Master Thesis

Prediction filter for real time tumor tracking using the treatment table

Author(s):
Jöhl, Alexander

Publication Date:
2014

Permanent Link:
https://doi.org/10.3929/ethz-a-010294731

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
Alexander Jöhl

Prediction Filter for Real Time Tumor Tracking Using the Treatment Table

Master Thesis

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich
Department of Radiation Oncology, University Hospital Zurich

Supervision (ETHZ)

Dr. Marianne Schmid Daners
Prof. Dr. Lino Guzzella

Supervision (USZ)

Dr. Stephanie Lang
Dr. Stephan Klöck

August 2014
Preface

Research in the context of this thesis, the Radiation Oncology, is mainly carried out by medical physicists and physicians. This also held true for Treatment Couch Tracking, but it was recognized, that the problems coming up here largely overlap with the typical problems in control engineering. The Radiation Oncology Department of the University Hospital Zurich then contacted the IDSC of the ETH Zurich and a joint research project was taken up.

During my internship I was looking for a Master Thesis project and was offered this work by Marianne Schmid Daners and I want to thank her for entrusting me this work. This decision was not only on her side but also on the side of Stephanie Lang and Stephan Klöck. To both I give my thanks, too.

On the technical side the assistance of David Brühlmann for helping with electrical problems concerning step motors and Daniel Wagner for building the sensor mounting was valuable and they also receive my thanks.

I thank Stephanie Ehrbar for correcting and improving my language in the following text.
# Contents

Zusammenfassung v
Abstract vii
Nomenclature ix

1 Introduction 1
  1.1 Radiation Oncology ................................................. 1
    1.1.1 Tumor Treatment ............................................. 1
    1.1.2 Tumor Motion .................................................. 1
    1.1.3 Tumor Motion Mitigation ..................................... 2
  1.2 System Description ................................................. 3
  1.3 Aim and Outline of the Thesis ..................................... 3
    1.3.1 Remarks ..................................................... 4

2 Methods and Materials 5
  2.1 System Modelling .................................................. 5
    2.1.1 Human Respiration ........................................... 5
    2.1.2 Treatment Couch .............................................. 5
    2.1.3 Sensors ....................................................... 11
    2.1.4 Simulation Implementation ................................... 11
  2.2 Predictive Filters Theory ......................................... 11
    2.2.1 normalized Least Mean Squares Filter ......................... 11
    2.2.2 Multistep Linear Filter ..................................... 12
    2.2.3 Support Vector Regression Filter ............................. 13
    2.2.4 Interacting Multiple Model Kalman Filter ................... 14
    2.2.5 Local Circular Model Kalman Filter ......................... 16
    2.2.6 Fourier Kalman Filter ....................................... 17
  2.3 Couch Control Theory ............................................. 18
    2.3.1 No Controller ................................................ 18
  2.4 Hardware Implementation ......................................... 18
  2.5 Performance Indices ................................................ 19

3 Results 21
  3.1 Respiration Characteristics ...................................... 21
  3.2 Protura Characteristics .......................................... 22
    3.2.1 Protura without Extension .................................. 22
    3.2.2 Protura with Extension ...................................... 24
    3.2.3 Parameter Check ............................................. 27
  3.3 Prediction Filter Simulations ...................................... 28
    3.3.1 Results of nLMS Filter ...................................... 28
    3.3.2 Results of MULIN Filter ..................................... 29
    3.3.3 Results of SVR Filter ....................................... 31
3.3.4 Results of IMM Kalman Filter ........................................... 33
3.3.5 Results of LCM Kalman Filter ........................................... 34
3.3.6 Results of Fourier Kalman Filter ....................................... 36
3.3.7 Comparison of Prediction Filter Simulations ......................... 37
3.4 Combined Simulations and Experiments ................................. 39

4 Discussion and Outlook ..................................................... 41
4.1 Respiration Characteristics .............................................. 41
4.2 Protura Performance ...................................................... 41
4.3 Prediction Filters .......................................................... 42
4.4 Tracking Simulations ...................................................... 42

5 Conclusion ................................................................. 45

A Protura Derivations and Parameters ................................... 47
A.1 Mechanical Model Derivation ............................................ 47
A.1.1 General Rotations ................................................... 55
A.2 Mechanical Model with Extension Plate .............................. 56
A.3 Protura Parameters ...................................................... 60

B Respiration ................................................................. 63
B.1 Respiration Curves ....................................................... 63
B.2 Respiration Characteristics .............................................. 69
Zusammenfassung

Zweck

Eine mögliche Behandlung von Krebs ist die Beststrahlung des Tumors durch ionisierende Strahlung, welche möglichst nur vom Tumor absorbiert werden sollte. Im Falle eines sich bewegenden Tumors muss dessen Bewegung bei der Therapieplanung und Behandlung berücksichtigt werden. Die hier betrachtete Methode zur Berücksichtigung der Tumorbewegung misst die Atembewegung des Patienten, bestimmt daraus die Tumorbewegung und bewegt den Patienten so, dass der Tumor sich relativ zum Behandlungsstrahl nicht bewegt. Die Sensoren für die Messung der Atembewegung haben zum Teil grosse Totzeiten, welche durch die Prädiktionsfilter ausgeglichen werden soll. Die Totzeit beschreibt die Dauer zwischen der Messung und der tatsächlichen Verfügbarkeit des Sensorsignals zur weiteren Nutzung. Die Genauigkeit verschiedener Prädiktionsfilter ist in dieser Arbeit evaluiert worden. Der Aktuator (Treatment Table) verursacht durch seine limitierte Dynamik auch Fehler, daher ist der Aktuator modelliert und zusammen mit den Reglern und Prädiktionsfiltern simuliert worden.

Methoden


Resultate

Das Modell des Systems hat gegenüber dem realen System einen durchschnittlichen RMS Fehler von 0.57 mm in Längsrichtung (x) des Aktuatorsystems und 0.5 mm in vertikaler (z) Richtung. Die laterale (y) Richtung wurde nicht gemessen. Die RMS Fehler der Prädiktionsfilter liegen, mit Ausnahme des normalized Least Mean Squares-Prädiktionsfilters (nLMS), zwischen 0.4 mm und
0.65 mm bei einer Prädiktionszeit von 300 ms. Die Standardabweichungen der RMS Fehler eines Prädiktionsfilters sind grösser als die Unterschiede zwischen den Prädiktionsfiltern. Mit Ausnahme des Support Vector Regression-Prädiktionsfilters (SVR) liegen die Rechenzeiten pro Schritt unter 1 ms, der SVR-Prädiktionsfilter liegt bei knapp 20 ms für MATLAB Implementierungen. Die durchschnittliche Erhöhung der RMS Fehler bei Tumortracking durch eine Totzeit von 100 ms beträgt 0.54 mm während die Reduktion durch Benutzung eines Prädiktionsfilter zur Kompensation der Totzeit 0.53 mm beträgt.

**Schlussfolgerung**

Abstract

Purpose
A possible method to treat cancer is the irradiation of the tumor with ionizing radiation, which should be deposited as much as possible in the tumor. In cases of moving tumors their movements have to be mitigated. The method considered in this thesis measures the respiratory motion and calculates the corresponding tumor motion and moves the patient in such a way that the tumor does not move relative to the beam. Some sensors measuring the respiratory motion have large delay times, which can be compensated by prediction filters. The delay time is the time between the measurement and the actual availability of the signal for further use. The accuracy of several prediction filters has been evaluated in this work. The actuator (Treatment Table) also causes errors due to its limited dynamics, therefore, it was modeled and simulated together with the controllers and prediction filters.

Methods
The actuator system, the Protura Treatment Couch (CIVCO Medical Solutions, Kalona, IA, USA) was modeled using methods from Mechanics and was implemented using MATLAB/Simulink (The MathWorks, Inc., Natick, MA, USA). Using measurements of the real system unknown parameters were identified. The measurements were carried out using laser triangulation systems (Micro Epsilon Messtechnik GmbH & Co. KG, Ortenburg, Germany). The six different prediction filters considered here were implemented in MATLAB and their parameters were optimized for two different sampling times and one prediction time. The performance of the prediction filters for a single respiration curve is determined by the Root Mean Square (RMS) of the differences (errors) between the estimated value and the actual value at the corresponding time instants. High performance is achieved by having small RMS errors. During the optimization for each parameter combination of the prediction filter the RMS error for 19 different respiration curves were determined and the average over the respiration curves was considered as the performance index. With the optimal parameter combination the behaviour of the RMS of the prediction filters depending on different prediction times were examined. Finally one prediction filter was built into the model of the system and the complete system was simulated.

Results
The model of the system has an average RMS error of 0.57 mm in longitudinal (x) direction of the actuator system and 0.5 mm in vertical (z) direction. The lateral (y) direction was not measured. The RMS errors of the prediction filters lie, with exception of the normalized Least Mean Squares (nLMS) filter, between 0.4 mm and 0.65 mm for a prediction time of 300 ms. The standard deviations of the RMS error of a prediction filter are larger than the differences between the prediction filters. Except for the Support Vector Regression (SVR) prediction filter the computing times per step are less than 1 ms, while the SVR filter is at nearly 20 ms (MATLAB implementations). The average increase of the error RMS during tumor tracking caused by a delay of 100 ms is 0.54 mm while the reduction using a prediction filter compensating the delay time is 0.53 mm.
Conclusion

The actuator system was designed for static positioning of patient, but it still can compensate the tumor motion at least partly. However, another actuator system or another internal controller needs to be considered, such that also faster tumor motion can be compensated well. The model developed here allows testing of different prediction filters and other components of the complete system without great effort and without the real system, which is mainly used for clinical operation and rarely available. The prediction performance depends strongly on the specific respiration curve, therefore, the predictability of the respiration curves need to be considered more closely in further work.


## Nomenclature

### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{\text{RMS}}$</td>
<td>Root Mean Square error of prediction or tracking</td>
<td>[mm]</td>
</tr>
<tr>
<td>$e_{\text{CI}}$</td>
<td>Confidence Interval error of prediction or tracking</td>
<td>[mm]</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling Time</td>
<td>[s]</td>
</tr>
<tr>
<td>$x$</td>
<td>longitudinal direction of treatment couch, superior-inferior for patient</td>
<td></td>
</tr>
<tr>
<td>$y$</td>
<td>lateral direction of treatment couch, left-right for patient</td>
<td></td>
</tr>
<tr>
<td>$z$</td>
<td>vertical direction of treatment couch, anterior-posterior for patient</td>
<td></td>
</tr>
<tr>
<td>$\delta$</td>
<td>number of steps to predict ahead</td>
<td></td>
</tr>
<tr>
<td>$\mu_{e,\text{RMS}}$</td>
<td>average $e_{\text{RMS}}$</td>
<td>[mm]</td>
</tr>
<tr>
<td>$\sigma_{e,\text{RMS}}$</td>
<td>standard deviation of $e_{\text{RMS}}$</td>
<td>[mm]</td>
</tr>
</tbody>
</table>

### Indices

- RMS Root Mean Square
- CI Confidence Interval

### Acronyms and Abbreviations

- CA: Constant Acceleration
- CI: Confidence Interval
- CoG: Center of Gravity
- CT: Computer Tomography
- CV: Constant Velocity
- DNA: Deoxyribonucleic Acid
- ETHZ: Eidgenössische Technische Hochschule
- FIR: Finite Impulse Response
- IDSC: Institute for Dynamic Systems and Control
- IMM: Interacting Multiple Model
- KF: Kalman Filter
- LCM: Local Circular Model
- LHS: Left Hand Side
- MLC: Multileaf Collimator
- MLP: Multilayer Perceptron
- MULIN: Multistep Linear
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nLMS</td>
<td>normalized Least Mean Squares</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RHS</td>
<td>Right Hand Side</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>SVR</td>
<td>Support Vector Regression</td>
</tr>
<tr>
<td>USZ</td>
<td>UniversitätsSpital Zürich</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis is written at the University Hospital Zurich (USZ) and the Institute for Dynamic Systems and Control (IDSC) at the Swiss Federal Institute of Technology Zurich (ETHZ). This work is mainly motivated by the Radiation-Oncology Department of the USZ but the methods considered in this thesis fall in the realm of the IDSC expertise.

1.1 Radiation Oncology

Oncology deals with cancer and Radiation Oncology is its branch dealing with the treatment of cancer with radiation. Radiation beams are aimed at the tumors to be treated and these beams ionize the atoms of the tissue. The ionization happens directly with an ionizing beam (e.g. electron beam) or indirectly by a non-ionizing beam (e.g. gamma rays). The ionizations induce damages to the DNA, which can lead to the loss of growth capability of the tumor cells. The detailed physics are well explained in the book by Khan [1].

1.1.1 Tumor Treatment

Before the actual tumor treatment, the position and shape of the tumor need to be determined. This is the imaging phase, where the Computer Tomography (CT) scans are taken. From the images the tumor position and shape can be determined with submillimeter precision, see for example [2]. Afterwards, the treatment is planned by delineating the tumor and adding some margin around it to account for any uncertainties [3]. This results in the Planning Target Volume. Using computer programs the irradiation dose to the body can be calculated for different configurations of the radiation beam. The irradiation dose of the tumor should be as high as possible, while the surrounding tissue should not receive more dose than the organ specific tolerance dose.

1.1.2 Tumor Motion

In some cases the tumor may move during the treatment, especially, if it is located in the torso. The respiratory motion greatly influences the position of thoracic, liver and adrenal glands tumors, although a part can be attributed to the heartbeat. In the case of lung tumors the motion may have a peak to peak motion of up to 34 mm [4]. Heartbeat contributes 1 to 4 mm [5]. Since the irradiation dose of the tumor is prioritized, in case of tumor motion the margins need to be increased, such that the tumor always receives high radiation doses. This also results in high radiation dose to healthy tissue.
1.1.3 Tumor Motion Mitigation

The methods to mitigate this problem can be categorized as follows [6]:

- Respiration gating
- Breath-hold
- Forced shallow-respiration
- Respiration synchronization, tumor tracking

The respiratory motion is somewhat periodic, so the tumor will be in a certain position repeatedly. In respiration gating this correlation between the respiratory motion and the tumor position is exploited by only switching on the radiation beam when the tumor is in a predetermined position. This requires observation of the respiratory motion and prolongs the treatment time. In the breath-hold method the patient stops breathing in complete exhale or inhale position for a few seconds and during this time the beam is switched on. This can be rather stressful or even impossible for the patient and also prolongs the treatment time. Using the forced shallow-respiration means that the moving amplitude of the tumor is reduced. This also means stress for the patient. In respiration synchronization or tumor tracking the beam or the patient is moved such that the motion of the tumor relative to the beam is compensated. Here again observation of the respiratory motion is needed.

There are several methods to do tracking: Moving the accelerator (the beam source), modifying the beam or moving the patient. The first concept was realized with the Cyberknife system [7]. There the accelerator is mounted on a robotic arm. Its disadvantage, compared to the other systems, is the long time it takes for one treatment session, due to its low dose rate. The Vero system [8] works with a different kind of robotic system and achieves shorter treatment times than the Cyberknife system. Their main drawback though is their high costs. The second method, modifying the beam, usually does not require special hardware, because it is often already built in a C-arm linear accelerator, the most used type of linear accelerator worldwide. This method is commonly called MLC Tracking, because of the hardware involved, the Multileaf Collimator (MLC). The MLC essentially acts as a stencil on the beam and can be modified online. It consists of multiple little metallic bars, which can move in one direction. There is one row on each side of the beam as shown in Figure 1.1. This setup results in high resolution and speed in one dimension but low resolution (due to the thickness of the little metallic bars) and speed in the other dimension. During tracking these bars move if needed, but tracking a movement in the perpendicular direction may not be feasible with the maximum speed of the bars. Additionally, the MLC is already used to change the outline of the beam dynamically, when the beam source is rotated around the patient.

Figure 1.1: Beam’s eye view of the Multileaf Collimator, the grey bars obstruct parts of the radiation beam, the arrows indicate the possible moving directions of the bars

The third method, moving the patient (also called Couch Tracking), has the advantage of using already available hardware, an actuated couch. The couch is commonly used for fine positioning of the patient before a treatment session, since the patient may not be in exactly the same position on the couch as during the imaging phase.

This thesis considers the method of Couch Tracking. To get an overview and a feel on what is possible, some results from literature are presented in the following. In [9] Couch Tracking
was implemented and they reached a Root Mean Square (RMS) error of 0.88 mm. In [10] a mean absolute error of 0.44 mm was achieved. At the USZ an actuated couch and sensors for measuring the respiration are available. These sensors differ in their sampling times and their delay times, which can be considerably large. The delay times have a significant influence on the behaviour of couch tracking. Therefore, prediction filters are developed, which predict the respiratory motion ahead. There are several prediction filters considered in this thesis, the first one being the normalized Least Mean Squares (nLMS) filter from [11], where it reaches a RMS error of 0.8546 mm. The second one is the Multistep Linear (MULIN) filter [12] reaching a RMS error of 0.169 mm. These two filters are model free filters and need very little computational effort. The Support Vector Regression (SVR) filter is presented in [13] and in [12] it reaches a RMS error of 0.129 mm. The SVR filter is based on machine learning and requires high computing power. The Interacting Multiple Model (IMM) Kalman filter [14] reaches a RMS error of 0.25 mm. The Local Circular Motion (LCM) Kalman filter [15] reaches a normalized RMS error of approximately 0.15. The normalized RMS error is the RMS error divided by the standard deviation of the input signal. The Fourier Kalman filter [16] is used for prediction of the heartbeat. It reached a RMS error of 1.01 mm.

1.2 System Description

The treatment couch tracking system consists of several elements. First there is the patient and its respiration. It is considered to be an external signal source, so it is assumed that there is no influence on the patient by the rest of the system. The sensors measure the respiratory motion of the patient. The sensor may introduce large delays in data transmission due to high amount of processing needed. These delays can be compensated by the use of prediction filters. The measured respiratory motion signals then are converted by an algorithm to the tumor motion signal. The tumor moves relative to the body of the patient, so the patient has to be moved in the opposite direction such that the tumor does not move relative to the inertial frame. Therefore, the tumor motion signal is multiplied by \(-1\) and fed to the controller. The controller sends a position signal to the treatment couch, which then moves accordingly. In Figure 1.2 the system is described by a signal flow diagram, the arrows represent signals and the blocks represent subsystems, which generate new signals and may be driven by other signals.

![Figure 1.2: System description, the blocks represent subsystems and the arrows represent signals.](image)

1.3 Aim and Outline of the Thesis

The prediction filters can be tested without using any other elements of the complete system, but to know the real error the complete system is needed. The time, during which the complete system is available for testing, is limited since it is already used in clinical operation. Therefore, a model of the complete system is derived to save time and enable better understanding of the complete system. In Section 2.1 the system is modeled, especially the treatment couch used at the USZ. In Section 2.2 six different prediction filters are presented and their mathematical reasoning
explained. The hardware setup is shown in Section 2.4. The respiration curves used in this thesis are analyzed in Section 3.1. The real treatment couch is characterized and the fit of its model is shown in Section 3.2. In Section 3.3 the prediction filters are simulated without the system and their results are presented. The prediction filter with the best performance was determined and chosen for the simulation of the model of the complete system. The results are shown in Section 3.4.

1.3.1 Remarks

We only considered translational movements of the couch to limit the complexity of the system, especially reducing the number of inputs and outputs.
Furthermore, if there is no mention of respiration to tumor conversion, only the respiratory motion was tracked.
Chapter 2

Methods and Materials

2.1 System Modelling

2.1.1 Human Respiration

The underlying model of the human respiration is considered to be unknown. As shown in the introduction in Figure 1.2, the human respiration is considered to have no signals driving it. For the experiments and simulations the signals are measured respiration curves of real patients and are shown in the Appendix B. An analysis of the available measurements is shown in Section 3.1.

2.1.2 Treatment Couch

The robotic treatment couch moving the patient is the Protura treatment couch by CIVCO Medical Systems. It is usually used for fine positioning of the patient before a treatment session. In this thesis it is used as the actuator moving the patient lying on it. It consists of a base and a platform, both connected by six legs in parallel. The joints connecting the legs and the base can move linearly, as shown in Figure 2.1. Therefore, each leg has one degree of freedom and having six legs, this results in a system with six degrees of freedom, three translational and three rotational for the platform. These kind of systems are called Stewart-Gough platforms. A literature survey did not reveal a model of this type of Stewart platform, therefore, one is derived here. The method of the modeling process follows the same principle as in [17].

Figure 2.1: The Protura System, including the $x,y,z$ directions of the Protura system.
Protura Mechanical Model

Due to its length the complete derivation of the equations of motion of the system is in the Appendix A, Section A.1. Here only the method (taken from [17]) and the final equations are shown. First a representative system is introduced, which is shown in Figure 2.2. Since the legs are not directly connected to each other, one leg at a time can be considered and for the other legs only the parameter values need to be changed.

Figure 2.2: The representative system. The capital letters designate points, the small letters designate vectors. \( O \) is the origin, \( T \) is the center of the platform, \( J \) is the leg-platform joint, \( B \) is the leg-base joint and \( B_0 \) is the initial \( B \) position.

Now the linear momentum balances and the angular momentum balances of the leg and the platform need to be formulated, starting with the linear momentum of the leg using the center of gravity of the leg and the angular momentum around point \( B \):

\[
mg + F_{rB} + F_{act} + F_s = m\ddot{r}_L
\]  
(2.1)

\[
M_{rB} + M_{rJ} + r_L \times (mg) + l \times F_s = m\ddot{r}_L + \ddot{b} + I_L \dddot{\omega}_L + \omega_L \times (I_L \omega_L)
\]  
(2.2)

Where \( m \) is the mass of a leg, \( I_L \) is the rotational inertia of a leg and \( g \) is the gravitational acceleration. Vectors and matrices are written in bold, \( t \) and \( \dot{t} \) are the first (velocity) and second (acceleration) time derivative of \( t \), respectively. \( \omega \) is the rotational velocity and \( \ddot{\omega} \) the rotational acceleration. The equations (2.1) and (2.2) are actually written down for each leg, so they appear six times. But since they all differ only by their numerical values, only one pair needs to be manipulated for the derivation.

Now the same is done for the platform, but the angular momentum balance is around point \( T \):

\[
m_T g + F_{ext} - \sum_{i=1}^{6} (F_s)_i = m_T\ddot{r}_T
\]  
(2.3)

\[- \sum_{i=1}^{6} (M_{rJ})_i + M_{ext} - \sum_{i=1}^{6} (p \times F_s)_i + r_T \times m_T g = m_T\ddot{r}_T - \ddot{t} + I_T \dddot{\omega}_T + \omega_T \times (I_T \omega_T)
\]  
(2.4)

The friction forces \( F_{rB}, M_{rB}, M_{rJ} \) will be specified after the derivation of the model to keep the model flexible. \( F_{act} \), which does not appear here directly but in the Appendix A, is used as the
control input to the mechanical model. \( F_{\text{ext}}, M_{\text{ext}} \) can be used to add other force elements, for example the load of a patient placed on the platform. \( F_s \) is the joint force at point \( J \) needed to respect the constraints introduced by the joint. These \( \left( F_s \right)_i \) need to be eliminated from the equations. The first step is to solve the leg balances of momentum for \( F_s \) such that the acceleration variables appear linearly on the other side of the resulting equations. Then, one can plug in these equations into the equations of momentum balances for the platform, such that \( \left( F_s \right)_i \) are eliminated.

There are several new variables introduced during the derivation to shorten the expressions. Their definitions are shown in the Appendix A.

The resulting equations are:

\[
\begin{align*}
\sum_{i=1}^{6}m_{s,i} \ddot{t} + m_T & \ddot{t} = \sum_{i=1}^{6} m_{s,i} \ddot{\dot{p}}_i - m_T \ddot{r}_T \\
& - \sum_{i=1}^{6} \ddot{t} \ddot{p}_i + \ddot{t} \ddot{I}_T = 0
\end{align*}
\]

**Protura Mechanical Model with Extension Plate**

At one of the short sides of the platform an extension plate can be attached. The joint may have play in the bearing and the extension plate might bend. Therefore, the extension plate is modeled as an additional rigid body. The play in the bearing and the bending are modeled by introducing one degree of freedom, namely the rotation around the side of the platform, by which the extension is attached. Additionally a spring damper element depending on the angle is assumed to act between the extension and the platform to account for bending. An overview is given in Figure 2.3.

![Figure 2.3: The representative system with the extension plate.](image)

We start with writing down the equations of momentum of the extension plate:

\[
\begin{align*}
m_E \ddot{r}_{OE} &= m_E g + F_E + F_{\text{load}} \\
I_E \ddot{\omega}_E + \omega_E \times I_E \omega_E + m_E r_E \times \ddot{s}_E &= M_E + M_{FD} + m_E r_E \times g + r_{\text{load}} \times F_{\text{load}}
\end{align*}
\]
accommodate for the new forces, the momentum equations of the platform have to be modified:

\[ m_T \mathbf{g} + \mathbf{F}_{\text{ext}} - \sum_{i=1}^{6} (\mathbf{F}_i) = m_T \ddot{\mathbf{r}}_T \]

\[- \sum_{i=1}^{6} (M_{i,j}) + \mathbf{M}_{\text{ext}} - \sum_{i=1}^{6} ((A_{iTT} \mathbf{p}) \times \mathbf{F}_i) + \mathbf{r}_T \times m_T \mathbf{g} - \mathbf{p}_E \times \mathbf{F}_E - \mathbf{M}_{\text{FD}} =\]

\[ m_T \mathbf{r}_T \times \ddot{\mathbf{r}}_T + \mathbf{I}_T \dot{\omega}_{TT} + \omega_{TT} \times (\mathbf{I}_T \omega_{TT}) \]

The complete derivation is in the Appendix A in Section A.2. The final equations of motion are:

\[
\begin{bmatrix}
A_1 + m_E E_3 \\
A_2 + m_E (\mathbf{E}_p + \ddot{\mathbf{r}}_E) \quad B_1 - m_E (\mathbf{E}_p + \ddot{\mathbf{r}}_E) \\
\quad m_E \mathbf{E}_p \ddot{\mathbf{r}}_E \\
B_2 + \mathbf{I}_E - m_E \mathbf{E}_p \mathbf{E}_p \\
\end{bmatrix}
\begin{bmatrix}
\ddot{x} \\
\ddot{y} \\
\ddot{z} \\
\end{bmatrix}
= \begin{bmatrix}
-C_1 \\
-C_2 \\
0 \\
0 \\
\end{bmatrix}
= \begin{bmatrix}
-\mathbf{V}_{FE} - \mathbf{F}_{FE} \\
-\mathbf{V}_{ME} - \mathbf{M}_{ME} - \mathbf{p}_E \times (\mathbf{V}_{FE} + \mathbf{F}_{FE}) \\
-\mathbf{e}_y^T \mathbf{V}_{ME} - \mathbf{e}_y^T \mathbf{M}_{ME} \\
0 \\
0 \\
\end{bmatrix}
\]

\[ (2.9) \]

Protura Internal Controller

The input of the model derived above are the six forces $F_{\text{act},i}$ acting on the base-leg joints. However, the real system has the position and orientation as inputs, and an internal controller converts the inputs into the needed forces. Therefore, an internal controller has to be added to the model. But since the aim is to get an overall system behaviour as close as possible to the real system, the controller developed here is not elaborate.

The input of the model derived above are the six forces $F_{\text{act},i}$ acting on the base-leg joints. However, the real system has the position and orientation as inputs, and an internal controller converts the inputs into the needed forces. Therefore, an internal controller has to be added to the model. But since the aim is to get an overall system behaviour as close as possible to the real system, the controller developed here is not elaborate.

The concept of the model of the internal controller consists of four parts: A reference conversion, a trajectory generator, a Proportional-Integrative-Derivative (PID) feedback part and a feedforward part, as shown in Figure 2.4. The reference input to the controller is the position $t_{\text{ref}}$ of the center of the platform and the rotation of the platform, represented by a quaternion $q_{\text{ref}}$. The use of quaternions is explained in the Appendix A in Section A.1.1. Using the kinematics of the model derived in the Appendix A, these reference values are converted to reference values for the base-leg joint position $\alpha_{i,\text{ref}}$. The first and second derivatives of the reference values for the base-leg joint position are limited by the trajectory generator. In the feedback part the difference between the actual position $\alpha_i$ of the joints and the trajectory positions $\alpha_{i,\text{traj}}$ are fed into a PID controller. The output of the PID controller constitutes the feedback part $F_{\text{act,FB},i}$. The feedforward part of $F_{\text{act},i}$ is calculated from a simplified mechanical model of the treatment couch and solved for $F_{\text{act,FF},i}$. The simplification is done by setting all time derivatives of the model to zero. The resulting $F_{\text{act,FF},i}$ equals the needed force to keep the system in steady state given a known load.
The derivation of the feedforward part of the controller is done as following: Take (A.86) and set the left hand side to zero.

$$0 = -\sum_{i=1}^{6} h_{i,j} + F_{\text{ext}} + m T g$$

$$0 = -\sum_{i=1}^{6} p_{i} \times h_{i,j} + M_{\text{ext}} + m T r_{T} \times g$$

Where $h_{i,j}$ is given by (from (A.80) and (A.78)):

$$h_{i} = 0 + h_{i} = -\frac{d}{dt} (0 + 0 + M_{i}) + \frac{d}{dt} (0 + 0) + \frac{1}{dt} F_{1}$$

Where $M_{1}$ and $F_{1}$ consist of frictional and gravitational forces, for the feedforward the friction is neglected. From (A.57) and (A.65):

$$M_{1} = 0 + 0 - m r_{L} \times g$$

$$F_{1} = -ld^{T} (m g + F_{\text{act}}) = -mld^{T} g - ld^{T} F_{\text{act,FF}}$$

With $d^{T} F_{\text{act,FF}} = d^{T} F_{\text{act,FF}}$, since the $d$ and $F_{\text{act,FF}}$ vectors are collinear. These equations plugged into (2.13) give:

$$h_{t} = \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g - \frac{d}{dt} F_{\text{act,FF}}$$

And this expression inserted into (2.11) and (2.12):

$$\sum_{i=1}^{6} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g - \frac{d}{dt} F_{\text{act,FF}} \right)_{i} = F_{\text{ext}} + m T g$$

$$\sum_{i=1}^{6} \left( \tilde{p}_{i} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g - \frac{d}{dt} F_{\text{act,FF}} \right) \right)_{i} = M_{\text{ext}} + m T r_{T} \times g$$

Rearranging the terms:

$$-\sum_{i=1}^{6} \frac{d}{dt} F_{\text{act,FF},i} = F_{\text{ext}} + m T g - \sum_{i=1}^{6} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g \right)_{i}$$

$$-\sum_{i=1}^{6} \tilde{p}_{i} \frac{d}{dt} F_{\text{act,FF},i} = M_{\text{ext}} + m T r_{T} \times g - \sum_{i=1}^{6} \tilde{p}_{i} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g \right)_{i}$$

So a system of linear equations is available, which can be solved for $F_{\text{act,FF},i}$ numerically:

$$\begin{pmatrix}
\frac{d}{dt} \frac{d}{dt} \\
\tilde{p}_{1} \frac{d}{dt} \\
\vdots \\
\tilde{p}_{6} \frac{d}{dt}
\end{pmatrix}
\begin{pmatrix}
F_{\text{act,FF},1} \\
\vdots \\
F_{\text{act,FF},6}
\end{pmatrix} =
\begin{pmatrix}
F_{\text{ext}} + m T g - \sum_{i=1}^{6} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g \right)_{i} \\
M_{\text{ext}} + m T r_{T} \times g - \sum_{i=1}^{6} \tilde{p}_{i} \left( \frac{d}{dt} (m r_{L} \times g) - m \frac{d}{dt} g \right)_{i}
\end{pmatrix}$$

The feedback $F_{\text{act,FB},i}$ has a simple formula:

$$F_{\text{act,FB},i} = k_{p} (\alpha_{\text{traj},i} - \alpha_{i}) + k_{i} \int (\alpha_{\text{traj},i} - \alpha_{i}) \, dt + k_{d} \frac{d}{dt} (\alpha_{\text{traj},i} - \alpha_{i})$$

Finally the output of the internal controller is:

$$F_{\text{act},i} = F_{\text{act,FB},i} + F_{\text{act,FF},i}$$
Trajectory Generator

The feedback part of the internal controller does not directly follow the reference, which is an external input. Instead there is a trajectory generator, which modifies the reference, such that certain constraints are respected. In this case the constraints are the velocity and the acceleration of the platform. The exact implementation of the trajectory generator is not known and is modeled by its input-output behavior. The method for trajectory generation used in the simulation is taken from [18]. The trajectory generator consists of two cascaded control loops, where the controllers have switching control laws. In Figure 2.5 the signal flow is shown. The output $r_{\text{traj}}$ follows the input $r_{\text{ref}}$ while respecting desired saturations on the first and second derivative of $r_{\text{traj}}$. These constraints are imposed on the signal by the control laws $L_1$ and $L_2$.

We call the first derivative of $r_{\text{traj}}$ the speed and the second derivative the acceleration. The corresponding parameters representing the constraints are: $v_{\text{max}}$ and $a_{\text{max}}$. The formula for $L_1$ is then as follows:

$$ u_1 (\dot{r}_{\text{traj}}, u_2) = \begin{cases} a_{\text{max}} & \text{if } \dot{r}_{\text{traj}} < u_2 \\ 0 & \text{if } \dot{r}_{\text{traj}} = u_2 \\ -a_{\text{max}} & \text{if } \dot{r}_{\text{traj}} > u_2 \end{cases} $$

(2.24)

The formulas for $L_2$ are:

$$ u_2 (r_{\text{ref}}, r_{\text{traj}}, \dot{r}_{\text{traj}}) = \begin{cases} v_{\text{max}} & \text{if } r_{\text{traj}} - r_{\text{ref}} < h_2 (\dot{r}_{\text{traj}}) \text{ or } r_{\text{traj}} - r_{\text{ref}} = h_2^{-} (\dot{r}_{\text{traj}}) \\ 0 & \text{if } r_{\text{traj}} - r_{\text{ref}} = \dot{r}_{\text{traj}} = 0 \\ -v_{\text{max}} & \text{if } r_{\text{traj}} - r_{\text{ref}} > h_2 (\dot{r}_{\text{traj}}) \text{ or } r_{\text{traj}} - r_{\text{ref}} = h_2^{+} (\dot{r}_{\text{traj}}) \end{cases} $$

(2.25)

$$ h_2 (\dot{r}_{\text{traj}}) = \begin{cases} h_2^{-} (\dot{r}_{\text{traj}}) = \frac{\dot{r}_{\text{traj}}^2}{2a_{\text{max}}} & \text{if } \dot{r}_{\text{traj}} < 0 \\ 0 & \text{if } \dot{r}_{\text{traj}} = 0 \\ h_2^{+} (\dot{r}_{\text{traj}}) = \frac{\dot{r}_{\text{traj}}^2}{2a_{\text{max}}} & \text{if } \dot{r}_{\text{traj}} > 0 \end{cases} $$

(2.26)

Here $u_1$ and $u_2$ are intermediate variables as shown in Figure 2.5. $h_2$ is an intermediate variable used inside $L_2$. The output $r_{\text{traj}}$ is sent through a low pass filter, since it chatters. An exemplary behavior of the trajectory generator is shown in Figure 2.6, the reference signal makes a step and the trajectory signal follows it as quickly as possible while respecting the speed and acceleration constraints. This can be seen by the finite slope and the finite curvature of the trajectory signal.

![Figure 2.5: Signal flow of the Trajectory Generator.](image)

![Figure 2.6: The step response of the trajectory generator. Note the finite slope and curvature of the trajectory.](image)
2.1.3 Sensors

The sensor was generically modeled with a zero order hold and a delay element. As shown in Figure 2.7, first the signal is delayed by some constant time and then the signal is sampled with a zero order hold. The zero order hold keeps the value of the output constant until a certain time has passed and then updates the output to the current input.

![Diagram of sensor model](image)

Figure 2.7: The delay element delays the signal by some constant time and the zero order hold keeps the output constant until a certain sampling time has passed and then updates the output to the current input.

2.1.4 Simulation Implementation

The models shown above are implemented using MATLAB/Simulink R2013b (The MathWorks, Inc., Natick, MA, USA). Simulink is a programming environment developed for implementing dynamic models, which have to be solved using numerical ordinary differential equations solvers. The models are implemented using blocks and arrows the same way as in Figures 2.4, 2.5 and 2.7.

2.2 Predictive Filters Theory

As mentioned in the introduction the sensors used to measure the respiration require a lot of computing time and therefore have a low frequency and high latency. In control systems this high latency is difficult to handle. In order to mitigate the effects of the high latency, prediction filters are used. In this thesis two model free filters, three model based filters and one machine learning filter are presented.

2.2.1 normalized Least Mean Squares Filter

The normalized Least Mean Squares (nLMS) filter is presented in [11]. The results of optimizing the parameters of the nLMS filter are shown in Section 3.3.1. Essentially this filter is a FIR (Finite Impulse Response) filter, whose weights are modified at each step. At timestep $k$, there is an input history of length $n$, comprising of past and present measurements $y$ and a vector of weights:

$$
\mathbf{y}(k) = \begin{bmatrix}
y(k) \\
y(k-1) \\
\vdots \\
y(k-n+1)
\end{bmatrix}, \quad \mathbf{w}(k) = \begin{bmatrix}
w_0(k) \\
w_1(k) \\
\vdots \\
w_{n-1}(k)
\end{bmatrix}
$$

(2.27)

The output of the filter is simply the weight vector multiplied with the input history vector:

$$
\hat{y}(k + \delta | k) = \mathbf{w}(k)^T \mathbf{y}(k)
$$

(2.28)

The crucial point is the determination of $\mathbf{w}(k)$. The idea is to minimize the cost, which is the expected square of the error, which is the difference between the presently measured value $y(k)$ and the predicted value $\hat{y}(k|k-\delta)$.

$$
e(k) = y(k) - \hat{y}(k|k-\delta) = y(k) - \mathbf{w}(k-\delta)^T \mathbf{y}(k-\delta)
$$

(2.29)

$$
C = E \left[ e(k)^2 \right]
$$

(2.30)
We take the gradient of the expected error with respect to the weights:
\[
\frac{\partial C}{\partial w(k)} = 2 \mathbb{E} \left[ \frac{\partial e(k)}{\partial w(k)} e(k) \right] = 2 \mathbb{E} [-y(k - \delta)e(k)]
\]  
(2.31)

The expected value is not known but is approximated by:
\[
\mathbb{E} [y(k - \delta)e(k)] \approx y(k - \delta)e(k)
\]  
(2.32)

Now the direction of the steepest ascend is available, so the weights \(w(k)\) can be modified to minimize the expected error, by taking a step in the opposite direction with a gain \(\mu_{\text{step}}\):
\[
w(k + 1) = w(k) - \frac{\mu_{\text{step}}}{2} \frac{\partial C}{\partial w(k)} = w(k) + \mu_{\text{step}}y(k - \delta)e(k)
\]  
(2.33)

Depending on the magnitude of \(y(k - \delta)\) the actual step may be large even though the error \(e(k)\) is small. To solve this, (2.33) is modified by introducing a variable step size:
\[
w(k + 1) = w(k) + \frac{\mu_{\text{step}}}{\epsilon + y(k - \delta)^2} y(k - \delta)e(k)
\]  
(2.34)

Where \(\epsilon\) is a small positive number used to avoid divisions by zero.

### 2.2.2 Multistep Linear Filter

The Multistep Linear (MULIN) filter is presented in [19] and [12]. The results are shown in Section 3.3.2. For the simplest version it is assumed that the difference between the present value \(y(k)\) and the past value \(y(k - 1)\) is similar to the difference between \(y(k + 1)\) and \(y(k)\). We define the difference function:
\[
\Delta_{(y,\delta)}(k) = y(k - \delta) - y(k)
\]  
(2.35)

Using the difference function one can predict \(\delta\) steps ahead by using this formula (MULIN0):
\[
\hat{y}(k + \delta|k) = y(k) - \Delta_{(y,\delta)}(k)
\]  
(2.36)

The performance deteriorates quickly if the assumption does not hold. To improve \(\Delta_{(y,\delta)}(k)\) is expanded in a similar way as in (2.36) by replacing \(y\) by \(\Delta_{(y,\delta)}\):
\[
\hat{y}_{\Delta}(k + \delta|k) = \Delta_{(y,\delta)}(k) - \Delta_{(\Delta_{(y,\delta)},l)}(k) = \Delta_{(y,\delta)}(k) - (\Delta_{(y,\delta)}(k - l) - \Delta_{(y,\delta)}(k))
\]  
(2.37)

Where \(l\) influences, how far past values of \(\Delta_{(y,\delta)}\) should be taken into account. Now in (2.36) \(\Delta_{(y,\delta)}(k)\) is replaced by \(\hat{y}_{\Delta}(y,\delta)(k)\) and get MULIN1:
\[
\hat{y}(k + \delta|k) = y(k) - 2\Delta_{(y,\delta)}(k) + \Delta_{(y,\delta)}(k - l)
\]  
(2.38)

Repeat the method in (2.37) for (2.38) for both \(\Delta_{(y,\delta)}(k)\) and \(\Delta_{(y,\delta)}(k - l)\) resulting in MULIN2:
\[
\hat{y}(k + \delta|k) = y(k) - 4\Delta_{(y,\delta)}(k) + 4\Delta_{(y,\delta)}(k - l) - \Delta_{(y,\delta)}(k - 2l)
\]  
(2.39)

And MULIN3:
\[
\hat{y}(k + \delta|k) = y(k) - 8\Delta_{(y,\delta)}(k) + 12\Delta_{(y,\delta)}(k - l) - 6\Delta_{(y,\delta)}(k - 2l) + \Delta_{(y,\delta)}(k - 3l)
\]  
(2.40)

Of course this can be repeated recursively ad infinitum. To reduce the sensitivity to noise an exponential smoothing factor \(\alpha\) can be used with the following modification:
\[
\tilde{y}_{\text{filt}}(k + \delta|k) = \alpha \hat{y}(k + \delta) + (1 - \alpha) \hat{y}(k - 1 + \delta)
\]  
(2.41)

It is a first order low pass filter acting on the prediction value and introduces a phase shift degrading the predictions.
2.2.3 Support Vector Regression Filter

The idea is from [13] and the library, which implements the methods, is from [20]. A good explanation of the theory behind Support Vector Regression (SVR) is given in [21]. The results are shown in 3.3.3. The support vector regression method is actually an optimization problem, which fits a hyperplane to the data. The optimization problem is stated as follows:

\[
\min _{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum _{i=1}^{n} (\xi _i + \xi _i^*) \\
\text{s.t. } y_i - (\mathbf{w}^T \mathbf{x}_i + b) \leq \epsilon + \xi _i \\
(\mathbf{w}^T \mathbf{x}_i + b) - y_i \leq \epsilon + \xi _i^* \\
\xi _i, \xi _i^* \geq 0 \forall i \in \{1, \ldots, n\}
\] (2.42)

Here the \((y_i, \mathbf{x}_i), i = 1, \ldots, n\) represent the available data, where \(y\) is the component considered to be dependent on \(\mathbf{x}\). Here \(y\) is the respiration and \(\mathbf{x}\) is the time. \(\mathbf{w}\) is the vector defining the hyperplane. The \(\epsilon\) allows a certain deviation of the data from the hyperplane without influencing the cost. Any larger deviations increase the values of \(\xi, \xi^*\) and therefore the cost. The problem is convex, so it is solvable with known algorithms. The solution then is:

\[
f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b
\] (2.43)

But in general the data is much better fitted with a nonlinear function. But directly using nonlinear functions instead of the hyperplane may cause loss of the convexity of the optimization problem. The idea is now to map the data into a higher dimensional space using some function:

\[
\phi : \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{m_2}, \mathbf{x} \mapsto \phi(\mathbf{x}), \text{ where } m_2 > m_1.
\]

This improves the possibility to fit a hyperplane to the data, the hyperplane being in higher dimensions as well. So in the optimization problem (2.42) the \(\mathbf{w}^T \mathbf{x}\) part is replaced by \(\mathbf{w}^T \phi(\mathbf{x})\). This, however, introduces a new problem, the mapping of the data into higher dimensions may be computationally intensive. It can be solved by considering the dual of (2.42):

\[
\max _{\alpha, \alpha^*} \frac{1}{2} - \sum _{i,j=1}^{n} \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) - \epsilon \sum _{i=1}^{n} (\alpha_i + \alpha_i^*) + \sum _{i=1}^{n} y_i (\alpha_i + \alpha_i^*) \\
\text{s.t. } 0 \leq \alpha_i \\
(\alpha_i - \alpha_i^*) \leq C \\
\sum _{i=1}^{n} (\alpha_i - \alpha_i^*) = 0, \forall i = 1, \ldots, n
\] (2.44)

Here the mapping of the data only appears as a scalar product and is replaceable by a so called kernel function:

\[
k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})
\] (2.45)

Instead of mapping the data into higher dimensions with nonlinear functions and then taking the scalar product, the kernel function is a nonlinear function that takes the original data and computes a scalar value avoiding higher dimensions. There are restrictions on what kind of functions can be used as a kernel function, for details see [21]. The solution is then computed as follows:

\[
f(\mathbf{x}) = \sum _{i=1}^{n} (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b
\] (2.46)

However using this method in real time is still not feasible, since at every step a new optimization problem has to be solved, because the old data is discarded and new data is added. [20] developed and implemented an algorithm, which allows us to find the solution to the optimization problem...
These are both linear systems and can be described by the system matrices with variance \( Q \) where \( x \) systems. The algorithm works as follows: matrix with its elements \( \pi \) to the CV model but with nonzero dynamics of the third state, the acceleration: noise is considered to be random accelerations. The constant acceleration (CA) model is similar while incrementally adding or removing data. In this application the \( x_i, \forall i = 1, \ldots, n \) are the last \( n \) time instances \( t(k-i), \forall i = 1, \ldots, n \) and \( y_i \) are the corresponding respiration positions. In every step the present measurement data is added, the prediction is computed by evaluating (2.46) at the desired \( x \) and then the oldest measurement data is discarded. The amount of data used in every step can be set as a parameter.

### 2.2.4 Interacting Multiple Model Kalman Filter

This filter, from [14], is model based. The results are shown in 3.3.4. It consists of two models describing the respiration in different phases and are implemented in two Kalman filters running in parallel. The results of the Kalman filters are combined by the interacting multiple model (IMM) filter. One model describes the constant velocity phases of the respiration and the other model the constant acceleration phases. The constant velocity (CV) model is formulated as follows:

\[
\begin{bmatrix}
\dot{x}_1(k) \\
\dot{x}_2(k) \\
\dot{x}_3(k)
\end{bmatrix} = 
\begin{bmatrix}
1 & T_s & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1(k-1) \\
x_2(k-1) \\
x_3(k-1)
\end{bmatrix} + 
\begin{bmatrix}
\frac{1}{2}T_s^2 \\
\frac{1}{2}T_s^2 \\
T_s
\end{bmatrix} v(k-1)
\]

\[y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + w(k) \tag{2.47}\]

Where \( x_1 \) is the position and \( x_2 \) is the velocity. The process noise \( v \) is assumed to be zero mean with variance \( Q \) and the measurement noise \( w \) is also zero mean with variance \( R \). The process noise is considered to be random accelerations. The constant acceleration (CA) model is similar to the CV model but with nonzero dynamics of the third state, the acceleration:

\[
\begin{bmatrix}
\dot{x}_1(k) \\
\dot{x}_2(k) \\
\dot{x}_3(k)
\end{bmatrix} = 
\begin{bmatrix}
1 & T_s & \frac{1}{2}T_s^2 \\
0 & 1 & T_s \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1(k-1) \\
x_2(k-1) \\
x_3(k-1)
\end{bmatrix} + 
\begin{bmatrix}
\frac{1}{2}T_s^2 \\
\frac{1}{2}T_s^2 \\
T_s
\end{bmatrix} v(k-1)
\]

\[y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + w(k) \tag{2.48}\]

These are both linear systems and can be described by the system matrices \( F_i, G_i, H_i, i \in \{CV, CA\} \). The complete hybrid system then is:

\[
\xi(k) = \sum_{i \in \{CV,CA\}} \mu_i(k-1) \left( F_i \xi_i(k-1) + G_i v_i(k-1) \right)
\]

\[y(k) = \sum_{i \in \{CV,CA\}} \mu_i(k-1) \left( H_i \xi_i(k-1) + w_i(k-1) \right) \tag{2.49}\]

\[\mu(k) = \Pi^T \mu(k-1)\]

Here \( \mu_i \in [0, 1] \) denotes the current probability of being in mode \( i \) and \( \Pi \) is the Markovian transition matrix with its elements \( \pi_{ij} \in [0, 1] \). The IMM filter is a state estimator for switching linear systems. The algorithm works as follows:

1. Mode probability prediction:

\[\mu_j(k|k-1) = \sum_i \pi_{ij} \mu_i(k-1) \tag{2.50}\]

2. Mixing probability

\[\mu_{ij}(k|k-1) = \pi_{ij} \frac{\mu_i(k-1)}{\mu_j(k|k-1)} \tag{2.51}\]
3. Initialization of local Kalman filters:
\[
\hat{\xi}_0(j|k-1) = \sum_i \hat{\xi}_i(k-1)\mu_{ij}(k|k-1)
\] (2.52)
\[
P_0(j|k-1) = \sum_i \left( P_i(k-1) + \left[ \hat{\xi}(k-1) - \hat{\xi}_0(j|k-1) \right] \left[ \hat{\xi}(k-1) - \hat{\xi}_0(j|k-1) \right]^T \right) \mu_{ij}(k|k-1)
\] (2.53)

4. Kalman filter process prediction step (\(\forall j \in \{CV, CA\}\)):
\[
\hat{\xi}_j(k|k-1) = F_j \hat{\xi}_0(j|k-1)
\] (2.54)
\[
P_j(k|k-1) = F_j P_0(j|k-1) F_j^T + G_j Q_j G_j^T
\] (2.55)

5. Residuals (process prediction errors):
\[
r_j(k) = y(k) - H_j \hat{\xi}_j(k|k-1)
\] (2.56)
\[
S_j(k) = H_j P_j(k|k-1) H_j^T + R_j
\] (2.57)

6. Kalman gains:
\[
K_j(k) = P_j(k|k-1) H_j^T S_j(k)^{-1}
\] (2.58)

7. Kalman filter measurement steps:
\[
\hat{\xi}_j(k) = \hat{\xi}_j(k|k-1) + K_j(k) r_j(k)
\] (2.59)
\[
P_j(k) = P_j(k|k-1) - K_j(k) S_j(k) K_j(k)^T
\] (2.60)

8. Gaussian likelihood:
\[
\Lambda_j(k) = \frac{1}{\sqrt{2\pi S_j(k)}} e^{-\frac{r_j(k)^2}{2S_j(k)}}
\] (2.61)

9. Update mode probability:
\[
\mu_j(k) = \frac{1}{\sum_j \Lambda_j(k)\mu_j(k|k-1)} \Lambda_j(k)\mu_j(k|k-1)
\] (2.62)

10. Combination:
\[
\hat{\xi}(k) = \sum_j \hat{\xi}_j(k)\mu_j(k)
\] (2.63)
\[
P(k) = \sum_j \left( P_j(k) + [\hat{\xi}_j(k) - \hat{\xi}(k)] [\hat{\xi}_j(k) - \hat{\xi}(k)]^T \right) \mu_j(k)
\] (2.64)

11. Predict the \(\delta\) samples ahead output:
\[
\hat{y}(k+1+\delta|k-1) = \sum_j H_j F_j^\delta \hat{\xi}_0(j|k-1)\mu_j(k-1)
\] (2.65)

Before executing this algorithm the states have to be initialized: \(\mu(0) = \mu_0, \hat{\xi}_i(0) = \hat{\xi}_{i,0}\) and \(P_j(0) = P_{j,0}\).
2.2.5 Local Circular Model Kalman Filter

This filter is taken from [15]. The results are shown in Section 3.3.5. Here the respiratory motion is modeled as the projection of the motion of a model. The dynamics of the model generate a circular motion:

\[
\ddot{x} = -\Omega \dot{z} \\
\ddot{z} = \Omega \dot{x} \\
\dot{\Omega} = 0
\]

The model has to be discretized such that it can be used in a Kalman filter. We define the discrete-time state vector \( x = [x(k), \dot{x}(k), \dot{z}(k), \Omega(k)] \).

The extended Kalman filter, where the Jacobian of \( f \) as defined in (2.67) is needed:

\[
x(k + 1) = f(x(k)) + v(k) = \begin{bmatrix}
1 & \sin(\Omega(k)T_s) & -1 & -\cos(\Omega(k)T_s) \\
0 & \cos(\Omega(k)T_s) & 0 & \sin(\Omega(k)T_s) \\
0 & \sin(\Omega(k)T_s) & \cos(\Omega(k)T_s) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} x(k) + v(k)
\]

\[
y(k) = Hx(k) + w(k) = [1 \ 0 \ 0 \ 0] x(k) + w(k)
\]

Where \( v(k) \) is the zero mean process noise with variance \( Q \) and \( w(k) \) is the measurement noise with variance \( R \). \( Q \) is assumed to have the following form:

\[
Q = \begin{bmatrix}
\frac{1}{2} q_1 T_s^3 & \frac{1}{2} q_2 T_s^2 & 0 & 0 \\
\frac{1}{2} q_2 T_s & q_3 T_s & 0 & 0 \\
0 & 0 & q_2 T_s & 0 \\
0 & 0 & 0 & q_3 T_s
\end{bmatrix}
\]

Together with the noise the dynamics have a locally circular motion (LCM). The filter is an extended Kalman filter, where the Jacobian of \( f \) as defined in (2.67) is needed:

\[
f_x(k) = \frac{\partial f}{\partial x}igg|_{x=k(k)} = \begin{bmatrix}
1 & \sin(\Omega(k)T_s) & -1 & -\cos(\Omega(k)T_s) \\
0 & \cos(\Omega(k)T_s) & 0 & \sin(\Omega(k)T_s) \\
0 & \sin(\Omega(k)T_s) & \cos(\Omega(k)T_s) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
f_{1\Omega}(k) = \frac{1}{\Omega(k)^2} \left( \left( \Omega(k)T_s \cos(\Omega(k)T_s) - \sin(\Omega(k)T_s) \right) \dot{x}(k) \\
- \left( \Omega(k)T_s \sin(\Omega(k)T_s) - 1 + \cos(\Omega(k)T_s) \right) \dot{z}(k) \right)
\]

\[
f_{2\Omega}(k) = -T_s \sin(\Omega(k)T_s) \dot{x}(k) - T_s \cos(\Omega(k)T_s) \dot{z}(k)
\]

\[
f_{3\Omega}(k) = T_s \cos(\Omega(k)T_s) \dot{x}(k) - T_s \sin(\Omega(k)T_s) \dot{z}(k)
\]

With this the algorithm can be stated as follows:

1. Process prediction step:

\[
\hat{x}(k+1|k) = f(\hat{x}(k|k))
\]

\[
P(k+1|k) = f_x(k|k)P(k|k)f_x(k|k)^T + Q(k)
\]

2. Measurement step:

\[
S(k+1) = HP(k+1|k)H^T + R(k+1)
\]

\[
K(k+1) = P(k+1|k)H^T S(k+1)^{-1}
\]

\[
\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1) \left( y(k+1) - H\hat{x}(k+1|k) \right)
\]

\[
P(k+1|k+1) = P(k+1|k) - K(k+1)S(k+1)K(k+1)^T
\]
3. Predict the $\delta$ samples ahead output:
\[
\hat{y}(k + \delta | k) = \hat{x}(k | k) + \frac{\sin(\Omega_T \delta)}{\Omega(k | k)} \hat{x}(k | k) - \frac{1 - \cos(\Omega_T \delta)}{\Omega(k | k)} \hat{z}(k | k)
\]  
(2.80)

### 2.2.6 Fourier Kalman Filter

Since the respiration is somewhat periodic, one can approximate the signal using a time varying finite Fourier series as in [16]. The results are shown in Section 3.3.6.

\[
y(k) = c(k) + \sum_{i=1}^{n} r_i(k) \sin(\theta_i(k))
\]
(2.81)

\[
\theta_i(k) = i \sum_{l=0}^{k} \omega(l) T_s + \phi_i(k)
\]
(2.82)

We define the state vector
\[
x(k) = [c(k), r_i(k), \omega(k), \theta_i(k)]^T, i \in \{1, \ldots, n\}
\]

The model then is:

\[
x(k+1) = Fx(k) + v(k)
\]
(2.83)

\[
y(k) = h(x(k)) + w(k) = c(k) + \sum_{i=1}^{n} r_i(k) \sin(\theta_i(k)) + w(k)
\]
(2.84)

Where $v$ is a zero mean process with variance $Q$ and $w$ is the measurement noise with variance $R$. $E_{n+1}$ is the unit matrix. Since the output equation is nonlinear the extended Kalman filter is employed. The partial derivative of $h(x)$ with respect to $x$ is:

\[
H^T(x(k)) = \left. \frac{\partial h}{\partial x} \right|_{x(k)}^T = \begin{bmatrix}
1 & \sin \theta_1(k) & \vdots & r_1(k) \cos(\theta_1(k)) \\
0 & \sin \theta_2(k) & \vdots & r_2(k) \cos(\theta_2(k)) \\
& \vdots & \ddots & \vdots \\
0 & & & \sin \theta_n(k) \cos(\theta_n(k))
\end{bmatrix}
\]
(2.85)

With this the algorithm is stated:

1. Process prediction step:
   \[
x(k+1 | k) = Fx(k)
   \]
   (2.86)
   \[
P(k+1 | k) = FP(k | k)F^T + Q
   \]
   (2.87)

2. Measurement step ($E$ is the unit matrix with corresponding dimensions):
   \[
   S = H(k+1 | k)P(k+1 | k)H^T(k+1 | k) + R
   \]
   (2.88)
   \[
   K = P(k+1 | k)H^T S^{-1}
   \]
   (2.89)
   \[
   \hat{x}(k+1 | k+1) = \hat{x} + K \left( y(k) - h(\hat{x}(k+1 | k)) \right)
   \]
   (2.90)
   \[
   P(k+1 | k+1) = \left( E - KH \right) P(k+1 | k)
   \]
   (2.91)
3. Predict the $\delta$ samples ahead output:

$$\hat{x}_{\text{pred}} = F^\delta x(k+1|k+1) \tag{2.92}$$
$$\hat{y}(k+1 + \delta|k+1) = h(\hat{x}_{\text{pred}}) \tag{2.93}$$

### 2.3 Couch Control Theory

#### 2.3.1 No Controller

Since the input to the Protura system already consists of the coordinates of the platform and the internal controller computes the forces needed, it is not imperative to develop an external controller. Therefore the "No Controller" concept is just sending the reference signal (the tumor motion) through without modifications.

### 2.4 Hardware Implementation

In Figure 2.8 the hardware setup is shown. The sensors consist of two laser triangulation systems (Micro Epsilon Messtechnik GmbH & Co. KG, Ortenburg, Germany). In some cases one laser is used for patient respiration measurement and the other for couch (Protura) motion. Their placement though can be arbitrary and depend on the experiment. The lasers send an analog signal to the computer, in which there is a data acquisition card (MF624, Humusoft, Prague, Czech Republic), that converts the analog signals to digital signals.

The Protura (CIVCO Medical Solutions, Kalona, IA, USA) moves the patient. It is controlled via a serial interface over the D-Sub connector at the computer.

The Hexapod (H840.5PD, Physik Instrumente GmbH & Co. KG, Karlsruhe/Palmbach, Germany) emulates the respiration. It is also controlled via a serial interface, though the additional converter allows it to be connected to an USB port at the computer as opposed to the D-Sub connector for the Protura.

The software sending and receiving the signals is programmed via Matlab/Simulink and Realtime Windows Target (The MathWorks, Inc., Natick, MA, USA) ensures the real time operation of the software.

![Diagram](Image)

*Figure 2.8: This picture is taken from [22] and shows the hardware and type of connections used for measurements.*
2.5 Performance Indices

Before looking at the results, the performance indices need to be defined. The error of prediction is defined as follows:

\[ e_y(k) = \hat{y}(k|k-\delta) - y(k) \]  

(2.94)

Similarly the tracking error of the control system is defined as:

\[ e_{\text{ref}}(k) = y_{\text{ref}}(k) - y(k) \]  

(2.95)

To summarize the behaviour of the error over the time the simulation or experiment runs, two different formulas are used. First the root mean square (RMS) of the error:

\[ e_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} e(k)^2} \]  

(2.96)

And second the confidence interval as defined in [14]. Here the interval is searched for which the error is located inside with probability \( p_{\text{CI}} \). The probability distribution of the error is assumed to be gaussian with mean \( \mu \) and standard deviation \( \sigma \).

\[ p_{\text{CI}}(\mu - \sigma m < e(k) < \mu + \sigma m) = \frac{2}{\sqrt{\pi}} \int_{0}^{m/\sqrt{2}} \exp(-u^2)du = \text{erf} \left( \frac{m}{\sqrt{2}} \right) \]  

(2.97)

Where erf is called the error function. The confidence interval is defined to be:

\[ e_{\text{CI}} = |\mu| + m\sigma \]  

(2.98)

The variables \( \mu \) and \( \sigma \) can be computed by the given error data, \( \mu \) is simply the mean of the data, while \( \sigma \) is the standard deviation of the data. And \( m \) can be computed by using the inverse \( \text{erf}^{-1} \) of the error function.

\[ m = \sqrt{2} \text{erf}^{-1}(p_{\text{CI}}) \]  

(2.99)

Where \( p_{\text{CI}} \) is the demanded confidence ( \( p_{\text{CI}} \in [0, 1] \) ).
Chapter 3

Results

3.1 Respiration Characteristics

The respiration curves are examined with respect to some characteristics. First the mean amplitude (mean to peak) of each curve is determined by computing the half of the mean distance between the maximum peaks and the minimum peaks that are next to each other. Further the mean and the maximum of the absolute speed of each respiration curve is calculated. This is also done for the mean and maximum of the absolute acceleration. Then there is also the fundamental frequency of the respiration curve, which is the frequency with the highest amplitude and is located between the interval of 0.1 Hz and 5 Hz. The minimum, maximum and average values taken over all respiration curves are shown in Table 3.1. The complete values are shown in the Appendix B in Table B.1.

Table 3.1: The minimum, maximum and average value, taken over all respiration curves, of the characteristics.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Unit</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampl.</td>
<td>[mm]</td>
<td>1.05</td>
<td>6.64</td>
<td>2.76</td>
</tr>
<tr>
<td>Fund. Freq.</td>
<td>[Hz]</td>
<td>0.15</td>
<td>0.43</td>
<td>0.26</td>
</tr>
<tr>
<td>Mean Speed</td>
<td>[mm/s]</td>
<td>0.80</td>
<td>4.59</td>
<td>2.51</td>
</tr>
<tr>
<td>Max. Speed</td>
<td>[mm/s]</td>
<td>1.98</td>
<td>18.53</td>
<td>7.31</td>
</tr>
<tr>
<td>Mean Acc.</td>
<td>[mm/s²]</td>
<td>1.31</td>
<td>8.08</td>
<td>4.63</td>
</tr>
<tr>
<td>Max. Acc.</td>
<td>[mm/s²]</td>
<td>4.37</td>
<td>35.71</td>
<td>15.51</td>
</tr>
</tbody>
</table>

Additionally, consider the correlations between the characteristics in Table 3.2. The correlation matrix is \( R \) computed from the covariance matrix \( C \) of the data given by the respiration characteristics with \( R(i,j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}} \). The fundamental frequency has a small negative correlation with the amplitude and a small positive correlation with the mean acceleration. The amplitude has a high correlation with the mean speed and a bit smaller with the maximum speed and the mean and maximum accelerations. There are high correlations between the maximum speed and the maximum acceleration and between the mean speed and the mean acceleration.

Table 3.2: The correlations between the characteristics.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampl.</td>
<td>1.00</td>
<td>-0.39</td>
<td>0.88</td>
<td>0.70</td>
<td>0.61</td>
<td>0.70</td>
</tr>
<tr>
<td>Dom. Freq.</td>
<td>-0.39</td>
<td>1.00</td>
<td>-0.01</td>
<td>-0.12</td>
<td>0.36</td>
<td>-0.03</td>
</tr>
<tr>
<td>Mean Speed</td>
<td>0.88</td>
<td>-0.01</td>
<td>1.00</td>
<td>0.75</td>
<td>0.90</td>
<td>0.79</td>
</tr>
<tr>
<td>Max. Speed</td>
<td>0.70</td>
<td>-0.12</td>
<td>0.75</td>
<td>1.00</td>
<td>0.69</td>
<td>0.96</td>
</tr>
<tr>
<td>Mean Acc.</td>
<td>0.61</td>
<td>0.36</td>
<td>0.90</td>
<td>0.69</td>
<td>1.00</td>
<td>0.77</td>
</tr>
<tr>
<td>Max. Acc.</td>
<td>0.70</td>
<td>-0.03</td>
<td>0.79</td>
<td>0.96</td>
<td>0.77</td>
<td>1.00</td>
</tr>
</tbody>
</table>
3.2 Protura Characteristics

3.2.1 Protura without Extension

Speed and Acceleration Limitations

The Protura system has saturations in the positions, the speed and the acceleration. The position saturations are ignored in the model and it is assumed, that during tumor tracking those saturations are not reached. The speed and acceleration saturations in the real system are respected by the trajectory generator and in the model the saturations are also respected by the trajectory generator, but are not built into the mechanical model itself. The official number of the maximum speed is \( 16 \text{ mm/s} \). From measurements, using a chirp signal input going from 0.01 Hz to 0.4 Hz with an amplitude of 10 mm, the maximum speeds and accelerations have been computed. There are six measurements, two, where the platform moved only in \( z \)-direction, two, where the platform moved only in \( x \)-direction and two where the platform moved in two directions, \( x \) and \( z \). In each pair, one of the two measurements was carried out without any load and the other with a person on the platform acting as a load. The extension was removed. The results are shown in the first half of Table 3.3.

<table>
<thead>
<tr>
<th>Platform:</th>
<th>max. speed [mm/s]</th>
<th>max. acceleration [mm/s²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input signal</td>
<td>25.01</td>
<td>61.79</td>
</tr>
<tr>
<td>( z ) without load</td>
<td>20.33</td>
<td>65.20</td>
</tr>
<tr>
<td>( z ) with load (61 kg)</td>
<td>20.39</td>
<td>65.69</td>
</tr>
<tr>
<td>( x ) without load</td>
<td>21.42</td>
<td>56.48</td>
</tr>
<tr>
<td>( x ) with load (61 kg)</td>
<td>21.60</td>
<td>57.63</td>
</tr>
<tr>
<td>( x, z ) without load</td>
<td>16.06</td>
<td>46.19</td>
</tr>
<tr>
<td>( x, z ) with load (70 kg)</td>
<td>16.21</td>
<td>44.60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Base-leg joints:</th>
<th>max. speed [mm/s]</th>
<th>max. acceleration [mm/s²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z ) without load</td>
<td>20.85</td>
<td>65.39</td>
</tr>
<tr>
<td>( z ) with load (61 kg)</td>
<td>20.43</td>
<td>64.26</td>
</tr>
<tr>
<td>( x ) without load</td>
<td>21.42</td>
<td>56.48</td>
</tr>
<tr>
<td>( x ) with load (61 kg)</td>
<td>21.60</td>
<td>57.63</td>
</tr>
<tr>
<td>( x, z ) without load</td>
<td>19.87</td>
<td>59.75</td>
</tr>
<tr>
<td>( x, z ) with load (70 kg)</td>
<td>19.92</td>
<td>52.92</td>
</tr>
</tbody>
</table>

Since the maximum speeds and maximum accelerations of the platform do not seem to be the same in all cases, the movement of the platform was converted to the movement of the base-leg joints. Their movement was then analyzed for their speeds and accelerations and the results are shown in the second half of Table 3.3. The values are closer to each other than the values for the platform. In the following results of simulations, the maximum speeds of the base-leg joints in the simulation are set to 20 mm/s and the maximum acceleration to 60 mm/s².

Dynamic Input Output Behaviour

The Protura followed a chirp signal reference for the \( x \) and \( z \)-direction. The \( y \)-direction and the rotations received zero as reference. A chirp signal is a sinusoidal signal, whose frequency increases linearly with time. In this case the frequency goes from 0.01 Hz to 1 Hz in 180 s and has an amplitude of 10 mm. The \( z \) and \( y \)-direction were measured and the Fourier transformations of the resulting signals and the input were taken. The ratio of the Fourier transformations of the measurement \( Y(\omega) \) and the input \( U(\omega) \) gives the input-output behaviour \( H(\omega) \) of the system.
Splitting the resulting complex values into magnitude and phase and plotting them depending on \( \omega \) gives us the Bode plots.

\[
H(\omega) = \frac{Y(\omega)}{U(\omega)}
\]  

(3.1)

The resulting plots are shown in Figures 3.1 and 3.2. The magnitude gives us the gain at certain frequencies from the input to the output and the phase tells us how far the output lags behind the input. For the plotting the logarithm of \( H(\omega) \) is taken: \( 20 \cdot \log_{10}(|H(\omega)|) \). In the Figure 3.1 at low frequencies up to about 0.2 Hz the magnitude is about 0 dB, so the Protura system follows the input fully. With increasing frequency the magnitude and the phase decrease. So the Protura system cannot move the full amplitude at higher frequencies and lags behind as well. The model of the Protura exhibits similar behaviour, the magnitudes are close up to 0.4 Hz and the phase deviates earlier at about 0.2 Hz. However, at higher frequencies the phases and magnitudes stay close.

Figure 3.1: A Bode plot from chirp reference input to \( z \)-motion of the Protura, when only \( z \) and \( x \) receive a chirp reference input.

In Figure 3.2, the \( y \)-direction should actually not move at all (magnitude should go to \( -\infty \)), but does. Here the input-output behaviour between the reference input to \( x \) and \( z \) and the motion of \( y \) is considered. At lower frequencies the behaviour has a very small magnitude, but with higher frequencies it increases and peaks at about 0.4 Hz. The results of the simulation are not as consistent with the measurement as in the case of Figure 3.1. At lower frequencies the simulation has a smaller magnitude and at 0.2 Hz the magnitudes are close to each other, but the simulation peaks earlier.

Figure 3.2: A magnitude plot from chirp reference input to \( y \) motion of the Protura, when only \( z \) and \( x \) receive a chirp reference input.
3.2.2 Protura with Extension

Influence of Extension

We start with a single step of the platform, which moves the platform 20 mm in positive $z$-direction. This step is repeated several times and each time the distance sensors measured another location. The locations are positioned along the $x$-axis of the platform (see Figure 3.3).

There are twelve measurement locations distributed with the distances $a = 200$ mm and $b = 100$ mm between each other and they all lie on the same straight. Since there are only two distance measurement sensors available, the step motion of the platform had to be repeated and each time the sensors were set to new measurement locations. Therefore, before each step motion of the platform, the sensors were calibrated. The data is plotted in Figure 3.4 for a few time instances.

The motion starts at two seconds, the lowest line in Figure 3.4. At this moment the platform is still in zero position. After 0.3 seconds, the motion has started and there is a height difference of about 2 mm from the first to the twelfth location. The inclination stays until the motion has completed at 3.5 s. The points do not make up two straights (platform and extension) in a perfect manner, this is attributed to errors in the positioning of the sensors. To find the actual slopes of the platform, the extension and all measurement locations together three straights have been fitted to the data, one straight using only the measurement points of the platform, one straight using only the measurement points of the extension plate and one straight over all measurement points. The behaviour of the slopes is shown in Figure 3.5.

Before the step the slopes of the fitted straights are zero. The slope of the extension plate decreases for nearly half a second, after which it seems to oscillate until the slopes go back to zero. This corresponds to the acceleration and constant speed intervals of the step motion as shown in Figure 3.6.
Stiffness of Extension

To find the stiffness of the extension, a load was placed on the extension at different locations, that vary in their distances to the joint between the platform and the extension. The height of the extension was measured at two locations, once at the joint position (a) and once at the other end of the extension (b), as shown in Figure 3.7.

The stiffness is assumed to be in a rotational spring acting on the joint between the platform and the extension, which are considered to be rigid bodies. By changing the location of the constant load, the torque around the joint is varied and the resulting angles are determined by measuring the vertical dislocation of the points a and b. The Protura did not move during measurement. Since the load and its positions ($x_d$) are known, the resulting torques are also known. The following equation is assumed to hold:

$$M_{load} = c_{R} \phi + c_{\text{residual}}$$  \hspace{1cm} (3.2)

where

$$M_{load} = m_{load} g x_d$$  \hspace{1cm} (3.3)

The expression (3.3) does not include the mass of the extension itself, because the mass already influences the calibration phase of the measurement. This means, that an actual nonzero angle
3.2. Protura Characteristics

of the extension is measured as zero. Due to friction forces the zero angle during the calibration process may not be equal to the zero angle after the calibration, because they are reached from different directions. Therefore, \( c_{\text{residual}} \) is introduced to catch any offsets. Since this equation depends linearly on its parameters, the least squares method is applied. The results are shown in Table 3.4.

Table 3.4: Rotational spring stiffness parameters determined by least squares

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_R ) [Nm/(^\circ)]</td>
<td>975</td>
</tr>
<tr>
<td>( c_{\text{residual}} ) [Nm]</td>
<td>6.44</td>
</tr>
</tbody>
</table>

Dynamic Input Output Behaviour of Extension

In this experiment a chirp signal was sent to the Protura system to move in the z-direction. The frequency of the chirp signal went from 0.01 Hz to 1 Hz. The configuration was the same as in Figure 3.7 but without the load. From the two measurement signals the angle of the extension was computed and then considered to be the output of the Protura system depending on the chirp input.

![Figure 3.8: A magnitude plot of the angle motion of the extension of the Protura system, when z receives a chirp reference input](image)

In Figure 3.8 the magnitude for lower frequencies up to 0.3 Hz is small, so there is little motion. But at higher frequencies the magnitude increases and peaks between 0.5 Hz and 0.6 Hz and decreases afterwards. The simulation also has small magnitude until 0.3 Hz, then it starts to increase. But it peaks before 0.5 Hz and stays smaller than the measurement values. This result is the closest achieved while varying the spring and damper values for the extension. The values then are 17.45 [Nm/\(^\circ\)] and 0.17 [N.m.s/\(^\circ\)]. So the spring stiffness found here is not the same as shown in Table 3.4.
3.2.3 Parameter Check

The results of simulations shown above are done while fitting the parameters of the Protura system. Therefore, other data is needed to check whether the determined parameters actually fit. The data is originally used in [22] and consists of respiration curves and conversion factors between the respiration signal and the tumor motion signal. The measurement sensors were placed on the platform of the Protura system, so there is no influence of the extension plate. Additionally, the y-direction of motion was not measured and is assumed to be zero. The respiration curves used in those measurements correspond to the signals 11 to 19 as shown in the Appendix B. Figure 3.9 shows the $\varepsilon_{\text{RMS}}$ of the differences between the simulation and the measurements, once in $x$ and once in $z$-direction.

![Figure 3.9](image.png)

Figure 3.9: The $\varepsilon_{\text{RMS}}$ of the error between the measurement and the simulation. The horizontal axis numbers correspond to the respiration curves shown in Appendix B. 16 appears twice, because, for this patient, there were two tumors with different conversion factors.

The average of the $\varepsilon_{\text{RMS}}$ over these measurements is 0.57 mm with standard deviation 0.35 mm in $x$-direction and in $z$-direction the average is 0.5 mm with a standard deviation of 0.77 mm.
3.3 Prediction Filter Simulations

3.3.1 Results of nLMS Filter

The nLMS filter was introduced in Section 2.2.1. There are several parameters influencing the filter performance:

- \( n \) is the number of data points used to compute the prediction (or the length of the weight vector)
- \( \delta \) is the number of steps the predicted value is ahead
- \( \mu_{\text{step}} \) is the step size for minimizing the expected error

The results are shown in Table 3.5. The left part shows the parameter range in which the parameters were varied with a certain stepsize. In the middle the parameter set, which minimizes both the \( e_{\text{RMS}} \) and the \( e_{\text{CI}} \). The achieved values are given on the right and for comparison the maximum values are given, too.

Table 3.5: Parameter-sweep range, the found minimizer and the minimum and maximum \( e_{\text{RMS}} \), \( e_{\text{CI}} \) values. All simulations were done with \( T_s = 0.05 \text{s} \) and a prediction time of 0.1 s

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer</th>
<th>( e_{\text{RMS}} ) [mm]</th>
<th>( e_{\text{CI}} ) [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>1 -- 10</td>
<td>1</td>
<td>( n )</td>
<td>2</td>
<td>min 0.12</td>
</tr>
<tr>
<td>( \mu_{\text{step}} )</td>
<td>0.05 -- 0.4</td>
<td>0.05</td>
<td>( \mu_{\text{step}} )</td>
<td>0.2</td>
<td>max 21.85</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.5 -- 1.0</td>
<td>0.05</td>
<td>( \alpha )</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

Another interesting aspect is the error dependent on the prediction time, but for that \( n \) and \( \mu_{\text{step}} \) need to be fixed. For this, the minimizer parameter set from Table 3.5 is used.

Figure 3.10: The black line with circle markers shows the mean \( e_{\text{RMS}} \) over all the respiration curves using the minimizer for the nLMS prediction filter from Table 3.5 and a sampling time \( T_s = 0.05 \text{s} \). The red lines with square markers show the standard deviation.

In Figures 3.10 and 3.11 the mean \( e_{\text{RMS}} (\mu_{e_{\text{RMS}}}) \) and the standard deviations (\( \sigma_{e_{\text{RMS}}} \)) over all respiration curves are plotted. The further the prediction is ahead, the higher is the error. The mean \( e_{\text{RMS}} \) and the standard deviation increase with higher prediction times. Comparing the two Figures, up to \( \delta \cdot T_s = 0.15 \text{s} \) the values of Figure 3.10 are smaller than the values of Figure 3.11. At higher prediction times the situation reverses.
0.05 0.1 0.15 0.2 0.25 0.3
0
0.5
1
1.5
δ ⋅ Ts [s]
eRMS [mm]
µ e,RMS
µ e,RMS ± σ e,RMS

Figure 3.11: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the nLMS prediction filter from Table 3.5 and a sampling time $T_s = 0.1$ s. The red lines with square markers show the standard deviation.

3.3.2 Results of MULIN Filter

The MULIN filter is introduced in Section 2.2.2. There are several parameters affecting the performance of the MULIN filter. The different MULIN filters, MULIN0, MULIN1, MULIN2, MULIN3 differ in the order of the approximation of the difference function. The other parameters are:

- $\alpha$ is the smoothing factor
- $l$ is the number of steps past values, which are used, are behind (only MULIN1, MULIN2, MULIN3)

In Table 3.6, the parameter variation range of the parameter sweep and its results are shown. For MULIN0 there is no $l$ parameter. The best results are achieved with MULIN1 and MULIN2 with the corresponding minimizing parameter set.

Table 3.6: Parametersweep range, the found minimizer and the minimum and maximum $e_{\text{RMS}}, e_{\text{CI}}$ values for MULIN filters. All simulations were done with $T_s = 0.05$ s and a prediction time of $0.1$ s

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer for</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$0.1 - 1$</td>
<td>0.1</td>
<td>MULIN0</td>
</tr>
<tr>
<td>$l$</td>
<td>$1 - 10$</td>
<td>1</td>
<td>MULIN1</td>
</tr>
<tr>
<td>$l$</td>
<td>$1 - 10$</td>
<td>1</td>
<td>MULIN2</td>
</tr>
<tr>
<td>$l$</td>
<td>$1 - 10$</td>
<td>1</td>
<td>MULIN3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors</th>
<th>$e_{\text{RMS}}$</th>
<th>$e_{\text{CI}}$</th>
<th>$e_{\text{RMS}}$</th>
<th>$e_{\text{CI}}$</th>
<th>$e_{\text{RMS}}$</th>
<th>$e_{\text{CI}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>min [mm]</td>
<td>0.11</td>
<td>0.22</td>
<td>0.06</td>
<td>0.12</td>
<td>0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>max [mm]</td>
<td>1.41</td>
<td>2.76</td>
<td>1.4</td>
<td>2.74</td>
<td>1.39</td>
<td>2.73</td>
</tr>
</tbody>
</table>

With the given minimizer, the prediction times are varied between $0.05$ s and $0.3$ s for MULIN1 and MULIN2 and for two sampling times $T_s = 0.05$ s and $T_s = 0.1$ s.

The $\mu_{e,\text{RMS}}$ of MULIN1 increases monotonically together with an also monotonically increasing standard deviation for both sampling times. The $\mu_{e,\text{RMS}}$ with a higher sampling time is not higher than with a smaller sampling time, though the standard deviation is slightly larger. The MULIN2 prediction filter has an initially slightly decreasing standard deviation but with higher prediction times it also increases monotonically. The $\mu_{e,\text{RMS}}$ increases monotonically in both cases.
Figure 3.12: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the MULIN1 prediction filter from Table 3.6 and a sampling time $T_s = 0.05$ s. The red lines with square markers show the standard deviation.

Figure 3.13: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the MULIN1 prediction filter from Table 3.6 and a sampling time $T_s = 0.1$ s. The red lines with square markers show the standard deviation.

Figure 3.14: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the MULIN2 prediction filter from Table 3.6 and a sampling time $T_s = 0.05$ s. The red lines with square markers show the standard deviation.
Figure 3.15: The black line with circle markers shows the mean $\hat{e}_{\text{RMS}}$ over all the respiration curves using the minimizer for the MULIN2 prediction filter from Table 3.6 and a sampling time $T_s = 0.1 \, \text{s}$. The red lines with square markers show the standard deviation.

### 3.3.3 Results of SVR Filter

This filter is presented in Section 2.2.3. The changeable parameters are:

- $\epsilon$ is the tolerance to deviation, in which the cost is not influenced
- $C$ is the weight of the cost due to deviation from the searched function
- $n$ is the number of data points considered in the optimization problem
- several different kernel functions are available with their own parameters, at most two ($\beta_1$, $\beta_2$)

Concerning the kernel functions only the radial basis function (RBF) is used. In the library by [20] five kernel functions are available, which are linear, polynomial, RBF, gaussian RBF, exponential RBF and multilayer perceptron (MLP). The linear kernel fits a hyperplane to the data, which gives bad predictions especially when the respiratory cycle is at the maximum or minimum. The polynomial kernel function is numerically unstable. The gaussian RBF is practically the same as the RBF. The MLP kernel is a switching function and does not fit a smooth curve.

The RBF has one kernel function parameter $\beta_1$, so together with $n$, there are four parameters. These parameters were varied in certain ranges and for each combination the averaged $e_{\text{RMS}}$ and $e_{\text{CI}}$ were determined. The results are shown in Table 3.7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer</th>
<th>$e_{\text{RMS}}$ [mm]</th>
<th>$e_{\text{CI}}$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>0.01 – 0.09</td>
<td>0.01</td>
<td>$\epsilon$ 0.01</td>
<td>min 0.08</td>
<td>min 0.16</td>
</tr>
<tr>
<td>$C$</td>
<td>80 – 180</td>
<td>50</td>
<td>$C$ 180</td>
<td>max 0.31</td>
<td>max 0.67</td>
</tr>
<tr>
<td>$n$</td>
<td>2 – 10</td>
<td>2</td>
<td>$n$ 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>2 – 10</td>
<td>2</td>
<td>$\beta_1$ 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the searched parameter range the differences the maximum and minimum of the $e_{\text{RMS}}$ is very small. This is also the case for the $e_{\text{CI}}$. The minimizers differ only by the sampling time $T_s$. For $T_s = 0.05 \, \text{s}$ only four data points are needed while $T_s = 0.1 \, \text{s}$ uses five.

Using the minimizer for $e_{\text{RMS}}$, the prediction time was varied between 0.05s and 0.3s once with $T_s = 0.05 \, \text{s}$ and once with $T_s = 0.1 \, \text{s}$. In Figures 3.16 and 3.17 the results are shown for each respiration curve. In both cases the $\mu_{e_{\text{RMS}}}$ increases monotonically and also in both cases the standard deviation increases monotonically.
Figure 3.16: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the SVR prediction filter from Table 3.7 and a sampling time $T_s = 0.05$ s. The red lines with square markers show the standard deviation.

Figure 3.17: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the SVR prediction filter from Table 3.7 and a sampling time $T_s = 0.1$ s. The red lines with square markers show the standard deviation.
3.3.4 Results of IMM Kalman Filter

This filter is presented in Section 2.2.4. Its parameters are:

- process noise variances $Q_i$, $i \in \{CV, CA\}$
- measurement noise variances $R_i$, $i \in \{CV, CA\}$
- Markovian transition matrix $\Pi$

The process noise variances constitute two parameters, the measurement noise variances are also two parameters and the markovian transition matrix another two, which are the entries on the diagonals ($\pi_{CV,CV}, \pi_{CA,CA}$). The non-diagonal entries can be computed from the given parameters, since the row sum of $\Pi$ has to be one. So there are six parameters in total.

These parameters were varied in certain ranges as shown in Table 3.8 and the minimizer of the parameter set is used for further simulations varying the prediction time.

Table 3.8: Parametersweep range, the found minimizer and the minimum and maximum $e_{RMS}$, $e_{CI}$ values. All simulations were done with $T_s = 0.05$ s and a prediction time of 0.1 s

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer of</th>
<th>$e_{RMS}$</th>
<th>$e_{CI}$</th>
<th>$e_{RMS}$ [mm]</th>
<th>$e_{CI}$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_{CV,CV}$</td>
<td>0.1 – 0.9</td>
<td>0.1</td>
<td>$\pi_{CV,CV}$</td>
<td>0.9</td>
<td>0.8</td>
<td>0.11 – 15.24</td>
<td>0.22 – 29.49</td>
</tr>
<tr>
<td>$\pi_{CA,CA}$</td>
<td>0.1 – 0.9</td>
<td>0.1</td>
<td>$\pi_{CA,CA}$</td>
<td>0.9</td>
<td>0.7</td>
<td>0.11 – 15.24</td>
<td>0.22 – 29.49</td>
</tr>
<tr>
<td>$Q_{CV}$</td>
<td>4000 – 18000</td>
<td>2000</td>
<td>$Q_{CV}$</td>
<td>18000</td>
<td>18000</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>$Q_{CA}$</td>
<td>4000 – 18000</td>
<td>2000</td>
<td>$Q_{CA}$</td>
<td>8000</td>
<td>8000</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>$R_{CV}$</td>
<td>0.05 – 0.2</td>
<td>0.05</td>
<td>$R_{CV}$</td>
<td>0.1</td>
<td>0.05</td>
<td>0.11 – 15.24</td>
<td>0.22 – 29.49</td>
</tr>
<tr>
<td>$R_{CA}$</td>
<td>0.05 – 0.2</td>
<td>0.05</td>
<td>$R_{CA}$</td>
<td>0.05</td>
<td>0.05</td>
<td>min</td>
<td>max</td>
</tr>
</tbody>
</table>

Using the minimizer for $e_{RMS}$ the prediction time is varied between 0.05 s and 0.3 s once with a sampling time $T_s$ of 0.05 s in Figure 3.18 and once with 0.1 s in Figure 3.19.

![Figure 3.18](image.png)

Figure 3.18: The black line with circle markers shows the mean $e_{RMS}$ over all the respiration curves using the minimizer for the IMM prediction filter from Table 3.8 and a sampling time $T_s = 0.05$ s. The red lines with square markers show the standard deviation.

For both sampling times the $\mu_{e,RMS}$ increases monotonically with the prediction time, but with $T_s = 0.1$ s the $e_{RMS}$ values are higher than with $T_s = 0.05$ s. This also holds for the standard deviations.
3.3. Prediction Filter Simulations

Figure 3.19: The black line with circle markers shows the mean $e_{RMS}$ over all the respiration curves using the minimizer for the IMM prediction filter from Table 3.8 and a sampling time $T_s = 0.1\, s$. The red lines with square markers show the standard deviation.

3.3.5 Results of LCM Kalman Filter

This filter is presented in Section 2.2.5. The parameters used here are

- $Q$, the process noise variance matrix, consists of three parameters $q_1$, $q_2$, $q_3$
- $R$, the measurement noise variance

The parameters of the filter were varied over certain ranges and steps as shown in Table 3.9. Also in Table 3.9, the average $e_{RMS}$ minimizer of the parameter set is shown. The simulations of the parameter sweep were carried out with $T_s = 0.05\, s$ and a prediction time of $0.1\, s$.

Table 3.9: Parametersweep range, the found minimizer and the minimum and maximum $e_{RMS}$, $e_{CI}$ values. All simulations were done with $T_s = 0.05\, s$ and a prediction time of $0.1\, s$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer</th>
<th>$e_{RMS}$ [mm]</th>
<th>$e_{CI}$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_1$</td>
<td>$100 - 1000$</td>
<td>100</td>
<td>$q_1$ 200</td>
<td>$q_2$ 5 min</td>
<td>$e_{RMS}$ 0.06 min $e_{CI}$ 0.12 max 3.04</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$5 - 15$</td>
<td>100</td>
<td>$q_2$ 5</td>
<td>$q_3$ 2000 max</td>
<td>$e_{RMS}$ 1.87 max $e_{CI}$ 3.04 min 0.4</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$200 - 2000$</td>
<td>200</td>
<td>$q_3$ 2000</td>
<td>$q_4$ 2000 max</td>
<td>$e_{RMS}$ 1.87 max $e_{CI}$ 3.04 min 0.4</td>
</tr>
<tr>
<td>$R$</td>
<td>$0.1 - 1.0$</td>
<td>0.1</td>
<td>$R$ 0.4</td>
<td>$q_5$ 2000 max</td>
<td>$e_{RMS}$ 1.87 max $e_{CI}$ 3.04 min 0.4</td>
</tr>
</tbody>
</table>

Using the minimizer of the parameter sweep, several simulation were run with varying prediction times, once with $T_s = 0.05\, s$ and once with $T_s = 0.1\, s$. The results are shown in Figures 3.20 and 3.21. For both sampling times, the errors increase monotonically with the prediction time. The standard deviations increase in both cases, too. Comparing the Figures 3.20 and 3.21 shows that the latter case has a bit higher errors.
Figure 3.20: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the LCM prediction filter from Table 3.9 and a sampling time $T_s = 0.05s$. The red lines with square markers show the standard deviation.

Figure 3.21: The black line with circle markers shows the mean $e_{\text{RMS}}$ over all the respiration curves using the minimizer for the LCM prediction filter from Table 3.9 and a sampling time $T_s = 0.1s$. The red lines with square markers show the standard deviation.
3.3.6 Results of Fourier Kalman Filter

This filter is presented in Section 2.2.6. One parameter influences the length of the sum and therefore the total number of parameters. To avoid this, the parameters, that have the same place in each summand, are pooled.

- \( m \) is the length of the Fourier sum

- \( q_c, q_r, q_\omega, q_\theta \) are the parameters for the process variance matrix \( Q \), in which they are placed on the diagonal like this: \[
\begin{bmatrix}
q_c & q_r & \cdots & q_r & q_\omega & \cdots & q_\omega & q_\theta \\
q_r & q_c & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots \\
q_\omega & \cdots & \cdots & q_\omega & q_\theta & \cdots & \cdots & \cdots \\
q_\theta & \cdots & \cdots & \cdots & \cdots & q_\theta & \cdots & \cdots \\
\end{bmatrix},
\]

- \( R \) is the measurement noise variance

A parametersweep in a certain region of the parameter space was carried out. From this, the minimizer is determined. The results are shown in Table 3.10. With these values the prediction time is varied once for \( T_s = 0.05 \) s and once for \( T_s = 0.1 \) s.

Table 3.10: Parametersweep range, the found minimizer and the minimum and maximum \( e_{\text{RMS}} \), \( e_{\text{CI}} \) values. All simulations were done with \( T_s = 0.05 \) s and a prediction time of 0.1 s

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step</th>
<th>Minimizer of</th>
<th>( e_{\text{RMS}} )</th>
<th>( e_{\text{CI}} )</th>
<th>( e_{\text{RMS}} ) [mm]</th>
<th>( e_{\text{CI}} ) [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>5−10</td>
<td>1</td>
<td>( m )</td>
<td>6</td>
<td>7</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>( q_c )</td>
<td>0.1−0.5</td>
<td>0.1</td>
<td>( q_c )</td>
<td>0.4</td>
<td>0.1</td>
<td>min 0.11</td>
<td>min 0.20</td>
</tr>
<tr>
<td>( q_r )</td>
<td>1.5−2</td>
<td>0.1</td>
<td>( q_r )</td>
<td>1.8</td>
<td>1.5</td>
<td>max 0.16</td>
<td>max 0.30</td>
</tr>
<tr>
<td>( q_\omega )</td>
<td>2−2.5</td>
<td>0.1</td>
<td>( q_\omega )</td>
<td>2.5</td>
<td>2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q_\theta )</td>
<td>0.1−0.5</td>
<td>0.1</td>
<td>( q_\theta )</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( R )</td>
<td>0.01−0.05</td>
<td>0.01</td>
<td>( R )</td>
<td>0.05</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The \( \mu_{e,RMS} \) increases monotonically for both sampling times. The standard deviations also increase monotonically with prediction time.

![Figure 3.22: The black line with circle markers shows the mean \( e_{\text{RMS}} \) over all the respiration curves using the minimizer for the Fourier prediction filter from Table 3.10 and a sampling time \( T_s = 0.05 \) s. The red lines with square markers show the standard deviation.](image-url)
Figure 3.23: The black line with circle markers shows the mean $e_{RMS}$ over all the respiration curves using the minimizer for the Fourier prediction filter from Table 3.10 and a sampling time $T_s = 0.1$ s. The red lines with square markers show the standard deviation.

### 3.3.7 Comparison of Prediction Filter Simulations

Here the errors resulting from the simulations of the different prediction filters are compared. In Figures 3.24 and 3.25 the simulations with the minimizing parameters found in the sections above and a prediction time of $0.3$ s are compared by looking at the $e_{RMS}$ averaged over all respiration curves. Additionally the standard deviations of the $e_{RMS}$ are shown.

Figure 3.24: The grey bars show the averages of the $e_{RMS}$ of the prediction filters over all $19$ respiration curves for a prediction time of $0.3$ s and $T_s = 0.05$ s. The red bars show the standard deviation. Note the difference in scale for the nLMS filter compared to the other filters.

The results for the nLMS filter show values nearly a magnitude higher, therefore its results are shown on a separate scale. This allows a better differentiation between the results of the other filters, while keeping the results complete. The smallest average $e_{RMS}$ is reached by the LCM filter in the case of $T_s = 0.05$ s and by the MULIN filter in the case of $T_s = 0.1$ s. In second place is the SVR filter for both sampling times. It also has the smallest standard deviation. With exception of the nLMS and the MULIN filters all filters have higher $e_{RMS}$ with $T_s = 0.1$ s than with $T_s = 0.05$ s.
though the differences vary. The difference for the SVR filter is very small, while the difference for the IMM filter is very high.

Another aspect is the computation time. Each prediction filter has a different amount of computations to do per step. The values are shown in Figure 3.26. They are from simulations with pure Matlab code. The SVR prediction filter is the slowest, while the nLMS and the MULIN prediction filters are the fastest. The IMM, the LCM and the Fourier prediction filters are slightly slower than the nLMS and the MULIN filters.
3.4 Combined Simulations and Experiments

At this point we are interested in the behaviour of the complete system. We look at simulations carried out without any delay, with a delay and with a delay together with a prediction filter. The LCM prediction filter was chosen due to its good results. Since only for respiration curves 11 to 19 respiration to tumor conversion values were known, only these respiration curves were used in this Section. For the simulations the center of the platform of the Protura system is the point of interest.

The simulation results are shown in Figure 3.27. The bars show the \( e_{\text{RMS}} \) of the norm of the tracking error vector over the first 100 s of the respiration curves. The white bars show the \( e_{\text{RMS}} \) when no tracking is used. The black bars show the tracking error in the ideal case without any delay introduced in the system. The middle red bars show the tracking error with a delay 0.1 s introduced by the sensors. On average the increase from tracking without delay to tracking with delay is 0.54 mm and all respiration curves are increased compared to the left black bars. The right grey bars show the tracking error with a delay of and the LCM prediction filter. The values decrease and reach the level of the system without delay. On average the decrease from tracking with delay to tracking with prediction is 0.53 mm.

In Figure 3.28 the corresponding measurement results are shown. Here also the \( e_{\text{RMS}} \) of the norm of the tracking error vector over the first 100 s is taken. But since the \( y \)-direction was not measured the error in \( y \)-direction is assumed to be zero. The left black bars show the real case and the right red bars show the corresponding simulated results. The differences are small though the simulations usually reach lower values.

Figure 3.27: The \( e_{\text{RMS}} \) of the norm of the tracking error vector in simulation. The delay is set to 0.1 s and the prediction time is set to the same value. The sampling time is 0.05 s. The prediction filter used the LCM filter. 16 appears twice, because this patient had two tumors with distinct conversion factors.

Figure 3.28: The \( e_{\text{RMS}} \) of the norm of the tracking error vector with a no delay system and for comparison the corresponding simulation results. The measurement results are from experiment carried out for [22]. 16 appears twice, because this patient had two tumors with distinct conversion factors.
Chapter 4

Discussion and Outlook

4.1 Respiration Characteristics

The average value of the amplitudes of the respiration curves used in this thesis is 2.76 mm as shown in Table 3.1. In [23] and [24] the mean amplitude is 5 mm. So the respiration curves here tend to have small amplitudes. The difference can be explained by differences in placement of the reference points of the respiratory motion measurement. The average fundamental frequency here is 0.26 Hz while in [23] it is 0.28 Hz and in [24] it is 0.26 Hz. These values are quite close and should therefore be able to represent the typical respiratory frequency of the population.

Comparing the values of speed and acceleration of the respiration characteristics and the Protura (Table 3.3) shows that the Protura should be able to follow the breathing signal. Though this does not have to hold for the tumor tracking, since the amplitude here may be larger than the respiratory amplitude. But looking at the fundamental frequency of the respiration curves and the Bode plot (Figure 3.1) shows that the dynamics of the Protura are too slow. At the average fundamental frequency there is already a nonzero phase and the magnitude is below 0 dB, so the Protura cannot do respiration tracking perfectly.

4.2 Protura Performance

The Protura system is a parallel manipulator which allows very good static positioning of patients. On the other hand dynamic tracking does not work as well. The Bode plot in Figure 3.1 shows that until about 0.2 Hz the Protura can do reference tracking well. Afterwards the magnitude and the phase decrease, so the Protura will not follow the complete range of motion and will also lag behind. This is a common limitation and is usually due to inertia, although here the saturations in speed and acceleration also play a part. This can be seen in the behaviour of the phase since it decreases too much for either a linear first order element or a second order element (For their definitions and Bode plot behaviours see [25]).

Another problem of the Protura is indicated in Figure 3.2. When the reference position is nonzero in \(x\) and \(z\)-direction and zero in \(y\)-direction, it still results in unwanted motion in the \(y\)-direction. This phenomenon results from the particular setup of the legs of the system as shown in Figure 2.1. The legs 1 and 4, oriented in \(y\)-direction, cause a force in \(y\)-direction on the platform even when the platform should only move in \(z\)-direction. The internal controller of the real Protura system does not seem to be able to completely compensate for this. This unwanted motion mainly increases the errors during tracking but could also negatively impact the comfort of the patient. Though, whether this will happen during tracking depends on the current respiratory behaviour, since the unwanted motion starts becoming relevant only at frequencies higher than 0.2 Hz.

Another part of the Protura system is the extension plate attached to one of the short ends of the platform. Figures 3.4 and 3.5 show that the platform causes torque around the \(y\)-axis of the platform. If there is also a patient placed on the extension the torque may be even greater and the
slopes of the platform and the extension as well. The controller can compensate the torque very well, when there is no motion needed, but during motion it cannot. During tracking this needs to be considered. Additionally the Figure 3.8 shows that at higher frequencies the extension plate starts oscillating, which may increase the error during tracking and decrease the patient comfort. In the course of this thesis the Protura system has been modeled using first principles. The internal controller though is only an approximation and has been tuned to get the model behaviour as close to the real system behaviour as possible. The Bode plot in Figure 3.1 shows that the model starts decreasing the magnitude at the same frequency as the real system. The slopes of the curves after the start of the decrease though are not quite the same. The phase of the model starts decreasing earlier and is practically always smaller than the phase of the real system. This may be attributed to the internal controller behaviour of the model since it is working at a smaller sampling frequency (to save simulation time) than the real system. The unwanted motion problem described above also happens in the model, but at lower frequencies and not quite as pronounced as in the real system. This can also be attributed to the internal controller. The extension plate was modeled as a rigid body and the joint between the platform and the extension with a linear rotational spring and a linear rotational damper. The spring damper was estimated in the static case (Table 3.4), but the values found for the closest magnitude behaviour (Figure 3.8) of the extension were completely different. But even with these values the match is not very good. Therefore the joint needs a closer look and a different model of the relationship between the angle of the extension and forces at the joint.

Still, Figure 3.9 shows a good fit of the parameters to the real system, since the differences of the positions between the model and the real system while tracking are small.

4.3 Prediction Filters

The Figures 3.24 and 3.25 show that the performances of the prediction filters are rather similar, except for the nLMS prediction filter. The nLMS prediction filter does not seem to be robust against prediction time increase, though with a higher sampling time the predictions improve. Since for both sampling times the same parameter values were used, the nLMS uses values that are further in the past in the case of larger sampling time. Therefore it seems that for a look further into the future the nLMS needs to see further into the past. Generally the predictions are worse with higher prediction times, which is expected, since there is more time in which the signal can change arbitrarily. Additionally, with higher prediction times the performance of the prediction filters depend more on respiration behaviour, which is expressed in the increasing standard deviation shown in the corresponding figures in Section 3.3.

This leads to the question how the respiration curves influence the performance of the prediction filters. Or formulated differently: What makes a respiratory motion predictable? This question remains for further work.

An important aspect of the prediction filters is their computational cost, especially in realtime operation there is a hard limit on how much time each step can take. The only relevant prediction filter here is the SVR filter, since it takes nearly 20 ms to compute. The SVR filter is essentially an optimization problem solver and each time new data is added it has to iterate until it converges to a new solution. This may take very long. The other prediction filters do each iteration of the convergence at each timestep. And the MULIN filter does not even have an internal state, so there is nothing to converge. Therefore the other filters are negligible regarding the computational effort needed, as shown in Figure 3.26.

These results need yet to be extended to noisy signals, since, for this work, only smooth respiration curves were considered.

4.4 Tracking Simulations

The complete system as shown in Figure 1.2 was implemented in MATLAB/Simulink and allows testing of different prediction filters, couch systems, sensor systems and external controllers. The
simulation results shown in Figure 3.27 indicate the effectiveness of the prediction filters (in this case LCM). The additional tracking error due to the delayed sensor signal, can be compensated effectively by using a prediction filter. But even in the case of zero delay the tracking error is not small, which can be attributed to the slow dynamics of the Protura. One could increase the prediction time to a higher value than the actual delay and compensate the lag caused by the Protura. A closer look at the contributions of all system parts to the tracking error is needed. The Figure 3.28 shows that for common respiration curves and respiration to tumor conversion factors the \( e_{\text{RMS}} \) of the system and the model are comparable. The simulation usually achieves smaller values, with exception of respiration curve 11. The real system still has some small delay, while the simulation does actually have zero delay. Additionally, the simulation results are run with noise free signals, which also improves the results compared to the real case. In the future the sensor model has to be extended adding noise to the system. Therefore, the noise of different sensor systems have to be quantified.
4.4. Tracking Simulations
Chapter 5

Conclusion

Even though the Protura is designed to do static positioning of the patient, it can compensate the tumor motion at least partly. But the compensation for faster tumor motion is still lacking. This thesis concentrated on reducing tracking error caused by delays, but the simulations showed, that also the actuator may contribute significantly to the total tracking error. A different internal controller or a different treatment couch, designed for tumor tracking, should improve the case of faster tumor motion.

The developed model of the real system allows evaluation of different respiration curves, different respiration to tumor correlations, different sensor systems, different prediction filters and different controllers for tumor tracking without requiring the real system at USZ, which is mainly occupied by clinical operation and, therefore, is rarely available. This will speed up further development. The prediction filters do not differ a lot in their results. The standard deviations of the results of each prediction filter are larger than the differences between the prediction filters. The best prediction filter, though, is the SVR filter, which for $T_s = 0.05 \text{s}$ and $T_s = 0.1 \text{s}$ has the second lowest average $e_{\text{RMS}}$ but also has the smallest standard deviation. And its computing time per step is smaller than $0.05 \text{s}$. Its disadvantage is its elaborate algorithm. Further work is needed on the dependency of the prediction performance on the specific respiration curve, because the standard deviation over the respiration curves is greater than the differences between mean $e_{\text{RMS}}$ of the prediction filters.
Appendix A

Protura Derivations and Parameters

A.1 Mechanical Model Derivation

The derivation of the mechanical model of the Protura system follows the same method as is used in [17].

The Protura has six legs. But since all the legs are not directly dependent on each other (no joints between them), it is sufficient to consider one representative leg together with the platform. This situation is shown in Figure A.1. There are three coordinates systems, one inertial system $I$, one leg-fixed system $L$ and one platform-fixed system $T$. A vector represented in either the $L$ or $T$ system has a lower prefix $L$ or $T$. If it is represented in the inertial system, the prefix is omitted. In the following, a list of variables and parameters, that will occur during the derivation, is given:

- $A_{IL}$: Transformation from leg coordinates to inertial coordinates
- $A_{IT}$: Transformation from platform coordinates to inertial coordinates
- $b$: vector from base joint in zero position ($B_0$) to actual position ($B$)(collinear to $d$)
- $b_0$: vector from origin ($O$) to base joint ($B_0$) (zero position)
- $C$: temporary variable for long expression, defined during derivation in (A.43)
- $d$: direction of travel of base joint (unit length)
- $E_3$: $3 \times 3$ unit matrix
- $F_{act}$: actuator force acting on leg at point $B$ for control of system
- $F_{ext}$: unspecified external force acting on platform at point $T$
- $F_{FB}$: linear friction force acting on leg at point $B$
- $F_s$: joint force between leg and platform at point $J$
- $g$: gravitational acceleration
- $h_s$: variable representing long expression, defined during derivation in (A.78)
- $h_t$: variable representing long expression, defined during derivation in (A.80)
- $I_L$: rotational inertia of a leg
- $I_T$: rotational inertia of platform
• **K**: variable representing long expression, defined during derivation in (A.41)

• **k**: auxiliary vector for rotation matrices of the legs

• **l**: vector from base joint (B) to platform joint (J)

• **M_1**: variable representing long expression defined during derivation in (A.57)

• **M_{ext}**: unspecified external torque acting on the platform

• **M_{rB}**: friction torque acting on leg at point B

• **M_{rJ}**: friction torque acting on leg and platform at point J

• **m**: mass of a leg

• **m_s**: variable representing long expression, defined during derivation in (A.77)

• **m_T**: mass of platform

• **p**: vector from platform center (T) to platform joint position (J)

• **q**: quaternion, which represents the orientation of the platform

• **r_L**: vector from base joint (B) to center of gravity of leg (L_{CG})

• **r_{OL}**: vector from origin (O) to center of gravity of leg (L_{CG})

• **r_T**: vector from platform center (T) to center of gravity of platform (T_{CG})

• **r_{OT}**: vector from origin (O) to center of gravity of platform (T_{CG})

• **s**: vector from origin (O) to platform joint (J)

• **t**: vector from origin (O) to platform center (T)

• **V_1**: variable representing long expression, defined during derivation in (A.56)

• **V_2**: variable representing long expression, defined during derivation in (A.62)

• **V_3**: variable representing long expression, defined during derivation in (A.69)

• **V_4**: variable representing long expression, defined during derivation in (A.72)

• **α**: base-leg joint position, scalar value

• **ω_IT**: rotational velocity of the platform

• **ω_IL**: rotational velocity of the leg

Some vectors that are constant in one coordinate system, but not in the inertial coordinate system:

\[
\begin{align*}
\mathbf{p} &= \mathbf{A}_{IT} \mathbf{r} \mathbf{p} \\
\mathbf{l} &= \mathbf{A}_{IL} \mathbf{r} \mathbf{l} \\
\mathbf{r}_T &= \mathbf{A}_{IT} \mathbf{r} \mathbf{r}_T \\
\mathbf{r}_L &= \mathbf{A}_{IL} \mathbf{r} \mathbf{r}_L
\end{align*}
\]
The kinematics of the platform are the starting point, meaning the movement of $T_{CG}$ and $J$ is described depending on the motion of $T$ and the rotation of the platform:

\[
\begin{align*}
\mathbf{r}_{OT} & = \mathbf{t} + \mathbf{r}_{T} \\
\dot{\mathbf{r}}_{OT} & = \dot{\mathbf{t}} + \omega_{IT} \times \mathbf{r}_{T} \\
\ddot{\mathbf{r}}_{OT} & = \ddot{\mathbf{t}} + \omega_{IT} \times \dot{\mathbf{r}}_{T} + \omega_{IT} \times (\omega_{IT} \times \mathbf{r}_{T}) \\
\mathbf{s} & = \mathbf{t} + \mathbf{p} \\
\dot{s} & = \dot{\mathbf{t}} + \omega_{IT} \times \mathbf{p} \\
\ddot{s} & = \ddot{\mathbf{t}} + \omega_{IT} \times \dot{\mathbf{p}} + \omega_{IT} \times (\omega_{IT} \times \mathbf{p})
\end{align*}
\]

Now the leg kinematics are considered. First the velocity of the base joint is described using the platform motion. The variable $\dot{s}$ is considered the independent variable, but later in the derivation it will be replaced by (A.9). The leg is assumed to be a rigid body, therefore the velocities of $B$ and $J$, projected on their connecting straight, have to be equal:

\[
\dot{s} \mathbf{1} = \dot{\mathbf{b}} \mathbf{1}
\]

Where $\mathbf{a}^T$ is the transposed of $\mathbf{a}$. Multiply both sides with $\mathbf{d}$:

\[
(\dot{s} \mathbf{1}) \mathbf{d} = (\dot{\mathbf{b}} \mathbf{1}) \mathbf{d}
\]

Since $\mathbf{d}$ is collinear to $\dot{\mathbf{b}}$:

\[
\mathbf{d} = \alpha \dot{\mathbf{b}}
\]

\[
(\dot{s} \mathbf{1}) \mathbf{d} = (\dot{\mathbf{b}} \mathbf{1}) \alpha \dot{\mathbf{b}}
\]

Move the scalar $\alpha$ inside the brackets:

\[
(\dot{s} \mathbf{1}) \mathbf{d} = (\alpha \dot{\mathbf{b}} \mathbf{1}) \dot{\mathbf{b}}
\]
Therefore, \( d \) is now inside the brackets and \( \dot{s} \) needs to be isolated now:

\[
\begin{align*}
\mathbf{d} (\dot{s}^T l)^T &= (\mathbf{d}^T l) \mathbf{b} \\
\mathbf{d} (l^T \dot{s}) &= (\mathbf{d}^T l) \mathbf{b} \\
(\mathbf{d}^T) \dot{s} &= (\mathbf{d}^T l) \mathbf{b} \\
\end{align*}
\] (A.16)

(A.17)

(A.18)

The base joint velocity can be expressed by the platform joint velocity:

\[
\dot{\mathbf{b}} = \mathbf{dl}^T \mathbf{d}^T l \dot{s} \\
\] (A.19)

The acceleration of the base joint is calculated by taking the time derivative of (A.19):

\[
\ddot{\mathbf{b}} = \mathbf{dl}^T \mathbf{d}^T l \ddot{s} + \mathbf{dl}^T \dot{s} - (\mathbf{d}^T l) \mathbf{dl}^T l \ddot{s} \\
\] (A.20)

Where \( \mathbf{i} \) is:

\[
\mathbf{i} = \omega_{IL} \times \mathbf{l} \\
\] (A.21)

But at this point \( \mathbf{l} \) or \( \mathbf{b} \) are unknown. The equation connecting these is:

\[
\mathbf{s} - \mathbf{l} = \mathbf{b}_0 + \mathbf{b} \\
\] (A.22)

This is only one vector equation for two unknown vectors (six scalar unknowns and three scalar equations). But it is known, that \( \mathbf{b} \) is collinear to \( \mathbf{d} \), so the number of scalar unknowns can be reduced to four:

\[
\mathbf{s} - \mathbf{l} = \mathbf{b}_0 + \alpha \mathbf{d} \\
\] (A.23)

Since the length of \( \mathbf{l} \) (being constant) is known, the set of points described by the left hand side (LHS) of above equation comprise a sphere with its center being \( \mathbf{s} \). The right hand side (RHS) results in points making up a straight in the direction of \( \mathbf{d} \) and supported by \( \mathbf{b}_0 \). Therefore, this equation describes the intersection of a straight with a sphere, which may have zero, one or two solutions. The following holds:

\[
\mathbf{l}^T \mathbf{l} = ||\mathbf{l}||^2 \\
\] (A.24)

Replacing the expressions in the brackets gives:

\[
(s - \mathbf{b}_0 - \alpha \mathbf{d})^T (s - \mathbf{b}_0 - \alpha \mathbf{d}) = ||\mathbf{l}||^2 \\
\] (A.25)

Expanding the equation and formulating it as a quadratic polynomial in \( \alpha \) gives:

\[
\alpha^2 ||\mathbf{d}||^2 - 2 \alpha \mathbf{d} \cdot (s - \mathbf{b}_0) + ||s - \mathbf{b}_0||^2 - ||\mathbf{l}||^2 = 0 \\
\] (A.26)

Since, in general, there will be two solutions, one has to be chosen. While running the simulation both solutions are evaluated, and the solution nearer to the chosen solution of the timestep before is selected. The simulation has to be set up in such a way, that situations, where the two solutions are near to each other, are avoided. One could avoid the ambiguity by integrating \( \dot{\mathbf{b}} \) in situations like these, but this introduces larger numerical errors and additional states.

The next step concerns the kinematics of the center of gravity of the leg and the rotation of the leg:

\[
\mathbf{r}_{OL} = \mathbf{b}_0 + \mathbf{b} + \mathbf{r}_L \\
\dot{\mathbf{r}}_{OL} = \dot{\mathbf{b}} + \omega_{IL} \times \mathbf{r}_L \\
\ddot{\mathbf{r}}_{OL} = \ddot{\mathbf{b}} + \omega_{IL} \times \mathbf{r}_L + \omega_{IL} \times (\omega_{IL} \times \mathbf{r}_L) \\
\omega_{IL} = \frac{1}{||\mathbf{l}||^2} \mathbf{l} \times (\dot{s} - \dot{\mathbf{b}}) \\
\dot{\omega}_{IL} = \frac{1}{||\mathbf{l}||^2} \left( (\omega_{IL} \times \mathbf{l}) \times (\dot{s} - \dot{\mathbf{b}}) + \mathbf{l} \times (\dot{s} - \dot{\mathbf{b}}) \right) \\
\] (A.27)

(A.28)

(A.29)

(A.30)

(A.31)
Subtracting \( \dot{b} \) from \( \dot{s} \) results in a vector with no component in the direction of \( l \). Furthermore, it is assumed that the rotation velocity \( \omega_{IL} \) also has no component in the direction of \( l \) (Universal Joints). This allows a simplification using the identity (A.32):

\[
\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c}) \mathbf{b} - (\mathbf{a} \cdot \mathbf{b}) \mathbf{c} \tag{A.32}
\]

First plug in the rotational velocity (A.30) into the expression for the rotational acceleration (A.31) and rearrange the terms:

\[
\dot{\omega}_{IL} = \frac{1}{||l||^2} \left( \frac{-1}{||l||^2} \left( l \times \left( l \times (\dot{s} - \dot{b}) \right) \right) \times (\dot{s} - \dot{b}) + l \times (\dot{s} - \dot{b}) \right) \tag{A.33}
\]

Apply the identity (A.32):

\[
\dot{\omega}_{IL} = \frac{1}{||l||^2} \left( \frac{-1}{||l||^2} \left( (l \cdot (\dot{s} - \dot{b})) l - (l \cdot l) (\dot{s} - \dot{b}) \right) \times (\dot{s} - \dot{b}) + l \times (\dot{s} - \dot{b}) \right) \tag{A.34}
\]

The first scalar product results in zero value since they are orthogonal to each other (no common components). Additionally there is a vector product of two parallel vectors which also results in a zero vector. Therefore the velocity expressions vanish, and the rotational acceleration is purely dependent on the acceleration values of the upper and lower joints:

\[
\dot{\omega}_{IL} = \frac{1}{||l||^2} \left( \frac{-1}{||l||^2} \left( (l \cdot (\dot{s} - \dot{b})) l \right) \right) \tag{A.35}
\]

Now the linear momentum balances and the angular momentum balances of the leg and the platform are formulated. First, we write the linear momentum of the leg using the center of gravity of the leg and the angular momentum around point \( B \):

\[
mg + \mathbf{F}_{rB} + \mathbf{F}_{act} + \mathbf{F}_s = m\ddot{r}_L \tag{A.36}
\]

\[
\mathbf{M}_{rB} + \mathbf{M}_{rJ} + \mathbf{r}_L \times (mg) + l \times \mathbf{F}_s = m \mathbf{r}_L \times \dot{\mathbf{b}} + \mathbf{I}_L \dot{\omega}_{IL} + \omega_{IL} \times (\mathbf{I}_L \omega_{IL}) \tag{A.37}
\]

Now the same is done for the platform, but the angular momentum balance is around point \( T \):

\[
m_T \mathbf{g} + \mathbf{F}_{ext} - \sum_{i=1}^{6} (\mathbf{F}_s)_i = m_T\ddot{r}_T \tag{A.38}
\]

\[- \sum_{i=1}^{6} (\mathbf{M}_{rJ})_i + \mathbf{M}_{ext} - \sum_{i=1}^{6} (\mathbf{p} \times \mathbf{F}_s)_i + \mathbf{r}_T \times m_T \mathbf{g} = m_T \mathbf{r}_T \times \ddot{\mathbf{t}} + \mathbf{I}_T \ddot{\omega}_{TT} + \omega_{TT} \times (\mathbf{I}_T \omega_{TT}) \tag{A.39}
\]

The friction forces \( \mathbf{F}_{rB}, \mathbf{M}_{rB}, \mathbf{M}_{rJ} \) will be specified after the derivation of the model to keep the model flexible. \( \mathbf{F}_{ext}, \mathbf{M}_{ext} \) can be used to add other force elements, for example the load of a patient situated on the platform.

Now there are six unknown forces \( \mathbf{F}_s \), which have to be eliminated. From translational momentum balance of the leg, it can be simplified:

\[
\mathbf{F}_s = \mathbf{K} \tag{A.40}
\]

Where \( \mathbf{K} \) is:

\[
\mathbf{K} = -mg - \mathbf{F}_{rB} - \mathbf{F}_{act} + m\ddot{r}_L \tag{A.41}
\]

Similarly angular momentum of the leg:

\[
l \times \mathbf{F}_s = \mathbf{C} \tag{A.42}
\]
Where $C$ is:

$$C = m \mathbf{r}_L \times \ddot{\mathbf{b}} + \mathbf{I}_L \dot{\omega}_I - \mathbf{I}_L \mathbf{L} \times (\mathbf{I}_L \omega_I) - M_{rB} - M_{rJ} - \mathbf{r}_L \times (mg) \quad (A.43)$$

Project the translational momentum in the direction $\mathbf{d}$:

$$\mathbf{F}_s \cdot \mathbf{d} = \mathbf{K} \cdot \mathbf{d} \quad (A.44)$$

Multiply the angular momentum with $\mathbf{d}$ by using the cross product:

$$(\mathbf{l} \times \mathbf{F}_s) \times \mathbf{d} = C \times \mathbf{d} \quad (A.45)$$

Use the identity (A.32):

$$\mathbf{d} \times (\mathbf{l} \times \mathbf{F}_s) = -C \times \mathbf{d} \quad (A.46)$$

$$((\mathbf{d} \cdot \mathbf{F}_s) \mathbf{l} - (\mathbf{d} \cdot \mathbf{l}) \mathbf{F}_s) = -C \times \mathbf{d} \quad (A.47)$$

Insert the projected translational momentum:

$$((\mathbf{K} \cdot \mathbf{d}) \mathbf{l} - (\mathbf{d} \cdot \mathbf{l}) \mathbf{F}_s) = -C \times \mathbf{d} \quad (A.48)$$

Solve for $\mathbf{F}_s$:

$$\mathbf{F}_s = \frac{1}{\mathbf{d} \cdot \mathbf{l}} ((\mathbf{K} \cdot \mathbf{d}) \mathbf{l} + \mathbf{C} \times \mathbf{d}) \quad (A.49)$$

The variable $\mathbf{F}_s$ has an acceleration term, so it cannot be computed during simulation directly. These terms can be isolated, but for that the cross product sometimes needs to be rewritten as following:

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \mathbf{b} = \tilde{\mathbf{a}} \mathbf{b} \quad (A.50)$$

$$\mathbf{F}_s = \frac{1}{\mathbf{d} \cdot \mathbf{l}} ((\mathbf{d})^T \mathbf{K} - \tilde{\mathbf{d}} \mathbf{C}) \quad (A.51)$$

$$= \frac{1}{\mathbf{d} \cdot \mathbf{l}} ((\mathbf{d})^T (m \tilde{\mathbf{r}}_L - \mathbf{F}_{act} - \mathbf{F}_{rB} - mg) - \tilde{\mathbf{d}} \mathbf{C}) \quad (A.52)$$

$$= \frac{1}{\mathbf{d} \cdot \mathbf{l}} ((\mathbf{d})^T (m \tilde{\mathbf{r}}_L - mg) + l (-\mathbf{F}_{act} - \mathbf{F}_{rB}) - \tilde{\mathbf{d}} \mathbf{C}) \quad (A.53)$$

The $\mathbf{F}_s$ is reformulated such that $\ddot{\mathbf{s}}$ is affine in the expression. Start with $\mathbf{C}$ omitting the coordinate system projection gives:

$$C = m \mathbf{r}_L \times \ddot{\mathbf{b}} + \mathbf{I}_L \dot{\omega}_I - \mathbf{I}_L \mathbf{L} \times (\mathbf{I}_L \omega_I) - M_{rB} - M_{rJ} - \mathbf{r}_L \times (mg) \quad (A.54)$$

$$= m \tilde{\mathbf{r}}_L \ddot{\mathbf{b}} + \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) + \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) \times \left( \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) \right) - M_{rB} - M_{rJ} + m \tilde{\mathbf{g}} r_L \quad (A.55)$$

The terms, that do not include the accelerations, are collected:

$$\mathbf{V}_1 = \left( \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) \right) \times \left( \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) \right) \quad (A.56)$$

$$\mathbf{M}_1 = -M_{rB} - M_{rJ} + m \tilde{\mathbf{g}} r_L \quad (A.57)$$

$$\mathbf{C} = m \tilde{\mathbf{r}}_L \ddot{\mathbf{b}} + \frac{\mathbf{I}_L \ddot{\mathbf{L}}}{|\mathbf{L}|^2} (\ddot{\mathbf{s}} - \dot{\mathbf{b}}) + \mathbf{V}_1 + \mathbf{M}_1 \quad (A.58)$$
Appendix A. Protura Derivations and Parameters

Next replace the \( \ddot{b} \):

\[
C = \left(m\ddot{r}_L - \frac{I_L \ddot{l}}{||l||^2}\right) \ddot{b} + \frac{I_L \ddot{l}}{||l||^2} \ddot{s} + V_1 + M_1
\]  
(A.59)

\[
= \left(m\ddot{r}_L - \frac{I_L \ddot{l}}{||l||^2}\right) \left(\frac{d\dot{I}^T}{d\dot{I}} \ddot{s} + \frac{d\dot{I}^T}{d\dot{I}} \ddot{s} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2} \ddot{s}\right) + \frac{I_L \ddot{l}}{||l||^2} \ddot{s} + V_1 + M_1
\]  
(A.60)

\[
= \left[m\ddot{r}_L \frac{d\dot{I}^T}{d\dot{I}} + \frac{I_L \ddot{l}}{||l||^2} \left(E_3 - \frac{d\dot{I}^T}{d\dot{I}}\right)\right] \ddot{s} + \left(m\ddot{r}_L - \frac{I_L \ddot{l}}{||l||^2}\right) \left[\frac{d\dot{I}^T}{d\dot{I}} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2}\right] \ddot{s}
\]  
(A.61)

\[
+ V_1 + M_1
\]

\[
V_2 = \left(m\ddot{r}_L - \frac{I_L \ddot{l}}{||l||^2}\right) \left[\frac{d\dot{I}^T}{d\dot{I}} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2}\right] \ddot{s}
\]  
(A.62)

\[
C = \left(m\ddot{r}_L \frac{d\dot{I}^T}{d\dot{I}} + \frac{I_L \ddot{l}}{||l||^2} \left(E_3 - \frac{d\dot{I}^T}{d\dot{I}}\right)\right] \ddot{s} + V_2 + V_1 + M_1
\]  
(A.63)

Now \( \ddot{s} \) is affine in \( C \). \( E_3 \) is the \( 3 \times 3 \) identity matrix. Doing the same for \( F_s \), while keeping \( C \), gives:

\[
F_s = \frac{1}{d\dot{I}^T}(ld^T)m\ddot{r}_L + F_1 - \ddot{d}C
\]  
(A.64)

\[
F_1 = -(ld^T) (mg + F_{act} + F_{ob})
\]  
(A.65)

Again replace the acceleration of the center of gravity of the log.

\[
F_s = \frac{1}{d\dot{I}^T} \left[(ld^T) m \left(\ddot{b} + \dot{\omega} \times r_L + \dot{\omega} \times (r_L \times \dot{r}_L)\right) + F_1 - \ddot{d}C\right]
\]  
(A.66)

\[
= \frac{1}{d\dot{I}^T} \left[(ld^T) m \left(\frac{d\dot{I}^T}{d\dot{I}} \ddot{s} + \frac{d\dot{I}^T}{d\dot{I}} \ddot{s} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2} \ddot{s}\right)
\]

\[
+ \left(\frac{1}{||l||^2} (\ddot{s} - \ddot{b})\right) \times r_L + \dot{\omega} \times (r_L \times \dot{r}_L)\right) + F_1 - \ddot{d}C
\]  
(A.67)

\[
= \frac{1}{d\dot{I}^T} \left[(ld^T) m \left(\left(\frac{d\dot{I}^T}{d\dot{I}} - \ddot{r}_L \frac{1}{||l||^2}\right) \ddot{s} + \ddot{r}_L \frac{1}{||l||^2} \ddot{b} + V_3\right) + F_1 - \ddot{d}C\right]
\]  
(A.68)

\[
V_3 = \frac{d\dot{I}^T}{d\dot{I}} \ddot{s} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2} \ddot{s} + \dot{\omega} \times (\dot{\omega} \times r_L)
\]  
(A.69)

\[
F_s = \frac{1}{d\dot{I}^T} \left[(ld^T) m \left(\left(\frac{d\dot{I}^T}{d\dot{I}} - \ddot{r}_L \frac{1}{||l||^2}\right) \ddot{s} + \ddot{r}_L \frac{1}{||l||^2} \left(\frac{d\dot{I}^T}{d\dot{I}} \ddot{s}\right)
\]

\[
+ \frac{d\dot{I}^T}{d\dot{I}} \ddot{s} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2} \ddot{s}\right) + V_3\right) + F_1 - \ddot{d}C\right]
\]  
(A.70)

\[
= \frac{1}{d\dot{I}^T} \left[(ld^T) m \left(\left(\frac{d\dot{I}^T}{d\dot{I}} - \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d\dot{I}^T}{d\dot{I}}\right) \ddot{s} + V_4 + V_3\right) + F_1 - \ddot{d}C\right]
\]  
(A.71)

\[
V_4 = \ddot{r}_L \frac{1}{||l||^2} \left(\frac{d\dot{I}^T}{d\dot{I}} \ddot{s} - \frac{(d\dot{I}^T)(d\dot{I})}{(d\dot{I})^2} \ddot{s}\right)
\]  
(A.72)

Now plug in the expression \( C \):

\[
F_s = \frac{ld^T}{d\dot{I}^T} m \left(\frac{d\dot{I}^T}{d\dot{I}} - \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d\dot{I}^T}{d\dot{I}}\right) \ddot{s} - \ddot{d} \frac{d\dot{I}^T}{d\dot{I}} C + \frac{ld^T}{d\dot{I}^T} m(V_4 + V_3) + \frac{1}{d\dot{I}^T} F_1
\]  
(A.73)
For further calculations:

\[ \frac{d}{dt} \begin{pmatrix} \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d l}{dr} \end{pmatrix} \tilde{s} = \frac{d}{dr} \left( \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d l}{dr} \right) \tilde{s} + V_2 + V_1 + M_1 \]  
(A.74)

\[ + \frac{d}{dt} m(V_4 + V_3) + \frac{1}{dt} F_1 \]

\[ = \left[ \frac{d}{dt} m \left( \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d l}{dr} \right) - \frac{d}{dt} \left[ m \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \left( E_3 - \frac{d l}{dr} \right) \right] \right] \tilde{s} \]
\[ - \frac{d}{dt} (V_2 + V_1 + M_1) + \frac{d}{dt} m(V_4 + V_3) + \frac{1}{dt} F_1 \]
(A.75)

For further calculations:

\[ F_s = m_s \tilde{s} + h_s \]
(A.76)

\[ m_s = \frac{d}{dt} m \left( \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \frac{d l}{dr} \right) - \frac{d}{dt} \left[ m \ddot{r}_L \frac{1}{||l||^2} + \ddot{r}_L \frac{1}{||l||^2} \left( E_3 - \frac{d l}{dr} \right) \right] \]
(A.77)

\[ h_s = - \frac{d}{dt} (V_2 + V_1 + M_1) + \frac{d}{dt} m(V_4 + V_3) + \frac{1}{dt} F_1 \]
(A.78)

Replace \( \tilde{s} \) :

\[ F_s = m_s \tilde{i} - m_s \tilde{p} \omega_{IT} + h_t \]
(A.79)

\[ h_t = m_s (\omega_{IT} \times (\omega_{IT} \times p)) + h_s \]
(A.80)

Now the expression found for \( F_s \) is inserted into the equations for the platform starting with the translational momentum:

\[ m_T g + F_{ext} - \sum_{i=1}^{6} \left( m_s \tilde{i} - m_s \tilde{p} \omega_{IT} + h_t \right)_i = m_T \left( \tilde{i} + \omega_{IT} \times \tilde{r}_T + \omega_{IT} \times (\omega_{IT} \times \tilde{r}_T) \right) \]

\[ (\sum_{i=1}^{6} m_s, i + m_T E_3) \tilde{i} + \left( -\sum_{i=1}^{6} m_s, i \tilde{p}_i - m_T \tilde{r}_T \right) \omega_{IT} = \]

\[ m_T g + F_{ext} = \sum_{i=1}^{6} h_t, i - m_T (\omega_{IT} \times (\omega_{IT} \times \tilde{r}_T)) \]
(A.81)

Next the angular momentum:

\[ - \sum_{i=1}^{6} M_{r, i} = \sum_{i=1}^{6} \left( \tilde{p} \times (m_s \tilde{i} - m_s \tilde{p} \omega_{IT} + h_t) \right)_i + \tilde{r}_T \times m_T g = \]

\[ \tilde{r}_T \times m_T \tilde{i} + I_T \omega_{IT} + \omega_{IT} \times (I_T \omega_{IT}) \]
\[ (\sum_{i=1}^{6} \tilde{p} m_s, i + m_T \tilde{r}_T) \tilde{i} + \left( -\sum_{i=1}^{6} \tilde{p} m_s, i \tilde{p}_i + I_T \right) \omega_{IT} = \]
\[ m_T r_T \times g + M_{ext} = \sum_{i=1}^{6} M_{r, i} - \sum_{i=1}^{6} \tilde{p} \times h_t, i - \omega_{IT} \times (I_T \omega_{IT}) \]
(A.82)

\[ m_T r_T \times g + M_{ext} - \sum_{i=1}^{6} M_{r, i} = \sum_{i=1}^{6} \tilde{p} \times h_t, i - \omega_{IT} \times (I_T \omega_{IT}) \]
(A.83)
And this gives the ordinary differential equations for the Protura system:

\[
\begin{align*}
\begin{pmatrix}
\sum_{i=1}^{6} m_{x,i} + m_T E_2 - \sum_{i=1}^{6} m_{s,i} \tilde{p}_i - m_T \tilde{r}_T \\
\sum_{i=1}^{6} \tilde{p}_i m_{x,i} + m_T \tilde{r}_T - \sum_{i=1}^{6} \tilde{p}_i m_{s,i} \tilde{p}_i + I_T
\end{pmatrix} \begin{pmatrix}
\dot{t} \\
\tilde{\omega}_{IT}
\end{pmatrix} &= (A.85) \\
- \left( \sum_{i=1}^{6} h_{t,i} \right) + \left( F_{ext} \right) + \left( m_T g \right) - \left( m_T \omega_{IT} \times (\omega_{IT} \times r_T) \right) - \left( \sum_{i=1}^{6} M_{r,i} \right) &= 0 (A.86)
\end{align*}
\]

A.1.1 General Rotations

The transformation matrix from platform system to inertial system \(A_{IT}\) can be calculated using the following formula:

\[
\frac{dA_{IT}}{dt} = \tilde{\omega}_{IT} A_{IT} (A.87)
\]

This introduces 9 additional states, which have to be algebraically constrained. The algebraic constraints are mainly due to the demanded orthogonality of transformation matrices. This is numerically problematic, so to avoid this, quaternions are used. Quaternions are similar to complex numbers (and also called hypercomplex numbers) and are often written this way:

\[
q = q_0 + q_1 i + q_2 j + q_3 k = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}
\]

Where \(q_0, q_1, q_2, q_3\) are real numbers and 1, \(i, j, k\) is the basis. It holds that:

\[
i^2 = j^2 = k^2 = ijk = -1
\] (A.89)

An introduction to quaternions is given in [26]. The differential equation for rotations using quaternions is:

\[
\frac{dq}{dt} = \frac{1}{2} \begin{pmatrix} 0 \\ \omega_{IT} \end{pmatrix} \otimes q (A.90)
\]

Where \(\otimes\) is the quaternion multiplication, which can be written as a matrix multiplication (omitting the quaternion basis):

\[
q \otimes r = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix} (A.91)
\]

This introduces only four states. And the only algebraic constraint is the norm of \(q\) required to be 1: \(||q|| = 1\). The rotation matrix \(A_{IT}\) is then computed as follows (assuming a normalized quaternion):

\[
A_{IT} = \begin{pmatrix}
1 - 2q_3^2 - 2q_2^2 & 2q_1q_2 - 2q_3q_0 & 2q_1q_3 + 2q_2q_0 \\
2q_1q_2 + 2q_3q_0 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_1q_0 \\
2q_1q_3 - 2q_2q_0 & 2q_2q_3 + 2q_1q_0 & 1 - 2q_1^2 - 2q_2^2
\end{pmatrix} (A.92)
\]

The transformation matrix from leg system to inertial system \(A_{IL}\) depends on the current configuration of the leg, which is algebraically dependent on the platform, so no integration is possible. An auxiliary vector \(k\) and the vector \(l\), describing the current leg orientation, are used. Then the
transformation matrix is computed as follows:

\[
x = \frac{1}{||l||} \]

\[
y = \frac{k \times x}{||k \times x||} \]

\[
z = x \times y
\]

\[A_{TE} = [x, y, z]\]

This implies that the auxiliary vector \(k\) and the leg vector \(l\) should never be parallel.

A.2 Mechanical Model with Extension Plate

At one of the short sides of the platform an extension plate can be attached. The joint may have play in the bearing and the extension plate might bend. Therefore, the extension plate is modeled as an additional rigid body. The play in the bearing and the bending are modeled by introducing one degree of freedom, namely the rotation around the side of the platform by which the extension is attached. Additionally, a spring damper element depending on the angle is assumed to act between the extension and the platform to account for bending. An overview is given in Figure 2.3. In the following a list of variables and parameters, that will occur during the derivation, is given. Any variable or parameter, that is already in the list in Section A.1, is omitted:

- \(A_1\) : variable representing long expression defined during derivation in (A.123)
- \(A_2\) : variable representing long expression defined during derivation in (A.126)
- \(A_{TE}\) : Transformation from extension plate to inertial coordinates
- \(A_{TE}\) : Transformation from extension plate to platform coordinates
- \(B_1\) : variable representing long expression defined during derivation in (A.123)
- \(B_2\) : variable representing long expression defined during derivation in (A.126)
- \(C_1\) : variable representing long expression defined during derivation in (A.123)
- \(C_2\) : variable representing long expression defined during derivation in (A.126)
- \(e_y\) : rotation axis of platform-extension joint
- \(F_{FE}\) : variable representing long expression defined during derivation in (A.117)
- \(F_E\) : joint force between the extension plate and the platform
- \(F_{\text{load}}\) : external force acting on the extension plate at position \(r_{\text{load}}\)
- \(I_E\) : rotational inertia of extension plate
- \(m_E\) : mass of extension plate
- \(M_E\) : joint torque between the extension plate and the platform
- \(M_{\text{FD}}\) : torque due to spring-damper element at the platform-extension joint
- \(M_{ME}\) : variable representing long expression defined during derivation in (A.122)
- \(p_E\) : vector from point \(T\) (center of platform) to point \(E\) (platform-extension joint)
- \(r_{\text{load}}\) : position of external force \(F_{\text{load}}\) acting on the extension plate
- \( \mathbf{r}_E \): vector from point \( E \) to center of gravity \( E_{CG} \)
- \( \mathbf{r}_{OE} \): vector from origin \( O \) to center of gravity \( E_{CG} \)
- \( \mathbf{s}_E \): vector from origin \( O \) to joint \( E \)
- \( V_{FE} \): variable representing long expression defined during derivation in (A.116)
- \( V_{ME} \): variable representing long expression defined during derivation in (A.121)
- \( \phi \): rotation angle of the extension plate relative to the platform
- \( \omega_{IE} \): rotational velocity of extension plate

Starting with writing down the equations of momentum of the extension plate give:

\[
m_E \ddot{\mathbf{r}}_{OE} = m_E \mathbf{g} + \mathbf{F}_E + \mathbf{F}_{load} \tag{A.97}
\]
\[
\mathbf{I}_E \dot{\omega}_{TE} + \mathbf{I}_E \omega_{TE} \times \mathbf{I}_E \dot{\omega}_{TE} + m_E \mathbf{r}_E \times \dot{\mathbf{s}}_E = \mathbf{M}_E + \mathbf{M}_{FD} + m_E \mathbf{r}_E \times \mathbf{g} + \mathbf{r}_{load} \times \mathbf{F}_{load} \tag{A.98}
\]

Where \( \mathbf{F}_E \) and \( \mathbf{M}_E \) are the forces due to the restriction of degrees of freedom, \( \mathbf{M}_{FD} \) is the torque due to the spring damper element and \( \mathbf{F}_{load} \) is the force due to loads on the extension. To accommodate for the new forces, the momentum equations of the platform have to be modified:

\[
m_T \mathbf{g} + \mathbf{F}_{ext} - \sum_{i=1}^{6} (\mathbf{F}_s)_i - \mathbf{F}_E = m_T \dot{\mathbf{r}}_T \tag{A.99}
\]
\[
- \sum_{i=1}^{6} (\mathbf{M}_{rj})_i + \mathbf{M}_{ext} - \sum_{i=1}^{6} ((\mathbf{A}_{ITT}) \times \mathbf{F}_s)_i + \mathbf{r}_T \times m_T \mathbf{g} - \mathbf{M}_E - \mathbf{p}_E \times \mathbf{F}_E - \mathbf{M}_{FD} = \\
\quad m_T \mathbf{r}_T \times \ddot{\mathbf{r}}_T + \mathbf{I}_T \dot{\omega}_{IT} + \dot{\omega}_{IT} \times (\mathbf{I}_T \omega_{IT}) \tag{A.100}
\]

Now the kinematics of \( \mathbf{s}_E \) and \( \mathbf{r}_{OE} \) are needed:

\[
\mathbf{s}_E = \mathbf{t} + \mathbf{A}_{ITT} \tau \mathbf{p}_E \tag{A.101}
\]
\[
\dot{\mathbf{s}}_E = \mathbf{t} + \omega_{IT} \times (\mathbf{A}_{ITT} \tau \mathbf{p}_E) \tag{A.102}
\]
\[
\ddot{\mathbf{s}}_E = \ddot{\mathbf{t}} + \omega_{IT} \times (\mathbf{A}_{ITT} \tau \mathbf{p}_E) + \omega_{IT} \times (\omega_{IT} \times (\mathbf{A}_{ITT} \tau \mathbf{p}_E)) \tag{A.103}
\]
Now plug the kinematics into the equations of momentum for the extension and solve for $F_E$ and $M_E$, such that the acceleration terms are isolated:

$$F_E = m_E (\dot{s}_E + \dot{\omega}_{IE} \times A_{IE} \dot{r}_E + \omega_{IE} \times (\omega_{IE} \times A_{IE} \dot{r}_E)) - m_E g - F_{load}$$

$$= m_E \left[ \dot{t} + \dot{\omega}_{IT} \times (A_{IT} \tau p_E) + \omega_{IT} \times (\omega_{IT} \times (A_{IT} \tau p_E)) + \left( \dot{\omega}_{IT} + \omega_{IT} \times \left(A_{IT} e_y \dot{\phi} \right) + A_{IT} e_y \ddot{\phi} \right) \times A_{IE} \dot{r}_E + \omega_{IE} \times (\omega_{IE} \times A_{IE} \dot{r}_E) \right]$$

$$- m_E g - F_{load}$$

$$= m_E \left[ \dot{t} + \dot{\omega}_{IT} \times (A_{IT} \tau p_E) + \omega_{IT} \times A_{IE} \dot{r}_E + \left(A_{IT} e_y \dot{\phi} \right) \times A_{IE} \dot{r}_E \right] + V_{FE} + F_{FE}$$

$$V_{FE} = m_E \left( \omega_{IT} \times (\omega_{IT} \times (A_{IT} \tau p_E)) + \left( \omega_{IT} \times \left(A_{IT} e_y \dot{\phi} \right) \right) \times A_{IE} \dot{r}_E \right)$$

$$+ \omega_{IE} \times (\omega_{IE} \times A_{IE} \dot{r}_E)$$

$$F_{FE} = - m_E g - F_{load}$$

$$M_E = I_E \omega_{IE} + \omega_{IE} \times I_E \omega_{IE} + m_E r_E \times \ddot{s}_E - M_{FD} - m_E r_E \times g - r_{load} \times F_{load}$$

$$= I_E \left( \omega_{IT} + \omega_{IT} \times \left(A_{IT} e_y \dot{\phi} \right) + A_{IT} e_y \ddot{\phi} \right) + \omega_{IE} \times I_E \omega_{IE}$$

$$+ m_E r_E \times (\dot{t} + \dot{\omega}_{IT} \times (A_{IT} \tau p_E) + \omega_{IT} \times (\omega_{IT} \times (A_{IT} \tau p_E)))$$

$$- M_{FD} - m_E r_E \times g - r_{load} \times F_{load}$$

$$= I_E \omega_{IT} + I_E A_{IT} e_y \ddot{\phi} + m_E r_E \times \ddot{t} + m_E r_E \times (\dot{\omega}_{IT} \times (A_{IT} \tau p_E))$$

$$+ V_{ME} + M_{ME}$$

$$V_{ME} = I_E \omega_{IT} \times \left(A_{IT} e_y \dot{\phi} \right) + \omega_{IE} \times I_E \omega_{IE} + m_E r_E \times (\omega_{IT} \times (\omega_{IT} \times (A_{IT} \tau p_E)))$$

$$M_{ME} = - M_{FD} - m_E r_E \times g - r_{load} \times F_{load}$$
Now these expressions have to be plugged into the momentum equations of the platform starting with the legs already incorporated. First the linear momentum:

\[
\begin{align*}
\left( \sum_{i=1}^{6} \mathbf{m}_{s,i} + m_T \mathbf{E}_3 \right) \ddot{\mathbf{r}}_T + \left( \sum_{i=1}^{6} \mathbf{m}_{s,i} \ddot{\mathbf{p}}_i - m_T \ddot{\mathbf{r}}_T \right) \omega_{IT} + \mathbf{F}_E \\
= m_T \mathbf{g} + \mathbf{F}_{ext} - \sum_{i=1}^{6} \mathbf{h}_{t,i} - m_T (\omega_{IT} \times (\omega_{IT} \times \mathbf{r}_T))
\end{align*}
\]  
\quad (A.123)

\[
\begin{align*}
\mathbf{A}_1 \ddot{\mathbf{r}}_T + \mathbf{B}_1 \omega_{IT} + m_\mathbf{E} \left[ \dot{\mathbf{r}}_T + \omega_{IT} \times (\mathbf{A}_{IT} \mathbf{r}_E) + \omega_{IT} \times \mathbf{A}_{IT} \mathbf{r}_E + \left( \mathbf{A}_{IT} \mathbf{e}_y \dot{\phi} \right) \times \mathbf{A}_{IT} \mathbf{r}_E \right] \\
= \mathbf{C}_1 - \mathbf{V}_{FE} - \mathbf{F}_{FE}
\end{align*}
\]  
\quad (A.124)

\[
\begin{align*}
(A_1 + m_\mathbf{E} \mathbf{E}_3) \dot{\mathbf{r}}_T + (\mathbf{B}_1 - m_\mathbf{E} (\ddot{\mathbf{r}}_E + \ddot{\mathbf{r}}_E)) \omega_{IT} + (-m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_y) \ddot{\phi} = \mathbf{C}_1 - \mathbf{V}_{FE} - \mathbf{F}_{FE}
\end{align*}
\]  
\quad (A.125)

Rotational momentum:

\[
\begin{align*}
\left( \sum_{i=1}^{6} \mathbf{p}_i \mathbf{m}_{s,i} + m_T \ddot{\mathbf{r}}_T \right) \ddot{\mathbf{r}}_E + \left( \sum_{i=1}^{6} \mathbf{p}_i \mathbf{m}_{s,i} \ddot{\mathbf{p}}_i - \mathbf{I}_T \right) \omega_{IT} + \mathbf{M}_E + \mathbf{p}_E \times \mathbf{F}_E = \\
= \mathbf{C}_2 - \mathbf{M}_{FD} - \mathbf{V}_{ME} - \mathbf{M}_{ME} - \mathbf{M}_E \mathbf{e}_y \times (\mathbf{V}_{FE} + \mathbf{F}_{FE})
\end{align*}
\]  
\quad (A.126)

\[
\begin{align*}
(A_2 + m_\mathbf{E} (\ddot{\mathbf{r}}_E + \ddot{\mathbf{r}}_E)) \dot{\mathbf{r}}_E + (\mathbf{B}_2 + \mathbf{I}_E - m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_E) \omega_{IT} + (\mathbf{I}_E - m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_E) \mathbf{e}_y \ddot{\phi} = \mathbf{C}_2 - \mathbf{M}_{FD} - \mathbf{V}_{ME} - \mathbf{M}_{ME} - \mathbf{M}_E \mathbf{e}_y \times (\mathbf{V}_{FE} + \mathbf{F}_{FE})
\end{align*}
\]  
\quad (A.127)

There are now six equations available, but there are seven degrees of freedom. The torque \(\mathbf{M}_E\) has a zero component corresponding to the additional degree of freedom. Therefore:

\[
\mathbf{e}_y^T \mathbf{M}_E = 0
\]  
\quad (A.129)

This applied to (A.120) gives:

\[
\begin{align*}
\left( m_\mathbf{E} \mathbf{e}_y^T \ddot{\mathbf{r}}_E \right) \ddot{\mathbf{r}}_E + \mathbf{e}_y^T (\mathbf{I}_E - m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_E) \omega_{IT} + \mathbf{e}_y^T \mathbf{I}_E \mathbf{e}_y \ddot{\phi} = -\mathbf{e}_y^T \mathbf{V}_{ME} - \mathbf{e}_y^T \mathbf{M}_{ME}
\end{align*}
\]  
\quad (A.130)

The complete equations of motion:

\[
\begin{bmatrix}
\mathbf{A}_1 + m_\mathbf{E} \mathbf{E}_3 & \mathbf{B}_1 - m_\mathbf{E} (\ddot{\mathbf{r}}_E + \ddot{\mathbf{r}}_E) & -m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_y \\
\mathbf{A}_2 + m_\mathbf{E} (\ddot{\mathbf{r}}_E + \mathbf{r}_E) & \mathbf{B}_2 + \mathbf{I}_E - m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_E & (\mathbf{I}_E - m_\mathbf{E} \ddot{\mathbf{r}}_E \mathbf{e}_E) \mathbf{e}_y \\
\mathbf{M}_{E} \mathbf{e}_y^T & \mathbf{e}_y^T \mathbf{I}_E \mathbf{e}_y & \mathbf{e}_y^T \mathbf{M}_{ME}
\end{bmatrix}
\begin{bmatrix}
\ddot{\mathbf{r}}_T \\
\omega_{IT} \\
\ddot{\phi}
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{V}_{FE} - \mathbf{F}_{FE} \\
-\mathbf{V}_{ME} - \mathbf{M}_{ME} \mathbf{e}_y \times (\mathbf{V}_{FE} + \mathbf{F}_{FE}) \\
-\mathbf{e}_y^T \mathbf{V}_{ME} - \mathbf{e}_y^T \mathbf{M}_{ME} \\
0 \\
0
\end{bmatrix}
\]  
\quad (A.131)
A.3 Protura Parameters

The Protura parameters given in this section were given by the company, that designed the parallel manipulator of the Protura system. The numbering of the legs corresponds to the numbering in Figure 2.1.

Table A.1: Locations of base-leg joints (Point $B$ as defined in A.1) in zero position ($B_0$) described in inertial coordinate system.

<table>
<thead>
<tr>
<th>Leg</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-Coordinate [mm]</td>
<td>-528.0</td>
<td>-269.8</td>
<td>269.8</td>
<td>509.0</td>
<td>210.3</td>
<td>-190.2</td>
</tr>
<tr>
<td>y-Coordinate [mm]</td>
<td>-32.3</td>
<td>-205.0</td>
<td>-205.0</td>
<td>-37.4</td>
<td>205.0</td>
<td>205.0</td>
</tr>
<tr>
<td>z-Coordinate [mm]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.2: Locations of platform-leg joints (Point $J$ as defined in A.1) described in platform coordinate system $\mathcal{T}$.

<table>
<thead>
<tr>
<th>Leg</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-Coordinate [mm]</td>
<td>-528.0</td>
<td>-113.6</td>
<td>113.6</td>
<td>509.0</td>
<td>334.1</td>
<td>-379.9</td>
</tr>
<tr>
<td>y-Coordinate [mm]</td>
<td>157.4</td>
<td>-205.0</td>
<td>-205.0</td>
<td>86.4</td>
<td>205.0</td>
<td>205.0</td>
</tr>
<tr>
<td>z-Coordinate [mm]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.3: Leg lengths and centers of gravity (CoG) described in leg coordinate system $\mathcal{L}$.

<table>
<thead>
<tr>
<th>Leg</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leg length [mm]</td>
<td>228.0</td>
<td>201.0</td>
<td>201.0</td>
<td>177.0</td>
<td>177.0</td>
<td>228.0</td>
</tr>
<tr>
<td>CoG x-Coordinate [mm]</td>
<td>114.0</td>
<td>100.5</td>
<td>100.5</td>
<td>88.5</td>
<td>88.5</td>
<td>114.0</td>
</tr>
<tr>
<td>CoG y-Coordinate [mm]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CoG z-Coordinate [mm]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table A.4: Rotational inertias of legs in leg coordinate system \( \mathcal{L} \) with respect to the CoG (in kg \( \cdot \) m\(^2\)).

| Inertia of Legs 1,6 [kg \( \cdot \) m\(^2\)] |  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.126 ( \cdot ) 10(^{-5})</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2.062 ( \cdot ) 10(^{-3})</td>
</tr>
<tr>
<td>0</td>
<td>2.08 ( \cdot ) 10(^{-3})</td>
</tr>
</tbody>
</table>

| Inertia of Legs 2,3 [kg \( \cdot \) m\(^2\)] |  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.892 ( \cdot ) 10(^{-5})</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1.46 ( \cdot ) 10(^{-3})</td>
</tr>
<tr>
<td>0</td>
<td>1.48 ( \cdot ) 10(^{-3})</td>
</tr>
</tbody>
</table>

| Inertia of Legs 4,5 [kg \( \cdot \) m\(^2\)] |  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.724 ( \cdot ) 10(^{-5})</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1.051 ( \cdot ) 10(^{-3})</td>
</tr>
<tr>
<td>0</td>
<td>1.065 ( \cdot ) 10(^{-3})</td>
</tr>
</tbody>
</table>

Table A.5: The mass and the rotational inertia described in the platform coordinate system \( \mathcal{T} \), as well as the CoG with respect to the geometrical center of the platform.

<table>
<thead>
<tr>
<th>Mass of platform [kg]</th>
<th>19.566</th>
</tr>
</thead>
</table>

| Rotational Inertia of platform [kg \( \cdot \) m\(^2\)] |  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.684 ( \cdot ) 10(^{-4})</td>
<td>-5.684 ( \cdot ) 10(^{-5})</td>
</tr>
<tr>
<td>2.72 ( \cdot ) 10(^{-2})</td>
<td>2.72</td>
</tr>
<tr>
<td>3.287</td>
<td>0</td>
</tr>
</tbody>
</table>

| Center of gravity of platform [mm] |  \\
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CoG x-Coordinate</td>
<td>4.33</td>
</tr>
<tr>
<td>CoG y-Coordinate</td>
<td>0.54</td>
</tr>
<tr>
<td>CoG z-Coordinate</td>
<td>9.65</td>
</tr>
</tbody>
</table>
Appendix B

Respiration

B.1 Respiration Curves

In the following the respiration curves used in this thesis to evaluate the performance of prediction filters and control systems are shown.

The figures do not show the full respiration curve over the whole measurement time, but rather a at least 100 s long segment, which shows the characteristic part of the curve. Mind the differences in the position scales.

![Respiration curve 1](image1)

Figure B.1: Respiration curve 1

![Respiration curve 2](image2)

Figure B.2: Respiration curve 2
Appendix B. Respiration

Figure B.7: Respiration curve 7

Figure B.8: Respiration curve 8

Figure B.9: Respiration curve 9

Figure B.10: Respiration curve 10
Figure B.11: Respiration curve 11

Figure B.12: Respiration curve 12

Figure B.13: Respiration curve 13

Figure B.14: Respiration curve 14
Figure B.15: Respiration curve 15

Figure B.16: Respiration curve 16

Figure B.17: Respiration curve 17

Figure B.18: Respiration curve 18
Figure B.19: Respiration curve 19
## B.2 Respiration Characteristics

Table B.1: The mean amplitude (mean to peak), the fundamental frequency, the mean speed, the maximum speed, the mean acceleration and the maximum acceleration values for each respiration curve.

<table>
<thead>
<tr>
<th></th>
<th>Ampl. [mm]</th>
<th>Fund. Freq. [Hz]</th>
<th>Mean Speed [mm/s]</th>
<th>Max. Speed [mm/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.50</td>
<td>0.29</td>
<td>1.64</td>
<td>5.73</td>
</tr>
<tr>
<td>2</td>
<td>4.59</td>
<td>0.17</td>
<td>3.01</td>
<td>9.38</td>
</tr>
<tr>
<td>3</td>
<td>1.92</td>
<td>0.31</td>
<td>2.05</td>
<td>6.21</td>
</tr>
<tr>
<td>4</td>
<td>3.97</td>
<td>0.16</td>
<td>2.56</td>
<td>11.81</td>
</tr>
<tr>
<td>5</td>
<td>1.05</td>
<td>0.19</td>
<td>0.80</td>
<td>1.98</td>
</tr>
<tr>
<td>6</td>
<td>1.09</td>
<td>0.21</td>
<td>0.89</td>
<td>2.78</td>
</tr>
<tr>
<td>7</td>
<td>3.66</td>
<td>0.29</td>
<td>3.85</td>
<td>18.53</td>
</tr>
<tr>
<td>8</td>
<td>2.36</td>
<td>0.43</td>
<td>3.25</td>
<td>7.01</td>
</tr>
<tr>
<td>9</td>
<td>3.46</td>
<td>0.20</td>
<td>2.96</td>
<td>12.89</td>
</tr>
<tr>
<td>10</td>
<td>2.99</td>
<td>0.34</td>
<td>3.52</td>
<td>6.71</td>
</tr>
<tr>
<td>11</td>
<td>5.38</td>
<td>0.21</td>
<td>4.59</td>
<td>10.28</td>
</tr>
<tr>
<td>12</td>
<td>1.71</td>
<td>0.40</td>
<td>1.79</td>
<td>5.38</td>
</tr>
<tr>
<td>13</td>
<td>2.23</td>
<td>0.31</td>
<td>2.36</td>
<td>5.08</td>
</tr>
<tr>
<td>14</td>
<td>1.83</td>
<td>0.35</td>
<td>1.99</td>
<td>5.58</td>
</tr>
<tr>
<td>15</td>
<td>6.64</td>
<td>0.17</td>
<td>4.58</td>
<td>10.95</td>
</tr>
<tr>
<td>16</td>
<td>1.85</td>
<td>0.20</td>
<td>1.42</td>
<td>2.97</td>
</tr>
<tr>
<td>17</td>
<td>1.16</td>
<td>0.24</td>
<td>0.98</td>
<td>3.00</td>
</tr>
<tr>
<td>18</td>
<td>2.19</td>
<td>0.34</td>
<td>2.89</td>
<td>6.91</td>
</tr>
<tr>
<td>19</td>
<td>2.88</td>
<td>0.15</td>
<td>2.63</td>
<td>5.73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean Acc. [mm/s²]</th>
<th>Max. Acc [mm/s²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.33</td>
<td>13.27</td>
</tr>
<tr>
<td>2</td>
<td>4.37</td>
<td>20.60</td>
</tr>
<tr>
<td>3</td>
<td>4.09</td>
<td>12.75</td>
</tr>
<tr>
<td>4</td>
<td>4.25</td>
<td>24.48</td>
</tr>
<tr>
<td>5</td>
<td>1.31</td>
<td>4.37</td>
</tr>
<tr>
<td>6</td>
<td>1.60</td>
<td>6.11</td>
</tr>
<tr>
<td>7</td>
<td>7.99</td>
<td>35.71</td>
</tr>
<tr>
<td>8</td>
<td>8.08</td>
<td>16.81</td>
</tr>
<tr>
<td>9</td>
<td>5.15</td>
<td>23.89</td>
</tr>
<tr>
<td>10</td>
<td>7.60</td>
<td>17.39</td>
</tr>
<tr>
<td>11</td>
<td>7.05</td>
<td>19.57</td>
</tr>
<tr>
<td>12</td>
<td>3.97</td>
<td>9.41</td>
</tr>
<tr>
<td>13</td>
<td>4.57</td>
<td>10.73</td>
</tr>
<tr>
<td>14</td>
<td>4.21</td>
<td>13.08</td>
</tr>
<tr>
<td>15</td>
<td>6.57</td>
<td>24.23</td>
</tr>
<tr>
<td>16</td>
<td>1.86</td>
<td>4.83</td>
</tr>
<tr>
<td>17</td>
<td>1.60</td>
<td>5.62</td>
</tr>
<tr>
<td>18</td>
<td>6.09</td>
<td>21.24</td>
</tr>
<tr>
<td>19</td>
<td>4.34</td>
<td>10.59</td>
</tr>
</tbody>
</table>
B.2. Respiration Characteristics
Bibliography


[13] F. Ernst and A. Schweikard, “Forecasting respiratory motion with accurate online support
vector regression (svrpred),” *International journal of computer assisted radiology and surgery*,

respiratory motion prediction for real-time IGRT,” *Physics in medicine and biology*, vol. 53,
no. 6, p. 1651, 2008.

dynamic model in an augmented space,” *Physics in medicine and biology*, vol. 56, no. 6,

compensation for beating heart intracardiac surgery,” *The International journal of robotics

platform through the newton-euler approach,” *Mechanism and Machine Theory*, vol. 33, no. 7,

input and internal variables: an application to trajectory planning,” in *8th IFAC Symposium

[19] F. Ernst and A. Schweikard, “Predicting respiratory motion signals for image-guided radio-
therapy using multi-step linear methods (mulin),” *International Journal of Computer Assisted

Science, University of Genoa, Italy*, 2007.


sinusoidal model to lung cancer patient respiratory motion,” *Medical physics*, vol. 32, no. 9,

Hochschulverlag AG, 2011.

one, 1999.
Title of work:
Prediction Filter for Real Time Tumor Tracking Using the Treatment Table

Thesis type and date:
Master Thesis, August 2014

Supervision (ETHZ):
Dr. Marianne Schmid Daners
Prof. Dr. Lino Guzzella

Supervision (USZ):
Dr. Stephanie Lang
Dr. Stephan Klöck

Student:
Name: Alexander Jöhl
E-mail: joehla@student.ethz.ch
Legi-Nr.: 09-928-599
Semester: 10

Statement regarding plagiarism:
By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

Zurich, 16. 9. 2014: __________________________________________
Eigenständigkeitserklärung


Die Dozentinnen und Dozenten können auch für andere bei ihnen verfasste schriftliche Arbeiten eine Eigenständigkeitserklärung verlangen.

Ich bestätige, die vorliegende Arbeit selbständig und in eigenen Worten verfasst zu haben. Davon ausgenommen sind sprachliche und inhaltliche Korrekturvorschläge durch die Betreuer und Betreuerinnen der Arbeit.

**Titel der Arbeit** (in Druckschrift):

```
Prediction Filter for Real Time Tumor Tracking Using the Treatment Table
```

**Verfasst von** (in Druckschrift):

```
Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich.
```

<table>
<thead>
<tr>
<th>Name(n):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jöhl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vorname(n):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexander</td>
</tr>
</tbody>
</table>

Ich bestätige mit meiner Unterschrift:
- Ich habe keine im Merkblatt „Zitier-Knigge“ beschriebene Form des Plagiats begangen.
- Ich habe alle Methoden, Daten und Arbeitsabläufe wahrheitsgetreu dokumentiert.
- Ich habe keine Daten manipuliert.
- Ich habe alle Personen erwähnt, welche die Arbeit wesentlich unterstützt haben.

Ich nehme zur Kenntnis, dass die Arbeit mit elektronischen Hilfsmitteln auf Plagiate überprüft werden kann.

<table>
<thead>
<tr>
<th>Ort, Datum</th>
<th>Unterschrift(en)</th>
</tr>
</thead>
</table>

Bei Gruppenarbeiten sind die Namen aller Verfasserinnen und Verfasser erforderlich. Durch die Unterschriften bürgen sie gemeinsam für den gesamten Inhalt dieser schriftlichen Arbeit.