HEART MOTION PREDICTION BASED ON ADAPTIVE ESTIMATION ALGORITHMS FOR ROBOTIC-ASSISTED BEATING HEART SURGERY

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Eser Erdem Tuna

September, 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Cenk Çavuşoğlu (Co-Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Hitay Özbay (Co-Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ömer Morgül

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Uluç Saranlı

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School of Engineering and Science

ABSTRACT

HEART MOTION PREDICTION BASED ON ADAPTIVE ESTIMATION ALGORITHMS FOR ROBOTIC-ASSISTED BEATING HEART SURGERY

Eser Erdem Tuna
M.S. in Electrical and Electronics Engineering
Supervisors: Assoc. Prof. Dr. Cenk Çavuşoğlu and Prof. Dr. Hitay Özbay
September, 2011

Robotic assisted beating heart surgery aims to allow surgeons to operate on a beating heart without stabilizers as if the heart is stationary. The robot actively cancels heart motion by closely following a point of interest (POI) on the heart surface—a process called Active Relative Motion Canceling (ARMC). Due to the high bandwidth of the POI motion, it is necessary to supply the controller with an estimate of the immediate future of the POI motion over a prediction horizon in order to achieve sufficient tracking accuracy. In this thesis two prediction algorithms, using an adaptive filter to generate future position estimates, are studied. In addition, the variation in heart rate on tracking performance is studied and the prediction algorithms are evaluated using a 3 degrees of freedom test-bed with prerecorded heart motion data.

Besides this, a probabilistic robotics approach is followed to model and characterize noise of the sensor system that collects heart motion data used in this study. The generated model is employed to filter and clean the noisy measurements collected from the sensor system. Then, the filtered sensor data is used to localize POI on the heart surface accurately. Finally, estimates obtained from the adaptive prediction algorithms are integrated to the generated measurement model with the aim of improving the performance of the presented approach.

Keywords: Active relative motion canceling, signal estimation, medical robotics, surgical robotics, probabilistic robotics.

ÖZET

ROBOTİK-DESTEKLİ KALP AMELİYATLARI İÇİN UYABİLEN TAHMİN ALGORİTMALARINA DAYALI KALP HAREKETİ TAHMİNİ

Eser Erdem Tuna
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticileri: Doç. Dr. Cenk Çavuşoğlu ve Prof. Dr. Hitay Özbay
Eylül, 2011

Robotik destekli atan kalp ameliyatı, cerrahlara atan kalp üzerinde dengeliyiciler olmadan, kalp sabitmişçesine çalşmaları için olanak sağlamaktadır. Robot, kalp yüzeyindeki bir ilgi noktasını etkin bir biçimde, yakından takip ederek kalp hareketini iptal eder. Bu yönteme "Etkin Göreceli Hareket Önleyici (EGHÖ)" denilmektedir. İlgi noktasının yüksek bant genişliğindeki hareketi nedeniyle, yeterli takip doğruluğunu sağlamak için, denetleyiciye ilgi noktasının hareketinin bir tahmin ufku boyunca yakın bir tahminini sağlamak gerekmektedir. Bu tezde, gelecekteki konum tahminini oluşturmak için uyabilen süzgeç kullanan iki tahmin algoritması çalışılmıştır. Buna ek olarak, kalp hızı değişiminin takip performansı üzerine etkisi çalışıldı ve tahmin algoritmaları 3 serbestlik derecesi olan bir sınama ortamı kullanılarak önceden kaydedilmiş kalp hareketi verileri ile değerlendirildi.

Bunların yanında, bu çalışmada kullanılan kalp hareket verilerini toplayan sezici sistemin gürültüsünü tanımlamak için olasılıksal bir robotik yaklaşım takip edildi. Oluşturulan model, sezici sistemden toplanan gürültülü ölçümleri süzmek ve temizlemek için istihdam edildi. Daha sonra, süzülmüş sezici ölçümler, ilgi noktasının kalp yüzeyindeki yerinin doğru bir şekilde belirlenmesi için kullanıldı. Son olarak, uyabilen tahmin algoritmalarından elde edilen tahminler, sunulan yaklaşımın performansını arttırmak amacıyla oluşturulan ölçüm modeline dahil edildi.

Anahtar sözcükler: Etkin göreceli hareket önleyici, sinyal tahmini, tıbbi robotik, cerrahi robotik, olasılıksal robotik.

Acknowledgement

First, I would like to express my deep gratitude to Dr. Cenk Çavuşoğlu for letting me involved in this research and giving me the opportunity of being a part of the MeRCIS lab. His endless support and guidance always kept me motivated and gave me hope throughout my study. I really appreciate his encouragement and steadfastness which influenced me and helped me to complete this work.

Especially, I am grateful to Dr. Hitay Özbay for his invaluable advice during these years. I would like to thank him for teaching me to think broadly and innovatively and leading me to the control systems field. His considerate and understanding personality as a mentor helped me to make the right choices in my career as a graduate student.

I am also thankful to Dr. Ömer Morgül and Dr. Uluç Saranlı for showing keen interest to the subject matter and accepting to read and review this thesis.

A special thanks goes to Dr. Özkan Bebek for all his help and support during the completion of this study. His enthusiasm in the surgical robotics, his inspiring ideas and our enlightening discussions provide significant contributions to the presented study which have been put into use in this thesis.

I am also appreciative of the generous support from Bilkent University, Department of Electrical and Electronics Engineering.

Last but not least, I am indebted to my family for their love and for believing in me.

Dedeme ve anneanneme

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Coronary Artery Bypass Graft Surgery

Coronary artery bypass graft (CABG) surgery requires surgeons to operate on blood vessels that move with high bandwidth. This rapid motion of heart makes it difficult to track these arteries by hand effectively [1]. Contemporary techniques either stop the heart and use a cardio-pulmonary bypass machine or passively restrain the beating heart with stabilizers in order to cancel the biological motion of heart during CABG surgery. However using on-pump CABG surgery might expose patient to suffer from long term cognitive loss due to complications that can occur during or after the surgeries as a consequence of stopping the heart [2]. Off-pump CABG surgery with stabilizers is limited to the front surface of the heart and significant residual motion is observed during stabilization [3].

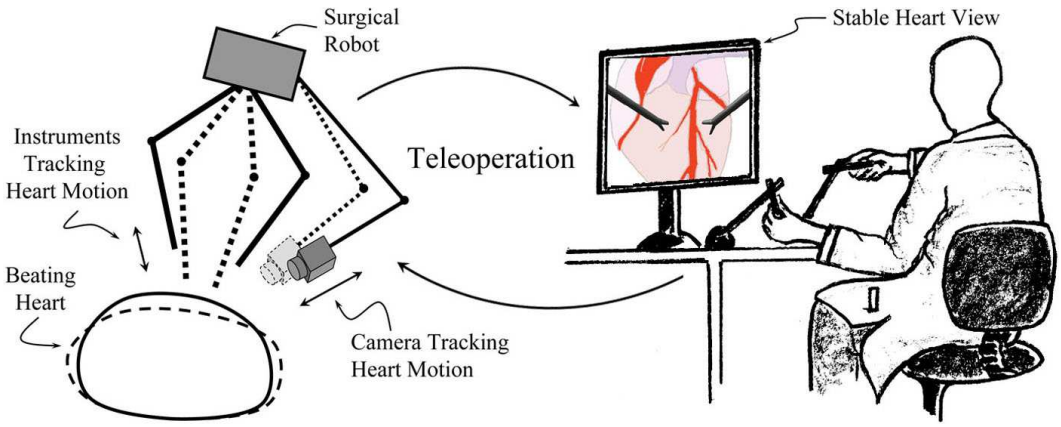

Figure 1.1: System Concept for Robotic Telesurgical System for Off-Pump CABG Surgery with Active Relative Motion Canceling (ARMC). Left: Surgical instruments and camera mounted on a robot actively tracking heart motion.

1.2 Robotic-Assisted Beating Heart Surgery

Robotic-assisted beating heart surgery emerges as a novel technology, which replaces the conventional surgical tools with robotic instruments. Figure 1.1 illustrates the proposed robotic assisted surgical system. In this system, a camera which is mounted on a robotic arm follows the heart motion. A surgical robot, which moves simultaneously with the heart, is used to track and cancel the relative three dimensional heart motion. By this way, it allows surgeon to experience the surgical site via stabilized views. Surgeon directly controls the surgical instruments through teleoperation and surgical instruments track heart motion and cancel the relative motion between heart and the instruments. Thus, the surgeon operates on heart as if it is stationary. This approach is called "Active Relative Motion Canceling (ARMC)". This would eliminate the need for stopping the heart and the use of cardio pulmonary bypass machine (the pump). Hence, robotic-assisted CABG surgery will prevent the occurrence of risks due to on-pump CABG surgery. It differs from the traditional off-pump CABG surgery with stabilizers since in the proposed robotic-assisted surgical system, heart motion is canceled with motion compensation. In contrast in the

traditional off-pump CABG surgery, heart is passively constrained with mechanical stabilizers [4].

1.3 Motion Estimation Algorithms for Model Based Active Relative Motion Canceling Algorithms

In CABG surgery, surgeon is required to operate on small blood vessels which move very rapidly. Their diameters vary from 0.5 to 2 mm and they have a quasiperiodic motion at the rates of 1 to 2 Hz. RMS tracking error for the position of a point of interest (POI) on the heart surface has to be in the order of 100-250 μm to perform precise operations on these vessels. The robotic tools need to track and manipulate a fast moving target with very high precision [5]. Causal error feedback control alone is not able to reduce the tracking error sufficiently such that surgery can be done on blood vessels on the heart surface. A predictive controller which implements a receding horizon model predictive control (RHMPC) in the feedforward path was found to be necessary [6, 7]. The proposed control architecture for designing motion estimation algorithm for ARMC on the beating heart surgery is shown in Figure 1.2

In the Model Based ARMC Algorithm architecture, the control algorithm fuses information from mechanical motion sensors which measure the heart motion. Motion of the point of interest (POI) has two dominant modes of motion, lung motion and heartbeat motion. These two modes are separated using proper filters. Lung motion has significantly lower frequency, and it can be canceled by a simple causal feedback controller. On the other hand, heartbeat motion has more demanding requirements in terms of the bandwidth of the motion that needs to be tracked. Thus a feedforward controller is required to cancel heartbeat motion.

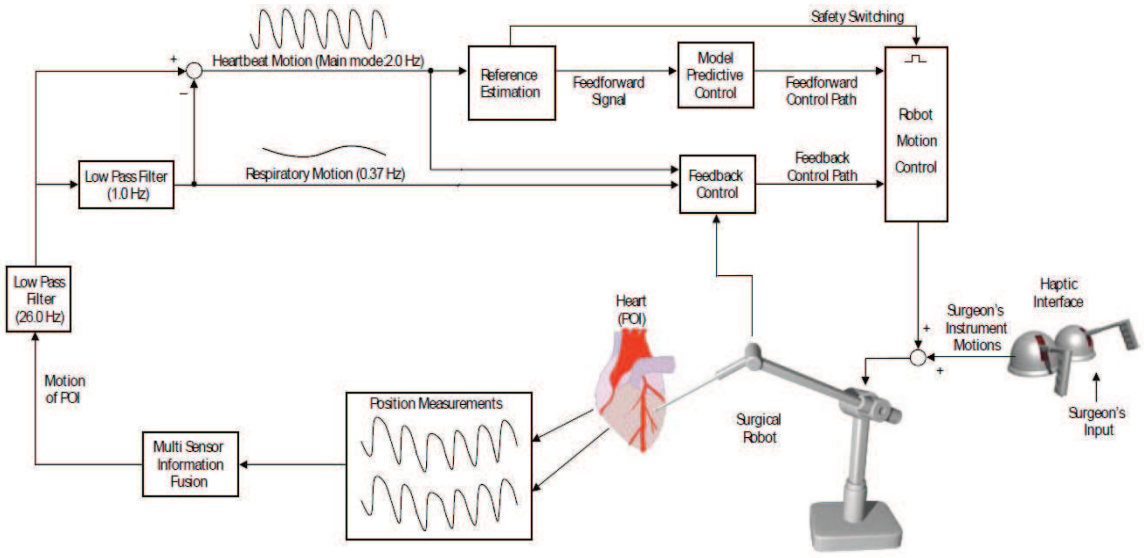

Figure 1.2: Proposed control architecture for designing Intelligent Control Algorithms for Active Relative Motion Cancelling on the beating heart surgery.

The confidence level reported by the heart motion model will be used to adaptively weight the amount of feedforward and feedback components used in the final control signal. This confidence level will also be used as a safety switching signal to turn off the feedforward component of the controller if an arrhythmia is detected, and switch to a further fail-safe mode if necessary. These safety features will be an important component of the final system.

The primary goal of this research is to improve the tracking performance of a surgical robot prototype as proof of concept that motion cancelation can be achieved. To this end, the tracking performance research has primarily been focused on developing estimation methods for use with a RHMPC. Such a predictive controller needs an estimate of the future motion of the POI on the heart surface. The estimate needs to be of a finite duration into the future, referred to as the prediction horizon.

In this thesis, heart motion prediction methods based on adaptive filtering techniques are studied. The implementations parametrize a linear system to predict the motion of the POI and rely on recursive least square adaptive filter algorithms. The presented methods differ as the first one assumes a linear system relation between the consecutive samples in the prediction horizon whereas the second method performs this parametrization independently for each point throughout the horizon. The effectiveness and feasibility of these algorithms are studied by simulations and on a 3-degree-of-freedom (DOF) hardware with constant and varying heart rate data.

The presented one step adaptive filter and the generalized adaptive filter are initially described by Franke *et al.* in [8] and [9] respectively. During the course of this research, these two algorithms are exhaustively and extensively studied. The bugs and errors in these predictors are fixed and the generalized predictor is completely reimplemented. As a result of this effort, the prediction performance of the algorithms and eventually the tracking performance of the intelligent control algorithms are significantly improved. These two algorithms are explained for the completion of the work and the results presented in this thesis.

1.4 A Probabilistic Robotics Approach for Sensing

A sensor that is in continuous contact with tissue is necessary for satisfactory tracking. The continuous contact sensor used in measuring the heart motion in the current literature is a Sonomicrometry system manufactured by Sonometrics Inc. (London, Ontario, Canada). A Sonomicrometer measures the distances within the soft tissue via ultrasound signals. A set of small piezoelectric crystals attached to the tissue are used to transmit and receive short pulses of ultrasound signal, and the time of flight of the sound wave as it travels between the transmitting and receiving crystals are

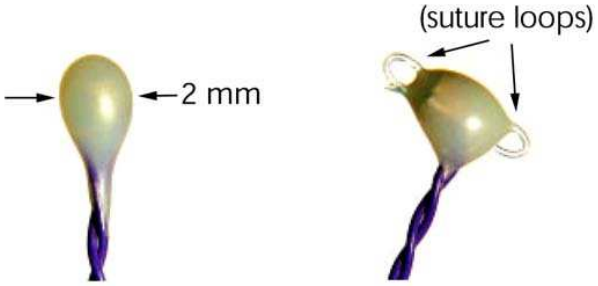

Figure 1.3: Piezoelectric crystals (courtesy of Sonometrics Corporation). Left: Standard piezoelectric crystal in 2 mm diameter that were used on the base plate. Right: Piezoelectric crystal with suture loops embedded to the crystal head. Loops are used to suture the crystal onto muscle.

measured (see Figure 1.3) [10].

The Sonomicrometric position sensor has been the sensor of choice in the earlier studies of this research, but obtaining precise position measurements is essential in closed loop control for tracking the beating heart. Despite sonomicrometric sensors are very accurate, they contain noise from ultrasound echoes. [4].

It is crucial to provide good quality of heart motion data to intelligent control algorithms to make sure that these algorithms follows and cancels motion of POI on heart surface accurately. In this part of the research, a probabilistic robotics approach is followed to model the noise of the Sonomicrometry sensor system and a Bayes estimator is used to filter and clean the noisy measurements collected from the Sonomicrometry sensor system. Then the POI on the heart surface is localized using this filtered measurements.

The main reason for following a probabilistic robotic approach is to represent uncertainty in the sensor system explicitly, using the calculus of probability theory. In other words, instead of relying on a single hypothesis as to what might be the

exact effect of the ultrasound echoes on measurements, the applied probabilistic algorithms represent information by probability distributions over a whole space of possible hypotheses.

1.5 Contributions

This thesis presents two heart motion prediction methods based on adaptive filtering techniques. The presented methods differ as the first one assumes a linear system relation between the consecutive samples in the prediction horizon whereas the second method performs this parametrization independently for each point throughout the horizon.

Although, the presented one step adaptive filter and the generalized adaptive filter are initially described by Franke *et al.* in [8] and [9] respectively, the bugs and errors in these predictors are completely fixed and the generalized predictor is completely re-implemented.

In the literature these predictors are tested with very limited and short duration of data. During the course of this research, these two algorithms are exhaustively and extensively studied with a wide range of different heart motion data. Predictors are tested with both constant and varying heart rate motion data. This is the first study that uses real varying heart rate data to perform heart motion tracking. As a result of this effort, the prediction performance of the algorithms and eventually the tracking performance of the intelligent control algorithms are significantly improved.

With the presented algorithms, the estimation of future POI motion is no longer the bottleneck in the heartbeat motion tracking since the necessary amount of RMS tracking error for the POI on the heart surface is achieved to perform precise operations.

In this work, an effective probabilistic robotics approach is applied to model and characterize noise of the sensor system that is used to collect heart motion data used in this study. The applied model completely covers the noise of the sensor system and effectively filters the noisy measurements. In an in-vivo heart tracking experiment, this preliminary approach will provide an online filtering mechanism for the noisy sensor measurements and localize point-of-interest on the heart surface accurately.

1.6 Thesis Outline

The rest of this thesis is organized as follows. Related work and analysis of the experimental heart motion data are given in Chapter 2. Problem formulation, the prediction methods and how the methods differ from each other to create estimations throughout the prediction horizon are explained in Chapter 3. Implementation details are also addressed in this chapter. In Chapter 4 simulation and experimental results are given. Chapter 5 describes the probabilistic approach that is applied the model noise in the sensor system and explains two different localization algorithms that are used to filter and clean noisy sensor measurements in order to accurately localize the Point-of-Interest(POI) on heart. The results of the localization algorithms are given in Chapter 6. Finally, the discussion and conclusions are presented in Chapter 7.

Chapter 2

Background

2.1 Related Works in Literature

This thesis is concerned with estimating the prediction horizon for RHMPC–a control scheme that relies on the estimate of the prediction horizon as a reference signal. There has already been several proposed ways to estimate motion of a POI on the heart surface.

Nakamura *et al.* [11] performed experiments to track the heart motion with a 4-DOF robot using a vision system to measure heart motion. The tracking error due to the camera feedback system was relatively large (error on the order of few millimeters in the normal direction) to perform beating heart surgery. There are also other studies in the literature on measuring heart motion. Thakor *et al.* [12] used a laser range finder system to measure one-dimensional motion of a rat's heart. Groeger *et al.* [13] used a two-camera computer vision system to measure local motion of heart and performed analysis of measured trajectories, and Koransky *et al.* [14] studied the stabilization of coronary artery motion afforded by passive cardiac stabilizers using

3-D digital sonomicrometer. Richa *et al.* developed a tracking algorithm for the heart surface based on a thin-plate spline deformable model [15], and an illumination compensation algorithm which can cope with arbitrary illumination changes on the heart [16].

Ortmaier *et al.* [17] used Takens Theorem to develop a robust prediction algorithm, anticipating periods of lost data when a tool obscured the visual tracking system. Estimates were generated from a linear combination of embedding vectors of previous heart data. The weights were chosen such that better estimating vectors are weighted more heavily. The algorithm had a global prediction technique that correlated ECG signals to heart motion. It was able to estimate the system behavior when visual contact of the landmark was lost for some period of time.

Ginhoux *et al.* [7] separated breathing motion from heart motion in the prediction algorithm. The breathing motion was treated as perfectly periodic, since the patient would be on a breathing machine. The heart motion was predicted by estimating the fundamental frequency, as well as the amplitude and phase of the first 5 harmonics. This prediction was used to estimate disturbance so that the controller could correct for it.

Rotella [6] used the previous cycle of heart motion data as an estimate of future behavior. This lead to problems since the POI motion was not perfectly periodic. Bebek and Cavusoglu [4] improved upon this prediction scheme by synchronizing heart periods using ECG data and separated heart and breathing motion, predicting only heart motion. Bebek noted that the prediction method still could be improved.

Bader *et al.* [18] presented a model-based approach for reconstructing the position of any arbitrary POI and for predicting the heart's surface motion in the intervention area. They model the motion of a POI on heart surface by means a pulsating membrane model. The membrane motion is described by means of a system of coupled linear partial differential equations (PDEs) and obtained a bank of lumped

systems after spatial discretization of the PDE solution space by the Finite Spectral Element Method. A Kalman filter is employed to estimate the state of the lumped systems by incorporating noisy measurements of the heart surface.

Yuen *et al.* [19] developed a 1-DOF ultrasound-guided motion compensation system for cardiac surgery. The surgical system integrates 3D ultrasound imaging and a robotic instrument with a predictive controller that compensates for the 50-100 ms imaging and image processing delays to ensure good tracking performance. Yuen *et al.* [20] used Extended Kalman Filter (EKF) algorithm to predict the future position of mitral valve annulus motion. The EKF filter is used to feed-forward the trajectory of a cardiac target in order to compensate time delays occurred due to the acquisition of motion data by the 3D ultrasound imaging. They tested the performance of EKF in prediction and tracking in the presence of high measurement noise and heart rate variability. They reported RMS synchronization errors of 1.5 mm for trajectories derived from clinical heart rate variability data.

This thesis introduces new estimation algorithms into the controller described in the earlier work of Rotella [6] and Bebek and Cavusoglu [4]. New prediction technique using adaptive filters are proposed and used in place of the prediction algorithm of Bebek and Cavusoglu [4]. Since the new predictors are parameterized by a least squares algorithm, the predictors are inherently robust to noise. The predictors only use observations close to and including the present making it less susceptible to differences between heart periods than the algorithm of Bebek and Cavusoglu [4]. Where as Ginhoux *et al* [7] formulated prediction for periodic POI motion, no assumptions are made towards periodicity of the system *a priori*, rather the predictors are unconstrained so that they can best mimic the motion of the POI. The adaptive prediction algorithms presented in this thesis tested with constant and varying heart rate motion data. This is the first study that uses real varying heart rate data to perform heart motion tracking.

In addition, tracking and prediction performance of the adaptive predictors presented in this thesis and the EKF predictor used in the study of Yuen *et al.* [20] are compared.

2.2 Analysis of Heart Data

In this section of the thesis, the experimentally collected varying heart rate motion data, which are used in this study, are described. Data were collected from three calves and all the study is performed with on benchtop with these pre-recorded data. First the collection of heart motion data will be explained. Then the analysis of varying heart rate motion data is presented.

2.2.1 Experimental Setup for Measurement of Heart Motion

The prerecorded data used in this study was collected using a Sonomicrometry system (Sonometrics Inc., Ontario, Canada). The Sonomicrometry system has also been the sensor of choice in our previous work for measuring heart motion for robotic ARMC [4]. A Sonomicrometer measures the distances within the soft tissue via ultrasound signals. A set of small piezoelectric crystals attached to the tissue are used to transmit and receive short pulses of ultrasound signal, and the "time of flight" of the sound wave as it travels between the transmitting and receiving crystals are measured. Using these data, the 3-D configuration of all the crystals can be calculated [10]. Absolute accuracy of the Sonomicrometry system is 250 μm (approximately 1/4 wavelength of the ultrasound) [21].

In the experimental set-up one crystal of the Sonomicrometry system was sutured on the heart . While collecting measurements, this crystal on the heart was placed in two different locations. The first location, which is referred to as 'Top' in the rest of

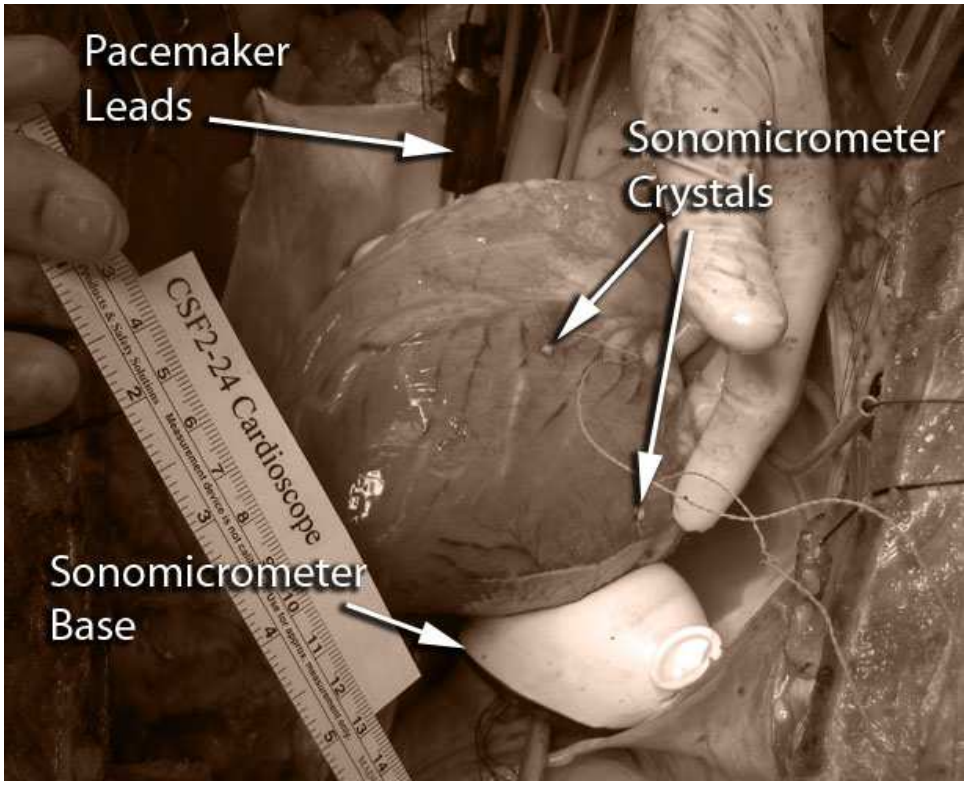

Figure 2.1: Experimental setup for measurement of heart motion. Two Sonomicrometer crystals that are sutured on the anterior and posterior surfaces of the heart are used for data collection. Pacemaker leads and Sonomicrometer base are also visible in the image.

the thesis, is located on the front surface. Specifically, the Sonomicrometry crystal was placed at 1 cm laterally from the left anterior descending coronary artery and 8 cm cranially from the LV apex. The second location, which is referred to as 'Side', is the location on the side surface of the heart. Specifically, in this case, the crystal was placed at 5 cm laterally from the left anterior descending coronary artery and 10 cm cranially from the LV apex. Five other crystals were asymmetrically mounted on a rigid plastic base of diameter 60 mm, on a circle of diameter 50 mm, forming a reference coordinate frame. This rigid plastic sensor base is placed in a rubber latex

balloon which is filled with a %9.5 glycerine solution. The reason of using such a set-up is to ensure a continuous line of sight between the base crystals and the crystal on the heart surface through a liquid medium for proper operation of Sonomicrometry sensor system. Figure 2.1 shows the experimental setup for measurement of heart motion. The Sonomicrometer crystals that are sutured on the heart can be seen from the figure. The pacemaker leads that are used to change the heart rate and the Sonomicrometer base are also visible.

Data were processed offline using the proprietary software provided with the system to calculate the 3-D motion of the POI. The only filtering performed on the data produced by the Sonomicrometry system was the (very limited) removal of the outliers, which occasionally occur as a result of ultrasound echoing effects. Although the Sonomicrometry system can operate at 2 kHz sampling rate for measuring the location of the POI crystal relative to the fixed base, in our test experiments, we have collected data at sampling rates of 257 Hz and 404 Hz in order to collect redundant measurements.

2.2.2 Analysis of Varying Heart Rate Motion Data

The motion of the heart surface is quasi-periodic in nature. The motion of the POI on the heart is primarily the superposition of two effects: motion due to the heart beating and motion due to breathing. Each of these signals closely resemble periodic signals.

In practice, the statistics of heart motion is likely to change during surgery. Such a change will result in variations in the underlying dynamics of the POI's motion. In order to explore the effects of these slow variations on the tracking performance and investigate how the adaptive algorithms will adjust to these changes, two distinct types of experimentally collected heart data is used in this study. The first type

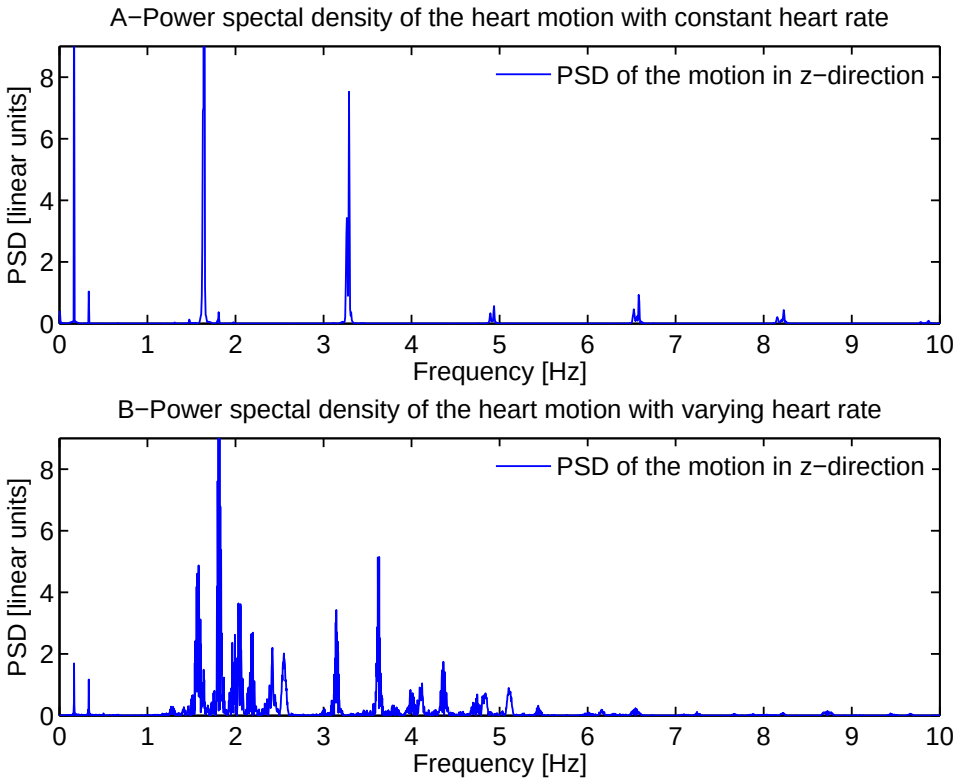

Figure 2.2: Power Spectral Density (PSD) of the heart motion in the z direction. (A) PSD of heart motion with constant heart rate. Tall, narrow peaks with the absence of intermittent frequencies indicate largely periodic motion of the heart. (B) PSD of heart motion with varying heart rate.

includes constant heart rate data whereas the second type includes varying heart rate data. From each calf a duration of 736-s, 472-s and 340-s of data are processed and used in this study.

Fourier analysis of the heart signal data with constant heart rate reveals how this periodic nature is prevalent (see Figure 2.2-A). The first peak corresponds to lung motion, which has the lower frequency with a fundamental period of approximately

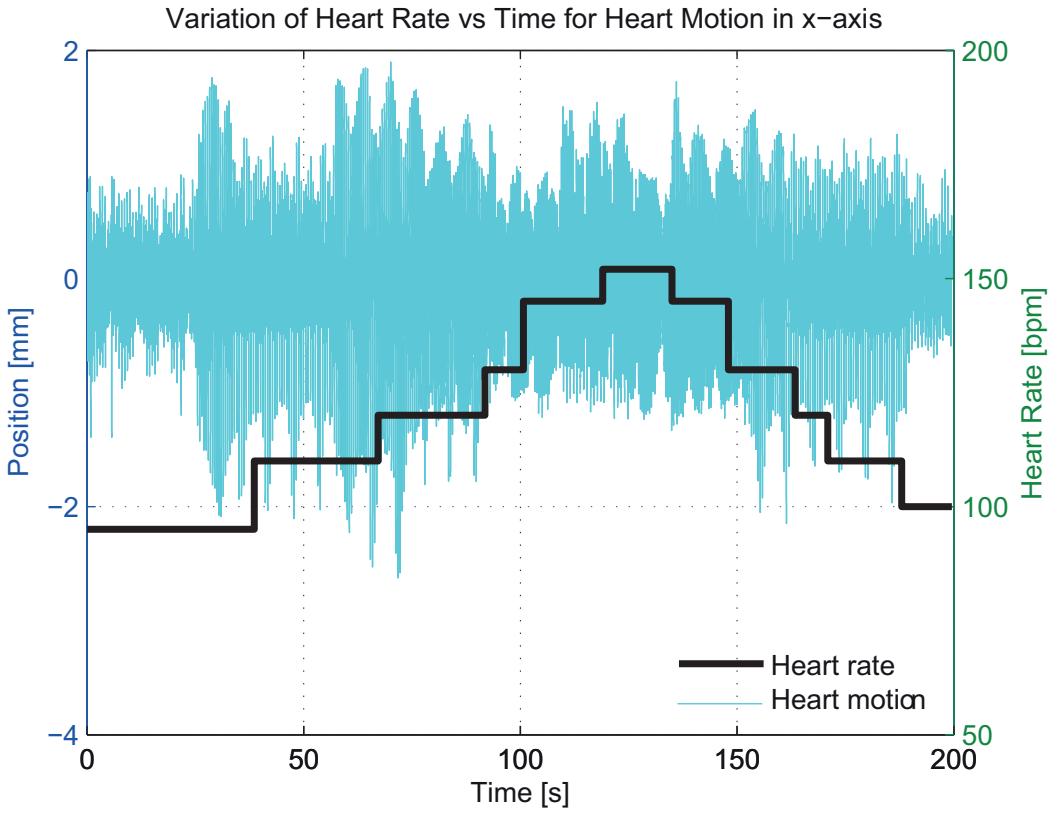

Figure 2.3: Variation of heart rate in the heart motion in x-direction.

0.17 Hz with first harmonic at 0.33 Hz is appearing significant. The heart motion itself has a fundamental frequency of 1.66 Hz, corresponding to 100 bpm, with the first four harmonics clearly visible in the figure. The peak displacement of the POI from its mean location was 8.39 mm, with a root-mean-square (RMS) value of 3.55 mm. The sharpness of these peaks indicate that the harmonics decay very little in time, meaning that the overall motion of the POI is similar to a superposition of periodic signals.

In order to change the heart rate, an artificial pacemaker is employed which uses electrical impulses to regulate heart rate, generated by electrodes contacting the

heart muscles. Initially heart is allowed to beat 40 seconds at 95 bpm. Then, the heart rate is gradually increased from 95 bpm to 152 bpm by approximately 10 bpm steps and then decreased in the same way, where heart is allowed to beat for at an average 15 seconds at a particular heart rate. Figure 2.3 shows the variation in heart rate with respect to time for the heart motion in x-direction. In the Fourier analysis of varying heart rate data, Figure 2.2-B, the first observable dominant mode at 0.17 Hz corresponds the breathing motion, similar to constant heart rate data, with a significant first harmonic at 0.33 Hz. The remainder of motion which is due to the beating of heart shows the fundamental frequencies of heart motion for different heart rates. The peaks at 1.58 Hz, 1.81 Hz, 2.03 Hz, 2.18 Hz, 2.42 Hz and 2.54 Hz correspond to heart rate of 95 bpm, 110 bpm, 120 bpm, 130 bpm, 145 bpm and 152 bpm, respectively. The peak displacement of the POI from its mean location was 7.43 mm, with a RMS value of 3.38 mm.

Chapter 3

Problem Definition and Methods

3.1 Problem Formulation

The control algorithm establishes the most essential part of the robotic tools for tracking heart motion during CABG surgery. Rapid motion of heart possesses demanding requirements on the control architecture in terms of the bandwidth of the motion that needs to be tracked. This necessity resulted in utilizing a feedforward algorithm in the control architecture in order to cancel high frequency components of heartbeat motion. In this study RHMPC (originally developed in [4]) was used as the feedforward control algorithm, requiring an estimate of the immediate future of the POI on the heart surface. If the feedforward controller has high enough precision to perform the necessary tracking, then the tracking problem can be reduced to predicting the estimated reference signal effectively [4].

The following notion will be used for formulating the motion estimation problem. Let z_i represent an observation at time i. In this case, z_i is a three dimensional column vector representing the location of the POI in Cartesian coordinates. At a

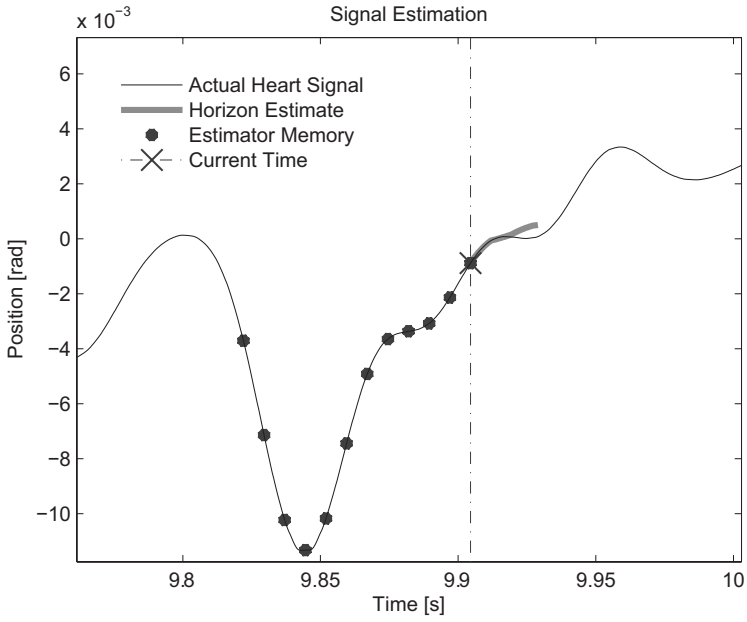

Figure 3.1: A schematic of the prediction problem. The circles represent past observations, now in memory, the 'X' is the current observation, and the short curve originating from there is the horizon estimate. The predictor takes the past observations and produces the horizon estimate from past observations.

given time step n, the observation z_n indicates the current *3D* position of the heart. Then, the observation z_{n-1} represents the previous position of heart at the time n; and the older observations will be referenced by decreasing subscript index, e.g., z_{n-5} is the observation from five samples ago. In a similar fashion, z_{n+1} represents the next observation at time n. Yet, this observation has not occurred, and will not be known until it becomes the present value. The estimate for the next observation at time n is introduced as $\hat{z}_{n+1}$.

Using this notation, the prediction problem can be posed as follows: Given the N-dimensional vector of known samples leading up to time n, $[z_n, z_{n-1}, \ldots, z_{n-N+1}]^T$, find the best estimate of the M-dimensional horizon, $[z_{n+1}, z_{n+2}, \ldots, z_{n+M}]^T$. Figure 3.1 provides a graphical schematic of this problem. The best estimate is defined

to be the one that minimizes the square of the estimation error, where the estimation error is the difference between the prediction and the observed value at that time. Once a method is established to predict the next observations, a sequence of future observations can be estimated. However, heart dynamics are nonlinear, which makes it quite challenging to parameterize a valid heart model in order to generate future predictions of POI from known samples.

Two adaptive filter based motion estimation algorithms are presented in this thesis to estimate reference signal, namely one step adaptive filter based motion estimation algorithm and a generalized adaptive filter based motion estimation algorithm. These two methods formulate and then parameterize the model of heart motion differently as described in the following two sections, Section 3.2 and Section 3.3.

3.2 One Step Motion Estimation Algorithm

An N^{th} order predictor has memory of past $N-1$ observations together with accessing to current observation. It generates the next expected observation in the prediction horizon based on these N observations. In order to generate the next future observation, what was just estimated is evaluated as if it was actually observed in the current time step and is added to the set of observations. Then the next value in the horizon is the next prediction that is obtained from this set. Proceeding in a similar fashion, any number of future estimates can be generated recursively till all the predictions for the horizon have been made.

In order to generate predictions in this way, the one step prediction function must be known. The motion of the POI interest is a continuous-time dynamic system, which is nonlinear as mentioned in the previous section. To establish a method for predicting future positions of POI, an equivalent discrete-time system has to be

used. Yet, neither the state space nor the dimension of the heart is obvious. So to simplify prediction method, a finite and low order state vector must be employed in the heart model. The state transition function for this approximate heart model is also nonlinear due to nonlinear dynamics of heart motion, which makes it difficult to parameterize.

The Fourier analysis of the heart signal data, presented in Section 2.2, reveals the quasiperiodic nature of the heart motion (see Figure 2.2-A). It shows that the heart position signal is composed of a main mode and additional harmonic where the motion is subject to small disturbance. This nature of heart motion allows the nonlinear state transition function to be approximated as linear by the following intuition.

The sharpness of the peaks of significant harmonics indicate that the harmonics decay very little in time, meaning that the overall motion of the POI is similar to a superposition of periodic signals. A linear system can easily be constructed that has a frequency response that mimics the heart signal's Fourier representation. The transient response would then resemble the observed heart data. Thus, given the current state of the actual heart signal as initial values for the system, the transient response would follow the actual heart motion−giving a prediction.

Finally, if the state was formulated as a stacked vector of past observations, then the determination of the initial state would be trivial. A linear system of the above specifications would meet the requirements for the heart model transition function. However, the model would still need to be parameterized in a way to statistically minimize the error of the prediction.

3.2.1 Model of Heart Motion

The heart position data consists of 3-dimensional vectors representing position. These vector samples are assumed to be generated from a vector autoregressive model (VAR). A VAR process has multiple output signals which are correlated with each other. The model is given by following equation [22]:

$$\vec{z}_k = \sum_{i=1}^{N} A_i \vec{z}_{k-i} + \vec{\gamma}_k \tag{3.1}$$

In this case, it is an N^{th} order VAR model. Each observation is given by a weighted sum of past observations, and is perturbed by noise given by $\vec{\gamma}_k$. Noise vector $\vec{\gamma}_k$ is assumed to be zero mean white noise. Since the linear combination of past observations account for correlation between observations, for any two noise vectors $\vec{\gamma}_k$ and $\vec{\gamma}_i$, $\vec{\gamma}_k$ is uncorrelated with $\vec{\gamma}_i$ for $i \neq k$. Since the noise vector is assumed to be white, it is not useful when generating predictions of future values. Therefore, when parameterizing the equation for the purpose of prediction, only the weighting matrices need to be estimated.

The VAR model given in (3.1) can be reformulated in state space canonical form as

$$\begin{aligned} \vec{X}_k &= \Phi \vec{X}_{k-i} + \Gamma \vec{v}_k \\ \vec{\sigma}_k &= C \vec{X}_k \end{aligned} \tag{3.2}$$

This system can be reformulated using an arbitrary state vector, however a stacked vector of past observations simplifies the determination of the initial state, parametrization of the state transition matrix Φ, and generation of the prediction

horizon. In this case, Φ is in canonical form and can be written as:

$$\Phi = \begin{bmatrix} A_1 & A_2 & \cdots & A_N \\ I & 0 & \cdots & 0 \\ 0 & I & & \vdots \\ \vdots & & \ddots & 0 \end{bmatrix} \tag{3.3}$$

Future observations of the system are given by solving the state space solution at time n. In order to find the expected trajectory, we take the expectation of (3.2) and find that the solution takes the form

$$E\{z_{n+k}\} = C\Phi^k \vec{X}_n \tag{3.4}$$

where the above formula gives the horizon estimate made at time n for a value k steps into the future. Note that since Φ^k is only computed for $k < M$, where M is the horizon length, Φ^k always remains finite. Therefore, stability of Φ is not a concern. Since $\vec{v}$ is unknown, but its expected value is zero by construction, it does not appear in the solution to the expected trajectory.

3.2.2 Adaptive Filter

The adaptive predictor consists of two principle parts: a linear filter and an adaption algorithm (see Figure 3.2). The input-output relation of the adaptive filter is determined by the linear filter. The adaptive filter's response is the response of the linear filter to the system's input. In this case, the linear filter will be a transversal filter. The adaptive algorithm changes the filter's weights in order to make the filter's output match the desired response in a statistical sense. The adaptive algorithm changes the filter's weights, so the filter is in fact not linear time-invariant. However, when the adaptive filter is adapting to a stationary signal, it will converge

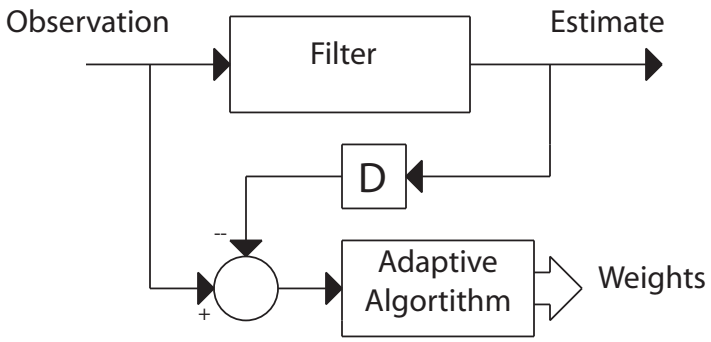

Figure 3.2: Adaptive predictor concept; the adaptive filter is arranged to minimize the error between the estimate for the current observation, calculated in the last iteration, and the actual observed value. In this way, the weights of the filter are statistically optimized to estimate one step ahead.

to a steady state, after which point it can be treated as being linear time-invariant.

If the adaptive algorithm is able to forget the past, just as it was able to converge to a stationary signal, it can track a signal with changing statistics [23]. In the special case that the statistics change slowly relative to the algorithm's ability to adapt, then the filter can track the ideal time-varying solution. Further, if the statistics change slowly relative to the length of the prediction horizon as well as the length of the state vector, then the adaptive filter can be considered to be locally linear time-invariant. The two afore mentioned conditions are the case with modeling the heart motion during most normal situations.

The adaption algorithm uses an exponential window to weight past observations so that more recent observations carry more weight. The exponential window was chosen because it can easily be implemented recursively. Due to this windowing, the adaptive predictor is able to track the heart signal even if the statistics of the heart signal change slowly with time.

3.2.3 Parametrization

Traditional system identification problems using adaptive filters arrange the filter such that the input to the filter is the system's input and the desired response is the system's desired response. In this way, the filter converges towards an approximation of the system's input-output relation. However, (Equation 3.2) is driven by white noise input vector $\vec{v}$. This input is unknown and unable to be predicted for future observations. Thus, deriving an input-output relationship for the heart motion would be impractical. Instead, the adaptive filter is arranged as a one-step predictor. The desired response is the heart position's current observation and the input to the adaptive filter is the previous heart observations. The adaptive filter adjusts its filter weights such that it generates the statistically best estimate for the next observation, given only the current and past observations.

In order to generate the predictions, the coefficient matrices, A_i, from (3.1), equivalently, the matrix Φ from (3.2) need to be estimated. The state transition matrix, Φ, is in controllable canonical form, so estimating A_i is sufficient to parameterize the estimated state transition matrix, denoted $\hat{\Phi}$. As can be seen from (3.1), the matrices A_i correspond to tap weights in a transversal filter. In a one-step predictor, when it has converged to a solution, its filter weights are precisely the matrices needed to parameterize $\hat{\Phi}$. In this way the adaptive algorithm estimates the matrix $\hat{\Phi}$.

3.2.4 Recursive Least Squares

Recursive least squares (RLS) was chosen as the adaptation algorithm to update the filter weights. RLS is a method that updates a least squares solution when a new piece of data is added. In practice, the RLS solution will approach the actual solution, even if the initial estimates for the solution were wrong. To formulate the RLS algorithm for vector samples, the one step prediction problem needs to be stated

as a least squares problem.

$$[z_{n-1}^T \quad z_{n-2}^T \quad \cdots \quad z_{n-N}^T]\ W^T = z_n^T \tag{3.5}$$

where the objective is to find W such that the square of the error between the two sides of the equation is minimized. At any time step z_n is the current position of POI on heart surface and $[z_{n-1}^T \quad z_{n-2}^T \quad \cdots \quad z_{n-N}^T]$ is the N dimensional vector of past positions. From this representation it is clear that

$$W = \ [A_1 \quad A_2 \quad \cdots \quad A_N] \tag{3.6}$$

where A_i are the weighting matrices from (3.1).

Using the statement of the least squares problem for the one step estimator in (3.5), the RLS algorithm can be derived. The derivation of the vector valued RLS algorithm is analogous to Haykin's derivation of the scalar case [23]. Since W is updated at every time step, the estimator is able to adapt to slowly changing heart behavior. Further, an exponential weighting factor can be introduced to produce a weighted least squares problem. This factor, λ is multiplied to each observation at each iteration, producing an exponential weighting of observations.

The RLS algorithm was formulated with past observations exponentially windowed such that the algorithm has the ability to forget the distant past. The exponential window parameter λ is referred to as the forgetting factor. When $\lambda = 1$, the RLS algorithm does not forget old observations, instead it has infinite memory. When $\lambda < 1$, observations are reduced in importance such that the least squares solution places a greater importance on minimizing error for the more recent observations and their prediction than on older ones. From the combination of weighted memory and convergence to the optimal solution, if the statistics of the heart motion change in time, the RLS algorithm is able to adapt to the new heart behavior.

3.2.5 Prediction

Following from (3.4), the one-step prediction is:

$$W \begin{bmatrix} z_n \\ z_{n-1} \\ \vdots \\ z_{n-N+1} \end{bmatrix} = \hat{z}_{n+1} \tag{3.7}$$

Once W is determined by (3.6), the stacked past observations vector is shifted down by one observation size and making the first past observation the current position, z_n. Then by matrix multiplication the one step prediction, $\hat{z}_{n+1}$, on the horizon is computed, which is is precisely the expected value of z_{n+1} from (3.1).

The prediction horizon of length M starting at time n is the solution to (3.4) with initial condition vector being the stacked vector of the past N observations.

In the actual implementation, predictions over the horizon length are generated by iterating this function several times. This avoids the computational complexity of calculating Φ^k and using it directly to compute the predictions. The calculation of $\vec{X}_n = \Phi\vec{X}_{n-1}$ is simplified by calculating $\hat{z}_{n+k}$ by (3.7), shifting the stacked observation vector $\vec{X}$ down by one observation size and making the first observation the current estimate. In this way, the computational complexity of iterating the state variable increases proportional to N, opposed to N^2. Since the observation matrix C from (3.2) simply retrieves the first observation from $\vec{X}$, multiplication by C is not necessary because the observation can be directly indexed and removed.

This recursive relationship can be written explicitly. If W is factored as $W =$

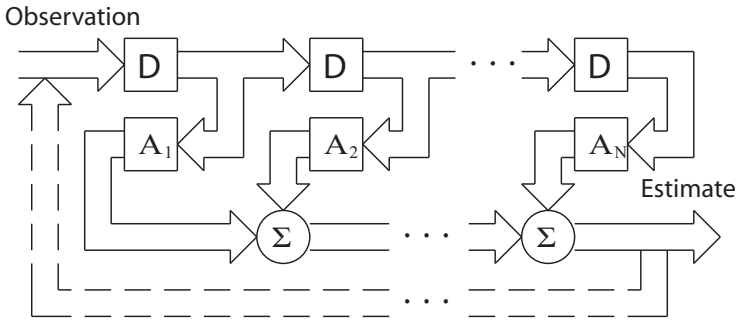

Figure 3.3: The generation of the horizon by the adaptive filter. The one step estimate is generated by use of a transversal filter weighting the past observations to produce an estimate for the next expected observation. When generating predictions for the horizon, the path is closed as the last estimate is treated as the current input. The prediction sequence is the collection of the estimate output each time the filter is iterated.

$C\Phi_{0,1}$ where

$$\Phi_{0,1} : [z_n, \ldots, z_{n-N+1}]^T \rightarrow [\hat{z}_{n+1}, z_n, \ldots, z_{n-N+2}]^T$$

$$C : [z_n, \ldots, z_{n-N+1}]^T \rightarrow z_n; \quad C = [I \;\; 0 \;\; \cdots \;\; 0],$$

then it is possible to define a matrix U such that it maps the memory of past observations to the expected horizon. In this case,

$$U = \begin{bmatrix} C\Phi_{0,1} \\ C\Phi_{0,1}^2 \\ \vdots \\ C\Phi_{0,1}^M \end{bmatrix} \tag{3.8}$$

$$U : (z_n, z_{n-1}, \ldots, z_{n-N+1}) \rightarrow (\hat{z}_{n+1}, \hat{z}_{n+2}, \ldots, \hat{z}_{n+M}).$$

Using the above described method for obtaining an estimate, the horizon is generated by collecting the next M estimates of the POI trajectory. Each time the process starts, the current state vector is composed of N-1 past observations together with the current observation. The first one-step prediction is generated by this state vector. Then, the state vector is shifted down by one observation size and the new prediction is used as the current observation. By following this procedure the next M estimates in the prediction horizon are generated (see Figure 3.3). This collection of M estimates is the expected POI trajectory given the N-1 past observations and the current observation. In order to generate next prediction horizon at the following time step, the aforementioned procedure is applied to the new state vector where the new state vector is composed of the new actual heart position data and corresponding $N-1$ past observations.

3.3 Generalized Linear Prediction

In Section 3.2, the optimal linear one step predictor, in the sense of prediction error magnitude, was formulated and used recursively to generate predictions. This method approximates the heart dynamics as being a linear discrete time system and leads to sub-ideal predictions, as the POI motion has nonlinear dynamics. In the generalized prediction method that is explained in this section, the assumption of a linear system relation between consecutive time samples is abandoned. Instead, a linear estimator for each point in the horizon is independently estimated. This is done by extending (3.7) as follows:

$$V \begin{bmatrix} z_n \\ z_{n-1} \\ \vdots \\ z_{n-N+1} \end{bmatrix} = \begin{bmatrix} \hat{z}_{n+1} \\ \hat{z}_{n+2} \\ \vdots \\ \hat{z}_{n+M} \end{bmatrix} \tag{3.9}$$

Where V is the estimation matrix that maps from the collection of observations to the expected horizon. In the same way as W was parameterized, RLS is used to determine V online and adaptively. However, since (3.9) contains the estimated values that are being solved for, it is unsuitable for implementation via RLS as is. This can be solved by assuming POI statistics to be stationary, or at least slowly varying, which makes V approximately constant. The assumption of time invariance of the heart statistics is utilized to introduce M delays so that all quantities have been observed when solving for V.

$$V \begin{bmatrix} z_{n-M} \\ z_{n-M-1} \\ \vdots \\ z_{n-N-M+1} \end{bmatrix} = \begin{bmatrix} z_{n-M+1} \\ z_{n-M+2} \\ \vdots \\ z_{n} \end{bmatrix} \tag{3.10}$$

The analogy can be made between (3.10) and an adaptive filter. The right hand side is the desired output and the observation vector on the left hand side is the input. Further, introducing the estimation matrices

$$\Phi_{0,i} : [z_n, \ldots, z_{n-N+1}]^T \rightarrow [\hat{z}_{n+i}, \hat{z}_{n+i+1}, \ldots, \hat{z}_{-N+n+i}]^T$$

for $1 \leq i \leq M$, then V can be decomposed similar to U in (3.8) as

$$V = \begin{bmatrix} C\Phi_{0,1} \\ C\Phi_{0,2} \\ \vdots \\ C\Phi_{0,M} \end{bmatrix} \tag{3.11}$$

The generalization of this prediction method results from the fact that, unlike in (3.8), $\Phi_{0,i}$ are parameterized independently and not, in general, equal to $\Phi_{0,1}^i$.

The removal of this constraint allows for the nonlinear dynamics throughout the prediction horizon to be better predicted by a linear estimator.

The predictor is implemented in a similar way to the previous vector RLS adaptive filter. The adaptive filter is formulated to solve the delayed estimation equation (3.10). This is equivalent to using a bank of n-step predictors, but is more computationally efficient. The largest cost in the RLS algorithm involves updating the inverse covariance matrix of the inputs. The generalized predictor is an improvement on to the one-step predictor, since in generalized predictor each estimate is using the same input vector. As a result the updating only needs to be done once, providing a dramatic reduction in computational complexity of one-step predictor when predictions are being made at many points throughout the horizon.

Chapter 4

Experimental Results

In this chapter, we evaluate the performance of the prediction methods that we presented in the previous chapter. In the literature these predictors are tested with very limited and short duration of data. In this research, these algorithms are extensively studied with a wide range of different heart motion data. This is the first study that uses real varying heart rate data to perform heart motion tracking.

We start by introducing our testbed and the performance metrics we use. Next, we compare the performance of existing methods in terms of these metrics on both constant and varying heart rate data. Finally, we compare the presented results with the reported values in the literature and show that the presented generalized predictor outperforms existing methods for heart motion tracking.

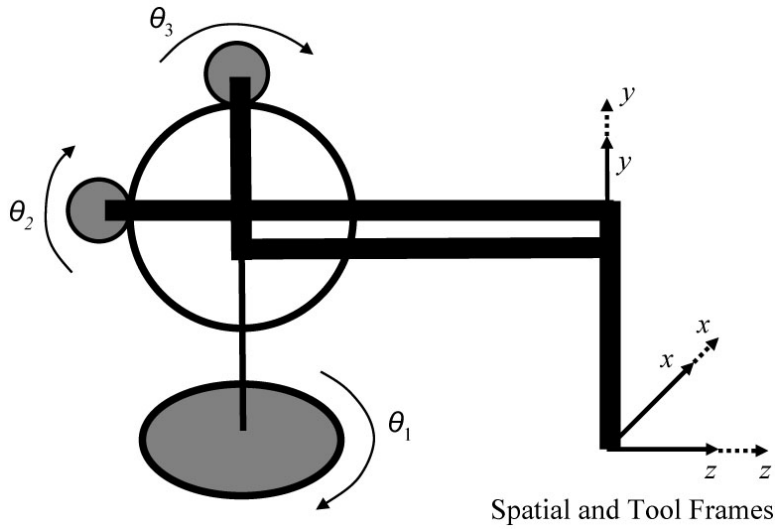

Figure 4.1: Zero Configuration of the PHANToM manipulator, also showing the axes movements and spatial and tool frames.

4.1 Experiments and Results

4.1.1 3-DOF Robotic Testbed

The proposed estimation algorithms were tested on a PHANToM Premium 1.5A haptic device, which is a 3-DOF robotic system. The nonlinearities of the system (i.e., gravitational effects, joint frictions, and Coriolis and centrifugal forces) were canceled independently from the controller. In order to maintain the accuracy of the experiments, the manipulator was brought to a selected home (zero) position, in the center of its workspace (more details can be found in [24]), before every experiment.

The controller used by Bebek and Cavusoglu [4] was modified to include the new prediction algorithms. The trials used the prerecorded heart motion data described in Section 2.2. The robot was made to follow the combined motion of heartbeat and breathing. The system was run using online streaming position data in place

of real-time measurements. The controller was implemented in xPC Target and run in real time with a sampling time of 0.5 ms on a Intel Xeon 2.33 GHz Core PC. The linearized robot model was controlled using RHMPC. The RHMPC was formulated to track the horizon estimate weighted by a quadratic objective function. The encoder positions on the PHANToM were recorded and these positions were transformed into end effector positions. The reported RMS errors are calculated from the difference between the prerecorded target point and the actual end effector position calculated from joint angles.

4.1.2 Simulation and Experimental Results

The same control method and reference data were used while running simulations and experiments. During the trials, a 16^{th} order correlated signal one-step estimator and a 10^{th} order generalized estimator predicting 4 different future points in the 25 ms horizon were used and quadratic interpolation was accounted for the intermittent points. The experiments were carried out using two different constant heart rate data and four different varying heart rate data.

Experiments were run 10 times with the estimation algorithms and again with the actual heart motion data as future signal reference for the prediction horizon. The later case represents a 'perfect' estimation, providing a performance base of the robotic system's capability. It was noted that the deviation between the trials are very small. Among these results, the maximum values for the *End-effector RMS and Maximum Position Errors in millimeters* in 3D and *RMS Control Effort in millinewton meters* are summarized in Table 4.1 for simulations and in Table 4.2 for experiments to project the worst cases. The results shown in Tables 4.1 and 4.2 are grouped with respect to type of the heart rate data collected from the animals. The position of the Sonomicrometer crystal on the heart surface, which are named as 'Top' position and 'Side' position are also stated.

SIMULATION AND EXPERIMENTAL RESULTS: The fixed heart rate data from animal 1 is 235 s long with a sampling rate of 257 Hz and from animal 2 is 472 s long with a sampling rate of 404 Hz. The sampling rate of all data sets with varying heart rate are 404 Hz. The duration of the varying heart rate data from animal 1 is 251 s for top position and 250 s for the side position. The duration for the varying heart from animal 3 is 200 s for top position and 140 s for side position.

Table 4.1: Simulation Results for End-Effector Tracking

A - RMS Position Error and MAX Position Error for the Control Algorithms

End-effector Tracking Results	RMS Position Error [mm] (Maximum Position Error [mm])					
Heart Rate	Fixed		Varying			
DataSet	Animal 1	Animal 2	Animal 1	Animal 3	Animal 1	Animal 3
Crystal Position	Top	Top	Top	Top	Side	Side
RHMPC with Exact Reference Information	0.488 (1.428)	0.237 (1.236)	0.231 (0.777)	0.197 (0.650)	0.194 (1.542)	0.231 (1.033)
RHMPC with One-Step Adaptive Filter Estimation	0.524 (1.953)	0.255 (1.460)	0.247 (1.098)	0.206 (0.917)	0.201 (2.163)	0.237 (1.195)
RHMPC with Generalized Adaptive Filter Estimation	0.481 (1.399)	0.235 (1.173)	0.229 (0.767)	0.195 (0.861)	0.191 (1.540)	0.230 (1.059)

B - RMS Control Effort for the Control Algorithms

End-effector Tracking Results	Control Effort [mNm]					
Heart Rate	Fixed		Varying			
DataSet	Animal 1	Animal 2	Animal 1	Animal 3	Animal 1	Animal 3
Crystal Position	Top	Top	Top	Top	Side	Side
RHMPC with Exact Reference Information	18.873	14.589	16.647	11.719	13.675	14.137
RHMPC with One-Step Adaptive Filter Estimation	26.685	21.801	37.991	18.010	30.402	20.027
RHMPC with Generalized Adaptive Filter Estimation	19.865	17.294	16.786	12.242	13.840	13.909

Table 4.2: Experiment Results for End-Effector Tracking

A - RMS Position Error and MAX Position Error for the Control Algorithms

End-effector Tracking Results	RMS Position Error [mm] (Maximum Position Error [mm])					
Heart Rate	Fixed		Varying			
DataSet	Animal 1	Animal 2	Animal 1	Animal 3	Animal 1	Animal 3
Crystal Position	Top	Top	Top	Top	Side	Side
RHMPC with Exact Reference Information	0.344 (1.238)	0.162 (0.912)	0.163 (0.780)	0.171 (0.559)	0.161 (0.538)	0.165 (0.906)
RHMPC with One-Step Adaptive Filter Estimation	0.404 (2.236)	0.176 (1.395)	0.181 (1.576)	0.199 (1.084)	0.173 (0.960)	0.188 (1.022)
RHMPC with Generalized Adaptive Filter Estimation	0.351 (1.291)	0.174 (1.022)	0.168 (0.827)	0.178 (0.615)	0.164 (0.572)	0.167 (0.972)

B - RMS Control Effort for the Control Algorithms

End-effector Tracking Results	Control Effort [mNm]					
Heart Rate	Fixed		Varying			
DataSet	Animal 1	Animal 2	Animal 1	Animal 3	Animal 1	Animal 3
Crystal Position	Top	Top	Top	Top	Side	Side
RHMPC with Exact Reference Information	54.379	28.512	25.350	21.593	24.390	27.260
RHMPC with One-Step Adaptive Filter Estimation	55.686	33.785	46.820	24.346	52.640	29.592
RHMPC with Generalized Adaptive Filter Estimation	54.948	29.699	25.760	22.082	24.635	27.830

Tracking results for a constant heart rate data with the one-step estimator in two different scales is shown in Figure 4.2 and results for varying heart rate data with the generalized adaptive filter estimation is shown in Figure 4.3. When Figure 4.2-A and Figure 4.3-A are compared, the variations in the heart rate can be observed from the pattern of the reference signal for x-axis in Figure 4.3-A. In Figure 4.2-B and Figure 4.3-B, magnitude of the end effector position error superimposed with the reference signal for the x-axis is shown.

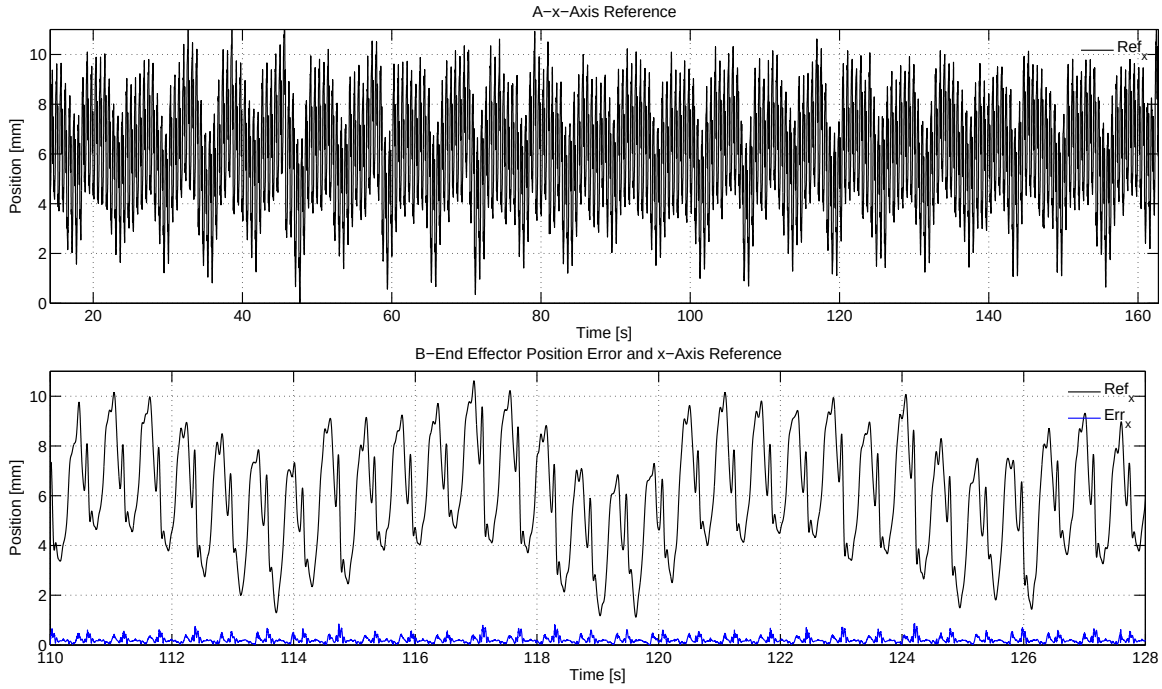

Figure 4.2: Tracking results for 157-s constant heart rate heart motion data in two different scales with RHMPC with One-Step Adaptive Filter Estimation. (A) The reference signal for the x-axis. (B) Magnitude of the end effector error (below) superimposed with the reference signal for the x-axis.

We believe that, the maximum error values are affected from the noise in the data collected by Sonomicrometry sensor as it is unlikely that the POI on the heart is capable of moving 5 mm in a few milliseconds. The data has been kept as-is without applying any filtering to eliminate these jumps in the sensor measurement data as currently we do not have an independent set of sensor measurements (such as from a vision sensor) that would confirm this conjecture.

As it can be seen from the results presented in Table 4.1, in our simulations the generalized estimator out performed the exact heart signal in terms of RMS Position error. This is likely due a combination of two factors. First, the simulation model is a linearized, reduced order model of the actual hardware. Second, the estimator has a robustness characteristic that makes its output less noisy than the actual

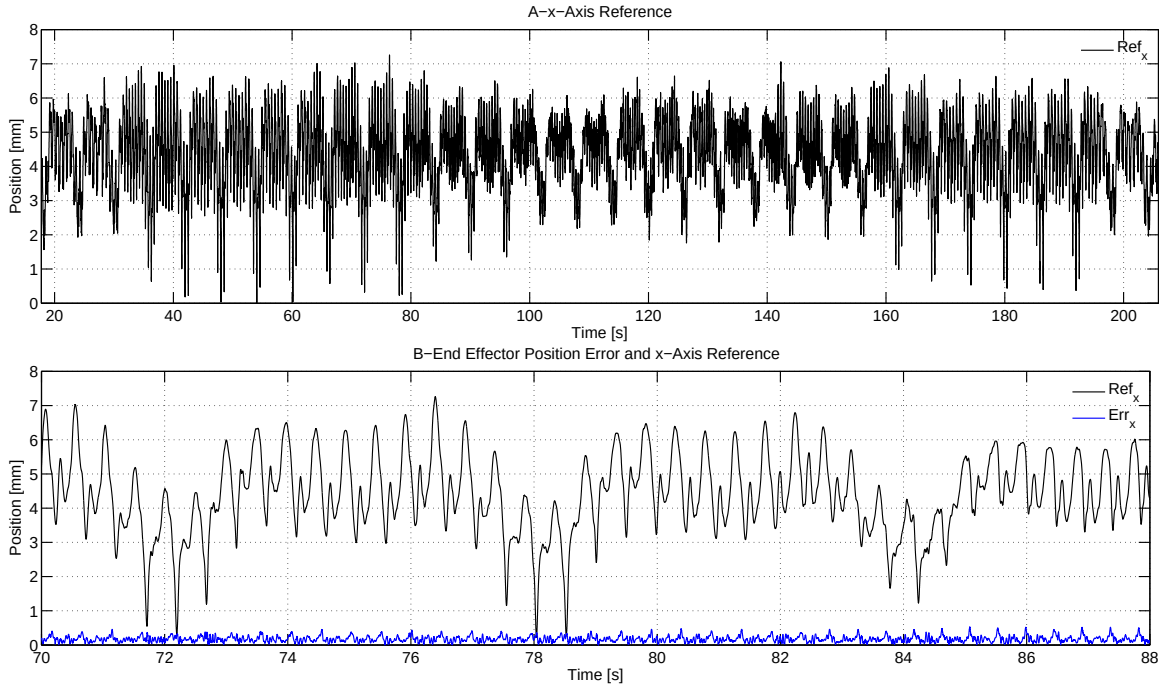

Figure 4.3: Tracking results for 200-s varying heart rate heart motion data in two different scales with RHMPC with Generalized Adaptive Filter Estimation. (A) The reference signal for the x-axis. (B) Magnitude of the end effector error (below) superimposed with the reference signal for the x-axis.

heart data. The combination of these two factors yields better results in the linear case. However, when the experiment is performed on the hardware, the effects of the nonlinearities are seen when the performance of the estimator-driven controller decreases. It should be noted that although the simulation provides valuable insight to the effectiveness of the controller, it is the experimental trials that are the best indicator of performance.

When the tracking results of the adaptive predictors are compared with each other, the generalized predictor outperforms the one-step predictor both in simulations and experiments. These results show that the nonlinear dynamics of the POI throughout the prediction horizon are better predicted by a generalized estimator.

From the results presented in Table 4.2, it can be observed that, in the experiments the controller with exact heart signal reference performs better than the one-step estimator and the generalized estimator in term RMS end effector error for both constant heart rate data and varying heart rate data. Maximum error and the control effort results for the exact heart signal are also smaller than the tracking results of one-step and generalized estimators, because the controller with exact heart signal reference represents the perfect estimation for heart motion tracking.

4.1.3 Discussion of the Results

At this point, it would be informative to compare the presented tracking results with the reported values in the literature.

Ginhoux *et al.* [7] used motion canceling through prediction of future heart motion using high-speed visual servoing with a model predictive controller. Their results indicated a tracking error variance on the order of 6-7 pixels (approximately 1.5-1.75 mm calculated from the 40 pixel/cm resolution reported in [7]) in each direction of a 3-DOF tracking task. Although it yielded better results than earlier studies using vision systems, the error was still very large to perform heart surgery.

Bebek and Cavusoglu used the past heartbeat cycle motion data, synchronized with the ECG data, in their estimation algorithms. They achieved 0.682 mm RMS end-effector position error on a 3-DOF robotic test-bed system [4].

Yuen *et al.* used an Extended Kalman Filter (EKF) algorithm with a quasiperiodic motion model to predict the path of mitral valve motion in order to compensate the time delay occurred from the 3-dimensional ultrasound (3DUS) measurements. They achieved 1.15 ± 0.004 mm RMS tracking error for a 1-DOF motion compensation instrument (MCI) in an in vitro 3DUS-guided servoing test. They stated that employing the EKF based predictor in time-delay compensation restores the tracking

Table 4.3: Experimental Results for End-Effector Tracking: RMS End-Effector and Maximum Position Errors for the Controller with EKF Predictor. The results for RLS-Based Adaptive Algorithms from Table 4.2-A also presented for comparison.

End-effector Tracking Results	RMS Position Error [mm] (Maximum Position Error [mm])					
Heart Rate	Fixed		Varying			
DataSet	Animal 1	Animal 2	Animal 1	Animal 3	Animal 1	Animal 3
Crystal Position	Top	Top	Top	Top	Side	Side
RHMPC with Exact Reference Information	0.344 (1.238)	0.162 (0.912)	0.163 (0.780)	0.171 (0.559)	0.161 (0.538)	0.165 (0.906)
RHMPC with One-Step Adaptive Filter Estimation	0.404 (2.236)	0.176 (1.395)	0.181 (1.576)	0.199 (1.084)	0.173 (0.960)	0.188 (1.022)
RHMPC with Generalized Adaptive Filter Estimation	0.351 (1.291)	0.174 (1.022)	0.168 (0.827)	0.178 (0.615)	0.164 (0.572)	0.167 (0.972)
RHMPC with EKF Estimation	1.148 (5.157)	0.386 (2.863)	0.515 (3.006)	0.523 (2.859)	0.433 (2.475)	0.449 (2.796)

performance of MCI to baseline tracking conditions in cases of delay. They reported that EKF gives better predictions than the AR filtering algorithms and last cycle method used by Bebek and Cavusoglu [4] in the presence of high noise and heart rate variability. Yuen *et al.* concluded since EKF explicitly models the quasiperiodic motion of heart it can adjust to rapid changes in heart rate better than other algorithms [20].

In order to compare the tracking performances using the proposed one-step and the generalized predictors with the EKF algorithm developed by Yuen *et al.* [20], the same hardware experiment described in Section 4.1.2 was repeated by employing the estimates generated by EKF in the RHMPC controller. The experimental results of these experiments, which include end-effector RMS position errors and maximum end-effector position errors, are presented in Table 4.3. The results for RLS-Based

Adaptive Algorithms from Table 4.2-A are also presented in Table 4.3 for comparison. The results of the experiments showed that the proposed adaptive algorithms outperformed the EKF-based algorithm in terms of tracking performance.

Simulation studies similar to the ones in [20] were conducted to compare the prediction performances of the one-step predictor, generalized predictor, EKF and last-cycle methods, in order to further investigate the tracking results presented in Table 4.3, which seemed to contradict the results reported by Yuen *et al.* [20]. In these simulations, the prediction performances of the algorithms were explored in the presence of measurement noise and heart rate variations.

In the first simulation study, the effect of measurement noise on the predictor performance on a constant heart rate motion data was evaluated. The motion data of POI on heart surface was downsampled to 28 Hz and corrupted by a additive zero-mean Gaussian noise with standard deviation $0.3 \leq \sigma_r \leq 3$ mm to match the conditions used in [20]. Similarly, the performance was evaluated for 1-step ahead prediction for a 10 s of data after 30 s of initialization time for each predictor. The EKF based predictor from [20] was also implemented with the parameters presented in that study for comparison. The RMS measurement error for each predictor obtained by averaging across 100 Monte Carlo trials are shown in Figure 4.4. Results show that EKF performs the best in the presence of high measurement noise when compared with the other algorithms.

In the second simulation study, the performance of the predictors in the presence of variations in heart rate were evaluated. The motion data was constructed similar to the way described in [20]. First part of the data included heart motion at a constant rate of 103 bpm with a duration of 30 s and the second part was a 10 s of motion data at a different heart rate (103 + ΔHR bpm), which was varied between -10 $\leq \Delta$HR $\leq$ 10 bpm. The motion data with varying heart rate was generated by compression and dilation of the trajectory of POI on heart surface. Heart motion

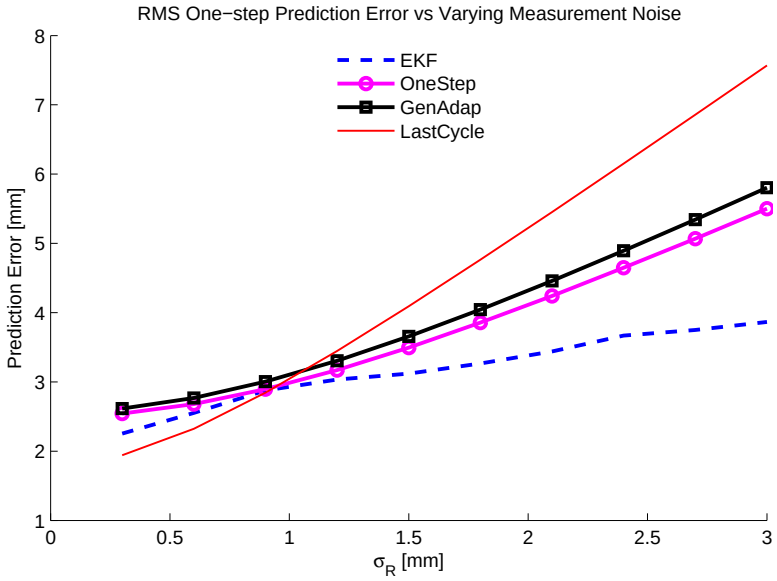

Figure 4.4: Plot showing the RMS prediction error results for a parametric simulation study where the predictors are tested in the presence of varying measurement noise at a sampling rate of 28 Hz.

data was again downsampled to 28 Hz and corrupted with additive white gaussian noise with $\sigma_R = 1.30$ mm. The performance of the predictors were evaluated only for the motion with varying heart rate and for the 1-step ahead predictions. The RMS errors were computed for 100 Monte Carlo trials and EKF was implemented again with the parameters given in [20]. The results presented in Figure 4.5 shows that EKF yielded better results than the AR filtering algorithms and last cycle method. One-step and generalized predictors provided similar results with former giving slightly better results. Finally, the last cycle method gave comparable results to adaptive predictors when variation in heart rate is small, yet the prediction error increases significantly when ΔHR increases.s

Results from these two simulation studies agree with the results reported in [20], indicating that EKF produces better one-step predictions than the adaptive algorithms and the last cycle method in the presence of high measurement noise and

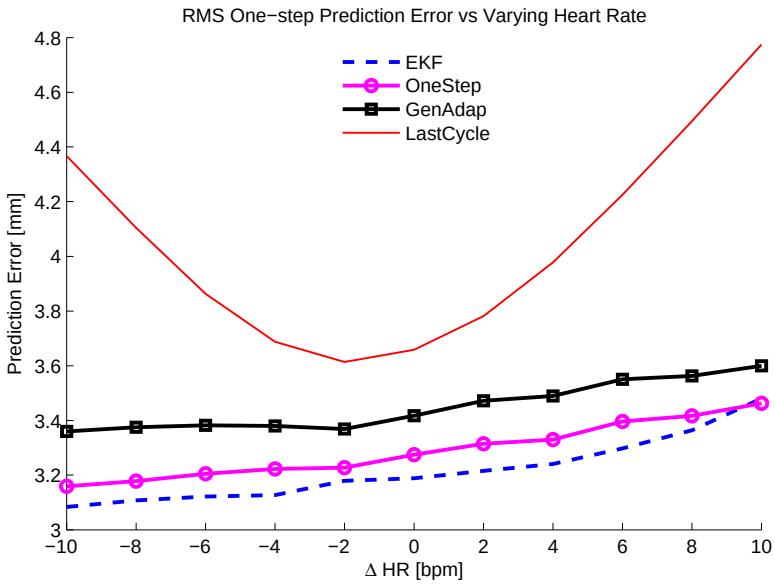

Figure 4.5: Plot showing the RMS prediction error results for a parametric simulation study where the predictors are tested in the presence of varying heart rate at a sampling rate of 28 Hz.

variations in heart rate at a sampling rate of 28 Hz. However, as the control algorithms employed in the real-time tracking hardware testbed operate at a sampling rate of 2 kHz, it is informative to also look at the prediction performance of the algorithms for varying heart rate at a sampling rate of 2 kHz. When the second simulation study was repeated at a 2 kHz sampling rate, it was observed that the one-step adaptive predictor produced better one-step predictions than the EKF predictor. Furthermore, the variations in the prediction performances of algorithms at different heart rates disappeared as a result of much higher sampling rate and no noticeable variations were observed in prediction performances at different heart rates as in Figure 4.5.

Based on these observations, the following reasons can be considered to explain the differences between the results presented in this study and the study of Yuen *et al.* [20]. First, and foremost, the comparison between the algorithms presented

in [20] was based on one-step prediction performances in simulation, whereas, the results reported above in Table 4.3 compares algorithms in terms of the tracking performances on a hardware testbed. Because of the high order and nonlinear dynamics of the robotic platform and the controllers employed, a better one-step prediction performance does not necessarily translate to a better tracking performance. Second, the sampling rates used in the two studies were different. The experiments, and as a result, the prediction and control algorithms, in this study used a sampling rate of 2 kHz. On the other hand, Yuen *et al.*, acquired the EKF predictions at the 3DUS sampling rate of 28 Hz. And, as mentioned before, the relative performances of the algorithms appeared to be different at different sampling rates. The third factor is the nature of the data used in the two studies. In this study, the 3D motion of a POI located on the surface of heart motion is used. On the other hand, Yuen *et al.* characterized the motion of mitral valve annulus by a 1D model. In addition, the hardware experiments in this study and in [20] represents two different cases, as the test beds used to evaluate the performance of the predictors are quite different. The differences in the inner dynamics of the experimental set-ups might lead to different tracking performance results. Namely, if RLS-based adaptive filters would have been employed for tracking in [20], the tracking performance of RLS-based predictors can be expected to be different when compared to this study.

The generalized predictor presented in this thesis represent the best results reported in the literature. These results show that the model predictive controller with the proposed generalized estimator and the exact reference data performed equally well, which indicates that the main cause of error is no longer the prediction but the performance limitations of the robot and controller. It is important to note that the results also need to be validated *in vivo*, which were the case in [7, 20].

Chapter 5

Probabilistic Robotics Approach

5.1 Motivation and Methodology

In robotic-assisted CABG surgery heart motion is canceled with motion compensation. To achieve this motion compensation, a predictive controller which implements a RHMPC in the feedforward path was found to be necessary as emphasized in section 1.2. Such a predictive controller needs an estimate of the future motion of the POI on the heart surface.

In order to estimate the future motion of POI on heart surface accurately and then subsequently cancel this motion effectively, it is essential to provide noise-free and good quality of heart motion data to these algorithms. For this purpose, it is required to filter and clean the measurements obtained from the sonomicrometry sensor system, which is used for measuring heart motion in this research.

For the one step and the generalized motion estimation algorithms presented in Section 3.2 and Section 3.3 the provided heart motion data is only filtered offline in

order the clean the ultrasound echoing effects of the Sonomicrometry sensor system. The tracking results presented in Section 4.1.2 are obtained via using this offline filtered data.

As stated, these tracking results need to be validated *in-vivo* (see Section 4.1.2). During an *in-vivo* experimental procedure it is not possible to do any offline filtering in order to clean the existing noise in measurements, since the incoming heart data would be online. In the light of these facts, an online processing method to clean the sensor measurements is found to be required.

Sonomicrometry sensor system is very accurate and the major source of error is the crystal geometry [10]. However, this system contains noise from the ultrasound echoes and it is prone to error due to the calibration between the base sensors and the robotic manipulator coordinate frame.

In order to represent these uncertainties explicitly in the Sonomicrometry sensor system a probabilistic robotics approach is followed in this research. Using the probability theory, the noise in the sensor system is characterized. Then, a recursive Bayes estimator is used to filter and clean the noisy measurements collected from the Sonomicrometer. Finally, the POI on the heart surface is localized using these filtered measurements.

5.2 Recursive State Estimation

The idea of estimating state from sensor measurements is the core of the probabilistic robotics. In robotic applications, sensors carry only partial information about certain quantities *i.e.* locations of a mobile robot and nearby obstacles and without a doubt these measurements are corrupted by noise. State estimation aims to recover state variables from the data gathered by the sensors.

In probabilistic robotics, one of the most essential concepts is *belief* distributions. A belief simply represents the robots internal knowledge about the state of the environment. In other words, the belief of a robot is the posterior distribution over the state of the environment. Probabilistic state estimation algorithms compute belief distributions over all possible states in the existing environment. The principal algorithm for calculating the belief is *the Bayes filter*. This recursive algorithm calculates the belief distribution from measurement and control data [25].

Being a recursive algorithm is an essential property of the Bayes filter, that is, the belief $bel(x_t)$ at time t is calculated from the belief $bel(x_{t-1})$ at time $t-1$. The input to the algorithm is the belief distribution at time $t-1$, together with the most recent control u_t and the most recent measurement z_t. The output is the belief distribution, $bel(x_t)$, at time t [25].

Algorithm 1 depicts the general algorithm for Bayes filtering.

Algorithm 1 Bayes Filter Algorithm

1: Bayes_Filter($bel(x_{t-1}), u_t, z_t$)
2: **for all** x_t **do**
3: $\quad \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx$
4: $\quad bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t)$
5: **end for**
6: **return** $bel(x_t)$

In the Bayes filter algorithm, the probability $p(x_t|u_t, x_{t-1})$ is the state transition probability. It designates the evolution of environmental state x_t over time as a function of robot controls u_t. The probability $p(z_t|x_t)$ is the measurement probability. The measurement probability designates the probability that the measurements z are generated from the environment state x where the measurements are regarded as the noisy projections of the state. These two probabilities $p(x_t|u_t, x_{t-1})$ and $p(z_t|x_t)$ together characterize the dynamical stochastic system of the robot and its environment.

The Bayes filter algorithm is composed of two fundamental steps, *prediction* or *control update* and *correction* or *measurement update* respectively. The prediction step, shown in Line 3, is performed by processing the control u_t. It is done by calculating a belief over the state x_t based on the prior belief over state x_{t-1} and the control u_t. Specifically, the predicted belief $\overline{bel}(x_t)$ is obtained by the integral of the product of two distributions; the prior belief assigned to x_{t-1}, and the state transition probability from x_{t-1} to x_t [25].

In the correction step, shown in Line 4, the predicted belief $\overline{bel}(x_t)$ is multiplied with the measurement probability of the sensor measurement z_t. It is done for each hypothetical posterior state x_t. The result is normalized, by a normalization constant η since the resulting product is generally not a probability and thus it may not integrate to 1. The result is the final belief $bel(x_t)$, which is returned in the final line of the algorithm. In order to compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$ [25].

In probabilistic robotics, different techniques are employed to implement Bayes filters. Each of these techniques relies on different assumptions regarding the state transition and measurement probabilities and the initial belief. Those different assumptions lead to different types of beliefs, and the algorithms for computing those belief distributions have different computational characteristics. Since an exact technique does not exist to calculate beliefs, they need to be approximated. In order to choose a suitable approximation to compute belief distributions, a trade-off must be made between the certain properties of the algorithm such as computation efficiency, accuracy of the approximation, and ease of implementation [25].

In the implementation of Bayes filtering algorithms for this study, sonomicrometry sensor system provides the sensor measurements, z, by computing the distance between the crystal sutured on the heart and each crystal located on the base plate.

The *3D* position of the POI on the heart surface would be treated as the environmental state variable, x, and the heart motion data will be used as the control data, u, throughout the presented study.

Rest of this chapter is organized as follows. In the next two sections two components for implementing the Bayes filtering algorithms will be described; the motion and the measurement models. The motion model deals with state transition probability, $p(x_t|u_t, x_{t-1})$ and measurement model deals with measurement probability, $p(z_t|x_t)$. Then, two different Bayes filtering algorithms will be explained, Extended Kalman Filter (EKF) and Particle Filters. Finally, the results concerning the localization of the POI on the heart surface by implementing these algorithms will be presented.

5.3 Motion Model

This section focuses on the motion model. Motion models covers the state transition probability $p(x_t|u_t, x_{t-1})$ and plays an essential role in the prediction step of the Bayes filtering algorithm presented in Line 3 of Algorithm 1.

The idea behind studying motion models by a probabilistic approach comes from the fact that the outcome of a control action is uncertain, due to control noise or unmodeled exogenous effects. Therefore the generated motion model will be suitable to the probabilistic state estimation techniques [25].

In order to implement the Bayes Filter algorithm a motion model of heart is needed. The motion of the heart is quasi-periodic in nature, including the heart beating and breathing motions. In Section 2.2.2, the Fourier analysis of the heart motion depicts that first harmonic of the breathing motion and first four harmonics of the heart beating motion is clearly visible

In order to simplify this motion, two different approaches are followed to construct a motion model. The first approach is the use a Brownian motion and the second approach is to construct a harmonic motion by using only the first two harmonics heart motion.

5.3.1 Brownian Motion

The Brownian motion, an idealized approximation to actual random physical processes, is used to represent the motion of heart beating. A Brownian motion which is also called a Wiener process is a stochastic process. It is a collection of random variables $S(t)$ that are defined on the same probability space (Ω, F, P), satisfying the following conditions:
1- $S(t) = 0,\ t = 0$;
2- With probability one, $S(t)$ is continuous in t;
3- $S(t)$ has stationary and independent increments, i.e. for any positive integer n and any $0 = t_0 < t_1 < \cdots < t_n$, the random variables $S(t_i)$ - $S(t_{i-1})$, $i = 1, \ldots, n$ are mutually independent, and $S(k+t)$ - $S(k)$ has the same distribution as $S(t)$ for any $k, t > 0$;
4- $S(t) \sim N(0, \sigma^2 t)$.
From *Property* 4, it is clear that $S(t)$ has the normal distribution with mean 0 and variance $\sigma^2 t$ for some constant σ^2. [26]

The state update equation of the Brownian motion is given by the following linear equation.

$$x_t = x_{t-1} + \varepsilon_t \tag{5.1}$$

Here, x_t and x_{t-1} are the three dimensional state vectors, which represent the

position of the crystal on heart surface at time t and $t-1$ respectively. x_t is of the form:

$$x_t = \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{z}_t \end{bmatrix} \tag{5.2}$$

where $\hat{x}_t$, $\hat{y}_t$, $\hat{z}_t$ represents the *3D* position coordinates of the crystal with respect to reference coordinate frame. ε_t is the uncertainty induced by the Brownian motion, which is represented by a zero-mean multivariate Gaussian distribution, $N(0, \Sigma)$. Σ is of the form:

$$\Sigma = \begin{bmatrix} \Sigma_x & 0 & 0 \\ 0 & \Sigma_y & 0 \\ 0 & 0 & \Sigma_z \end{bmatrix} \tag{5.3}$$

$(\Sigma_x, \Sigma_y, \Sigma_z)$ are computed from the corresponding axes of three dimensional heart motion data which is only filtered offline to remove the outliers.

5.3.2 Harmonic Motion

Harmonics motion is another approximation to the actual heart motion. Although it is not simple as the Brownian motion, it resembles the actual heart motion better than the Brownian motion since it comprises actual components of the heart motion.

The spectral analysis of the heart motion in Section 2.2 show that heart motion, y_t, can be approximated by a certain number of harmonics. By using an $m - order$ Fourier series with constant offset and the first two harmonics ($m = 2$) of the heart

motion, the following simple approximation to a single axis of the actual heart motion can be obtained:

$$u_t = c_0 + \sum_{m=1}^{2} a_m sin(mw_0 t) + b_m cos(mw_0 t) \tag{5.4}$$

where in Equation 5.4, c_0 is the constant offset, $a_m \in \mathbb{R}^3$'s and $b_m \in \mathbb{R}^3$'s are the Fourier series coefficients and w_0 is the heart rate. The Fourier series coefficients are obtained by taking $N - point$ FFT of the heart motion data and the constant offset is set to the mean of the position values of the related reference axis. Equation 5.4 shows the harmonic approximation for a single axis of the three dimensional heart motion. This approximation is applied for each of the three axes to generate a three dimensional harmonic approximation of the actual heart motion. Figure 5.1 shows the constructed 2^{nd} order harmonics approximation superimposed on the reference signal for z-axis.

At any time t, the update equation of the original heart motion and harmonic motion model can be written as:

$$y_{t+\Delta t} = y_t + \Delta y_t \tag{5.5}$$

$$u_{t+\Delta t} = u_t + \Delta u_t \tag{5.6}$$

In the above equation y_t represents the three dimensional actual heart motion and u_t represents three dimensional harmonic motion. Similarly, Δy_t is a the three dimensional vector representing the increment of actual heart motion and Δu_t is a three dimensional vector representing the increment of harmonic motion.

Since the heart motion, y_t, at time t is approximated by the harmonic motion u_t, the increment Δy_t can be also approximated by Δu_t. In other words,

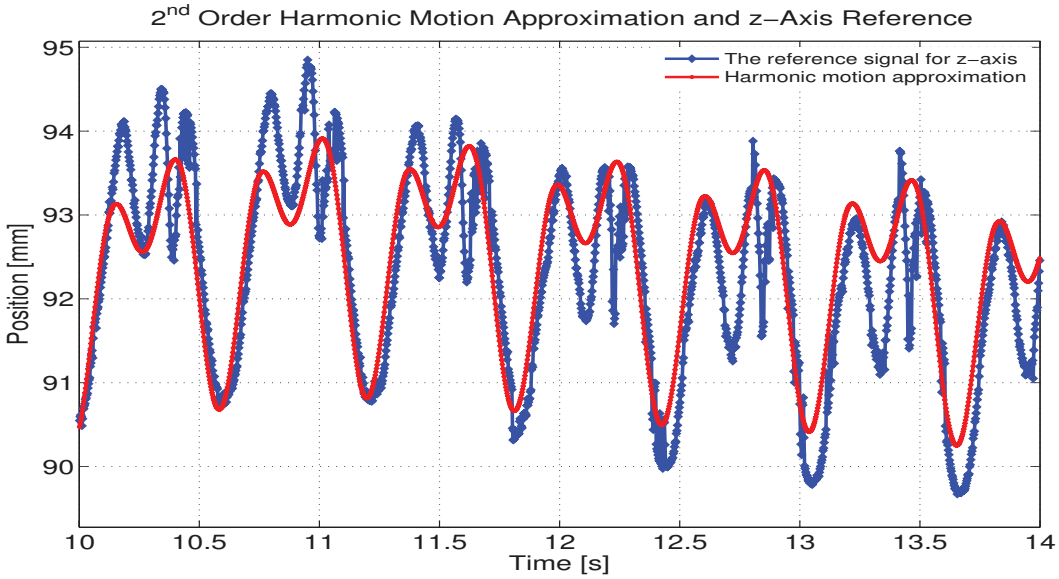

Figure 5.1: 2^{nd} order harmonics approximation superimposed on the reference signal for z-axis.

$$\Delta y_t \approx \Delta u_t \tag{5.7}$$

With this approximate, state update equation for the harmonic motion model is:

$$x_t = x_{t-1} + \Delta u_t + \varepsilon_{r_t} \tag{5.8}$$

Again, x_t and x_{t-1} represent the three dimensional position of the crystal on heart surface at time t and $t-1$ respectively. Δu_t is the update increment of the harmonics motion as shown in Equation 5.7. ε_{r_t} is the uncertainty induced by the process noise, which is represented by a zero-mean multivariate Gaussian distribution, $N(0, \Sigma_r)$, of the form:

$$\Sigma_r = \begin{bmatrix} \Sigma_{r_x} & 0 & 0 \\ 0 & \Sigma_{r_y} & 0 \\ 0 & 0 & \Sigma_{r_z} \end{bmatrix} \tag{5.9}$$

$(\Sigma_{r_x}, \Sigma_{r_y}, \Sigma_{r_z})$ are computed from the corresponding axes of the three dimensional remaining motion;

$$r_t = y_t - u_t \tag{5.10}$$

5.4 Measurement Model

This section will describe the probabilistic models of sensor measurements $p(z_t|x_t)$, that are crucial for the measurement update step. Probabilistic robotics explicitly models the noise in sensor measurements. Such models account for the inherent uncertainty in the robots sensors.

5.4.1 Sonomicrometry Sensor System

It is mentioned in Section 2.2 that Sonomicrometry sensor system is employed to collect the heart motion data used in this study. The schematic that shows the Sonomicrometry sensor model can be seen in Figure 5.2.

The sonomicrometer setup obtained from SonoMetrics Corporation has six channels for piezoelectric crystals. In this system one piezoelectric crystal was sutured on the heart and five other crystals were asymmetrically mounted on a rigid plastic base forming a reference coordinate frame in order to measure the motion POI on

heart surface.

The geometrical placement of the piezoelectric crystals on the base will affect the formation of the uncertainty geometry of the sensor. Thus, sensors should be mounted asymmetrically to prevent having homogeneous solutions since solutions depend on geometrical placement. In order to minimize the uncertainty geometry of the sensor, the base crystals should be placed evenly on a circle. Both of these could be satisfied by placing the crystals on a circle slightly shifting them from their original evenly spaced positions [10].

The *3D* Position coordinates and the distances between the crystals are given in Figure 5.3. Here the base crystals are named as $\{q_1, q_2, \cdots, q_5\}$. There are ten possible reference coordinate frame combinations that can be formed from the five base crystals, where the crystal q_1 is located at the origin in the shown default coordinate frame (see Appendix A).

All of the crystals are assumed to be well calibrated. Only possible source of error is due to the crystals geometry, which only affects the absolute value of the distance measurements. The most significant uncertainties in the measurements are due to the ultrasound echoes. Besides these uncertainties, no errors due to the inaccuracy of the flight time measurements, uniform speed of sound in the medium of measurements, and no weak signal reception are assumed [27].

The sensor system in Figure 5.2 provides 10 different sensor measurement to the user with 5 of them are independent. The first 5 independent measurements are obtained by treating 5 base crystals transmitters and the 6^{th} crystal on heart as receiver. The duplicates of these measurements are obtained by treating the crystal on heart as transmitter and base crystals as receivers.

Hence, a set of sensor data at time t which includes all of these 10 measurements can be written as:

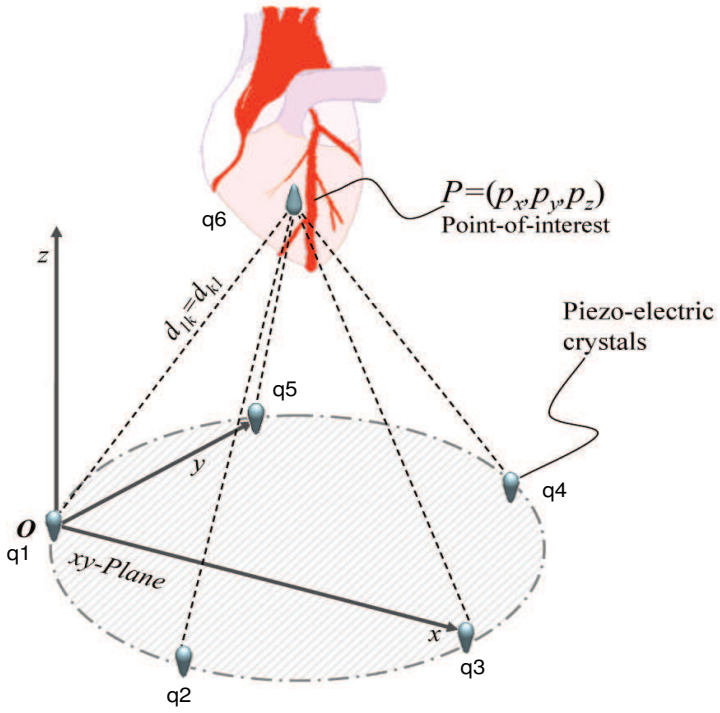

Figure 5.2: Sonomicrometry sensor model: Five crystals were mounted to a base to measure the distance of a sixth crystal attached to the heart.

$$z_t = \{z_t^1, \cdots, z_t^{10}\} \tag{5.11}$$

The accuracy of these measurements can be verified by calculating them from the true state x_t, the *3D* position coordinate crystal attached near the POI on heart surface (see Equation 5.12).

$$\begin{aligned} \hat{z_t^i} &= h_i(x_t), \\ h_i(x_t) &= ||x_t - q_i|| \end{aligned} \tag{5.12}$$

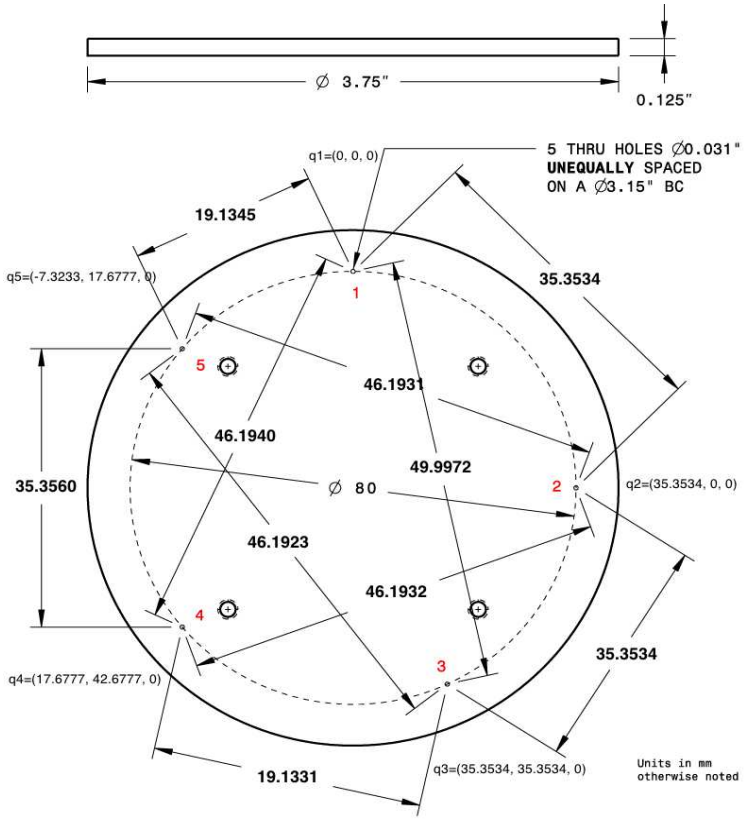

Figure 5.3: Sonomicrometry base plate, *3D* crystal position coordinates and crystal distances in mm.

where x_t represents the *3D* position coordinates of the crystal on heart surface and q_i for $i = 1, \cdots, 5$ represents the *3D* position coordinates of the base crystals.

The nonlinear Equation 5.12 is the Euclidean distances between the known *3D* position coordinates of the each base crystal and the *3D* position coordinate crystal attached on the heart. In the reference coordinate frame this nonlinear equation is expanded as:

$$h_i(x_t) = \sqrt{(\hat{x}_t - q_{i_x})^2 + (\hat{y}_t - q_{i_y})^2 + (\hat{z}_t - q_{i_z})^2} \tag{5.13}$$

where $(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ is the coordinate of the crystal on the heart surface as shown in Equation 5.2 and (q_{ix}, q_{iy}, q_{iz}) is the coordinate of crystal q_i on the base.

In order to compute the *3D* position coordinates of the POI on heart surface, at least 4 crystals are necessary. The position information of the the crystal attached next to the POI relative to the origin can be calculated using geometric triangulation method [27].

Approximating the Sonomicrometry error by a Gaussian noise distribution with a constant standard deviation, σ, will be the simplest approach to generate a measurement model. A Gaussian noise distribution will sufficient enough to capture the basic uncertainties in the sensor system and it will provide a convenient way to filter and clean the noisy measurements. Constructing the measurement model by a Gaussian noise distribution will allow to implement Bayes Filtering by Extended Kalman Filter Algorithm.

Although the Gaussian noise distribution will be sufficient enough to model noise in sensor measurements, in some cases a more detailed measurement model is needed to capture the uncertainties. Such a model will be presented next.

5.4.2 Sonomicrometry Measurement Model

In order to generate the measurement model, first, the Euclidean distances between the crystal on the heart and each base crystal are computed. Then, the differences, *innovations*, between the actual measurements obtained by sonomicrometer channels and these computations are calculated.

$$\alpha_t = z_t - \hat{z}_t \tag{5.14}$$

With the innovation obtained from the measurements and the computed data is shown in Equation 5.14, the measurement probability can be rewritten as:

$$p(\alpha_t) = p(z_t|x_t) \tag{5.15}$$

The next step is to generate the normalized histogram of these innovations in order to visualize the amount of error between the measurements and computations. The generated model incorporates two types of measurement errors, which are essential to capture all possible source of error: small measurement noise and random unexplained noise.

The major source of the small measurement noise is the crystal geometry as mentioned in Section 5.4.1. Since the ultrasound wave is broadcast and received by the leading edge of the piezoelectric crystal surface, the originally measured distance may not be the true distance between the geometric centers of respective crystals. For example, if the crystals are oriented edge-to-edge, then diameter of the piezoelectric disc, which surrounds the crystals, should be subtracted from the initially measured distance. Therefore these piezoelectric discs add error to the sonomicrometry measurements [10].

This small measurement noise can be characterized by subtracting the $2mm$ diameter of the piezoelectric crystal initially during the computation of the distances between the location of the each base crystal and the location of the crystal attached on the heart. This calculation will improve the overall noise model by initially getting rid of the deflection bias due to the crystal geometry.

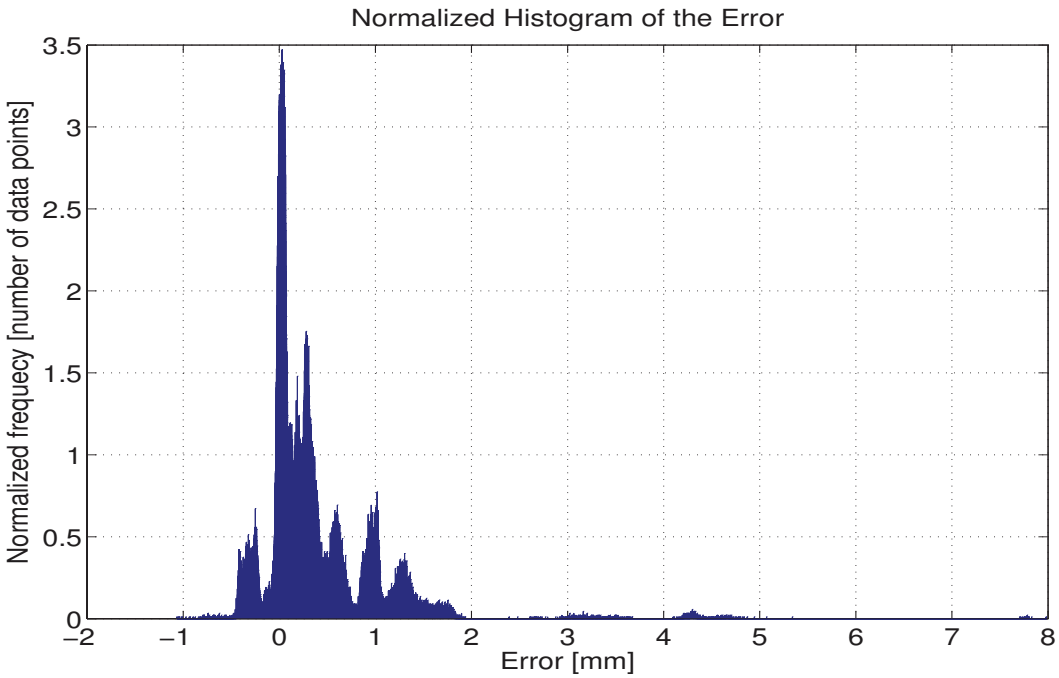

Figure 5.4: Normalized histogram of the error between the sensor measurements and actual data.

The histogram shown in Figure 5.4 will provide the essential information to generate the noise model of the sensor system. It can be observed that most of the error is distributed around *0 mm* with significant bumps exist from *-0.5 mm* to *2 mm*. Outside this range, the amount of observed error is relatively small.

This error can be approximated by a narrow Gaussian noise distribution, $p_{hit}(z_t|x_t)$. The mean, μ, and covariance, σ, of this is distribution is determined according to the frequency of the error values in this range. The resulting Gaussian distribution has has a $\mu = 0.3$ and a $\sigma = 0.15$.

The remaining component of the generated model, random unexplained noise, is describing the ultrasound echoes of the Sonomicrometry sensor system. To keep things simple, the effects of ultrasound echoes will be modeled using a uniform distribution, $p_{rand}(z_t|x_t)$, spread over the entire error range seen in Figure 5.4.

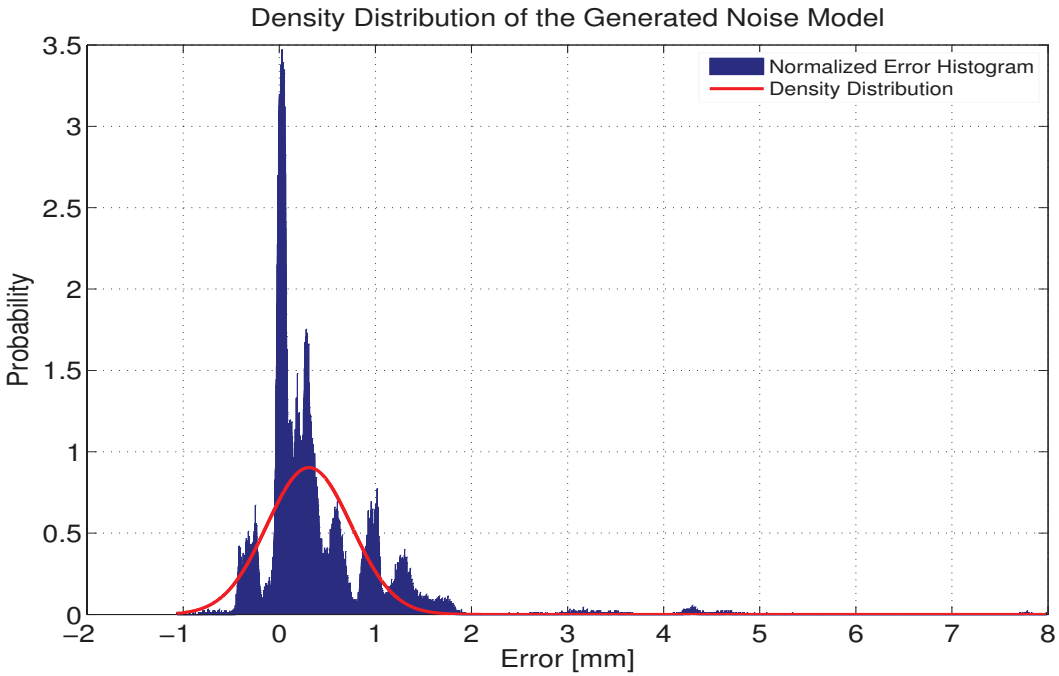

Figure 5.5: Density distribution of the noise model superimposed on the normalized error histogram

These two different distributions are now mixed by a weighted average, defined by the parameters w_{hit} and w_{rand} with $w_{hit} + w_{rand} = 1$. The weights are determined according to the frequency of the error values for the corresponding distributions. The Equation 5.16 gives the resulting distribution.

$$p(z_t|x_t) = \begin{pmatrix} w_{hit} \\ w_{rand} \end{pmatrix}^T \cdot \begin{pmatrix} p_{hit}(z_t|x_t) \\ p_{rand}(z_t|x_t) \end{pmatrix} \tag{5.16}$$

The density resulting from this linear combination of the individual densities is shown in Figure 5.5. It can be noticed that the basic characteristics of both basic models are still present in this combined density.

Now with the necessary components to implement Bayes filter, the motion model,

$p(x_t|u_t, x_{t-1})$, and measurement model, $p(z_t|x_t)$, are generated, two different Bayes filtering algorithms will be presented next.

5.5 Extended Kalman Filter Algorithm

Extended Kalman Filter (EKF) is a variation of Gaussian Filters. Gaussian techniques based on the same the basic idea that beliefs are represented by multivariate normal distributions, which is shown in Equation 5.17.

$$p(x) = det(2\pi\Sigma)^{\frac{-1}{2}} \exp\{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\} \tag{5.17}$$

.

This density over the variable x is characterized by two sets of parameters: which are the mean and the covariance. The Kalman filter represents beliefs by these moments. At time t, the belief $bel(x_t)$ is represented by the mean, μ_t and the covariance Σ_t.

The input of the Kalman filter is the belief at time $t-1$, is represented by μ_{t-1} and Σ_{t-1}. To update these parameters, Kalman filter require the control u_t and the measurement z_t. The output is the belief at time t which is represented by μ_t and the covariance Σ_t [25].

The Kalman filter is implemented for a restricted class of problems with linear state transitions and linear measurements. Since the arguments of the measurement probability, $p(z_t|x_t)$, is not linear in this study (see Equation 5.12), the linearity assumption is violated and so Kalman filter algorithm is not applicable in its current form.

The EKF is a variation of Kalman Filter which is modified to handle nonlinear cases. EKF calculates an approximation to the true belief. It represents this approximation by a Gaussian. Specifically, the belief $bel(x_t)$ is represented by the mean, μ_t and the covariance Σ_t. Thus, the EKF inherits from the Kalman filter the basic belief representation, but it differs in that this belief is only approximate, not exact as was the case in Kalman filters.

In EKF the state transition probability and the measurement probabilities are governed by nonlinear functions g and h, respectively:

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{5.18}$$

$$z_t = h(x_t) + \delta_t \tag{5.19}$$

where ε_t and δ_t describe the process noise and measurement noise respectively. The process noise ε_t is of the same dimension with the state vector x_t. It is a zero mean Gaussian distribution with covariance R_t. Similarly, the measurement noise δ_t is of the same dimension with the measurement vector z_t. It is a zero mean Gaussian distribution with covariance Q_t. The nonlinear functions g and h are approximated by linearization via $TaylorExpansion$.

Now, the EKF algorithm is depicted in Algorithm 2 [25]:

Algorithm 2 Extended Kalman Filter Algorithm

1: Extended Kalman_Filter$(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$
2: $\bar{\mu}_t = g(u_t, \mu_{t-1})$
3: $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$
4: $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
7: **return** (μ_t, Σ_t)

The nonlinear measurement update equation, $h(\bar{\mu}_t)$, in Line 5 of Algorithm 2 is already given in Equation 5.13. H_t is the *Jacobian* of the $h(\bar{\mu}_t)$. It is obtained by the linearization of $h(\bar{\mu}_t)$ via Taylor series expansion around $\bar{\mu}_t$. H_t is given by:

$$H_t = h^{'}(\bar{\mu}_t) \tag{5.20}$$

$$H_t = \begin{pmatrix} \frac{\partial h_1}{\partial \bar{\mu}_x} & \frac{\partial h_1}{\partial \bar{\mu}_y} & \frac{\partial h_1}{\partial \bar{\mu}_z} \\ \frac{\partial h_2}{\partial \bar{\mu}_x} & \frac{\partial h_2}{\partial \bar{\mu}_y} & \frac{\partial h_2}{\partial \bar{\mu}_z} \\ & \vdots & \\ \frac{\partial h_m}{\partial \bar{\mu}_x} & \frac{\partial h_m}{\partial \bar{\mu}_y} & \frac{\partial h_m}{\partial \bar{\mu}_z} \end{pmatrix} \tag{5.21}$$

where m is the dimension of the measurement vector z.

On the other hand, the nonlinear state transition equation, $g(u_t, \mu_{t-1})$, in Line 2 will be defined differently for the two motion models, Brownian motion and harmonics motion, presented in Section 5.3.

In the first case, Brownian Motion, state update equation is $g(u_t, \mu_{t-1})$ will be linear which is already presented in Equation 5.1. Thus the Jacobian, G_t of the linearization of $g(u_t, \mu_{t-1})$ is of the form:

$$G_t = I \tag{5.22}$$

where I is the identity matrix with $I \in \mathbb{R}^n$ and n is the dimension of the state x.

With these motion and measurement models EKF algorithm for Brownian motion model will be:

In the second case, harmonic motion model is utilized. For this case the state update equation is given in Equation 5.8. Likewise the Brownian Motion Model state

Algorithm 3 Extended Kalman Filter Algorithm with Brownian Motion Model

1: Brownian_EKF($\mu_{t-1}, \Sigma_{t-1}, z_t$)
2: $\bar{\mu}_t = \mu_{t-1}$
3: $\bar{\Sigma}_t = \Sigma_{t-1} + R_t$
4: $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
7: **return** (μ_t, Σ_t)

update Jacobian, $G_t = I$ and the EKF algorithm for Harmonic motion is given as:

Algorithm 4 Extended Kalman Filter Algorithm with Harmonic Motion Model

1: Harmonics_EKF($\mu_{t-1}, \Sigma_{t-1}, \Delta u_t, z_t$)
2: $\bar{\mu}_t = \mu_{t-1} + \Delta u_t$
3: $\bar{\Sigma}_t = \Sigma_{t-1} + R_t$
4: $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
6: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
7: **return** (μ_t, Σ_t)

The purpose of using EKF algorithm lies in its simplicity and its computational efficiency. It is computationally extremely tractable since it represents the belief distribution by a multivariate Gaussian distribution.

On the other hand, representing the posterior, $p(z|x)$, by a Gaussian has important ramifications. Especially, Gaussians are unimodal, in other words they posses a single maximum. Although such a posterior is characteristic of many tracking problems in robotics, in which the posterior is focused around the true state with a small margin of uncertainty, gaussian posteriors are a poor match for many global estimation problems. The reason is in many global estimation problems, distinct hypotheses exist, each of which forming its own mode in the posterior [25].

In the next section a nonparametric approach for implementing the Bayes Filter is presented, which will overcome this single hypothesis ramification.

5.6 Particle Filter Algorithm

Particle Filter is another approach to Bayes filter implementation and it represents the posterior distribution, $p(z|x)$, by a finite number of samples unlike the Extended Kalman Filter which relies on finite functional form and represents the posterior by a multivariate Gaussian distribution. For this purpose, particle filters regarded as nonparametric filters [25].

The samples, *particles*, which represent the posterior distribution are denoted as:

$$X_t := x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]} \tag{5.23}$$

where M represents the total number of particles in particle set X_t. Each particle $x_t^{[m]}$ in this set with $1 \leq m \leq M$ represents a single hypothesis for the possible true state at time t. The key idea of particle filters is to approximate and represent the belief distribution $bel(x_t)$ by a finite number of random particles X_t.

Likewise the Extended Kalman Filter algorithm the belief, $bel(x_t)$, at time t is computed from the the belief, $bel(x_{t-1})$, at the previous time step $t-1$. The following algorithm depicts the basic Particle Filter Algorithm [25].

In Algorithm 5 inputs are the particle set at the previous time step $t-1$, $\bar{X}_{t-1}$, control input at time t, u_t and measurement at time t, z_t. Two essential steps of the Bayes filtering state transition distribution, $p(x_t|u_t, x_{t-1})$, and measurement distribution $p(z_t|x_t)$ are presented in Lines 4 and 5 respectively.

State transition is implemented by sampling from this distribution, $p(x_t|u_t, x_{t-1})$. For this purpose, a hypothetical state x_t is generated from the particle set X_{t-1} based on the particle x_{t-1} by employing the control input u_t. By generating M particles

Algorithm 5 Particle Filter Algorithm

1: Particle_Filter($\mu_{t-1}, X_{t-1}, u_t, z_t$)
2: $\bar{X}_t = X_t = \emptyset$
3: $\eta = 0$
4: **for** $m = 1 \rightarrow M$ **do**
5: $\quad sample\ x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]})$
6: $\quad w_t^{[m]} = p(z_t|x_t^{[m]})$
7: $\quad \eta = \eta + w_t^{[m]}$
8: $\quad \bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
9: **end for**
10: **for** $m = 1 \rightarrow M$ **do**
11: $\quad w_t^{[m]} = w_t^{[m]}/\eta$
12: **end for**
13: **for** $m = 1 \rightarrow M$ **do**
14: $\quad draw\ x_t^{[i]} from\ \bar{X}_t\ with \propto w_t^{[i]}$
15: $\quad add\ x_t^{[i]} to\ X_t$
16: **end for**
17: **return** X_t

in this way, the prediction belief distribution $\overline{bel}(x_t)$ is represented. It is important to note that the M particles are generated independently from each other.

In order to incorporate the measurement, z_t, into particle set important factor, $w_t^{[m]}$ is calculated by using the measurement probability: $w_t^{[m]} = p(z_t|x_t^{[m]})$. The importance factor $w_t^{[m]}$ represents the weight of the particle $x_t^{[m]}$ and these weighted particles approximates the belief distribution at time t, $bel(x_t)$.

The most essential part of the particle filter algorithm is shown in *Lines 12-16*, the *resampling*. In the resampling process M particles are drawn with replacement from the temporary particle set $\bar{X}_t$ in which the particles are distributed relative to $\overline{bel}(x_t)$. Probability of drawing a sample is proportional to the normalized weight, $w_t^{[i]}$, of each particle $x_t^{[i]}$. The normalization of the weights $w_t^{[i]}$ are carried out in *Lines 9 - 11*.

The key idea of the resampling process is incorporating the effect normalized importance factors into the particle set. As a result of resampling, the particle set $\bar{X}_t$ is transformed into another particle set X_t of the same size which contains samples according to the distribution $bel(x_t) = \eta p(z_t|x_t^{[m]})\overline{bel}(x_t)$.

However the resampling process arouses an important problem regarding the complete representation of the original density with the generated particles. This problem is presented next.

5.6.1 Sampling Variance

A significant issue about the particle filter algorithm is the sampling variance. The statistics obtained by the samples such as mean and variance differ from the statistics of the original density distribution from which these samples are drawn. This variation is called as sampling variance [25].

The resampling step of the Algorithm 5 is the major cause of this increase sampling variance which might result with the incomplete representation of the original density with the generated particles. The repetitive resampling of the particles will cause decrease in the diversity of the particles in the final particle set, X_t, which represents the posterior belief distribution, $bel(x_t)$.

This problem can be solved with *low variance sampling*. In the low variance sampling, particles are generated via a sequential sampling process unlike the basic particle filter algorithm, which follows an independent sampling process (see Algorithm 5). This algorithm is given in Algorithm 6 [25].

For this purpose, initially a r between 0 and M^{-1} is chosen. Then, by repeatedly adding M^{-1} to r the particles can be selected. A unique particle is selected by the following formula;

Algorithm 6 Low Variance Sampler

1: Low Variance Sampler(X_t, W_t)
2: $\bar{X}_t = \emptyset$
3: $r = rand(0, M^{-1})$
4: $c = w_t^{[1]}$
5: $i = 1$
6: **for** $m = 1 \rightarrow M$ **do**
7: $\quad U = r + (m-1) \cdot M^{-1}$
8: $\quad$ **while** $U > c$ **do**
9: $\quad\quad i = i + 1$
10: $\quad\quad c = c + w_t^{[i]}$
11: $\quad$ **end while**
12: $\quad add\ x_t^{[i]} to\ \bar{X}_t$
13: **end for**
14: **return** $\bar{X}_t$

$$i = \underset{j}{\operatorname{argmin}} \sum_{m=1}^{j} w_t^{[m]} \geq U \tag{5.24}$$

where U is any number between 0 and 1. There are three advantages of the low-variance sampler. First, the sample space is sampled in a more systematic way as opposed to independent selection. Second, if all samples have identical weights the collection of particles will be the same after re-sampling. Third, the algorithm has a lower complexity $O(M)$ as opposed to the complexity of independent selection $O(M \log M)$ [25].

With the low variance sampler shown above the particle filter algorithm for the harmonic motion model is given below.

In Algorithm 7, Σ_r (see Equation 5.9) is the covariance of the three dimensional remaining motion shown in Equation 5.10 and k in *Line 8* is the number of measurements. $v_t^{[j]}$ is an auxiliary variable which represents the probability of the the

Algorithm 7 Particle Filter Algorithm for Harmonic Motion Model

1: Harmonics_Particle_Filter($\mu_{t-1}, X_{t-1}, \Delta u_t, z_t$)
2: $\bar{X}_t = X_t = \emptyset$
3: $\eta = 0$
4: **for** $m = 1 \rightarrow M$ **do**
5: $\quad sample\ r_t^{[m]} \sim N(0, \Sigma_r)$
6: $\quad \bar{\mu}_t^{[m]} = \mu_{t-1} + \Delta u_t + r_t^{[m]}$
7: $\quad w_t^{[m]} = 1$
8: $\quad$ **for** $j = 1 \rightarrow k$ **do**
9: $\quad\quad \hat{z}_j^{[m]} = ||\bar{\mu}_t^{[m]} - q_j||$
10: $\quad\quad \alpha_j^{[m]} = z_t - \hat{z}_j^{[m]}$
11: $\quad\quad v_t^{[j]} = p(\alpha_j^{[m]})$
12: $\quad\quad w_t^{[m]} = w_t^{[m]} \cdot v_t^{[j]}$
13: $\quad$ **end for**
14: $\quad \eta = \eta + w_t^{[m]}$
15: $\quad \bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
16: **end for**
17: **for** $m = 1 \rightarrow M$ **do**
18: $\quad w_t^{[m]} = w_t^{[m]} / \eta$
19: **end for**
20: $r = rand(0, M^{-1})$
21: $c = w_t^{[1]}$
22: $i = 1$
23: **for** $m = 1 \rightarrow M$ **do**
24: $\quad U = r + (m-1) \cdot M^{-1}$
25: $\quad$ **while** $U > c$ **do**
26: $\quad\quad i = i + 1$
27: $\quad\quad c = c + w_t^{[i]}$
28: $\quad$ **end while**
29: $\quad add\ x_t^{[i]} to\ X_t$
30: **end for**
31: $\mu_t = \mathbb{E}[X_t]$
32: **return** (μ_t, X_t)

innovation particle $\alpha_j^{[m]}$. For each particle, this probability is directly computed from the innovation measurement model presented in Section 5.4.

The Algorithm 7 initially computes the prediction belief distribution, $\overline{bel}(x_t)$ which is represented by the temporary particle set $\bar{X}_t$. $\bar{X}_t$ is generated by initially sampling from the remaining motion. Then the posterior belief distribution, $bel(x_t)$, represented by particle set X_t, is obtained by resampling with low variance sampler. Finally, the expected position of the POI on heart surface is computed by taking the weighted expectation of the particles in X_t.

For the Brownian motion model *Lines 5-6* of the Algorithm 7 is updated in the following way:

Algorithm 8 Brownian Motion Model for the Particle Filter Algorithm

1: $sample\ r_t^{[m]} \sim N(0, \Sigma)$
2: $\bar{\mu}_t^{[m]} = \mu_{t-1} + r_t^{[m]}$

where Σ is the covariance of the Brownian motion as shown in Equation 5.3.

Chapter 6

Evaluation of the Probabilistic Algorithms

In this chapter, we comprehensively evaluate the performance of the probabilistic algorithms that are presented in the previous chapter. Initially, we test the algorithms on a 70 s of Sonomicrometry data. After the algorithms are verified with this independent data, we applied these algorithms to a 60 sec heart motion data and show that they effectively filter the noisy heart motion data. Finally, one-step estimates obtained from the generalized adaptive predictor (see Section 3.3) algorithm is employed as the motion model of the applied probabilistic approach. Thus, two distinctive studies presented in this thesis are linked to each other.

6.1 Verification with the Independent Sensor Data

In order to collect the 70 s of Sonomicrometry data, first the rigid plastic sensor base, shown in Figure 5.3, was placed in a rectangular glass tank which is filled with distilled water. The 6^{th}, *moving*, crystal of the Sonomicrometry system was attached to the tip of the PHANToM manipulator. Then, the tip of the manipulator was hanged down into the water bath with the piezoelectric crystal surface is looking towards the rigid base. Finally, a custom circular motion was applied to the PHANToM device and ten incoming measurements from these six channels were recorded.

Together with these ten measurements, the *3D* position coordinates of the moving crystal were also computed. This position is calculated by the SonoVIEW Software (Sonometrics Inc., Ontario, Canada) via geometric triangulation method, which is mentioned the previous chapter. The *3D* position of the moving crystal was also computed by PHANToM manipulator. The encoder positions on the PHANToM were recorded and these positions were transformed into end effector positions.

The *3D* position coordinates of the moving crystal computed by the Sonomicrometer and PHANToM manipulator can be seen Figure 6.1.

The simulations for all of the four algorithms;
i) EKF with Brownian motion model
ii) EKF with Harmonic motion model
iii) Particle Filter with Brownian motion model
iv) Particle Filter with Harmonic motion model
were tested with the same 70 s of data.

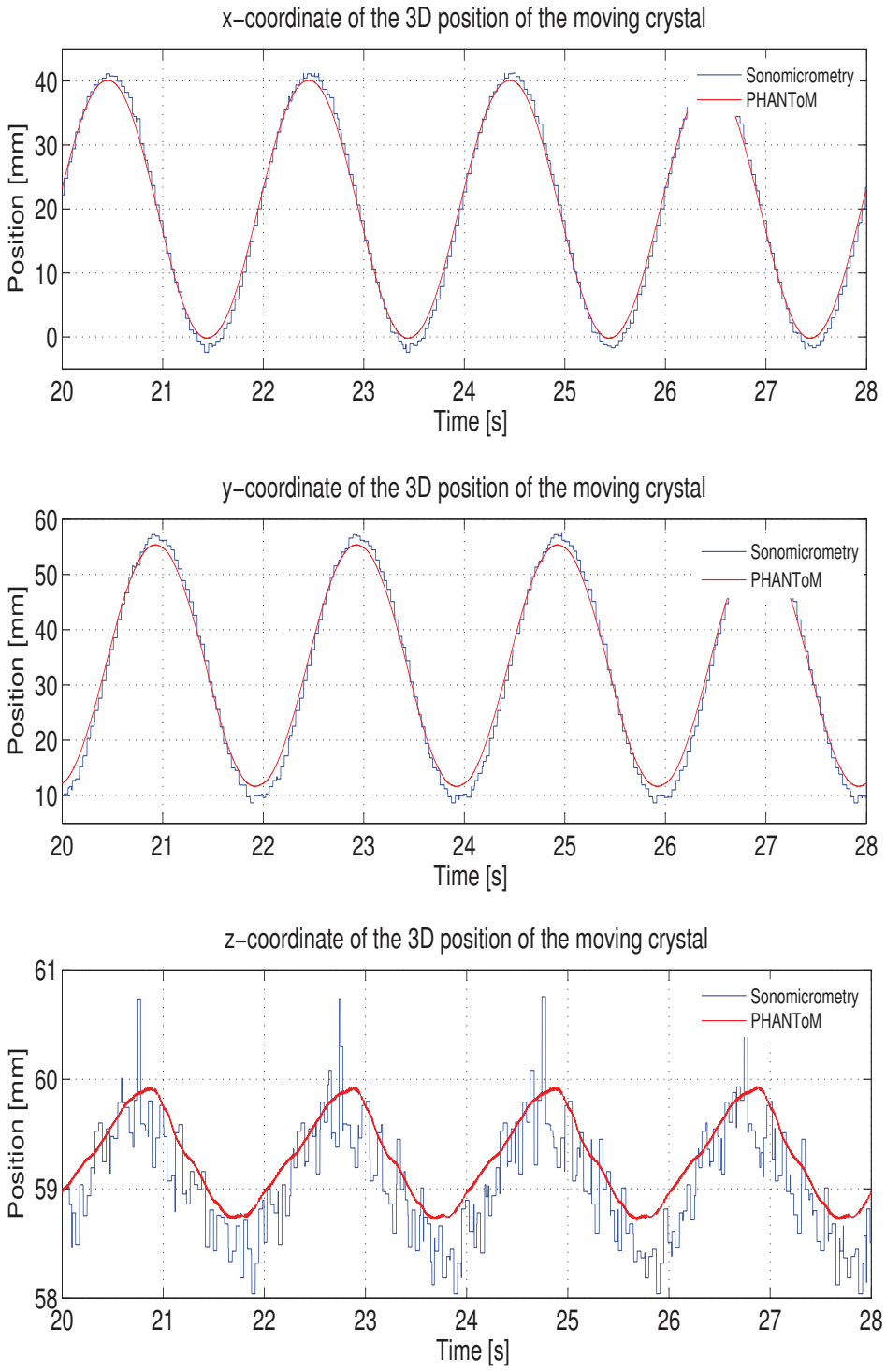

Figure 6.1: The *3D* position coordinates of the moving crystal computed by Sonomicrometry and PHANToM.

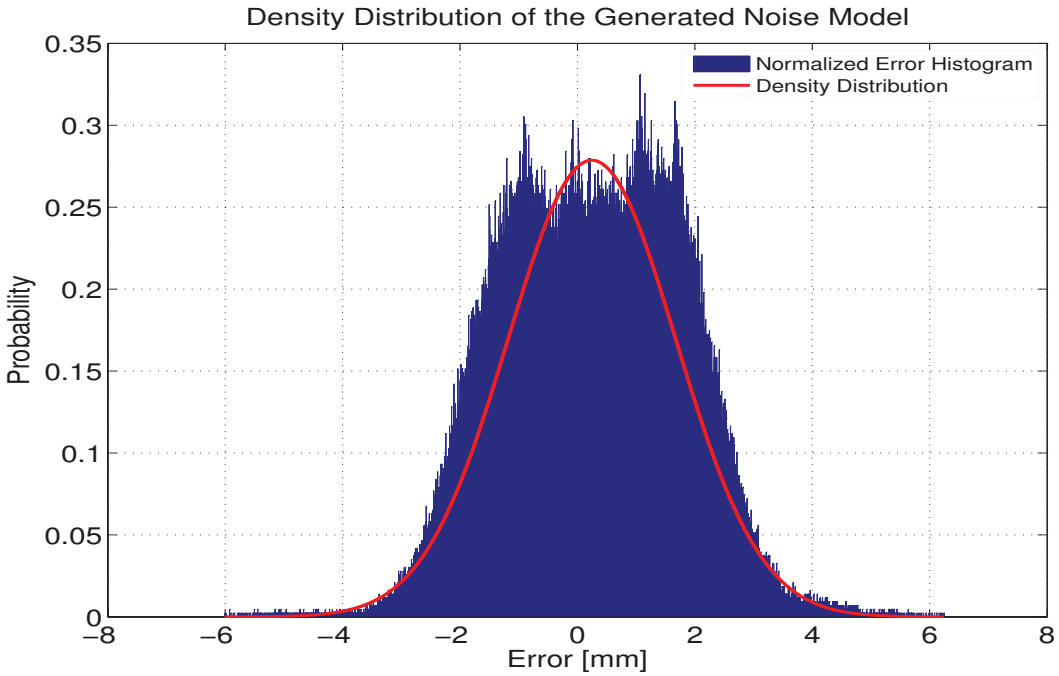

Figure 6.2: Density distribution of the noise model superimposed on the normalized error histogram.

In the simulations, the end effector positions of the PHANToM manipulator represents a 'perfect' localization, providing a performance base for the probabilistic algorithms during the filtering of the channel measurements and localization of the moving crystal. The performance metric that was used for the evaluation of the algorithms is the three dimensional Root Mean Square Error (RMSE) between the localized position of the moving crystal by probabilistic algorithms and the position computed by Sonomicrometry. This error was compared with the *3D* RMSE between the position computed by Sonomicrometry and the position computed by PHANToM to show the performance of the algorithms.

For the particle filter algorithms, the measurement model is shown in Figure 6.2. 500 particles were used for the representation of distributions and simulations performed 10 times with worst result is presented.

Table 6.1: Simulation Results for a 70 sec long Sonomicrometer Data: RMSE for the Probabilistic Localization Algorithms.

Localization Results	RMS Position Error [mm]
Baseline Error between Sonomicrometry and PHANToM	2.4130
EKF with Brownian motion model	2.9860
EKF with Harmonic motion model	2.5129
Particle Filter with Brownian motion model	2.6454
Particle Filter with Harmonic motion model	2.5042

Localization results of the probabilistic algorithms in terms of *3D* RMSE for the 70 sec Sonomicrometry data are shown in Table 6.1. Second row shows the baseline RMS error between the Sonomicrometry sensor system and PHANToM manipulator. Remaining rows show the RMS error for between the Sonomicrometry sensor system and the four distinct probabilistic algorithms.

Filtering results for a certain channel measurement by the and EKF with harmonic motion model and the localized *3D* position of the moving crystal is shown in Figure 6.3 and Figure 6.4 respectively. These two figures are provided in order to exemplify and demonstrate how the algorithms are filtering the incoming Sonomicrometry sensor data and compute the position of the moving crystal with these filtered measurements.

When the RMS error results of the localization algortihms are compared with each other, particle filter algorithm with harmonic motion model yields the best results. The effect of the harmonic motion model in the performance of localization can be also observed. Both the EKF and particle filter algorithms provide better results with harmonic motion model than Brownian motion model. These results together

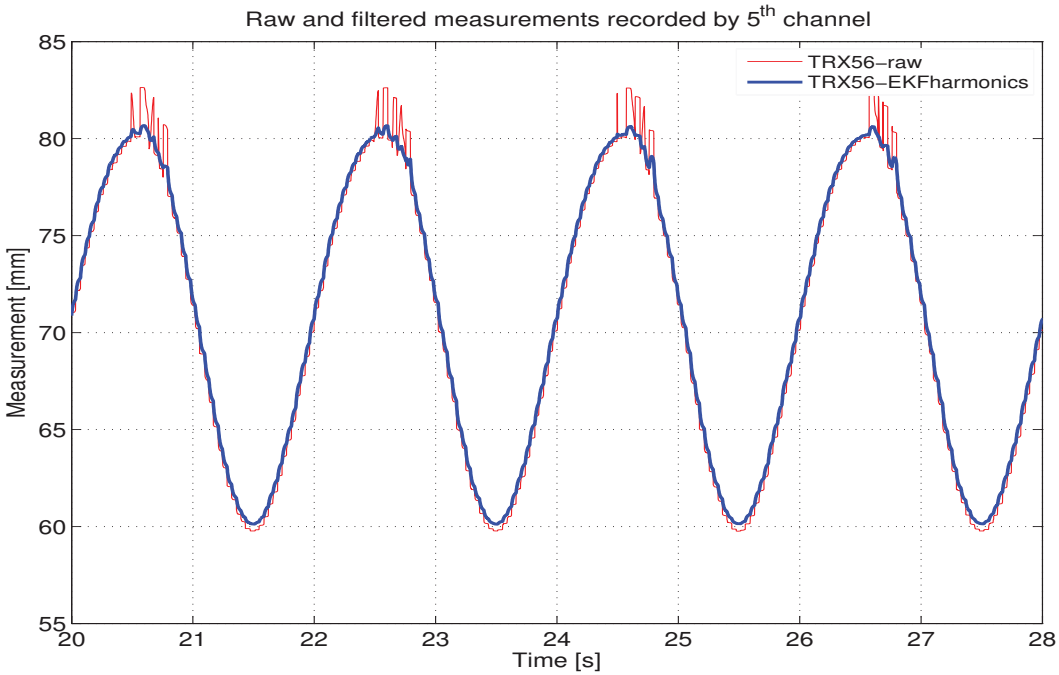

Figure 6.3: Raw and filtered sensor data by EKF with harmonic motion model gathered by 5^{th} channel are presented.

with the Figures 6.3 and 6.4 show that the presented probabilistic algorithms filter the incoming noisy measurements effectively and yields accurate localization of the moving crystal.

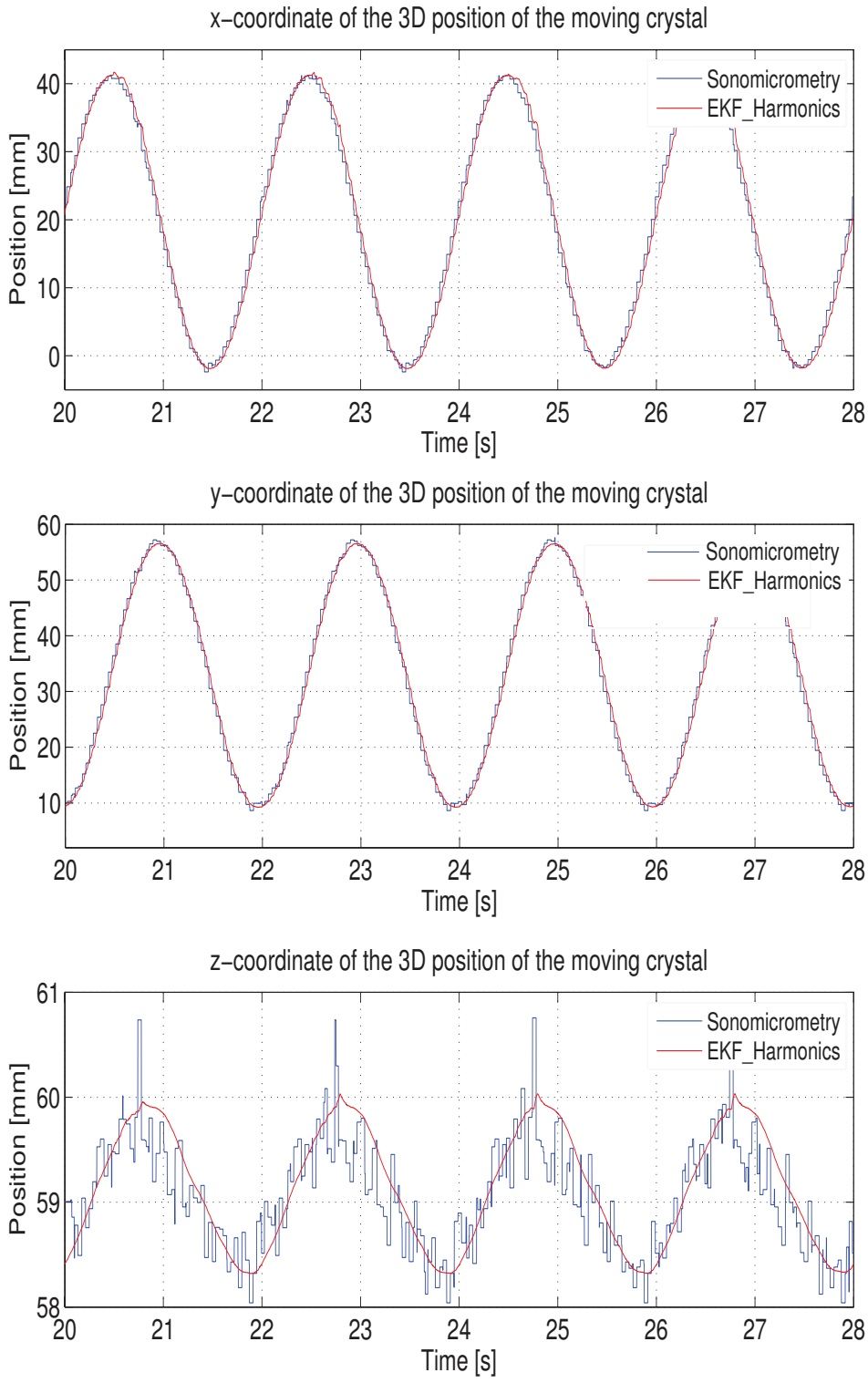

Figure 6.4: The *3D* position coordinates of the moving crystal computed by Sonomicrometry and localized by EKF with harmonic motion model.

6.2 Application to the Heart Motion Data

In this section, the results regarding the application of the localization algorithms to a 60 sec constant heart rate motion data are presented. It is already shown in the previous section that these algorithms are working properly. Since the initial goal of this study is to develop an online filtering mechanism for incoming sensor data, it is essential to test the presented algorithms with the prerecorded heart motion data. The harmonic motion model which comprises 2^{nd} order harmonic approximation of the heart motion data was presented in Section 5.3.2.

The incoming channel measurements during an *in-vivo* data collection are very noisy. These noisy measurements causes sign shifting in the geometric triangulation method, which is used by Sonomicrometer to compute the *3D* position of the POI, and thus yields incorrect results for the *3D* coordinates of the POI (see Figure 6.5).

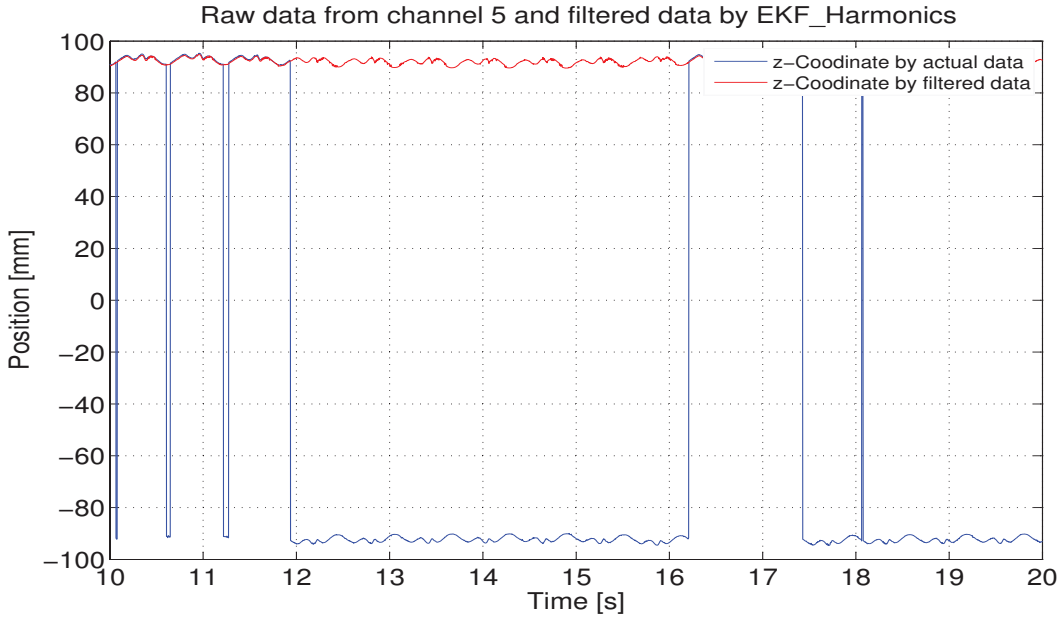

Figure 6.5: z-Coordinate of the *3D* Position of POI computed separately by raw measurements and offline filtered measurements.

Table 6.2: Simulation Results for a 60 sec long Constant Heart Rate Motion Data: RMSE for the Probabilistic Localization Algorithms.

Localization Results	RMS Position Error [mm]
EKF with Brownian motion model	1.3859
EKF with Harmonic motion model	1.3271
Particle Filter with Brownian motion model	1.1484
Particle Filter with Harmonic motion model	1.0506

For this purpose an updated performance metric is required to evaluate the algorithms. The performance metric that was used for the evaluation of the algorithms is the *3D* RMS error between the localized position of the POI on heart surface by probabilistic algorithms and *3D* position computed via geometric triangulation method by offline filtered measurements. Again, 500 particles were used for the representation of probability distributions and the simulations performed 10 times with worst result is presented. Localization results of the probabilistic algorithms in terms of *3D* RMSE for the 60 sec heart motion data are shown in Table 6.2.

The filtered heart motion data from channel 5 by the EKF with harmonic motion model and the localized *3D* position of the POI is shown in Figure 6.6 and Figure 6.7 respectively. Again, these two figures are provided in order demonstrate and show that how the algorithms are filtering the heart motion data and localize *3D* position of the POI on heart surface.

When the RMS error results of the localization algorithms are compared with each other, particle filter algorithm with harmonic motion model yields the best results. Both the EKF and particle filter algorithms provide better results with harmonic motion model than Brownian motion model. These results together with the Figures 6.6 and 6.7 show that the presented probabilistic algorithms filter the

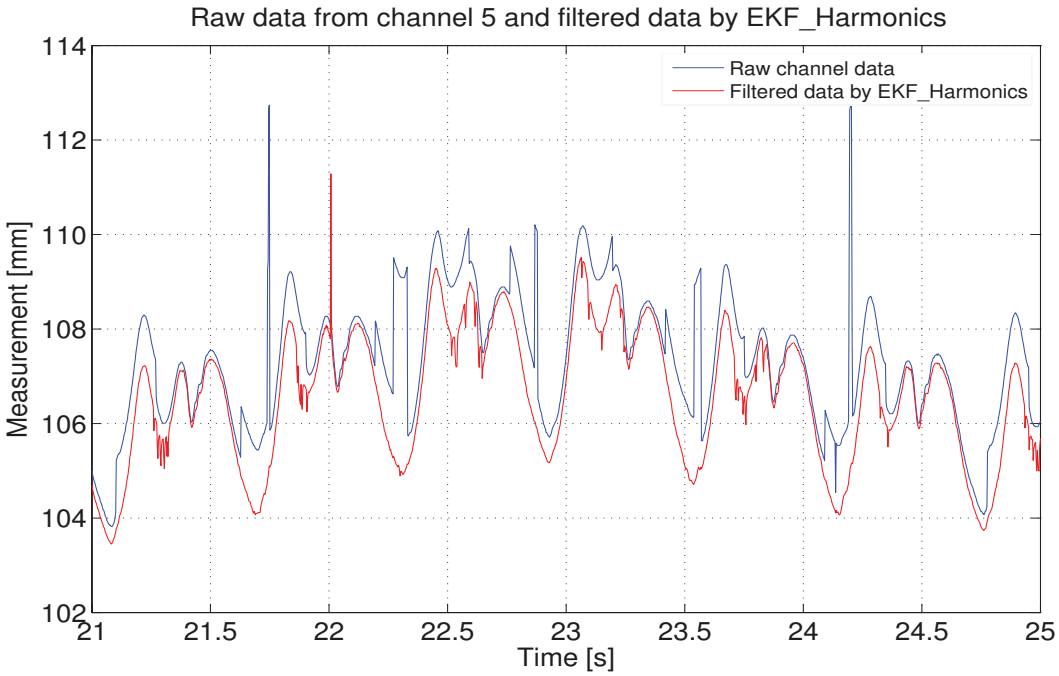

Figure 6.6: Raw and filtered sensor data by EKF with harmonic motion model gathered by 5^{th} channel are presented.

heart motion effectively and yields accurate localization of the POI.

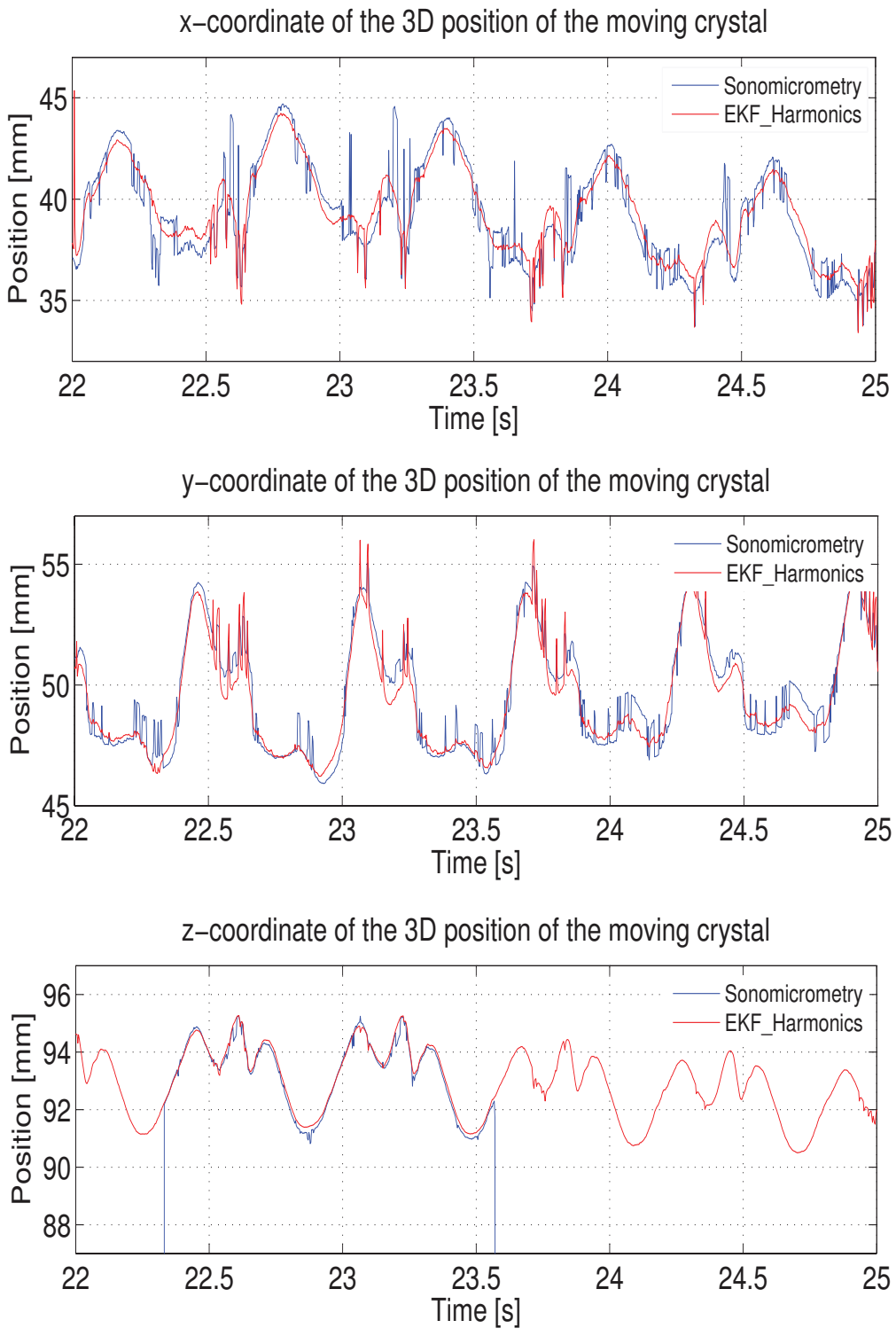

Figure 6.7: The *3D* position coordinates of the POI computed by Sonomicrometry and localized by EKF with harmonic motion model.

6.3 Generalized Adaptive Predictor as Motion Model

In this part of this chapter, two distinct studies presented in this thesis are linked. In this manner the generalized adaptive predictor is used to generate motion model for the EKF and particle filter algorithms.

By employing the one-step estimates generated by the generalized adaptive filter, the corresponding motion model for the algorithms is constructed in the following way.

At any time t, the update equation of the generalized motion model can be written as:

$$\Delta gaf_t = gaf_{t+1} - gaf_t \tag{6.1}$$

where gaf_{t+1} is the one step prediction generated by the adaptive predictor and gaf_t is the x_t, which is the current *3D* position of the heart data. Then, the state update equation in Line 2 of EKF Algorithm (Algorithm 2) is updated as;

$$\mu_{t+1} = \mu_t + \Delta gaf_t \tag{6.2}$$

and consecutively in Line 5 of Particle Filter Algorithm (Algorithm 5) is updated as;

$$\bar{\mu}_t^{[m]} = \mu_{t-1} + \Delta gaf_t + r_t^{[m]} \tag{6.3}$$

Table 6.3: Simulation Results for a 60 sec long Constant Heart Rate Motion Data: RMSE for the Probabilistic Localization Algorithms.

Localization Results	RMS Position Error [mm]
EKF with Generalized motion model	1.2257
Particle Filter with Generalized motion model	1.0876

where $r_t^{[m]}$ is sampled from the distribution $r_t^{[m]} \sim N(0, \Sigma_{gaf})$ with Σ_{gaf} is the covariance obtained from the three dimensional remaining motion between one-step estimates and current position of POI (see Equation 6.1).

Localization results of the EKF and Particle Filter algorithms for the 60 sec heart motion data are shown in Table 6.3.

From Table 6.3 it can be seen that integrating the one-step estimates obtained from the generalized adaptive filter results in considerable improvement in the performance of EKF algorithm. Although it enhances the performance of the particle filter algorithm when compared to Brownian motion model, it yields slightly worse results than the harmonic motion model.

6.4 Discussion of the Results

At this point, it would be informative to discuss the algorithms and their performance presented in the previous sections.

In this chapter, we initially evaluate the presented EKF and particle filter algorithms on a Sonomicrometry sensor data. The presented results in Table 6.1 and

Figures 6.3 and 6.4 show that the algorithms are working properly and yields reasonable localization results for a custom generated circular motion.

After verifying the implementation of the algorithms, we applied them to a constant rate heart motion data. Although the channel measurements for this study comprise more noise (see Figure 6.6) than the previous study, both of the EKF and particle filter algorithms filter and clean these noisy channels effectively and accurately localize *3D* position of the crystal attached near the POI (see Figure 6.7 and Table 6.2).

In all of the studies, harmonic motion model yields better results than the Brownian motion model which is no surprise since the harmonic approximation includes significant information about the state of the environment.

Finally, we integrate the generalized adaptive predictor presented in Section 3.3 into the motion model of the localization algorithms. The one-step estimates generated by the adaptive predictor improved the performance of the EKF algorithm precisely (see Table 6.3).

When the two localization algorithms are compared in terms of computational efficiency, EKF algorithm is much more computationally tractable. The duration for EKF algorithm to process the 60 sec heart motion data is on the order of $2-3$ sec whereas the particle filter algorithm, with 500 particles, process the same data in approximately 6000 sec. Therefore, despite particle filter provides better filtering of the noisy measurements, EKF is much more computationally efficiently and can be easily implemented.

Chapter 7

Conclusion

In this thesis, two distinct studies are presented.

In the first study, a one-step and a generalized estimator for predicting the horizon estimate for the model predictive controller are presented. Three different sets of experiments are performed with constant heart rate and varying heart rate to evaluate the performance of the proposed algorithms.

The experimental RMS error on the order of $0.160 - 0.350$ mm obtained using the generalized estimator described in this thesis represents a significant improvement in tracking performance compared to earlier studies. These results show that the estimation of future POI motion is no longer the bottleneck in the heartbeat motion tracking since the necessary amount of RMS tracking error in the order of 100-250 μm for the POI on the heart surface is achieved to perform precise operations.

Furthermore the results showed that if the heart statistics change, then adaptive predictors are able to adjust to these changes sufficiently quickly and yield good tracking results. However, if the statistics change abruptly and significantly, such as in an arrhythmia, actions must be taken to minimize the effect of poor predictions.

Another way to improve tracking quality is to incorporate other types of data into the estimation scheme. One such possibility is to include the electrocardiogram (ECG) signal into the observations. In this way, the predictor is able to use the electrical signals that activate heart contraction in order to improve the prediction as in [4]. This may improve performance during heart contractions, when rapid POI motion occurs.

In the second part of the thesis a probabilistic approach to filter and clean the measurements obtained from the Sonomicrometry sensor system, which is used for measuring heart motion in this research.

The implementation of the two probabilistic algorithms, EKF and Particle Filter, is verified with a custom generated circular motion and then the performance of these algorithms are evaluated on a heart motion data. Subsequently, the generalized predictor presented in the first part of the thesis integrated to improve the performance of the algorithms.

The *3D* RMS Position errors on the order of $1.000 - 1.400$ mm obtained using the generalized estimator described in this thesis represents a sufficient localization performance for the POI on the heart surface. When the computational efficiency and ease of implementation of the algorithms are taken into account, these result show that EKF algorithm can be further developed and integrated as an online filtering mechanism for the sensor measurements.

In order to emerge the second part of the thesis as a publishable study and developed into an online process, a secondary sensor system is required. Such a sensor system will provide online independent measurements which will be used as the baseline data during the filtering of the Sonomicrometry measurements.

In the on-going development of this setup, two high-speed cameras are currently employed for measuring heart motion and providing the independent measurement

for filtering algorithms. Currently, a successful calibration between the Sonomicrometry sensor system, high-speed camera system and PHANToM manipulator is trying to be achieved before using these localization algorithms in an *in-vivo* experiment.

Bibliography

[1] A. L. Trejos, S. E. Salcudean, F. Sassani, and S. Lichtenstein, "On the feasibility of a moving support for surgery on the beating heart," in *Proc. of Medical Image Computing and Computer-Assisted Interventions (MICCAI)*, Cambridge, UK, September 1999, pp. 1088–1097.

[2] M. F. Newman, J. L. Kirchner, B. Phillips-Bute, V. Gaver, H. Grocott, R. H. Jones *et al.*, "Longitudinal assessment of neurocognitive function after coronary-artery bypass surgery," *New England Journal of Medicine*, vol. 344, no. 6, pp. 395–402, February 2001.

[3] M. Lemma, A. Mangini, A. Redaelli, and F. Acocella, "Do cardiac stabilizers really stabilize? experimental quantitative analysis of mechanical stabilization," *Interactive CardioVascular and Thoracic Surgery*, no. 4, pp. 222–226, March 2005.

[4] O. Bebek and M. C. Cavusoglu, "Intelligent control algorithms for robotic-assisted beating heart surgery," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 468–480, June 2007.

[5] M. C. Cavusoglu, J. Rotella, W. S. Newman, S. Choi, J. Ustin, and S. S. Sastry, "Control algorithms for active relative motion cancelling for robotic assisted off-pump coronary artery bypass graft surgery," in *Proc. of the 12th International*

Conference on Advanced Robotics (ICAR), Seattle, WA, USA, July 2005, pp. 431–436.

[6] J. Rotella, "Predictive tracking of quasi periodic signals for active relative motion cancellation in robotic assisted coronary artery bypass graft surgery," M.S. Thesis, Case Western Reserve University, Cleveland, OH, USA, August 2004, Advisor: M. Cenk Cavusoglu.

[7] R. Ginhoux, J. A. Gangloff, M. F. DeMathelin, L. Soler, J. Leroy, M. M. A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 67–79, February 2005.

[8] T. J. Franke, O. Bebek, and M. C. Cavusoglu, "Improved prediction of heart motion using an adaptive filter for robot assisted beating heart surgery," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, October-November 2007, pp. 509–515.

[9] T. J. Franke, M. C. Cavusoglu, and O. Bebek, "Prediction of heartbeat motion with a generalized adaptive filter," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 2008, pp. 2916–2921.

[10] M. B. Ratcliffe, K. B. Gupta, J. T. Streicher, E. B. Savage, D. K. Bogen, and J. L. H. Edmunds, "Use of sonomicrometry and multidimensional scaling to determine the three-dimensional coordinates of multiple cardiac locations: Feasibility and initial implementation," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 6, pp. 587–598, June 1995.

[11] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat synchronization for robotic cardiac surgery," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, Seoul, Korea, May 2001, pp. 2014–2019.

[12] A. Thakral, J. Wallace, D. Tomlin, N. Seth, and N. V. Thakor, "Surgical motion adaptive robotic technology (S.M.A.R.T.): Taking the motion out of physiological motion," in *Proc. of 4th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Utrecht, The Netherlands, October 2001, pp. 317–325.

[13] M. Groeger, T. Ortmaier, W. Sepp, and G. Hirzinger, "Tracking local motion on the beating heart," in *Proc. of the SPIE Medical Imaging Conference*, vol. 4681 of SPIE, San Diego, CA, USA, February 2002, pp. 233–241.

[14] M. L. Koransky, M. L. Tavana, A. Yamaguchi, and R. Robbins, "Quantification of mechanical stabilization for the performance of offpump coronary artery surgery," in *Proc. of the Meeting of the International Society for Minimally Invasive Cardiac Surgery (ISMICS)*, Munich, Germany, June 2001, (Abstract).

[15] R. Richa, A. P. L. Bo, and P. Poignet, "Motion prediction for tracking the beating heart," in *Proc. of the Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*, Vancouver, British Columbia, Canada, August 2008, pp. 3261–3264.

[16] R. Richa, P. Poignet, and C. Liu, "Deformable motion tracking of the heart surface," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 2008, pp. 3997–4003.

[17] T. Ortmaier, M. Groeger, D. H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 10, pp. 1729–1740, October 2005.

[18] T. Bader, A. Wiedemann, K. Roberts, and U. D. Hanebeck, "Model-based motion estimation of elastic surfaces for minimally invasive cardiac surgery," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007, pp. 2261–2266.

[19] S. G. Yuen, P. M. Novotny, and R. D. Howe, "Quasiperiodic predictive filtering for robot-assisted beating heart surgery," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, May 2008, pp. 3875–3880.

[20] S. G. Yuen, D. T. Kettler, P. M. Novotny, R. D. Plowes, and R. D. Howe, "Robotic motion compensation for beating heart intracardiac surgery," *International Journal of Robotics Research*, vol. 28, no. 10, pp. 1355–1372, October 2009.

[21] G. Kalogeros, *Personal Communication*, Sonometrics Corporation, Ontario, Canada.

[22] W. Wei, *Time Series Analysis - Univariate and Multivariate Methods*. Addison-Wesley Publishing Company, New York, 1990.

[23] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, New Jersey: Prentice Hall, 2001.

[24] M. C. Cavusoglu, D. Feygin, and F. Tendick, "A critical study of the mechanical and electrical properties of the phantom haptic interface and improvements for highperformance control," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 6, pp. 555–568, 2002.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 1st ed. Cambridge, MA, USA: The MIT Press, 2006.

[26] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford University Press Inc., New York, 2001.

[27] O. Bebek, "Robotic-assisted beating heart surgery," Ph.D. Dissertation, Case Western Reserve University, Cleveland, OH, USA, May 2008, Advisor: M. Cenk Cavusoglu.

Appendix A

Sonomicrometer Least Squares Equations

For each group of four crystals, the position of the crystal attached on the point of interest (POI) on heart surface is calculated relative to the three crystals fixed on the base. First crystal on the base is selected as the origin of the coordinate frame. Second crystal forms the x-axis together with the crystal at the origin, and the third crystal forms the xy-plane together with the x-axis. Four different coordinate frames, (α, β, γ, δ), from five base crystals can be constructed in the following way:

$$\begin{array}{llll} \alpha_1\text{: } 1-2-3 & \beta_1\text{: } 1-3-4 & \gamma_1\text{: } 2-3-4 & \delta_1\text{: } 4-5-1 \\ \alpha_2\text{: } 1-2-4 & \beta_2\text{: } 1-3-5 & \gamma_2\text{: } 2-3-5 & \delta_2\text{: } 4-5-2 \\ \alpha_3\text{: } 1-2-5 & & & \delta_3\text{: } 4-5-3 \end{array} \tag{A.1}$$

Let the position of the fourth crystal be $P(x, y, z)$; x $= [x \;\; y]^T$ denote the xy-coordinates with respect to the coordinate frame; and d be the measured distance between two crystals. If x is known, the distance of the fourth crystal from the base frame, z, can be computed from raw crystal measurements using trigonometry. The

position of the fourth crystal can be computed as:

$$\underbrace{\begin{bmatrix} 2x_2 & 2y_2 \\ 2x_3 & 2y_3 \end{bmatrix}}_{\mathrm{A}_1} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\mathrm{x}_\alpha} = \underbrace{\begin{bmatrix} d_1^2 - d_2^2 + x_2^2 + y_2^2 \\ d_1^2 - d_3^2 + x_3^2 + y_3^2 \end{bmatrix}}_{\mathrm{b}_1} \tag{A.2}$$

$$z = +\sqrt{d_3^2 - x^2 - y^2} \tag{A.3}$$

Then for n possible measurements, there are n linear equations.

$$\begin{aligned} \mathrm{A}_1\mathrm{x}_1 &= \mathrm{b}_1 \\ \mathrm{A}_2\mathrm{x}_2 &= \mathrm{b}_2 \\ &\vdots \\ \mathrm{A}_\mathrm{n}\mathrm{x}_\mathrm{n} &= \mathrm{b}_\mathrm{n} \end{aligned} \tag{A.4}$$

Similar solutions can be grouped under the same coordinate frame such as:

$$x_n = \begin{cases} x_\alpha, \ n = 1, 2, 3 \\ x_\beta, \ n = 4, 5 \\ x_\gamma, \ n = 6, 7 \\ x_\delta, \ n = 8, 9, 10 \end{cases}$$

$$\begin{array}{llll} \mathrm{A}_1\mathrm{x}_\alpha = \mathrm{b}_1 & \mathrm{A}_4\mathrm{x}_\beta = \mathrm{b}_4 & \mathrm{A}_6\mathrm{x}_\gamma = \mathrm{b}_6 & \mathrm{A}_8\mathrm{x}_\delta = \mathrm{b}_8 \\ \mathrm{A}_2\mathrm{x}_\alpha = \mathrm{b}_2 & \mathrm{A}_5\mathrm{x}_\beta = \mathrm{b}_5 & \mathrm{A}_7\mathrm{x}_\gamma = \mathrm{b}_7 & \mathrm{A}_9\mathrm{x}_\delta = \mathrm{b}_9 \\ \mathrm{A}_3\mathrm{x}_\alpha = \mathrm{b}_3 & & & \mathrm{A}_{10}\mathrm{x}_\delta = \mathrm{b}_{10} \end{array} \tag{A.5}$$

Let g be a homogeneous transformation matrix:

$$\mathrm{g} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0\,0 & 1 \end{bmatrix} \tag{A.6}$$

where position vector $\mathbf{p}$ describes translations with respect to a reference frame, and orientation matrix $\mathbf{R}$ describes rotations. Then inverse transformation matrix of g is:

$$\mathrm{g}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{p} \\ 0\,0 & 1 \end{bmatrix} \tag{A.7}$$

Using the transformation matrices, all of the measurements can be expressed under the same coordinate frame.

$$\begin{aligned}
\mathrm{g}_{\alpha\beta}\begin{bmatrix} \mathrm{x}_\beta \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathrm{x}_\beta \\ 1 \end{bmatrix} = \mathrm{g}_{\alpha\beta}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} \\
\mathrm{g}_{\alpha\gamma}\begin{bmatrix} \mathrm{x}_\gamma \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathrm{x}_\gamma \\ 1 \end{bmatrix} = \mathrm{g}_{\alpha\gamma}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} \\
\mathrm{g}_{\alpha\delta}\begin{bmatrix} \mathrm{x}_\delta \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} \mathrm{x}_\delta \\ 1 \end{bmatrix} = \mathrm{g}_{\alpha\delta}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix}
\end{aligned} \tag{A.8}$$

Lets define a truncated transformation $\tilde{\mathrm{g}}$ and its identity in the following way.

$$\tilde{\mathrm{g}}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{p} \end{bmatrix} \tag{A.9}$$

$$\tilde{\mathrm{I}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{A.10}$$

$$
\begin{aligned}
&\mathrm{A}_n\mathrm{x}_\alpha = \mathrm{b}_n \longrightarrow \mathrm{A}_n\tilde{\mathrm{I}}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_n, \qquad n = 1,2,3 \\
&\mathrm{A}_n\mathrm{x}_\beta = \mathrm{b}_n \longrightarrow \mathrm{A}_n\mathrm{g}_{\alpha\beta}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_n, \qquad n = 4,5 \\
&\mathrm{A}_n\mathrm{x}_\gamma = \mathrm{b}_n \longrightarrow \mathrm{A}_n\mathrm{g}_{\alpha\gamma}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_n, \qquad n = 6,7 \\
&\mathrm{A}_n\mathrm{x}_\delta = \mathrm{b}_n \longrightarrow \mathrm{A}_n\mathrm{g}_{\alpha\delta}^{-1}\begin{bmatrix} \mathrm{x}_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_n, \qquad n = 8,9,10
\end{aligned}
\tag{A.11}
$$

Then, all equations can be combined into a single linear equation:

$$
\begin{aligned}
&\mathrm{A}_1\tilde{\mathrm{I}}\mathrm{x}_\alpha = \mathrm{b}_1 \\
&\qquad\vdots \\
&\mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1}\begin{bmatrix} x_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_4 \\
&\qquad\vdots \\
&\mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1}\begin{bmatrix} x_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_6 \\
&\qquad\vdots \\
&\mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1}\begin{bmatrix} x_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_8 \\
&\qquad\vdots \\
&\mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1}\begin{bmatrix} x_\alpha \\ 1 \end{bmatrix} = \mathrm{b}_{10}
\end{aligned}
\quad \equiv \quad
\begin{bmatrix} \mathrm{A}_1\tilde{\mathrm{I}} \\ \vdots \\ \mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1} \\ \vdots \\ \mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1} \\ \vdots \\ \mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1} \\ \vdots \\ \mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1} \end{bmatrix}
\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} =
\begin{bmatrix} \mathrm{b}_1 \\ \vdots \\ \mathrm{b}_4 \\ \vdots \\ \mathrm{b}_6 \\ \vdots \\ \mathrm{b}_8 \\ \vdots \\ \mathrm{b}_{10} \end{bmatrix}
\tag{A.12}
$$

$$\left(\begin{bmatrix} \mathrm{A}_1\tilde{\mathrm{I}} \\ \vdots \\ \mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1} \\ \vdots \\ \mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1} \\ \vdots \\ \mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1} \\ \vdots \\ \mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1} \end{bmatrix} \tilde{\mathrm{I}}_1 \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \mathrm{A}_1\tilde{\mathrm{I}} \\ \vdots \\ \mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1} \\ \vdots \\ \mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1} \\ \vdots \\ \mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1} \\ \vdots \\ \mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1} \end{bmatrix} \tilde{\mathrm{I}}_2 \right) = \begin{bmatrix} \mathrm{b}_1 \\ \vdots \\ \mathrm{b}_4 \\ \vdots \\ \mathrm{b}_6 \\ \vdots \\ \mathrm{b}_8 \\ \vdots \\ \mathrm{b}_{10} \end{bmatrix} \tag{A.13}$$

where,

$$\tilde{\mathrm{I}}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \; and \; \tilde{\mathrm{I}}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{A.14}$$

$$\underbrace{\left(\begin{bmatrix} \mathrm{A}_1\tilde{\mathrm{I}} \\ \vdots \\ \mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1} \\ \vdots \\ \mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1} \\ \vdots \\ \mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1} \\ \vdots \\ \mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1} \end{bmatrix} \tilde{\mathrm{I}}_1 \right)}_{\mathrm{A}} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathrm{b}_1 \\ \vdots \\ \mathrm{b}_4 \\ \vdots \\ \mathrm{b}_6 \\ \vdots \\ \mathrm{b}_8 \\ \vdots \\ \mathrm{b}_{10} \end{bmatrix} - \begin{bmatrix} \mathrm{A}_1\tilde{\mathrm{I}} \\ \vdots \\ \mathrm{A}_4\mathrm{g}_{\alpha\beta}^{-1} \\ \vdots \\ \mathrm{A}_6\mathrm{g}_{\alpha\gamma}^{-1} \\ \vdots \\ \mathrm{A}_8\mathrm{g}_{\alpha\delta}^{-1} \\ \vdots \\ \mathrm{A}_{10}\mathrm{g}_{\alpha\delta}^{-1} \end{bmatrix} \tilde{\mathrm{I}}_2}_{\mathrm{b}} \tag{A.15}$$

Using linear least squares, a solution to $\mathrm{Ax} = \mathrm{b}$ can be found as:

$$\mathrm{x} = (\mathrm{A}^T\mathrm{A})^{-1}\mathrm{A}^T\mathrm{b} \tag{A.16}$$