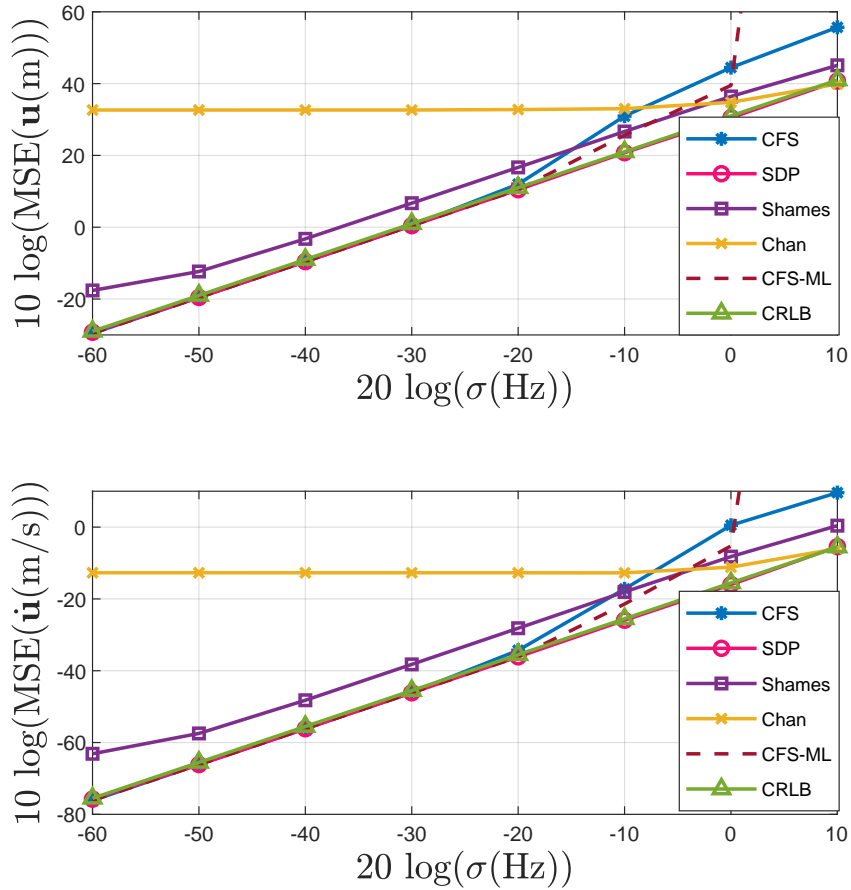


Figure 4.2: Performance of the proposed methods at different σ levels for single-time case when f_o is available.

Sequential Multiple-Time

We use $M = 16$ and $N = 5$ for this test. The algorithm updates the location of the object sequentially as k increases. Fig. 4.6 shows the performance at $k = 4$ when $\sigma_{f_o} = 0$. The sequential algorithm using CFS is denoted by S-CFS and that using SDP is S-SDP. When $\sigma \leq 0.05$ Hz, both the S-CFS and the S-SDP have optimal performance. At larger noise level, the S-CFS leaves the CRLB accuracy whereas the

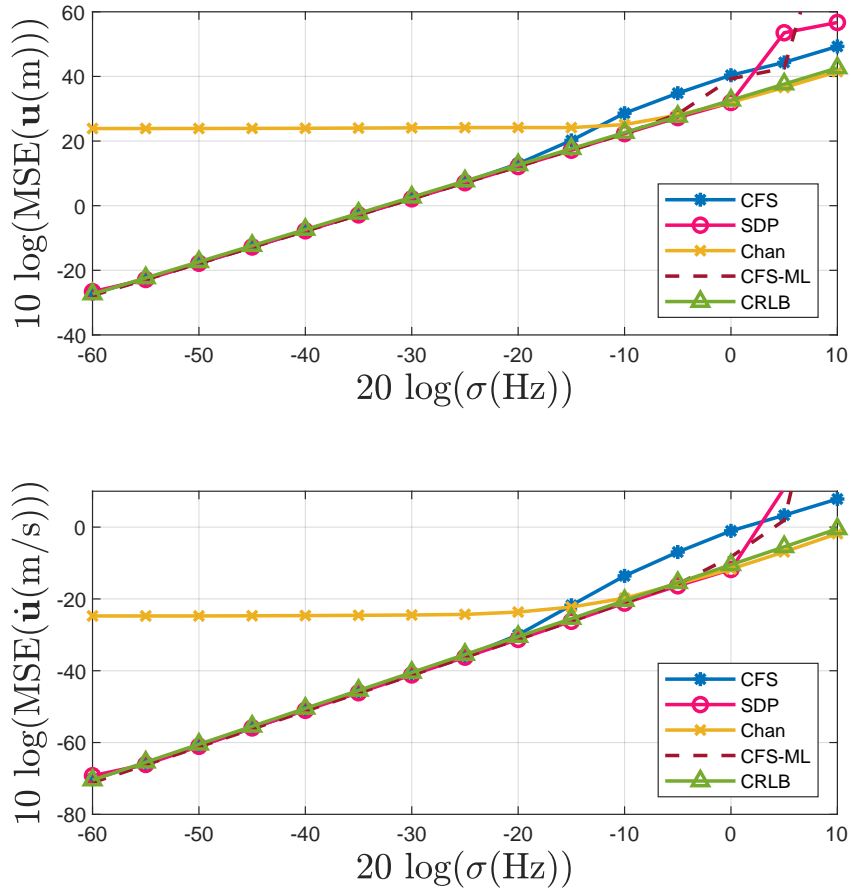


Figure 4.3: Performance of the proposed methods at different σ levels for single-time case when f_o is unavailable.

S-SDP continue to perform well.

4.7.2 Performance w.r.t. Sensor Position and Carrier Frequency Errors

The covariance matrix of sensor position errors is randomly set as

$$\mathbf{Q}_s = \sigma_s^2 \text{diag}\{ [6, 5, 8, 9, 8, 4, 8, 9, 6, 3, 3, 7, 9, 3, 6, 8, 8, 3, 3, 4, 5, 4, 6, 8, 7, 8,$$

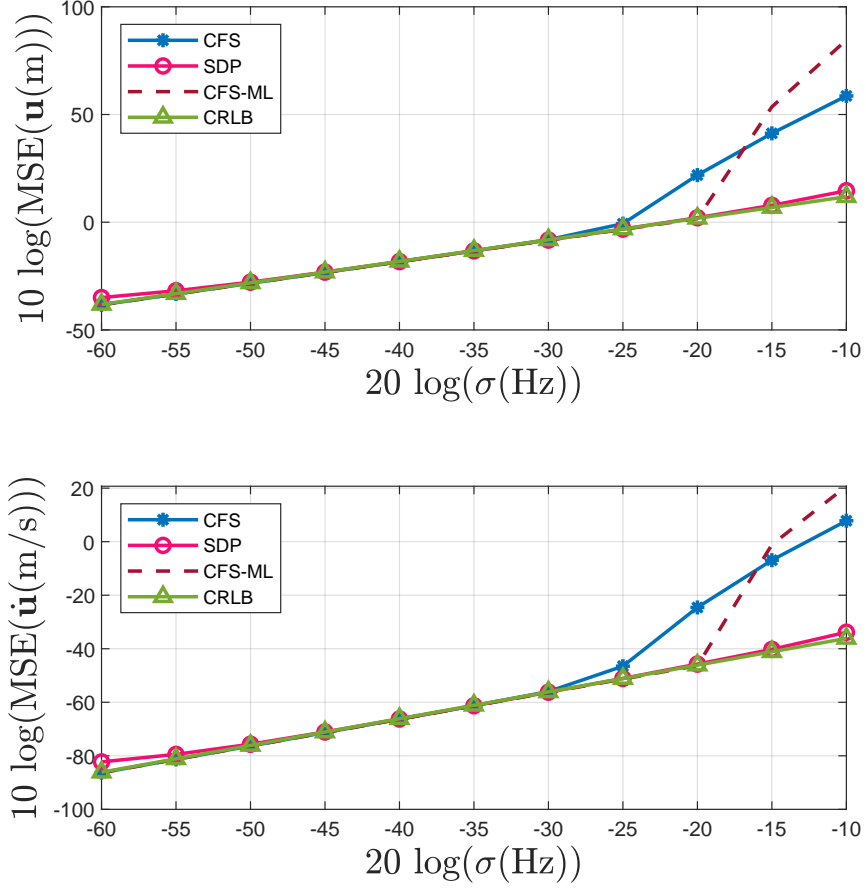


Figure 4.4: Performance of the proposed methods at different σ levels for multiple-time when f_o is available.

$$4, 9, 4, 8, 3, 3, 6, 6, 7, 7, 3, 9, 4, 2, 6, 3, 4, 3, 3, 4, 5, 4]/10\}, \quad (4.85)$$

where the unit of σ_s is meters. Fig. 4.7 gives the estimation performance with respect to σ_s when the carrier frequency error is absent, where the measurement noise covariance matrix is arbitrarily chosen as $\mathbf{Q}_n = 10^{-7} \text{diag}\{ [4, 3, 6, 7, 8, 1, 8, 10, 3, 0.1, 1, 4, 9, 1, 6, 8] \}$. Fig. 4.8 presents the results as σ_{f_o} increases while keeping $\sigma_s = 0$ and $\mathbf{Q}_n = 10^{-4} \mathbf{I}_{16}$. Both CFS and SDP show similar performance by having optimal

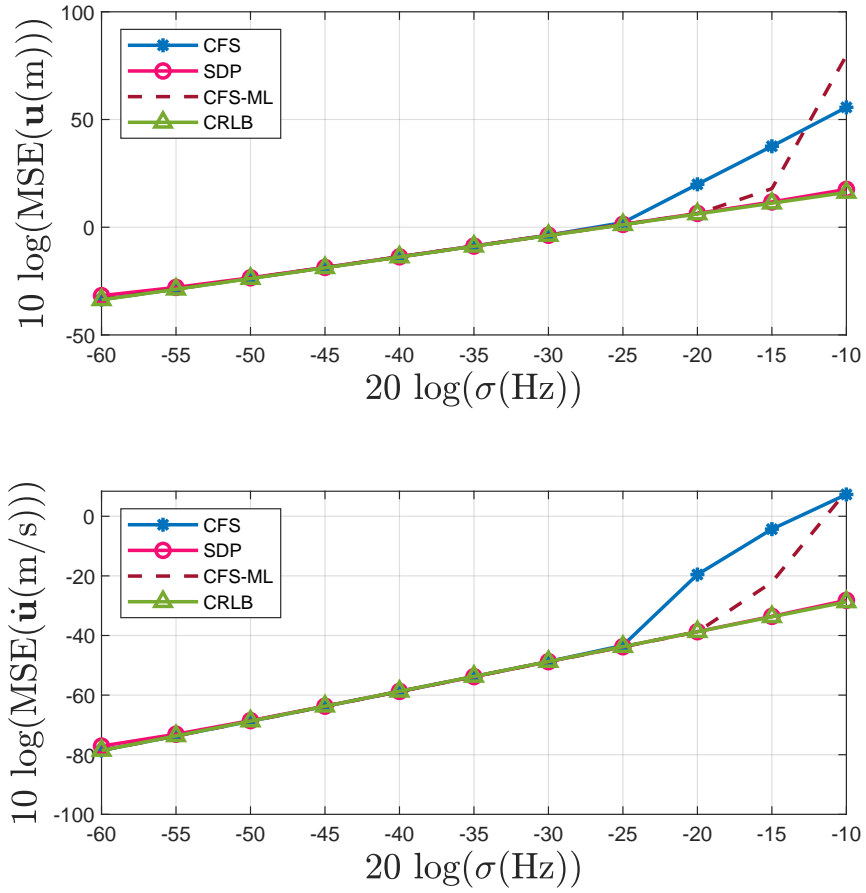


Figure 4.5: Performance of the proposed methods at different σ levels for multiple-time when f_o is unavailable.

accuracy over the region of small to moderate levels of errors. CFS-ML outperforms the others by its ability to reach the CRLB at larger noise level. When we neglect the error of carrier frequency by setting σ_{f_o} to 0 in (4.9) for CFS, the performance becomes much worse unless σ_{f_o} is very small where the accuracy is dominated by the measurement noise.

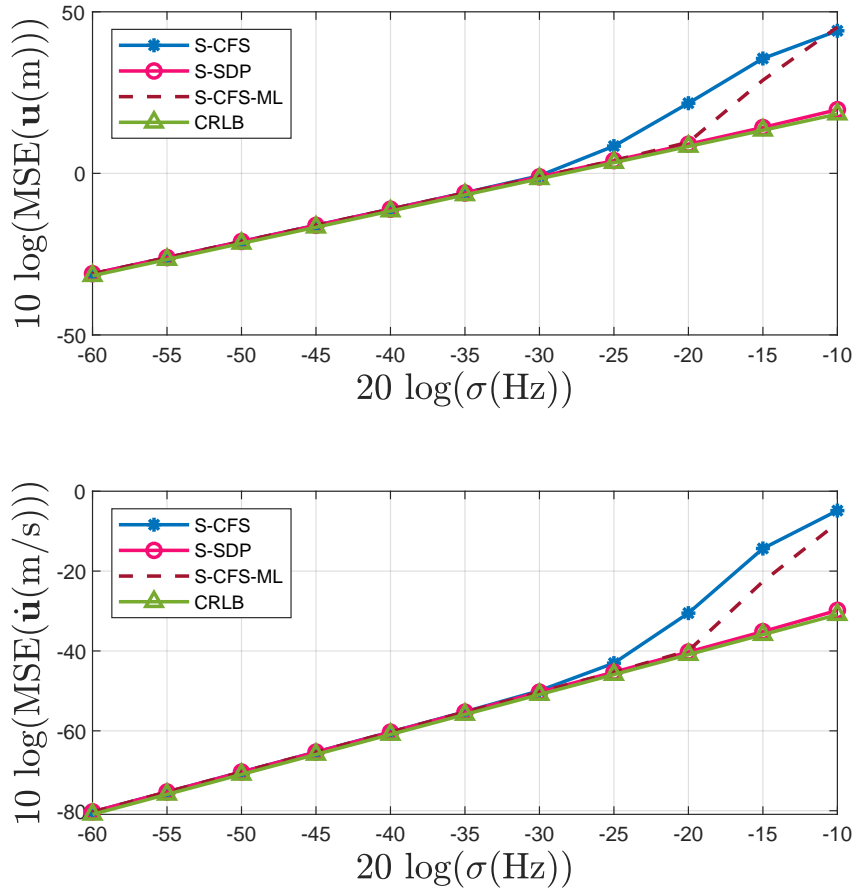


Figure 4.6: Performance of the proposed methods at different σ levels for the sequential estimation when f_o is available.

4.7.3 Sequential Estimation for Multiple-Time as Time k Increasing

Using $M = 16$ sensors and having the noise level at $\sigma = 0.1$ Hz, the MSE of the object location estimate at successive k values is plotted in Fig. 4.9 using only one randomly generated configuration where the sensor position error and carrier frequency noise are absent. We include the extended Kalman filter (EKF) [104] denoted by EKF as a comparison to our proposed sequential method. The state vector of EKF is $[\mathbf{u}^T, \dot{\mathbf{u}}^T]^T$

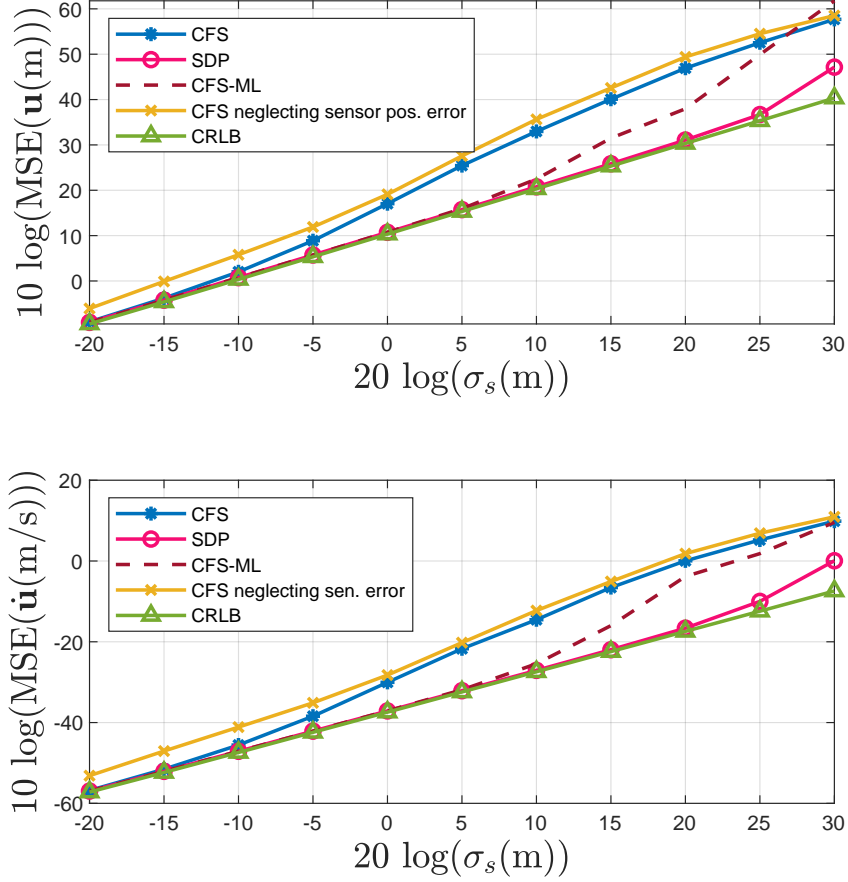


Figure 4.7: Performance of the proposed methods at different σ_s levels for single-time case.

and the input at each instant is the frequency measurement. It is initialized with the object state vector equal to $\mathbf{0}$ and the state covariance matrix by [112]

$$\mathbf{Q}_{state} = \text{diag} \{ 1500^2 \mathbf{I}_2/3, 500^2/3, 10^2 \mathbf{I}_3/9 \}. \quad (4.86)$$

The proposed sequential algorithms S-CFS and S-SDP are able to locate the object and track it well with the optimal CRLB accuracy for the tested values of k . While

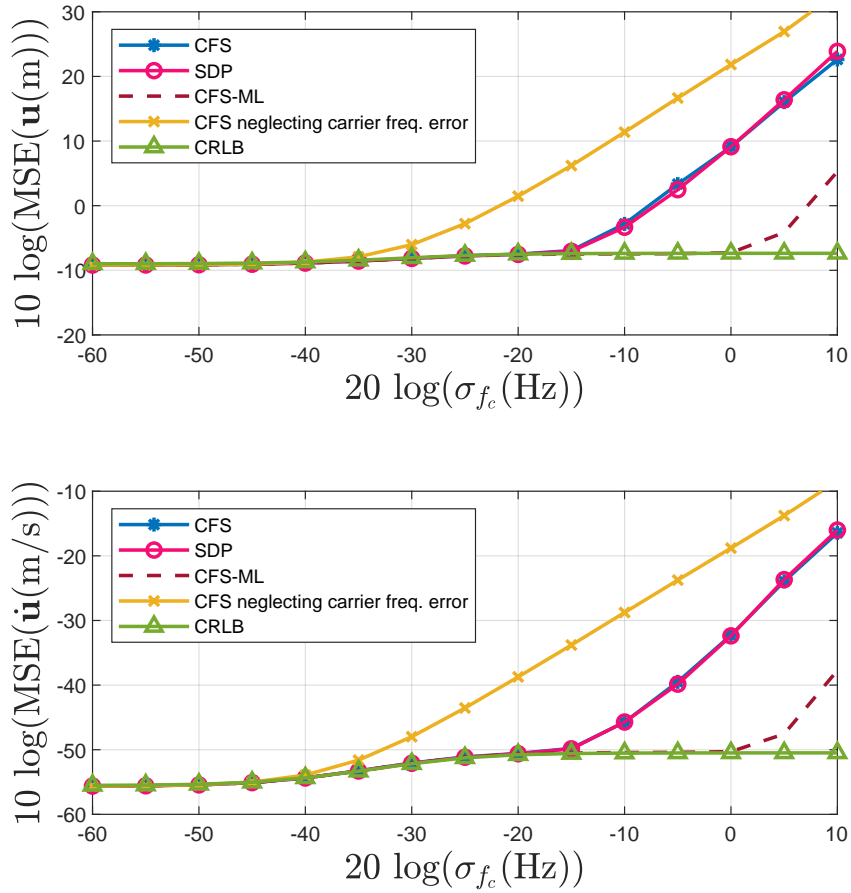


Figure 4.8: Performance of the proposed methods at different σ_{f_o} levels for single-time case when f_o is available.

EKF eventually can reach the CRLB when k is sufficiently large, they considerably outperform the EKF when k is small.

The processing times of the algorithms at several k values from Matlab implementation are recorded in Table 4.7, under the settings of Fig. 4.9. EKF is most computationally efficient, which takes about 1/8-th of the processing time of S-CFS. However, its tracking performance is inadequate. Both sequential algorithms, S-CFS and S-SDP, maintain nearly the same amounts of processing times regardless of k

and they are less than those of their batch counterparts, B-CFS and B-SDP. When k increases, the B-CFS processing time increases at almost constant rate and becomes large. The time consumed by the B-SDP does not vary much with k as it is dominated by the relaxation process which is mostly dependent on the number of unknowns and the number of constraints rather than the measurement data size.

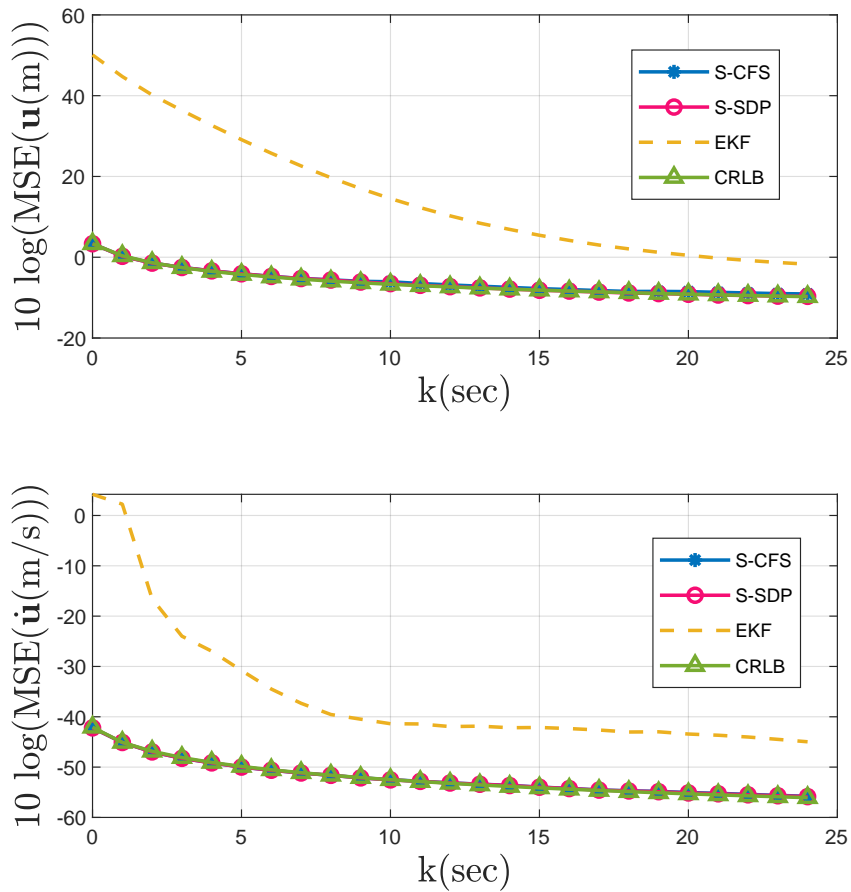


Figure 4.9: Performance of the proposed methods at different k for the sequential estimation when f_o is available.

4.7.4 Sensor Number

Figs. 4.10 and 4.11 give, respectively, the performance when using different number of sensors for the single-time and multiple-time scenarios. The settings are the same as in Fig. 4.2 and Fig. 4.4 for the two cases, while keeping the measurement noise level at $\sigma = 0.01$ Hz.

For the single-time scenario in Fig. 4.10, CFS needs at least 14 sensors to work properly due to the significant number of unknowns in (4.20). SDP attains the CRLB with only 9 sensors. Shames cannot reach the CRLB even when the number of sensors is large and has a very long processing time. Chan has poor performance regardless of the number of sensors due to the accuracy limited by the grid size of solution search. Using smaller grid size could improve the accuracy but it will be at the expense of consuming longer processing time.

Fig. 4.11 shows similar observations for the multiple-time case. CFS now requires 9 sensors to operate and SDP can reach the CRLB with only 6 sensors.

4.8 Summary

In this chapter, We have investigated the problem of locating a moving object in 3-D by using frequency measurements. Several scenarios including the single-time and the multiple-time, both having carrier frequency known with error or fully unknown are studied in the possible presence of sensor position errors. For each individual scenario, we proposed two solutions based on the algebraic closed-form approach or the semi-definite relaxation. The first is computationally efficient that consumes small amount of computation time and the latter is noise resilient that can tolerate large amount

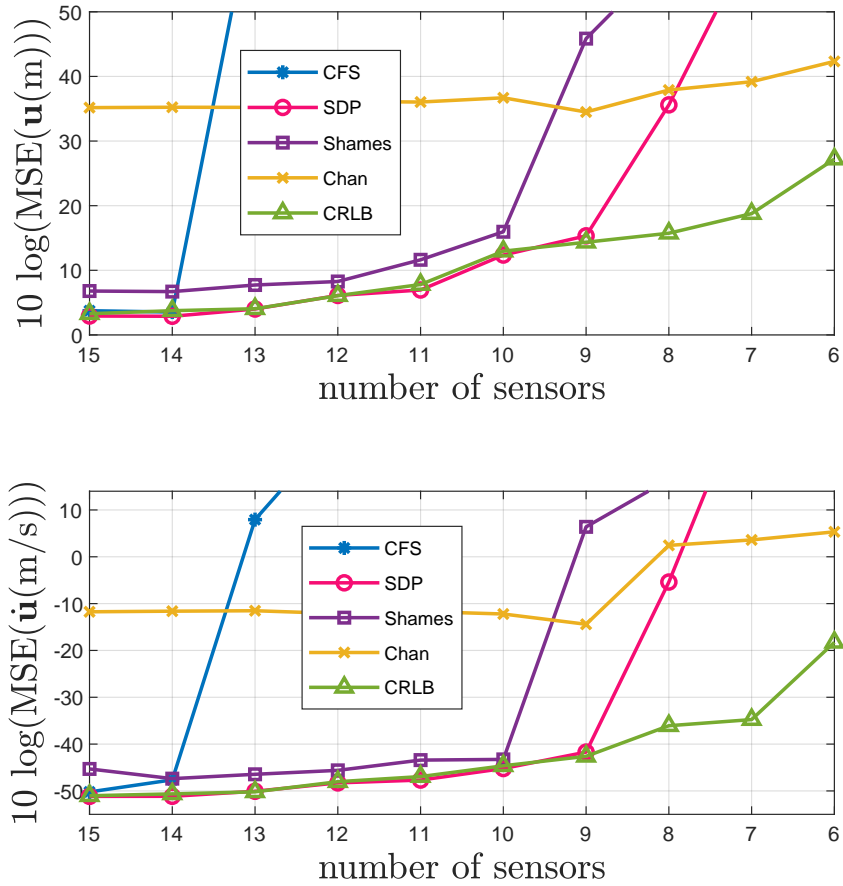


Figure 4.10: Performance of the proposed methods using different number of sensors for single-time case when f_o is available.

of noise. The semi-definite relaxation solution has several hundreds of constraints and we have evaluated each and listed the crucial ones that improve performance. Both solution methods attain the CRLB performance under their intended operating environments. We have also demonstrated CFS is an effective initialization for the iterative implementation of the MLE. Furthermore, a sequential estimator is developed for the multiple-time scenario where the object location parameters are updated upon a new observation arrives. Finally, the study contains theoretical analysis to prove

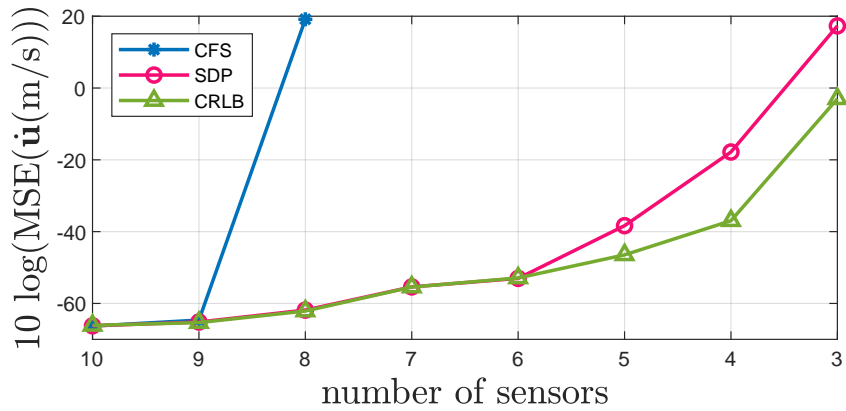
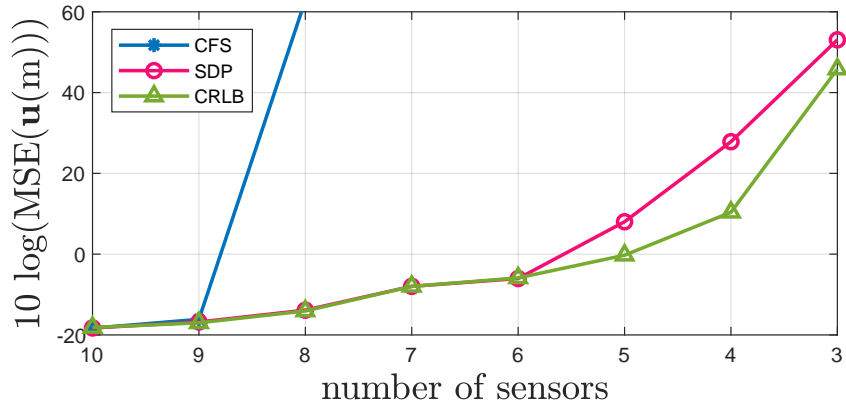


Figure 4.11: Performance of the proposed methods using different number of sensors for multiple-time when f_o is available.

that under small Gaussian noise, the algebraic solution and the sequential estimator can reach the CRLB accuracy having available and unavailable carrier frequency, with the simulations providing the validation.

Chapter 5

Moving Sensors Scenario

The measurement model given in (2.2) is valid when the sensors are fixed. To consider the sensor movement, we use the following model

$$f_{k,i} = f_o - \frac{f_o^o(\mathbf{u}_k^o - \mathbf{s}_{k,i}^o)^T(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)}{c \|\mathbf{u}_k^o - \mathbf{s}_{k,i}^o\|} + n_{k,i}, \quad (5.1)$$

where $\mathbf{s}_{k,i}^o \in \mathbb{R}^d$ and $\dot{\mathbf{s}}_{k,i}^o \in \mathbb{R}^d$ are the true position and velocity of sensor i at time instant k , and d is the localization dimension. $r_{k,i} = \|\mathbf{u}_k^o - \mathbf{s}_{k,i}^o\|$ is the Euclidean distance at instant k between the object and the noisy position of sensor i . Collecting $\mathbf{s}_{k,i}^o$ and $\dot{\mathbf{s}}_{k,i}^o$ over M sensors and N time observations gives

$$\mathbf{s}^o = [\mathbf{s}_1^{oT}, \mathbf{s}_2^{oT}, \dots, \mathbf{s}_k^{oT}, \dots, \mathbf{s}_{N-1}^{oT}]^T = \mathbf{s} - \Delta\mathbf{s}, \quad (5.2)$$

$$\dot{\mathbf{s}}^o = [\dot{\mathbf{s}}_1^{oT}, \dot{\mathbf{s}}_2^{oT}, \dots, \dot{\mathbf{s}}_k^{oT}, \dots, \dot{\mathbf{s}}_{N-1}^{oT}]^T = \dot{\mathbf{s}} - \Delta\dot{\mathbf{s}}, \quad (5.3)$$

and \mathbf{s}_k^o and $\dot{\mathbf{s}}_k^o$ are

$$\mathbf{s}_k^o = [\mathbf{s}_{k,1}^{oT}, \mathbf{s}_{k,2}^{oT}, \dots, \mathbf{s}_{k,M}^{oT}]^T, \quad (5.4)$$

$$\dot{\mathbf{s}}_k^o = [\dot{\mathbf{s}}_{k,1}^{oT}, \dot{\mathbf{s}}_{k,2}^{oT}, \dots, \dot{\mathbf{s}}_{k,M}^{oT}]^T. \quad (5.5)$$

$\Delta \mathbf{s}$ and $\Delta \dot{\mathbf{s}}$ are uncorrelated zero-mean Gaussian distributed error with covariance matrices given by \mathbf{Q}_s and $\mathbf{Q}_{\dot{s}}$ respectively. To solve for the unknowns \mathbf{u}^o and $\dot{\mathbf{u}}^o$, we assume that enough measurements are available and the errors of sensor position, sensor velocity and observation noise are uncorrelated and not significant in which the second and higher error terms are negligible. We shall first reformulate the problem to get an objective function with some second order constraints. Then, two solutions will be proposed to minimize the objective function and obtain the object location estimate. We shall also address the cases single-time and multiple-time measurements separately.

5.1 Formulation

We start by expressing the measurement model (5.1) in terms of the available quantities f_o , $\mathbf{s}_{k,i}$ and $\dot{\mathbf{s}}_{k,i}$ as

$$f_{k,i} = f_o - \frac{f_o(\mathbf{u}_k^o - \mathbf{s}_{k,i})^T(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i})}{c r_{k,i}} + \varepsilon_{k,i} + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|), \quad (5.6)$$

where $\varepsilon_{k,i}$ is the composite error term given by

$$\varepsilon_{k,i} = \left(\frac{\boldsymbol{\rho}_{k,i}^{oT}(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i})}{c} - 1 \right) \Delta f_o + \frac{-f_o}{c r_{k,i}^o} (\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i})^T \mathbf{P}_{k,i}^{\perp o} \Delta \mathbf{s}_{k,i} - \frac{f_o}{c} \boldsymbol{\rho}_{k,i}^{oT} \Delta \dot{\mathbf{s}}_{k,i} + n_{k,i}. \quad (5.7)$$

$\boldsymbol{\rho}_{k,i} = (\mathbf{u}_k^o - \mathbf{s}_{k,i})/r_{k,i}$ is a unit length vector pointing from $\mathbf{s}_{k,i}$ to \mathbf{u}_k^o and $\mathbf{P}_{k,i}^{\perp o} = \mathbf{I} - \boldsymbol{\rho}_{k,i}^o \boldsymbol{\rho}_{k,i}^{oT}$ is the orthogonal projection matrix of $\boldsymbol{\rho}_{k,i}^o$. Let $\mathbf{d}_{f,k}$ be the $M \times 1$ vector whose i -th element is $-1 + \boldsymbol{\rho}_{k,i}^{oT}(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)/c$, $\mathbf{D}_{s,k}$ as the $M \times dM$ matrix whose i -th row is zero except the elements $\mathbf{D}_{s,k}(i, d \times (i-1) + 1 : d \times (i)) = -(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)^T \mathbf{P}_{k,i}^{\perp o} f_o^o / (c r_{k,i}^o)$, and $\mathbf{D}_{\dot{s},k}$ as the $M \times dM$ matrix whose i -th row is zero except the elements $\mathbf{D}_{\dot{s},k}(i, d \times (i-1) + 1 : d \times (i)) = -\frac{f_o^o}{c} \boldsymbol{\rho}_{k,i}^{oT}$. Collecting the composite errors from different sensors at time k gives

$$\boldsymbol{\varepsilon}_k = \mathbf{d}_{f,k} \Delta f_o + \mathbf{D}_{s,k} \Delta \mathbf{s} + \mathbf{D}_{\dot{s},k} \Delta \dot{\mathbf{s}} + \mathbf{n}_k. \quad (5.8)$$

Defining the vector $\mathbf{d}_f = [\mathbf{d}_{f,0}^T, \mathbf{d}_{f,1}^T, \dots, \mathbf{d}_{f,N-1}^T]^T$, the matrix $\mathbf{D}_s = [\mathbf{D}_{s,0}^T, \mathbf{D}_{s,1}^T, \dots, \mathbf{D}_{s,N-1}^T]^T$, and the matrix $\mathbf{D}_{\dot{s}} = [\mathbf{D}_{\dot{s},0}^T, \mathbf{D}_{\dot{s},1}^T, \dots, \mathbf{D}_{\dot{s},N-1}^T]^T$, the error vector of all MN measurements $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}_0^T, \dots, \boldsymbol{\varepsilon}_{N-1}^T]^T$ is

$$\boldsymbol{\varepsilon} = \mathbf{d}_f \Delta f_o + \mathbf{D}_s \Delta \mathbf{s} + \mathbf{D}_{\dot{s}} \Delta \dot{\mathbf{s}} + \mathbf{n}. \quad (5.9)$$

$\boldsymbol{\varepsilon}$ remains Gaussian and has the covariance matrix equal to

$$\mathbf{Q}_{\boldsymbol{\varepsilon}} = E[\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T] = \sigma_{f_o}^2 \mathbf{d}_f \mathbf{d}_f^T + \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{D}_{\dot{s}} \mathbf{Q}_{\dot{s}} \mathbf{D}_{\dot{s}}^T + \mathbf{Q}_n. \quad (5.10)$$

To proceed further, it is more convenient to work with a scaled version of the normalized Doppler shift

$$d_{k,i} = c(f_{k,i} - f_o)/f_o. \quad (5.11)$$

It has the meaning of the object range rate at time k observed by sensor i . Rearranging (5.6) gives

$$(d_{k,i} - \varepsilon_{k,i}c/f_o) r_{k,i} = -(\mathbf{u}_k^o - \mathbf{s}_{k,i})^T (\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|). \quad (5.12)$$

Recall that $r_{k,i} = \|\mathbf{u}_k^o - \mathbf{s}_{k,i}\|$ which is defined before (5.1), squaring both sides yields

$$\begin{aligned} -2(c/f_o)d_{k,i}r_{k,i}^2\varepsilon_{k,i} + d_{k,i}^2 (\|\mathbf{s}_{k,i}\|^2 - 2\mathbf{s}_{k,i}^T \mathbf{u}_k^o + \|\mathbf{u}_k^o\|^2) &= (\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o)^2 + (\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o)^2 + (\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i})^2 \\ + (\mathbf{u}_k^{oT} \dot{\mathbf{u}}^o)^2 - 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{u}_k^{oT} \dot{\mathbf{u}}^o - 2\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}_k^o + 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{u}}^{oT} \mathbf{u}_k^o - 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o - 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \\ + 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|). \end{aligned} \quad (5.13)$$

\mathbf{u}_k^o is the object position at the time k . It is dependent on \mathbf{u}^o and $\dot{\mathbf{u}}^o$ only. Expressing it by (2.1), we obtain the following

$$(\mathbf{u}_k^{oT} \dot{\mathbf{u}}^o)^2 = (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 + 2k\mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o + k^2 (\dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o)^2, \quad (5.14a)$$

$$\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{u}_k^{oT} \dot{\mathbf{u}}^o = \dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + k\dot{\mathbf{s}}_{k,i}^T (\mathbf{u}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o + \dot{\mathbf{u}}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o) + k^2 \dot{\mathbf{s}}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o, \quad (5.14b)$$

$$\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o \mathbf{u}_k^{oT} \dot{\mathbf{u}}^o = \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \mathbf{u}^o + k\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o, \quad (5.14c)$$

$$\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \mathbf{u}_k^{oT} \dot{\mathbf{u}}^o = \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{u}}^{oT} \mathbf{u}^o + k\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o, \quad (5.14d)$$

$$\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o = \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o + k\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \dot{\mathbf{u}}^o, \quad (5.14e)$$

Inserting (5.14) in (5.13), we get

$$\begin{aligned} -2(c/f_o)d_{k,i}r_{k,i}^2\varepsilon_{k,i} + d_{k,i}^2 (\|\mathbf{s}_{k,i}\|^2 - 2\mathbf{s}_{k,i}^T \mathbf{u}_k^o + \|\mathbf{u}_k^o\|^2) &= (\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o)^2 + (\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o)^2 + (\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i})^2 \\ + (\mathbf{u}_k^{oT} \dot{\mathbf{u}}^o)^2 + 2k\mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o + k^2 (\dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o)^2 - 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{u}_k^{oT} \dot{\mathbf{u}}^o - 2k\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \end{aligned}$$

$$\begin{aligned}
& -2k^2 \dot{\mathbf{s}}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o - 2(\mathbf{s}_{k,i}^T + k \dot{\mathbf{s}}_{k,i}^T) \dot{\mathbf{u}}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o + 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{u}}^{oT} \mathbf{u}^o + 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \\
& - 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o - 2\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i}^T \dot{\mathbf{u}}^o + 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o + o(\varepsilon_{k,i}) + \Delta f_o o(1) \\
& + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|). \quad (5.15)
\end{aligned}$$

The terms of $(\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o)^2$, $(\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o)^2$ and $\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o$, can be expanded into

$$(\mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o)^2 = x_{k,i}^2 \dot{x}^{o2} + y_{k,i}^2 \dot{y}^{o2} + z_{k,i}^2 \dot{z}^{o2} + 2x_{k,i} y_{k,i} \dot{x}^o \dot{y}^o + 2x_{k,i} z_{k,i} \dot{x}^o \dot{z}^o + 2y_{k,i} z_{k,i} \dot{y}^o \dot{z}^o, \quad (5.16a)$$

$$\begin{aligned}
(\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o)^2 &= \dot{x}_{k,i}^2 x^{o2} + \dot{y}_{k,i}^2 y^{o2} + \dot{z}_{k,i}^2 z^{o2} + 2\dot{x}_{k,i} \dot{y}_{k,i} x^o y^o + 2\dot{x}_{k,i} \dot{z}_{k,i} x^o z^o + 2\dot{y}_{k,i} \dot{z}_{k,i} y^o z^o \\
&+ k^2 \dot{x}_{k,i}^2 \dot{x}^{o2} + k^2 \dot{y}_{k,i}^2 \dot{y}^{o2} + k^2 \dot{z}_{k,i}^2 \dot{z}^{o2} + 2k^2 \dot{x}_{k,i} \dot{y}_{k,i} \dot{x}^o \dot{y}^o + 2k^2 \dot{x}_{k,i} \dot{z}_{k,i} \dot{x}^o \dot{z}^o \\
&+ 2k^2 \dot{y}_{k,i} \dot{z}_{k,i} \dot{y}^o \dot{z}^o + 2k \dot{x}_{k,i}^2 x^o \dot{x}^o + 2k \dot{y}_{k,i}^2 y^o \dot{y}^o + 2k \dot{z}_{k,i}^2 z^o \dot{z}^o \\
&+ 2k \dot{x}_{k,i} \dot{y}_{k,i} (y^o \dot{x}^o + x^o \dot{y}^o) + 2k \dot{x}_{k,i} \dot{z}_{k,i} (z^o \dot{x}^o + x^o \dot{z}^o) \\
&+ 2k \dot{y}_{k,i} \dot{z}_{k,i} (y^o \dot{z}^o + z^o \dot{y}^o), \quad (5.16b)
\end{aligned}$$

$$\begin{aligned}
\dot{\mathbf{s}}_{k,i}^T \mathbf{u}_k^o \mathbf{s}_{k,i}^T \dot{\mathbf{u}}^o &= x_{k,i} \dot{x}_{k,i} x^o \dot{x}^o + y_{k,i} \dot{y}_{k,i} y^o \dot{y}^o + z_{k,i} \dot{z}_{k,i} z^o \dot{z}^o + y_{k,i} \dot{x}_{k,i} x^o \dot{y}^o + z_{k,i} \dot{x}_{k,i} x^o \dot{z}^o \\
&+ x_{k,i} \dot{y}_{k,i} y^o \dot{x}^o + z_{k,i} \dot{y}_{k,i} y^o \dot{z}^o + x_{k,i} \dot{z}_{k,i} z^o \dot{x}^o + y_{k,i} \dot{z}_{k,i} z^o \dot{y}^o + k x_{k,i} \dot{x}_{k,i} \dot{x}^{o2} \\
&+ k y_{k,i} \dot{y}_{k,i} \dot{y}^{o2} + k z_{k,i} \dot{z}_{k,i} \dot{z}^{o2} + k (y_{k,i} \dot{x}_{k,i} + x_{k,i} \dot{y}_{k,i}) \dot{x}^o \dot{y}^o \\
&+ k (z_{k,i} \dot{x}_{k,i} + x_{k,i} \dot{z}_{k,i}) \dot{x}^o \dot{z}^o + k (z_{k,i} \dot{y}_{k,i} + y_{k,i} \dot{z}_{k,i}) \dot{y}^o \dot{z}^o. \quad (5.16c)
\end{aligned}$$

Inserting (5.16) in (5.15) and rearranging,

$$\begin{aligned}
& 2(c/f_o) d_{k,i} r_{k,i}^2 \varepsilon_{k,i} + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|) = d_{k,i}^2 \mathbf{s}_{k,i}^T \mathbf{s}_{k,i} - (\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i})^2 \\
& + 2(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i} - d_{k,i}^2 \mathbf{s}_{k,i})^T \mathbf{u}^o + 2(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} (k \dot{\mathbf{s}}_{k,i} + \mathbf{s}_{k,i}) - k d_{k,i}^2 \mathbf{s}_{k,i})^T \dot{\mathbf{u}}^o + (d_{k,i}^2 - \dot{x}_{k,i}^2) x^{o2} \\
& + (d_{k,i}^2 - \dot{y}_{k,i}^2) y^{o2} + (d_{k,i}^2 - \dot{z}_{k,i}^2) z^{o2} + (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (x_{k,i} + k \dot{x}_{k,i})^2) \dot{x}^{o2}
\end{aligned}$$

$$\begin{aligned}
& + (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (y_{k,i} + k \dot{y}_{k,i})^2) \dot{y}^{o2} + (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (z_{k,i} + k \dot{z}_{k,i})^2) \dot{z}^{o2} \\
& - 2\dot{x}_{k,i} \dot{y}_{k,i} x^o y^o - 2\dot{x}_{k,i} \dot{z}_{k,i} x^o z^o - 2\dot{y}_{k,i} \dot{z}_{k,i} y^o z^o - 2(x_{k,i} + k \dot{x}_{k,i})(y_{k,i} + k \dot{y}_{k,i}) \dot{x}^o \dot{y}^o \\
& - 2(x_{k,i} + k \dot{x}_{k,i})(z_{k,i} + k \dot{z}_{k,i}) \dot{x}^o \dot{z}^o - 2(y_{k,i} + k \dot{y}_{k,i})(z_{k,i} + k \dot{z}_{k,i}) \dot{y}^o \dot{z}^o \\
& + 2(k d_{k,i}^2 - k \dot{x}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - x_{k,i} \dot{x}_{k,i}) x^o \dot{x}^o + 2(k d_{k,i}^2 - k \dot{y}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - y_{k,i} \dot{y}_{k,i}) y^o \dot{y}^o \\
& + 2(k d_{k,i}^2 - k \dot{z}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - z_{k,i} \dot{z}_{k,i}) z^o \dot{z}^o - 2\dot{x}_{k,i} (k \dot{y}_{k,i} + y_{k,i}) x^o \dot{y}^o - 2\dot{y}_{k,i} (k \dot{x}_{k,i} + x_{k,i}) y^o \dot{x}^o \\
& - 2\dot{z}_{k,i} (k \dot{x}_{k,i} + x_{k,i}) z^o \dot{x}^o - 2\dot{x}_{k,i} (k \dot{z}_{k,i} + z_{k,i}) x^o \dot{z}^o - 2\dot{y}_{k,i} (k \dot{z}_{k,i} + z_{k,i}) y^o \dot{z}^o \\
& - 2\dot{z}_{k,i} (k \dot{y}_{k,i} + y_{k,i}) z^o \dot{y}^o + 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + 2(\mathbf{s}_{k,i} + k \dot{\mathbf{s}}_{k,i})^T \dot{\mathbf{u}}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + 2k \dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \\
& + 2(k^2 \dot{\mathbf{s}}_{k,i} + k \mathbf{s}_{k,i})^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o - (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 - 2k \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o - k^2 (\dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o)^2 \quad (5.17)
\end{aligned}$$

5.1.1 Single-Time Measurement

We set $k = 0$ and drop the time zero index for simplicity, (5.17) reduces to

$$\begin{aligned}
& 2(c/f_o) d_i r_i^2 \varepsilon_i + o(\varepsilon_i) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_i\|) + o(\|\Delta \dot{\mathbf{s}}_i\|) = d_i^2 \mathbf{s}_i^T \mathbf{s}_i - (\mathbf{s}_i^T \dot{\mathbf{s}}_i)^2 \\
& + 2(\mathbf{s}_i^T \dot{\mathbf{s}}_i \dot{\mathbf{s}}_i - d_i^2 \mathbf{s}_i)^T \mathbf{u}^o + 2\mathbf{s}_i^T \dot{\mathbf{s}}_i \mathbf{s}_i^T \dot{\mathbf{u}}^o + \mathbf{a}_i^T \mathbf{b}^o + 2\dot{\mathbf{s}}_i^T \mathbf{u}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + 2\mathbf{s}_i^T \dot{\mathbf{u}}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o - (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2, \quad (5.18)
\end{aligned}$$

\mathbf{a}_i and \mathbf{b}^o are vectors given in (5.25) and (5.26) and in (5.28) and (5.29) for 2-D and 3-D localization respectively. (5.18) remains to be a highly nonlinear equation with respect to \mathbf{u}^o and $\dot{\mathbf{u}}^o$. We shall formulate the localization problem as a constrained optimization. Let the unknown vector be

$$\boldsymbol{\varphi}^o = \left[\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}, \mathbf{b}^{oT}, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \mathbf{u}^{oT}, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT}, (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 \right]^T. \quad (5.19)$$

Also, define the matrix

$$\mathbf{A} = \begin{bmatrix} -2(\mathbf{s}_1^T \dot{\mathbf{s}}_1 \dot{\mathbf{s}}_1 - d_1^2 \mathbf{s}_1)^T & -2\mathbf{s}_1^T \dot{\mathbf{s}}_1 \mathbf{s}_1^T & -\mathbf{a}_1^T & -2\dot{\mathbf{s}}_1^T & -2\mathbf{s}_1^T & 1 \\ -2(\mathbf{s}_2^T \dot{\mathbf{s}}_2 \dot{\mathbf{s}}_2 - d_2^2 \mathbf{s}_2)^T & -2\mathbf{s}_2^T \dot{\mathbf{s}}_2 \mathbf{s}_2^T & -\mathbf{a}_2^T & -2\dot{\mathbf{s}}_2^T & -2\mathbf{s}_2^T & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -2(\mathbf{s}_M^T \dot{\mathbf{s}}_M \dot{\mathbf{s}}_M - d_M^2 \mathbf{s}_M)^T & -2\mathbf{s}_M^T \dot{\mathbf{s}}_M \mathbf{s}_M^T & -\mathbf{a}_M^T & -2\dot{\mathbf{s}}_M^T & -2\mathbf{s}_M^T & 1 \end{bmatrix}, \quad (5.20)$$

the vector

$$\mathbf{h} = \left[d_1^2 \mathbf{s}_1^T \mathbf{s}_1 - (\mathbf{s}_1^T \dot{\mathbf{s}}_1)^2, d_2^2 \mathbf{s}_2^T \mathbf{s}_2 - (\mathbf{s}_2^T \dot{\mathbf{s}}_2)^2, \dots, d_M^2 \mathbf{s}_M^T \mathbf{s}_M - (\mathbf{s}_M^T \dot{\mathbf{s}}_M)^2 \right]^T, \quad (5.21)$$

and the $M \times M$ matrix

$$\mathbf{B} = 2 \frac{c}{f_o} \text{diag} \{ [d_1 r_1^2, d_2 r_2^2, \dots, d_M r_M^2] \}. \quad (5.22)$$

Over $i = 1, 2, \dots, M$, (5.18) forms the matrix equation after dropping the second and higher order error terms,

$$\mathbf{B} \boldsymbol{\varepsilon} \simeq \mathbf{h} - \mathbf{A} \boldsymbol{\varphi}^o. \quad (5.23)$$

The approximation is valid when the error is small. The covariance matrix for the equation error (5.23) is equal to

$$\text{cov}(\mathbf{B} \boldsymbol{\varepsilon}) = \mathbf{B} E [\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T] \mathbf{B}^T = \mathbf{B} \mathbf{Q}_\varepsilon \mathbf{B}^T, \quad (5.24)$$

where \mathbf{Q}_ε is given by (5.10) with $N = 1$. We shall build the optimization problem for 2-D and 3-D localization separately in the following two subsections.

2-D Localization

The transformed observation equation in (5.17) describes the relation among the unknowns and the available data in the 3-D coordinates. Considering the 2-D version of the problem and single-time measurement, (5.17) reduces to (5.18) with the following definitions of \mathbf{a}_i and \mathbf{b}^o

$$\mathbf{a}_i = [(d_i^2 - \dot{x}_i^2), (d_i^2 - \dot{y}_i^2), -x_i^2, -y_i^2, -2\dot{x}_i\dot{y}_i, -2x_i y_i, -2(\mathbf{s}_i^T \dot{\mathbf{s}}_i + x_i \dot{x}_i), \\ -2(\mathbf{s}_i^T \dot{\mathbf{s}}_i + y_i \dot{y}_i), -2y_i \dot{x}_i, -2x_i \dot{y}_i]^T. \quad (5.25)$$

$$\mathbf{b}^o = [x^{o2}, y^{o2}, \dot{x}^{o2}, \dot{y}^{o2}, x^o y^o, \dot{x}^o \dot{y}^o, x^o \dot{x}^o, y^o \dot{y}^o, x^o \dot{y}^o, y^o \dot{x}^o]^T. \quad (5.26)$$

The unknown vector $\boldsymbol{\varphi}^o$ in (5.23) has 19 elements but the number of independent variables is only 4. Fifteen constraints are necessary to relate the elements of the variable $\boldsymbol{\varphi}$ for the estimation of $\boldsymbol{\varphi}^o$. Based on $\boldsymbol{\varphi}^o$ defined in (5.19), Table 5.1 shows the individual components of $\boldsymbol{\varphi}$ and lists the relations among the elements. The localization problem can be cast as a weighted least-squares (WLS) optimization under a set of constraints as follows:

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (5.27a)$$

$$\text{s.t. All the constraints listed on the right column of Table 5.1.} \quad (5.27b)$$

The 15 quadratic constraints (5.27b) come from the relations among the nuisance variables $\boldsymbol{\varphi}(5 : 19)$ and the actual unknowns $\boldsymbol{\varphi}(1 : 4)$. There are other second order relations among the elements of $\boldsymbol{\varphi}$ are redundant in the formulation (5.27). They will be used for the SDP solution method to improve the tightness of the optimization

Table 5.1: The Elements of φ and The 15 Constraints for The 2-D Single-Time Measurement Case.

Elements	Relations
$\varphi(1) = x$	$\varphi(5) = \varphi(1)^2$
$\varphi(2) = y$	$\varphi(6) = \varphi(2)^2$
$\varphi(3) = \dot{x}$	$\varphi(7) = \varphi(3)^2$
$\varphi(4) = \dot{y}$	$\varphi(8) = \varphi(4)^2$
$\varphi(5) = x^2$	
$\varphi(6) = y^2$	$\varphi(9) = \varphi(1)\varphi(2)$
$\varphi(7) = \dot{x}^2$	$\varphi(10) = \varphi(3)\varphi(4)$
$\varphi(8) = \dot{y}^2$	$\varphi(11) = \varphi(1)\varphi(3)$
$\varphi(9) = xy$	$\varphi(12) = \varphi(2)\varphi(4)$
$\varphi(10) = \dot{x}\dot{y}$	$\varphi(13) = \varphi(1)\varphi(4)$
$\varphi(11) = x\dot{x}$	$\varphi(14) = \varphi(2)\varphi(3)$
$\varphi(12) = y\dot{y}$	
$\varphi(13) = x\dot{y}$	$\varphi(15) = \varphi(1)\varphi(12) + \varphi(3)\varphi(5)$
$\varphi(14) = y\dot{x}$	$\varphi(16) = \varphi(2)\varphi(11) + \varphi(4)\varphi(6)$
$\varphi(15) = x(x\dot{x} + y\dot{y})$	$\varphi(17) = \varphi(1)\varphi(7) + \varphi(4)\varphi(14)$
$\varphi(16) = y(x\dot{x} + y\dot{y})$	$\varphi(18) = \varphi(2)\varphi(8) + \varphi(3)\varphi(13)$
$\varphi(17) = \dot{x}(x\dot{x} + y\dot{y})$	$\varphi(19) = \varphi(1)\varphi(17) + \varphi(2)\varphi(18)$
$\varphi(18) = \dot{y}(x\dot{x} + y\dot{y})$	
$\varphi(19) = (x\dot{x} + y\dot{y})^2$	

when it is approximated with SDR.

The constrained optimization problem (5.27) will be solved using unconstrained minimization or convex optimization that will be described in Sections 5.2 and 5.3 respectively.

3-D Localization

For 3-D localization, the vectors \mathbf{a}_i and \mathbf{b}^o of (5.18) are defined as follows

$$\begin{aligned}
 \mathbf{a}_i = & [(d_i^2 - \dot{x}_i^2), (d_i^2 - \dot{y}_i^2), (d_i^2 - \dot{z}_i^2), -x_i^2, -y_i^2, -z_i^2, -2\dot{x}_i\dot{y}_i, -2\dot{x}_i\dot{z}_i, -2\dot{y}_i\dot{z}_i, \\
 & -2x_iy_i, -2x_iz_i, -2y_iz_i, -2(\mathbf{s}_i^T \dot{\mathbf{s}}_i + x_i\dot{x}_i), -2(\mathbf{s}_i^T \dot{\mathbf{s}}_i + y_i\dot{y}_i), -2(\mathbf{s}_i^T \dot{\mathbf{s}}_i + z_i\dot{z}_i), \\
 & -2y_i\dot{x}_i, -2z_i\dot{x}_i, -2x_i\dot{y}_i, -2z_i\dot{y}_i, -2x_i\dot{z}_i, -2y_i\dot{z}_i]^T, \quad (5.28)
 \end{aligned}$$

$$\mathbf{b}^o = [x^{o2}, y^{o2}, z^{o2}, \dot{x}^{o2}, \dot{y}^{o2}, \dot{z}^{o2}, x^o y^o, x^o z^o, y^o z^o, \dot{x}^o \dot{y}^o, \dot{x}^o \dot{z}^o, \dot{y}^o \dot{z}^o, x^o \dot{x}^o, y^o \dot{y}^o, z^o \dot{z}^o, x^o \dot{y}^o, x^o \dot{z}^o, y^o \dot{x}^o, y^o \dot{z}^o, z^o \dot{x}^o, z^o \dot{y}^o]^T. \quad (5.29)$$

The unknown vector $\boldsymbol{\varphi}^o$ has 34 elements in this case with only 6 independent variables. The elements $\boldsymbol{\varphi}^o(7 : 34)$ are polynomial function of the first 6 elements of $\boldsymbol{\varphi}^o$. Thus, we shall build 28 constraints to relate the dependent variables with the actual unknowns, $\boldsymbol{\varphi}^o(1 : 6)$. The following weighted least-square (WLS) optimization along with the 28 constraints form the localization problem for this case. The constraints are directly derived from the definition of $\boldsymbol{\varphi}^o$.

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (5.30a)$$

$$\text{s.t. All the constraints listed in right column of Table 5.2,} \quad (5.30b)$$

5.1.2 Multiple-Time Measurements

The transformed observation equation at a certain time instant k is (5.17). For simplicity, we incorporate the vectors $\mathbf{a}_{k,i}$ and \mathbf{b}^o to get

$$\begin{aligned} 2(c/f_o)d_{k,i}r_{k,i}^2\varepsilon_{k,i} + o(\varepsilon_{k,i}) + \Delta f_o o(1) + o(\|\Delta \mathbf{s}_{k,i}\|) + o(\|\Delta \dot{\mathbf{s}}_{k,i}\|) &= d_{k,i}^2 \mathbf{s}_{k,i}^T \mathbf{s}_{k,i} - (\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i})^2 \\ &+ 2(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i} - d_{k,i}^2 \mathbf{s}_{k,i})^T \mathbf{u}^o + 2(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} (k\dot{\mathbf{s}}_{k,i} + \mathbf{s}_{k,i}) - k d_{k,i}^2 \mathbf{s}_{k,i})^T \dot{\mathbf{u}}^o + \mathbf{a}_{k,i}^T \mathbf{b}^o \\ &+ 2\dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + 2(\mathbf{s}_{k,i} + k\dot{\mathbf{s}}_{k,i})^T \dot{\mathbf{u}}^o \mathbf{u}^{oT} \dot{\mathbf{u}}^o + 2k\dot{\mathbf{s}}_{k,i}^T \mathbf{u}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o + 2k(k\dot{\mathbf{s}}_{k,i} + \mathbf{s}_{k,i})^T \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \\ &- (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2 - 2k \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o - k^2 (\dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o)^2, \quad (5.31) \end{aligned}$$

Table 5.2: The Elements of φ and The 28 Constraints for The 3-D Single-Time Measurement Case.

Elements	Relations
$\varphi(1) = x$	$\varphi(7) = \varphi(1)^2$
$\varphi(2) = y$	$\varphi(8) = \varphi(2)^2$
$\varphi(3) = z$	$\varphi(9) = \varphi(3)^2$
$\varphi(4) = \dot{x}$	$\varphi(10) = \varphi(4)^2$
$\varphi(5) = \dot{y}$	$\varphi(11) = \varphi(5)^2$
$\varphi(6) = \dot{z}$	$\varphi(12) = \varphi(6)^2$
$\varphi(7) = x^2$	
$\varphi(8) = y^2$	$\varphi(13) = \varphi(1)\varphi(2)$
$\varphi(9) = z^2$	$\varphi(14) = \varphi(1)\varphi(3)$
$\varphi(10) = \dot{x}^2$	$\varphi(15) = \varphi(2)\varphi(3)$
$\varphi(11) = \dot{y}^2$	$\varphi(16) = \varphi(4)\varphi(5)$
$\varphi(12) = \dot{z}^2$	$\varphi(17) = \varphi(4)\varphi(6)$
$\varphi(13) = xy$	$\varphi(18) = \varphi(5)\varphi(6)$
$\varphi(14) = xz$	$\varphi(19) = \varphi(1)\varphi(4)$
$\varphi(15) = yz$	$\varphi(20) = \varphi(2)\varphi(5)$
$\varphi(16) = \dot{x}\dot{y}$	$\varphi(21) = \varphi(3)\varphi(6)$
$\varphi(17) = \dot{x}\dot{z}$	$\varphi(22) = \varphi(1)\varphi(5)$
$\varphi(18) = \dot{y}\dot{z}$	$\varphi(23) = \varphi(1)\varphi(6)$
$\varphi(19) = x\dot{x}$	$\varphi(24) = \varphi(2)\varphi(4)$
$\varphi(20) = y\dot{y}$	$\varphi(25) = \varphi(2)\varphi(6)$
$\varphi(21) = z\dot{z}$	$\varphi(26) = \varphi(3)\varphi(4)$
$\varphi(22) = x\dot{y}$	$\varphi(27) = \varphi(3)\varphi(5)$
$\varphi(23) = x\dot{z}$	
$\varphi(24) = y\dot{x}$	
$\varphi(25) = y\dot{z}$	$\varphi(28) = \varphi(1)\varphi(19) + \varphi(2)\varphi(22) + \varphi(3)\varphi(23)$
$\varphi(26) = z\dot{x}$	$= \varphi(4)\varphi(7) + \varphi(5)\varphi(13) + \varphi(6)\varphi(14)$
$\varphi(27) = z\dot{y}$	$\varphi(29) = \varphi(1)\varphi(24) + \varphi(2)\varphi(20) + \varphi(3)\varphi(25)$
$\varphi(28) = x(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(13) + \varphi(5)\varphi(8) + \varphi(6)\varphi(15)$
$\varphi(29) = y(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(30) = \varphi(1)\varphi(26) + \varphi(2)\varphi(27) + \varphi(3)\varphi(21)$
$\varphi(30) = z(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(14) + \varphi(5)\varphi(15) + \varphi(6)\varphi(9)$
$\varphi(31) = \dot{x}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(31) = \varphi(1)\varphi(10) + \varphi(2)\varphi(16) + \varphi(3)\varphi(17)$
$\varphi(32) = \dot{y}(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(19) + \varphi(5)\varphi(24) + \varphi(6)\varphi(26)$
$\varphi(33) = \dot{z}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(32) = \varphi(1)\varphi(16) + \varphi(2)\varphi(11) + \varphi(3)\varphi(18)$
$\varphi(34) = (x\dot{x} + y\dot{y} + z\dot{z})^2$	$= \varphi(4)\varphi(22) + \varphi(5)\varphi(20) + \varphi(6)\varphi(27)$
	$\varphi(33) = \varphi(1)\varphi(17) + \varphi(2)\varphi(18) + \varphi(3)\varphi(12)$
	$= \varphi(4)\varphi(23) + \varphi(5)\varphi(25) + \varphi(6)\varphi(21)$
	$\varphi(34) = \varphi(1)\varphi(31) + \varphi(2)\varphi(32) + \varphi(3)\varphi(33)$
	$= \varphi(4)\varphi(28) + \varphi(5)\varphi(29) + \varphi(6)\varphi(30)$

and $\mathbf{a}_{k,i}$ and \mathbf{b}^o are given by (5.41) and (5.42) for 2-D localization and (5.44) and (5.45) for 3-D localization. By collecting in each term the lumped variable involving the object position and velocity, we define the unknown vector as

$$\boldsymbol{\varphi}^o = \left[\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}, \mathbf{b}^{oT}, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \mathbf{u}^{oT}, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT}, (\mathbf{u}^{oT} \dot{\mathbf{u}}^o)^2, \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT}, \mathbf{u}^{oT} \dot{\mathbf{u}}^o \dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o, (\dot{\mathbf{u}}^{oT} \dot{\mathbf{u}}^o)^2 \right]^T. \quad (5.32)$$

Let \mathbf{A}_k be the matrix having the i -th row

$$\begin{aligned} \mathbf{A}_k(i, :) = & \left[-2 \left(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} \dot{\mathbf{s}}_{k,i} - d_{k,i}^2 \mathbf{s}_{k,i} \right)^T, -2 \left(\mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} (k \dot{\mathbf{s}}_{k,i} + \mathbf{s}_{k,i}) - k d_{k,i}^2 \mathbf{s}_{k,i} \right)^T, -\mathbf{a}_{k,i}^T, \right. \\ & \left. -2 \dot{\mathbf{s}}_{k,i}^T, -2 (\mathbf{s}_{k,i} + k \dot{\mathbf{s}}_{k,i})^T, 1, -2k \dot{\mathbf{s}}_{k,i}^T, -2k (k \dot{\mathbf{s}}_{k,i} + \mathbf{s}_{k,i})^T, 2k, k^2 \right], \quad (5.33) \end{aligned}$$

the length M vector \mathbf{h}_k be

$$\begin{aligned} \mathbf{h}_k = & \left[d_{k,1}^2 \mathbf{s}_{k,1}^T \mathbf{s}_{k,1} - (\mathbf{s}_{k,1}^T \dot{\mathbf{s}}_{k,1})^2, d_{k,2}^2 \mathbf{s}_{k,2}^T \mathbf{s}_{k,2} - (\mathbf{s}_{k,2}^T \dot{\mathbf{s}}_{k,2})^2, \dots, \right. \\ & \left. d_{k,M}^2 \mathbf{s}_{k,M}^T \mathbf{s}_{k,M} - (\mathbf{s}_{k,M}^T \dot{\mathbf{s}}_{k,M})^2 \right]^T, \quad (5.34) \end{aligned}$$

and the size $M \times M$ matrix \mathbf{B}_k be

$$\mathbf{B}_k = 2 \frac{c}{f_o} \text{diag} \left\{ \left[d_{k,1} r_{k,1}^2, d_{k,2} r_{k,2}^2, \dots, d_{k,M} r_{k,M}^2 \right] \right\}. \quad (5.35)$$

Putting together \mathbf{A}_k , \mathbf{h}_k and \mathbf{B}_k for $k = 0, 1, \dots, N-1$ separately such that

$$\mathbf{A} = \left[\mathbf{A}_0^T, \mathbf{A}_1^T, \dots, \mathbf{A}_{N-1}^T \right]^T, \quad (5.36)$$

$$\mathbf{h} = [\mathbf{h}_0^T, \mathbf{h}_1^T, \dots, \mathbf{h}_{N-1}^T]^T, \quad (5.37)$$

$$\mathbf{B} = \text{diag} \{ \mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{N-1} \}, \quad (5.38)$$

we can represent all MN equations over $i = 1, \dots, M$ and $k = 0, \dots, N - 1$ of (5.31) in a matrix form as

$$\mathbf{B}\boldsymbol{\varepsilon} \simeq \mathbf{h} - \mathbf{A}\boldsymbol{\varphi}^o, \quad (5.39)$$

where the second and higher order errors terms have been ignored and the approximation is reasonable when the errors are small. The covariance matrix of (5.39) is given by

$$\text{cov}(\mathbf{B}\boldsymbol{\varepsilon}) = \mathbf{B} E [\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T] \mathbf{B}^T = \mathbf{B} \mathbf{Q}_\varepsilon \mathbf{B}^T, \quad (5.40)$$

and \mathbf{Q}_ε is given by (5.10) with $N > 1$. The following two subsections present the optimization problem for 2-D and 3-D localization separately.

2-D Localization

In the 2-D coordinates, we define the vectors $\mathbf{a}_{k,i}$ and \mathbf{b}^o of (5.31) as

$$\begin{aligned} \mathbf{a}_{k,i} = & [(d_{k,i}^2 - \dot{x}_{k,i}^2), (d_{k,i}^2 - \dot{y}_{k,i}^2), ((k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i}) - (x_{k,i} + k\dot{x}_{k,i})^2), \\ & ((k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i}) - (y_{k,i} + k\dot{y}_{k,i})^2), -2\dot{x}_{k,i}\dot{y}_{k,i}, -2(x_{k,i} + k\dot{x}_{k,i})(y_{k,i} + k\dot{y}_{k,i}), \\ & 2(k d_{k,i}^2 - k\dot{x}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - x_{k,i}\dot{x}_{k,i}), 2(k d_{k,i}^2 - k\dot{y}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - y_{k,i}\dot{y}_{k,i}), \\ & -2\dot{x}_{k,i}(k\dot{y}_{k,i} + y_{k,i}), -2\dot{y}_{k,i}(k\dot{x}_{k,i} + x_{k,i})]^T, \quad (5.41) \end{aligned}$$

$$\mathbf{b}^o = [x^{o2}, y^{o2}, \dot{x}^{o2}, \dot{y}^{o2}, x^o y^o, \dot{x}^o \dot{y}^o, x^o \dot{x}^o, y^o \dot{y}^o, x^o \dot{y}^o, y^o \dot{x}^o]^T, \quad (5.42)$$

Table 5.3: The Elements of $\boldsymbol{\varphi}$ and The 21 Constraints for The 2-D Multiple-Time Measurements Case.

Elements	Constraints
$\varphi(1) = x$	$\varphi(5) = \varphi(1)\varphi(1)$
$\varphi(2) = y$	$\varphi(6) = \varphi(2)\varphi(2)$
$\varphi(3) = \dot{x}$	$\varphi(7) = \varphi(3)\varphi(3)$
$\varphi(4) = \dot{y}$	$\varphi(8) = \varphi(4)\varphi(4)$
$\varphi(5) = x^2$	$\varphi(9) = \varphi(1)\varphi(2)$
$\varphi(6) = y^2$	$\varphi(10) = \varphi(3)\varphi(4)$
$\varphi(7) = \dot{x}^2$	$\varphi(11) = \varphi(1)\varphi(3)$
$\varphi(8) = \dot{y}^2$	$\varphi(12) = \varphi(2)\varphi(4)$
$\varphi(9) = xy$	$\varphi(13) = \varphi(1)\varphi(4)$
$\varphi(10) = \dot{x}\dot{y}$	$\varphi(14) = \varphi(2)\varphi(3)$
$\varphi(11) = x\dot{x}$	$\varphi(15) = \varphi(1)\varphi(11) + \varphi(2)\varphi(13)$
$\varphi(12) = y\dot{y}$	$\varphi(16) = \varphi(3)\varphi(9) + \varphi(4)\varphi(6)$
$\varphi(13) = x\dot{y}$	$\varphi(17) = \varphi(1)\varphi(7) + \varphi(2)\varphi(10)$
$\varphi(14) = y\dot{x}$	$\varphi(18) = \varphi(3)\varphi(13) + \varphi(4)\varphi(12)$
$\varphi(15) = x(x\dot{x} + y\dot{y})$	$\varphi(19) = \varphi(1)\varphi(17) + \varphi(2)\varphi(18)$
$\varphi(16) = y(x\dot{x} + y\dot{y})$	$\varphi(20) = \varphi(1)\varphi(8) + \varphi(3)\varphi(11)$
$\varphi(17) = \dot{x}(x\dot{x} + y\dot{y})$	$\varphi(21) = \varphi(2)\varphi(7) + \varphi(4)\varphi(12)$
$\varphi(18) = \dot{y}(x\dot{x} + y\dot{y})$	$\varphi(22) = \varphi(3)\varphi(7) + \varphi(4)\varphi(10)$
$\varphi(19) = (x\dot{x} + y\dot{y})^2$	$\varphi(23) = \varphi(3)\varphi(10) + \varphi(4)\varphi(8)$
$\varphi(20) = x(\dot{x}^2 + \dot{y}^2)$	$\varphi(24) = \varphi(3)\varphi(17) + \varphi(4)\varphi(18)$
$\varphi(21) = y(\dot{x}^2 + \dot{y}^2)$	$\varphi(25) = \varphi(3)\varphi(22) + \varphi(4)\varphi(23)$
$\varphi(22) = \dot{x}(\dot{x}^2 + \dot{y}^2)$	
$\varphi(23) = \dot{y}(\dot{x}^2 + \dot{y}^2)$	
$\varphi(24) = (x\dot{x} + y\dot{y})(\dot{x}^2 + \dot{y}^2)$	
$\varphi(25) = (\dot{x}^2 + \dot{y}^2)^2$	

In this case $\boldsymbol{\varphi}^o$ has 25 elements and the individual variables are shown in Table 5.3. The number of actual unknowns is only 4 and the elements are related to each other. Table 5.3 tabulates the second order relations that relates nuisance variables of $\boldsymbol{\varphi}$ with the object location independent variables. The optimization for the multiple-time measurements case, using the weighting matrix in (5.47) with N larger than one, is

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (5.43a)$$

$$\text{s.t. All the constraints listed on the right column of Table 5.3.} \quad (5.43b)$$

The relations in Table 5.3 are used to impose constraints among the 25 elements of $\boldsymbol{\varphi}$, for the purpose to fix the number of independent variables to only 4. There are other redundant second order relations among the elements of $\boldsymbol{\varphi}$ that are not needed in the formulation but they will be used later to improve the optimization when (5.43) is approximated with SDR.

3-D Localization

We define the vectors $\mathbf{a}_{k,i}$ and \mathbf{b}^o of (5.31) here as

$$\begin{aligned} \mathbf{a}_{k,i} = & [(d_{k,i}^2 - \dot{x}_{k,i}^2) , (d_{k,i}^2 - \dot{y}_{k,i}^2) , (d_{k,i}^2 - \dot{z}_{k,i}^2) , (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (x_{k,i} + k \dot{x}_{k,i})^2) , \\ & (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (y_{k,i} + k \dot{y}_{k,i})^2) , (k^2 d_{k,i}^2 - 2k \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - (z_{k,i} + k \dot{z}_{k,i})^2) , -2\dot{x}_{k,i} \dot{y}_{k,i} , \\ & -2\dot{x}_{k,i} \dot{z}_{k,i} , -2\dot{y}_{k,i} \dot{z}_{k,i} , -2(x_{k,i} + k \dot{x}_{k,i})(y_{k,i} + k \dot{y}_{k,i}) , -2(x_{k,i} + k \dot{x}_{k,i})(z_{k,i} + k \dot{z}_{k,i}) , \\ & -2(y_{k,i} + k \dot{y}_{k,i})(z_{k,i} + k \dot{z}_{k,i}) , 2(k d_{k,i}^2 - k \dot{x}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - x_{k,i} \dot{x}_{k,i}) , \\ & 2(k d_{k,i}^2 - k \dot{y}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - y_{k,i} \dot{y}_{k,i}) , 2(k d_{k,i}^2 - k \dot{z}_{k,i}^2 - \mathbf{s}_{k,i}^T \dot{\mathbf{s}}_{k,i} - z_{k,i} \dot{z}_{k,i}) , \\ & -2\dot{x}_{k,i} (k \dot{y}_{k,i} + y_{k,i}) , -2\dot{y}_{k,i} (k \dot{x}_{k,i} + x_{k,i}) , -2\dot{z}_{k,i} (k \dot{x}_{k,i} + x_{k,i}) , -2\dot{x}_{k,i} (k \dot{z}_{k,i} + z_{k,i}) , \\ & -2\dot{y}_{k,i} (k \dot{z}_{k,i} + z_{k,i}) , -2\dot{z}_{k,i} (k \dot{y}_{k,i} + y_{k,i})]^T , \quad (5.44) \end{aligned}$$

$$\begin{aligned} \mathbf{b}^o = & [x^{o2} , y^{o2} , z^{o2} , \dot{x}^{o2} , \dot{y}^{o2} , \dot{z}^{o2} , x^o y^o , x^o z^o , y^o z^o , \dot{x}^o \dot{y}^o , \dot{x}^o \dot{z}^o , \dot{y}^o \dot{z}^o , x^o \dot{x}^o , y^o \dot{y}^o , \\ & z^o \dot{z}^o , x^o \dot{y}^o , x^o \dot{z}^o , y^o \dot{x}^o , y^o \dot{z}^o , z^o \dot{x}^o , z^o \dot{y}^o]^T , \quad (5.45) \end{aligned}$$

and in the same way, we build the following optimization problem,

$$\min_{\boldsymbol{\varphi}} J = (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi})^T \mathbf{W} (\mathbf{h} - \mathbf{A}\boldsymbol{\varphi}), \quad (5.46a)$$

$$\text{s.t. All the constraints listed on the right column of Table 5.4.} \quad (5.46b)$$

$\boldsymbol{\varphi}$ in (5.46) has 42 unknown elements of which only 6 are independents. The elements $\boldsymbol{\varphi}(7 : 42)$ are directly related to the first 6 unknowns of $\boldsymbol{\varphi}$ as shown in Table 5.4. These relations represent the key for our developed algebraic solution and also the SDP solution through the SDR method.

The matrix \mathbf{W} mentioned in (5.27a), (5.30a), (5.43a) and (5.46a) is an approximation for the weighting matrix and it is set to

$$\mathbf{W} = \text{cov}(\mathbf{B}\boldsymbol{\epsilon})^{-1}, \quad (5.47)$$

where $\text{cov}(\mathbf{B}\boldsymbol{\epsilon})$ is given in (5.24) and (5.40) for single-time and multiple-time measurements respectively.

5.2 Algebraic Solution

We shall present in this section an algebraic closed-form solution to solve the constrained WLS problems for localization. The method is computationally attractive and will reach the CRLB performance in the small noise environment. First, we assume the elements of $\boldsymbol{\varphi}$ are independent variables to obtain the WLS solution. Next, we exploit the constraints to refine the estimate through nonlinear transformation

Table 5.4: The Elements of φ and The 36 Constraints for The 3-D Multiple-Time Measurements Case.

Elements	Constraints
$\varphi(1) = x$	$\varphi(7) = \varphi(1)^2$
$\varphi(2) = y$	$\varphi(8) = \varphi(2)^2$
$\varphi(3) = z$	$\varphi(9) = \varphi(3)^2$
$\varphi(4) = \dot{x}$	$\varphi(10) = \varphi(4)^2$
$\varphi(5) = \dot{y}$	$\varphi(11) = \varphi(5)^2$
$\varphi(6) = \dot{z}$	$\varphi(12) = \varphi(6)^2$
$\varphi(7) = x^2$	$\varphi(13) = \varphi(1)\varphi(2)$
$\varphi(8) = y^2$	$\varphi(14) = \varphi(1)\varphi(3)$
$\varphi(9) = z^2$	$\varphi(15) = \varphi(2)\varphi(3)$
$\varphi(10) = \dot{x}^2$	$\varphi(16) = \varphi(4)\varphi(5)$
$\varphi(11) = \dot{y}^2$	$\varphi(17) = \varphi(4)\varphi(6)$
$\varphi(12) = \dot{z}^2$	$\varphi(18) = \varphi(5)\varphi(6)$
$\varphi(13) = xy$	$\varphi(19) = \varphi(1)\varphi(4)$
$\varphi(14) = xz$	$\varphi(20) = \varphi(2)\varphi(5)$
$\varphi(15) = yz$	$\varphi(21) = \varphi(3)\varphi(6)$
$\varphi(16) = \dot{x}\dot{y}$	$\varphi(22) = \varphi(1)\varphi(5)$
$\varphi(17) = \dot{x}\dot{z}$	$\varphi(23) = \varphi(1)\varphi(6)$
$\varphi(18) = \dot{y}\dot{z}$	$\varphi(24) = \varphi(2)\varphi(4)$
$\varphi(19) = x\dot{x}$	$\varphi(25) = \varphi(2)\varphi(6)$
$\varphi(20) = y\dot{y}$	$\varphi(26) = \varphi(3)\varphi(4)$
$\varphi(21) = z\dot{z}$	$\varphi(27) = \varphi(3)\varphi(5)$
$\varphi(22) = x\dot{y}$	$\varphi(28) = \varphi(1)\varphi(19) + \varphi(2)\varphi(22) + \varphi(3)\varphi(23)$
$\varphi(23) = x\dot{z}$	$= \varphi(4)\varphi(7) + \varphi(5)\varphi(13) + \varphi(6)\varphi(14)$
$\varphi(24) = y\dot{x}$	$\varphi(29) = \varphi(1)\varphi(24) + \varphi(2)\varphi(20) + \varphi(3)\varphi(25)$
$\varphi(25) = y\dot{z}$	$= \varphi(4)\varphi(13) + \varphi(5)\varphi(8) + \varphi(6)\varphi(15)$
$\varphi(26) = z\dot{x}$	$\varphi(30) = \varphi(1)\varphi(26) + \varphi(2)\varphi(27) + \varphi(3)\varphi(21)$
$\varphi(27) = z\dot{y}$	$= \varphi(4)\varphi(14) + \varphi(5)\varphi(15) + \varphi(6)\varphi(9)$
$\varphi(28) = x(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(31) = \varphi(1)\varphi(10) + \varphi(2)\varphi(16) + \varphi(3)\varphi(17)$
$\varphi(29) = y(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(19) + \varphi(5)\varphi(24) + \varphi(6)\varphi(26)$
$\varphi(30) = z(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(32) = \varphi(1)\varphi(16) + \varphi(2)\varphi(11) + \varphi(3)\varphi(18)$
$\varphi(31) = \dot{x}(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(22) + \varphi(5)\varphi(20) + \varphi(6)\varphi(27)$
$\varphi(32) = \dot{y}(x\dot{x} + y\dot{y} + z\dot{z})$	$\varphi(33) = \varphi(1)\varphi(17) + \varphi(2)\varphi(18) + \varphi(3)\varphi(12)$
$\varphi(33) = \dot{z}(x\dot{x} + y\dot{y} + z\dot{z})$	$= \varphi(4)\varphi(23) + \varphi(5)\varphi(25) + \varphi(6)\varphi(21)$
$\varphi(34) = (x\dot{x} + y\dot{y} + z\dot{z})^2$	$\varphi(34) = \varphi(1)\varphi(31) + \varphi(2)\varphi(32) + \varphi(3)\varphi(33)$
$\varphi(35) = x(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(35) = \varphi(4)\varphi(19) + \varphi(5)\varphi(22) + \varphi(6)\varphi(25)$
$\varphi(36) = y(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(36) = \varphi(4)\varphi(23) + \varphi(5)\varphi(20) + \varphi(6)\varphi(26)$
$\varphi(37) = z(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(37) = \varphi(4)\varphi(24) + \varphi(5)\varphi(27) + \varphi(6)\varphi(21)$
$\varphi(38) = \dot{x}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(38) = \varphi(4)\varphi(10) + \varphi(5)\varphi(16) + \varphi(6)\varphi(17)$
$\varphi(39) = \dot{y}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(39) = \varphi(4)\varphi(16) + \varphi(5)\varphi(11) + \varphi(6)\varphi(18)$
$\varphi(40) = \dot{z}(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$\varphi(40) = \varphi(4)\varphi(17) + \varphi(5)\varphi(18) + \varphi(6)\varphi(12)$
$\varphi(41) = (x\dot{x} + y\dot{y} + z\dot{z})(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)$	$= \varphi(4)\varphi(28) + \varphi(5)\varphi(29) + \varphi(6)\varphi(30)$
$\varphi(42) = (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^2$	$\varphi(41) = \varphi(1)\varphi(38) + \varphi(2)\varphi(39) + \varphi(3)\varphi(40)$
	$= \varphi(4)\varphi(31) + \varphi(5)\varphi(32) + \varphi(6)\varphi(33)$
	$\varphi(42) = \varphi(4)\varphi(38) + \varphi(5)\varphi(39) + \varphi(6)\varphi(40)$

[12].

Although the matrix and vector variables \mathbf{A} , \mathbf{B} , \mathbf{h} and \mathbf{W} are defined differently for the four different cases discussed in 5.1.1 and 5.1.2, the solution to the optimization problems given in (5.27), (5.30), (5.43) and (5.46) is

$$\boldsymbol{\varphi} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{h}. \quad (5.48)$$

When the noise in \mathbf{A} is not significant, we can reasonably approximate the covariance matrix of the WLS solution $\boldsymbol{\varphi}$ by [12],

$$\text{cov}(\boldsymbol{\varphi}) \simeq (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1}. \quad (5.49)$$

The weighting matrix \mathbf{W} is given by (5.47) and it is unavailable since \mathbf{B} and \mathbf{Q}_ϵ are function of the true object location. Nevertheless, we can construct \mathbf{W} by using the noisy elements of (5.48) obtained from its initial estimate when setting $\mathbf{W} = \mathbf{I}$ in (5.48). The error resulting for the solution of $\boldsymbol{\varphi}$ can be neglected as the WLS solution is not sensitive to the error in the weighting matrix [9, 10].

Now, we shall utilize the relations among the elements of $\boldsymbol{\varphi}$ to improve the estimation accuracy. Let us introduce the separate unknown vector

$$\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T, \quad (5.50)$$

Also, let the pseudo data vector be the WLS solution (5.48). The relations among the elements of $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}^o$ can be used to perform a nonlinear transformation in which every element in the data vector can be expressed in terms of the elements of $\boldsymbol{\varphi}$ and

θ° . In particular,

$$\varphi(1 : 2d) = \theta^\circ(1 : 2d) + \Delta\varphi(1 : 2d), \quad (5.51a)$$

$$\varphi(2d + 1) = \varphi(1)\theta^\circ(1) - \varphi^\circ(1)\Delta\varphi(1) + \Delta\varphi(2d + 1), \quad (5.51b)$$

$$\varphi(2d + 2) = \varphi(2)\theta^\circ(2) - \varphi^\circ(2)\Delta\varphi(2) + \Delta\varphi(2d + 2), \quad (5.51c)$$

$$\varphi(3d + 1) = \varphi(d + 1)\theta^\circ(d + 1) - \varphi^\circ(d + 1)\Delta\varphi(d + 1) + \Delta\varphi(3d + 1), \quad (5.51d)$$

$$\varphi(3d + 2) = \varphi(d + 2)\theta^\circ(d + 2) - \varphi^\circ(d + 2)\Delta\varphi(d + 2) + \Delta\varphi(3d + 2), \quad (5.51e)$$

$$\begin{aligned} \varphi(4d + 1) = & \frac{1}{2}\varphi(2)\theta^\circ(1) + \frac{1}{2}\varphi(1)\theta^\circ(2) - \frac{1}{2}\varphi^\circ(2)\Delta\varphi(1) - \frac{1}{2}\varphi^\circ(1)\Delta\varphi(2) \\ & + \Delta\varphi(4d + 1), \end{aligned} \quad (5.51f)$$

where $\Delta\varphi$ is the estimation error of the WLS solution. The nonlinear transformation in (5.51) is valid for all the different cases of localization scenarios we are discussing in this chapter. The transformation for the rest of elements in the data vector is given in (5.52), in which we coded the indices of the data vector and reveal it for the four different cases in Table 5.5.

$$\varphi(c1) = \varphi(3)\theta^\circ(3) - \varphi^\circ(3)\Delta\varphi(3) + \Delta\varphi(9), \quad (5.52a)$$

$$\varphi(c2) = \varphi(6)\theta^\circ(6) - \varphi^\circ(6)\Delta\varphi(6) + \Delta\varphi(12), \quad (5.52b)$$

$$\varphi(c3) = \frac{1}{2}\varphi(3)\theta^\circ(1) + \frac{1}{2}\varphi(1)\theta^\circ(3) - \frac{1}{2}\varphi^\circ(1)\Delta\varphi(3) - \frac{1}{2}\varphi^\circ(3)\Delta\varphi(1) + \Delta\varphi(14), \quad (5.52c)$$

$$\varphi(c4) = \frac{1}{2}\varphi(3)\theta^\circ(2) + \frac{1}{2}\varphi(2)\theta^\circ(3) - \frac{1}{2}\varphi^\circ(2)\Delta\varphi(3) - \frac{1}{2}\varphi^\circ(3)\Delta\varphi(2) + \Delta\varphi(15), \quad (5.52d)$$

$$\begin{aligned} \varphi(c5) = & \frac{1}{2}\varphi(d + 2)\theta^\circ(d + 1) + \frac{1}{2}\varphi(d + 1)\theta^\circ(d + 2) - \frac{1}{2}\varphi^\circ(d + 2)\Delta\varphi(d + 1) \\ & - \frac{1}{2}\varphi^\circ(d + 1)\Delta\varphi(d + 2) + \Delta\varphi(c5), \end{aligned} \quad (5.52e)$$

$$\varphi(c6) = \frac{1}{2}\varphi(6)\theta^\circ(4) + \frac{1}{2}\varphi(4)\theta^\circ(6) - \frac{1}{2}\varphi^\circ(4)\Delta\varphi(6) - \frac{1}{2}\varphi^\circ(6)\Delta\varphi(4) + \Delta\varphi(17), \quad (5.52f)$$

$$\varphi(c7) = \frac{1}{2}\varphi(6)\theta^\circ(5) + \frac{1}{2}\varphi(5)\theta^\circ(6) - \frac{1}{2}\varphi^\circ(5)\Delta\varphi(6) - \frac{1}{2}\varphi^\circ(6)\Delta\varphi(5) + \Delta\varphi(18), \quad (5.52g)$$

$$\begin{aligned}\varphi(c8) &= \frac{1}{2}\varphi(d+1)\theta^o(1) + \frac{1}{2}\varphi(1)\theta^o(d+1) - \frac{1}{2}\varphi^o(d+1)\Delta\varphi(1) - \frac{1}{2}\varphi^o(1)\Delta\varphi(d+1) \\ &\quad + \Delta\varphi(c8),\end{aligned}\tag{5.52h}$$

$$\begin{aligned}\varphi(c9) &= \frac{1}{2}\varphi(d+2)\theta^o(2) + \frac{1}{2}\varphi(2)\theta^o(d+2) - \frac{1}{2}\varphi^o(d+2)\Delta\varphi(2) - \frac{1}{2}\varphi^o(2)\Delta\varphi(d+2) \\ &\quad + \Delta\varphi(c9),\end{aligned}\tag{5.52i}$$

$$\varphi(c10) = \frac{1}{2}\varphi(6)\theta^o(3) + \frac{1}{2}\varphi(3)\theta^o(6) - \frac{1}{2}\varphi^o(6)\Delta\varphi(3) - \frac{1}{2}\varphi^o(3)\Delta\varphi(6) + \Delta\varphi(21),\tag{5.52j}$$

$$\begin{aligned}\varphi(c11) &= \frac{1}{2}\varphi(d+2)\theta^o(1) + \frac{1}{2}\varphi(1)\theta^o(d+2) - \frac{1}{2}\varphi^o(d+2)\Delta\varphi(1) - \frac{1}{2}\varphi^o(1)\Delta\varphi(d+2) \\ &\quad + \Delta\varphi(c11),\end{aligned}\tag{5.52k}$$

$$\varphi(c12) = \frac{1}{2}\varphi(6)\theta^o(1) + \frac{1}{2}\varphi(1)\theta^o(6) - \frac{1}{2}\varphi^o(6)\Delta\varphi(1) - \frac{1}{2}\varphi^o(1)\Delta\varphi(6) + \Delta\varphi(23),\tag{5.52l}$$

$$\begin{aligned}\varphi(c13) &= \frac{1}{2}\varphi(d+1)\theta^o(2) + \frac{1}{2}\varphi(2)\theta^o(d+1) - \frac{1}{2}\varphi^o(d+1)\Delta\varphi(2) - \frac{1}{2}\varphi^o(2)\Delta\varphi(d+1) \\ &\quad + \Delta\varphi(c13),\end{aligned}\tag{5.52m}$$

$$\varphi(c14) = \frac{1}{2}\varphi(6)\theta^o(2) + \frac{1}{2}\varphi(2)\theta^o(6) - \frac{1}{2}\varphi^o(6)\Delta\varphi(2) - \frac{1}{2}\varphi^o(2)\Delta\varphi(6) + \Delta\varphi(25),\tag{5.52n}$$

$$\varphi(c15) = \frac{1}{2}\varphi(4)\theta^o(3) + \frac{1}{2}\varphi(3)\theta^o(4) - \frac{1}{2}\varphi^o(4)\Delta\varphi(3) - \frac{1}{2}\varphi^o(3)\Delta\varphi(4) + \Delta\varphi(26),\tag{5.52o}$$

$$\varphi(c16) = \frac{1}{2}\varphi(5)\theta^o(3) + \frac{1}{2}\varphi(3)\theta^o(5) - \frac{1}{2}\varphi^o(5)\Delta\varphi(3) - \frac{1}{2}\varphi^o(3)\Delta\varphi(5) + \Delta\varphi(27),\tag{5.52p}$$

$$\begin{aligned}\varphi(c17) &= \frac{1}{2}\varphi(11)\theta^o(1) + \frac{1}{2}\varphi(13)\theta^o(2) + \frac{1}{2}\varphi(5)\theta^o(3) + \frac{1}{2}\varphi(9)\theta^o(4) - \frac{1}{2}\varphi^o(3)\Delta\varphi(5) \\ &\quad - \frac{1}{2}\varphi^o(4)\Delta\varphi(9) - \frac{1}{2}\varphi^o(1)\Delta\varphi(11) - \frac{1}{2}\varphi^o(2)\Delta\varphi(13) + \Delta\varphi(15),\end{aligned}\tag{5.52q}$$

$$\begin{aligned}\varphi(c18) &= \frac{1}{2}\varphi(14)\theta^o(1) + \frac{1}{2}\varphi(12)\theta^o(2) + \frac{1}{2}\varphi(9)\theta^o(3) + \frac{1}{2}\varphi(6)\theta^o(4) - \frac{1}{2}\varphi^o(4)\Delta\varphi(6) \\ &\quad - \frac{1}{2}\varphi^o(3)\Delta\varphi(9) - \frac{1}{2}\varphi^o(2)\Delta\varphi(12) - \frac{1}{2}\varphi^o(1)\Delta\varphi(14) + \Delta\varphi(16),\end{aligned}\tag{5.52r}$$

$$\begin{aligned}\varphi(c19) &= \frac{1}{2}\varphi(7)\theta^o(1) + \frac{1}{2}\varphi(10)\theta^o(2) + \frac{1}{2}\varphi(11)\theta^o(3) + \frac{1}{2}\varphi(14)\theta^o(4) - \frac{1}{2}\varphi^o(1)\Delta\varphi(7) \\ &\quad - \frac{1}{2}\varphi^o(2)\Delta\varphi(10) - \frac{1}{2}\varphi^o(3)\Delta\varphi(11) - \frac{1}{2}\varphi^o(4)\Delta\varphi(14) + \Delta\varphi(17),\end{aligned}\tag{5.52s}$$

$$\begin{aligned}\varphi(c20) &= \frac{1}{2}\varphi(10)\theta^o(1) + \frac{1}{2}\varphi(8)\theta^o(2) + \frac{1}{2}\varphi(13)\theta^o(3) + \frac{1}{2}\varphi(12)\theta^o(4) - \frac{1}{2}\varphi^o(2)\Delta\varphi(8) \\ &\quad - \frac{1}{2}\varphi^o(1)\Delta\varphi(10) - \frac{1}{2}\varphi^o(4)\Delta\varphi(12) - \frac{1}{2}\varphi^o(3)\Delta\varphi(13) + \Delta\varphi(18),\end{aligned}\tag{5.52t}$$

$$\varphi(c21) = \frac{1}{2}\varphi(17)\theta^o(1) + \frac{1}{2}\varphi(18)\theta^o(2) + \frac{1}{2}\varphi(15)\theta^o(3) + \frac{1}{2}\varphi(16)\theta^o(4) - \frac{1}{2}\varphi^o(3)\Delta\varphi(15)$$

$$-\frac{1}{2}\varphi^o(4)\Delta\varphi(16) - \frac{1}{2}\varphi^o(1)\Delta\varphi(17) - \frac{1}{2}\varphi^o(2)\Delta\varphi(18) + \Delta\varphi(19), \quad (5.52u)$$

$$\begin{aligned} \varphi(c22) &= \frac{1}{2}\varphi(7)\theta^o(1) + \frac{1}{2}\varphi(11)\theta^o(3) + \varphi(13)\theta^o(4) - \frac{1}{2}\varphi^o(1)\Delta\varphi(7) - \frac{1}{2}\varphi^o(3)\Delta\varphi(11) \\ &\quad - \varphi^o(4)\Delta\varphi(13) + \Delta\varphi(20), \end{aligned} \quad (5.52v)$$

$$\begin{aligned} \varphi(c23) &= \frac{1}{2}\varphi(8)\theta^o(2) + \varphi(14)\theta^o(3) + \frac{1}{2}\varphi(12)\theta^o(4) - \frac{1}{2}\varphi^o(2)\Delta\varphi(8) - \frac{1}{2}\varphi^o(4)\Delta\varphi(12) \\ &\quad - \varphi^o(3)\Delta\varphi(14) + \Delta\varphi(21), \end{aligned} \quad (5.52w)$$

$$\varphi(c24) = \varphi(7)\theta^o(3) + \varphi(10)\theta^o(4) - \varphi^o(3)\Delta\varphi(7) - \varphi^o(4)\Delta\varphi(10) + \Delta\varphi(22), \quad (5.52x)$$

$$\varphi(c25) = \varphi(10)\theta^o(3) + \varphi(8)\theta^o(4) - \varphi^o(4)\Delta\varphi(8) - \varphi^o(3)\Delta\varphi(10) + \Delta\varphi(23), \quad (5.52y)$$

$$\begin{aligned} \varphi(c26) &= \frac{1}{2}\varphi(22)\theta^o(1) + \frac{1}{2}\varphi(23)\theta^o(2) + \frac{1}{2}\varphi(17)\theta^o(3) + \frac{1}{2}\varphi(18)\theta^o(4) - \frac{1}{2}\varphi^o(3)\Delta\varphi(17) \\ &\quad - \frac{1}{2}\varphi^o(4)\Delta\varphi(18) - \frac{1}{2}\varphi^o(1)\Delta\varphi(22) - \frac{1}{2}\varphi^o(2)\Delta\varphi(23) + \Delta\varphi(24), \end{aligned} \quad (5.52z)$$

$$\varphi(c27) = \varphi(22)\theta^o(3) + \varphi(23)\theta^o(4) - \varphi^o(3)\Delta\varphi(22) - \varphi^o(4)\Delta\varphi(23) + \Delta\varphi(25), \quad (5.52aa)$$

$$\begin{aligned} \varphi(c28) &= \frac{1}{2}\varphi(19)\theta^o(1) + \frac{1}{2}\varphi(22)\theta^o(2) + \frac{1}{2}\varphi(23)\theta^o(3) + \frac{1}{2}\varphi(7)\theta^o(4) + \frac{1}{2}\varphi(13)\theta^o(5) \\ &\quad + \frac{1}{2}\varphi(14)\theta^o(6) - \frac{1}{2}\varphi^o(4)\Delta\varphi(7) - \frac{1}{2}\varphi^o(5)\Delta\varphi(13) - \frac{1}{2}\varphi^o(6)\Delta\varphi(14) - \frac{1}{2}\varphi^o(1)\Delta\varphi(19) \\ &\quad - \frac{1}{2}\varphi^o(2)\Delta\varphi(22) - \frac{1}{2}\varphi^o(3)\Delta\varphi(23) + \Delta\varphi(28), \end{aligned} \quad (5.52ab)$$

$$\begin{aligned} \varphi(c29) &= \frac{1}{2}\varphi(24)\theta^o(1) + \frac{1}{2}\varphi(20)\theta^o(2) + \frac{1}{2}\varphi(25)\theta^o(3) + \frac{1}{2}\varphi(13)\theta^o(4) + \frac{1}{2}\varphi(8)\theta^o(5) \\ &\quad + \frac{1}{2}\varphi(15)\theta^o(6) - \frac{1}{2}\varphi^o(5)\Delta\varphi(8) - \frac{1}{2}\varphi^o(4)\Delta\varphi(13) - \frac{1}{2}\varphi^o(6)\Delta\varphi(15) - \frac{1}{2}\varphi^o(2)\Delta\varphi(20) \\ &\quad - \frac{1}{2}\varphi^o(1)\Delta\varphi(24) - \frac{1}{2}\varphi^o(3)\Delta\varphi(25) + \Delta\varphi(29), \end{aligned} \quad (5.52ac)$$

$$\begin{aligned} \varphi(c30) &= \frac{1}{2}\varphi(26)\theta^o(1) + \frac{1}{2}\varphi(27)\theta^o(2) + \frac{1}{2}\varphi(21)\theta^o(3) + \frac{1}{2}\varphi(14)\theta^o(4) + \frac{1}{2}\varphi(15)\theta^o(5) \\ &\quad + \frac{1}{2}\varphi(9)\theta^o(6) - \frac{1}{2}\varphi^o(6)\Delta\varphi(9) - \frac{1}{2}\varphi^o(4)\Delta\varphi(14) - \frac{1}{2}\varphi^o(5)\Delta\varphi(15) - \frac{1}{2}\varphi^o(3)\Delta\varphi(21) \\ &\quad - \frac{1}{2}\varphi^o(1)\Delta\varphi(26) - \frac{1}{2}\varphi^o(2)\Delta\varphi(27) + \Delta\varphi(30), \end{aligned} \quad (5.52ad)$$

$$\begin{aligned} \varphi(c31) &= \frac{1}{2}\varphi(10)\theta^o(1) + \frac{1}{2}\varphi(16)\theta^o(2) + \frac{1}{2}\varphi(17)\theta^o(3) + \frac{1}{2}\varphi(19)\theta^o(4) + \frac{1}{2}\varphi(24)\theta^o(5) \\ &\quad + \frac{1}{2}\varphi(26)\theta^o(6) - \frac{1}{2}\varphi^o(1)\Delta\varphi(10) - \frac{1}{2}\varphi^o(2)\Delta\varphi(16) - \frac{1}{2}\varphi^o(3)\Delta\varphi(17) - \frac{1}{2}\varphi^o(4)\Delta\varphi(19) \\ &\quad - \frac{1}{2}\varphi^o(5)\Delta\varphi(24) - \frac{1}{2}\varphi^o(6)\Delta\varphi(26) + \Delta\varphi(31), \end{aligned} \quad (5.52ae)$$

$$\begin{aligned}
\varphi(c32) &= \frac{1}{2}\varphi(16)\theta^o(1) + \frac{1}{2}\varphi(11)\theta^o(2) + \frac{1}{2}\varphi(18)\theta^o(3) + \frac{1}{2}\varphi(22)\theta^o(4) + \frac{1}{2}\varphi(20)\theta^o(5) \\
&+ \frac{1}{2}\varphi(27)\theta^o(6) - \frac{1}{2}\varphi^o(2)\Delta\varphi(11) - \frac{1}{2}\varphi^o(1)\Delta\varphi(16) - \frac{1}{2}\varphi^o(3)\Delta\varphi(18) - \frac{1}{2}\varphi^o(5)\Delta\varphi(20) \\
&- \frac{1}{2}\varphi^o(4)\Delta\varphi(22) - \frac{1}{2}\varphi^o(6)\Delta\varphi(27) + \Delta\varphi(32), \tag{5.52af}
\end{aligned}$$

$$\begin{aligned}
\varphi(c33) &= \frac{1}{2}\varphi(17)\theta^o(1) + \frac{1}{2}\varphi(18)\theta^o(2) + \frac{1}{2}\varphi(12)\theta^o(3) + \frac{1}{2}\varphi(23)\theta^o(4) + \frac{1}{2}\varphi(25)\theta^o(5) \\
&+ \frac{1}{2}\varphi(21)\theta^o(6) - \frac{1}{2}\varphi^o(3)\Delta\varphi(12) - \frac{1}{2}\varphi^o(1)\Delta\varphi(17) - \frac{1}{2}\varphi^o(2)\Delta\varphi(18) - \frac{1}{2}\varphi^o(6)\Delta\varphi(21) \\
&- \frac{1}{2}\varphi^o(4)\Delta\varphi(23) - \frac{1}{2}\varphi^o(5)\Delta\varphi(25) + \Delta\varphi(33), \tag{5.52ag}
\end{aligned}$$

$$\begin{aligned}
\varphi(c34) &= \frac{1}{2}\varphi(31)\theta^o(1) + \frac{1}{2}\varphi(32)\theta^o(2) + \frac{1}{2}\varphi(33)\theta^o(3) + \frac{1}{2}\varphi(28)\theta^o(4) + \frac{1}{2}\varphi(29)\theta^o(5) \\
&+ \frac{1}{2}\varphi(30)\theta^o(6) - \frac{1}{2}\varphi^o(4)\Delta\varphi(28) - \frac{1}{2}\varphi^o(5)\Delta\varphi(29) - \frac{1}{2}\varphi^o(6)\Delta\varphi(30) - \frac{1}{2}\varphi^o(1)\Delta\varphi(31) \\
&- \frac{1}{2}\varphi^o(2)\Delta\varphi(32) - \frac{1}{2}\varphi^o(3)\Delta\varphi(33) + \Delta\varphi(34), \tag{5.52ah}
\end{aligned}$$

$$\begin{aligned}
\varphi(c35) &= \frac{3}{4}\varphi(10)\theta^o(1) + \frac{1}{4}\varphi(19)\theta^o(4) + \varphi(22)\theta^o(5) + \varphi(23)\theta^o(6) - \frac{3}{4}\varphi^o(1)\Delta\varphi(10) \\
&- \frac{1}{4}\varphi^o(4)\Delta\varphi(19) - \varphi^o(5)\Delta\varphi(22) - \varphi^o(6)\Delta\varphi(23) + \Delta\varphi(35), \tag{5.52ai}
\end{aligned}$$

$$\begin{aligned}
\varphi(c36) &= \frac{3}{4}\varphi(11)\theta^o(2) + \varphi(24)\theta^o(4) + \frac{1}{4}\varphi(20)\theta^o(5) + \varphi(25)\theta^o(6) - \frac{3}{4}\varphi^o(2)\Delta\varphi(11) \\
&- \frac{1}{4}\varphi^o(5)\Delta\varphi(20) - \varphi^o(4)\Delta\varphi(24) - \varphi^o(6)\Delta\varphi(25) + \Delta\varphi(36), \tag{5.52aj}
\end{aligned}$$

$$\begin{aligned}
\varphi(c37) &= \frac{3}{4}\varphi(12)\theta^o(3) + \varphi(26)\theta^o(4) + \varphi(27)\theta^o(5) + \frac{1}{4}\varphi(21)\theta^o(6) - \frac{3}{4}\varphi^o(3)\Delta\varphi(12) \\
&- \frac{1}{4}\varphi^o(6)\Delta\varphi(21) - \varphi^o(4)\Delta\varphi(26) - \varphi^o(5)\Delta\varphi(27) + \Delta\varphi(37), \tag{5.52ak}
\end{aligned}$$

$$\begin{aligned}
\varphi(c38) &= \varphi(10)\theta^o(4) + \varphi(16)\theta^o(5) + \varphi(17)\theta^o(6) - \varphi^o(4)\Delta\varphi(10) - \varphi^o(5)\Delta\varphi(16) \\
&- \varphi^o(6)\Delta\varphi(17) + \Delta\varphi(38), \tag{5.52al}
\end{aligned}$$

$$\begin{aligned}
\varphi(c39) &= \varphi(16)\theta^o(4) + \varphi(11)\theta^o(5) + \varphi(18)\theta^o(6) - \varphi^o(5)\Delta\varphi(11) - \varphi^o(4)\Delta\varphi(16) \\
&- \varphi^o(6)\Delta\varphi(18) + \Delta\varphi(39), \tag{5.52am}
\end{aligned}$$

$$\begin{aligned}
\varphi(c40) &= \varphi(17)\theta^o(4) + \varphi(18)\theta^o(5) + \varphi(12)\theta^o(6) - \varphi^o(6)\Delta\varphi(12) - \varphi^o(4)\Delta\varphi(17) \\
&- \varphi^o(5)\Delta\varphi(18) + \Delta\varphi(40), \tag{5.52an}
\end{aligned}$$

$$\varphi(c41) = \frac{1}{2}\varphi(38)\theta^o(1) + \frac{1}{2}\varphi(39)\theta^o(2) + \frac{1}{2}\varphi(40)\theta^o(3) + \frac{1}{2}\varphi(31)\theta^o(4) + \frac{1}{2}\varphi(32)\theta^o(5)$$

$$\begin{aligned}
& + \frac{1}{2}\varphi(33)\theta^o(6) - \frac{1}{2}\varphi^o(4)\Delta\varphi(31) - \frac{1}{2}\varphi^o(5)\Delta\varphi(32) - \frac{1}{2}\varphi^o(6)\Delta\varphi(33) - \frac{1}{2}\varphi^o(1)\Delta\varphi(38) \\
& - \frac{1}{2}\varphi^o(2)\Delta\varphi(39) - \frac{1}{2}\varphi^o(3)\Delta\varphi(40) + \Delta\varphi(41), \tag{5.52ao}
\end{aligned}$$

$$\begin{aligned}
\varphi(c42) & = \varphi(38)\theta^o(4) + \varphi(39)\theta^o(5) + \varphi(40)\theta^o(6) - \varphi^o(4)\Delta\varphi(38) - \varphi^o(5)\Delta\varphi(39) \\
& - \varphi^o(6)\Delta\varphi(40) + \Delta\varphi(42). \tag{5.52ap}
\end{aligned}$$

The nonlinear transformation in matrix form, after dropping the second and higher order error terms, is

$$\tilde{\mathbf{B}}\Delta\varphi \simeq \varphi - \tilde{\mathbf{A}}\boldsymbol{\theta}^o. \tag{5.53}$$

The matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are the key elements of this transformation and they are constructed differently for each localization case. Table 5.6 and Table 5.7 define the $\tilde{\mathbf{A}}$ matrix for the four different cases. The $\tilde{\mathbf{B}}$ matrix is given by

$$\tilde{\mathbf{B}} = \mathbf{I} - \mathbf{C}, \tag{5.54}$$

where \mathbf{C} is sparse matrix with non-zero elements defined in Table 5.8 for the 2-D localization and in Table 5.9 for the 3-D localization. The WLS solution for $\tilde{\boldsymbol{\varphi}}^o$ is

$$\boldsymbol{\theta} = (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \boldsymbol{\varphi}. \tag{5.55}$$

$\tilde{\mathbf{W}}$ is set as

$$\tilde{\mathbf{W}} = \tilde{\mathbf{B}}^{-T} (\mathbf{A}^T \mathbf{W} \mathbf{A}) \tilde{\mathbf{B}}^{-1}, \tag{5.56}$$

which is an approximation of $E[\tilde{\mathbf{B}}\Delta\varphi\Delta\varphi^T\tilde{\mathbf{B}}^T]^{-1}$ where (5.49) has been used. The

Table 5.5: The indices of the data vector for the four different cases corresponding to each code index

Code	2-D Single-Time	2-D Multiple-Time	3-D Single-Time	3-D Multiple-Time
c1	-	-	9	9
c2	-	-	12	12
c3	-	-	14	14
c4	-	-	15	15
c5	10	10	16	16
c6	-	-	17	17
c7	-	-	18	18
c8	11	11	19	19
c9	12	12	20	20
c10	-	-	21	21
c11	13	13	22	22
c12	-	-	23	23
c13	14	14	24	24
c14	-	-	25	25
c15	-	-	26	26
c16	-	-	27	27
c17	15	15	-	-
c18	16	16	-	-
c19	17	17	-	-
c20	18	18	-	-
c21	19	19	-	-
c22	-	20	-	-
c23	-	21	-	-
c24	-	22	-	-
c25	-	23	-	-
c26	-	24	-	-
c27	-	25	-	-
c28	-	-	28	28
c29	-	-	29	29
c30	-	-	30	30
c31	-	-	31	31
c32	-	-	32	32
c33	-	-	33	33
c34	-	-	34	34
c35	-	-	-	35
c36	-	-	-	36
c37	-	-	-	37
c38	-	-	-	38
c39	-	-	-	39
c40	-	-	-	40
c41	-	-	-	41
c42	-	-	-	42

Table 5.6: The entries of $\tilde{\mathbf{A}}_{19 \times 4}$ and $\tilde{\mathbf{A}}_{34 \times 6}$ matrices for the 2-D Single-Time and 3-D Single-Time cases respectively.

Row	2-D Single-Time				3-D Single-Time					
	Col 1	Col 2	Col 3	Col 4	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
-	1	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0	1	0	0
4	$\varphi(1)$	0	0	0	0	0	0	0	1	0
5	0	$\varphi(2)$	0	0	0	0	0	0	0	1
6	0	0	$\varphi(3)$	0	$\varphi(1)$	0	0	0	0	0
7	0	0	0	$\varphi(4)$	0	$\varphi(2)$	0	0	0	0
8	$.5\varphi(2)$	$.5\varphi(1)$	0	0	0	0	$\varphi(3)$	0	0	0
9	0	0	$.5\varphi(4)$	$.5\varphi(3)$	0	0	0	$\varphi(4)$	0	0
10	$.5\varphi(3)$	0	$.5\varphi(1)$	0	0	0	0	0	$\varphi(5)$	0
11	0	$.5\varphi(4)$	0	$.5\varphi(2)$	0	0	0	0	0	$\varphi(6)$
12	$.5\varphi(4)$	0	0	$.5\varphi(1)$	$.5\varphi(2)$	$.5\varphi(1)$	0	0	0	0
13	0	$.5\varphi(3)$	$.5\varphi(2)$	0	$.5\varphi(3)$	0	$.5\varphi(1)$	0	0	0
14	$.5\varphi(11)$	$.5\varphi(13)$	$.5\varphi(5)$	$.5\varphi(9)$	0	$.5\varphi(3)$	$.5\varphi(2)$	0	0	0
15	$.5\varphi(14)$	$.5\varphi(12)$	$.5\varphi(9)$	$.5\varphi(6)$	0	0	0	$.5\varphi(5)$	$.5\varphi(4)$	0
16	$.5\varphi(7)$	$.5\varphi(10)$	$.5\varphi(11)$	$.5\varphi(14)$	0	0	0	$.5\varphi(6)$	0	$.5\varphi(4)$
17	$.5\varphi(10)$	$.5\varphi(8)$	$.5\varphi(13)$	$.5\varphi(12)$	0	0	0	0	$.5\varphi(6)$	$.5\varphi(5)$
18	$.5\varphi(17)$	$.5\varphi(18)$	$.5\varphi(15)$	$.5\varphi(16)$	$.5\varphi(4)$	0	0	$.5\varphi(1)$	0	0
19	-	-	-	-	0	$.5\varphi(5)$	0	0	$.5\varphi(2)$	0
20	-	-	-	-	0	0	$.5\varphi(6)$	0	0	$.5\varphi(3)$
21	-	-	-	-	$.5\varphi(5)$	0	0	0	$.5\varphi(1)$	0
22	-	-	-	-	$.5\varphi(6)$	0	0	0	0	$.5\varphi(1)$
23	-	-	-	-	0	$.5\varphi(4)$	0	$.5\varphi(2)$	0	0
24	-	-	-	-	0	$.5\varphi(6)$	0	0	0	$.5\varphi(2)$
25	-	-	-	-	0	0	$.5\varphi(4)$	$.5\varphi(3)$	0	0
26	-	-	-	-	0	0	$.5\varphi(5)$	0	$.5\varphi(3)$	0
27	-	-	-	-	$.5\varphi(19)$	$.5\varphi(22)$	$.5\varphi(23)$	$.5\varphi(7)$	$.5\varphi(13)$	$.5\varphi(14)$
28	-	-	-	-	$.5\varphi(24)$	$.5\varphi(20)$	$.5\varphi(25)$	$.5\varphi(13)$	$.5\varphi(8)$	$.5\varphi(15)$
29	-	-	-	-	$.5\varphi(26)$	$.5\varphi(27)$	$.5\varphi(21)$	$.5\varphi(14)$	$.5\varphi(15)$	$.5\varphi(9)$
30	-	-	-	-	$.5\varphi(10)$	$.5\varphi(16)$	$.5\varphi(17)$	$.5\varphi(19)$	$.5\varphi(24)$	$.5\varphi(26)$
31	-	-	-	-	$.5\varphi(16)$	$.5\varphi(11)$	$.5\varphi(18)$	$.5\varphi(22)$	$.5\varphi(20)$	$.5\varphi(27)$
32	-	-	-	-	$.5\varphi(17)$	$.5\varphi(18)$	$.5\varphi(12)$	$.5\varphi(23)$	$.5\varphi(25)$	$.5\varphi(21)$
33	-	-	-	-	$.5\varphi(31)$	$.5\varphi(32)$	$.5\varphi(33)$	$.5\varphi(28)$	$.5\varphi(29)$	$.5\varphi(30)$
34	-	-	-	-						

Table 5.7: The entries of $\tilde{\mathbf{A}}_{25 \times 4}$ and $\tilde{\mathbf{A}}_{42 \times 6}$ matrices for the 2-D Multiple-Time and 3-D Multiple-Time cases respectively.

Row	2-D Multiple-Time				3-D Multiple-Time					
	Col 1	Col 2	Col 3	Col 4	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
-	1	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0
3	0	0	0	1	0	0	0	1	0	0
4	$\varphi(1)$	0	0	0	0	0	0	0	1	0
5	0	$\varphi(2)$	0	0	0	0	0	0	0	1
6	0	0	$\varphi(3)$	0	$\varphi(1)$	0	0	0	0	0
7	0	0	0	$\varphi(4)$	0	$\varphi(2)$	0	0	0	0
8	$.5\varphi(2)$	$.5\varphi(1)$	0	0	0	0	$\varphi(3)$	0	0	0
9	0	0	$.5\varphi(4)$	$.5\varphi(3)$	0	0	0	$\varphi(4)$	0	0
10	$.5\varphi(3)$	0	$.5\varphi(1)$	0	0	0	0	0	$\varphi(5)$	0
11	0	$.5\varphi(4)$	0	$.5\varphi(2)$	0	0	0	0	0	$\varphi(6)$
12	$.5\varphi(4)$	0	0	$.5\varphi(1)$	$.5\varphi(2)$	$.5\varphi(1)$	0	0	0	0
13	0	$.5\varphi(3)$	$.5\varphi(2)$	0	$.5\varphi(3)$	0	$.5\varphi(1)$	0	0	0
14	$.5\varphi(11)$	$.5\varphi(13)$	$.5\varphi(5)$	$.5\varphi(9)$	0	$.5\varphi(3)$	$.5\varphi(2)$	0	0	0
15	$.5\varphi(14)$	$.5\varphi(12)$	$.5\varphi(9)$	$.5\varphi(6)$	0	0	0	$.5\varphi(5)$	$.5\varphi(4)$	0
16	$.5\varphi(7)$	$.5\varphi(10)$	$.5\varphi(11)$	$.5\varphi(14)$	0	0	0	$.5\varphi(6)$	0	$.5\varphi(4)$
17	$.5\varphi(10)$	$.5\varphi(8)$	$.5\varphi(13)$	$.5\varphi(12)$	0	0	0	0	$.5\varphi(6)$	$.5\varphi(5)$
18	$.5\varphi(17)$	$.5\varphi(18)$	$.5\varphi(15)$	$.5\varphi(16)$	$.5\varphi(4)$	0	0	$.5\varphi(1)$	0	0
19	$.5\varphi(7)$	0	$.5\varphi(11)$	$\varphi(23)$	0	$.5\varphi(5)$	0	0	$.5\varphi(2)$	0
20	0	$.5\varphi(8)$	$\varphi(14)$	$.5\varphi(12)$	0	0	$.5\varphi(6)$	0	0	$.5\varphi(3)$
21	0	0	$\varphi(7)$	$\varphi(10)$	$.5\varphi(5)$	0	0	0	$.5\varphi(1)$	0
22	0	0	$\varphi(7)$	$\varphi(8)$	$.5\varphi(6)$	0	0	0	0	$.5\varphi(1)$
23	$.5\varphi(22)$	$.5\varphi(23)$	$.5\varphi(17)$	$.5\varphi(18)$	0	$.5\varphi(4)$	0	$.5\varphi(2)$	0	0
24	0	0	$\varphi(22)$	$\varphi(23)$	0	$.5\varphi(6)$	0	0	0	$.5\varphi(2)$
25	-	-	-	-	0	0	$.5\varphi(4)$	$.5\varphi(3)$	0	0
26	-	-	-	-	0	0	$.5\varphi(5)$	0	$.5\varphi(3)$	0
27	-	-	-	-	$.5\varphi(19)$	$.5\varphi(22)$	$.5\varphi(23)$	$.5\varphi(7)$	$.5\varphi(13)$	$.5\varphi(14)$
28	-	-	-	-	$.5\varphi(24)$	$.5\varphi(20)$	$.5\varphi(25)$	$.5\varphi(13)$	$.5\varphi(8)$	$.5\varphi(15)$
29	-	-	-	-	$.5\varphi(26)$	$.5\varphi(27)$	$.5\varphi(21)$	$.5\varphi(14)$	$.5\varphi(15)$	$.5\varphi(9)$
30	-	-	-	-	$.5\varphi(10)$	$.5\varphi(16)$	$.5\varphi(17)$	$.5\varphi(19)$	$.5\varphi(24)$	$.5\varphi(26)$
31	-	-	-	-	$.5\varphi(16)$	$.5\varphi(11)$	$.5\varphi(18)$	$.5\varphi(22)$	$.5\varphi(20)$	$.5\varphi(27)$
32	-	-	-	-	$.5\varphi(17)$	$.5\varphi(18)$	$.5\varphi(12)$	$.5\varphi(23)$	$.5\varphi(25)$	$.5\varphi(21)$
33	-	-	-	-	$.5\varphi(31)$	$.5\varphi(32)$	$.5\varphi(33)$	$.5\varphi(28)$	$.5\varphi(29)$	$.5\varphi(30)$
34	-	-	-	-	$.75\varphi(10)$	0	0	$.25\varphi(19)$	$\varphi(22)$	$\varphi(23)$
35	-	-	-	-	0	$.75\varphi(11)$	0	$\varphi(24)$	$.25\varphi(20)$	$\varphi(25)$
36	-	-	-	-	0	0	$.75\varphi(12)$	$\varphi(26)$	$\varphi(27)$	$.25\varphi(21)$
37	-	-	-	-	0	0	0	$\varphi(10)$	$\varphi(16)$	$\varphi(17)$
38	-	-	-	-	0	0	0	$\varphi(16)$	$\varphi(11)$	$\varphi(18)$
39	-	-	-	-	0	0	0	$\varphi(17)$	$\varphi(18)$	$\varphi(12)$
40	-	-	-	-	$.5\varphi(38)$	$.5\varphi(39)$	$.5\varphi(40)$	$.5\varphi(31)$	$.5\varphi(32)$	$.5\varphi(33)$
41	-	-	-	-	0	0	0	$\varphi(38)$	$\varphi(39)$	$\varphi(40)$
42	-	-	-	-	-	-	-	-	-	-

Table 5.8: The non-zero entries of $\mathbf{C}_{19 \times 19}$ and $\mathbf{C}_{25 \times 25}$ matrices for the 2-D Single-Time and 2-D Multiple-Time cases respectively.

Row	2-D Single-Time \ 2-D Multiple-Time				Row	2-D Multiple-Time			
-	Col 1	Col 2	Col 3	Col 4	-	Col 7	Col 10	Col 11	Col 13
5	$\varphi^o(1)$	0	0	0	20	$.5\varphi^o(1)$	0	$.5\varphi^o(3)$	$\varphi^o(4)$
6	0	$\varphi^o(2)$	0	0	22	$\varphi^o(3)$	$\varphi^o(4)$	0	0
7	0	0	$\varphi^o(3)$	0	-	-	-	-	-
8	0	0	0	$\varphi^o(4)$	-	-	-	-	-
9	$.5\varphi^o(2)$	$.5\varphi^o(1)$	0	0	-	-	-	-	-
10	0	0	$.5\varphi^o(4)$	$.5\varphi^o(3)$	-	-	-	-	-
11	$.5\varphi^o(3)$	0	$.5\varphi^o(1)$	0	-	-	-	-	-
12	0	$.5\varphi^o(4)$	0	$.5\varphi^o(2)$	-	-	-	-	-
13	$.5\varphi^o(4)$	0	0	$.5\varphi^o(1)$	-	-	-	-	-
14	0	$.5\varphi^o(3)$	$.5\varphi^o(2)$	0	-	-	-	-	-
-	Col 5	Col 9	Col 11	Col 13	-	Col 8	Col 10	Col 12	Col 14
15	$.5\varphi^o(3)$	$.5\varphi^o(4)$	$.5\varphi^o(1)$	$.5\varphi^o(2)$	21	$.5\varphi^o(2)$	0	$.5\varphi^o(4)$	$\varphi^o(3)$
-	-	-	-	-	23	$\varphi^o(4)$	$\varphi^o(3)$	0	0
-	Col 6	Col 9	Col 12	Col 14	-	Col 17	Col 18	Col 22	Col 23
16	$.5\varphi^o(4)$	$.5\varphi^o(3)$	$.5\varphi^o(2)$	$.5\varphi^o(1)$	24	$.5\varphi^o(3)$	$.5\varphi^o(4)$	$.5\varphi^o(1)$	$.5\varphi^o(2)$
-	-	-	-	-	25	0	0	$\varphi^o(3)$	$\varphi^o(4)$
-	Col 7	Col 10	Col 11	Col 14	-	-	-	-	-
17	$.5\varphi^o(1)$	$.5\varphi^o(4)$	$.5\varphi^o(3)$	$.5\varphi^o(4)$	-	-	-	-	-
-	Col 8	Col 10	Col 12	Col 13	-	-	-	-	-
18	$.5\varphi^o(2)$	$.5\varphi^o(1)$	$.5\varphi^o(4)$	$.5\varphi^o(3)$	-	-	-	-	-
-	Col 15	Col 16	Col 17	Col 18	-	-	-	-	-
19	$.5\varphi^o(3)$	$.5\varphi^o(4)$	$.5\varphi^o(1)$	$.5\varphi^o(2)$	-	-	-	-	-

Table 5.9: The non-zero entries of $\mathbf{C}_{34 \times 34}$ and $\mathbf{C}_{42 \times 42}$ matrices for the 3-D Single-Time and 3-D Multiple-Time cases respectively.

Row	3-D Single-Time \ 3-D Multiple-Time						Row	3-D Multiple-Time			
-	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	-	Col 10	Col 19	Col 22	Col 23
7	$\varphi(1)$	0	0	0	0	0	35	$.75\varphi(1)$	$.25\varphi(4)$	$.5\varphi(5)$	$.5\varphi(6)$
8	0	$\varphi(2)$	0	0	0	0	-	-	-	-	-
9	0	0	$\varphi(3)$	0	0	0	-	-	-	-	-
10	0	0	0	$\varphi(4)$	0	0	-	-	-	-	-
11	0	0	0	0	$\varphi(5)$	0	-	-	-	-	-
12	0	0	0	0	0	$\varphi(6)$	-	-	-	-	-
13	$.5\varphi(2)$	$.5\varphi(1)$	0	0	0	0	-	-	-	-	-
14	$.5\varphi(3)$	0	$.5\varphi(1)$	0	0	0	-	-	-	-	-
15	0	$.5\varphi(3)$	$.5\varphi(2)$	0	0	0	-	-	-	-	-
16	0	0	0	$.5\varphi(5)$	$.5\varphi(4)$	0	-	-	-	-	-
17	0	0	0	$.5\varphi(6)$	0	$.5\varphi(4)$	-	-	-	-	-
18	0	0	0	0	$.5\varphi(6)$	$.5\varphi(5)$	-	-	-	-	-
19	$.5\varphi(4)$	0	0	$.5\varphi(1)$	0	0	-	-	-	-	-
20	0	$.5\varphi(5)$	0	0	$.5\varphi(2)$	0	-	-	-	-	-
21	0	0	$.5\varphi(6)$	0	0	$.5\varphi(3)$	-	-	-	-	-
22	$.5\varphi(5)$	0	0	0	$.5\varphi(1)$		-	-	-	-	-
23	$.5\varphi(6)$	0	0	0	0	$.5\varphi(1)$	-	-	-	-	-
24	0	$.5\varphi(4)$	0	$.5\varphi(2)$	0		-	-	-	-	-
25	0	$.5\varphi(6)$	0		0	$.5\varphi(2)$	-	-	-	-	-
26	0	0	$.5\varphi(4)$	$.5\varphi(3)$	0	0	-	-	-	-	-
27	0	0	$.5\varphi(5)$	0	$.5\varphi(3)$	0	-	-	-	-	-
-	Col 7	Col 8	Col 9	Col 13	Col 14	Col 15	-	Col 11	Col 20	Col 24	Col 25
28	$.5\varphi(4)$	0	0	$.5\varphi(5)$	$.5\varphi(6)$	0	36	$.75\varphi(2)$	$.25\varphi(5)$	$.5\varphi(4)$	$.5\varphi(6)$
29	0	$.5\varphi(5)$	0	$.5\varphi(4)$	0	$.5\varphi(6)$	-	-	-	-	-
30	0	0	$.5\varphi(6)$	0	$.5\varphi(4)$	$.5\varphi(5)$	-	-	-	-	-
-	Col 10	Col 11	Col 12	Col 16	Col 17	Col 18	-	Col 12	Col 21	Col 26	Col 27
31	$.5\varphi(1)$	0	0	$.5\varphi(2)$	$.5\varphi(3)$	0	37	$.75\varphi(3)$	$.25\varphi(6)$	$.5\varphi(4)$	$.5\varphi(5)$
32	0	$.5\varphi(2)$	0	$.5\varphi(1)$	0	$.5\varphi(3)$	-	-	-	-	-
33	0	0	$.5\varphi(3)$	0	$.5\varphi(1)$	$.5\varphi(2)$	-	-	-	-	-
-	Col 19	Col 20	Col 22	Col 23	Col 24	Col 25	-	Col 10	Col 11	Col 16	Col 17
28	$.5\varphi(1)$	0	$.5\varphi(2)$	$.5\varphi(3)$	0	0	38	$\varphi(4)$	0	$\varphi(5)$	$\varphi(6)$
29	0	$.5\varphi(2)$	0	0	$.5\varphi(1)$	$.5\varphi(3)$	39	0	$\varphi(5)$	$\varphi(4)$	0
31	$.5\varphi(4)$	0	0	0	$.5\varphi(5)$	0	-	-	-	-	-
32	0	$.5\varphi(5)$	$.5\varphi(4)$	0	0	0	-	-	-	-	-
-	Col 21	Col 23	Col 25	Col 26	Col 27	Col 28	-	Col 12	Col 17	Col 18	Col 19
30	$.5\varphi(3)$	0	0	$.5\varphi(1)$	$.5\varphi(2)$	0	39	0	0	$\varphi(6)$	0
31	0	0	0	$.5\varphi(6)$	0	0	40	$\varphi(6)$	$\varphi(4)$	$\varphi(5)$	0
32	0	0	0	0	$.5\varphi(6)$	0	-	-	-	-	-
33	$.5\varphi(6)$	$.5\varphi(4)$	$.5\varphi(5)$	0	0	0	-	-	-	-	-
-	Col 28	Col 29	Col 30	Col 31	Col 32	Col 33	-	Col 31	Col 32	Col 33	Col 38
34	$.5\varphi(4)$	$.5\varphi(5)$	$.5\varphi(6)$	$.5\varphi(1)$	$.5\varphi(2)$	$.5\varphi(3)$	41	$.5\varphi(4)$	$.5\varphi(5)$	$.5\varphi(6)$	$.5\varphi(1)$
-	-	-	-	-	-	-	42	0	0	0	$\varphi(4)$
-	-	-	-	-	-	-	-	Col 39	Col 40	Col 41	Col 42
-	-	-	-	-	-	-	41	$.5\varphi(2)$	$.5\varphi(3)$	0	0
-	-	-	-	-	-	-	42	$\varphi(5)$	$\varphi(6)$	0	0

covariance matrix of $\boldsymbol{\theta}$ under the small error region can be approximated by [12]

$$\text{cov}(\boldsymbol{\theta}) \simeq (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1}, \quad (5.57)$$

where the noise in $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$ are small and can be neglected.

5.3 SDP Solution

We shall transform the optimization problem in (5.27) to a semi-definite programming (SDP) problem and solve it through SDR technique. This method is robust against noise, requires fewer sensors compared to the algebraic method in Section 5.2 and can operate in difficult sensors-object geometries. However, it is less favourable in terms of computational complexity and software/hardware design requirements. The objective function in (5.27) irrespective of the different definitions for \mathbf{h} , \mathbf{A} , $\boldsymbol{\varphi}$ and \mathbf{W} for the four localization cases can be expressed as

$$\bar{J}(\boldsymbol{\Phi}, \boldsymbol{\varphi}) = \text{tr}(\mathbf{A}^T \mathbf{W} \mathbf{A} \boldsymbol{\Phi}) - 2\mathbf{h}^T \mathbf{W} \mathbf{A} \boldsymbol{\varphi}. \quad (5.58)$$

The term $\mathbf{h}^T \mathbf{W} \mathbf{h}$ is dropped as it is independent of the unknown vector $\boldsymbol{\varphi}$, and $\boldsymbol{\Phi} = \boldsymbol{\varphi} \boldsymbol{\varphi}^T$ is a matrix that has rank equal to one. The SDP problem is constructed as a minimizer of (5.58) under the rank relaxation of $[[\boldsymbol{\Phi} \ \boldsymbol{\varphi}]^T \ [\boldsymbol{\Phi}^T \ 1]^T]$ to one and several affine constraints that are basically come from the relations among the elements of $\boldsymbol{\varphi}$ and $\boldsymbol{\Phi}$ as shown in Table 5.11-Table 5.13. The mathematical model of

Table 5.10: Constraints of the SDP Solution for each Localization Case

Localization Case	Constraints
2-D Single-Time	The corresponding constraints in Table 5.11
2-D Multiple-Time	All the constraints in Table 5.11
3-D Single-Time	All the constraints in Table 5.12 and the corresponding constraints in Table 5.13
3-D Multiple-Time	All the constraints in Table 5.12 and Table 5.13

the SDP problem is

$$\min_{\Phi, \varphi} \bar{J}(\Phi, \varphi), \quad (5.59a)$$

$$\text{s.t.} \quad \begin{bmatrix} \Phi & \varphi \\ \varphi^T & 1 \end{bmatrix} \succeq 0, \quad (5.59b)$$

The constraints corresponding to each localization case indicated in Table 5.10. (5.59c)

(5.59) can be solved using some optimization packages such as CVX [94]. The SDP problem in (5.59) is valid for all the localization cases discussed in Section 5.1 and can be used separately for each localization case when the variables \mathbf{h} , \mathbf{A} , φ and \mathbf{W} are defined accordingly. The object location estimate is extracted from the independent elements of φ after the SDP is been relaxed,

$$\boldsymbol{\theta} = \varphi(1 : 2d). \quad (5.60)$$

\mathbf{W} in (5.58) is not known due to the requirement of the true object location. Nevertheless, \mathbf{W} can be set to identity for the first solution trial and then we can use this solution to get an approximate of \mathbf{W} for the second trial.

Table 5.11: Constraints on φ and Φ for the SDP solution of the 2-D localization

2-D Single-Time\2-D Multiple-Time	2-D Multiple-Time	2-D Multiple-Time	2-D Multiple-Time
$\Phi(1, 6) = \Phi(2, 9)$	$\Phi(6, 10) = \Phi(12, 14)$	$\Phi(1, 21) = \Phi(2, 20)$	$\Phi(11, 25) = \Phi(20, 22)$
$\Phi(1, 7) = \Phi(3, 11)$	$\Phi(6, 11) = \Phi(9, 14)$	$\Phi(1, 22) = \Phi(3, 20)$	$\Phi(12, 20) = \Phi(13, 21)$
$\Phi(1, 8) = \Phi(4, 13)$	$\Phi(6, 13) = \Phi(9, 12)$	$\Phi(1, 23) = \Phi(4, 20)$	$\Phi(12, 22) = \Phi(14, 23)$
$\Phi(1, 9) = \Phi(2, 5)$	$\Phi(6, 15) = \Phi(9, 16)$	$\Phi(2, 22) = \Phi(3, 21)$	$\Phi(12, 24) = \Phi(16, 23)$
$\Phi(1, 10) = \Phi(3, 13)$	$\Phi(6, 17) = \Phi(14, 16)$	$\Phi(2, 23) = \Phi(4, 21)$	$\Phi(12, 25) = \Phi(21, 23)$
$\Phi(1, 11) = \Phi(3, 5)$	$\Phi(6, 18) = \Phi(12, 16)$	$\Phi(3, 23) = \Phi(4, 22)$	$\Phi(13, 24) = \Phi(15, 23)$
$\Phi(1, 12) = \Phi(2, 13)$	$\Phi(6, 19) = \Phi(16, 16)$	$\Phi(5, 21) = \Phi(9, 20)$	$\Phi(13, 25) = \Phi(20, 23)$
$\Phi(1, 13) = \Phi(4, 5)$	$\Phi(7, 8) = \Phi(10, 10)$	$\Phi(5, 22) = \Phi(11, 20)$	$\Phi(14, 24) = \Phi(16, 22)$
$\Phi(1, 14) = \Phi(2, 11)$	$\Phi(7, 9) = \Phi(11, 14)$	$\Phi(5, 23) = \Phi(13, 20)$	$\Phi(14, 25) = \Phi(21, 22)$
$\Phi(1, 16) = \Phi(2, 15)$	$\Phi(7, 12) = \Phi(10, 14)$	$\Phi(5, 24) = \Phi(15, 20)$	$\Phi(15, 21) = \Phi(16, 20)$
$\Phi(1, 17) = \Phi(3, 15)$	$\Phi(7, 13) = \Phi(10, 11)$	$\Phi(5, 25) = \Phi(20, 20)$	$\Phi(15, 22) = \Phi(17, 20)$
$\Phi(1, 18) = \Phi(4, 15)$	$\Phi(7, 15) = \Phi(11, 17)$	$\Phi(6, 20) = \Phi(9, 21)$	$\Phi(15, 23) = \Phi(18, 20)$
$\Phi(2, 7) = \Phi(3, 14)$	$\Phi(7, 16) = \Phi(14, 17)$	$\Phi(6, 22) = \Phi(14, 21)$	$\Phi(15, 24) = \Phi(19, 20)$
$\Phi(2, 8) = \Phi(4, 12)$	$\Phi(7, 18) = \Phi(10, 17)$	$\Phi(6, 23) = \Phi(12, 21)$	$\Phi(15, 25) = \Phi(20, 24)$
$\Phi(2, 10) = \Phi(3, 12)$	$\Phi(7, 19) = \Phi(17, 17)$	$\Phi(6, 24) = \Phi(16, 21)$	$\Phi(16, 22) = \Phi(17, 21)$
$\Phi(2, 11) = \Phi(3, 9)$	$\Phi(8, 9) = \Phi(12, 13)$	$\Phi(6, 25) = \Phi(6, 10)$	$\Phi(16, 23) = \Phi(18, 21)$
$\Phi(2, 12) = \Phi(4, 6)$	$\Phi(8, 11) = \Phi(10, 13)$	$\Phi(6, 25) = \Phi(12, 14)$	$\Phi(16, 24) = \Phi(19, 21)$
$\Phi(2, 13) = \Phi(4, 9)$	$\Phi(8, 14) = \Phi(10, 12)$	$\Phi(7, 20) = \Phi(11, 22)$	$\Phi(16, 25) = \Phi(21, 24)$
$\Phi(2, 14) = \Phi(3, 6)$	$\Phi(8, 15) = \Phi(13, 18)$	$\Phi(7, 21) = \Phi(14, 22)$	$\Phi(17, 23) = \Phi(18, 22)$
$\Phi(2, 17) = \Phi(3, 16)$	$\Phi(8, 16) = \Phi(12, 18)$	$\Phi(7, 23) = \Phi(10, 22)$	$\Phi(17, 24) = \Phi(19, 22)$
$\Phi(2, 18) = \Phi(4, 16)$	$\Phi(8, 17) = \Phi(10, 18)$	$\Phi(7, 24) = \Phi(17, 22)$	$\Phi(17, 25) = \Phi(22, 24)$
$\Phi(3, 8) = \Phi(4, 10)$	$\Phi(8, 19) = \Phi(18, 18)$	$\Phi(7, 25) = \Phi(22, 22)$	$\Phi(18, 24) = \Phi(19, 23)$
$\Phi(3, 10) = \Phi(4, 7)$	$\Phi(9, 10) = \Phi(11, 12)$	$\Phi(8, 20) = \Phi(13, 23)$	$\Phi(18, 25) = \Phi(23, 24)$
$\Phi(3, 12) = \Phi(4, 14)$	$\Phi(9, 17) = \Phi(11, 16)$	$\Phi(8, 21) = \Phi(12, 23)$	$\Phi(19, 25) = \Phi(24, 24)$
$\Phi(3, 13) = \Phi(4, 11)$	$\Phi(9, 18) = \Phi(12, 15)$	$\Phi(8, 22) = \Phi(10, 23)$	
$\Phi(3, 18) = \Phi(4, 17)$	$\Phi(9, 19) = \Phi(15, 16)$	$\Phi(8, 24) = \Phi(18, 23)$	
$\Phi(5, 6) = \Phi(9, 9)$	$\Phi(10, 15) = \Phi(11, 18)$	$\Phi(8, 25) = \Phi(23, 23)$	
$\Phi(5, 7) = \Phi(11, 11)$	$\Phi(10, 16) = \Phi(12, 17)$	$\Phi(9, 22) = \Phi(11, 21)$	
$\Phi(5, 8) = \Phi(13, 13)$	$\Phi(10, 19) = \Phi(17, 18)$	$\Phi(9, 23) = \Phi(12, 20)$	
$\Phi(5, 10) = \Phi(11, 13)$	$\Phi(11, 12) = \Phi(13, 14)$	$\Phi(9, 24) = \Phi(15, 21)$	
$\Phi(5, 12) = \Phi(9, 13)$	$\Phi(11, 16) = \Phi(14, 15)$	$\Phi(9, 25) = \Phi(20, 21)$	
$\Phi(5, 14) = \Phi(9, 11)$	$\Phi(11, 18) = \Phi(13, 17)$	$\Phi(10, 20) = \Phi(11, 23)$	
$\Phi(5, 16) = \Phi(9, 15)$	$\Phi(11, 19) = \Phi(15, 17)$	$\Phi(10, 21) = \Phi(12, 22)$	
$\Phi(5, 17) = \Phi(11, 15)$	$\Phi(12, 15) = \Phi(13, 16)$	$\Phi(10, 24) = \Phi(17, 23)$	
$\Phi(5, 18) = \Phi(13, 15)$	$\Phi(12, 17) = \Phi(14, 18)$	$\Phi(10, 25) = \Phi(22, 23)$	
$\Phi(5, 19) = \Phi(15, 15)$	$\Phi(12, 19) = \Phi(16, 18)$	$\Phi(11, 21) = \Phi(14, 20)$	
$\Phi(6, 7) = \Phi(14, 14)$	$\Phi(13, 19) = \Phi(15, 18)$	$\Phi(11, 23) = \Phi(13, 22)$	
$\Phi(6, 8) = \Phi(12, 12)$	$\Phi(14, 19) = \Phi(16, 17)$	$\Phi(11, 24) = \Phi(15, 22)$	
$\varphi(5) = \Phi(1, 1)$	$\varphi(13) = \Phi(1, 4)$	$\varphi(20) = \Phi(1, 8) + \Phi(3, 11)$	
$\varphi(6) = \Phi(2, 2)$	$\varphi(14) = \Phi(2, 3)$	$\varphi(21) = \Phi(2, 7) + \Phi(4, 12)$	
$\varphi(7) = \Phi(3, 3)$	$\varphi(15) = \Phi(1, 11) + \Phi(1, 12)$	$\varphi(22) = \Phi(3, 7) + \Phi(4, 10)$	
$\varphi(8) = \Phi(4, 4)$	$\varphi(16) = \Phi(1, 14) + \Phi(2, 12)$	$\varphi(23) = \Phi(3, 10) + \Phi(4, 8)$	
$\varphi(9) = \Phi(1, 2)$	$\varphi(17) = \Phi(1, 7) + \Phi(2, 10)$	$\varphi(24) = \Phi(3, 17) + \Phi(4, 18)$	
$\varphi(10) = \Phi(3, 4)$	$\varphi(18) = \Phi(1, 10) + \Phi(2, 8)$	$\varphi(25) = \Phi(3, 22) + \Phi(4, 23)$	
$\varphi(11) = \Phi(1, 3)$	$\varphi(19) = \Phi(1, 17) + \Phi(2, 18)$		
$\varphi(12) = \Phi(2, 4)$			

Table 5.12: Constraints on Φ for the SDP solution of the 3-D Single-Time and 3-D Multiple-Time localization

$\Phi(1, 8) = \Phi(2, 13)$	$\Phi(3, 23) = \Phi(6, 14)$	$\Phi(8, 19) = \Phi(13, 24)$	$\Phi(11, 24) = \Phi(16, 20)$	$\Phi(16, 28) = \Phi(19, 32)$
$\Phi(1, 9) = \Phi(3, 14)$	$\Phi(3, 24) = \Phi(4, 15)$	$\Phi(8, 21) = \Phi(15, 25)$	$\Phi(11, 25) = \Phi(18, 20)$	$\Phi(16, 29) = \Phi(20, 31)$
$\Phi(1, 10) = \Phi(4, 19)$	$\Phi(3, 25) = \Phi(6, 15)$	$\Phi(8, 22) = \Phi(13, 20)$	$\Phi(11, 26) = \Phi(16, 27)$	$\Phi(16, 30) = \Phi(26, 32)$
$\Phi(1, 11) = \Phi(5, 22)$	$\Phi(3, 26) = \Phi(4, 9)$	$\Phi(8, 23) = \Phi(13, 25)$	$\Phi(11, 28) = \Phi(22, 32)$	$\Phi(16, 33) = \Phi(17, 32)$
$\Phi(1, 12) = \Phi(6, 23)$	$\Phi(3, 27) = \Phi(5, 9)$	$\Phi(8, 26) = \Phi(15, 24)$	$\Phi(11, 29) = \Phi(20, 32)$	$\Phi(16, 34) = \Phi(31, 32)$
$\Phi(1, 13) = \Phi(2, 7)$	$\Phi(3, 31) = \Phi(4, 30)$	$\Phi(8, 27) = \Phi(15, 20)$	$\Phi(11, 30) = \Phi(27, 32)$	$\Phi(17, 20) = \Phi(18, 24)$
$\Phi(1, 14) = \Phi(3, 7)$	$\Phi(3, 32) = \Phi(5, 30)$	$\Phi(8, 28) = \Phi(13, 29)$	$\Phi(11, 31) = \Phi(16, 32)$	$\Phi(17, 22) = \Phi(18, 19)$
$\Phi(1, 15) = \Phi(2, 14)$	$\Phi(3, 33) = \Phi(6, 30)$	$\Phi(8, 30) = \Phi(15, 29)$	$\Phi(11, 33) = \Phi(18, 32)$	$\Phi(17, 27) = \Phi(18, 26)$
$\Phi(1, 16) = \Phi(4, 22)$	$\Phi(4, 11) = \Phi(5, 16)$	$\Phi(8, 31) = \Phi(24, 29)$	$\Phi(11, 34) = \Phi(32, 32)$	$\Phi(17, 28) = \Phi(19, 33)$
$\Phi(1, 17) = \Phi(4, 23)$	$\Phi(4, 12) = \Phi(6, 17)$	$\Phi(8, 32) = \Phi(20, 29)$	$\Phi(12, 13) = \Phi(23, 25)$	$\Phi(17, 29) = \Phi(24, 33)$
$\Phi(1, 18) = \Phi(5, 23)$	$\Phi(4, 16) = \Phi(5, 10)$	$\Phi(8, 33) = \Phi(25, 29)$	$\Phi(12, 14) = \Phi(21, 23)$	$\Phi(17, 30) = \Phi(21, 31)$
$\Phi(1, 19) = \Phi(4, 7)$	$\Phi(4, 17) = \Phi(6, 10)$	$\Phi(8, 34) = \Phi(29, 29)$	$\Phi(12, 15) = \Phi(21, 25)$	$\Phi(17, 32) = \Phi(18, 31)$
$\Phi(1, 20) = \Phi(2, 22)$	$\Phi(4, 18) = \Phi(5, 17)$	$\Phi(9, 10) = \Phi(26, 26)$	$\Phi(12, 16) = \Phi(17, 18)$	$\Phi(17, 34) = \Phi(31, 33)$
$\Phi(1, 21) = \Phi(3, 23)$	$\Phi(4, 20) = \Phi(5, 24)$	$\Phi(9, 11) = \Phi(27, 27)$	$\Phi(12, 19) = \Phi(17, 23)$	$\Phi(18, 28) = \Phi(22, 33)$
$\Phi(1, 22) = \Phi(5, 7)$	$\Phi(4, 21) = \Phi(6, 26)$	$\Phi(9, 12) = \Phi(21, 21)$	$\Phi(12, 20) = \Phi(18, 25)$	$\Phi(18, 29) = \Phi(20, 33)$
$\Phi(1, 23) = \Phi(6, 7)$	$\Phi(4, 22) = \Phi(5, 19)$	$\Phi(9, 13) = \Phi(14, 15)$	$\Phi(12, 22) = \Phi(18, 23)$	$\Phi(18, 30) = \Phi(21, 32)$
$\Phi(1, 24) = \Phi(2, 19)$	$\Phi(4, 23) = \Phi(6, 19)$	$\Phi(9, 16) = \Phi(26, 27)$	$\Phi(12, 24) = \Phi(17, 25)$	$\Phi(18, 34) = \Phi(32, 33)$
$\Phi(1, 25) = \Phi(2, 23)$	$\Phi(4, 25) = \Phi(6, 24)$	$\Phi(9, 17) = \Phi(21, 26)$	$\Phi(12, 26) = \Phi(17, 21)$	$\Phi(19, 20) = \Phi(22, 24)$
$\Phi(1, 26) = \Phi(3, 19)$	$\Phi(4, 27) = \Phi(5, 26)$	$\Phi(9, 18) = \Phi(21, 27)$	$\Phi(12, 27) = \Phi(18, 21)$	$\Phi(19, 21) = \Phi(23, 26)$
$\Phi(1, 27) = \Phi(3, 22)$	$\Phi(4, 32) = \Phi(5, 31)$	$\Phi(9, 19) = \Phi(14, 26)$	$\Phi(12, 28) = \Phi(23, 33)$	$\Phi(19, 25) = \Phi(23, 24)$
$\Phi(1, 29) = \Phi(2, 28)$	$\Phi(4, 33) = \Phi(6, 31)$	$\Phi(9, 20) = \Phi(15, 27)$	$\Phi(12, 29) = \Phi(25, 33)$	$\Phi(19, 27) = \Phi(22, 26)$
$\Phi(1, 30) = \Phi(3, 28)$	$\Phi(5, 12) = \Phi(6, 18)$	$\Phi(9, 22) = \Phi(14, 27)$	$\Phi(12, 30) = \Phi(21, 33)$	$\Phi(19, 29) = \Phi(24, 28)$
$\Phi(1, 31) = \Phi(4, 28)$	$\Phi(5, 17) = \Phi(6, 16)$	$\Phi(9, 23) = \Phi(14, 21)$	$\Phi(12, 31) = \Phi(17, 33)$	$\Phi(19, 30) = \Phi(26, 28)$
$\Phi(1, 32) = \Phi(5, 28)$	$\Phi(5, 18) = \Phi(6, 11)$	$\Phi(9, 24) = \Phi(15, 26)$	$\Phi(12, 32) = \Phi(18, 33)$	$\Phi(19, 32) = \Phi(22, 31)$
$\Phi(1, 33) = \Phi(6, 28)$	$\Phi(5, 21) = \Phi(6, 27)$	$\Phi(9, 25) = \Phi(15, 21)$	$\Phi(12, 34) = \Phi(33, 33)$	$\Phi(19, 33) = \Phi(23, 31)$
$\Phi(2, 9) = \Phi(3, 15)$	$\Phi(5, 23) = \Phi(6, 22)$	$\Phi(9, 28) = \Phi(14, 30)$	$\Phi(13, 16) = \Phi(19, 20)$	$\Phi(19, 34) = \Phi(28, 31)$
$\Phi(2, 10) = \Phi(4, 24)$	$\Phi(5, 25) = \Phi(6, 20)$	$\Phi(9, 29) = \Phi(15, 30)$	$\Phi(13, 17) = \Phi(19, 25)$	$\Phi(20, 21) = \Phi(25, 27)$
$\Phi(2, 11) = \Phi(5, 20)$	$\Phi(5, 33) = \Phi(6, 32)$	$\Phi(9, 31) = \Phi(26, 30)$	$\Phi(13, 18) = \Phi(20, 23)$	$\Phi(20, 23) = \Phi(22, 25)$
$\Phi(2, 12) = \Phi(6, 25)$	$\Phi(7, 8) = \Phi(13, 13)$	$\Phi(9, 32) = \Phi(27, 30)$	$\Phi(13, 21) = \Phi(14, 25)$	$\Phi(20, 26) = \Phi(24, 27)$
$\Phi(2, 14) = \Phi(3, 13)$	$\Phi(7, 9) = \Phi(14, 14)$	$\Phi(9, 33) = \Phi(21, 30)$	$\Phi(13, 26) = \Phi(14, 24)$	$\Phi(20, 28) = \Phi(22, 29)$
$\Phi(2, 15) = \Phi(3, 8)$	$\Phi(7, 10) = \Phi(19, 19)$	$\Phi(9, 34) = \Phi(30, 30)$	$\Phi(13, 27) = \Phi(14, 20)$	$\Phi(20, 30) = \Phi(27, 29)$
$\Phi(2, 16) = \Phi(4, 20)$	$\Phi(7, 11) = \Phi(22, 22)$	$\Phi(10, 11) = \Phi(16, 16)$	$\Phi(13, 30) = \Phi(14, 29)$	$\Phi(20, 31) = \Phi(24, 32)$
$\Phi(2, 17) = \Phi(4, 25)$	$\Phi(7, 12) = \Phi(23, 23)$	$\Phi(10, 12) = \Phi(17, 17)$	$\Phi(13, 31) = \Phi(19, 29)$	$\Phi(20, 33) = \Phi(25, 32)$
$\Phi(2, 18) = \Phi(5, 25)$	$\Phi(7, 15) = \Phi(13, 14)$	$\Phi(10, 13) = \Phi(19, 24)$	$\Phi(13, 32) = \Phi(20, 28)$	$\Phi(20, 34) = \Phi(29, 32)$
$\Phi(2, 19) = \Phi(4, 13)$	$\Phi(7, 16) = \Phi(19, 22)$	$\Phi(10, 14) = \Phi(19, 26)$	$\Phi(13, 33) = \Phi(23, 29)$	$\Phi(21, 22) = \Phi(23, 27)$
$\Phi(2, 20) = \Phi(5, 8)$	$\Phi(7, 17) = \Phi(19, 23)$	$\Phi(10, 15) = \Phi(24, 26)$	$\Phi(13, 34) = \Phi(28, 29)$	$\Phi(21, 24) = \Phi(25, 26)$
$\Phi(2, 21) = \Phi(3, 25)$	$\Phi(7, 18) = \Phi(22, 23)$	$\Phi(10, 18) = \Phi(16, 17)$	$\Phi(14, 16) = \Phi(19, 27)$	$\Phi(21, 28) = \Phi(23, 30)$
$\Phi(2, 22) = \Phi(5, 13)$	$\Phi(7, 20) = \Phi(13, 22)$	$\Phi(10, 20) = \Phi(16, 24)$	$\Phi(14, 17) = \Phi(19, 21)$	$\Phi(21, 29) = \Phi(25, 30)$
$\Phi(2, 23) = \Phi(6, 13)$	$\Phi(7, 21) = \Phi(14, 23)$	$\Phi(10, 21) = \Phi(17, 26)$	$\Phi(14, 18) = \Phi(21, 22)$	$\Phi(21, 31) = \Phi(26, 33)$
$\Phi(2, 24) = \Phi(4, 8)$	$\Phi(7, 24) = \Phi(13, 19)$	$\Phi(10, 22) = \Phi(16, 19)$	$\Phi(14, 20) = \Phi(15, 22)$	$\Phi(21, 32) = \Phi(27, 33)$
$\Phi(2, 25) = \Phi(6, 8)$	$\Phi(7, 25) = \Phi(13, 23)$	$\Phi(10, 23) = \Phi(17, 19)$	$\Phi(14, 24) = \Phi(15, 19)$	$\Phi(21, 34) = \Phi(30, 33)$
$\Phi(2, 26) = \Phi(3, 24)$	$\Phi(7, 26) = \Phi(14, 19)$	$\Phi(10, 25) = \Phi(17, 24)$	$\Phi(14, 25) = \Phi(15, 23)$	$\Phi(22, 30) = \Phi(27, 28)$
$\Phi(2, 27) = \Phi(3, 20)$	$\Phi(7, 27) = \Phi(14, 22)$	$\Phi(10, 27) = \Phi(16, 26)$	$\Phi(14, 29) = \Phi(15, 28)$	$\Phi(22, 33) = \Phi(23, 32)$
$\Phi(2, 30) = \Phi(3, 29)$	$\Phi(7, 29) = \Phi(13, 28)$	$\Phi(10, 28) = \Phi(19, 31)$	$\Phi(14, 31) = \Phi(19, 30)$	$\Phi(22, 34) = \Phi(28, 32)$
$\Phi(2, 31) = \Phi(4, 29)$	$\Phi(7, 30) = \Phi(14, 28)$	$\Phi(10, 29) = \Phi(24, 31)$	$\Phi(14, 32) = \Phi(22, 30)$	$\Phi(23, 29) = \Phi(25, 28)$
$\Phi(2, 32) = \Phi(5, 29)$	$\Phi(7, 31) = \Phi(19, 28)$	$\Phi(10, 30) = \Phi(26, 31)$	$\Phi(14, 33) = \Phi(21, 28)$	$\Phi(23, 34) = \Phi(28, 33)$
$\Phi(2, 33) = \Phi(6, 29)$	$\Phi(7, 32) = \Phi(22, 28)$	$\Phi(10, 32) = \Phi(16, 31)$	$\Phi(14, 34) = \Phi(28, 30)$	$\Phi(24, 30) = \Phi(26, 29)$
$\Phi(3, 10) = \Phi(4, 26)$	$\Phi(7, 33) = \Phi(23, 28)$	$\Phi(10, 33) = \Phi(17, 31)$	$\Phi(15, 16) = \Phi(20, 26)$	$\Phi(24, 33) = \Phi(25, 31)$
$\Phi(3, 11) = \Phi(5, 27)$	$\Phi(7, 34) = \Phi(28, 28)$	$\Phi(10, 34) = \Phi(31, 31)$	$\Phi(15, 17) = \Phi(21, 24)$	$\Phi(24, 34) = \Phi(29, 31)$
$\Phi(3, 12) = \Phi(6, 21)$	$\Phi(8, 9) = \Phi(15, 15)$	$\Phi(11, 12) = \Phi(18, 18)$	$\Phi(15, 18) = \Phi(20, 21)$	$\Phi(25, 34) = \Phi(29, 33)$
$\Phi(3, 16) = \Phi(4, 27)$	$\Phi(8, 10) = \Phi(24, 24)$	$\Phi(11, 13) = \Phi(20, 22)$	$\Phi(15, 31) = \Phi(24, 30)$	$\Phi(26, 32) = \Phi(27, 31)$
$\Phi(3, 17) = \Phi(4, 21)$	$\Phi(8, 11) = \Phi(20, 20)$	$\Phi(11, 14) = \Phi(22, 27)$	$\Phi(15, 32) = \Phi(20, 30)$	$\Phi(26, 34) = \Phi(30, 31)$
$\Phi(3, 18) = \Phi(5, 21)$	$\Phi(8, 12) = \Phi(25, 25)$	$\Phi(11, 15) = \Phi(20, 27)$	$\Phi(15, 33) = \Phi(21, 29)$	$\Phi(27, 34) = \Phi(30, 32)$
$\Phi(3, 19) = \Phi(4, 14)$	$\Phi(8, 14) = \Phi(13, 15)$	$\Phi(11, 17) = \Phi(16, 18)$	$\Phi(15, 34) = \Phi(29, 30)$	
$\Phi(3, 20) = \Phi(5, 15)$	$\Phi(8, 16) = \Phi(20, 24)$	$\Phi(11, 19) = \Phi(16, 22)$	$\Phi(16, 21) = \Phi(17, 27)$	
$\Phi(3, 21) = \Phi(6, 9)$	$\Phi(8, 17) = \Phi(24, 25)$	$\Phi(11, 21) = \Phi(18, 27)$	$\Phi(16, 23) = \Phi(17, 22)$	
$\Phi(3, 22) = \Phi(5, 14)$	$\Phi(8, 18) = \Phi(20, 25)$	$\Phi(11, 23) = \Phi(18, 22)$	$\Phi(16, 25) = \Phi(17, 20)$	

Table 5.13: Constraints on φ and Φ for the SDP solution of the 3-D localization

3-D Single-Time \ 3-D Multiple-Time	3-D Multiple-Time		
$\varphi(7) = \Phi(1, 1)$	$\Phi(1, 36) = \Phi(2, 35)$	$\Phi(12, 37) = \Phi(21, 40)$	$\Phi(21, 36) = \Phi(25, 37)$
$\varphi(8) = \Phi(2, 2)$	$\Phi(1, 37) = \Phi(3, 35)$	$\Phi(12, 38) = \Phi(17, 40)$	$\Phi(21, 38) = \Phi(26, 40)$
$\varphi(9) = \Phi(3, 3)$	$\Phi(1, 38) = \Phi(4, 35)$	$\Phi(12, 39) = \Phi(18, 40)$	$\Phi(21, 39) = \Phi(27, 40)$
$\varphi(10) = \Phi(4, 4)$	$\Phi(1, 39) = \Phi(5, 35)$	$\Phi(12, 41) = \Phi(33, 40)$	$\Phi(21, 41) = \Phi(30, 40)$
$\varphi(11) = \Phi(5, 5)$	$\Phi(1, 40) = \Phi(6, 35)$	$\Phi(12, 42) = \Phi(40, 40)$	$\Phi(21, 42) = \Phi(37, 40)$
$\varphi(12) = \Phi(6, 6)$	$\Phi(2, 37) = \Phi(3, 36)$	$\Phi(13, 37) = \Phi(14, 36)$	$\Phi(22, 37) = \Phi(27, 35)$
$\varphi(13) = \Phi(1, 2)$	$\Phi(2, 38) = \Phi(4, 36)$	$\Phi(13, 38) = \Phi(19, 36)$	$\Phi(22, 40) = \Phi(23, 39)$
$\varphi(14) = \Phi(1, 3)$	$\Phi(2, 39) = \Phi(5, 36)$	$\Phi(13, 39) = \Phi(20, 35)$	$\Phi(22, 41) = \Phi(28, 39)$
$\varphi(15) = \Phi(2, 3)$	$\Phi(2, 40) = \Phi(6, 36)$	$\Phi(13, 40) = \Phi(23, 36)$	$\Phi(22, 42) = \Phi(35, 39)$
$\varphi(16) = \Phi(4, 5)$	$\Phi(3, 38) = \Phi(4, 37)$	$\Phi(13, 41) = \Phi(28, 36)$	$\Phi(23, 36) = \Phi(25, 35)$
$\varphi(17) = \Phi(4, 6)$	$\Phi(3, 39) = \Phi(5, 37)$	$\Phi(13, 42) = \Phi(35, 36)$	$\Phi(23, 41) = \Phi(28, 40)$
$\varphi(18) = \Phi(5, 6)$	$\Phi(3, 40) = \Phi(6, 37)$	$\Phi(14, 36) = \Phi(15, 35)$	$\Phi(23, 42) = \Phi(35, 40)$
$\varphi(19) = \Phi(1, 4)$	$\Phi(4, 39) = \Phi(5, 38)$	$\Phi(14, 38) = \Phi(19, 37)$	$\Phi(24, 37) = \Phi(26, 36)$
$\varphi(20) = \Phi(2, 5)$	$\Phi(4, 40) = \Phi(6, 38)$	$\Phi(14, 39) = \Phi(22, 37)$	$\Phi(24, 40) = \Phi(25, 38)$
$\varphi(21) = \Phi(3, 6)$	$\Phi(5, 40) = \Phi(6, 39)$	$\Phi(14, 40) = \Phi(21, 35)$	$\Phi(24, 41) = \Phi(29, 38)$
$\varphi(22) = \Phi(1, 5)$	$\Phi(7, 36) = \Phi(13, 35)$	$\Phi(14, 41) = \Phi(28, 37)$	$\Phi(24, 42) = \Phi(36, 38)$
$\varphi(23) = \Phi(1, 6)$	$\Phi(7, 37) = \Phi(14, 35)$	$\Phi(14, 42) = \Phi(35, 37)$	$\Phi(25, 41) = \Phi(29, 40)$
$\varphi(24) = \Phi(2, 4)$	$\Phi(7, 38) = \Phi(19, 35)$	$\Phi(15, 38) = \Phi(24, 37)$	$\Phi(25, 42) = \Phi(36, 40)$
$\varphi(25) = \Phi(2, 6)$	$\Phi(7, 39) = \Phi(22, 35)$	$\Phi(15, 39) = \Phi(20, 37)$	$\Phi(26, 39) = \Phi(27, 38)$
$\varphi(26) = \Phi(3, 4)$	$\Phi(7, 40) = \Phi(23, 35)$	$\Phi(15, 40) = \Phi(21, 36)$	$\Phi(26, 41) = \Phi(30, 38)$
$\varphi(27) = \Phi(3, 5)$	$\Phi(7, 41) = \Phi(28, 35)$	$\Phi(15, 41) = \Phi(29, 37)$	$\Phi(26, 42) = \Phi(37, 38)$
$\varphi(28) = \Phi(2, 22) + \Phi(4, 7) + \Phi(6, 14)$	$\Phi(7, 42) = \Phi(35, 35)$	$\Phi(15, 42) = \Phi(36, 37)$	$\Phi(27, 41) = \Phi(30, 39)$
$\varphi(29) = \Phi(2, 19) + \Phi(3, 25) + \Phi(5, 8)$	$\Phi(8, 35) = \Phi(13, 36)$	$\Phi(16, 35) = \Phi(19, 39)$	$\Phi(27, 42) = \Phi(37, 39)$
$\varphi(30) = \Phi(1, 26) + \Phi(5, 15) + \Phi(6, 9)$	$\Phi(8, 37) = \Phi(15, 36)$	$\Phi(16, 36) = \Phi(20, 38)$	$\Phi(28, 36) = \Phi(29, 35)$
$\varphi(31) = \Phi(1, 10) + \Phi(3, 17) + \Phi(4, 20)$	$\Phi(8, 38) = \Phi(24, 36)$	$\Phi(16, 37) = \Phi(26, 39)$	$\Phi(28, 37) = \Phi(30, 35)$
$\varphi(32) = \Phi(1, 16) + \Phi(2, 11) + \Phi(3, 18)$	$\Phi(8, 39) = \Phi(20, 36)$	$\Phi(16, 40) = \Phi(17, 39)$	$\Phi(28, 38) = \Phi(31, 35)$
$\varphi(33) = \Phi(4, 23) + \Phi(5, 25) + \Phi(6, 21)$	$\Phi(8, 40) = \Phi(25, 36)$	$\Phi(16, 41) = \Phi(31, 39)$	$\Phi(28, 39) = \Phi(32, 35)$
$\varphi(34) = \Phi(1, 31) + \Phi(3, 33) + \Phi(5, 29)$	$\Phi(8, 41) = \Phi(29, 36)$	$\Phi(16, 42) = \Phi(38, 39)$	$\Phi(28, 40) = \Phi(33, 35)$
	$\Phi(8, 42) = \Phi(36, 36)$	$\Phi(17, 35) = \Phi(19, 40)$	$\Phi(28, 41) = \Phi(34, 35)$
	$\Phi(9, 35) = \Phi(14, 37)$	$\Phi(17, 36) = \Phi(24, 40)$	$\Phi(28, 42) = \Phi(35, 41)$
	$\Phi(9, 36) = \Phi(15, 37)$	$\Phi(17, 37) = \Phi(21, 38)$	$\Phi(29, 37) = \Phi(30, 36)$
	$\Phi(9, 38) = \Phi(26, 37)$	$\Phi(17, 39) = \Phi(18, 38)$	$\Phi(29, 38) = \Phi(31, 36)$
	$\Phi(9, 39) = \Phi(27, 37)$	$\Phi(17, 41) = \Phi(31, 40)$	$\Phi(29, 39) = \Phi(32, 36)$
	$\Phi(9, 40) = \Phi(21, 37)$	$\Phi(17, 42) = \Phi(38, 40)$	$\Phi(29, 40) = \Phi(33, 36)$
	$\Phi(9, 41) = \Phi(30, 37)$	$\Phi(18, 35) = \Phi(22, 40)$	$\Phi(29, 41) = \Phi(34, 36)$
	$\Phi(9, 42) = \Phi(37, 37)$	$\Phi(18, 36) = \Phi(20, 40)$	$\Phi(29, 42) = \Phi(36, 41)$
	$\Phi(10, 35) = \Phi(19, 38)$	$\Phi(18, 37) = \Phi(21, 39)$	$\Phi(30, 38) = \Phi(31, 37)$
	$\Phi(10, 36) = \Phi(24, 38)$	$\Phi(18, 41) = \Phi(32, 40)$	$\Phi(30, 39) = \Phi(32, 37)$
	$\Phi(10, 37) = \Phi(26, 38)$	$\Phi(18, 42) = \Phi(39, 40)$	$\Phi(30, 40) = \Phi(33, 37)$
	$\Phi(10, 39) = \Phi(16, 38)$	$\Phi(19, 36) = \Phi(24, 35)$	$\Phi(30, 41) = \Phi(34, 37)$
	$\Phi(10, 40) = \Phi(17, 38)$	$\Phi(19, 37) = \Phi(26, 35)$	$\Phi(30, 42) = \Phi(37, 41)$
	$\Phi(10, 41) = \Phi(31, 38)$	$\Phi(19, 39) = \Phi(22, 38)$	$\Phi(31, 39) = \Phi(32, 38)$
	$\Phi(10, 42) = \Phi(38, 38)$	$\Phi(19, 40) = \Phi(23, 38)$	$\Phi(31, 40) = \Phi(33, 38)$
	$\Phi(11, 35) = \Phi(22, 39)$	$\Phi(19, 41) = \Phi(28, 38)$	$\Phi(31, 41) = \Phi(34, 38)$
	$\Phi(11, 36) = \Phi(20, 39)$	$\Phi(19, 42) = \Phi(35, 38)$	$\Phi(31, 42) = \Phi(38, 41)$
	$\Phi(11, 37) = \Phi(27, 39)$	$\Phi(20, 35) = \Phi(22, 36)$	$\Phi(32, 40) = \Phi(33, 39)$
	$\Phi(11, 38) = \Phi(16, 39)$	$\Phi(20, 37) = \Phi(27, 36)$	$\Phi(32, 41) = \Phi(34, 39)$
	$\Phi(11, 40) = \Phi(18, 39)$	$\Phi(20, 38) = \Phi(24, 39)$	$\Phi(32, 42) = \Phi(39, 41)$
	$\Phi(11, 41) = \Phi(32, 39)$	$\Phi(20, 40) = \Phi(25, 39)$	$\Phi(33, 41) = \Phi(34, 40)$
	$\Phi(11, 42) = \Phi(39, 39)$	$\Phi(20, 41) = \Phi(29, 39)$	$\Phi(33, 42) = \Phi(40, 41)$
	$\Phi(12, 35) = \Phi(23, 40)$	$\Phi(20, 42) = \Phi(36, 39)$	$\Phi(34, 42) = \Phi(41, 41)$
	$\Phi(12, 36) = \Phi(25, 40)$	$\Phi(21, 35) = \Phi(23, 37)$	
	$\varphi(35) = \Phi(1, 10) + \Phi(5, 22) + \Phi(6, 23)$	$\varphi(39) = \Phi(4, 16) + \Phi(5, 11) + \Phi(6, 18)$	
	$\varphi(36) = \Phi(2, 12) + \Phi(4, 24) + \Phi(5, 20)$	$\varphi(40) = \Phi(4, 17) + \Phi(5, 18) + \Phi(6, 12)$	
	$\varphi(37) = \Phi(3, 11) + \Phi(4, 26) + \Phi(6, 21)$	$\varphi(41) = \Phi(1, 38) + \Phi(2, 39) + \Phi(3, 40)$	
	$\varphi(38) = \Phi(4, 10) + \Phi(5, 16) + \Phi(6, 17)$	$\varphi(42) = \Phi(4, 38) + \Phi(5, 39) + \Phi(6, 40)$	

5.4 CRLB

The CRLB derived in Chapter 2 is valid only for the fixed sensor scenario and does not consider the additional error coming from the sensor movement. In this section, we shall change the derivation slightly to consider the error of sensor velocity.

We define the nuisance variable vector as

$$\boldsymbol{\alpha}^o = [f_o^o, \mathbf{s}^{oT}, \dot{\mathbf{s}}^{oT}]^T, \quad (5.61)$$

and the observation vector as

$$\mathbf{g} = [\mathbf{f}^T, f_o, \mathbf{s}^T, \dot{\mathbf{s}}^T]^T. \quad (5.62)$$

Since the location vector for estimation is $\boldsymbol{\theta}^o = [\mathbf{u}^{oT}, \dot{\mathbf{u}}^{oT}]^T$ and the parameter vector for evaluating the CRLB is $\boldsymbol{\phi}^o = [\boldsymbol{\theta}^{oT}, \boldsymbol{\alpha}^{oT}]^T$, the logarithm of the probability density function under the Gaussian data will be

$$\begin{aligned} \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{g}) &= \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{f}) + \ln \mathcal{L}(\boldsymbol{\phi}^o; f_o) + \ln \mathcal{L}(\boldsymbol{\phi}^o; \mathbf{s}) + \ln \mathcal{L}(\boldsymbol{\phi}^o; \dot{\mathbf{s}}) \\ &= \kappa - \frac{1}{2} (\mathbf{f} - \mathbf{f}^o)^T \mathbf{Q}_n^{-1} (\mathbf{f} - \mathbf{f}^o) - \frac{1}{2} \sigma_{f_o}^{-2} (f_o - f_o^o)^2 - \frac{1}{2} (\mathbf{s} - \mathbf{s}^o)^T \mathbf{Q}_s^{-1} (\mathbf{s} - \mathbf{s}^o) \\ &\quad - \frac{1}{2} (\dot{\mathbf{s}} - \dot{\mathbf{s}}^o)^T \mathbf{Q}_{\dot{\mathbf{s}}}^{-1} (\dot{\mathbf{s}} - \dot{\mathbf{s}}^o), \end{aligned} \quad (5.63)$$

where κ is a constant. The CRLB can be partitioned into a 2×2 block matrix as

$$\text{CRLB}(\boldsymbol{\phi}^o) = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\phi}^o \partial \boldsymbol{\phi}^{oT}} \right]^{-1} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y}^T & \mathbf{Z} \end{bmatrix}^{-1}. \quad (5.64)$$

The blocks in (5.64) are corresponding to the estimation performance of $\boldsymbol{\theta}^o$ and $\boldsymbol{\alpha}^o$, and they are given by

$$\mathbf{X} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\theta}^o \partial \boldsymbol{\theta}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\theta}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (5.65a)$$

$$\mathbf{Y} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\theta}^o \partial \boldsymbol{\alpha}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\theta}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}}, \quad (5.65b)$$

$$\mathbf{Z} = -E \left[\frac{\partial^2 \ln f(\mathbf{g}; \boldsymbol{\phi}^o)}{\partial \boldsymbol{\alpha}^o \partial \boldsymbol{\alpha}^{oT}} \right] = \frac{\partial \mathbf{f}^{oT}}{\partial \boldsymbol{\alpha}^o} \mathbf{Q}_n^{-1} \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}} + \mathbf{Q}_\alpha^{-1}, \quad (5.65c)$$

where $\mathbf{Q}_\alpha = \text{diag}(\sigma_{f_o}^2, \mathbf{Q}_s, \mathbf{Q}_{\dot{s}})$. The expressions for the partial derivatives in (5.65) are given in Appendix E. Applying the partitioned matrix inversion formula [101] on (5.64) to get, from the upper left block

$$\text{CRLB}(\boldsymbol{\theta}^o) = (\mathbf{X} - \mathbf{Y}\mathbf{Z}^{-1}\mathbf{Y}^T)^{-1} = \mathbf{X}^{-1} + \mathbf{X}^{-1}\mathbf{Y}(\mathbf{Z} - \mathbf{Y}^T\mathbf{X}^{-1}\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}^{-1}. \quad (5.66)$$

The first term on the right of (5.66) is the CRLB of $\boldsymbol{\theta}^o$ when there is no sensor location error or carrier frequency noise. The second term in (5.66) is the additional performance loss in the presence of Δf_o , $\Delta \mathbf{s}$ and $\Delta \dot{\mathbf{s}}$.

5.5 Analysis

In this section, we shall prove that the covariance matrix of the proposed algebraic solution can reach the CRLB accuracy under the first order analysis in which the second and higher order noise terms are very small to be neglected when some conditions are satisfied. We shall start with the theoretical covariance matrix given in (5.57) as a general equation for all the localization cases and then we shall give the

conditions for each specific case separately according to the definition of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{W}}$. Inserting (5.47) and (5.56) in (5.57) and taking the inverse of both sides up to the first order noise component give

$$\text{cov}(\boldsymbol{\theta})^{-1} = \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_\varepsilon^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \quad (5.67)$$

For convenience, we express (5.10) as $\mathbf{Q}_\varepsilon = \mathbf{Q}_n + \mathbf{D} \mathbf{Q}_\alpha \mathbf{D}^T$, where $\mathbf{D} = [\mathbf{d}_f, \mathbf{D}_s, \mathbf{D}_\dot{s}]$ and \mathbf{Q}_α is defined below (5.65). The inverse of \mathbf{Q}_ε , from the matrix inversion lemma [104], is

$$\mathbf{Q}_\varepsilon^{-1} = \mathbf{Q}_n^{-1} - \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1}. \quad (5.68)$$

Inserting (5.68) in (5.67) gives

$$\begin{aligned} \text{cov}(\boldsymbol{\theta})^{-1} &= \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}} \\ &\quad - \tilde{\mathbf{A}}^T \tilde{\mathbf{B}}^{-T} \mathbf{A}^T \mathbf{B}^{-T} \mathbf{Q}_n^{-1} \mathbf{D} (\mathbf{Q}_\alpha^{-1} + \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Q}_n^{-1} \mathbf{B}^{-1} \mathbf{A} \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{A}}. \end{aligned} \quad (5.69)$$

\mathbf{B}^{-1} and $\tilde{\mathbf{B}}^{-1}$ can be evaluated analytically since \mathbf{B} is diagonal and $\tilde{\mathbf{B}}$ is sparse. Replacing \mathbf{A} , \mathbf{B} , $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ in (5.69) by their true values and inserting (5.65) in (5.66) with the gradients given in Appendix E, the following two equations can be shown to be true upon using direct algebraic evaluation

$$\mathbf{B}^{o-1} \mathbf{A}^o \tilde{\mathbf{B}}^{o-1} \tilde{\mathbf{A}}^o = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (5.70a)$$

$$\mathbf{D} = \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}}. \quad (5.70b)$$

Table 5.14: Small Noise Conditions For $k = 0, 1, \dots, N - 1$ and $i = 1, 2, \dots, M$

No.	2-D Single-Time	2-D Multiple-Time	3-D Single-Time	3-D Multiple-Time
1	$\frac{n_i}{f_i^o - f_o^o} \simeq 0$	$\frac{n_{k,i}}{f_{k,i}^o - f_o^o} \simeq 0$	$\frac{n_i}{f_i^o - f_o^o} \simeq 0$	$\frac{n_{k,i}}{f_{k,i}^o - f_o^o} \simeq 0$
2	$\frac{\Delta f_o}{f_i^o - f_o^o} \simeq 0$	$\frac{\Delta f_o}{f_{k,i}^o - f_o^o} \simeq 0$	$\frac{\Delta f_o}{f_i^o - f_o^o} \simeq 0$	$\frac{\Delta f_o}{f_{k,i}^o - f_o^o} \simeq 0$
3	$\frac{\Delta x_i}{x_i^o} \simeq 0$ $\frac{\Delta y_i}{y_i^o} \simeq 0$	$\frac{\Delta x_{k,i}}{x_{k,i}^o} \simeq 0$ $\frac{\Delta y_{k,i}}{y_{k,i}^o} \simeq 0$	$\frac{\Delta x_i}{x_i^o} \simeq 0$ $\frac{\Delta y_i}{y_i^o} \simeq 0$ $\frac{\Delta z_i}{z_i^o} \simeq 0$	$\frac{\Delta x_{k,i}}{x_{k,i}^o} \simeq 0$ $\frac{\Delta y_{k,i}}{y_{k,i}^o} \simeq 0$ $\frac{\Delta z_{k,i}}{z_{k,i}^o} \simeq 0$
4	$\frac{\Delta x_i}{r_i^o} \simeq 0$ $\frac{\Delta y_i}{r_i^o} \simeq 0$	$\frac{\Delta x_{k,i}}{r_{k,i}^o} \simeq 0$ $\frac{\Delta y_{k,i}}{r_{k,i}^o} \simeq 0$	$\frac{\Delta x_i}{r_i^o} \simeq 0$ $\frac{\Delta y_i}{r_i^o} \simeq 0$ $\frac{\Delta z_i}{r_i^o} \simeq 0$	$\frac{\Delta x_{k,i}}{r_{k,i}^o} \simeq 0$ $\frac{\Delta y_{k,i}}{r_{k,i}^o} \simeq 0$ $\frac{\Delta z_{k,i}}{r_{k,i}^o} \simeq 0$
5	$\frac{\dot{\Delta} x_i}{\dot{x}_i^o} \simeq 0$ $\frac{\dot{\Delta} y_i}{\dot{y}_i^o} \simeq 0$	$\frac{\dot{\Delta} x_{k,i}}{\dot{x}_{k,i}^o} \simeq 0$ $\frac{\dot{\Delta} y_{k,i}}{\dot{y}_{k,i}^o} \simeq 0$	$\frac{\dot{\Delta} x_i}{\dot{x}_i^o} \simeq 0$ $\frac{\dot{\Delta} y_i}{\dot{y}_i^o} \simeq 0$ $\frac{\dot{\Delta} z_i}{\dot{z}_i^o} \simeq 0$	$\frac{\dot{\Delta} x_{k,i}}{\dot{x}_{k,i}^o} \simeq 0$ $\frac{\dot{\Delta} y_{k,i}}{\dot{y}_{k,i}^o} \simeq 0$ $\frac{\dot{\Delta} z_{k,i}}{\dot{z}_{k,i}^o} \simeq 0$
6	$\frac{\mathbf{s}_i^{oT} \Delta \mathbf{s}_i}{\mathbf{s}_i^{oT} \dot{\mathbf{s}}_i^o} \simeq 0$	$\frac{\mathbf{s}_{k,i}^{oT} \Delta \mathbf{s}_{k,i}}{\mathbf{s}_{k,i}^{oT} \dot{\mathbf{s}}_{k,i}^o} \simeq 0$	$\frac{\mathbf{s}_i^{oT} \Delta \mathbf{s}_i}{\mathbf{s}_i^{oT} \dot{\mathbf{s}}_i^o} \simeq 0$	$\frac{\mathbf{s}_{k,i}^{oT} \Delta \mathbf{s}_{k,i}}{\mathbf{s}_{k,i}^{oT} \dot{\mathbf{s}}_{k,i}^o} \simeq 0$
7	$\frac{\dot{\mathbf{s}}_i^{oT} \Delta \mathbf{s}_i}{\mathbf{s}_i^{oT} \dot{\mathbf{s}}_i^o} \simeq 0$	$\frac{\dot{\mathbf{s}}_{k,i}^{oT} \Delta \mathbf{s}_{k,i}}{\mathbf{s}_{k,i}^{oT} \dot{\mathbf{s}}_{k,i}^o} \simeq 0$	$\frac{\dot{\mathbf{s}}_i^{oT} \Delta \mathbf{s}_i}{\mathbf{s}_i^{oT} \dot{\mathbf{s}}_i^o} \simeq 0$	$\frac{\dot{\mathbf{s}}_{k,i}^{oT} \Delta \mathbf{s}_{k,i}}{\mathbf{s}_{k,i}^{oT} \dot{\mathbf{s}}_{k,i}^o} \simeq 0$
8	$\frac{\Delta \varphi(l)}{\varphi^o(l)} \simeq 0$ $l = 1, 2, \dots, 18$	$\frac{\Delta \varphi(l)}{\varphi^o(l)} \simeq 0$ $l = 1, 2, \dots, 18,$ $22, 23$	$\frac{\Delta \varphi(l)}{\varphi^o(l)} \simeq 0$ $l = 1, 2, \dots, 33$	$\frac{\Delta \varphi(l)}{\varphi^o(l)} \simeq 0$ $l = 1, 2, \dots, 33,$ $38, 39, 40$
9	$\frac{\Delta \varphi(l)}{r_i^o} \simeq 0$ $l = 1, 2, \dots, 4$	$\frac{\Delta \varphi(l)}{r_{k,i}^o} \simeq 0$ $l = 1, 2, \dots, 4$	$\frac{\Delta \varphi(l)}{r_i^o} \simeq 0$ $l = 1, 2, \dots, 6$	$\frac{\Delta \varphi(l)}{r_{k,i}^o} \simeq 0$ $l = 1, 2, \dots, 6$

(5.70) can be achieved when the approximation of \mathbf{A} , \mathbf{B} , $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ by their true values is valid. The conditions to assure this validation for the four localization cases are listed in Table 5.14.

The first and second conditions of Table 5.14 impose that the Doppler shift generated by the relative movement among the object and the sensors is relatively large compared to the measurement noise and the error of carrier frequency, which is expected in order for the localization by frequency only measurements to be possible. The third and fourth conditions require the sensor position error relative to the true

sensor position coordinates and to the true sensor-object distance to be small. Condition 5 implies that the error of sensor velocity is relatively small compared with the true values of the sensor velocity coordinates. Conditions 7 and 8 demand that the error associated with the dot product between the position and velocity of the sensor is relatively small compared with the true value of this product. Condition 8 requires the elements of $\tilde{\mathbf{A}}$ that are coming from the first stage solution to have sufficiently small error compared to their true values and condition 9 needs the error of the elements in $\tilde{\mathbf{C}}$ to be small relative to the sensors-object distances.

When the conditions of Table 5.14 are satisfied, we get

$$\mathbf{B}^{-1}\mathbf{A}\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{A}} \simeq \frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}}, \quad (5.71)$$

and as a result,

$$\text{cov}(\boldsymbol{\theta}) \simeq \text{CRLB}(\boldsymbol{\theta}). \quad (5.72)$$

5.6 Simulations

To test the performance of the proposed solutions, we shall simulate the problem of localizing a moving object in the underwater environment using the acoustic signals. The object travels at a constant speed of $\|\mathbf{u}^o\| = 10$ and radiate a single tone at $f_o^o = 15$ kHz with speed of propagation $c = 1500$ m/s [1, 83, 81]. The measurements have covariance matrices $\mathbf{Q}_k = \sigma^2 \mathbf{I}_M$, $k = 0, 1, \dots, N - 1$, and the unit of σ is Hz. The mean-square error (MSE) of the object location estimate will be used for the performance measure and it is computed by averaging 1,000 ensemble runs. The CRLB derived in Section 5.4 is presented as a performance reference. We shall denote

the closed-form solution by CFS and the semi-definite programming solution by SDP in the result figures. The SDP solution is obtained using the CVX toolbox [94]. To handle the numerical aspect of CVX, two scaling factors, s_1 and s_2 , will be used. s_1 is applied to the sensor position vector \mathbf{s} defined in (5.2) and s_2 is applied to the sensor velocity vector $\dot{\mathbf{s}}$ defined in (5.3) and to the quantity $d_{k,i}$ given by (5.11). We assume f_o is exactly known and the sensor position and sensor velocity errors are absent.

5.6.1 2-D Single-Time

For this localization case, we used $M = 20$ sensors with positions and velocities shown in Table 5.6.1. The sensor positions satisfy the condition $\|\mathbf{s}_i - \mathbf{s}_j\| \geq 150$ m for $i, j = 1, 2, \dots, M$ and $i < j$ to avoid near degenerated geometries. The object is placed at $\mathbf{u}_0^o = [1607.01 \quad 1472.35]^T$ m outside the sensors region with velocity $\dot{\mathbf{u}}^o = [7.5726 \quad 6.5311]^T$ m/s. The scaling factors for using the CVX are $s_1 = 5.6 \times 10^{-4}$ and $s_2 = 6.1 \times 10^{-2}$.

Fig. 5.1 shows the estimation accuracy of the proposed methods as the frequency measurement noise level σ increases. The MSE of the CFS and SDP methods reach the CRLB accuracy level for both position and velocity estimation excluding the position estimation for SDP at $\sigma = 3.2$ mHz, possibly due to CVX numerical accuracy, and also the CFS accuracy at $\sigma = 1$ Hz, due to the threshold effect.

5.6.2 2-D Multiple-Time

In this case, only $M = 4$ sensors are used and each sensor collects $N = 15$ measurements throughout its random movement. The sensors positions and velocities for

Table 5.15: Sensor Positions in (m) and Velocities in (m/s) for 2-D Single-Time Localization

i	x_i	y_i	\dot{x}_i	\dot{y}_i
1	-44.16	-311.56	-5.9757	-3.3623
2	-180.90	587.49	2.2498	-4.0959
3	447.67	-278.44	-6.9171	0.7386
4	39.70	39.88	6.7859	-5.385
5	300.42	223.29	-6.2975	4.3638
6	509.53	119.83	1.6936	-6.4772
7	-263.74	-375.92	5.9557	-1.4524
8	-377.65	-682.32	1.7847	-6.1058
9	-507.07	67.76	-2.4296	0.7225
10	-206.74	-110.62	6.269	1.029
11	461.93	354.00	4.139	-2.618
12	-561.46	-244.15	-0.8908	5.0193
13	740.70	-179.93	-1.5429	2.5112
14	379.52	-574.18	-3.6148	1.1946
15	-564.68	-580.38	-6.8161	2.5024
16	134.35	422.09	-2.5331	4.748
17	728.85	642.94	1.3534	-1.5602
18	208.66	743.82	3.8845	-3.0508
19	528.36	698.54	6.5255	-0.5112
20	636.30	-705.82	1.498	-4.8675

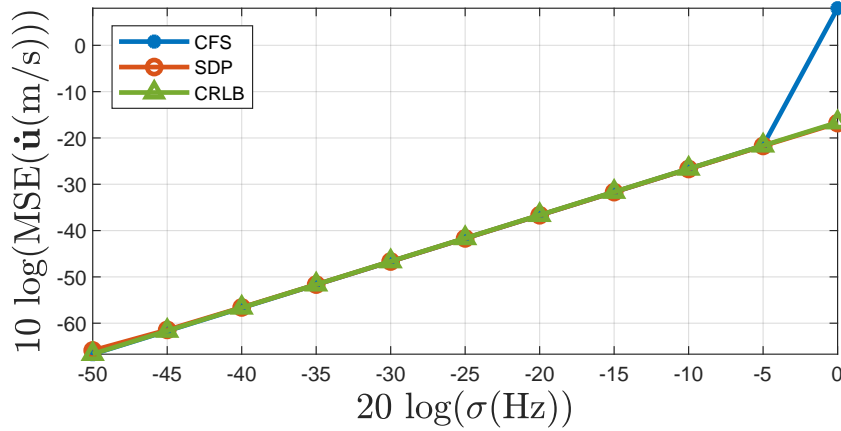
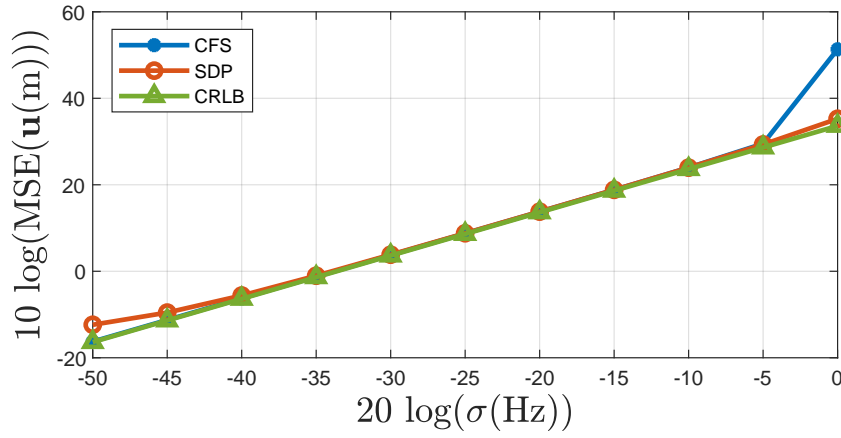


Figure 5.1: Performance of the proposed methods at different σ levels for 2-D single-time case. (a) position estimation, (b) velocity estimation.

$k = 0, 1, \dots, N - 1$ are shown in Table 5.16. The object starts the linear motion at $\mathbf{u}_0^o = [-188.96 \ 491.34]^T$ m with velocity $\dot{\mathbf{u}}^o = [-0.7598 \ -9.9711]^T$ m/s and ends at $\mathbf{u}_{14}^o = [-199.6 \ 351.74]^T$ m. The CVX scaling factors s_1 and s_2 are both set to 0.01.

Fig. 5.2 illustrates the performance of the proposed methods for this localization case where only few sensors are available but using multiple-time measurements from each sensor. The absolute performance improved compared to Fig. 5.1 as we have larger number of measurements in total. Nevertheless, the relative performance

Table 5.16: Sensor Positions in (m) and Velocities in (m/s) for 2-D Multiple-Time Localization

k	i	$x_{k,i}$	$y_{k,i}$	$\dot{x}_{k,i}$	$\dot{y}_{k,i}$	k	i	$x_{k,i}$	$y_{k,i}$	$\dot{x}_{k,i}$	$\dot{y}_{k,i}$
0	1	746.99	-285.99	3.8611	8.2414	8	1	810.37	-284.30	9.0958	-0.3082
0	2	463.54	-631.35	9.3566	1.1507	8	2	523.63	-606.07	8.138	-4.7586
0	3	-43.16	-341.02	-6.8761	1.618	8	3	0.85	-323.26	6.0098	-3.7123
0	4	521.74	424.69	-1.4801	-9.3201	8	4	572.21	479.81	7.436	5.8103
1	1	755.47	-282.68	8.4782	3.3088	9	1	817.59	-278.76	7.2234	5.5363
1	2	472.41	-628.15	8.8644	3.2084	9	2	530.33	-612.71	6.6972	-6.6346
1	3	-36.94	-337.66	6.216	3.3556	9	3	6.19	-327.89	5.3331	-4.6322
1	4	529.70	429.76	7.959	5.0703	9	4	579.73	474.11	7.5188	-5.7028
2	1	760.51	-275.10	5.0447	7.575	10	1	826.55	-277.19	8.9634	1.5769
2	2	477.50	-620.21	5.0936	7.9326	10	2	536.26	-620.03	5.9359	-7.3237
2	3	-31.40	-333.28	5.5379	4.3852	10	3	10.27	-333.66	4.0868	-5.7617
2	4	536.57	436.23	6.8684	6.4715	10	4	586.31	467.34	6.582	-6.7625
3	1	769.38	-273.07	8.871	2.0331	11	1	835.65	-277.50	9.0955	-0.3182
3	2	482.95	-612.52	5.4541	7.6892	11	2	541.72	-627.72	5.4572	-7.687
3	3	-26.96	-327.79	4.4449	5.4901	11	3	14.10	-339.59	3.8274	-5.9372
3	4	542.71	443.40	6.1389	7.1672	11	4	591.64	459.56	5.3299	-7.7876
4	1	778.48	-272.96	9.1003	0.115	12	1	844.52	-279.54	8.8707	-2.0346
4	2	489.19	-605.45	6.232	7.0734	12	2	546.86	-635.62	5.1363	-7.905
4	3	-23.03	-321.92	3.934	5.8671	12	3	18.17	-345.37	4.069	-5.7743
4	4	548.32	450.98	5.6118	7.587	12	4	596.75	451.62	5.1093	-7.934
5	1	787.49	-274.28	9.0043	-1.3234	13	1	852.72	-283.49	8.1999	-3.9485
5	2	497.19	-600.47	8.0053	4.9786	13	2	551.96	-643.55	5.1076	-7.9236
5	3	-18.82	-316.24	4.2009	5.679	13	3	23.34	-350.18	5.1673	-4.8164
5	4	553.53	458.85	5.2137	7.8659	13	4	601.95	443.74	5.1966	-7.8772
6	1	792.40	-281.94	4.9173	-7.6582	14	1	860.29	-288.55	7.5654	-5.0589
6	2	506.44	-598.67	9.253	1.8038	14	2	557.94	-650.83	5.9791	-7.2885
6	3	-11.83	-317.24	6.9924	-1.0024	14	3	29.31	-353.96	5.9688	-3.7778
6	4	558.92	466.59	5.3908	7.7455	14	4	608.17	436.65	6.2205	-7.0965
7	1	801.27	-283.99	8.8665	-2.0526						
7	2	515.49	-601.31	9.0482	-2.6458						
7	3	-5.15	-319.55	6.6767	-2.3065						
7	4	564.77	474.00	5.851	7.4041						

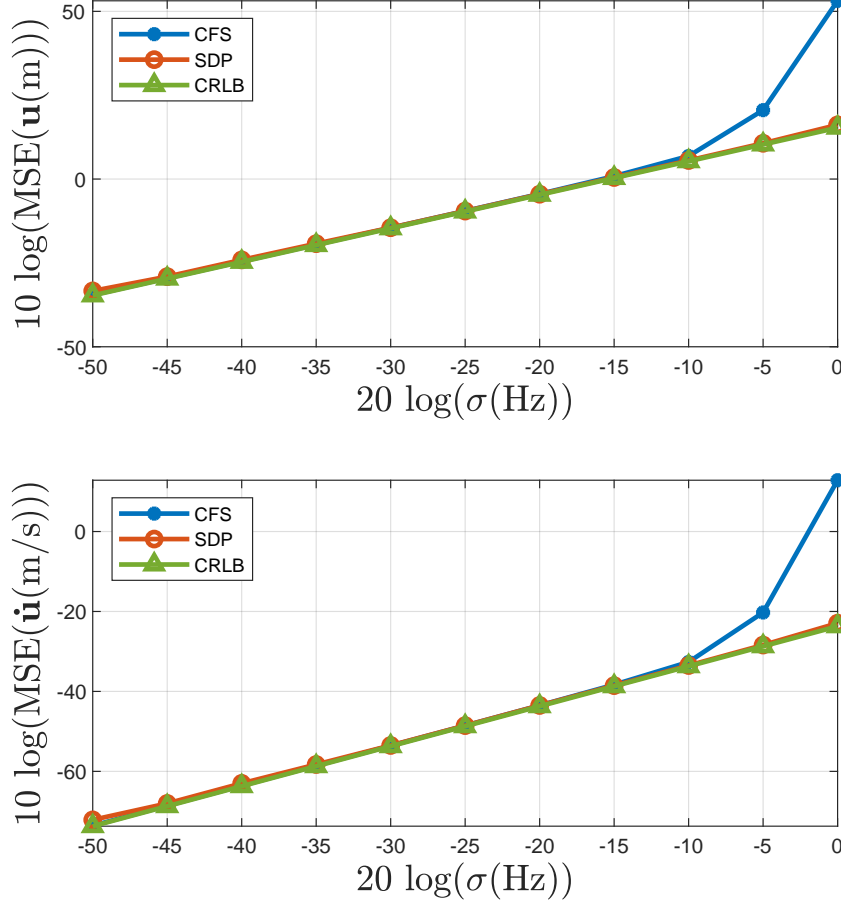


Figure 5.2: Performance of the proposed methods at different σ levels for 2-D multiple-time case. (a) position estimation, (b) velocity estimation.

among the CFS and SDP algorithms remains similar except the threshold effect of CFS appears at $\sigma = 0.56$ Hz and the SDP follows the CRLB very well in this case.

5.6.3 3-D Single-Time

For the 3-D single-time case, the unknown vector φ in (5.30) has 34 elements; therefore, we used $M = 35$ sensors to accommodate the algebraic solution requirement. The positions and velocities for this case are shown in Table 5.6.3. The minimum

distance among the sensors are imposed to be no less than 250 m and the scaling factors $s_1 = 5.63 \times 10^{-4}$ and $s_2 = 8.1 \times 10^{-2}$ are used. The true position and velocity of the object are set as $\mathbf{u}_0^o = [-14.71 \quad 992.2 \quad -1356.53]^T$ m and $\dot{\mathbf{u}}^o = [-5.0886 \quad 2.9613 \quad -8.6430]^T$ m/s.

Fig. 5.3 shows the estimation performance for the proposed methods when large number of sensors are used to locate a moving object in the 3-D space. The CFS and SDP algorithms exhibit optimum accuracy when σ is equal or less than 0.5 Hz. Higher σ levels can still be tolerated by the SDP method but it will cause the CFS to leave the CRLB accuracy.

5.6.4 3-D Multiple-Time

We use $M = 4$ sensors for this localization case and the number of successive measurements for each sensor is $N = 20$. The sensors locations are given in Table 5.18. The scaling factors s_1 and s_2 for use with CVX in obtaining the SDR solution are 0.015. The object movement starts at $\mathbf{u}_0^o = [-524.37 \quad -831.08 \quad 549.01]^T$ and ends at $\mathbf{u}_{19}^o = [-566.55 \quad -948.45 \quad 613.26]^T$ with constant velocity $\dot{\mathbf{u}}^o = [-2.2198 \quad -6.1775 \quad 3.3817]^T$.

Fig. 5.4 gives the estimation performance as the noise level σ increases when only 4 sensors with multiple-time measurements are used for the 3-D object localization. The SDP velocity estimation slightly deviate from the CRLB at $\sigma = 3.2$ mHz due to the CVX numerical accuracy limit and the threshold effect appears on the CFS algorithm when σ is larger than 0.5 Hz. Otherwise, both the CFS and SDP solutions perform very well in reaching the CRLB.

Table 5.17: Sensor Positions in (m) and Velocities in (m/s) for 3-D Single-Time Localization

i	x_i	y_i	z_i	\dot{x}_i	\dot{y}_i	\dot{z}_i
1	41.83	-206.72	539.89	0.7738	1.6133	4.7143
2	-4.87	-116.38	-524.85	4.6771	-4.8408	-5.048
3	-344.35	-193.66	-428.95	2.4653	-4.3342	1.8965
4	-481.78	546.54	454.49	-1.0883	-5.2303	-1.1167
5	-470.06	-159.43	52.05	0.0349	3.5272	-2.4159
6	-470.23	108.24	-127.74	3.0438	1.1788	0.9633
7	394.19	79.31	-728.67	-3.0827	5.7171	-2.5574
8	-638.72	337.34	6.21	5.7533	-5.2548	-2.3914
9	549.99	-557.65	491.69	2.0803	5.6624	-3.1808
10	-274.88	361.88	654.14	-0.4382	-3.6867	-0.9863
11	642.72	250.05	-224.01	-2.5141	-2.3323	-5.068
12	-601.02	-144.89	-279.45	-4.6114	3.3999	1.6901
13	-193.31	186.87	252.84	-3.2119	-4.8618	-0.6891
14	594.58	612.51	-601.58	-3.893	1.1395	1.1096
15	-622.39	-75.92	372.39	-4.0489	5.4512	-2.2167
16	-48.58	337.21	532.54	5.0095	1.0994	3.5499
17	-464.80	392.13	239.44	2.9073	-1.9004	0.2159
18	-620.32	-533.79	24.34	5.3742	2.8665	-0.7975
19	417.98	-363.18	290.10	-1.9452	-0.5624	-5.2735
20	-590.62	-708.32	545.38	4.1296	-3.5323	-2.2578
21	87.36	327.66	-325.66	-5.3217	-3.4894	0.1171
22	655.90	391.34	641.73	3.5027	5.7393	0.947
23	35.70	-265.93	196.75	-5.4708	1.2978	2.6842
24	274.52	407.95	-471.59	-3.2451	0.0086	4.1627
25	-311.16	615.55	-691.30	-5.7376	-5.2862	0.1318
26	150.65	591.39	423.48	-4.4309	2.9829	-4.4513
27	-181.19	659.02	128.06	-2.8	2.8342	-1.9115
28	268.43	-347.39	739.44	-4.6474	-5.3978	-2.6595
29	372.08	212.18	740.58	-5.5725	2.6022	-2.262
30	-700.67	-313.47	415.36	3.571	2.422	-2.8766
31	-388.60	423.64	-267.34	1.1884	2.6506	2.8262
32	-737.46	130.50	-378.85	-3.537	5.9034	-0.2801
33	587.05	-515.19	-461.30	-5.674	3.0132	5.4079
34	410.57	739.39	663.38	-4.5391	4.5071	-0.7607
35	571.14	503.14	263.78	-2.4108	2.5939	-4.2892

Table 5.18: Sensor Positions in (m) and Velocities in (m/s) for 3-D Multiple-Time Localization

k	i	x_i	y_i	z_i	\dot{x}_i	\dot{y}_i	\dot{z}_i	k	i	x_i	y_i	z_i	\dot{x}_i	\dot{y}_i	\dot{z}_i
0	1	505.61	-119.53	48.96	-10.16	-3.242	5.2772	10	1	541.08	-72.55	13.28	2.40	0.5033	11.6478
0	2	-69.72	108.41	-226.60	-0.58	1.4588	9.4766	10	2	-40.09	122.74	-261.30	-0.60	9.5683	0.5991
0	3	-192.30	-438.54	56.84	9.73	1.6639	2.4846	10	3	-161.57	-398.25	6.09	-4.25	7.8974	4.824
0	4	-228.49	108.47	-482.84	7.27	-0.9038	-7.9633	10	4	-179.46	148.47	-511.28	-5.40	4.9859	7.9491
1	1	515.33	-113.43	52.12	3.04	11.0651	3.1564	11	1	542.44	-67.08	23.77	-4.96	2.6669	10.4843
1	2	-63.47	112.95	-220.89	4.59	6.2178	5.7046	11	2	-34.33	125.41	-268.51	4.76	4.1969	-7.208
1	3	-187.37	-436.18	48.26	-4.62	2.9445	-8.5852	11	3	-159.74	-388.82	9.47	-3.19	9.0597	3.3817
1	4	-221.16	115.51	-479.10	3.51	9.5334	3.7372	11	4	-178.70	157.74	-505.74	-4.76	7.9962	5.5322
2	1	521.83	-103.55	50.81	-1.31	11.7567	-1.3156	12	1	540.29	-56.86	29.47	-5.00	9.1731	5.6993
2	2	-64.22	112.00	-230.42	1.20	0.1528	-9.5289	12	2	-28.53	132.63	-271.07	2.47	8.922	-2.5626
2	3	-181.73	-427.73	49.01	0.75	10.1273	0.7515	12	3	-159.28	-391.92	19.16	2.99	0.9674	9.687
2	4	-218.08	125.83	-478.05	1.04	10.7221	1.0489	12	4	-180.45	164.50	-497.47	-5.33	4.5002	8.2732
3	1	524.20	-92.07	48.73	-2.04	11.5415	-2.0719	13	1	539.34	-55.95	41.30	-1.31	0.1448	11.8295
3	2	-60.65	120.72	-232.26	-1.81	9.2532	-1.84	13	2	-26.12	140.70	-266.47	-4.04	7.3967	4.6063
3	3	-179.99	-417.95	46.78	-2.18	9.6919	-2.2357	13	3	-160.02	-390.35	29.19	-1.71	0.2966	10.0334
3	4	-218.66	134.06	-485.06	-5.34	6.2831	-7.0105	13	4	-177.11	160.33	-488.06	4.65	2.6384	9.4126
4	1	522.48	-82.59	41.74	-5.66	7.7962	-6.9906	14	1	549.90	-59.95	45.06	3.57	10.715	3.759
4	2	-59.88	127.54	-238.98	-4.80	4.9046	-6.7199	14	2	-26.01	145.60	-258.21	-4.22	2.5044	8.259
4	3	-181.01	-410.28	40.15	-5.03	5.8772	-6.6214	14	3	-163.31	-386.91	38.19	-4.21	2.2258	9.0013
4	4	-220.63	139.56	-494.17	-4.92	3.1501	-9.1136	14	4	-170.77	156.91	-479.97	5.38	4.7864	8.0837
5	1	517.70	-76.57	32.66	-5.87	4.9664	-9.0857	15	1	555.30	-60.32	55.66	4.81	2.4534	10.6047
5	2	-60.46	130.38	-248.14	-2.76	0.8726	-9.159	15	2	-29.08	151.56	-251.33	-4.80	4.6837	6.8759
5	3	-183.46	-405.04	31.77	-4.76	3.2798	-8.384	15	3	-156.27	-390.77	44.46	4.94	6.3248	6.2678
5	4	-213.20	132.17	-496.90	2.64	10.1353	-2.7299	15	4	-161.48	157.41	-474.44	4.76	7.9948	5.5336
6	1	515.12	-75.12	21.13	-2.86	0.7324	-11.5301	16	1	561.97	-56.68	64.82	5.85	4.8592	9.1556
6	2	-56.13	121.92	-246.74	-1.38	9.4026	1.3966	16	2	-27.58	150.25	-241.94	1.95	0.4156	9.3955
6	3	-182.07	-406.17	21.75	1.76	0.3144	-10.0244	16	3	-154.29	-391.04	54.44	1.96	0.3912	9.9853
6	4	-204.92	128.78	-503.00	5.04	7.3917	-6.0949	16	4	-153.68	162.03	-468.53	4.95	7.5925	5.9139
7	1	521.64	-76.12	11.22	5.49	3.6537	-9.9083	17	1	563.12	-55.58	53.02	-1.58	0.2144	-11.7945
7	2	-58.66	124.52	-255.64	-3.36	1.3703	-8.8942	17	2	-23.96	148.86	-233.15	3.55	1.5648	8.7885
7	3	-182.50	-406.00	11.57	-0.46	0.021	-10.1723	17	3	-144.46	-389.70	52.17	-2.22	9.6749	-2.2741
7	4	-198.10	129.68	-511.35	5.31	4.3823	-8.3499	17	4	-146.89	170.09	-466.06	2.41	10.2594	2.4716
8	1	532.54	-74.88	6.59	4.26	10.1051	-4.6248	18	1	565.25	-44.81	48.40	-4.26	10.1082	-4.6208
8	2	-58.39	124.36	-265.24	0.31	0.0101	-9.6006	18	2	-16.61	148.61	-226.97	4.73	5.6324	6.1778
8	3	-172.49	-406.82	9.88	1.67	9.9024	-1.6898	18	3	-137.63	-386.60	59.06	5.07	5.519	6.8913
8	4	-189.09	134.59	-514.80	3.28	9.7196	-3.4571	18	4	-143.63	177.29	-473.45	-5.40	5.7713	-7.395
9	1	542.44	-70.52	1.64	4.51	9.8369	-4.9579	19	1	567.22	-55.57	43.70	4.32	10.0436	-4.7033
9	2	-49.66	122.15	-261.90	-3.13	8.4481	3.3345	19	2	-11.43	151.86	-219.56	4.71	3.8875	7.4112
9	3	-167.35	-405.10	1.26	4.59	2.8857	-8.62	19	3	-129.96	-379.92	58.56	-0.50	10.1577	-0.5053
9	4	-182.17	141.64	-519.23	4.03	9.0186	-4.4203	19	4	-145.22	185.92	-479.78	-5.14	7.1174	-6.3338

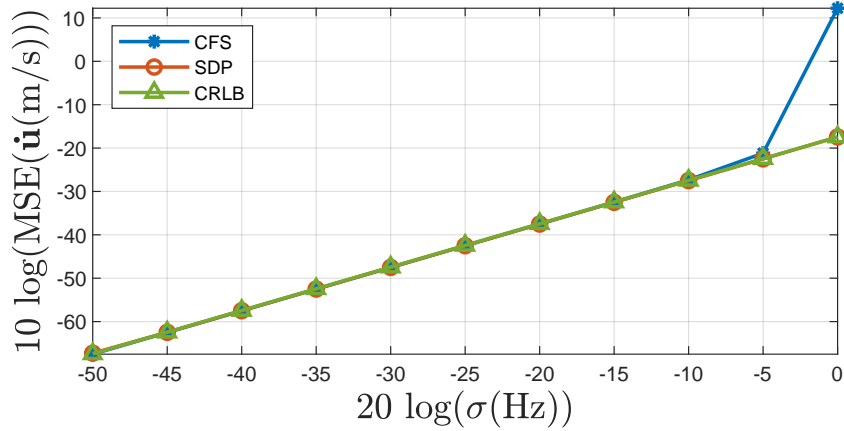
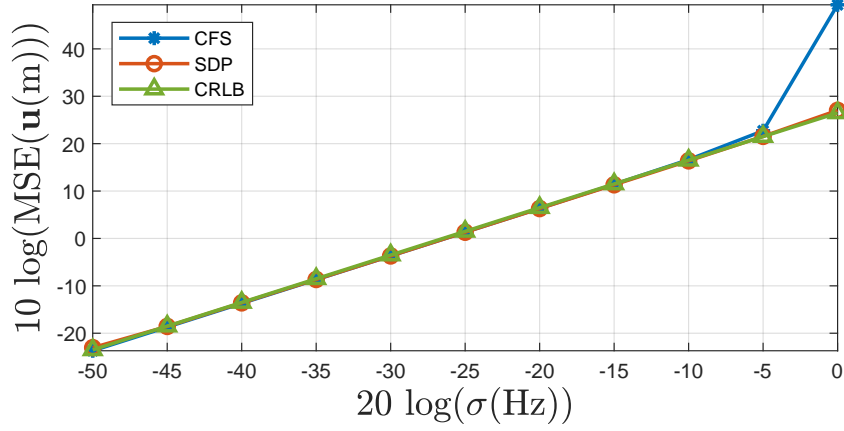


Figure 5.3: Performance of the proposed methods at different σ levels for 3-D single-time case. (a) position estimation, (b) velocity estimation.

5.7 Summary

In this Chapter, we investigated the moving sensors scenario where each sensor moves along nonlinear trajectory with random speed and collects frequency measurement from the object. Incorporating the sensor movement in the frequency measurement

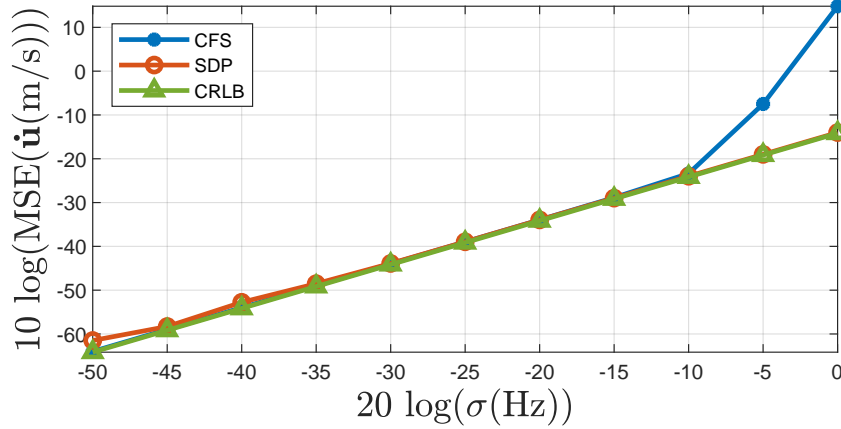
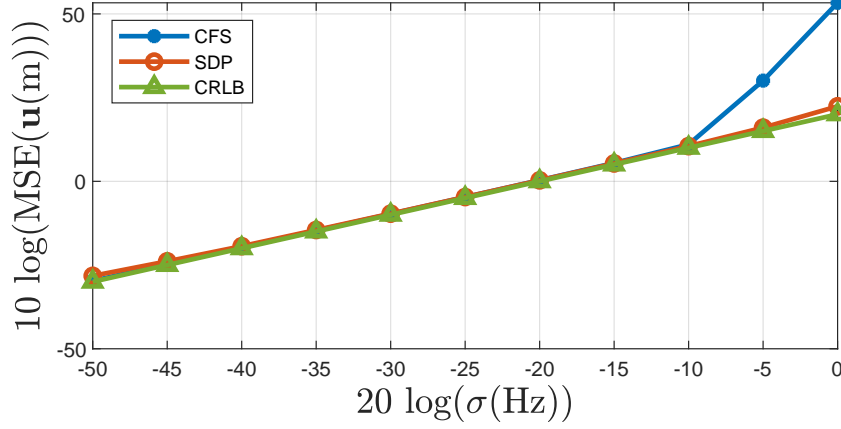


Figure 5.4: Performance of the proposed methods at different σ levels for 3-D multiple-time case. (a) position estimation, (b) velocity estimation.

model results in an increase in the complexity of the model and significant increase in the number of auxiliary variables. The consequence is a large increase in the number of constraints when forming the optimization problem especially in the SDP solution procedure. Also, We have derived the error term associated with the pseudo linear formation including the sensor velocity error which will define later, during the analysis study, additional small noise conditions needed to implement the CFS algorithm.

In addition, we have investigated the sensors movement and found out that the circular movement of the sensors is not appropriate for the pseudo linear formation and can deficient the additional information coming from the sensor movement and makes it vain to reduce the number of required sensors.

Both the closed form solution CFS and the semi-definite programming SDP algorithms derived using the same WLS formulation with a set of constraints. Matlab simulation is used to test the performance of the proposed methods. Both algorithms perform well in reaching the CRLB accuracy. In terms of computational complexity, the CFS estimator is more attractive than the SDP. It is favorable when regular localization geometry such as those having near optimum sensor placement [106] are used. CFS requires additional sensors and low data error to operate and it can only reach the CRLB performance under the small noise conditions specified in Table 5.14. The SDP estimator is much better in handling poorer localization geometry, operating with fewer sensors and working with higher data error.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research investigates the localization of a moving object in position and velocity, by observing the emitted frequency from the object that is subject to the Doppler shift effect at a number of stationary or moving sensors. Previous attempts rely on exhaustive grid search or numerical polynomial optimization to obtain a solution. We proposed a constrained optimization to formulate the localization problem, which enables the problem to be solved efficiently using the linear optimization method to reach a closed-form solution CFS or the semi-definite programming SDP technique to achieve a noise resilient estimate. The CFS has the best computational efficiency among all the other estimators but requires additional sensors and small frequency observation noise. The SDP gives much better performance in handling poorer localization geometry, operating with fewer sensors and working at higher measurement

noise level but needs further processing time and more complicated hardware and software units.

The algorithms are derived for several different scenarios based on the same formulation approach. We started with the 2-D localization, stationary sensors and single-time measurement scenario. Then, we developed the algorithms to multiple-time observations, which helped to reduce the number of sensors required and produce better location accuracy assuming the motion of the object is linear with constant speed. After that, 3-D localization is considered which is more complicated regarding the number of auxiliary variables associated with the pseudo linear formulation and also in handling much more quadratic constraints that relate the independent unknown and the nuisance variables. Later, we considered the sensors are moving along nonlinear trajectories with random speed and derived the CFS and SDP solutions for the 2-D single-time, 2-D multiple-time, 2-D single-time and 3-D multiple-time localization scenarios. The presence of errors in the carrier frequency and the sensor positions are considered. The non-cooperative scenario where the carrier frequency is completely not known is also addressed. Analysis validates the proposed closed-form solution in reaching the Cramer-Rao Lower Bound accuracy under Gaussian noise over the small error region. Simulations support the performance of the proposed solutions.

The proposed CFS algorithm appears to require small frequency measurement errors to operate well. CFS can serve as an effective initialization to the iterative implementation of the ML Estimator, especially in the single-time measurement case where the deviation from the CRLB seems gradual. While the SDP solution gives better performance when the SNR is low or the number of sensors is small, careful

scaling of the input quantities and parameters may be needed to improve the numerical accuracy in which the scaling adjustment could be dependent on the SDP solver used. Incorporating an SDP solver may need certain hardware and software that could be prohibitive in some practical applications. In general, the CFS algorithm should be used under an environment where the localization geometry is typical, the number of sensors is sufficient and the SNR is high, or when the computation complexity is an important factor for consideration. In the situation where the number of sensors is limited or the SNR is low while the complexity is not a crucial factor, the SDP solution is recommended.

Without accounting for the sensor position error by assuming their locations are accurate although they have error, the MSE is nearly 4 dB higher than the CRLB at low noise level and deviates significantly from the bound early when the noise level increases. When accounting for the sensor position error, both CFS and SDP reach the CRLB accuracy and the latter performs well even when the sensor position error is large. When the available carrier frequency has some error but assuming it as accurate in CFS, the performance is highly degraded unless the error is very small and the measurement noise becomes the dominant over the carrier frequency error. The results manifest the importance of taking the error associated with the carrier frequency or the sensor locations into consideration when designing a localization algorithm.

6.2 Future Work

The proposed solutions for the problem of object localization using Doppler shifted frequency measurements can be further investigated to improve the performance and accommodate other localization scenarios. Additional research can be done on improving the noise resilient ability of the algebraic solution, reducing the number of constraints for the SDP algorithm, and better handling the sensor position and velocity errors. The problem of optimum sensor placement is another interesting topic that can help to reduce the required number of sensors when the measurement noise level is high. The mobility pattern of the sensor can be investigated to draw a clear picture of its effect on the localization ability and accuracy.

The problem of object localization using multiple-time measurements investigated in this research is a special case of object tracking under a linear motion constant velocity model. The proposed algorithm may be applicable for other motion pattern of the object if the observation period is short enough that constant velocity model holds approximately within. Our future work is to extend the proposed algorithms to the more complex object tracking scenario where the motion is nonlinear and the velocity is not constant. One approach is to incorporate a disturbance term in the object trajectory model to account for the position variations and apply a dynamic velocity model, such as a constant acceleration model, to describe the velocity changes over time.

Appendix A

The partial derivatives in (2.41) are:

$$\frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} = \frac{\partial}{\partial \boldsymbol{\theta}^{oT}} [\mathbf{f}_0^{oT}, \mathbf{f}_1^{oT}, \dots, \mathbf{f}_{N-1}^{oT}]^T, \quad (\text{A.1a})$$

$$\frac{\partial \mathbf{f}_k^o}{\partial \boldsymbol{\theta}^{oT}} = \frac{\partial}{\partial \boldsymbol{\theta}^{oT}} [f_{k,1}^o, f_{k,2}^o, \dots, f_{k,M}^o]^T, \quad (\text{A.1b})$$

$$\frac{\partial f_{k,i}^o}{\partial \boldsymbol{\theta}^{oT}} = \left[\frac{\partial f_{k,i}^o}{\partial \mathbf{u}^{oT}}, \frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{u}}^{oT}} \right], \quad (\text{A.1c})$$

$$\frac{\partial f_{k,i}^o}{\partial \mathbf{u}^{oT}} = \frac{-f_o^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|} \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o}, \quad (\text{A.1d})$$

$$\frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{u}}^{oT}} = \frac{-f_o^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|} (k \dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o} + (\mathbf{u}_k^o - \mathbf{s}_i^o)^T). \quad (\text{A.1e})$$

$$\frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}} = \frac{\partial}{\partial \boldsymbol{\alpha}^{oT}} [\mathbf{f}_0^{oT}, \mathbf{f}_1^{oT}, \dots, \mathbf{f}_{N-1}^{oT}]^T, \quad (\text{A.2a})$$

$$\frac{\partial \mathbf{f}_k^o}{\partial \boldsymbol{\alpha}^{oT}} = \frac{\partial}{\partial \boldsymbol{\alpha}^{oT}} [f_{k,1}^o, f_{k,2}^o, \dots, f_{k,M}^o]^T, \quad (\text{A.2b})$$

$$\frac{\partial f_{k,i}^o}{\partial \boldsymbol{\alpha}^{oT}} = \left[\frac{\partial f_{k,i}^o}{\partial f_o^o}, \frac{\partial f_{k,i}^o}{\partial \mathbf{s}_i^{oT}} \right], \quad (\text{A.2c})$$

$$\frac{\partial f_{k,i}^o}{\partial f_o^o} = 1 - \frac{(\mathbf{u}_k^o - \mathbf{s}_i^o)^T \dot{\mathbf{u}}^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|}, \quad (\text{A.2d})$$

$$\frac{\partial f_{k,i}^o}{\partial \mathbf{s}^{oT}} = \frac{f_o^o}{c} \left[\mathbf{0}_{d(i-1)}^T, \frac{\dot{\mathbf{u}}^{oT} \mathbf{P}_{k,i}^{\perp o}}{\|\mathbf{u}_k^o - \mathbf{s}_i^o\|}, \mathbf{0}_{d(N-i)}^T \right], \quad (\text{A.2e})$$

where d is the dimension of localization and the matrix $\mathbf{P}_{k,i}^{\perp o}$ is defined below (3.3).

\mathbf{b}_{f_o} and \mathbf{B}_s in (2.36) are

$$\mathbf{b}_{f_o} = \frac{\partial f_o^o}{\partial \boldsymbol{\phi}^{oT}} = [\mathbf{0}_{2d}^T, 1, \mathbf{0}_{dM}^T], \quad (\text{A.3a})$$

$$\mathbf{B}_s = \frac{\partial \mathbf{s}}{\partial \boldsymbol{\phi}^{oT}} = [\mathbf{0}_{dM \times (2d+1)}, \mathbf{I}_{dM}]. \quad (\text{A.3b})$$

Appendix B

From (4.20), (4.29) and representing an estimated value (\cdot) by $(\cdot)^o + \Delta(\cdot)$, it is direct to validate the following relations for the elements of $\tilde{\mathbf{h}}$ in (4.30) after dropping the second order error terms,

$$\begin{aligned} \tilde{h}(4) &= \varphi(1)\tilde{\varphi}^o(1) + \varphi(2)\tilde{\varphi}^o(2) + \varphi(3)\tilde{\varphi}^o(3) - \varphi^o(1)\Delta\varphi(1) \\ &\quad - \varphi^o(2)\Delta\varphi(2) - \varphi^o(3)\Delta\varphi(3) + \Delta\varphi(4), \end{aligned} \tag{B.1a}$$

$$\begin{aligned} \tilde{h}(8) &= \frac{1}{2}\varphi(6)\tilde{\varphi}^o(4) + \frac{1}{2}\varphi(5)\tilde{\varphi}^o(5) - \frac{1}{2}\varphi^o(6)\Delta\varphi(5) \\ &\quad - \frac{1}{2}\varphi^o(5)\Delta\varphi(6) + 2\varphi(8)\Delta\varphi(8) - \Delta\varphi(8)^2, \end{aligned} \tag{B.1b}$$

$$\begin{aligned} \tilde{h}(9) &= \frac{1}{2}\varphi(7)\tilde{\varphi}^o(4) + \frac{1}{2}\varphi(5)\tilde{\varphi}^o(6) - \frac{1}{2}\varphi^o(7)\Delta\varphi(5) \\ &\quad - \frac{1}{2}\varphi^o(5)\Delta\varphi(7) + 2\varphi(9)\Delta\varphi(9) - \Delta\varphi(9)^2, \end{aligned} \tag{B.1c}$$

$$\begin{aligned} \tilde{h}(10) &= \frac{1}{2}\varphi(7)\tilde{\varphi}^o(5) + \frac{1}{2}\varphi(6)\tilde{\varphi}^o(6) - \frac{1}{2}\varphi^o(7)\Delta\varphi(6) \\ &\quad - \frac{1}{2}\varphi^o(6)\Delta\varphi(7) + 2\varphi(10)\Delta\varphi(10) - \Delta\varphi(10)^2, \end{aligned} \tag{B.1d}$$

$$\begin{aligned} \tilde{h}(11) &= \varphi(8)\tilde{\varphi}^o(2) + \varphi(9)\tilde{\varphi}^o(3) + \varphi(1)\tilde{\varphi}^o(4) - \varphi^o(5)\Delta\varphi(1) \\ &\quad - \varphi^o(2)\Delta\varphi(8) - \varphi^o(3)\Delta\varphi(9) + \Delta\varphi(11), \end{aligned} \tag{B.1e}$$

$$\begin{aligned}\tilde{h}(12) &= \varphi(8)\tilde{\varphi}^o(1) + \varphi(10)\tilde{\varphi}^o(3) + \varphi(2)\tilde{\varphi}^o(5) - \varphi^o(6)\Delta\varphi(2) \\ &\quad - \varphi^o(1)\Delta\varphi(8) - \varphi^o(3)\Delta\varphi(10) + \Delta\varphi(12),\end{aligned}\tag{B.1f}$$

$$\begin{aligned}\tilde{h}(13) &= \varphi(9)\tilde{\varphi}^o(1) + \varphi(10)\tilde{\varphi}^o(2) + \varphi(3)\tilde{\varphi}^o(6) - \varphi^o(7)\Delta\varphi(3) \\ &\quad - \varphi^o(1)\Delta\varphi(9) - \varphi^o(2)\Delta\varphi(10) + \Delta\varphi(13),\end{aligned}\tag{B.1g}$$

$$\begin{aligned}\tilde{h}(14) &= \varphi(11)\tilde{\varphi}^o(1) + \varphi(12)\tilde{\varphi}^o(2) + \varphi(13)\tilde{\varphi}^o(3) - \varphi^o(1)\Delta\varphi(11) \\ &\quad - \varphi^o(2)\Delta\varphi(12) - \varphi^o(3)\Delta\varphi(13) + \Delta\varphi(14).\end{aligned}\tag{B.1h}$$

Appendix C

From the definition of φ^o in (4.44), it is simple to validate the following relations, which form constraints to the WLS problem for multiple-time scenario:

$$\varphi(7) = \varphi(1)^2 + \varphi(2)^2 + \varphi(3)^2, \quad (\text{C.1a})$$

$$\varphi(8) = \varphi(1)\varphi(4) + \varphi(2)\varphi(5) + \varphi(3)\varphi(6), \quad (\text{C.1b})$$

$$\varphi(9) = \varphi(4)^2, \quad (\text{C.1c})$$

$$\varphi(10) = \varphi(5)^2, \quad (\text{C.1d})$$

$$\varphi(11) = \varphi(6)^2, \quad (\text{C.1e})$$

$$\varphi(12) = \varphi(4)\varphi(5), \quad (\text{C.1f})$$

$$\varphi(13) = \varphi(4)\varphi(6), \quad (\text{C.1g})$$

$$\varphi(14) = \varphi(5)\varphi(6), \quad (\text{C.1h})$$

$$\begin{aligned} \varphi(15) &= \varphi(1)\varphi(9) + \varphi(2)\varphi(12) + \varphi(3)\varphi(13) \\ &= \varphi(4)\varphi(8), \end{aligned} \quad (\text{C.1i})$$

$$\begin{aligned}\varphi(16) &= \varphi(1)\varphi(12) + \varphi(2)\varphi(10) + \varphi(3)\varphi(14) \\ &= \varphi(5)\varphi(8),\end{aligned}\tag{C.1j}$$

$$\begin{aligned}\varphi(17) &= \varphi(1)\varphi(13) + \varphi(2)\varphi(14) + \varphi(3)\varphi(11) \\ &= \varphi(6)\varphi(8),\end{aligned}\tag{C.1k}$$

$$\varphi(18) = \varphi(4)\varphi(9) + \varphi(5)\varphi(12) + \varphi(6)\varphi(13),\tag{C.1l}$$

$$\varphi(19) = \varphi(4)\varphi(12) + \varphi(5)\varphi(10) + \varphi(6)\varphi(14),\tag{C.1m}$$

$$\varphi(20) = \varphi(4)\varphi(13) + \varphi(5)\varphi(14) + \varphi(6)\varphi(11),\tag{C.1n}$$

$$\varphi(21) = \varphi(4)\varphi(18) + \varphi(5)\varphi(19) + \varphi(6)\varphi(20),\tag{C.1o}$$

$$\begin{aligned}\varphi(22) &= \varphi(1)\varphi(18) + \varphi(2)\varphi(19) + \varphi(3)\varphi(20) \\ &= \varphi(4)\varphi(15) + \varphi(5)\varphi(16) + \varphi(6)\varphi(17),\end{aligned}\tag{C.1p}$$

$$\varphi(23) = \varphi(1)\varphi(15) + \varphi(2)\varphi(16) + \varphi(3)\varphi(17).\tag{C.1q}$$

Appendix D

This Appendix examines the complexity of the algebraic and SDP solutions for multiple-time scenario, when the carrier frequency is available. The single-time scenario is a special case where the number of successive time observations N is 1. The study concentrates on the batch processing for multiple-time case. The complexity difference with the sequential version is summarized at the end.

D.1 Algebraic Solution

The algebraic CFS involves one least-squares computation for initializing the first stage weighting matrix and two WLS calculations from the two stages. The overall complexity is dominated by the evaluation of the weighting matrix (4.25). Table D.1 lists the detailed operations and the associated computational costs in terms of the number of multiplications for the algebraic solution [113, 114]. When the second stage is repeated $\eta - 1$ times ($\eta \geq 1$) to improve $\widetilde{\mathbf{W}}$, The computational complexity is

$$\begin{aligned}
& M^3N^3 + 3M^3N^2 + 18M^3N + (2L + 3)M^2N^2 + 3M^2N \\
& + (2L^2 + 5L + 1)MN + (2/3 + 4\eta)L^3 + (4 + \eta(1 + 2K))L^2 \\
& + \eta(K + 3)LK + \eta(K/3 + 2)K^2, \quad (\text{D.1})
\end{aligned}$$

where M is the number of sensors, N is the number of measurements in consecutive time instants, L is the length of the pseudo unknown vector $\boldsymbol{\varphi}$ in the first stage and K is the length of the unknown vector $\tilde{\boldsymbol{\varphi}}$ that is equal to 6.

D.2 SDP Solution

The SDP solution requires the formation of the objective function (4.40) and the solution of the relaxed SDP (4.60). Also, (4.60) is solved twice, first with \mathbf{W} set to the identity matrix to obtain an initial solution and second with \mathbf{W} formed by the initial solution to obtain the final solution. Generating the objective function for the first SDP solution uses the operations in entries 1-2 of Table D.1 and that for the final SDP solution needs entries 4-9. There are extra MN multiplications in entries 2 and 9 due to the factor 2 in (4.40). Obtaining the objective functions take the computational cost of

$$\begin{aligned}
& M^3N^3 + 3M^3N^2 + 18M^3N + (2L + 3)M^2N^2 \\
& + 3M^2N + (2L^2 + 6L + 3)MN. \quad (\text{D.2})
\end{aligned}$$

The worst-case complexity of solving an SDP is [110]

$$O\left(\sqrt{\mu}\left(m^3 + m^2 \sum_{j=1}^{N_{sd}} (n_j^{sd})^2 + m \sum_{j=1}^{N_{sd}} (n_j^{sd})^3\right) \ln(1/\epsilon)\right), \quad (\text{D.3})$$

where m is the number of equality constraints, N_{sd} is the number of semidefinite cone constraints, n_j^{sd} is the dimension of the j -th semidefinite cone, μ is the barrier parameter for measuring the geometric complexities of the cones involved,

$$\mu = \sum_{j=1}^{N_{sd}} n_j^{sd}, \quad (\text{D.4})$$

and $\epsilon > 0$ is the solution precision. The relaxed SDP (4.60) has $C + 1$ linear equality constraints and one semidefinite cone constraint of size $L + 1$ (hence $N_{sd} = 1$ and $n_1^{sd} = L + 1$). Thus, the worst-case complexity is on the order of

$$O\left(\sqrt{L}(C^3 + C^2L^2 + CL^3) \ln(1/\epsilon)\right). \quad (\text{D.5})$$

The proposed SDP solution calls an SDP solver twice and the amount in (D.5) needs to be doubled.

D.3 Sequential Estimation

The sequential multiple-time algorithm requires an extra step to update the solution by the object location estimate from the current single time measurement. Table D.2 lists the extra computational cost of this additional step.

Table D.1: Computational Costs of the Algebraic Solution CFS

Operations (LS)	Computational Costs
$\mathbf{A}^T \mathbf{A}$	$MNL(L+1)$
$\mathbf{A}^T \mathbf{h}$	$2MNL$
$(\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{h})$	$L^3/3 + 2L^2$
Operations (First WLS)	Computational Costs
$\sigma_{f_o}^2 \mathbf{d}_f \mathbf{d}_f^T$	$(M^2 N^2 + MN) + 2MN(3M)^2$
$+ \mathbf{D}_s \mathbf{Q}_s \mathbf{D}_s^T + \mathbf{Q}_n$	$+ 3M^2 N(MN + 1)$
$\mathbf{Q}_\varepsilon^{-1}$	$M^3 N^3$
$\mathbf{B}^{-1} \mathbf{Q}_\varepsilon^{-1} \mathbf{B}^{-1}$	$2M^2 N^2$
$\mathbf{A}^T \mathbf{W}$	$2LM^2 N^2$
$(\mathbf{A}^T \mathbf{W}) \mathbf{A}$	$MNL(L+1)$
$(\mathbf{A}^T \mathbf{W}) \mathbf{h}$	$2MNL$
$(\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{W} \mathbf{h})$	$L^3/3 + 2L^2$
Operations (Second WLS)	Computational Costs
$\tilde{\mathbf{B}}^{-1}$	L^3
$\tilde{\mathbf{B}}^{-1} (\mathbf{A}^T \mathbf{W} \mathbf{A}) \tilde{\mathbf{B}}^{-1}$	$2L^3 + (L^3 + L^2)$
$\tilde{\mathbf{A}}^T \tilde{\mathbf{W}}$	$2KL^2$
$(\tilde{\mathbf{A}}^T \tilde{\mathbf{W}}) \tilde{\mathbf{A}}$	$LK(K+1)$
$(\tilde{\mathbf{A}}^T \tilde{\mathbf{W}}) \tilde{\mathbf{h}}$	$2KL$
$(\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}})^{-1} (\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{h}})$	$K^3/3 + 2K^2$

Table D.2: Extra Computational Costs for the Solution Update of Sequential Process

Operations	Computational Costs
$\mathbf{H}_k \boldsymbol{\theta}_{\{k-1\}}$	$K/2$
$\tilde{\mathbf{B}}^{-1}$	L^3
$\tilde{\mathbf{B}}^{-1} (\mathbf{A}^T \mathbf{W} \mathbf{A}) \tilde{\mathbf{B}}^{-1}$	$2L^3 + (L^3 + L^2)$
$\tilde{\mathbf{A}}^T \tilde{\mathbf{W}} \tilde{\mathbf{A}}$	$2KL^2 + LK(K+1)$
$\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}}$	$K^2/2$
$(\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}}) \mathbf{H}_{\{k\}}$	$K^2/2$
$(\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}}) [\boldsymbol{\theta}_k^T \ \boldsymbol{\theta}_{\{k-1\}}^T]^T$	$2K(2K)$
$(\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}} \mathbf{H}_{\{k\}})^{-1}$	$K^3/3 + 2K^2$
$\cdot (\mathbf{H}_{\{k\}}^T \mathbf{W}_{\{k\}} [\boldsymbol{\theta}_k^T \ \boldsymbol{\theta}_{\{k-1\}}^T]^T)$	

Appendix E

The partial derivatives in (5.65) are:

$$\frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\theta}^{oT}} = \frac{\partial}{\partial \boldsymbol{\theta}^{oT}} [\mathbf{f}_0^{oT}, \mathbf{f}_1^{oT}, \dots, \mathbf{f}_{N-1}^{oT}]^T, \quad (\text{E.1a})$$

$$\frac{\partial \mathbf{f}_k^o}{\partial \boldsymbol{\theta}^{oT}} = \frac{\partial}{\partial \boldsymbol{\theta}^{oT}} [f_{k,1}^o, f_{k,2}^o, \dots, f_{k,M}^o]^T, \quad (\text{E.1b})$$

$$\frac{\partial f_{k,i}^o}{\partial \boldsymbol{\theta}^{oT}} = \left[\frac{\partial f_{k,i}^o}{\partial \mathbf{u}^{oT}}, \frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{u}}^{oT}} \right], \quad (\text{E.1c})$$

$$\frac{\partial f_{k,i}^o}{\partial \mathbf{u}^{oT}} = \frac{-f_o^o}{c \|\mathbf{u}_k^o - \mathbf{s}_{k,i}^o\|} (\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)^T \mathbf{P}_{k,i}^{\perp o}, \quad (\text{E.1d})$$

$$\frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{u}}^{oT}} = \frac{-f_o^o}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|} (k(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)^T \mathbf{P}_{k,i}^{\perp o} + (\mathbf{u}_k^o - \mathbf{s}_i^o)^T). \quad (\text{E.1e})$$

$$\frac{\partial \mathbf{f}^o}{\partial \boldsymbol{\alpha}^{oT}} = \frac{\partial}{\partial \boldsymbol{\alpha}^{oT}} [\mathbf{f}_0^{oT}, \mathbf{f}_1^{oT}, \dots, \mathbf{f}_{N-1}^{oT}]^T, \quad (\text{E.2a})$$

$$\frac{\partial \mathbf{f}_k^o}{\partial \boldsymbol{\alpha}^{oT}} = \frac{\partial}{\partial \boldsymbol{\alpha}^{oT}} [f_{k,1}^o, f_{k,2}^o, \dots, f_{k,M}^o]^T, \quad (\text{E.2b})$$

$$\frac{\partial f_{k,i}^o}{\partial \boldsymbol{\alpha}^{oT}} = \left[\frac{\partial f_{k,i}^o}{\partial f_o^o}, \frac{\partial f_{k,i}^o}{\partial \mathbf{s}^{oT}}, \frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{s}}^{oT}} \right], \quad (\text{E.2c})$$

$$\frac{\partial f_{k,i}^o}{\partial f_o^o} = 1 - \frac{(\mathbf{u}_k^o - \mathbf{s}_i^o)^T (\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)}{c \|\mathbf{u}_k^o - \mathbf{s}_i^o\|}, \quad (\text{E.2d})$$

$$\frac{\partial f_{k,i}^o}{\partial \mathbf{s}^o T} = \frac{f_o^o}{c} \left[\mathbf{0}_{d((i+kM)-1)}^T, \frac{(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)^T \mathbf{P}_{k,i}^{\perp o}}{\|\mathbf{u}_k^o - \mathbf{s}_i^o\|}, \mathbf{0}_{d(NM-(i+kM))}^T \right], \quad (\text{E.2e})$$

$$\frac{\partial f_{k,i}^o}{\partial \dot{\mathbf{s}}^o T} = \frac{f_o^o}{c} \left[\mathbf{0}_{d((i+kM)-1)}^T, \frac{k(\dot{\mathbf{u}}^o - \dot{\mathbf{s}}_{k,i}^o)^T \mathbf{P}_{k,i}^{\perp o} + (\mathbf{u}_k^o - \mathbf{s}_i^o)^T}{\|\mathbf{u}_k^o - \mathbf{s}_i^o\|}, \mathbf{0}_{d(NM-(i+kM))}^T \right], \quad (\text{E.2f})$$

where d is the dimension of localization.

Bibliography

- [1] L. Yang, L. Yang, and K. C. Ho, “Moving target localization in multistatic sonar by differential delays and doppler shifts,” *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1160–1164, Sep. 2016.
- [2] A. Beck, P. Stoica, and J. Li, “Exact and approximate solutions of source localization problems,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, 2008.
- [3] D. C. Torney, “Localization and observability of aircraft via doppler shifts,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 1163–1168, 2007.
- [4] Y. Xiao, P. Wei, and T. Yuan, “Observability and performance analysis of bi/multi-static doppler-only radar,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 1654–1667, 2010.
- [5] Z. Li, S. E. Dosso, and D. Sun, “Motion-compensated acoustic localization for underwater vehicles,” *IEEE J. Ocean. Eng.*, vol. 41, no. 4, pp. 840–851, 2016.

- [6] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, “Direct localization for massive mimo,” *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, 2017.
- [7] B. Wang and Y. Tian, “Distributed network localization: Accurate estimation with noisy measurement and communication information,” *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5927–5940, 2018.
- [8] K. Dogancay, “Uav path planning for passive emitter localization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 2, pp. 1150–1166, 2012.
- [9] K. C. Ho and W. Xu, “An accurate algebraic solution for moving source location using tdoa and fdoa measurements,” *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2453–2463, Sep. 2004.
- [10] K. C. Ho, X. Lu, and L. Kovavisaruch, “Source localization using tdoa and fdoa measurements in the presence of receiver location errors: Analysis and solution,” *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 684–696, Feb. 2007.
- [11] K. J. Cameron and D. J. Bates, “Geolocation with fdoa measurements via polynomial systems and ransac,” in *Proc. IEEE Radar Conf. (RadarConf18)*, Boston, Apr. 2018, pp. 0676–0681.
- [12] Y. T. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location,” *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.

- [13] Y. Weng, W. Xiao, and L. Xie, "Total least squares method for robust source localization in sensor networks using tdoa measurements," *Int. J. Distributed Sensor Networks*, vol. 2011, pp. 1–8, Nov. 2011.
- [14] K. C. Ho, L. Kovavisaruch, and H. Parikh, "Source localization using tdoa with erroneous receiver positions," in *Proc. IEEE Int. Symp. Circuits Syst., Vancouver*, vol. 3, May 2004, pp. 453–456.
- [15] G. Wang, "A semidefinite relaxation method for energy-based source localization in sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2293–2301, Jun. 2011.
- [16] K. Dogancay, "Bias compensation for the bearings-only pseudolinear target track estimator," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 59–68, 2006.
- [17] Yiu-Tong Chan, H. Yau Chin Hang, and Pak-chung Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 10–16, 2006.
- [18] Y. Wang and K. C. Ho, "Tdoa positioning irrespective of source range," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1447–1460, 2017.
- [19] L. A. Romero and J. Mason, "Evaluation of direct and iterative methods for overdetermined systems of toa geolocation equations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1213–1229, 2011.
- [20] K. Yang, J. An, X. Bu, and G. Sun, "Constrained total least-squares location algorithm using time-difference-of-arrival measurements," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1558–1562, 2010.

- [21] L. Rui and K. C. Ho, “Algebraic solution for joint localization and synchronization of multiple sensor nodes in the presence of beacon uncertainties,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 9, pp. 5196–5210, 2014.
- [22] L. Yang and K. C. Ho, “An approximately efficient tdoa localization algorithm in closed-form for locating multiple disjoint sources with erroneous sensor positions,” *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4598–4615, 2009.
- [23] K. C. Ho, “Bias reduction for an explicit solution of source localization using tdoa,” *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2101–2114, 2012.
- [24] L. Rui and K. C. Ho, “Efficient closed-form estimators for multistatic sonar localization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 600–614, 2015.
- [25] Z. Fei, Y. Tiejun, and H. Shunji, “Toa estimation algorithm based on multi-search,” *J. Syst. Eng. Electron.*, vol. 16, no. 3, pp. 561–565, 2005.
- [26] Y. Xie, Y. Wang, P. Zhu, and X. You, “Grid-search-based hybrid toa/aoa location techniques for nlos environments,” *IEEE Commun. Lett.*, vol. 13, no. 4, pp. 254–256, 2009.
- [27] Kaiqiang Liang, Zhen Huang, and Jiazhi He, “A passive localization method of single satellite using toa sequence,” in *2016 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, 2016, pp. 1795–1798.
- [28] K. Yang, G. Wang, and Z.-Q. Luo, “Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals,” *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2775–2784, Jul. 2009.

- [29] E. Xu, Z. Ding, and S. Dasgupta, “Reduced complexity semidefinite relaxation algorithms for source localization based on time difference of arrival,” *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1276–1282, Sep. 2011.
- [30] K. W. K. Lui, F. K. W. Chan, and H. C. So, “Semidefinite programming approach for range-difference based source localization,” *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1630–1633, 2009.
- [31] G. Wang, A. M. So, and Y. Li, “Robust convex approximation methods for tdoa-based localization under nlos conditions,” *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3281–3296, Jul. 2016.
- [32] Z. Wang, J. Luo, and X. Zhang, “A novel location-penalized maximum likelihood estimator for bearing-only target localization,” *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6166–6181, 2012.
- [33] Y. Wang and K. C. Ho, “An asymptotically efficient estimator in closed-form for 3-d aoa localization using a sensor network,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6524–6535, 2015.
- [34] Y. Sun, K. C. Ho, and Q. Wan, “Eigenspace solution for aoa localization in modified polar representation,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2256–2271, 2020.
- [35] A. N. Bishop, B. D. O. Anderson, B. Fidan, P. N. Pathirana, and G. Mao, “Bearing-only localization using geometrically constrained optimization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 1, pp. 308–320, 2009.

- [36] I. Shames, A. N. Bishop, and B. D. O. Anderson, “Analysis of noisy bearing-only network localization,” *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 247–252, 2013.
- [37] R. W. Ouyang, A. K. Wong, and C. Lea, “Received signal strength-based wireless localization via semidefinite programming: Noncooperative and cooperative schemes,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1307–1318, 2010.
- [38] H. Ketabalian, M. Biguesh, and A. Sheikhi, “A closed-form solution for localization based on rss,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 912–923, 2020.
- [39] Z. Wang, H. Zhang, T. Lu, and T. A. Gulliver, “Cooperative rss-based localization in wireless sensor networks using relative error estimation and semidefinite programming,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 483–497, 2019.
- [40] Y. Wang and K. C. Ho, “Unified near-field and far-field localization for aoa and hybrid aoa-tdoa positionings,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1242–1254, 2018.
- [41] S. Tomic, M. Beko, and R. Dinis, “3-d target localization in wireless sensor networks using rss and aoa measurements,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3197–3210, 2017.
- [42] M. McGuire, K. N. Plataniotis, and A. N. Venetsanopoulos, “Data fusion of power and time measurements for mobile terminal location,” *IEEE Trans. Mobile Comput.*, vol. 4, no. 2, pp. 142–153, 2005.

- [43] G. Wang, Y. Li, and N. Ansari, “A semidefinite relaxation method for source localization using tdoa and fdoa measurements,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 853–862, Feb. 2013.
- [44] Y. Wang and Y. Wu, “An efficient semidefinite relaxation algorithm for moving source localization using tdoa and fdoa measurements,” *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 80–83, Jan. 2017.
- [45] J. T. Broad and L. M. Savage, “Frequency difference of arrival (fdoa) for geolocation,” Jul. 11 2017, US Patent 9,702,960.
- [46] S. M. Nejad and S. Olyaei, “Accuracy improvement by nonlinearity reduction in two-frequency laser heterodyne interferometer,” in *Proc. 13th IEEE Int. Conf. Electron., Circuits Syst. (ICECS), Nice*, Dec. 2006, pp. 914–917.
- [47] O. O. Bezvesilniy, V. V. Vynogradov, and D. M. Vavriv, “High-accuracy doppler measurements for airborne sar applications,” in *Proc. European Radar Conf. EuRAD2008, Amsterdam*, Oct. 2008, pp. 29–32.
- [48] S. Bi, X. Gao, V. M. Lubecke, O. B. Lubecke, D. Matthews, and X. L. Liu, “A multi-arc method for improving doppler radar motion measurement accuracy,” in *Proc. IEEE/MTT-S Int. Microw. Symposium - IMS*, Jun. 2018, pp. 244–247.
- [49] X. Gao, J. Xu, A. Rahman, V. Lubecke, and O. Boric-Lubecke, “Arc shifting method for small displacement measurement with quadrature cw doppler radar,” in *Proc. IEEE MTT-S Int. Microw Symposium (IMS)*, Jun. 2017, pp. 1003–1006.

- [50] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, “A survey of motion planning and control techniques for self-driving urban vehicles,” *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [51] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *IEEE Trans. Syst., Man, Cybern. Part C*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.
- [52] U. Ritzinger, J. Puchinger, and R. F. Hartl, “A survey on dynamic and stochastic vehicle routing problems,” *Int. J. Production Research*, vol. 54, no. 1, pp. 215–231, 2016.
- [53] Y. Xue, W. Su, H. Wang, D. Yang, and J. Ma, “A model on indoor localization system based on the time difference without synchronization,” *IEEE Access*, vol. 6, pp. 34 179–34 189, 2018.
- [54] A. Catovic and Z. Sahinoglu, “The cramer-rao bounds of hybrid toa/rss and tdoa/rss location estimation schemes,” *IEEE Communications Letters*, vol. 8, no. 10, pp. 626–628, 2004.
- [55] J. Mason and L. Romero, “Toa/foa geolocation solutions using multivariate resultants,” *Navigation*, vol. 52, no. 3, pp. 163–177, 2005.
- [56] J. Li, F. Guo, and W. Jiang, “A linear-correction least-squares approach for geolocation using fdoa measurements only,” *Chinese Journal of Aeronautics*, vol. 25, no. 5, pp. 709–714, May 2012.
- [57] S. Stein, “Algorithms for ambiguity function processing,” *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 29, no. 3, pp. 588–599, Jun. 1981.

- [58] Y.-T. Chan and K. Ho, “Joint time-scale and tdoa estimation: analysis and fast approximation,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2625–2634, 2005.
- [59] Z. Gong, C. Li, F. Jiang, and J. Zheng, “Auv-aided localization of underwater acoustic devices based on doppler shift measurements,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2226–2239, 2020.
- [60] X. Zhong, A. B. Premkumar, and W. Wang, “Direction of arrival tracking of an underwater acoustic source using particle filtering: Real data experiments,” in *IEEE 2013 Tencon - Spring*, 2013, pp. 420–424.
- [61] P. Tarrío, A. M. Bernardos, and J. R. Casar, “Weighted least squares techniques for improved received signal strength based localization,” *Sensors*, vol. 11, no. 9, pp. 8569–8592, 2011.
- [62] N. J. Giordano, *College physics: reasoning and relationships*. Brooks/Cole, Cengage Learning, 2010.
- [63] J. J. O’Connor and E. F. Robertson, “Christian andreas doppler,” *MacTutor History of Mathematics archive*. University of St Andrews, 1998.
- [64] J. Genyuk, “Christian doppler,” Jan 2009.
- [65] S. Weinberg, “The first three minutes. a modern view of the origin of the universe,” *Bantam, New York*, 1977.
- [66] V. L. Pisacane, “The legacy of transit: Guest editor’s introduction,” 1998.

- [67] B. Soltanian, A. M. Demirtas, M. Renfors *et al.*, “Reduced-complexity fft-based method for doppler estimation in gnss receivers,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 143, 2014.
- [68] Y. Zhang, M. G. Amin, and F. Ahmad, “Narrowband frequency-hopping radars for the range estimation of moving and vibrating targets,” in *Radar Sensor Technol. XII*, vol. 6947. International Society for Optics and Photonics, 2008, p. 694709.
- [69] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, “The challenges of building mobile underwater wireless networks for aquatic applications,” *Ieee Network*, vol. 20, no. 3, pp. 12–18, 2006.
- [70] X. Lurton, *An introduction to underwater acoustics: principles and applications*. Springer Science & Business Media, 2002.
- [71] D. B. Kilfoyle and A. B. Baggeroer, “The state of the art in underwater acoustic telemetry,” *IEEE Journal of oceanic engineering*, vol. 25, no. 1, pp. 4–27, 2000.
- [72] Z. Gong, C. Li, F. Jiang, R. Su, R. Venkatesan, C. Meng, S. Han, Y. Zhang, S. Liu, and K. Hao, “Design, analysis, and field testing of an innovative drone-assisted zero-configuration localization framework for wireless sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 322–10 335, 2017.
- [73] X. Su, I. Ullah, X. Liu, and D. Choi, “A review of underwater localization techniques, algorithms, and challenges,” *Journal of Sensors*, vol. 2020, 2020.

- [74] E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela, “Mimo radar: An idea whose time has come,” in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No. 04CH37509)*. IEEE, 2004, pp. 71–78.
- [75] A. M. Haimovich, R. S. Blum, and L. J. Cimini, “Mimo radar with widely separated antennas,” *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2007.
- [76] T. Tirer and A. J. Weiss, “High resolution localization of narrowband radio emitters based on doppler frequency shifts,” *Signal Process.*, vol. 141, pp. 288–298, 2017.
- [77] J. Yin, D. Wang, Y. Wu, and R. Liu, “Direct localization of multiple stationary narrowband sources based on angle and doppler,” *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2630–2633, Dec. 2017.
- [78] O. Bar-Shalom and A. J. Weiss, “Emitter geolocation using single moving receiver,” *Signal Process.*, vol. 105, pp. 70–83, Dec. 2014.
- [79] B. H. Lee, Y. T. Chan, F. Chan, H. Du, and F. A. Dilkes, “Doppler frequency geolocation of uncooperative radars,” in *Proc. IEEE Military Commun. Conf. (MILCOM), Orlando, USA*, Oct. 2007, pp. 1–6.
- [80] A. Amar and A. J. Weiss, “Localization of narrowband radio emitters based on doppler frequency shifts,” *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5500–5508, Nov. 2008.

- [81] N. H. Nguyen and K. Doğançay, “Closed-form algebraic solutions for 3-d doppler-only source localization,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6822–6836, Oct. 2018.
- [82] L. Deng, P. Wei, Z. Zhang, and H. Zhang, “Doppler frequency shift based source localization in presence of sensor location errors,” *IEEE Access*, vol. 6, pp. 59 752–59 760, Jan. 2018.
- [83] Y. T. Chan and F. L. Jardine, “Target localization and tracking from doppler-shift measurements,” *IEEE J. Ocean. Eng.*, vol. 15, no. 3, pp. 251–257, Jul. 1990.
- [84] Y. T. Chan, “A 1-d search solution for localization from frequency measurements,” *IEEE J. Ocean. Eng.*, vol. 19, no. 3, pp. 431–437, Jul. 1994.
- [85] Y. T. Chan and J. J. Towers, “Passive localization from doppler-shifted frequency measurements,” *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2594–2598, Oct. 1992.
- [86] Y. Kalkan and B. Baykal, “Target localization methods for frequency-only mimo radar,” in *2010 IEEE 18th Signal Process. Commun. Applications Conf.* IEEE, 2010, pp. 439–442.
- [87] B. Friedlander and J. O. Smith, “Localization of multiple targets from doppler measurements,” in *1984 American Control Conference.* IEEE, 1984, pp. 593–598.

- [88] I. Shames, A. N. Bishop, M. Smith, and B. D. O. Anderson, “Doppler shift target localization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 1, pp. 266–276, Jan. 2013.
- [89] D. Henrion, J.-B. Lasserre, and J. Löfberg, “Gloptipoly 3: moments, optimization and semidefinite programming,” *Optimization Methods & Software*, vol. 24, no. 4-5, pp. 761–779, 2009.
- [90] J. F. Sturm, “Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,” *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.
- [91] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, “Sdpt3—a matlab software package for semidefinite programming, version 1.3,” *Optimization methods and software*, vol. 11, no. 1-4, pp. 545–581, 1999.
- [92] J. B. Lasserre, “A semidefinite programming approach to the generalized problem of moments,” *Mathematical Programming*, vol. 112, no. 1, pp. 65–92, 2008.
- [93] R. M. Freund, “Introduction to semidefinite programming (sdp),” *Massachusetts Institute of Technol.*, pp. 8–11, 2004.
- [94] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming,” Jan 2020.
- [95] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, “Semidefinite programming approaches for sensor network localization with noisy distance measurements,” *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, 2006.

- [96] R. W. Ouyang, A. K.-S. Wong, and C.-T. Lea, “Received signal strength-based wireless localization via semidefinite programming: Noncooperative and cooperative schemes,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1307–1318, 2010.
- [97] G. Wang and K. Yang, “A new approach to sensor node localization using rss measurements in wireless sensor networks,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1389–1395, 2011.
- [98] D. Den Hertog, *Interior point approach to linear, quadratic and convex programming: algorithms and complexity*. Springer Science & Business Media, 2012, vol. 277.
- [99] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.
- [100] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.
- [101] L. L. Scharf, *Statistical signal processing*. Reading, MA: Addison-Wesley, 1991, vol. 98.
- [102] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004.
- [103] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, “Semidefinite programming based algorithms for sensor network localization,” *ACM Trans. Sens. Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.

- [104] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.
- [105] J. F. Epperson, *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.
- [106] N. H. Nguyen and K. Doğançay, “Optimal sensor placement for doppler shift target localization,” in *2015 IEEE Radar Conference (RadarCon)*. IEEE, 2015, pp. 1677–1682.
- [107] M. M. Ahmed, K. Ho, and G. Wang, “Localization of a moving source by frequency measurements,” *IEEE Trans. Signal Process.*, vol. 68, pp. 4839–4854, 2020.
- [108] S. Nardone, A. G. Lindgren, and K. Gong, “Fundamental properties and performance of conventional bearings-only target motion analysis,” *IEEE Trans. Autom. Control*, vol. 29, no. 9, pp. 775–787, Sep. 1984.
- [109] V. J. Aidala, “Kalman filter behavior in bearings-only tracking applications,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, no. 1, pp. 29–39, Jan. 1979.
- [110] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications (MPS-SIAM Series on Optimization)*. Philadelphia, PA, USA: SIAM, 2001.
- [111] T. Jia, D. K. Ho, H. Wang, and X. Shen, “Localization of a moving object with sensors in motion by time delays and doppler shifts,” *IEEE Trans. Signal Process.*, 2020.

- [112] Y. Chan and S. Rudnicki, “Bearings-only and Doppler-bearing tracking using instrumental variables,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 4, pp. 1076–1083, Oct. 1992.
- [113] M. Yano, J. D. Penn, G. Konidakis, and A. T. Patera, “Math, numerics, and programming: For mechanical engineers,” 2013.
- [114] J. A. Belloch, B. Bank, F. D. Igual, E. S. Quintana-Ortí, and A. M. Vidal, “Solving weighted least squares (wls) problems on arm-based architectures,” *J. Supercomput.*, vol. 73, no. 1, pp. 530–542, 2017.

VITA

Musaab M. Ahmed was born in Nineveh, Iraq. He received his B.Sc. (with first class honors) degree in Telecommunications Engineering from the University of Mosul, Nineveh, Iraq, in 2011, and his M.S. degree in Electrical Engineering and Ph.D. Minor degree in Statistics from the University of Missouri-Columbia, MO, USA, in 2016 and 2020 respectively. He pursued his Ph.D. degree with the Department of Electrical Engineering and Computer Science, University of Missouri-Columbia, MO, USA.

Musaab is Married to Maryam Al-Sabbagh and they have four children: Abdullah, Zainab, Zumurrud and Abdul-Kareem. After the Ph.D. degree, Musaab plans to return to Iraq and work as a teacher and a researcher for the Ministry of Higher Education and Scientific Research.