KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Robust Estimation and Compensation of Semi-Periodic Physiological Movements for Robot-Assisted Surgery

Florian De Clercq

Thesis submitted for the degree of
Master of Science in Mechanical
Engineering

Thesis supervisors:
Prof. dr. ir. Emmanuel Vander Poorten
Prof. dr. dr. Wouter Oosterlinck
Prof. dr. ir. Joris De Schutter

Assessors:
Prof. dr. ir. Harry van Lenthe
Dr. ir. Erwin Aertbeliën

Mentor:
Dr. ir. Gianni Borghesan

Academic year 2019 – 2020

Preface

My gratitude goes towards my promoters and mentor, who made this subject available and allowed me to work on it. I would like to explicitly thank my mentor, Gianni, for his advice and guidance. My sincerest appreciation goes towards all the professors, assistants and staff at our faculty who consistently ensure the high quality of our education. I would like to thank the jury for reading the text.

On a more personal note, I thank my parents for providing the means that enabled these past five years. Special credit goes to my grandparents, Paul and Gisella, for their unconditioned support.

Florian De Clercq

Contents

Abstract

This work designs a motion-compensation system for physiological semi-periodic movements in the context of robot-assisted surgical procedures. Special attention is paid to coronary artery bypass graftings or 'bypass surgery'. The overall objective is to render the presence of any incision larger than those necessary for keyhole procedures obsolete. This is made possible provided such motion-compensation system.

The thesis commences by elaborating the context and explaining why the employment of the intended system leads to less-invasive surgery. Then the design of the system is tackled. The concept of state estimation is introduced along with specific algorithms, after which research related to the heart is presented. This leads to the establishment of assumptions that are exploited throughout the work. The introductory part is concluded with the presentation of an approximative model of the to-be-compensated heart motion. Attention is also paid to the incorporation of respiratory movement. This model is confluenced with the state estimation concepts towards the implementation and validation of several state estimators. Upon comparison, appropriate selections are made for further use. In this context, an extended Kalman filter (EKF) provides the best results. Additionally, a filter is designed with the purpose of providing an on-line differentiation of an arbitrary signal. It is compared to more traditional methods and shows superior performance. The chapter on filter implementation is concluded by a part on the exploitation of the predictive information of an electrocardiogram (ECG) signal through several methods. Eventually, to this end, sensor fusion is performed by an unscented Kalman filter (UKF). This method is proven to be the most beneficial when it comes to exploiting the ECG signal. The aforementioned systems are separately integrated along with two distinct control strategies. The behaviour of these systems is assessed, in nominal as well as in anomalous circumstances. The systems have advantages and drawbacks but both reflect promising performance.

Eventually, the expansion of the state estimator towards a realistic three-dimensional system is discussed. An approximative semispherical surface model is introduced and a UKF is built which maintains it in real-time. The system is further improved and methods for on-line consistency assessment are described and implemented. This final part is concluded by a series of simulations that evaluate the designed system.

Samenvatting

Deze thesis behandelt het ontwerp van een bewegingscompensatiesysteem voor quasi-periodieke fysiologische bewegingen, bestemd voor gebruik in robotchirurgische context. Bijzondere aandacht is besteed aan bypass-hartoperaties. Het vooropgestelde doel is het overbodig maken van incisies met een grotere omvang dan deze aanwezig bij kijkoperaties.

Het werk vangt aan met een contextuele schets en een uiteenzetting die het verband tussen het beoogde systeem en minimaal-invasieve chirurgie verheldert. Daarop volgt het systeemontwerp. Eerst wordt het idee achter toestandschatting geïntroduceerd in parallel met specifieke algoritmen. Vervolgens worden enkele cardiologische onderzoeksresultaten aangekaart. Deze leiden tot vaststellingen en aannames die doorheen het werk worden geëxploiteerd. Dit inleidend deel wordt afgesloten met de voorstelling van een hartverplaatsingsmodel. Aandacht wordt daarbij ook besteed aan de inclusie van de ademhalingsverplaatsing. Dit model wordt verenigd met de uiteengezette toestandschattingsmethoden om tot de implementatie en validatie van verschillende toestandsschatters te komen. Deze worden vergeleken en gepaste keuzes worden gemaakt voor verder gebruik. In deze context levert de extended Kalman filter (EKF) de beste resultaten. Verder wordt ook een filter ontworpen die moet voorzien in een 'on-line' afleiding van een arbitrair signaal. Hij wordt vergeleken met meer conventionele methoden en vertoont betere prestaties. Het hoofdstuk over filterimplementatie wordt afgesloten door een deel over het aanwenden van het elektrocardiogramsignaal (ECG) en het op verschillende manieren exploiteren van de voorspellende informatie die het bevat. Het summum hiervan is een 'sensor fusion', uitgevoerd door een unscented Kalman filter (UKF). Dit blijkt de meest performante methode. De vermelde systemen worden tezamen geïmplementeerd in twee disjuncte controlesystemen. Het gedrag van deze systemen wordt getoetst in nominale en afwijkende omstandigheden. De systemen hebben voor- en nadelen maar leveren beiden veelbelovende resultaten.

Uiteindelijk wordt de uitbreiding van de huidige toestandschatter naar een meer realistisch driedimensionaal systeem besproken. Een benaderend sferisch oppervlak wordt geïntroduceerd, vergezeld door een UKF die het in 'real time' updatet. Het systeem wordt beschreven en 'on-line' consistentiebeoordelingsmethoden van de meting worden toegelicht. Dit laatste deel wordt afgesloten met een reeks simulaties die het ontwerp toetsen.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

BW	bandwidth
CABG	coronary artery bypass grafting
CH	control horizon
CPB	cardiopulmonary bypass
ECG	electrocardiogram
EKF	extended Kalman filter
KF	Kalman filter
LKF	linear Kalman filter
LPF	low-pass filter
LSE	least-square estimator
LTI	linear time-invariant
MHE	moving horizon estimator
MIDCAB	minimally invasive direct coronary artery bypass grafting
MO	measured output
MPC	model predictive control
MSE	mean square error
MT	mini-thoracotomy
MV	manipulated variable
NIS	normalized innovation squared
NTI	non-linear time-invariant
OPCAB	off-pump coronary artery bypass grafting
OR	operation theater
PH	prediction horizon

PID	proportional-integral-derivative
POI	point of interest
PSD	power spectral density
RACAB	robot-assisted coronary artery bypass grafting
RR	respiratory rate
SNIS	summed normalized innovation squared
SSR	state-space representation
TF	transfer function
UKF	unscented Kalman filter
UT	unscented transformation
UZ	universitair ziekenhuis (Dutch for university hospital)

Symbols

State Estimation

k	Count of arbitrary discrete time step
$\mathbf{x}_k$	State vector at time step k
$\mathbf{f}()$	State-space system equations
z_k	Measurement at time step k
$h()$	Measurement equation
R_0	Invariant measurement noise variance
Q_0	Invariant process noise covariance matrix
R_k	Measurement noise variance at time step k
Q_k	Process noise covariance matrix at time step k
A	Linear system equations
C	Linear measurement equations
$\mathbf{w}_k$	Process noise at time step k
v_k	Measurement noise at time step k
$\mathcal{N}_n$	Multivariate distribution of dimension n (no subscript for $n = 1$)
$\hat{\mathbf{x}}_k^-$	A priori (predicted) state estimate at time step k
$\hat{z}_k^-$	A priori (predicted) measurement estimate at time step k
$\hat{P}_k^-$	A priori (predicted) error covariance matrix estimate at time step k
P_k	Maintained error covariance matrix at time step k
y_k	Used on Figure 2.1 instead of z_k
S_k	Innovation covariance matrix
K_k	Kalman gain matrix
I_n	Identity matrix of order n
F_k	Jacobian of system function $\mathbf{f}()$
H_k	Jacobian of measurement function $h()$
X_k	Collection of sigma-points and corresponding weights at time step k
$\mathbf{x}_k^i$	ith sigma-point at time step k
W^j	jth weight of sigma-point at time step k
$\mathbf{x}_k^{-,i}$	ith a priori (predicted) sigma-point at time step k
$\mathrm{Cov}()$	(Cross-)covariance of variable(s)
ν_k	Innovation at time step k $(z_k - h(\hat{\mathbf{x}}_k^-))$
NIS_k	NIS at time step k
SNIS_k	SNIS at time step k
J_{MHE}	Cost function of MHE
w_{meas}	MHE weight for measurement
w_{states}	MHE weight for states
$\mathbf{x}_{k,l}$	State vector for time step l with respect to state vector at time step k
N_{hor}	Horizon maintained by MHE

State Estimation

$\Delta T_{\mathrm{ecg,p}}$	Time interval between ECG peak (QRS) and heart movement peak
ω_0	Heart beat angular frequency
y	Estimated displacement
C_k	Offset at time step k
a_{ik}	Amplitude of ith sine at time step k
ω_i	Angular frequency of ith sine
t	Continuous time
ϕ_i	Phase shift of ith sine
y_r	Actual displacement
Δt	Time step
B	Input matrix of state-space
D	Feedforward matrix of state-space
$\mathbf{y}$	Output vector
u	Input vector (control vector)
ω_k	Heart beat angular frequency at time step k
J	Jacobian
θ_{ik}	Argument of ith sine at time step k

Filter Implementation

$()_g$	Subscript denoting value used for signal generation
c	Arbitrary scaling factor
v_k	Velocity at time step k
x_k	Displacement at time step k
q	Arbitrary velocity process noise covariance scalar
r	Arbitrary measurement noise variance scalar
f_c	Cutoff frequency
T_p	Period of periodic ECG signal
$\phi_{\Delta,k}$	Phase difference between expected peak in heart movement due to ECG signal and current state estimates at time step k
$\mathrm{T}_{\mathrm{int}}$	Magnitude of the time interval in which a maximum is searched
$\mathbf{\Delta T}_{\mathrm{rel}}$	Vector with relative time differences corresponding to $\mathrm{T}_{\mathrm{int}}$
$\Delta T^*_{\mathrm{rel}}$	Relative time difference at location of maximum value in search interval
$\Delta_{\mathrm{ecg},k}$	Expected value of triangular pulse signal at time step k
$R_{0,\mathrm{disp}}$	Invariant measurement noise of displacement measurement
$R_{0,\mathrm{ecg}}$	Invariant measurement noise of ECG pulse ‘measurement’

Robot Control

e	Error signal
$C(s)$	Control function in s-domain
j	Imaginary unit ($j = \sqrt{-1}$)
s	Complex frequency of the Laplace domain ($s = \omega j$)
k_p	Proportionate gain
k_d	Derivative gain
k_i	Integral gain
s_0	Zero in the Laplace domain
K	Proportionate gain $K = k_p$
I	Integral gain $I = k_i$
$Y(s)$	Output in s-domain
$X(s)$	Input (reference) in s-domain
$P(s)$	Plant function in s-domain
$\dot{\hat{x}}_e$	Estimated velocity of environment (in this case the heart velocity)
Δt_{MPC}	Time step maintained by MPC algorithm
N_{PH}	MPC prediction horizon
N_{CH}	MPC control horizon
J_{MPC}	MPC cost function
u	Control signal (i.e. setpoint velocity)
$\dot{u}$	Derivative of control signal (i.e. ‘setpoint’ acceleration)
r_k	Reference displacement at time step k
x_{pj}	Simulated output displacement of internal plant depending on u
x_h	Human operator input displacement
x_e	Environment displacement
$()^m$	Superscript denoting measurement
τ	Time constant of low-pass system
ΔT_{sim}	Execution time of simulation

Three-Dimensional Motion Tracking

$(x_e, y_e, z_e)_k$	Coordinate of robot end effector at time step k
R_s	Radius of semispherical surface
$(x_c, y_c, z_c)_k$	Coordinate of center or semispherical surface at time step k
$\theta_{e,k}$	Polar angle of robot end effector at time step k
$\phi e, k$	Azimuth angle of robot end effector at time step k
$(x, y, z)_{\mathrm{sol}}$	Coordinate which is the solution of geometric equations
$()_{\mathrm{resp}}$	Subscript denoting unit in the context of respiratory movement (as opposed to heart beat movement)
$F()$	Arbitrary function
ϵ_k	Altered ‘NIS’ value at time step k
$\mathrm{tr}()$	Matrix trace operator

Chapter 1

Background and Goals

This chapter introduces the background of the work. It first lays the groundwork on which subsequently the thesis goals and modi operandi are determined.

1.1 Introduction

Robotic surgery is gaining in popularity. Surgical robots are controlled over a distance by a surgeon who is seated at a master console. Here, he moves a pair of joysticks through which motions are recorded and sent as reference input to surgical robot controllers. Ideally, the surgical robot closely follows the motion commands by the surgeon. Today's systems are quite powerful and intuitive to such extent that surgeons report to have the impression as if they were directly operating on the patient (i.e. without intervening robot). Technical developments have for example lead to greater mechanical bandwidth and haptic feedback, in turn resulting in quicker response and more accurate input from the operator.

Delicate tasks can be done with modern robots, like operations on the heart [3]. The robot is helpful here as it enhances precision and ergonomics. One feature that is not fully developed and that could be a big asset for robotic surgery is motion compensation, as beating heart procedures are beneficial for patient's outcome. Alternatively, a cardiopulmonary bypass (CPB) is charged with the patient's heart functioning. Deploying an accurate motion compensation system, the surgeon would have the impression that he is operating on a static heart because the robot would automatically take care of the movements of the beating heart.

1.2 Coronary Artery Bypass Grafting (CABG)

A CABG (pronounced 'cabbage') is commonly known as *bypass surgery* or *heart bypass*. It is a general term used to address a family of surgical procedures where an obstructed coronary artery gets bypassed. Such obstruction prevents the supply of oxygen-rich blood to the heart, with the possibility of eventually leading to severe chest pain, shortness of breath or even a heart attack. Figure 1.1 displays a heart with two obstructed arteries as well as the intended result of the procedure. These

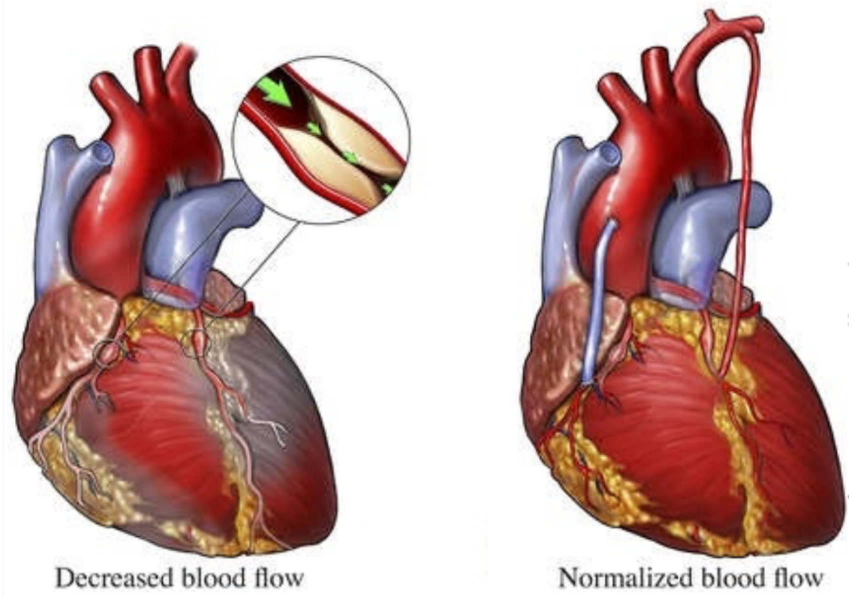


Figure 1.1: Heart with obstructed arteries (left) and after CABG (right) [1]

are often planned in advance but in some cases, they might also be performed in emergency situations e.g. when the patient suffers from cardiac arrest. The CABG is the most performed heart surgery in the world and one of the most common procedures performed during U.S. hospital stays, with over 200 000 cases in 2014 [4, 5].

1.2.1 The Procedure

In general, a CABG involves the following 4 steps. First, a median sternotomy is performed. This procedure provides access to the heart. An incision is made along the sternum (breastbone) after which it is divided, and the chest is opened as shown in Figure 1.2. Next, the movement of the heart is prohibited or minimized in some way, depending on the procedure. In the most conservative scenario, a CPB is used to this end. Here, a machine ("the pump") bypasses the heart and takes over its function. Once these preparations have been successfully executed, an artery is used to bypass the obstruction and is eventually sewn in place. Finally, the sternum of the patient is closed.

An alternative, less invasive method with some pronounced advantages [6] is the off-pump coronary artery bypass grafting (OPCAB), where the CPB is omitted by mechanically stabilizing the heart and clamping it with a device. Further refinements to the OPCAB have resulted in the less invasive minimally invasive direct coronary artery bypass grafting (MIDCAB).

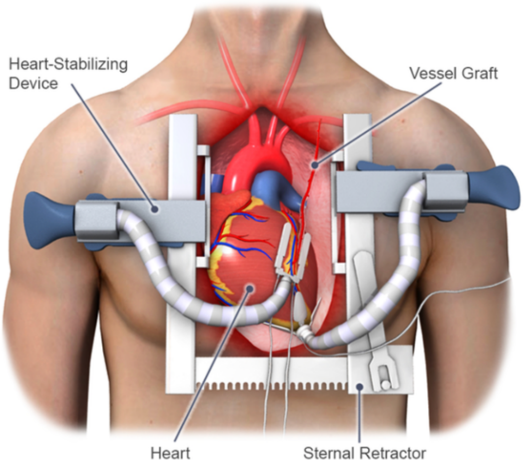


Figure 1.2: Median Sternotomy [2]

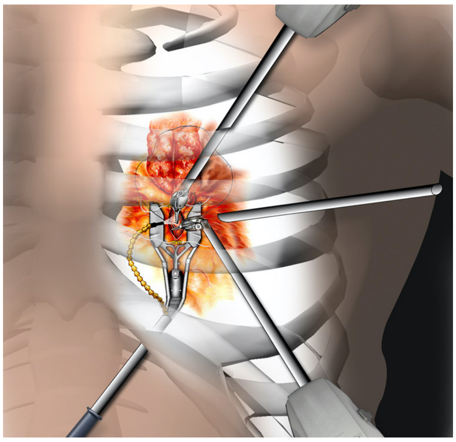


Figure 1.3: RACAB illustration. (Property of Intuitive Surgical Inc.)

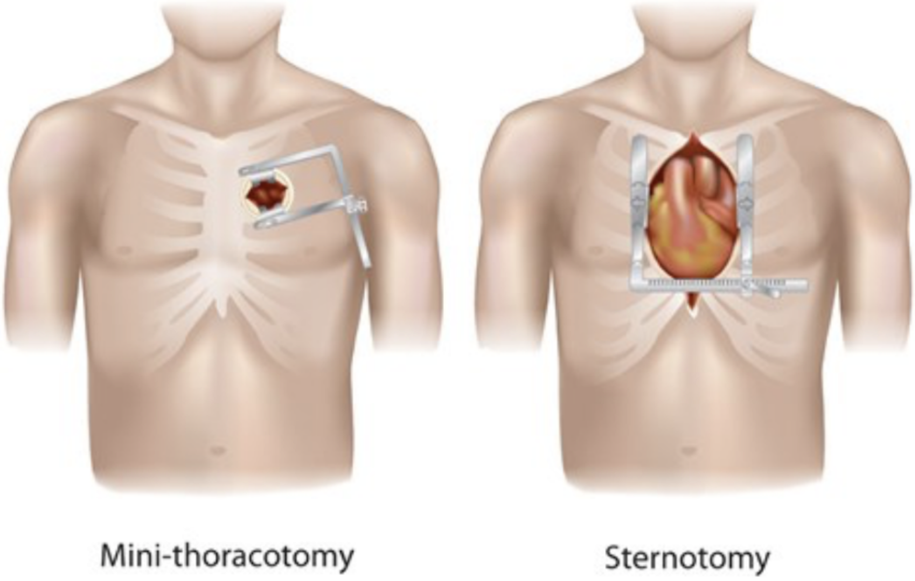


Figure 1.4: comparison of median sternotomy (right) with less invasive mini-thoracotomy (left) [7]

1.2.2 MIDCAB

During a MIDCAB, the median sternotomy is replaced by a mini-thoracotomy (MT) (see Figure 1.4). This is a significantly smaller incision made between the ribs in the area of the heart. Tangible benefits of the MT over the median sternotomy include less pain, faster postoperative recovery, and better cosmetic results. [8] Even though the MT introduces a substantial improvement, it still involves opening up the thorax and cutting through muscle. The resulting procedure is as described below.

First, the MT is performed: an incision between the ribs grants access to the pericardium, a double walled sac containing the heart. After this sac is opened, the operator can directly manipulate the heart. Then a series of further preparations for the bypass is executed. Amongst those are measures to gain further access to parts of the heart, identifying, clipping and clamping arteries and checking for homeostasis. In what is left of the procedure an artery is sewn in place to form the actual bypass.

1.2.3 RACAB

The latest development in the context of the CABG procedure is the increased use of robot assistance. Wenhui Gong et al. [3] have suggested short-term and mid-term outcomes of the RACAB are favourable in comparison with the MIDCAB. Globally, a number of hospitals have started to alter some of their procedures in favour of these robotic systems. Amongst those is the university hospital of the Catholic University of Leuven: UZ Leuven. The added value of robotic surgery becomes clear in the way the MIDCAB is altered.

The first and longest part of the procedure is performed through keyholes in the thorax (the chest). To this end, a Da Vinci robot (Intuitive Inc.) is deployed (Figure 1.3). These holes are used to insert robot end effectors and a camera into the thorax. The setup enables the surgeon to perform a significant part of the aforementioned preparations without the need for an incision. Further advantages of this robotic approach are more precision, stability and visibility in comparison with a manual procedure. Because of the movement of the heart, the last part of the procedure cannot be performed through the robot, as the operator would not be able to control the robot as to maintain a constant gap between the point of interest and the robot end effector. Instead, an MT is made through which the surgeon stitches manually. This incision is reduced in size as compared with the one made during the MIDCAB. The purpose of the MT is twofold: to allow for the placement of a stabilising device, and to grant the surgeon access to the heart.

1.3 Thesis Goals

The currently used setup introduces opportunity for a more extensive deployment of the robot during the RACAB. This thesis aims to design a motion-compensation system for the robot in order to allow the surgeon to perform the entire procedure using the robot, hereby omitting the need for an MT. The required system accurately estimates the motion that is to be compensated and controls the robot with the objective of maintaining a constant distance between the end effectors and camera on one hand, and the point of interest on the other. In this scenario, from the surgeon's perspective, he would seem to be working on a motionless heart. The surgeon's input from the console is superimposed on the motion-compensation.

In order to maintain accuracy, the tracking error of the system should be a fraction of a millimetre, as the artery itself has a section of only a couple of millimetres. Avoiding the MT would yield a series of advantages: significantly faster revalidation, diminished likeliness for complications and the possibility to treat some earlier declined patients [9]. On top of that, during these long procedures, the robot console helps the surgeon to remain focused and work more accurately.

As the range of applications of the intended system is more extensive than the situation described above, and as the system would ideally be applicable to the range of surgical robots that exist today, this work aims to develop a general system without referring to one specific platform.

First, for modelling, prediction and estimation purposes, a one-dimensional model of the heart is developed. This is a continuous task throughout the entire work as the sequential development of the other parts of the system might have implications for it. Next, a suitable state estimator is designed and selected which makes use of this model. The estimations are employed towards a twofold purpose: to filter out measurement noise and improve the control strategy. Then the actual control problem is tackled. Different methods are discussed, implemented and compared. These chapters are supported by simulations. Due to the global COVID-19 crisis in the spring of 2020, the intention of actually implementing the system on a test setup had to be abandoned in favour of a more extensive expansion towards three-dimensional functionality. Eventually, this expansion is discussed.

Chapter 2

State Estimation

This chapter introduces methods and algorithms that aim to provide as accurate as possible information about the current state of a system. To this end, the concepts of Kalman filtering and moving horizon estimation are elaborated and the possibilities to manipulate the filter's behaviour are summarised. The chapter is concluded by discussing the possibility of on-line consistency assessment.

2.1 Introduction

Consider a discrete non-linear time-invariant (NTI) system without control input, described by $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k)$ with corresponding measurement equation $z_k = h(\mathbf{x}_k)$. This notation implies that, for this specific system, $\mathbf{x}_k \in \mathbb{R}^{n \times 1}$ and $z_k \in \mathbb{R}$. These dimensions remain assumed throughout this chapter in correspondence with chapter 3. The *state estimation problem* is the problem of obtaining information about $\mathbf{x}_k$ considering only z_k. This problem corresponds to many real-world engineering problems as usually the internal states $\mathbf{x}_k$ of a system are a priori unknown.

2.2 Kalman Filtering

A Kalman filter (KF) is a recursive Bayesian[1] filter that employs an internal model in order to estimate the states of a system, based on a series of measurements. These are usually distorted by noise. The accuracy of the model is implicitly assumed to be limited in the sense that an amount of process noise is taken into account. Process noise is a measure for the uncertainty of the model states or, in the context of a KF, the inaccuracy of a model. There exists an entire family of KFs, of which a subset is considered in the context of this thesis. The filter assumes process and measurement noise to be additive and Gaussian with measurement noise variance $R_0 \in \mathbb{R}$ and process noise covariance $Q_0 \in \mathbb{R}^{n \times n}$. These values are implicitly presumed to be

---

[1] The Bayesian interpretation relies on probability expressing a degree of belief in an event. The degree of belief may be based on prior knowledge about the event, such as the results of previous experiments, or on personal beliefs about the event.

time varying when their subscript changes: R_k and Q_k. The actual system and measurement equations, where $\mathbf{f}$ and h are known functions, then become:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k \\ z_k &= h(\mathbf{x}_k) + v_k \end{aligned} \tag{2.1}$$

Or for a linear system:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + \mathbf{w}_k \\ z_k &= C\mathbf{x}_k + v_k \end{aligned} \tag{2.2}$$

The distribution of the noises can be represented as shown hereunder.

$$\mathbf{w}_k \sim \mathcal{N}_n(0, Q_k) \tag{2.3}$$

$$v_k \sim \mathcal{N}(0, R_k) \tag{2.4}$$

n in Equation 2.3 is the dimension of the system i.e. the amount of states. $\mathcal{N}_n$ represents the multivariate normal distribution. This distribution is the generalisation of the one-dimensional normal distribution (i.e. Gaussian distribution) to higher dimensions.

2.2.1 In general

Based on the previous estimated state $\hat{\mathbf{x}}_{k-1}$, the KF *predicts* the state vector $\hat{\mathbf{x}}_k^-$ through the internal model and uses it to calculate a predicted measurement $\hat{z}_k^-$[2]. Concomitantly, the error covariance matrix P_k^-, an estimation of the covariance[3] of the system states, is predicted. The predictions are then *assimilated* with the current measurement z_k. From the difference between these values, taking into account the initially set measurement and process noises, the state estimate and error covariance matrix are adjusted resulting in $\hat{\mathbf{x}}_k$ and P_k. These states in turn allow for the calculation of a measurement estimate through $\hat{z}_k = h(\hat{\mathbf{x}}_k)$. A qualitative illustration of the principle is shown in Figure 2.1: the current predicted measurement $\hat{\mathbf{x}}_k^-$ and actual measurement y_k $(= z_k)$ are compared with their (co)variances taken into account. The filter then produces a ‘better’ estimate $\hat{\mathbf{x}}_k$ with corresponding error covariance matrix P_k.

2.2.2 Linear Kalman filter (LKF)

When a linear model describes the system and measurement equation, the filter becomes a linear Kalman filter (LKF). For linear time-invariant (LTI) systems with additive Gaussian process and measurement noise, as depicted in Equation 2.2 and Equation 2.3, the LKF is an optimal state estimator which solves the linear–quadratic–Gaussian control problem (LQG). The LKF algorithm employing these equations is shown in algorithm 1.

---

[2] Note on notation: the $-$ superscript implies a predicted value, $\hat{}$ an estimate and a k subscript refers to a variable in the kth step of the algorithm.

[3] In statistics, covariance is a measure of the joint variability of two random variables. The covariance of n variables is represented by an $n \times n$ matrix. Along the matrix diagonal the variance of each corresponding variable is found.

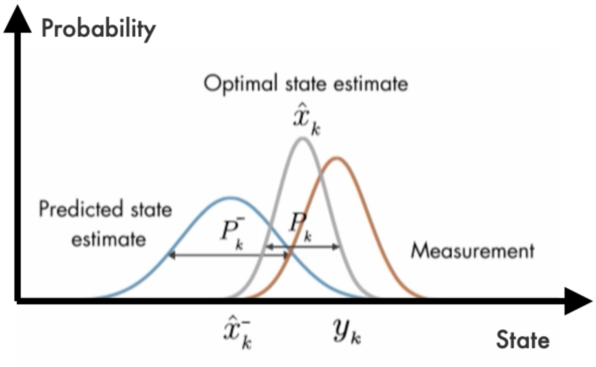


Figure 2.1: Qualitative illustration of KF working principle: the current predicted measurement $\hat{\mathbf{x}}_k^-$ and actual measurement $y_k(=z_k)$ are compared with their (co)variances taken into account. P_k^- is the predicted error covariance matrix, P_k^- the updated one. An optimal output estimation is then made. [10]

Algorithm 1: Time invariant linear Kalman filter

```
Result: Optimal measurement and state estimates $\hat{z}_i$ and $\hat{\mathbf{x}}_i$
initialize $P_0, \mathbf{x}_0, R_0$ and $Q_0$;
while (1) do
    // prediction step
    $\hat{x}_k^- = A\hat{x}_{k-1}$;
    $P_k^- = AP_{k-1}A^T + Q_0$;
    // assimilation (update) step
    $S_k = CP_k^- C^T + R_0$;
    $K_k = P_k^- C^T S_k^{-1}$;
    $\hat{x}_k = \hat{x}_k^- + K_k(z_k - C\hat{x}_k^-)$;
    $P_k = (I - K_k C)P_k^-$;
    // calculate output
    $\hat{z}_k = C\hat{x}_k$;
    $k \to k+1$
end
```

2.2.3 Extended Kalman filter (EKF)

If the system incorporates nonlinear system or measurement equations, as is the more general case of Equation 2.1, the equations of algorithm 1 cannot be applied. The EKF solves this problem by local linearisation. Consequently, a condition for the EKF to be applicable is for the equations $\mathbf{f}$ and h to be differentiable. Since this method approximates the functions by a linear model, a system with high nonlinearity leads to poor EKF performance.

The linearisation is incorporated by the Jacobians of the nonlinear equations.

More details are obtained from Equation 2.5.

$$\begin{aligned} F_k &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}_{k-1}} \\ H_k &= \frac{\partial h}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}_k^-} \end{aligned} \tag{2.5}$$

In practice the Jacobian can either be calculated analytically or numerically. The former is preferred if a relatively simple analytical model is available, the latter for numerical or complex analytical models.

The nonlinear functions and calculations of the Jacobians make for the EKF algorithm of algorithm 2.

Algorithm 2: extended Kalman filter

```
Result: Measurement and state estimates $\hat{z}_i$ and $\hat{\mathbf{x}}_i$
initialize $P_0, \mathbf{x}_0$;
while (1) do
    // prediction step
    $\hat{x}_k^- = f(\hat{x}_{k-1})$;
    calculate $F_k$ from $\hat{x}_{k-1}$;
    $P_k^- = F_k P_{k-1} F_k^T + Q_0$;
    // assimilation (update) step
    calculate $H_k$ from $\hat{x}_k^-$;
    $S_k = H_k P_k^- H_k^T + R_0$;
    $K_k = P_k^- H_k^T S_k^{-1}$;
    $\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-))$;
    $P_k = (I - K_k C) P_k^-$;
    // calculate output
    $\hat{z}_k = h(\hat{x}_k)$;
    $k \to k + 1$
end
```

2.2.4 Unscented Kalman filter (UKF)

In the case of nonlinear equations, the unscented Kalman filter (UKF) competes with the EKF. The UKF is a so-called sigma-point Kalman filter. Those are a class of Kalman filters that employ statistical linearisation as a way of omitting numerical or analytical differentiation, as well as avoiding corruption of the posterior mean and covariance due to nonlinearities. To this end the unscented transformation (UT) is performed.

The unscented transformation (UT)

The UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation.

First, a state distribution is obtained by selecting a set of sample points, called sigma-points, with known mean $\mathbf{x}_{k-1}$ and covariance P_{k-1}. Each point is subsequently propagated through the system resulting in a transformed cloud of points. Finally the statistics of the new points are used to compute a new mean and covariance, $\mathbf{x}_k^-$ and P_k^- respectively.

Algorithm

The UKF iteratively executes three steps: calculation of the sigma-points, forecasting and assimilation.

Sigma-point selection An extensively discussed topic in literature is the selection of the UKFs sigma-points. Most disquisitions plead for smaller sets of points -yielding less computational expense- without loss of accuracy. With an unlimited range of possibilities, this topic is well beyond the scope of this paper. It is from here on assumed that the following, common set of $2n+1$ points, provides a reliable measure for UKF performance.

$\mathbf{x}_{k-1}^j$ denotes the jth point based on mean $\mathbf{x}_{k-1}$. Each point is accompanied by a weight W^j to result in a set

$$X_{k-1} = \{(\mathbf{x}_{k-1}^j, W^j) \,|\, j = 0 \mathinner{..} 2n\}. \tag{2.6}$$

The points are then calculated from the following equations where $W^0 \in (-1, 1)$ and $(\sqrt{X})_i$ denotes the ith row or column of the matrix square root of X.

$$\begin{aligned}
\mathbf{x}_{k-1}^0 &= \mathbf{x}_{k-1} && && (2.7)\\
\mathbf{x}_{k-1}^i &= \mathbf{x}_{k-1} + \Big(\sqrt{\frac{n}{1-W^0} P_{k-1}}\Big)_i && \forall i = 1 \mathinner{..} n && (2.8)\\
\mathbf{x}_{k-1}^i &= \mathbf{x}_{k-1} - \Big(\sqrt{\frac{n}{1-W^0} P_{k-1}}\Big)_i && \forall i = 1 \mathinner{..} n && (2.9)\\
W^j &= \frac{1-W_0}{2n} && \forall j && (2.10)
\end{aligned}$$

Forecast step Once the points are attained, they are propagated through the system:

$$\mathbf{x}_k^{-,j} = \mathbf{f}(\mathbf{x}_{k-1}^j) \tag{2.11}$$

The forecast mean and covariance are calculated using these transformed points:

$$\mathbf{x}_k^- = \sum_{j=0}^{2n} W^j \mathbf{x}_k^{-,j} \tag{2.12}$$

$$P_k^- = \sum_{j=0}^{2n} W^j (\mathbf{x}_k^{-,j} - \mathbf{x}_k^-)(\mathbf{x}_k^{-,j} - \mathbf{x}_k^-)^T + Q_{k-1} \tag{2.13}$$

Similarly, the observations corresponding to the points are calculated along with their mean and covariance. Additionally the cross-covariance[4] between the transformed points and these observations is calculated in Equation 2.17.

$$z_{k-1}^{-,j} = h(\mathbf{x}_{k-1}^j) \tag{2.14}$$

$$z_{k-1}^- = \sum_{j=0}^{2n} W^j \mathbf{z}_{k-1}^{-,j} \tag{2.15}$$

$$\mathrm{Cov}(\tilde{z}_{k-1}^-) = \sum_{j=0}^{2n} W^j (\mathbf{z}_{k-1}^{-,j} - z_{k-1}^-)(z_{k-1}^{-,j} - z_{k-1}^-)^T + R_k \tag{2.16}$$

$$\mathrm{Cov}(\tilde{\mathbf{x}}_k^-, \tilde{z}_{k-1}^-) = \sum_{j=0}^{2n} W^j (\mathbf{x}_k^{-,j} - \mathbf{x}_k^-)(z_{k-1}^{-,j} - \mathbf{z}_{k-1}^-)^T \tag{2.17}$$

Assimilation step The update formulas show clear resemblance with those presented for the EKF in algorithm 2. In case of linear equations and Gaussian noise, both the UKF and EKF reduce to a linear Kalman filter (LKF).

$$K_k = Cov(\tilde{\mathbf{x}}_k^-, \tilde{\mathbf{z}}_{k-1}^-) Cov(\tilde{\mathbf{z}}_{k-1}^-)^{-1} \tag{2.18}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + K_k(\mathbf{z}_k - \mathbf{z}_{k-1}^-) \tag{2.19}$$

$$P_k = P_k^- - K_k Cov(\tilde{\mathbf{z}}_{k-1}^-) K_k^T \tag{2.20}$$

Summary The resulting algorithm is presented by algorithm 3. For clarity, the equations are not copied and a more qualitative description is maintained [11, 12, 13].

2.2.5 Tuning the Kalman filter (KF)

Once a suitable algorithm is selected and implemented, the challenge of designing a well-performing KF lies in tuning its parameters.

Model The first ‘parameter’ to be established is the maintained model. This selection is often a trade-off. An accurate model, closely approximating the reality, has higher dimensions and therefore increases computational cost. Additionally, it is often unnecessary as relatively small deviations are be embodied by process noise.

---

[4] The cross-covariance is the covariance of two processes. The cross-covariance between two states $a \in \mathbb{R}^{n \times 1}$ and $b \in \mathbb{R}^{m \times 1}$ is presented by an $n \times m$ matrix. The cross-covariance between two equal states a equals the covariance, an $n \times n$ matrix.

Algorithm 3: unscented Kalman filter

```
Result: Measurement and state estimates $\hat{z}_i$ and $\hat{\mathbf{x}}_i$
while (1) do
    // sigma-point calculation
    choose $\mathbf{x}_{k-1}^{j}$ and $W^j$;
    // prediction step
    calculate $\mathbf{x}_k^-$ and $P_k^-$;
    calculate $\mathbf{z}_{k-1}^-$, $Cov(\tilde{\mathbf{z}}_{k-1}^-)$ and $Cov(\tilde{\mathbf{x}}_k^-, \tilde{\mathbf{z}}_{k-1}^-)$;
    // assimilation (update) step
    calculate $K_k$, $\mathbf{x}_k$ and $P_k$;
    // calculate output
    $\hat{z}_k = h(\mathbf{x}_k)$;
    $k \rightarrow k+1$
end
```

In order to account for environmental changes, exceptions, etc. without increasing the model dimensions, switching between several different models on the account of a monitor is also an option. Special care needs to be taken when transitioning between models.

Measurement R_k and process noise Q_k With the model in place, the go-to tuneable variables are the process and measurement noise. The KF steady-state performance is determined by the ratio Q_k/R_k. If the ratio becomes small, much measurement noise is implied, and the filter tends more towards the modelled (predicted) results. For high values, the model becomes less accurate and more weight is given to the measurements. These variables also provide the means for on-line filter tuning: if they are altered during operation, the behaviour of the filter is manipulated accordingly. This for instance can prove to be a useful asset if an event is registered which is known to be outside of the scope of the model. Increasing Q_k then allows for greater deviations from the predicted measurements.

Initial state $\mathbf{x}_0$ and error covariance matrix P_0 These two variables define the initial state the filter is in. They are intertwined as they represent the Gaussian distribution the filter is initialised with. In case of high certainty of the initial state, P_0 can be chosen small or even equal to Q_0. If not, the filter converges (or diverges) more rapidly if P_0 is higher. The error covariance matrix usually converges to a constant value over time. A too large P_0 might, especially in combination with a poorly chosen initial state (see next), trigger the system to converge to faulty states. A possible method to provide a 'blind' $\mathbf{x}_0$ estimate is solving a least-squares problem using the internal model.

Time step Δt As it is intended to feed the KF independent measurements[5], the time step is -if possible- selected according to the availability of these measurement. This interval has to exceed the computation time of one filter iteration by some margin. An important remark to take into account is that the KF is able to skip an assimilation step if no new or useful measurement is available. Then only the predicted values are propagated. This however is not the intended behaviour.

Jacobian As explained in subsection 2.2.3, the calculation of the Jacobian can either be performed analytically or numerically. Furthermore, there also exists a possibility of including second or higher order terms, in the form of a Hessian[6]. In practice this leads to high computational cost, with limited improvement only when the measurement noise is small [14].

Sigma-points For completeness, for the case of a UKF, sigma-point selection is also listed referring to section 2.2.4.

2.2.6 On-line consistency test

The discussed filters are highly sensitive to consistency between the measured value and their internal model. If they are fed data which is inconsistent, they quickly lose their purpose as the estimated states become meaningless. In order to distinct useful data and algorithmically detect hazardous circumstances, the need for an on-line consistency test rises. [15]

NIS & SNIS

The linear Kalman filter (LKF) and extended Kalman filter (EKF) are, considering their analytical character, particularly suitable for implementation of such a test. A convenient solution is the normalized innovation squared (NIS). As the name partly reveals, the NIS is calculated by squaring the innovation $\nu_k = z_k - h(\hat{x}_k^-)$ and normalising it with the innovation covariance $S_k = H_k P_k^- H_k^T + R_k$. Its formula is shown by Equation 2.21. As only present information of the current time step is required, the on-line NIS calculation is computationally inexpensive. Similarly, the summed normalized innovation squared (SNIS) is calculated by summing the latest M NIS' in order to expose more consistently present issues.

$$\mathrm{NIS}_k = \nu_k' S_k^{-1} \nu_k \tag{2.21}$$

$$\mathrm{SNIS}_k = \sum_{i=k-M}^{k} \mathrm{NIS}_i \tag{2.22}$$

---

[5] The measurement variance ideally fulfils Equation 2.4.

[6] The Hessian matrix contains the second order partial derivatives of a function. It is the transpose of the Jacobian of the gradient of a function.

If the filter assumptions are met, ν_{ki} is Gaussian distributed which makes the NIS$_k$ χ^2-distributed with m degrees of freedom. The χ^2-distribution with m degrees of freedom is the distribution of the sum of the squares of m Gaussian distributed variables. $m = 1$ in this case, the dimension of the measurement equation. SNIS$_k$ is then χ^2-distributed with M degrees of freedom, the amount of summed terms. Knowing the distribution allows for the calculation of a confidence interval, e.g. 95%, for which inclusion of the (S)NIS is then assessed.

UKF consistency test

As the calculation of the innovation covariance S_k requires linear or linearised measurement equations H_k, it's calculation is less convenient for the UKF. A less rigorous approach circumvents this issue by using the trace of the error covariance matrix P_k for normalisation as shown in Equation 2.23 [16].

$$\epsilon_k = \nu_k' \mathrm{tr}(P_k)^{-1} \nu_k \tag{2.23}$$

This formula accompanies a predominantly experimental approach, especially when it comes to selecting a 'consistency criterion' in the form of $\epsilon_{\mathrm{thres}}$, a threshold value. Equation 2.23 is not χ^2-distributed. However, within certain limits (e.g. when the filter has converged and P_k is not volatile) it provides a useful criterion. Alternatively, an estimate for H_k is made numerically, at a computational cost.

2.3 Moving Horizon Estimation (MHE)

A state estimation technique that competes with Kalman filtering is moving horizon estimation (MHE). An important difference lies in the fact that the MHE at every time step considers the last N_{hor} measurement values instead of only the last one. N_{hor} is called the 'horizon', hence moving horizon estimator (MHE). As the KF, it uses an internal model to compare expected output with measurements. It does so by solving an optimisation problem. At every algorithm time step Δt, the states $\mathbf{x}_k$ are sought that minimise Equation 2.24.

$$J_{MHE} = \sum_{j=1}^{N} w_{meas}(h(\mathbf{x}_{k,k-j+1}) - z_{k-j+1}) + \sum_{j=1}^{N} \mathbf{w}_{states}(\mathbf{x}_{k,k-j+1} - \hat{\mathbf{x}}_{k-1,k-j+1}) \tag{2.24}$$

Here $\mathbf{x}_{k,k-j+1}$ is the state vector at time step $k-j+1$ with respect to $\mathbf{x}_k$, the variable of the optimisation problem at time step k. I.e. $\mathbf{x}_{k,k-j+1} = A^{1-j}\mathbf{x}_k$ for a linear system. These values are compared with the results of the previous optimisation problem: $\hat{\mathbf{x}}_{k-1,k-j+1}$ and with z_k, the measurement at time step k. w_{meas} and $\mathbf{w}_{states}$ are weights used to tune the estimator. $\mathbf{w}_{states}$ is a row vector with the same dimension as the system, which is presumed to be scaled by a scalar w_{states}. E.g. $\mathbf{w}_{states} = w_{states} * [1, 1, 1, 1, 1, 1]$. A higher ratio w_{meas}/w_{states} results in more weight for the measurement, resulting in faster deviation from the previous estimation and vice versa. This ratio is comparable with Q_0/R_0 in the context of the invariant KF.

When the assumptions of additive Gaussian noise hold, $N_{\text{hor}} = 1$ and the system equations are linear, the MHE is, like the LKF, and optimal estimator. Both then provide the same result [17].

Chapter 3

The Heart Model

This chapter introduces a mathematical model for one-dimensional heart movement. An accurate heart model is an important aspect in the development of the state estimator, and some (whether or not predictive) forms of control.

3.1 Heart movement

Information about characteristics of the heart motion during the CABG is crucial for the purposes of this work. To allow for overall clarity and due to a lack of available experimental data in this area, some assumptions are made.

3.1.1 Frequency spectrum

In Bebek et al. [18] a power spectral density (PSD) of a heart movement is obtained through an experiment with an adult porcine. This animal's heart is widely accepted to be similar to a human one [19]. Sonomicrometry, a technique of measuring the distance between piezoelectric crystals based on the speed of acoustic signals through the medium (in this case human tissue) they are embedded in, is used to wirelessly measure the movement of the heart. From these movements the frequency spectrum is calculated. The result is displayed in Figure 3.1. In the PSD, the heart beat and respiratory frequencies are distinguished at 2.0 Hz and 0.37 Hz respectively. The peak at 4.0 Hz results from the first harmonic of the heartbeat movement. It is worth noting that, concerning the heartbeat motion, the lion's share of the movement is represented by the ground frequency and its first harmonic. This is also assumed to be the case for the model design. Deviations from this assumption are resolved by accounting for more harmonics.

3.1.2 Electrocardiogram (ECG)

Another important characteristic of heart motion is its relation with the simultaneously recorded ECG signal. The purpose of this passage is to demonstrate the close correlation between the heart movement and ECG, and to ascertain that to a certain extent the movement of the heart can be predicted through the latter.

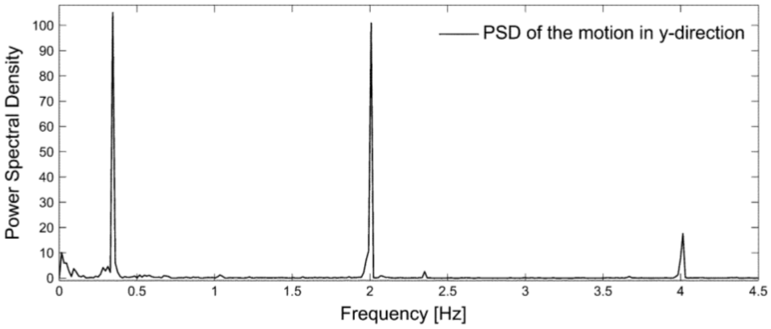
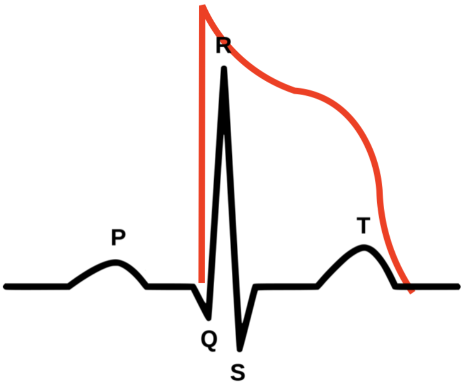


Figure 3.1: PSD of heart movement in sonomicrometric experiment. Dominant modes are observed at 0.37 Hz and 2.0 Hz, corresponding to respiratory movement and the heartbeat rate respectively. The peak at 4.0 Hz is the first harmonic of the heartbeat [18]



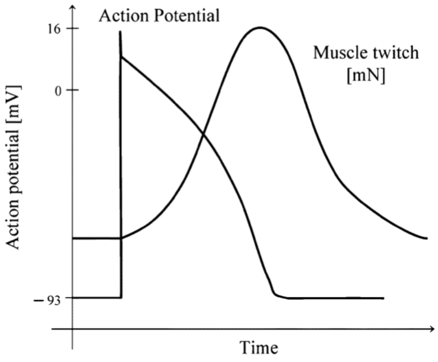


Figure 3.2: Illustration of the relationship between action potential (red) and ECG signal (black). The steep rising of the action potential occurs in parallel with the ECG peak in the QRS complex.

Figure 3.3: Illustration of the relationship between action potential and ventricular muscle force. Lag between the excitation and the peak force is about 200 ms long and peak force approximately coincides with repolarisation.[18]

The ECG signal measurement is a direct result of the ventricular action potential, a voltage change which is a by-product of the mechanism responsible for the contraction of the heart muscles [20]. Figure 3.2 reveals the relation between the action potential and the ECG signal. As these signals are inherently linked, the ECG signal, unquestionably present in the operation theater (OR), provides the means to estimate the shape of the action potential. This curve in turn reveals information about the ventricular muscle force, which directly leads to the heart displacement of which the prediction is intended. As is illustrated by Figure 3.3, the peak force of the ventricular heart muscle coincides with repolarisation, which is observed by the settling of the action potential. This repolarisation typically occurs 200 ms after excitation, the action potential peak.

These observations link the ECG peak in the QRS complex, the large peak in the ECG signal,[1] to the heart movement. Determining a general correlation linking the time delay between the ECG peak in the QRS complex and the peak movement of the heart $\Delta T_{ecg,p}$ with environmental parameters such as the angular frequency of the heart ω_0 is beyond the scope of this work. It is from here on assumed $\Delta T_{ecg,p}$ is approximately constant with $\Delta T_{ecg,p} \approx 200\,\mathrm{ms}$.

3.2 Model concept

The intended system does not exert any control over the heart movement. It makes use of the fact that any periodic signal can be written as a sum of an (infinite) amount of sinusoids ($N \to \infty$) and an offset:

$$y(t) = C_0 + \sum_{i=1}^{N} a_i \sin(\omega_i t + \phi_i) \tag{3.1}$$

where $\omega_i = i\omega_0$ and ω_0 is a certain angular frequency. The '0' subscript from this pulsatance is from hereon dropped. $y(t)$ is the time-dependent displacement of the heart according to the model. The displacement of the heart $y_r(t)$ can be compared with expression Equation 3.1 by writing it as follows:

$$y_r(t) = C_0(t) + \sum_{i=1}^{N} a_i(t) \sin(\omega_i(t) t + \phi_i(t)) \tag{3.2}$$

The discrete-time variants of these expressions are

$$y[k] = C_0 + \sum_{i=1}^{N} a_i \sin(\omega_i \Delta t k + \phi_i) \tag{3.3}$$

and

$$y[k] = C_0[k] + \sum_{i=1}^{N} a_i[k] \sin(\omega_i[k] \Delta t k + \phi_i[k]) \tag{3.4}$$

---

[1] In electrophysiology, the ECG signal is analysed by segmenting it into distinct phases. The QRS complex is the narrow phase in which the largest peak occurs.

respectively, where k represents the step and Δt the time step between the occurrence of k and $k + 1$.

In what follows, for simplicity and based on subsection 3.1.1, a model with two sinusoids is used where the second frequency is fixed to be twice as great as the first one.

3.3 State Space Representations (SSRs)

As chapter 2 illustrates, the intended system involves numerous calculations and is consequently designed to work with and on digital hardware. Therefore, and for simulation purposes, the need for a discrete-time state-space representation (SSR)[2] rises.

An LTI discrete-time SSR in general has the following lay-out:

$$\mathbf{x}[k+1] = A\mathbf{x}[k] + B\mathbf{u}[k] \tag{3.5}$$

$$\mathbf{y}[k] = C\mathbf{x}[k] + D\mathbf{u}[k] \tag{3.6}$$

where $\mathbf{x}[k]$ is the state vector, $\mathbf{y}[k]$ the output vector and $\mathbf{u}[k]$ the input or control vector [21]. In the elaborated context there is no input, the measurement equation Equation 3.6 is non-linear, and the output as well as measurement function are scalar, leading to these equations:

$$\mathbf{x}[k+1] = A\mathbf{x}[k] \tag{3.7}$$

$$y[k] = h(\mathbf{x}[k]) \tag{3.8}$$

There are several ways to design an SSR for the discussed heart model. In what follows, 3 models are described and compared.

3.3.1 Explicit phase state-space representation (SSR)

This is the most interpretable model. The state-vector being used is $\mathbf{x}_k = [C, a_1, a_2, \phi_1, \phi_2, \omega]_k^T$. The model itself is

$$\mathbf{x}_{k+1} = I_6\mathbf{x}_k \tag{3.9}$$

$$y_k = C_k + a_{1,k}\sin(\omega_k k\Delta t + \phi_{1,k}) + a_{2,k}\sin(2\omega_k k\Delta t + \phi_{2,k}) \tag{3.10}$$

where I_6 represents the identity matrix of 6th order. The Jacobian of the measurement equation for this system is

$$\begin{split} J_1 = [1, \sin(\omega k\Delta t + \phi_1), \sin(2\omega k\Delta t + \phi_2), a_1\cos(\omega k\Delta t + \phi_1), a_2\cos(2\omega k\Delta t + \phi_2), \\ a_1 k\Delta t\cos(\omega k\Delta t + \phi_1) + 2a_2 k\Delta t\cos(2\omega k\Delta t + \phi_2)] \end{split} \tag{3.11}$$

---

[2] A state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations (continuous) or difference equations (discrete).

The Jacobian of a system provides insight in the stability of the system which is a relevant criterion for selecting one to be used in an EKF. Judging the stability of a system in absolute terms is complicated and unnecessary. The Jacobians provide information that is used to compare the systems.

3.3.2 Implicit phase state-space representation (SSR)

A second SSR is similar to the one above. The measurement equation is recalculated using the sine angle sum identity $\sin(a+b) = \cos(a)\sin(b) + \sin(a)\cos(b)$ as follows:

$$\begin{aligned} y_k &= C_k + a_{1,k}\sin(\omega_k k\Delta t + \phi_{1,k}) + a_{2,k}\sin(2\omega_k k\Delta t + \phi_{2,k}) && (3.12) \\ &= C_k + a_{1,k}\cos(\phi_{1,k})\sin(\omega_k k\Delta t) + a_{1,k}\sin(\phi_{1,k})\cos(\omega_k k\Delta t) \\ &\quad + a_{2,k}\cos(\phi_{2,k})\sin(2\omega_k k\Delta t) + a_{2,k}\sin(\phi_{2,k})\cos(2\omega_k k\Delta t) && (3.13) \end{aligned}$$

This formulation introduces the opportunity to change the state vector:

$$\mathbf{x}_k = [C, a_1\cos(\phi_1), a_1\sin(\phi_1), a_2\cos(\phi_2), a_2\sin(\phi_2), \omega]_k^T \tag{3.14}$$

The system equations remain the same. The measurement Jacobian now becomes:

$$\begin{aligned} J_2 = [&1, \sin(\omega k\Delta t), \cos(\omega k\Delta t), \sin(2\omega k\Delta t), \cos(2\omega k\Delta t), \\ &a_1 k\Delta t\cos(\omega k\Delta t + \phi_1) + 2a_2 k\Delta t\cos(2\omega k\Delta t + \phi_2)] \end{aligned} \tag{3.15}$$

The last element of the Jacobian remains the same. Comparing the 2nd, 3rd, 4th and 5th element of J_1 with the 2nd, 4th, 3rd and 5th element of J_2 respectively, shows that J_2 is similar to J_1 but with less dependencies, likely yielding a more stable EKF. This can serve as a basis to assume the second system is superior to the first one for the purpose of this thesis.

3.3.3 Alternative state-space representation (SSR)

A last implementation with less arbitrary system equations is as follows [22]:

$$\begin{bmatrix} C \\ a_1 \\ a_2 \\ \omega_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \Delta t & 1 & 0 \\ 0 & 0 & 0 & 2\Delta t & 0 & 1 \end{bmatrix} \begin{bmatrix} C \\ a_1 \\ a_2 \\ \omega_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}_{k} \tag{3.16}$$

$$y_k = C_k + a_{1,k}\sin\theta_{1,k} + a_{2,k}\sin\theta_{2,k} \tag{3.17}$$

and yields the following Jacobian:

$$J_3 = [1, \sin(\theta_1), \sin(\theta_2), 0, a_1\cos(\theta_1), a_2\cos(\theta_2)] \tag{3.18}$$

The simplicity of the measurement equation and Jacobian associated with this system make it suitable for the purposes of this work. This is the system used throughout what follows.

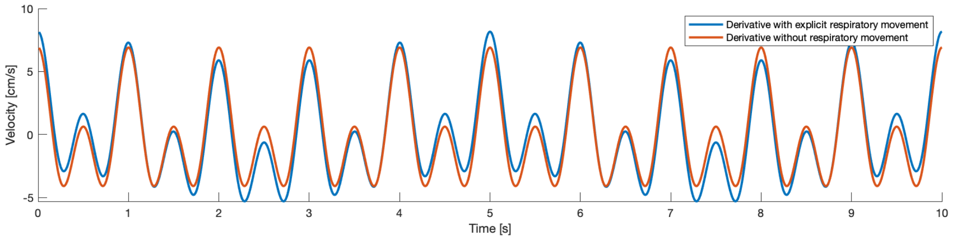


Figure 3.4: Derivative of displacement signal with (red) and without (blue) omission of respiratory movement in estimation model.

3.4 Respiratory movement

So far, respiratory movement has been neglected. One way of coping with it is to introduce a second ground frequency to the system, so that it looks like this.

$$\begin{bmatrix} C \\ a_1 \\ a_2 \\ a_3 \\ \omega_0 \\ \theta_1 \\ \theta_2 \\ \omega_1 \\ \theta_3 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\Delta t & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t & 1 \end{bmatrix} \begin{bmatrix} C \\ a_1 \\ a_2 \\ a_3 \\ \omega_0 \\ \theta_1 \\ \theta_2 \\ \omega_1 \\ \theta_3 \end{bmatrix}_{k} \tag{3.19}$$

$$y_k = C_k + a_{1,k} \sin \theta_{1,k} + a_{2,k} \sin \theta_{2,k} + a_{3,k} \sin \theta_{3,k} \tag{3.20}$$

Equation 3.19 introduces three new states and therefore three extra degrees of freedom as compared to Equation 3.16. For measurement correction purposes it is arguably justified to omit this modification by properly tuning the process noise of the filter. More specifically, by increasing the expected noise of the offset. However, as in chapter 5, where the design of the control is tackled, a feed-forward control strategy is deployed, the derivative of the estimated signal is of vital importance. Ideally this velocity signal is come by, by analytically calculating it from the estimated states. As dropping this respiratory term significantly increases the error on the estimated derivative, it is included in the model. Figure 3.4 shows this phenomenon for the signal of Equation 3.21. The first term embodies the respiratory movement.

$$y(t) = \sin(0.2 * 2\pi t) + 0.5 * \sin(2\pi t) + 0.3 * \sin(2 * 2\pi t) \tag{3.21}$$

The blue signal is the actual velocity, the red one is the displacement's derivative when the respiratory term is omitted. Nonetheless, for clarity, in what follows respiratory-related considerations are frequently omitted.

3.5 Conclusion

Analysis of the heart movement PSD and electrocardiac characteristics reveal exploitable properties. A model for the movement of the heart is set forth and corresponding SSRs are proposed. One is selected to serve future purposes based on its complexity and dependency in context of the EKF. The chapter is concluded by explaining why respiratory movement cannot be omitted in the model.

Chapter 4

Filter Implementation

4.1 Objective

In this chapter the state estimation concepts introduced in the previous one are deployed towards a twofold purpose.

On one hand there is the envisioning of an on-line improvement of the heart displacement measurement. Taking into account the periodicity of the movement it is intended to filter out sensor noise, consequently allowing for less biased robot control and better motion compensation. The second objective is to provide additional information that is either necessary or engenders an improvement for (some of) the applied control strategies. An example is section 4.5, where a KF is implemented for estimation of velocity from a given displacement. This signal is required for a feed forward velocity control strategy. It is not the only way to achieve this derivation, but it is later proven to increase overall performance.

4.2 EKF - UKF comparison

This section aims to compare the two filters performance-wise.

4.2.1 Test signal

To allow for testing the implemented filters, a test signal corresponding to subsection 3.3.3 is generated which embodies Gaussian process noise. Afterwards measurement noise is added. The maintained dimensional units are centimetres [cm], as this is the order of magnitude of the movement. Specific values are set as follows.

The g subscript refers to a variable that corresponds to the generated signal.

$$\Delta t = 10^{-3}\mathrm{s} \tag{4.1}$$

$$\mathbf{x}_{0g}^T = [1.5, 0.5, 0.3, 2\pi, 0, 0] \tag{4.2}$$

$$R_{kg} = R_{0g} = 10^{-3}\mathrm{cm}^2 \tag{4.3}$$

$$Q_{kg} = Q_{0g} = 10^{-5}\begin{bmatrix} 1\,\mathrm{cm}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1\,\mathrm{cm}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1\,\mathrm{cm}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10\,\mathrm{s}^{-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.4}$$

The process noise variance of the phases is zero in order to maintain -to a certain extent- the shape of the signal. I.e. the relative phase difference between the sines is not expected to change. The heart rate is initialised at 60 BPM ($\omega = 2\pi\,{}^{\mathrm{rad}}\!/\!_{\mathrm{s}}$). Prior to the addition of measurement noise $v_k \sim \mathcal{N}(0, R_0)$, the signal is low-passed[1] using a low-pass filter (LPF) with a cut-off frequency of 5 Hz. This is physiologically justified because human tissue has mechanical low-pass characteristics. The resulting heart movement and a zoom with on which the generated measurements are displayed are shown in Figure 4.1 and Figure 4.2 respectively.

4.2.2 EKF implementation

From chapter 2, algorithm 2 is implemented. Considering Equation 3.16 and Equation 3.17, the calculation of F_k is superfluous and $F_k = A$ for every iteration. The calculation of H_k, based on Equation 3.18, results in Equation 4.5:

$$H_k = [1, \sin(\hat{x}_{5,k}^-), \sin(\hat{x}_{6,k}^-), 0, \hat{x}_{2,k}^- \cos(\hat{x}_{5,k}^-), \hat{x}_{3,k}^- \cos(\hat{x}_{6,k}^-)] \tag{4.5}$$

R_k is set to R_{0g}, and $\mathbf{x}_0^T = \mathbf{x}_{0g}^T$. Two series of simulations are executed where Q_0 is varied by scaling Q_{0g} and I_6, and the mean square error (MSE) between the output and the actual movement is calculated. Mapping this provides insight in the filter's sensitivity to a deviating process noise covariance matrix and shows the best-obtainable results. I.e. the algorithm is repeatedly applied using $Q_0 = cQ_{0g}$ and $Q_0 = cI_6$ with varying c. The error covariance is initialised by $P_0 = Q_0 * 10^2$.[2] The results are given by Figure 4.5: the optimal c is ≈ 1.0 as expected and the corresponding MSEs are $1.4 * 10^{-4}\,\mathrm{cm}^2$ and $1.6 * 10^{-4}\,\mathrm{cm}^2$. The MSEs are calculated from the 3000th (3 s) time step onward, well beyond the threshold of initialisation effects.

Figure 4.2 and Figure 4.3 show the actual movement, the measurements and the EKF estimation for $c = 1$. Accurate tracking is obtained.

---

[1] The used filter performs zero phase filtering with 60 dB attenuation. For more information, see the MATLAB documentation on *lowpass()*.

[2] P_0 could be set to Q_0 as the initial state is known to be correct. Increasing it however makes filter convergence observable, increases the possibility of convergence to an unintended state and might reveal faults or instabilities.

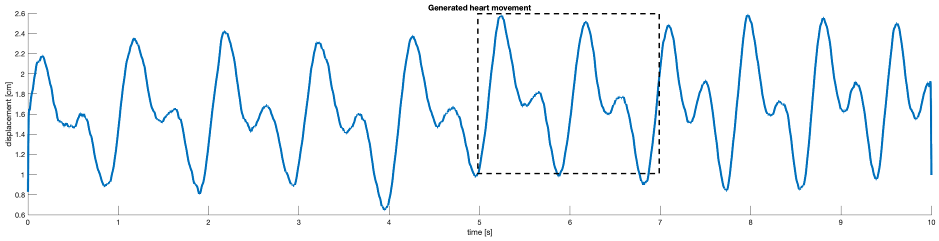


Figure 4.1: The generated heart motion without noise. The dashed line shows the zoomed area of Figure 4.2.

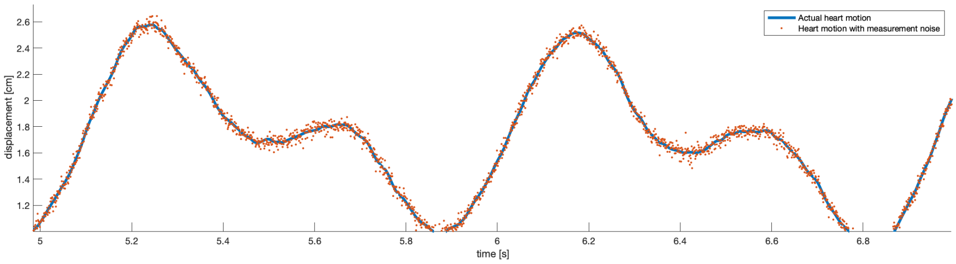


Figure 4.2: Zoom of the generated heart motion including generated measurement values.

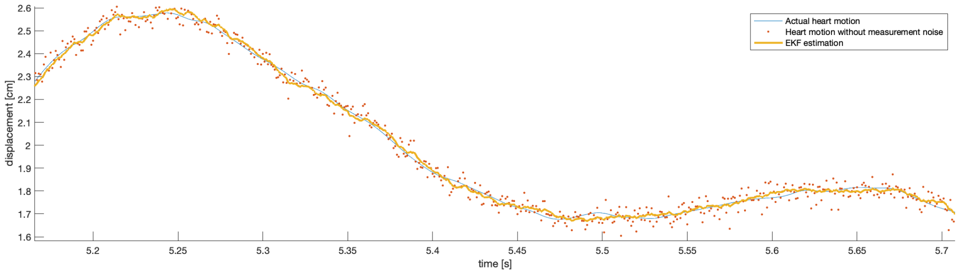


Figure 4.3: Strong zoom on the generated heart motion, generated measurement values and EKF output signal.

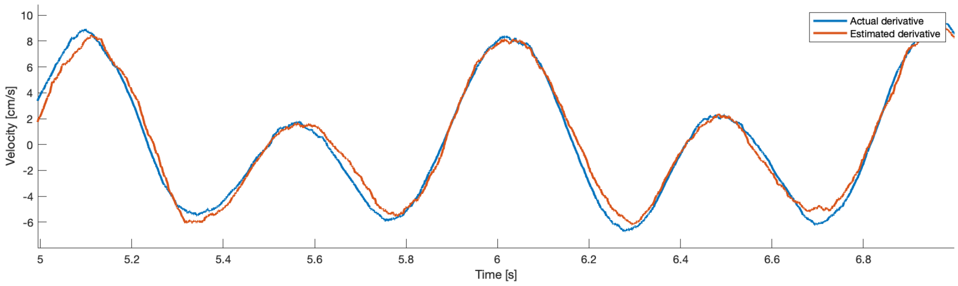


Figure 4.4: Actual derivative of input movement along with estimated derivative based on EKF states. The estimated derivative is shown for two different filter solutions.

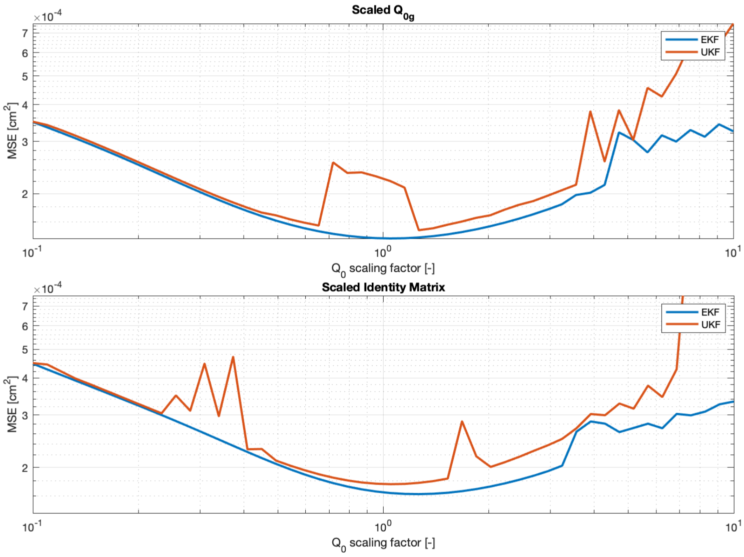


Figure 4.5: Double plot of the MSEs of the UKF and EKF in function of the scaled process noise matrix. The authentic process noise of the generated signal (top) and the identity matrix (bottom) are used.

The simulation is performed again with the same measurement signal and $Q_0 = Q_{0g}$ with $Q(5,5) = Q(6,6) = 10^{-6}\,\mathrm{rad}^2$, granting the estimation of the arguments a certain amount of freedom. Depending on the robot control strategy, the estimated derivative of the displacement is of paramount importance. Consider e.g. a feed forward velocity controller. The velocity is therefore shown in Figure 4.4. Four predicted states in case of wrong initialisation $\mathbf{x}_0 = [2, 0.7, 0.2, 2\pi + 1, 0, 0]^T$ are plotted in Figure 4.6 along with the actual states from the generated signal. After the filter has converged (after about 2 s) the actual states are accurately tracked, albeit with a small delay.

4.2.3 UKF implementation

Similarly, a UKF algorithm corresponding to algorithm 3 is implemented and tested with the same measurement signal. The sigma-points are calculated according to section 2.2.4. The remaining parameters are set equal to those used for the EKF. The result of Q_0 optimisation is shown in Figure 4.5. Once again, the minimal value is found for $c \approx 1.0$ corresponding to an MSE of $1.5 * 10^{-4}\,\mathrm{cm}^2$. This result is comparable to that of the EKF. The results for state estimation and measurement filtering are barely distinguishable from Figure 4.6 and Figure 4.3 respectively.

Two observations are attained from the comparison in Figure 4.5. Firstly, around the optimal value the plot for the UKF shows some discontinuity which reflects a higher sensitivity to parameter tuning. Secondly, and more importantly, for larger deviations in Q_k, the curve quickly becomes significantly steeper for the UKF. This reveals an important flaw of the UKF in this specific scenario as for the real world

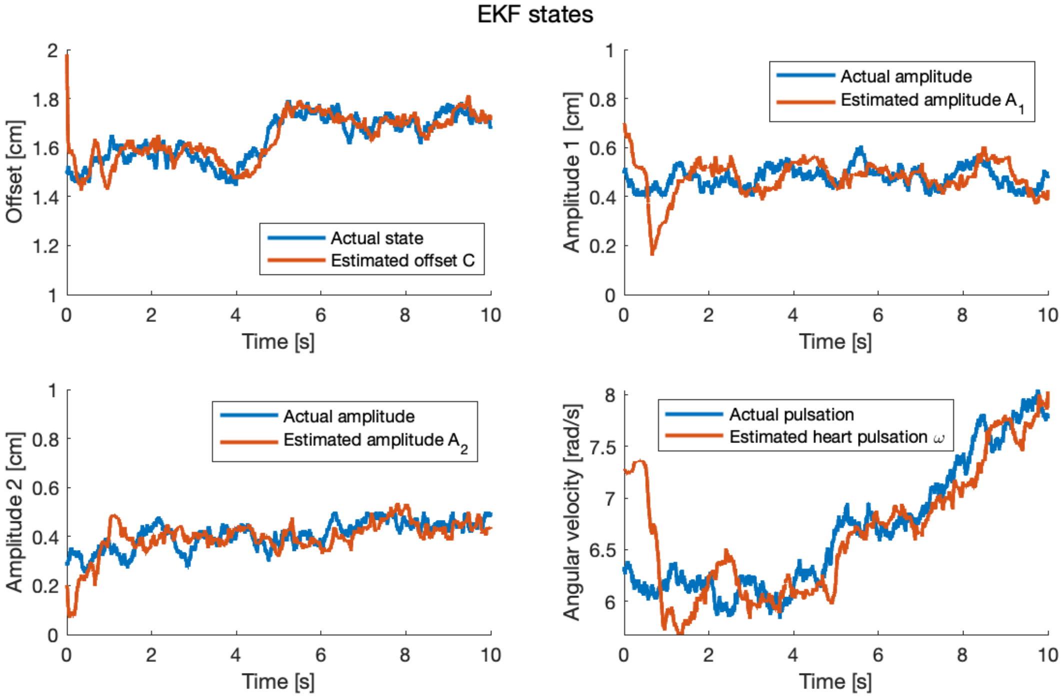


Figure 4.6: EKF estimated states and actual states. (States 1 through 4.)

system the process noise covariance matrix Q_0 has to be estimated and does not equal Q_{0g}.

Moreover the EKF is remarkably less time consuming. The simulation was executed (all 10000 steps) in 0.31 s as compared to 2.56 s for the UKF[3]. The time cost for individual corresponding simulation steps shows the same ratio. On the basis of these observations, the EKF is selected for further use.

4.3 Moving Horizon Estimation (MHE)

The MHE algorithm is elaborated in section 2.3. This section assesses its performance in the context of periodic motion estimation.

4.3.1 Parameters

A test signal is generated with Q_{0g} from Equation 4.4 and $R_0 = 10^{-3}\,\text{cm}^2$. Moving horizon estimation is more computationally intensive as compared to (extended) Kalman filtering. The time step is therefore increased to $\Delta t = 10\,\text{ms}$. The performance in terms of MSE is compared with that of an EKF with the same time step. Simulations are done with a horizon of $N_{\text{hor}} = 1, 5$ and 10. Both filters

---

[3] The simulations were 'warm-executed' on a machine featuring a 6th generation 'Skylake' 2.6GHz quad-core i7 processor with Radeon Pro 450 dGPU, in similar circumstances.

are initialised with the correct states, those of Equation 4.2. The EKF process covariance matrix is set to $Q_0 = I_6 * 10^{-4}$ and the filter is initialised with $P_0 = I_6 * 10^{-2}$. Corresponding to this covariance matrix, the MHE state weights are chosen $\mathbf{w}_{\text{states}} = w_{\text{states}} * [1, 1, 1, 1, 1, 1]$.

4.3.2 Results

MHE performance is mainly determined by the horizon N_{hor} and the ratio of the weights, w_{states}/w_{meas}. The results of the MHE simulations are shown in Figure 4.7. In this range, increasing N_{hor} increases accuracy. The difference between the most accurate simulations of $N_{\text{hor}} = 5$ and $N_{\text{hor}} = 10$ is negligible, less than 1%. However, initial state error rejection significantly increases with increasing horizon, to the extent that for $N_{\text{hor}} = 1$, no convergence is reached for the trumped initial state vector $\mathbf{x}_0$ from subsection 4.2.2. This finding increases the required computational power to result in acceptable behaviour. From Figure 4.7 it is also derived that the MHE approaches the EKF closely in terms of performance, but fails to reach/surpass it. The optimal case for $N_{\text{hor}} = 5$ is $w_{states}/w_{meas} = 0.1$, which is plotted in Figure 4.8 along with the result for the Kalman filter. Similar behaviour is observed. Both their estimated states along with the actual states are shown in Figure 4.9. Once again, the MHE is only able to provide similar but inferior results.

When the MHE weights and EKF variances are varied separately, both filters deliver better results. Parallels are however clear with the results above. The MHE does not achieve the same performance as the Kalman filter. Additionally, tuning the MHE is more difficult and the algorithm's performance is more prone to weight deviations.

These findings, along with the fact that for these specific simulations the MHE algorithm takes over 100 times longer to run when compared to the EKF algorithm, result in abandoning the moving horizon estimator (MHE) in favour of the Kalman filter (KF).

4.4 Other Methods

A significant number of methods for on-line estimation (or time delay compensation) remains. Yuen et al. [22] have shown the superiority of the EKF over the autoregressive (AR) model with either least-squares estimator or fading memory estimator for periodic movement. Further methods are not discussed in the context of this work.

4.5 Velocity estimation

The need for the attainment of an on-line velocity computation corresponding to a displacement signal is translated to an additional state estimation problem. The need for this (on-line) information emerges from the robot control (chapter 5). Feed forward velocity control strategies require the velocity signal to be available with the

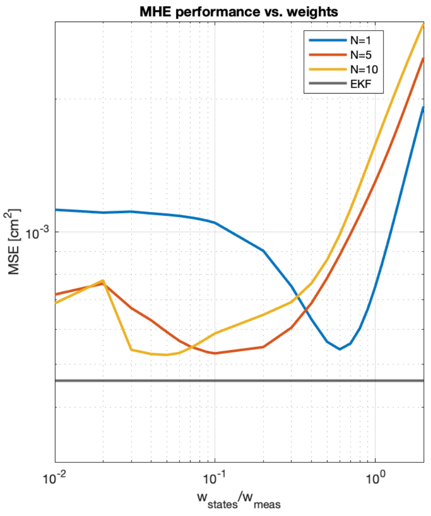


Figure 4.7: MSE results for MHE simulations with varying horizon. Corresponding EKF result is given by horizontal grey line.

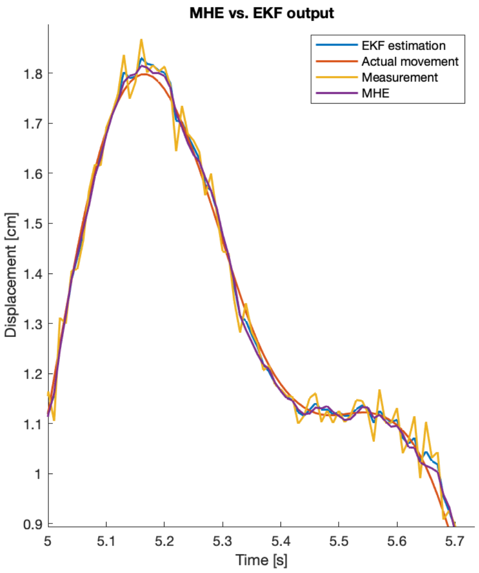


Figure 4.8: Test signal with results for EKF and MHE for $w_{states}/w_{meas} = 0.1$ and $N_{hor} = 5$.

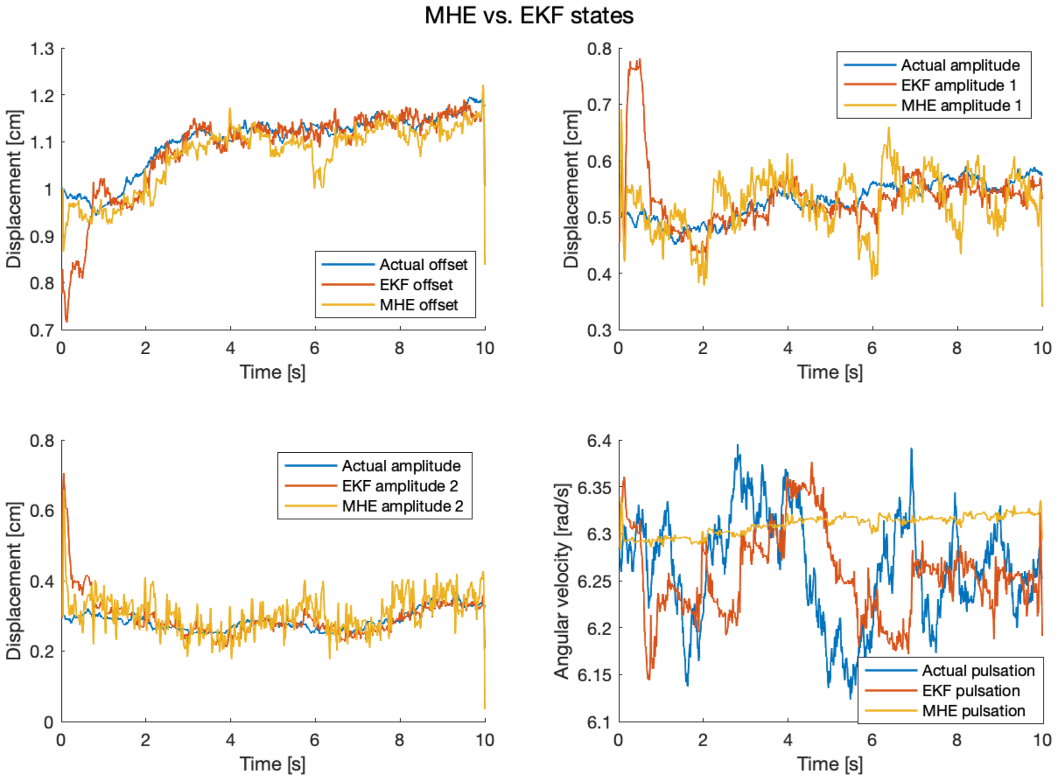


Figure 4.9: MHE states (yellow) against EKF states (red). The true states are displayed in blue.

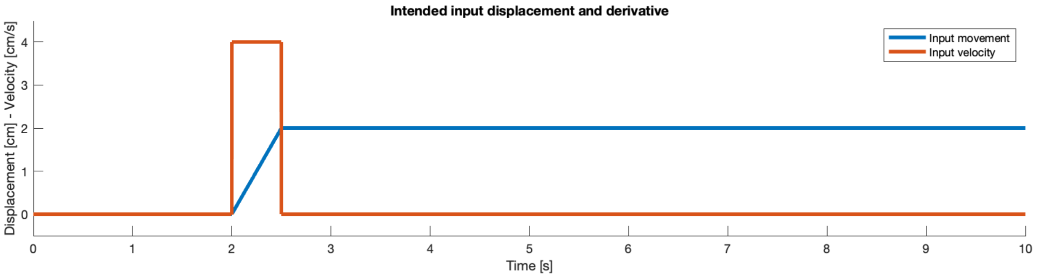


Figure 4.10: Linear input signal and derivative for velocity estimation.

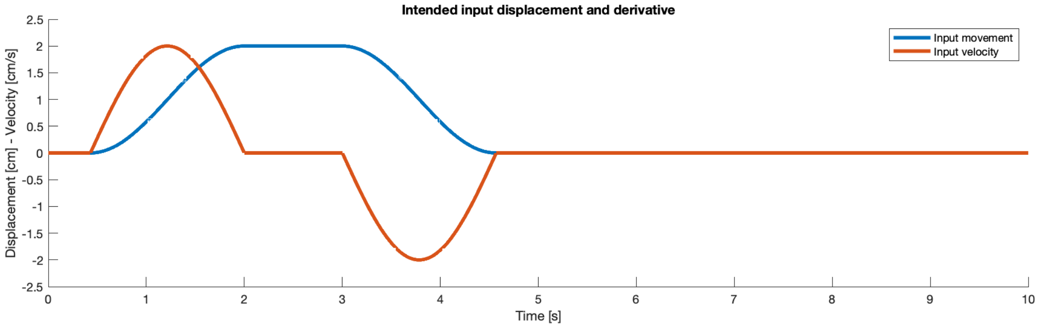


Figure 4.11: (Co)sinusoidal input signal and derivative for velocity estimation.

displacement. This is especially true for the operator's input signal, as no model is available.

The input displacement is expected to contain a certain amount of noise, originating from e.g. discretisation from the digital input console of the operator, or measurement noise. Simple numerical differentiation $v_k = (x_k - x_{k-1})/\Delta t$ would confluence this noise with the small time step that is maintained, resulting in a differentiation signal with considerable noise amplification. A multi-point numerical differentiation partly resolves this issue, but also introduces an inherent time delay. Two other solutions are considered.

The first one is a numerical differentiation of the *low-passed* displacement signal, provided an efficient trade-off between time delay and noise reduction is established. The second is a Kalman filter (KF) which maintains a model that allows for calculation of the derivation. In contemplation of testing these methods, two inherently distinctive test signals are created.

4.5.1 Test signals

The first test signal supplies a linear displacement which leads to a block pulse velocity, see Figure 4.10. A displacement of 2 cm is desired in 0.5 s. The second test signal has a rather block pulse displacement with sinusoidal flanks and therefore a sinusoidal velocity profile as shown in Figure 4.11. A small amount of Gaussian noise is added to the displacement signals with a variance of $R_0 = 10^{-7}\,\text{cm}^2$.

4.5.2 Derivation of low-passed signal

The signals are propagated through a low-pass Butterworth filter. An on-line numerical differentiation of the latest two values is then performed which provides the velocity signal. In order to minimise the time delay, the order and cutoff frequency of the filter are optimised to the lowest possible MSE between actual and estimated velocity for both signals separately. A lower order and/or higher cutoff leads to smaller time delay but passes more noise.

4.5.3 LKF implementation

Ab initio, Equation 4.6 is used. It correlates displacement x with velocity $\dot{x}$ and presumes a constant process noise covariance matrix Q_0 and measurement noise variance R_0 of Equation 4.7. The model implies a constant velocity is expected.

$$\begin{aligned} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}_{k+1} &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}_k \\ y_k &= x_k + \dot{x}_k \Delta t \end{aligned} \tag{4.6}$$

$$\begin{aligned} Q_0 &= \begin{bmatrix} 0 & 0 \\ 0 & q \end{bmatrix} \\ R_0 &= r \end{aligned} \tag{4.7}$$

The layout of Q_0 accompanies the chosen system equations and must be understood as follows. At every time step the displacement is updated taking into account the previous velocity. The absent variance on the displacement ensures that the difference between the previous and current displacement is traced back to the updated velocity. The measurement (current displacement) is then perceived as a direct result of the velocity. If a variance on the displacement is present as well, a part of the measured value is understood as process noise and the effect of the velocity, and therefore the velocity itself as well, is underestimated.

A possible modification to the system implies constant acceleration $\ddot{x}$ and linear velocity $\dot{x}$. The system equations and noises then become those of Equation 4.8 and Equation 4.9.

$$\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_k \tag{4.8}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.9}$$

The effects of a change to these modified equations is predictable: for the linear signal of Figure 4.10, inferior performance is expected as the signal solely contains pieces of constant velocity. If the operator input embodies quadratic shapes rather than linear ones, the 3DOF system improves the estimation. Equation 4.6 and Equation 4.7 are adopted for further purposes.

4.5.4 Comparison

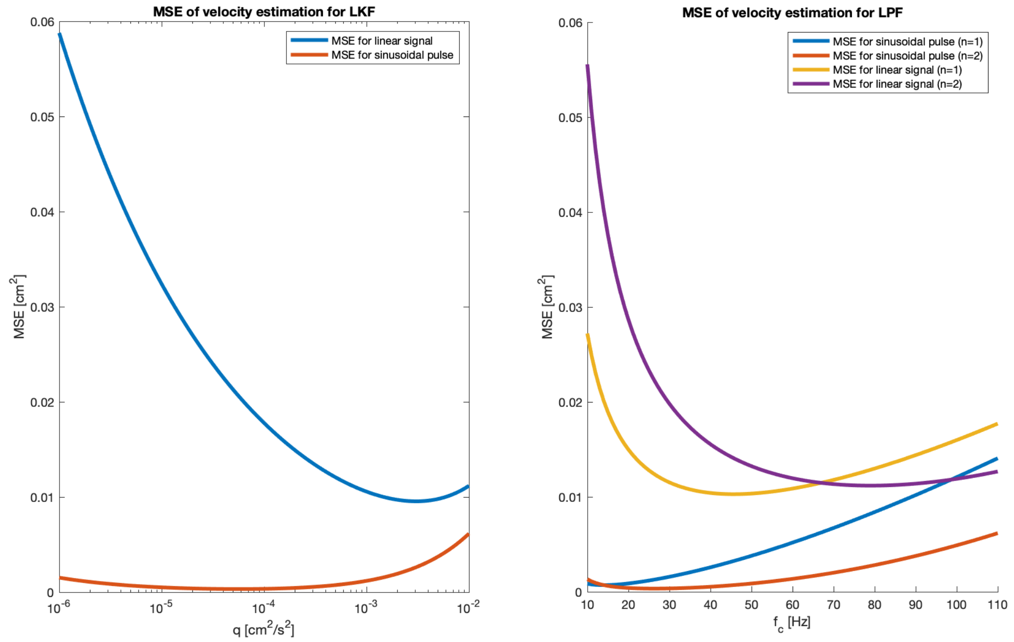
The MSEs are iteratively calculated for both systems as their parameters are altered. The LKF is tuned by shifting q from 10^{-6} to $10^{-2}\,\mathrm{cm^2/s^2}$, the LPF by ranging the cutoff frequency f_c from 10 Hz to 110 Hz. For the latter, favourable results are obtained with Butterworth filters of either first or second order n. The results of these tests are plotted in Figure 4.12a and Figure 4.12b respectively.

The following conclusions are drawn. Different filter orders are preferred for the two signals: the estimation of the sinusoidal pulse is more accurate with a second order filter while for the linear profile the first order filter performs better. At their optimal points, the KF performs slightly better. For the sinusoidal and linear signal 3% and 8% respectively. Considering that altering the filter characteristics on-line is to be omitted, two points are selected that compromise in terms of performance for both signals. The Butterworth filter is set to first order with $f_c = 300\,\mathrm{Hz}$ and for the Kalman filter $q = 10^{-3}\,\mathrm{cm^2/s^2}$. The Kalman's superior results are now more pronounced with improvements of 36% and 11% respectively. Additionally, the rise time of the LP solution is consistently larger, as shown in Figure 4.13. Similar rise times as those of the KF can only be achieved by applying LPF characteristics that propagate large amounts of noise resulting in unsuitable performance. The Kalman filter (KF) is adopted as solution for the derivation problem.

4.6 EKF improvement through ECG

As is elaborated in subsection 3.1.2, the ECG signal contains predictive information about the heart movement. Due to the lack of a dynamic ECG–heart-movement model, the assumption of a quasi-constant time delay $\Delta T_{\mathrm{ecg},p}$ between ECG peak and heart movement peak is made: $\Delta T_{\mathrm{ecg},p} \approx 200\,\mathrm{ms}$. The ECG signal is assumed the be available in real time. More particularly, the locations of the peaks in the QRS complexes are presumed to be known when, or shortly after, they occur. If these are not explicitly passed, they are calculated from the ECG signal by comparing the current value to the mean of the signal. A peak is detected when this value crosses a certain threshold (e.g. 150 mV).

The exploitation of the signal is twofold. It is in the first place used to predict on-line deviations from the model. The point in time where the ECG places the next motion peak is compared to the current expected peak according to the model. Secondly, it is also used to increase robustness by converting and inserting the measured heart rate directly into the EKF as angular frequency state ω. Generation of test data and the two mentioned exploitations are topics tackled in the following three subsections.



(a) Results for LKF

(b) Results for LPF

Figure 4.12: Comparison of LPF and LKF for velocity estimation.

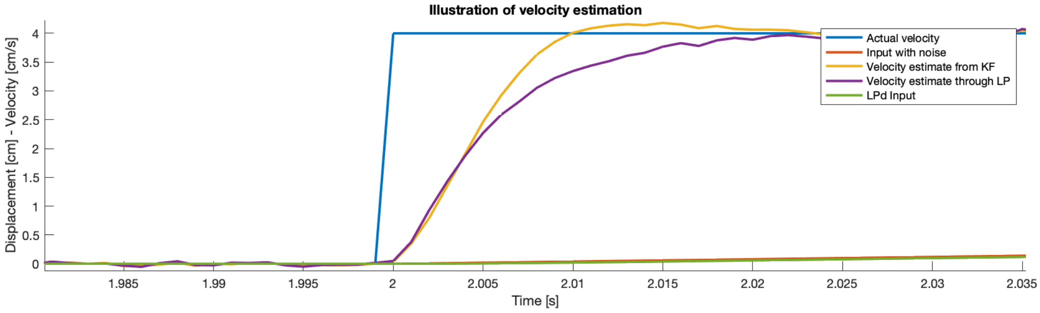


Figure 4.13: Velocity estimate rise times for the two discussed methods. Parameters: first order Butterworth LP, $f_c = 30\,\text{Hz}$. $q_{KF} = 10^{-3}\,\text{cm}^2/\text{s}^2$.

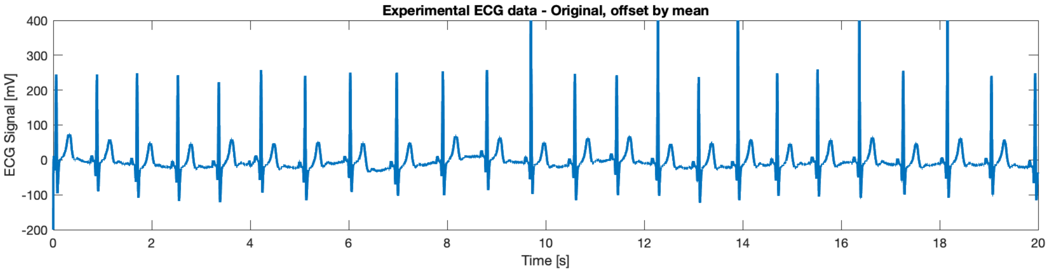


Figure 4.14: Original ECG signal

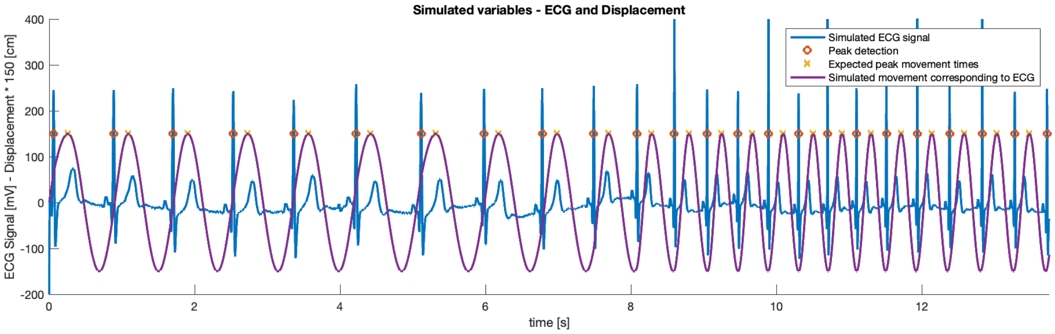


Figure 4.15: Manipulated ECG signal with detection of peaks and time of expected peak movement indicated. The first 5 s are the same as those from Figure 4.14, the next 5 s are used to scale up the heart rate to twice the nominal (from 70 BPM to 140 BPM approx.). Constant amplitude sinusoid (purple) with peaks corresponding to ECG signal.

4.6.1 Test data

Data allowing for testing the algorithm is created by altering experimental ECG data [23][4]. The data, displayed in Figure 4.14 is offset by its own mean value (518 mV). An interval of 20 s is selected where the heart rate is about 70 BPM. The [5 s, 10 s] interval is used to gradually speed up the frequency to twice the heart rate, which is maintained until the end. The resulting signal is 13.75 s long and shown in Figure 4.15. A threshold of 150 mV is selected for peak detection. The detections are indicated with ‘o’s, the expected peaks in the heart motion -200 ms later- with ‘x’. A sine wave of constant amplitude with peaks hitting these time marks is then created as a means of simulating the heart motion. These makings are shown in Figure 4.15 as well.

4.6.2 Heart rate linking

The extended Kalman filter (EKF) implemented in subsection 4.2.2 has the heart frequency ω explicitly present as a state in $\mathbf{x}$, the state vector. By setting it to the

---

[4] The D1NAMO dataset contains ECG data for research on diabetes. Data from the subset of healthy subjects is used.

measured heart rate from the ECG signal, the robustness of the filter is increased. A rapid raise in heart rate could for example initially be ‘explained’ by the EKF through the process noise of the offset, amplitudes and sine arguments, resulting in a lagging state estimation for ω. This phenomenon is to a small extent observed in Figure 4.6. Fixing this value directly also introduces a time delay, as the new frequency is only calculated after the QRS peak has passed. This delay is partially compensated by the time delay between the ECG signal and heart movement. In the lion’s share of the cases the resulting effect is modest as ω does not deviate much between subsequent periods.

This method is implemented and compared as follows. A simplified model of the earlier described EKF is implemented, omitting a second sine, the offset and amplitude. The resulting filter is iteratively applied to the simulated signal of Figure 4.15 and distorted with additive Gaussian measurement noise with variance $R_0 = 10^{-2}\,\mathrm{cm}^2$. The diagonal elements of Q_0 have been optimised separately, resulting in $\mathrm{Diag}(Q_0) = (1.3\,\mathrm{rad}^2/\mathrm{s}^2, 0.04\,\mathrm{rad}^2)$.

Subsequently, the alternative algorithm with direct ω feed is implemented and Q_0 (with only one diagonal element, in position $[2, 2]$) is optimised as well, resulting in $Q_{0,(2,2)} = 0.04\,\mathrm{rad}^2$. The resulting displacement estimation in terms of MSE is consistently about 10% better[5].

4.6.3 Motion prediction

The next expected peak of the current model maintained by the EKF is known from the states, as is the approximate peak in heart movement due to the ECG signal –about 200 ms in advance. The corresponding phase difference $\phi_{\Delta,k}$ is a measure for the current state error being made by the filter. Fixing the sine argument to match the expected motion eliminates every degree of freedom of the sine, restricting the filter’s possibilities to cope with real world process uncertainties. Instead the process noise covariance matrix Q_0 is tuned on-line, resulting in a time dependent Q_k.

Q_k is increased linearly with increasing $\phi_{\Delta,k}$. This results in Equation 4.10.

$$Q_k = Q_0(1 + c|\phi_{\Delta,k}|) \tag{4.10}$$

Q_0 as well as c are iteratively altered in a search for optimal values, leading to $Q_{0,(2,2)} = 0.03\,\mathrm{rad}^2$ and $c = 2\,\mathrm{rad}^{-1}$. The resulting system now consistently performs about 20% better than the one omitting ECG information.

Analysis of the resulting signals reveals that improvements are predominantly present when the heart rate increases, which is where the standard EKF output deviates most from the actual signal. This phenomenon is perceived in Figure 4.16.

4.6.4 Drawbacks

There are some drawbacks to the implemented ECG-exploitations. First of all, fixing one of the states based on a discrete event, and concomitantly taking away

---

[5] This result varies slightly between simulations due to the stochastic measurement noise.

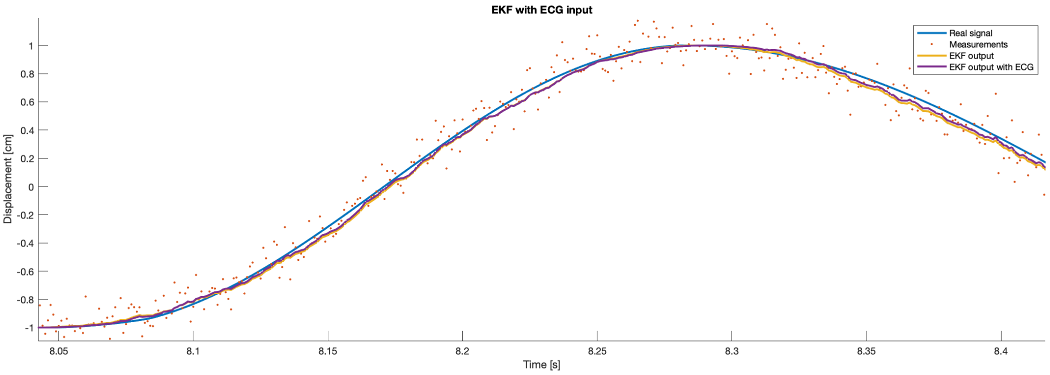


Figure 4.16: Zoom on simulation in the particular region where the heart rate rapidly increases. The deviation of the standard EKF signal (yellow) and improvement of the improved one (purple) are observed, especially where the former deviates most.

its degree of freedom, interferes with robustness in unpredicted scenarios. The response towards a sudden shock (step) for example shows more ‘inertia’, resulting in a larger tracking error. Fixing the frequency does increase robustness in the nominal case, assuming the ECG frequency is correct. Robustness is here interpreted as the rejection of convergence to erroneous states. E.g. when the frequency estimate is too high/low and the error is captured by fluctuation in offset and amplitudes. A second problem is that intentionally increasing the process noise covariance matrix when convergence to the accurate states has been achieved, increases the chances of nonetheless ending up with anomalous states. The third and last problem is the presence of the aforementioned inherent time delay of the ECG frequency estimation, due to the fact that it is only computed once a peak is detected. This delay is of little consequence in the nominal case, but when the heart beat rhythm suddenly changes, it ensures a delay in the estimated frequency. Consequentially, the motion tracking has the propensity to lag in those cases.

These three problems are tackled by exploiting the ECG signal in a more robust and elegant fashion: a fusion of the two measurements with respect to the model.

4.7 Sensor fusion

Once again, the dynamic ECG–heart-motion model is presumed to be known with a constant time delay $\Delta T_{\mathrm{ecg},p}$ between ECG peak and peak in heart movement. Based on these detected ECG peaks, a deterministic reference signal is designed which serves as second measured signal, in parallel with the measured displacement. Triangular pulses that start when the peak is measured are used to this end. The measurement function is altered in order to, apart from the expected displacement, also calculate the expected triangular signal. Because this introduces a numeric aspect to the implementation, the extended Kalman filter (EKF) is replaced by an unscented Kalman filter (UKF).

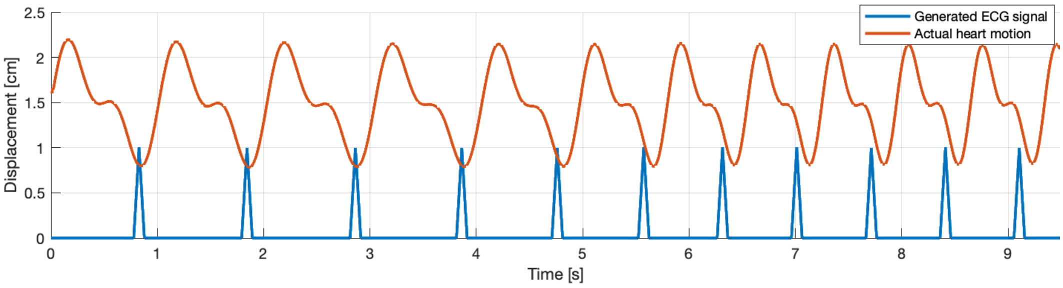


Figure 4.17: The generated signals for the UKF sensor fusion. The generated movement (red) and triangular ECG pulses (blue) are shown.

4.7.1 ECG signal

Because a signal with discrete function value differences such as the Dirac delta function is not suitable for this purpose, the detected peaks are translated into a continuous signal[6] which the UKF produces as well. A triangular pulse train is used where the pulses are placed immediately upon detection of the ECG peaks.

The used displacement signal is created with $\mathrm{Diag}(Q_0) = (10^{-7}\,\mathrm{cm}^2, 10^{-7}\,\mathrm{cm}^2, 10^{-7}\,\mathrm{cm}^2, 10^{-5}\,{}^{\mathrm{rad}^2}\!/\!_{\mathrm{s}^2}, 0\,\mathrm{rad}^2, 0\,\mathrm{rad}^2)$, and the frequency is artificially increased by $3\,{}^{\mathrm{rad}}\!/\!_{\mathrm{s}}$ in $3\,\mathrm{s}$ starting at $t = 4\,\mathrm{s}$. The resulting generated states are plotted in Figure 4.18 (yellow). Measurement noise with $R_{0,\mathrm{disp}} = 10^{-3}\,\mathrm{cm}^2$ is added. The generated triangular signal and displacement are shown in Figure 4.17. The time delay $\Delta T_{\mathrm{ecg},p}$ is increased to 400 ms for three reasons. The 200 ms that was assumed before was based on the heart beat of an animal with a significantly faster frequency than the average human one. Physiologically, the ECG peaks are registered *before or with* the upward heart movement (as is now the case in Figure 4.17) and, lastly, because it makes the observed effect of the sensor fusion more pronounced. The generated triangular pulses are symmetrical, 100 ms wide and have unit height.

4.7.2 UKF design

The measurement function which is maintained by the ‘Sensor Fusion UKF’ is given by algorithm 4. It now returns two outputs. First the expected displacement is calculated. What remains serves the purpose of calculating the expected triangular pulse value in function of the current state. First the search interval $\mathrm{T}_{\mathrm{int}}$ is limited to one period of the signal, symmetrically around the current time step. A vector $\mathbf{\Delta T}_{\mathrm{rel}}$ is created which contains the relative time differences corresponding to the current time step for every element of the period $\mathrm{T}_{\mathrm{int}}$. This symmetrically generated interval is offsetted by the maintained ECG triangle–motion-peak lead. The estimated displacement signal **localWave** is created with respect to the current states $\mathbf{x}_k$. The maximum of the wave is found along with its corresponding relative time difference

---

[6] I.e. a continuous discrete signal: the signal itself is discrete but the subsequent values are close to one another.

$\Delta\mathrm{T}^*_{\mathrm{rel}}$. This is then used to calculate the expected current value of the triangular signal $\Delta_{\mathrm{ecg},k}$. Any undefined functions in algorithm 4 refer to MATLAB syntax.

The values used in algorithm 4 for generation of the triangular profile (0.35, 0.05 and 20 respectively) correspond to these elaborated in subsection 4.7.1.

In addition to the mentioned changes, the measurement variance is replaced by a measurement covariance matrix:

$$R_0 = \begin{bmatrix} R_{0,\mathrm{disp}} & 0 \\ 0 & R_{0,\mathrm{ecg}} \end{bmatrix} \tag{4.11}$$

Algorithm 4: Measurement Function (UKF Sensor Fusion)

```
function measurementFunction($\boldsymbol{x}_k$):
    $\hat{z}_k = x_{1,k} + x_{2,k}\sin x_{5,k} + x_{3,k}\sin x_{6,k}$

    $\mathrm{T}_{\mathrm{int}} = 2\pi/x_{4,k}$
    $\boldsymbol{\Delta}\mathbf{T}_{\mathrm{rel}} = (-\mathrm{round}(\mathrm{T}_{\mathrm{int}}/2, 3) + 0.35 : \Delta t : \mathrm{round}(\mathrm{T}_{\mathrm{int}}/2, 3) + 0.35)$
    $\mathbf{localWave} = x_{2,k}\sin(x_{5,k} + x_{4,k}\boldsymbol{\Delta}\mathbf{T}_{\mathrm{rel}}) + x_{3,k}\sin(x_{6,k} + 2x_{4,k}\boldsymbol{\Delta}\mathbf{T}_{\mathrm{rel}})$
    $[\sim, \mathrm{ind}] = \max(\mathbf{localWave})$
    $\Delta\mathrm{T}^*_{\mathrm{rel}} = \boldsymbol{\Delta}\mathbf{T}_{\mathrm{rel}}(\mathrm{ind}) - 0.35$

    if $\Delta\mathrm{T}^*_{\mathrm{rel}} < 0.05$ && $\Delta\mathrm{T}^*_{\mathrm{rel}} \geq 0$ then
        $\Delta_{\mathrm{ecg},k} = 20(-\Delta\mathrm{T}^*_{\mathrm{rel}} + 0.05)$
    else if $\Delta\mathrm{T}^*_{\mathrm{rel}} > -0.05$ && $\Delta\mathrm{T}^*_{\mathrm{rel}} \leq 0$ then
        $\Delta_{\mathrm{ecg},k} = 20(\Delta\mathrm{T}^*_{\mathrm{rel}} + 0.05)$
    else
        $\Delta_{\mathrm{ecg},k} = 0$

    return $[\hat{z}_k, \Delta_{\mathrm{ecg},k}]'$
```

4.7.3 Insights

The performance of the filter in the context of the additional functionality is influenced by the triangular signal and the measurement noise $R_{0,\mathrm{ecg}}$. Placing the triangles earlier increases the predictive aspect of the filter. This placement is physiologically limited by $\Delta T_{\mathrm{ecg},p}$. Decreasing the width makes the momentary influence of the ECG signal shorter. A longer triangle manipulates the states (mostly the frequency) for a longer period in time. As the pulse is a reflection of a momentary detection (the ECG peak), this manipulation becomes more irrelevant the longer it lasts. On the other hand, the shorter the pulse is the less time the filter is permitted to adapt to the new measurements. When it comes to the measurement noise $R_{0,\mathrm{ecg}}$, a larger value implies more measurement noise and less influence by the ECG. Since no measurement but only discretisation noise is present, the value is chosen relatively low, presuming the ECG detections and the dynamic ECG–heart-motion model are accurate.

4.7.4 Simulation

The resulting states for the simulation with $R_{0,\text{ecg}} = 10^{-5}$ are shown in Figure 4.18, along with the states of the generated signal and the estimation by the conventional UKF. The corresponding triangular pulses with a detailed pulse are also plotted in Figure 4.19. Each ECG detection is indicated by a grey dashed line. The small lasting difference between the pulses results from the difference between the peak search algorithms. The UKF finds the top of the curve analytically, based on the states, while the ECG signal is generated by locating the actual top. The latter is partly determined by the process noise as well. This characteristic allows the difference to be perceived as a sort of noise on the dynamic ECG–heart-motion model.[7] In the plot of the states, the frequency in particular, the updates due to the ECG are perceivable. As the model presumes a constant frequency, the states with sensor fusion tend to converge to those without after the detected peak has passed. This behaviour is influenced by pulse shape, as discussed before.

The improvement of the sensor fusion algorithm over the conventional UKF in terms of MSE is significant for the displacement tracking (35% improvement) as well as for the estimated velocity (50% improvement). The conventional UKF disposes of the possibility to more rapidly respond to the increase in heart frequency than is portrayed here, provided an increase in the variance of the frequency state is carried through. This however comes at the cost of deteriorated performance in nominal scenarios.

4.8 Conclusion

In correspondence with chapter 2, an extended Kalman filter (EKF), unscented Kalman filter (UKF) and MHE are implemented and compared. Based on the results, the EKF is selected for further use. Next a linear Kalman filter (LKF) is designed and implemented with the purpose of providing an on-line differentiation estimate. This system is compared with the more conservative numeric differentiation and has proven to perform superiorly. The chapter is concluded by exhausting the possibility of using an ECG signal to improve filter performance. Simplistic methods as well as sensor fusion are used and compared. The latter yields significant improvements in comparison with the former.

---

[7] I.e. this inaccuracy can be omitted but this small difference is more realistic than a perfect prediction.

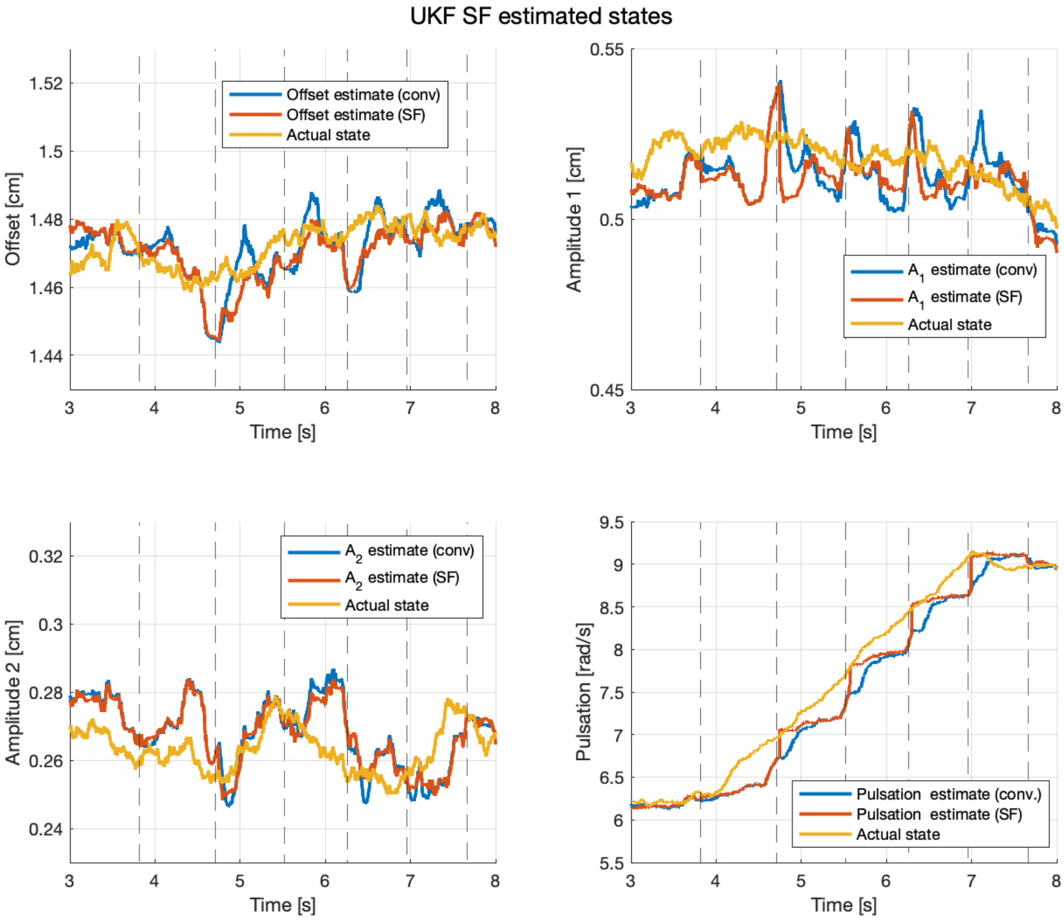


Figure 4.18: Actual (generated) states (yellow) along with the states estimated by the UKF with sensor fusion (red) and those by the conventional UKF (blue). The time steps where the ECG peaks are registered are shown by the grey dashed lines.

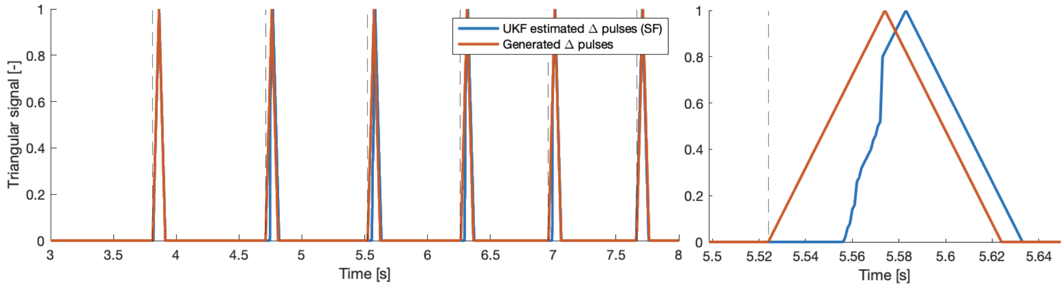


Figure 4.19: UKF sensor fusion triangular pulses. The measured pulse (red) and estimated pulse (blue) are shown. The blue pulse adapts to match the red one. The lasting difference is explained in subsection 4.7.4.

Chapter 5

Robot Control

5.1 Introduction

This chapter introduces and compares strategies concerning the control of the surgical robot. More specifically, two methods are applied. First the more conservative proportional-integral-derivative (PID) control is implemented. Then, exhausting the possibility of prediction, model predictive control (MPC) is applied. The two strategies are briefly explained, applied to the motion-compensation system and compared with different tuning parameters. These simulations are repeatedly performed making use of an ideal and a more realistic plant model. *The numerical results serve solely as a means of comparing the two systems. Performance is heavily dependent on the deployed hardware and further mechanical design choices.*

5.2 Proportional control

5.2.1 Concept of PID control

A PID controller is a feedback controller that (in the most general case) applies a three-term control formula to the measured error value $e(t)$, the difference between the reference value and measured variable. As the name divulges, this formula consists of three terms as shown in Equation 5.1 (in the s-domain)[1]. General feedback control loops adopt the scheme of Figure 5.1.

$$C(s) = k_p + k_d s + \frac{k_i}{s} \tag{5.1}$$

5.2.2 Proposed P(I) controller

A variant of this controller is introduced. The scheme is is shown in Figure 5.2 (marked in pink) along with the earlier described KFs of chapter 4. The derivative term is omitted as it yields little to no improvement. For simplicity, the integral term is ignored as well at first. The plant is presumed to have an internal velocity

---

[1] I.e. the Laplace domain where $s = \omega j$.

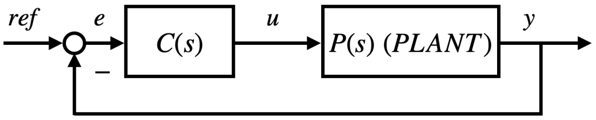


Figure 5.1: Feedback control scheme. $C(s)$ is the control formula, $P(s)$ (a model of) the plant.

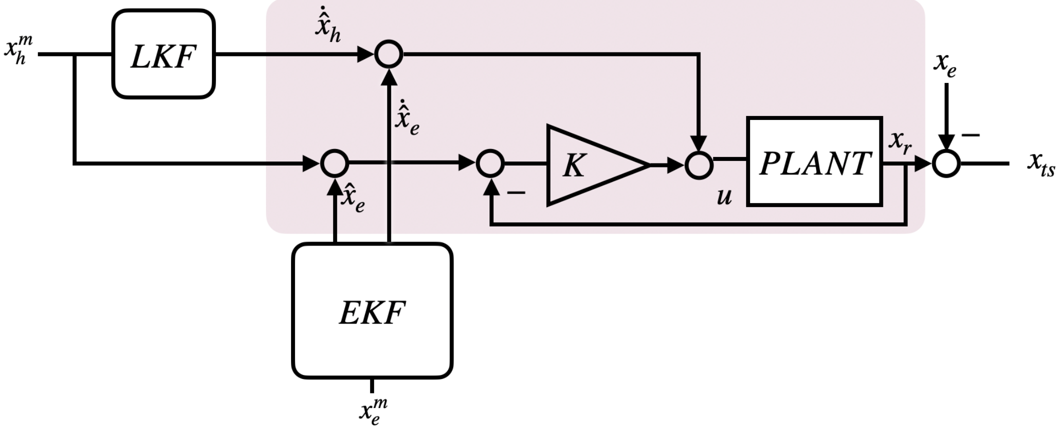


Figure 5.2: Combination of proportionate and feedforward control integrated with KFs. The control loop itself is marked in pink. LKF and EKF denote the linear and extended Kalman filter respectively. PLANT the robot, velocity controller inclusive.

controller. The proposed system uses the velocity as feed-forward signal. The feedback loop attempts to compensate for errors of the feed-forward control by responding to the observed tracking error. Under the assumption of perfect estimation, and considering the plant to be an ideal integrator $1/s$, the transfer function (TF) of the system depicted in Figure 5.2 is Equation 5.3 and can be compared with the basic proportionate feedback control Equation 5.2 (proportionate control-formula case of Figure 5.1). In this ideal case, the improved control loop delivers perfect tracking as the TF equals 1. Equation 5.2 however has one pole and low-pass characteristics.

$$\frac{Y(s)}{X(s)} = \frac{P(s)}{1+KP(s)} \xrightarrow{P(s)=s^{-1}} \frac{1}{s+K} \tag{5.2}$$

$$\frac{Y(s)}{X(s)} = \frac{P(s)(s+K)}{1+KP(s)} \longrightarrow \frac{s+K}{s+K} = 1 \tag{5.3}$$

Ab initio, $\dot{\hat{x}}_e$ is calculated analytically from the estimated states according to Equation 5.4. This is readdressed in section 5.5.

$$\dot{\hat{x}}_e = \hat{x}_2\hat{x}_4 cos(\hat{x}_5) + 2\hat{x}_3\hat{x}_4 cos(\hat{x}_6) \tag{5.4}$$

5.3 Model Predictive Control (MPC)

5.3.1 Concept of MPC

Model predictive control (MPC) is a strategy which employs an internal model of the plant in order to predict output behaviour. It repeatedly solves an optimisation problem in which it minimises a cost function for the plant's control signal. For the case of tracking a constant signal this for instance often comes down to minimising the difference between the reference and the expected output for the following N_{PH} values. At each time step the control value for the current time step, calculated during the previous one, is applied.

The concept is illustrated by Figure 5.3. MPC is a multivariable control algorithm with behaviour dependent on the following parameters.

Time step Δt_{MPC} This is the time step used by the controller. It is the interval within which the control signal remains constant. A smaller Δt_{MPC} increases performance. It establishes a more precise control signal and reduces the control delay. This delay equals Δt_{MPC} and is inherently present in the system as Δt_{MPC} is the time needed to solve the optimisation problem. Consequently, the time step must also be chosen large enough to allow for the optimiser to solve the problem.

Prediction and control horizon N_{PH} and N_{CH} (see Figure 5.3) can take different values, with $N_{\text{PH}} \geqslant N_{\text{CH}}$. The prediction horizon is the amount of time steps maintained for solving the optimisation problem and thus simulating the future response of the model. The control horizon is the amount of time steps over which the predicted control signal is allowed to vary. An increase of either value improves control accuracy at the cost of computational complexity.

Constraints The control effort as well as the output of the plant can be subject to constraints. These manifest themselves in a constrained optimisation problem, which in turn yields a higher computational cost.

Cost function The to-be-optimised cost function J_{MPC} determines the objective of the optimisation. It for instance reflects the relative importance between reference tracking and minimising the control effort. J_{MPC} can also be used to assign different weights to steps in the prediction horizon. E.g. the first couple of steps are made more important.

5.3.2 Proposed MPC controller

Before applying the controller, reference predictions r have to be calculated. To calculate the $k+j$th prediction at time step k, Equation 5.5 is deployed where k_{MPC} and k_{EKF} correspond to the current (possibly distinctive) time intervals. Additionally, as a means of countering the MPC's inherent time delay, the predictions are shifted by Δt_{MPC}, resulting in Equation 5.6. If the MPC and EKF time steps are equal,

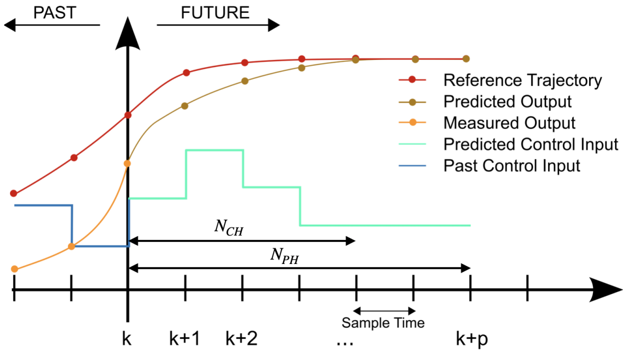


Figure 5.3: Schematic illustration of MPC method. Control and prediction horizon are denoted by N_{CH} and N_{PH} respectively.

the 'EKF' and 'MPC' subscripts in Equation 5.5 become irrelevant as is assumed in Equation 5.6. These equations correspond to the 'predictions' block of Figure 5.4. The cost function maintained by the controller's optimiser is given by Equation 5.7. x_{pj} is the simulated output of the system depending on the optimised variable (the p-subscript denoting 'predictive'), the velocity control signal is u. This signal is not directly present in the cost as power consumption is assumed to be irrelevant in contrast to tracking accuracy. For the magnitude and rate of change of u however, limitations are imposed on $u_{\min}, u_{\max}$ and $\dot{u}_{\min}, \dot{u}_{\max}$ respectively, as the system's dynamics are limited. [24]

$$
\begin{aligned}
r_{k_{\text{MPC}}+j} &= \hat{x}_{k_{\text{EKF}}1} + \hat{x}_{k_{\text{EKF}}2} sin(\hat{x}_{k_{\text{EKF}}5} + j\Delta t_{\text{MPC}} \hat{x}_{k_{\text{EKF}}4}) + \\
&\qquad \hat{x}_{k_{\text{EKF}}3} sin(\hat{x}_{k_{\text{EKF}}6} + 2j\Delta t_{\text{MPC}} \hat{x}_{k_{\text{EKF}}4}) && (5.5) \\
r_{k+j-1} &= \hat{x}_{k1} + \hat{x}_{k2} sin(\hat{x}_{k5} + j\Delta t \hat{x}_{k4}) + \hat{x}_{k3} sin(\hat{x}_{k6} + 2j\Delta t \hat{x}_{k4}) && (5.6) \\
J_{\text{MPC}} &= \sum_{j=1}^{N_{\text{PH}}} (r_j - x_{pj})^2 && (5.7)
\end{aligned}
$$

The resulting control system is integrated as schematically depicted by Figure 5.4. The EKF's estimated states $\hat{x}$ are used to make N_{PH} predictions r. The current human (operator) input x_h^m is added before they are fed to the MPC controller, which also takes the measured output (MO) into account. Based on these the manipulated variable (MV) u is calculated and used to control the plant.

5.4 Simulations

A combination of MATLAB and Simulink is used to validate the designed systems and tune the parameters. The same signals are applied with identical noise, the KFs are deployed and in the first instance an ideal plant model is maintained.

The operator input of section 4.5 is used, together with the generated heart movement of Figure 5.5 (in red, behind purple). Gaussian noise is added to the signal

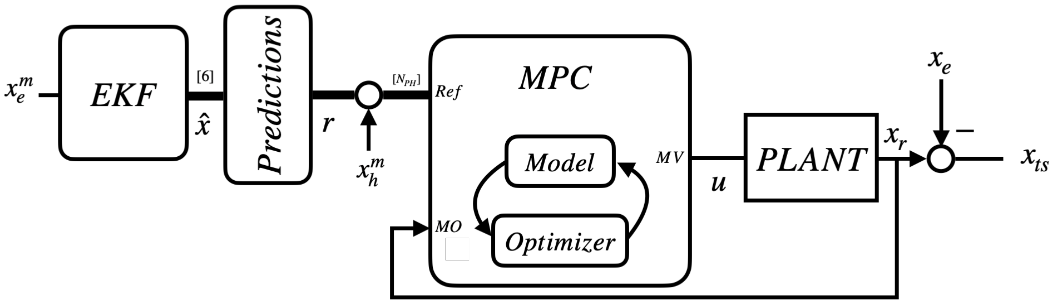


Figure 5.4: Integrated MPC controller. The estimated states $\hat{\mathbf{x}}$ of the EKF are used to calculate N_{PH} predictions r_i of the environmental movement x_e. The current human (operator) input x_h is added before the values enter the MPC. The controller respectively has two inputs and one output: the reference signal, measured output (MO) and manipulated variable (MV).

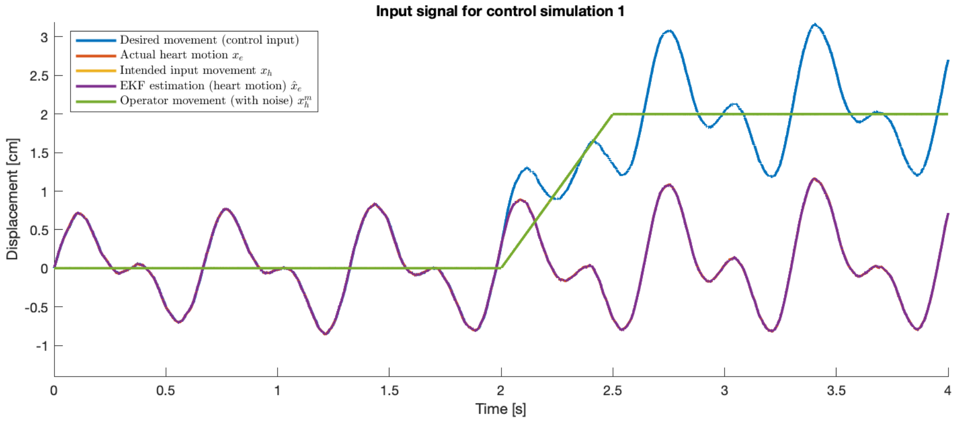


Figure 5.5: Control simulation input displacements. The EKF displacement estimation (purple) sums with the operator's desired movement (green) to result in the desired robot movement (blue).

with $R_0 = 10^{-4}\,\text{cm}^2$. The plot also shows the resulting desired robot movement (blue).

5.4.1 Ideal plant simulations

P(I) control The implemented proportionate control loop is first tested with different gains. Increasing the gain K monotonously increases the tracking accuracy (until it bumps into the limitations concerning discretisation noise, which occur around $K_{\text{instable}} = 500\,\text{s}^{-1}$). The results for $K = 10\,\text{s}^{-1}$ are plotted in Figure 5.6. The MSE of the resulting error signal is $6.8 * 10^{-4}\,\text{cm}^2$. The results for a more conservative ($K = 2\,\text{s}^{-1}$) and higher gain ($K = 20\,\text{s}^{-1}$) are shown in the left table of Table 5.1. Also included are results for a system with additional integral action: a PI controller. The achievable improvements are often limited but notable. They are

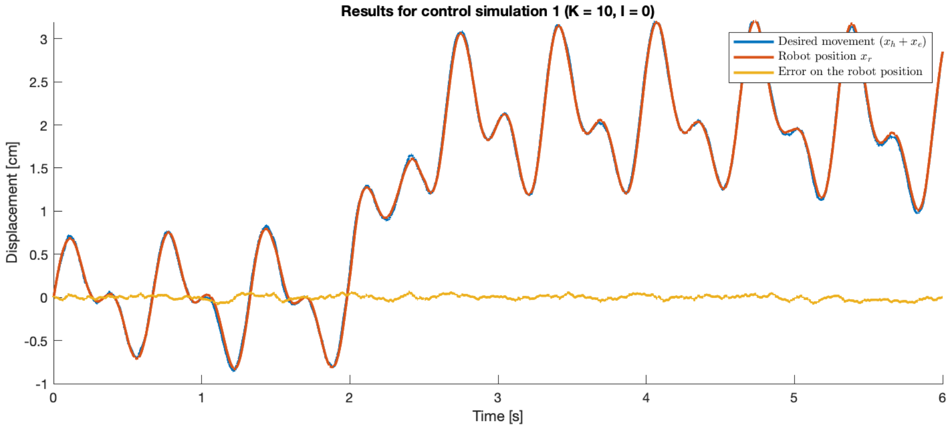


Figure 5.6: Proportionate control simulation results. The desired movement (blue) which is only perceived by the system in the presence of measurement noise is compared to the resulting robot movement (red). The difference is also plotted (yellow).

in the range of $5 - 35\%$. The numerical values of the I-gains are chosen so that they either achieve the best performance, or that further increase only yields negligible improvement.

Model predictive control (MPC) As the ideal plant model is used in combination with shifted predictions (i.e. the MPC time delay is compensated as discussed in subsection 5.3.2 and shown by Equation 5.6), the controller achieves accurate tracking and the only errors made are the estimation errors made by the KFs and those that result from process noise. For a time step of 1 ms his results in an MSE of $3.5 * 10^{-5}\,\mathrm{cm}^2$, regardless of the chosen prediction horizon (PH) or control horizon (CH). A time step of 10 ms leads to $2.2 * 10^{-4}\,\mathrm{cm}^2$, mainly due to the delayed response to the operator input.

It is clear that this unrealistic virtual environment fails to expose real world behaviour of the intended systems. A more realistic simulation is envisioned which takes into account limited plant dynamics and imposes limitations on the control signal.

5.4.2 Limited dynamics simulation

In addition to the integrator, low-pass characteristics are added with a bandwidth of about 30 Hz, translating into a time constant $\tau = 5\,\mathrm{ms}$. The control effort and its rate are limited by $|u_{min,max}| < 0.5\,\mathrm{m/s}$ and $|\dot{u}_{min,max}| < 5\,\mathrm{m/s^2}$. These values are chosen with respect to the achievements of modern high performance linear actuators that could e.g. be used as robot end effector [25, 26]. They are not specifically related to any equipment or scenario but intend to simulate an environment which allows for comparison of the control systems. The same test signals are used.

Ideal plant		
Control Gains		
$K[\mathrm{s}^{-1}]$	$I[-]$	MSE [cm^2]
2	0	$3.6 * 10^{-3}$
	5	$2.3 * 10^{-3}$
10	0	$6.8 * 10^{-4}$
	5	$6.3 * 10^{-4}$
20	0	$3.9 * 10^{-4}$
	10	$3.7 * 10^{-4}$

Non-ideal plant		
Control Gains		
$K[\mathrm{s}^{-1}]$	$I[-]$	MSE [cm^2]
2	0	$5.4 * 10^{-3}$
	2	$4.4 * 10^{-3}$
10	0	$2.1 * 10^{-3}$
20	0	$1.2 * 10^{-3}$
100	0	$2.2 * 10^{-4}$

Table 5.1: Results for P(I)-control simulations with ideal (left) and more realistic (right) plant model. No integral action is added if no improvement was obtained.

P(I) control The right table in Table 5.1 shows the results for these more realistic simulations. The results are directly comparable with those of the left table, obtained through the ideal plant model. Adding integral action with proportionate gains $K = 10\,\mathrm{s}^{-1}$ and $K = 20\,\mathrm{s}^{-1}$ yields no further improvement. These results are therefore omitted. The peak-to-peak errors corresponding to $K = 2\,\mathrm{s}^{-1}; I = 2$, $K = 20\,\mathrm{s}^{-1}; I = 0$ and $K = 100\,\mathrm{s}^{-1}; I = 0$ are 3.5 mm, 1.8 mm and 0.8 mm respectively.

Furthermore, the systems closed loop stability is -making use of the current model- favourable. The input to a step input (consequently without feed-forward) has no overshoot for $K = 20\,\mathrm{s}^{-1}$ and a 4% overshoot for $K = 100\,\mathrm{s}^{-1}$.

Model predictive control (MPC) The simulations are performed with an internal MPC time step Δt_{MPC} of either 1 ms or 10 ms. The aforementioned restrictions are imposed and the prediction and control horizon (N_{PH} and N_{CH}) are varied. The simulation occupies 10 s of 'simulated time' and the input signals are sampled at $\Delta t = 1\,\mathrm{ms}$. The results are summarised in Table 5.2. This table shows the simulation time ΔT_{sim}[2], the MSE of the error between the eventual robot position and the original (unbiased) heart motion, the peak-to-peak error ϵ_{pp} and the overshoot Δh of the output towards the internal plant, in response to a step input.

The results show that the prediction and control horizons must be selected sufficiently high in order ensure stability. '*unstable*' denotes behaviour of which the apparent 'damping' is insufficient and the excitation due to a step response does not fade. This value is dependent on the phase delay of the plant. A controller with time step $\Delta t_{\mathrm{MPC}} = 1\,\mathrm{ms}$ applied to the current plant ($\tau = 5\,\mathrm{ms}$) reaches this kind of stability with a horizon of 6 ms for instance, whereas the same controller applied to a plant with slower dynamics ($\tau' = 20\,\mathrm{ms}$) only reaches with $N_{\mathrm{PH}} = 11$. This is an important parameter to take into account when designing an MPC for a specific plant.

---

[2] The simulations were 'warm-executed' on a machine featuring a 6th generation 'Skylake' 2.6GHz quad-core i7 processor with Radeon Pro 450 dGPU, in similar circumstances.

Parameters		Output measurements			
Δt_{MPC}[ms]	$N_{\mathrm{PH}}/N_{\mathrm{CH}}$	ΔT_{sim}[s]	MSE [cm^2]	ϵ_{pp}[mm]	$\Delta h[-]$
10	1/1		*unstable*		
	2/2	1.0	$3.5 * 10^{-4}$	1.4	35%
	10/10	1.3	$3.5 * 10^{-4}$	1.4	6%
1	2/2		*unstable*		
	8/8	4.0	$7.9 * 10^{-5}$	0.6	18%
	10/10	4.3	$7.8 * 10^{-5}$	0.6	13%
	20/20	7.5	$7.8 * 10^{-5}$	0.6	4%
	50/50	9.4	$7.8 * 10^{-5}$	0.6	4%

Table 5.2: Results for MPC simulation with non-ideal plant model.

Once favourable behaviour is reached, further increasing the horizon yields elevated robustness but little further improvement on the error in the simulation. The computational effort however rapidly increases. When the horizon is further expanded the performance has the tendency to slowly deteriorate. This is explained by the fact that the errors of the movement predictions r that are further in the future increase due to process noise.

5.5 Unmodelled behaviour

This section provides a fully integrated simulation for compensation of motion that is not comprehended by the internal model. Operator input is omitted. The systems are compared performance-wise and further safety measures are discussed.

5.5.1 Simulation

A signal is composed which lacks one of the heart movement beats. It is shown in blue in Figure 5.8. Due to the inconsistency of the provided model the states calculated by the EKF are inaccurate, even though the error of the estimated movement is limited (see Figure 5.9a in green). The proposed PID and MPC controllers are for their functioning dependent on these states. The PID controller uses them to analytically calculate a velocity estimate through Equation 5.4 and the MPC employs them in order to calculate reference predictions. I.e. the control strategy is threatened in these unmodelled circumstances. The effects are investigated by means of a simulation using the discussed signal. The PID and MPC results are shown in Figure 5.8, Figure 5.9a and Figure 5.9b in purple and red respectively. The MPC uses $N_{\mathrm{PH}} = N_{\mathrm{CH}} = 10$ with $\Delta t_{\mathrm{MPC}} = 1\,\mathrm{ms}$ and the PID has $K = 20\,\mathrm{s}^{-1}$. The latter shows a large peak which tends to follow the unmanipulated signal: the state estimates are non-conform and bias the feed-forward signal. Similarly, the MPC predictions r amplify the EKF error, albeit to a smaller extent.

The need for the possibility of more robust control rises. One solution is to use the linear Kalman filter (LKF) of section 4.5 to estimate $\dot{\hat{x}}_e$ instead of calculating it analytically. These results are shown in yellow. This allows the PID to even outperform the MPC in terms of 'inconsistency rejection', against the cost of an inherent (section 4.5) time delay, less accuracy and modestly higher computational complexity. In order to observe the impact of this change, the simulations of section 5.4 are repeated with the same signal, using $K = 20\,\mathrm{s}^{-1}$ and $I = 0$. The operator input is omitted. The respective mean square errors (MSEs) for LKF and analytical estimation are $4.0 * 10^{-3}\,\mathrm{cm}^2$ and $1.2 * 10^{-3}\,\mathrm{cm}^2$. I.e. this solution has a significant negative impact on the overall performance in nominal circumstances.

5.5.2 On-line consistency test

The epicentre of the problem corresponds to the ability of making an algorithmic distinction between nominal and unmodelled measurement. This functionality creates the possibility of on-line shifting to the optimal control strategy. If the PID is used, the velocity estimation is for instance ideally calculated analytically in the nominal case and through the LKF elsewhere. The foundations of an appropriate consistency test have been introduced in section 2.2.6.

Figure 5.10 shows the EKF measurement and output for a signal similar to the one of Figure 5.9a, together with it's SNIS for $M = 5$ which corresponds to Equation 2.22. In this example, 10 ms of SNIS values outside of the 99% confidence interval is used as a criterion to detect severe inconsistencies. Figure 5.10 also shows when inconsistency is flagged according to this criterion. Consequently, the control strategy can upon detection be changed to a more robust one for a determined time, until certain 'safety conditions' are fulfilled. Selection of M, the confidence interval and the time interval to pass before intervention are trade-offs between responsiveness and robustness of the selection.

5.5.3 Predictive ECG peaks

Section 4.6 explains how by comparing the detected ECG peaks with the expected location of the next movement peak, the process noise covariance matrix is altered which leads to improved tracking. Similarly, if the heart skips a beat, the lack of such peak is available information and can be used as an indicator for a switch to a different control strategy.

For the model that has been maintained so far, at discrete time step k, the expected location of the next peak is calculated by solving Equation 5.8 in interval Equation 5.9. It is expressed by t_Δ, the time difference between the current time and that of the next expected peak. A constant delay between ECG peak and movement peak $\Delta T_{\mathrm{ecg,p}}$ of 200 ms is assumed here. A time interval (see Figure 5.7, grey dashed line) is imposed after which the heart beat is supposed to be skipped or unacceptably delayed. The control strategy is then altered accordingly. The applicability of this algorithm depends on $\Delta T_{\mathrm{ecg,p}}$ as well as the heart movement. In the particular case of Figure 5.7 for instance, the detection occurs too late and most of the damage

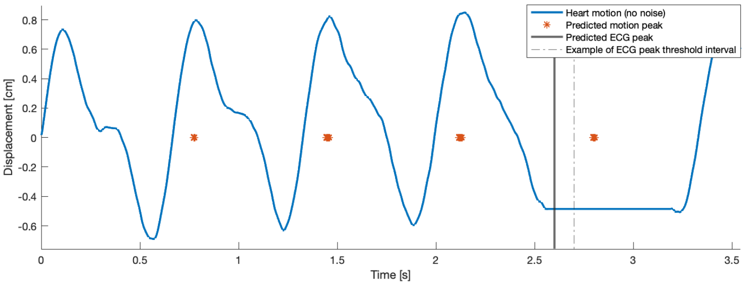


Figure 5.7: Heart skipping beat with indication of predicted peaks. The presumed location of the ECG peak is shown as well, along with an example of a threshold time for detection of a skipped beat.

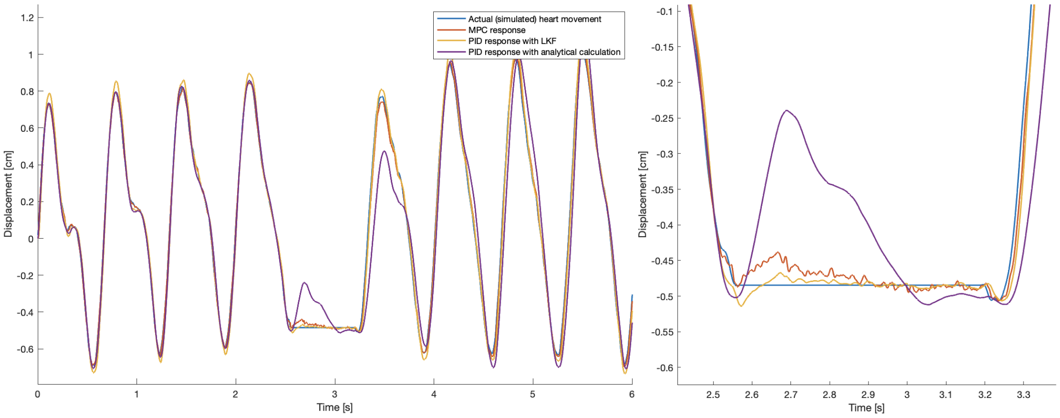


Figure 5.8: Performance comparison of different control systems in unforeseen (unmodelled) scenario. The PID systems maintain a gain of $K = 20s^{-1}$ without integrating action. The earlier discussed low-pass model is used.

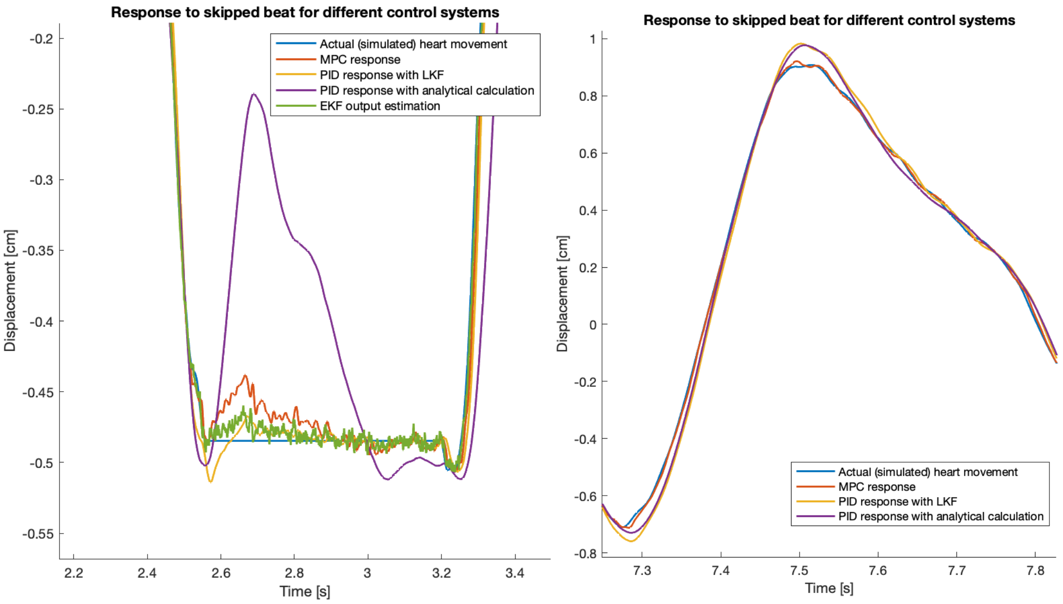
(see Figure 5.8) is already done. Here using the SNIS as detection method is more sensible.

$$a_{1k}\cos(\theta_{1k} + \omega_k t_\Delta) + 2\omega_k a_{2k}\cos(\theta_{2k} + 2\omega_k t_\Delta) = 0 \tag{5.8}$$

$$t_\Delta \in \left(\frac{2\pi - \theta_{1k}}{\omega_k}, \frac{3\pi - \theta_{1k}}{\omega_k} \right) \tag{5.9}$$

5.6 Conclusion

The feed forward P(I) controller uses conservative technology to maintain favourable results. Its performance is driven by the feedforward control and therefore the velocity estimation is of vital importance. A feedback circuit with gain K is used to compensate any deviations, which is why -supposing the velocity estimation is as good as possible- K determines the P-control performance. Increasing K increases



(a) Zoom on area of interest in Figure 5.8

(b) Zoom on signal of Figure 5.8

Figure 5.9

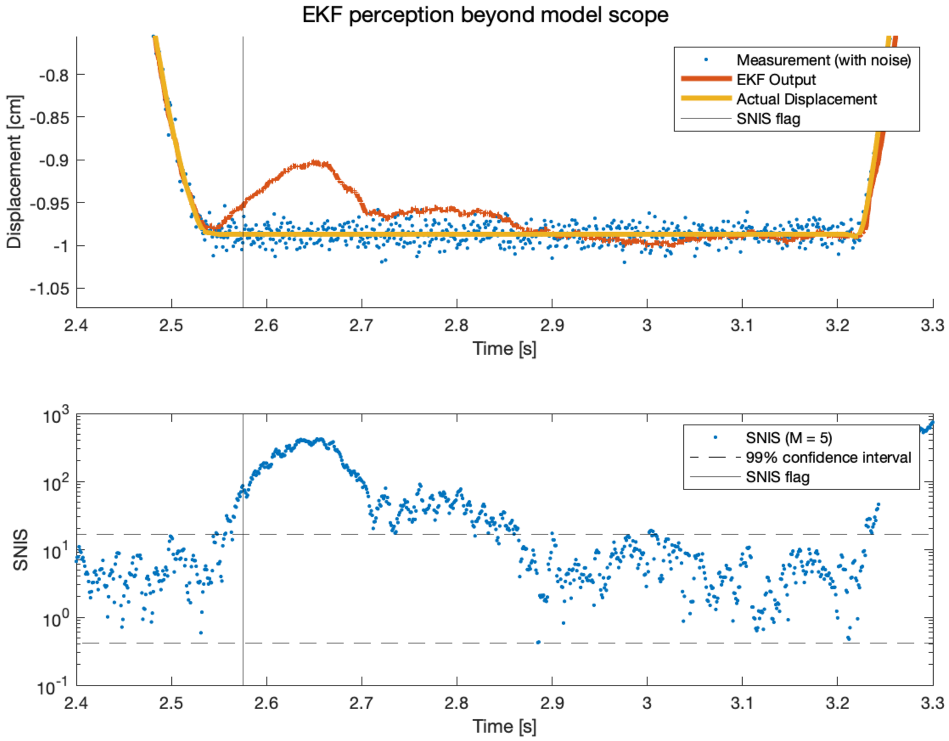


Figure 5.10: EKF output and SNIS for input beyond model scope. A vertical line indicates where, maintaining the elaborated criterion, anomalous circumstances are detected.

the accuracy. The MPC shows the potential to be a superior solution but is heavily subject to the availability of computational power.

After establishing this, attention is paid to behaviour in circumstances outside the model's scope. It was shown that the PID controller only provides acceptable results in this scenario if the velocity is estimated differently. To this end an LKF is applied which negatively impacts overall performance. Therefore, the chapter is concluded by showing results for an on-line consistency test which allows for on-line distinction between control strategies.

The trends exposed in this section are in line with those obtained with the sinusoidal pulse as operator input. They also hold for in/decreased noise on the signals.

Chapter 6

Three-Dimensional Motion Tracking

6.1 Introduction

Taking into account the objective of designing a robot motion-compensation system for a surgical platform, the expansion of the one-dimensional considerations of the previous chapters towards a three-dimensional system is a final, crucial step.

The general idea is to maintain a real-time estimate of the surface which contains the point of interest (POI) by approximating it with a certain model and updating it at every time step. This approximated 3D surface at a time step k compares to the scalar $\hat{z}_k$ of chapter 2 and further. In this chapter a semispherical surface is used, yielding 4 degrees of freedom: the location of the centre and the radius. A sketch of the principle is shown in Figure 6.1.

Considering the absent availability of an accurate close-range 3D distance sensor, the measurement is presumed to be done by a laser mounted on the robot end effector. Consequently, the measurement remains one-dimensional. Figure 6.2 shows a geometric view of the situation where the end effector trajectory is in the $z = 0$ plane, perpendicular to the laser measurement direction.

6.2 3D Extended Kalman filter (EKF)

6.2.1 Simplified model

In a first illustrative approach, the laser is assumed to provide a unidirectional measurement along the z-axis and the heart motion is neglected (static scenario). The system states $\mathbf{x}$ are the 4 parameters that determine the surface: the coordinates of the centre and the radius R_s. Furthermore, the current location of the laser $(x_e, y_e, z_e)_k$ is presumed to be known[1]. The measurement equation Equation 6.3 is then analytically available which allows for the Jacobian to be calculated explicitly.

---

[1] The subscript 'e' is for '(end) effector'.

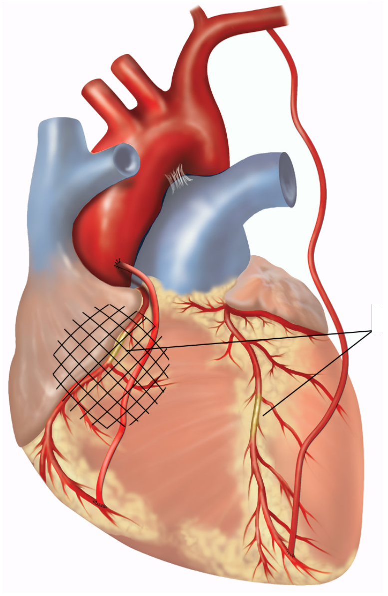


Figure 6.1: Illustration of POI on heart with spherical approximation of surface.

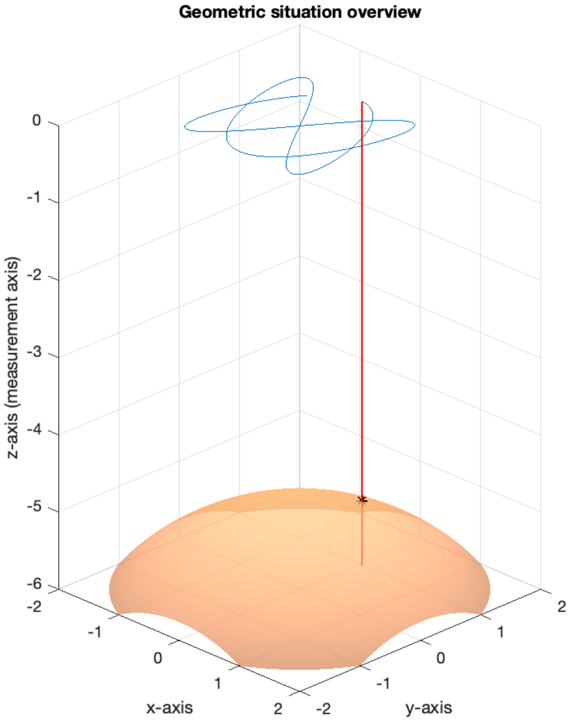


Figure 6.2: Geometric overview of estimated surface (copper) with end effector trajectory (blue) and laser measurement line (red).

$$\mathbf{x}_k = [x_c, y_c, z_c, R_s]_k^T \tag{6.1}$$

$$A = I_4 \tag{6.2}$$

$$z_k^m = z_{c,k} + z_{e,k} - \sqrt{R_s^2 - (x_{e,k} - x_{c,k})^2 - (y_{e,k} - y_{c,k})^2} \tag{6.3}$$

6.2.2 Orientational freedom

In correspondence to a realistic surgical robot, the laser (end effector) is granted orientational freedom. This orientation is captured through a polar angle $\theta_{e,k}$ and azimuth angle $\phi_{e,k}$ at every time step k. These angles correspond to the spherical coordinate system depicted in Figure 6.3. An updated geometric overview is given by Figure 6.4.

Maintaining the same surface model as subsection 6.2.1, the measurement equation now changes to Equation 6.4 where $(x, y, z)_{\text{sol}}$ is the solution of Equation 6.5 for (x, y, z) and represents the measured point on the surface. The first and second equations define the measurement line of the laser, the third the semispherical surface. Equation 6.5 only holds under the conditions of Equation 6.6. The consideration of the other cases is a topic of subsection 6.3.2.

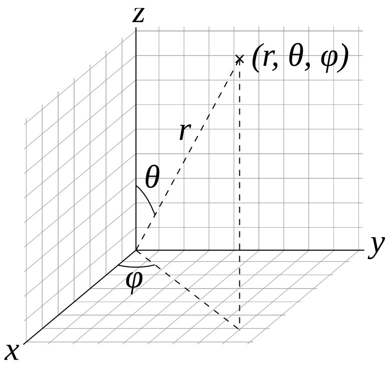


Figure 6.3: Spherical coordinate system with polar angle θ and azimuth angle ϕ.

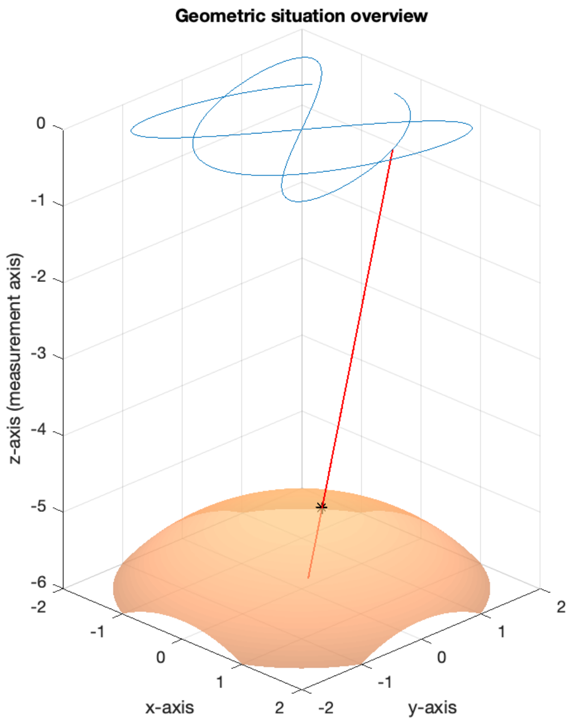


Figure 6.4: Geometric overview of estimated surface (copper) with end effector trajectory (blue) and inclined laser measurement line (red). $\theta_e = \pi + 0.2\,\text{rad}$, $\phi_e = 0.1\,\text{rad}$ and $(x_e, y_e, z_e) = (1.1, 0.46, 0)\,\text{cm}$.

$$z_k^m = \sqrt{(x_{e,k} - x_{\text{sol},k})^2 + (y_{e,k} - y_{\text{sol},k})^2 + (z_{e,k} - z_{\text{sol},k})^2} \tag{6.4}$$

$$\begin{cases} \dfrac{x - x_{e,k}}{\sin\theta_{e,k}\cos\phi_{e,k}} - \dfrac{y - y_{e,k}}{\sin\theta_{e,k}\sin\phi_{e,k}} = 0 \\ \dfrac{y - y_{e,k}}{\sin\theta_{e,k}\sin\phi_{e,k}} - \dfrac{z - z_{e,k}}{\cos\theta_{e,k}} = 0 \\ z - z_{c,k} - \sqrt{R_s^2 - (y - y_{c,k})^2 - (x - x_{c,k})^2} = 0 \end{cases} \tag{6.5}$$

$$\begin{cases} \sin\theta_{e,k}\cos\phi_{e,k} \neq 0 \\ \sin\theta_{e,k}\sin\phi_{e,k} \neq 0 \\ \cos\theta_{e,k} \neq 0 \end{cases} \tag{6.6}$$

No convenient analytical solution is attainable for these more profound equations. Additionally, the dependency of filter performance on the existence of an analytical formulation is to be omitted as interchangeability of the surface shape is a vital asset of the intended system. Depending on the location of the POI on the heart, a surface model change is due. This makes any advantage of the EKF over the UKF fade away. The latter provides a more prominent solution in coping with numerical models as it omits the need for numerical differentiation by means of the unscented transformation (UT). Therefore, in what follows, an unscented Kalman filter (UKF) is designed for the purpose of 3D heart motion estimation.

6.3 3D Unscented Kalman filter (UKF)

6.3.1 Model equations

A more complete model is maintained as heart beat and respiratory motion are added. For clarity, the results in this section only consider monofrequential movement for either. Results where the first harmonic of the heart beat motion is included are provided in Appendix A. Based on observations of MIDCAB footage, the heart movement is by the model presumed to manifest itself in a varying sphere radius R_s, and the respiratory movement is monodirectional in the z-direction. The system and measurement equations used in the filter are Equation 6.7 and Equation 6.8 respectively. The equations to be solved are given by Equation 6.9 where Equation 6.6 are required conditions.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ R_s \\ A \\ \omega \\ \theta \\ A_{\text{resp}} \\ \omega_{\text{resp}} \\ \theta_{\text{resp}} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ R_s \\ A \\ \omega \\ \theta \\ A_{\text{resp}} \\ \omega_{\text{resp}} \\ \theta_{\text{resp}} \end{bmatrix}_{k} \tag{6.7}$$

$$z_k^m = \sqrt{(x_{e,k} - x_{\text{sol},k})^2 + (y_{e,k} - y_{\text{sol},k})^2 + (z_{e,k} - z_{\text{sol},k})^2} \tag{6.8}$$

$$\begin{cases} \dfrac{x - x_{e,k}}{\sin\theta_{e,k}\cos\phi_{e,k}} - \dfrac{y - y_{e,k}}{\sin\theta_{e,k}\sin\phi_{e,k}} = 0 \\ \dfrac{y - y_{e,k}}{\sin\theta_{e,k}\sin\phi_{e,k}} - \dfrac{z - z_{e,k}}{\cos\theta_{e,k}} = 0 \\ z - (z_{c,k} + A_{\text{resp},k}\sin\theta_{\text{resp},k}) \;... \\ \qquad ... \; - \sqrt{(R_s + A_k\sin\theta_k)^2 - (y - y_{c,k})^2 - (x - x_{c,k})^2} = 0 \end{cases} \tag{6.9}$$

6.3.2 Numeric solver

The traditional measurement equation is replaced by a system of equations which has to be solved numerically in real time. A solver is deployed which solves Equation 6.9 for $(x, y, z)_{\text{sol},k} = F(\text{states}, x_{e,k}, y_{e,k}, z_{e,k}, \theta_{e,k}, \phi_{e,k})_k$ at every time step Δt. The conditions of Equation 6.6 are *not* required in order for a solution to exist. If any of the conditions is violated Equation 6.9 has to be reformulated.

For instance: if $\sin\theta_k = 0$, Equation 6.10 is used instead.

$$\begin{cases} x - x_{e,k} = 0 \\ y - y_{e,k} = 0 \\ z - (z_{c,k} + A_{\text{resp},k} \sin\theta_{\text{resp},k}) \ \dots \\ \qquad \dots \ - \sqrt{(R_s + A_k \sin\theta_k)^2 - (y - y_{c,k})^2 - (x - x_{c,k})^2} = 0 \end{cases} \tag{6.10}$$

To account for the limits of numeric calculations, in the condition test a small threshold value is used instead of zero. Four distinctive scenarios are possible in which this reformulation is necessary. This results in an *if…elseif…else*-case with four conditions and the nominal equations in the *else*-entry. The applied solver employs a trust-region dogleg algorithm [27]. At every time step $2n + 1$ sigma-points are propagated through the system. In this case $n = 10$ which translates to the solver being addressed 21 times within every time step.

Consequently, facilitating fast solver convergence is of vital importance for the filter's performance. In the local region around the measured point, and within nominal geometric circumstances[2], the problem is strongly convex. Convergence to a unique solution is therefore guaranteed if the starting point of the algorithm is sufficiently close to the solution. In the presence of a small time step (e.g. 10 ms) and considering limited robot dynamics, the subsequent measured points are close to one another. Therefore this problem is tackled by using the previous solution of the equations as the starting point of the current problem. As a result, the solver converges within a significantly lower amount of iterations. In the simulation of section 6.4 this results in an MSE difference[3] of 0.001% as compared to using the exact solution when only 2 iterations are allowed for the optimiser. This explains why altering the solver's tolerance to a larger value -albeit a fraction of the standard deviation of the sensor noise- further reduces the amount of iterations. Furthermore, these $2n + 1$ optimisation problems are independent and therefore suited for parallelisation. Parallel computing is a type of computation in which many calculations or the execution of processes are carried out simultaneously [28]. Profiling the code learns that for the simulation of section 6.4, 80% of the computation time is spent on solving one of these optimisation problems. As a result, the designed algorithm is about 80% parallelisable. This means that the algorithm is 5 times faster in the theoretic case where an unlimited amount of CPU cores is available.

---

[2] If e.g. the measurement line cuts the surface at a small angle, two nearby solutions may exist or the problem might not be convex in the considered region.

[3] On the tracking error of the UKF.

6.3.3 Data elimination

Inconsistent data

In order to increase the filter's stability, inconsistent data is not fed to the UKF. The consistency test discussed in section 2.2.6 is shown here again and used for the algorithm.

$$\epsilon_k = \nu_k' tr(P_k)^{-1} \nu_k$$

Because no innovation covariance is available and this modified normalized innovation squared (NIS) is calculated making use of the error covariance matrix' trace, an arbitrary threshold value ϵ_{thres} must be experimentally determined.

Out of bounds data

The surface tracked by the filter is meant to be a good approximation within a certain region around the POI. I.e. when deviating more towards the borders of the semisphere, the error between the surface and heart becomes larger. In order to prevent measurements outside of this area of interest to bias the parameters of the surface, only data close enough to the POI is considered. Figure 6.5 shows an example in correspondence to Figure 6.2. The black stripe-dot line illustrates a proposed threshold for measurements to be considered. Even though this threshold area is depicted in 2D, in reality it is a 3D 'threshold volume', e.g. in the shape of a sphere. When at a certain time step the algorithm eliminates a data point for usage, the correction step is omitted and two consecutive prediction steps are performed.

6.4 Simulation

6.4.1 Lissajous tracking trajectory

The trajectory followed by the robot end effector plays an important role in constructing the model. From one hand, ideally, the samples measured are as divers as possible, distributed along the area of interest. From the other, the robot imposes limitations as its dynamics are limited. A raster with sharp edges cannot be tracked accurately by the robot as it requires significant bandwidth. The sinusoidal scanning trajectory of Figure 6.6 (left) resolves that problem but also moves slowly in one direction. A large number of consecutive samples come from within a narrow area, so it fails to provide a fast overview of the situation.

An elegant solution is provided by Lissajous scanning trajectories. The parametrised equations take the form of Equation 6.11.

$$\begin{aligned} x(t) &= A \sin at \\ y(t) &= B \sin bt \end{aligned} \tag{6.11}$$

If the ratio $\frac{a}{b}$ is rational, the produced curves are closed. This is not a strict requirement for the purposes of this work. Figure 6.6 (middle and right) shows results for $\frac{2}{3}$ and $\frac{4}{5}$ respectively. The $\frac{2}{3}$-profile has so far been illustratively used in

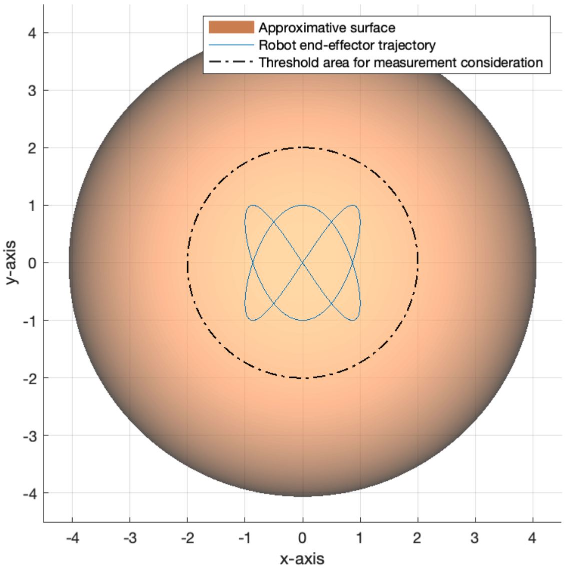


Figure 6.5: Zenith view of semisphere and trajectory corresponding to Figure 6.2 with example of area for data acceptance (stripe-dot line).

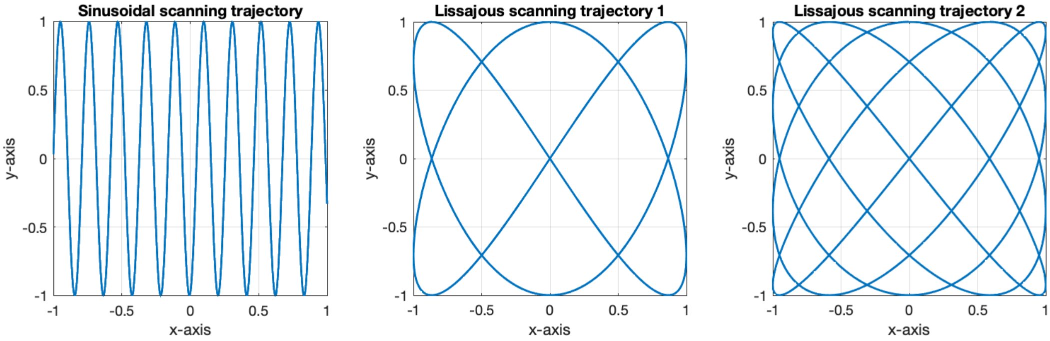


Figure 6.6: 3 realistic (limited BW) tracking profiles. One sinusoidal scanning trajectory (left) and two Lissajous profiles with different frequency ratios (middle and right, 2:3 and 4:5 respectively).

Figure 6.2, Figure 6.4 and Figure 6.5. These trajectories are especially beneficial as they cover large area in little time and additionally require only a or b -whichever is highest- as maximum bandwidth (BW). The dimension of the covered raster is $2A$ by $2B$. Because of the aforementioned reasons, Lissajous tracking has been proven beneficial in various mechanical scanning applications [29, 30].

6.4.2 Parameters

The UKF time step is $\Delta t = 10\,\mathrm{ms}$. The heart is assumed to locally approximate a sphere of radius $R_s = 4\,\mathrm{cm}$ and centre $(0, 0, -8)\,\mathrm{cm}$. Heart beat motion is incorporated by a sinusoidally varying radius with $\omega = 2\pi\,{}^{\mathrm{rad}}\!/_{\mathrm{s}}$ (60 BPM) and amplitude $A = 1\,\mathrm{cm}$. I.e. the radius varies between 3 and 5 cm. Respiratory movement is translational in the z-direction with $\omega_{\mathrm{resp}} = 12/60 * 2\pi\,{}^{\mathrm{rad}}\!/_{\mathrm{s}}$ (12 RR) and amplitude $A_{\mathrm{resp}} = 2\,\mathrm{cm}$, resulting in the z_c coordinate varying between -10 and -6 cm.

The laser position and orientation are determined by three coordinates and two angles, given in cm and rad:

$$x_{e,k} = \sin 4k\Delta t \tag{6.12}$$

$$y_{e,k} = \cos 6k\Delta t \tag{6.13}$$

$$z_{e,k} = 0 \tag{6.14}$$

$$\theta_{e,k} = \pi(1 + \frac{\sin k\Delta t}{18}) \tag{6.15}$$

$$\phi_{e,k} = 5k\Delta t \tag{6.16}$$

This environment is simulated, the measurement is calculated and Gaussian noise with zero mean and variance $R_{\mathrm{meas}} = 10^{-2}\,\mathrm{cm}^2$ is added.

The filter is set and initialised with $R_0 = 10^{-2}\,\mathrm{cm}^2$, $Q_0 = I_{10} * 10^{-7}$ and $P_0 = I_{10} * 10^{-1}$. The initial states $\mathbf{x}_0$ deviate from the real ones $\mathbf{x}_{0,\mathrm{actual}}$.

$$\mathbf{x}_{0,\mathrm{actual}} = [0, 0, -8, 4, 1, 2\pi, 0, 2, 1.26, 0]^T \tag{6.17}$$

$$\mathbf{x}_0 = [0.5, -0.5, -7, 3, 2, 2\pi + 1, 0, 1, 0.76, 0]^T \tag{6.18}$$

6.4.3 Results

Figure 6.7 shows a plot of the measurement without noise, the measurement and the UKF output. The norm of the error covariance matrix P is plotted and indicates convergence. The state estimates in function of time are shown in Figure 6.8 together with the actual values. All states show sensible convergence. z_c and R_s show slight deviations. They compensate each other locally. Considering the first control strategy of chapter 5, an important criterion for assessment of accuracy is the estimated velocity as compared to the actual velocity of a tracked point. The estimated velocity of the center point of the semisphere in this simulation is compared with its actual velocity in Figure B.5 (spherical surface line).

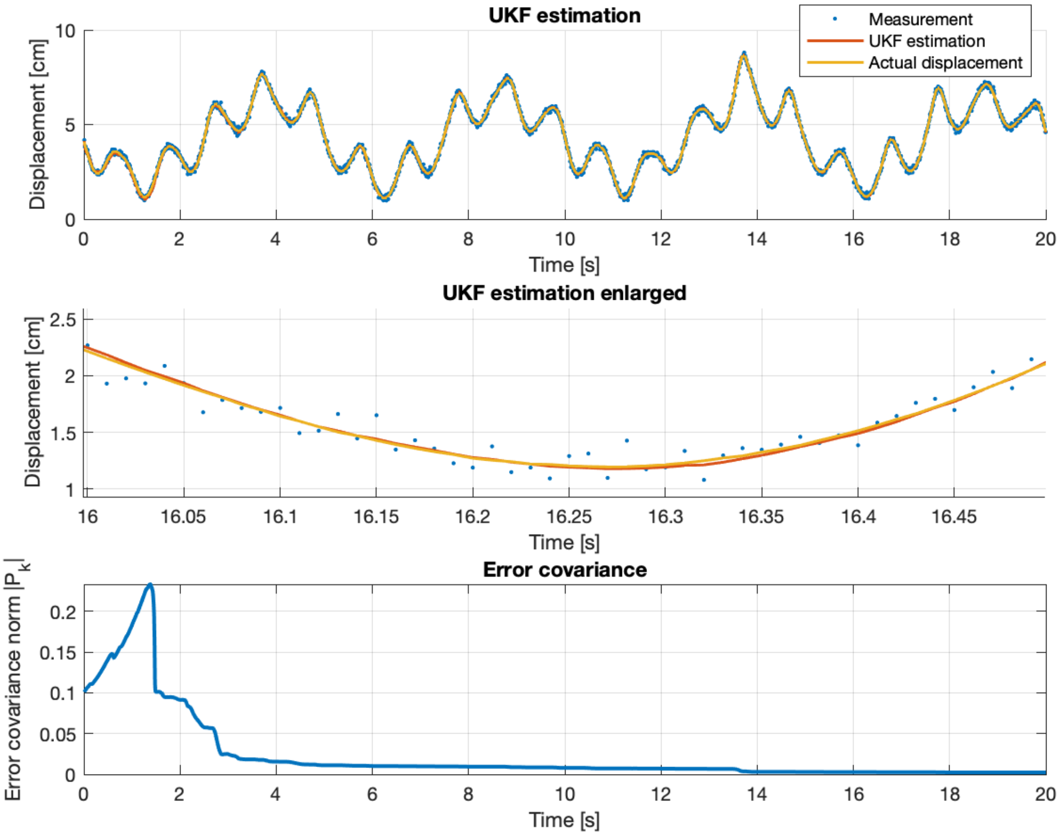


Figure 6.7: 3D UKF output estimation, together with measurement and signal without noise (top). The norm of the error covariance matrix (bottom) is also plotted.

A similar simulation where the model is extended with a harmonic for the heart beat motion is found in Appendix A. Appendix D contains a geometric overview of this simulation at three time steps.

6.4.4 Unmodelled behaviour

In order to illustrate the purpose of the NIS, the measurement signal is altered and an anomaly is created at around $t = 7\,\mathrm{s}$, in the region where the filter's convergence has significantly advanced. Q_k is increased by a factor 10. The results for the simulation without data elimination are shown in Figure 6.9. As the data makes suspect, the estimated states deviate significantly as is shown in Figure C.1 (Appendix C).

Figure 6.10 shows the same simulation where data elimination is applied based on the NIS criterion. ϵ_{thres} is fixed at 15. The estimated signal from the anomaly onward tracks the reality accurately and the state estimates don't deviate as shown by Figure C.2 (Appendix C). As the surface of the heart in not smooth but instead harbors irregularities such as epicardial fat and arteries, this criterion is proposed as

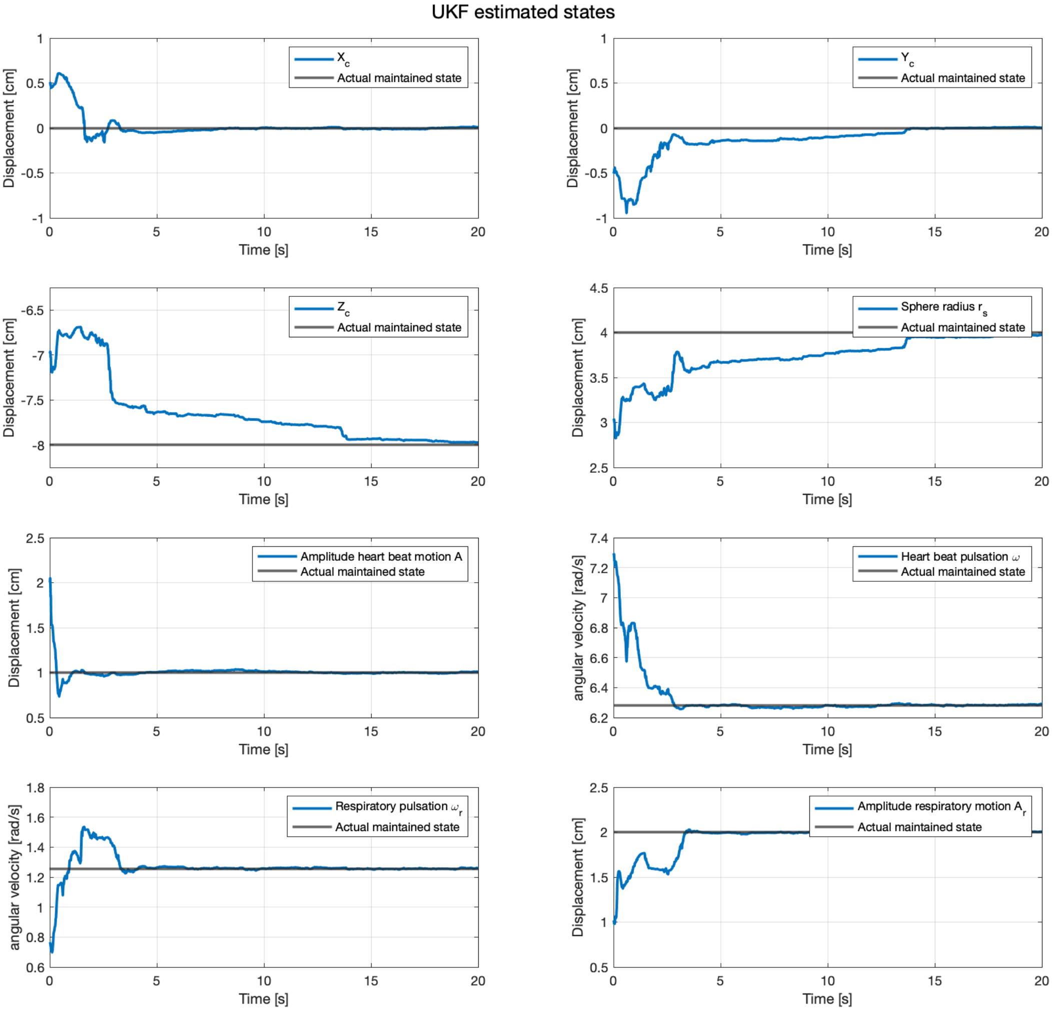


Figure 6.8: 3D UKF estimated states.

a means to filter them out and improve the accuracy of the tracking surface in the area of interest of the heart.

Furthermore, two simulations are performed in which an ellipsoidal surface is tracked and approximated with the so far maintained spherical profile. The second simulation uses a smaller scanning area and illustrates the importance of limiting this area when it comes to rejecting approximation errors. The velocity of a point on the surface, an important value to be estimated for the control (chapter 5), is also compared with that of the corresponding point in case of a spherical surface, and plotted against the actual velocity in both cases. This simulation and its results are presented in Appendix B.

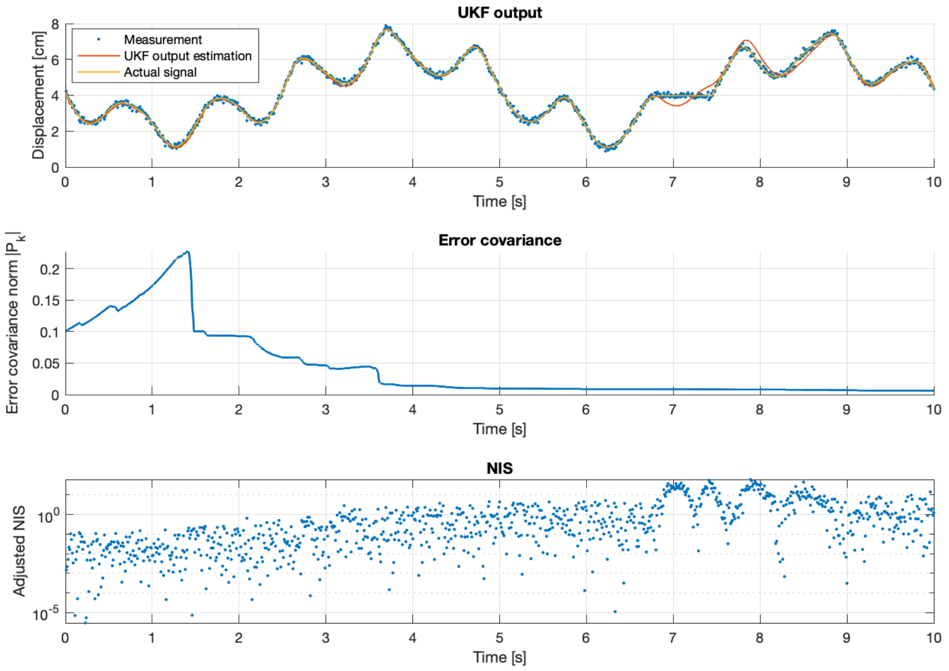


Figure 6.9: Simulated results of 3D UKF with trumped signal without data elimination. The corresponding system states are given by Figure C.1.

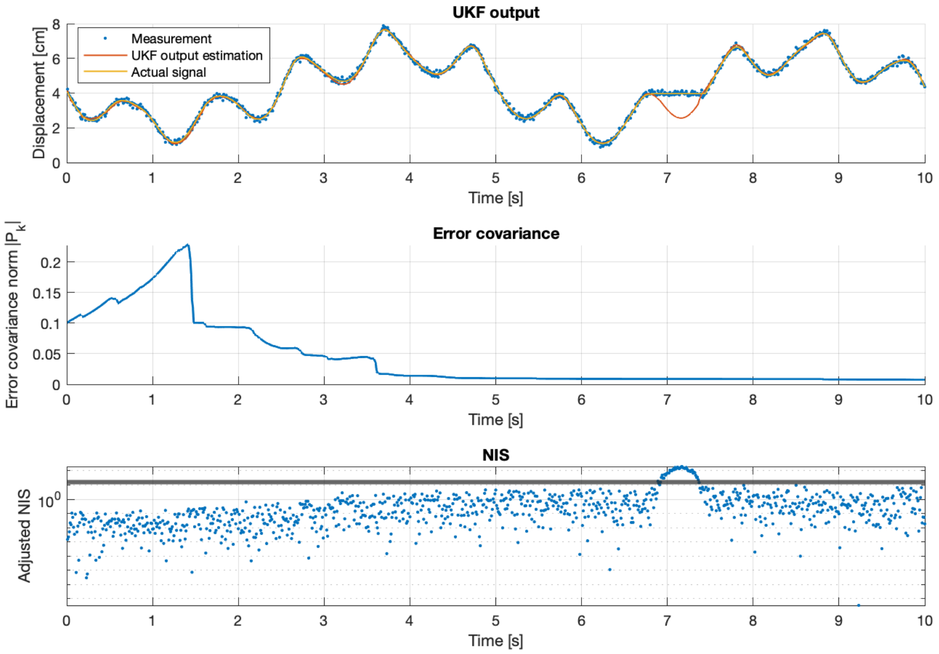


Figure 6.10: Simulated results of 3D UKF with trumped signal with data elimination. The horizontal line in the bottom plot shows the elimination criterion. The corresponding system states are given by Figure C.2.

6.5 Conclusion

The demand of geometrical degrees of freedom renders an EKF unsuitable for 3D movement. That path is abandoned in favour of a UKF. Three-dimensional system equations are designed along with a measurement equation incorporating a numeric solver. Two methods of measurement data elimination are elaborated and a Lissajous tracking profile is proposed. The entire system is evaluated through simulations and favourable results are attained.

Chapter 7

Conclusion

Heart procedures such as the coronary artery bypass grafting (CABG) show potential for improvement when confronted with a powerful motion-compensation system. Improvements are noticeable on the side of the surgeon because of the ability to use the robot for the entire procedure, e.g. more precise control, extended periods of concentration and less tiredness, as well as on the patient's end with shorter revalidation periods, a decrease in complication likeliness and the possibility for earlier declined patients to be reconsidered.

Research of the heart learns that the ECG signal contains predictive information of the movement. It additionally shows that the lion's share of heart beat movement is contained in only two frequencies. Three when the respiratory movement is accounted for.

From the comparison of the extended Kalman filter (EKF), unscented Kalman filter (UKF) and moving horizon estimator (MHE) for nonlinear periodic one-dimensional motion estimation, the EKF proves to be most prominent solution in the basic scenario. It is the most accurate filter, is least sensitive to changes in tuning parameters and, on account of the availability of an analytical Jacobian, significantly less time consuming when compared to the others. On-line detection of anomalies and inconsistent data is enabled by the NIS and SNIS consistency tests. The ECG signal is used to add predictive aspects to the EKF. When a sensor fusion is attempted, the EKF is replaced by a UKF. Simulations prove the improvements brought about by these alterations.

A linear Kalman filter (LKF) brings relief in the search for an accurate on-line differentiation method. It proves to be a superior all-round solution as compared to the more traditional method: numerically differentiating a low-passed signal. Its main benefit is a reduced time delay.

A combination of feedback and feedforward control embodies the more traditional approach in the comparison of two control strategies. The other being model predictive control (MPC). When assisted by the aforementioned state estimators, both demonstrate adequate performance. The MPC manages to improve tracking to an extent which cannot be overseen. This improvement however comes at a significant computational cost. A fully integrated simulation in which the behaviour of the input signal temporarily does not match the internal model of the EKF/UKF and MPC, reveals flaws of both systems. An on-line consistency test is proposed along with a solution in case of anomaly detection.

In the context of the envisioning of a real-world motion compensation system, the one-dimensional EKF/UKF is expanded in order to create a three-dimensional motion estimator. The maintained model is that of a semispherical surface which locally approximates the heart surface. The unavoidable need for geometrical degrees of freedom, together with the demand of an interchangeable approximative surface render the EKF unsuitable. In response, an unscented Kalman filter (UKF) is designed which successfully carries out the task. Further optimisations to the filter algorithm, a proposition for a mechanical scanning trajectory and an algorithmic criterion for data elimination result in a suitable system. A statement which is supported by a series of simulations.

The design process of this work opens doors towards and creates room for further research. Future steps towards a fully operational motion-compensation system start with the assimilation between these disquisitions and a specific robot platform. This allows for three-dimensional expansion of the control system. Other improvements lie in a dynamic model between the ECG signal and heart displacement. Eventually, these efforts lead to an experimental validation of the system, concomitantly allowing for additional tuning of the countless parameters, thresholds and other variables which influence system performance.

Appendices

Appendix A

3D UKF simulation with first harmonic

An extra harmonic of the heart beat movement is added. Two extra states, a sine-argument and the sine-amplitude, represent it in the system. The simulation is executed with the following settings/parameters:

$$x_{e,k} = \sin 4k\Delta t \tag{A.1}$$
$$y_{e,k} = \cos 6k\Delta t \tag{A.2}$$
$$z_{e,k} = 0 \tag{A.3}$$
$$\theta_{e,k} = 0 \tag{A.4}$$
$$\phi_{e,k} = 0 \tag{A.5}$$

$$Q_{\text{filter}} = I_{12} * 10^{-6} \tag{A.6}$$
$$Q_0 = I_{12} \tag{A.7}$$
$$R_{\text{actual}} = 10^{-2} \tag{A.8}$$
$$R_{\text{filter}} = 10^{-2} \tag{A.9}$$
$$\mathbf{x} = [x_c, y_c, z_c, R_s, A_1, A_1, \omega, \theta_1, \theta_2, A_{\text{resp}}, \omega_{\text{resp}}, \theta_{\text{resp}}]^T \tag{A.10}$$
$$\mathbf{x}_{0,\text{actual}} = [0, 0, -8, 4, 1, 0.7, 2\pi, 0, 0, 2, 1.26, 0]^T \tag{A.11}$$
$$\mathbf{x}_0 = [0.5, -0.5, -7, 3, 1.2, 2, 2\pi + 1, 0, 0, 1, 0.76, 0]^T \tag{A.12}$$

The states are plotted in Figure A.1, sufficient convergence is present. The output is given by Figure A.2.

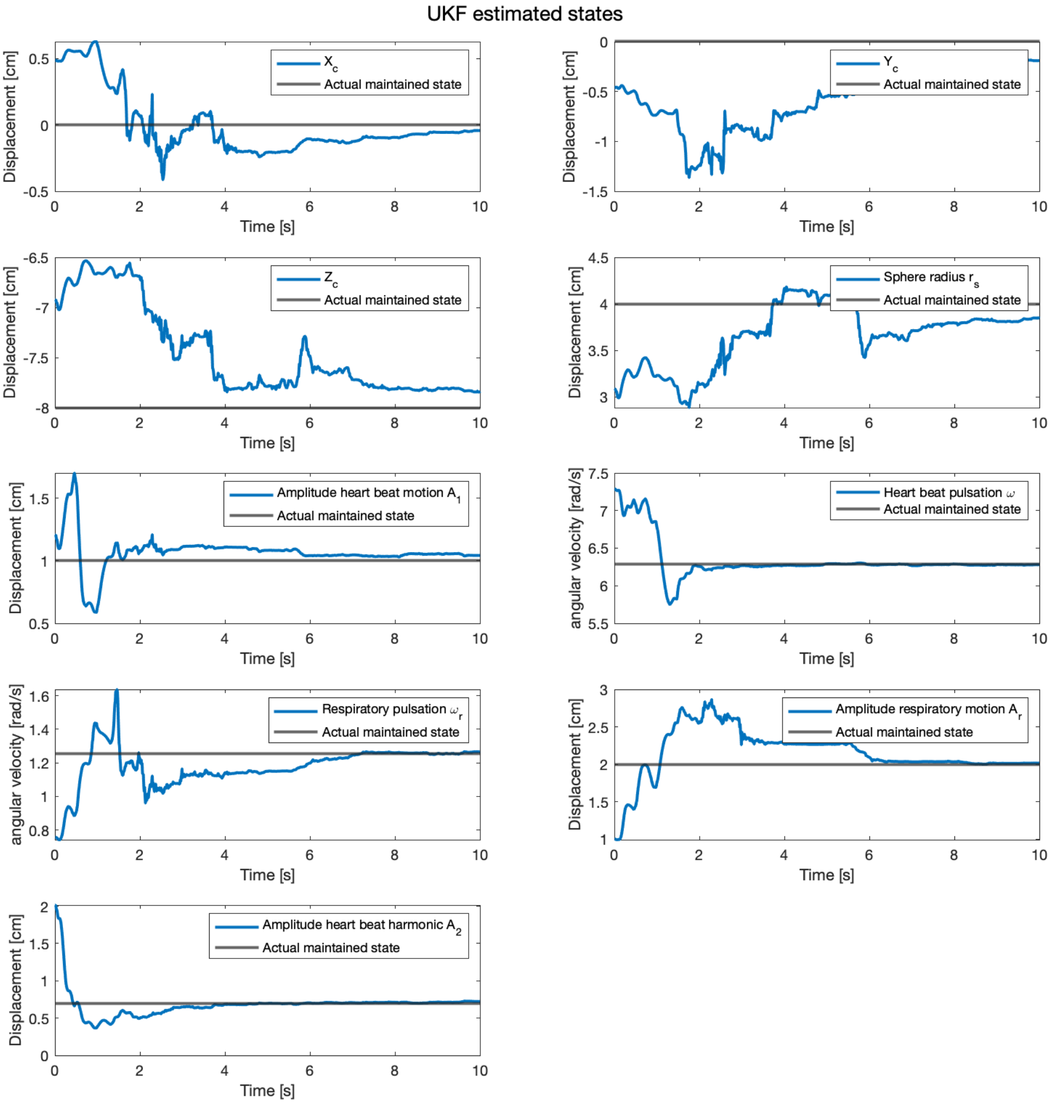


Figure A.1: States corresponding to simulation with heart beat harmonic.

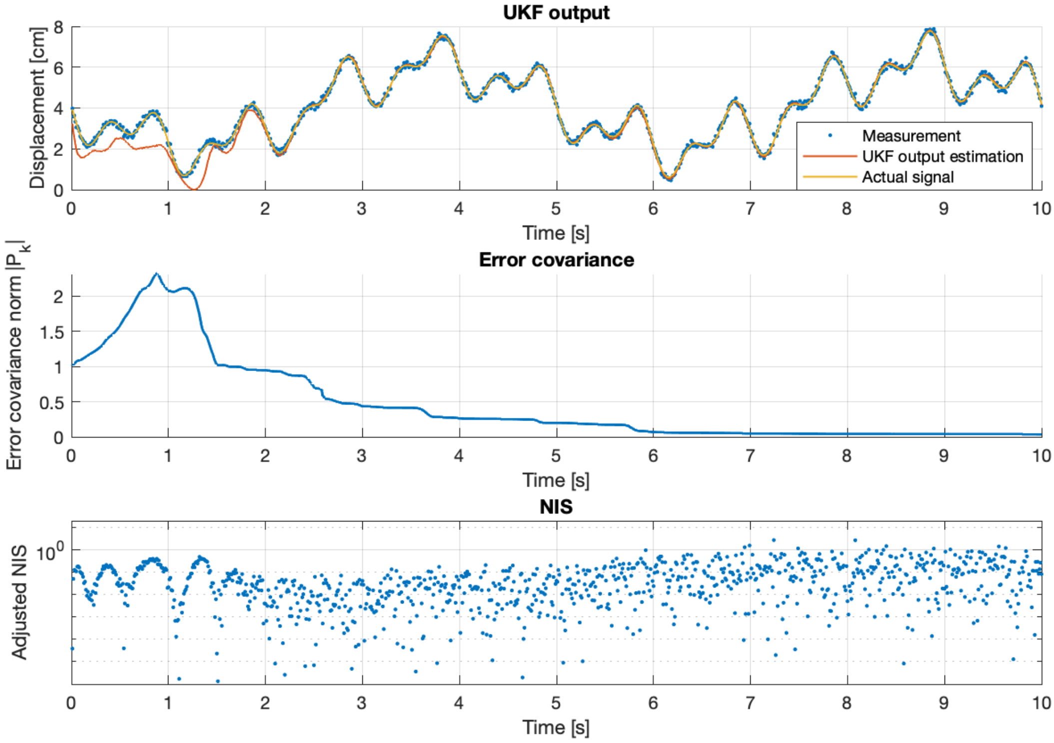


Figure A.2: In and output, error covariance and NIS corresponding to simulation with heart beat harmonic.

Appendix B

3D UKF simulation of ellipsoidal surface

In the context of investigating the robustness of the 3D UKF of chapter 6, the algorithm is set to approximate an ellipsoidal surface with a semispherical model.

The simulation variables and filter settings are similar to those of Appendix A and shown hereunder.

$$\begin{aligned}
x_{e,k} &= \sin 4k\Delta t && \text{(B.1)}\\
y_{e,k} &= \cos 6k\Delta t && \text{(B.2)}\\
z_{e,k} &= 0 && \text{(B.3)}\\
\theta_{e,k} &= \pi(1+\frac{\sin k\Delta t}{18}) && \text{(B.4)}\\
\phi_{e,k} &= 5k\Delta t && \text{(B.5)}
\end{aligned}$$

$$\begin{aligned}
Q_k &= I_{12} * 10^{-6} && \text{(B.6)}\\
P_0 &= I_{12} * 10^{-1} && \text{(B.7)}\\
R_{\text{actual}} &= 10^{-2} && \text{(B.8)}\\
R_k &= 10^{-2} && \text{(B.9)}\\
\mathbf{x} &= [x_c, y_c, z_c, R_s, A_1, A_1, \omega, \theta_1, \theta_2, A_{\text{resp}}, \omega_{\text{resp}}, \theta_{\text{resp}}]^T && \text{(B.10)}\\
\mathbf{x}_{0,\text{actual}} &= [0, 0, -8, 4, 1, 0.7, 2\pi, 0, 0, 2, 1.26, 0]^T && \text{(B.11)}\\
\mathbf{x}_0 &= [0.5, -0.5, -7, 3, 1.2, 2, 2\pi+1, 0, 0, 1, 0.76, 0]^T && \text{(B.12)}
\end{aligned}$$

Attention has to be paid to $\mathbf{x}_{0,\text{actual}}$, as it implies the actual model to be a (semi)sphere. Instead, an ellipsoid is used with its y-axis 25% longer than the x-axis, so 10 cm as opposed to 8 cm. The estimated output and states are shown by Figure B.1 and Figure B.2 respectively. The filter interprets the lower curvature of the surface in the y-direction by overestimating the radius of the sphere and underestimating the z-coordinate. In this case the UKF shows the propensity to converge slower.

Mathematically speaking, considering a smaller area of interest results in a decreased absolute error that results from a certain difference in surface curvature. It is therefore recommended to put effort into minimizing the considered heart surface. The effect of this adjustment is illustrating by an identical simulation except for $(x_e, y_e, z_e)_k$, which is scaled down as follows.

$$x_{e,k} = 0.5 \sin 4k\Delta t \tag{B.13}$$

$$y_{e,k} = 0.5 \cos 6k\Delta t \tag{B.14}$$

$$z_{e,k} = 0 \tag{B.15}$$

The corresponding output and states are given by Figure B.3 and Figure B.4 respectively. The improvement is mainly observed through the mean square error (MSE) of the tracking error after convergence. In the last 5 s of the first simulation it equals $1.8 * 10^{-3}\,\text{cm}^2$ as compared to $7.8 * 10^{-4}\,\text{cm}^2$ for the second one.

As has been illustrated before, the accuracy of the derivative of the estimated displacement is an important indicator of estimator performance, because a feedforward control strategy heavily depends on it. The actual velocity of the center point of the ellipsoid is the same as the one in the simulation of subsection 6.4.3. The estimated velocities of that simulation and this one with the ellipsoidal surface are compared with the actual velocity in Figure B.5. As is to be expected, the velocity estimate deviates more when the surface fit is less accurate.

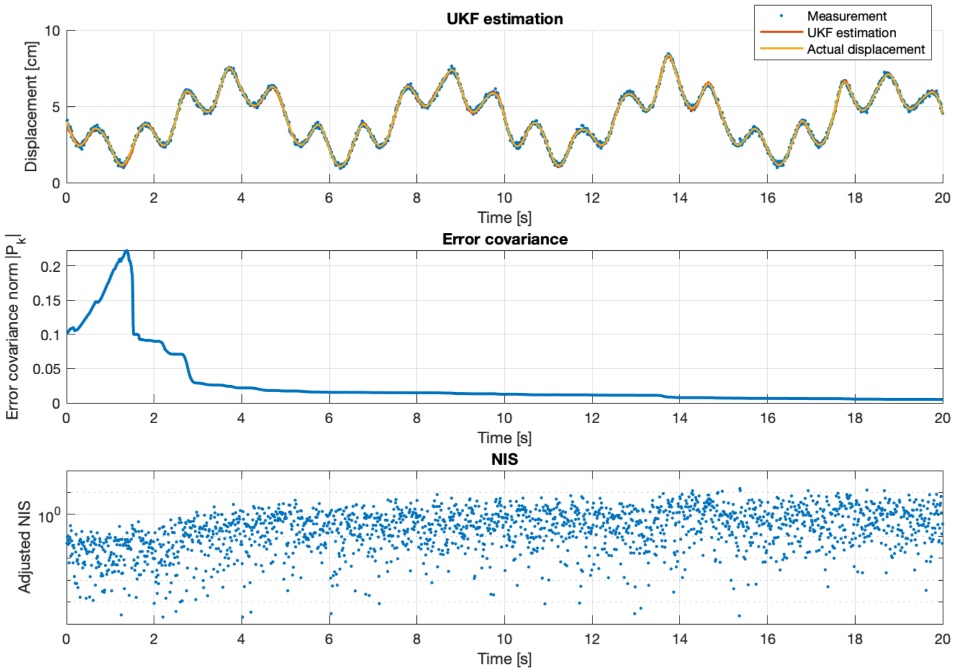


Figure B.1: UKF estimated tracking of deviating surface (ellipsoid).

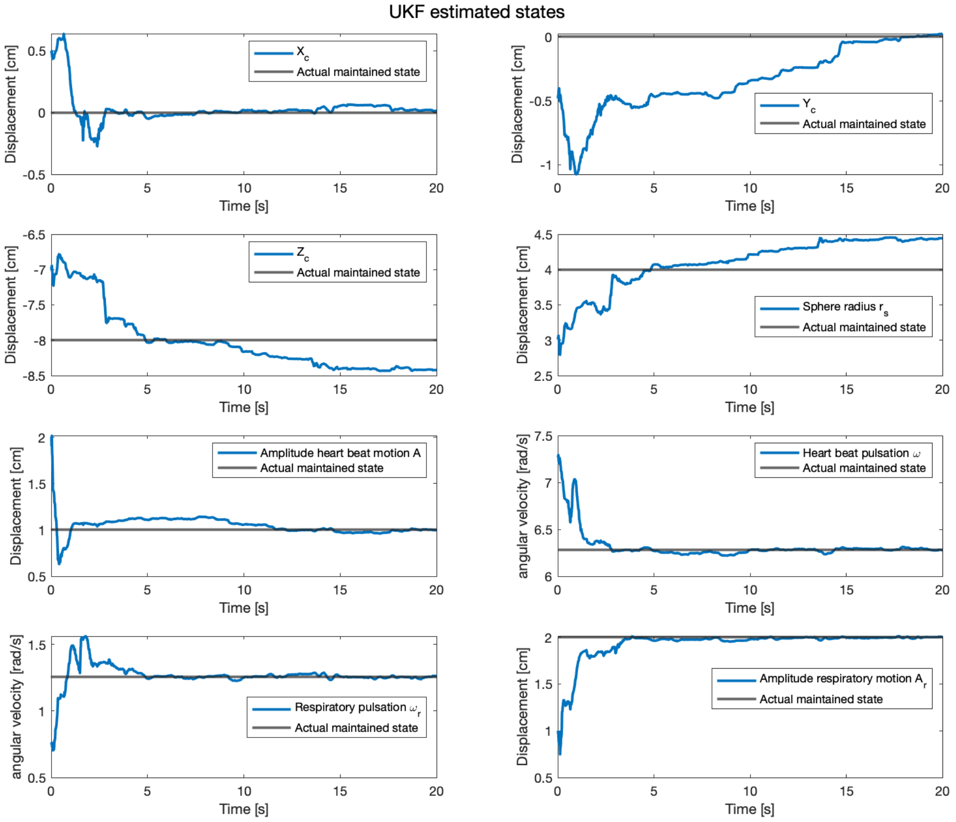


Figure B.2: UKF estimated states of deviating surface (ellipsoid).

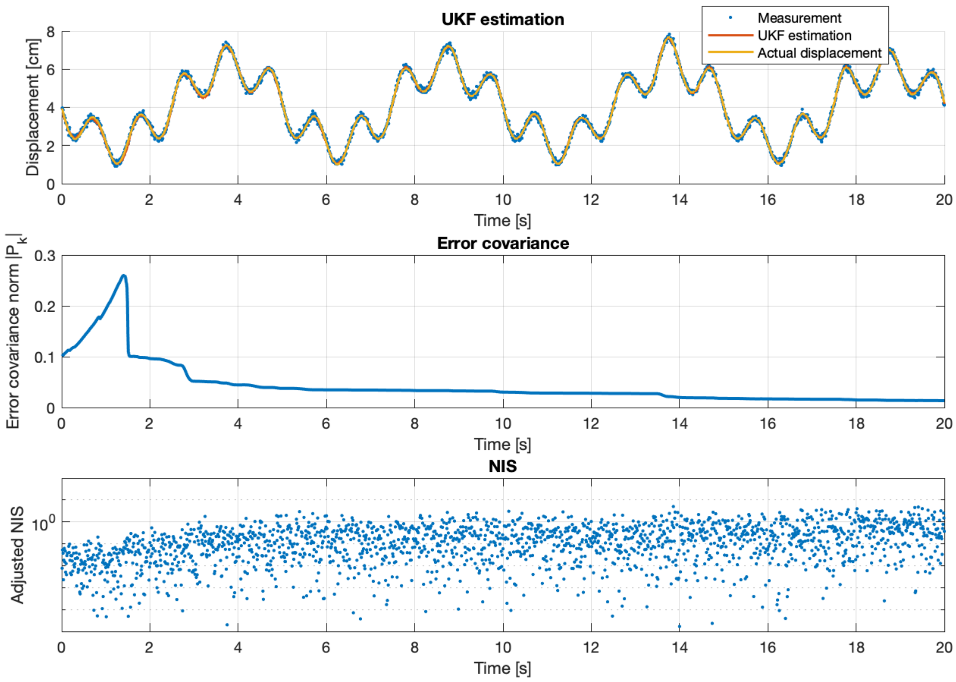


Figure B.3: UKF estimated tracking of deviating surface (ellipsoid) with smaller trajectory.

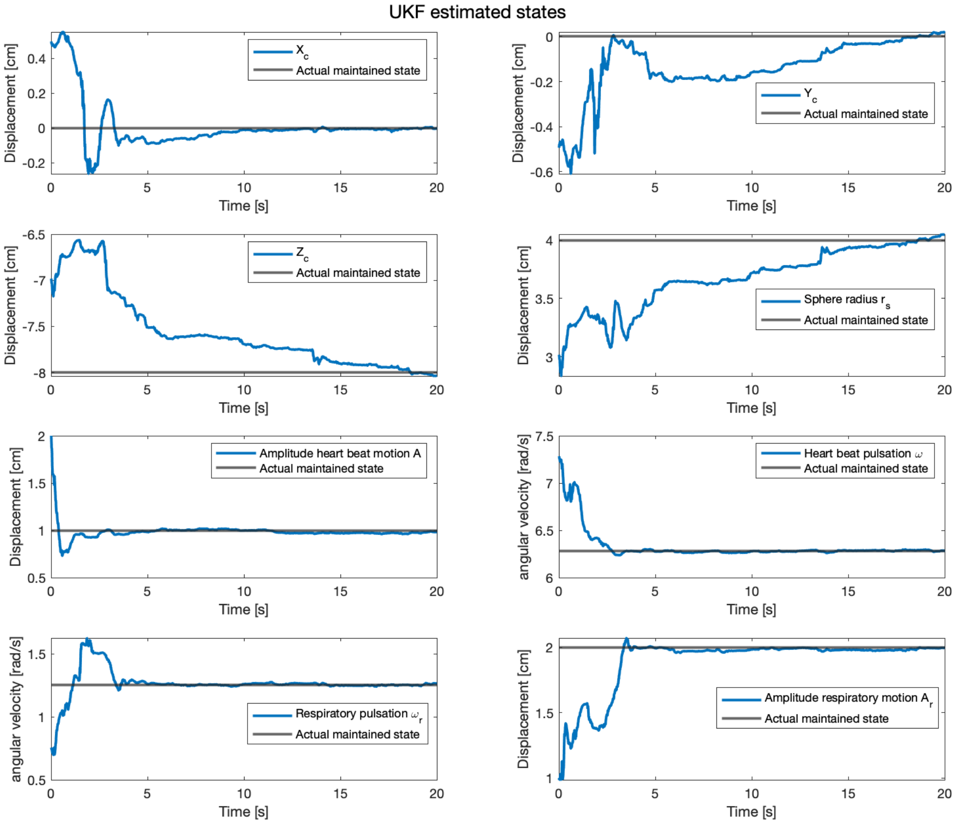


Figure B.4: UKF estimated states of deviating surface (ellipsoid) with smaller trajectory.

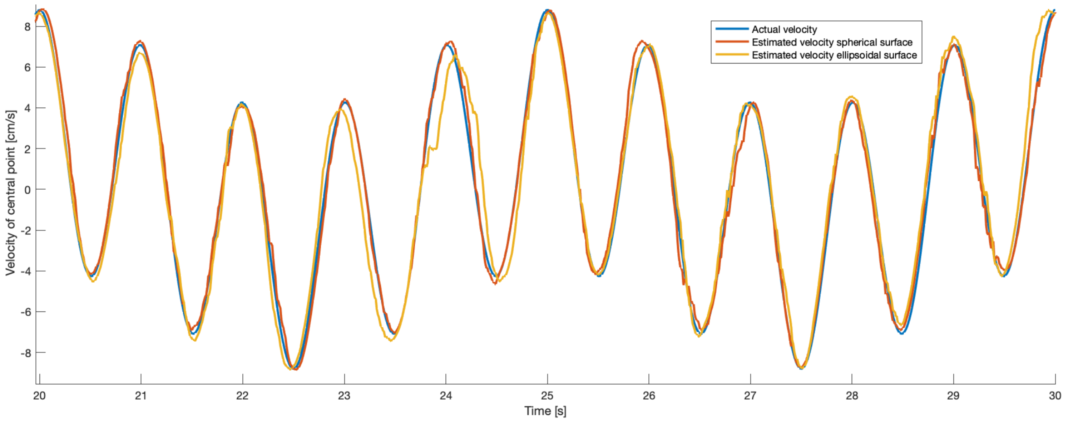


Figure B.5: Comparison of estimated velocities from simulation with ellipsoidal surface and the simulation of subsection 6.4.3 (with spherical surface) with the actual velocity. The velocity of the central (upper) point of the surfaces are considered. The estimation with the spherical surface provides a more accurate velocity.

Appendix C

Simulation with out-of-model signal

This appendix contains the states corresponding to the two compared simulations of subsection 6.4.4.

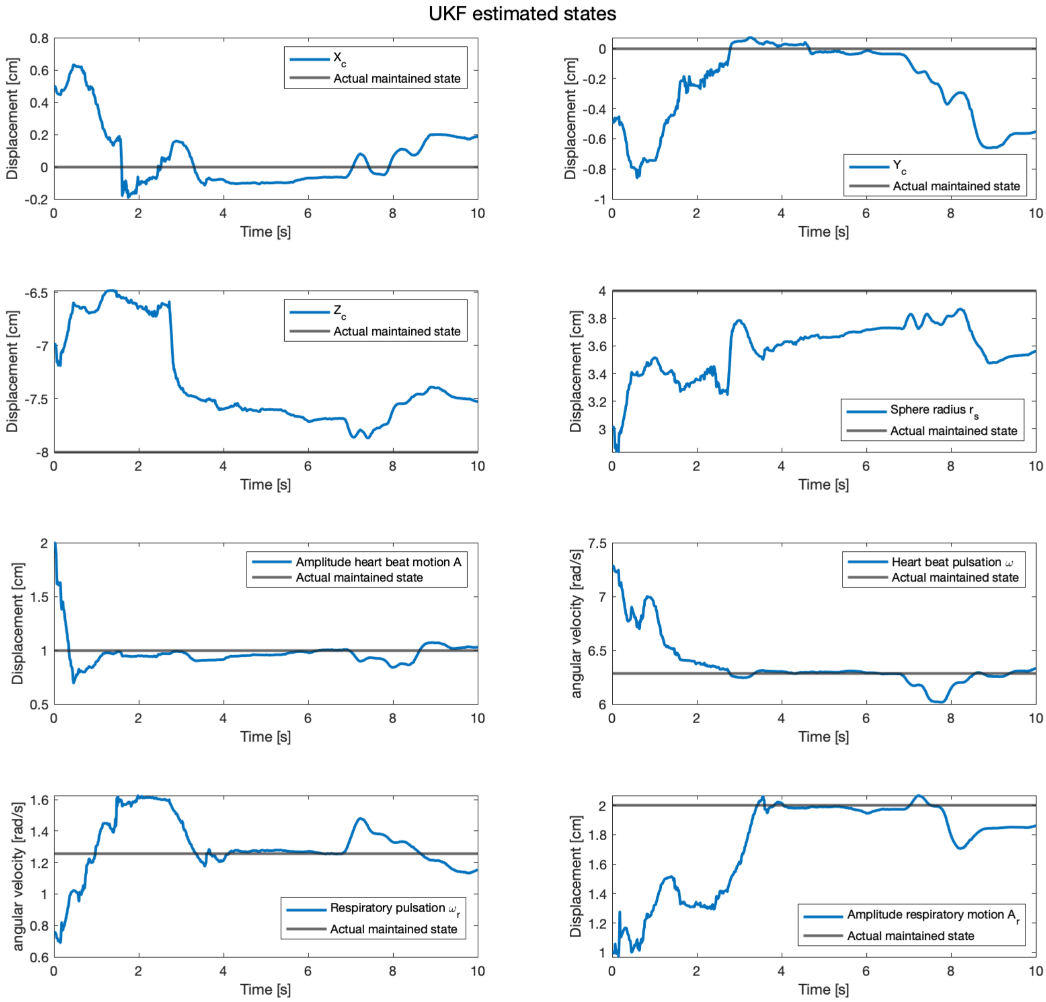


Figure C.1: UKF states for signal containing anomaly –without NIS test

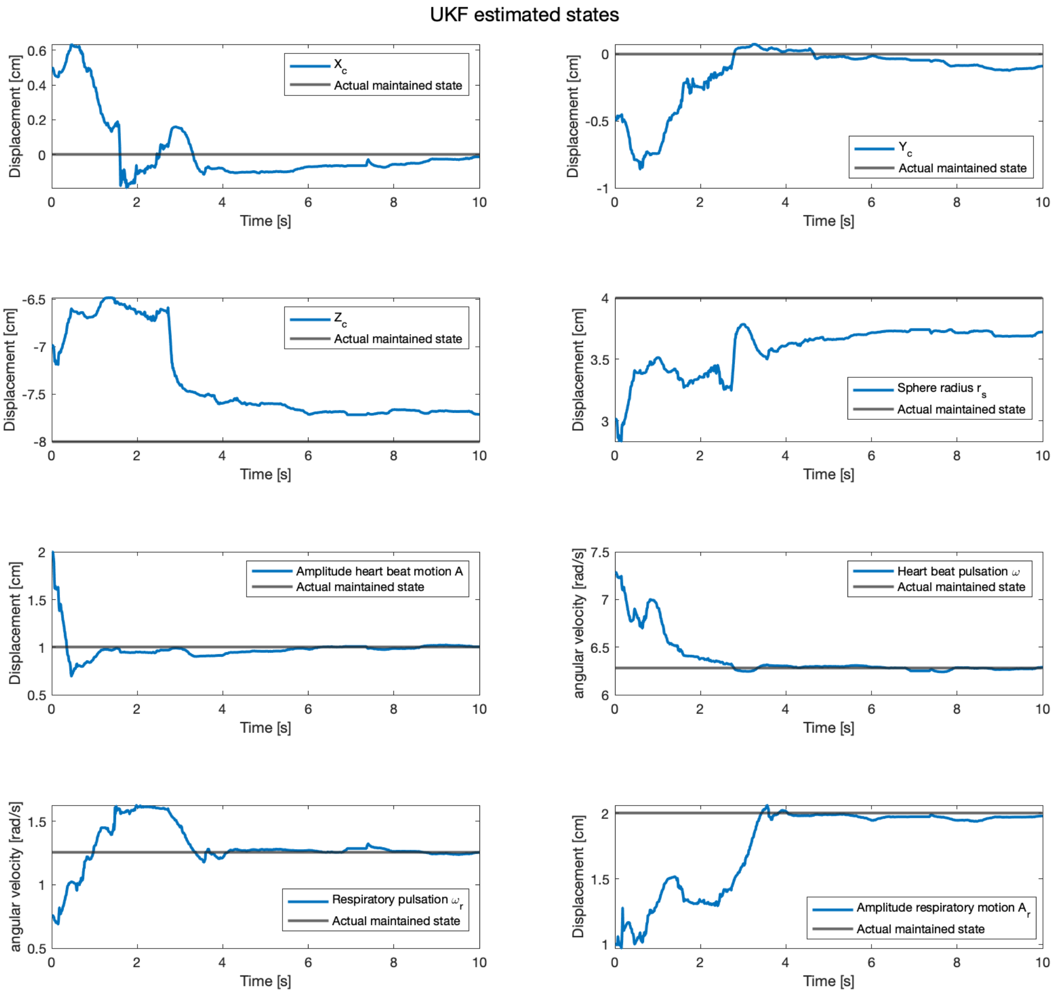


Figure C.2: UKF states for signal containing anomaly –with NIS test

Appendix D

Illustration of 3D UKF simulation

Figure D.1, Figure D.2, Figure D.3 and Figure D.4 illustrate the simulation at times 0 s, 2 s and 7 s (of a 10 s simulation). The last plot is a top view after the simulation has completed. Every previous measured point (black asterisk) remains plotted. The simulation corresponds to the one of section 6.3.

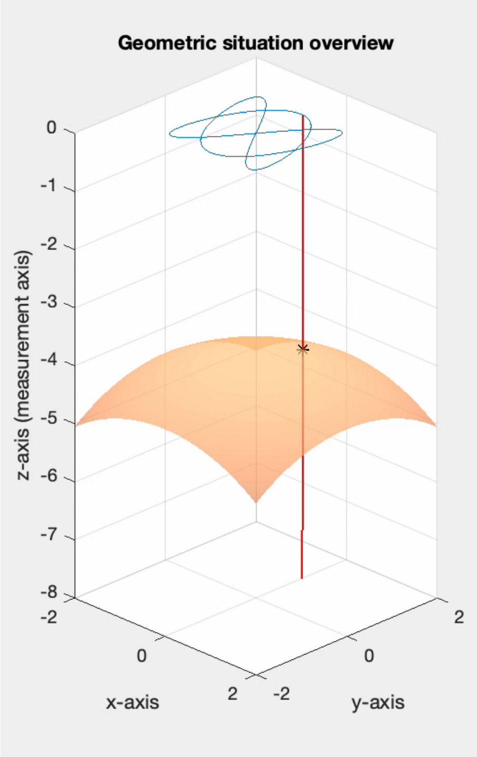


Figure D.1: At start

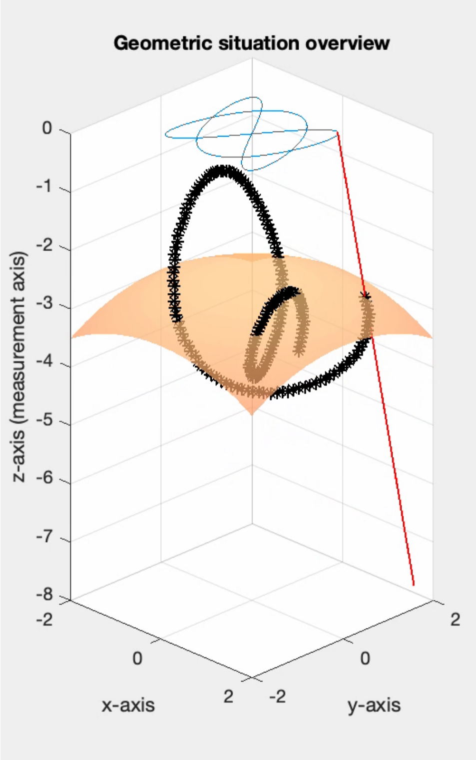


Figure D.2: After approx. 2 s.

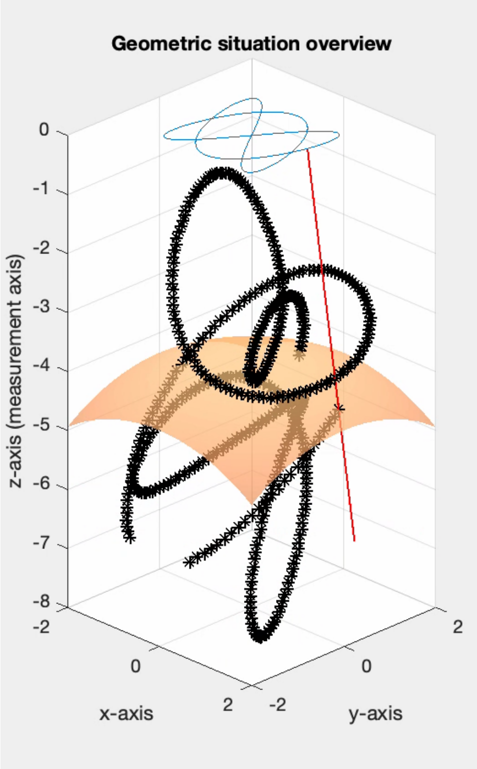


Figure D.3: After approx. 5 s.

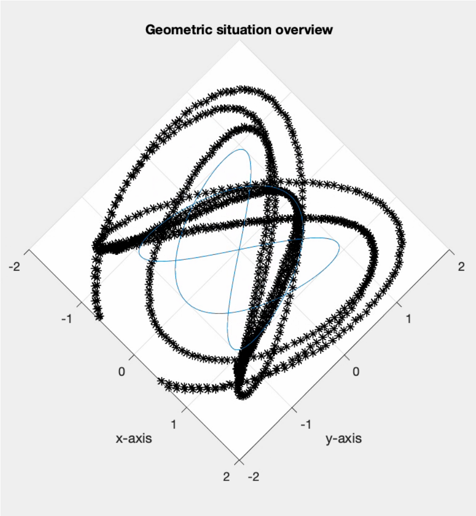


Figure D.4: After ending, top view.

Bibliography

[1] Cancer Care of Western New York, "Coronary artery bypass grafting." URL: https://www.cancercarewny.com/content.aspx?chunkiid=14782, last checked on 20-02-2020.

[2] The Society of Thoracic Surgeons, "Coronary artery disease." URL: https://ctsurgerypatients.org/adult-heart-disease/coronary-artery-disease, last checked on 19-02-2020.

[3] W. Gong and J. Cai, "Robot-assisted coronary artery bypass grafting improves short- term outcomes compared with minimally invasive direct coronary artery bypass grafting," *Journal of Thoracic Disease*, pp. 459–468, 2016.

[4] M. Diodato and E. G. Chedrawy, "Coronary artery bypass surgery: the past, present and future of myocardial revascularisation," 2014. Article ID: 726158.

[5] A. Weiss and A. Elixhauser, "Trends in operating room procedures in u.s. hospitals, 2001-2011," *Healthcare Cost Util. Project*, pp. 1–14, 01 2014.

[6] O. Yadava and A. Kundu, "On or off pump coronary artery bypass grafting – is the last word out?," *Indian Heart Journal*, vol. 65, 2013.

[7] University of California San Francisco, "Minimally invasive aortic valve surgery." URL: https://adultctsurgery.ucsf.edu/conditions--procedures/minimally-invasive-aortic-valve-surgery.aspx, last checked on 20-02-2020.

[8] G. Mariscalco and F. Musumeci, "The minithoracotomy approach: A safe and effective alternative for heart valve surgery," *Ann Thorac Surg*, 2014. 2014;97:356-64.

[9] S. G. Raja, "Short-term clinical outcomes and long-term survival of minimally invasive direct coronary artery bypass grafting," 2018.

[10] Mathworks, "An optimal state estimator." URL: https://uk.mathworks.com/videos/understanding-kalman-filters-part-3-optimal-state-estimator--1490710645421.html, last checked on 28-04-2020, 2020.

[11] E. Wan and R. van der Merwe, "The unscented kalman filter," *Wiley Publishing*, 2001.

[12] H. Bruyninckx and T. Lefevre, "Kalman filters for nonlinear systems: A comparison of performance.," 2001.

[13] A. Gelb, "Applied optimal estimation," *M.I.T. Press*, 1974.

[14] G. Einicke, *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. Amazon Prime Publishing, 2019.

[15] R. Piche, "Online tests of kalman filter consistency," *International Journal of Adaptive Control and Signal Processing*, vol. 30, no. 1, pp. 115–124, 2016.

[16] Y. Bar-Shalom and X.-R. Li, *Estimation and tracking: principles, techniques, and software*. The Artech House radar library, Boston: Artech house, 1993.

[17] J. B. Rawlings, *Moving Horizon Estimation*. London: Springer London, 2013.

[18] O. Bebek and M. C. Cavusoglu, "Intelligent control algorithms for robotic-assisted beating heart surgery," *IEEE Transaction on Robotics*, pp. 468–480, 2007.

[19] M. Swindle and A. Smith, "Comparative anatomy and physiology of the pig," *Scandinavian Journal of Laboratory Animal Science*, vol. 25, pp. 11–21, 01 1998.

[20] T. Watanabe, P. M. Rautaharju, and T. F. McDonald, "Ventricular action potentials, ventricular extracellular potentials, and the ecg of guinea pig.," *Circulation Research*, vol. 57, no. 3, pp. 362–373, 1985.

[21] W. L. Brogan, *Modern Control Theory (3rd Ed.)*. USA: Prentice-Hall, Inc., 1991.

[22] S. G. Yuen and D. T. Kettler, "Robotic motion compensation for beating heart intracardiac surgery," *The International Journal of Robotics Research*, pp. 1355–1372, 2009.

[23] "The open d1namo dataset: A multi-modal dataset for research on non-invasive type 1 diabetes management." URL: `https://zenodo.org/record/1421616#.Xo3-pi2Q3RY`, last checked on 08-04-2020, 2018.

[24] G. Vrooijink and A. Denasi, "Model predictive control of a robotically actuated delivery sheath for beating heart compensation," *International Journey of Robotics Research*, pp. 193–209, 2017.

[25] Faulhaber, "Linear dc-servomotors series lm 1247 specifications." URL: `https://www.faulhaber.com/fileadmin/Import/Media/EN_LM1247_11_FMM.pdf`, last checked on 03-05-2020, 2020.

[26] PI USA, "V-277 pimag high-load linear actuator." URL: `https://static.pi-usa.us/fileadmin/user_upload/physik_instrumente/files/datasheets/V-277-Datasheet.pdf`, last checked on 03-05-2020, 2019.

[27] Mathworks, “Equation solving algorithms.” URL: `https://uk.mathworks.com/help/optim/ug/equation-solving-algorithms.html#f51887`, last checked on 26-04-2020, 2020.

[28] G. S. Almasi and A. Gottlieb, *Highly Parallel Computing*. USA: Benjamin-Cummings Publishing Co., Inc., 1989.

[29] E. Csencsics and G. Schitter, “System design and control of a resonant fast steering mirror for lissajous-based scanning,” *IEEE/ASME Transactions on Mechatronics*, vol. PP, pp. 1–1, 07 2017.

[30] S. Z. Sullivan, R. D. Muir, J. A. Newman, M. S. Carlsen, S. Sreehari, C. Doerge, N. J. Begue, R. M. Everly, C. A. Bouman, and G. J. Simpson, “High frame-rate multichannel beam-scanning microscopy based on lissajous trajectories,” *Opt. Express*, vol. 22, pp. 24224–24234, Oct 2014.