PREDICTION OF RENAL RESPIRATORY MOTION FOR COMPENSATION DURING PERCUTANEOUS NEEDLE INSERTION IN ROBOT ASSISTED SURGERY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET ÇETINKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2018

Approval of the thesis:

PREDICTION OF RENAL RESPIRATORY MOTION FOR COMPENSATION DURING PERCUTANEOUS NEEDLE INSERTION IN ROBOT ASSISTED SURGERY

submitted by **MEHMET ÇETİNKAYA** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Aydan Müşerref Erkmen _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Aydan Müşerref Erkmen
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Uluç Saranlı
Computer Eng. Dept., METU _____

Assist. Prof. Dr. Emre Özkan
Electrical and Electronics Eng. Dept., METU _____

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Eng. Dept., Hacettepe University _____

Date: **21.09.2018**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET ÇETINKAYA

Signature :

ABSTRACT

PREDICTION OF RENAL RESPIRATORY MOTION FOR COMPENSATION DURING PERCUTANEOUS NEEDLE INSERTION IN ROBOT ASSISTED SURGERY

Çetinkaya, Mehmet
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Aydan Müşerref Erkmen

September 2018, 132 pages

In this study, the aim is to design a respiratory motion prediction algorithm which can be used to compensate for this physiological disturbance in medical operations where respiration limits operation accuracy. For this purpose, a new Kalman filter has been developed for tracking quasi-periodic signals approximated as finite Fourier series. Instead of relying on approximations provided by Extended Kalman Filter or Unscented Kalman Filter, our filter performs the exact calculation of the mean and covariances of interest. Our results indicate that the theoretically derived mean and covariance calculations result in either comparable or better estimation performance in terms of convergence speed and output estimation error depending on the circumstances. We then employ an expectation maximization algorithm to find the maximum likelihood estimates of the process noise with known measurement noise statistics. Coupled with this, the new filter is able to track the output despite breathing irregularities. However, the degree of irregularity may cause divergence from the assumed underlying model.

Keywords: Quasi-Periodic Signal Tracking, Nonlinear Kalman Filtering, Exact Kalman Filtering, Adaptive Kalman Tuning

ÖZ

ROBOT DESTEKLİ CERRAHİ İLE YAPILAN PERKÜTAN İĞNE GİRİŞİMİ SIRASINDA SOLUNUM KAYNAKLI BÖBREK HAREKELERİNİN KOMPANSE EDİLMESİ İÇİN KESTİRİLMESİ

Çetinkaya, Mehmet
Yüksek Lisans, Elektrik ve Elektronik Müh. Bölümü
Tez Yöneticisi : Prof. Dr. Aydan Müşerref Erkmen

Eylül 2018 , 132 sayfa

Bu çalışmanın amacı, hassasiyet gerektiren tıbbi operasyonlarda oluşabilecek solunum kaynaklı bozuklukları gidermede kullanılabilecek solunuma dayalı hareket kestirim algoritmasının modellenmesidir. Bu amaçla, sonlu Fourier serisi yaklaşımı yapılmış yarı periyodik sinyallerin takibi için yeni bir Kalman filtresi geliştirildi. Geliştirdiğimiz filtre, Genişletilmiş Kalman veya Kokusuz Kalman Filtre'lerinin sağladığı yaklaşık ortalama ve kovaryans değerlerini kullanmak yerine, bu değerleri tam olarak hesaplamaktadır. Aldığımız sonuçlara göre, teorik olarak hesaplanan bu değerler yakınsama hızı ve çıktı kestirim hatası bazında iyileşme sağlamaktadır. Süreç gürültüsünün en çok olabilirlik kestirimi, bilinen bir ölçüm gürültüsü varsayımı ile beklenti maksimizasyonu algoritması kullanılarak bulunmuştur. Oluşabilecek solunum düzensizliklerine rağmen, bahsi geçen filtreleme yöntemi ve beklenti maksimizasyonunun birlikte kullanımı ile çıktı kestirimi yapılabilmektedir. Fakat, düzensizliğin derecesi varsayılan modelden ıraksamaya sebep olabilmektedir.

Anahtar Kelimeler: Yarı Periyodik Sinyal Takibi, Doğrusal Olmayan Kalman Filtreleme, Yaklaşık Olmayan Kalman Filtreleme, Uyarlamalı Kalman Ayarı

To my family

ACKNOWLEDGMENTS

First off, I would like to express my appreciation of my supervisor, Prof. Dr. Aydan M. Erkmen, for her supervision, criticism and discussions throughout this study.

I would like to take this opportunity to acknowledge the insightful guidance, encouragement and constructive support I got from Assoc. Prof. Umut Orguner, without whom this work would have ended at an incomplete level. I am also thankful to my thesis committee members Prof. Uluç Saranlı, Assist. Prof. Emre Özkan and Assist. Prof. Yakup Özkazanç for their advice and suggestions as well as their time for being there.

I also would like to thank my labmates, Efe, Zafer and Alperen for the many cheer- and beer-ful nights; you all gave me a great amount of support when I was in dire need of it. My puffs of smoke would have been lonely if it weren't for you guys. Your presence at the lab was a great source of desire to work, especially late nights.

I think this paragraph should start with an apology in advance: I am sorry just in case I forget any one of you. An initial version of this paragraph was about to end in a long detailed one. To keep things neat and less personal, I would like to thank Artun, Dilem, Asil, Kerem, Selin, Büşra, Hande, Selim and many others for being a good support.

I would also like to thank Scientific and Technological Research Council of Turkey (TÜBİTAK) for their financial support during my M.Sc. studies. I would like to thank ELMED for providing us the data for we have used in validating our methods.

Last but not least, I would like to thank each and every member of my family for their support in getting me where I am today. I could not share sufficient amount of time while producing this work, which I promise to make up for. This promise is also to my niece, Naz, for our lack of time together this summer. Thank you mom and dad, Asie and İsmet, for bringing me up to who I am. Thank you my sister and brother,

Zeynep and İsmet, for guiding me all my life. Thank you Naz for being the lovely niece you are. Thank you my grandmother, Alime, for the unfading memories even after three years after you passed away.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

C/DT	Continuous or Discrete Time
EKF	Extended Kalman Filter
EM	Expectation Maximization
ExKF	Exact Moment Matching Kalman Filter
GRV	Gaussian Random Variable/Vector
KF	Kalman Filter
MCT	Monte Carlo Transformation
ML	Maximum Likelihood
NA	Not Available
PCA	Principal Component Analysis
p(s)d	Positive (Semi-)Definite
r.v.	Random Variable
RMSE	Root Mean Square Error
SS	State Space
UKF	Unscented Kalman Filter
w.r.t.	With Respect To

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Humans and robots have complementary strengths in medical operations. When compared to humans, robots have poorer judgment and less versatility in adapting to new situations. Their limited dexterity and manipulator-eye coordination still restrict their widespread usage. However, they can be used as assistant to a surgeon's capabilities as they are stable with repeatable moves and they are not prone to fatigue. Moreover, robot design allows different motion scales whereas humans have limited motion scale due to their presized end effectors. Furthermore, robots are immune to radiation and infection that improves surgical safety. Robots can also provide more consistent results as they have better geometric accuracy and they can integrate big data from multiple sensors [12]. Within the balance of this thesis, we aim at assistive biorobotics and focus on a situation where a robot is used to enhance surgeon's capabilities.

There are various sources of physiological disturbances that limit surgeon's capabilities in medical operations, the most significant ones being respiratory motion of the patient, his/her heartbeat and tremor of the surgeon [13]. Examples to the operations where respiratory motion can be a problem are, among others, cardiac interventions, radiotherapy and minimally invasive procedures such as percutaneous access to kidneys. Heartbeat has a more local effect, only causing movements in the chest and upper abdomen. Tremor, in contrary to the other two, originates not from the patient but from the surgeon and it is the involuntary movement of the hand of the surgeon.

What these disturbances have in common is that they are quasi-periodic and that medical operation accuracy can benefit greatly from the precise tracking of them.

To get a better idea of these quasi-periodic disturbances, a visualization of quasi-periodic motion can be seen in Fig. 1.1.a which shows a regular breathing pattern. Fig. 1.1.b shows a breathing pattern with intermittent stops, resulting in frequency changes of the signal. 1.1.c is a signal with changing DC component and 1.1.d depicts a signal with changing amplitude. Quasi-periodic movement of the heart, at a higher frequency than the respiratory one, modulated with the respiratory motion is shown in Fig. 1.1.e. Their overall combination is given in Fig. 1.1.f.

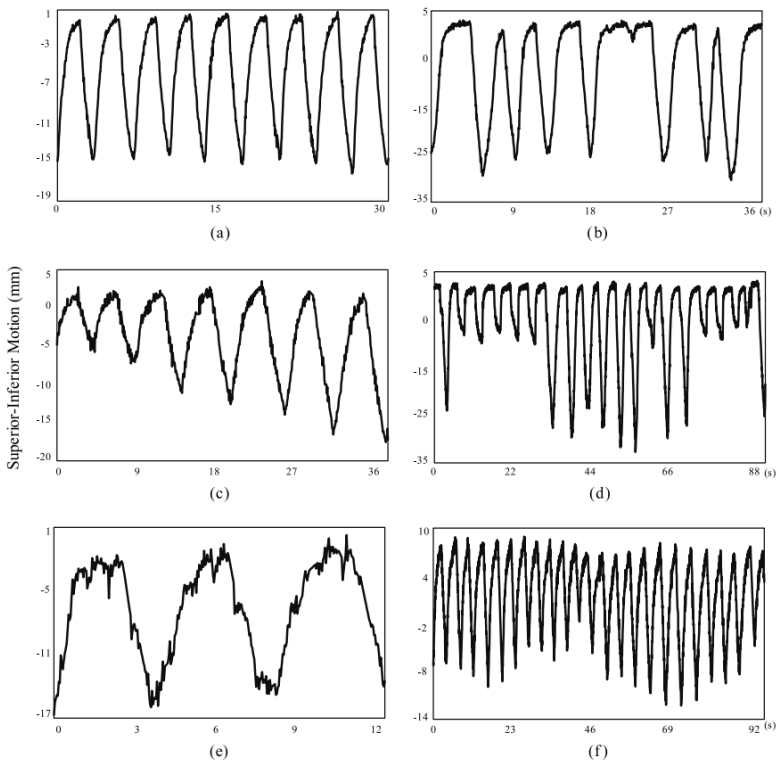

Figure 1.1: Patterns of breathing, (a) regular breathing, irregular breathing with (b) changing frequency, (c) changing baseline, (d) changing amplitude, (e) cardiac motion modulated with respiratory motion and (f) combination of them, obtained from [1]

To introduce an operation where respiration acts as a physiological disturbance, percutaneous nephrolithotomy (PCNL) is a term used to describe the medical operation of accessing the kidney using a needle inserted through the skin with the aim to access, destroy and extract kidney stones. On a word by word basis, percutaneous, nephro-, -litho- and -tomy mean through the skin, kidney, stone and removal, re-

spectively. It is accepted as the standard procedure for the removal of stones with a diameter larger than 2 cm, although other factors such as stone chemical composition and stone location in kidneys may make PCNL necessary. It is a minimally invasive procedure, with reduced morbidity and recovery time when compared to its open surgery counterpart. However, it still remains a challenging problem with approximately 11% of the surgeons performing PCNL achieve kidney access [14]. This problem arises from the combined effect of the lack of skill of the surgeon, lack of three dimensional information due to the limitations of two dimensional imaging - the two dimensional fluoroscopy screen. This imaging procedure also exposes the patient to levels of radiation higher than the limit one person should be exposed to within a single year [14].

Respiratory motion causes problems during PCNL operations as well. Placement of the needle needs an accuracy of 5 mm so that it can be punctured to the desired calyx of the kidney, which are veinless regions of kidney that constitute a safe entry point. When such operations are performed by hand, re-puncture is usually necessary as hand driven error rates are high. Moreover, respiratory motion may cause movements in the region of interest up to several centimeters, which is far more than the required accuracy stated at the beginning of this paragraph. The usual approach in dealing with respiratory motion is halting the respiration, but this cessation may not last longer than 25 seconds [15]. That is why, the operation is segmented into intervals of at most 25 seconds; which cause the operation to become inefficient, with prolonged operation time and, thus, increased costs. Also, the respiration needs to be stopped at the same point of respiratory cycle so that the target entry point is in the same point in space at each cessation. This requirement is prone to error as it is "controlled manually by the anesthesiologist" [15]. On top of this, being able to stop respiration requires the usage of general anesthesia and mechanical ventilation. On the other hand, if we were to predict and compensate for respiratory motion especially around the punctuation area of the kidney as it is our aim in this thesis, this would enable the usage of local anesthesia, thus decreasing operation time and patient morbidity.

1.2 Objectives and Goals

One of the key challenges during a robotic surgery is that instantaneous motion control of a mechanical device cannot be achieved. Data acquisition, its processing for the determination of new robot position and mechanical actuation all add up to the overall system latency as can be seen in Fig. 1.2. Current data acquisition systems are relatively fast with a common approximate latency around 10 ms [2]. Data processing for the determination of robot position is dependent on the computations used for motion prediction and inverse kinematics of the robot; general latency of data processing in actual clinical systems can be accepted to be in the range of 1 ms to 20 ms [2]. The last and the major source of latency comes from the robotic actuation, with common approximate delays of 75 to 100 ms. In total, these add up to a latency of 100 to 150 ms that has to be compensated for by motion prediction [2].

The sources of latencies are not the only problem associated with respiratory motion prediction. Patients show varying, irregular patterns of respiratory motion as was shown in Fig. 1.1. Our objective in this study is to predict respiratory motion adaptively with patient acceptable clinical accuracies despite changing breathing patterns.

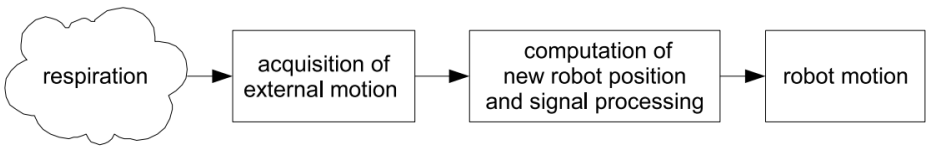

Figure 1.2: Sources of latencies obtained from [2]

1.3 Methodology

Prediction of respiratory motion is a well-studied subject with the earliest work dating back to 1997 [1], with various model-free and model-based methods developed for this task. One of the most widely used method is a sinusoidal model with slowly varying frequency and amplitude coefficients for tracking quasi-periodic signals, which we intend to use in this work. Both an Extended Kalman Filter (EKF) and an Unscented Kalman Filter (UKF) have been studied for such a model. However, to the best of our knowledge, only the first order EKF [16] has so far been examined and

UKF [17] was used for a simple nonlinear system with a single harmonic. What is more is that the underlying model used in both is the worse of the two possible alternatives: states are chosen such that the system is more nonlinear. Instead, we will use an Exact Moment Matching KF (ExKF) for the tracking problem with a more suitable state space representation that favors linearity more [18]. Both the UKF and EKF use approximations for the calculation of mean and covariance after a random variable is passed through the nonlinearity. In contrast, our filter calculates these parameters exactly. By doing so, it is expected that ExKF should be as good as a Monte Carlo KF using much less computational resources with higher accuracy than EKF and UKF [19]. This higher accuracy can tackle the motion prediction problem better, which is needed because of the latencies.

In contrast to previous Kalman filter implementations for tracking respiratory motion, we will use a second KF for the adaptation of process noise covariances to improve the accuracy of tracking. Manual tuning of process noise covariance is error prone and process noise covariances are not stationary but may change during tracking due to changing breathing patterns. For these reasons, automatic tuning is needed.

1.4 Contribution

The main contribution of this thesis is the development of an ExKF for tracking quasi-periodic signals and show its superior performance when compared to the traditional nonlinear KFs. Its novelty lies in the exact calculation of first two moments of the nonlinear sinusoidal observation model. Although there were one other study for the calculation of mean and covariance for a mapping such as $\sin\theta$, its derivation did not allow the extension for non-scalar case. Different from their approach, we used the linearity property of expectation and differentiation as well as the Moment Generating Function of a Gaussian random vector for calculations. With such an approach, mean and variance of signals in the form of $a\sin\theta$, for jointly Gaussian a and θ, are found explicitly. This is subsequently extended for multi harmonic tracking.

ExKF is then shown to be better than the other two filters when the initial estimates associated with the states are erroneous. Although all filters have comparable per-

formances when the initial estimate in the current angle θ is exactly known, ExKF outperforms the other two when this initial estimate is wrong. This is beneficial for tracking a quasi-periodic motion as we do not know at which instant and phase we start tracking the actual signal. Moreover, our filter converges faster than the other two when the estimates of the current angle θ and amplitude coefficients are wrong. Apart from convergence, it is shown that the newly developed filter is able to track a quasi-periodic signal when covariance associated with either (i) the amplitude coefficient or (ii) the current angle is high. These imply that it should better be able to track signals with (i) changing amplitude and (ii) changing frequency both of which are observed in respiratory motion. These results clearly favor the usage of ExKF over UKF and EKF.

Although the respiratory motion shows a wide variety of irregularities, Kalman filtering has been done with manually tuned systems. Instead of manual tuning, an expectation maximization (EM) algorithm will be used to iteratively find the maximum likelihood estimates of the process noise covariance. Inspired by the idea from projected gradient descent, we perform a projection after each maximization to guide the search for faster convergence. This also enables the covariance matrix to fit the desired structure.

1.5 Outline of the Thesis

The outline of this thesis is as follows.

In chapter 2 a literature review is presented regarding the studies on respiratory motion. Specifically, respiratory motion and its mechanics are discussed, requirements of robotic platforms for respiratory motion simulations are then presented. Existing modeling algorithms are then briefly discussed.

Chapter 3 focuses on respiratory motion as a tracking problem. State space models and equations needed for EKF and UKF are first presented. Existing KFs are then discussed along with the need for the development of an ExKF, followed by a derivation of our filter. Next, an Expectation Maximization based identification of the process noise covariance is presented.

Chapter 4 compares all three Kalman filters in terms of state and output RMSEs using simulation data, which is followed by the demonstration of the validity of the results on a respiratory motion trace from a patient.

Chapter 5 evaluates the performance of the ExKF coupled with an EM algorithm for tracking a simulated quasi-periodic signal as well as their performances on real respiratory motion trace.

Chapter 6 lists our concluding remarks as well as possible future research directions.

CHAPTER 2

LITERATURE SURVEY

Respiratory motion tracking requires prediction of respiratory motion due to unavoidable latencies of robotic platforms. Correlation, on the other hand, enables the prediction of internal movement from externally placed markers. When these two are coupled together, they enable the prediction of internal organ position with the externally placed markers at a desired time horizon. In this work, we aim to tackle the prediction problem.

We will begin by first introducing the respiratory motion mechanics and how this dynamics change during general anesthesia for possible benefits in percutaneous operations in Section 2.1. Section 2.2 surveys the range of kidney motion in order to build up the background for requirements of a respiratory motion platform that should be built in order to test the accuracy of the overall accuracy of the percutaneous access system as a respiratory motion simulator, which is given in section 2.3. Section 2.4 gives the common evaluation metrics used in respiratory motion prediction. Section 2.5 introduces the Principle Component Analysis which is commonly used in for dimensionality reduction in respiratory motion prediction algorithms. Finally, section 2.6 presents some portion of the prediction algorithms that currently exist on literature as well as our justifications as to why we opted not to use them.

2.1 Respiratory Motion and Its Mechanics

2.1.1 Respiratory Motion Under Free Breathing

Breathing consists of two phases: inhalation (inspiration or breathing in) and exhalation (expiration or breathing out). In mammals, inhalation is mainly accomplished by the contraction of diaphragm, increasing thoracic (i.e. belonging to upper body) volume and, consequently, lung volume [3]. This increase is accompanied by a decrease in lung pressure, allowing atmospheric air which is at a higher pressure to flow into lungs. The opposite case occurs for exhalation.

To dig a little bit deeper as in [3], contraction of diaphragm straightens its parachute shape (see Fig. 2.1) and expands lung volumes by pulling them downwards. This contraction, though inadvertently, causes mobility in both the upper body and abdominal organs. The inhalation phase is also assisted by the external intercostal muscles (i.e. muscles located at the outside of ribs), which contract and engender the elevation of the rib cage, increasing lung volume by as much as 25% (see Appendix A for details) comparable to 75% contribution of diaphragm activity. In addition to diaphragm and intercostals, further muscles depicted in Fig. 2.1 are enacted for deep inhalation to allow greater thoracic expansion. These muscles are sternocleidomastoid muscles, the scalene muscles and the pectoralis minor muscles elevating the sternum, first two ribs and 3^{rd} to 5^{th} ribs, respectively. These are collectively termed as accessory muscles since their contribution is negligible.

Exhalation, on the other hand, is usually a passive process. When both diaphragm and intercostal muscles are relaxed (or released), diaphragm moves upwards and rib cage moves inwards, both into their natural state due to their elasticity. Active expiration occurs only in deep breathing, where internal intercostals are contracted to reduce rib cage volume and several abdominal muscles are enacted to push the diaphragm further up by compressing abdominal organs. These are all coordinated through the respiratory center of the brain, which fires rhythmic impulses to enable the required contractions. This center is also connected to the cerebral cortex that enables the voluntary control of breathing.

As the target group of our implementation consists of patients under anesthesia, the next subsection is devoted to the description of changes that occur due to anesthesia.

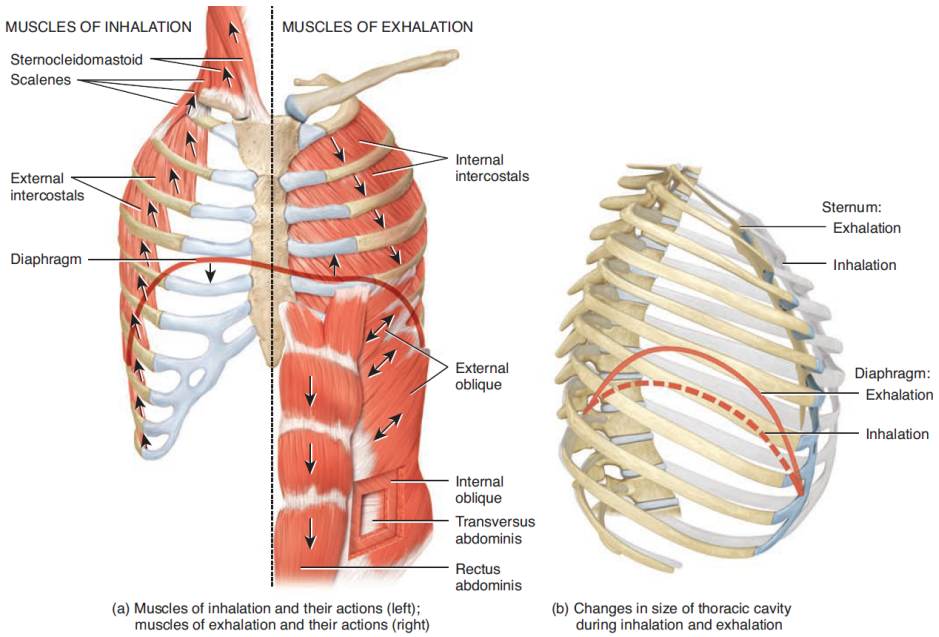

Figure 2.1: Respiratory muscles assisting breathing, obtained from [3], p.937

2.1.2 Respiratory Motion Under Anesthesia

Respiratory cycle takes a different turn when anesthetic agents are used [20]. These agents are used to relieve or suppress pain, and most of them affect the control of breathing by suppressing the respiratory center. This suppression may be observed even in sub-anesthetic concentrations and is generally a side effect.

Different areas of the brain are responsible for inhalation and exhalation, and these areas are affected by anesthesia differently. According to Pelosi et al. [20], "diaphragm activity is fairly well preserved" in the inhalation phase of respiratory cycle, but the contraction of external intercostal muscles is lost. In contrast, exhalation, which is a passive process for awake patients under normal breathing, becomes active under anesthesia. The reason for this activation is unknown but the result is the contraction of internal intercostal muscles. It should be noted that not all of anesthetic agents have these exact effects, but these are the general conclusions from the aforementioned source.

What these imply for the problem at hand are as follows:

- Baseline shifts in respiratory motion are still possible even under general anesthesia due to active expiration.
- Range of abdominal organ motion is expected to increase due to preserved diaphragm activity while thoracic organ motion is expected to decrease due to loss of intercostal muscle activity.

A more quantitative approach is presented in the next subsection.

2.1.3 Respiration Induced Motion: Range and Hysteresis

As was previously discussed in the introduction, sub-centimetric accuracy (0.5cm) is needed for percutaneous access to kidneys while the respiratory motion is in great excess of this range. Here, we will present a survey on the range of respiratory induced motion that kidneys undergo.

Rehder et al. [4] reported that general anesthesia where spontaneous breathing is allowed or aided, reduces the percent rib cage contribution to total lung volume change (termed as tidal volume) in a range of $11 - 34\%$. Consequently, the diaphragm contribution is increased with the same percentage values. This is attributed to the previously discussed loss of intercostal activity during inhalation. What this suggests is that chest wall motion is expected to decrease while abdominal one is expected to increase due to the decrease in thoracic contribution to the total tidal volume.

Mechanical ventilation is an artificial ventilation where the spontaneous breathing is replaced by an external machine. Under general anesthesia with complete mechanical ventilation, percent rib cage contribution to tidal volume increases, making the chest wall motion more prominent while diminishing the abdominal motion. This change is attributed to different elasticity levels of the rib cage and diaphragm. These results are depicted in Fig. 2.2 and offer benefits during abdominal laparoscopic surgeries such as percutaneous renal access. The most important benefit would be the decrease in the amplitude of abdominal motion that we are trying to eliminate. The second benefit of mechanical ventilation would be the stabilization of breathing patterns; mechanical

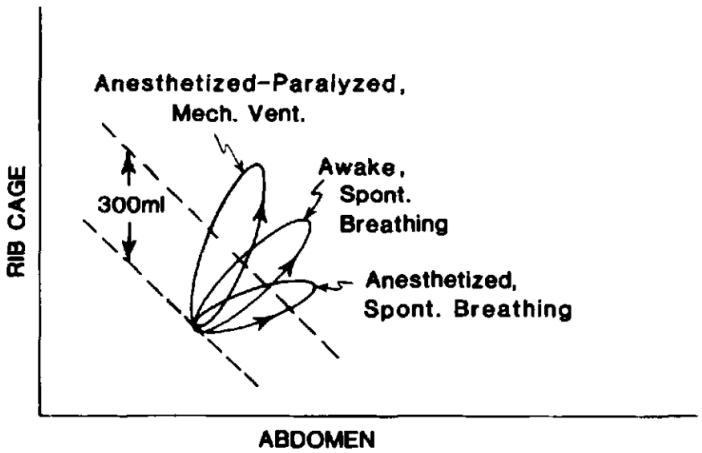

Figure 2.2: Changes in percentage of rib cage and diaphragm contribution to tidal volume, obtained from [4], p.738

ventilation does not eliminate the irregularities but makes the respiratory motion more predictable [21]. An example of this will be presented under the simulation chapter.

As a final remark, using anesthesia is not the only variable that is affecting the respiratory motion. Although we accept the suggestion that the range of abdominal motion decreases under general anesthesia with complete mechanical ventilation as correct, there are other factors that may change the range of renal respiratory motion. Take, for example, the inversely proportional respiratory motion and respiratory rate (frequency) relationship. Sorensen et al. [22] reported that least respiratory movement of a kidney stone occurred when the patient was anesthetized and had the highest respiratory rate, completely agreeing with the benefit stated at the end of the previous paragraph. Yet, highest respiratory motion occurred for a patient with the lowest respiratory rate, and this is reported to happen even under mechanical ventilation. There were also awake, spontaneous breathers under the same study, which we would have disregarded based only on the mechanical ventilation argument that "mechanical ventilation decreases abdominal motion". Therefore, we should focus on the the whole range of respiratory motion studies in the next section instead of focusing solely on the ones for complete mechanical ventilation.

2.2 Effect of Respiration on Kidney Motion

Any medical robotic platform should be thoroughly tested before using it on patients. That is why a physical device on which one can test the developed algorithms is needed. With this goal in mind, we will compile the studies on kidney respiratory motion to find the requirements of a possible respiratory motion simulator. In the rest of this thesis we will use the body directional terms that are generally accepted as shown in Table 2.1 [3].

Table 2.1: Body Directions

Superior (cranial)	Upwards or towards the head
Inferior (caudal)	Downwards, or away from the head
Anterior	Towards the front of the body
Posterior	Towards the back of the body

Pham et al. [23], compiled the results of 25 publications (related to studies on approximately 450 patients) focusing on renal respiratory motion. A brief version of the results are given in Table 2.2. First thing to be noted is that deep breathing causes the largest amount of respiratory movement as expected. Next, in accordance with the nature of the diaphragm movement, the greatest movement appears in Superior-Inferior direction. Kidneys also move in Anterior-Posterior and Left-Right directions, although the latter is negligibly small. NA's in the last two lines indicate that those measurements are not available.

Table 2.2: Kidney Motion Records

Direction	Condition	Mean (mm)	Range [Min, Max] (mm)
Superior-Inferior	Deep	18	[7, 86]
Superior-Inferior	Free	10	[4, 35]
Anterior-Posterior	Free	5	[NA]
Left-Right	Free	2	[NA]

The only study we could find regarding the speed and acceleration of kidney respiratory motion is written by Davies et al. [24]. The measurements were taken both from awake thoracic and abdominal breathers and are measured in superior-inferior direction. A summary of their measurements are given in Table 2.3. They observed that greater amplitude of respiratory motion was observed for abdominal breathers; this seems to suggest that mechanically ventilated patients, as in our case, would benefit

from such a result. They have also recorded age, sex, weight for any trends regarding these, but did not reach concluding remarks, suggesting further research on this was needed. A recent study regarding the these variables [25] concluded that kidney respiratory motion is "independent of age, sex, height or body weight"; that is, no definite conclusion can be drawn by looking at these variables. Another conclusion was that renal respiratory motion shows wide variations and it is necessary to evaluate respiratory motion for each individual patient.

2.3 Respiratory Motion Simulator

In this section, we will use the information gathered in the previous one to build up the requirements for a robotic platform which needs to be built to test the overall accuracy of algorithms the medical robotic system uses. As we have no records gathered from patients, we will use the following rudimentary algebraic manipulation for determining our own motion simulator: Maximum acceleration and velocity values given in Table 2.3 will be scaled by maximum displacements given in both tables. This gives internal motion simulation platform requirements as: capable of 86 mm displacement, $\max(v_{\text{Table 2.3}})\frac{\max(d_{\text{Table 2.2}})}{\max(d_{\text{Table 2.3}})} = 91$ mm/s velocity and $\max(a_{\text{Table 2.3}})\frac{\max(d_{\text{Table 2.2}})}{\max(d_{\text{Table 2.3}})} = 215$ mm/s^2 acceleration for simulating the motion in superior inferior direction. For the anterior posterior direction using mean values and adding a safety margin of 50%, those values are 8 mm, 6 mm/s and 12 mm/s^2. Left-right displacement is negligibly small but its requirements, if needed, are 3 mm, 2 mm/s and 5 mm/s^2.

Table 2.3: Kidney Motion Records; Acceleration, Velocity and Displacement

Measurement Type	Mean $\pm$ SD	Min	Max
Displacement (mm)	11 ± 4	5	16
Velocity (mm/s)	8 ± 2	5	17
Acceleration (mm/s^2)	18 ± 8	8	40

An example of a motion simulator is given in Fig. 2.3. This simulator has been developed for testing the accuracy of CyberKnife with Synchrony (Accuray, Sunnyvale, CA) [5], a robotic radiation delivery system for the treatment of tumors. It has two independent parts: robotic platform given on the left hand side is what the authors call the tumor motion simulator (TMS) to simulate the internal organ motion. It con-

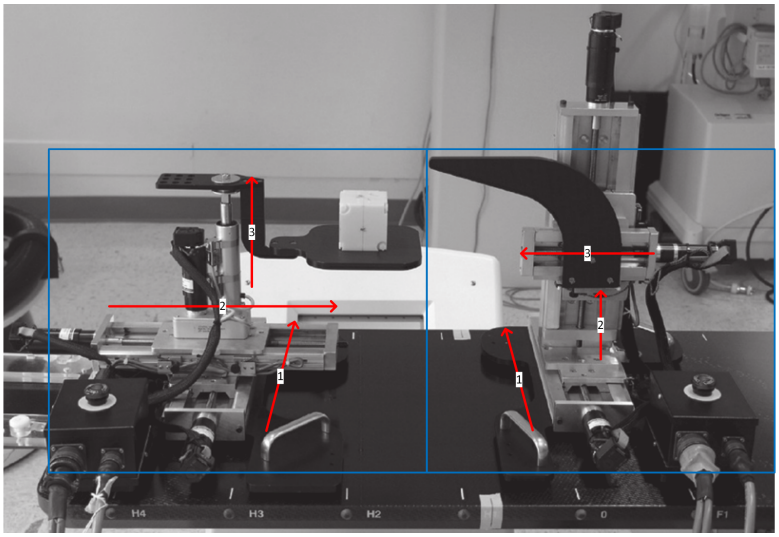

Figure 2.3: An example of respiratory motion simulator, adapted from [5]

sists of two linear slides, 1 and 2, for the movements in left-right and superior-inferior direction, while the last one, 3, is a linear actuator in anterior-posterior direction. The handle added to this third actuator should be made of nonmetallic materials so that X-ray imaging techniques can also be tested without obstructing its scene. The second robotic platform, skin motion simulator (SMS), merely consists of three linear slides to simulate external motion. The reason for using two independent platforms is that internal and external respiratory motion can be quite complex. Using two platforms should be sufficient to generate any correlated motion between the two. This is not the only solution available to simulate the respiratory motion (see, for example, commercially available QUASAR™ Respiratory Motion Phantom [26]) but the group of an SMS and TMS is a sufficient and easy to build solution. These platforms provide a clue for our own motion simulator with the aforementioned requirements.

2.4 Evaluation Metrics

There are various performance evaluation metrics that can be used in evaluating the accuracy of respiratory motion prediction algorithms, such as mean error, mean absolute error and maximum absolute error. However, the most commonly used one among them is Root Mean Square Error (RMSE) [21]. As its advantage over the others, RMSE considers the whole error signal while putting more weigh on large errors.

It is given as in Eq. (2.1), where y_i is the true signal to be tracked, $\hat{y}_i$ is the estimated signal and N is the length of y_i. Whatever underlying model we use to predict respiration, this is the only metric that makes sense as it is the only variable that we can measure. For example, we will use a state space model to predict respiration; yet, as we have no access to the true underlying states, what we can get from the system is the measured output and the only comparison we can make without any assumptions is to use it for prediction accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.1}$$

2.5 Principal Component Analysis

A common approach used in speeding up the estimation of external respiratory motion is using Principal Component Analysis (PCA) for dimensionality reduction [2]. PCA is an effective method for structuring data into bins such that the variance of projected data is maximized. As the external respiratory motion is mainly one dimensional, using PCA to find the motion in that dimension is beneficial for two reasons. First, the number of data points needed to be processed decreases. Second, signal to noise ratio in that dimension is maximized.

To give a short introduction sufficient for reducing the three dimensional respiratory motion record into one, we will use the explanations given in [27], which is the introductory example of PCA. What we are seeking is a 3 dimensional vector $\boldsymbol{u}_1$ that contains the maximum variance of a window of measurement sequence. As we are only interested in the direction of $\boldsymbol{u}_1$ not its magnitude, we can arbitrarily set $||\boldsymbol{u}_1|| = 1$. Mean of the original data is given in 2.2c and mean of the projected data after the translating the original data by an amount of its mean is 0. Its variance, which we seek to maximize, is given in 2.2e. In order to maximize this variance with the constraint that $||\boldsymbol{u}_1|| = 1$, a Lagrange multiplier can be introduced as in equation 2.2f. By differentiating this w.r.t. $\boldsymbol{u}_1$ and equating the result to zero, the necessary condition for extrema is given in equation 2.2g. This form of structure for $\boldsymbol{u}_1$ simply states that $\boldsymbol{u}_1$ is the right eigenvector corresponding to the eigenvalue λ_1 of $\boldsymbol{S}$ and maximizing

this is equivalent to choosing the largest eigenvalue λ_1 as well as its corresponding eigenvector $\boldsymbol{u}_1$.

$$\boldsymbol{x}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{2.2a}$$

$$x_T = \boldsymbol{u}_1^T(\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{x}}) \tag{2.2b}$$

where

$$\boldsymbol{\mu}_{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{2.2c}$$

$$\boldsymbol{S} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{x}})^T \tag{2.2d}$$

$$S_T = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{u}_1^T\boldsymbol{x}_i - \boldsymbol{u}_1^T\boldsymbol{\mu}_{\boldsymbol{x}})^2 = \boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1 \tag{2.2e}$$

$$\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1 + \lambda_1(1 - \boldsymbol{u}_1^T\boldsymbol{u}_1) \tag{2.2f}$$

$$\boldsymbol{S}\boldsymbol{u}_1 = \lambda_1\boldsymbol{u}_1 \tag{2.2g}$$

For respiratory motion tracking, the authors suggest that [2] at least five cycles of respiration to be used in calculating $\boldsymbol{\mu}_{\boldsymbol{x}}$ and $\boldsymbol{u}_1^T$. We will use their suggestion without any modifications.

2.6 Respiratory Motion Prediction Algorithms

As it was stated in the first chapter, respiratory motion prediction is a well studied subject, several outstanding studies can be seen in Fig. 2.4. The algorithms are classified into three parts: model-based and model-free methods as well as the hybrid methods that use the combination of them. We will briefly discuss some of them as well as their shortcomings in the following sections.

2.6.1 Model Based Methods

One of the most frequently cited respiratory motion algorithm is called "Adaptive & Sinusoidal" model and it first fits a sinusoidal model given in equation 2.3 to a

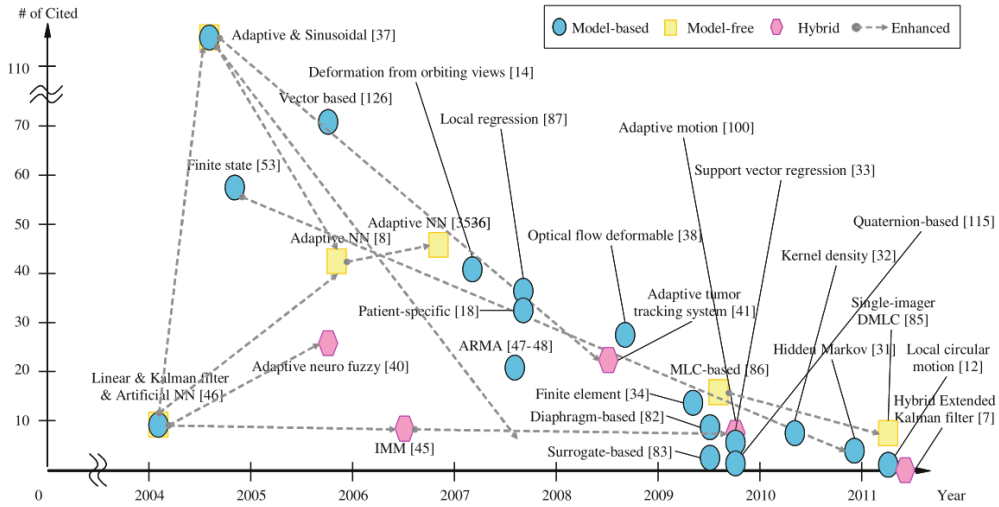

Figure 2.4: Respiratory motion prediction algorithms, obtained from [6]

signal history of chosen length. Then it uses the current measurement and updates it according to the difference provided by the fitted sinusoidal model. This method has two shortcomings. According to the authors, one of them is that it does not cope well against latencies larger than 200 ms. The second one is that this model is based on a single harmonic and respiratory motion is clearly not such a signal. Its extension into multiple harmonic case is straight forward, which we will introduce in the next chapter when we give the state space model of filtering equations, but will give our criticism then.

$$x_{fitted}(t_n) = A\sin(Bt_n + C) + D \tag{2.3}$$

$$x_{pred}(t_n + \Delta) = x_{measured}(t_n) + (x_{fitted}(t_n + \Delta) - x_{fitted}(t_n)) \tag{2.4}$$

Another previous attempt uses a similar, multiharmonic sinusoidal model what they term as a "weighted-frequency Fourier linear combiner" [15]. They use a cascaded gradient descent approach for finding Fourier series coefficients and angular speed of the signal given in 2.5. The outer loop performs the frequency adaptation with a lower gain (consequently, with a lower speed) and the inner one performs the coefficient adaptation with a larger gain (consequently, with a larger speed). However, their approach does not take into account the noise associated with the measurements.

This is clearly apparent from their suggestion that a model order of M=2 or M=3 is suggested as higher order coefficients are likely to adapt to noise instead of the underlying signal.

$$x_{fitted}(t) = \sum_{k=1}^{M} \left(a_k \cos(t_n \omega) + b_k \sin(t_n \omega) \right) \tag{2.5}$$

Work conducted by Vedam et al. [28] approaches the tracking of respiratory motion as a Finite State Model in which they introduce discrete states with constant velocities to model inhale, exhale and end-of-exhale states of respiration which can be seen in Fig. 2.5. Their original work was a deterministic model where transitions are determined from current the speed estimated from the current and last measurement as well as time it takes from the last state transition. They have extended it to a probabilistic model using Hidden Markov Models where state transitions are determined probabilistically where state transition probabilities are determined from these variables. However, even this extension did not allow thier model to become "good enough". More specifically, even with a prediction horizon of 200 ms, their prediction errors exceeded the defined threshold value of 2.5 mm 80% of the time, with clear lags in the predicted signal.

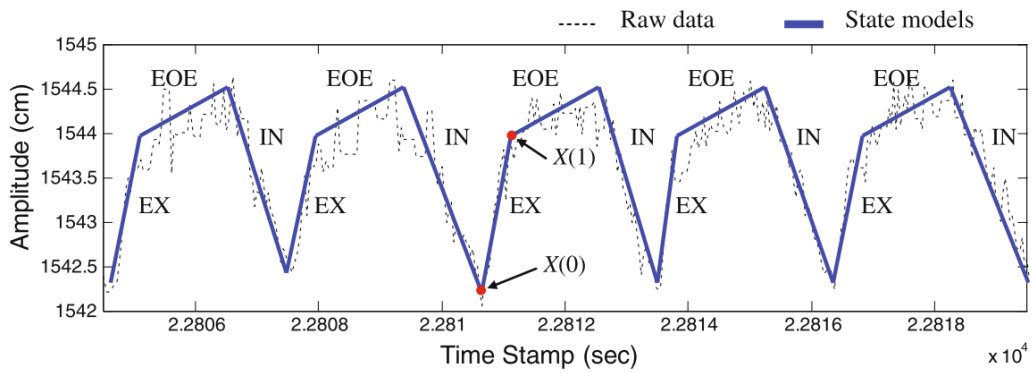

Figure 2.5: Finite state model of respiration, obtained from [6]

One last respiration model, "Vector based model" or more specifically more with significant citations is introduced by Low et al. [7]. It uses the total amount of air -the tidal volume- as well as the current flow to capture the hysteresis observed in internal organs. Total amount of air captures the linear part of motion observed using the vector $\overrightarrow{r_v}$ as can be seen in Fig. 2.6, where v is the tidal volume and α & $\hat{\boldsymbol{r}}_v$ are

the parameters to be found. Similarly, hysteresis is captured by $\overrightarrow{rf}$ where f is the patient's airflow and β & $\hat{\boldsymbol{r}}_f$ are the parameters to be found. By separating the model into inhale and exhale through f, this model is able to capture the hysteresis between the two modes of respiration. However, even though it was shown to be one of the most cited respiratory motion models, it does not fit the purposes of prediction. This model better suits a correlation model between internal and external sensors.

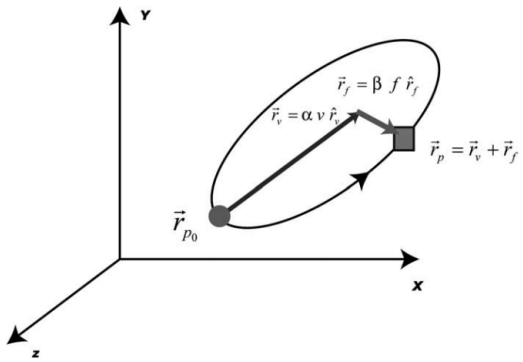

Figure 2.6: Vector based model of respiration, obtained from [7]

2.6.2 Model Free Methods

The most prominent model free method is Adaptive Neural Network (ANN) as can be seen in Fig. 2.4 [6]. The most general structure of a two-layered NN can be seen in Fig. 2.7. Inputs given to the system are the measured position over a training window n_i while the outputs are the future positions with desired prediction horizon. The idea behind the NN is simple: pass the desired inputs as well as hidden layers from a nonlinear activation function, their choice being the most commonly used sigmoid function ($f(x) = \frac{1}{1+e^{-x}}$), and learn the weights (w_{ij}, w_{jk}) that connect the layers. We won't be using this method for two reasons. The first reason is that their justification of using a NN is "highly irregular breathing patterns are difficult or impossible to model bio-mechanically", which should not happen under mechanical ventilation. The second reason, on the other hand, is that the boundaries of what a NN has learned may not be known clearly as they are black box models.

Another recent respiratory motion model is developed by using Gaussian process regression models [21]. The idea is to fit a "kernel function", an example of which

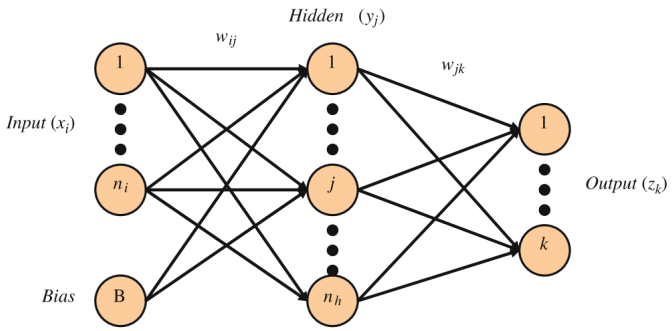

Figure 2.7: Two-layered NN architecture for respiratory motion prediction, obtained from [6]

is given in equation 2.6, and try to find the hyperparameters that relate the L2-norm distance r to the correlations of observed data points. This L2-norm difference corresponds to differences in observation times as far as time-series tracking is concerned. Quasi-periodic kernel function given in equation 2.6 is a natural choice for respiratory motion tracking problem. The hyperparameter θ_p captures the periodic nature/correlation of respiratory motion while the hyperparameter θ_l captures the inter-period variations by allowing less correlation and higher forgetting factor between each period (θ_s is a parameter that captures just the information related to the variance of the output). Although it is a good fit for respiratory tracking problem the problem associated with this method is its computational burden. The training time to find the parameters of interest is at the order of minutes as our initial attempts and authors' later publications reveal, which is in great excess of real-time tracking requirements. That is, sampling period of most external markers is in the range of $25-100$ milliseconds and the model we trained should have a shorter training time than this sampling period. That is the main problem with such machine learning algorithms, Support Vector Regression and Relevance Vector Machines being some examples of them. They are not fit for real-time tracking as their training times are higher than sensor sampling periods, making their usage in real time tracking implausible if retraining is often needed.

$$k_{QP} = \theta_S^2 \exp\left(-\frac{r^2}{2\theta_L^2}\right) \exp\left(-\frac{\sin^2[(2\pi/\theta_p)r]}{2}\right) \tag{2.6}$$

CHAPTER 3

PROPOSED METHODOLOGY

In this section, we will introduce the overall structure of our filtering problem, which can be seen in Fig. 3.1. Principal Component Analysis is used to reduce the 3-dimensional quasi-periodic signal (x_i, y_i, z_i) obtained from the sensor to 1D quasi-periodic signal (y_k) as external respiratory motion is mainly 1D. Subscript is changed with the only purpose of consistency within this chapter. Exact Moment Matching Kalman Filter (ExKF), or the other filters, will use this linearly projected measurement to find the the updated state estimates $\hat{\boldsymbol{x}}_{k|k}$ as well as the next measurement $\hat{y}_{k+1}$ and state prediction $\hat{\boldsymbol{x}}_{k+1|k}$. The tuning block will use these updated estimates as well as their predictions to find the Kalman process noise $\boldsymbol{Q}_k$. PCA was already introduced in the previous chapter. The state space model will be introduced in section 3.1.1, followed by the literature research on nonlinear Kalman filters in section 3.1.3. Then we will introduce our ExKF in section 3.2 as well as an EM algorithm tuned for our needs in section 3.3.

3.1 Mathematical Background

3.1.1 State Space Representation

In this section state space (SS) representation of the filtering problem will be derived. SS representation will be obtained in first continuous time and subsequently discretized; this approach will allow a computationally efficient multi-step prediction by eliminating the need for recursive estimation which is needed when only discrete model is used. Multi-step prediction is needed for respiratory motion tracking as

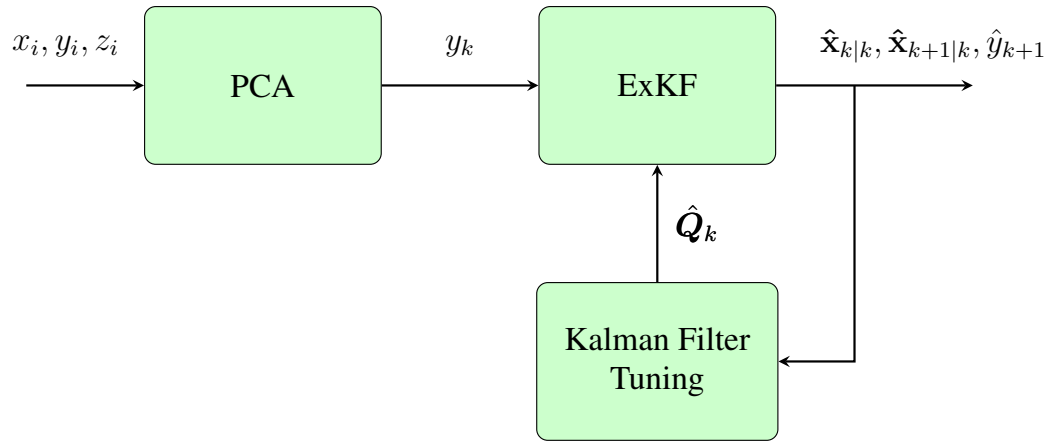

Figure 3.1: Block diagram of the proposed method

the latencies associated with sensors, processing their data and robotic actuation may exceed the estimation horizon provided by a single step predictor such as a Kalman filter.

The previous section and the section 1.1 have introduced the characteristics of the respiratory motion as a quasi-periodic signal. This quasi-periodic signal to be tracked is given by equation (3.1a) (in rectangular coordinates) or (3.1b) (in polar coordinates), where w_0 is the slowly changing fundamental frequency (the breathing frequency), a_0 (r_0) is the DC component of the signal (capturing the baseline shifts in Fig. 1.1.c), m is the order of Fourier series representation, a_i & b_i's (r_i's) are the respective amplitude coefficients and $v(t)$ is assumed to be additive white Gaussian noise with statistical properties given in equations (3.1c) and (3.1d).

$$y(t) = a_0(t) + \sum_{i=1}^{m} \Big(a_i(t)\cos(iw_0(t)t) + b_i \sin(iw_0(t)t) \Big) + v(t) \tag{3.1a}$$

$$= r_0(t) + \sum_{i=1}^{m} \Big(r_i \sin(iw_0(t)t + \phi_i(t)) \Big) + v(t) \tag{3.1b}$$

where

$$\mathbb{E}\{v(t)\} = 0 \tag{3.1c}$$

$$\mathbb{E}\{v(t_1)v(t_2)\} = \begin{cases} \sigma_v^2 & \text{if } t_1 = t_2 \\ 0 & \text{otherwise} \end{cases} \tag{3.1d}$$

There are two options for the state space representation of the above system: either use a nonlinear state transition matrix and make the measurement equation linear or use a linear state transition matrix with a nonlinear measurement model. Gustafsson (p. 323 of [29]) stated that linear time invariant system assumption is needed to have an exact discrete time (DT) state noise propagation; that is the reason why we will be adapting the latter. Having an exact DT noise propagation also enables us making multi step prediction more efficiently. That is, if we were to perform a multi step prediction of respiratory motion, we can do this in one state transition step, instead of repetitively performing the state propagation equations. This is in accordance with the aforementioned efficient multi-step prediction requirement.

Then, the choice for SS model boils down to whether to use the rectangular model in equation (3.1a) or polar model in (3.1b). To put the above equations into a more compact form, we make the following changes as was done in [30]. Different from their derivation is performing these steps in continuous time (CT) rather than DT similar to the approach of sine wave tracking problem presented in [31], where the authors tracked only a single sine wave. Note that b_1 in the rectangular form equation (3.2b) is intentionally set to zero as this information can be captured by a_1 and $\theta(t)$; thus, it is logical to eliminate the redundant state b_1 [30]. Secondly, the two forms seem to be identically equivalent when the underlying dynamics are deterministic with a simple change of variables. However, degree of nonlinearities involved in each model differs a lot, and that is expected to have an effect on the filtering performance.

$$\dot{\theta}(t) = \omega_0(t) \text{ and } \dot{\omega_0}(t) = \mathrm{w}_\omega(t) \tag{3.2a}$$

$$y(t) = a_0(t) + a_1(t)\cos(\theta(t)) + \sum_{i=2}^{m} \Big(a_i(t)\cos(i\theta(t)) + b_i(t)\sin(i\theta(t)) \Big) + v(t) \tag{3.2b}$$

$$y(t) = r_0(t) + \sum_{i=1}^{m} \Big(r_i \sin(i\theta(t) + \phi_i(t)) \Big) + v(t) \tag{3.2c}$$

Parker et al. [30] stated that tracking in rectangular coordinates has a superior performance to that in polar coordinates in radar tracking applications, without any references but with an appreciation of their colleagues for pointing this out. Yet, their

assertions seems reasonable when you consider that $m+1$ states in equation (3.2c) pass through a severe nonlinearity of a sine function while only one state is passed through that nonlinearity in rectangular coordinate form. This was demonstrated to be the case for tracking quasi-periodic signals, where Kim and McNames [18] have compared the performance of an Extended Kalman Smoother for tremor amplitude and frequency tracking. They showed that using rectangular form yields faster frequency tracking convergence and lower NMSE (Normalized Mean Square Error) than its polar counterpart. That is why we choose the model in equation 3.2b as the respiratory motion is a quasi-periodic signal like tremor. With these choices, the SS model then becomes as in equations (3.3a) - (3.3c), where $\mathbf{w} \in \mathbb{R}^{2m+1}$ is zero-mean process noise vector with covariance matrix $\boldsymbol{Q}_c = \mathbb{E}(\mathbf{w}\mathbf{w}^T) \in \mathbb{R}^{(2m+1)\times(2m+1)}$.

$$\boldsymbol{x}(t) = \begin{bmatrix} a_0(t) & a_1(t) & \dots & a_m(t) & b_2(t) & \dots & b_m(t) & \theta(t) & \omega(t) \end{bmatrix}^T \tag{3.3a}$$

$$\dot{\boldsymbol{x}} = \underbrace{\left[\begin{array}{c|cc} \mathbf{0}_{2m\times 2m} & \mathbf{0}_{2m\times 2} \\ \hline \mathbf{0}_{2\times 2m} & \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \end{array}\right]}_{\triangleq \boldsymbol{F}_c} \boldsymbol{x} + \underbrace{\left[\begin{array}{c|c} \boldsymbol{\mathcal{I}}_{2m\times 2m} & \mathbf{0}_{2m\times 1} \\ \hline \mathbf{0}_{2\times 2m} & \begin{matrix} 0 \\ 1 \end{matrix} \end{array}\right]}_{\triangleq \boldsymbol{G}_c} \mathbf{w} \tag{3.3b}$$

$$y(t) = \underbrace{a_0 + a_1\cos(\theta(t)) + \sum_{i=2}^{m}\Big(a_i\cos(i\theta(t)) + b_i\sin(i\theta(t))\Big)}_{\triangleq h(\boldsymbol{x}(t))} + v(t) \tag{3.3c}$$

In the formulation above, rate of change in amplitude coefficients is modeled as Wiener processes; thus they behave as Brownian or random walk models ($\dot{a}_i = \sigma_{a_i}^2$ etc.). Phase of the system, on the other hand, is modeled as a constant velocity model where the angular velocity is under random walk ($\dot{\theta} = \omega$ and $\dot{\omega} = \sigma_w^2$). Using equation 4.90 of [32], state transition matrix $\boldsymbol{\Phi}(t_1, t_2)$ of the above system can be obtained as in equation (3.4). While proceeding to the second line, nice structure of $\boldsymbol{F}_c$ allowed us to eliminate higher order terms. Discrete state transition matrix and discretized process noise covariance (using equation 4.112 of [32]) are given in equations (3.5) and (3.6), respectively, where T_s is the sampling period and $\boldsymbol{Q}_{ab}$ and q_ω are the block

diagonal entries of $\boldsymbol{Q}_c$.

$$\begin{aligned}\boldsymbol{\Phi}(t_1,t_2) = \boldsymbol{\Phi}(t_1-t_2) = e^{\int_{t_1}^{t_2}\boldsymbol{F}_c dt} = e^{\boldsymbol{F}_c(t_1-t_2)} = \sum_{n=0}^{\infty}\frac{\boldsymbol{F}_c^n(t_1-t_2)^n}{n!} \\ = \mathcal{I} + \boldsymbol{F}_c(t_1-t_2) = \left[\begin{array}{c|cc} \boldsymbol{\mathcal{I}}_{2m\times 2m} & \multicolumn{2}{c}{\mathbf{0}_{2m\times 2}} \\ \hline \multirow{2}{*}{$\mathbf{0}_{2\times 2m}$} & 1 & t_1-t_2 \\ & 0 & 1 \end{array}\right]\end{aligned} \tag{3.4}$$

$$\boldsymbol{F}_k = \boldsymbol{\Phi}(T_s) = \left[\begin{array}{c|cc} \boldsymbol{\mathcal{I}}_{2m\times 2m} & \multicolumn{2}{c}{\mathbf{0}_{2m\times 2}} \\ \hline \multirow{2}{*}{$\mathbf{0}_{2\times 2m}$} & 1 & T_s \\ & 0 & 1 \end{array}\right] \tag{3.5}$$

$$\begin{aligned}\boldsymbol{Q}_k &= \int_0^{T_s}\boldsymbol{\Phi}(T_s-\tau)\boldsymbol{G}_c\boldsymbol{Q}_c\boldsymbol{G}_c^T\boldsymbol{\Phi}(T_s-\tau)^T d\tau \\ &= \int_0^{T_s}\boldsymbol{\Phi}(T_s-\tau)\left[\begin{array}{c|c}\boldsymbol{\mathcal{I}} & \mathbf{0} \\ \hline \mathbf{0} & \begin{array}{cc}0 & \\ & 1\end{array}\end{array}\right]\left[\begin{array}{cc}\boldsymbol{Q}_{ab} & \mathbf{0} \\ \mathbf{0} & q_w\end{array}\right]\left[\begin{array}{c|cc}\boldsymbol{\mathcal{I}} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & 0 & 1\end{array}\right]\boldsymbol{\Phi}(T_s-\tau)^T d\tau \\ &= \int_0^{T_s}\left[\begin{array}{c|cc}\boldsymbol{\mathcal{I}} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & 1 & T_s-\tau \\ & 0 & 1\end{array}\right]\left[\begin{array}{c|cc}\boldsymbol{Q}_{ab} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & 0 & 0 \\ & 0 & q_w\end{array}\right]\left[\begin{array}{c|cc}\boldsymbol{\mathcal{I}} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & 1 & 0 \\ & T_s-\tau & 1\end{array}\right] d\tau \\ &= \int_0^{T_s}\left[\begin{array}{c|cc}\boldsymbol{Q}_{ab} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & (T_s-\tau)^2 q_w & (T_s-\tau)q_w \\ & (T_s-\tau)q_w & q_w\end{array}\right] d\tau \\ &= \left[\begin{array}{c|cc}\boldsymbol{Q}_{ab}T_s & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \mathbf{0} & q_w T_s^3/3 & q_w T_s^2/2 \\ & q_w T_s^2/2 & q_w T_s\end{array}\right]\end{aligned} \tag{3.6}$$

Overall state space equations in DT are listed below.

$$\boldsymbol{x}_{k+1} = \left[\begin{array}{c|cc} \boldsymbol{\mathcal{I}}_{2m\times 2m} & \multicolumn{2}{c}{\mathbf{0}_{2m\times 2}} \\ \hline \multirow{2}{*}{$\mathbf{0}_{2\times 2m}$} & 1 & T_s \\ & 0 & 1 \end{array}\right] \boldsymbol{x}_k + \mathbf{w}_k \tag{3.7a}$$

$$\begin{aligned} y_k &= h(\boldsymbol{x_k}) + v(k) = a_{0,k} + a_{1,k}\cos(\theta_k) \\ &\quad + \sum_{i=2}^{m} \Big(a_{i,k}\cos(i\theta_k) + b_{i,k}\sin(i\theta_k) \Big) + v_k \end{aligned} \tag{3.7b}$$

where

$$\boldsymbol{x}_k = \begin{bmatrix} a_{0,k} & a_{1,k} & \dots & a_{m,k} & b_{2,k} & \dots & b_{m,k} & \theta_k & \omega_k \end{bmatrix}^T \in \mathbb{R}^{(2m+2)} \tag{3.7c}$$

$$\begin{aligned} \boldsymbol{Q}_k &= \left[\begin{array}{c|cc} \boldsymbol{Q}_{ab}T_s & \multicolumn{2}{c}{\mathbf{0}} \\ \hline \multirow{2}{*}{$\mathbf{0}$} & q_w T_s^3/3 & q_w T_s^2/2 \\ & q_w T_s^2/2 & q_w T_s \end{array}\right] \in \mathbb{R}^{(2m+2)\times(2m+2)} \\ \boldsymbol{Q}_{ab} &= \mathbb{E}\left\{ (\boldsymbol{A} - \boldsymbol{\mu}_A)(\boldsymbol{A} - \boldsymbol{\mu}_A)^T \right\} \text{ for } \boldsymbol{A} = \begin{bmatrix} a_0 & a_1 & \dots & a_m & b_1 & \dots & b_m \end{bmatrix}^T \\ v_k &\sim \mathcal{N}(0; \sigma_v^2) \text{ and } \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}; \boldsymbol{Q}_k) \end{aligned} \tag{3.7d}$$

3.1.2 Bayesian Filtering

In this section a brief introduction introduction to Bayesian filtering will be provided. Interested readers are advised to read [33] and [34] for further details. In the remainder of this chapter, states and measurements are as given in equations (3.7a)-(3.7d).

Now, consider the following discrete time system from [34]. Most of the following explanations can be traced back to this source.

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{3.8a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{3.8b}$$

where Eq. (3.8a) is the process or dynamic model, $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the state vector of order n_x at time k, $\boldsymbol{f}_k$ is the state transition function, $\boldsymbol{w}_k$ is the process noise, Eq. (3.8b) is the measurement or observation model, $\boldsymbol{v}_k$ is the measurement noise and $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ is the measurement vector at time k. The goal is to obtain the following conditional densities: state transition density in Eq. (3.9a) and measurement model density in Eq. 3.9b.

$$\boldsymbol{x}_{k+1} \sim p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_{0:k}) \tag{3.9a}$$

$$\boldsymbol{y}_k \sim p(\boldsymbol{y}_k|\boldsymbol{x}_{0:k}, \boldsymbol{y}_{1:k-1}) = p(\boldsymbol{y}_k|\boldsymbol{x}_k) \tag{3.9b}$$

Our goal then becomes obtaining a recursive calculation of the estimation problem. Let $\boldsymbol{Y}_k = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k\}$ be the sequence of measurements. Then, using the previous conditional independence rules as well as applications of Bayes' rule and total probability theorem, one can obtain the following distributions. This conditional independence is acceptable in our problem as the state dynamics govern the evolution of states, no matter what the measurement we get from them. That is why, the next state $p(\boldsymbol{x}_{k+1})$ is independent from the measurement $\boldsymbol{Y}_k$ once we know the value of $p(\boldsymbol{x}_k)$.

Prediction or Time Update:

$$\begin{aligned} p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_k) &= \int p(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k|\boldsymbol{Y}_k) d\boldsymbol{x}_k \\ &= \int p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{Y}_k) p(\boldsymbol{x}_k|\boldsymbol{Y}_k) d\boldsymbol{x}_k \\ &= \int p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{Y}_k) d\boldsymbol{x}_k \end{aligned} \tag{3.10a}$$

Filtering or Measurement Update:

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) = \frac{p(\boldsymbol{x}_k, \boldsymbol{y}_k|\boldsymbol{Y}_{k-1})}{p(\boldsymbol{y}_k|\boldsymbol{Y}_{k-1})} = \underbrace{\frac{\overbrace{p(\boldsymbol{y}_k|\boldsymbol{x}_k)}^{\text{Likelihood}}\overbrace{p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})}^{\text{Prior}}}{\int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1})d\boldsymbol{x}_k}}_{\text{Normalization Constant}} \tag{3.10b}$$

Obtaining the distributions given in equations (3.10a) and (3.10b) is the key point in Bayesian filtering method. However, it is almost impossible to get those exact PDFs as analytical expressions except under very limited circumstances [34]. This happens to be the case for us as well; that is why we will be using Kalman filters to approximate those integrals. Also, marginalization integral in Eq. (3.10b) is often intractable. For linear systems with additive Gaussian noise, these can be solved optimally with some additional milder constraints such as the requirement of observability. Other methods such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) or any other nonlinear Kalman filtering scheme approximate these densities with the joint Gaussian assumption. Provided that the nonlinearities involved are mild and the process and measurement noise is not too far from Gaussian, such filters may be promising candidates to the filtering problem [34]. Stating from the same source, given an exact mean and covariance for any distribution, Gaussian density assumption can be justified to be useful because the Gaussian density is the maximum entropy distribution with that mean and covariance.

3.1.2.1 Kalman Filter

A special case when the above likelihood has an exact, recursive solution is when the traditional Kalman Filter (KF) is applicable [35]. Its recursive nature makes the KF a great online estimation algorithm thanks to its low computational burden; this burden can be further improved for LTI systems by allowing calculation of the Kalman gain offline [36]. KF is an optimal (in minimum mean square error sense) and a consistent filter provided that the following conditions are met.

- State transition ($\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k)$) and measurement ($\boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{v}_k)$) functions are lin-

ear.

- Process noise and measurement noise are additive.
- Process noise $\boldsymbol{w}_k$ and measurement noise $\boldsymbol{v}_k$ as well as the distribution of the initial estimate are Gaussian of known statistical properties with the usual assumption of zero-mean and independent noise components.

When the above assumptions are valid, Kalman Filter equations can be summarized as in Table 3.1 [36].

Table 3.1: Standard Kalman Filter Algorithm

State Space Model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{3.11a}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{3.11b}$$

Initialization:

$$\mathbb{E}\left\{\boldsymbol{x}_{0|-1}\right\} \triangleq \hat{\boldsymbol{x}}_0 = \mathbb{E}\left\{\boldsymbol{x}_0\right\} \tag{3.12a}$$

$$\boldsymbol{P}_{0|-1} \triangleq \boldsymbol{P}_0 = \mathbb{E}\left\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right\} \tag{3.12b}$$

Prediction or Time Update: (A priori state and its covariance estimate)

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k|k} \tag{3.13a}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k \boldsymbol{P}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q}_k \tag{3.13b}$$

Kalman Gain:

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_k^T \underbrace{(\boldsymbol{H}_k \boldsymbol{P}_{k+1|k} \boldsymbol{H}_k^T + \boldsymbol{R}_{k+1})^{-1}}_{\mathrm{cov}(\boldsymbol{y}_{k+1|k})^{-1}} \tag{3.14}$$

Filtering or Measurement Update: (A posteriori state and its covariance estimate)

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1} \underbrace{(\boldsymbol{y}_{k+1} - \underbrace{\boldsymbol{H}_k \hat{\boldsymbol{x}}_{k+1|k}}_{\hat{\boldsymbol{y}}_{k+1|k}})}_{\text{Innovation}} \tag{3.15a}$$

$$\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{\mathcal{I}} - \boldsymbol{K}_{k+1}\boldsymbol{H}_k)\boldsymbol{P}_{k+1|k} \tag{3.15b}$$

Assumptions:

$$\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_0; \boldsymbol{P}_0) \tag{3.16a}$$

$$\boldsymbol{w}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}; \boldsymbol{Q}_k) \tag{3.16b}$$

$$\boldsymbol{v}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}; \boldsymbol{R}_k) \tag{3.16c}$$

With the Gaussianity assumptions given in equations (3.16a)-(3.16c), prediction and measurement update distributions are Gaussian and given as follows.

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_{k+1}; \hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{P}_{k+1|k}) \tag{3.17a}$$

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k) = \mathcal{N}(\boldsymbol{x}_k; \hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k}) \tag{3.17b}$$

where

$$p(\boldsymbol{x}; \boldsymbol{\mu}_x, \boldsymbol{P}_x) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{P}_x|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_x)^T \boldsymbol{P}_x^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_x)} \tag{3.17c}$$

When the measurement noise given in Eq. (3.14) is considerably higher than the process noise, Kalman gain will have small eigenvalues. This, in turn, will result in filtered estimates to depend more on process dynamics than on measurement values. Conversely, when the measurement noise is much lower than process noise, Kalman gain will be high, resulting in the filtered estimates to be more dependant on the measured values than the process dynamics.

The key point in designing a KF lies in the determination of process and measurement noise parameters, which collectively can be termed as KF tuning. That is why it is

important to note the effects of overestimating and underestimating both of these parameters. Collecting the results of demonstrations given in [37] for a single target tracking case, effects of wrong KF tuning are as follows. Underestimating process noise and, equivalently, overestimating measurement noise result in time correlated innovation sequence $\boldsymbol{e}_k \triangleq \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k$. That is because the measurements are not given proper importance and estimated states cannot catch up with the real ones. Another observation is that $\boldsymbol{e}_k$ falls too frequently outside the 2σ bounds, which should not happen more frequently than 5% of the time. The converse case, on the other hand, causes $\boldsymbol{e}_k$ to rarely leave the 2σ bounds with no apparent time correlation. One last thing to note about these observations is overestimating the process noise is equivalent to underestimating measurement noise; this was proved in [38] for a general Extended Kalman Filter (EKF) and demonstrated in [39] that it is the ratio of process noise to measurement noise that determines the Kalman gain and, hence, its convergence properties. As the linear KF is a special case of the EKF, it is natural to assume that these properties would be applicable.

3.1.3 Nonlinear Kalman Filters

Standard Kalman filter given in the previous section requires both the system and the measurement functions to be linear, which is rarely the case in most of the practical systems [36]. Its extensions into nonlinear filtering are sub-optimal algorithms that try to approximate the posterior distribution by a Gaussian one using different approximations, several ones of which are discussed below.

3.1.3.1 Extended Kalman Filter

Extended Kalman Filter (EKF) is the most widely used approximation in nonlinear Kalman filtering [36]. It is based on the approximation of the nonlinear function based on its Taylor series expansion. The most commonly used EKF is based on the linearization of the nonlinear function about the mean of the current estimate, and propagate mean and covariance according to the linearized dynamics as in a regular KF. This is sometimes referred as a first order EKF. Depending on the severity of

the nonlinearity involved, one may have to adopt to the its extension, a second order EKF, for a better approximation of the transformed variable at the cost of increased computational complexity. In the light of our problem, even the simulated results with a second order quasi-periodic signal shows EKF1 to be insufficient while EKF2 fails with the increased model order of 5.

Below, a second order EKF where both the process and measurement noises are additive introduced, starting with some notational conventions. First, define the following Jacobian matrices for a scalar function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and for a vector-valued function $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$ w.r.t. a vector $\boldsymbol{x} \in \mathbb{R}^n$. Hessian given in Eq. (3.18c) can be readily obtained by using the equations given below.

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n \tag{3.18a}$$

$$\boldsymbol{f}'(\boldsymbol{x}) \triangleq \nabla_{\boldsymbol{x}} \boldsymbol{f} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \dfrac{\partial f_1}{\partial \boldsymbol{x}} & \dfrac{\partial f_2}{\partial \boldsymbol{x}} & \cdots & \dfrac{\partial f_m}{\partial \boldsymbol{x}} \end{bmatrix} \in \mathbb{R}^{m \times n} \tag{3.18b}$$

$$\boldsymbol{f}''(\boldsymbol{x}) \triangleq \nabla^2_{\boldsymbol{x}} f = \frac{\partial^2 f}{\partial \boldsymbol{x} \partial \boldsymbol{x}^T} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1 \partial x_1} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \tag{3.18c}$$

For $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x; \boldsymbol{P}_x)$, the idea is to approximate the posterior density after using the Taylor series expansion of $\boldsymbol{f}(\boldsymbol{x})$ around the current mean $\boldsymbol{\mu}_x$ perturbed with the covariance $\boldsymbol{P}_x$. Using the results and notation in [19], the following is the Taylor series approximation of the output. Let the notation $[a_i]_i$ denote a vector $\boldsymbol{a}$ whose i^{th}

element is a_i. Similarly, $[A_{ij}]_{ij}$ denotes a matrix $\boldsymbol{A}$ whose $(i,j)^{th}$ element is A_{ij}.

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \cong \boldsymbol{f}(\boldsymbol{\mu}_x) + \boldsymbol{f}'(\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x) + \frac{1}{2} \underbrace{\begin{bmatrix} (\boldsymbol{x} - \boldsymbol{\mu}_x)^T \boldsymbol{f}_1''(\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x) \\ (\boldsymbol{x} - \boldsymbol{\mu}_x)^T \boldsymbol{f}_2''(\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x) \\ \vdots \\ (\boldsymbol{x} - \boldsymbol{\mu}_x)^T \boldsymbol{f}_m''(\boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x) \end{bmatrix}}_{\triangleq [(\boldsymbol{x}-\boldsymbol{\mu}_x)^T \boldsymbol{f}_i''(\boldsymbol{\mu}_x)(\boldsymbol{x}-\boldsymbol{\mu}_x)]_i} + \mathcal{O}(||\boldsymbol{x} - \boldsymbol{\mu}_x||^3) \quad (3.19)$$

Using the terms up to the second order of this expansion, the mean and covariance of $\boldsymbol{y}$ can be approximated as shown in equations (3.20c) and (3.20d), respectively. Using the terms up to first order results in dropping the Hessian terms, and mean & covariance approximations are also shown below for the sake of completeness. For a complete derivation, see equations (5a) to (11b) of [19]. TT1 (Taylor transformation 1) and TT2 (Taylor transformation 2) are abbreviations that the authors used to represent which order of Taylor approximation is used for statistical property transformations. After this approximation, most of the calculations of second order EKF can be obtained in a straightforward manner as shown in Table 3.2, where both the process and the measurement noises are assumed to be additive.

TT1:

$$\boldsymbol{\mu}_y = \boldsymbol{f}(\boldsymbol{\mu}_x) \quad (3.20a)$$

$$\boldsymbol{\Sigma}_y = \boldsymbol{f}'(\boldsymbol{\mu}_x)\boldsymbol{\Sigma}_x(\boldsymbol{f}'(\boldsymbol{\mu}_x))^T \quad (3.20b)$$

TT2:

$$\boldsymbol{\mu}_y = \boldsymbol{f}(\boldsymbol{\mu}_x) + \frac{1}{2}[\mathrm{tr}(\boldsymbol{f}_i''(\boldsymbol{\mu}_x)\boldsymbol{\Sigma}_x)]_i \quad (3.20c)$$

$$\boldsymbol{\Sigma}_y = \boldsymbol{f}'(\boldsymbol{\mu}_x)\boldsymbol{\Sigma}_x(\boldsymbol{f}'(\boldsymbol{\mu}_x))^T + \frac{1}{2}\Big[\ \mathrm{tr}\left(\boldsymbol{f}_i''(\boldsymbol{\mu}_x)\boldsymbol{\Sigma}_x\boldsymbol{f}_j''(\boldsymbol{\mu}_x)\boldsymbol{\Sigma}_x\right)\Big]_{ij} \quad (3.20d)$$

where

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \text{ and } \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x; \boldsymbol{\Sigma}_x) \implies \boldsymbol{y} \overset{approx}{\sim} \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \tag{3.20e}$$

Using higher order terms would result in better mean and covariance estimates; however, in general, it is difficult to obtain even the Hessian matrix let alone the higher order derivatives. On top of that, such a computationally expensive approach may make the tracking problem inexecutable in real time. One last thing to note here is that process and measurement noises are additive in equations (3.24b) and (3.25b) when their covariance matrices are sufficiently larger than the effect of the addition of Hessian based terms (more strictly speaking, when $\boldsymbol{Q}_k$ has much larger eigenvalues than $\frac{1}{2}\,\mathrm{tr}(.)$ term in equation (3.24b), and similarly for $\boldsymbol{R}_k$) even the standard first order EKF would have a comparable performance to its higher counterparts [19]. Hence, in short, when the noise term dominates the time or measurement update covariance terms, even the first order EKF may provide sufficiently well approximations. This is the very same reason that a third order EKF may not compensate for its computational burden with increased accuracy and it is not advised to go further than this order of approximation.

For our system, Jacobian and Hessian calculations are given below. Their calculation is straightforward by the application of Jacobian and Hessian definitions in equations (3.18a) and (3.18c) to our measurement model given in equation (3.7b).

$$h'(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} h = \begin{bmatrix} 1 \\ \cos\theta \\ \cos 2\theta \\ \vdots \\ \cos m\theta \\ \sin 2\theta \\ \vdots \\ \sin m\theta \\ \sum_{i=1}^{m}(-ia_i \sin i\theta) + \sum_{i=2}^{m}(ib_i \cos i\theta) \\ 0 \end{bmatrix} \tag{3.21a}$$

$$h''(\boldsymbol{x}) = \begin{bmatrix} & & & \frac{\partial^2 h}{\partial a_0 \partial \theta} & \\ & \mathbf{0} & & \frac{\partial^2 h}{\partial a_1 \partial \theta} & \mathbf{0} \\ & & & \vdots & \\ & & & \frac{\partial^2 h}{\partial b_m \partial \theta} & \\ \frac{\partial^2 h}{\partial a_0 \partial \theta} & \frac{\partial^2 h}{\partial a_1 \partial \theta} & \cdots \quad \frac{\partial^2 h}{\partial b_m \partial \theta} & \frac{\partial^2 h}{\partial^2 \theta} & 0 \\ & \mathbf{0} & & 0 & 0 \end{bmatrix} \tag{3.21b}$$

$$= \begin{bmatrix} & & & 0 & \\ & & & -\sin\theta & \\ & & & -2\sin 2\theta & \\ & \mathbf{0} & & \vdots & \mathbf{0} \\ & & & -m\sin m\theta & \\ & & & 2\cos 2\theta & \\ & & & \vdots & \\ & & & m\cos m\theta & \\ 0 & -\sin\theta \quad -2\sin 2\theta & \ldots & m\cos m\theta & h_1(\theta) \quad 0 \\ & \mathbf{0} & & 0 & 0 \end{bmatrix} \tag{3.21c}$$

where

$$h_1(\theta) = -\sum_{i=1}^{m}(i^2 a_i \cos i\theta) - \sum_{i=2}^{m}(i^2 b_i \sin i\theta) \tag{3.21d}$$

Finally, note that state transition dynamics are linear in our system, that is $f(\boldsymbol{x}) = \boldsymbol{F}\boldsymbol{x}$. Therefore, $\boldsymbol{f}'(\boldsymbol{x}) = \boldsymbol{F}$ and $\boldsymbol{f}''(\boldsymbol{x}) = \mathbf{0}$ which reduces the second order EKF equations to those of a linear KF.

Table 3.2: Second Order Extended Kalman Filter Algorithm

State Space Model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{3.22a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.22b}$$

Initialization:

$$\mathbb{E}\left\{\boldsymbol{x}_{0|-1}\right\} \triangleq \hat{\boldsymbol{x}}_0 = \mathbb{E}\left\{\boldsymbol{x}_0\right\} \tag{3.23a}$$

$$\boldsymbol{\Sigma}_{0|-1} \triangleq \boldsymbol{\Sigma}_0 = \mathbb{E}\left\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right\} \tag{3.23b}$$

Prediction or Time Update: (A priori state and its covariance estimate)

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_{k|k}) + \frac{1}{2}[\mathrm{tr}(\boldsymbol{f}''_{k,i}(\hat{\boldsymbol{x}}_{k|k})\boldsymbol{\Sigma}_{k|k})]_i \tag{3.24a}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{k+1|k} &= \boldsymbol{f}'_k(\hat{\boldsymbol{x}}_{k|k})\boldsymbol{\Sigma}_{k|k}(\boldsymbol{f}'_k(\hat{\boldsymbol{x}}_{k|k}))^T \\ &\quad + \frac{1}{2}\Big[\,\mathrm{tr}\left(\boldsymbol{f}''_{k,i}(\hat{\boldsymbol{x}}_{k|k})\boldsymbol{\Sigma}_{k|k}\boldsymbol{f}''_{k,j}(\hat{\boldsymbol{x}}_{k|k})\boldsymbol{\Sigma}_{k|k}\right)\Big]_{ij} + \boldsymbol{Q}_k\end{aligned} \tag{3.24b}$$

Kalman Gain:

$$\hat{\boldsymbol{y}}_{k+1|k} = \boldsymbol{h}_{k+1}(\boldsymbol{x}_{k+1|k}) + \frac{1}{2}[\mathrm{tr}(\boldsymbol{h}''_{k+1,i}\boldsymbol{\Sigma}_{k+1|k})] \tag{3.25a}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{y_{k+1|k}} &= \boldsymbol{h}'_{k+1}(\hat{\boldsymbol{x}}_{k+1|k})\boldsymbol{\Sigma}_{k+1|k}(\boldsymbol{h}'_{k+1}(\hat{\boldsymbol{x}}_{k+1|k}))^T \\ &\quad + \frac{1}{2}\Big[\,\mathrm{tr}\left(\boldsymbol{h}''_{k+1,i}(\hat{\boldsymbol{x}}_{k+1|k})\boldsymbol{\Sigma}_{k+1|k}\boldsymbol{h}''_{k+1,j}(\hat{\boldsymbol{x}}_{k+1|k})\boldsymbol{\Sigma}_{k+1|k}\right)\Big] \\ &\quad + \boldsymbol{R}_{k+1}\end{aligned} \tag{3.25b}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_{k+1|k}(\boldsymbol{h}'_{k+1})^T(\boldsymbol{x}_{k+1|k})\boldsymbol{\Sigma}^{-1}_{y_{k+1|k}} \tag{3.25c}$$

Filtering or Measurement Update: (A posteriori state and its covariance estimate)

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}) \tag{3.26a}$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = (\boldsymbol{\mathcal{I}} - \boldsymbol{K}_{k+1}\boldsymbol{h}'_{k+1}(\hat{\boldsymbol{x}}_{k+1|k}))\boldsymbol{\Sigma}_{k+1|k} \tag{3.26b}$$

Assumptions:

$$\boldsymbol{x}_0 \sim \mathcal{N}(\hat{\boldsymbol{x}}_0; \boldsymbol{\Sigma}_0) \tag{3.27a}$$

$$\boldsymbol{w}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}; \boldsymbol{Q}_k) \tag{3.27b}$$

$$\boldsymbol{v}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}; \boldsymbol{R}_k) \tag{3.27c}$$

3.1.3.2 Unscented Kalman Filter

EKF approximations are based on linearization of nonlinear functions around the current mean without considering the uncertainty (or spread) associated with the current estimates. On top of this, they may lead to inconsistent covariance estimates; that is, they may fail to satisfy the positive semidefiniteness of left hand side of equation (3.28) [40]. This underestimation of covariance matrix $\mathbf{\Sigma}_y$ would result in a large Kalman gain and may lead to filter divergence. Overestimating $\mathbf{\Sigma}_y$ should be preferred provided that it is not greatly in excess of the actual one; greatly overestimating would have caused a lag in tracking true states as was discussed in KF explanations for overestimating measurement noise, and equivalently, $\mathbf{\Sigma}_y$.

$$\mathbf{\Sigma}_y - \mathbb{E}\left\{(\boldsymbol{y} - \hat{\boldsymbol{y}})(\boldsymbol{y} - \hat{\boldsymbol{y}})^T\right\} \geq \mathbf{0} \tag{3.28}$$

As an alternative to linearization, Julier et al. [41] introduced the Unscented Transform (UT) to deal with this problem. Their intuition was that "with a fixed number of parameters it should be easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function transformation". In UT (or any other sigma point based transformations that are used in Cuberature Kalman Filter, Central Difference Kalman Filter, etc.) the idea is to generate a carefully selected deterministic set of sigma points around the current mean with their spread controlled by the current uncertainty. These sigma points are then passed through the nonlinearity and weighted average of these are used to calculate the mean and variance of the output as discussed in equations (3.29a)-(3.29n) and as depicted in 3.1.3.2 (the depiction representing the 2 dimensional case).

Equation of interest:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{3.29a}$$

Define the Cholesky decomposition of a PSD matrix as its numerically stable and efficient square root:

$$\mathbf{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T \implies \boldsymbol{L} = \sqrt{\mathbf{\Sigma}} \tag{3.29b}$$

Generate $2n_x + 1$ sigma points around the current mean of $\boldsymbol{x}$.

$$\boldsymbol{X}_0 = \boldsymbol{\mu}_x \tag{3.29c}$$

$$\boldsymbol{X}_i = \boldsymbol{\mu}_x + \left(\sqrt{(n_x + \lambda)\boldsymbol{\Sigma}_x}\right)_i \qquad i = 1, \ldots, n_x \tag{3.29d}$$

$$\boldsymbol{X}_{i+n_x} = \boldsymbol{\mu}_x - \left(\sqrt{(n_x + \lambda)\boldsymbol{\Sigma}_x}\right)_i \qquad i = 1, \ldots, n_x \tag{3.29e}$$

Associated weights for the sigma points (where superscripts indicate whether they will be used for mean or covariance calculation):

$$W_0^{(m)} = \frac{\lambda}{n_x + \lambda} \qquad \text{(central mean weight)} \tag{3.29f}$$

$$W_0^{(c)} = W_0^{(m)} + (1 - \alpha^2 + \beta) \qquad \text{(central covariance weight)} \tag{3.29g}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n_x + \lambda)} \qquad i = 1, \ldots, 2n_x \tag{3.29h}$$

Calculation of the sample mean and sample covariance of $\boldsymbol{y}$:

$$\boldsymbol{\mu}_y = \sum_{i=0}^{2n_x} W_i^{(m)} \boldsymbol{f}(\boldsymbol{X}_i) \tag{3.29i}$$

$$\boldsymbol{\Sigma}_y = \sum_{i=0}^{2n_x} W_i^{(c)} (\boldsymbol{f}(\boldsymbol{X}_i) - \boldsymbol{\mu}_y)(\boldsymbol{f}(\boldsymbol{X}_i) - \boldsymbol{\mu}_y)^T \tag{3.29j}$$

Parameters to be tuned:

λ : resulting scaling parameter controlling the distribution of sigma points

$$\lambda = \alpha^2(n_x + \kappa) - n_x \tag{3.29k}$$

β : parameter to be scaled according to the distribution of $\boldsymbol{x}$

$\beta = 2$ for Gaussian distributions (3.29l)

α : primary scaling parameter usually set to a small positive value

$$\alpha \in [10^{-3}, 1] \tag{3.29m}$$

κ : secondary scaling parameter

$\kappa = 0$ or $\kappa = 3 - n_x$ usual choice for κ acc. to [41] or [8] (3.29n)

There is no general consensus as to which parameter selection would result in better

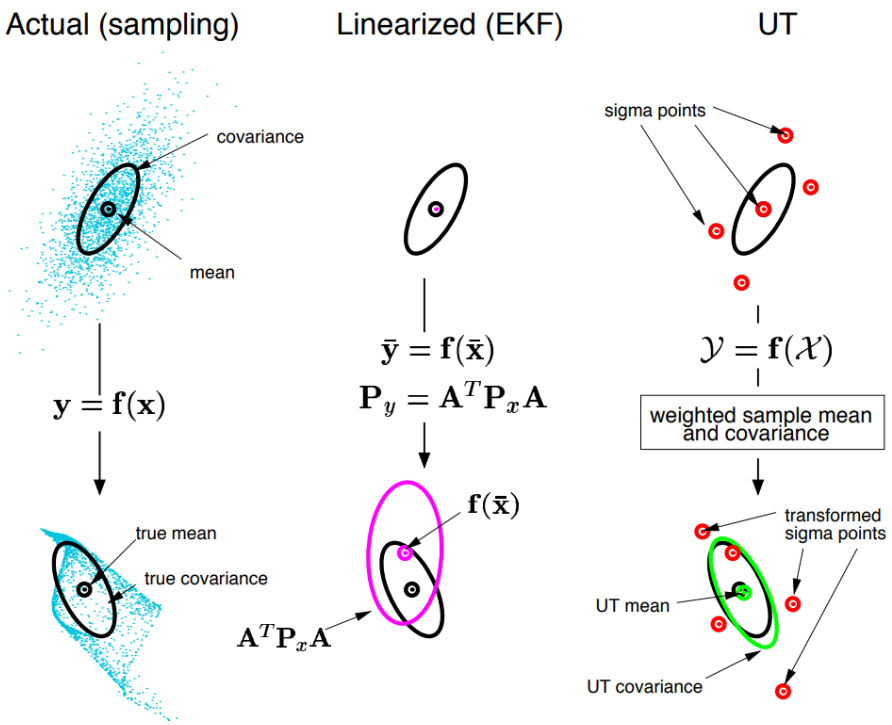

Figure 3.2: Actual (sampling based), first order linearization and UT based mean and covariance propagation (obtained from [8])

capturing the statistical properties of $\boldsymbol{y}$. For example, the choice of $\kappa = 3 - n_x$ given in [41], it is guaranteed that mean and covariance estimation of $\boldsymbol{y}$ are correct up to and including the third order moment of $\boldsymbol{x}$ in Taylor series based approximations. This seems fine unless the value of κ may become negative, which may result in $\lambda < 0$ and consequently $\boldsymbol{\Sigma}_y \not\succ 0$. Also, as was demonstrated in [9] for tracking a single sine wave, $\alpha = 1$, $\beta = 2$ and $\kappa = 0$ resulted in far better mean and covariance estimates than the previous case (provided that the input standard deviation is bounded by 2; their results can be seen in Fig. 3.3.). Thus, the optimal choice of their values are problem specific and should be examined for the problem at hand. We will use their tuning values for the UKF. With this choice, λ in equation 3.29k becomes 0; central and non-central covariance weights in equations 3.29g and 3.29h become 2 and $\frac{1}{n_x}$, respectively. Then, the positive semidefiniteness of $\boldsymbol{\Sigma}_y$ is asserted by having these covariance weights as non-negative [42]. The non-augmented UKF equations are summarized in Table 3.3.

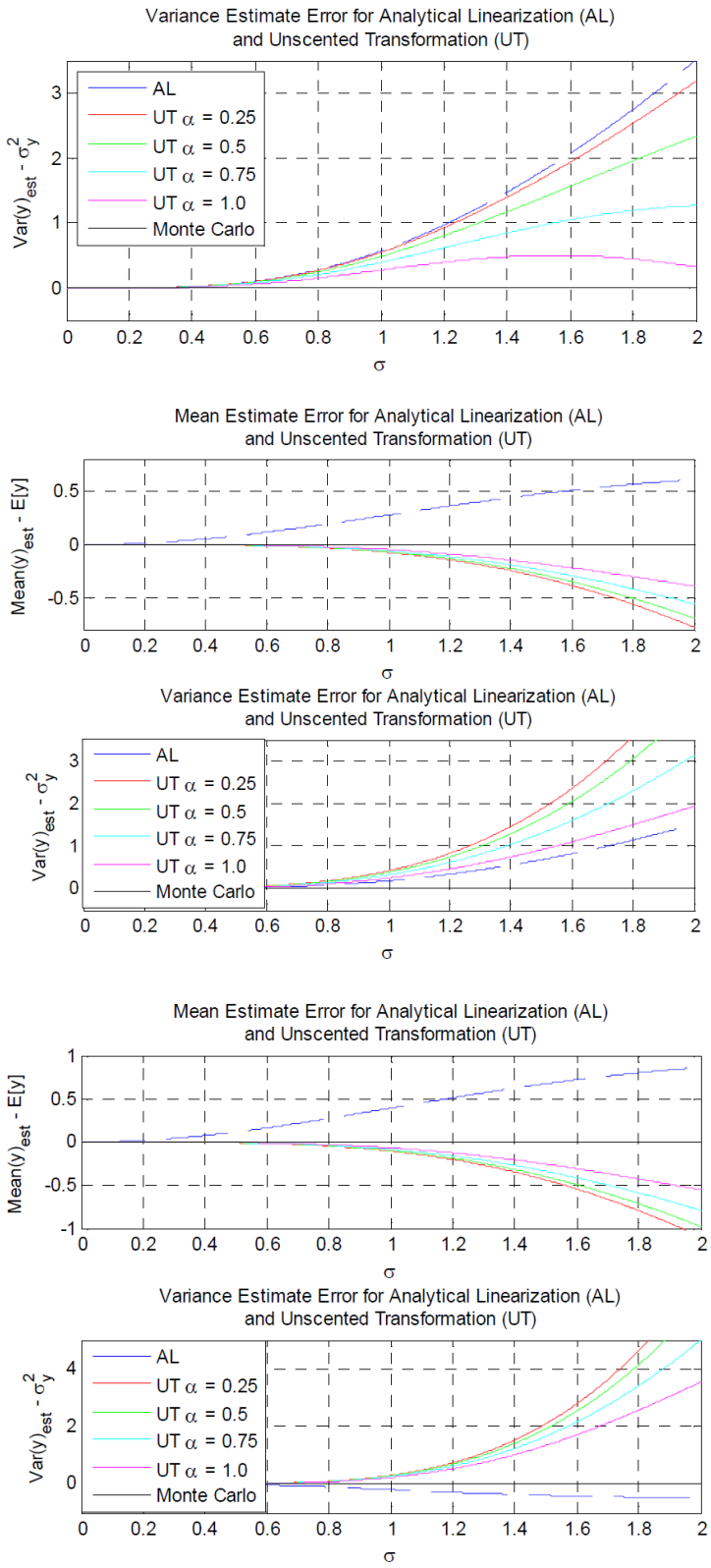

Figure 3.3: Mean and variance estimation provided by TT1 and UT for $y = \sin x$ for $\mu_x = 0$ (top), $\pi/4$ (middle) and $\pi/2$ (bottom) as a function of σ_x, obtained from [9]

Table 3.3: Unscented Kalman Filter Algorithm

State Space Model:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{3.30a}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.30b}$$

Initialization:

$$\mathbb{E}\left\{\boldsymbol{x}_{0|-1}\right\} \triangleq \hat{\boldsymbol{x}}_0 = \mathbb{E}\left\{\boldsymbol{x}_0\right\} \tag{3.31a}$$

$$\boldsymbol{\Sigma}_{0|-1} \triangleq \boldsymbol{\Sigma}_0 = \mathbb{E}\left\{(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right\} \tag{3.31b}$$

Generate sigma points according to equations (3.29c) - (3.29e) with $\boldsymbol{\mu}_x = \hat{\boldsymbol{x}}_{k|k}$ and $\boldsymbol{\Sigma}_x = \hat{\boldsymbol{\Sigma}}_{k|k}$.

Prediction or Time Update: (A priori state and its covariance estimate)

$$\hat{\boldsymbol{X}} = \boldsymbol{f}_{k+1}(\boldsymbol{f}(\boldsymbol{X})), \qquad \hat{\boldsymbol{x}}_{k+1|k} = \sum_{i=0}^{2n_x} W_i^{(m)} \hat{\boldsymbol{X}}_i \tag{3.32a}$$

$$\boldsymbol{\Sigma}_{k+1|k} = \sum_{i=0}^{2n_x} W_i^{(c)} (\hat{\boldsymbol{X}}_i - \hat{\boldsymbol{x}}_{k+1|k})(\hat{\boldsymbol{X}}_i - \hat{\boldsymbol{x}}_{k+1|k})^T \tag{3.32b}$$

Kalman Gain (After the regeneration of sigma points):

$$\hat{\boldsymbol{Y}} = \boldsymbol{h}_{k+1}(\boldsymbol{f}(\boldsymbol{X}_i)), \qquad \hat{\boldsymbol{y}}_{k+1|k} = \sum_{i=0}^{2n_x} W_i^{(m)} \hat{\boldsymbol{Y}}_i \tag{3.33a}$$

$$\boldsymbol{\Sigma}_{y_{k+1|k}} = \sum_{i=0}^{2n_x} W_i^{(c)} (\hat{\boldsymbol{Y}}_i - \hat{\boldsymbol{y}}_{k+1|k})(\hat{\boldsymbol{Y}}_i - \hat{\boldsymbol{y}}_{k+1|k}) + \boldsymbol{R}_{k+1} \tag{3.33b}$$

$$\boldsymbol{\Sigma}_{xy_{k+1|k}} = \sum_{i=0}^{2n_x} W_i^{(c)} (\hat{\boldsymbol{X}}_i - \hat{\boldsymbol{x}}_{k+1|k})(\hat{\boldsymbol{Y}}_i - \hat{\boldsymbol{y}}_{k+1|k}) \tag{3.33c}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_{xy_{k+1|k}} \boldsymbol{\Sigma}_{y_{k+1|k}}^{-1} \tag{3.33d}$$

Filtering or Measurement Update: (A posteriori state and its covariance estimate)

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}) \tag{3.34a}$$

$$\boldsymbol{\Sigma}_{k+1|k+1} = \boldsymbol{\Sigma}_{k+1|k} - \boldsymbol{K}_{k+1} \boldsymbol{\Sigma}_{y_{k+1|k}}^{-1} \boldsymbol{K}_{k+1}^T \tag{3.34b}$$

3.2 Exact Moment Matching Kalman Filtering

All of the Kalman filters previously mentioned (even those that are slightly touched upon as sigma point based Kalman filters) attribute their superior success in tracking to less erroneous mean and covariance estimation. The choice of $\beta = 2$ being optimal for a GRV x was demonstrated by the authors of the original UKF [41] for perfectly estimating the mean and covariance of $y = x^2$. This has led us to think that, the closer one gets to the true mean and covariance, the better estimation performance is expected to be obtained. Beginning with this thought, the following will be a collection of the results found in the literature. Note that the restriction that the output of nonlinear transformation being Gaussian is still not loosened in this section, so the filter described is still suboptimal but should be better than those described previously.

Downsides of EKF were previously touched open under Section 3.1.3.2, which are (i) disregarding the current uncertainty associated with current estimates during linearization and (ii) possible divergence due to underestimation of Σ_y. Two of the shortcomings of UKF was specified as tuning α, β and K which is not straightforward and is problem specific (although it may be argued that it is more straightforward than the derivations to be shown later whose benefit is the increased accuracy in tracking). Moreover, the tuning is not only problem specific but also depends on the state estimation variances. Consider, for example, the nonlinear transformation given in Eq. (3.35a), which is part of our measurement model. If the input standard deviation were limited to 2, the previously stated parameter choice results in the best covariance estimates with $\alpha = 1$. However, if one considers the distribution given in Eq. (3.35b), it is clear that sample covariance estimate tends towards 0 as the transformed sigma points are equal to each other, which happens due to *sigma point collapse*. A caption from [10] has a clear demonstration of this case for an almost exact sigma point collapse, which can be seen in Fig. 3.4 with sigma points depicted with the red crosses and approximated pdf with the red pdf curve. By performing exact moment matching, our aim is to obtain a distribution close to the one shown with the green curve, not the erroneous one that can be obtained from the wrongly tuned UKF. Note that multimodality of the output is still visible (with two clear peaks at the output) and can be represented better with the sum of two Gaussian variables; however, we

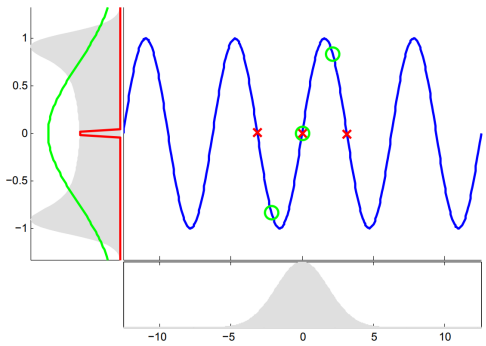

Figure 3.4: UKF sigma point collapse obtained from [10]

will still approximate the output as unimodal as a simplifying assumption. We will show in chapter 4 that even with this approximation, exact moment matching still outperforms the approximations of Taylor transformation or unscented transform.

$$y = g(x) = \sin(x) \tag{3.35a}$$

$$x \sim \mathcal{N}(0; \pi^2) \tag{3.35b}$$

$$\lambda = \alpha^2(n_x + K) - n_x = \alpha^2 - 1 \implies \sqrt{(n_x + \lambda)\sigma_x^2} = \alpha\sigma_x \tag{3.35c}$$

$$\boldsymbol{X} = \begin{bmatrix} g(\mu_x) & g(\mu_x + \alpha\sigma_x) & g(\mu_x - \alpha\sigma_x) \end{bmatrix} = \begin{bmatrix} \sin(0) & \sin(\pi) & \sin(-\pi) \end{bmatrix} = \mathbf{0} \tag{3.35d}$$

3.2.1 Monte Carlo Transformation

Let us first define the Monte Carlo Transformation (MCT) that can be used for the mean and covariance estimation. Given a sufficiently large number of samples, MCT is asymptotically the best transformation in calculating the parameters of interest thanks to the law of large numbers [19]. "Given a sufficiently large number of samples" is naturally the root cause of its computational burden. However, the idea behind is quite simple and it is to generate a large number of samples from a normal distribution, pass them through the nonlinearity and calculate the sample mean and

covariance according to the equations shown below.

$$\boldsymbol{x}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_x; \boldsymbol{\Sigma}_x) \qquad i = 1, 2, \ldots, N \tag{3.36a}$$

$$\boldsymbol{y}^{(i)} = f(\boldsymbol{x}^{(i)}) \tag{3.36b}$$

$$\boldsymbol{\mu}_y = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{y}^{(i)} \tag{3.36c}$$

$$\boldsymbol{\Sigma}_y = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{y}^{(i)} - \boldsymbol{\mu}_y)(\boldsymbol{y}^{(i)} - \boldsymbol{\mu}_y)^T \tag{3.36d}$$

3.2.2 Moment Matching Kalman Filters

Luca et al. [43] first developed an exact Kalman filter for polynomial systems. They have developed a general framework for such systems and have given analytical expressions for mean and covariance transformation through Chebyshev polynomials of order up to 4. They have demonstrated that (i) EKF fails in the estimation of transformed statistics even under mild nonlinearities, (ii) UKF greatly improves these results, and (iii) exact transformation gives the correct output which totally agrees with the MCT described above. They have also, through simulations, shown that $\boldsymbol{\Sigma}_y^{EKF} \leq \boldsymbol{\Sigma}_y^{UKF} \leq \boldsymbol{\Sigma}_y^{ExPKF}$ (ExPKF refers to Exact Polynomial KF). What this means is that EKF and UKF gives more optimistic results than the ExPKF, which may lead to local filter divergence. Another consequence is that associated Kalman gains follow a similar pattern, resulting in ExPKF having the most superior performance in tracking. In a separate work, Saha et al. [44] independently developed what is actually the same filter with some different terminology (for interested readers they named it as KF with Exact Moment Matching), and they have also demonstrated that this filter has far better RMSE estimates than EKF/UKF. What is important and different in their work is that the performance of the new filter (in RMSE sense) is found to be comparable to Unscented Particle Filter (refer to the previous source [8] for details) with much less computation time (approximately 1% of the particle filter).

In a completely separate work, Rhudy et al. [9] revisited the same problem with the addition of several nonlinear functions to enhance this method. Their derivation

started with a zero mean Gaussian r.v.. Using the following equations, they have first found the mean and variance of $\sin x$ and $\cos x$. A simple trick used in the last step of Eq. (3.37g) is the fact that if $x \sim \mathcal{N}(\mu; \sigma^2)$ then $2x \sim \mathcal{N}(2\mu; 4\sigma^2)$.

$$x \sim \mathcal{N}(0; \sigma^2) \tag{3.37a}$$

$$\implies f(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{3.37b}$$

$$\implies M(r) = \mathbb{E}\left\{e^{rx}\right\} \triangleq \int_{-\infty}^{\infty} e^{rx} f(x) dx = e^{\frac{1}{2}\sigma^2 r^2} \tag{3.37c}$$

kth order moment (where !! is the double factorial):

$$\mathbb{E}\left\{x^k\right\} = \left.\frac{\partial M(r)}{\partial r}\right|_{r=0} = \begin{cases} 0 & \text{for } k \text{ odd} \\ \sigma^k (k-1)!! & \text{for } k \text{ even} \end{cases} \tag{3.37d}$$

Then,

$$\mathbb{E}\left\{\sin x\right\} = \mathbb{E}\left\{\sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)!}\right\} = \sum_{i=0}^{\infty} \frac{(-1)^i \cancelto{0}{\mathbb{E}\left\{x^{2i+1}\right\}}}{(2i+1)!} = 0 \tag{3.37e}$$

$$\begin{aligned} \mathbb{E}\left\{\cos x\right\} &= \mathbb{E}\left\{\sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!}\right\} = \sum_{i=0}^{\infty} \frac{(-1)^i \mathbb{E}\left\{x^{2i}\right\}}{(2i)!} \\ &= \sum_{i=0}^{\infty} \frac{(-1)^i \sigma^{2i} (2i-1)!!}{(2i)!} = \sum_{i=0}^{\infty} \frac{(-1)^i \sigma^{2i}}{(2i)!!} = e^{-\frac{\sigma^2}{2}} \end{aligned} \tag{3.37f}$$

$$\mathrm{var}(\sin x) = \mathbb{E}\left\{\sin^2 x\right\} - \cancelto{0}{\mathbb{E}\left\{\sin x\right\}^2} = \mathbb{E}\left\{\frac{1 - \cos 2x}{2}\right\} = \frac{1}{2}(1 - e^{-2\sigma^2}) \tag{3.37g}$$

$$\begin{aligned} \mathrm{var}(\cos x) &= \mathbb{E}\left\{\cos^2 x\right\} - \mathbb{E}\left\{\cos x\right\}^2 = \mathbb{E}\left\{\frac{1 + \cos 2x}{2}\right\} - e^{-\sigma^2} \\ &= \frac{1}{2}(1 + e^{-2\sigma^2}) - e^{-\sigma^2} = \frac{1}{2}(1 - e^{-\sigma^2})^2 \end{aligned} \tag{3.37h}$$

Extensions of these for non-zero mean case follows the trivial trigonometric identities

as follows.

$$x \sim \mathcal{N}(\mu; \sigma^2) \text{ and } x_1 \triangleq x - \mu, \implies x_1 \sim \mathcal{N}(0; \sigma^2) \tag{3.38a}$$

$$\begin{aligned}\mathbb{E}\{\sin x\} &= \mathbb{E}\{\sin(x_1 + \mu)\} = \mathbb{E}\left\{\cancelto{0}{\sin x_1}\cos\mu + \cos x_1 \sin\mu\right\} \\ &= \sin\mu e^{-\frac{\sigma^2}{2}}\end{aligned} \tag{3.38b}$$

$$\begin{aligned}\mathbb{E}\{\cos x\} &= \mathbb{E}\{\cos(x_1 + \mu)\} = \mathbb{E}\left\{\cos x_1 \cos\mu - \cancelto{0}{\sin x_1}\sin\mu\right\} \\ &= \cos\mu e^{-\frac{\sigma^2}{2}}\end{aligned} \tag{3.38c}$$

$$\begin{aligned}\text{var}(\sin x) &= \mathbb{E}\left\{\sin^2(x_1 + \mu)\right\} - \mathbb{E}\{\sin(x_1 + \mu)\}^2 \\ &= \mathbb{E}\left\{\frac{1 - \cos(2x_1 + 2\mu)}{2}\right\} - \sin^2\mu e^{-\sigma^2} \\ &= \frac{1}{2}(1 - \cos 2\mu e^{-2\sigma^2}) - e^{-\sigma^2}\frac{1 - \cos 2\mu}{2} \\ &= \frac{1}{2}(1 - e^{-\sigma^2})(1 + \cos 2\mu e^{-\sigma^2})\end{aligned} \tag{3.38d}$$

$$\begin{aligned}\text{var}(\cos x) &= \mathbb{E}\left\{\cos^2(x_1 + \mu)\right\} - \mathbb{E}\{\cos(x_1 + \mu)\}^2 \\ &= \mathbb{E}\left\{\frac{1 + \cos(2x_1 + 2\mu)}{2}\right\} - \cos^2\mu e^{-\sigma^2} \\ &= \frac{1}{2}(1 + \cos 2\mu e^{-2\sigma^2}) - e^{-\sigma^2}\frac{1 + \cos 2\mu}{2} \\ &= \frac{1}{2}(1 - e^{-\sigma^2})(1 - \cos 2\mu e^{-\sigma^2})\end{aligned} \tag{3.38e}$$

3.2.3 Our Moment Matching Kalman Filter

What was done in the original paper [9] was only with a single sinusoidal function for a single state. Now, we will extend their work for a more general case to include the effect of higher order harmonics and more than one states. In order for us to run the Kalman filter equation, we need to find the Kalman gain which requires finding $\mathbf{\Sigma}_{xy}$ and $\mathbf{\Sigma}_y$. Also expected value of y, $\hat{y}$, will be needed in the Kalman update step. After finding these, the equations given in 3.5 are directly available from our state formulation.

Starting with the simple case of $\hat{y} = \hat{y}_{k+1|k}$, it is given as in Eq. (3.39a). Note that

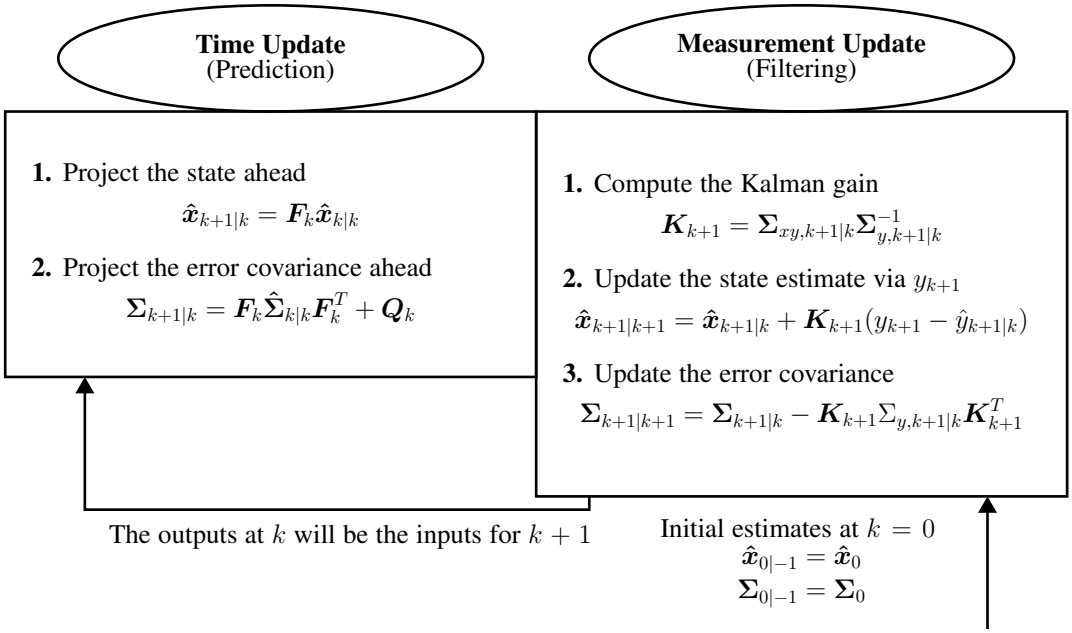

Figure 3.5: Kalman filter equations

for deterministic a_i and random θ, this also verifies the result in equations (3.38b)-(3.38c).

$$\hat{y} = \sum_{i=0}^{m} \mathbb{E}\left\{a_i \cos i\theta\right\} + \sum_{i=2}^{m} \mathbb{E}\left\{b_i \sin i\theta\right\} \tag{3.39a}$$

where

$$\mathbb{E}\left\{a_i \cos i\theta\right\} = (\mu_{a_i} \cos i\mu_\theta - i\Sigma_{a_i\theta} \sin i\mu_\theta) e^{-\frac{i^2\Sigma_\theta}{2}} \tag{3.39b}$$

$$\mathbb{E}\left\{b_i \sin i\theta\right\} = (\mu_{b_i} \sin i\mu_\theta + i\Sigma_{b_i\theta} \cos i\mu_\theta) e^{-\frac{i^2\Sigma_\theta}{2}} \tag{3.39c}$$

Proof: Moment generating function of a Gaussian random vector is given by (3.40a) [45]. Letting $\boldsymbol{x} = \begin{bmatrix} a_i & \theta \end{bmatrix}^T$, $\boldsymbol{t} = \begin{bmatrix} t_1 & t_2 \end{bmatrix}^T$, taking the partial derivatives w.r.t. t_1 and evaluating at $t_1 = 0$ yields (3.40b). Letting $t_2 = ji$ where $j = \sqrt{-1}$ and i is an integer, we found the desired results by equating the real and imaginary parts.

$$M_{\boldsymbol{x}}(\boldsymbol{t}) = \mathbb{E}\left\{e^{\boldsymbol{t}^T\boldsymbol{x}}\right\} = e^{\boldsymbol{t}^T\boldsymbol{\mu}_x + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{\Sigma}_x\boldsymbol{t}} \tag{3.40a}$$

$$\left.\frac{\partial M_{\boldsymbol{x}}(\boldsymbol{t})}{\partial t_1}\right|_{t_1=0} = \mathbb{E}\left\{a_i e^{t_2\theta}\right\} = (\mu_{a_i} + \Sigma_{a_i\theta} t_2) e^{t_2\mu_\theta + \frac{1}{2}t_2^2\Sigma_\theta} \tag{3.40b}$$

$$\begin{aligned}\left.\frac{\partial M_{\boldsymbol{x}}(\boldsymbol{t})}{\partial t_1}\right|_{t_1=0,t_2=ji} &= \mathbb{E}\left\{a_i\cos i\theta + j a_i \sin i\theta\right\} \\ &= (\mu_{a_i} + \Sigma_{a_i\theta} ji)(\cos i\mu_\theta + j\sin i\mu_\theta) e^{-\frac{1}{2}i^2\Sigma_\theta}\end{aligned} \tag{3.40c}$$

Working on a bit more complicated scenario, we will show that $\operatorname{cov}(x, a_k\cos k\theta)$ is given by (3.41a). Variable x can be equal to any one of the state variables.

$$\begin{aligned}\operatorname{cov}(x, a_k\cos\theta_k k) = \Big((\Sigma_{xa_k} - k^2\Sigma_{x\theta}\Sigma_{a_k\theta})\cos k\mu_\theta \\ - k\Sigma_{x\theta}\mu_{a_k}\sin k\mu_\theta\Big) e^{-\frac{k^2\Sigma_\theta}{2}}\end{aligned} \tag{3.41a}$$

$$\begin{aligned}\operatorname{cov}(x, b_k\sin k\theta) = \Big((\Sigma_{xb_k} - k^2\Sigma_{x\theta}\Sigma_{b_k\theta})\sin k\mu_\theta \\ + k\Sigma_{x\theta}\mu_{a_k}\cos k\mu_\theta\Big) e^{-\frac{k^2\Sigma_\theta}{2}}\end{aligned} \tag{3.41b}$$

Proof: Again start with a general MGF of a Gaussian vector with $\boldsymbol{x} = \begin{bmatrix} x & a_k & \theta \end{bmatrix}^T$, $\boldsymbol{t} = \begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}^T$. Differentiating w.r.t. t_1 and evaluating at $t_1 = 0$ yields (3.42a); repeating the same for t_2 gives (3.42b). Finally, setting $t_3 = jk$, separating real & imaginary parts as before and subtracting $\mu_{a_i}\mu_{a_k}\cos k\theta$ results in (3.41a).

$$\left.\frac{\partial M_{\boldsymbol{x}}(\boldsymbol{t})}{\partial t_1}\right|_{t_1=0} = \mathbb{E}\left\{x e^{t_2 a_k + t_3\theta}\right\} = (\mu_x + t_2\Sigma_{xa_k} + t_3\Sigma_{x\theta})\; M_{\boldsymbol{x}}(\boldsymbol{t})|_{t_1=0} \tag{3.42a}$$

$$\begin{aligned}\left.\frac{\partial^2 M_{\boldsymbol{x}}(\boldsymbol{t})}{\partial t_1\partial t_2}\right|_{t_1=t_2=0} &= \mathbb{E}\left\{x a_k e^{t_3\theta}\right\} \\ &= \left(\Sigma_{xa_k} + (\mu_x + t_3\Sigma_{x\theta})(\mu_{a_k} + t_3\Sigma_{a_k\theta})\right) e^{t_3\mu_\theta + \frac{t_3^2\Sigma_\theta}{2}}\end{aligned} \tag{3.42b}$$

$$\begin{aligned}\left.\frac{\partial^2 M_{\boldsymbol{x}}(\boldsymbol{t})}{\partial t_1\partial t_2}\right|_{t_1=t_2=0,t_3=jk} &= \mathbb{E}\left\{x a_k\cos\theta + j x a_k\sin\theta\right\} \\ &= \left(\Sigma_{xa_k} + (\mu_x + jk\Sigma_{x\theta})(\mu_{a_k} + jk\Sigma_{a_k\theta})\right) \\ &\cdot (\cos\mu_\theta + j\sin\mu_\theta) e^{\frac{-k^2\Sigma_\theta}{2}}\end{aligned} \tag{3.42c}$$

$$\mathbb{E}\left\{xa_k \cos k\theta\right\} = \Big[(\Sigma_{xa_k} + \mu_x\mu_{a_k} - k^2\Sigma_{x\theta}\Sigma_{a_k\theta}) \cos k\mu_\theta - k(\Sigma_{x\theta}\mu_{a_k} + \Sigma_{a_k\theta}\mu_x) \sin k\mu_\theta\Big] e^{-\frac{1}{2}k^2\sigma^2} \tag{3.42d}$$

$$\mathbb{E}\left\{xb_k \sin k\theta\right\} = \Big[(\Sigma_{xb_k} + \mu_x\mu_{b_k} - k^2\Sigma_{x\theta}\Sigma_{b_k\theta}) \sin k\mu_\theta + k(\Sigma_{x\theta}\mu_{b_k} + \Sigma_{b_k\theta}\mu_x) \cos k\mu_\theta\Big] e^{-\frac{1}{2}k^2\sigma^2} \tag{3.42e}$$

Then, using the linearity of covariance (as a direct result of linearity of expectation), we found state and output correlation vector given as below.

$$\mathbf{\Sigma}_{xy,k+1|k} = \sum_{i=0}^{m} \begin{bmatrix} \mathrm{cov}(a_0, a_i \cos i\theta) \\ \vdots \\ \mathrm{cov}(\omega, a_i \cos i\theta) \end{bmatrix} + \sum_{i=2}^{m} \begin{bmatrix} \mathrm{cov}(a_0, b_i \sin i\theta) \\ \vdots \\ \mathrm{cov}(\omega, b_i \sin i\theta) \end{bmatrix} \tag{3.43}$$

Finally, we calculated the last parameter needed to complete the filtering equations, which is $\Sigma_y = \mathrm{var}(y)$. The trick used at equation (3.44d) is similar to the one used in [9], where the multiplication is converted into addition using trigonometric identities. Expectations in (3.44b) can be obtained from (3.42b). The final expectations are readily obtainable from Eq. (3.42d) and (3.42e) with a simple change of indices. Overall filtering equations are then given in Fig. 3.5. Note that the prediction equations are linear, and regular KF prediction steps are applicable.

$$\begin{aligned} \Sigma_{y,k+1|k} &= \sum_{i=0}^{m} \mathrm{cov}(a_i \cos i\theta, y) + \sum_{i=2}^{m} \mathrm{cov}(b_i \sin i\theta, y) + R \\ &= \sum_{i=0}^{m}\sum_{j=0}^{m} \mathrm{cov}(a_i \cos i\theta, a_j \cos j\theta) + \sum_{i=2}^{m}\sum_{j=2}^{m} \mathrm{cov}(b_i \sin i\theta, b_j \sin j\theta) \\ &\quad + 2\sum_{i=0}^{m}\sum_{j=2}^{m} \mathrm{cov}(a_i \cos i\theta, b_j \sin j\theta) + R \end{aligned} \tag{3.44a}$$

where

$$\begin{aligned} \mathrm{cov}(a_i \cos i\theta, a_j \cos j\theta) &= \mathbb{E}\left\{a_i \cos i\theta a_j \cos j\theta\right\} - \mu_{a_i \cos i\theta}\mu_{a_j \cos j\theta} \\ &= 0.5\mathbb{E}\left\{a_i a_j \cos(i-j)\theta\right\} + 0.5\mathbb{E}\left\{a_i a_j \cos(i+j)\theta\right\} - \mu_{a_i \cos i\theta}\mu_{a_j \cos j\theta} \end{aligned} \tag{3.44b}$$

$$\begin{aligned}\operatorname{cov}(a_i \cos i\theta, b_j \sin j\theta) &= \mathbb{E}\left\{a_i \cos i\theta b_j \sin j\theta\right\} - \mu_{a_i \cos i\theta}\mu_{b_j \sin j\theta} \\ &= 0.5\mathbb{E}\left\{a_i b_j \sin(i+j)\theta\right\} - 0.5\mathbb{E}\left\{a_i b_j \sin(i-j)\theta\right\} - \mu_{a_i \cos i\theta}\mu_{b_j \sin j\theta}\end{aligned} \tag{3.44c}$$

$$\begin{aligned}\operatorname{cov}(b_i \sin i\theta, b_j \sin j\theta) &= \mathbb{E}\left\{b_i \sin i\theta b_j \sin j\theta\right\} - \mu_{b_i \sin i\theta}\mu_{b_j \sin j\theta} \\ &= 0.5\mathbb{E}\left\{b_i b_j \cos(i-j)\theta\right\} - 0.5\mathbb{E}\left\{b_i b_j \cos(i+j)\theta\right\} - \mu_{b_i \sin i\theta}\mu_{b_j \sin j\theta}\end{aligned} \tag{3.44d}$$

3.2.3.1 MCT Verification

One of the most prevalent criticism that the EKF gets is that obtaining and coding of Hessian & Jacobian matrices is error prone. This is even more valid for the ExKF described above, both for the derivation and the coding. This situation gets even more severe when one tries to make the computation memory bound by avoiding repetitive calculations of $\sin(.)$, $\cos(.)$ and $e^{(.)}$ and storing their values in an array. That is why some verification of the results would be beneficial. In this subsection, we will demonstrate that the calculation of the exact mean and covariance matches the one given by MCT. Number of Monte Carlo points were chosen as $N = 30 \cdot 10^6$. The procedure to do so is as follows.

1. Generate a random vector $v \in \mathbb{R}^{2m+2 \times 2m+2}$.

2. Check the singularity of $P = v^T v$. If singular, go back to Step 1.

3. Generate a random mean vector $m \in \mathbb{R}^{2m+2}$.

4. Find $\hat{y}$, Σ_y and $\mathbf{\Sigma}_{xy}$ using MCT & exact calculation using the above mean and covariance matrices.

$$P = \begin{bmatrix} -0.0815 & -0.0082 & -0.0628 & -0.0360 \\ -0.1507 & 0.0790 & 0.0786 & -0.0224 \\ -0.1022 & -0.0330 & -0.0141 & 0.1503 \\ -0.0370 & 0.0125 & 0.0044 & 0.0684 \end{bmatrix} \text{ and } m = \begin{bmatrix} -0.1608 \\ 0.9850 \\ -0.6011 \\ 0.0443 \end{bmatrix}$$

$$\implies \hat{y}_{Ex} = \hat{y}_{MCT} = 0.6514, \Sigma_{y,MCT} = \Sigma_{y,Ex} = 0.03626$$

$$\begin{aligned} P_{xy,Ex} &= 100 * \begin{bmatrix} 3.1360 & 0.1842 & 0.6173 & -1.6878 \end{bmatrix}^T \\ \boldsymbol{P}_{xy,MCT,1} &= 100 * \begin{bmatrix} 3.1359 & 0.1839 & 0.6169 & -1.6879 \end{bmatrix}^T \\ \boldsymbol{P}_{xy,MCT,2} &= 100 * \begin{bmatrix} 3.1356 & 0.1842 & 0.6176 & -1.6875 \end{bmatrix}^T \\ \boldsymbol{P}_{xy,MCT,3} &= 100 * \begin{bmatrix} 3.1361 & 0.1837 & 0.6169 & -1.6874 \end{bmatrix}^T \end{aligned} \tag{3.45}$$

When one looks at the results shown in 3.45, there is very close agreement between MCT and Exact Transformation. Only in the calculation of $\boldsymbol{P}_{xy,MCT}$ approximately 0.2% differences were observed. In order to see whether our formulation was wrong, we have run 2 more MCTs, which showed that MCT gave somewhat inconsistent results at such low resolution. Increasing the number of data points used would have solved this problem but limited computer memory made this impossible. Another test case was run with increased covariance matrix eigenvalues (by approximately a factor of 2), this also shows a close agreement between MCT and Exact Transformation.

$$P = \begin{bmatrix} 0.2495 & -0.0459 & 0.1550 & 0.0249 \\ -0.0459 & 0.0438 & 0.0074 & 0.0731 \\ 0.1550 & 0.0074 & 0.2334 & 0.0978 \\ 0.0249 & 0.0731 & 0.0978 & 0.2322 \end{bmatrix} \text{ and } m = \begin{bmatrix} -0.0639 \\ 0.5022 \\ 1.9087 \\ 0.4122 \end{bmatrix}$$

$$\implies \hat{y}_{Ex} = \hat{y}_{MCT} = -0.2183, \Sigma_{y,MCT} = 0.2013 \cong \Sigma_{y,Ex} = 0.2012$$

$$\begin{aligned} P_{xy,Ex} &= 100 * \begin{bmatrix} 19.8009 & -6.1904 & 5.4920 & -3.7703 \end{bmatrix}^T \\ \boldsymbol{P}_{xy,MCT} &= 100 * \begin{bmatrix} 19.7922 & -6.1873 & 5.4884 & -3.7678 \end{bmatrix}^T \end{aligned} \tag{3.46}$$

We will also compare the results of MCT/Exact Transformation to those of the TT2

and UT so that some simulation result discussions can be referred back to here.

Table 3.4: Comparison of Different Transformations

1. $\boldsymbol{x} \sim \mathcal{N}([0; 1; \mathbf{0}; \pi/2]; 0.1 \cdot \mathrm{diag}[1, 1, \mathbf{1}, 0.01])$		
Transformation	$y \sim \mathcal{N}(.; .)$	$\boldsymbol{P}_{xy}$
TT2	0.950; 0.210	$[0.100, 0.100, 0, 0]^T$
UT	0.952; 0.212	$[0.100, 0.100, 0, 0]^T$
MCT	0.951; 0.196	$[0.100, 0.095, -10^{-5}, -10^{-6}]^T$
Exact	0.951; 0.196	$[0.100, 0.095, 0, 0]^T$
2. $\boldsymbol{x} \sim \mathcal{N}([0; 1; \mathbf{0}; \pi/2]; 0.1 \cdot \mathrm{diag}[1, 1, \mathbf{10}, 0.01])$		
Transformation	$y \sim \mathcal{N}(.; .)$	$\boldsymbol{P}_{xy}$
TT2	0.500; 0.700	$[0.100, 0.100, 0, 0]^T$
UT	0.646; 0.826	$[0.100, 0.100, 0, 0]^T$
MCT	0.606; 0.357	$[0.100, 0.061, 10^{-4}, 10^{-6}]^T$
Exact	0.607; 0.357	$[0.100, 0.061, 0, 0]^T$
3. $\boldsymbol{x} \sim \mathcal{N}([0; 1; \boldsymbol{\pi}/\mathbf{4}; \pi/2]; 0.1 \cdot \mathrm{diag}[1, 1, \mathbf{1}, 0.01])$		
Transformation	$y \sim \mathcal{N}(.; .)$	$\boldsymbol{P}_{xy}$
TT2	0.672; 0.208	$[0.100, 0.071, -0.071, 0]^T$
UT	0.673; 0.200	$[0.100, 0.071, -0.066, 0]^T$
MCT	0.673; 0.198	$[0.100, 0.067, -0.067, 0]^T$
Exact	0.673; 0.198	$[0.100, 0.067, -0.067, 0]^T$
3. $\boldsymbol{x} \sim \mathcal{N}([0; 1; \boldsymbol{\pi}/\mathbf{4}; \pi/2]; 0.1 \cdot \mathrm{diag}[1, 1, \mathbf{10}, 0.01])$		
Transformation	$y \sim \mathcal{N}(.; .)$	$\boldsymbol{P}_{xy}$
TT2	0.354; 0.950	$[0.100, 0.071, -0.071, 0]^T$
UT	0.457; 0.567	$[0.100, 0.071, -0.066, 0]^T$
MCT	0.429; 0.466	$[0.100, 0.043, -0.043, 10^{-8}]^T$
Exact	0.429; 0.466	$[0.100, 0.043, -0.043, 0]^T$

Variables that have changed between each trial are written in bold. Our general observations are as follows.

- UT does a fairly good job in estimating $\hat{y}$, irrespective of the current mean of θ. Its estimation performance gets worse with increased Σ_θ. TT2 works well only when Σ_θ is low.

- Both the TT2 and UT overestimate the output covariance. This would lower the Kalman gain, resulting in slower convergence. Overestimation gets more pronounced when the current angle θ, the third state, is non-zero.

- When elements of $\boldsymbol{x}$ are uncorrelated, correlation between y and a_0 is perfectly estimated by all methods thanks to the linear relationship between y and a_0.

3.3 Kalman Filter Tuning

Kalman filter tuning is the process of finding the process and measurement noise covariances. As manual tuning is often prone to errors and as it is likely that these parameters may change over time, there is a need for adapting the KF parameters to the incoming data. Early work by Mehra [46] introduced and categorized KF tuning methods into 4: covariance matching, correlation, Bayesian and Maximum Likelihood (ML).

Covariance matching technique has a basic idea that covariance obtained from sample innovation sequence has to match its theoretical one. Innovations obtained for the output are sampled over a window and measurement noise variance $\boldsymbol{R}$ is obtained by subtracting the prior output variance from the covariance of this sequence. Process noise covariance is obtained similarly. One may also use a low pass filtering or forgetting factor approach instead of windowing as presented in [39] to increase the major strength of this method, which is its computational speed. However, assuring positive definiteness in this method is its major drawback. Subtraction step used in finding $\boldsymbol{R}$ may result in this. Ad-hoc methods such as forcing $\boldsymbol{R}$ to be diagonal and replacing diagonal entries with their absolute values was suggested to overcome this problem [47], although a more educated fix with smoothed output estimates were later introduced. As it should be apparent from even this basic fact that covariance matching techniques are suboptimal. Also measurement noise estimates are generally more accurate than process noise estimates [47]. Despite such major drawbacks, such techniques are preferred thanks to their simplicity and computational efficiency.

Correlation techniques make use of the fact that the innovation sequence should be zero-mean, white and Gaussian. Sample autocorrelation functions are generated from the innovation sequence and they are used to find the process and measurement noise covariances. A unique solution to $\boldsymbol{R}$ can be obtained with this method for linear systems; however, a unique solution for $\boldsymbol{Q}$ is guaranteed with strict restrictions that the system is fully observable and the number of measurements is greater than the number of states [46]. For nonlinear systems, on the other hand, even the positive definiteness of $\boldsymbol{R}$ is not guaranteed [47], and correlation techniques may not give sound results even for linear time-varying systems while trying to find $\boldsymbol{Q}$. That is

why, similar to the covariance matching techniques they require ad-hoc procedures to assure positive definiteness and are suboptimal.

3.3.1 Maximum Likelihood Tuning with Expectation Maximization

Maximum Likelihood (ML) estimate of $\boldsymbol{\theta}$ is the vector that maximizes the probability of $\boldsymbol{\theta}$ given a set of observed output sequence, $\boldsymbol{Y}_N$, where $\boldsymbol{\theta}$ may include any of process and noise parameters. Note that it is equivalent to maximize the likelihood and log-likelihood as the logarithmic function is monotonic.

$$\hat{\boldsymbol{\theta}}^{ML} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{Y}_N) = \arg\max_{\boldsymbol{\theta}} \underbrace{\log p(\boldsymbol{\theta}|\boldsymbol{Y}_N)}_{L_{\boldsymbol{\theta}}(\boldsymbol{Y}_N)} \tag{3.47}$$

The probability given above can be expanded using Bayes' rule as follows [46]. One can safely ignore the normalization constant given in equation 3.48 while trying to maximize the probability on the left hand side. Furthermore, if no prior information is available on the noise parameters $\boldsymbol{\theta}$, the term $p(\boldsymbol{\theta})$ can be dropped as well. What this implies is that maximum likelihood estimate of $\boldsymbol{\theta}$ coincides with the $\boldsymbol{\theta}$ that maximizes the likelihood of the measurement sequence when no information is available regarding $\boldsymbol{\theta}$.

$$p(\boldsymbol{\theta}|\boldsymbol{Y}_N) = \frac{p(\boldsymbol{Y}_N|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{Y}_N)} \tag{3.48}$$

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{Y}_N) \propto p(\boldsymbol{Y}_N|\boldsymbol{\theta}) &= p(\boldsymbol{y}_N, \boldsymbol{Y}_{N-1}|\boldsymbol{\theta}) \\ &= p(\boldsymbol{y}_N|\boldsymbol{Y}_{N-1}, \boldsymbol{\theta})p(\boldsymbol{Y}_{N-1}|\boldsymbol{\theta}) \\ &\dots \\ &= p(\boldsymbol{y}_N|\boldsymbol{Y}_{N-1}, \boldsymbol{\theta})p(\boldsymbol{y}_{N-1}|\boldsymbol{Y}_{N-2}, \boldsymbol{\theta}) \dots p(\boldsymbol{y}_0) \end{aligned} \tag{3.49}$$

Finding probabilities given in equation 3.49 are not directly obtainable from KF equations. Moreover, optimization procedure after this may be problematic if the probabilities are not obtained parametrically. That is why, we will be using a standard approach of Expectation Maximization (EM) algorithm to find the ML estimate of $\boldsymbol{\theta}$. It is an iterative procedure that finds a maximum for maximum likelihood esti-

mation and this maximum does not necessarily correspond to the global one. As a common approach in EM, we have to introduce some latent variables. The natural choice of unobserved variables in a KF are unobserved states as was done in [48]. Letting $\boldsymbol{\theta} = \boldsymbol{Q}$, EM algorithm is then given as follows.

1. Set an initial value for $\boldsymbol{Q}_0$ and $l = 1$.
2. Expectation step: Calculate the expectation

$$\Gamma(\boldsymbol{Q}; \boldsymbol{Q}_{l-1}) = \mathbb{E}\left\{\log p(\boldsymbol{Y}_N, \boldsymbol{X}_N|\boldsymbol{Q})|\boldsymbol{Q}_{l-1}, \boldsymbol{Y}_N\right\}$$

3. Maximization Step: Maximize the next iterated value of $\boldsymbol{Q}$ using

$$\boldsymbol{Q}_l = \arg\max_{\boldsymbol{Q}} \Gamma(\boldsymbol{Q}; \boldsymbol{Q}_{l-1})$$

4. Iterate until convergence: Stop if $||\boldsymbol{Q}_l - \boldsymbol{Q}_{l-1}|| < \epsilon$; otherwise go back to step 2 with $l = l + 1$.

Citing from the same source, maximization step has the following structure. What it means is that smoothed expectations of $\mathbb{E}\left\{\boldsymbol{w}_i\boldsymbol{w}_i^T|\boldsymbol{Q}_{l-1}, \boldsymbol{Y}_N\right\}$ can be used to calculate the next iterated value and this expectation is given as in equation 3.51.

$$\boldsymbol{Q}_l = \frac{1}{N}\sum_{i=1}^{N-1} \mathbb{E}\left\{\boldsymbol{w}_i\boldsymbol{w}_i^T|\boldsymbol{Q}_{l-1}, \boldsymbol{Y}_N\right\} \tag{3.50}$$

$$= \boldsymbol{Q}_{l-1} + \frac{1}{N}\boldsymbol{Q}_{l-1}\left(\sum_{i=1}^{N-1} \boldsymbol{P}_{i+1|i}^{-1}\left(\boldsymbol{P}_{i+1|N} - \boldsymbol{P}_{i+1|i} + \boldsymbol{a}\boldsymbol{a}^T\right)\boldsymbol{P}_{i+1|i}^{-1}\right)\boldsymbol{Q}_{l-1} \tag{3.51}$$

where

$$\boldsymbol{a} \triangleq (\hat{\boldsymbol{x}}_{i+1|N} - \hat{\boldsymbol{x}}_{i+1|i}) \tag{3.52}$$

There is one little problem with the above form of the EM algorithm. It needs to find $(2m + 2) \times (2m + 2)$ number of parameters for a model order m in order to find the correct process covariance matrix. That is, even for a second order system, 36 parameters are needed to be found. Our initial experimentation with this produced

extremely slow convergence and inaccurate results under simulation with diagonal noise entries. In order to reduce the number of parameters to be found, we decided to apply a projection step after the maximization step in order to guide the search into an a-priori form of covariance matrix as a set of feasible solutions. This idea is inspired by Projected Gradient Descent method used in optimization problems [49]. The projection will be onto a block diagonal matrix as shown in equation 3.53: amplitude coefficient covariance portion is assumed to be diagonal to speed up the convergence. Angular velocity and position block will be of the form given in equation 3.54. The final goal is to minimize the Frobenius norm given in equation 3.55. With the diagonality assumption of $\boldsymbol{Q}_{ab}$, it is clear that only keeping the diagonal entries of $\boldsymbol{Q}_{l,(1:2m)(1:2m)}$ minimizes this norm error and it is given as in equation 3.56.

$$\boldsymbol{Q}_{l,BD} = \begin{bmatrix} \boldsymbol{Q}_{ab}T_s & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_{\omega} \end{bmatrix} \tag{3.53}$$

$$\boldsymbol{Q}_{\omega} = \begin{bmatrix} q_w T_s^3/3 & q_w T_s^2/2 \\ q_w T_s^2/2 & q_w T_s \end{bmatrix} \tag{3.54}$$

$$||\boldsymbol{Q}_l - \boldsymbol{Q}_{l,BD}||_2 = \sum_{i,j}^{2m+2} |q_{ij} - q_{ij,BD}|^2 \tag{3.55}$$

$$\boldsymbol{Q}_{ab}T_s = \mathrm{diag}(\boldsymbol{Q}_{l,(1:2m)(1:2m)}) \tag{3.56}$$

Finding q_w is a bit more challenging. Minimization of norm error given in equation 3.55 is equivalent to the minimization given in equation 3.57 as far as the parameters of $\boldsymbol{Q}_w$ are concerned, where $\mathrm{vec}(\cdot)$ is the vectorization. This is equivalent to find the minimum norm error for the linear system of equations given in equation 3.59. Since the matrix $\boldsymbol{A}$ is full column rank, it left inverse exists and is given in equation 3.60, where the second manipulation can be made as $\boldsymbol{A}^T\boldsymbol{A}$ is a scalar. Finally, the explicit solution for q_w is given as in equation 3.61.

$$||q_w \begin{bmatrix} T_s^3/3 \\ T_s^2/2 \\ T_s^2/2 \\ T_s \end{bmatrix} - \mathrm{vec}(\hat{\boldsymbol{Q}}_w)||_2 \text{ where } \hat{\boldsymbol{Q}}_w = \boldsymbol{Q}_{l,(2m+1:2m+2)(2m+1:2m+2)} \tag{3.57}$$

$$\text{Let } \boldsymbol{A} \triangleq \begin{bmatrix} T_s^3/3 \\ T_s^2/2 \\ T_s^2/2 \\ T_s \end{bmatrix}. \tag{3.58}$$

$$\boldsymbol{A} q_w = \mathrm{vec}(\hat{\boldsymbol{Q}}_w) \tag{3.59}$$

$$\boldsymbol{A}^+ = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T = \frac{\boldsymbol{A}^T}{\boldsymbol{A}^T \boldsymbol{A}} \tag{3.60}$$

$$q_w = \frac{\boldsymbol{A}^T \mathrm{vec}(\hat{\boldsymbol{Q}}_w)}{\boldsymbol{A}^T \boldsymbol{A}} \tag{3.61}$$

These two projections given in equations 3.56 and 3.61 can be inserted after the third step of EM algorithm to give projected EM algorithm.

As a final remark for the above projection is that we do not suggest that the projected covariance estimates are the optimum ones. However, this approach allows faster convergence when the DT covariances are separated by an order of magnitude. In our early simulations, it was impossible for Σ_θ to approach its true value $q_w T_s^3/3$ which is an order of magnitude lower than Σ_ω as the sampling period is $T_s = \frac{1}{20}$. Decreasing ϵ may have solved this problem with a great increase in computational resources. It was thanks to this modification that Σ_θ could be guided to its true value faster, that is with less number of iterations.

Pseudocode for the combined KF and EM algorithm can be seen in Algorithm 1. Note that until the number of available data points reach the number required by the EM algorithm, regular filtering equations are applied. After that limit is exceeded, EM is run at least once with the arrival of a new data point and it may be run more than once if the convergence criterion of EM is not met. As a results, process noise covariance

estimation is performed online.

Algorithm 1: Online EM & KF

Initialization:;
$\boldsymbol{Q}_{=}\boldsymbol{Q}_{\text{manual}}$;
Generate two FIFO buffers of length N for state prediction and filtered state estimate vectors;
Do the same for state prediction covariance and filtered state estimation covariance matrices;
Iterations:;
Do a measurement update as in Fig. 3.5;
Push the filtered state estimates and their covariances into their FIFO buffers;
if *k>Number of data points N* **then**
    Set $l = 0$ and $\boldsymbol{Q}_0 = \boldsymbol{Q}$;
    repeat
        Update the process noise according equation 3.51;
    until $||\boldsymbol{Q}_l - \boldsymbol{Q}_{l-1}|| < \epsilon$;
    $\boldsymbol{Q} = \boldsymbol{Q}_l$;
end
Increment k;
Do a time update as in Fig. 3.5;
Push the state predictions and their covariances into their FIFO buffers;

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, we will describe the results obtained from both simulation and our own setup. We discuss the data generation process used in our simulations with justifications based on respiratory motion data obtained from a patient in Section 4.1. A comparative discussion in terms of convergence speeds and output RMSE between the Kalman filters is presented in Section 4.2. Section 4.3.1 introduces sensor used for respiratory motion tracking as well as the sensor placements. Section 4.3.2 uses respiratory records obtained from patients to compare the filtering performance of 3 nonlinear Kalman filters to justify the benefits of ExKF over the others.

4.1 Simulation Environment

In this section, we will describe how we have determined the data generation process used in our simulations. Figure 4.1 shows the Fourier transform of the chest movement data obtained from an artificially ventilated patient. Ventilation frequency is 16 breaths per minute, justifying the peaks around $\frac{16}{60} = 0.667$Hz and at higher harmonics. There is no DC offset in the data as PCA removes this offset. Also, frequency components of respiratory motion do not exhibit sharp peaks around the harmonics but are somewhat spread around them, indicating the quasi-periodic nature of the movement.

A more quantitative analysis of this signal yields that $a_1 \approx 0.496$ and $b_1 \approx 0.602$. Our model assumes that the first order harmonic is to be tracked with a single coefficient: that is, information in b_1 can be captured by θ and a_1 as we have previously discussed.

This results in $a_1 \leftarrow \sqrt{a_1^2 + b_1^2} \approx 0.780$. With this in mind, much of the information is contained in the first harmonic, with an approximate decay ratio of $\frac{a_2}{a_1} = 1/4$. This also holds for the higher order harmonics ($\frac{a_{n+1}}{a_n}$). As we have analyzed a single respiratory motion record, we will assume and force this limit as the twice of $1/4$ during simulations.

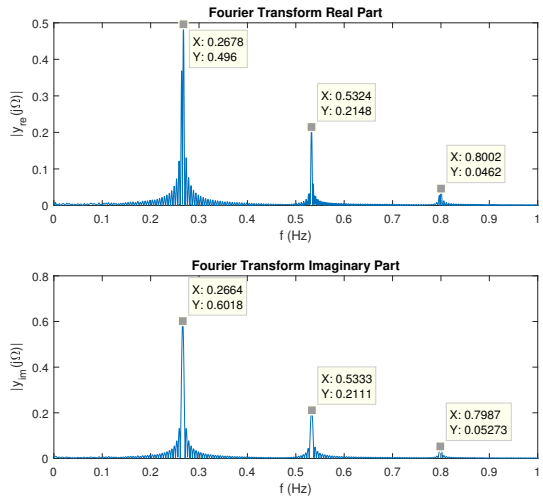

Figure 4.1: Fourier Transform of respiratory motion trace

After examining these, we will follow a similar approach to what was done in [50] to simulate the system: generate an initial state as in Eq. (4.1), and perturb each state with noise. Different from their approach, we will keep the angular velocity noise Σ_ω relatively lower than those of the Fourier series coefficients in order to keep angular velocity changes almost zero so as to reflect a consistently periodic nature of mechanical ventilation. As was discussed in [15], the second model based method described in Section 2.6.1, approximation orders with $m = 2$ and $m = 3$ are recommended; otherwise, higher order coefficients are likely to adapt to noise as their model does not take measurement noise statistics into account. Also considering the case given in Fig. 4.1, higher order coefficients contribute little to the explanation of the output. With a decay ratio of $1/4$, power associated with third and higher order coefficients corresponds to approximately 6.25% of the total power of the signal. We will use their recommendation only when we artificially generate data. We will try to track the real respiratory motion trace with different orders of Fourier series

approximations in order to find the best underlying model in Chapter 5.

$$\boldsymbol{x}_0 = \begin{bmatrix} 0 & 1 & 0.5 & 0.25 & \dots & 0.5 & 0.25 & \dots & \theta_0 & \omega \end{bmatrix}^T \tag{4.1}$$

Finally, it would be impossible to differentiate the systems given in Eqs. (4.2)-(4.3) by only looking at the output y since both measurements are equivalent. That is, it is impossible to both track θ by a factor of π and the sign of odd order coefficients. Therefore, an error of π is equivalent to an error of 0 if the true sign and estimated sign of the coefficients tracked are different. Similarly, error in the coefficient of interest would be equal to $e_i = a_i - (-\hat{a}_i) = a_i + \hat{a}_i$ in the same case. Alternatively, one has two options: either track only the amplitude of odd order coefficients, letting θ oscillate between $-\pi$ to π, or restrict θ to be bounded in 0 to π, making signs of coefficients important. We opted to track the amplitude of a_1 and reflected its sign to θ whenever it became negative.

$$y = a_0 + a_1 \cos(\theta_k) + \sum_{i=2}^{m} \Big(a_i \cos(i\theta_k) + b_i \sin(i\theta_k) \Big) \tag{4.2}$$

$$\begin{aligned} y = {} & a_0 - a_1 \cos(\theta_k + \pi) + \sum_{i \text{ even}} \Big(a_i \cos(i\theta_k + i\pi) + b_i \sin(i\theta_k + i\pi) \Big) \\ & + \sum_{i \text{ odd}} \Big(a_i \cos(i\theta_k + i\pi) + b_i \sin(i\theta_k + i\pi) \Big) \end{aligned} \tag{4.3}$$

4.2 Simulation Results and Discussion

We will conduct a comparative performance test of ExKF on our system against UKF, EKF1 and EKF2 In order to test the usefulness of ExKF, we will test it against the UKF and EKF2 under different conditions with artificially generated data for a second order Fourier series approximation with $m = 2$ with a reasoning discussed in the previous section. Also note that increasing the value of m deteriorates convergence properties of all filters, with UKF and EKF2 being unable to converge most of the time as shown in Section 4.3.2. That is why the choice of $m = 2$ yields more comparative results. For reproducibility of the results, we use the following variables common on

our tests.

- Sampling frequency is chosen as 20Hz as it is the sampling frequency of the device we will be using for data collection,
- Fundamental period or "breathing period" is chosen as 5 seconds,
- Number of data points, or epochs, is 1000 as convergence results can be observed with them, and
- Number of Monte Carlo simulations is $N = 1000$.

- Case 1:

$$\begin{aligned}
\boldsymbol{Q}_{ab} &= \mathrm{diag}(0,0,0,0), q_\omega = 0 \\
R &= 0.01^2, \boldsymbol{x}_0 = \begin{bmatrix} 0 & 1 & 0.5 & 0.5 & \pi/2 & 2*\pi/5 \end{bmatrix}^T \\
\hat{\boldsymbol{x}}_0 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2*\pi/5 \end{bmatrix}^T, \boldsymbol{P}_0 = \mathrm{diag}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^2
\end{aligned}$$

The first test case is devoted to examine the convergence properties of these filters. To achieve this, we aim to keep the states constant; consequently, process noise variances are intentionally set to zero so that no variations occur in $\boldsymbol{x}$. In order to make the measurement covariance calculation dominated by the nonlinearity, measurement noise variance is chosen small: that is, we have chosen a value for R as 0.01^2 so that Σ_y value of 0.263 increases to the value of $\Sigma_{y,measured} = 0.2631$ for the above set of parameters, with less than 1% contribution of measurement noise. By doing so, output covariance is mostly affected by the nonlinearity instead of the linear additive part.

Initial uncertainty is chosen as 1 standard deviation away from each state so that errors due to overestimating or underestimating initial estimate covariance are avoided. Initial state estimates are all chosen as 0 except for ω to reflect the uncertainty in the coefficients and θ, both of which would be unknown if we were to track a quasi-periodic signal as in our results with real data. That is to say that coefficients to be tracked as well as at the angular position would be unknown a-priori at the nitialization of tracking. When we examined the MCT verification of ExKF in section 3.2.3.1,

we have observed that ExKF had the highest Kalman gain when the uncertainty associated with the current angle θ was high (as it is the case here with $\Sigma_{\theta_0} = \pi^2/4$). That is why our expectation is that ExKF should have the fastest convergence among all filters.

RMSE state errors can be seen in Figure 4.2. Note that UKF cannot handle such large initialization errors; it cannot track either one of the states θ or a_1 as can be justified with residual errors even after $t > 40$. The same goes for EKF1. ExKF and EKF2 both converged to the true model, with ExKF having smaller errors initially; this is true especially for θ. This seems to suggest that ExKF is more robust to initial parameter estimation errors than both of the filters, with UKF having the worst performance. Before evaluating this suggestion more thoroughly, we wish to compare the performances in output prediction as presented in Table 4.1. UKF has the worst RMSE as 0.652 while EKF2 provides 63% improvement over it. ExKF, on the other hand, has an approximate improvement as 89%, making it the best filter for this test case. EKF1, on the other hand, has clear lags in tracking both θ and a_1 despite its reduced output RMSE of 0.144.

Table 4.1: Simulation Results - Output RMSE for Different Test Cases

Case #	**RMSE$_{\text{EKF1}}$**	**RMSE$_{\text{EKF2}}$**	**RMSE$_{\text{UKF}}$**	**RMSE$_{\text{ExKF}}$**
1	0.144	0.239	0.652	0.072
2	0.350	0.829	0.841	0.145
3	0.033	0.030	0.038	0.031
4	0.091	0.429	0.814	0.104
5	0.150	0.514	0.870	0.115
6	0.108	$3.43 \cdot 10^6$	0.665	0.899

In order to check whether the ExKF has the fastest convergence among the three filters, the other two being EKF2 & UKF, we decided to generate more sub-cases similar to case #1. Initial estimate of θ is chosen from the discrete group of $\begin{bmatrix} 0 & \frac{\pi}{4} & \frac{\pi}{2} & \frac{3\pi}{4} & \pi \end{bmatrix}$ as its initial estimation error affects the convergence properties of the filters the most. Fourier series coefficients on the other hand were drawn randomly from

$$\begin{bmatrix} 0 & 1 & 0.5 & 0.5 \end{bmatrix}^T \cdot a$$

where a has a zero mean normal distribution with standard deviation α. Now, one

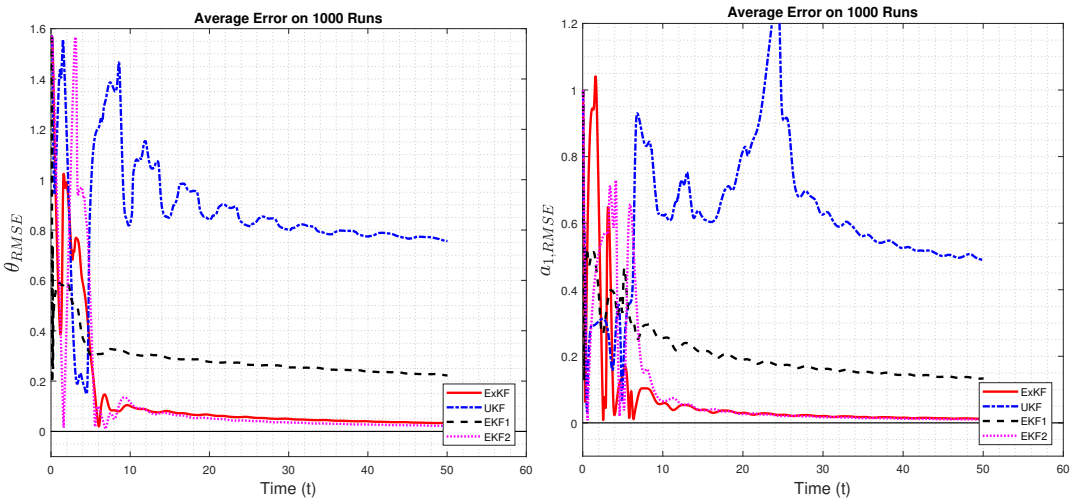

Figure 4.2: RMSE of θ (left) and a_1 (right) in test case # 1

of the problems faced with this approach is when the ratios of $|a_1/a_2|$ and $|a_1/b_2|$ were larger than 0.5 individually. Similarly, having the initial estimate in a_1 too large for filters to handle resulted in divergences. That is why the random drawing procedure was repeated for ratios exceeding the threshold of 0.5 or for values of $|a_1|$ exceeding 3. These values are determined experimentally: when they were exceeded filter convergences were achieved inconsistently. This is just a region where our filter outperforms the other two, outside of which the filtering behaviour was intractably unexplainable with frequent output and state divergences. RMSE vs. time plots of these subcases are provided subsequently on pages 68-72 for the sake of clarity.

First comparison is done for $\theta_0 = 0$ with α values from $0.1, 0.5, 1$ and 1.5. As can be seen in Fig. 4.3 on page 68 convergence speeds of all of the filters are equivalent no matter what the value of α is. This means that all of the filters converge to the true model without any performance differences provided that the current angle θ is estimated correctly.

Second comparison is done for $\theta_0 = \pi/4$. The RMSE values in estimation of a_1 can be seen in Fig. 4.4 on page 69. All of the filters are able to converge to the true state if $\alpha = 0.1$ with approximately the same speed. However, for increased values of α, UKF and EKF2 have deteriorating performances. UKF cannot cope well with large initialization errors when the error in $\hat{\theta}$ is large as can be seen for $\alpha = 1$. None of the filters are able to converge to true a_1 for $\alpha = 1.5$ with a residual RMSE of

approximately 0.2.

Third subcase is devoted to $\theta_0 = \pi/2$ and RMSE plots of a_1 can be seen in Fig. 4.5 on page 70. UKF cannot converge to true a_1 no matter what the initial error is. EKF2 does a fairly good job, with a residual state RMSE greater than 0.1 only apparent when $\alpha = 1$ or $\alpha = 1.5$. ExKF is clearly the best filter especially under this subcase.

Fourth subcase is for $\theta_0 = 3\pi/4$ and is presented in Fig. 4.6 on page 71. Just like the case in $\theta_0 = \pi/4$, EKF2 always has residual errors. UKF can only converge for $\alpha = 0.1$ while ExKF showed no convergence problems in these trials except for the extremely slow convergence when $\alpha = 0.1$.

Fifth subcase is for $\theta_0 = \pi$. Now, we expected and obtained identical results to that of the $\theta_0 = 0$. If one accepts the fact that sign of a_1 in $a_1 \cos\theta$ as well as a phase difference of π in θ is indistinguishable by observing the output, only the errors in the amplitude of a_1 should matter. To clarify this point, if $\hat{a}_1$ converges to true $a_1 = 1$ in the first subcase, $\hat{a}_1$ may converge to -1 in the fifth one. Provided that the phase difference due to sign differences are reflected to $\hat{\theta}$, both subcases should provide identical results. Identical results for $\theta_0 = 0$ and $\theta_0 = \pi$ can be seen by comparing Figures 4.3 (on p. 68) and 4.7 (on p. 72). In my personal trials, we found that similar cases occur for higher values of θ_0; that is why we took the liberty to terminate the tests at π without falling out of generality.

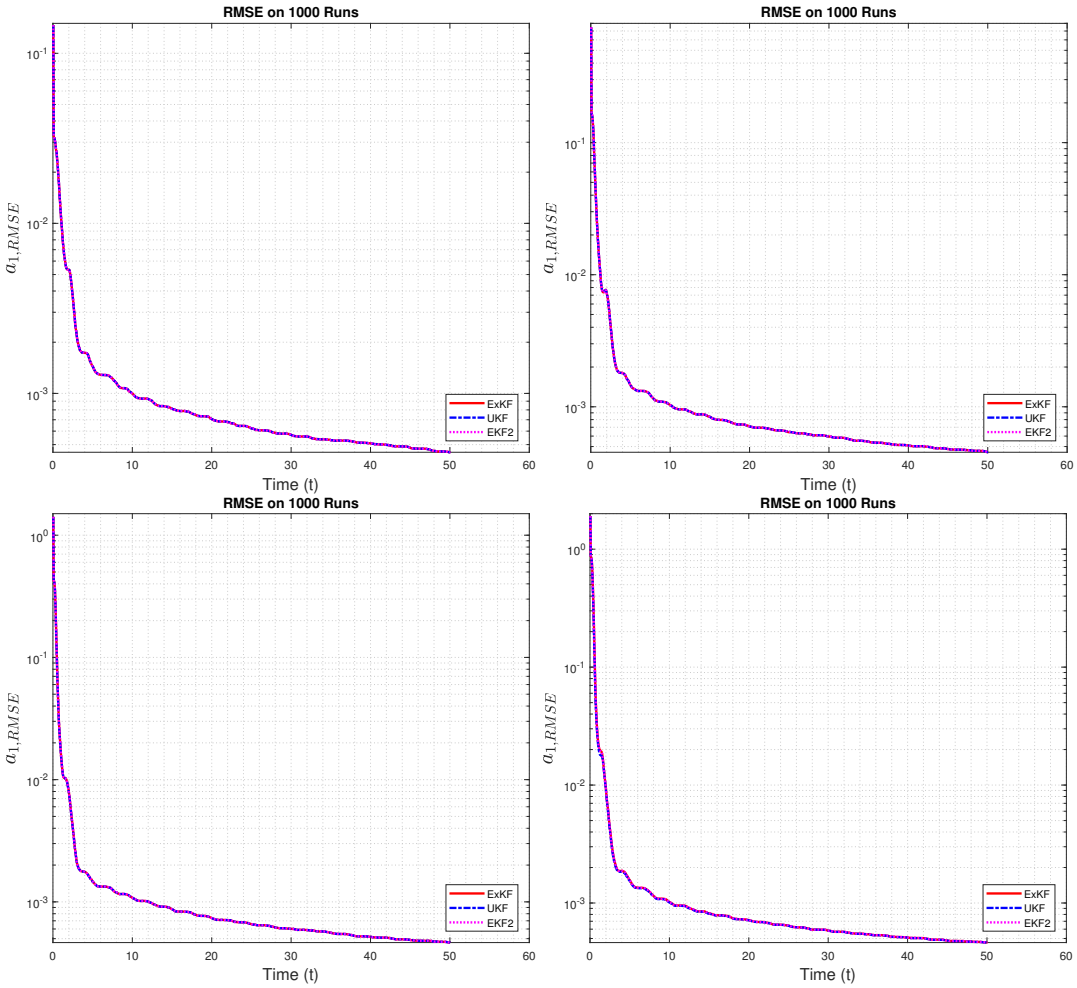

Figure 4.3: RMSE of a_1 for $\alpha = 0.1$ (top left), $\alpha = 0.5$ (top right), $\alpha = 1$ (bottom left) and $\alpha = 1.5$ (bottom right) for $\theta_0 = 0$

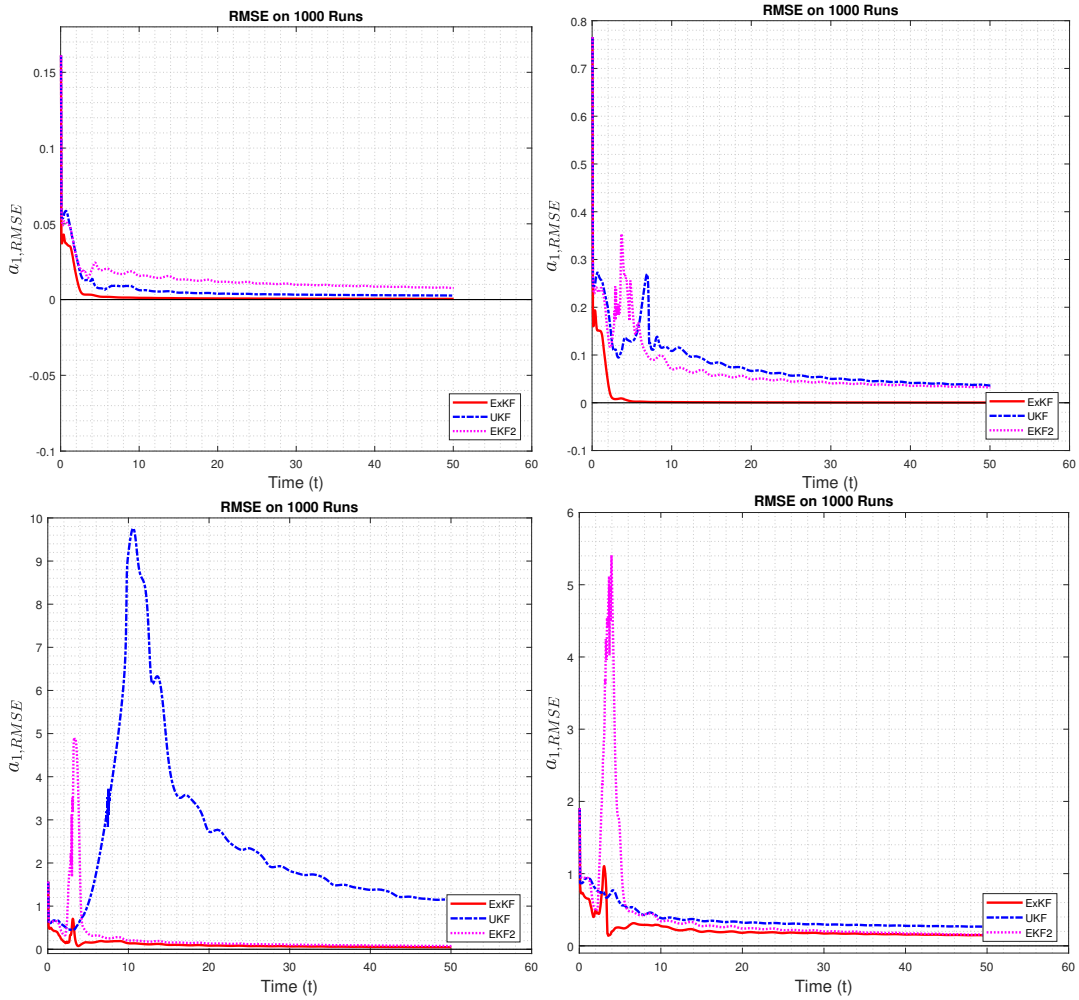

Figure 4.4: RMSE of a_1 for $\alpha = 0.1$ (top left), $\alpha = 0.5$ (top right), $\alpha = 1$ (bottom left) and $\alpha = 1.5$ (bottom right) for $\theta_0 = \pi/4$

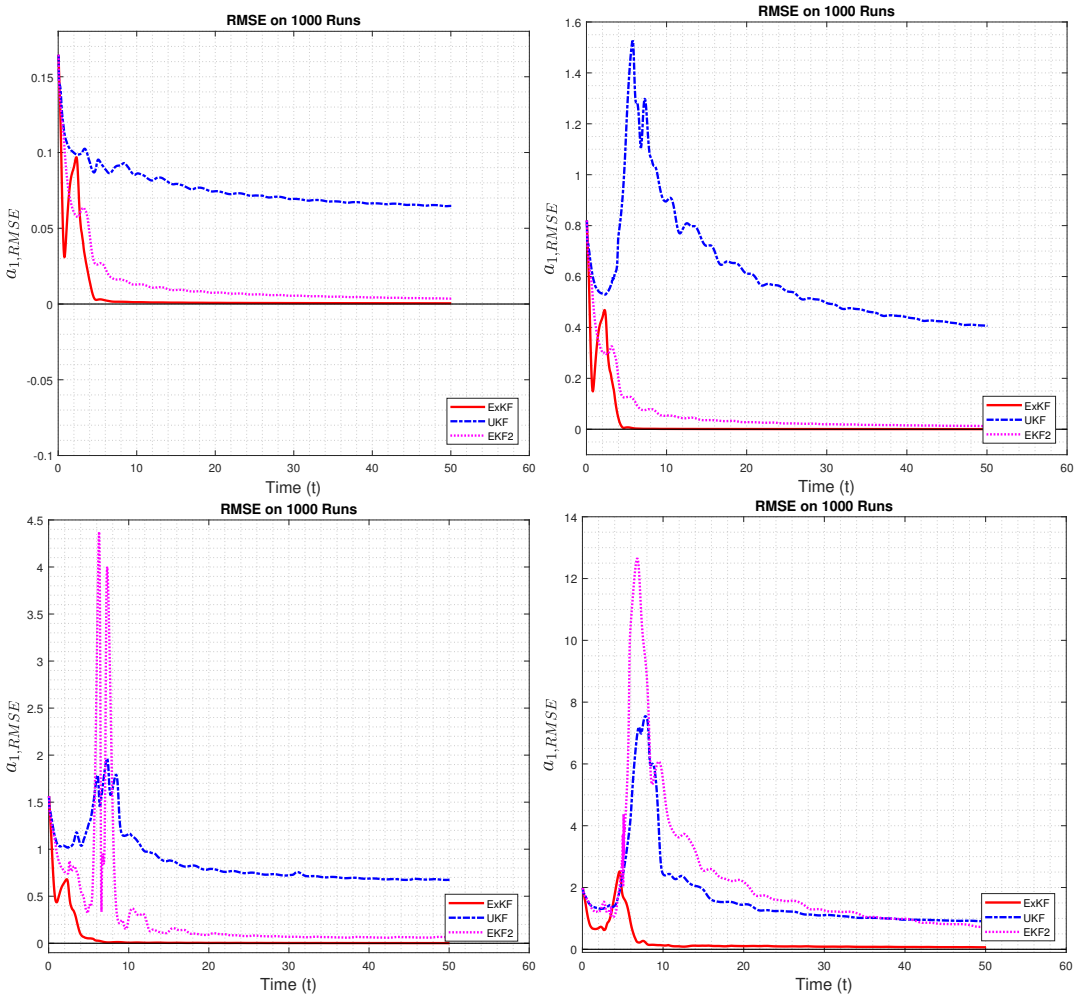

Figure 4.5: RMSE of a_1 for $\alpha = 0.1$ (top left), $\alpha = 0.5$ (top right), $\alpha = 1$ (bottom left) and $\alpha = 1.5$ (bottom right) for $\theta_0 = \pi/2$

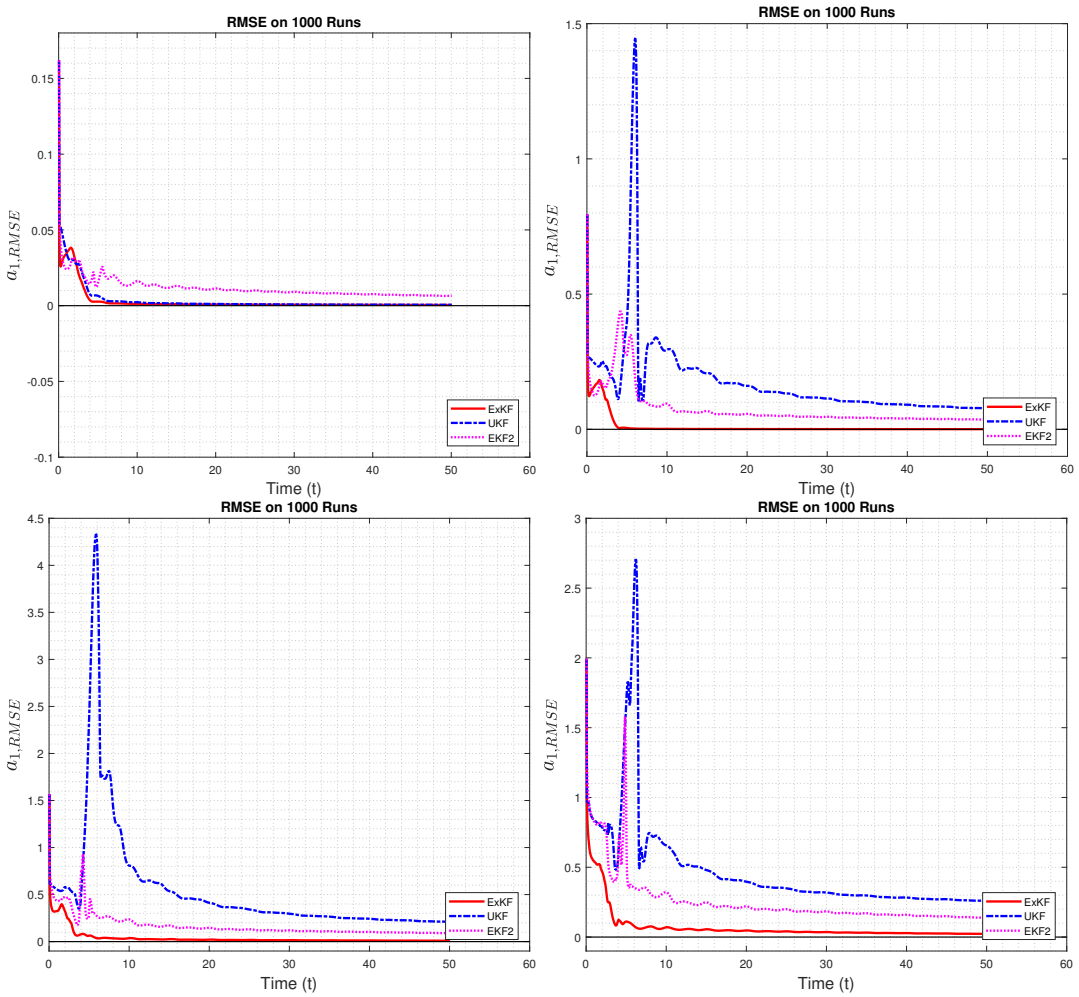

Figure 4.6: RMSE of a_1 for $\alpha = 0.1$ (top left), $\alpha = 0.5$ (top right), $\alpha = 1$ (bottom left) and $\alpha = 1.5$ (bottom right) for $\theta_0 = 3\pi/4$

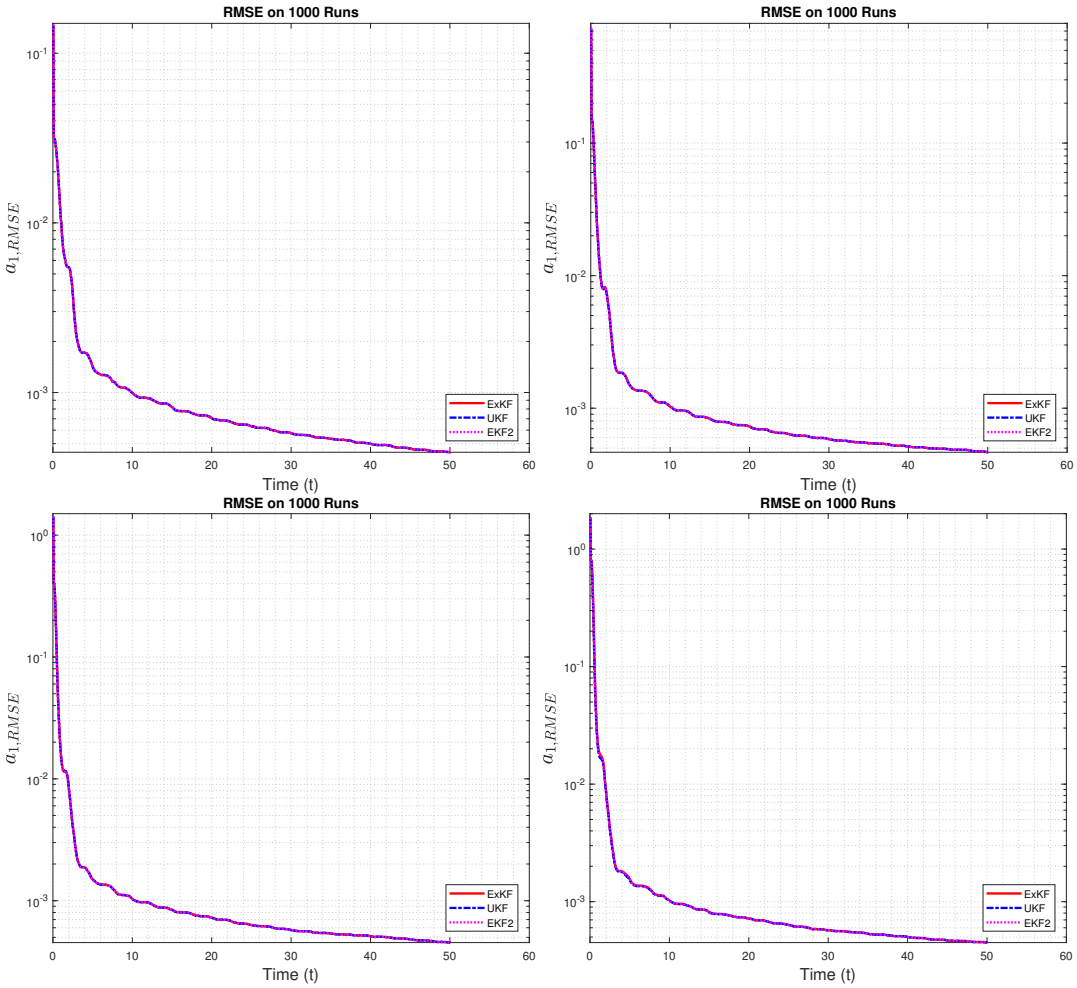

Figure 4.7: RMSE of a_1 for $\alpha = 0.1$ (top left), $\alpha = 0.5$ (top right), $\alpha = 1$ (bottom left) and $\alpha = 1.5$ (bottom right) for $\theta_0 = \pi$

- Case 2: Case 1 with $R = 0.1^2$

Going back to the main cases, we intend to test the filters with an increased measurement noise. When the measurement noise variance is increased from 0.01^2 to 0.1^2, we expect all the filters to perform worse than the first case. RMSE results of state estimates can be seen on Fig. 4.8. Just like it was in the first test case, neither UKF nor EKF1 can track the states. EKF2, in contrast, loses its ability to track both of the states. These can be justified by the over-estimation of the output variance by both filters; with increased measurement noise, Kalman gain remains so low that states cannot be tracked accurately. ExKF is still able to track a_1 but has a clear lag in tracking θ. Their performances on output RMSE-wise converged to each other. Considering the output, UKF demonstrates a value of 0.841, and that of the EKF2 demonstrates a similar performance with a slight amelioration of 2% in the output RMSE value. ExKF and EKF1 have approximate improvements in output RMSE of 83% and 58% over that of UKF, respectively, despite the inability of EKF1 to track the true states. Percentage improvements over UKF decreases when compared to case 1 and all filters have deteriorating performances with increased R.

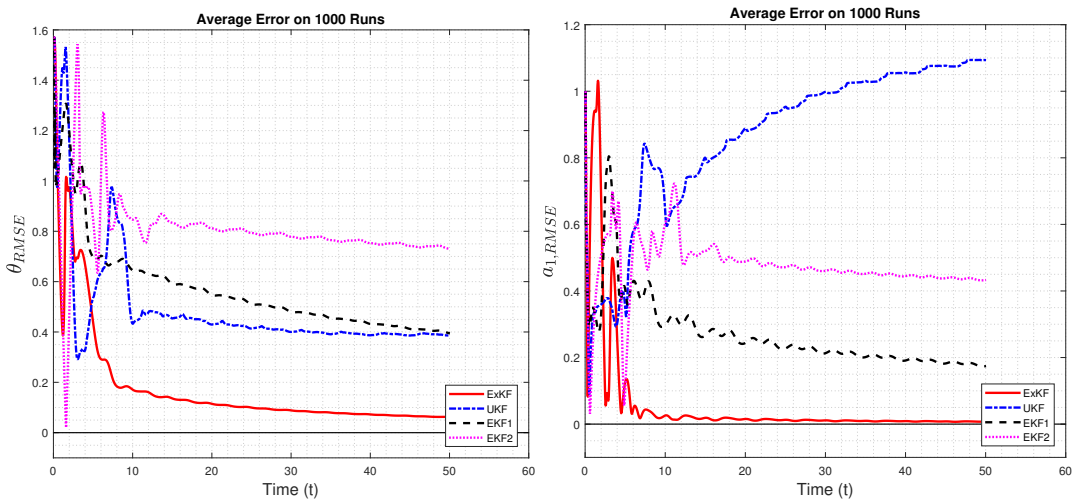

Figure 4.8: RMSE of θ (left) and a_1 (right) in test case # 2

- Case 3: Case 1 with $\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 0.75 * \boldsymbol{x}_{0,1-5} & \boldsymbol{x}_{0,6} \end{bmatrix}^T$.

In Case 3, our aim is to examine filter performances with a fair initial estimate all states but one. The exception is the angular velocity state ω_0 which is fairly well

known for a mechanically ventilated patient. State RMSE of a_1 and θ can be seen in Fig. 4.9. Although no significant error differences can be observed under this case, UKF still has the largest initial errors in both of the states. ExKF and EKF2 have indistinguishable performances in tracking a_1 but EKF2 has an initial advantage over ExKF for tracking θ, between $t = 0$ to $t = 3$ seconds. EKF1 can track the amplitude coefficient almost with the same convergence speed but it has a small lag in tracking θ. In accordance with these results, output RMSE for UKF, ExKF, EKF1 and EKF2 are 0.038, 0.031, 0.033 and 0.030, respectively.

Our conclusion from these three test cases as well as the subcases of case 1 is that ExKF is better than or on par with the UKF/EKF1/EKF2 in terms of convergence. Faster convergence allows ExKF to have less RMSE in tracking respiratory motion as a quasi-periodic signal.

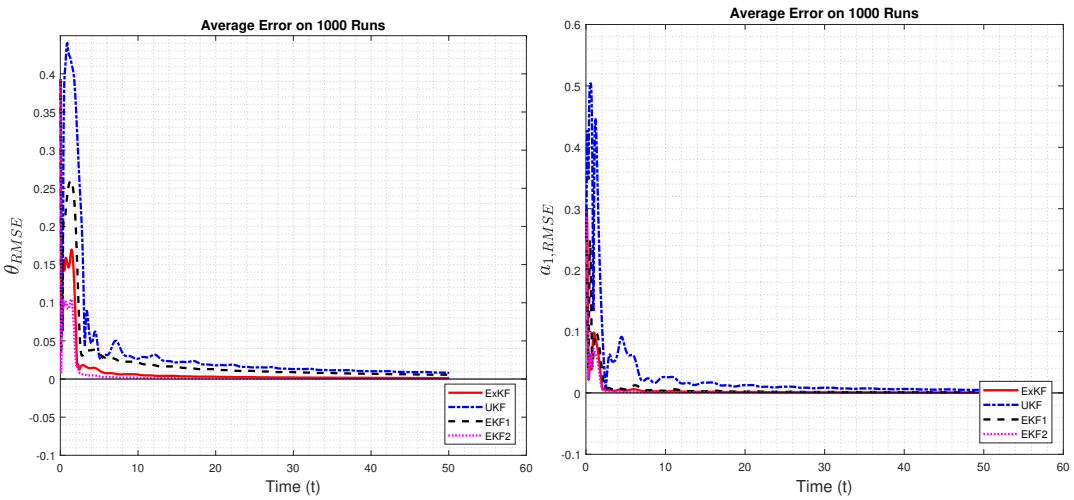

Figure 4.9: θ (left) and a_1 (right) tracking error in test case # 3

- Case 4: Case 1 with $q_\omega = 0.001^2$ and $\boldsymbol{Q}_{ab} = 0.005^2\,\mathrm{diag}(0.1, 1, 1, 1)$.

Having examined the convergence properties of the filters with constant states, we will be performing simulations with added process noise in Case 4, which constitutes a more realistic scenario. In this test case, coefficient process noise variance is chosen slightly larger than the angular velocity noise variance. Their variances are swapped at a later case, Case 5. Moreover, there is no clue on the values of Fourier series coefficients when one starts tracking a respiratory motion model, that is why their initial estimates are set to zero in this test case. Finally, θ is a variable that oscillates

between $(-\pi, \pi]$ and filtering of respiratory motion may start at any instant. That is why, the initial error $\theta_0 - \hat{\theta}_0$ is given a value of $\pi/2$. This value will be changed in several subcases as it has a great impact on the convergence abilities of filters.

RMSE in a_1 and θ can be seen in Fig. 4.10. UKF cannot converge to true a_1 or θ. EKF1 cannot converge to true a_1 and tracking θ shows oscillatory behaviour. To compare the remaining two, EKF2 has larger overshoots in θ and ExKF has larger overshoots in a_1; however, ExKF converges to the true values much faster than EKF2. These are all reflected to the output RMSEs as well: UKF has an RMSE of 0.814, EKF2 of 0.429 and ExKF of 0.104. A slight increase in these values when compared to case 1 can be attributed to the added process noise. The only filter that benefits from increased process noise is EKF1, which has an output RMSE of 0.091. However, its inability to stably track θ or a_1 is a clear indication of convergence to a wrong model.

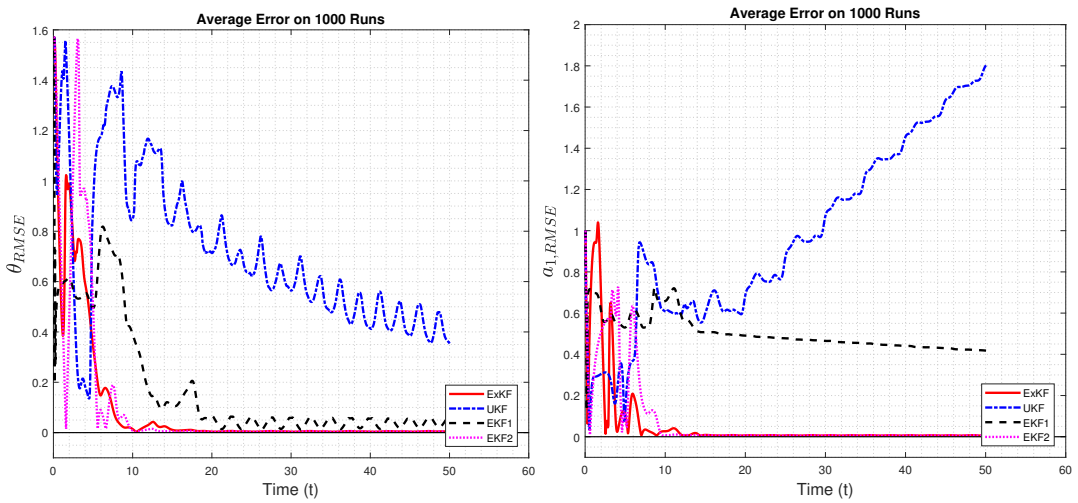

Figure 4.10: RMSE of θ (left) and a_1 (right) in test case # 4

A closer evaluation of convergence to a_1 for different values of $\hat{\theta}_0 - \theta_0$ can be seen in Fig. 4.11. Note again that all filters are again equivalent for a zero initial angle error. UKF is able to track a_1 for $\hat{\theta}_0 - \theta_0 = 0$ or $\pi/4$; larger initial angle errors causes UKF to diverge. EKF2 achieves an RMSE of 0.01 in almost all cases in less than $t = 15$ seconds, only having a lousy performance for $\hat{\theta}_0 - \theta_0 = 3\pi/4$. ExKF achieves this convergence performance in all cases, usually outperforming EKF2 in most of the time. An initial error of $\hat{\theta}_0 - \theta_0 = \pi/4$ seems to favor EKF2 over the others; however, calculation of exact mean and variance seems to favor ExKF more especially when the initial errors associated with $\hat{\theta}$ are high.

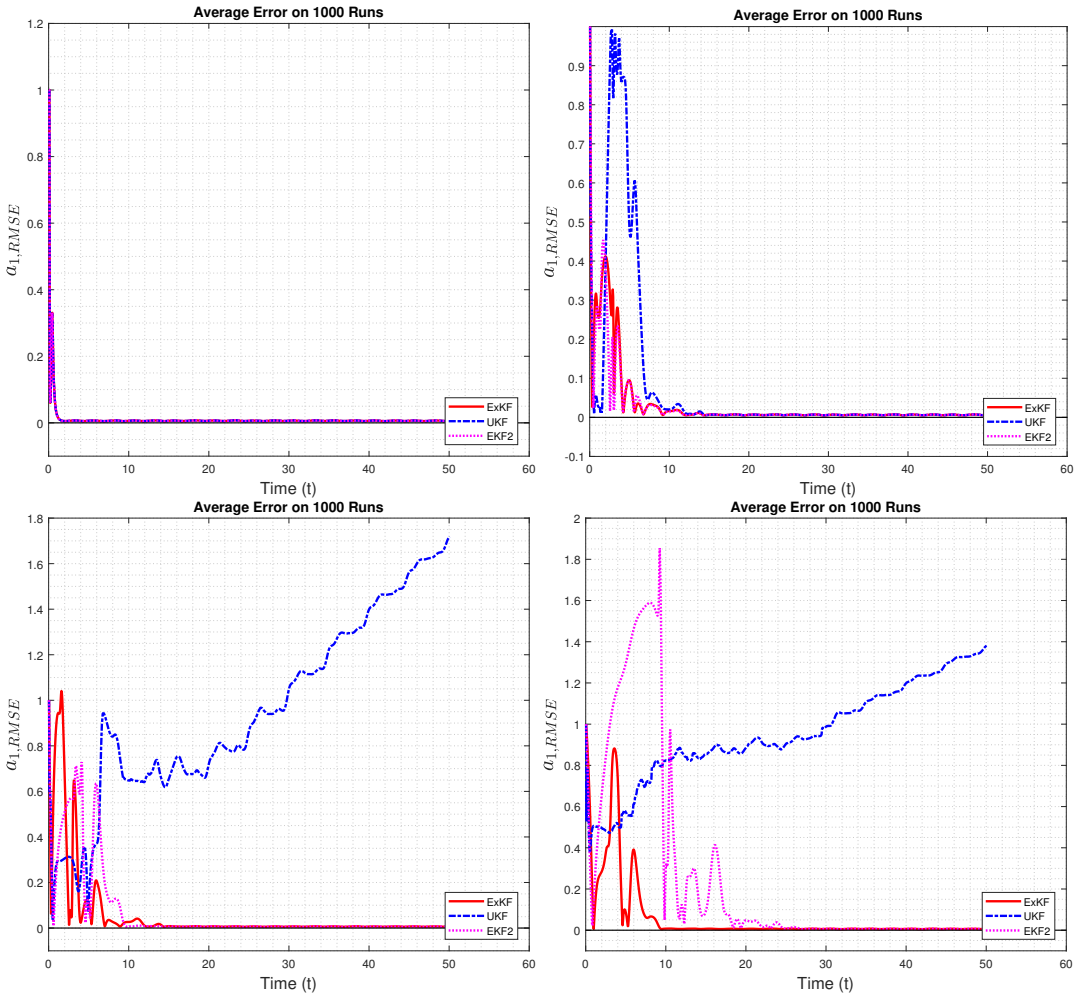

Figure 4.11: RMSE of a_1 for $\hat{\theta}_0 - \theta_0$ equal to 0 (top left), $\pi/4$ (top right), $\pi/2$ (bottom left) and $3\pi/4$ (bottom right) for the subcases of case # 4

- Case 5: Case 4 with $q_\omega = 0.005^2$ and $\boldsymbol{Q}_{ab} = 0.001^2\,\mathrm{diag}(0.1, 1, 1, 1)$.

The next case is devoted to examine the filter performances when the variance of the state θ passing through the severe nonlinearity of a sinusoidal function is increased. What is different from the previous case is that the variance associated with angular velocity is higher than the variances of the Fourier series coefficients. Our expectation, which can be confirmed when the results are compared to the previous case, is that all filters are likely to perform worse for the following two reasons. Firstly, approximations provided by UKF, EKF1 and EKF2 will be less adequate. See Fig. 3.3 on page 42 as a reminder of the fact that worse mean and variance estimations are obtained for a sinusoidal mapping of $\sin x$ with increased variance of x. Secondly, the Gaussianity assumption used by all filters is violated more; refer to Fig. 3.4 on

page 45 to see the introduction of multimodality of the same sinusoidal mapping with increased variance of x. In accordance with these expectations, EKF1, EKF2 and UKF have divergence problems as can be seen in Fig. 4.12; this divergence is newly introduced to EKF2 as it was convergent in the previous case. ExKF converges to the true value of θ and a_1 at around $t = 25$ and $t = 20$ seconds, respectively, in contrast to $t = 15$ seconds in the previous case. Their output RMSEs have also increased. Still, ExKF still has the lowest output RMSE of 0.115, with 78% improvement over EKF2 with an RMSE of 0.514 and with 87% improvement over UKF with an RMSE of 0.870. EKF1 still has a low output RMSE of 0.150 despite its worse state estimation performance.

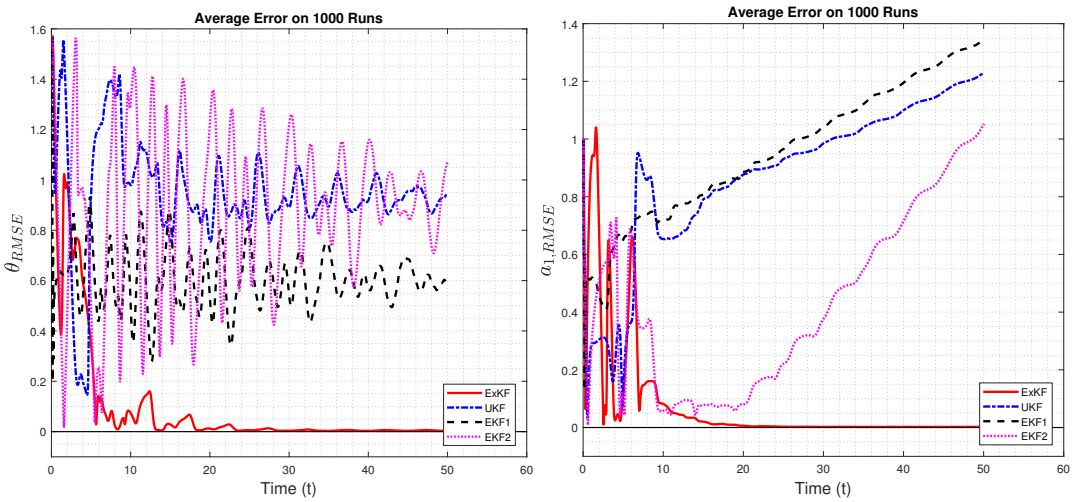

Figure 4.12: RMSE of θ (left) and a_1 (right) in test case # 5

Similar to the previous case, now convergence of filters with respect to initial angle estimation error will be examined, with all results of RMSEs presented in Fig. 4.13. This simulation also verifies that all filters are equivalent provided that initial angle estimate as well as its covariance are known. UKF cannot track a_1 only when $\hat{\theta}_0 - \theta_0 = \pi/2$ and in other cases its response shows initially large oscillations. Similar to the previous case, EKF2 cannot track a_1 when $\hat{\theta}_0 - \theta_0 = \pi/2$ and it has a residual errors in a_1 when $\hat{\theta}_0 - \theta_0 = 3\pi/4$. ExKF, on the other hand, can converge to a_1 in approximately 20 seconds, a slight increase from the previous case. This can, again, be attributed to the violation of Gaussianity for $\sin x$, the output of which ExKF approximates as Gaussian.

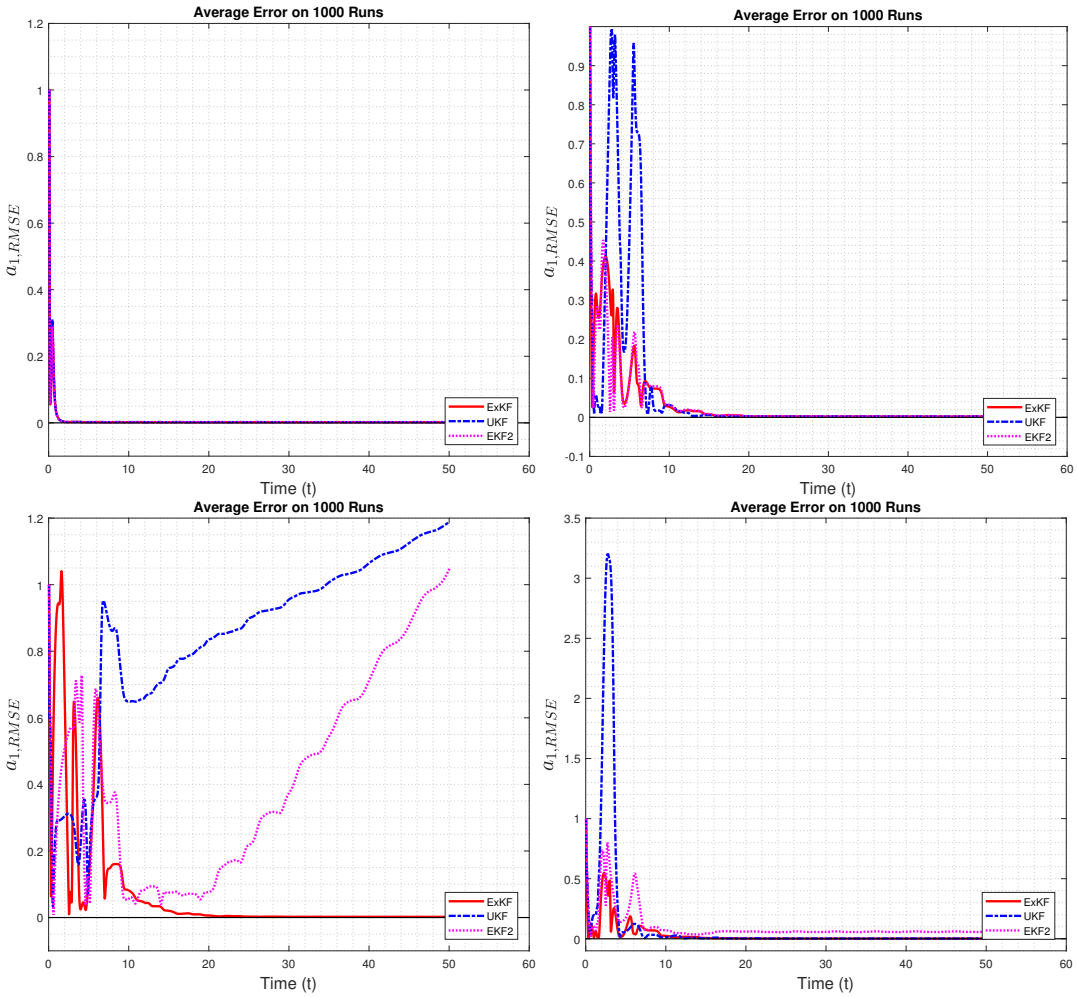

Figure 4.13: RMSE of a_1 for $\hat{\theta}_0 - \theta_0$ equal to 0 (top left), $\pi/4$ (top right), $\pi/2$ (bottom left) and $3\pi/4$ (bottom right) for the subcases of case # 5

- Case 6: Case 4 with $\boldsymbol{x}_0 = \text{diag}(0, 0, 0, 0, 0, 0)$.

The last case, Case 6, is devoted to wrong convergence results provided by UKF, a single run of tracking results can be seen in Fig. 4.14. This case corresponds to the one that both ExKF and EKF2 have their superiority over UKF. Both filters are divergent, while EKF2 unnecessarily increases the estimated mean and variance. UKF, on the other hand, has an apparent and inaccurate output convergence around $t = 40$ seconds and tries to converge to a model that does not fit the underlying one. Further evaluation of these results yield that all filters stop tracking the current angle θ, which can be seen on the left hand side of average error plot on Fig. 4.15; and average error plot for a_1 goes to zero only for the UKF. However, if one examines the convergence in a single run, the result can be seen in Fig. 4.16. Estimated coefficients

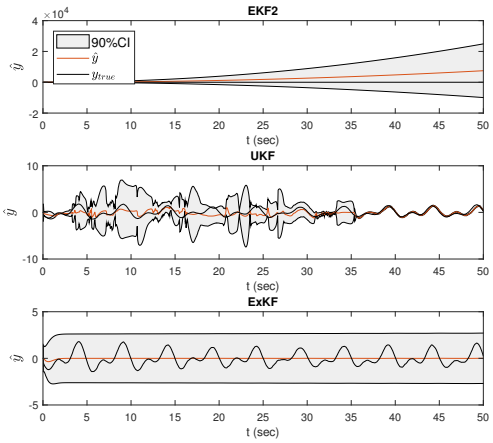

Figure 4.14: A single run of output tracking with wrong initial frequency estimate

no longer posses a random walk behaviour, unlike the ones that are used in simulation. A consistency test would be needed for the UKF to detect divergence, which would require more memory and computational resources. Interested readers are advised to read [51]; however, as we do not intend to use the UKF for tracking respiratory motion, we will not be checking its consistency.

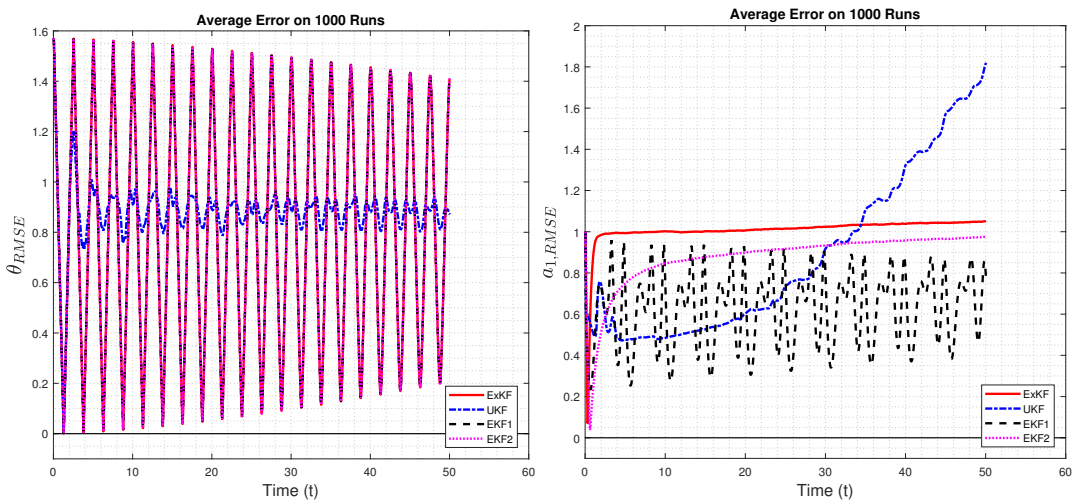

Figure 4.15: RMSE of θ (left) and a_1 (right) tracking errors in test case # 6

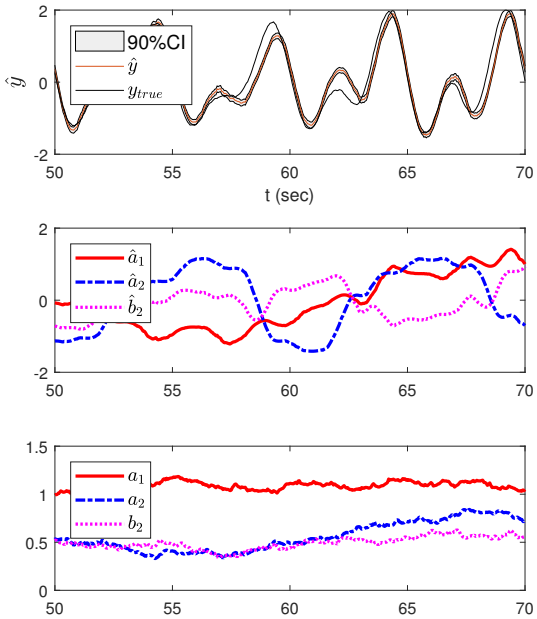

Figure 4.16: UKF incorrect convergence in test case # 6, estimated output (Top), estimated states (middle), true states (bottom)

In this section, we have compared the newly developed filter, ExKF, with two traditional filters. Below are the summary of conclusions drawn from the simulations that are beneficial for us while tracking a quasi-periodic signal such as respiratory motion.

1. From first two cases, the region where the Fourier series coefficients showed a decay ratio of at least 0.5, ExKF can converge to true coefficients while EKF1, EKF2 and UKF frequently cannot. Whenever there is another filter that can converge to the true model, ExKF is demonstrated to be at least on par with it with the unlikely choice of estimating the initial angle perfectly. Under nonzero initial angle errors, ExKF converges the fastest.

2. ExKF provides better state estimates both when the coefficients of the quasi-periodic signal change (case 4) and when the frequency of respiration change (case 5); this implies that respiratory motion can be tracked with less errors using ExKF.

3. UKF and EKF1 frequently have convergence problems.

4. EKF2 rarely has convergence problems, but it is generally slower than ExKF.

Our expectation from these simulation studies is that ExKF should also be the best

filter in tracking respiratory motion, with the fastest convergence irrespective of the initial error in θ. This is expected to hold on real data as well. However, we are aware of the fact that simulations are performed for a finite order model, whereas infinitely many coefficients can represent a periodic or quasi-periodic signal. It is possible that some coefficients may adapt to this high frequency components or completely avoid modeling it.

4.3 Experimental Results for Filter Comparison

4.3.1 Experimental Data Collection

This section describes the environment as well as the way measurements were taken to form the experimental database. Measurements are the outputs of position sensors we will introduce in detail. Position sensors are placed on front thorax or front abdomen of mechanically ventilated patients under general anesthesia to record respiratory motion. Respiratory motion measurements were collected by Elmed Electronics & Medical Industry & Trade Incorporated [52] with our collaboration and comprise of the motion traces observed at the sternum of patients, which is the frontal bone connecting ribs. The choice of such a placement is solely due to the recommendations of the operating room staff in order not to endanger the contamination of surgical sterile zone. Thanks to Elmed that organized the data collection environment, we are in possession of three respiratory motion records that we will use in our trials with real data in this thesis work: a regular breathing pattern for a 41-year-old, slightly overweight female (which will be used extensively in this study as patient 1 if not otherwise stated), a motion pattern with DC shift irregularities belonging to a 51-year-old, slightly overweight male (patient 2) and a motion pattern with DC shift & amplitude changes belonging to a 28-year-old male of normal weight (patient 3). During the data collection, all three patients were lying on their backs and they were undergoing laparoscopic surgeries. Motion records were taken approximately 15 minutes after the start of the operation. We are only interested in the type of breathing patterns present in these records. Age, sex and weight information is presented here for the sake of completeness as we do not intend to infer any information from these vari-

ables. That is because the number of samples is limited to draw any conclusions, causing us to resort to the results that was stated in section 2.2: that the respiratory motion is "independent of age, sex, height or body weight" [25].

Sensors used in collecting breathing data from the described patients belong to two classes of sensors: electromagnetic and optical position tracking systems. NDI Aurora [11] is one of the few commercially available products that can be used for electromagnetic position measurements. It uses a magnetic field generator (see Fig. 4.17 top left) to sequentially excite three orthogonal coils at every measurement cycle to obtain the position and the orientation of the tip of the sensor by measuring the induced voltages across the coils. Together with its required peripherals for amplification, digitization and processing, this sensor can measure the relative coordinates of field detecting coils which changes according to the breathing pattern of the thorax of the patient. When one considers the field generator, coils and peripherals as a system, its output is the position and orientation measurement of coil tips without requiring external inputs. We are inherently assuming that regular or irregular breathing motion is to reflect itself to external breathing motion which we can measure with a position sensor and the measurements we get from the output of NDI Aurora are indicators of breathing regularities. NDI Polaris, which can be seen in see Fig. 4.17 center left, is an optical position tracker to record breathing patterns and it uses an infrared illuminator to get the position of infrared reflective marker by using its two detectors. This optical measurement system gives more accurate measurements than its electromagnetic counterpart. Bottom left Fig. 4.17 belongs to the measurements taken by the NDI Aurora system, which are more noisy than bottom right Fig. 4.17 belonging to the NDI Polaris system. The main advantage of the electromagnetic position sensor over its optical tracking counterparts is that no line of sight is required between the field generator and the coils. This is the main reason for choosing it for data collection because obstruction of view is very likely in an operating room with various medical devices that are present.

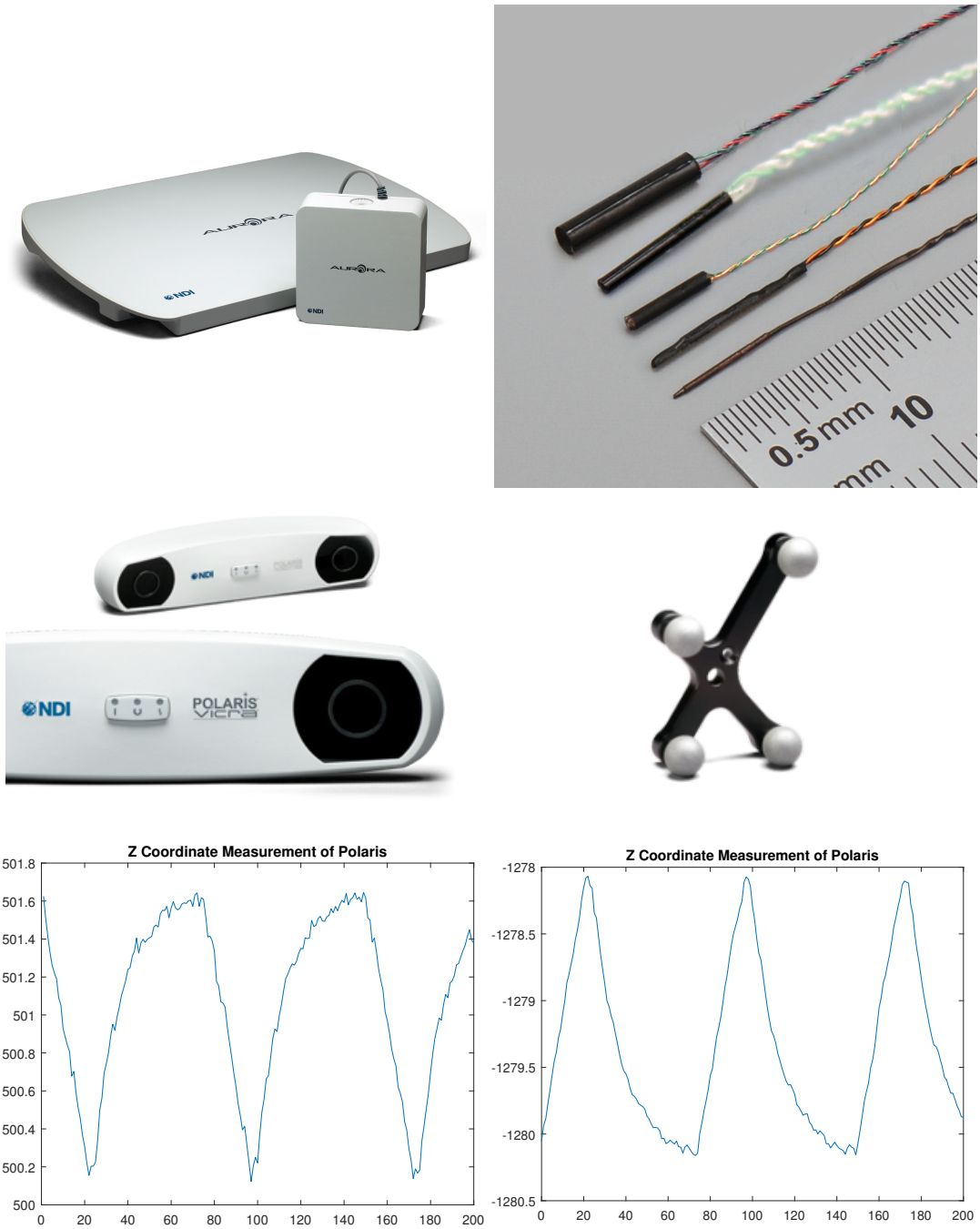

Figure 4.17: NDI Aurora field generator (top left) and field detecting coils (top right), NDI Polaris IR illuminator & detector (center left), IR reflector (center right) obtained from [11]. Respiratory motion measurements recorded with Aurora (bottom left) and Polaris (bottom right)

4.3.2 Results from Own Setup Data

What follows next is the demonstration of filtering performances on mechanically ventilated patients in the real environment described in section 4.3.1 from which real data was obtained.

One of the methods that we have used without any modifications is data dimensionality reduction using PCA. Although this simplifies our filtering problem, processing each dimension independently may increase the prediction accuracy. Krauss et al. [53] compared several models and found that PCA decreased the accuracy in tracking. They have also found out that the decrease in accuracy is dependent on the prediction model used; however, in none of their comparisons error deterioration was more than %5, so such an investigation is unlikely to greatly improve our estimation performance. Yet, processing data without applying of PCA can be informative to see whether the data shows any hysteresis, which may help us in phase estimation. We will present the filtered output without PCA at section 5.1.

Data is preprocessed using PCA to remove the DC offset of measurements as well as to project it to a subspace where the variance of data is maximized. See Fig. 4.18 for an example of this input output relationship, where PCA_1 is to be fed to our filters.

The order of the model is chosen as $m = 5$, which will be shown in the next chapter as the optimum one. Measurement standard deviation is given as $\sqrt{R} = 0.0137$ mm for each of the x, y, z axis of the Aurora 6-DoF magnetic tracker [54]; and since the PCA operation does not alter the variance of data (and equivalently noise) measurement standard deviation is empirically set to $\sqrt{R} = 0.0137$. As for the tuning of the process noise, angular velocity variance is set to a small value to reflect our belief in the almost exact periodicity under fully mechanical respiration ($\sqrt{\sigma_\omega} = 0.0001$). Coefficient covariances were gradually decreased under random walk behaviour were observed for all coefficients. Initial state estimates follow a similar pattern as in the simulations (i.e. $\begin{bmatrix} 0 & 1 & 0.5 & 0.5^2 & \dots \end{bmatrix}$): these estimates are also in accordance with the FS coefficients given at the beginning of this chapter. Such a good estimate is of benefit especially for UKF and EKF2 as they have poorer convergence properties. Initial standard deviations for coefficient estimates are given a large value of 1, slightly

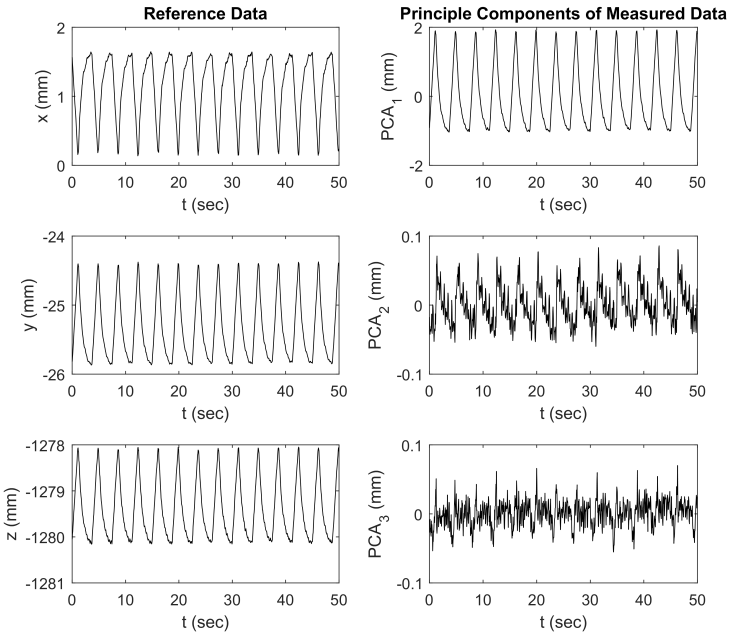

Figure 4.18: PCA inputs and outputs

larger than one third of the peak to peak amplitude of the observed signal: this was done with the assumption that the FS coefficients may not be larger than the amplitude of the signal, which clearly is the case if one refers to Fig. 4.1. Breathing frequency is $\omega_0 = 2\pi\frac{16}{60}$ (for 1/5 breaths per second) with high confidence ($\Sigma_{0,\omega} = 10^{-4}$) as the patient is mechanically ventilated with 16 breadths per minute. Initial angle estimate is $\theta_0 = 0$ with a confidence level that is bounded it between $-\pi$ and π ($\Sigma_{0,\omega} = (\frac{\pi}{3})^2$).

The results can be seen in Fig. 4.19. EKF2 has an initial overestimation of the output variance (top right subfigure) and it cannot handle wrong initial angular position estimate, as the value of a_1 stays around 0. EKF2 tries to explain the oscillatory behaviour of the output by altering the values of Fourier series coefficients instead of tracking the current angle (top left figure). UKF, on the other hand, underestimates the output variance and only after 10 seconds of oscillatory behaviour it can track the output (center right subfigure). ExKF could capture the exact output around $t = 5$ seconds (bottom right subfigure). ExKF also shows a more stable convergence than UKF if one examines the variance associated with the predicted output.

As the initial estimate of θ was important during simulations, we wanted to try two different values for it. The results for $\hat{\theta}_0 = \pi/2$ can be seen in Fig. 4.20. As the

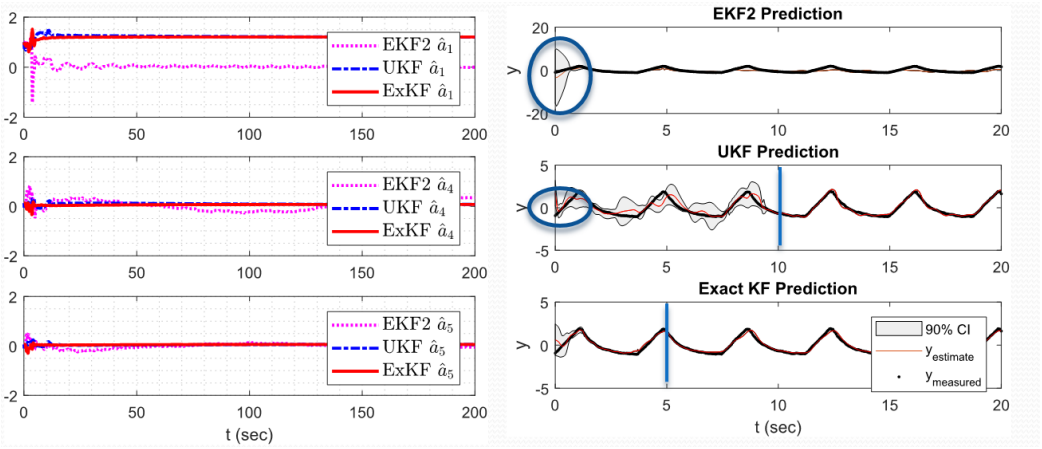

Figure 4.19: Experimental results on real data - correct angular frequency estimate, predicted outputs (left) and predicted states (right) with $\hat{\theta}_0 = 0$

tracked signal is the same as the previous case, the coefficients should converge to the same values. EKF2 again overestimates the initial output variance, although the amount of overestimation is much less than the previous case and it cannot track the output as it is the case for $\hat{\theta}_0 = 0$. UKF greatly underestimates the output variance in this case, resulting in premature convergence of a_1 to a wrong value. This premature convergence results in increased output variance at around $t = 6$ seconds and wrong output estimation after that point. ExKF achieves output convergence in about $t = 5$ seconds and its state estimate a_1 is equal in magnitude to that of the previous case. As we have stated at the end of section 4.1 that we are only interested in the amplitude of a_1, reflecting its sign to θ, both converged values are equivalent.

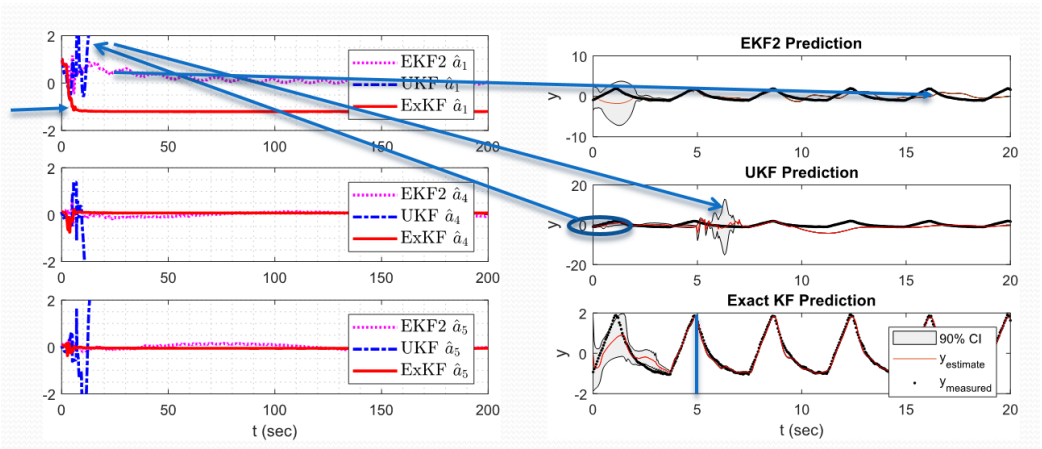

Figure 4.20: Experimental results on real data - correct angular frequency estimate, predicted outputs (left) and predicted states (right) with $\hat{\theta}_0 = \pi/2$

With $\hat{\theta}_0 = \pi$, overestimation of initial output variance by EKF2 further decreased, enabling EKF2 to be able to track the output. UKF again fails to converge to the true output. Both EKF2 and ExKF can achieve output convergence in about 5 seconds if one examines the left plot given in Fig. 4.21. EKF2 converges to a negative value of a_1 and ExKF converges to its positive value, with both being equivalent with the justification given in the last sentence of the paragraph above.

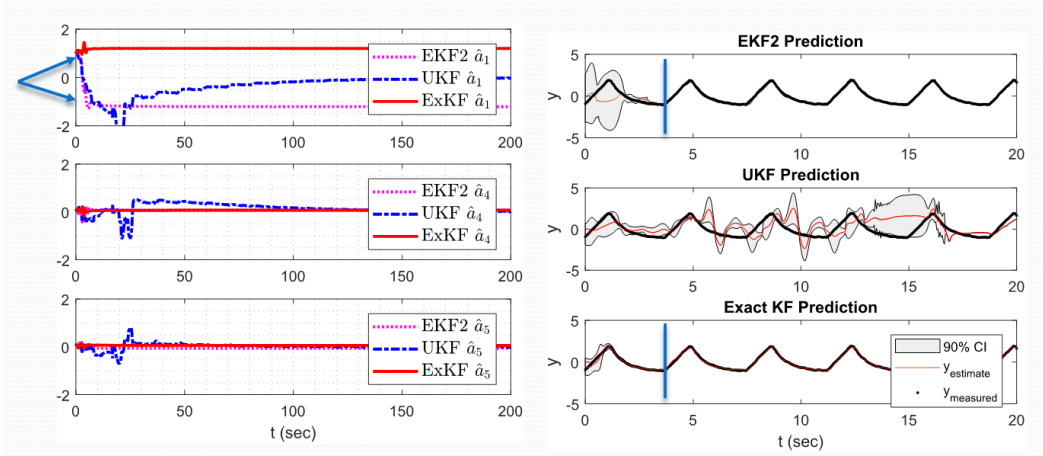

Figure 4.21: Experimental results on real data - correct angular frequency estimate, predicted outputs (left) and predicted states (right) with $\hat{\theta}_0 = \pi$

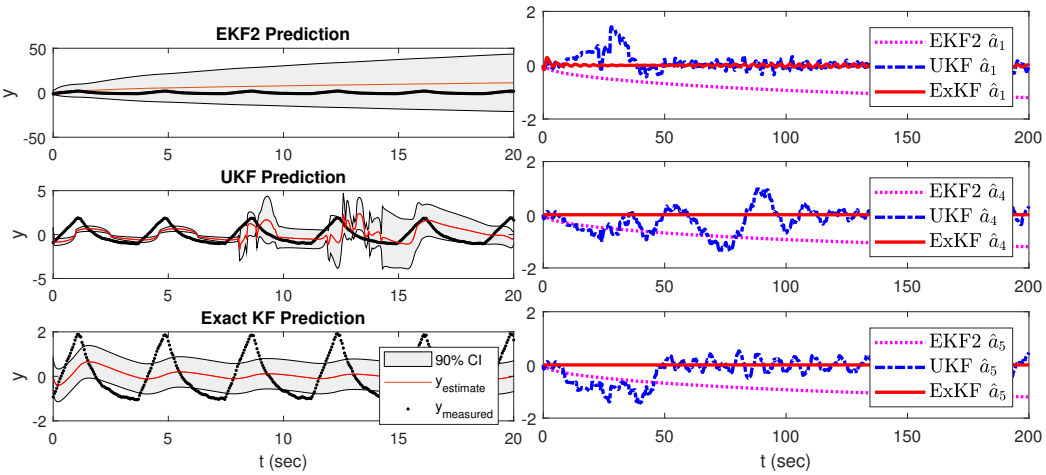

Figure 4.22: Experimental results on real data - wrong angular frequency estimate, predicted outputs (left) and predicted states (right)

One last trial was performed for estimating the breathing fundamental period incorrectly as 0 with all other parameters as stated at the beginning of this section. As can be seen in Fig. 4.22, ExKF and EKF2 show divergent estimates while the UKF tries to track the output by altering the coefficients. Its divergence can only be detected by

looking at the sequence of state residuals and check whether they show a chi-square distribution, which would increase both the memory and computational requirements. As a final remark, note that the measurements y in ExKF predictions (bottom left plot in Fig. 4.22) frequently fall outside of the 3σ range unlike the case in our simulations. This indicates that process noise tuning is incorrect for some states, which is expected as the manual filter tuning is prone to errors.

Results from these trials indicate that,

1. Conclusions from simulations are valid from the previous section, in the sense that ExKF is better at tracking respiratory motion. Its convergence speed is faster than UKF, and is comparable to EKF2. However, in our trials with three different angle estimates, it is the only filter that achieves convergence in all cases whereas the other two could not achieve convergence in 2 out of 3 cases.

2. Manual tuning of all of the filters are erroneous; thus, there is a need for automatic process noise tuning for better accuracy at tracking.

3. Building another filter for tuning on top of a slow or frequently divergent one can make the tuning process problematic. That is why we won't be proceeding with the two filters, EKF2 and UKF, in the next section.

CHAPTER 5

SENSITIVITY ANALYSIS IN FILTER TUNING OF OUR BREADTH TRACKING SYSTEM

In this chapter, we will be tackling the sensitivity of tuning of our filter to various parameters. First, we will evaluate how much information is lost due to the application of PCA for data projection. Then, process noise covariance tuning results will be presented in section 5.2 using artificially generated data in order to see the effect of stopping criterion as well as the number of data points used in each run. In the same section, we will analyze how the accuracy of tuning is affected by these parameters by comparing the tuned variances against the simulated ones. In section 5.3, we will evaluate the combined performance of ExKF and its tuning on the respiratory records of mechanically ventilated patients based on output estimation accuracy as well as the fitness to the assumed Gaussian model.

5.1 Filtering Without PCA

In this section we will evaluate how much information is lost due to the usage of PCA and the projection process. Fig. 5.1 shows a portion of data and its projection. Examining the top view of this data, it is apparent that the hysteresis information is lost with this projection.

Filtering all three dimensions separately gives the filtered estimates shown in Fig. 5.2, where the top row is for raw x measurements, the second and the third ones are for raw y and z measurements, and the last row is for the processed data. Now, as can be seen in the figure, the ExKF can process each dimensions separately, as well as their

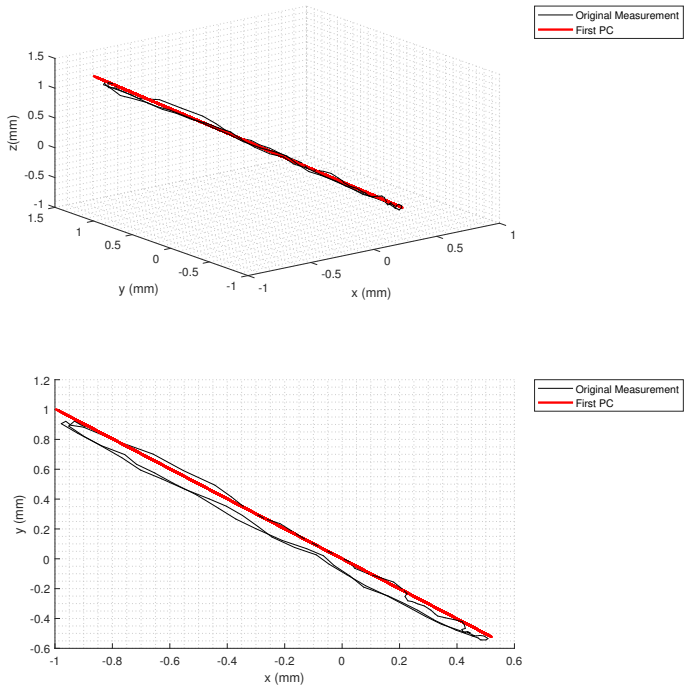

Figure 5.1: Original measurement and its projection, arbitrary 3D view (top) and top view (bottom)

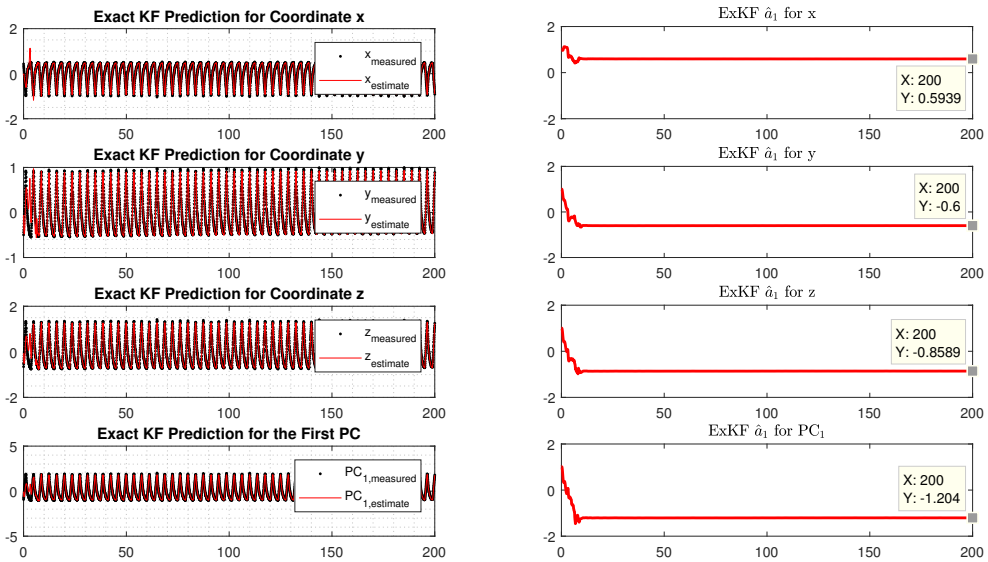

Figure 5.2: Filtered output predictions (left) and estimated first order coefficients a_1 (right)

projection onto the first PC. We will examine the amount of information loss through the first order coefficient estimate a_1 by making use of the cursor measurements as follows:

$$\sqrt{a_{1,x}^2 + a_{1,y}^2 + a_{1,z}^2} = 1.208 > a_{1,PC} = 1.204$$

The amplitude of motion captured by the three Kalman filters, one for each dimension, is larger than the amplitude of information captured by the first PC. However, if we compare their amplitude ratios, $a_{1,PC}$ is greater that 99% of $\sqrt{a_{1,x}^2 + a_{1,y}^2 + a_{1,z}^2}$. That is why, the information carried in the first PC is sufficient for respiratory motion prediction and processing each dimension separately unnecessarily increases the computational cost of filtering by three times.

5.2 Kalman Filter Tuning in Simulation

5.2.1 Effect of Varying the Stopping Criterion

In this section, tuning parameter selection process is justified through simulations. The stopping criterion is the 2-norm of the updated covariance being less than a threshold, that is, when $||\boldsymbol{Q}_l - \boldsymbol{Q}_{l-1}|| < \epsilon$ and we will start with the justification of the first parameter ϵ which terminates the expectation maximization algorithm. We adopt the same parameters used in case 4 of the simulation section 4.2 which are $q_\omega = 0.001^2$ and $\boldsymbol{Q}_{ab} = 0.005^2\,\mathrm{diag}(0.1, 1, 1, 1)$ to reflect our belief that mechanical ventilation allows strict periodicity through ω. Initial covariance estimates are chosen as 1000 times the original ones and $\epsilon = 10^{-6}$. This value of ϵ is changed within this subsection to see its effect on the process noise convergence. The other parameter chosen for this tuning is the number of data points used for training, with N=300 to cover 3 breathing cycles. This parameter will also be altered at a later stage of this analysis (section 5.2.2) to record its effect on filter tuning. The result of the experiment with $\epsilon = 10^{-6}$ can be found in Fig. 5.3. The expectation maximization algorithm stops after 44 iterations and it can bring the coefficient variances to approximately $10 - 20$ times the real one as can be seen in Fig. 5.3.b-d. More specifically,

Σ_{a_1} in b, Σ_{a_2} in c and Σ_{b_2} in d are 11, 12 and 13 times their true values. However, since the covariances associated with angular position and speed are low, errors in their estimates are large when compared to the real ones, which can be seen in Fig. 5.3.e-f. This shows that either the stopping criterion needs to be decreased or the EM algorithm may not be run only once. The latter also makes sense as the process noise covariance does not have to be stationary and online EM is needed for adjustments when $\boldsymbol{Q}$ changes.

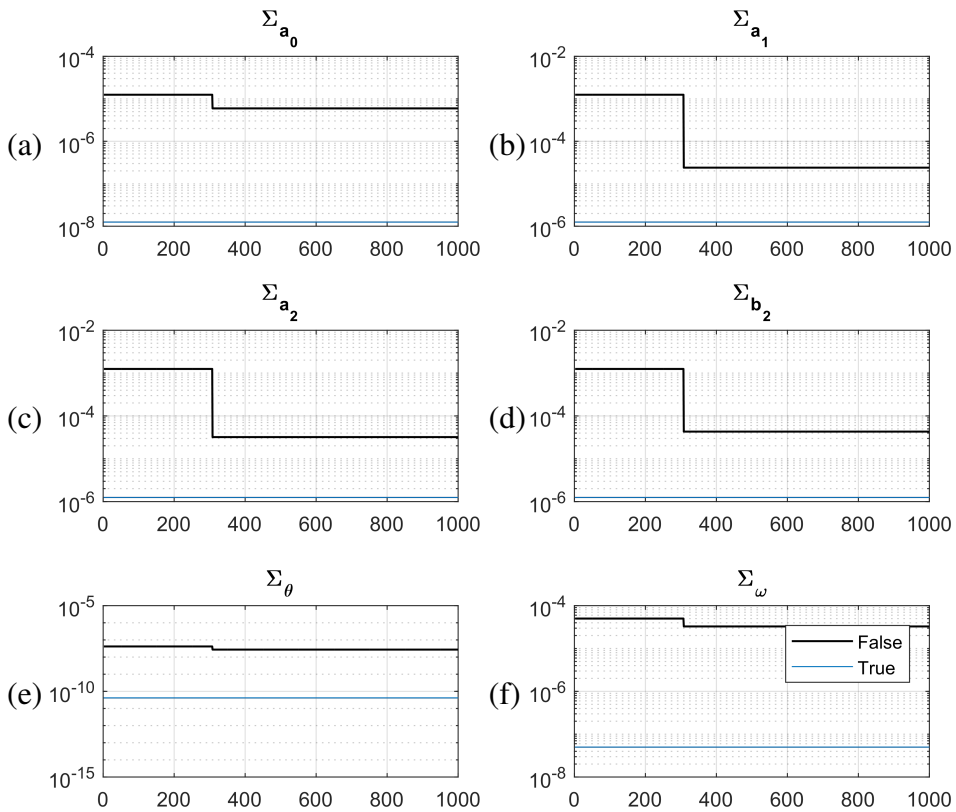

Figure 5.3: Simulation results of variance estimates for overestimating the process noise with a single stopping criterion as $\epsilon = 10^{-6}$ (one trial)

The basic remedy to fix the residual errors in variances is decreasing ϵ lo a lower value although this would increase the computational requirements of the EM algorithm. Our trial with decreasing ϵ to 10^{-8} provides far better variance estimates as can be seen in Fig. 5.4. Looking at Fig. 5.4.b, it can be seen that the variance of the first FS coefficient can be estimated to a value of 1.1 times the real one, which is an acceptable level as compared to the manual tuning procedure. The other variance estimates show slight improvements as well: Σ_{a_2} reaches 11 times the real one (Fig. 5.4.c) as compared to the value of 12 for the case with $\epsilon = 10^{-6}$ while Σ_{b_2} in Fig. 5.4.d remains at 14 times the actual one. The number of iterations the EM algorithm needs increases to approximately 450 iterations, almost a tenfold increase as compared to $\epsilon = 10^{-6}$. This increase can be justified with the fact that iterations are performed even for smaller updates in $\boldsymbol{Q}$ thanks to smaller stopping criterion.

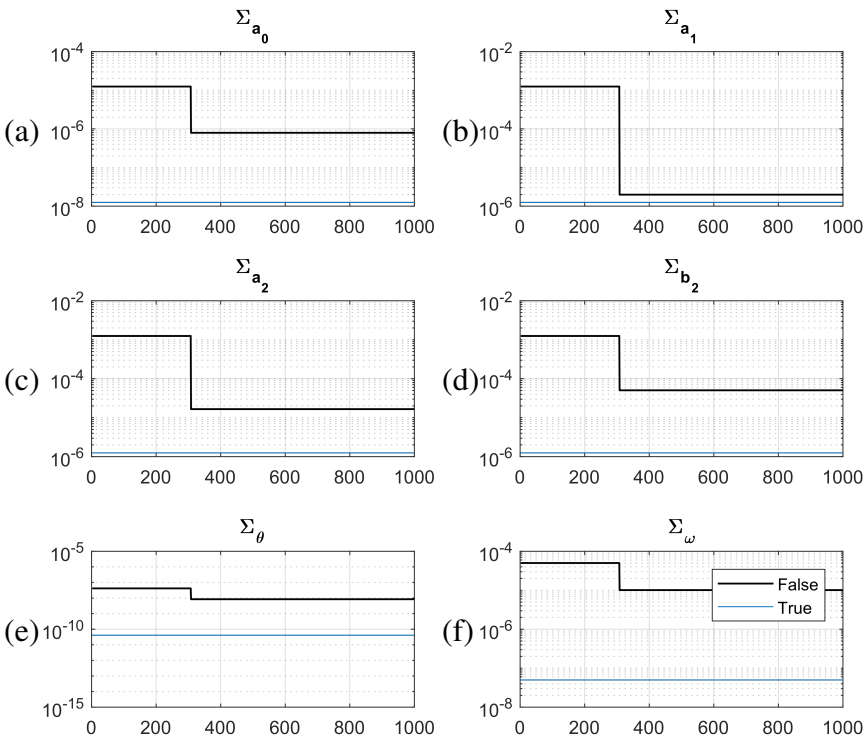

Figure 5.4: Simulation results of variance estimates for overestimating the process noise with a single stopping criterion $\epsilon = 10^{-8}$ (one trial)

Forcing the solution a bit further with further decreased ϵ to a value of 10^{-10} gives the result in Fig. 5.5. As compared to the previous case with $\epsilon = 10^{-8}$, variance associated with a_1 is underestimated this time, see Fig. 5.5.b. Variance estimate of a_2 in Fig. 5.5.c shows a slight and insignificant improvement of decreasing from 11 to 10 times the original variance while the variance estimate associated with b_2 in Fig. 5.5.d does not show any improvement. Only estimates that benefit from this decrease are the variance estimates of θ and ω, but that comes at a huge expense as the number of iterations needed reaches the value of 4400, another tenfold increase. What is undesirable is that the combined effect of filtering and tuning lasts slightly less than 50 seconds, on a PC with a CPU speed of 3.20 GHz; this causes our prediction algorithm to almost lose its ability to operate on real time as 1000 data points correspond to a data sequence covering a time interval of 50 seconds. That is why further decrease of ϵ will not be beneficial and will not be tried.

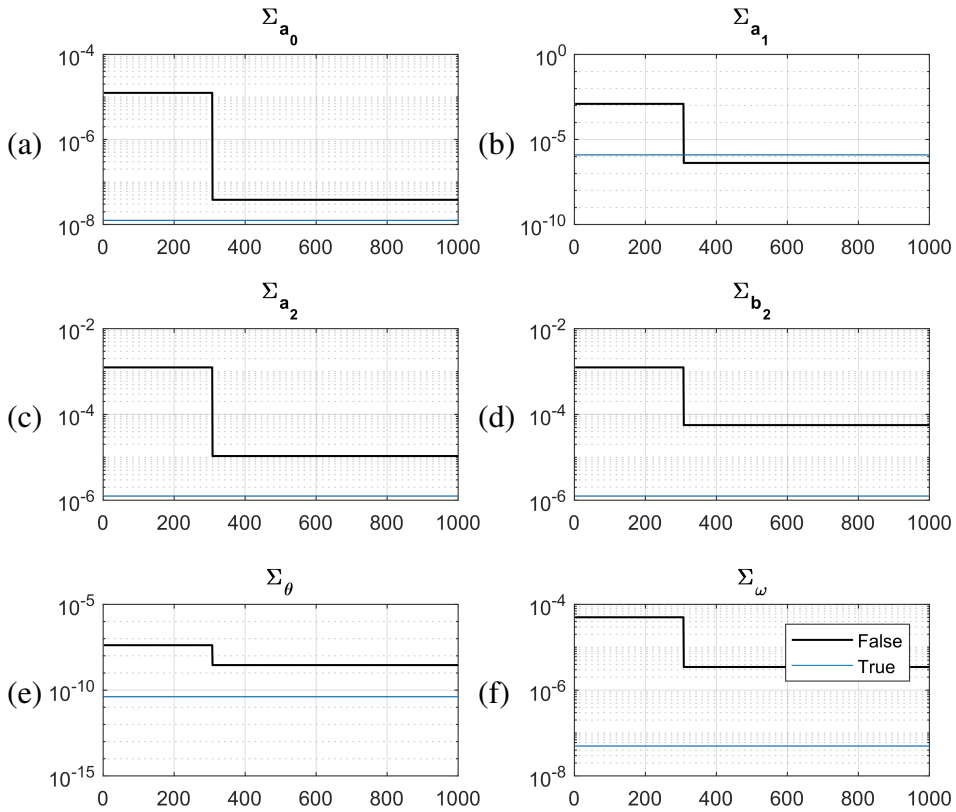

Figure 5.5: Simulation results of variance estimates for overestimating the process noise with a single stopping criterion $\epsilon = 10^{-10}$ (one trial)

As an alternative of decreasing ϵ more, we tried setting different values for norm matrix updates: ϵ_1 for the subblock of coefficient covariance estimates and ϵ_2 for angular velocity and position estimates. Such settings yield approximately the same results as obtained in the previous cases, the results of which we shall not report here due to its insignificance. The values we chose were $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-8}$ which give almost identical results to that of $\epsilon = 10^{-6}$; the choice of $\epsilon_1 = 10^{-6}$ and $\epsilon_2 = 10^{-10}$ produces almost equivalent results as $\epsilon = 10^{-8}$. That is why, we omit their inclusion in this thesis work.

Instead a better solution was found using N number of data points in each EM iteration, performing EM algorithm at least once upon the arrival of a new data point and stopping with the same criterion as in the case that can be seen in page 92 which is iterating until L2 norm convergence. By setting $\epsilon = 10^{-6}$, the result can be seen in 5.6. Initial convergence jumps are comparable to those in Fig. 5.3. As an example, consider the plot given in Fig. 5.6.b: Σ_{a_1} starts with 1000 times the true variance and the initial iterations of EM at N=300 brings this value to 200 times the true one exactly like the one EM is run once at N=300 without online updates as in Fig. 5.3.b. Estimated coefficient variances do show some oscillatory behaviour after approaching the true variances. Estimated variance of ω, on the other hand, oscillates somewhere between 1.5 and 0.5 times the true variance, far better than the ones in the coefficients.

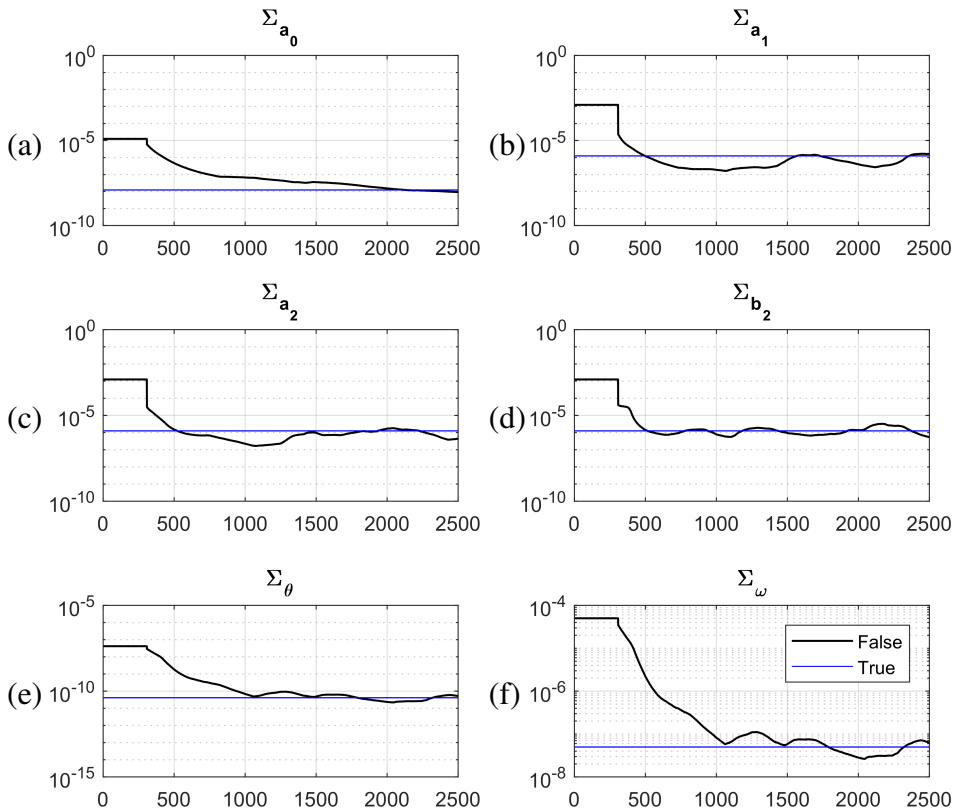

Figure 5.6: Simulation results of variance estimates for overestimating the process noise with a single stopping criterion $\epsilon = 10^{-6}$ and continuous updates (one trial)

Further decreasing or increasing ϵ only changed the initial amount of jumps observed until the stopping criterion is met. As an example, left hand side of the Fig. 5.7 shows the variance estimates of a_1, b_2 and ω for $\epsilon = 10^{-8}$ while the right hand side plots show the variance estimates of the same variables for $\epsilon = 10^{-4}$. Setting $\epsilon = 10^{-8}$ gives the correct variance estimate for a_1 even in the initial iterations at N=300 as can be seen in Fig. 5.7.a. However, angular velocity convergence in Fig. 5.7.e and 5.7.f are still comparable in the sense that convergence is first achieved at around N=1000. The oscillations observed after estimates reach the true variance values still persist: an order of magnitude of error can be seen for Σ_{b_2} with $\epsilon = 10^{-8}$ in Fig. 5.7.c and an order of magnitude of error can be seen for Σ_{a_1} with $\epsilon = 10^{-4}$ in Fig. 5.7.b. Note that there are no multiple EM iterations after N=300 in all plots as there are no step-wise changes after N=300. That is why it can be concluded that this oscillatory behaviour is independent from the value of ϵ.

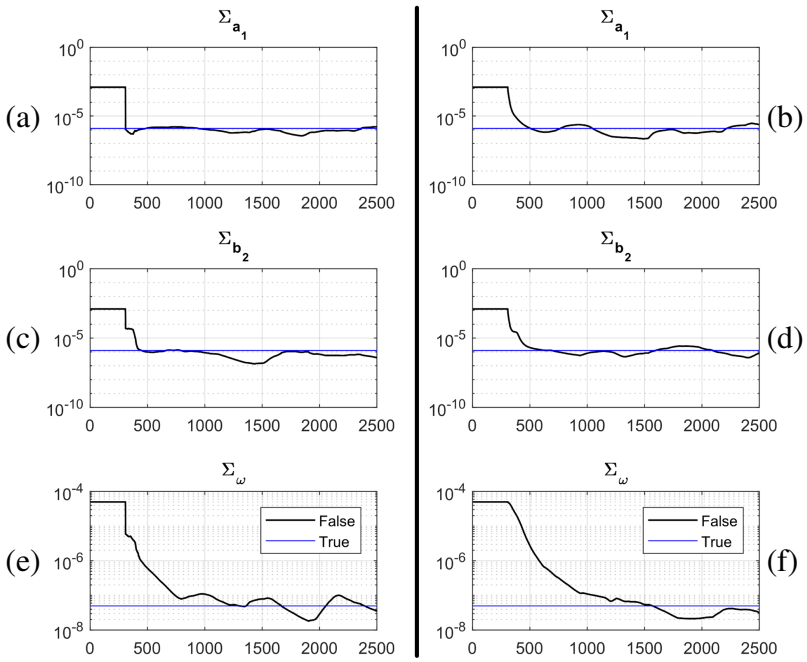

Figure 5.7: Simulation results of variance estimates for overestimating the process noise with a single stopping criterion $\epsilon = 10^{-8}$ (left) and $\epsilon = 10^{-4}$ (right) (one trial)

Table 5.1 shows the approximate convergence times for different values of ϵ. Although the steady state behaviour is oscillatory, what we take as rise time is the first time the variance reaches the value 10% close to the actual one. a_0 is disregarded in all of the coefficient variance rise times as it takes approximately 2500 number of epochs for it to reach the true value no matter what the ϵ value is. With $\epsilon = 10^{-2}$, the EM algorithm always meets the stopping criterion, resulting in a single EM run each time new data arrive. With $\epsilon = 10^{-4}$, EM runs 4 times more initially. Rise times of all variances remaining almost the same. $\epsilon = 10^{-6}$ improves the rise times of θ and ω with a slight additional computational burden. Choosing $\epsilon = 10^{-8}$ improves coefficient variance rise times by approximately 20% by also doubling the number of EM iterations. As its effect on rise times is substantial, we opt to choose ϵ as 10^{-6}.

Table 5.1: Simulation Results - Total Number of EM Iterations and Approximate Rise Times for Different Values of ϵ

ϵ	10^{-2}	10^{-4}	10^{-6}	10^{-8}
Coefficient Var. Rise Times	500	500	500	400
θ Var. Rise Time	1500	1250	1000	1000
ω Var. Rise Time	1500	1500	1000	1000
Number of EM Iterations	2196	2200	2239	4544

One final remark before we proceed to the effect of number of data points used in EM algorithm is that the initial estimate in $\boldsymbol{Q}$ should be chosen greater than the actual one for ϵ to be useful. That is, if $\hat{\boldsymbol{Q}}_0$ has much smaller eigenvalues than $\boldsymbol{Q}$, the stopping criterion would always be satisfied with $|10^{-6} - \epsilon_1 10^{-6}| < 10^{-6}$ for a small positive number ϵ_1. That is why, overestimating $\hat{\boldsymbol{Q}}_0$ should be preferred for ϵ to be effective in speeding up convergence. Otherwise, the results we obtained showed that convergence speed is as slow as setting up a small value of $\epsilon = 10^{-2}$.

5.2.2 Effect of Varying the Number of Data Points

In this section, we will examine how changing number of data points used in our simulations affects the convergence properties of the EM algorithm. At first, we will start the tuning process with less number of data points than a breathing period. To do so, we choose the number of data points as N=50, covering half period of a breathing

cycle. The result can be seen in Fig. 5.8. It is clear from these results that the EM algorithm fails to capture the true variances, especially with Σ_θ and Σ_ω being divergent in Fig. 5.8.e and 5.8.f, respectively.

To understand the effect of this divergence better, it is worth remembering the apparent convergence achieved by UKF under case # 6 in section 4.2. In this case, UKF has an apparent output convergence even if the estimated FS coefficients do not show a random walk behaviour. Interested readers are advised to refer to Fig. 4.16 on page 80 to reexamine this behavior. This case shows that Kalman filter does not take into account whether a state innovation sequence shows a random walk behavior as long as the output convergence is achieved, despite the fact that the underlying model does show a random walk behavior. Such a misleading output convergence is more likely when we employ two Kalman filters: one for state prediction and one for process noise prediction.

This divergence can be explained as follows: initial output convergence is achieved by altering the values of θ and ω thanks to their initially overestimated covariances. Then, this causes the EM algorithm to further increase the variances associated with these parameters. This, in turn, results in larger changes in θ and ω. The two Kalman filters continue to engage in this vicious circle despite the violation of Gaussianity in state innovation sequences in θ and ω. Moreover, EM takes approximately 10000 iterations trying to explain the measurement, that is, trying to achieve output convergence.

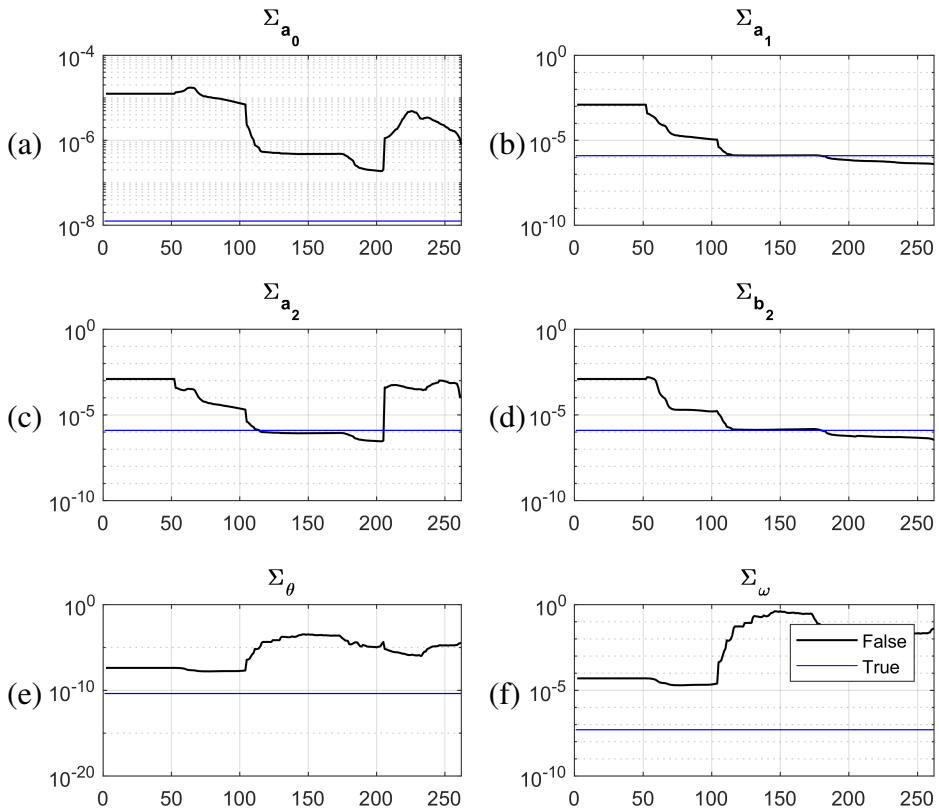

Figure 5.8: Simulation results of variance estimates for overestimating the process noise with $\epsilon = 10^{-6}$ and N=50

Increasing N to 100 brings it to a critical limit in terms of estimated variance convergence. As the frequency of respiration may either (i) increase or (ii) decrease during the simulations, the EM algorithm may (i) capture or (ii) fail to capture a full cycle of respiratory motion. A case where the EM is able to converge is shown in Fig. 5.9.a and a case there it is not able to converge is shown in Fig. 5.9.b. In our trials, the results were divergent more than 50% of the time. The EM algorithm takes approximately 2760 evaluations whenever the model is convergent, while it took more than 60000 evaluations for the result given in right hand side of Fig. 5.9 even when the simulation was stopped with reduced number of epochs as 1700.

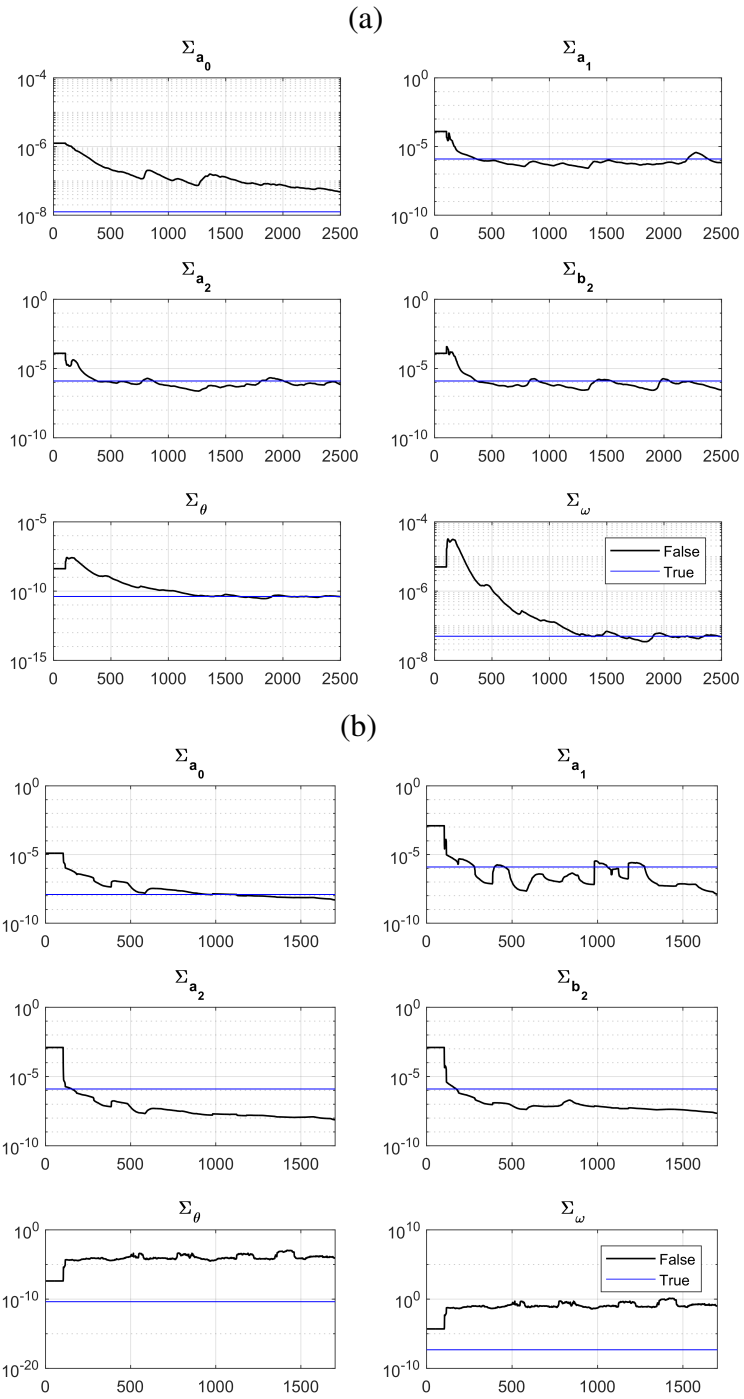

Figure 5.9: Simulation results of variance estimates for overestimating the process noise with $\epsilon = 10^{-6}$ and N=100, convergence (top) and divergence (bottom)

Now, the convergence results for N=200 and N=300 can be seen in Fig. 5.10 (on page 104) and for N=400 and N=500 can be seen in Fig. 5.11 (on page 105), with number of iterations EM performed as well as computational time presented in Table 5.2. All window lengths provide almost comparable predictions of Σ_{a_2} and Σ_{b_2} when one compares these two figures: a similar order of magnitude of error is observed in subplots c and d no matter what the number of data points is used. Increasing number of data points used in EM improves the performance of the estimator in finding Σ_{a_1} while its performance deteriorates for Σ_{ω}. To demonstrate this, we shall compare the Fig. 5.11.b and 5.11.f for the case with N=400 to the Fig. 5.11.h and 5.11.l for the case with N=500, where this is the most apparent. Starting with the process noise estimate of a_1, Σ_{a_1} is chosen as $1.25 \cdot 10^{-6}$ in both simulations. Using 500 data points bounds $\hat{\Sigma}_{a_1}$ between 0.5 to $2 \cdot 10^{-6}$ while using 400 data points bounds it between 0.3 to $1 \cdot 10^{-6}$, where the latter bound does not even contain Σ_{a_1}. As for the process noise estimate, Σ_{ω} is chosen as $5 \cdot 10^{-8}$. Using 500 data points bounds $\hat{\Sigma}_{\omega}$ between 1 to $10 \cdot 10^{-8}$ while using 400 data points bounds it between 1.1 to $5 \cdot 10^{-8}$, where the latter has a tighter bound.

As for the computational burden, increasing N decreases the initial amount of iterations needed until reaching the stopping criterion. To clarify this point, we have a total number of 2500 data points and a training window size of N=200 implies that the EM algorithm runs 2300 times if the stopping criterion is always satisfied, giving us the remaining number of iterations that are run initially as 132. Thus, the remaining number of initial iterations decreases with increasing N. This makes sense if one considers the fact that more information is embedded in a window length of larger size; therefore, it should be easier to extract this information with less number of iterations. However, this does not imply less computations as it is shown in the table with increasing computational cost as N increases, limiting our ability for real time tracking. We have decided to stop evaluating the effect of increasing N at 500 because even at this rate, the algorithm is barely able to run at real time. That is, for a sampling rate of 20 Hz, 2500 data points correspond to a data duration of 125 seconds. A real time tracking algorithm should process this data in less than 125 seconds, that the choice of N=500 barely achieves.

Before proceeding to experimenting on real data from breathing patterns of a me-

Table 5.2: Simulation Results - Total Number of EM Iterations and Approximate Rise Times for Different Values of ϵ

N	50	100	200	300	400	500
Number of Iterations	NA	NA	2432	2245	2139	2037
Initial Number of Iterations	NA	NA	132	45	39	37
Computation Time (sec)	NA	NA	62	89	111	118

chanically ventilated patient, our conclusions from this section are:

- The EM algorithm does provide better results that can adapt to underlying process noise. However, its accuracy in estimating coefficient noise variances may be accurate up to an order of magnitude. This accuracy can be increased by increasing the number of data points used in training at the cost of deteriorating accuracy in estimating angular velocity variance. However, elimination of covariance estimate errors does not seem possible with this approach due to the nonlinearities involved in state estimations.

- Finite order modelling of respiratory motion may cause several problems when we use real data. If the assumed order is insufficient in verifying the model against real data, covariance estimation may blow up, which we will show in the next section.

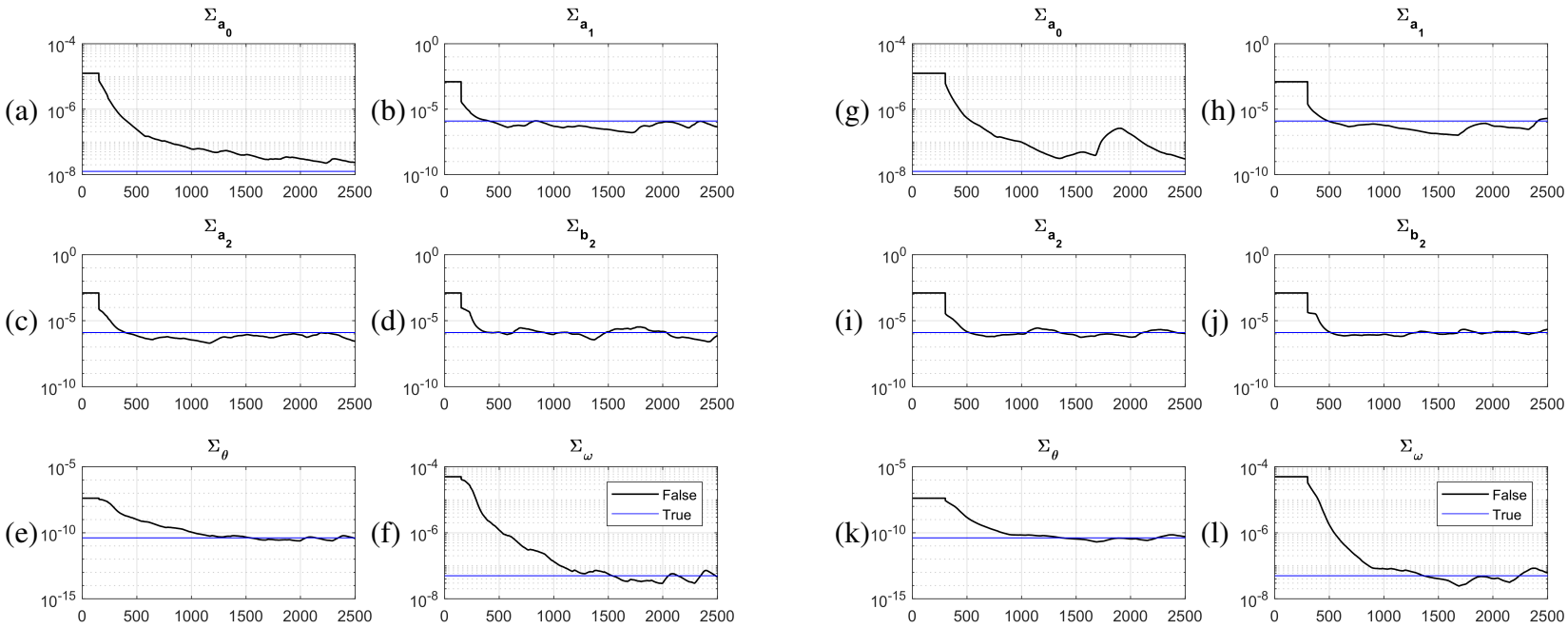

Figure 5.10: Simulation results of variance estimates for overestimating the process noise with $\epsilon = 10^{-6}$, N=200 (left) and N=300 (right)

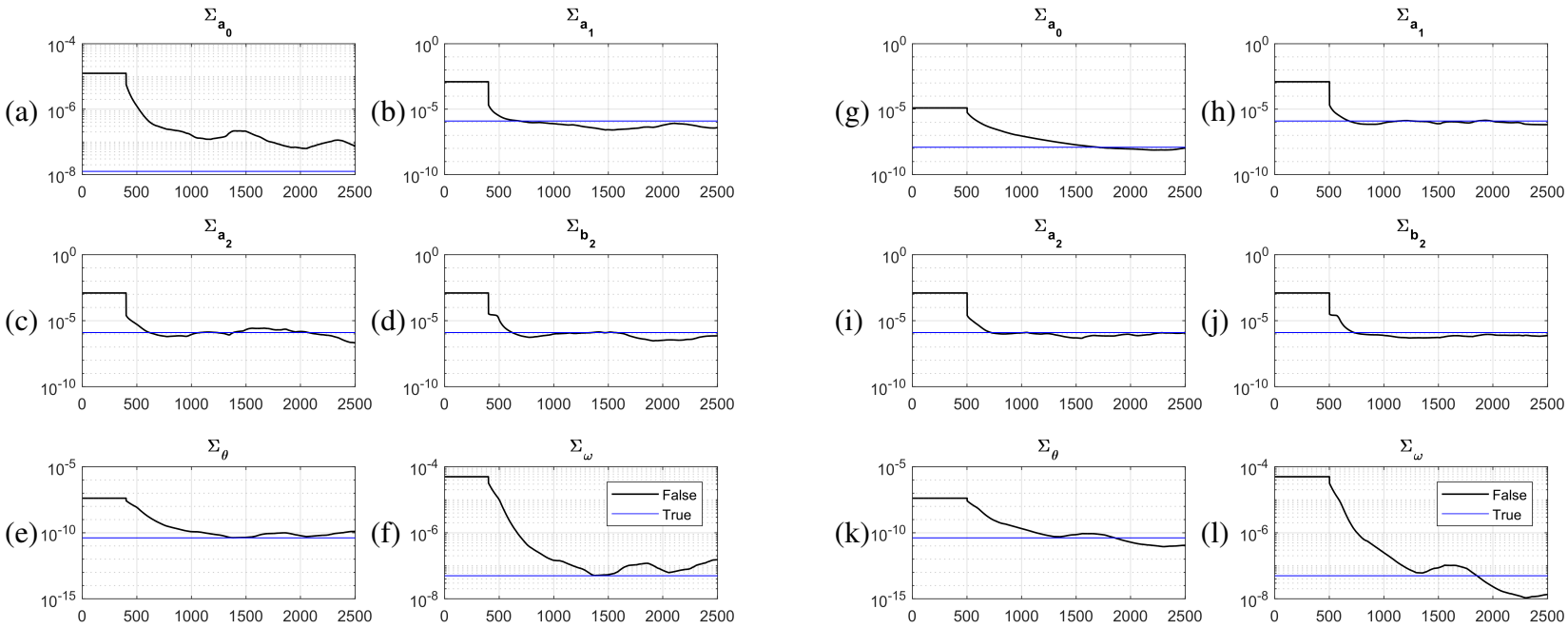

Figure 5.11: Simulation results of variance estimates for overestimating the process noise with $\epsilon = 10^{-6}$, N=400 (left) and N=500 (right)

5.3 Real Data

In this section, we will evaluate the combined performance of ExKF and its tuning on the respiratory records of mechanically ventilated patients. The first pattern of respiratory motion record is the same as in section 4.3.2 of the previous chapter. We will restate the choice for ω_0 and θ_0 for the sake of clarity. "Breathing frequency is $\omega_0 = \frac{10\pi}{60}$ (for 1/5 breaths per second) with high confidence ($\Sigma_{0,\omega} = 10^{-4}$) as the patient is mechanically ventilated with 12 breadths per minute. Initial angle estimate is $\theta_0 = 0$ with a confidence level that bounds it between $-\pi$ and π ($\Sigma_{0,\omega} = (\frac{\pi}{3})^2$)."

Our initial trials started by changing the number of data points used in the EM algorithm. Unfortunately, no matter what the order of the model used was, we observed that a training window less than N=4 breathing cycles gave divergent results for overestimating the process noise 10000 times the manually tuned one. That is why, we present the real data run starting with N=4 cycles.

For a second order Fourier series model $m = 2$, the convergence results can be seen in Fig. 5.12 on page 109. As can be seen in Fig. 5.12.b, expected value of one of the highest order Fourier series coefficients, in this case a_2, does not show a random walk behaviour bearing the burden of finite order approximation. What is worse is that the EM algorithm seems to favor this non-Gaussian behaviour. This can be seen from the plots of expected value of a_2 and its variance in Fig. 5.12.b and 5.12.h, respectively. EM first runs at around N_1=400 corresponding to t=40 seconds. Especially after this instant, a_2 adapts to the high frequency content of the data which can be seen in Fig. 5.12.b. That is, its estimated value captures the positive peaks observed at the output in Fig. 5.12.m when the mode of respiration changes from exhale to inhale. Moreover, changes in a_2 do not behave like a Gaussian random walk model due to the periodicity of its estimated value. Also, the information carried in this coefficient is corrupted because a_2 does not converge to its true value of 0.45 which we will when we increase the model order. Estimated output obtained by ignoring this corrupted coefficient can be seen in Fig. 5.12.m with an RMSE of 0.053. Large errors seen in the estimated output indicate that a second order model fails to track respiratory motion. On the computational performance, EM takes approximately 16000 iterations to fit this model at an approximate time cost of 390 seconds. This huge cost of

computational burden makes sense as the EM tries to find a better alternative while the assumed model cannot converge to the underlying one.

Increasing model order to $m = 3$ gives the results given in Fig. 5.13 on page 110. Compared to the previous case, a_2 fits the assumed model with a value of 0.45. This time, it is the third order coefficient b_3 that does not fit the assumed random walk model, taking the cost of a finite order model approximation. This can be seen in Fig. 5.13.g. Information captured by this coefficient is corrupted just like the information captured by a_2 is with a model order $m = 2$. Estimated output obtained by excluding b_3 can be seen in Fig. 5.13.q with an RMSE of 0.021. When one compares the estimated output in Fig. 5.13.q for $m = 3$ to that of $m = 2$ in Fig. 5.12.m, estimated output is found to approach the measured one with increased model order. Also, the number of EM iterations drops significantly to a number of approximately 7900 iterations calculated in about 220 seconds. This performance increase in both the computational performance as well as the improved output estimation accuracy is an indication of approaching the true model and verifying it with actual data.

Setting $m = 4$ and $m = 5$ yield similar results, with one of the highest order coefficients overfitting the high frequency content of measurements. The case for $m = 5$ is given in Fig. 5.14 on page 111. We are skipping the plot for $m = 4$ for the sake of not spreading this section as it gives similar results as for $m = 5$. We are only reporting $m = 5$ as it is the highest order model that verifies the model based on measurements with the lowest computational time of 161 seconds, which makes this model good for real time tracking as the computational cost associated with this is less than the data duration which is 200 seconds long. The magnitudes of coefficient estimates a_5 and b_5 after convergence are about 0.1 and 0.02 as can be seen in Fig. 5.14.f and 5.14.j, respectively. Having $\sqrt{a_5^2 + b_5^2} \cong 0.102$, their magnitude is much smaller than the amplitude of measured respiratory motion, which is approximately 3, and ignoring these coefficients in the output estimation adds little error to output estimation error. Estimating the output by ignoring these coefficients results in a maximum error of 0.25 mm at the instants where mode of respiration changes from inhale to exhale (which are the sharp tips observed at positive values of the measured signal constituting a small portion of the breathing cycle). Apart from this, the estimated signal fits fairly well the measured one with an RMSE of 0.016.

Increasing m further to 6 gives the result in Fig. 5.15 on page 112. Non random walk behaviour is observed in a_5 with a_6 convergent to zero, b_5 and b_6 finally converging to zero. Increasing m to 6 did not improve the model as it is again the 5^{th} order coefficient that accounts for the high frequency content. Total runtime has also increased to approximately 190, thanks to the added complexity with no beneficial reason (output RMSE is still at 0.016 as in the case with $m = 5$) despite the decrease in the total number of EM iterations to approximately 4600.

As a final remark on this case, computational costs associated with each approximation order are given in Table 5.3. Average EM and average update time are increasing with increasing model order, and average update time is affected more severely than the average EM time from this increase. Number of EM iterations, on the other hand, are decreasing. What dominates the total runtime is the duration of EM algorithm: it is roughly 1000 times more costly than average ExKF update time. That is why, decrease in total runtime should be explained by the number of EM iterations and its average evaluation time: percentage decrease in number of EM iterations are larger than percentage increase of average EM time when we gradually increase m up to 5, which does not hold when m is increased further.

Table 5.3: Filtering of Real Data - Total Number of EM Iterations, Total Runtime and Output RMSE vs. Model Order for a Training Window Covering 4 Breathing Cycles

Order m	# of EM Iterations	Total Runtime (sec)	Average Update Time (μsec)	Average EM Time (msec)	Output RMSE (mm)
2	15754	392	39	25	0.0531
3	7983	215	46	27	0.0211
4	5200	170	49	30	0.0171
5	4783	161	53	31	0.0160
6	4679	190	59	34	0.0160
7	4341	186	65	37	0.0163

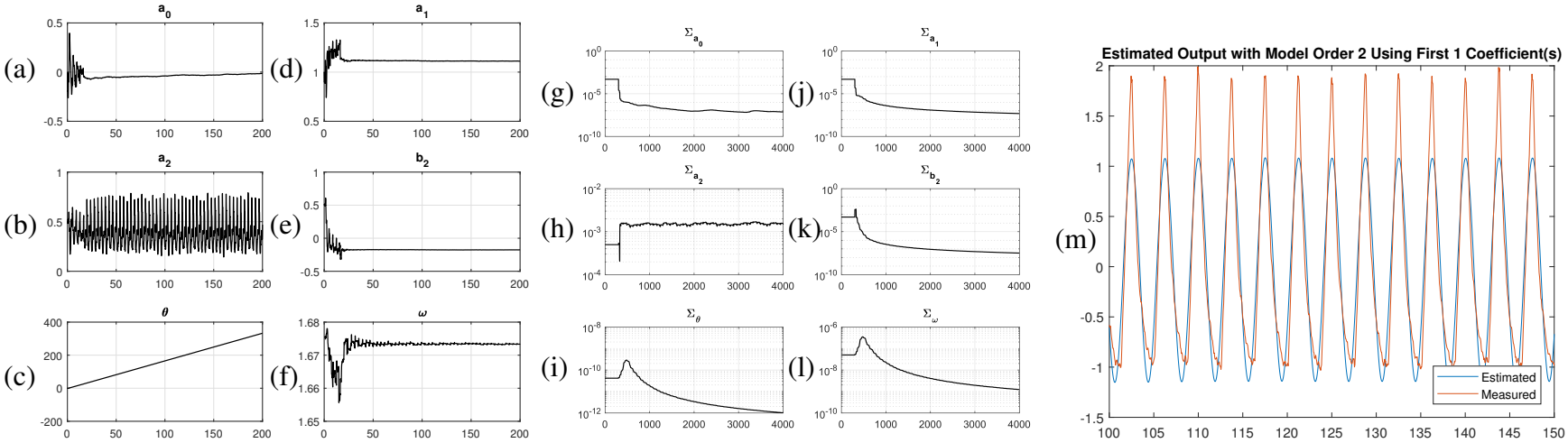

Figure 5.12: Estimated coefficients (top left), estimated variances (top right) and estimated output using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=2 and N=4 breathing cycles

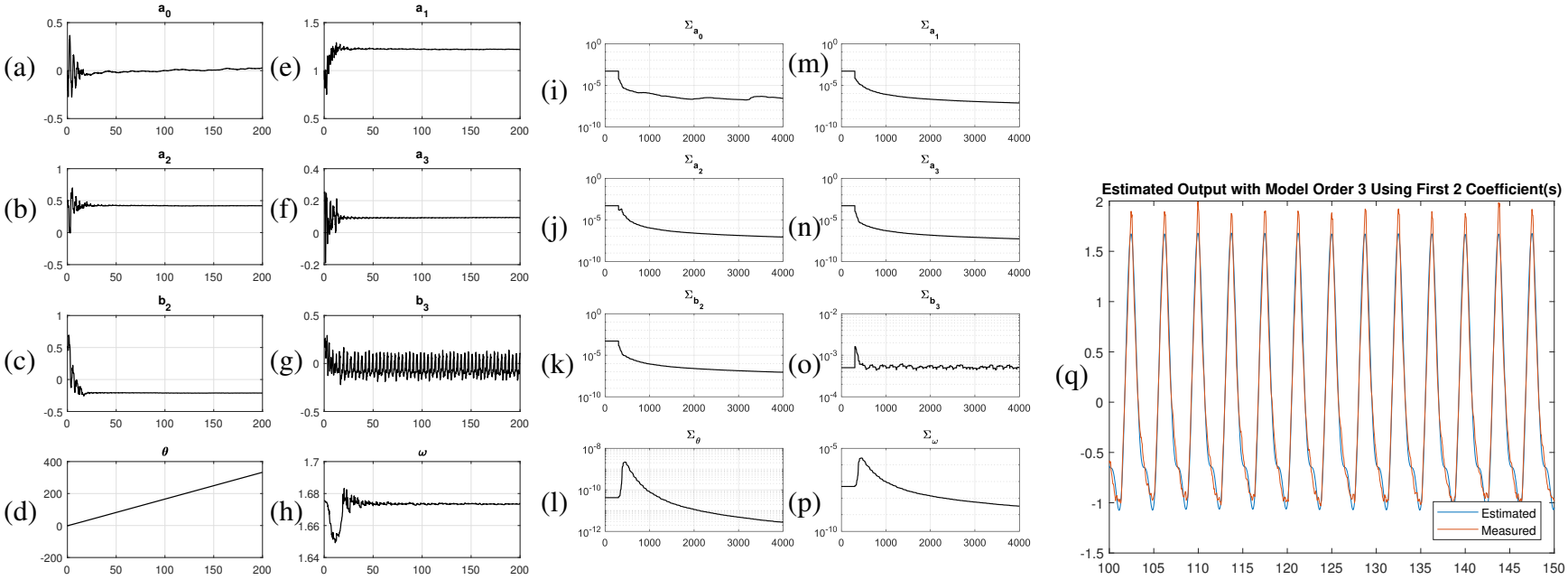

Figure 5.13: Estimated coefficients (left), estimated variances (center) and estimated output (right) using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=3 and N=4 breathing cycles

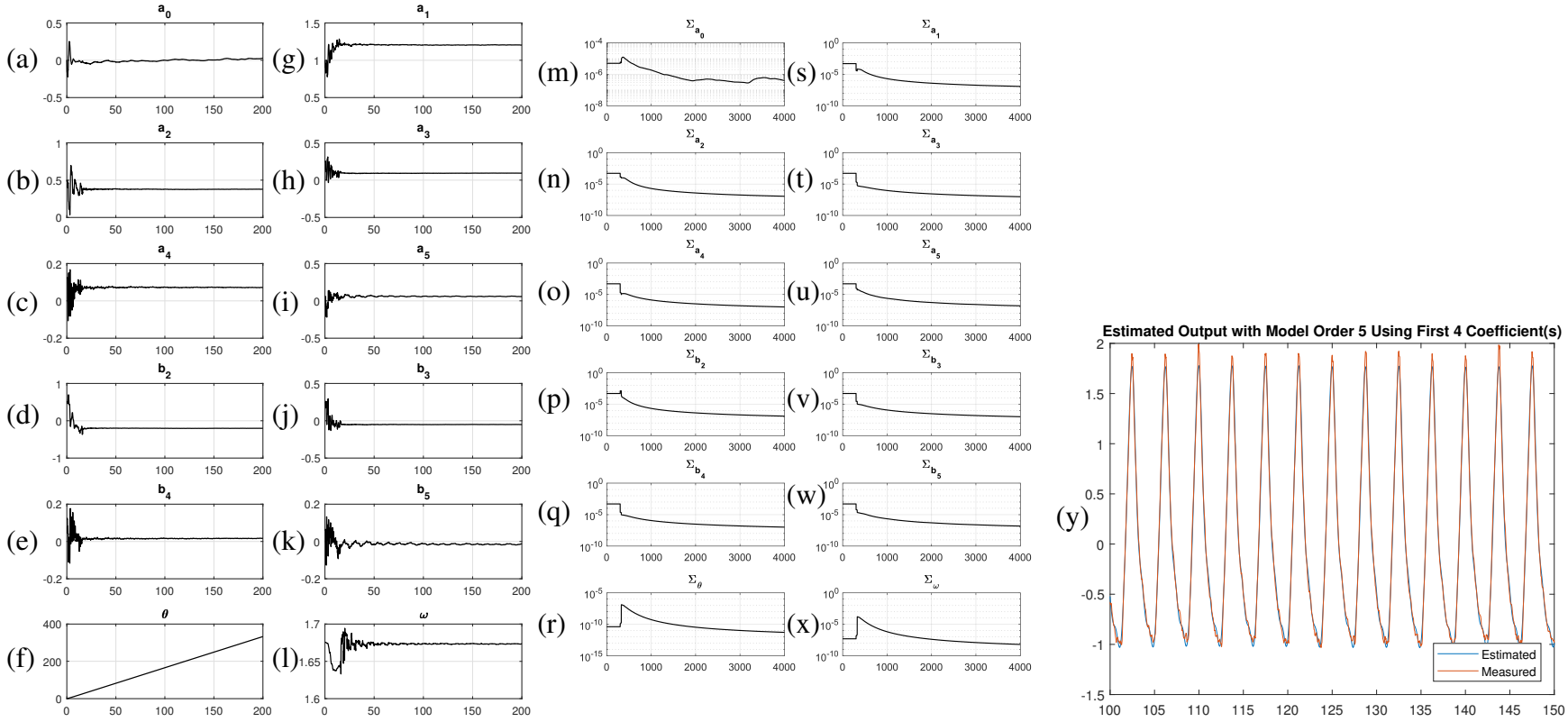

Figure 5.14: Estimated coefficients (left), estimated variances (center) and estimated output (right) using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=5 and N=4 breathing cycles

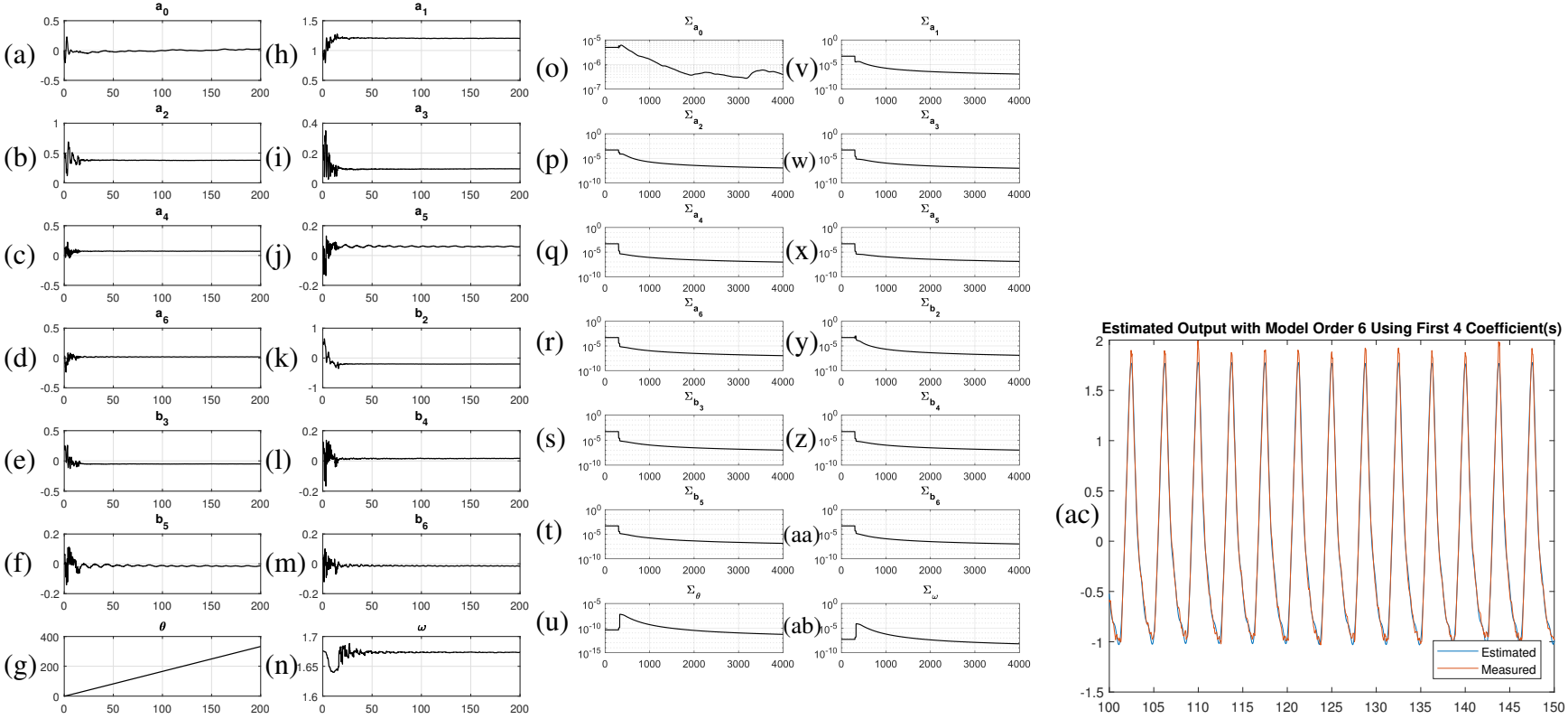

Figure 5.15: Estimated coefficients (left), estimated variances (center) and estimated output (right) using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=6 and N=4 breathing cycles

Increasing the number of data points used in the EM algorithm to cover 5 breathing cycles instead of 4 gives similar results to the previous case. The fifth order coefficient still does not behave in accordance with the assumed model, which can be seen in Fig. 5.16.k and Fig. 5.17.f for $m = 5$ and $m = 6$ with an output RMSE of 0.0159 and 0.0158, respectively. There is again no improvement in increasing m to 6; high frequency content is still captured by the fifth order coefficient while the average EM and ExKF update cost increased from 41 to 45 msec and from 63 to 60 μsec, respectively.

As for the computation complexity, computational costs associated with each approximation order are given in Table 5.4. In accordance with the case when the number of data points covers 4 breathing cycles, average update and EM times are again increasing with increasing m while the total number of EM iterations are decreasing. Different from the case of covering 4 breathing cycles, this time least amount of total runtime is for the model with $m = 4$. However, as the the fifth order model adds to the accuracy of output estimation with non-zero fourth order coefficients ($a_4 \cong 0.08$ in Fig. 5.4.c and $b_4 \cong 0$ in Fig. 5.4.e), the slight increase in computational requirements is acceptable. As a further justification of this acceptability, average EM time which dominates the total computational cost (41 msec) is less than the data arrival period which is 50 msec.

Table 5.4: Filtering of Real Data - Total Number of EM Iterations and Total Runtime vs. Model Order for a Training Window Covering 5 Breathing Cycles

Order m	# of EM Iterations	Total Runtime Time (sec)	Average Update Time (μsec)	Average EM Time (msec)
2	13638	398	40	29
3	7957	247	44	31
4	4301	162	47	36
5	3914	177	53	41
6	3881	201	60	45
7	3848	197	66	48

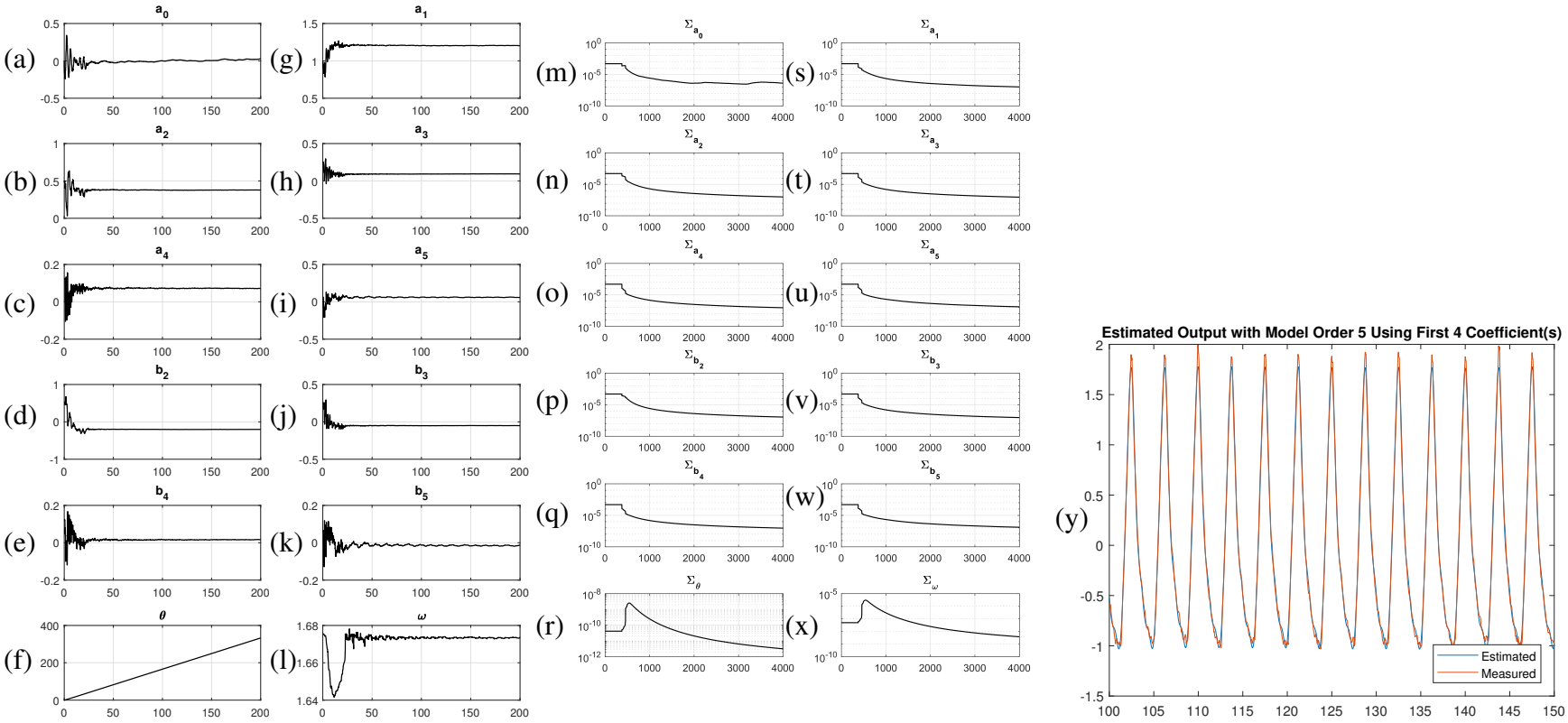

Figure 5.16: Estimated coefficients (left), estimated variances (center) and estimated output (right) using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=5 and N=5 breathing cycles

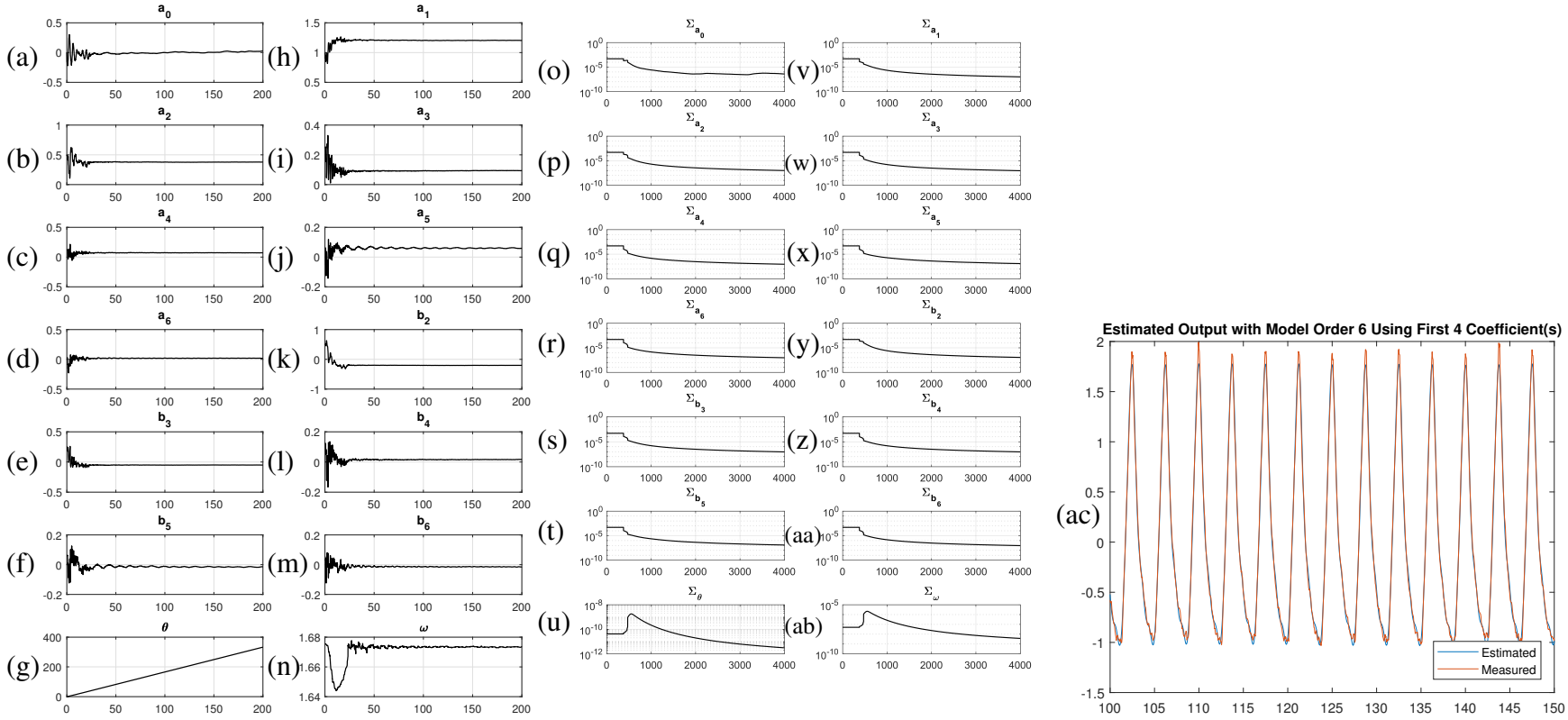

Figure 5.17: Estimated coefficients (left), estimated variances (center) and estimated output (right) using coefficients up to the order where coefficients fail to show a random walk behaviour for a model order m=6 and N=5 breathing cycles

Next cases will be devoted to patients with more irregular breathing patterns as for patient 2. Results for the second patient can be seen in Fig. 5.18. If one refers to the measured output given in Fig. 5.18.z, where the measured output is magnified between $t = 80$ and $t = 100$, it is apparent that this respiratory motion shows a DC shift. Estimated value of a_0 in the Fig. 5.18.a does not show a Gaussian random walk behaviour. To see whether this behavior is in agreement with the measured signal, a closer examination of the results is needed. The baseline shift at the output in Fig. 5.18.z starts around $t \cong 85$ seconds. This leads the EM algorithm to an increase of the variance associated with a_0 around N $\cong 1800$ (Fig. 5.18.m), which enables the estimated value of a_0 to increase its value around $t \cong 90$ seconds (Fig. 5.18.a). This increase makes sense when one compares the plots given in Fig. 5.18.y and 5.18.z, where the plot in Fig. 5.18.z shows the estimated output obtained by ignoring a_0 failing to capture this DC shift. This DC shift also hits the estimated value of ω as the periodicity of the signal is violated, and this can be observed by the increase in Σ_ω estimate in Fig. 5.18.z around N $\cong 1800$. However, thanks to adaptation of model adaptation by EM, ω can gradually approach its steady state value as can be seen in Fig. 5.18.x. Again, the behaviour of the fifth order coefficients is not Gaussian as can be seen in 5.18.i and 5.18.k, just like the case where the breathing pattern was regular as in patient 1.

As for its computational burden, the EM algorithm takes a total of 4467 iterations, each with average EM time of 32 msec and each update took 53 μsec. The total amount of runtime was 145 sec with an estimated output RMSE of 0.0203.

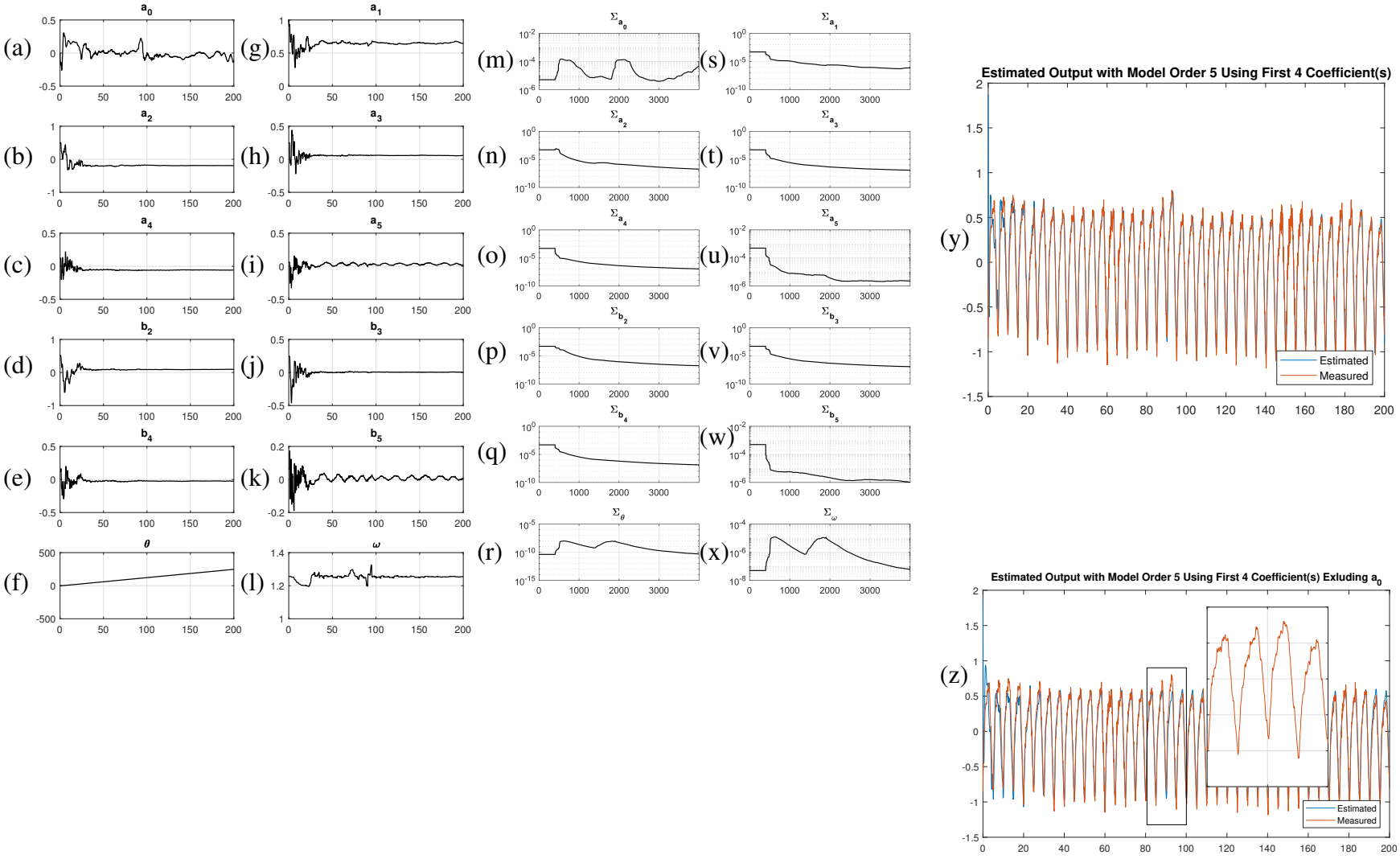

Figure 5.18: Patient 2, estimated coefficients (left), estimated variances (center) and estimated output using coefficients up to 4th order including a_0 (right, top) and exluding a_0 (right, bottom) for a model order m=5 and N=4 breathing cycles

One last respiratory motion record can be seen in Fig. 5.19 from the third patient with the most irregular breathing patterns showing DC level and amplitude shifts. Convergence towards steady state variance values cannot be achieved by most of the parameters (see 5.19.m, 5.19.s and 5.19.w). Increased value of a_0 around t=75, 110 and 150 in Fig. 5.19.a is in agreement with the DC shift in the measured respiratory motion when one examines the measured output at around t=75, 110 and 150 in Fig. 5.19.y, just like in the previous patient where only the DC shifts were observed. The output estimation performance is also in agreement with the measurements as can be seen in Fig. 5.19.y. However, most of the coefficients and ω do not follow a random walk behaviour if one examines the plots in Fig. 5.19 from g to l: what this indicates is that the chosen model is a bad fit for this breath pattern. Respiratory motion with high irregularity cannot be modeled with our model. As a side indicator of this: the EM algorithm takes approximately 10500 iterations, a number comparable to that we get when greatly underestimating the model order for a regular breathing pattern with $m = 2$. That is, number of EM iterations increases greatly when there is a mismatch between the assumed model and the true one. What is worse is that the processing of data corresponding to a duration of 200 seconds took approximately 480 seconds, a clear violation of real time tracking requirements.

In this section, we have evaluated the performance of our filter with the data collected from three different patients: one with a regular breathing pattern, one with a breathing pattern with DC shifts and one with amplitude and DC shifts. Our results indicate that:

- Automatic tuning of process noise covariance matrix is easily achievable when breathing is regular as it is the case for the first patient. The only problem associated with this was the selection of model order. We have experimentally decided that $m = 5$ was the best choice.
- Our model can handle the DC shifts in the output as was demonstrated with the data from the second patient even though random walk behaviour is lost for a_0. The choice of $m = 5$ seems to be sufficient for this respiratory motion trace as well.
- Data with frequent irregularities cannot be modeled with our assumed model

which is clearly demonstrated in the base of patient 3 with DC shifts and amplitude changes. Although the output is seemingly captured, the violation of Gaussian innovation in our estimated variables clearly indicate this unfitness.

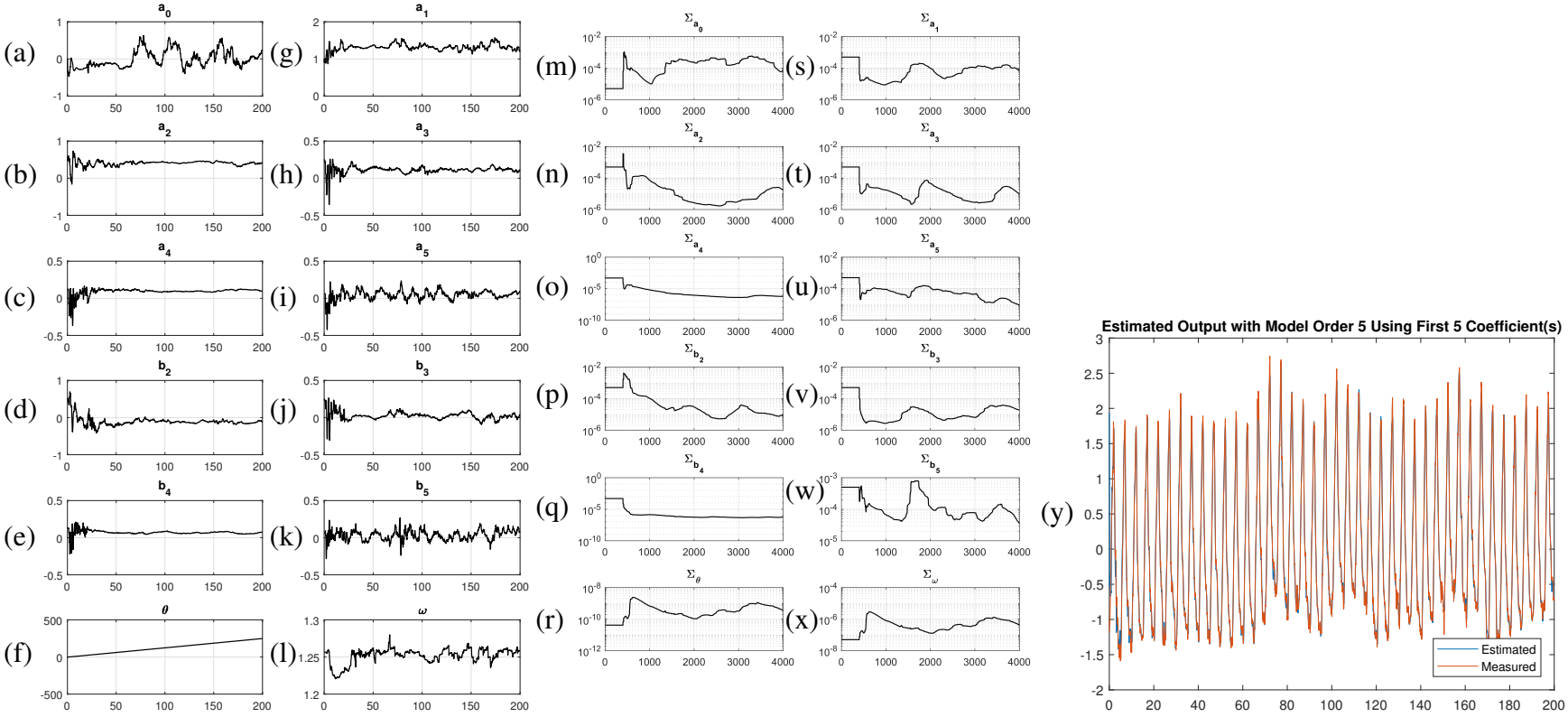

Figure 5.19: Patient 3, estimated coefficients (left), estimated variances (center) and estimated output using coefficients up to 5th order (right) for a model order m=5 and N=4 breathing cycles

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, two traditional non-linear Kalman filters, EKF2 and UKF, are compared against a newly developed ExKF called Exact Moment Matching Kalman Filter for tracking respiratory motion traces. Their performances are first compared in terms of state convergence speeds using artificially generated data, ExKF has been demonstrated to have the fastest convergence especially when the initial estimate in θ is varied thanks to the theoretically derived mean and covariance calculations that do not rely on approximations. Respiratory motion records were collected using NDI Aurora electromagnetic tracking system from mechanically ventilated patients undergoing laparoscopic surgeries. Convergence effect was emphasized when we used the real respiratory motion data, and we increased the model order to $m = 5$. Increased degree of nonlinearity causes EKF2 and UKF to frequently diverge, while ExKF can achieve convergence when the others failed.

EM based maximum likelihood estimation of process noise covariance is also evaluated. It is shown through simulation that variances associated with Fourier series coefficients can be brought to an order of magnitude close to the true ones. When applied to real data, variances high order coefficients adapt to unmodeled high frequency content; this case is especially pronounced when the model order is severely underestimated with $m = 2$ and $m = 3$. Using such observations as well as the associated computational costs, the choice of $m = 5$ seems to be the best choice for respiratory motion tracking. Moving on to more irregular patterns of breathing, our model can handle and adapt to the level shifts observed at the output. However, with frequent changes in amplitude and baseline shifts baseline shifts, real time requirements of the

tracking algorithm is violated.

The studies we have come across to so far had only focused on the output estimation error as the evaluation metric of the proposed state space model, without paying much attention to the fitness of the underlying model. Although the fitness can be acceptable for the breathing regular breathing patterns and breathing patterns with level shifts, the real time tracking ability of our method and the Gaussianity assumption are clearly violated for patterns with frequent changes in amplitude on top of level shifts. In conclusion, the proposed method is able to track respiratory motion traces of mechanically ventilated patients but its computational cost needs to be improved as will be discussed in the next section.

6.1 Future Work

We have implemented an online EM algorithm that updates the process noise covariance whenever a new measurement arrives. However, we still impose multiple iterations by rerunning the EM until convergence is achieved. Although that enables us to track regular and irregular breathing patterns with mild irregularities, real time tracking of signals with high irregularities is impractical. Instead, one can try using an online EM with forgetting factor to guarantee that EM is run only once at the arrival of a new measurement in order to speed up the computations.

We have demonstrated that the effect of initial estimate in θ is crucial for the convergence properties of the filters. In fact, when this angle is known, all three filters have equivalent convergence speeds as the problem reduces to the estimation of states of a linear time varying system. As an alternative to using a nonlinear KF, one can try to tackle this tracking problem with a particle filter. However, it is quite likely that a large number of particles would be needed even with $m = 5$. A marginalized particle filter can also be used to marginalize out the angle (θ) dependence of our measurement model, after which the measurement model becomes equivalent to a linear time varying equation of the remaining states. This method seems to be a possible candidate with promising improvements over ExKF that we have introduced.

As a final remark, biological systems are complex in the sense that a simple input-

output relationship may not be obtained. However, recording and evaluating some vital signs such as heartbeat rate and blood pressure may be of benefit for respiratory motion tracking, both of which are available in a clinical environment. These vital signs are all effective in respiratory motion patterns of spontaneously breathing patients. However, as far as we know, their effect on respiratory motion patterns of mechanically ventilated patients have not been studied so far. They may initiate the irregularities under mechanical ventilation and we may use such an information to check whether our method may be used for tracking the respiratory motion. Even if such a relationship may not be found, these signs may indicate that effect of anesthetic agents has worn off, and this wearing off may initiate sudden breathing irregularities as the mechanical ventilation is replaced with the spontaneous one. If such a relationship can be found, it would improve the surgical safety as the operation can be halted at those times.

REFERENCES

[1] C. N. Riviere, R. S. Rader, and N. V. Thakor. Adaptive canceling of physiological tremor for improved precision in microsurgery. *IEEE Transactions on Biomedical Engineering*, 45(7):839–846, 1998.

[2] Floris Ernst. *Compensating for quasi-periodic motion in robotic radiosurgery*, volume 9781461419. Springer, New York, NY, 2012.

[3] Gerard J. Tortora and Bryan H. Derrickson. *Principles of anatomy and physiology*. New York ; Chichester : Wiley, 2006., New York, 11 edition, 2006.

[4] Kai Rehder and H. Michael Marsh. Respiratory Mechanics During Anesthesia and Mechanical Ventilation. *Handbook of Physiology, The Respiratory System, Mechanics of Breathing*, 2011.

[5] Kenneth H Wong, Sonja Dieterich, Jonathan Tang, and Kevin Cleary. Quantitative measurement of CyberKnife robotic arm steering. *Technology in Cancer Research and Treatment*, 6(6):589–594, 2007.

[6] Suk Jin Lee and Yuichi Motai. *Prediction and Classification of Respiratory Motion*, volume 525. 2014.

[7] Daniel A. Low, Parag J. Parikh, Wei Lu, James F. Dempsey, Sasha H. Wahab, James P. Hubenschmidt, Michelle M. Nystrom, Maureen Handoko, and Jeffrey D. Bradley. Novel breathing motion model for radiotherapy. *International Journal of Radiation Oncology Biology Physics*, 63(3):921–929, nov 2005.

[8] EA Wan and R Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *... 2000. AS-SPCC. The IEEE 2000*, 2000.

[9] Matthew Rhudy, Yu Gu, and Marcello R. Napolitano. An analytical approach for comparing linearization methods in EKF and UKF: Regular paper. *International Journal of Advanced Robotic Systems*, 10:1–9, 2013.

[10] Ryan Turner and Carl Edward Rasmussen. Model based learning of sigma points in unscented Kalman filtering. *IEEE International Workshop on Machine Learning for Signal Processing*, pages 178–183, 2010.

[11] Northern Digital Inc., Waterloo, Ontario, Canada.

[12] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer International Publishing, Cham, 2 edition, 2016.

[13] Cameron N. Riviere, Jacques Gangloff, and Michel De Mathelin. Robotic compensation of biological motion to enhance surgical accuracy. *Proceedings of the IEEE*, 94(9):1705–1715, 2006.

[14] Gyanendra R Sharma, Pankaj N Maheshwari, Anshu G Sharma, Reeta P Maheshwari, Ritwik S Heda, and Sakshi P Maheshwari. Fluoroscopy guided percutaneous renal access in prone position. *World journal of clinical cases*, 3(3):245–64, mar 2015.

[15] C. N. Riviere, A. Thakral, I. I. Iordachita, G. Mitroi, and D. Stoianovici. Predicting respiratory motion for active canceling during percutaneous needle insertion. *Annual Reports of the Research Reactor Institute, Kyoto University*, 4:3477–3480, 2001.

[16] L Ramrath, A Schlaefer, F Ernst, S Dieterich2, and A Schweikard. Prediction of respiratory motion with a multi-frequency based Extended Kalman Filter. In *Proceedings of the 21st International Conference and Exhibition on Computer Assisted Radiology and Surgery*, pages 56–58, 2007.

[17] Manivannan Sundarapandian, Ramakrishnan Kalpathi, and R. Alfredo Siochi. Respiratory motion prediction from CBCT image observations using UKF. In *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, volume 1, pages 559–562, 2015.

[18] Sunghan Kim and James McNames. Detecting and tracking tremor in spike trains using the rectangular model based extended Kalman smoother. *Journal of Neuroscience Methods*, 188(1):97–104, 2010.

[19] Fredrik Gustafsson and Gustaf Hendeby. Some relations between extended and unscented Kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2012.

[20] Paolo Pelosi, Claudia Brusasco, and Marcelo Gama de Abreu. Mechanical Ventilation during General Anesthesia. In *Principles and Practice of Mechanical Ventilation*, pages 597–627. McGraw-Hill Education, 2013.

[21] Robert Dürichen. *From Univariate to Multivariate Respiratory Motion Compensation*. PhD thesis, Universitat zu Lübeck, 2014.

[22] Mathew D. Sorensen, Michael R. Bailey, Anup R. Shah, Ryan S. Hsi, Marla Paun, and Jonathan D. Harper. Quantitative Assessment of Shockwave Lithotripsy Accuracy and the Effect of Respiratory Motion [*]. *Journal of Endourology*, 26(8):1070–1074, 2012.

[23] D. Pham, T. Kron, F. Foroudi, M. Schneider, and S. Siva. A Review of Kidney Motion under Free, Deep and Forced-Shallow Breathing Conditions: Implications for Stereotactic Ablative Body Radiotherapy Treatment. *Technology in Cancer Research & Treatment*, 13(4):315–323, 2014.

[24] S. C. Davies, A. L. Hill, R. B. Holmes, M. Halliwell, and P. C. Jackson. Ultrasound quantitation of respiratory organ motion in the upper abdomen. *British Journal of Radiology*, 67(803):1096–1102, 1994.

[25] Hideomi Yamashita, Mami Yamashita, Masahiko Futaguchi, Ryousuke Takenaka, Shino Shibata, Kentaro Yamamoto, Akihiro Nomoto, Akira Sakumi, Satoshi Kida, Yoshihiro Kaneko, Shigeharu Takenaka, Takashi Shiraki, and Keiichi Nakagawa. Individually wide range of renal motion evaluated by four-dimensional computed tomography. *SpringerPlus*, 3(1):1–7, 2014.

[26] NA. QUASAR™ Respiratory Motion Phantom.

[27] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 53. 2013.

[28] S. S. Vedam, P. J. Keall, A. Docef, D. A. Todor, V. R. Kini, and R. Mohan. Predicting respiratory motion for four-dimensional radiotherapy. *Medical Physics*, 31(8):2274–2283, jul 2004.

[29] Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, Ltd, Chichester, UK, oct 2001.

[30] Philip J. Parker and Brian D O Anderson. Frequency tracking of nonsinusoidal periodic signals in noise. *Signal Processing*, 20(2):127–152, 1990.

[31] Jouni Hartikainen and Simo Särkkä. Optimal filtering with Kalman filters and smoother - Manual for Matlab toolbox EKF/UKF, 2008.

[32] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using Matlab*. John Wiley & Sons, Inc., Hoboken, NJ, USA, nov 2014.

[33] Zhe Chen. Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. *Statistics*, 182(1):1–69, 2003.

[34] Michael Roth, Gustaf Hendeby, and Fredrik Gustafsson. Nonlinear Kalman Filters Explained : A Tutorial on Moment Computations and Sigma Point Methods. *Journal of Advances in Information Fusion*, pages 1–21, 2016.

[35] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, mar 1960.

[36] John L. Crassidis and John L. Junkins. *Optimal estimation of dynamic systems*. CRC Press, 2012.

[37] Ian Reid. Estimation II, 2001.

[38] B.F. La Scala and R.R. Bitmead. Design of an extended Kalman filter frequency tracker. *IEEE Transactions on Signal Processing*, 44(3):739–742, 2000.

[39] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. Adaptive Adjustment of Noise Covariance in Kalman Filter for Dynamic State Estimation. *CoRR*, 2017.

[40] Simon J. Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. *Upublished*, pages 1–27, 1996.

[41] Simon J Julier and Jeffrey K Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. *Spie 3068*, pages 182–193, 1997.

[42] Simon Julier, Jeffrey Uhlmann, and H Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filter and estimators. *IEEE Transactions on Autonatic Control*, 45(3):447–482, 2000.

[43] Mihai Bogdan Luca, Stéphane Azou, Gilles Burel, and Alexandru Serbanescu. On exact Kalman filtering of polynomial systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(6):1329–1340, 2006.

[44] S Saha, P K Mandal, Y Boers, H Driessen, and A Bagchi. Gaussian proposal density using moment matching in SMC methods. *Stat Comput*, 19:203–208, 2009.

[45] Hogg Craig. *Introduction to Mathematical Statistics*. Pearson, 7 edition, 2012.

[46] Raman K. Mehra. Approaches to Adaptive Filtering. *IEEE Transactions on Automatic Control*, 17(5):693–698, 1972.

[47] Shyam Mohan M, Naren Naik, R. M. O. Gemson, and M. R. Ananthasayanam. Introduction to the Kalman Filter and Tuning its Statistics for Near Optimal Estimates and Cramer Rao Bound. Technical Report February, Dept. of Electrical Engineering, IIT Kanpur, India, 2015.

[48] Patrik Axelsson, Umut Orguner, Fredrik Gustafsson, and Mikael Norrlöf. ML Estimation of Process Noise Variance in Dynamic Systems. Technical report, Department of Electrical Engineering, Linköping University, 2010.

[49] A N Iusem. On the convergence properties of the projected gradient method for convex optimization. *Computational & Applied Mathematics*, 22(1):37–52, 2003.

[50] Floris Ernst, Alexander Schlaefer, and Achim Schweikard. Prediction of Respiratory Motion with Wavelet-Based Multiscale Autoregression. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, number Pt 2, pages 668–675, 2007.

[51] Robert Piché. Online tests of Kalman filter consistency. *International Journal of Adaptive Control and Signal Processing*, 30(1):115–124, 2016.

[52] Elmed Electronics & Medical Industry & Trade Incorporated.

[53] Andreas Krauss, Simeon Nill, and U. Oelfke. The comparative performance of four respiratory motion predictors for real-time tumour tracking. *Physics in Medicine and Biology*, 56(16):5303–5317, 2011.

[54] Floris Ernst, Alexander Schlaefer, and Achim Schweikard. Smoothing of respiratory motion traces for motion-compensated radiotherapy. *Medical Physics*, 37(1):282–294, 2010.

[55] A. De Troyer, S. Kelly, P. T. Macklem, and W. A. Zin. Mechanics of intercostal space and actions of external and internal intercostal muscles. *Journal of Clinical Investigation*, 75(3):850–857, 1985.

Appendix A

EFFECTS OF INTERCOSTAL MUSCLES ON BREATHING

Intercostal muscles have a peculiar way of assisting respiratory motion and will be discussed here in some detail. Work presented here is based on what is so called as Hamberger's law [55]. Consider the diagram shown in Fig. A.1. Blue ellipses represent two individual ribs, and intercostal muscles are shown as red lines. Their orientation in space is as shown and the orientation is important for their functionality in pulling the rib cage up or down. Spine acts as a pivot point for both inhalation and exhalation while sternum moves freely with the ribs. To simplify the problem, right hand side of the figure can be used with a single intercostal muscle represented in 1D. When an external intercostal is contracted for inspiration, it pulls both ribs closer to each other with equal force, f. This creates torques on both ribs, given as $\tau_1 = -f \times r_1$ and $\tau_2 = f \times r_2$. Since the lower part has a longer lever arm, its torque is greater in magnitude. Net overall torque for this case is pointing outwards ($\odot$) and the resulting angular movement is in counter clockwise direction with a simple application of right hand rule. Thus, the ribs should move upwards. The opposite case occurs for internal intercostal muscle contraction with τ_2 being smaller than τ_1.

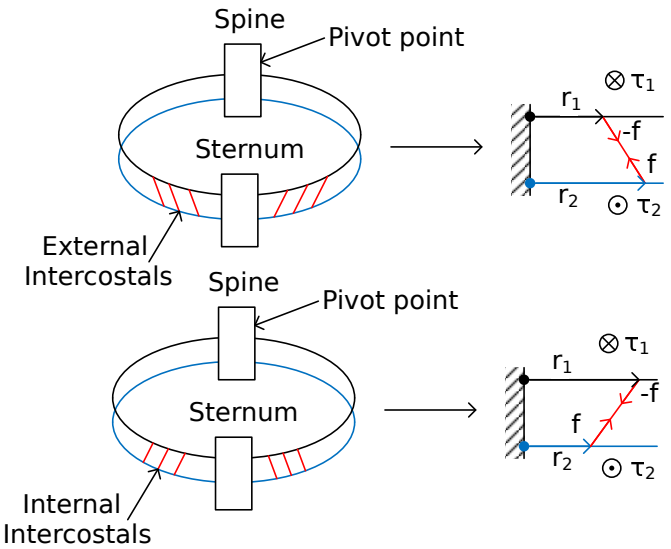

Figure A.1: External and internal intercostal muscles