

ESTIMATION OF GROUND REACTION FORCES USING FOREARM
CRUTCHES INSTRUMENTED WITH PRESSURE SENSORS AND
ACCELEROMETERS

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

BY

ÇAĞLAR SEYLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2016

Approval of the thesis:

**ESTIMATION OF GROUND REACTION FORCES USING FOREARM
CRUTCHES INSTRUMENTED WITH PRESSURE SENSORS AND
ACCELEROMETERS**

submitted by **ÇAĞLAR SEYLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Uluç Saranlı
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Çağatay Candan
Electrical and Electronics Engineering Department, METU

Assoc. Prof. Dr. Uluç Saranlı
Computer Engineering Department, METU

Assoc. Prof. Dr. Ertan Onur
Computer Engineering Department, METU

Assoc. Prof. Dr. Çiğdem Gündüz Demir
Computer Engineering Department, Bilkent University

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU

Date:

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

Name, Last Name: ÇAĞLAR SEYLAN

Signature :

ABSTRACT

ESTIMATION OF GROUND REACTION FORCES USING FOREARM CRUTCHES INSTRUMENTED WITH PRESSURE SENSORS AND ACCELEROMETERS

Seylan, Çağlar

M.S., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Uluç Saranlı

February 2016, 82 pages

The use of crutches is critical for successful restoration of walking mobility through lower-body robotic orthosis that externally support and replace the functions of knee and hip joints. In this context, crutches can also provide useful sensory data, allowing the estimation of system state, postural stability, controller performance as well as user intention to regulate controller actions. In this thesis, we describe design and analysis of a crutch system instrumented with accelerometer and pressure sensors to estimate ground reaction forces on their point of contact, providing a well-defined sensory output for such applications. We propose an angle-dependent quadratic model to map pressure data to force components, which we identify using least-squares methods. First, we show performance of the model for specific crutch angles. Then, we evaluate the model and show the results for crutch angles other than those used for training. Finally, we present the evaluation and analysis of the model under dynamic conditions in which the crutch angle is varied by time.

Keywords: Least Squares, Optimization, Smart Crutch

ÖZ

BASINCA DUYARLI SENSÖRLER VE İVMEÖLÇERLER İLE DONATILMIŞ KANEDYEN TİPİ KOLTUK DEĞNEKLERİYLE YERYÜZÜ TEPKİ KUVVETİ TAHMİNİ

Seylan, Çağlar

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Doç. Dr. Uluç Saranlı

Şubat 2016 , 82 sayfa

Diz ve kalça eklemlerinin fonksiyonlarını harici olarak destekleyen ve onların yerini alan robotik ortez sistemleri ile yürüme bozukluklarının tedavisinde koltuk değneklerinin kullanılması oldukça önemlidir. Bu kapsamda, koltuk değneklerinden alınabilecek uygun sensör verileri sistem durumunun, postüral dengenin ve kontrolcü performansının kestirilmesine zemin hazırlayabileceği gibi kullanıcının hareket niyetinin algılanarak kontrolcüye gerekli komutların yollanmasında da kullanılabilir. Bu tez kapsamında, bu tip uygulamalarda kullanılacak, yere temas eden uç noktasındaki yeryüzü tepki kuvvetini kestirebilen basınca duyarlı sensörler ve ivmeölçerler ile donatılmış kanedyen tipi bir koltuk değneğinin tasarımı ve analizi anlatılmıştır. Basınç verilerinin tepki kuvvetlerine eşlenmesi için açığa bağlı karesel bir model önerilmiş ve model parametreleri en küçük kareler yöntemi ile öğrenilmiştir. İlk olarak modelin yalnızca parametreleri öğrenmek için veri toplanan açılardaki performansı değerlendirilmiş ve ilgili sonuçlar gösterilmiştir. Daha sonra model diğer açılarda sınanmış ve bu açılardaki performansı gösterilmiştir. Son olarak model değnek açısının zamana bağlı olarak değiştiği dinamik koşullar altında sınanmış ve sonuçları sunulmuştur.

Anahtar Kelimeler: En Küçük Kareler, Optimizasyon, Akıllı Koltuk Değneği

To you as a reader..

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Uluç Saranlı, for his continuous support and patience during my M.Sc. thesis. Throughout the work, he always guided me to the right way whenever I faced with an obstacle. Without any doubt, I can say that I cannot accomplish this thesis without his advices and vision.

I would also like to thank Scientific and Technological Research Council of Turkey (TÜBİTAK) for their support with 2210-A National Scholarship Programme for M.Sc. Students.

Moreover, I would like to thank GDS Engineering for the mechanical and electrical components they provided for this work.

Last but not least, I want to thank my family for all kinds of their support during my M.Sc. studies.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vi
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS	xviii
CHAPTERS	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Contribution	2
1.3 Organization of the Thesis	2
2 BACKGROUND AND RELATED WORK	5
2.1 Instrumented Crutch Systems and Force Sensing	5
2.2 Lower-Limb Exoskeleton Systems	8
2.2.1 Lower-limb Robotic Exoskeleton Systems for Power Augmentation	9

2.2.2	Lower-Limb Robotic Exoskeleton Systems for Rehabilitation	11
2.2.2.1	Lower-limb Robotic Exoskeleton Systems for Patients Having only Partially Disabled Limbs	11
2.2.2.2	Lower-limb Robotic Exoskeleton Systems for Patients Having Completely Disabled Limbs	13
2.3	Least-Squares Methods	16
3	DESIGN OF THE CRUTCH PLATFORM	19
3.1	Mechanical Structure	19
3.2	Electronic Design	21
3.3	The Communication Infrastructure	23
3.3.1	An Overview of the Original URB Architecture	23
3.3.2	URB Extensions and Implementation	27
3.4	The Data Acquisition System	30
4	ESTIMATION OF GROUND REACTION FORCES FROM SENSOR DATA	35
4.1	GRF Estimation in 2-Dimensions	35
4.1.1	Angle-Dependent Force Transfer Model	36
4.1.2	Data-Driven Identification of the Force Transfer Model	38
4.1.3	Identification of Angle-Dependent Sensor Delay	39
4.1.4	Estimation of GRF Vectors for Arbitrary Crutch Angles	40

4.2	Experimental Results for 2-Dimensions	41
4.2.1	The Experimental Procedure	41
4.2.2	Error Metrics	42
4.2.3	Angle-Dependent Sensor Delay Characteristic	43
4.2.4	Model Accuracy for Crutch Angles Used for Training	44
4.2.5	Model Accuracy for Arbitrary Crutch Angles	47
4.2.6	Accuracy Under Dynamic Conditions	51
4.3	GRF Estimation in 3-Dimensions	55
4.3.1	Angle-Dependent Force Transfer Model	55
4.3.2	Data-Driven Identification of the Force-Transfer Model	57
4.3.3	Identification of Angle-Dependent Sensor Delay	58
4.3.4	Estimation of GRF Vectors for Arbitrary Crutch Angles	58
4.4	Experimental Results for 3-Dimensions	61
4.4.1	The Experimental Procedure	61
4.4.2	Error Metrics	64
4.4.3	Angle-Dependent Sensor Delay Characteristic	64
4.4.4	Model Accuracy for Crutch Angles Used for Training	66
4.4.5	Model Accuracy for Arbitrary Crutch Angles	70
4.4.6	Accuracy Under Dynamic Conditions	72

5	CONCLUSION AND FUTURE WORK	77
	REFERENCES	79

LIST OF TABLES

TABLES

Table 4.1	Average, minimum and maximum errors across 20 dynamic experiments in 2D.	54
Table 4.2	Average, minimum and maximum errors across 5 fast dynamic experiments in 2D.	55
Table 4.3	Exact values of errors with corresponding standard deviation for each data pair at which data collected. \bar{E}_L values are written after divided by 100.	69
Table 4.4	Resulting errors with corresponding standard deviations using the proposed interpolation method. $Point_a$ represents point at which gain parameters and the sensor delay wanted to be approximated. $Point_i$ represents the edges of the triangle used in the interpolation. Exact positions of the points with associated labels are shown in 4.18.	71
Table 4.5	Average, minimum and maximum errors across 20 dynamic experiments in 3D.	76
Table 4.6	Average, minimum and maximum errors across 5 fast dynamic experiments in 3D.	76

LIST OF FIGURES

FIGURES

Figure 2.1 The instrumented forearm crutch system with its architecture, variables used, and functionalities of its components (Reproduced from [26]).	7
Figure 2.2 BLEEX platform worn on the body (Reproduced from [15]).	10
Figure 2.3 HAL-3 platform.	12
Figure 2.4 eLEGS platform and with some of its parts indicated (Reproduced from [34]).	15
Figure 3.1 ErgoTech model forearm crutch of FDI company on which our work is based.	20
Figure 3.2 ErgoTech model forearm crutch after modifications, which is called as the smart crutch.	20
Figure 3.3 a) The rubber tip of the ErgoTech crutch equipped with printed circular piece, b) Bottom view of the rubber tip covered with double-sided sticky tape, c) A single FSR sensor covered with cellulose tape, d) Four FSRs placed on bottom side of the double-sided sticky tape, e) Single-sided tape was placed to cover all four FSRs, f) Final assembly was covered with RTV-2 type silicon and mounted on the tip of the crutch.	21
Figure 3.4 Cross-section of the silicon shoe indicating the places of rubber tip, double-sided sticky tape, 4 FSRs, single-sided sticky tape, RTV-2 type silicon with certain dimensions.	22
Figure 3.5 Force-resistance characteristic of the FlexiForce A-301 model pressure sensor (Reproduced from [3]).	23
Figure 3.6 The sensor board that acts as a URB node.	24
Figure 3.7 An example system highly suitable for URB. s_i represents sensors, a_i represents actuators and μC_i represents microcontrollers. Protocols used across each channel with frequencies of read or write requests are also shown.	26

Figure 3.8	Byte types used in the original URB protocol with their fields. . . .	27
Figure 3.9	Byte types used in our extended URB protocol with their fields. . .	28
Figure 3.10	CPU side of the extended URB protocol for write operations. . . .	29
Figure 3.11	CPU side of the extended URB protocol for read operations.	32
Figure 3.12	Node side of the extended URB protocol.	33
Figure 3.13	The RTD CME137686LX PC104 model single-board PC which hosts the main CPU in our data acquisition system.	34
Figure 3.14	The overall structure of our data acquisition system. Protocols used across each channel with frequencies of read or write requests are also shown.	34
Figure 4.1	Experimental platform for estimating GRF vectors showing the coordinate frames of both our force platform and the accelerometer along with the placements of FSRs relative to these coordinate frames. The two pictures from different views of the same platform is shown to clearly indicate crutch tilt angles α and β	36
Figure 4.2	Scatter plot of ground reaction force samples collected during all of the 11 experiments for $\beta = 4^\circ$	42
Figure 4.3	The graph of the estimated sensor delay (in milliseconds) vs. β (in degrees). For each β , estimations from 11 experiments were averaged. The shaded region shows the standard deviation range.	44
Figure 4.4	Percentage RMS errors in the estimates of F_z (top graph) and F_y (bottom graph) vs. β graph. For each β , resulting errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.	45
Figure 4.5	Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph. For each β , resulting errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.	46
Figure 4.6	Errors in the estimates of F_z (top graph, in percentage), and F_y (bottom graph, in percentage) vs. β (in degrees) in cartesian coordinates graph at interpolated angles for $\Delta\beta = 4^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.	48

Figure 4.7 Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph at interpolated angles for $\Delta\beta = 4^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation. 49

Figure 4.8 Percentage errors in the estimates of F_z (top graph), and F_y (bottom graph) vs. β (in degrees) in cartesian coordinates at interpolated angles for $\Delta\beta = 8^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged. Shaded regions indicate standard deviation. 50

Figure 4.9 Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph at interpolated angles for $\Delta\beta = 8^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation. 51

Figure 4.10 Graphs showing estimation results in cartesian coordinates for one of the dynamic experiments. The graph at the top shows β as a function of time. Black trajectories in the second from top and the second from bottom graphs show F_y and F_z , respectively. Blue trajectories in the second from top and the second from bottom graphs show \hat{F}_y and \hat{F}_z , respectively. The middle and bottom graphs show the difference between the ground-truth and estimated forces in y and z axis, respectively. 53

Figure 4.11 Graphs showing estimation results in polar coordinates for one of the dynamic experiments. Top graphs shows β as a function of time. Middle graphs shows the angle between the estimated GRF vector and ground-truth GRF vector. Bottom graph shows ratio of the norm of the estimated GRF vector to ground-truth GRF vector. 54

Figure 4.12 Three different crutch angle pairs (α_1, β_1) , and (α_2, β_2) , (α_3, β_3) at which gain matrices and the sensor delay are identified. (α, β) is the angle pair at which we want to approximate gain matrices and the sensor delay. A_1, A_2 , and A_3 represents the areas of the small triangles on which they are written. 59

Figure 4.13 (α, β) angle pairs at which data were collected. Points with ‘x’ represent data used for training the model for dynamic experiments. Points with ‘o’ represent data used for testing the method to extend to model to arbitrary crutch angles. Points with both ‘x’ and ‘o’ represent data used for both purposes. 62

Figure 4.14 Scatter plot of ground reaction force samples collected during all of the 17 experiments for $(\alpha = 2^\circ, \beta = 2^\circ)$	63
Figure 4.15 The graph of the estimated sensor delay (in milliseconds) vs. (α, β) (in degrees) for the first quadrant. For each (α, β) , estimations from 17 experiments were averaged.	65
Figure 4.16 Percentage errors in the estimates of F_z (top graph), F_y (middle graph), and F_x (bottom graph) vs. (α, β) . For each (α, β) , resulting errors for 17 experiments are averaged arithmetically.	67
Figure 4.17 Errors in terms of the angles between the ground-truth GRF vector and the estimated GRF vector (bottom graph, in degrees) and ratio of norm of the ground-truth vector and the estimated GRF vector (bottom graph) vs. (α, β) . For each (α, β) , resulting errors for 17 experiments are averaged arithmetically.	68
Figure 4.18 Exact positions of (α, β) pairs with associated labels used to test the proposed interpolation method.	70
Figure 4.19 Resulting Delaunay triangulation of the points used in the learning process to use the proposed interpolation method.	73
Figure 4.20 Graphs showing estimation results in cartesian coordinates for one of the dynamic experiments. The two topmost graphs show α and β as a function of time, respectively. Black trajectories in the third, fifth, and seventh from top graphs show $F_x, F_y,$ and $F_z,$ respectively. Blue trajectories in the same graphs show $\hat{F}_x, \hat{F}_y,$ and $\hat{F}_z,$ respectively. The fourth, sixth, and eighth graphs show the difference between the ground-truth and estimated forces in x, y, and z axis, respectively.	74
Figure 4.21 Graphs showing estimation results in polar coordinates for one of the dynamic experiments.	75

LIST OF ABBREVIATIONS

2D	2-Dimensions
3D	3-Dimensions
API	Application Programming Interface
CPU	Central Processing Unit
DoF	Degree of Freedom
FSR	Force Sensitive Resistor
GRF	Ground Reaction Force
PC	Personal Computer
PDA	Portable Digital Assistant
SCI	Spinal Cord Injured
RMS	Root Mean Squared
URB	Universal Robot Bus

CHAPTER 1

INTRODUCTION

1.1 Motivation

Today, many spinal cord injury (SCI) and stroke patients are obliged to use wheelchair. Sitting in such chairs for long hours without moving lower-limbs threatens health of SCI and stroke patients in many aspects such as increased coronary artery disease risk [17], increased osteoporosis risk [10], and increased risk of malfunctions on hormonal and endocrine system [22]. Thanks to advances on sensory, actuating, and microprocessing technologies, life quality of patients having such limb immobilities have been increasing as usage of robotic exoskeletal orthosis systems has become widespread [11].

In previous years, lower-limb robotic orthosis systems designed for SCI and stroke patients were successfully controlled by teleoperation. Nonetheless, the state-of-the-art has been shifting towards completely leaving control of such systems to the wearer. Patients able to move their lower limbs enough to carry the exoskeleton system do not need any crutch or cane while walking with the help of the system. However, patients who cannot move their lower-limbs in any way, as in the case of a stroke, inevitably need the use of a crutch or a cane to balance himself/herself because such systems usually have only one degree of freedom (DoF) at the hip and knee joints. Our motivation in this study is to design, analyse and evaluate an instrumented crutch system to be used along with lower-limb exoskeleton systems that measures ground reaction forces (GRF) at its contact point with the ground in real-time. We estimate that such a system will not only help the wearer to keep his/her balance but also will extract

triggering moments of gait events and estimate the wearer's motion intention, allowing the wearer to walk in a much more natural way than in the case of teleoperation control.

1.2 Contribution

Our first contribution in this work include the design of a new instrumented forearm crutch platform using low-cost pressure sensors and accelerometers to support the estimation of ground reaction forces. Secondly and more importantly, we propose a simple yet accurate quadratic model with delay for the relation between sensor outputs and the GRF, whose gain parameters we carefully identify through systematic experiments and ground-truth GRF measurements. Firstly, focusing on only sagittal plane forces and motion, we use systematic, statistically valid fitting and validation experiments to show that for a given crutch angle linear least-squares identification methods can be used to learn a model that can predict GRF components with percentage errors less than 10%. We also show that models for specific training angles can be generalized to a continuous range of crutch angles through interpolation, while still keeping estimation errors below 10%. Then, we investigate the performance of our model under dynamic conditions with time-varying crutch angle and GRF vectors. Finally, we generalize our investigation in the sagittal plane to 3 dimensional GRF estimation and give the results with brief analysis and discussion.

1.3 Organization of the Thesis

This thesis consists of 5 chapters. Chapter 1 states motivation of our study, our contributions to the area and describes organization of this thesis. Chapter 2 briefly introduces lower-limb robotic exoskeleton systems with examples of these systems which we consider significant. Chapter 2 also introduces instrumented crutch systems with possible sensors to measure ground reaction forces and tilt angle and describes some of the instrumented crutch systems in the literature. Moreover, possible methods for parameter estimation are given in this chapter. Chapter 3 briefly describes our instrumented crutch design in three subsections by firstly introducing the mechanical

structure, then electronic design and finally the communication infrastructure along with the data acquisition system. Chapter 4 describes our method to map signals obtained from the sensors to ground reaction forces, gives the experimental results of the method, and evaluates the results obtained. Chapter 5 summarizes the study within the scope of this thesis and states the work to be done in the future.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Instrumented Crutch Systems and Force Sensing

In rehabilitation and gait training, correct usage of crutches and canes by lower-limb disabled patients is critical. While correct usage of canes and crutches lead to better recovery [21], using them incorrectly may even worsen the situation [29]. Thus, instrumented crutch systems are used for gait training to monitor and track their usage whether they are used in the correct way or not. Although usage of the instrumented crutch or cane systems has not become widespread to capture motion primitives of patients and to control lower-limb robotic exoskeleton systems, investigating them in terms of ground reaction forces and tilt angle estimation is important for our study. This subsection is devoted to explaining designs of some instrumented crutch or cane systems in the literature.

In [27] the design of an instrumented cane for gait analysis for rehabilitative purposes was described. The design was quite simple, using only two kinds of sensors: strain gages, and FSR sensors. Two strain gages were placed on the metal body of the cane to measure the bending of the cane. Only one FSR sensor was placed at the tip of the cane to measure axial forces. Signals obtained from these sensors were processed in a Motorola 68HC11 microprocessor. Metrics used in the gait analysis were the peak force, contact duration, and force-time integral. If the applied weight on the cane surpassed 25% of the body weight, the user was informed with audio feedback.

Another instrumented cane designed for gait analysis was described in [23]. A six-channel AMTI force transducer was placed on the body of the cane, a few centimeters

above the tip to measure ground reaction forces at the ground contact point. Motion of the cane was tracked using a Vicon 370 motion capture system, consisting of 6 cameras and three reflective markers placed on various points on the cane. With this system, ground reaction forces at the tip of the cane were measured and the motion of the cane was very accurately tracked. However, such systems are very expensive and might be unsuitable to be used outside laboratory environments. In this case, they were used to obtain ground-truth data rather than prototyping.

Another study [30] introduced a virtual environment locomotor system with an instrumented cane to be used for gait analysis and biofeedback for rehabilitative purposes. The lower section of the cane had a ball joint, with an AMTI MC25-500 tri-axial force transducer attached with which the ground reaction forces could be measured. The cane could be attached to the left or the right side of the treadmill. The system was used on 5 stroke patients and 5 healthy control subjects. It was observed that the gait speed improved and gait variability decreased among stroke patients who trained with the cane.

In [26], the authors describe an instrumented forearm crutch system that was developed in the University of Southampton in collaboration with physiotherapists in the Southampton General Hospital. The system was intended to be used by physiotherapists in a laboratory environment to investigate whether patients can correctly use the cane. For this purpose, the system was expected to provide measurements and feedback for weight applied on the cane, cane tilt angle, and hand position on the grip to therapist and the user. The system consisted of two instrumented forearm crutches, one of which was slave and the other one the master, as well as an external computer. An FSR sensor was placed on the pole of each crutch to measure applied weight on them. Tilt angles of the crutches were measured with STmicro LIS3LV02DL MEMS tri-axial accelerometers. Hand position on the grip was measured with membrane potentiometers embedded inside the grip. The slave crutch sent its measurements to the master crutch through a wireless communication module. The master crutch processed slave crutch measurements along with its own measurements and provided audio feedback to the user on whether the crutches were used in the correct way or not. Moreover, both the measurements of the slave crutch and the master crutch were sent to an external computer via wireless communication modules for real-time moni-

toring and storage for further analysis. Figure 2.1 shows the architecture of this crutch platform with variables used in the system, and the functionalities of each component.

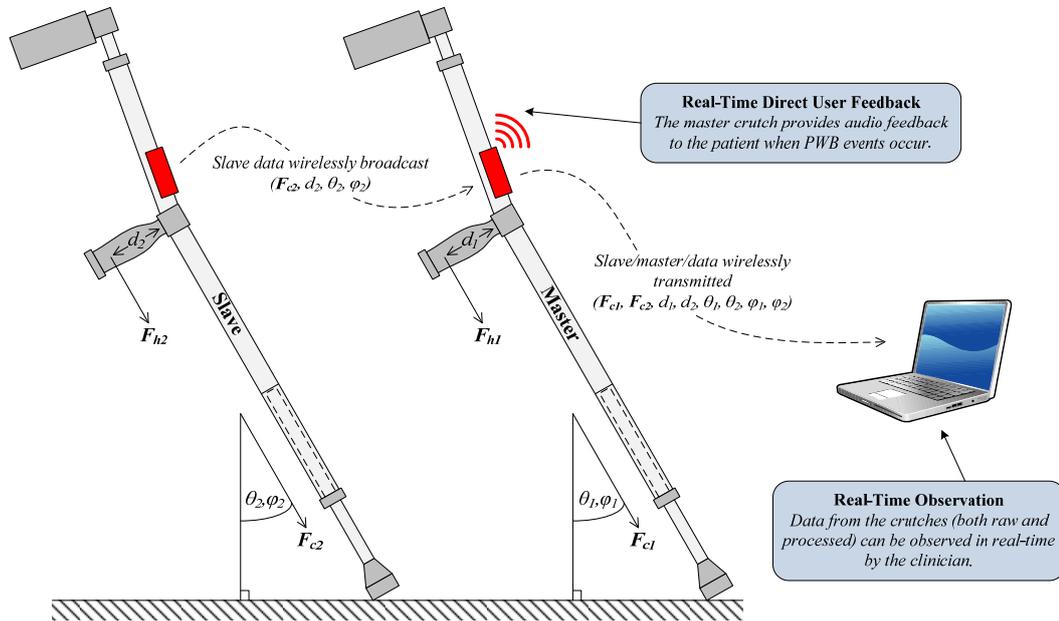


Figure 2.1: The instrumented forearm crutch system with its architecture, variables used, and functionalities of its components (Reproduced from [26]).

The iWA smart cane system was also developed for similar purposes [9]. With this instrumented smart cane, monitoring usage of the cane and providing feedback to the patient and the therapist were aimed. The system consisted of a smart cane instrumented with sensors and electronic equipment, a portable digital assistant (PDA) unit, and an external computer. A two DoF IDG-500 model gyroscope from InvenSense Inc. was placed on the body of the cane to measure angular velocity. In addition, a three-axis MMA7260Q model accelerometer from Freescale Semiconductor Inc. was placed on the body to measure linear acceleration. The orientation of the cane was computed by giving measurements from both sensors as inputs to a Kalman filter. Body weight exerted along the cane axis is measured by using 1-DoF SLC13/0250 model load cell from RDP Electronics Ltd. which was placed just above the tip of the cane. Data obtained from the sensors were sent to the PDA unit via a Bluetooth module. The PDA unit provides visual feedback about the usage of the cane and shows statistics such as activity duration, and number of steps in the action to the user. For further storage and analysis, data saved in the PDA could be transferred to the external computer via a USB cable.

Some of other instrumented crutch systems designed for monitoring gait training of a patient by means of estimating GRF, load bearing and tilt angle are described in [25, 33, 35].

In a related but different study, ground reaction forces at the contact points of feet with the ground of legged robots moving in high speeds are computed in real time [7]. To sense the ground reaction forces, 9 barometric pressure sensors arranged as a 3-by-3 matrix were embedded in a polyurethane rubber. Artificial neural networks approach was used to map the 9 input signals obtained from the barometers to 3 output signals, namely F_x , F_y , F_z . Before using the footpads in the field, the neural networks are trained by attaching the footpads to a CNC milling machine and generating various vertical and horizontal forces with the machine. According to the experimental results, the method estimates vertical (F_z) and shear (F_x , F_y) forces quite accurately.

2.2 Lower-Limb Exoskeleton Systems

Robotic exoskeletons are wearable, electromechanical, mobile robotic systems designed to help people move their limbs by replacing some or all of the joints of its wearer with actuators. They are used both for power augmentation and rehabilitation purposes. Exoskeletons are worn by healthy people for power augmentation purposes such as carrying heavy loads, military applications, and fire fighting. On the other hand, exoskeletons can also be worn in rehabilitation applications by people who cannot move or only partially move at least one of their limbs. Such applications include diagnosis of gait disorder, gait training in physiotherapy, gait analysis, restoring movement ability to a completely disabled limb.

Exoskeletons can be categorized based on the part of the body over which they are worn as lower limb exoskeleton systems, upper-limb exoskeleton systems, and full-body exoskeleton systems. The instrumented crutch in this work is intended to be used to control lower-limb exoskeleton systems. Consequently, lower-limb exoskeleton platforms and control methods will be described in this section.

2.2.1 Lower-limb Robotic Exoskeleton Systems for Power Augmentation

In the scope of this work, by lower-limb robotic exoskeleton systems for power augmentation, we refer to lower-limb robotic orthosis systems intended to be used by healthy people to enhance strength of their muscles, not to be used by people having immobilities with their lower extremities. Power augmenting exoskeletons commonly used for applications requiring carrying of heavy loads such as military applications, firefighting, rescuing etc. [39]. We summarize two of such exoskeletons in terms of their platforms and control approaches.

BLEEX

The first autonomous lower-limb robotic exoskeleton system designed for power augmentation applications was BLEEX [6]. It was specifically designed to help the wearer carry heavy loads such as weaponry, food, rescue equipment etc.. The primary goal was to make the platform highly maneuverable, durable, robust, and lightweight. The project started out in 2000 at UC Berkeley and was funded by the Defense Advanced Research Projects Agency (DARPA).

BLEEX consists of two legs, a backpack to carry the payload, and a power supply unit mounted just below the backpack [40]. Each leg consists of three parts, the thigh, the shank, and the foot, and has seven Degrees of Freedom (DoF), three at the hip (abduction/adduction, extension/flexion, rotation), one at the knee (extension/flexion), and three on the ankle (dorsiflexion/plantar flexion, abduction/adduction, rotation). Although BLEEX has 7 DoFs, DoFs of extension/flexion at the ankle, knee, hip and abduction/adduction at the hip are actuated for power saving purposes. Figure 2.2 shows BLEEX platform worn on the body.

The control methodology was designed to minimize the forces between the exoskeleton and the wearer when the wearer attempts to make a move [20]. For this purpose, instead of measuring the forces on various contact points by using sensors, the controller uses the total force exerted on the exoskeleton in a positive feedback controller.



Figure 2.2: BLEEX platform worn on the body (Reproduced from [15]).

ROBOKNEE

RoboKnee is another lower-limb exoskeleton system designed for power augmentation [31]. It enhances strength and endurance of knee muscles, enabling its wearer to be able to carry heavy loads. Using it with only one knee, the wearer can do deep knee bend movements multiple times without being exhausted while carrying a 60 kg backpack.

RoboKnee consists of a knee brace, a series elastic actuator between the upper and lower parts of the knee brace to actuate the knee joint, two load cells placed under rigid-bottom shoes to measure vertical ground reaction force under the shoes, and necessary electronic equipment. RoboKnee has only one DoF (extension/flexion) at the knee joint.

High transparency was aimed in the design of RoboKnee, wherein, user intention was estimated by using the knee joint angle and vertical ground reaction forces at the contact points of the shoes with the ground. By using the ground reaction force vector

and the knee joint angle, the torque required to bring the knee to a static situation was estimated. The required torque was then generated through positive feedback force amplification.

Other than these, studies on power augmenting lower-limb exoskeleton include [?,31, 38]. For the sake of not rambling the topic, we do not summarize these exoskeletons in a detailed way.

2.2.2 Lower-Limb Robotic Exoskeleton Systems for Rehabilitation

In this subsection, lower-limb robotic exoskeleton systems designed for rehabilitation purposes will be described and exemplified. We divide lower-limb robotic exoskeleton systems for rehabilitation purposes into two subcategory further as those designed for patients having only partially disordered limbs, and those designed for patients having completely disabled limbs. We have needed to do such a categorization because control approaches of the two type differs radically from each other.

2.2.2.1 Lower-limb Robotic Exoskeleton Systems for Patients Having only Partially Disabled Limbs

Patients having only partially disabled limbs can still move their disabled limbs weakly. Elderly people and some of the SCI patients can be considered in this category. Most of the control approaches of robotic exoskeleton systems designed for such people uses myoelectric signals and moments at the limb joints to estimate the user intention. We summarize two of such exoskeletons, one using myoelectric signals, and one using joint moments to control the device in terms of their platform designs and control approaches.

HAL

One of the most widely known and well developed robotic exoskeleton systems is the Hybrid Assistive Limb (HAL) platform. It was jointly developed by Tsukuba University of Japan and the Cyberdine robotics company to expand physical capabilities of humans, particularly people with physical disabilities. Its first prototype

was developed in late 1990's, and was quite a bulky exoskeleton, being far from practical. Nevertheless, many improvements were done on it since then. Today, HAL has two different versions in operation: HAL-3, which is a lower-limb exoskeleton system [19], and HAL-5, which is a full-body exoskeleton system [36].

The HAL-3 platform consists of two legs, a backpack to carry the power supply unit and electronic equipment, shoes instrumented with force sensors, encoders at the hip and knee joints and electrodes placed on the thighs to measure myoelectric signals. Each leg has three joints at the hip, knee, and ankle and both the hip and the knee joints are actuated and have a single DoF for extension and flexion. Figure 2.3 shows the HAL-3 platform.

Control of the HAL-3 platform relies on estimating the intention of the wearer. Myoelectric signals are processed in an online manner and along with the angles of the hip and knee joints and the floor reaction forces, the motion intention of the wearer is estimated and an associated motion trajectory is computed. After that, actuators realize the motion, following the computed trajectory.



Figure 2.3: HAL-3 platform.

LOKOMAT

Lokomat is an example of lower-limb robotic exoskeleton systems designed for rehabilitation purposes [8]. Treadmill walking improves the treatment process for Spinal Cord Injury (SCI) and stroke patients. Originally, Lokomat was designed in late 1990s to automate the treadmill walking treatment process and was successfully used in several rehabilitation centres successfully. After a few years, in early 2000s, it was proposed to use Lokomat as a gait disorder diagnosis and gait research tool [18].

The overall Lokomat platform consists of an exoskeleton, a suspension system to support the wearer, a treadmill to walk over, a computer to execute safety and control tasks, and another computer for the supervisor physiotherapist. Each leg has two DoFs, one at the hip (extension/flexion) and one at the knee (extension/flexion). Force sensors are placed at the upper and lower limb segments to measure forces exerted on these segments. Moreover, a potentiometer and a position controller are placed at each joint.

Lokomat's control algorithm of seeks to minimize interaction torques between the exoskeleton and the wearer in two stages. In the first stage, forces acting on the exoskeleton segments are measured with force sensors. In the next stage, gait patterns to minimize these interaction forces are generated in an online fashion. The platform then follows these gait patterns.

Other than these two lower-limb robotic exoskeleton systems, [12, 13] and [5, 28] describe designs of systems with control methods based on myoelectric signals and joint moments, respectively.

2.2.2.2 Lower-limb Robotic Exoskeleton Systems for Patients Having Completely Disabled Limbs

Control of lower-limb robotic exoskeleton systems for patients having completely disabled limbs are more difficult than control of the systems in the other two category because no myoelectric signal can be detected from disabled limbs of such people. Also, as such people cannot move their disabled limbs in any way, no joint moment at the disabled limbs can be detected. Note that, such people have to use crutches or

canes to balance themselves as lower-limb exoskeleton systems do not have enough DoFs at the hip and knee joints. In this case, motion intention should be recognized by other means such as by using motion of the crutches or canes, by using the signals obtained from the sensors on the crutches or canes if they are instrumented with sensors, or by using the motion of healthy limbs. In this subsection, we provide two well developed systems in the literature for the usage of patients having completely disabled lower-limbs and summarize them in terms of their platform designs and control methods.

eLEGS

eLEGS [34] is a lower-limb robotic exoskeleton system designed by Berkeley Bionics to help patients who completely lost their ability not only to walk but also to realize standing up and sitting down motions. It was unveiled in 2011, with the goal of being used in rehabilitation centers under medical supervision.

eLEGS consists of a backpack to carry battery and computer units, two legs having joints at the hip, knee, and ankle, braces on thighs, shins, sacrum, and feet to align the exoskeleton with the lower-body properly, and two crutches to help the wearer balance himself. Only hip and knee joints are actuated and they have one DoF for extension and flexion. Potentiometers and encoders are placed at the hip and knee joints to measure joint angles. Two inertial measurement units (IMU) consisting of accelerometer and gyroscope units are placed on the arms to measure arm angles. In addition, force sensors are embedded into the exoskeleton shoes to measure the GRF vector under the shoes. Moreover, crutches are instrumented with force sensors to measure the body weight falling on the crutches.

Motion of the eLEGS platform is based on four states, which are *left swing*, *right swing*, *right double stance*, and *left double stance*. In any given moment, eLEGS can be in one of these states. The Control component of eLEGS consists of three layers: the top layer decides on the state that the exoskeleton should be in, the middle layer decides on the trajectory that should be generated, and the bottom layer ensures that the joints follow a particular trajectory through a PID controller. The intention of the wearer is estimated with data obtained from all sensors and a high level control decision is taken by the top layer. Subsequently, the required motion is realized by

the bottom layer based on the trajectory generated by the middle layer. Figure 2.4 shows the eLEGS platform and identifies some of its components.

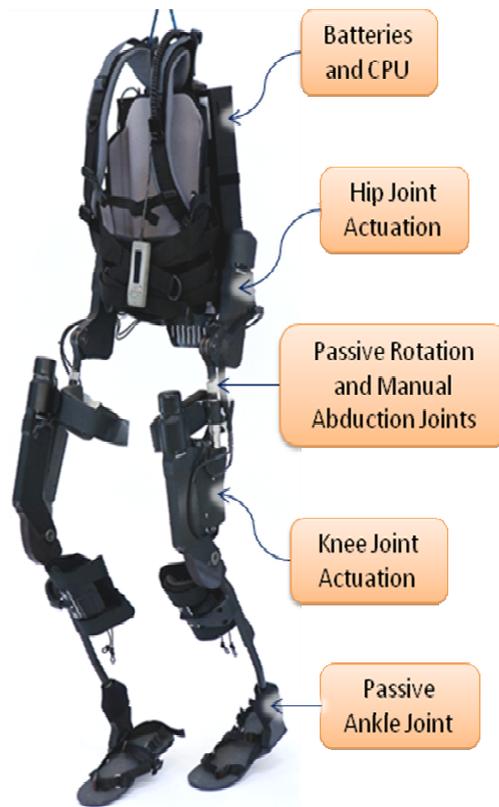


Figure 2.4: eLEGS platform and with some of its parts indicated (Reproduced from [34]).

Wearable Gait Measurement System

The wearable gait measurement system was developed in Tsukuba University of Japan and was unveiled in 2014 [16]. It was designed to help patients who can move only one of their legs normally but the other leg is completely disordered as in the case of hemiplegia. Up to now, the system was tested only on healthy people.

The wearable gait measurement system consists of one leg of the robot suit HAL-3, with IMUs on the thigh and shank, a smart forearm crutch instrumented with an IMU and FSR sensors, and shoes instrumented with FSR sensors. The computation is done on the smart cane so it has a wireless module to send necessary commands to the robotic leg.

The control algorithm of the wearable gait measurement system is based on the algo-

rithms described in [37]. In summary, the generation of the reference trajectory for the robotic leg is based on principal component analysis (PCA) on inter-joint coupling of upper and lower limbs. While motion of lower limbs is captured by the IMUs on the robotic leg, the motion of upper limbs is captured by the IMU on the instrumented crutch. In addition, FSR sensors under the tip of the crutch and the shoe provides ground contact information to the controller.

2.3 Least-Squares Methods

Least-squares estimation refers to an approach in regression analysis seeking to find approximate solutions to overdetermined systems, i.e., systems in which the number of equations are greater than the number of unknowns. There are many different kinds of least-squares procedures. The difference between these procedures originates from the structure of the mathematical model and the criteria to be optimized [24]. In this context, we will briefly introduce the linear least-squares method in which the output of the mathematical model depends linearly on the unknown parameters and the data.

Consider an overdetermined linear system,

$$F_i = \sum_{j=1}^n P_{ij}k_j, \quad (i = 1, 2, \dots, m), \quad (2.1)$$

having m linear equations each with n unknown coefficients k_1, k_2, \dots, k_n with the assumption $m > n$.

The system can be rewritten compactly in the matrix form as

$$\mathbf{F} = \mathbf{M}\mathbf{u}, \quad (2.2)$$

where

$$\mathbf{F} := \begin{bmatrix} F_1 & F_2 & \dots & F_m \end{bmatrix}^T,$$

$$\mathbf{M} := \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots \\ P_{m1} & P_{m2} & \dots & P_{mn} \end{bmatrix},$$

$$\mathbf{u} := \begin{bmatrix} k_1 & k_2 & \dots & k_n \end{bmatrix}^T.$$

Even though this system has no exact solution, we can try to find an optimal solution by minimizing the difference between the true output and the estimated output. Let us define the error between these outputs as a function of parameters as

$$E(\mathbf{u}) := \|\mathbf{F} - \mathbf{M}\mathbf{u}\|^2 \quad (2.3)$$

which can be further written as

$$\mathbf{E}(\mathbf{u}) = (\mathbf{F} - \mathbf{M}\mathbf{u})^T(\mathbf{F} - \mathbf{M}\mathbf{u}) = \mathbf{F}^T\mathbf{F} - \mathbf{u}^T\mathbf{M}^T\mathbf{F} - \mathbf{F}^T\mathbf{M}\mathbf{u} + \mathbf{u}^T\mathbf{M}^T\mathbf{M}\mathbf{u}.$$

The fact that $\mathbf{u}^T\mathbf{M}^T\mathbf{F} = \mathbf{F}^T\mathbf{M}\mathbf{u}$ leads us to

$$\mathbf{E}(\mathbf{u}) = \mathbf{F}^T\mathbf{F} - 2\mathbf{u}^T\mathbf{M}^T\mathbf{F} + \mathbf{u}^T\mathbf{M}^T\mathbf{M}\mathbf{u}. \quad (2.4)$$

Taking the derivative of Equation 2.4 with respect to \mathbf{u} gives us

$$\frac{\partial \mathbf{E}(\mathbf{u})}{\partial \mathbf{u}} = -2\mathbf{M}^T\mathbf{F} + 2\mathbf{M}^T\mathbf{M}\mathbf{u}. \quad (2.5)$$

Equating Equation 2.5 to zero and solving it for \mathbf{u} gives us the parameters that best approximate the data \mathbf{F} , and often denoted by $\hat{\mathbf{u}}$ as

$$\hat{\mathbf{u}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}. \quad (2.6)$$

If the output in the mathematical model linearly depends on the parameters, Equation 2.6 gives the optimal estimation of the parameters. Nevertheless, it should be noted that if the data is too noisy and the noise is not cancelled properly prior to learning the parameters, the model can be fit into the noise with the learned parameters.

CHAPTER 3

DESIGN OF THE CRUTCH PLATFORM

In this section, the design of the crutch platform with the system used for data acquisition will be briefly described. Firstly, sensor placement and manufacturing process of the silicon shoe in which the FSRs are embedded will be described. Then, electronic components used in the smart crutch platform will be described. After that, communication infrastructure, namely, small scale network protocol used to ensure coordination of the components and provide communication between these components in the smart crutch platform will be explained in a detailed way. Finally, data acquisition system used to collect data for training and testing the GRF estimation method will be described.

3.1 Mechanical Structure

We used ErgoTech forearm crutch model of FDI company, shown in Figure 3.1, as the basis of our smart crutch design. Three main modifications have been made to the forearm crutch: The addition of a silicon shoe at the bottom of the crutch, attachment of a sensor board on the body of the crutch, and the development of a software to establish communication between the crutch and the CPU. The smart crutch with these modifications can be seen in Figure 3.2.

The first modification, the silicon tip (white shoe at the tip of the crutch in Figure 3.2) was introduced to prevent FSRs from being damaged during contact with the ground, and to help distribution of reaction forces from the ground evenly over the FSRs. Since the surface of rubber tip which contacts with the ground is not planar,



Figure 3.1: ErgoTech model forearm crutch of FDI company on which our work is based.



Figure 3.2: ErgoTech model forearm crutch after modifications, which is called as the smart crutch.

a circular and planar piece was produced with a 3D printer and placed at the bottom of the rubber tip. After that, double-sided sticky tape was placed at the bottom of the rubber tip. This tape prevents the FSRs from being damaged under shear forces, acting like a soft barrier between the hard surface of the printed circular piece and fragile FSRs. When the RTV-2 type silicon touches the FSR directly, it can damage the sensor. To prevent this, the FSRs were covered with cellulose tape to prevent them from touching the silicon. Subsequently, the 4 FSRs were placed on the double-sided tape. To ensure that the FSRs stay still and preserve their placements in the presence of shear forces, single sided sticky tape was draped over them. Finally, the tip assembly augmented with FSRs and tapes was placed in a cylindrical mould and RTV-2 type silicon was poured into the mould. After the silicon is frozen, the final assembly was affixed at the tip of the crutch. The process is summarized in Figure 3.3. The cross-section of the final assembly can also be seen in Figure 3.4.

Inside the black box (visible on the body of the platform shown in Figure 3.2), there is a sensor board performing initial processing of data coming from FSRs and accelerometer (The design of this sensor board is presented in Section 3.2.). The purpose of the black box is to protect the sensor board from potential physical damages.

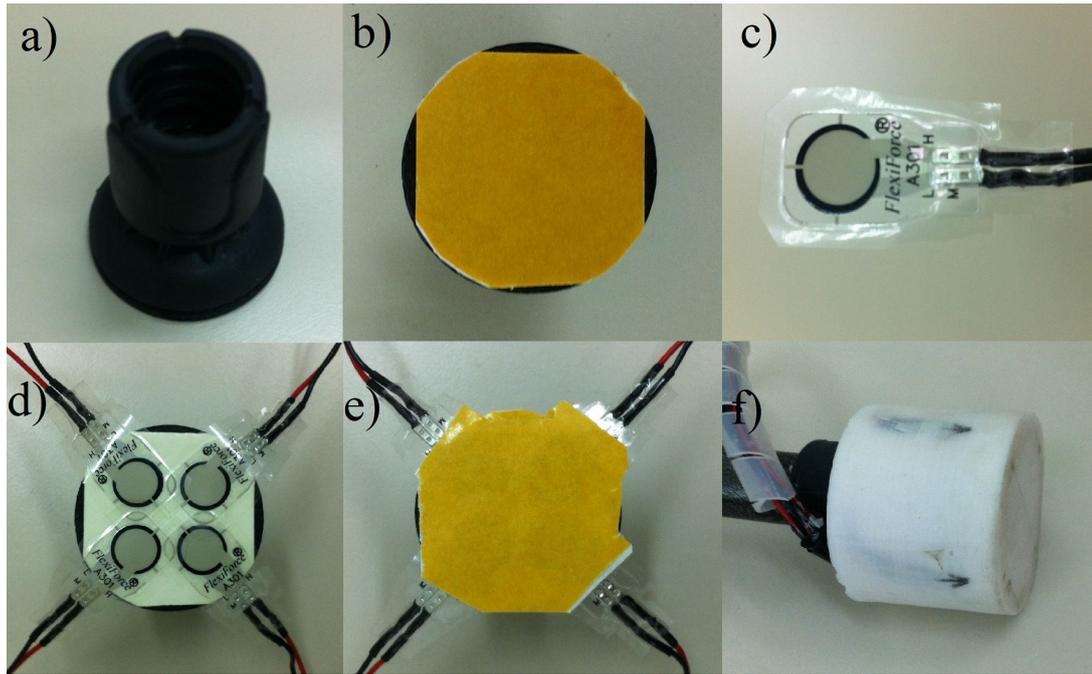


Figure 3.3: a) The rubber tip of the ErgoTech crutch equipped with printed circular piece, b) Bottom view of the rubber tip covered with double-sided sticky tape, c) A single FSR sensor covered with cellulose tape, d) Four FSRs placed on bottom side of the double-sided sticky tape, e) Single-sided tape was placed to cover all four FSRs, f) Final assembly was covered with RTV-2 type silicon and mounted on the tip of the crutch.

3.2 Electronic Design

To sense the GRF at the contact point of the smart crutch with the ground, we used the FlexiForce A-301 model pressure sensors. These sensors can sense forces up to 445N, which meets our requirements. Its force-resistance and force-conductance characteristics can be seen in Figure 3.5 which was reproduced from the FlexiForce user manual [3].

To measure the orientation of the smart crutch, an ADIS16209 dual-axis accelerometer manufactured by Analog Devices was used. Detailed descriptions for the functionality of this sensor can be found in [1]. Acceleration readings from this sensor were converted to degrees for inclination degrees of freedom afterwards.

A central CPU was used to process data obtained from the four FSRs and the ac-

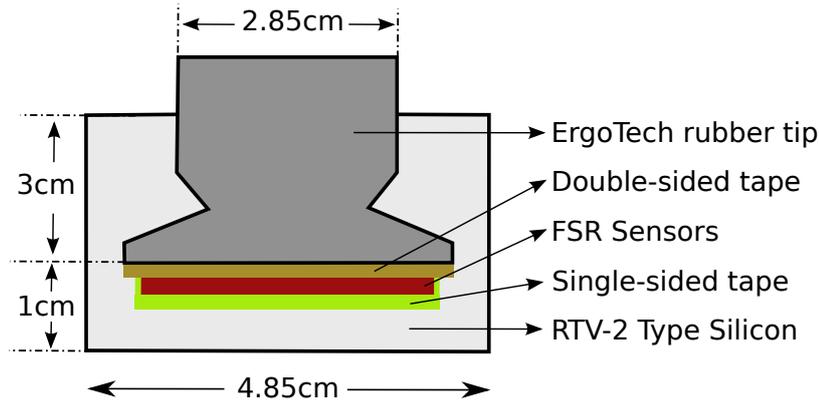


Figure 3.4: Cross-section of the silicon shoe indicating the places of rubber tip, double-sided sticky tape, 4 FSRs, single-sided sticky tape, RTV-2 type silicon with certain dimensions.

celerometer to estimate GRF at the contact point of the smart crutch and ground. However, this CPU is not directly responsible for acquiring data from the FSRs and the accelerometer. Instead, a sensor board of our design was used to collect data from the FSRs and the accelerometer at a desired frequency and to send them to the CPU upon a request. Responsibilities of this sensor board can be summarized as follows:

1. Obtaining data from the FSRs and performing analog to digital conversion.
2. Communicating with the accelerometer and obtaining its readings.
3. Sending acquired data to the CPU upon request. In other words, acting as a node in the Universal Robot Bus (URB) architecture, which will be explained briefly in Section 3.3.

There are 8 analog inputs on the sensor board four of which were used to interface with the FSRs. The ADIS accelerometer connects to the sensor board through dedicated connectors. A MSP430F2274 model microcontroller from Texas Instruments was used to convert analog data obtained from the FSRs to digital data, to obtain acceleration data by communicating with the accelerometer, and to send these data to the CPU on request via RS485 physical layer network protocol. This is a commonly used microcontroller from Texas Instruments, featuring 16-bit architecture and a 16 MHz clock frequency [4]. Figure 3.6 shows a top view of the sensor board, indicating

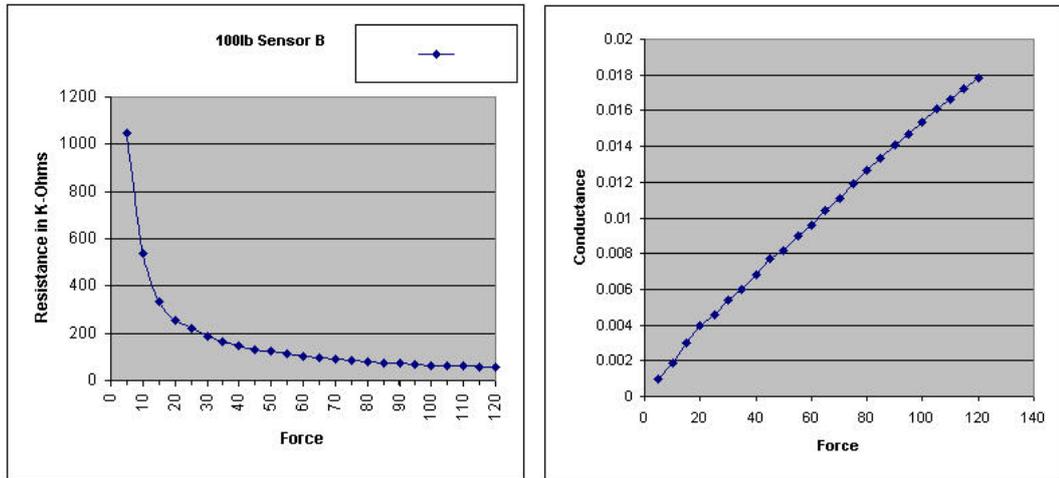


Figure 3.5: Force-resistance characteristic of the FlexiForce A-301 model pressure sensor (Reproduced from [3]).

placements of the analog and accelerometer pins, power, RS485, and programming interfaces, and the microcontroller.

Finally, we used a PC104 single-board computer as the main CPU. This CPU unit was necessary for our data collection experiments, and will not be necessary for standalone operation of the smart crutch. The CPU unit is presented in Section 3.4.

3.3 The Communication Infrastructure

The communication infrastructure we use for the overall system, including the experimental platform, is based on the URB architecture [32]. However, in this work, the original URB architecture was modified in a few different ways. In this section, the URB architecture will first be introduced. After that, our modifications on the URB architecture and its implementation will be explained.

3.3.1 An Overview of the Original URB Architecture

URB is a small scale network architecture specifically designed to establish communication and ensure coordination between a central, powerful CPU and different types of sensors and actuators. URB assumes the presence of nodes to which several

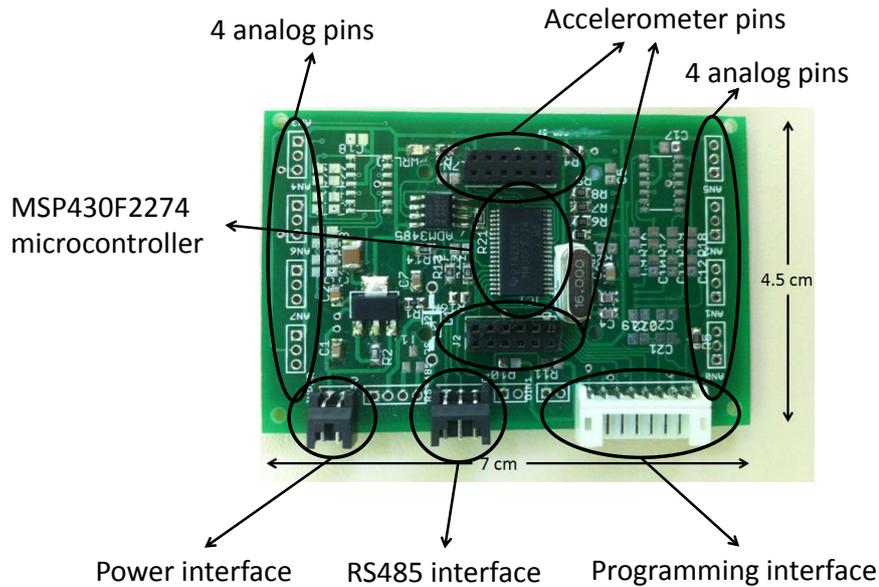


Figure 3.6: The sensor board that acts as a URB node.

sensors and/or actuators are attached, and that there is a single CPU to which all the nodes in the system are attached. In such a system,

1. Different kinds of sensors and/or actuators can be used,
2. Nodes can collect data from several sensors at different frequencies. For example, a node can collect data from an FSR at 500 Hz, while it can collect data from another sensor at 300 Hz.
3. The physical layer network protocol between a node and the CPU can be different from the physical layer network protocol between another node and the CPU. For example, while the protocol between a node and the CPU can be RS485, the protocol between another node and the CPU can be RS232 at the same time,
4. The URB framework provides real-time performance guarantee,
5. URB provides a uniform application programming interfaces (API) to develop nodes and CPU applications more rapidly and modularly.
6. URB provides mechanism for automatic synchronization of different kinds of sensors and/or actuators.

Thanks to these features, URB is highly suitable for our system where there may be several nodes collecting data from different kinds of sensors at different frequencies. Figure 3.7 shows an example system for which usage of URB is highly suitable. In this example system, s_i represent sensors, a_i represent actuators and μC_i represent microcontrollers, corresponding to individual nodes in the URB. μC_1 reads analog signals from sensors s_1 and s_2 and samples the signals at 500Hz. μC_2 reads analog signals from sensors s_3 , s_4 , and s_5 and samples the signal from s_3 at 400Hz and the signals from s_4 and s_5 at 200Hz. μC_3 reads data from sensor s_6 at 100Hz through I^2C protocol and sends data to actuator a_1 at 300Hz through the SPI protocol. μC_1 , μC_2 , and μC_3 connect to the CPU with a shared bus. The CPU sends requests to all of the microcontrollers at 500 Hz through the RS485 protocol. μC_4 sends data to actuators a_2 and a_3 at 100Hz through the I^2C protocol. The CPU sends requests to the microcontroller at 100Hz through the RS232 protocol. μC_5 sends data to actuator a_4 at 200Hz through SPI protocol. The CPU sends requests to the microcontroller at 200 Hz. URB facilitates the implementation and deployment of such heterogeneous systems.

The original URB is a two-tier network architecture where there are bridge components between the CPU and a group of nodes sharing the same bus. In this two-tier structure, the connection between the CPU and the bridge circuit is called as the uplink and the connection between the bridge circuit and the group of nodes is called as the downlink. Logically, however, the CPU directly sees all nodes consisting of "message boxes".

URB supports up to 16 message boxes for each node. Each message box is double-buffered and each buffer can be at most 32 bytes in length. 8 of the message boxes are called "outboxes" and 8 of them are called "inboxes". Outboxes contain data to be sent to the CPU, and inboxes contain data sent by the CPU. While data held in message boxes are application specific, outbox 0 always holds node identification information and inbox 0 always accepts protocol commands coming from the CPU.

The URB protocol, there are 4 types of bytes used in communication: flag and size byte, address byte, data byte, and checksum byte. The most significant 3 bits of the flag and size byte are reserved for flags and the remaining bits hold the size of

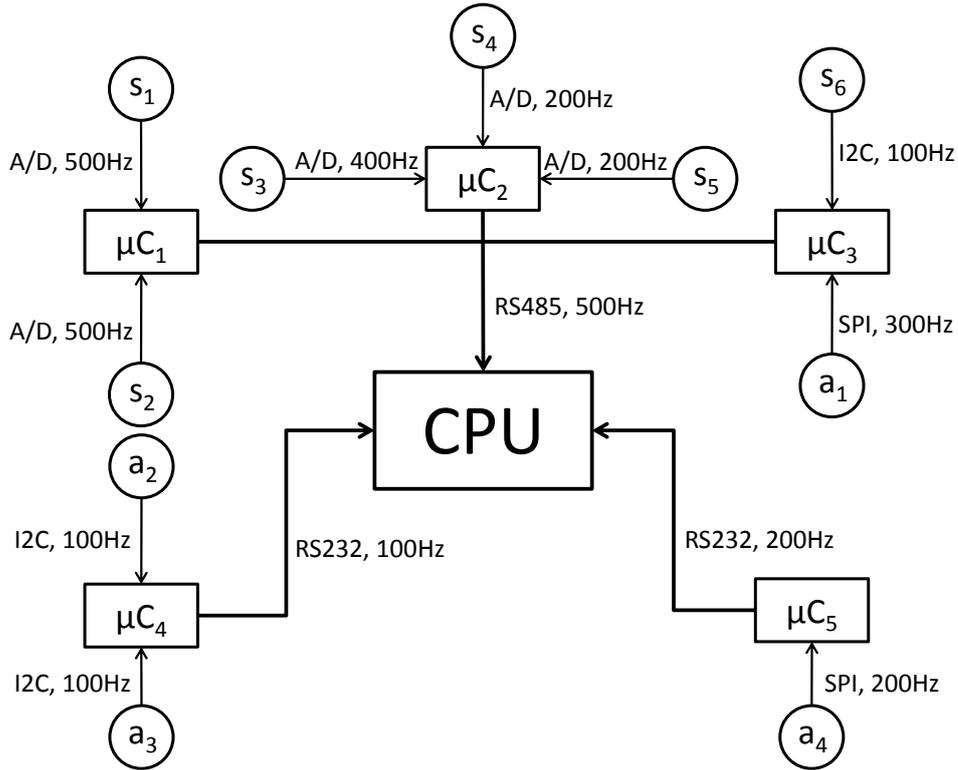


Figure 3.7: An example system highly suitable for URB. s_i represents sensors, a_i represents actuators and μC_i represents microcontrollers. Protocols used across each channel with frequencies of read or write requests are also shown.

message to be sent in bytes. The most significant 4 bits of the address byte hold the node address and the remaining bits hold the message box id to which the data will be written. Node address 0 is used for broadcasting, so the number of nodes on a data bus can at most be 15. Data bytes simply contain the message to be sent. The checksum byte is used to verify the integrity of data. These byte types are illustrated in Figure 3.8.

Even though all byte types are used in the uplink, downlink protocols can infer the message size on their own. Consequently, flag and size byte are not used across the downlink. Whether the message is a request or response the flag and size byte is first sent so that the receiver can infer the size of the message. Then, the address byte is sent so that the message will be considered only by its target node indicated by the node address field. After that, data bytes are sent one by one, followed by the checksum byte. Figure 3.8 summarizes these bytes and their content.

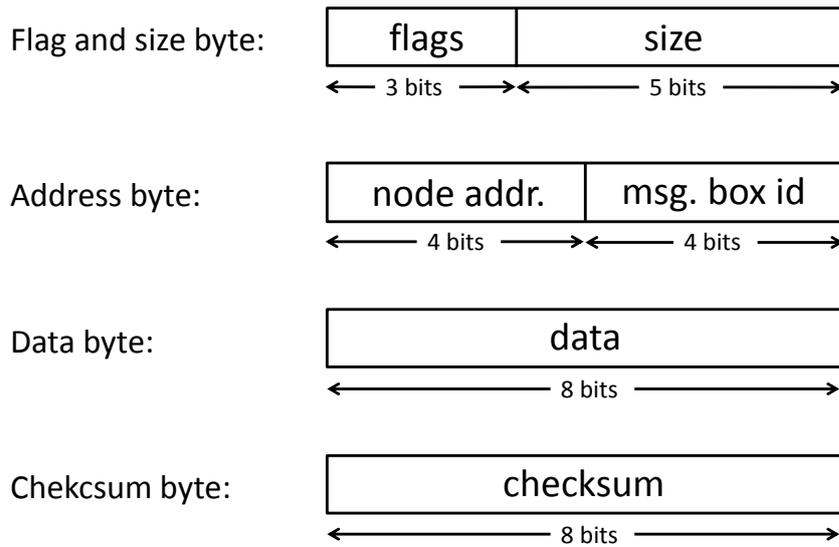


Figure 3.8: Byte types used in the original URB protocol with their fields.

3.3.2 URB Extentions and Implementation

Although our communication infrastructure for the crutch platform is based on the URB protocol, we use the following modifications for our systems:

1. Bridge circuits allow physical layer network protocols between nodes and the CPU. However, our platform only uses the RS485 physical layer network protocol between the CPU and the nodes. Consequently, our platform does not use bridge circuits.
2. In our implementation, only three types of bytes were used across the communication channel: address byte, data byte, and the checksum byte. The most significant bit of the address byte is always 1, and most significant bits of data bytes and the checksum byte are always 0. With this implementation, the address byte is placed at the head of a packet can be easily differentiated from other bytes in the packet, allowing easy detection of the header for the next packet in case of a communication errors, such as bit flips or dropped bytes.

As stated above, only three types of bytes were used in our URB implementation. The most significant bit of the address byte is always 1. 4th to 6th bits of the address byte are reserved for the node address. The node address 0 is again reserved

for broadcasting, so there can only be 7 nodes on the data, rather than the 15 in the original URB. Similarly, the 1st to 3rd bits of the address byte are reserved for encoding the message box id to which data will be written or from which data will be read. The least significant bit of address bytes holds a flag indicating whether the request is a read request or a write request. The most significant bit of the data byte is always 0. Consequently, we have 7 bits per byte to hold the message rather than 8 bits. Due to this fact, our URB implementation converts the 8-bit data stream into a 7-bit data stream. Conversely, after receiving the data payload, the 7-bit data stream is converted back to the original 8-bit content. These conversions can be seen as the addition of an extra layer to the original URB protocol. Similarly, the most significant bit of the checksum byte is always 0. With these byte types, the receiver can infer the size of the data payload. Hence, the flag and size byte is not used in our extended URB protocol.

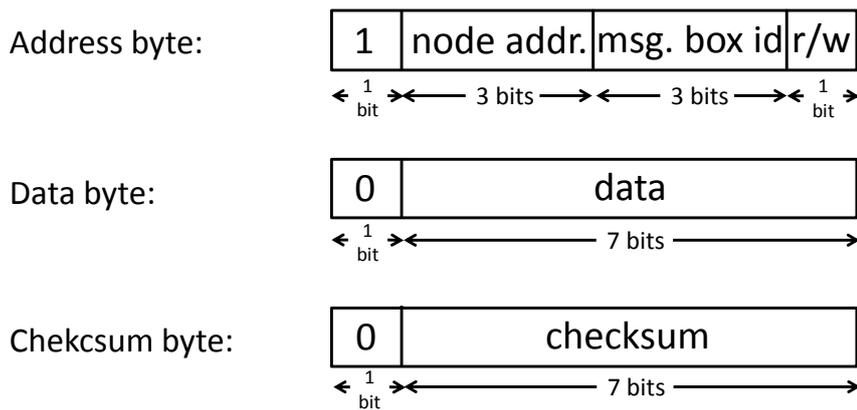


Figure 3.9: Byte types used in our extended URB protocol with their fields.

If the CPU selects write to a node operation, it prepares address byte by making the least significant bit of the address byte 0 and by adjusting the bits indicating the node on which the write operation will be performed and the bits indicating the inbox on which the data will be written. After sending the address byte, it encodes 8-bit classical bytes of which the data consist into 7-bit bytes. Then, it sends data bytes to the channel. Finally it sends the checksum byte so that the node can understand whether the data is corrupted or not. A diagram summarizing these operations is shown in Figure 3.10.

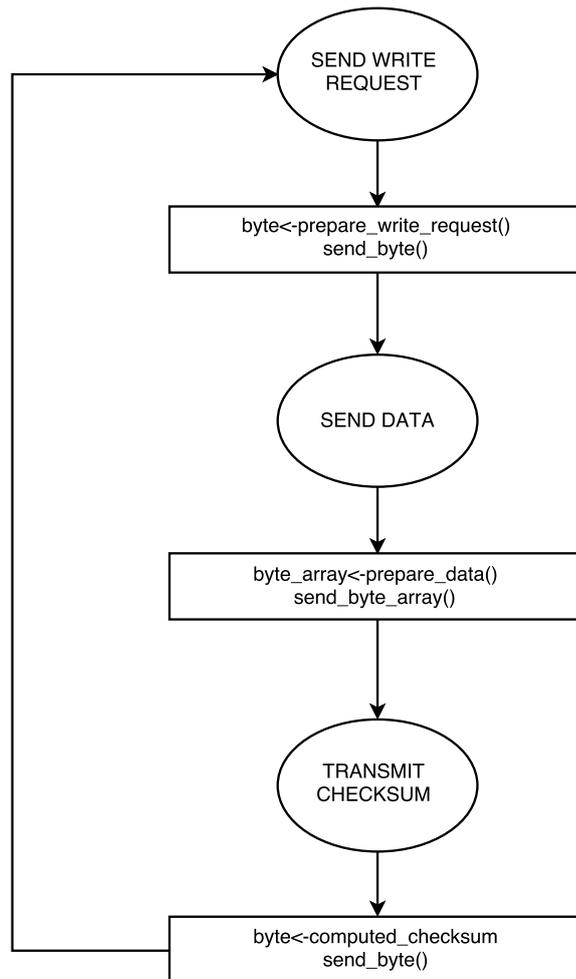


Figure 3.10: CPU side of the extended URB protocol for write operations.

If the CPU selects read from a node operation, it prepares address byte by making the least significant bit of the address byte 1 and by adjusting the bits indicating the node on which the read operation will be performed and the bits indicating the outbox from which the data will be read. After sending the address byte, it starts to wait for the packet. The data packet may come to the CPU in several chunks. The CPU checks whether most significant bits of the data bytes are 0 or not. If most significant bit of at least one of them is 1, the CPU discards whole data. After the CPU receives all of the data bytes, it receives the checksum to investigate whether the data is corrupted or not. If it is corrupted, the CPU discards all of the data. If it is not corrupted, 7-bit bytes forming the data are encoded into 8-bit classical bytes and the operation is done. A diagram summarizing these operations is shown in Figure 3.11.

If a node receives a byte when it is idle, first of all, it checks the most significant bit of the byte whether it is an address byte or not. If it is not an address byte, it simply ignores the byte. If it is an address byte, it extracts node address field of the address byte and checks whether it is equal to its own id or not. If not, the node again simply discards the byte. If it is equal to its own id, it further extracts message box id of the address byte to identify the message box on which the operation will be performed. After that, it checks r/w field of the byte to find out whether the request is a write operation or read operation. If r/w bit is 0, the request is a write operation. In this case, the node waits for the data bytes. When it receives a byte, it checks most significant bit of the byte to find out whether it is a data byte or not. If the most significant bit is 1, it discards the received data and goes into idle state. If the most significant bit is 0, it writes the byte to the appropriate inbox. Please note that the incoming data are encoded into 7-bit bytes so the node decodes the data into classical 8-bit bytes as it writes them to the inbox. Finally, the node receives the checksum byte and checks whether the received data are corrupted or not. If it is corrupted, it simply discards the received data and goes into idle state. If r/w bit of the received address byte is 1, the request is a read operation. In this case, the node sends the data in the requested outbox one by one. Please note that data in the outbox consist of classical 8-bit bytes so prior to sending the data to the CPU, the data are encoded into 7-bit bytes. Finally, the node sends the checksum byte to the CPU so that the CPU can find out whether the data are corrupted or not. A diagram summarizing these procedure is shown in Figure 3.12.

3.4 The Data Acquisition System

The data acquisition system we use to collect data from all sources consists of four components: The smart crutch platform which collects FSR and accelerometer data, AMTI HE6X6 force platform system to collect ground truth measurements of GRF at the tip of the smart crutch, RTD CME137686LX model single board PC to request data from the smart crutch and force platform, and an external PC to permanently store data for analysis and learning purposes.

Analog voltage signals coming from the four FSRs are sampled and digitized by the

MSP430F2274 microcontroller at 400 Hz. The microcontroller communicates with the accelerometer through an I^2C connector and reads lateral and foreaft acceleration data at 100 Hz. There are two outboxes in the microcontroller, one to hold pressure data from the four FSRs, and one to hold acceleration data along both axes.

Ground-truth GRF measurements are collected using an AMTI HE6X6 6-axis force platform [2]. The force platform measures forces along three dimensions (F_x, F_y, F_z) and moments along three dimensions (M_x, M_y, M_z) with a very high accuracy.

As our main CPU for the URB architecture, we used a RTD CME137686LX model single-board PC with a 500 MHz AMD Geode processor and a 32-bit architecture. Figure 3.13 shows a picture of this board which communicates with the smart crutch through an RS485 protocol and sends read requests to the microcontroller at 500 Hz to read relevant outboxes. Similarly, it communicates with the force measurement sensor through an RS232 connector, reading force values at 200 Hz. Since all these operations need to be realized in real-time, the QNX Neutrino real-time operating system was used on the RTD board. For storage and further analysis, the CPU sends the data to an external PC through a TCP/IP link. A diagram depicting the overall structure of our data acquisition system is shown in Figure 3.14.

For storage and further analysis, the CPU sends the data to an external PC with TCP/IP protocol. A diagram depicting overall picture of the data acquisition system is shown in Figure 3.14.

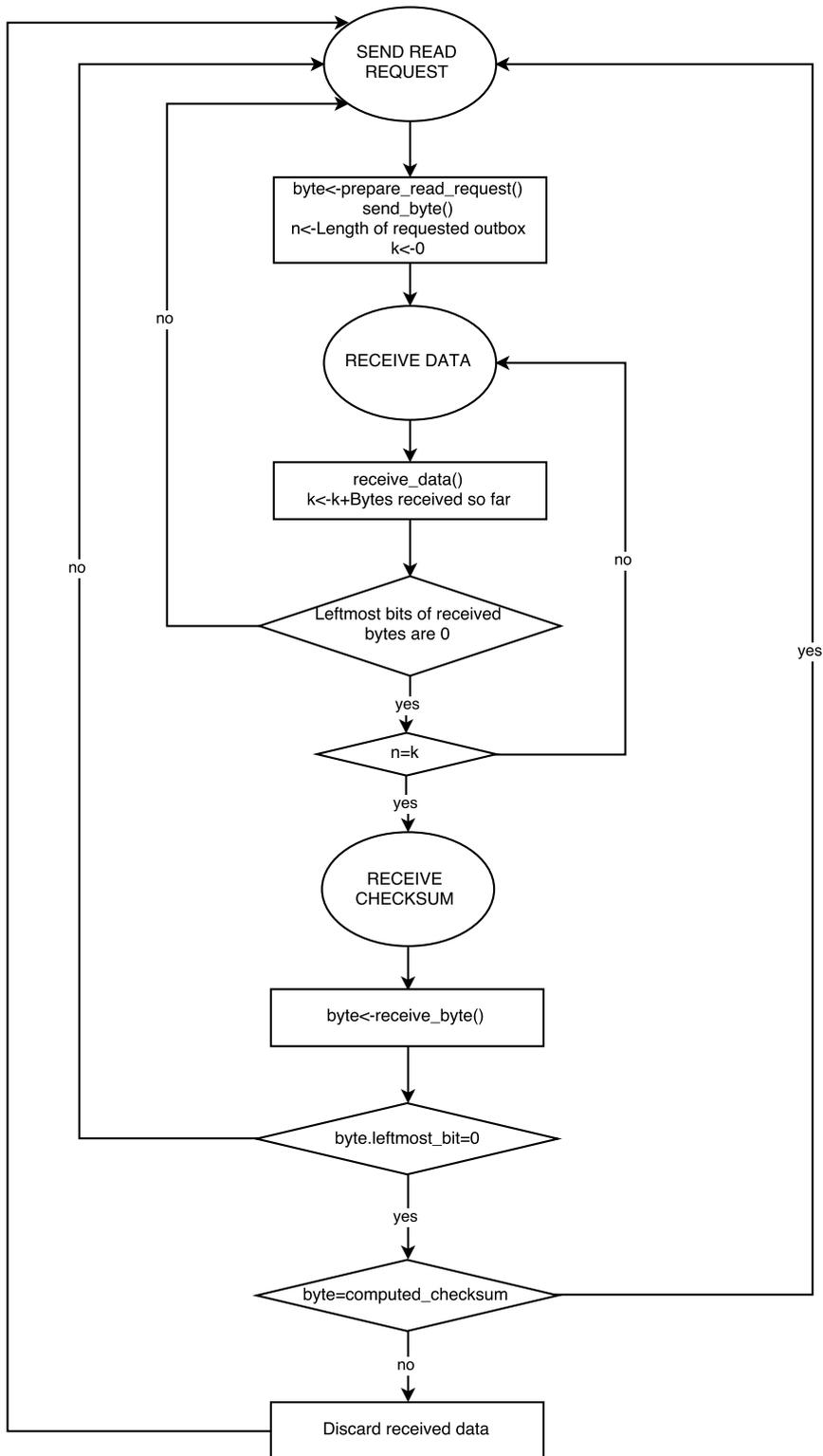


Figure 3.11: CPU side of the extended URB protocol for read operations.

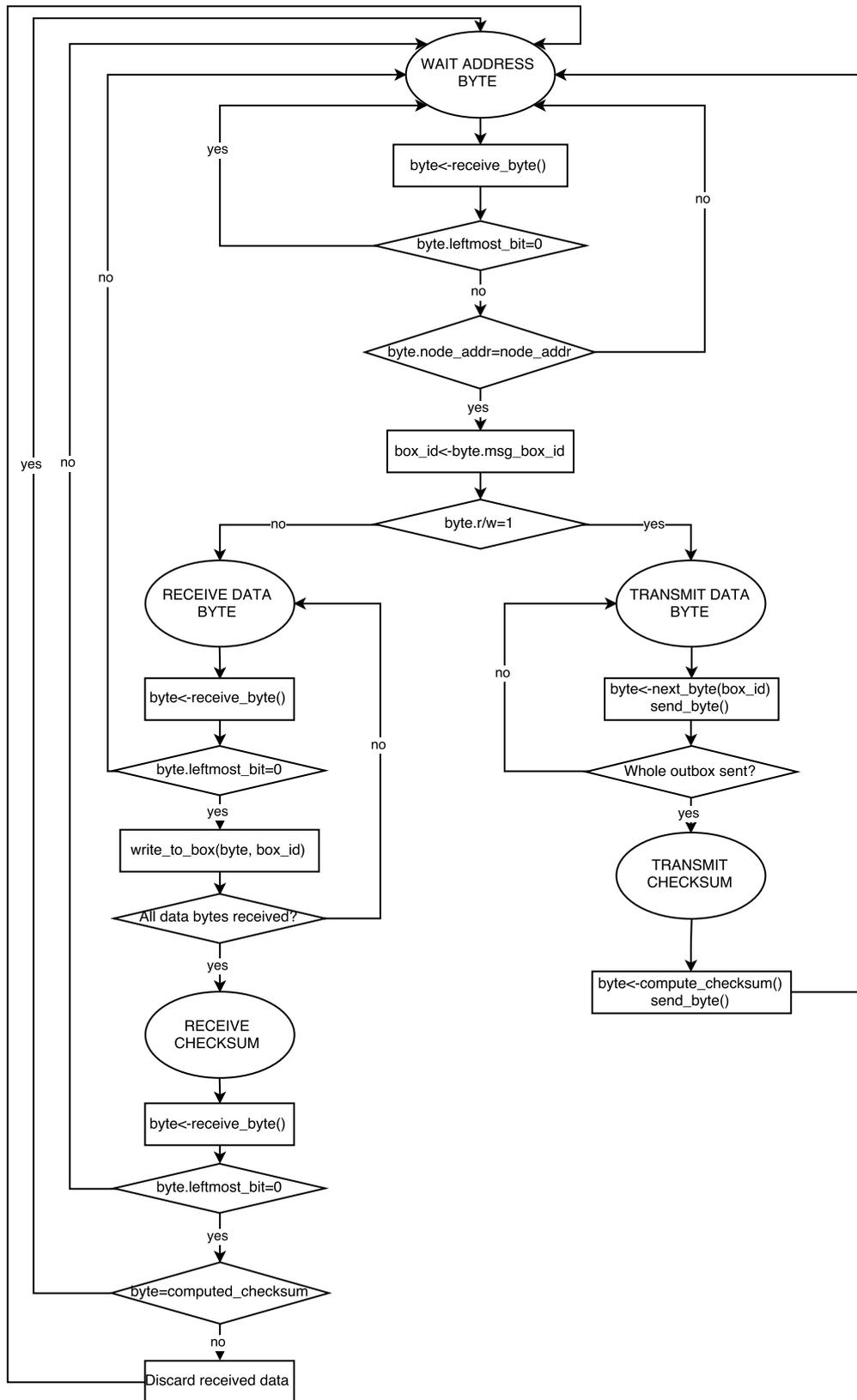


Figure 3.12: Node side of the extended URB protocol.

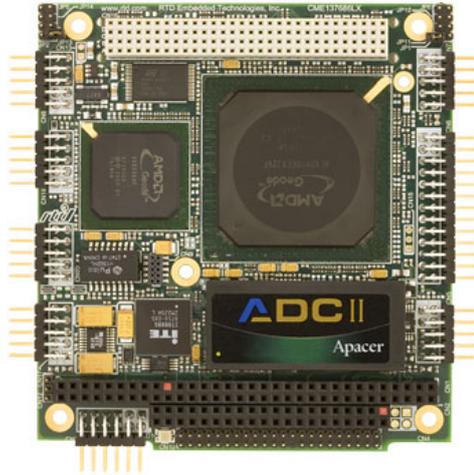


Figure 3.13: The RTD CME137686LX PC104 model single-board PC which hosts the main CPU in our data acquisition system.

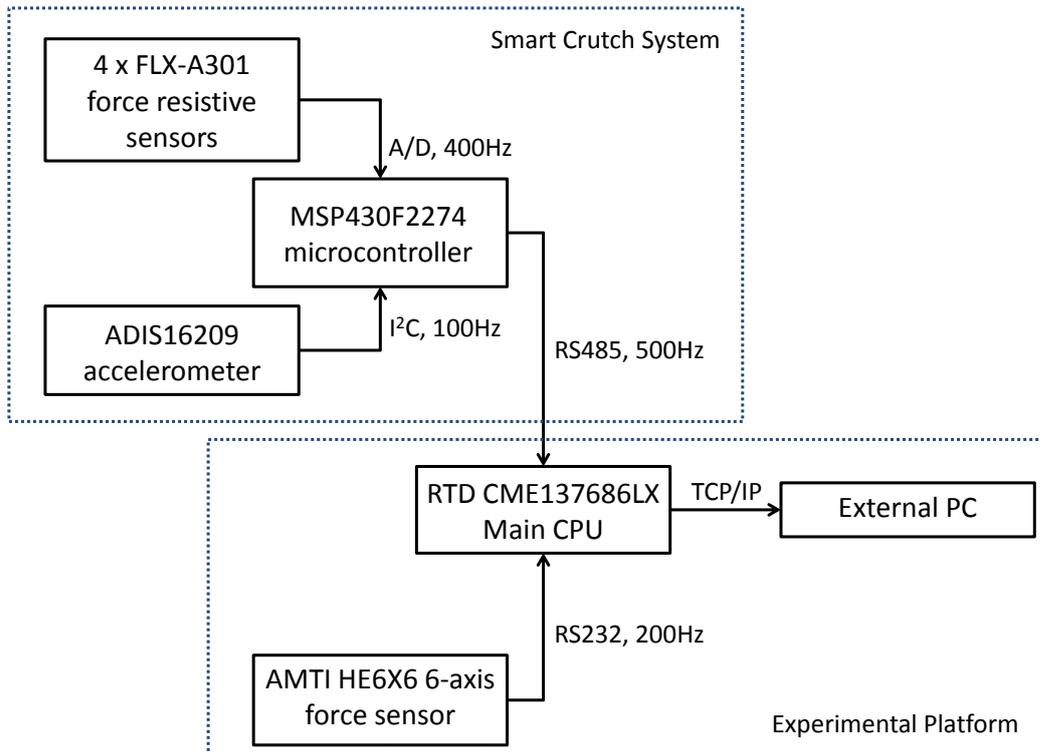


Figure 3.14: The overall structure of our data acquisition system. Protocols used across each channel with frequencies of read or write requests are also shown.

CHAPTER 4

ESTIMATION OF GROUND REACTION FORCES FROM SENSOR DATA

In this chapter, the method we propose to map pressure sensor data to GRF vectors with corresponding results will be briefly explained. Instead of directly describing estimation of GRF in 3D (3-dimensions), we first constraint ourselves to GRF estimation only in the sagittal plane then extent it to 3D. The reason of this is two-fold. Firstly, data acquisition procedure was exhausting so instead of blindly collecting data for 3D, we first tested the method in 2D (2-dimensions) to get an idea about the angle gaps between the tilt angles at which data were collected for better training. Secondly, testing methods in 2D is less time consuming both in terms of their implementation and computation time. Thus, when we ensure that the methods work well in 2D, then we extend them to 3D.

4.1 GRF Estimation in 2-Dimensions

Figure 4.1 shows our experimental platform to estimate ground reaction forces (GRF) with coordinate frames of both the force platform and the accelerometer along with the placements of FSRs relative to the coordinate frames. In this section, we will describe our method for mapping pressure sensor and accelerometer data to GRF vectors in 2D. More specifically, the method for estimation of F_y , and F_z from P_1 , P_2 , P_3 , P_4 , a_y will be presented in this section. Subsequently, Section 4.2 will present the estimation performance.

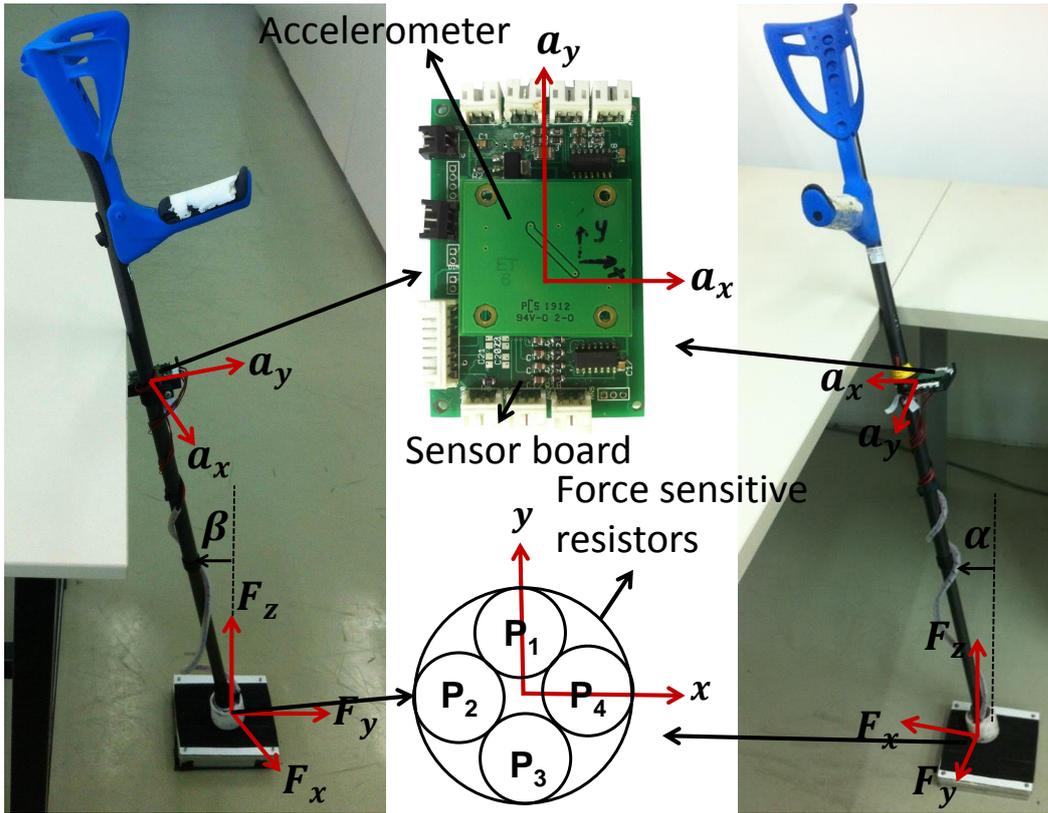


Figure 4.1: Experimental platform for estimating GRF vectors showing the coordinate frames of both our force platform and the accelerometer along with the placements of FSRs relative to these coordinate frames. The two pictures from different views of the same platform is shown to clearly indicate crutch tilt angles α and β .

4.1.1 Angle-Dependent Force Transfer Model

Our estimation method is based on the assumption that, given a crutch angle β with the vertical (shown in Figure 4.1), the GRF vector in 2D depends quadratically on pressure readings from four FSRs symmetrically placed into the silicon shoe at the bottom end of the crutch (can be seen in Figure 4.1). It should be noted that, the relationship can be different for different values of β . Moreover, we also propose that the delay in the transfer of the actual GRF under the silicon shoe to the FSR sensors due to the soft structure of the silicon should also be considered. According to these

assumptions and observations, the proposed model takes the form

$$\hat{\mathbf{F}}(t - t_d(\beta)) := \begin{bmatrix} \hat{F}_y(t - t_d(\beta)) \\ \hat{F}_z(t - t_d(\beta)) \end{bmatrix} = \mathbf{K}_L(\beta) \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{bmatrix} + \mathbf{K}_Q(\beta) \begin{bmatrix} P_1^2(t) \\ P_2^2(t) \\ P_3^2(t) \\ P_4^2(t) \end{bmatrix}. \quad (4.1)$$

In this model,

1. β is the angle of the crutch with the vertical and is computed from accelerometer measurements along the y-axis using

$$\beta = -\sin^{-1}\left(\frac{a_y}{g}\right), \quad (4.2)$$

2. $P_i(t)$ are pressure readings from the FSR sensors,
3. $t_d(\beta)$ is the angle-dependent sensor delay in the transfer of the GRF vector to the FSR sensors due to the soft structure of the silicon shoe,
4. $\mathbf{K}_L(\beta)$ is an angle-dependent linear gain matrix defined as

$$\mathbf{K}_L(\beta) := \begin{bmatrix} k_{L,y1}(\beta) & k_{L,y2}(\beta) & k_{L,y3}(\beta) & k_{L,y4}(\beta) \\ k_{L,z1}(\beta) & k_{L,z2}(\beta) & k_{L,z3}(\beta) & k_{L,z4}(\beta) \end{bmatrix}, \quad (4.3)$$

5. $\mathbf{K}_Q(\beta)$ is an angle-dependent quadratic gain matrix defined as

$$\mathbf{K}_Q(\beta) := \begin{bmatrix} k_{Q,y1}(\beta) & k_{Q,y2}(\beta) & k_{Q,y3}(\beta) & k_{Q,y4}(\beta) \\ k_{Q,z1}(\beta) & k_{Q,z2}(\beta) & k_{Q,z3}(\beta) & k_{Q,z4}(\beta) \end{bmatrix}. \quad (4.4)$$

It should be noted that, as the force-resistance characteristics of the FSR sensors are nonlinear, using only linear terms does not accurately capture the relationship between $P_i(t)$ and $[F_y(t), F_z(t)]^T$. Adding quadratic terms leads to better estimation results by capturing the nonlinear force-resistance characteristics of the FSR sensors in the system.

Under the assumption that delay and gain matrices are constant for a particular β , we identified delay and gain matrices through systematic experiments. The method we used to identify these gain matrices will be described in Sections 4.1.2 and 4.1.3, respectively.

4.1.2 Data-Driven Identification of the Force Transfer Model

Identification of gain matrices and the delay in our force transfer model introduced in Section 4.1.1 requires the collection of sufficient amount of synchronized ground truth force (F_y, F_z), angle (β), and pressure (P_1, P_2, P_3, P_4) data. In this section, our choice of identification method for the gain matrices associated with a particular β will be described, under the assumption that the sensor delay characteristic is known beforehand.

Suppose that we have a collection of data for a particular β consisting of N samples of ground truth force (F_y, F_z), and pressure data (P_1, P_2, P_3, P_4), recorded in a synchronized fashion to yield pressure and force vectors, \mathbf{P}_j and \mathbf{F}_j , with $j = 1, \dots, N$. Additionally, let us suppose that the delay T_d of the particular choice of β is known. With this information and the proposed force transfer model, it is possible to identify the gain matrices $\mathbf{K}_L(\beta)$ and $\mathbf{K}_Q(\beta)$ by using standard linear least squares methods. To this end, we define an error metric for the difference between estimated forces and ground truth forces, which is

$$E(\beta, T_d) := \sum_{j=N_d}^N \|\hat{\mathbf{F}}(\mathbf{P}_j) - \mathbf{F}_{(j-N_d)}\|^2. \quad (4.5)$$

In this equation, $\hat{\mathbf{F}}(\mathbf{P}_j)$ denotes the estimation from the model introduced in Section 4.1.1, using the measured pressure data \mathbf{P}_j , and $\mathbf{F}_{(j-N_d)}$ denotes the measured and recorded ground truth force data delayed in time by N_d samples representing the delay T_d in Equation 4.1.

To identify the linear and the quadratic gain matrices using linear least-squares, we rearrange the gain parameters defined in Equations 4.3 and 4.4 into the two vectors

$$\mathbf{u}_y := \left[k_{L,y1} \quad \dots \quad k_{L,y4} \quad k_{Q,y1} \quad \dots \quad k_{Q,y4} \right]^T, \quad (4.6)$$

$$\mathbf{u}_z := \left[k_{L,z1} \quad \dots \quad k_{L,z4} \quad k_{Q,z1} \quad \dots \quad k_{Q,z4} \right]^T. \quad (4.7)$$

Similarly, ground truth force recordings for each axis are collected into the two vec-

tors

$$\mathbf{F}_y := \begin{bmatrix} F_{y,1} & \dots & F_{y,(N-N_d)} \end{bmatrix}^T, \quad (4.8)$$

$$\mathbf{F}_z := \begin{bmatrix} F_{z,1} & \dots & F_{z,(N-N_d)} \end{bmatrix}^T. \quad (4.9)$$

Finally, pressure sensor readings are assembled into a single matrix as

$$\mathbf{M} := \begin{bmatrix} P_{1,N_d} & \dots & P_{4,N_d} & P_{1,N_d}^2 & \dots & P_{4,N_d}^2 \\ \dots & & & & & \\ P_{1,N} & \dots & P_{4,N} & P_{1,N}^2 & \dots & P_{4,N}^2 \end{bmatrix}. \quad (4.10)$$

To take the delay into account, the first N_d samples were left out of definition in Equation 4.10. Also, to make the lengths of vectors \mathbf{F}_y and \mathbf{F}_z compatible with the matrix \mathbf{M} , the last N_d samples were left out of the vectors \mathbf{F}_y and \mathbf{F}_z in Equations 4.8 and 4.9. Substituting these definitions into the error metric in Equation 4.5 yields simpler form of the error metric

$$E(\beta, T_d, \mathbf{u}_y, \mathbf{u}_z) = \|\mathbf{M}\mathbf{u}_y - \mathbf{F}_y\|^2 + \|\mathbf{M}\mathbf{u}_z - \mathbf{F}_z\|^2. \quad (4.11)$$

Based on this error metric, we estimate the gain parameters by independently using standard linear least-squares on first and second terms of Equation 4.11 resulting in

$$\hat{\mathbf{u}}_y = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}_y, \quad (4.12)$$

$$\hat{\mathbf{u}}_z = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}_z. \quad (4.13)$$

The experimental procedure for assessing the estimation performance of this method with corresponding detailed results are presented in Section 4.2.4.

4.1.3 Identification of Angle-Dependent Sensor Delay

As mentioned in Section 4.1.1, the soft structure of the silicon shoe covering the FSR sensors at the tip of the crutch results in the GRF under the silicon shoe being transferred to the FSR sensors in a delayed manner. In this section, we present a method to experimentally identify this delay.

It should first be noted that the relationship between the delay and β is nonlinear, which eliminates the possibility of using linear optimization methods for its identification. However, we can still compute T_d that minimizes the error in Equation 4.11 for a particular angle β constructed around the least-squares estimates of gain parameters $\hat{\mathbf{u}}_y$ and $\hat{\mathbf{u}}_z$. In other words, for a particular β , we define $T_d(\beta)$ as the sensor delay that minimizes the error when we estimate gain parameters through linear least-squares. More formally, we can estimate the sensor delay for a particular β angle by solving the optimization problem,

$$\hat{t}_d(\beta) = \underset{t_d}{\operatorname{argmin}} E(\beta, t_d, \hat{\mathbf{u}}_y, \hat{\mathbf{u}}_z), \quad (4.14)$$

based on the error metric given in Equation 4.11, and the gain vectors estimated by using Equations 4.12 and 4.13. To solve this one-dimensional optimization problem, `fminbnd` function of MATLAB was used. Details of the experimental procedure with associated results are presented in Section 4.2.3.

4.1.4 Estimation of GRF Vectors for Arbitrary Crutch Angles

We have described methods to identify the gain matrices and the sensor delay in the model given in Equation 4.1 only for particular β angles for which data were collected. In order to estimate gain matrices and the sensor delay at angles for which no data were collected, we propose to linearly interpolate the matrices and sensor delay. More formally, suppose that the gain matrices and the sensor delay are identified at two different crutch angles β_1 and β_2 . For an arbitrary β angle, as long as $\beta_1 < \beta < \beta_2$, we propose to estimate the gain matrices as

$$\mathbf{K}_L(\beta) = \mathbf{K}_L(\beta_1) \frac{\beta_2 - \beta}{\beta_2 - \beta_1} + \mathbf{K}_L(\beta_2) \frac{\beta - \beta_1}{\beta_2 - \beta_1}, \quad (4.15)$$

$$\mathbf{K}_Q(\beta) = \mathbf{K}_Q(\beta_1) \frac{\beta_2 - \beta}{\beta_2 - \beta_1} + \mathbf{K}_Q(\beta_2) \frac{\beta - \beta_1}{\beta_2 - \beta_1}. \quad (4.16)$$

Similarly, we can estimate the angle-dependent sensor delay as

$$T_d(\beta) = T_d(\beta_1) \frac{\beta_2 - \beta}{\beta_2 - \beta_1} + T_d(\beta_2) \frac{\beta - \beta_1}{\beta_2 - \beta_1} \quad (4.17)$$

under the assumption that the dependence of gains and the delay on β is locally approximately linear. Details of the experimental procedure with associated results in

estimating gain matrices and the sensor delay for arbitrary angles are presented in Section 4.2.5.

4.2 Experimental Results for 2-Dimensions

4.2.1 The Experimental Procedure

Two kinds of experiments have been conducted to test the proposed system identification method and evaluate its performance. First, stationary experiments were conducted, in which the crutch angle β was kept constant. Second, dynamic experiments were done, wherein the crutch angle β was varied slowly.

Stationary experiments were used to evaluate the performance and accuracy of the least-squares method used to identify the gain matrices and the one-dimensional optimization method used to identify the sensor delay, both at training angles and intermediate angles. In these experiments, we focused on systematically collecting data at different crutch angles to be used both in training and later evaluation of the model. More precisely, we collected data in the interval $\beta \in [-14^\circ, 14^\circ]$, which is roughly the interval of crutch angles swept during walking with robotic orthoses, with increments of 2° . At each angle, data were collected using the platform in Figure 4.1, recording

- pressure data (P_1, P_2, P_3, P_4),
- acceleration data (a_y) which is later converted to β by using Equation 4.2,
- ground-truth values of GRF data (F_y, F_z),

while external forces were manually applied to the crutch, keeping β as constant as possible. A trial of 11 experiments were conducted for each angle, and associated data was filtered and recorded to be used in training the model and evaluation of its performance. Data collected for $\beta = 4^\circ$ is illustrated in Figure 4.2.

Dynamic experiments focused on evaluating the performance of the model under realistic conditions. A total of 20 experiments were conducted to measure the performance of the model under dynamic conditions, wherein the position of the tip of the

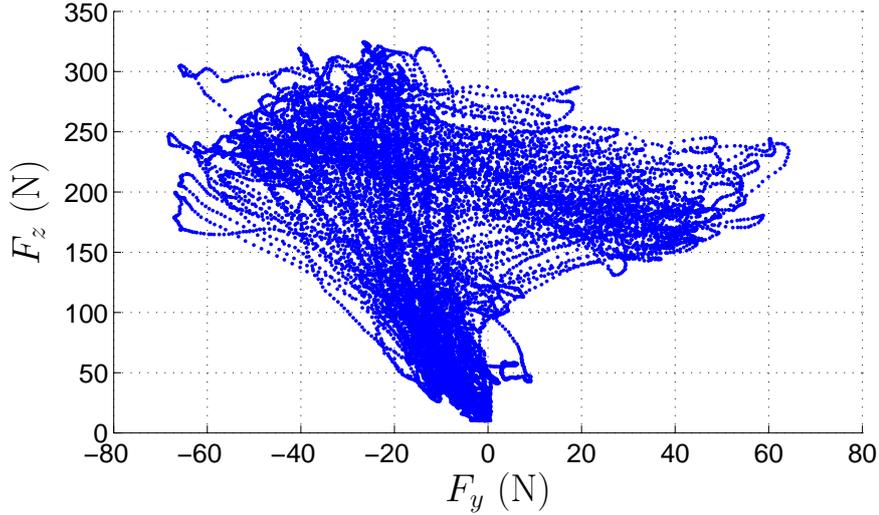


Figure 4.2: Scatter plot of ground reaction force samples collected during all of the 11 experiments for $\beta = 4^\circ$.

crutch was preserved, while moving the crutch back and forth such that β was varied in the interval $[-14^\circ, 14^\circ]$ with a reasonably slow angular velocity. Data collected for dynamic experiments were not used for training purposes, but were only used to evaluate model performance. Moreover, we conducted 5 more dynamic experiments in which β was varied fast to state the accuracy of the model under high speed dynamic conditions.

4.2.2 Error Metrics

To measure the accuracy of the proposed system identification method, RMS errors in both axes were computed for each leave-one-out run by using the error functions

$$E_{rms,y} := \frac{100}{\max_j \|\mathbf{F}_j\|} \sqrt{\frac{1}{N} \sum_{j=1}^N (\hat{F}_{j,y} - F_{j-N_d,y})^2}, \quad (4.18)$$

$$E_{rms,z} := \frac{100}{\max_j \|\mathbf{F}_j\|} \sqrt{\frac{1}{N} \sum_{j=1}^N (\hat{F}_{j,z} - F_{j-N_d,z})^2}. \quad (4.19)$$

The normalization factors in these error definitions correspond to the norm of the maximum GRF observed during the experiment from which data were obtained.

Equations 4.18 and 4.19 give us an idea about the estimation error along each axis but

it is hard to get an idea about the angle between the ground truth GRF vector and the estimated GRF vector, and ratio of the norm of the ground truth GRF vector and the estimated GRF vector, in other words, errors in polar coordinates. Thus, in addition to these error metrics, we define error metrics

$$E_{\Theta} := \frac{1}{N} \sum_{j=1}^N \cos^{-1} \left(\frac{\hat{\mathbf{F}}_j \cdot \mathbf{F}_j}{\|\hat{\mathbf{F}}_j\| \|\mathbf{F}_j\|} \right), \quad (4.20)$$

$$E_L := \frac{1}{N} \sum_{j=1}^N \left| \frac{\|\hat{\mathbf{F}}_j\|}{\|\mathbf{F}_j\|} - 1 \right|. \quad (4.21)$$

Equation 4.20 is the average of the angle between the estimated GRF vector and ground-truth GRF vector for a data set. Equation 4.21 gives us an idea about the ratio of the estimated GRF vector to ground-truth GRF vector. We used the ratio by subtracting 1 from it instead of leaving it as it is. If we do not subtract 1 from the ratio, the result is always around 1 due to unbiased error.

As mentioned earlier, 11 experiments were conducted for each β during stationary experiments. To measure the average estimation errors in cartesian coordinates and polar coordinates for a particular crutch angle β , arithmetic means of errors for all experiments at that angle were computed as

$$\bar{E}_{rms,y} := \frac{1}{11} \sum_{i=1}^{11} E_{rms,y,i}, \quad (4.22)$$

$$\bar{E}_{rms,z} := \frac{1}{11} \sum_{i=1}^{11} E_{rms,z,i}, \quad (4.23)$$

$$\bar{E}_{\Theta} := \frac{1}{11} \sum_{i=1}^{11} E_{\Theta,i}, \quad (4.24)$$

$$\bar{E}_L := \frac{1}{11} \sum_{i=1}^{11} E_{L,i}. \quad (4.25)$$

In the sequel, these error metrics will be used to evaluate model performance.

4.2.3 Angle-Dependent Sensor Delay Characteristic

In the first phase of the experiments, we identified the sensor delay $T_d(\beta)$ for crutch angles at which data were collected by using the method described in Section 4.1.3.

To do this, we used leave-one-out cross validation method [14]. In summary, from among 11 sets of data for each β angle, 10 were used for training while the remaining 1 data set was used for testing. This was repeated with 11 different data sets. To be more specific, at a particular β , the training set was used to compute $\hat{\mathbf{u}}_y$ and $\hat{\mathbf{u}}_z$ in Equation 4.14 and $T_d(\beta)$. For each β , the arithmetic mean of sensor delays estimated in each leave-one-out run at a particular β angle was computed to estimate the sensor delay at that particular β angle. The identified sensor delay characteristic is shown in Figure 4.3.

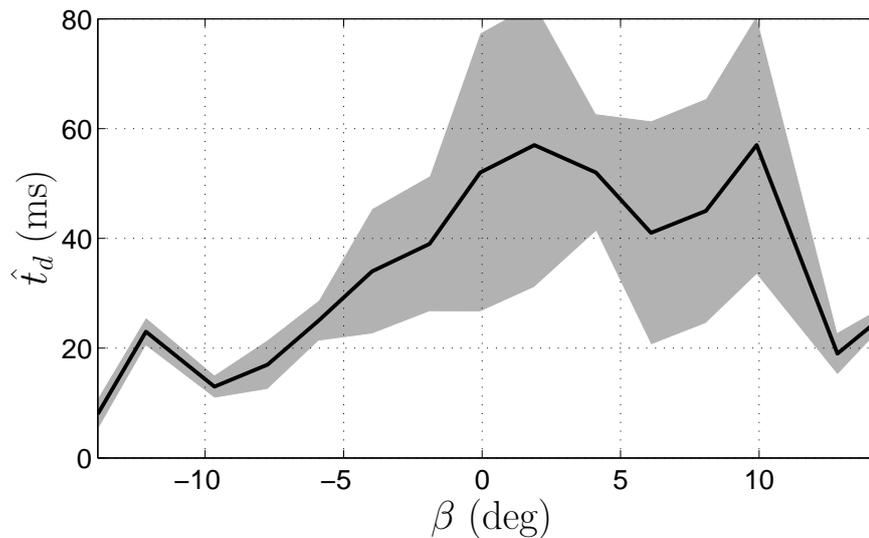


Figure 4.3: The graph of the estimated sensor delay (in milliseconds) vs. β (in degrees). For each β , estimations from 11 experiments were averaged. The shaded region shows the standard deviation range.

As seen from Figure 4.3, the delay gets larger as magnitude of the crutch angle gets smaller. This is expected because the compaction volume of the soft silicon caused by pressure is large for small $|\beta|$ which leads to higher delay values with higher standard deviations.

4.2.4 Model Accuracy for Crutch Angles Used for Training

Before assessing the performance of the force transfer model for arbitrary crutch angles, we first focus on assessing validity of the estimated gain parameters and the sensor delay at crutch angles at which the data were collected. Please note that, sen-

sor delay characteristics as a function of β were already identified up to this point, and associated results presented in Section 4.2.3. In this phase of the experiments, only gain parameters were estimated and the validity of these parameters along with the estimated sensor delay characteristics are assessed. To do this, leave-one-out cross validation method was used again. In each of the 11 runs for each β angle, 10 out of 11 data sets were used for training, while the remaining data set was used for testing. Errors computed for the testing data set in each run were arithmetically averaged by using Equations 4.22 and 4.23. Graphs presenting average percentage RMS errors in the estimation of F_z and F_y are shown in Figure 4.4.

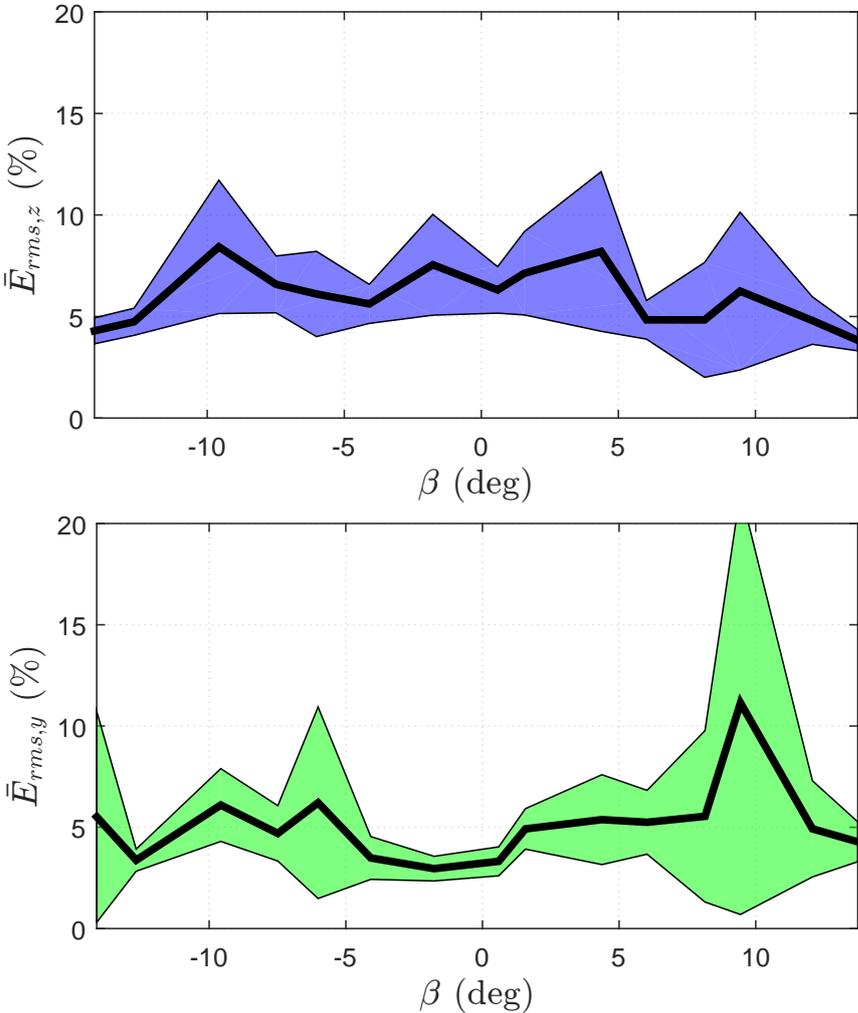


Figure 4.4: Percentage RMS errors in the estimates of F_z (top graph) and F_y (bottom graph) vs. β graph. For each β , resulting errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.

Moreover, errors in polar coordinates were computed using (4.20) and (4.21). Resulting errors were averaged by using (4.24) and (4.25), respectively. Graphs presenting average errors in polar coordinates are shown in Figure 4.5.

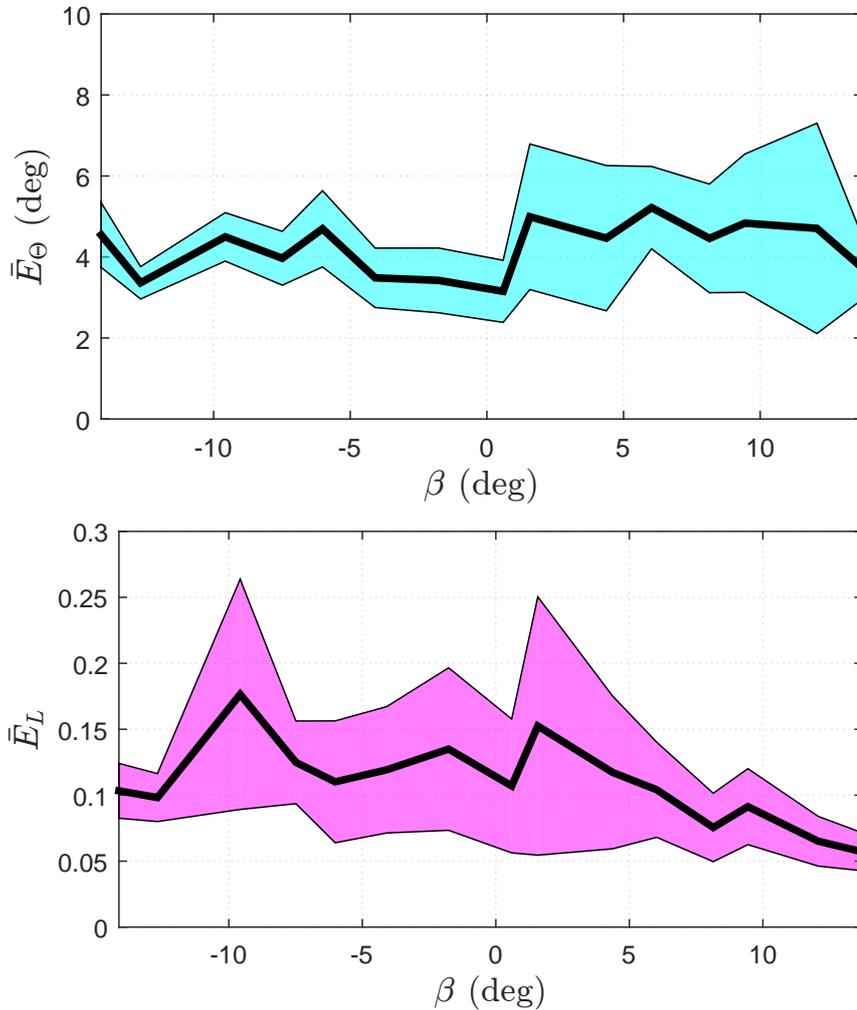


Figure 4.5: Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph. For each β , resulting errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.

As seen from Figure 4.4, estimation errors in the z-axis are below 8% for all β angles and estimation errors in the y-axis are below 6% for all β angles. Also, as seen from Figure 4.5, average angle between the estimated GRF vector and ground-truth GRF vector is around 5 degrees and error in the norm of estimated GRF vector is around

0.1. Even though these results only present errors for a very limited scenario in which the crutch angle is kept constant and evaluation done only for training angles, they still show the accuracy of estimated gain parameters and the sensor delay and confirm that the proposed force transfer model performs well enough in mapping pressure sensor data to 2D GRF vector data at least in controlled settings.

4.2.5 Model Accuracy for Arbitrary Crutch Angles

The interpolation method proposed in Section 4.1.4 enables us to use the model in Section 4.1.1 for crutch angles other than those at which the data were collected as long as $\beta \in [-14^\circ, 14^\circ]$. To assess the performance of this interpolation method, we take all possible (β_1, β_2) pairs such that $\beta_2 - \beta_1 = \Delta\beta$ and measure estimation errors at $\beta = (\beta_1 + \beta_2)/2$ by using a model interpolated from β_1 and β_2 . With this way, we choose the testing angle such that it will be farthest from the boundary angles used for the interpolation, giving us the worst case performance for the chosen (β_1, β_2) pair.

We first focus on $\Delta\beta = 4^\circ$. As stated in Section 4.2.1, 11 experiments have been conducted for each β , and errors in F_y and F_z are found for each experiment by using error functions given in Equations 4.18, 4.19, 4.20, and 4.21. Errors found in the 11 experiments are arithmetically averaged by using Equations 4.22, 4.23, 4.24, and 4.25, respectively. Graphs presenting average percentage RMS errors estimating F_y and F_z are shown in Figure 4.6. Moreover, Graphs presenting average errors in polar coordinates are shown in Figure 4.7.

As seen from Figures 4.6 and 4.7, estimation of 2D GRF vectors at intermediate crutch angles by linearly interpolating gain parameters and sensor delays with $\Delta\beta = 4^\circ$ does not considerably increase the error in the z-axis, staying below 9% for all β . However, errors in the y-axis for large $|\beta|$ are increased, but the results are still promising for the intended usage domain of the smart crutch.

Error graphs for $\Delta\beta = 8^\circ$ in cartesian coordinates and in polar coordinates are shown in Figure 4.8 and in Figure 4.9, respectively. Errors are computed exactly in the same way when $\Delta\beta = 4^\circ$. These figures show that estimation error in both axis increases considerably, which in turn suggests that choosing $\Delta\beta = 8^\circ$ is not suitable for neither

at stationary arbitrary angles nor under dynamic conditions.

The similarity between the error graphs of $\bar{E}_{rms,z}$ and \bar{E}_L is a consequence of the fact that norm of the GRF vector heavily depends on F_z . Likewise, the similarity between the error graphs of $\bar{E}_{rms,y}$ and \bar{E}_θ is a consequence of the fact that angle between the GRF vector and the vertical is a result of shear forces.

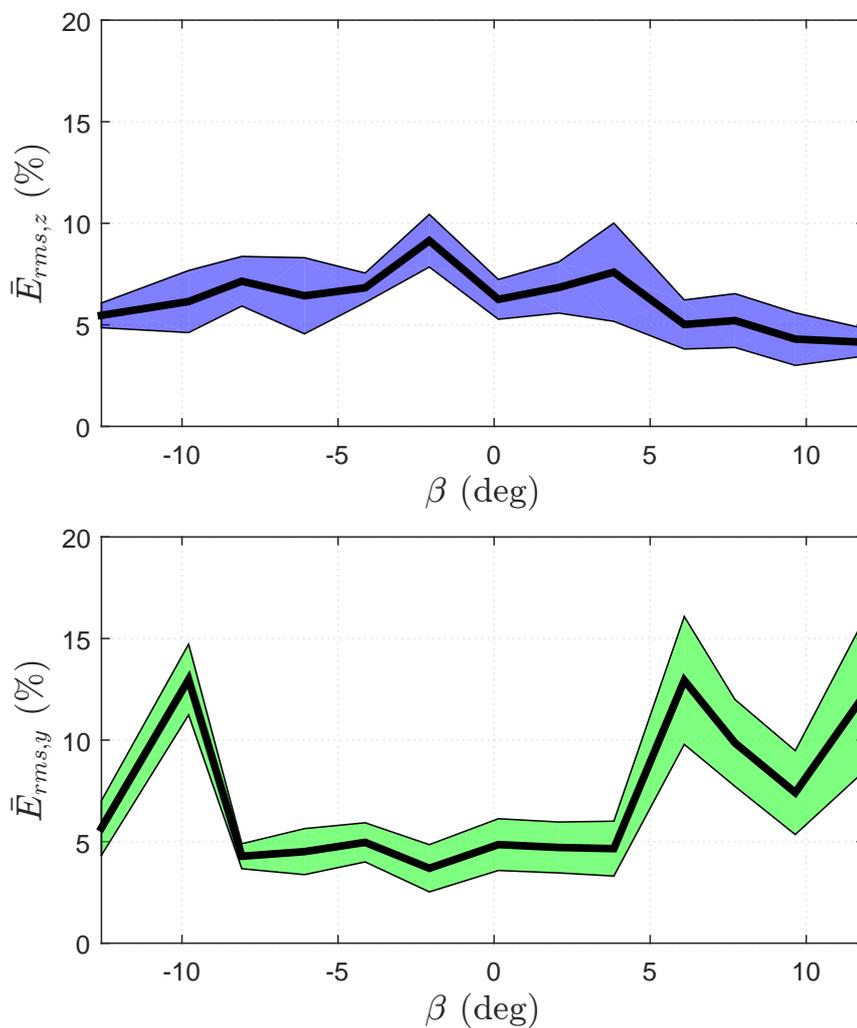


Figure 4.6: Errors in the estimates of F_z (top graph, in percentage), and F_y (bottom graph, in percentage) vs. β (in degrees) in cartesian coordinates graph at interpolated angles for $\Delta\beta = 4^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.

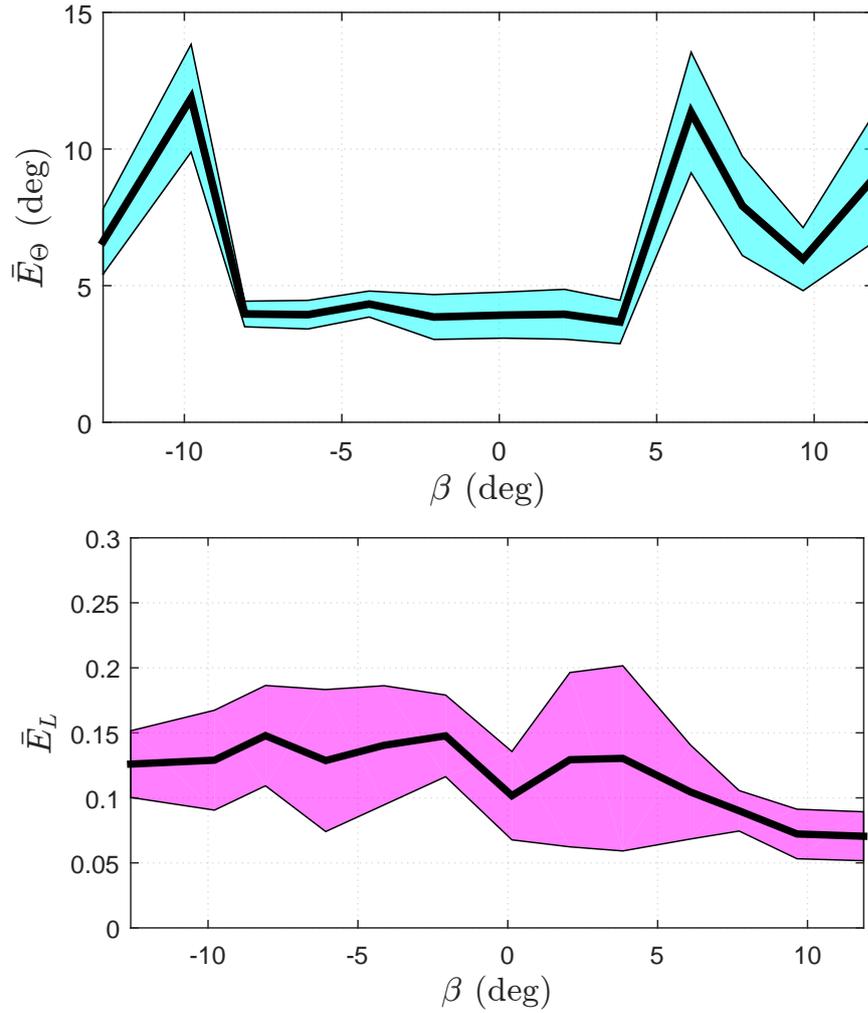


Figure 4.7: Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph at interpolated angles for $\Delta\beta = 4^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.

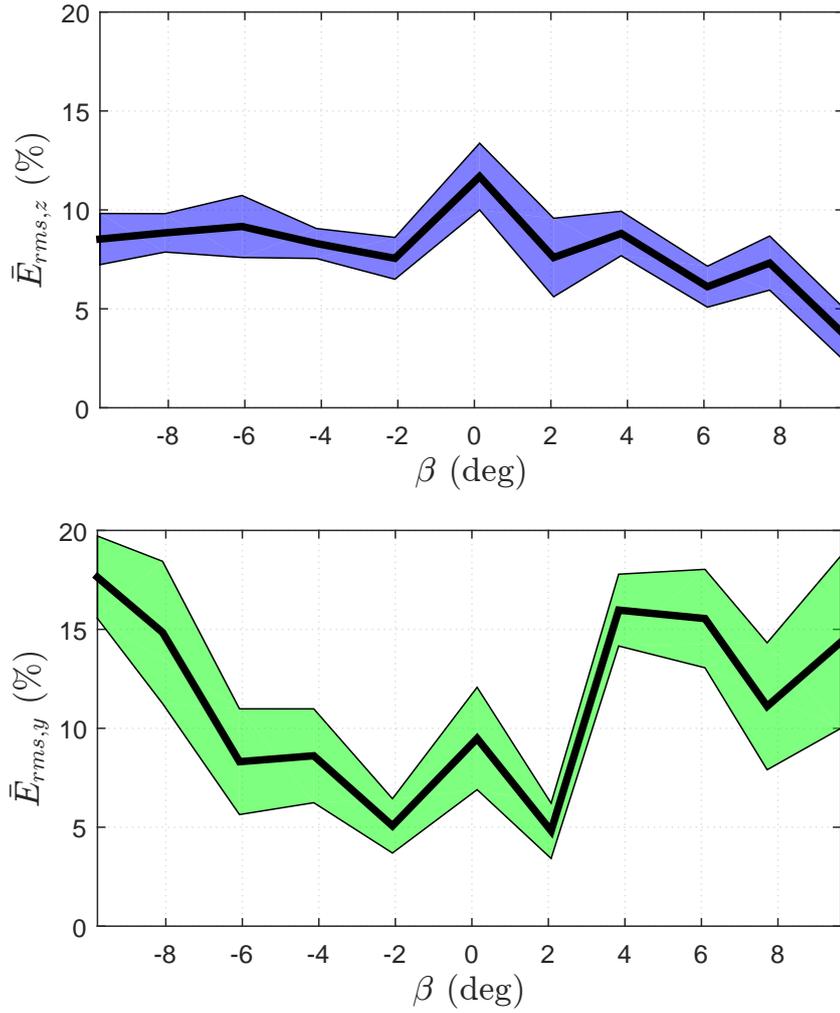


Figure 4.8: Percentage errors in the estimates of F_z (top graph), and F_y (bottom graph) vs. β (in degrees) in cartesian coordinates at interpolated angles for $\Delta\beta = 8^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged. Shaded regions indicate standard deviation.

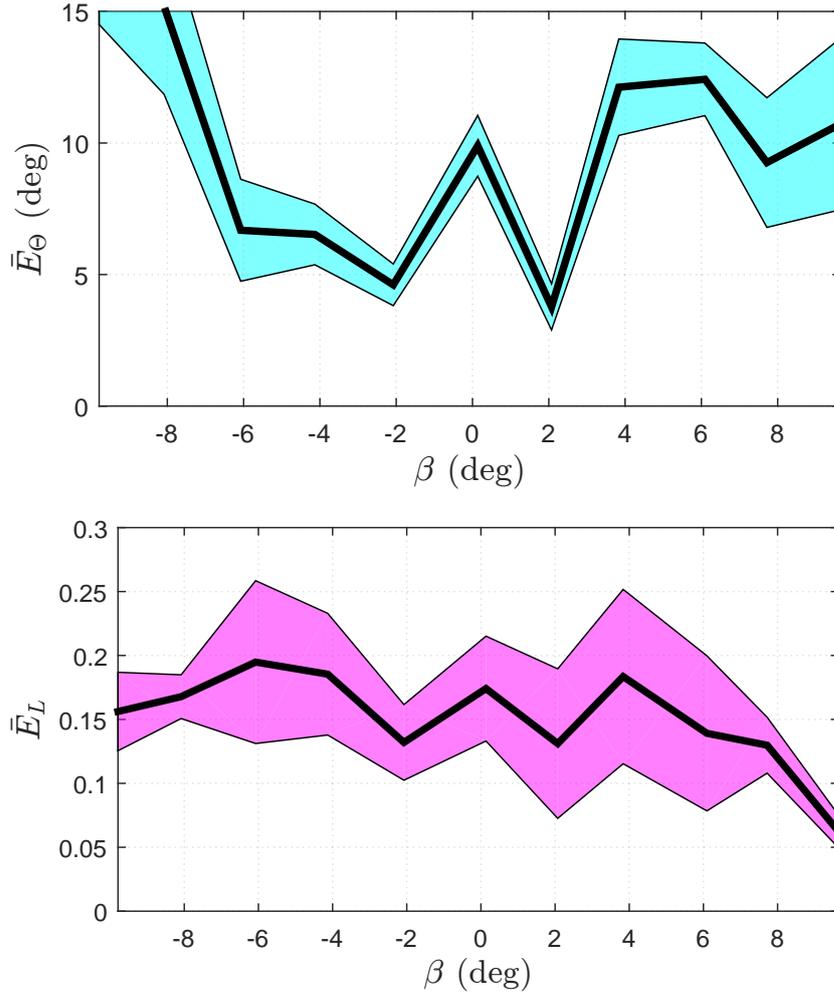


Figure 4.9: Errors in terms of the angle between the estimated GRF vector and ground-truth GRF vector (top graph) and the ratio of the norm of the estimated GRF vector to the ground-truth GRF vector (bottom graph) vs. β graph at interpolated angles for $\Delta\beta = 8^\circ$. For each $\beta = (\beta_1 + \beta_2)/2$, percentage errors for 11 experiments are averaged arithmetically. Shaded regions indicate standard deviation.

4.2.6 Accuracy Under Dynamic Conditions

Up to now, performance results of the proposed model in Equation 4.1 were presented only for constant β . However, in potential applications of the smart crutch system such as the control of a robotic orthosis systems, the crutch will exhibit dynamic, quasi-periodic behaviour, rather than staying still. This makes the crutch angle to be a function of time. In this last phase of experiments, the accuracy of the model under

such dynamic conditions are evaluated.

It should first be noted that under dynamic conditions, the model in Equation 4.1 cannot be directly used since it needs the knowledge of pressure sensor values ahead of time as a result of the sensor delay. To make the model causal, the estimation of the GRF vector needs to be delayed. In this regard, let us call maximum value of the sensor delay among all β angles as

$$t_d^{max} := \max \hat{t}_d(\beta). \quad (4.26)$$

If we delay pressure signals to be processed by t_d^{max} , we may be able to estimate GRF vectors in real-time, only delayed by t_d^{max} . When we incorporate maximum sensor delay, t_d^{max} and the time varying $\beta(t)$ into the model, Equation 4.1 becomes,

$$\begin{aligned} \hat{\mathbf{F}}(t - t_d^{max}) = & \mathbf{K}_L(\beta(t)) \mathbf{P}(t + t_d(\beta(t)) - t_d^{max}) \\ & + \mathbf{K}_Q(\beta(t)) \mathbf{P}^2(t + t_d(\beta(t)) - t_d^{max}) . \end{aligned} \quad (4.27)$$

According to Figure 4.3, $t_d^{max} = 58ms$ for our crutch platform. Consequently, when we buffer pressure sensor readings such that the signals will be processed with a 58ms delay, we can estimate the GRF vector in real-time only with this slight delay. One potential flaw with this approach is the fact that the sensor delay corresponding to the current β is used (This may lead inaccurate results because β may have been changed relative to the β corresponding to delayed pressure sensor readings). As mentioned in Section 4.2.1, 20 experiments were conducted to measure performance of the model under dynamic conditions. Graphs presenting estimation results in cartesian coordinates in one of these experiments are shown in Figure 4.10, and graphs presenting estimation results in polar coordinates of the same experiment are shown in Figure 4.11. Moreover, average, maximum, and minimum errors in both axes for 20 dynamic experiments are shown in Table 4.1.

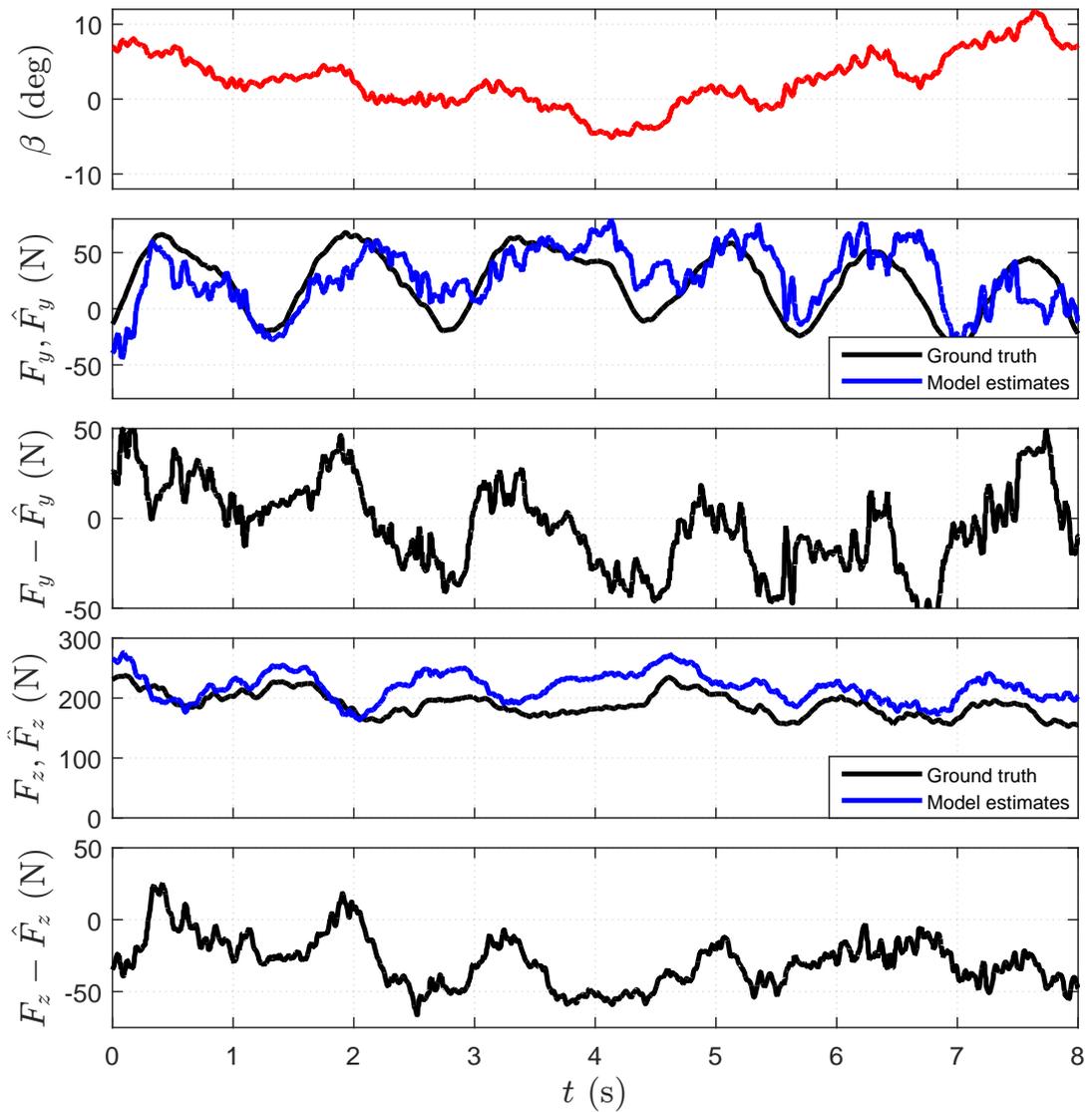


Figure 4.10: Graphs showing estimation results in cartesian coordinates for one of the dynamic experiments. The graph at the top shows β as a function of time. Black trajectories in the second from top and the second from bottom graphs show F_y and F_z , respectively. Blue trajectories in the second from top and the second from bottom graphs show \hat{F}_y and \hat{F}_z , respectively. The middle and bottom graphs show the difference between the ground-truth and estimated forces in y and z axis, respectively.

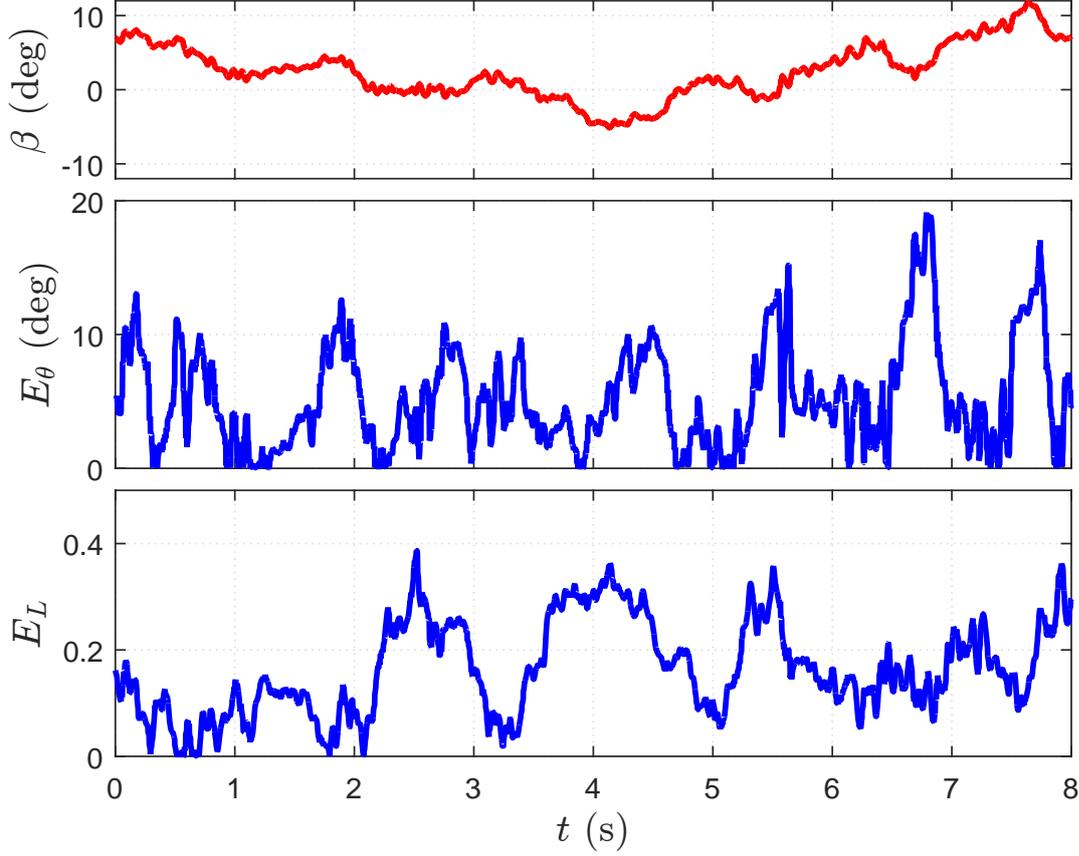


Figure 4.11: Graphs showing estimation results in polar coordinates for one of the dynamic experiments. Top graphs shows β as a function of time. Middle graphs shows the angle between the estimated GRF vector and ground-truth GRF vector. Bottom graph shows ratio of the norm of the estimated GRF vector to ground-truth GRF vector.

Table 4.1: Average, minimum and maximum errors across 20 dynamic experiments in 2D.

	Avg.	Min	Max
$E_{rms,y}$ (%)	12.4 ± 3.17	11.54	19.11
$E_{rms,z}$ (%)	9.12 ± 2.71	6.54	15.77
E_θ (deg)	7.59 ± 1.73	5.46	11.77
E_L	0.12 ± 0.03	0.08	0.19

As seen from Table 4.1, average error in z-axis is below 10% and average error in y-axis is below 13%. Moreover, angle between the estimated and ground-truth GRF vectors is around 7.5 degrees and norms of estimated and ground-truth GRF vectors are different from each other 12% indicating the proposed model along with identifi-

cation methods are promising to be used for 3D GRF estimation.

It should be noted dramatic changes in acceleration leads to erroneous inclination angle estimations if the inclination angle estimation is based on accelerometer readings. Consequently, if the crutch is moved in a fast manner, the error under dynamic conditions increases drastically. To verify this, we conducted 5 more experiments in which the crutch angle β changed in a fast manner. Average, maximum, and minimum errors in both axes for these experiments are shown in Table 4.2.

Table 4.2: Average, minimum and maximum errors across 5 fast dynamic experiments in 2D.

	Avg.	Min	Max
$E_{rms,y}$ (%)	25.85 ± 3.17	17.78	38.01
$E_{rms,z}$ (%)	10.25 ± 2.71	8.16	42.26
E_{θ} (deg)	14.26 ± 2.93	11.87	19.38
E_L	0.14 ± 0.03	0.11	0.19

As seen from Table 4.2, fast move of the crutch leads high errors in the GRF vector estimation. Thus, the system should be moved at a reasonable velocity while being used under dynamic conditions.

4.3 GRF Estimation in 3-Dimensions

We used the same experimental platform in the 2D case which is shown in Figure 4.1. In this section, we will describe our method for mapping pressure sensor and accelerometer data to GRF vectors in 3D. More specifically, the method for estimation of F_x , F_y , and F_z from P_1 , P_2 , P_3 , P_4 , a_x , and a_y will be presented in this section. Subsequently, Section 4.4 will present the estimation performance.

4.3.1 Angle-Dependent Force Transfer Model

Similar to Section 4.1.1, our estimation method is again based on the assumption that the GRF vector in 3D depends quadratically on the pressure readings from the FSRs embedded in the silicon shoe. However, this time we have two crutch angles, namely

α and β which can be clearly seen from Figure 4.1. This means, the relationship can be different for different values of α and β . Furthermore, this time the delay caused by the soft silicon structure of the shoe will not only depend on α , but also will depend on β . According to these assumptions, the model in Equation 4.1 takes the form

$$\begin{aligned} \hat{\mathbf{F}}(t - t_d(\alpha, \beta)) &:= \begin{bmatrix} \hat{F}_x(t - t_d(\alpha, \beta)) \\ \hat{F}_y(t - t_d(\alpha, \beta)) \\ \hat{F}_z(t - t_d(\alpha, \beta)) \end{bmatrix} \\ &= \mathbf{K}_L(\alpha, \beta) \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{bmatrix} + \mathbf{K}_Q(\alpha, \beta) \begin{bmatrix} P_1^2(t) \\ P_2^2(t) \\ P_3^2(t) \\ P_4^2(t) \end{bmatrix}. \end{aligned} \quad (4.28)$$

In this model,

1. α is the angle of the crutch with the vertical along x-axis and is computed from accelerometer measurements along the x-axis using

$$\alpha = \sin^{-1}\left(\frac{a_x}{g}\right), \quad (4.29)$$

β has the same definition in Equation 4.1 which is the angle of the crutch with the vertical along y-axis and is computed by using Equation 4.2,

2. $P_i(t)$ are pressure readings from the FSR sensors,
3. $t_d(\alpha, \beta)$ is the angle-dependent sensor delay in the transfer of the GRF vector to the FSR sensors due to the soft structure of the silicon shoe,
4. $\mathbf{K}_L(\beta)$ is an angle-dependent linear gain matrix defined as

$$\mathbf{K}_L(\alpha, \beta) := \begin{bmatrix} k_{L,x1}(\alpha, \beta) & k_{L,x2}(\alpha, \beta) & k_{L,x3}(\alpha, \beta) & k_{L,x4}(\alpha, \beta) \\ k_{L,y1}(\alpha, \beta) & k_{L,y2}(\alpha, \beta) & k_{L,y3}(\alpha, \beta) & k_{L,y4}(\alpha, \beta) \\ k_{L,z1}(\alpha, \beta) & k_{L,z2}(\alpha, \beta) & k_{L,z3}(\alpha, \beta) & k_{L,z4}(\alpha, \beta) \end{bmatrix}, \quad (4.30)$$

5. $\mathbf{K}_Q(\beta)$ is an angle-dependent quadratic gain matrix defined as

$$\mathbf{K}_Q(\alpha, \beta) := \begin{bmatrix} k_{Q,x1}(\alpha, \beta) & k_{Q,x2}(\alpha, \beta) & k_{Q,x3}(\alpha, \beta) & k_{Q,x4}(\alpha, \beta) \\ k_{Q,y1}(\alpha, \beta) & k_{Q,y2}(\alpha, \beta) & k_{Q,y3}(\alpha, \beta) & k_{Q,y4}(\alpha, \beta) \\ k_{Q,z1}(\alpha, \beta) & k_{Q,z2}(\alpha, \beta) & k_{Q,z3}(\alpha, \beta) & k_{Q,z4}(\alpha, \beta) \end{bmatrix}. \quad (4.31)$$

4.3.2 Data-Driven Identification of the Force-Transfer Model

We used the same method described in Section 4.1.2 to identify gain matrices in 3D. This time, the method requires the collection of sufficient amount of synchronized 3D GRF (F_x, F_y, F_z) , angle (α, β) , and pressure (P_1, P_2, P_3, P_4) data. Similar to Section 4.1.2, we assume that the sensor delay characteristic is known for a particular (α, β) crutch angle pair.

As in Section 4.1.2, suppose that we have a collection of data for a particular (α, β) pair consisting of N samples of ground truth force (F_x, F_y, F_z) , and pressure data (P_1, P_2, P_3, P_4) , recorded in a synchronized fashion to yield pressure and force vectors, \mathbf{P}_j and \mathbf{F}_j , with $j = 1, \dots, N$. Additionally, let us suppose that the delay T_d of the particular choice of (α, β) is known. With this information and the proposed force transfer model, we identify the gain matrices $\mathbf{K}_L(\alpha, \beta)$ and $\mathbf{K}_Q(\alpha, \beta)$ by using standard linear least squares methods. This time, our error metric is

$$E(\alpha, \beta, T_d) := \sum_{j=N_d}^N \|\hat{\mathbf{F}}(\mathbf{P}_j) - \mathbf{F}_{(j-N_d)}\|^2. \quad (4.32)$$

In this equation, $\hat{\mathbf{F}}(\mathbf{P}_j)$ denotes the estimation from the model introduced in Section 4.3.1, using the measured pressure data \mathbf{P}_j , and $\mathbf{F}_{(j-N_d)}$ denotes the measured and recorded ground truth force data delayed in time by N_d samples representing the delay T_d in Equation 4.28.

To be able to use linear least-squares, we rearrange the gain parameters in Equations 4.30 and 4.31 into three vectors \mathbf{u}_x , \mathbf{u}_y , and \mathbf{u}_z . \mathbf{u}_x is defined as

$$\mathbf{u}_x := \left[k_{L,x1} \quad \dots \quad k_{L,x4} \quad k_{Q,x1} \quad \dots \quad k_{Q,x4} \right]^T. \quad (4.33)$$

Definitions of \mathbf{u}_y and \mathbf{u}_z are given in Equations 4.6 and 4.7, respectively.

In 3D, ground truth force recordings for each axis are collected into three vectors \mathbf{F}_x , \mathbf{F}_y , and \mathbf{F}_z . \mathbf{F}_x is defined as

$$\mathbf{F}_x := \left[F_{x,1} \quad \dots \quad F_{x,(N-N_d)} \right]^T. \quad (4.34)$$

Definitions of \mathbf{F}_y and \mathbf{F}_z are given in Equations 4.8 and 4.9, respectively.

As in Section 4.1.2, pressure sensor readings are assembled in to matrix \mathbf{M} which is defined in Equation 4.10.

After throwing first N_d samples out of Equation 4.10 and also making lengths of vectors \mathbf{F}_x , \mathbf{F}_y , \mathbf{F}_z compatible with the matrix \mathbf{M} , we substitute these definitions into the error metric in Equation 4.32 yielding

$$E(\alpha, \beta, T_d, \mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z) = \|\mathbf{M}\mathbf{u}_x - \mathbf{F}_x\|^2 + \|\mathbf{M}\mathbf{u}_y - \mathbf{F}_y\|^2 + \|\mathbf{M}\mathbf{u}_z - \mathbf{F}_z\|^2 . \quad (4.35)$$

Based on this error metric, we estimate the gain parameters by independently using standard linear least-squares on first, second, and third terms of Equation 4.35. With this method we find estimated gain parameters $\hat{\mathbf{u}}_x$, $\hat{\mathbf{u}}_y$, and $\hat{\mathbf{u}}_z$. While $\hat{\mathbf{u}}_x$ results in

$$\hat{\mathbf{u}}_x = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}_x, \quad (4.36)$$

$\hat{\mathbf{u}}_y$ and $\hat{\mathbf{u}}_z$ are given in Equations 4.12 and 4.13, respectively.

The experimental procedure for assessing estimation performance of this method with corresponding detailed results are presented in Section 4.4.4.

4.3.3 Identification of Angle-Dependent Sensor Delay

Sensor delay caused by the soft structure of the silicon shoe is identified in 3D almost exactly the same way in Section 4.3.3. This time, we define the sensor delay for a particular (α, β) pair as $T_d(\alpha, \beta)$ and estimate t by solving the optimization problem,

$$\hat{t}_d(\alpha, \beta) = \underset{t_d}{\operatorname{argmin}} E(\alpha, \beta, t_d, \hat{\mathbf{u}}_x, \hat{\mathbf{u}}_y, \hat{\mathbf{u}}_z) , \quad (4.37)$$

based on the error metric given in Equation 4.35, and the gain vectors estimated by using Equations 4.36, 4.12, 4.13. As in the case of Section 4.3.3, we solve the optimization problem by using `fminbnd` function of MATLAB. The details of the experimental procedure with associated results are presented in Section 4.4.3.

4.3.4 Estimation of GRF Vectors for Arbitrary Crutch Angles

We extend the interpolation method presented in section 4.1.4 to 3D in order to estimate 3D gain matrices and the 3D sensor delay at angles for which no data were

collected. The analogue of the linear interpolation method used in Section 4.1.4 in 3D is called as barycentric interpolation. To better understand this method, suppose gain matrices and the sensor delay are identified at three different crutch angle pairs (α_1, β_1) , and (α_2, β_2) , (α_3, β_3) . We want to approximate gain matrices and the sensor delay at an arbitrary angle pair (α, β) inside of the triangle formed by (α_1, β_1) , (α_2, β_2) , and (α_3, β_3) . Figure 4.12 depicts the situation.

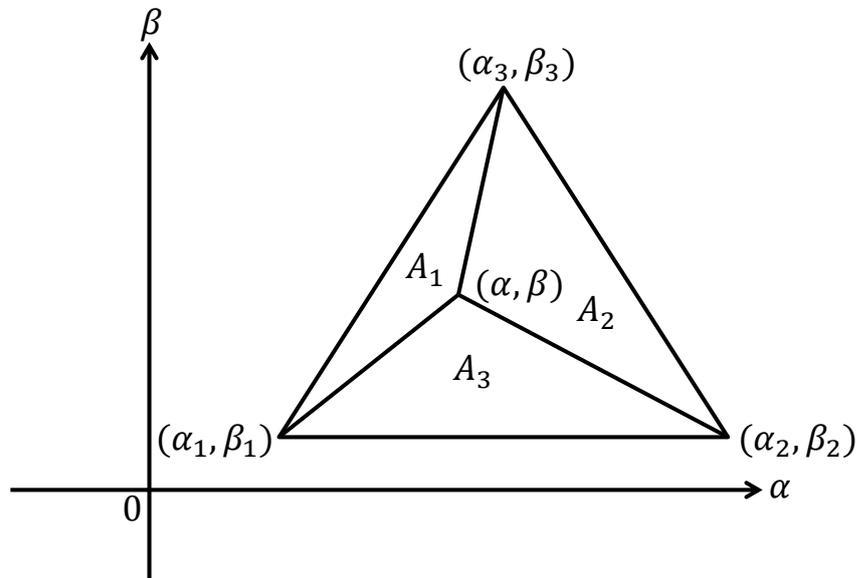


Figure 4.12: Three different crutch angle pairs (α_1, β_1) , and (α_2, β_2) , (α_3, β_3) at which gain matrices and the sensor delay are identified. (α, β) is the angle pair at which we want to approximate gain matrices and the sensor delay. A_1 , A_2 , and A_3 represents the areas of the small triangles on which they are written.

In Figure 4.12, A_1 , A_2 , and A_3 represents the areas of the small triangles on which

they are written and can be computed as,

$$A_1 := \frac{1}{2} \begin{vmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha & \beta & 1 \\ \alpha_3 & \beta_3 & 1 \end{vmatrix}, \quad (4.38)$$

$$A_2 := \frac{1}{2} \begin{vmatrix} \alpha_2 & \beta_2 & 1 \\ \alpha & \beta & 1 \\ \alpha_3 & \beta_3 & 1 \end{vmatrix}, \quad (4.39)$$

$$A_3 := \frac{1}{2} \begin{vmatrix} \alpha_1 & \beta_1 & 1 \\ \alpha_2 & \beta_2 & 1 \\ \alpha & \beta & 1 \end{vmatrix}, \quad (4.40)$$

respectively.

Moreover,

$$A_T := A_1 + A_2 + A_3 \quad (4.41)$$

represents the area of the large triangle formed by (α_1, β_1) , (α_2, β_2) , (α_3, β_3) . For the arbitrary (α, β) pair, we propose to estimate the gain matrices as,

$$\mathbf{K}_L(\alpha, \beta) = \mathbf{K}_L(\alpha_1, \beta_1) \frac{A_1}{A_T} + \mathbf{K}_L(\alpha_2, \beta_2) \frac{A_2}{A_T} + \mathbf{K}_L(\alpha_3, \beta_3) \frac{A_3}{A_T}, \quad (4.42)$$

$$\mathbf{K}_Q(\alpha, \beta) = \mathbf{K}_Q(\alpha_1, \beta_1) \frac{A_1}{A_T} + \mathbf{K}_Q(\alpha_2, \beta_2) \frac{A_2}{A_T} + \mathbf{K}_Q(\alpha_3, \beta_3) \frac{A_3}{A_T}. \quad (4.43)$$

Similarly, we can estimate the angle-dependent sensor delay as

$$T_d(\alpha, \beta) = T_d(\alpha_1, \beta_1) \frac{A_1}{A_T} + T_d(\alpha_2, \beta_2) \frac{A_2}{A_T} + T_d(\alpha_3, \beta_3) \frac{A_3}{A_T} \quad (4.44)$$

under the assumption that the dependence of gains and the delay on β is locally approximately linear.

Given a set of (α, β) pairs in the α - β plane with associated gain matrices and sensor delays at these points, the method requires the triangulation of the plane so that the arbitrary angle pair resides in one of the triangles. Any triangulation method is suitable for this interpolation method as long as the edges of the triangles do not cross each other.

Details of the experimental procedure with associated results in estimating gain matrices and the sensor delay for arbitrary angle pairs are presented in Section 4.4.5.

4.4 Experimental Results for 3-Dimensions

4.4.1 The Experimental Procedure

The strategy described in 4.2.1 is followed to assess validity of the arguments proposed for 3D GRF estimation. First, stationary experiments were conducted, in which the crutch angle pair (α, β) was kept constant. Second, dynamic experiments were conducted, in which the crutch angle pair (α, β) was varied reasonably slowly.

As the data acquisition procedure and experiments are much more tedious in 3D than the case of 2D, we collected data only in the first quadrant in the α - β plane in the intervals $\alpha \in [0^\circ, 14^\circ]$, $\beta \in [0^\circ, 14^\circ]$. To be able to better evaluate the performance of the method on the axes, we also collected data in the intervals $\alpha \in [-2^\circ, 6^\circ]$, $\beta \in [-2^\circ, 6^\circ]$ also. Data collected in these intervals are used to evaluate the performance of the method described in Section 4.3.4. Figure 4.13 shows exact pairs of (α, β) at which data are collected in the α - β plane. At each angle pair, data were collected using the platform shown in 4.1, recording

- pressure data (P_1, P_2, P_3, P_4) ,
- acceleration data (a_x, a_y) which is later converted to α and β by using Equation 4.29 and 4.2, respectively,
- ground truth values of GRF data (F_x, F_y, F_z) ,

while external forces were applied to the crutch, keeping α and β as constant as possible exactly in the same way described in Section 4.2.1. We conducted more experiments in 3D than in 2D, a trial of 17 experiments for each angle pair which was 11 in 2D, because we need more data points to learn the gain parameters and sensor delay characteristic in 3D than in the case of 2D. The associated data for each angle pair was again filtered and recorded to be used in training the model and evaluation of its performance. Data collected for $(\alpha = 2^\circ, \beta = 2^\circ)$ are illustrated in Figure 4.14.

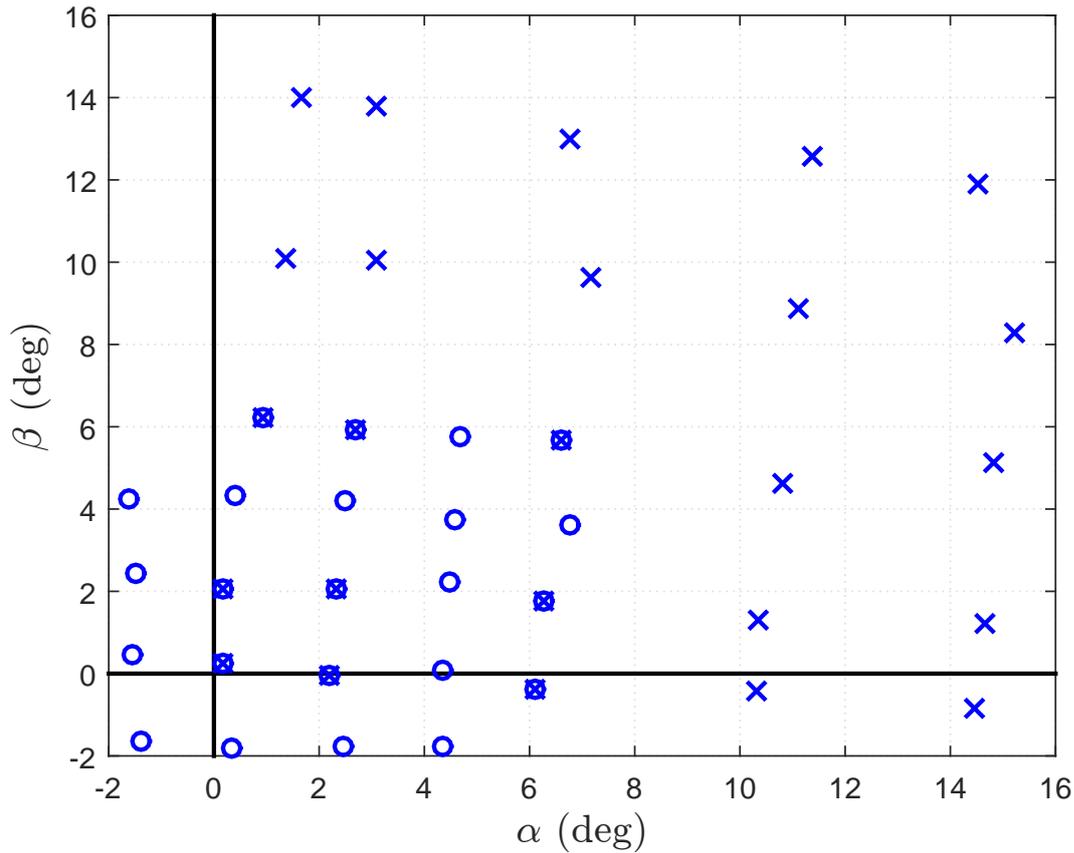


Figure 4.13: (α, β) angle pairs at which data were collected. Points with ‘x’ represent data used for training the model for dynamic experiments. Points with ‘o’ represent data used for testing the method to extend to model to arbitrary crutch angles. Points with both ‘x’ and ‘o’ represent data used for both purposes.

Dynamic experiments were conducted almost exactly in the way described in 4.2.1, conducting 20 experiments to measure the performance of the model under dynamic conditions, this time in 3D only in the first quadrant as we identified the gain matrices and the sensor delay characteristic only for the first quadrant. In these 20 experiments, α and β varied with a reasonably slow angular velocity in the intervals $\alpha \in [0^\circ, 14^\circ]$, $\beta \in [0^\circ, 14^\circ]$. Moreover, as in the case of 2D, we conducted 5 more dynamic experiments in which α and β were varied fast to state the accuracy of the model under high speed dynamic conditions.

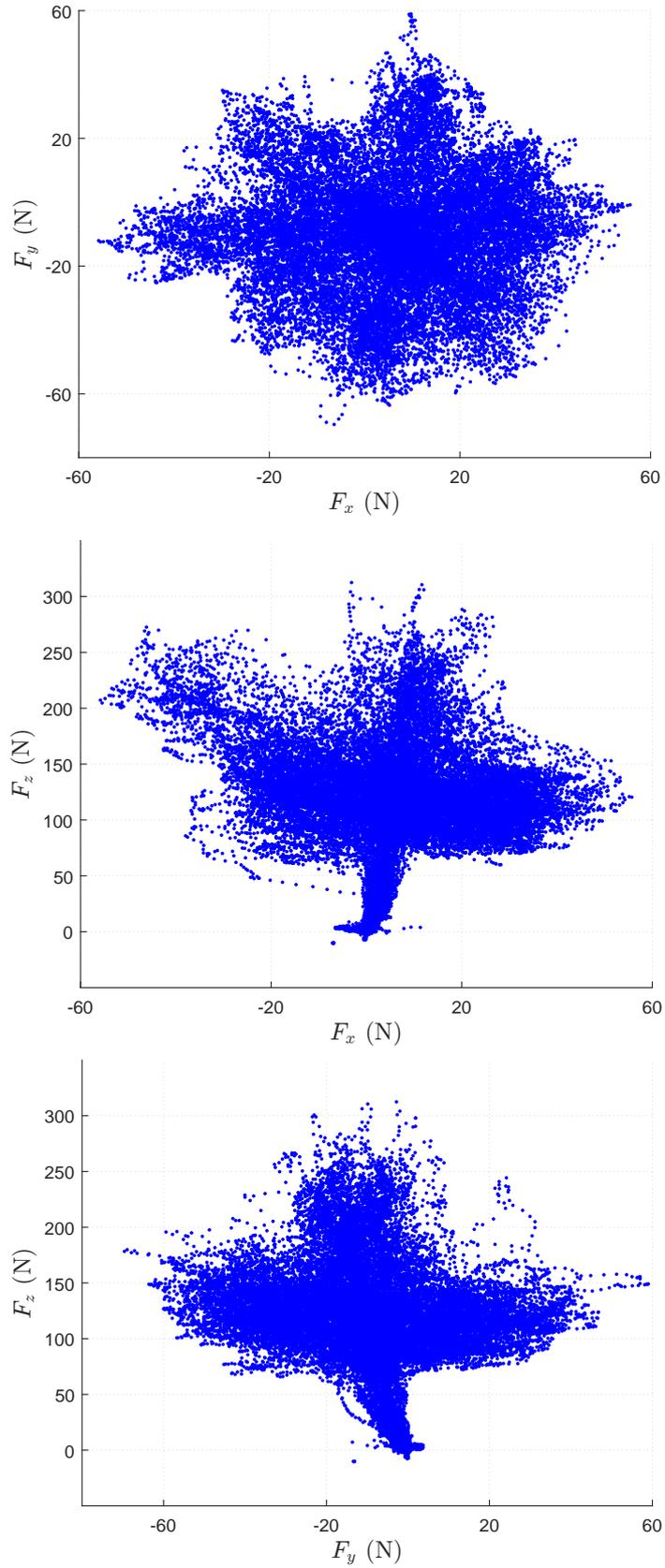


Figure 4.14: Scatter plot of ground reaction force samples collected during all of the 17 experiments for $(\alpha = 2^\circ, \beta = 2^\circ)$.

4.4.2 Error Metrics

To evaluate the performance of the proposed system identification method, RMS errors in three axes were computed. To find RMS errors in y and z axes, Equations 4.18 and 4.19 are used, respectively. In 3D, we add

$$E_{rms,x} := \frac{100}{\max_j \|\mathbf{F}_j\|} \sqrt{\frac{1}{N} \sum_{j=1}^N (\hat{F}_{j,x} - F_{j-N_d,x})^2} \quad (4.45)$$

to Equations 4.18 and 4.19 to compute the RMS error in x axis.

As mentioned earlier, 17 experiments were conducted for each (α, β) angle pair during stationary experiments. To measure the average estimation errors in each of the three axis for a particular crutch angle pair (α, β) , arithmetic means of errors for all experiments at that angle pair were computed as

$$\bar{E}_{rms,x} := \frac{1}{17} \sum_{i=1}^{17} E_{rms,x,i}, \quad (4.46)$$

$$\bar{E}_{rms,y} := \frac{1}{17} \sum_{i=1}^{17} E_{rms,y,i}, \quad (4.47)$$

$$\bar{E}_{rms,z} := \frac{1}{17} \sum_{i=1}^{17} E_{rms,z,i}, \quad (4.48)$$

$$\bar{E}_{\Theta} := \frac{1}{17} \sum_{i=1}^{17} E_{\Theta,i}, \quad (4.49)$$

$$\bar{E}_L := \frac{1}{17} \sum_{i=1}^{17} E_{L,i}. \quad (4.50)$$

In the sequel, these error metrics will be used to evaluate the model performance.

4.4.3 Angle-Dependent Sensor Delay Characteristic

In the first phase of the experiments, we identified the sensor delay $T_d(\alpha, \beta)$ for crutch angle pairs at which data were collected by using the method described in 4.3.3. Similar to the case in 2D, we used leave-one-out cross validation method, that is, from among 17 sets of each data for each (α, β) angle pair, 16 were used for training while the remaining 1 data set was used for testing. This was repeated with 17 data

sets. At a particular (α, β) , the training set was used to compute \hat{u}_x , \hat{u}_y , and \hat{u}_z in Equation 4.37 and $T_d(\alpha, \beta)$. For each (α, β) , the arithmetic mean of sensor delays estimated in each leave-one-out run at a particular (α, β) angle pair was computed to estimate the sensor delay at that particular (α, β) angle pair. The identified sensor delay characteristic in the first quadrant is shown in Figure 4.15.

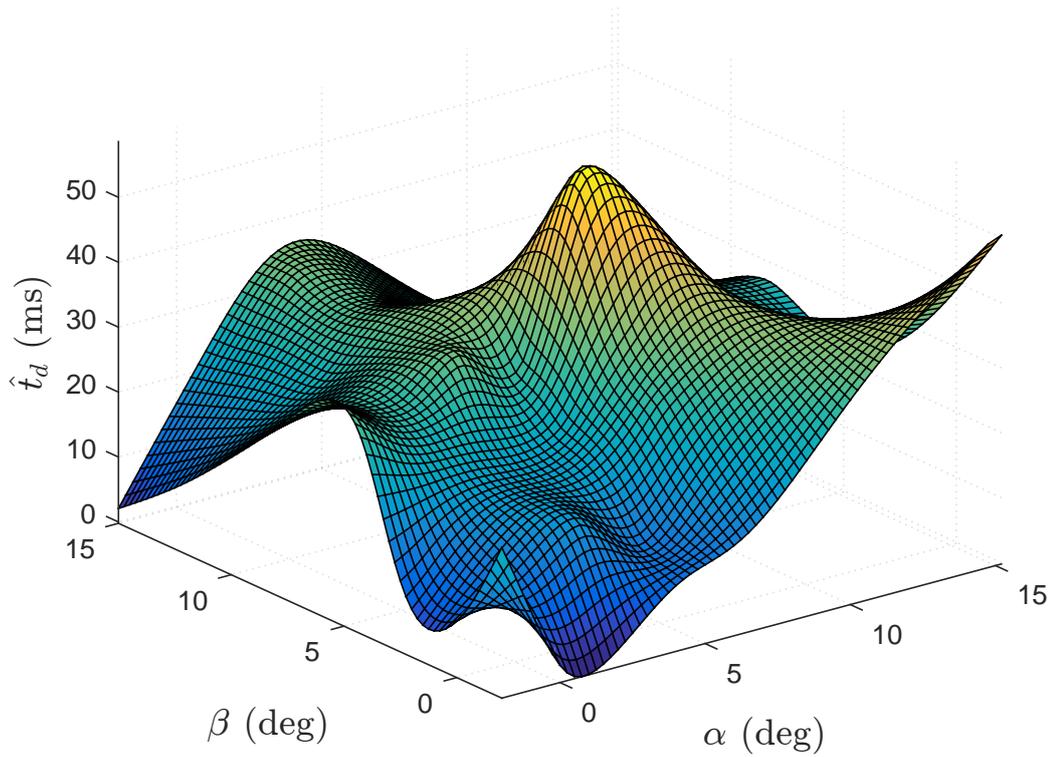


Figure 4.15: The graph of the estimated sensor delay (in milliseconds) vs. (α, β) (in degrees) for the first quadrant. For each (α, β) , estimations from 17 experiments were averaged.

We expect that the sensor delay would be greatest around $(\alpha = 0^\circ, \beta = 0^\circ)$ and gets lower to the edge angles as the compaction volume of the soft silicon structure of the tip is largest at vertical angles and gets lower towards edge tilt angles. However, Figure 4.15 does not confirm this claim. One cause of this might be the fact that the forces applied to train the model were not distributed homogeneous enough in the $F_x-F_y-F_z$ space as it can be seen from Figure 4.14. Another cause might be the possibility of nonhomogeneous stiffness of the soft silicon structure. To say more clearly, a sub-volume of the silicon tip near edge can be softer than a sub-volume around center which causes unexpected sensor delay characteristic in Figure 4.15.

4.4.4 Model Accuracy for Crutch Angles Used for Training

Sensor delay characteristic as a function of (α, β) were identified up to this point and associated results were presented in Section 4.4.3. In this phase of the experiments, we assess validity of gain parameters and this estimated sensor delay characteristic at the crutch angles used for training the model. We follow the same strategy with the counterpart of this phase of experiments in 2D which was presented in Section 4.2.4. That is, leave-one-out cross validation is used. In each of the 17 runs for each (α, β) angle pair, 16 out of 17 data sets were used for training, while the remaining data set was used for testing. Errors computed for the testing data set in each run were arithmetically averaged by using Equations 4.46, 4.47, 4.48, 4.49, and 4.50. Graphs presenting average errors in the estimation of F_x , F_y , and F_z are shown in Figure 4.16. Graphs presenting errors between the ground-truth GRF vectors and estimated GRF vectors in terms of degrees and in terms of ratios of their norms are shown in Figure 4.17. As exact positions of the angle pairs and associated error values cannot be seen clearly, and standard deviations of the errors are not represented in Figures 4.16 and 4.17, exact positions of the angle pairs with associated error values and standard deviations are presented in Table 4.3.

As seen from Figure 4.16, errors in all of the x, y, and z axis are below 8% for almost all (α, β) angle pairs in the first quadrant. Moreover, as seen from Figure 4.17, angle between the ground-truth and estimated GRF vector is below 10 degrees, and difference between the norms of the estimated and ground-truth GRF vectors are below 10% for almost all angle pairs. From the figures, we observe that error increases around $(\alpha = 6^\circ, \beta = 6^\circ)$. One reason of this increase might be the possibility of deflections arising inside the silicon tip when $(\alpha = 6^\circ, \beta = 6^\circ)$ which may cause low quality pressure sensor data. The deflection might be observed also from the high sensor delay values in Figure 4.15. Furthermore, it can be observed from the figures that the error tends to increase to the edge angles. This is expected because quality of the pressure signals decreases as tilt angle of the crutch increases. Nevertheless, the results still show that identified gain parameters and sensor delays at the training angles are valid enough to train the model to be used for GRF estimation under time-varying conditions.

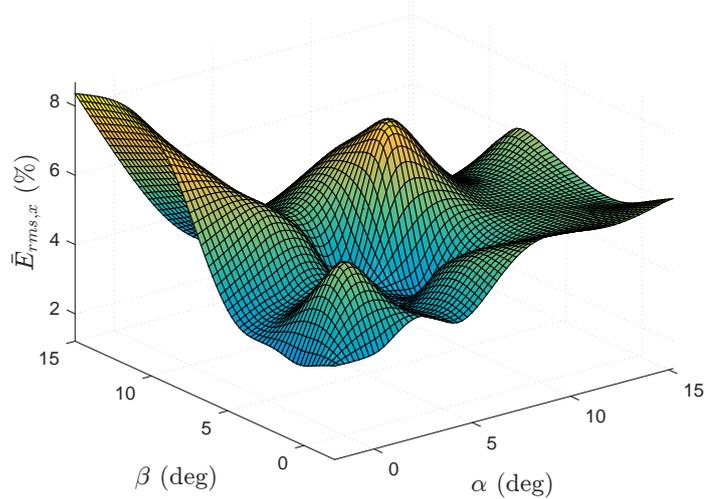
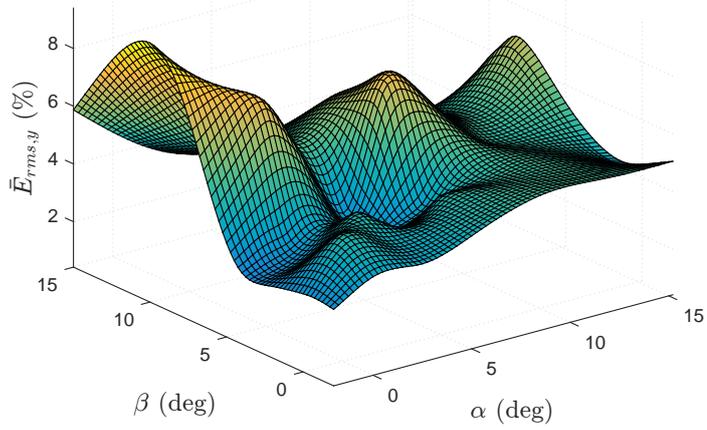
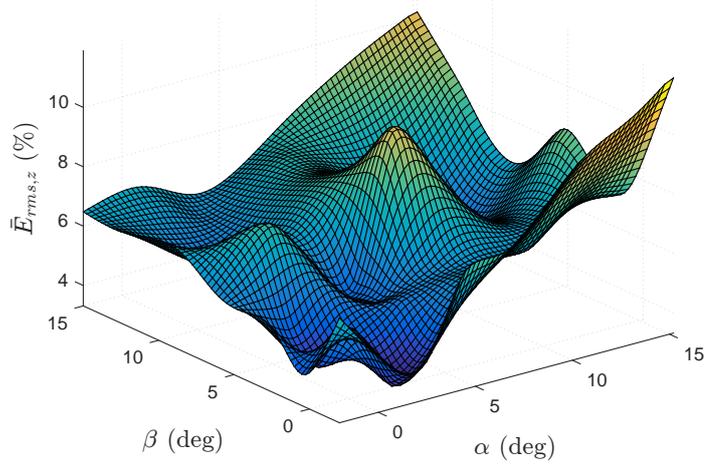


Figure 4.16: Percentage errors in the estimates of F_z (top graph), F_y (middle graph), and F_x (bottom graph) vs. (α, β) . For each (α, β) , resulting errors for 17 experiments are averaged arithmetically.

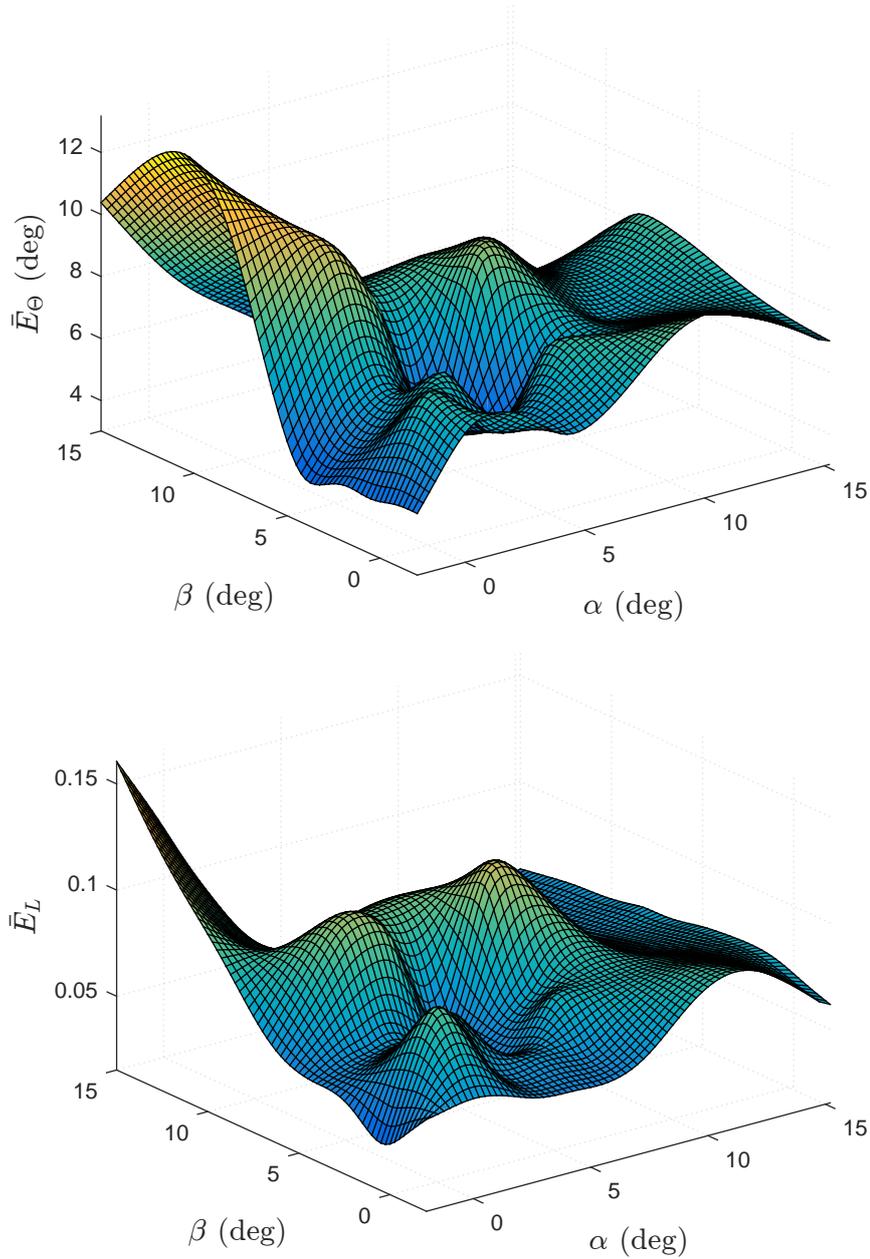


Figure 4.17: Errors in terms of the angles between the ground-truth GRF vector and the estimated GRF vector (bottom graph, in degrees) and ratio of norm of the ground-truth vector and the estimated GRF vector (bottom graph) vs. (α, β) . For each (α, β) , resulting errors for 17 experiments are averaged arithmetically.

Table 4.3: Exact values of errors with corresponding standard deviation for each data pair at which data collected. \bar{E}_L values are written after divided by 100.

(α, β)	$\bar{E}_{rms,x}$ (%)	$\bar{E}_{rms,y}$ (%)	$\bar{E}_{rms,z}$ (%)	\bar{E}_θ (deg)	$\bar{E}_L/100$
-1.39,-1.64	4.03 ± 0.54	3.42 ± 0.53	6.1 ± 2.77	5.78 ± 0.86	5 ± 5
0.34,-1.83	4.23 ± 0.29	3.97 ± 0.54	4.98 ± 0.94	7.42 ± 0.72	6 ± 1
2.45,-1.76	4.67 ± 0.39	3.86 ± 1.1	4.38 ± 1.23	7.07 ± 0.9	6 ± 1
4.36,-1.75	4.24 ± 0.67	4.17 ± 0.49	6.73 ± 2.62	6.28 ± 0.8	5 ± 1
-1.57,0.45	4.08 ± 0.32	3.43 ± 0.37	4.65 ± 1.1	5.47 ± 0.45	5 ± 1
0.17,0.27	6.08 ± 1.33	4.68 ± 1.34	5.62 ± 1.52	7.81 ± 1.45	10 ± 3
2.2,-0.04	4.8 ± 0.87	4.18 ± 0.93	3.37 ± 0.44	6.32 ± 1.44	6 ± 3
4.35,0.09	4.22 ± 0.49	4.24 ± 0.99	4.28 ± 0.66	6.42 ± 1.11	6 ± 1
6.11,-0.37	5.27 ± 1.4	4.67 ± 0.81	5.99 ± 2.6	7.32 ± 1.36	6 ± 1
10.32,-0.44	5.51 ± 0.52	5.08 ± 0.69	8.23 ± 1.98	8.35 ± 0.79	9 ± 3
14.47,-0.86	5.8 ± 0.47	4.77 ± 0.64	9.94 ± 2.98	7.29 ± 0.73	7 ± 1
-1.47,2.44	3.93 ± 0.75	2.97 ± 0.53	5.03 ± 1.82	5.16 ± 0.79	9 ± 5
0.18,2.05	4.96 ± 0.56	3.67 ± 0.73	3.89 ± 0.54	6.48 ± 0.92	7 ± 2
2.33,2.06	4.7 ± 0.65	4.66 ± 0.63	4.21 ± 0.68	7.52 ± 1.59	8 ± 2
4.48,2.21	3.81 ± 0.8	3.52 ± 0.49	5.25 ± 1.94	5.17 ± 0.96	5 ± 1
6.27,1.75	4.73 ± 0.82	4.56 ± 0.64	5.76 ± 2.65	7.63 ± 1.19	8 ± 2
10.35,1.3	5.24 ± 0.87	5.05 ± 0.9	6.24 ± 2.37	7.69 ± 1.21	8 ± 1
14.67,1.21	5.38 ± 0.75	4.34 ± 0.39	7.35 ± 1.88	7.3 ± 0.77	7 ± 2
-1.62,4.24	3.86 ± 0.44	3.16 ± 0.49	5.37 ± 1.83	5 ± 0.63	5 ± 1
0.4,4.33	4.35 ± 0.55	3.82 ± 0.42	4.72 ± 1.04	6.7 ± 1.18	6 ± 1
2.5,4.21	4.58 ± 0.83	3.92 ± 0.68	5.51 ± 1.08	6.15 ± 0.85	6 ± 1
4.59,3.73	3.91 ± 1.03	3.14 ± 0.73	4.99 ± 0.95	4.63 ± 0.94	5 ± 1
6.77,3.6	3.68 ± 1.32	3.88 ± 1.44	6.31 ± 3.36	5.04 ± 1.78	7 ± 3
0.93,6.21	6.1 ± 0.6	7.44 ± 0.67	7.47 ± 3.19	10.24 ± 1.11	10 ± 2
2.69,5.94	5.36 ± 0.46	5.45 ± 0.7	6.12 ± 1.38	8.29 ± 0.76	10 ± 2
4.68,5.76	3.79 ± 0.67	3.43 ± 0.75	5.07 ± 1.28	4.16 ± 0.74	5 ± 1
6.61,5.68	7.4 ± 0.69	7.16 ± 0.81	9.55 ± 2.57	8.86 ± 0.87	1 ± 3
10.91,4.63	5.36 ± 0.62	4.95 ± 0.69	6.38 ± 1.49	6.82 ± 0.6	8 ± 1
14.82,5.14	5.65 ± 0.59	5.42 ± 0.6	8.3 ± 2.09	7.96 ± 0.93	7 ± 1
1.36,10.09	6.25 ± 0.61	6.09 ± 1.15	5.21 ± 1.12	10.02 ± 1.3	8 ± 1
3.09,10.03	4.96 ± 0.68	4.51 ± 0.72	5.74 ± 1.47	8.12 ± 1.4	8 ± 3
7.18,9.62	6.47 ± 0.78	6.04 ± 1.22	7.49 ± 3.86	8.22 ± 1.62	8 ± 7
11.12,8.86	5.14 ± 0.6	4.8 ± 0.64	4.55 ± 1.24	7.18 ± 0.8	6 ± 1
15.21,8.26	6.09 ± 0.7	6.9 ± 1.13	6.64 ± 2.27	8.29 ± 0.96	7 ± 2
1.68,14.02	5.05 ± 0.87	4.23 ± 0.86	6.27 ± 1.99	7.47 ± 1.29	8 ± 1
3.07,13.78	4.14 ± 0.63	3.73 ± 0.53	5.79 ± 1.56	1.6 ± 1	5 ± 1
6.77,12.97	4.13 ± 0.46	4.06 ± 0.83	7.2 ± 2.86	5.58 ± 0.8	5 ± 1
11.38,12.56	3.39 ± 0.67	3.35 ± 0.72	6.85 ± 1.62	4.93 ± 1.03	5 ± 1
14.52,11.9	3.57 ± 0.79	3.51 ± 0.93	7.88 ± 3.04	5.68 ± 1.35	6 ± 4

4.4.5 Model Accuracy for Arbitrary Crutch Angles

To assess the performance of the interpolation method proposed in Section 4.3.4 we used data collected at the points represented with ‘o’ and the points represented with both ‘o’ and ‘x’ in Figure 4.13 are used. Exact positions of the points with associated labels used to test the interpolation method are shown in Figure 4.18.

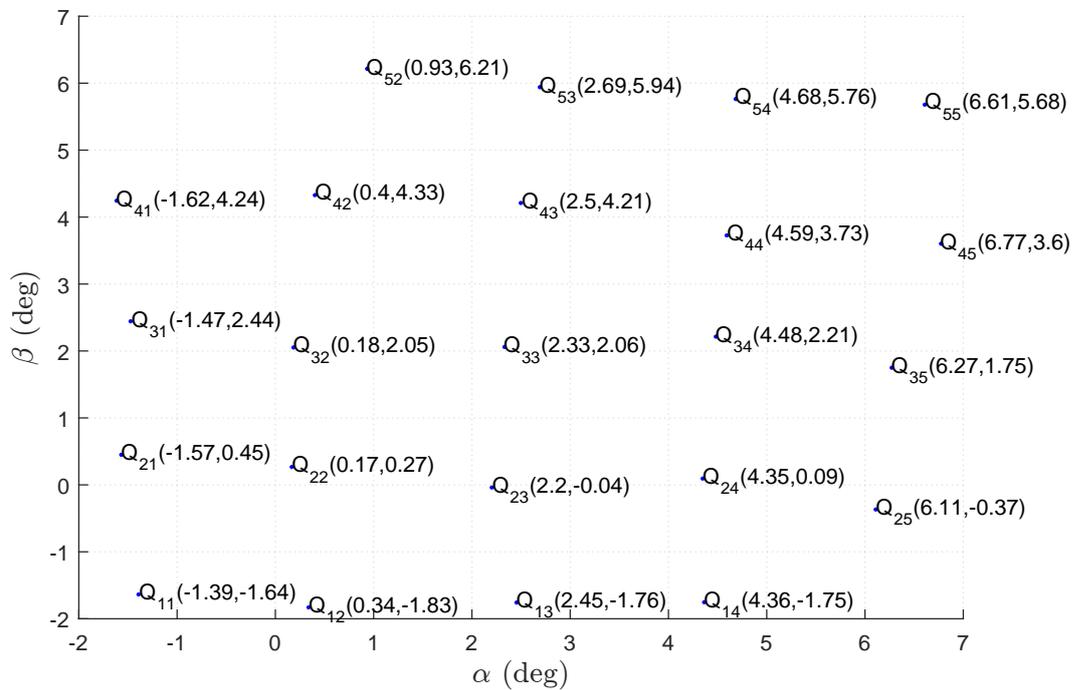


Figure 4.18: Exact positions of (α, β) pairs with associated labels used to test the proposed interpolation method.

To approximate gain parameters and the sensor delay at a data point, we used closest three points on which it resides almost at the middle of the triangle formed by them. 9 experiments were conducted to test the proposed interpolation method. Points used in the approximation and point to be approximated with associated errors and standard deviations are shown in Table 4.4.

Table 4.4: Resulting errors with corresponding standard deviations using the proposed interpolation method. $Point_a$ represents point at which gain parameters and the sensor delay wanted to be approximated. $Point_i$ represents the edges of the triangle used in the interpolation. Exact positions of the points with associated labels are shown in 4.18.

<i>Experiment#</i>	1	2	3	4	5	6	7	8	9
$Point_a$	Q_{22}	Q_{23}	Q_{24}	Q_{32}	Q_{33}	Q_{34}	Q_{42}	Q_{43}	Q_{44}
$Point_1$	Q_{11}	Q_{12}	Q_{33}	Q_{42}	Q_{22}	Q_{23}	Q_{41}	Q_{32}	Q_{33}
$Point_2$	Q_{13}	Q_{14}	Q_{13}	Q_{21}	Q_{24}	Q_{44}	Q_{33}	Q_{34}	Q_{35}
$Point_3$	Q_{32}	Q_{33}	Q_{25}	Q_{23}	Q_{43}	Q_{25}	Q_{53}	Q_{53}	Q_{54}
$\bar{E}_{rms,x}$	9.36	5.76	4.41	6.73	6.18	5.79	5.97	6.05	3.93
	±	±	±	±	±	±	±	±	±
	2.15	0.61	0.8	0.85	0.83	2.21	0.84	0.91	1.4
$\bar{E}_{rms,y}$	14.47	5.81	5.75	5.39	5.93	4.04	5.02	5.24	4.89
	±	±	±	±	±	±	±	±	±
	5.06	0.86	1.53	0.85	0.95	0.83	1.28	1.67	1.23
$\bar{E}_{rms,z}$	6.94	4.02	4.25	5.32	4.63	5.94	4.99	7.57	5.08
	±	±	±	±	±	±	±	±	±
	2.37	0.54	0.69	0.81	0.56	1.88	1.03	1.07	0.9
\bar{E}_Θ	11.89	6.97	7.66	8.41	8.82	6.96	8.22	8.13	6.04
	±	±	±	±	±	±	±	±	±
	2.2	0.9	0.98	1.07	1.65	2.29	1.69	1.06	1.27
\bar{E}_L	0.12	0.08	0.07	0.09	0.08	0.06	0.08	0.13	0.06
	±	±	±	±	±	±	±	±	±
	0.04	0.01	0.02	0.02	0.01	0.01	0.02	0.02	0.01

4th and 7th experiments test the performance of the method in the transition between the first and second quadrants as the point to be approximated lies on the positive side of y-axis. As seen from Table 4.4, errors in all axis are below 6%, angle and difference of the norms between the estimated and ground-truth GRF vectors are below 9 degrees and 9%, respectively. These low errors states that transition between first and second quadrants does not increase error.

2nd and 3rd experiments test the performance of the method in the transition between the first and fourth quadrants as the point to be approximated lies on the positive side of x-axis. As seen from Table 4.4, errors in all axis are below 6%, angle and difference of the norms between the estimated and ground-truth GRF vectors are below 8 degrees and 8%, respectively. These low errors states that transition between

first and fourth quadrants does not increase error.

1st experiment tests the performance of the method in the transition between the first and third quadrants as the point to be approximated lies on the origin. As seen from Table 4.4, errors in all axes increase considerably which is further approved by the increase in the errors in polar coordinates. These increases are due to the fact that while we are approximating gain parameters and sensor delay at origin, we use points from different quadrants.

5th, 6th, 8th, and 9th experiments test performance of the method only in the first quadrant without quadrant transition. As seen from Table 4.4, errors in all axis are below 8%, angle and difference of the norms between the estimated and ground-truth GRF vectors are below 9 degrees and 8%, respectively. These low errors states that if the method is only used in the first quadrant, the errors do not increase considerably.

4.4.6 Accuracy Under Dynamic Conditions

Similar to Section 4.2.6, to test the performance of the model under dynamic conditions, we cannot directly use the model in Equation 4.28 as it requires the knowledge of pressure sensor values ahead of time due to sensor delay. Thus, we have to delay the GRF vector estimation. Let us call the maximum value of the sensor delay among all (α, β) angle pairs as

$$t_d^{max} := \max \hat{t}_d(\alpha, \beta). \quad (4.51)$$

As in 2D case, if we delay pressure sensor signals to be processed by t_d^{max} , we may be able to estimate GRF vectors in real-time. The model in Equation 4.28 becomes

$$\begin{aligned} \hat{\mathbf{F}}(t - t_d^{max}) = & \mathbf{K}_L(\alpha(t), \beta(t)) \mathbf{P}(t + t_d(\alpha(t), \beta(t)) - t_d^{max}) \\ & + \mathbf{K}_Q(\alpha(t), \beta(t)) \mathbf{P}^2(t + t_d(\alpha(t), \beta(t)) - t_d^{max}). \end{aligned} \quad (4.52)$$

Points represented with ‘x’ and both ‘x’ and ‘o’ in Figure 4.13 are used to train the model. To triangulate the points in order to be able to use the interpolation model proposed in Section 4.3.4 we used Delaunay triangulation. Please note that any triangulation method is suitable as long as the edges of the triangles do not cross each other. Resulting triangulation is shown in Figure 4.19.

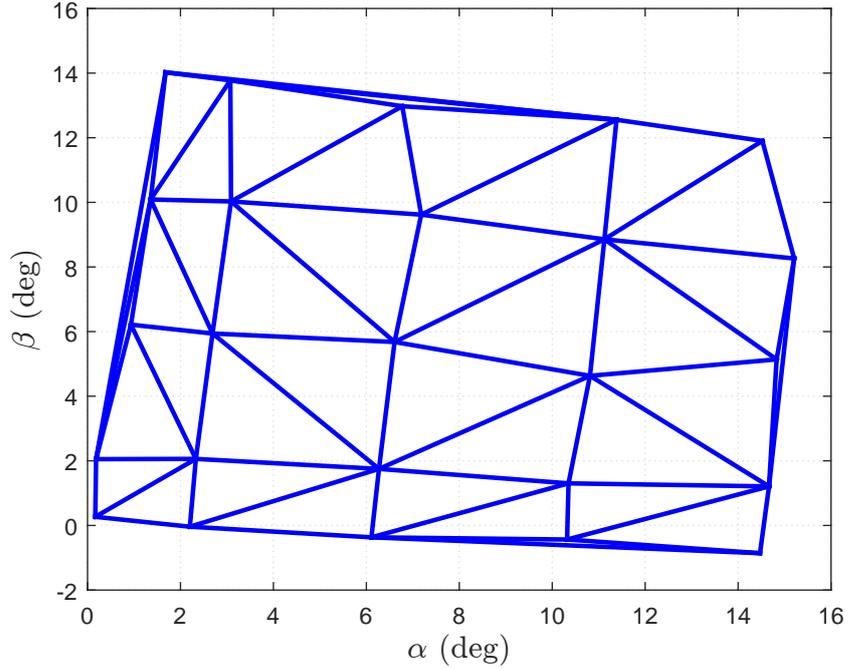


Figure 4.19: Resulting Delaunay triangulation of the points used in the learning process to use the proposed interpolation method.

Very similar to 2D case, $t_d^{max} = 58ms$ according to Figure 4.15. The potential flaw originating from the fact that the sensor delay corresponding to current (α, β) is used. As mentioned in Section 4.4.1, 20 experiments were conducted to measure performance of the model under dynamic conditions in which tip of the crutch is kept still while touching the ground and body of the crutch is moved such that (α, β) will be in the first quadrant in α - β plane. Graphs presenting estimation results in cartesian coordinates in one of these experiments are shown in Figure 4.20, and graphs presenting estimation results in polar coordinates of the same experiment are shown in Figure 4.21. Moreover, average, maximum, and minimum errors in both axes for 20 dynamic experiments are shown in Table 4.5.

To test the performance of the method with high crutch velocity, we conducted 5 more experiments. Average, maximum, and minimum errors in terms of all the five error metrics for 5 fast dynamic experiments are shown in 4.6.

As seen from Table 4.5, average percentage RMS error is below 6% in z-axis and are around 8% in shear axes. Moreover, angle between the estimated and ground-truth GRF vectors is around 10 degrees and norms of the estimated and ground-truth GRF

vectors are different from each other 9%.

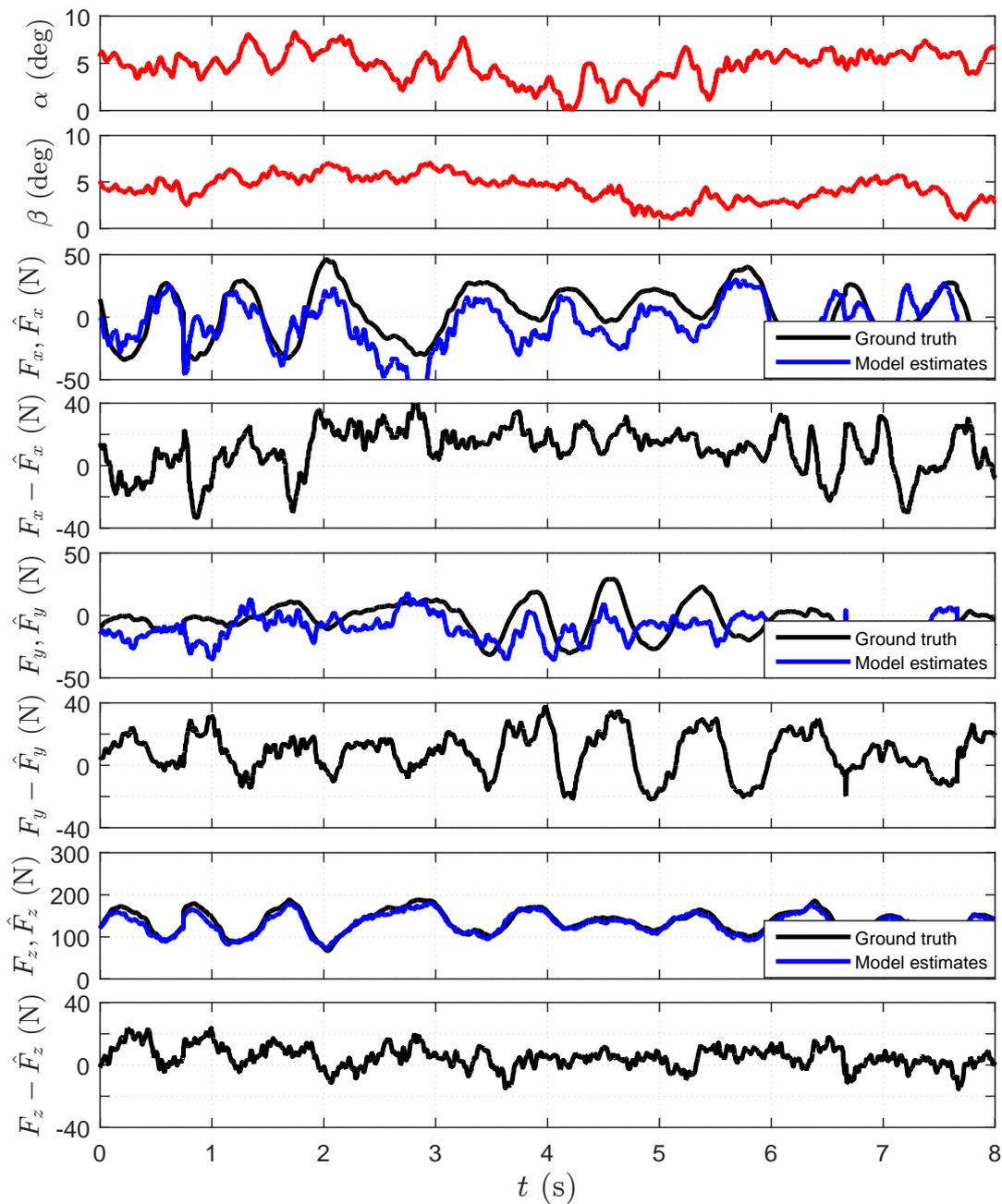


Figure 4.20: Graphs showing estimation results in cartesian coordinates for one of the dynamic experiments. The two topmost graphs show α and β as a function of time, respectively. Black trajectories in the third, fifth, and seventh from top graphs show F_x , F_y , and F_z , respectively. Blue trajectories in the same graphs show \hat{F}_x , \hat{F}_y , and \hat{F}_z , respectively. The fourth, sixth, and eighth graphs show the difference between the ground-truth and estimated forces in x, y, and z axis, respectively.

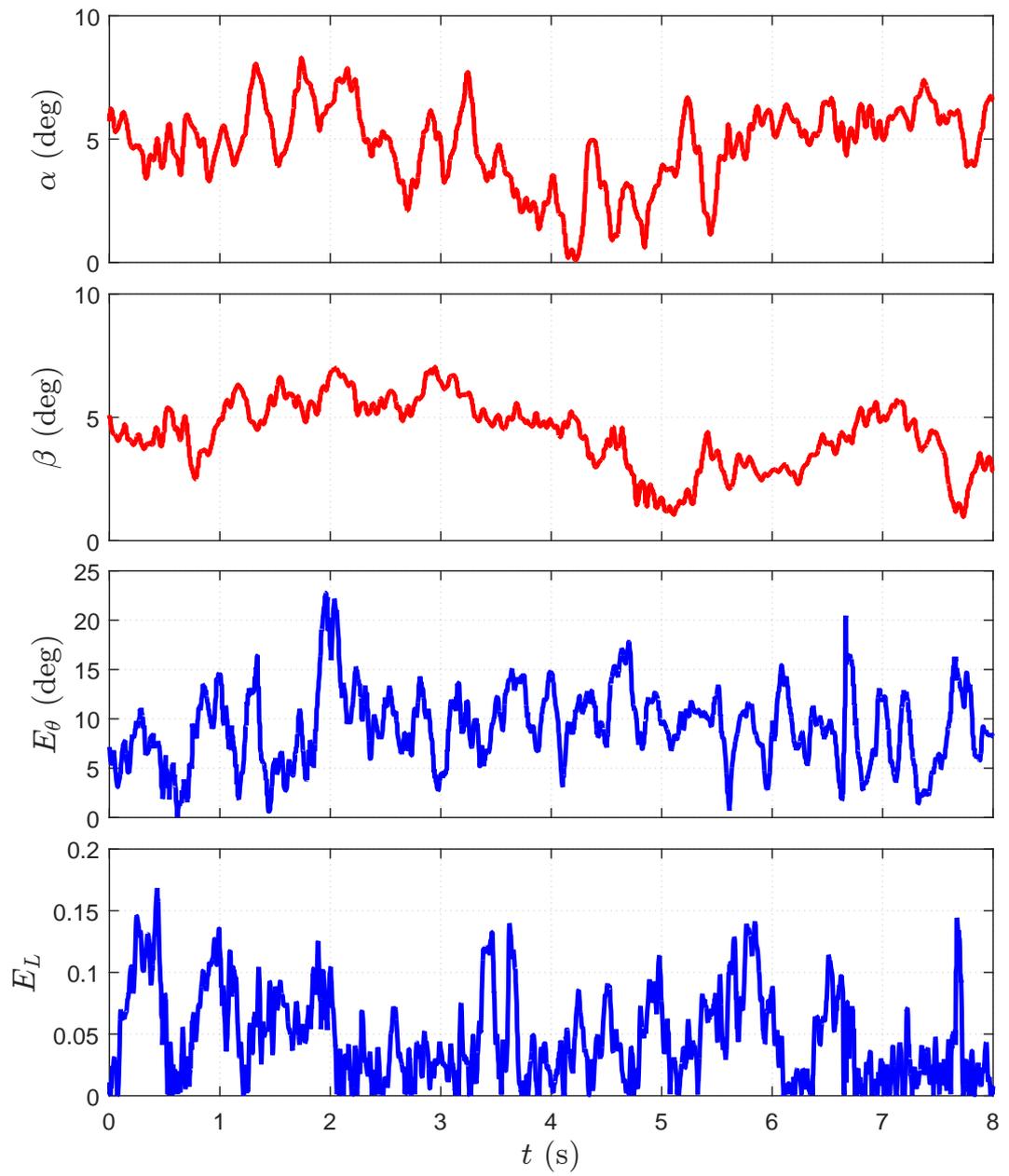


Figure 4.21: Graphs showing estimation results in polar coordinates for one of the dynamic experiments.

As in the 2D case, dramatic changes in acceleration leads to erroneous inclination angle estimations which can be confirmed from the high error values in Table 4.6.

Table 4.5: Average, minimum and maximum errors across 20 dynamic experiments in 3D.

	Avg.	Min	Max
$E_{rms,x}$ (%)	8.17 ± 1.72	5.8	12.93
$E_{rms,y}$ (%)	7.62 ± 1.33	6.08	10.93
$E_{rms,z}$ (%)	5.82 ± 1.18	3.13	7.66
E_θ (deg)	9.8 ± 1.48	7.31	12.49
E_L	0.09 ± 0.02	0.06	0.12

Table 4.6: Average, minimum and maximum errors across 5 fast dynamic experiments in 3D.

	Avg.	Min	Max
$E_{rms,x}$ (%)	49.11 ± 23.08	24.27	76.01
$E_{rms,y}$ (%)	42.19 ± 29.25	17.60	90.67
$E_{rms,z}$ (%)	18.53 ± 10.10	7.16	34.77
E_θ (deg)	22.78 ± 3.32	7.31	12.49
E_L	0.29 ± 0.12	0.12	0.44

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this study, we have introduced a low-cost crutch system instrumented with pressure sensors and an inclinometer to estimate ground reaction forces, towards the long-term goal of using GRF information to estimate user intention for controlling lower-body robotic orthosis.

The work consists of two major parts. We first designed of smart crutch platform with its mechanical structure, electronic design, communication infrastructure, and data acquisition platform. Then, we proposed quadratic model including sensory delays for how the GRF can be computed from pressure sensor measurements, whose gain matrices were computed using least-squares methods from systematically collected sensor and ground-truth force data. We cross validated our system identification results using the leave-one-out method, establishing that the resulting model can predict GRF vectors with less than 10% errors in all axes in 3D. We have also shown that the model successfully generalizes its predictions to crutch angles for which no data was collected, maintaining errors below 10% when training angles are separated by less than 4 degrees. Finally, we have also shown that the model performs well under dynamically changing crutch angles and external forces, maintaining average estimation errors below 10% in 3D.

The most important source of error in this system is the soft silicon structure of the manufactured tip which introduces high noise to the pressure sensors due to high vibration under high shear forces. To come over this issue, soft silicon of the tip should be replaced with a high quality rubber. Second most important source of error is using an accelerometer to measure tilt angle of the crutch. As mentioned, accelerometer

gives erroneous results under dynamic conditions which increases as the velocity increases. To overcome this problem, accelerometers will be replaced instead of gyro sensors which will not be affected by high acceleration changes. Another source of error is the fact that we computed only one sensor delay characteristic for all four pressure sensors. As compaction volume of the silicon tip under each sensor are different, computing delay characteristic of each pressure sensor separately is the most suitable approach. However, as this will require high computational power, we are not planning to do such a sensor delay identification in the future.

In the near future, we intend to use these GRF estimates from crutches to estimate user intention and commands, including the initiation of individual steps and the determination of step height and length, for our lower-body robotic orthosis platform. Finally, we hope to iterate on the SCP platform, improving estimation accuracy through more accurate sensing of crutch orientation by using gyro sensors, improved modelling of force transmission and better integration of pressure sensors by using a high quality rubber at the tip of the crutch.

In the far future, methods to map pressure sensor and orientation data obtained from the crutch to actuators at the hip and knee joints in the robotic orthosis system in real-time will be investigated. With this final system, patients who cannot move their lower extremities in any way will be able to walk completely independently.

REFERENCES

- [1] ADIS16209 high accuracy, dual-axis digital inclinometer and accelerometer datasheet.
- [2] AMTI HE6X6 force platform datasheet.
- [3] Tekscan FlexiForce sensors user's manual.
- [4] Texas Instruments MSP430 microcontroller family datasheet.
- [5] M. Chandrapal, X. Chen, W. Wang, B. Stanke, and N. L. Pape. Preliminary evaluation of intelligent intention estimation algorithms for an actuated lower-limb exoskeleton. *International Journal of Advanced Robotic Systems*, February 2013.
- [6] A. Chu, H. Kazerooni, and A. Zoss. On the biomimetic design of the berkeley lower extremity exoskeleton (BLEEX). *IEEE Proceedings of the International Conference on Robotics and Automation*, April 2005.
- [7] M. Y. Chuah and S. Kim. Enabling force sensing during ground locomotion: A bio-inspired, multi-axis, composite force sensor using discrete pressure mapping. *IEEE Sensors Journal*, 14(5), May 2014.
- [8] G. Colombo, M. Joerg, R. Schreier, and V. Dietz. Treadmill training of paraplegic patients using a robotic orthosis. *Journal of Rehabilitation Research and Development*, 37(6):693–700, November 2000.
- [9] P. Culmer, P. Brooks, D. Strauss, D. Ross, M. Levesley, R. O'Connor, and B. Bhakta. An instrumented walking aid to assess and retrain gait. *IEEE Transactions on Mechatronics*, 19(1), February 2014.
- [10] R. P. Dickenson, W. C. Hutton, and J. R. R. Stott. The mechanical properties of bone in osteoporosis. *The Bone and Joint Journal*, 63(2):233–238, August 1981.
- [11] A. M. Dollar and H. Herr. Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art. *IEEE Transactions on Robotics*, 24(1):144–158, February 2008.
- [12] D. P. Ferris, K. E. Gordon, G. S. Sawicki, and Peethambaran. An improved powered angle-foot orthosis using proportional myoelectric control. *Gait and Posture*, 23(4):425–428, June 2006.

- [13] C. Fleischer, C. Reinicke, and G. Hommel. Predicting the intended motion with emg signals for an exoskeleton orthosis controller. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2005.
- [14] S. Geisser. *Predictive Inference*. Chapman and Hall, New York, 1993.
- [15] J. Ghan and H. Kazerooni. System identification for the berkeley lower extremity exoskeleton (BLEEX). *IEEE Proceedings of the International Conference on Robotics and Automation*, May 2006.
- [16] M. Hassan, H. Kadone, K. Suzuki, and Y. Sankai. Wearable gait measurement system with an instrumented cane for exoskeleton control. *IEEE Sensors Journal*, 14(1), 2014.
- [17] A. L. Hicks, K. A. Martin, D. S. Ditor, A. E. Latimer, C. Craven, J. Bugaresti, and N. McCartney. Long-term exercise training in persons with spinal cord injury: Effects on strength, arm ergometry performance and psychological well-being. *Spinal Cord*, 41:34–43, 2003.
- [18] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari. Robotic orthosis lokomat: A rehabilitation and research tool. *Neuromodulation*, 6(2):108–115, April 2003.
- [19] H. Kawamoto, S. Lee, S. Kanbe, and Y. Sankai. Power assist method for HAL-3 using emg-based feedback controller. *IEEE International Conference on Systems, Man and Cybernetics*, 2:1648–1653, October 2003.
- [20] H. Kazerooni, J. L. Racine, L. Huang, and R. Steger. On the biomimetic design of the berkeley lower extremity exoskeleton (BLEEX). *IEEE Proceedings of the International Conference on Robotics and Automation*, April 2005.
- [21] T. S. Kuan, J. Y. Tsou, and F. C. Su. Hemiplegic gait of stroke patients: The effect of using a cane. *Physical Medicine and Rehabilitation*, 80(7):777–784, July 1999.
- [22] A. B. Lehmann, D. Johnston, and O. F. W. James. The effects of old age and immobility on protein turnover in human subjects with some observations on the possible role of hormones. *Age and Aging*, 18(3):148–157, July 1988.
- [23] A. Liggings, D. Coiro, G. Lange, T. Johnston, B. Smith, and J. McCarthy. The case for using instrumented crutches during gait analysis. *Proceedings of the IEEE Bioengineering Conference*, 2002.
- [24] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, Inc., New York, 1968.
- [25] U. Lugris, J. Carlin, A. Luaces, and J. Cuadrado. Gait analysis system for spinal cord-injured subjects assisted by active orthosis and crutches. *Journal of Multi-Body Dynamics*, July 2013.

- [26] G. V. Merrett, M. A. Ettabib, C. Peters, G. Hallett, and N. M. White. Augmenting forearm crutches with wireless sensors for lower limb rehabilitation. *Measurement Science and Technology*, 21(12), October 2010.
- [27] P. Moran, G. Harris, K. Acharya, H. Zhu, and J. Wertsch. A biofeedback cane system: Instrumentation and subject application results. *IEEE Transactions on Rehabilitation Engineering*, 3(1), March 1995.
- [28] G. A. Ollinger, J. E. Colgate, M. A. Peshkin, and A. Goswami. Active-impedance control of a lower-limb assistive exoskeleton. *IEEE International Conference on Rehabilitation Robotics*, June 2007.
- [29] K. A. Opila, A. C. Nicol, and J. P. Paul. Upper limb loadings of gait with crutches. *Journal of Biomedical Engineering*, 109(4):285–290, August 1987.
- [30] C. Perez, A. Oates, L. Hughey, and J. Fung. Development of a force-sensing cane instrumented within a treadmill-based virtual reality locomotor system. *Proceedings of the IEEE International Conference on Virtual Rehabilitation*, pages 154–159, 2009.
- [31] J. Pratt, K. B. T., C. J. Morse, and S. H. Collins. The RoboKnee: An exoskeleton for enhancing strength and endurance during walking. *IEEE Proceedings of the International Conference on Robotics and Automation*, April 2004.
- [32] U. Saranlı, A. Avci, and M. C. Ozturk. A modular, real-time fieldbus architecture for mobile robotic platforms. *IEEE Transactions on Instrumentation and Measurement*, 60(3):916–927, March 2011.
- [33] E. Sardini, M. Serpelloni, and M. Lancini. Wireless instrumented crutches for force and movement measurements for gait monitoring. *IEEE Transactions on Instrumentation and Measurement*, December 2015.
- [34] K. A. Strausser, S. T. A., A. B. Zoss, H. Kazerooni, and B. C. Bennett. Mobile exoskeleton for spinal cord injury: Development and testing. *ASME Dynamic Systems and Control Conference*, October 2011.
- [35] N. Tsuda, Y. Sakai, K. Fujii, S. Tarao, Y. Nomura, and N. Kato. Simplified measurement and visualization tool for crutch walk training. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2012.
- [36] A. Tsukahara, H. Y., and Y. Sankai. Standing-up motion support for paraplegic patient with robot suit HAL. *IEEE International Conference on Rehabilitation Robotics*, June 2009.
- [37] H. Vallery, E. H. Asseldonk, M. Buss, and H. van der Kooij. Reference trajectory generation for rehabilitation robots: Complementary limb motion estimation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(1):23–30, February 2009.

- [38] C. J. Walsh, D. Paluska, K. Pasch, W. Grand, A. Valiente, and H. Herr. Development of a lightweight, underactuated exoskeleton for load-carrying augmentation. *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2006.
- [39] C. J. Yang, J. F. Zhang, Y. Chen, Y. M. Dong, and Y. Zhang. A review of exoskeleton-type systems and their key technologies. *Journal of Mechanical Engineering Science*, 222(8):1599–1612, August 2008.
- [40] A. B. Zoss, H. Kazerooni, and A. Chu. Biomechanical design of the berkeley lower extremity exoskeleton (BLEEX). *IEEE/ASME Transactions on Mechatronics*, 11(2):128–138, April 2006.