ORTHOROBY ROBOTIC SYSTEM FOR ORTHOPEDIC SURGERY

by

Yasin GÜVEN

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronics Engineering

Yeditepe University
2012

ORTHOROBY ROBOTIC SYSTEM FOR ORHTOPEDIC SURGERY

APPROVED BY:

Assoc.Prof. Dr.Duygun Erol Barkana ....
(Supervisor)

Assoc.Prof. Dr. Cem Ünsalan ....

Asst.Prof. Dr.Sedat Şişbot ....

DATE OF APPROVAL: 30/07.2012

ACKNOWLEDGEMENTS

This study was supported by the Support Programme for Scientific and Technological Research Projects (TÜBİTAK-1001) under Grant 108E092.

ABSTRACT

ORTHOROBY ROBOTIC SYSTEM FOR ORTHOPEDIC SURGERY

Bone cutting is extensively used in straightening bone deformities, extending bone length, artificial joint implants applications, and removal of bone regions inflicted on by tumors, infections, and other similar agents. It is important to accomplish the bone cutting operation with minimal error and with least damage to the surrounding tissues. Recent research in orthopedic surgery indicates that robotic systems can improve the precision and accuracy of the surgery by minimizing the error and damage to the surrounding tissues. In this thesis, an orthopedic robotic system called OrthoRoby and an intelligent control architecture that will be used to control OrthoRoby is developed. A user interface is integrated into the OrthoRoby system. Real-time experiments have been performed on a bone model to evaluate the efficiency of the OrthoRoby.

ÖZET

ORTOPEDİK CERRAHİ AMAÇLI ORTHOROBY ROBOTİK SİSTEMİ

Ortopedik cerrahide, kemik kesimleri kemik eğriliklerinin düzeltilmesinde, boy uzatılmasında, ekleme yapay protez uygulamalarında ve kemiğin bölgesel olarak çıkartılması gereken durumlarda (tümör, enfeksiyon gibi) sıklıkla kullanılmaktadır. Kemik kesim operasyonlarının hatasız ve çevre dokulara en az zarar verilerek yapılması tedavi süresinin kısalması ve komplikasyonların önlenmesi açısından önem teşkil etmektedir. Ortopedik cerrahide yapılan son araştırmalar, robot sistemlerinin, hatayı ve çevre dokulara zararı minimize ederek ameliyatların hassasiyetini ve doğruluğunu arttırabileceğini belirtmektedir. Bu çalışmada, kemik kesme ameliyatlarında kullanılmak üzere ortopedik cerrahi robot sistemi OrthoRoby ve OrthoRoby'i kontrol etmek için akıllı kontrol mimarisi geliştirildi. Bir kullanıcı arayüzü OrthoRoby sistemine entegre edildi. Kadavra kemiği üzerinde gerçek zamanlı deneyler OrthoRoby'nin kullanılabilirliğini test etmek için yapıldı.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

V	Voltage
A	Ampere
C_1	Camera 1
C_2	Camera 2
α_n	Angle between markers and horizontal axis
h_n	Distance between markers on vertical axis
a	Horizontal distance between markers seen by camera 1
b	Horizontal distance between markers seen by camera 2
L_n	Distance between markers
l_d	Desired leg length
$\dot{l}_d$	Desired leg length velocity
$\ddot{l}_d$	Desired leg length acceleration
l_a	Actual leg length
$\dot{l}_a$	Actual leg length velocity
K_d	Low level controller derivative term gain
K_p	Low level controller proportional term gain
K_i	Low level controller integral term gain
e_q	Error
T_{dist}	Disturbance torque
T_{ctrl}	Control torque
PID	Proportional integral derivative
DoF	Degree of freedom
PWM	Pulse width modulation
kHz	Kilohertz
kW	Kilowatt
RPM	Revolutions per minute
PLC	Programmable logic controller
HD	High definition
3D	Three dimensional

CT	Computed tomography
UI	User interface

1. INTRODUCTION

Orthopedic surgery is one of the most common operations in hospitals. Most bone related orthopedic surgeries are performed to straighten bone deformities, to extend bone length, and to remove bone regions inflicted on by tumors and infections. Current manual surgical techniques may result in inaccurate placing and balancing of hip replacements, knee components, or soft-tissues. In recent years, computer-assisted robotic systems have been developed for orthopedic surgeries, which improve the precision and accuracy of the surgery and in turn lead to better long-term outcomes.

Various orthopedic surgery robotic systems have been developed to perform orthopedic surgeries in an accurate and safe manner. Some of these robotic systems use serial manipulators and some of them use parallel manipulators. Robodoc [1], Caspar, Acrobot [2], Arthrobot [3] and [4] are well-known orthopedic surgical robots that belong to the serial manipulators with large workspace which are somewhat heavy and suffer from low stiffness and accuracy, and possess low nominal load/weight ratio.

Parallel robots are also used for orthopedic surgery robots, which have specific advantages over serial robots such as better stiffness and precise positioning capability. Parallel manipulators are closed kinematic structures that hold requisite rigidity to yield a high payload to self-weight ratio. MARS is one of the well-known patient-mounted parallel robot [5, 6]. Similar to the MARS miniature orthopedic robot, MBARS [7] robot employs a parallel platform architecture. MBARS has been used for machining the femur to allow a patella implant to be positioned [7, 8]. Additionally, serial devices are significantly more prone to failure than parallel robots because in a serial device, one failure can cause the robot to move dramatically, whereas in a parallel structure, one failure will have little effect on the overall position of the robot. This is important because surgeon requires the device to maintain its last position in the event of a catastrophic failure. In this thesis an orthopedical surgical robotic system OrthoRoby, which consists of a parallel robot and a cutting tool, has been developed.

A control architecture is developed for orthopedic surgical robotic system OrthoRoby to complete bone cutting operation in desired and safe manner [9-11]. The control architecture consists of OrthoRoby robotic system, user interface, cameras, a high-level controller and a low-level controller. In this thesis, a hybrid system modelling technique is used to design the high-level controller, and a computed-torque controller is used for the low-level controller of the OrthoRoby to track the desired cutting trajectory [12]. A user interface is developed and integrated into the robotic system to operate bone cutting surgery.

Section 2 introduces previously proposed orthopedic surgical robotic systems. In Section 3, details of OrthoRoby and cartesian system, OrthoRoby's low-level and high-level controller, user interface and the camera system are presented. The experimental results of OrthoRoby system are presented in Section 4. Conclusion of the thesis and plans for future work are given in Section 5.

2. BACKGROUND

Robodoc [1] seen in Figure 2.1, Caspar, Acrobot [2] seen in Figure 2.2, Arthrobot [3] seen in Figure 2.3 and [4] are well-known orthopedic surgical robots that belong to the serial manipulators with large workspace which are somewhat heavy and suffer from low stiffness and accuracy, and possess low nominal load/weight ratio.

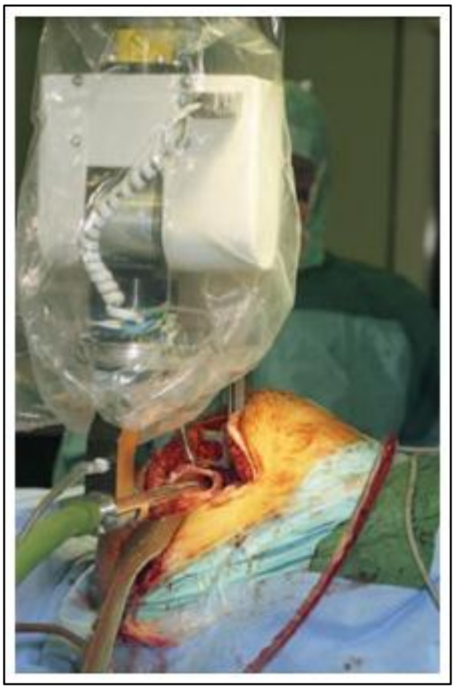


Figure 2.1. Robodoc

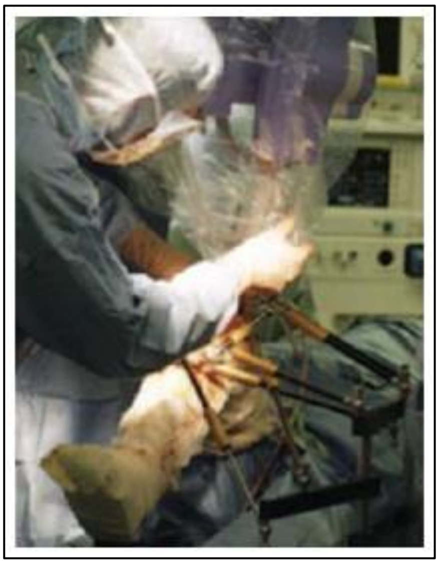


Figure 2.2. Acrobot

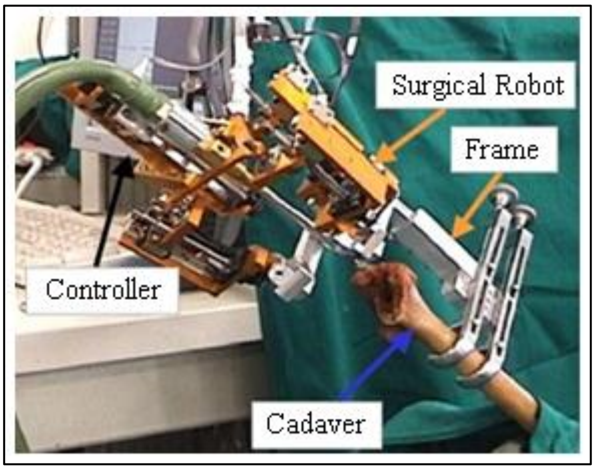


Figure 2.3. Arthrobot

Parallel robots are also used for orthopedic surgery robots, which have specific advantages over serial robots such as better stiffness and precise positioning capability. Parallel manipulators are closed kinematic structures that hold requisite rigidity to yield a high payload to self-weight ratio. MARS is one of the well-known patient-mounted parallel robot [5, 6]. Similar to the MARS miniature orthopedic robot, MBARS [7] seen in Figure 2.4 robot employs a parallel platform architecture. MBARS has been used for machining the femur to allow a patella implant to be positioned [7, 8].

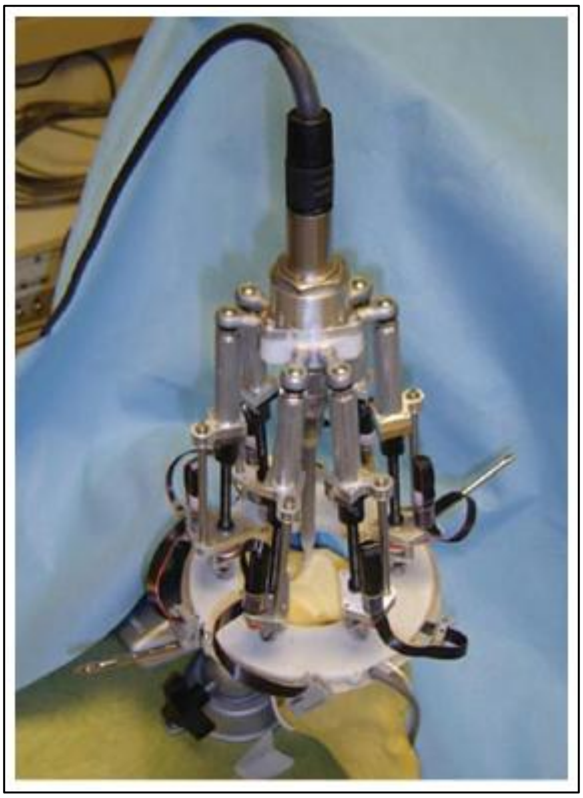


Figure 2.4. MBARS parallel robot

Compact robot system for image-guided orthopedic surgery (CRIGOS) is another parallel robot developed for planning of surgical interventions and for supervision of the robotic device [13]. Additionally, Orthdoc [14] and Hexapod [15] use parallel manipulators for orthopedic surgery. Hybrid bone-attached robot (HyBAR) has also been developed with a parallel and serial hybrid kinematic configuration for joint arthroplasty [12]. A parallel robot has been developed with automatic bone drilling carriage [16]. Parallel manipulators are preferred for orthopedic surgeries because they provide advantages in medical robotics such as small accumulated positioning errors and high stiffness [17]. On the other hand, Praxiteles is another patient-mounted surgical robot which comprised of 2 motorized degrees of freedom (DoF) whose axes of rotation are arranged in parallel, and are precisely aligned to the implant cutting planes with a 2 DoF adjustment mechanism [18].

3. METHODOLOGY

First, general control architecture of OrthoRoby is presented in this section. Then, control hardware of OrthoRoby is given. Finally, design of the low level and high level controllers of OrthoRoby are presented.

3.1. CONTROL ARCHITECTURE

A control architecture is developed for orthopedic surgical robotic system OrthoRoby to complete bone cutting operation in desired and safe manner [9-11]. The control architecture consists of OrthoRoby robotic system, user interface, cameras, a high-level controller and a low-level controller as shown in Figure 3.1.

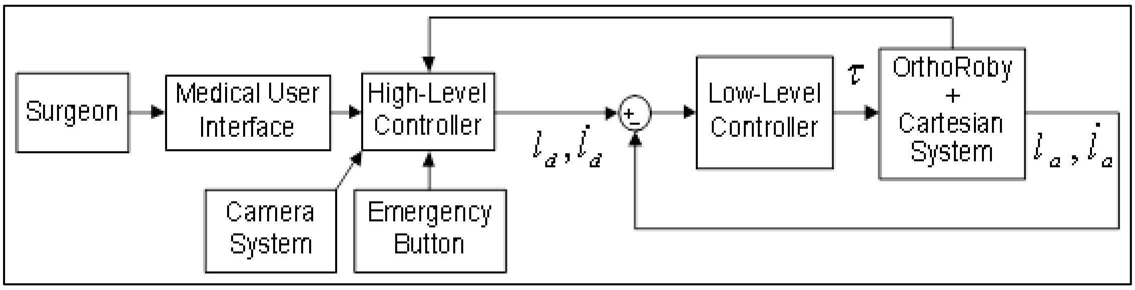


Figure 3.1. Control architecture of OrthoRoby

3.2. ORTHOROBY SYSTEM

OrthoRoby system consists of parallel robot, cutting tool, cartesian system and a camera system as shown in Figure 3.2.

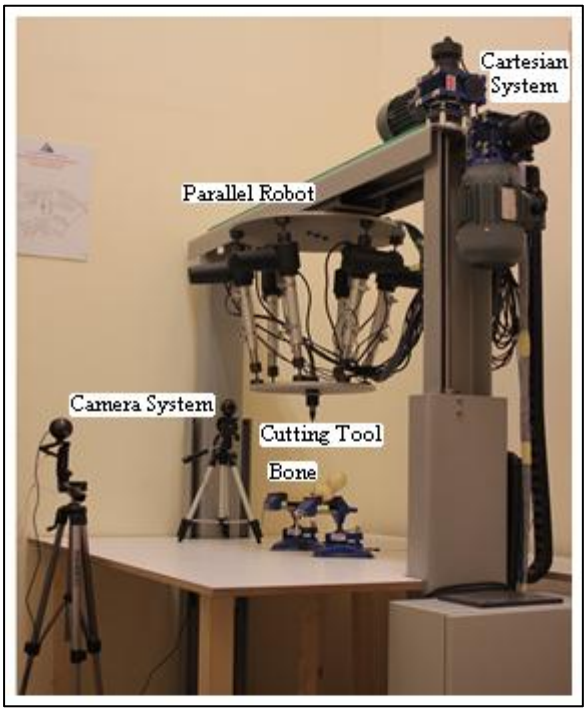


Figure 3.2. OrthoRoby robotic system with cartesian and camera system

OrthoRoby is developed considering the well-known parallel robot Stewart platform. Stewart platform has a moving platform connected to the base platform by linear actuators called legs. Each leg is connected to the moving platform and the base platform by spherical joints, universal joints and revolute joints. In our previous work a 6-6 spherical-prismatic-spherical (SPS) Stewart platform is selected for OrthoRoby as like as MARS and CRIGOS robots [19]. OrthoRoby parallel robot consists of two circular plates connected by six linear actuators as seen in Figure 3.3a. The plates are connected by six linear actuators CARE33H (SKF) which have a stroke length of 150mm and each can take a load up to 60kg. The maximum force generated by each actuator can be as much as 600N. The actuators have encoders attached to them to determine the position of the robot. The actuators are connected to the base and moving platform by spherical joints. The spherical joint connectors are manufactured so that the actuators can be connected to the base and moving platforms properly. The spherical joints have pivot angle 400. Cutting tool, which is placed in the middle of the moving platform of the parallel robot, is selected as Dremel 400 Digital (Dremel Inc). Cutting tool is attached on the moving platform in such a way that the height of the tool can be adjusted. The mechanical system of OrthoRoby is shown in Figure 3.3b.

The OrthoRoby is controlled via a 3.2GHz Pentium 4 PC with 2GB of RAM. The hardware is controlled through the MatLab Real Time Workshop Toolbox from Mathworks, and WinCon from Quanser Consulting. All data inputs and outputs are handled by the Quanser Q8 board. The leg lengths of the robot are acquired using encoders of CARE33H with a sampling time of 0.001 seconds from a Quanser Q8 card. The torque output to the OrthoRoby is sent with the same card with the same sampling time. A control card is developed to drive DC motors (actuators) of OrthoRoby as shown in Figure 3.3c. The developed board includes six sets of PWM generators, H-bridge amplifiers and logic circuits. Each set includes three inputs for speed reference, direction and enable signals. Direction signal is used by logic circuit to determine direction of CARE33H's movement. Enable signal is used to activate H-bridge amplifier's outputs. Direction and enable signals are digital signals received from digital outputs of Quanser Q8 board. Speed reference is an analog signal received from analog outputs of Quanser Q8 board. Using speed reference signal PWM generator controls CARE33H's speed. Frequency is set to 40 kHz for smooth movement PWM. H-bridge that is connected to PWM generator's output can drive CARE33H at high currents up to 7A. The used H-bridge can drive two motors up to 3.5A, so two outputs are used in parallel connection to achieve 7A max current. Logic gates are used to control CARE33H's movement direction. Using Quanser Q8 board, position feedback of the actuators is received from internal encoders of actuators and transmitted to the control algorithm running on Matlab. A power supply is used to provide 5V and 12V to the control card. OrthoRoby within cutting operation is shown in Figure 3.3d.

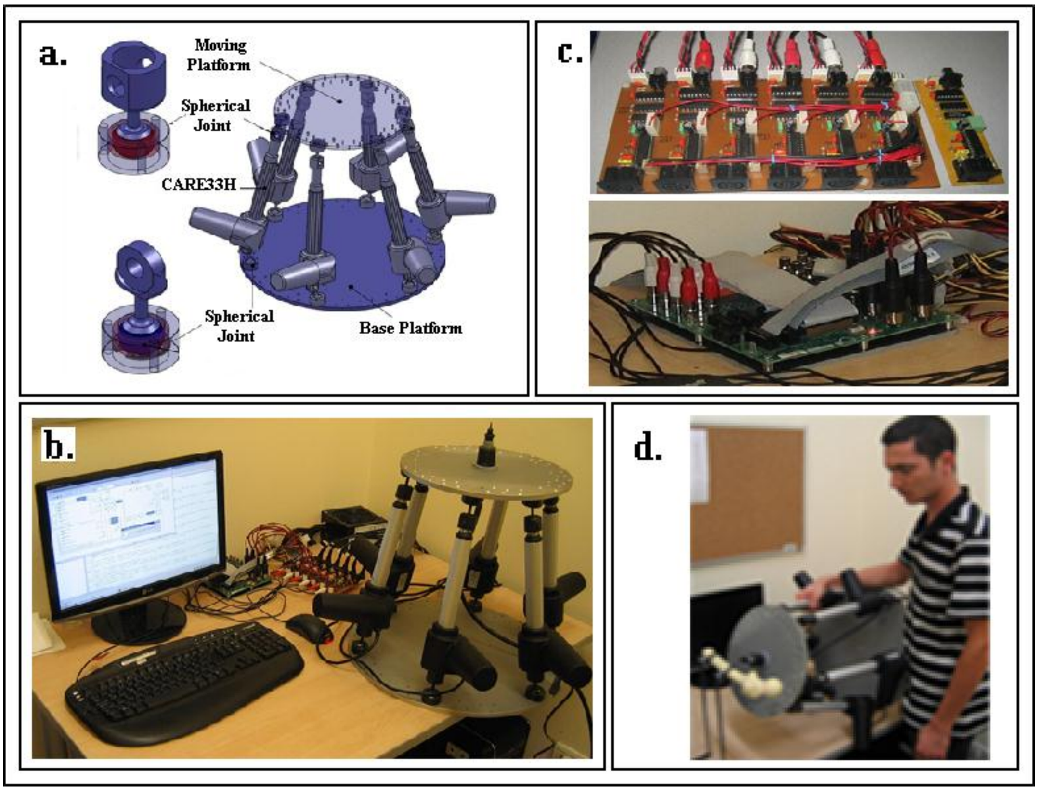


Figure 3.3. OrthoRoby, a. Parallel robot design details, b. OrthoRoby robotic system, c. Electrical connections, d. OrthoRoby within bone cutting operation

A Cartesian system is designed and manufactured for OrthoRoby’s initial positioning over bone before cutting procedure. Two Gamak three phase standard asynchronous motors with 0.25 kW rated power, and 220 or 380 rated volts were used in Cartesian system. A reducer with 1:12 rate is used to reduce 1380 revolutions per minute (RPM) of motors. Two MOELLER frequency inverters are used to drive asynchronous motors. Autonics rotary encoders with supply voltage range of 12-24V are used to read positions of motors and to determine if the mechanism reached the desired position. Also four Autonics inductive sensors are used to limit the movement of the mechanism at each edges. Three LTM linear screw drivers are used to transfer motor motion to the mechanism. ABBA rail and cart systems have been used to decrease friction during movement of motors. Six rails and eight carts were used on the system. Touch panel has been added to the system for surgeon to use the Cartesian system easily. Delta PLC has been used for software control of data on touch panel. A power supply with 24V, 2A rated output has been used to feed Cartesian system. Additionally, all inputs and outputs are protected with fuses to use the Cartesian system in orthopedic surgery safely.

Two Logitech C600 HD webcam with fixed focus cameras, which are labeled as C_1 and C_2, are integrated into the control architecture of OrthoRoby to detect if OrthoRoby is close enough to the bone and to measure the depth of cutting during the operation (Figure 3.4). Maximum resolution of the cameras is 2 megapixels, however only 400 pixels by 400 pixels have been cropped and used from the center of images to eliminate optical distortion caused by lens of cameras, and to obtain a more planar image plane. Markers are placed on the bone and on the OrthoRoby's cutting tool. The images taken from the cameras are processed and sent as a command to high-level controller to activate the cutting tool of OrthoRoby. OrthoRoby's cutting tool is activated when OrthoRoby is close enough to the bone. Additionally, the images from the cameras are used to stop the motion of OrthoRoby when the cutting tool reaches to the desired depth in the bone.

Image planes of camera 1 and camera 2 are shown in Figure 3.4. on which L_1, α_1 and L_2, α_2 are drawn respectively. L_1, α_1 and L_2, α_2 are calculated in the image plane of cameras. L_1 and L_2 represents the distance between the markers on image planes seen from the positions of camera 1 (C_1) and camera 2 (C_2), respectively. L is the 3D distance between two markers in pixel unit. α_1 and α_2 represent the slope between the two markers seen from the positions of C_1 and C_2. h_1 and h_2 represent the vertical distance between the markers seen from the positions of C_1 and C_2. h is the vertical distance between markers that is equal to h_1 and h_2 .

Initially, pixel values obtained from images are converted into millimeter values of the bone environment. First a ruler is placed in front of C_1 at the distance of markers, and an image is captured. Then the number of pixels in one centimeter is counted in vertical and horizontal axes. The same procedure can be used for camera 2. The values that are obtained are then used to convert pixel values of images into millimeter values. Positions of the markers located on cutting tool and bone are determined by processing real-time images captured from two cameras simultaneously. Values of $L_{1,2}$, $\alpha_{1,2}$ and $h_{1,2}$ and a, b are calculated using marker positions. Then a, b and h values are used to calculate L. L is calculated using Equation 3.1. Millimeter equivalent movement of the cutting tool inside the bone is calculated using the pixel per centimeter values.

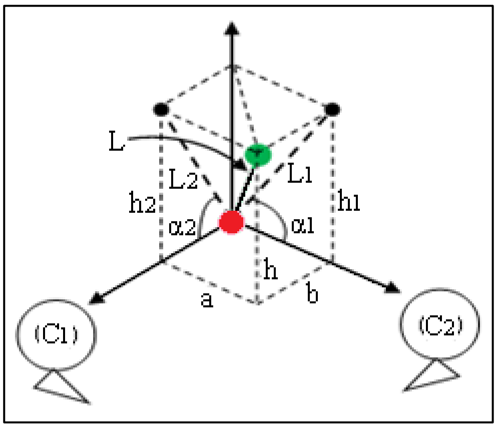


Figure 3.4: Camera interface

$$L = \sqrt{a^2 + b^2 + h^2} \tag{3.1}$$

A robot-support system is developed for surgeon to remotely control the cutting operation (Figure 3.5). Surgeon can immediately stop and continue cutting operation or completely stop the system during an emergency situation using this robot-support system. Additionally, a potentiometer is placed on the support system to be able to change the driller’s speed during the cutting operation.

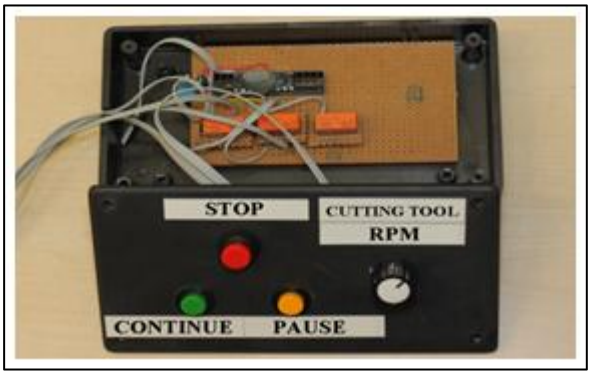


Figure 3.5. Robot-support system: stop, continue and pause with cutting tool rpm setting

3.3. CONROLLERS OF ORTHOROBY

The control of OrthoRoby has a low-level device controller and a high-level decision-making controller. These two controllers are responsible to perform the cutting operation in a desired and safe manner during the surgery. A high-level controller is used to allocate the cutting task responsibility to the low- level controller based on the task requirements and specific events that may arise during the bone cutting task performance. Let us first present low-level controller and then the high-level controller of the control architecture.

Intelligent control mechanism has three main parts that are plant, interface and high-level controller as shown in Figure 3.6. A high-level controller may be used to allocate task responsibility based on the task requirements and specific events that may arise during the task performance. The high-level controller and the low-level assistive controllers, in general, may not communicate directly because each may operate in different domains. While the low-level controllers may operate in a continuous way, the high-level controller may need to make intermittent decisions in a discrete manner. Therefore, an interface is required that can convert continuous-time signals to sequences of discrete symbols and vice versa.

Hybrid system theory provides mathematical tools that can accommodate both continuous and discrete systems in a unified manner. As a result, in this work, we use hybrid system modelling technique to design the presented intelligent control architecture. A hybrid system model has three parts, a “plant,” a “controller” (supervisor), and an Interface [21]. In order to avoid confusion about terminology, we call the “controller” in a hybrid system model a high-level controller. The continuous part, identified as the “plant” represents both parallel robot and cutting tool devices and their low-level controllers. The interface consists of a generator and an actuator. The interface accepts symbolic inputs via the actuator (control symbols) and produces symbolic outputs (plant symbols) via the generator. Note that the term actuator here does not mean the actuator of the OrthoRoby.

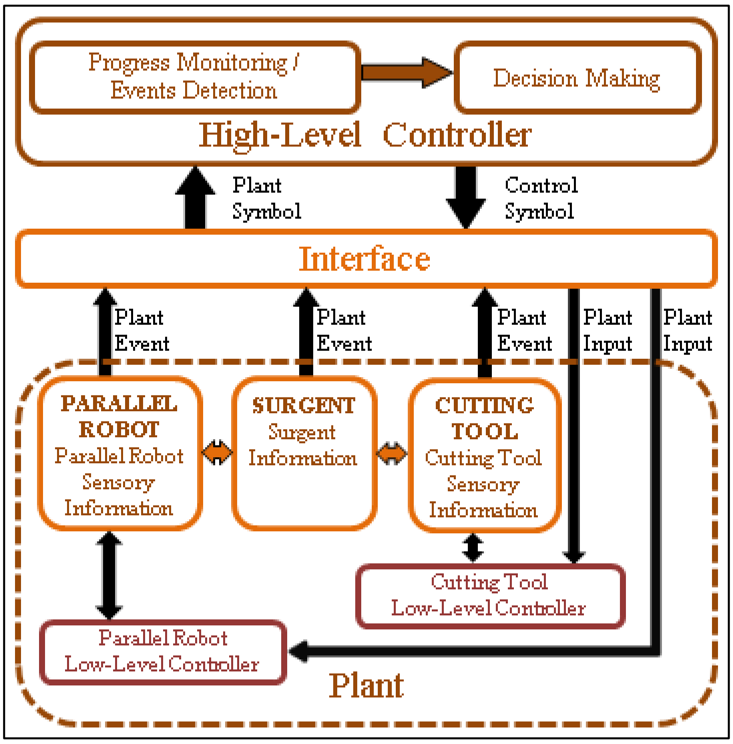

Progress Monitoring / Events Detection
Decision Making
High-Level Controller
Plant Symbol
Control Symbol
Interface
Plant Event
Plant Event
Plant Event
Plant Input
Plant Input
PARALLEL ROBOT Parallel Robot Sensory Information
SURGENT Surgent Information
CUTTING TOOL Cutting Tool Sensory Information
Cutting Tool Low-Level Controller
Parallel Robot Low-Level Controller
Plant

Figure 3.6. Intelligent control mechanism

3.3.1. Low Level Controller

Computed-torque controller is used as the low-level controller of the OrthoRoby to track the desired cutting trajectory as seen in Figure 3.7. Computed torque control is a model-based method, which uses the robot dynamics in the feedback loop for linearization and decoupling [20].

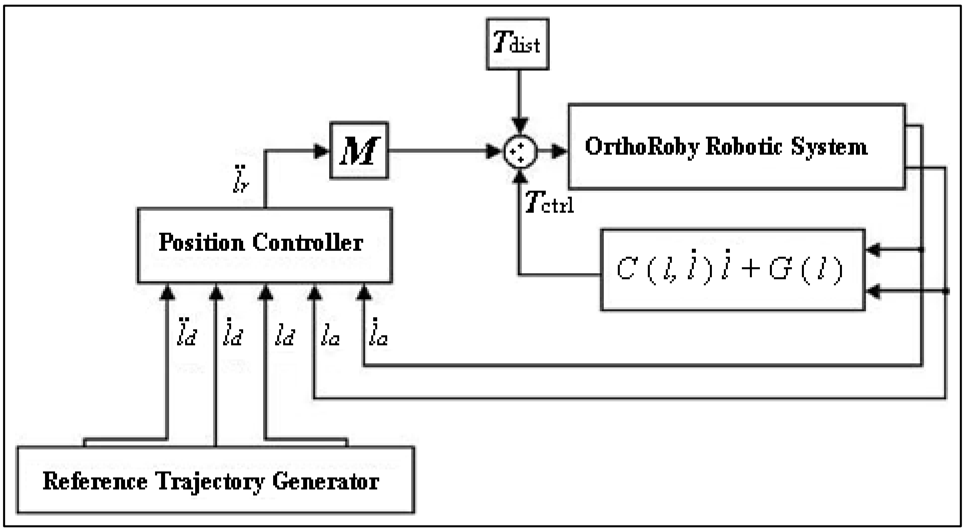


Figure 3.7: Computed torque control for OrthoRoby

Consider the control input which consists of an inner nonlinear compensation loop and an outer loop with an exogenous control signal $\ddot{l}_r$. Substituting this control law into the dynamical model of the robot manipulator that is shown in Equation 3.2, it follows as in Equation 3.3.

$$M(l)\ddot{l}_r + C(l,\dot{l})\dot{l}_r + G(l) + \tau_{dist} = \tau_{ctrl} \tag{3.2}$$

$$\ddot{l} = \ddot{l}_r \tag{3.3}$$

It is important to note that this control input converts a complicated nonlinear controller design problem into a simple design problem for a linear system consisting of decoupled subsystems. One approach to the outer-loop control is propositional–integral-derivative (PID) feedback, as in Equation 3.4.

$$\ddot{l}_r = \ddot{l}_d + K_v(\dot{l}_d - \dot{l}_a) + K_p(l_d - l_a) + K_i\left(\int l_d - \int l_a\right) \tag{3.4}$$

Where e_q equals subtraction of l_a from l_d and in which case the overall control input becomes as shown in Equation 3.5 and the resulting linear error dynamics are presented in the following equation where the convergence of the tracking error to zero is guaranteed.

K_v , K_p and K_i are the derivative, proportional and integral gains, respectively as shown in Equation 3.6.

$$M(l)\left(\ddot{l}_d + K_v(\dot{l}_d - \dot{l}_a) + K_p(l_d - l_a) + K_i\left(\int l_d - \int l_a\right)\right) + C(l,\dot{l})\dot{l} + G(l) + T_{dist} = T_{ctrl} \quad (3.5)$$

$$\ddot{e}_q + K_v\dot{e}_q + K_p e_q + K_i \int e_q = 0 \quad (3.6)$$

The low level computed torque controller is implemented on Simulink from Mathworks. The solver of the Simulink model is configured to run in continuous time, thus low level controller operates in continuous time.

3.3.2. High Level Controller

A high-level controller is needed to make intermittent decisions in a discrete manner. In this thesis, a hybrid system modelling technique is used to design the high-level controller. A set of hypersurfaces that separate different discrete states are defined for the high-level controller. The hypersurfaces are not unique and are decided considering the capabilities of the OrthoRoby system as shown in Table 3.1. Note that the hypersurfaces could be extended or modified for other bone cutting tasks based on the task requirements and the capabilities of the OrthoRoby.

Table 3.1. Hypersurfaces

$h_1 = (sb == 1)$	start button (sb) is a binary value, which will be 1 when it is pressed and 0 when it is released.
$h_2 = \lvert x - x_t \rvert < \lvert \in \rvert$	x and x_t are the parallel robot position and the bone's position, respectively, ∈ is a value used to determine if the parallel robot is close enough to the bone's position.
$h_3 = \lvert x_{ct} \rvert \geq \lvert x_{db} - \in_{ct} \rvert^{cto} == 1$	cto (cutting tool on) is a binary value, which will be 1 when it is pressed and 0 when it is released. x_{ct} and x_{db} are the cutting tool depth and the depth in the bone, respectively. $\in_{ct}$ is a value used to determine if the cutting tool is close enough to the desired depth in the bone.
$h_4 = \lvert x - x_i \rvert < \lvert \in_1 \rvert$	$\lvert x \rvert$ and x_i are the parallel robot position and the initial position of the task, respectively, $\in_1$ is a value used to determine if the robot is close enough to the initial position.
$h_5 = \begin{Bmatrix} l_l < l < l_u \\ \tau_{ctrl} \geq \tau_{rth} \end{Bmatrix}$	l_l and l_u represent the set of lower and upper limits of the parallel robots legs, respectively and l is the set of actual leg lengths. τ_{ctrl} and τ_{rth} are the torque applied to the actuators of the parallel robot device and the threshold value, respectively.
$h_6 = (eb == 1)$ $h_7 = (pb == 1)$ $h_8 = (pb == 0)\hat{}(eb == 0)$	Emergency button (eb) and pause button (pb) are binary values, which will be 1 when it is pressed and 0 when it is released.

Each region in the state space of the plant, bounded by the hypersurfaces, is associated with a state of the plant. A plant event occurs when a hypersurface is crossed. A plant event generates a plant symbol to be used by the high-level controller. The high-level controller is responsible for coordinating the activation of parallel robot and the cutting tool devices based on both task requirements and the safety requirements of the task. Each event is converted to a plant symbol. The next discrete state is activated based on the current discrete state and the associated plant symbol as shown in Table 3.2.

In order to notify the low-level controllers the next course of action in the new discrete state, the high-level controller generates a set of symbols, called control symbols. In this application, the purpose of the high-level controller is to activate/deactivate the parallel robotic device and the cutting tool device of OrthoRoby system in a coordinated manner so that these devices are activated or deactivated in the desired order so that the bone cutting operation does not enter critical regions of the state space in order to ensure safety. When new control actions are required for a bone cutting operation, new control states can easily be included in the set of the states. The transition function uses the current control state and the plant symbol to determine the next control action that is required to update the bone cutting operation. The high-level controller generates a control symbol which is unique for each state. The low-level controller cannot interpret the control symbols directly. Thus the interface converts the control symbols into continuous outputs, which are called plant inputs. The plant inputs are then sent to the low-level controllers to modify the bone cutting operation. Table 3.3 and Table 3.4 present control states and control symbols, respectively. To our knowledge, such an intelligent control mechanism has not been explored before for orthopedic surgical robotic systems. The high-level controller is also modeled using Stateflow that is a tool in Matlab from Mathworks as shown in Figure 3.8.

Table 3.2. Plant symbols

$\tilde{x}_1$	The parallel robot approaches towards the bone, which is generated when h_1 is crossed.
$\tilde{x}_2$	The parallel robot reaches the bone, which is generated when h_2 is crossed.
$\tilde{x}_3$	The cutting tool reaches the desired cutting depth, which is generated when h_3 is crossed.
$\tilde{x}_4$	Parallel robot goes back to starting position, which is generated when h_4 is crossed.
$\tilde{x}_5$	Safety related issues happened such as the parallel robot leg lengths are out of limits, or the parallel robot applied fence is above its threshold (when h_5 is crossed), or emergency button is pressed (when h_6 is crossed), or surgeon pressed pause button (when h_7 is crossed)
$\tilde{x}_6$	The surgeon releases the pause button, which is generated when h_8 is crossed
$\tilde{x}_{61}$ $\tilde{x}_{62}$ $\tilde{x}_{63}$	If the surgeon presses pause button when the parallel robot is approaching towards the bone, then plant symbol $\tilde{\mathrm{x}}_{61}$ is generated. Similarly if the surgent presses pause button when the bone cutting tool is on, then the plant symbol $\tilde{\mathrm{x}}_{62}$ is generated. If the surgent presses pause button when robot is returning back to original position, then then the plant symbol $\tilde{\mathrm{x}}_{63}$ is generated.

Table 3.3. Control states

$\tilde{s}_{1f}$	The parallel robot device alone is active to move towards the bone.
$\tilde{s}_{1b}$	The parallel robot device alone is active to move back to the starting position.
$\tilde{s}_{2}$	Both the parallel robot device and the cutting tool device are active.
$\tilde{s}_{3}$	Both the parallel robot device and the cutting tool device are idle.
$\tilde{s}_{4m}$	Memory state after surgeon says "stop" while $\tilde{s}_{4m}$ (where m = 1, 2) is active.
$\tilde{s}_{5m}$	Continue state when the surgeon wants to continue with the task while $\tilde{s}_{4m}$ (where m = 1, 2).

Table 3.4. Control symbols

$\tilde{r}_{1f}$	Drive parallel robot device to perform primitive motion to move towards the bone.
$\tilde{r}_{1b}$	Drive, parallel robot device to perform primitive motion to move back to the starting position
$\tilde{r}_{2}$	Drive cutting tool device to cut the bone.
$\tilde{r}_{3}$	Make the parallel robot and cutting tool devices idle.

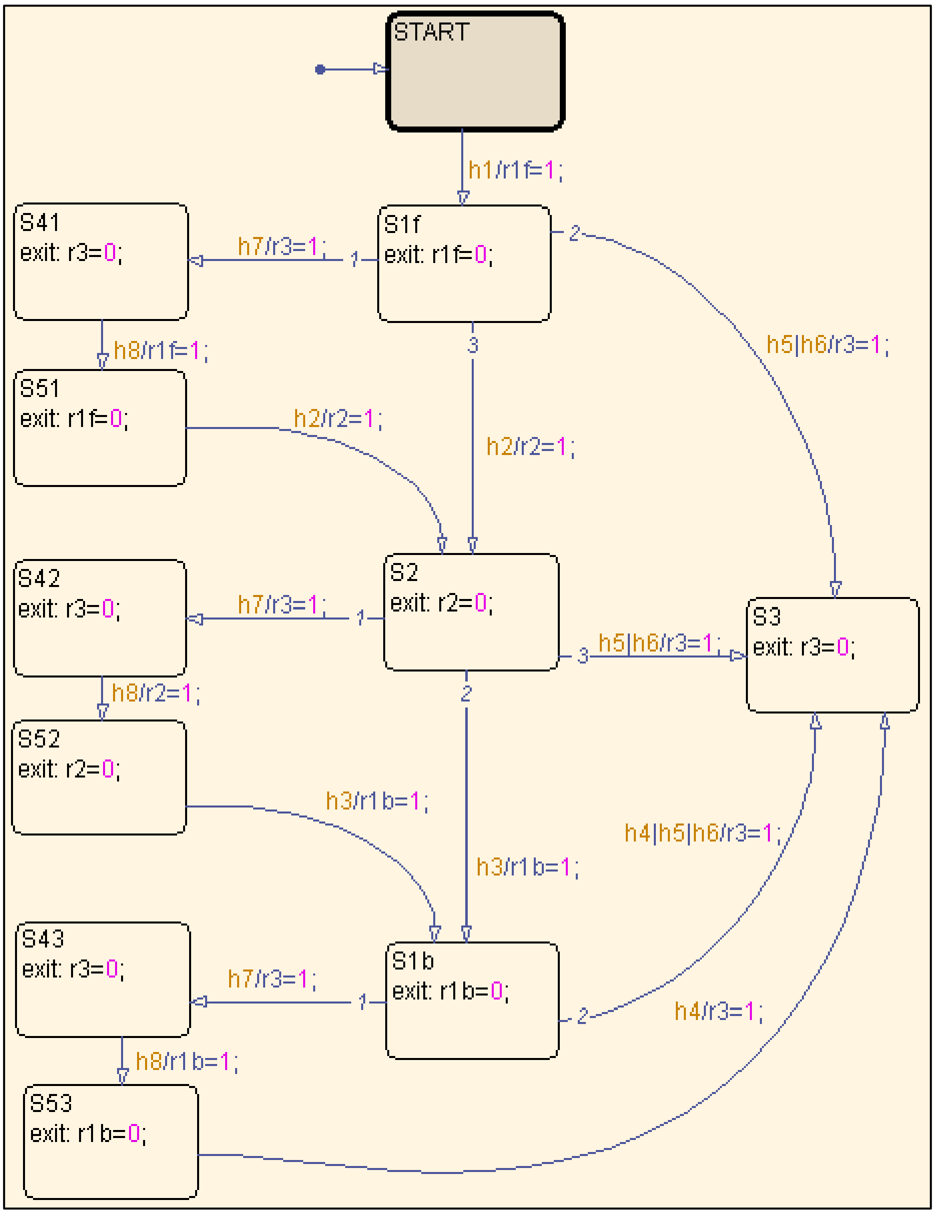
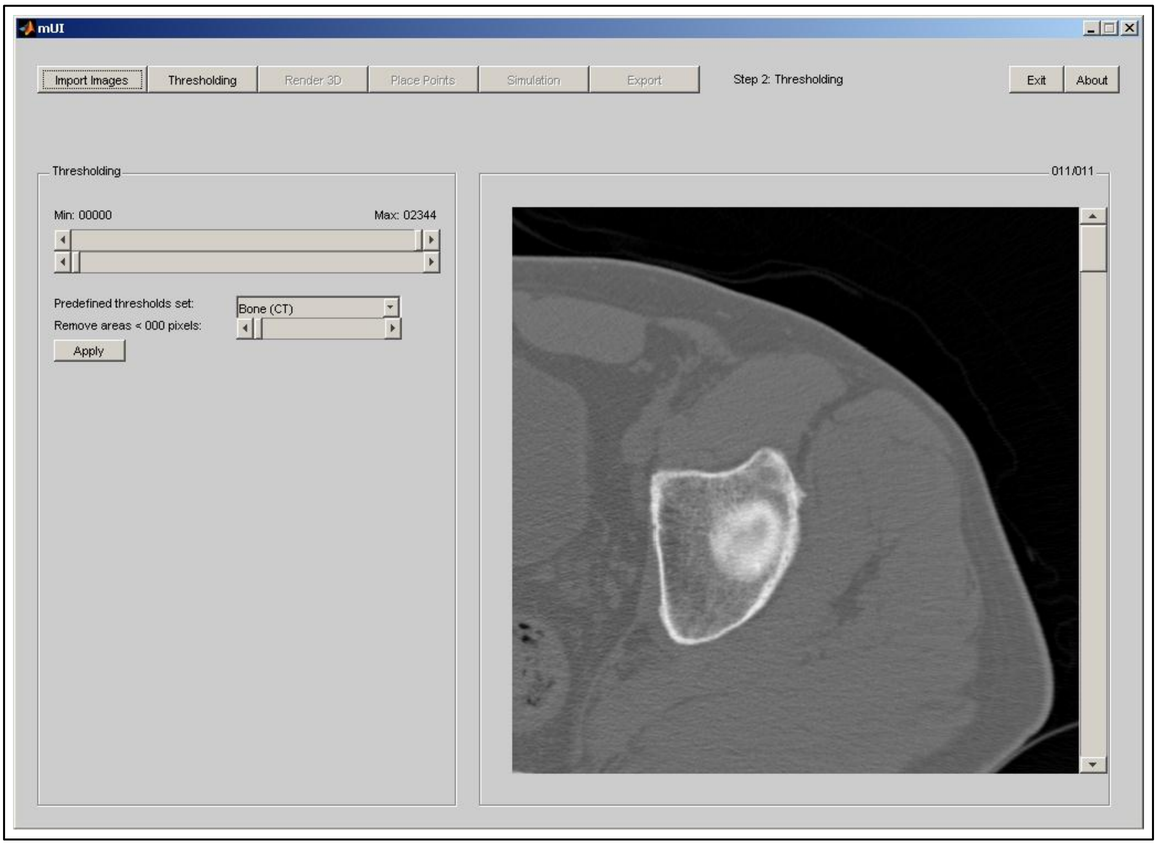


Figure 3.8. High-level controller’s stateflow model

3.4. USER INTERFACE

A user interface has been developed and integrated into the robotic system to allow surgeon easily control bone cutting operation. The necessity of obtaining accurate results for posterior validation with experimental values implied an adequate modeling of the bone structure in terms of 3D modeling. The initial step concerning the bone anthropometrical definition is a Computer-Tomography (CT) scan of the femur region of patients in a Philips® Brilliance CT equipment. The geometric models are obtained from 3D reconstruction of CT images of the patients which are taken from Yeditepe University Hospital. The CT images are taken with intervals of 1 mm in the neutral position. Surgeon uses user interface (UI) to define the bone cutting trajectory. UI uses CT images of the patients' bone as inputs and outputs a bone cutting trajectory. The origin and axes of CT images were matched with the origins and axes of camera system and OrthoRoby during the generation of the trajectory.

Surgeon can import DICOM images of the patient's bone using the tool panel and then display the images using display panel (Figure 3.9). Tool panel shows segmentation tools and display panel shows imported slices. When surgeon presses the 'Import Images' button, then a file selection dialog appears that shows DICOM images of the patient's bone. Surgeon can select single or multiple DICOM images. When DICOM images are read, then BT images are stored in the memory. BT image is a 512 pixels by 512 pixels by n array, where n is the number of slices to be read. In this thesis, we import 200 slices.



Step 2: Thresholding

Min: 00000

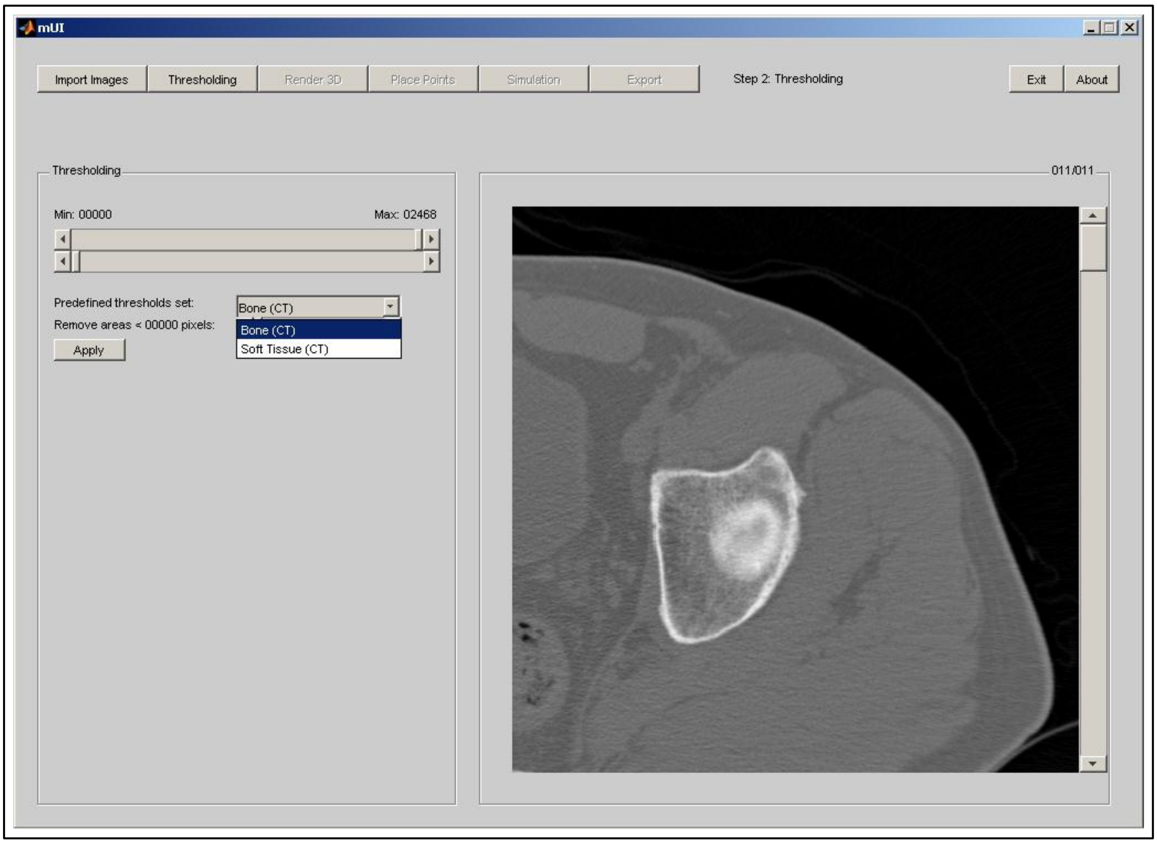
Max: 02344

Predefined thresholds set: Bone (CT)

Remove areas < 000 pixels:

Figure 3.9. Tool panel and display panel

When slices of CT images of the patient are used both soft tissue and bone are imported. Thus, segment bone needs to be separated from the soft tissue then it will be possible to render 3D model of the patient’s bone. A semiautomatic histogram based thresholding method is used to segment medical images [22]. Segmentation process is performed by defining minimum and maximum threshold values manually or by choosing one of the predefined thresholds (Figure 3.10). When segmentation process starts, grayscale images are automatically converted into black and white images by defining binary one to the grayscale pixel values in the range of threshold values and binary zero to the grayscale pixel values that are out of the range of threshold values as can be seen in Figure 3.11. A slider, which sets the minimum area of noise to remove the noise from the binary image, is used to refine the thresholding results.



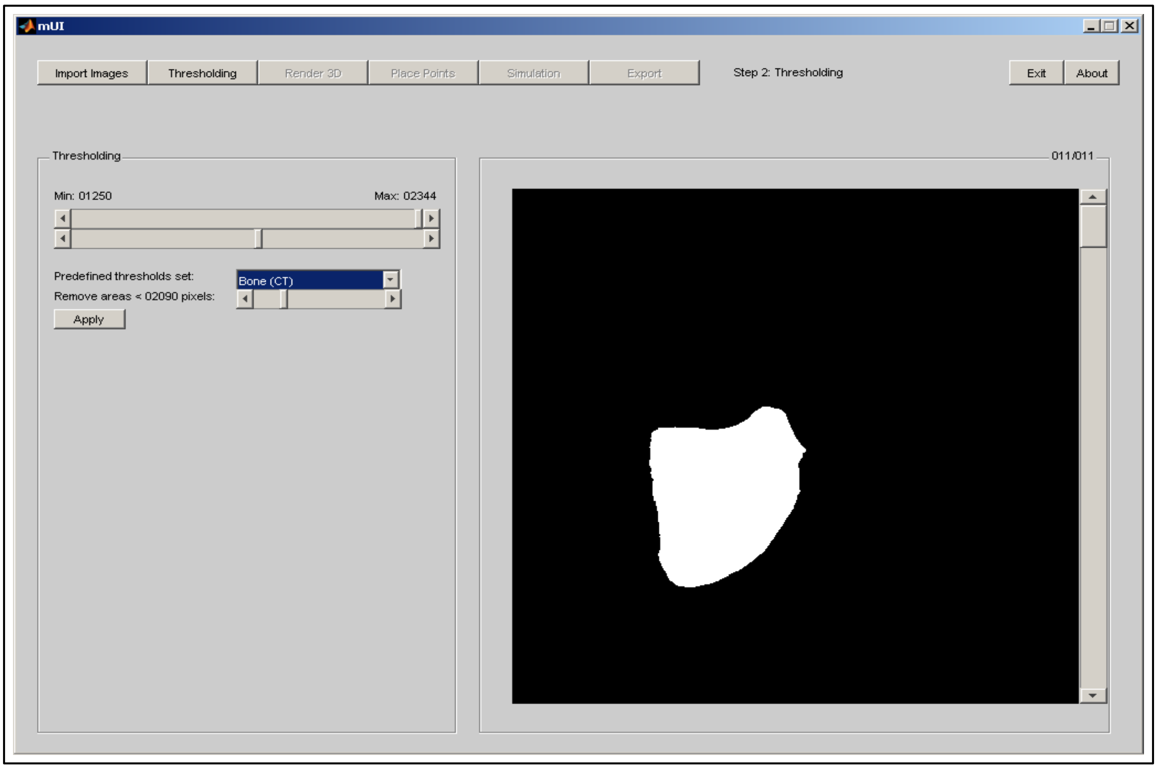
Min: 00000

Max: 02468

Predefined thresholds set: Bone (CT)

Remove areas < 00000 pixels:

Figure 3.10. Predefined thresholds



Step 2: Thresholding

Thresholding

Min: 01250

Max: 02344

Predefined thresholds set: Bone (CT)

Remove areas < 02090 pixels:

Figure 3.11. Thresholding preview

The next step of the UI is rendering that is the process of generating 3D model from segmented CT images. Two types of 3D plot methods are generally used for rendering which are triangular surface plot and triangular mesh plot [23]. In this thesis, triangular mesh plot method is selected because of its fast computation [24, 25]. When triangular mesh plot method is applied to the segmented CT images, white areas in the patient's segmented slices are converted into point fields with equal spaces and then point cloud is generated from these fields. Generated point cloud is used as nodes while plotting mesh triangles. Finally, rendered 3D bone model is generated. An example thresholding result is shown in Figure 3.12. The quality of the generated 3D model mostly depends on the generated point cloud.

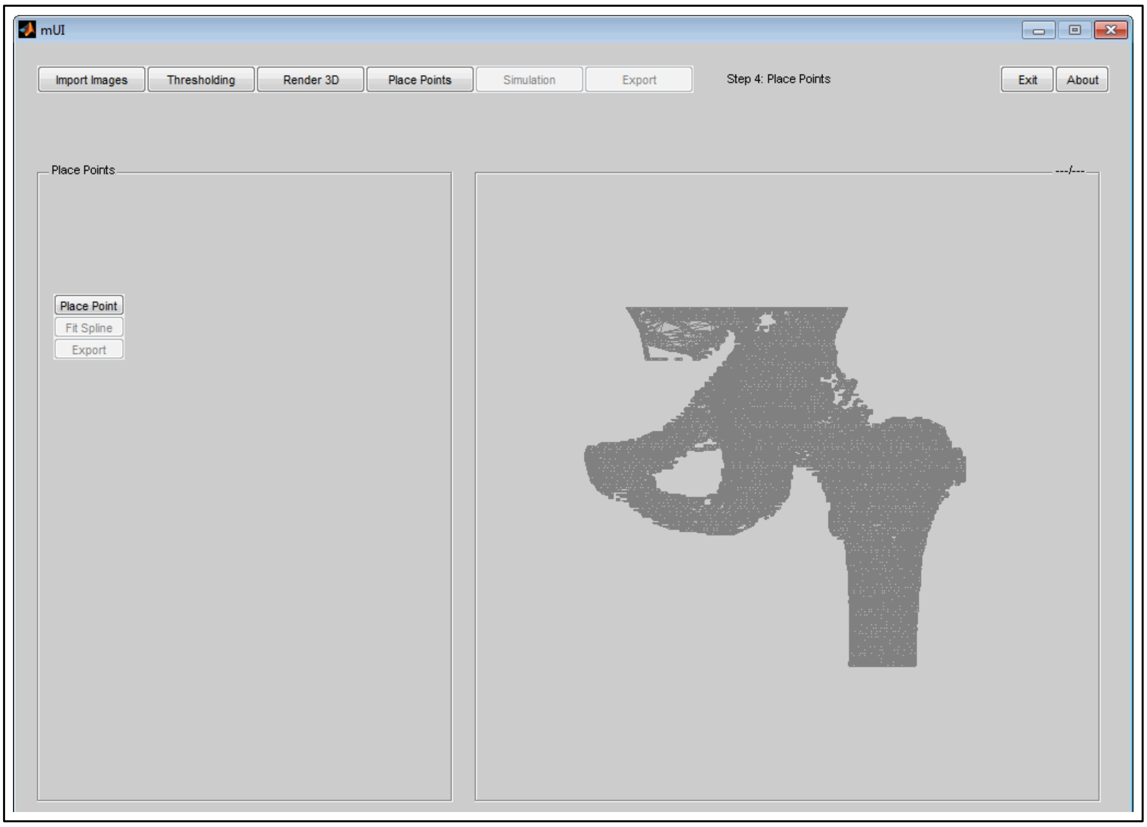


Figure 3.12. Rendered 3d model

When 3D model is rendered, then it is time for the surgeon to place points on the bone model to generate desired cutting trajectory. Surgeon can rotate 3D model for better view to place points on the model as shown in Figure 3.13. Positions of selected points are stored in the memory and 3D spline method is used to generate a smooth curve through a stored list of points in the order in which they are placed by the surgeon.

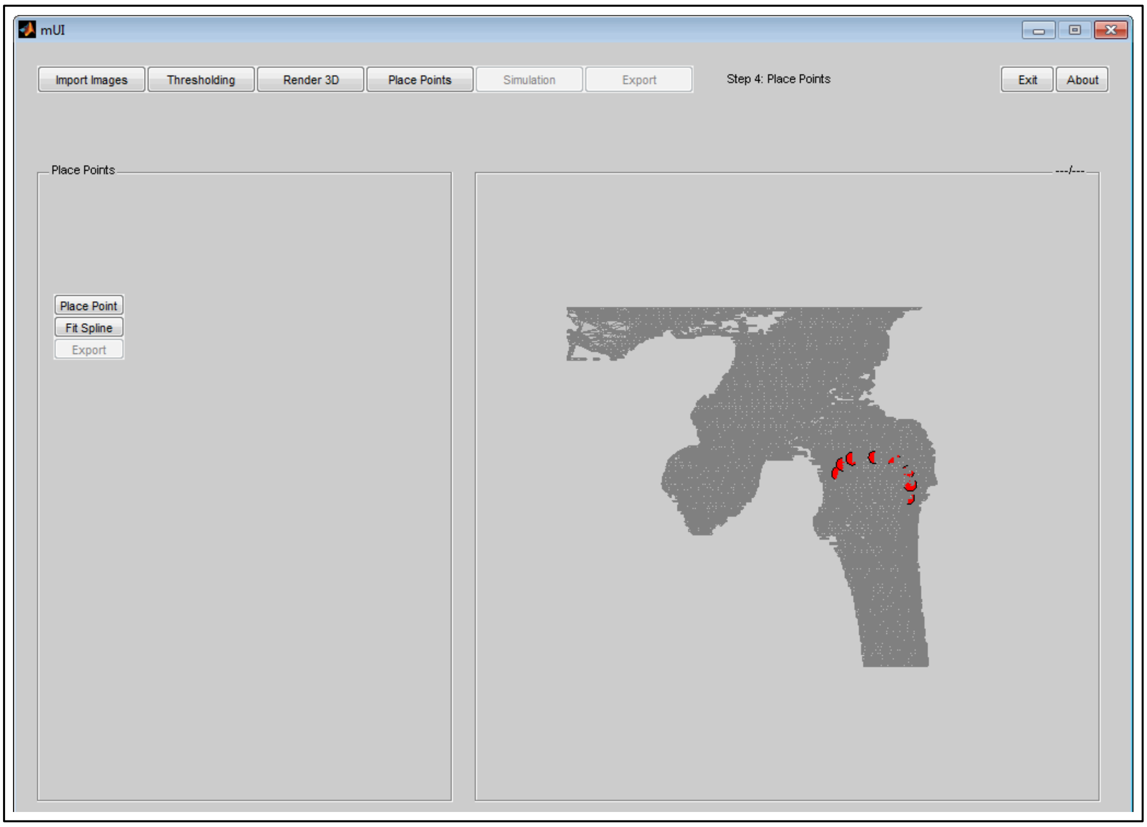


Figure 3.13. Placing points on 3d model

4. EXPERIMENTAL RESULTS

Initially experiments are performed to evaluate the camera system and user interface. Later low level controller gains are determined with series of experiments. Finally, the control architecture of OrthoRoby has been evaluated.

4.1. EVALUATION OF CAMERA SYSTEM

Initially experiments are performed to evaluate the camera system that is integrated into OrthoRoby system. The count of frames processed is calculated from the cameras during movement of OrthoRoby. Various methods are evaluated in calculation of frames processed per second. Positions of the markers are located over cutting tool and bone. Three dimensional position values of the markers have been calculated to activate the cutting tool when it is close enough to bone, to cut bone in desired depth and to take OrthoRoby back to starting point safely.

Two image processing methods have been evaluated and their performances are compared to determine positions of the markers. In first method, captured images have been processed at once by comparing their red, green and blue channels with lower and upper thresholds [26]. Necessary calculations have been performed by determining specific threshold values for two cameras separately to identify green marker on cutting tool and red marker on bone. These threshold values varies with different lighting conditions, however positions of markers are determined without any problem in constant conditions. In second method, captured images have firstly been separated into red, green and blue channels, and a mask is generated for each marker using suitable lower and upper thresholds. Then this mask is merged with original image to get the segmented image of markers. Positions of the markers are found by detecting edges in segmented images at the end of the process [27]. It has been observed that second method is preferable for small amount of images, however it is not preferable for real-time systems because of high processing load. The camera system was able to process 3.53 frames per second with first method and 3.4 frames per second with second method in same lighting conditions. The first method is selected in this study. Distance between the markers are calculated along moving direction using the first method, to determine if OrthoRoby is close enough to cutting area (Figure 4.1).

OrthoRoby’s cutting trajectory is determined with Dr Muharrem Inan and 0.75mm/s is selected for OrthoRoby’s speed. Distances between the markers are calculated and compared using images captured from each camera (Figure 4.2). OrthoRoby repeats the same movement 10 times, and the results show that distance error between two cameras are about 0.1 cm.

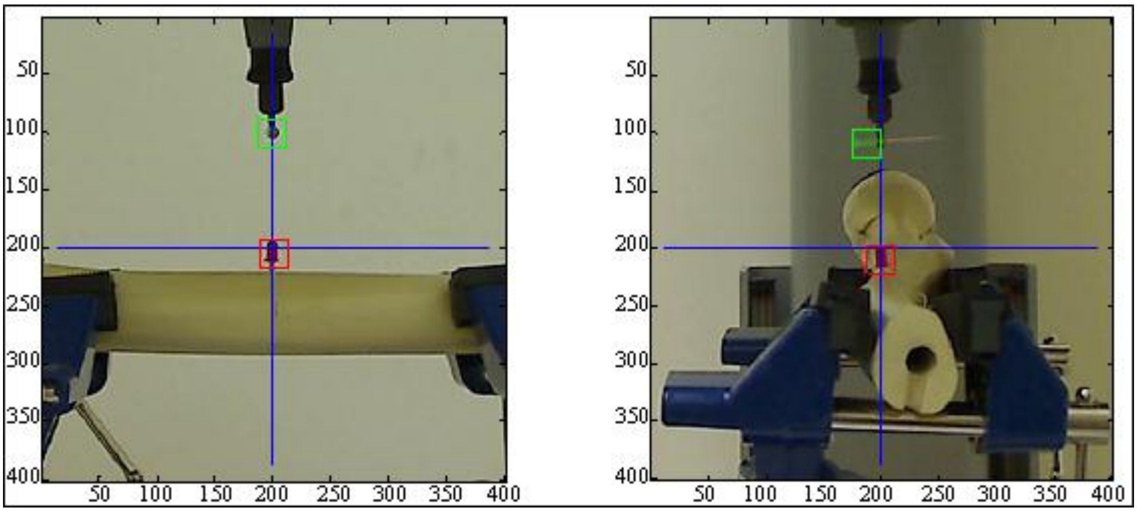


Figure 4.1. Marker distance calculation setup using two cameras

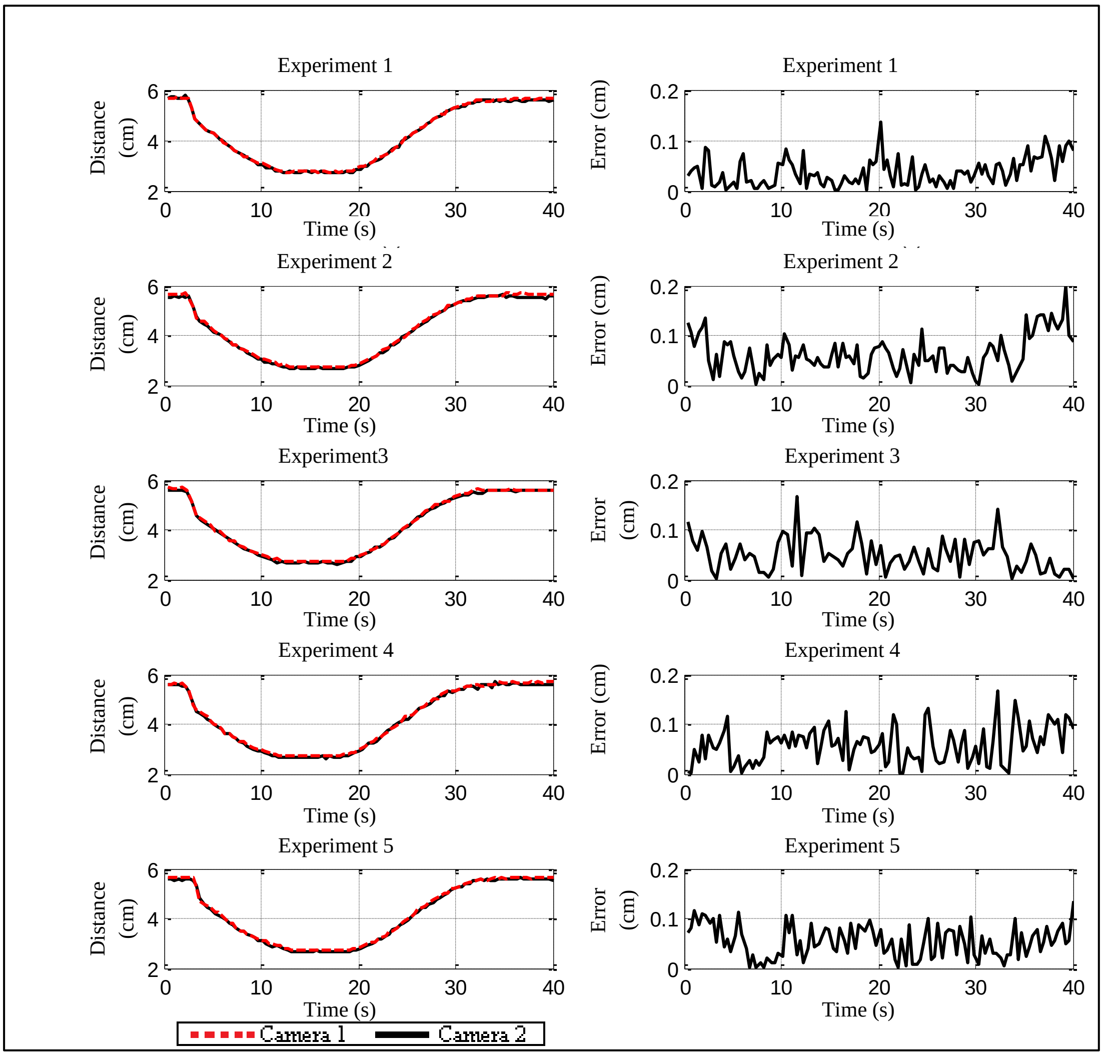


Figure 4.2. Distances between markers obtained from two cameras

Difference between mean distance values obtained from two cameras, and robot leg lengths obtained from encoders are evaluated for OrthoRoby's movements determined (Figure 4.3). It is observed that the maximum difference between robot leg lengths and mean leg length values obtained from two cameras are 0.4 cm. Robot's sampling frequency is 100 Hz and camera system's sampling frequency is 3 Hz, thus there is 100/3 times difference in calculated leg length values. In this situation it is observed that maximum error rate has been about 0.5 cm in repeated experiments, however this error value has been obtained in a small time interval when robot starts movement. When robot continues movement, the observed maximum error value has been about 0.2 cm which means that camera system calculates OrthoRoby's movement correctly.

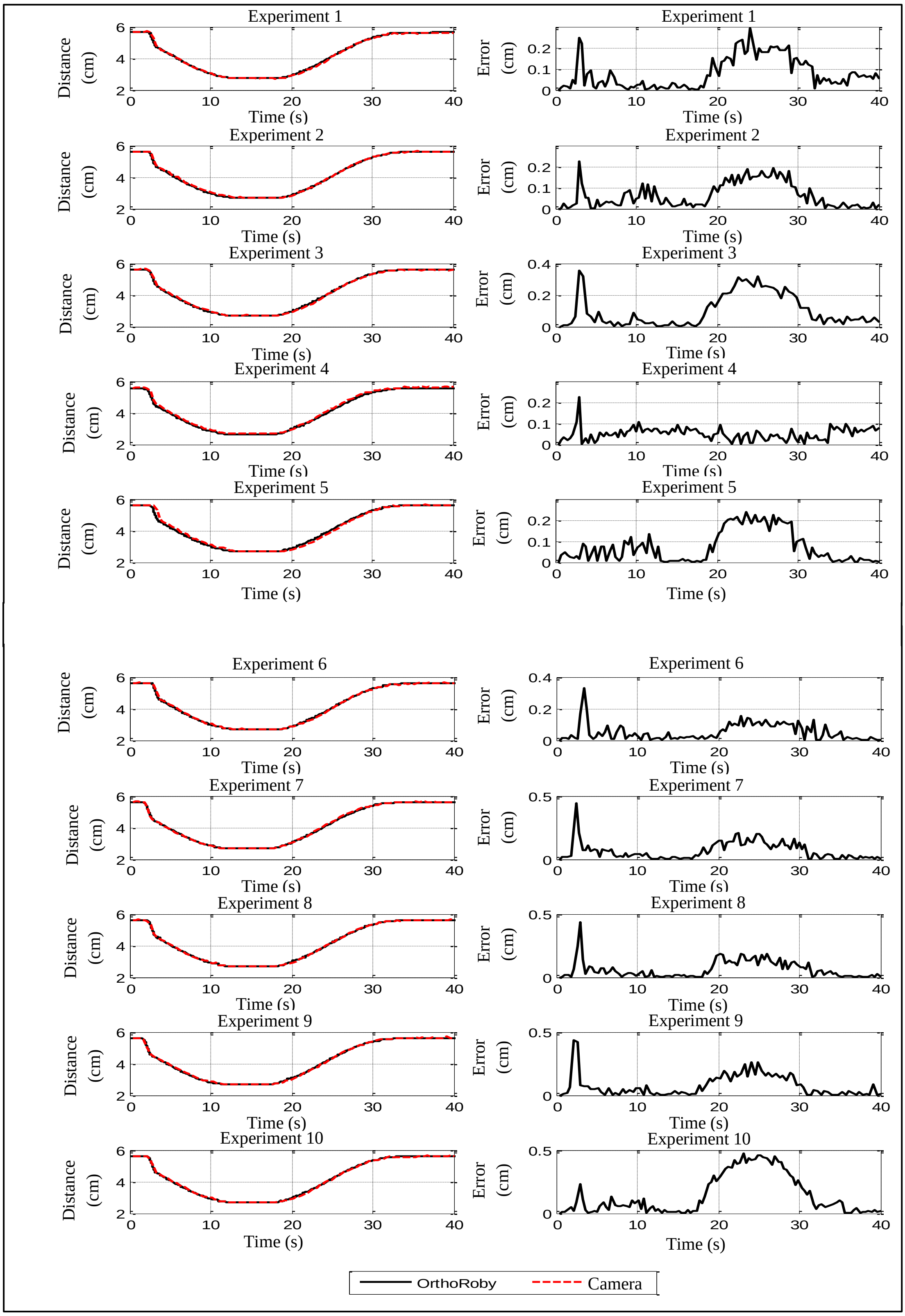


Figure 4.3. Robot leg lengths and mean lengths calculated by two cameras (Experiment 1 - Experiment 10)

Robot leg lengths and mean lengths calculated by two cameras have also been evaluated at OrthoRoby's movement at different speeds. Minimum, maximum, standard deviation and mean of the difference between two values are found and shown in Table 4.1.

Table 4.1. Minimum, maximum, standard deviation and mean of the difference between robot leg lengths and lengths calculated by two cameras

Speed (mm/s)	Minimum Error (cm)	Maximum Error (cm)	Standard Deviation (cm)	Mean (cm)
0.375	0	0.2372	0.0644	0.0988
0.1875	0	0.2186	0.0537	0.0924
0.125	0	0.2902	0.0702	0.0803
0.09375	0	0.1669	0.0398	0.0444
0.075	0	0.2377	0.0361	0.0487

Markers are used with camera system to start cutting tool when OrthoRoby moves towards the bone and stops cutting procedure when it reaches desired depth. Cutting procedure has been performed by activating cutting tool while camera system has been observing cutting tool's vertical distance to the bone if it has been close enough. Cutting tool has been disabled after it reaches desired depth by observing vertical depth with camera system during this time interval. It has been decided that cutting tool can be activated when vertical distance between cutting tool and bone reaches 4cm. The cutting tool has been disabled when it moves 1.25cm inside bone. Lengths calculated by two cameras have been presented in Figure 4.4. The time interval when cutting tool is active has been presented on the right side of Figure 4.4. It is observed that difference between the distance values obtained from two cameras is 0.05cm.

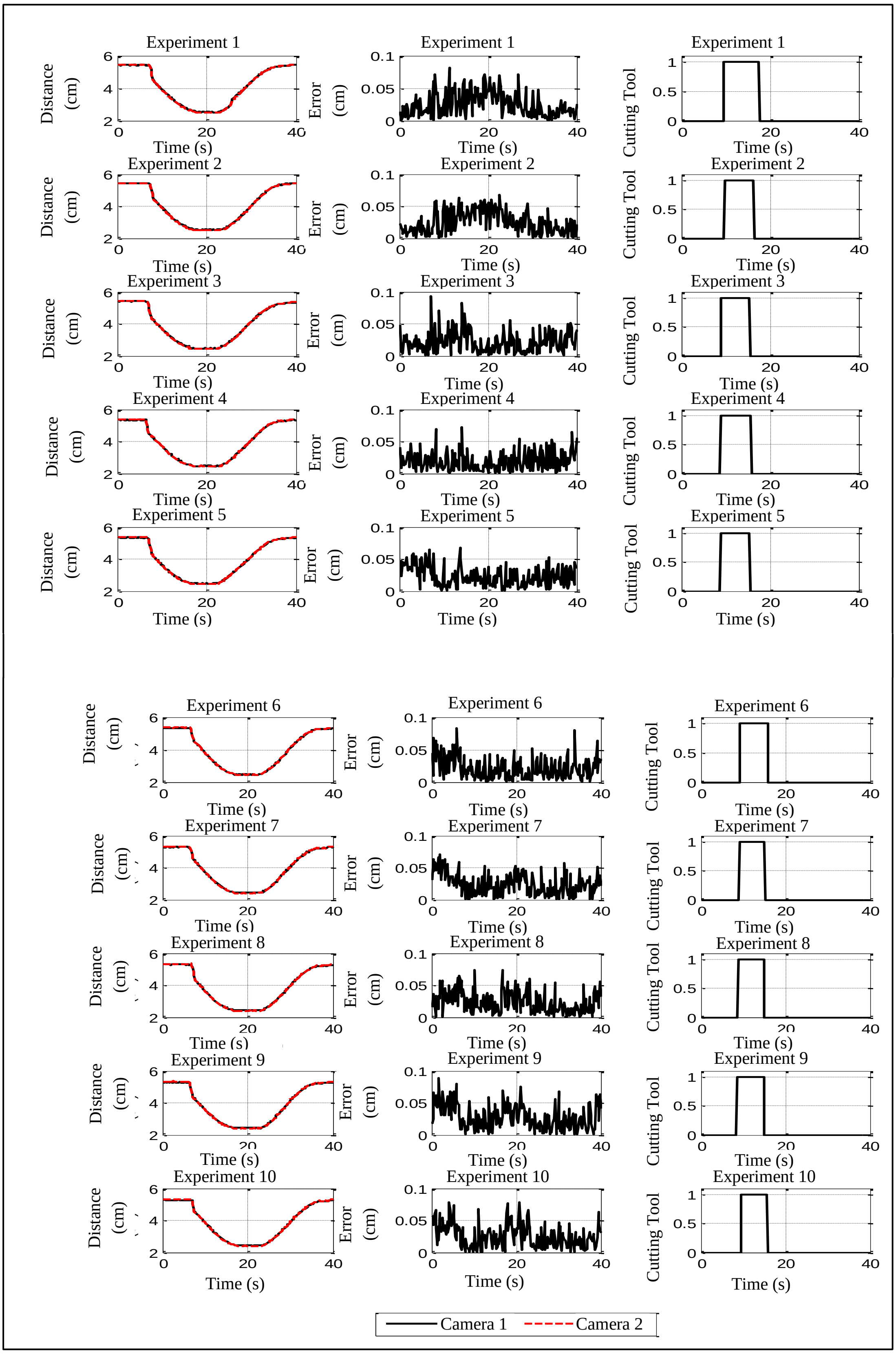


Figure 4.4. Comparison of distance between markers obtained from two cameras during all cutting procedures and activation status of cutting tool (Experiment 1-Experiment 10)

Comparison of OrthoRoby robotic system’s leg lengths and mean length values obtained from two cameras are shown in Figure 4.5. It is observed that the difference between two lengths is about 0.5cm. Therefore, OrthoRoby’s leg length movement has been calculated correctly even during cutting procedure.

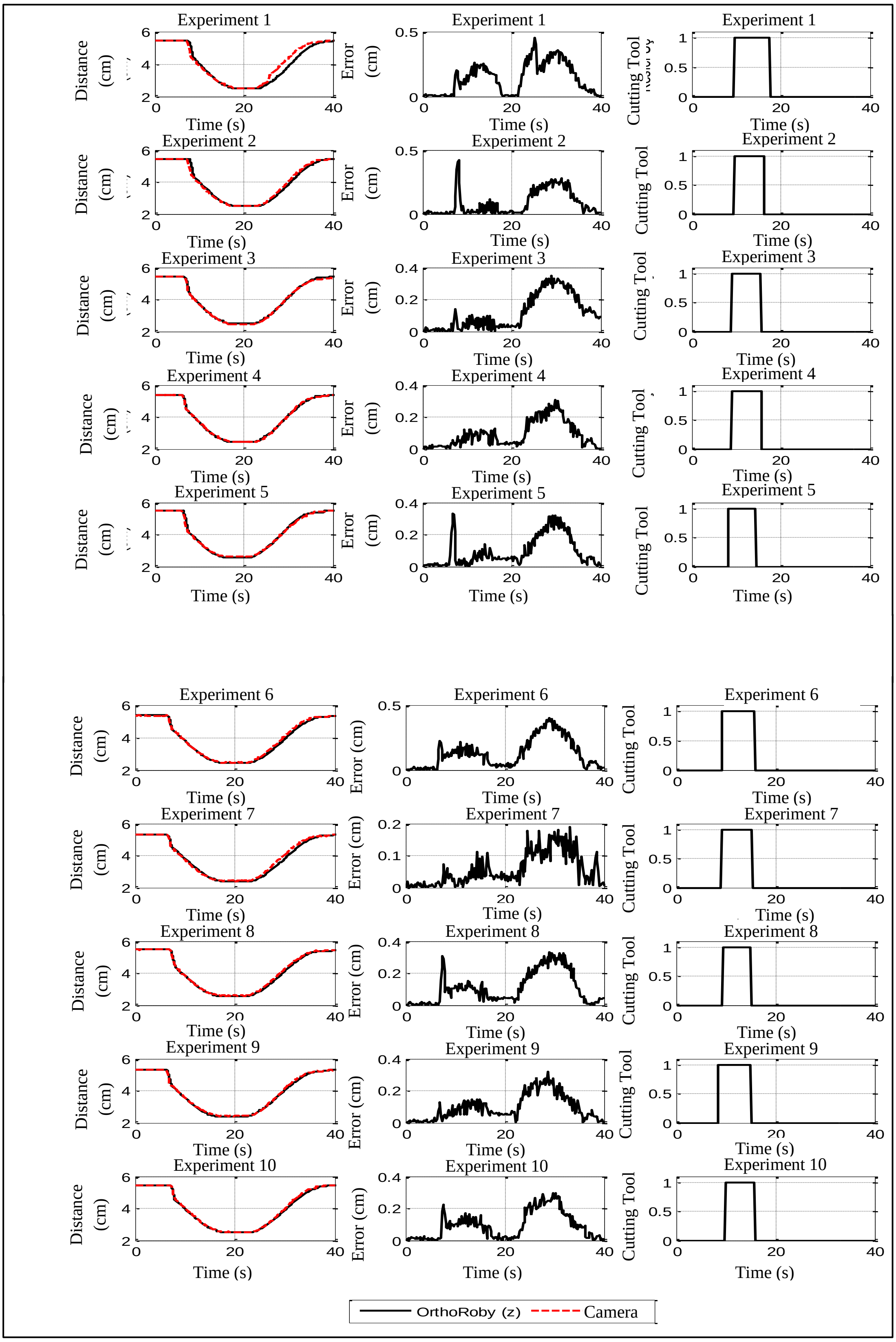


Figure 4.5. Comparison of distance between markers obtained from two cameras during all cutting procedures and activation status of cutting tool (Experiment 1-Experiment 10)

OrthoRoby completed cutting procedure at different speeds and mean lengths calculated by two cameras and robot leg lengths have been compared. Minimum, maximum, standard deviation and mean value of the difference between two values are calculated as shown in Table 4.2.

Table 4.2. Minimum, maximum, standard deviation and mean value of the difference between mean lengths calculated by two cameras and robot leg lengths (with bone cutting)

Speed (mm/s)	Minimum Error (cm)	Maximum Error (cm)	Standard Deviation (cm)	Mean (cm)
0.375	0	0.2126	0.0568	0.0708
0.1875	0	0.4576	0.0564	0.0892
0.125	0	0.3148	0.0921	0.1281
0.09375	0	0.3981	0.0406	0.0617
0.075	0	0.3579	0.0552	0.0762

Results of the experiments with camera system shows that the cutting tool with markers is useful to determine the closeness to complete cutting procedure with desired cut depth.

4.2. EVALUATION OF USER INTERFACE

Two experiments are designed to evaluate the user interface. In the first experiment surgeon places points on 3D model of patient’s bone as shown in Figure 4.6. Desired bone cutting trajectory generated using placed points is shown in Figure 4.7. X, Y, Z defines OrthoRoby’s cutting tool position and α (alpha), β (beta), γ (gamma) defines OrthoRoby’s cutting tool orientation (Figure 4.7).

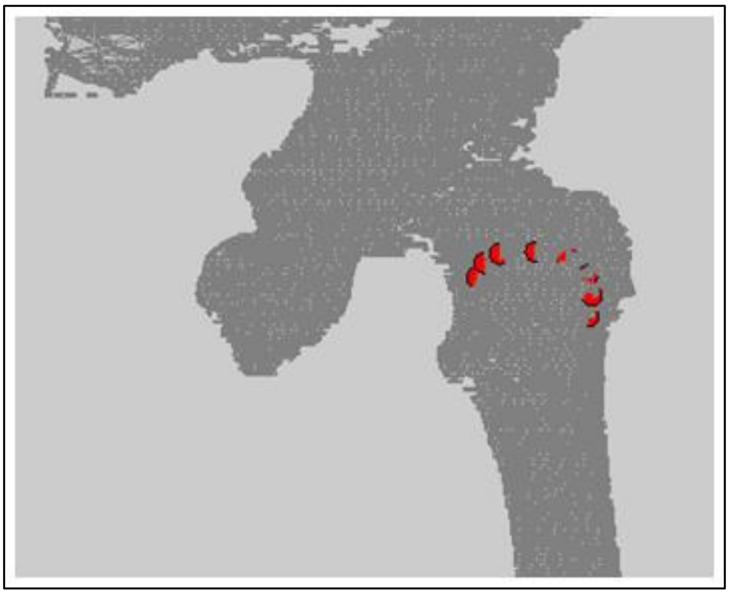


Figure 4.6. Placing points on 3D model

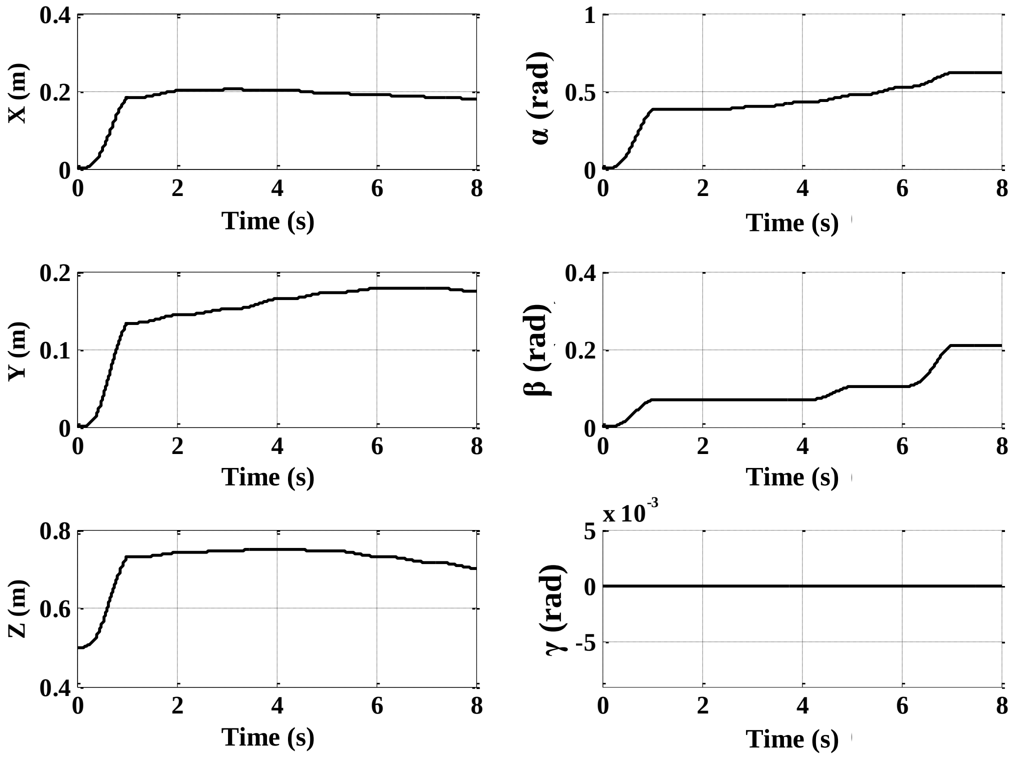


Figure 4.7. Desired bone cutting trajectory

In the second experiment, surgeon uses the interface to modify bone cutting trajectory after a problem which may occur during bone cutting procedure. For example, surgeon places new points on patient's 3D bone model and generates a new bone cutting trajectory (Figure 4.8). The bone cutting trajectory generates using new points placed by surgeon has been shown in Figure 4.9. It can be seen that position have not change much, however orientation is different. Note that it is possible to design different trajectories using user interface with different point choices of surgeon.

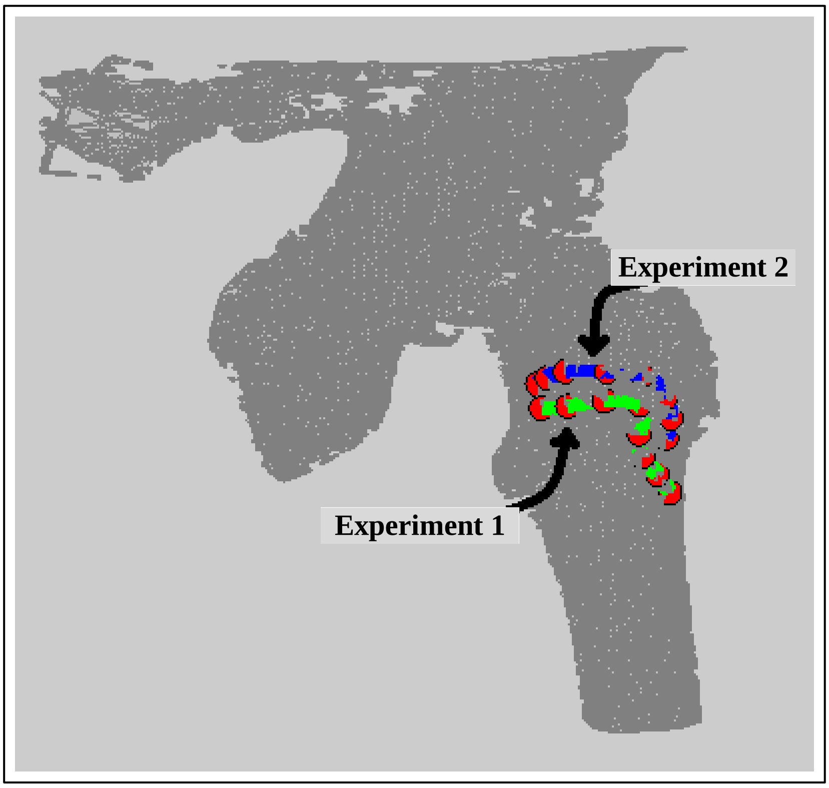


Figure 4.8. Modification of points on 3D Model

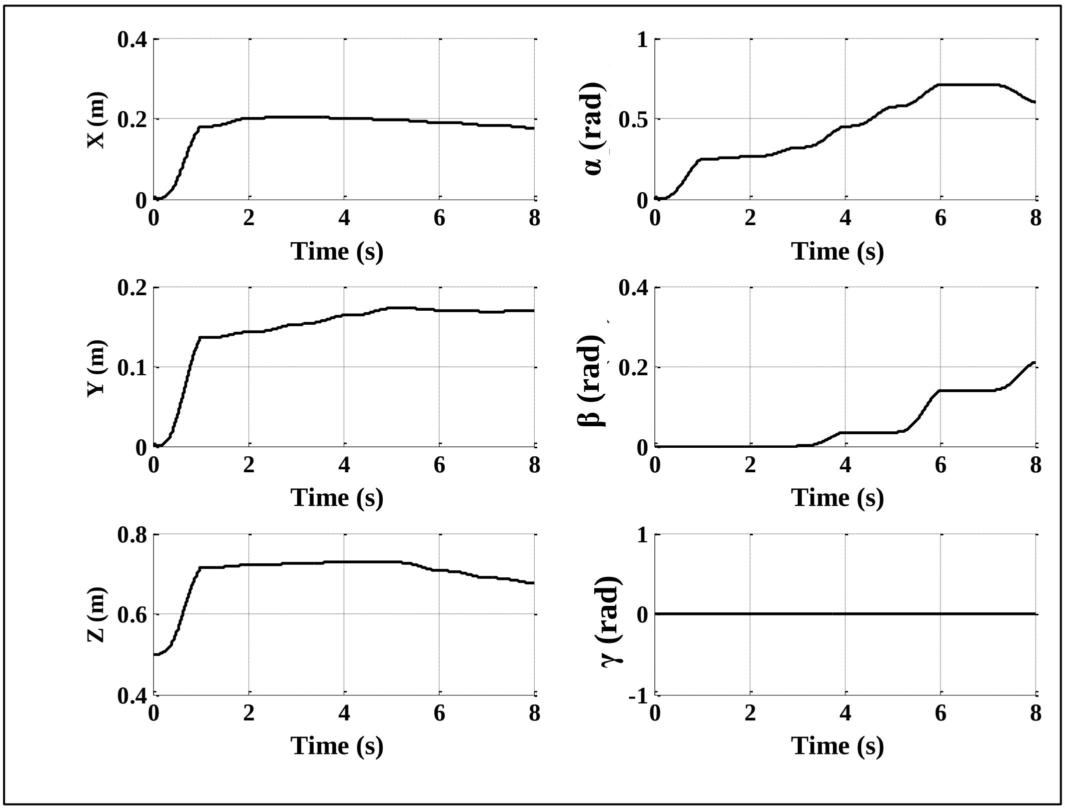


Figure 4.9. Bone cutting trajectory when unexpected situation happened

4.3. DETERMINATION OF LOW LEVEL CONTROLLER GAINS

Best computed torque controller gains have been found for OrthoRoby to complete bone cutting procedure with least error. OrthoRoby's bone cutting procedure has been evaluated using different controller gains. Low-level controller gains have been presented as K_p for proportional gain, K_i for integral gain and K_d for derivative gain. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during the experiments with different control gains have been presented for each experiment. Totally 14 experiments have been completed and 5 most significant of them are presented in here.

Oscillation of OrthoRoby’s movement has been checked to determine the best gain values for computed torque controller. Gains with minimum oscillations have been selected. Frequency components of leg length errors have also been calculated to determine how much oscillation has been occurred during the experiments. Discrete Fourier Transform method is used to calculate frequency components of leg length errors [28]. As shown in Equation 4.1, each sample of leg length errors during the experiments are represented as $x_i(n)$ and their calculated frequency components are represented as $X_i[k]$.

$$X_i[k] = \sum_{n=0}^{N-1} x_i[n] e^{-j2\pi kn/N} \tag{4.1}$$

In first experiment, it is observed that fifth leg’s frequency components included more high frequencies than that of other legs as shown in Figure 4.10. So that, fifth leg’s K_p value is decreased 25 percent and K_i value is also decreased 25 percent for next experiment as shown in Table 4.3.

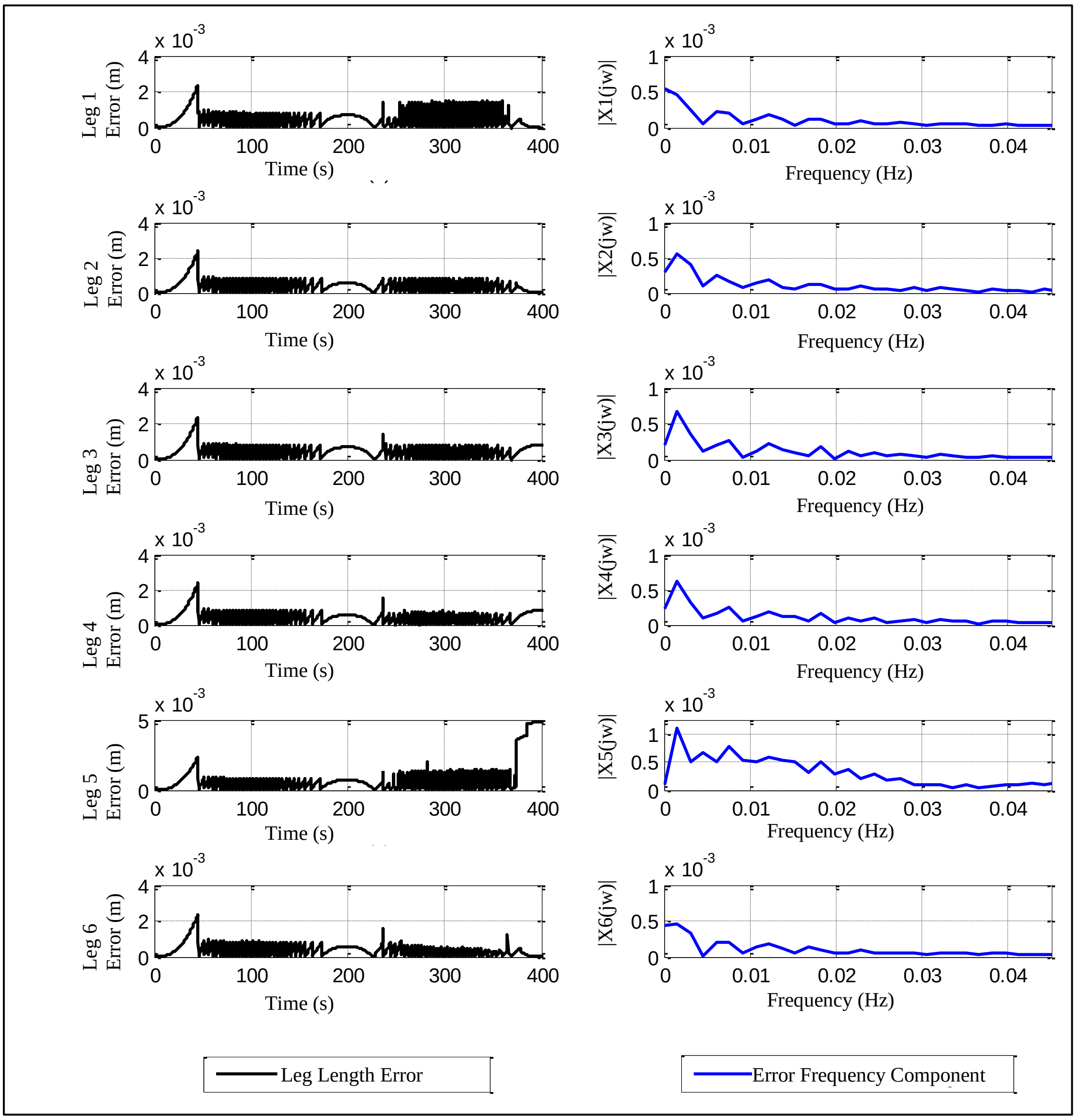


Figure 4.10. OrthoRoby's leg length errors and their frequency components (Experiment 1)

Table 4.3. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during experiments with different control gains (Experiment 1)

Leg	K_p	K_i	K_d	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	10000	2000	3125	2,3665	0	0,35623	0,40162
2	10000	2000	3125	2,3586	0	0,32877	0,4141
3	10000	2000	3125	2,3787	0	0,32954	0,46813
4	10000	2000	3125	2,3599	0	0,32835	0,42492
5	10000	2000	3125	4,812	0	1,1032	0,71351
6	10000	2000	3125	2,3749	0	0,3307	0,37058

In second experiment, it is observed that leg 5’s high frequency component decreased and increased for leg 1 and 6 as shown in Figure 4.11. For next experiment K_d value of all legs are decreased 20 percent as shown in Table 4.4.

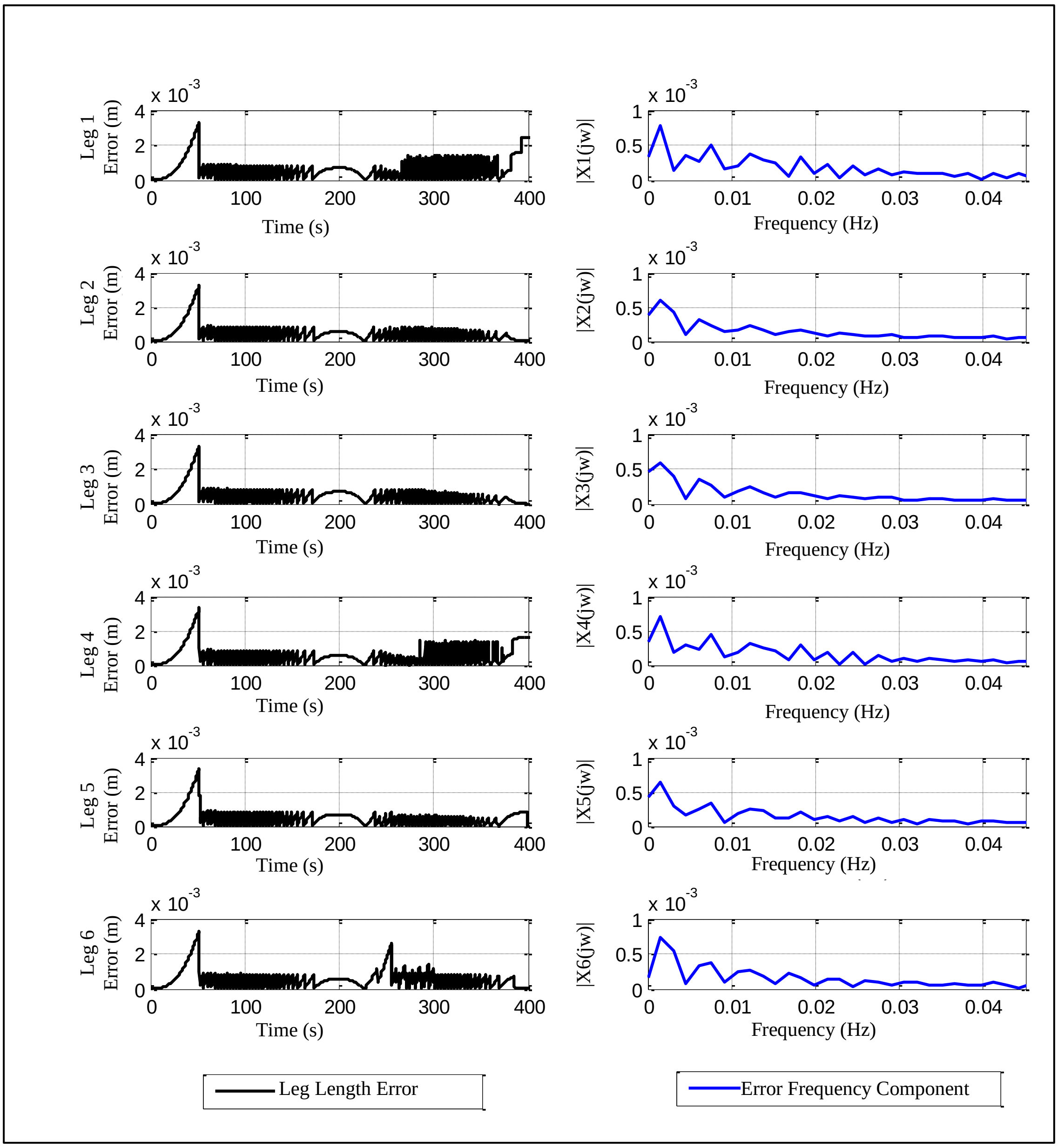


Figure 4.11. OrthoRoby's leg length errors and their frequency components (Experiment 2)

Table 4.4. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during experiments with different control gains (Experiment 2)

Leg	K_p	K_i	K_d	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	10000	2000	3125	3,3189	0	0,5899	0,5459
2	10000	2000	3125	3,3038	0	0,4477	0,4315
3	10000	2000	3125	3,3108	0	0,4516	0,4402
4	10000	2000	3125	3,3054	0	0,5167	0,4979
5	7500	1500	3125	3,3189	0	0,4529	0,4540
6	10000	2000	3125	3,3087	0	0,5001	0,5303

In third experiment, it is observed that leg 4 and 5’s high frequency components are increased as shown in Figure 4.12. For next experiment K_d values are decreased 17 percent for all legs as shown in Table 4.5.

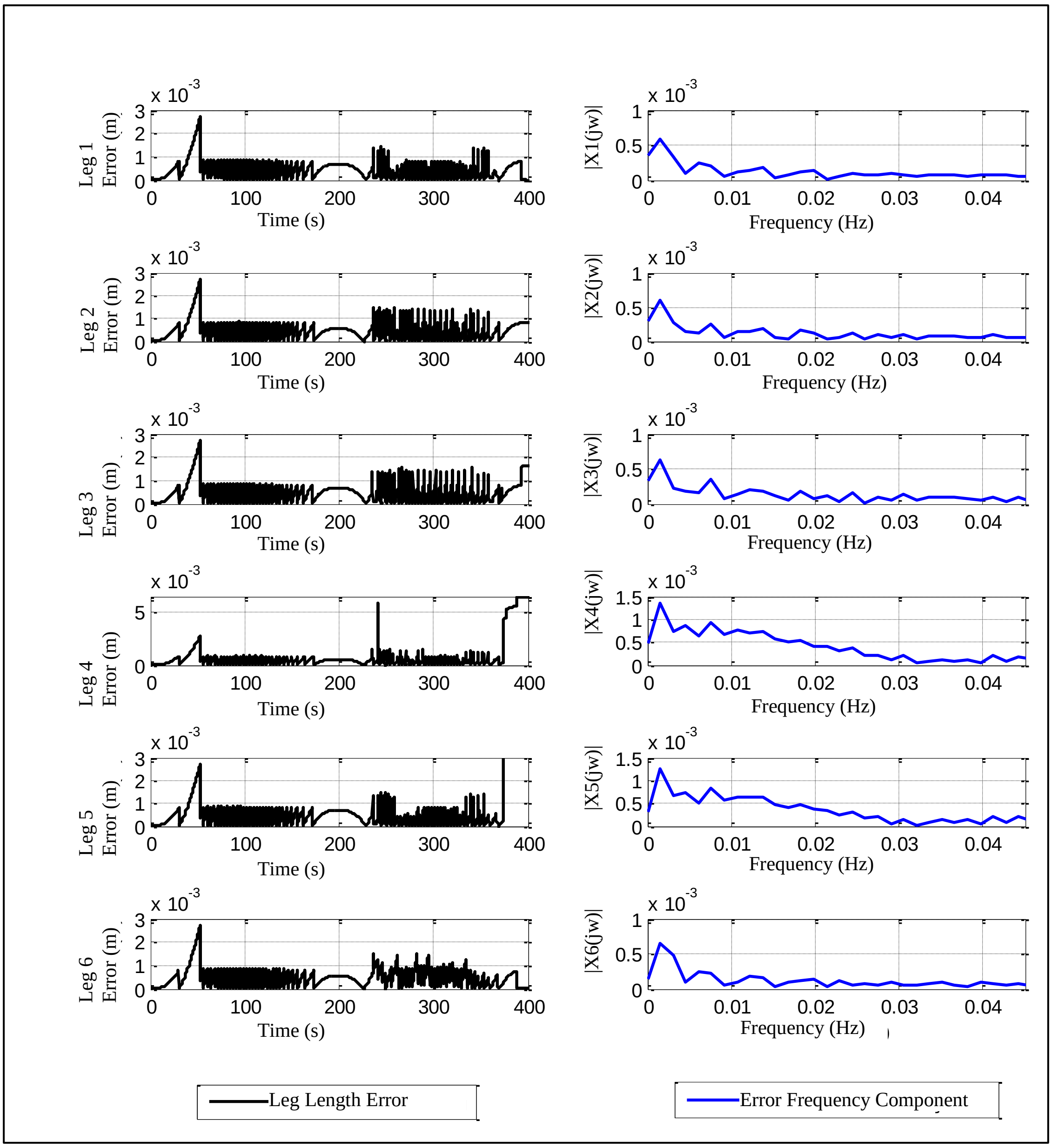


Figure 4.12. OrthoRoby's leg length errors and their frequency components (Experiment 3)

Table 4.5. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during experiments with different control gains (Experiment 3)

Leg	K_p	K_i	K_d	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	9375	1500	3750	2,7066	0	0,35664	0,42322
2	10000	2000	3750	2,7189	0	0,35455	0,4349
3	9375	1500	3750	2,7334	0	0,40291	0,4719
4	10000	2000	3750	6,416	0	1,4691	0,81592
5	9375	1500	3750	5,614	0	1,2552	0,7619
6	10000	2000	3750	2,7156	0	0,37244	0,47301

In fourth experiment, again all legs' high frequency components remain same, but leg 6's response delay remained as shown in Figure 4.13 thus its K_p gain has been increased 5 percent for the next experiment as shown in Table 4.15.

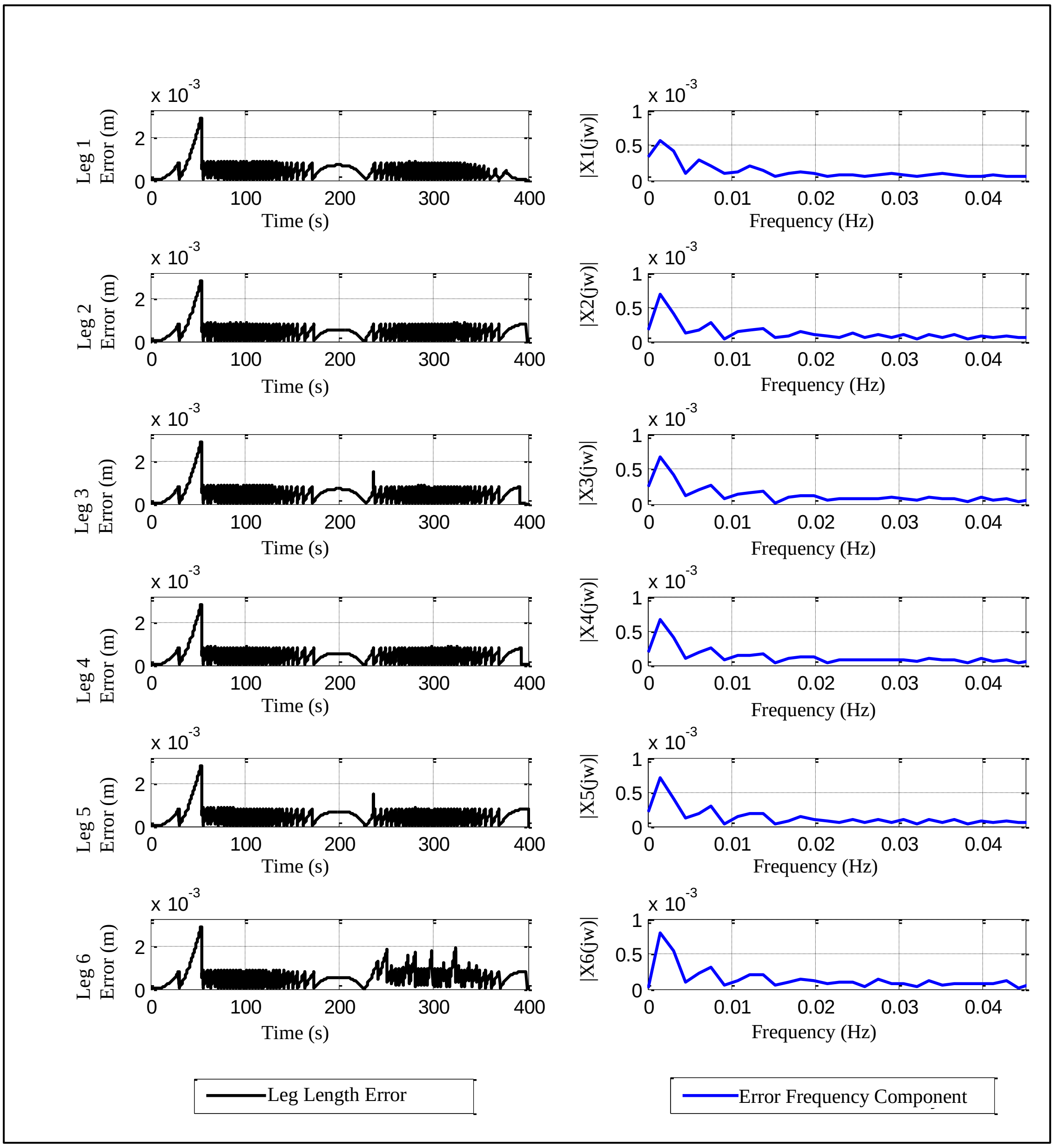


Figure 4.13. OrthoRoby's leg length errors and their frequency components (Experiment 4)

Table 4.6. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during experiments with different control gains (Experiment 4)

Leg	K_p	K_i	K_d	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	9375	1500	4375	2,8948	0	0,37555	0,43322
2	10000	1500	4375	2,8642	0	0,36296	0,4661
3	9375	1500	4375	2,8896	0	0,37057	0,46459
4	10000	1500	4375	2,8625	0	0,36544	0,45077
5	9375	1354	4375	2,869	0	0,36513	0,4831
6	12500	2000	4375	2,8795	0	0,41076	0,55547

In fifth experiment it is observed that high frequency components of all legs remain same and leg 6’s response delay reaches at its minimum level as shown in Figure 4.14. Therefore computed torque controller gains used in this experiment are chosen as best gains, and the K_p, K_i, K_d gains presented on Table 4.7, Experiment 5 are used.

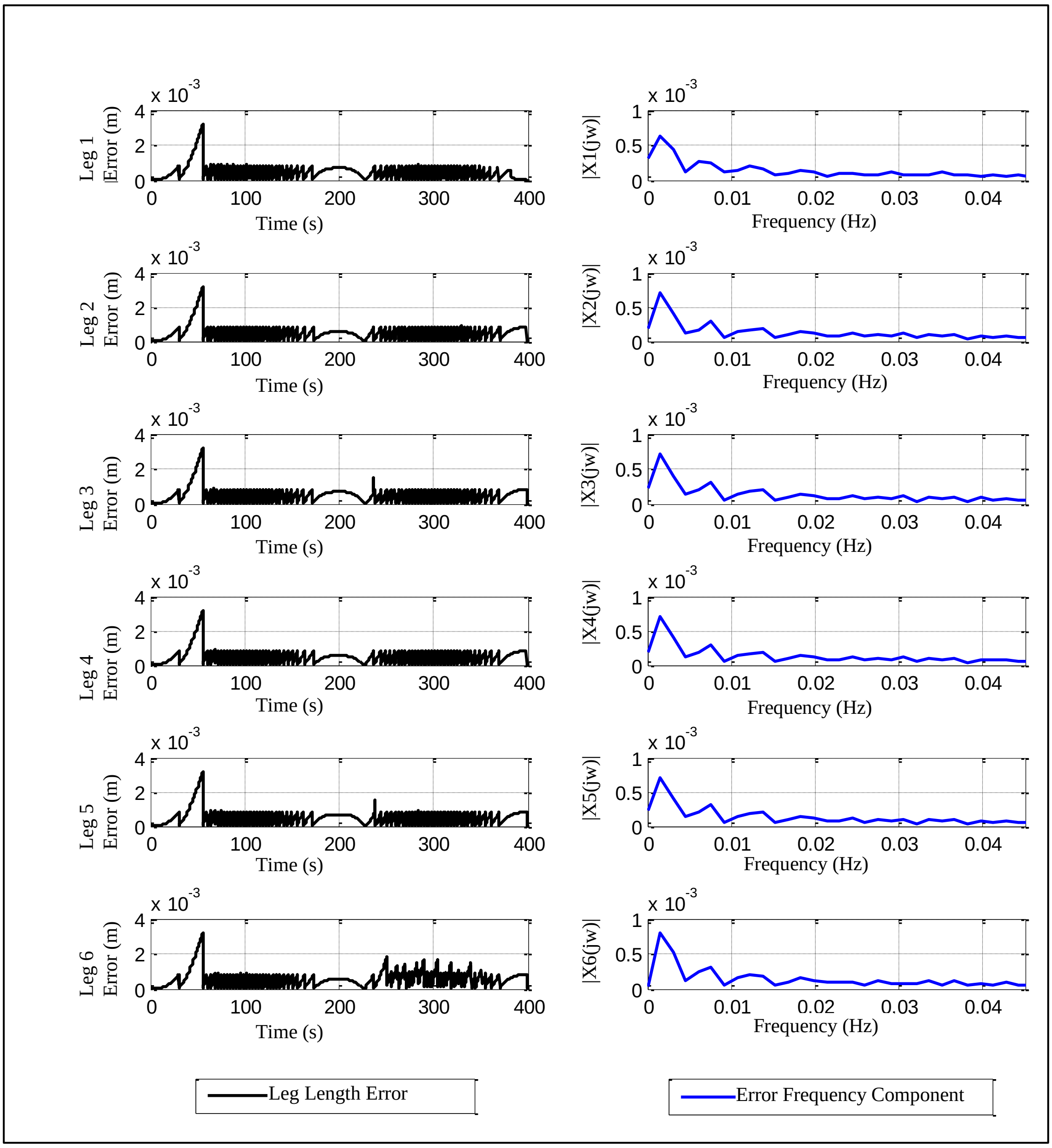


Figure 4.14. OrthoRoby’s leg length errors and their frequency components (Experiment 5)

Table 4.7. Minimum, maximum, standard deviation and main values of difference between desired and actual leg lengths during experiments with different control gains (Experiment 5)

Leg	K_p	K_i	K_d	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	9375	1500	4375	3,1961	0	0,41154	0,4508
2	10000	1500	4375	3,1653	0	0,40002	0,47293
3	9375	1500	4375	3,1943	0	0,40501	0,48593
4	10000	1500	4375	3,1653	0	0,40057	0,47233
5	9375	1354	4375	3,1711	0	0,40248	0,48769
6	13125	2000	4375	3,1795	0	0,43902	0,55513

4.4. EVALUATION OF ORTHOROBY’S CONTROL ARCHITECTURE

OrthoRoby and its control architecture has also been evaluated using real bone in cutting procedure. In all evaluations OrthoRoby is asked to move forward 5 cm in 200 seconds along z-axis, and move back to starting position. A similar material, a piece of wood with 3 cm thickness is used before real-time experiments on bone. Camera system observes OrthoRoby’s movement, and sends commands to high-level controller to start and stop the cutting tool as required. The desired task is to move OrthoRoby towards the wood, to cut it and to move OrthoRoby back to starting position. This task has been repeated three times. Minimum, maximum, standard deviation and mean of robot leg length error are shown in Table 4.8. Error figures and error frequency components are shown in Figure 4.15 and 4.16, for one of three times repeated experiments respectively.

Table 4.8. Minimum, maximum, standard deviation and mean values of difference between desired and actual robot leg lengths

Experiment Number	Leg	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	1	3,7208	0	0,4837	0,4605
	2	3,7310	0	0,4429	0,4058
	3	3,7386	0	0,4399	0,4249
	4	3,7208	0	0,4473	0,3910
	5	3,7540	0	0,4418	0,3956
	6	3,7438	0	0,4820	0,4978
2	1	3,4379	0	0,4115	0,4048
	2	3,4558	0	0,4092	0,4084
	3	3,4650	0	0,4064	0,4209
	4	3,4386	0	0,4101	0,3818
	5	3,4724	0	0,4072	0,4031
	6	3,4533	0	0,4346	0,4890
3	1	3,1318	0	0,3790	0,3833
	2	3,1049	0	0,3725	0,3914
	3	3,1012	0	0,3685	0,4057
	4	3,1190	0	0,3777	0,3663
	5	3,1460	0	0,3746	0,3921
	6	3,1261	0	0,4028	0,4717

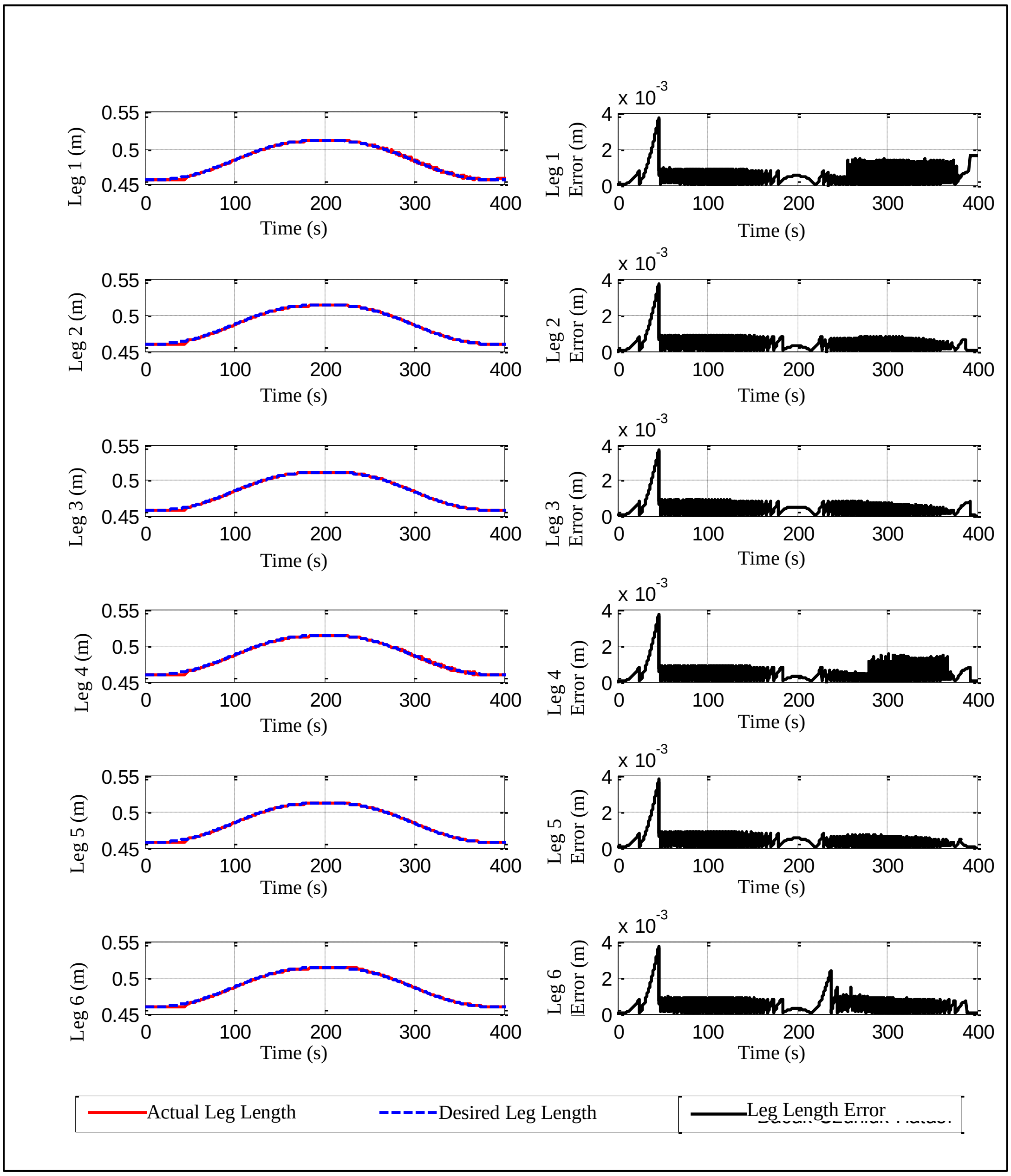


Figure 4.15. OrthoRoby’s leg length changes and their errors (without cutting wood)

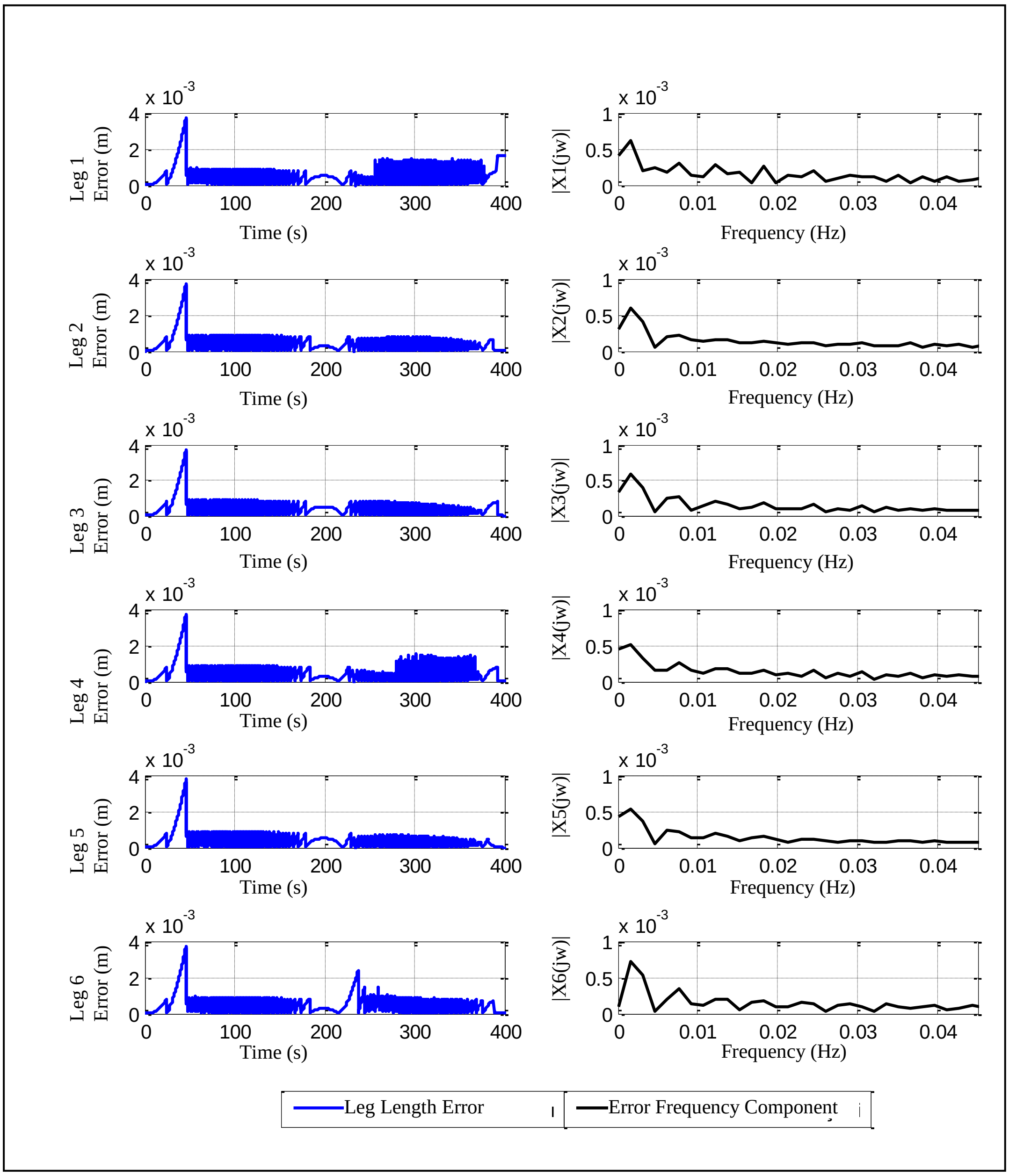


Figure 4.16. OrthoRoby’s leg length errors and their frequency components (without cutting wood)

Next, OrthoRoby repeats the cutting task explained above, but this time cutting tool cut the wood and the movement has been evaluated. Minimum, maximum, standard deviation and mean values of leg length errors are shown in Table 4.9. This task has been repeated three times and error figure and error frequency components are shown for one of them in Figure 4.17 and Figure 4.18 respectively.

Table 4.9. Minimum, maximum, standard deviation and mean values of difference between desired and actual robot leg lengths (cutting wood)

Experiment Number	Leg	Maximum Error (mm)	Minimum Error (mm)	Standard Deviation (mm)	Mean (mm)
1	1	3,3498	0	0,3964	0,3997
	2	3,3726	0	0,3994	0,4229
	3	3,3864	0	0,3960	0,4371
	4	3,3556	0	0,3978	0,4030
	5	3,3864	0	0,3960	0,4366
	6	3,3677	0	0,4140	0,4938
2	1	2,6236	0	0,3420	0,3927
	2	2,6042	0	0,4619	0,5293
	3	2,5964	0	0,3847	0,5105
	4	2,6084	0	0,4135	0,4937
	5	2,6340	0	0,3362	0,4617
	6	2,6104	0	0,3612	0,4882
3	1	3,0234	0	0,3691	0,3849
	2	3,0352	0	0,3939	0,4724
	3	3,0636	0	0,3656	0,4579
	4	3,0285	0	0,3728	0,4222
	5	3,0546	0	0,3668	0,4031
	6	3,0352	0	0,3906	0,4755

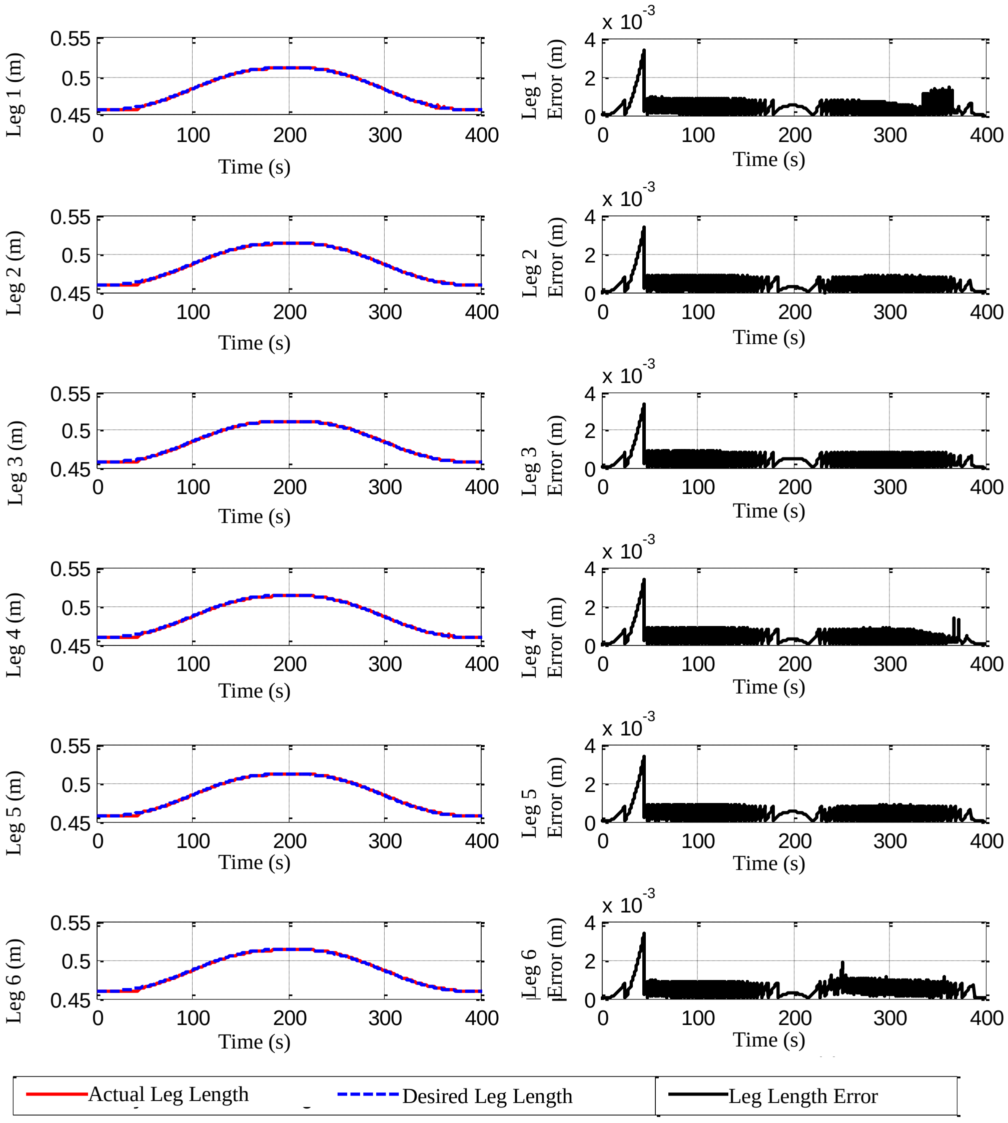


Figure 4.17. OrthoRoby’s leg length changes and their errors (cutting wood)

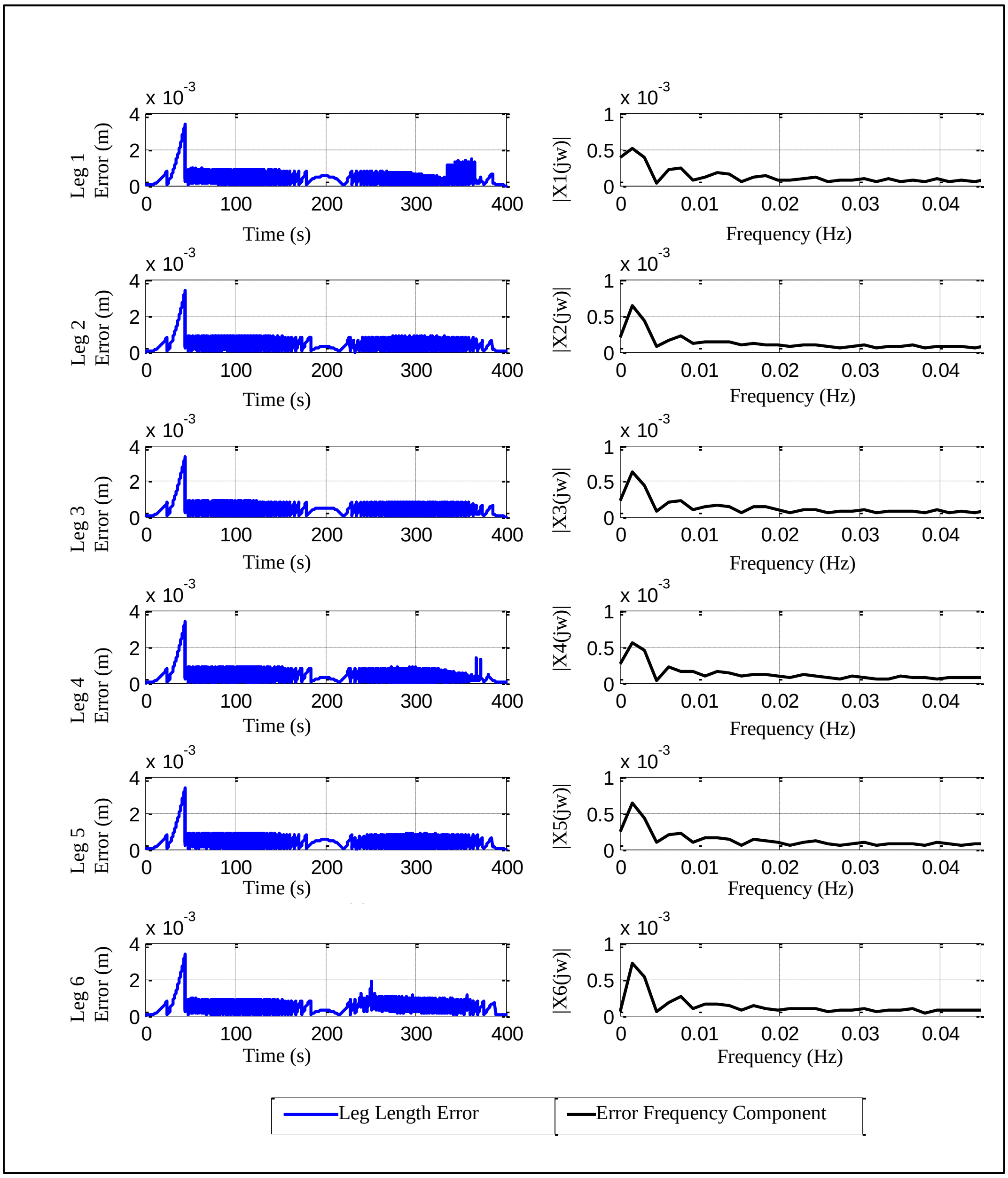


Figure 4.18. OrthoRoby’s leg length errors and their frequency components (cutting wood)

Table 4.10 presents leg length error changes obtained from experiments with and without cutting. Experiments with cutting are shown as “Experimentx.a” in table while experiments without cutting are shown as “Experimentx.b”. Changes in length errors are calculated for each leg in millimeter unit by subtracting two values obtained from experiments with and without cutting. In the experiments complete with 5 cm movement, it has been observed that mean error changes for each leg are (-0.0756)-0.0164, 0.0145-

0.1379, 0.0122-0.1048, 0.0120-0.1274, 0-0.0696 and (-0.0223)-0.0221. It can be said that these change values are reasonable for this range of movement. As represented in Table 4.18 values in four columns of three rows noted for each legs are close to each other. In this case it has easily been understood that the effective experiments are ones with cutting bone. Thus it is less probable to get same results each time in experiments with cutting than experiments without cutting. The reason is varying resistance force against OrthoRoby and cutting tool during bone cutting experiments. It is also observed that leg 6's error is less than the other ones and in inverse direction (Table 4.19). The reason for that is higher internal friction of leg 6 than the other legs. On the other hand, it can be said that leg 1's error rates is effected by leg6's internal friction, because legs 1-6, 2-3 and 4-5 moves as pairs as a result of OrthoRoby's kinematics structure.

Table 4.10. Comparison of error differences occurred during experiments with and without cutting

		Experiment 1.a	Experiment 2.a	Experiment 3.a
Robot Leg 1	Experiment 1.b	-0.0608	-0.0678	-0.0756
	Experiment 2.b	-0.0051	-0.0121	-0.0199
	Experiment 3.b	-0.0756	0.0164	0.0038
Robot Leg 2	Experiment 1.b	0.0171	0.1235	0.0666
	Experiment 2.b	0.0145	0.1209	0.0640
	Experiment 3.b	0.0315	0.1379	0.0810
Robot Leg 3	Experiment 1.b	0.0122	0.0856	0.0330
	Experiment 2.b	0.0162	0.0896	0.0370
	Experiment 3.b	0.0314	0.1048	0.0522
Robot Leg 4	Experiment 1.b	0.0120	0.1027	0.0312
	Experiment 2.b	0.0212	0.1119	0.0404
	Experiment 3.b	0.0367	0.1274	0.0559
Robot Leg 5	Experiment 1.b	0.0410	0.0661	0.0075
	Experiment 2.b	0.0335	0.0586	0
	Experiment 3.b	0.0445	0.0696	0.0110
Robot Leg 6	Experiment 1.b	-0.0040	-0.0096	-0.0223
	Experiment 2.b	0.0048	-0.0008	-0.0135
	Experiment 3.b	0.0221	0.0165	0.0038

Finally, control architecture has been evaluated with a cadaver bone. OrthoRoby is asked to move towards the bone and then cutting tool is activated to drill the bone (Figure 4.19). OrthoRoby's desired and actual leg lengths, leg length errors and error frequency components for each leg are shown in Figure 4.20. High-level controller's state changes during bone cutting are presented in Figure 4.21. First, $\tilde{s}_{1f}$ state is activated and OrthoRoby moves towards cadaver bone as shown in Figure 4.21. Then, $\tilde{s}_2$ state is activated and cutting tool started cutting as shown in Figure 4.22 and the cadaver bone has been cut. Later, $\tilde{s}_{1b}$ has been activated and OrthoRoby moves back to starting point. Finally, $\tilde{s}_3$ has been activated and OrthoRoby's movement is complete. The results show that OrthoRoby robotic system and control architecture can be used to cut cadaver bone successfully.

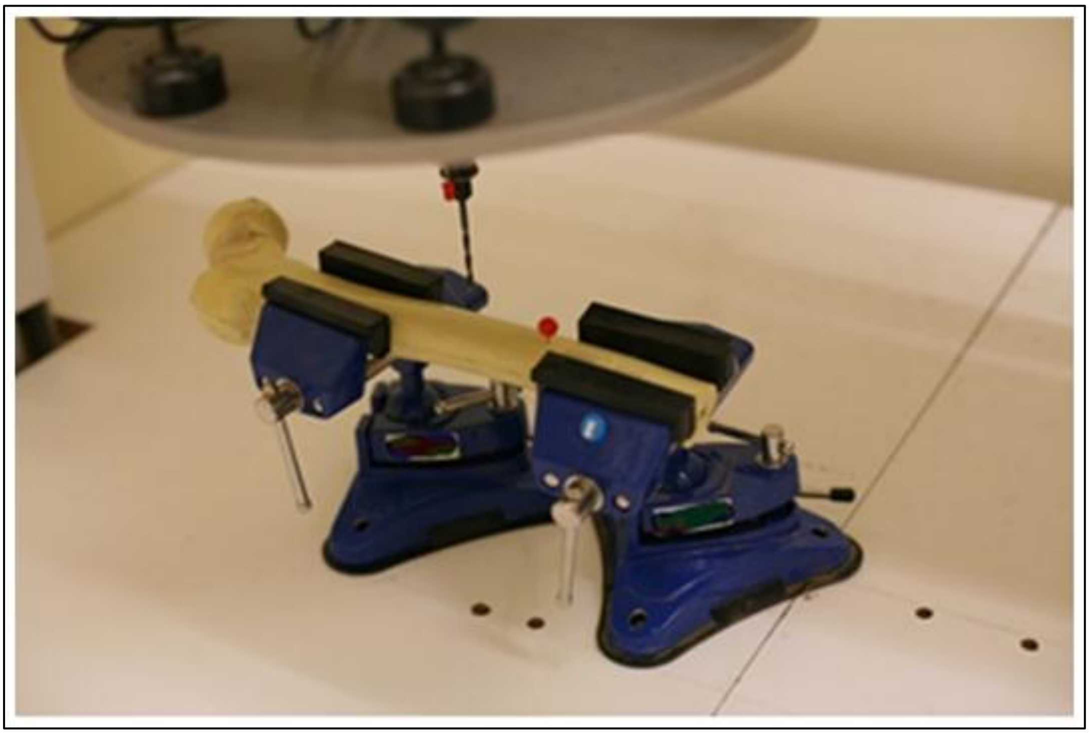


Figure 4.19. OrthoRoby cutting cadaver bone

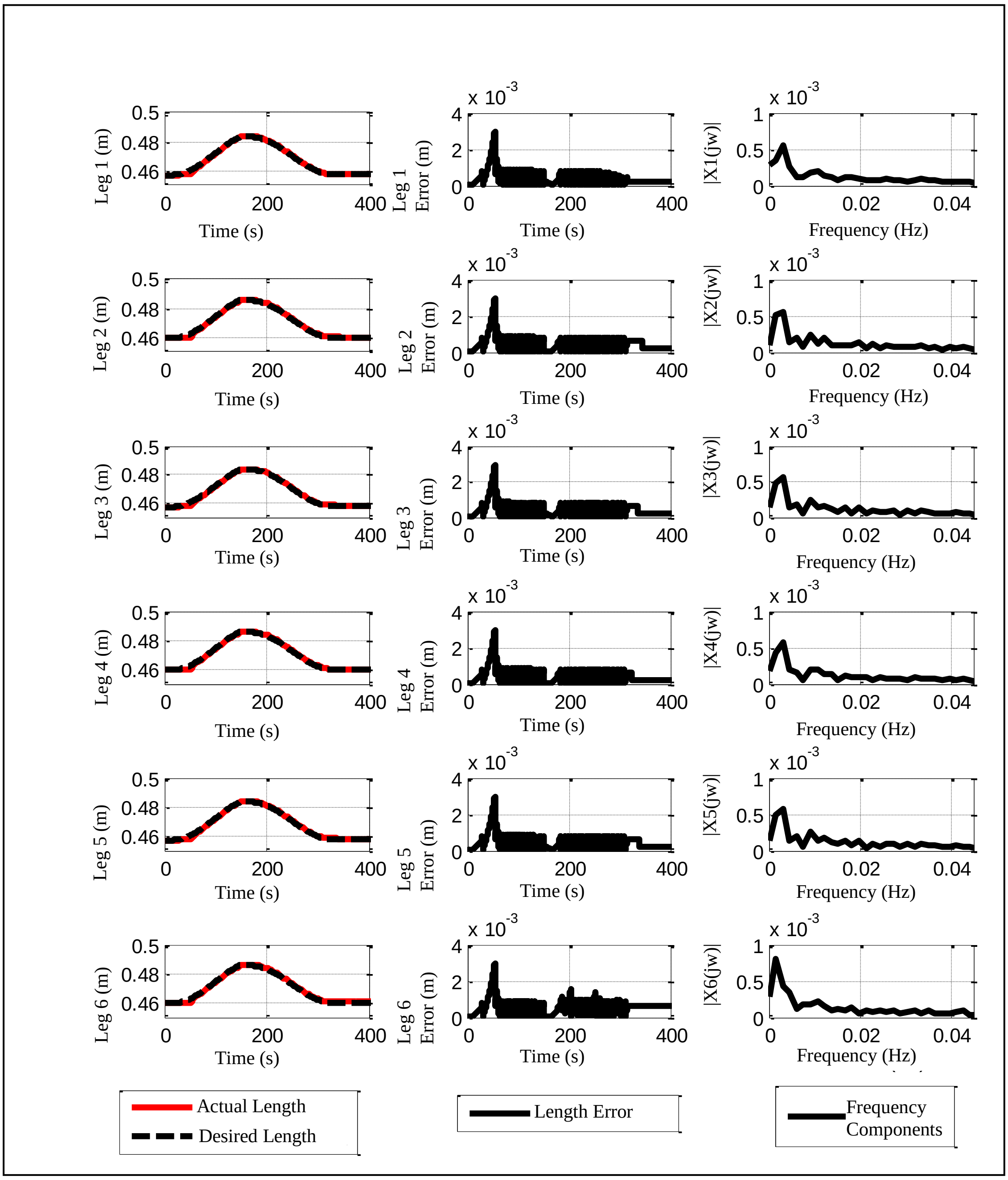


Figure 4.20. OrthoRoby’s leg length changes during cadaver bone cutting procedure

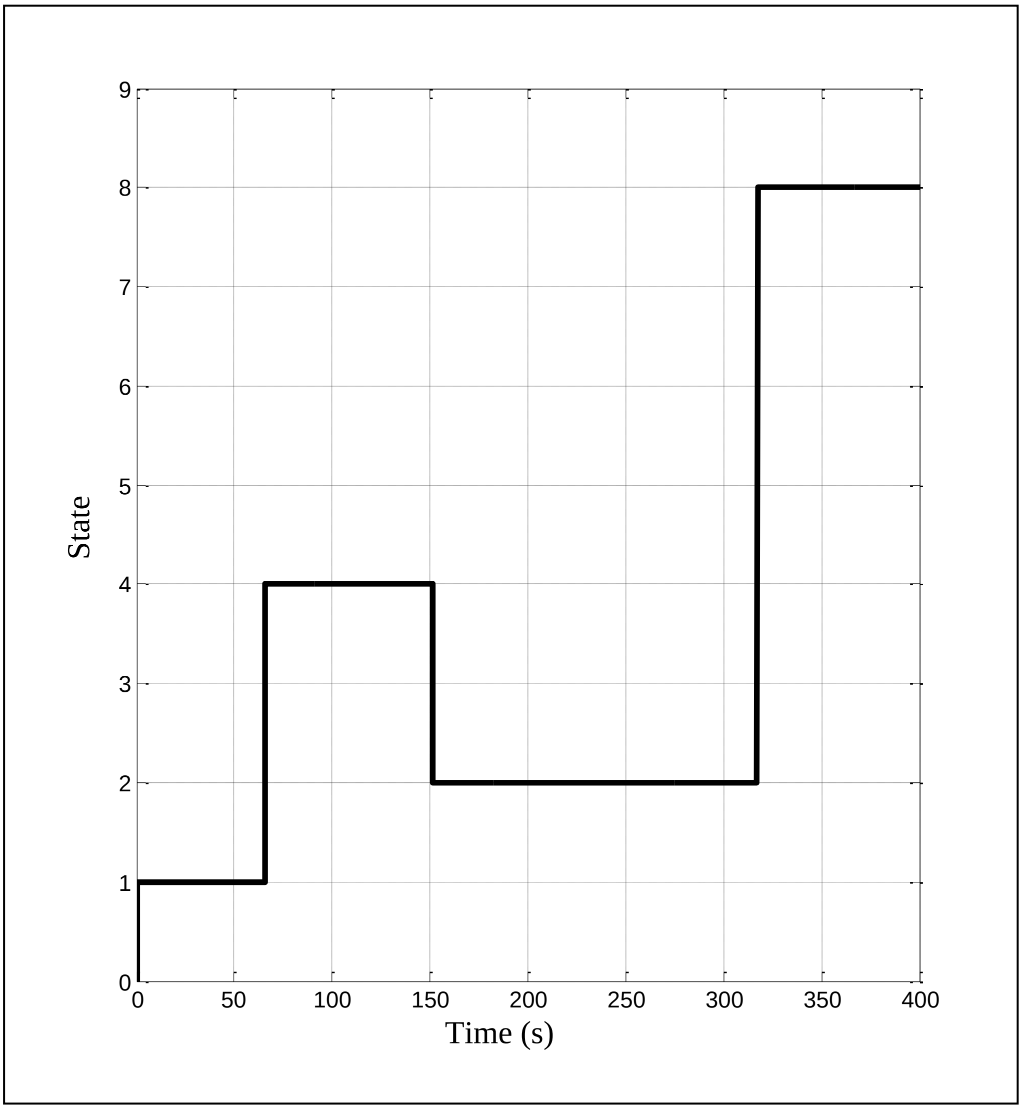


Figure 4.21. High-level controller's state changes while cutting cadaver bone (S_{1f}:*1*, S_{1b}:*2*, S_2:*4*, S_3:*8*)

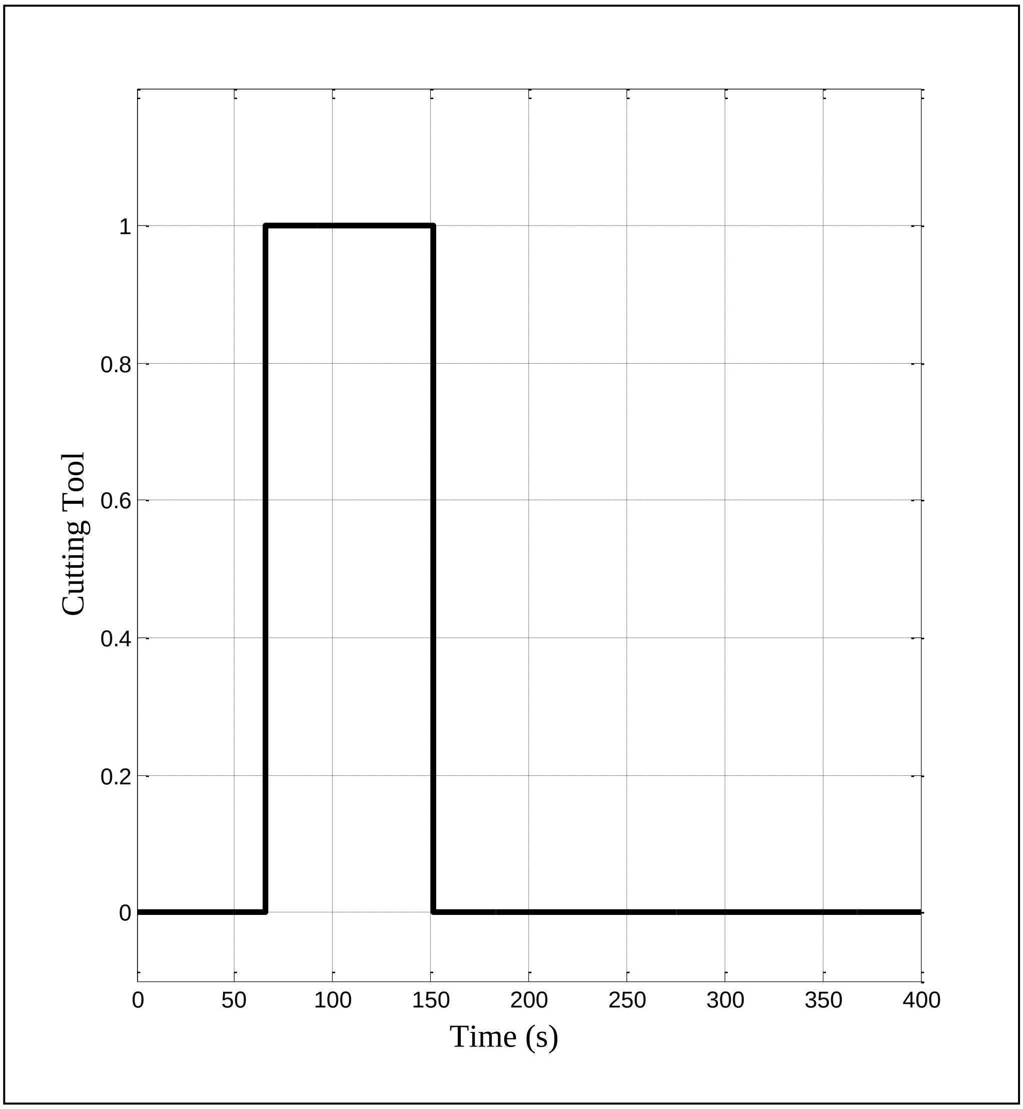


Figure 4.22. Cutting tool’s state while cutting cadaver (On:1, Off:0)

5. DISCUSSION AND CONCLUSION

In this thesis, an orthopedic robot called OrthoRoby, which is planned to be used in bone-cutting operations, is developed. OrthoRoby consists of a parallel robot and a cutting tool. A Cartesian system has been integrated into OrthoRoby to hold the parallel robot and to place the cutting tool of OrthoRoby close to the bone that will be cut. An intelligent control architecture has been developed that systematically combines a high-level controller with low-level controllers of the OrthoRoby to enable bone-cutting operations in a safe and desired manner. Intelligent control architecture has been integrated with a user interface to generate desired bone cutting trajectories based on patient's 3D bone model.

The intelligent control mechanism developed in this thesis can supervise the parallel robot, the cutting tool and the cartesian system to produce necessary coordinated motion to complete a given cutting bone operation in a desired manner. To our knowledge, such an intelligent control architecture has not been explored for orthopedic surgical robotic systems. Additionally, the proposed intelligent control architecture can monitor the progress and the safety of the cutting operation such that necessary dynamic modifications of the operation can be made (if needed) to complete the given operation in a safe manner. Medical robots are safety-critical systems, and safety should be considered from the very beginning of the design process and during the surgical operation.

User interface is used by the surgeon to define the required cutting trajectories. It can be observed that patient-specific preoperation planning through creation of 3D virtual environments using user interface is possible. It is also noticed that surgeon can easily define and modify the cutting trajectory using this user interface.

For bone cutting surgical robots that utilize structural rigidity and an automated decision mechanism are major issues. The cadaver bone cutting experiments of the OrthoRoby and its intelligent control architecture yielded promising results about the structural rigidity by its design, and automated decision by its intelligent control mechanism.

Note that in free space, when there is no contact with the bone, a low-level controller that satisfies motion tracking and disturbance rejection is sufficient. However, the interaction dynamics of the controlled system also depend on the dynamics of the bone. It was observed that computed-torque controller can become unstable with inexact cancellation due to parameter uncertainties and unmodeled dynamics.

As a future work, other control methods will be used to ensure OrthoRoby’s stability. Functionality of user interface will be extended adding new functions. Note also that this is a feasibility study for the proposed orthopedic surgical robotic system OrthoRoby to be used in the future for orthopedic surgery.

REFERENCES

1. Schulz, A. P., S., Klaus, C., Queitsch, A. V., Haugwitz, J., Meiners, B., Kienast, M., Tarabolsi, M, Kammal and C., Jürgens, “Results of Total Hip Replacement Using the Robodoc Surgical Assistant System: Clinical Outcome and Evaluation of Complications for 97 Procedures”, *The International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 3, No. 4, pp. 301-306, 2007.

2. Jakopec, M., F. R., Baena, S. J., Harris, P., Gomes, J. Cobb and B. L., Davies, “The Hands-On Orthopaedic Robot “Acrobot”: Early Clinical Trials of Total Knee Replacement Surgery”, *IEEE Trans. on Robotics and Automation*, Vol. 19, No. 5, pp. 902-911, 2003.

3. McEwen, J. A., C. R., Bussani, G. F., Auchinleck and M. J., Breault, “Development and initial clinical evaluation of pre-robotic and robotic retraction systems for surgery”, *Proceedings of the Annual International Conference of the IEEE Engineering in Engineering in Medicine and Biology Society*, Vol.3, pp. 881-882, 1989.

4. Davies, B., “A Review of Robotics in Surgery”, *Proceedings of the Institution of Mechanical Engineers, Journal of Engineering in Medicine* Vol. 214, pp. 129–140, 2000.

5. Shoham, M., M., Burman, E., Zehavi, L., Joskowicz, E., Batkilin and Y., Kunicher, “Bone-mounted Miniature Robot for Surgical Procedures: Concept and Clinical Applications”, *IEEE Transactions on Robotics and Automation*, Vol. 19, pp. 893-901, 2003.

6. Pechlivanis, I., G., Kiriyanthan, M., Engelhardt, M., Scholz, S., Lucke, A., Harders and K., Schmieder, “Percutaneous Placement of Pedicle Screws in the Lumbar Spine Using a Bone Mounted Miniature Robotic System, First Experiences and Accuracy of Screw Placement”, *Spine Journal*, Vol. 34, No. 4, pp. 392–398, 2009.

7. Wolf, A., B., Jaramaz, B., Lisien and A. M., Digioia, "MBARS: Mini Bone-attached Robotic System for Joint Arthroplasty", *International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 1, No.2, pp.101-121, 2005.

8. Jaramaz, B., M. A., Hafez and A. M., Digioia, "Computer-Assisted Orthopaedic Surgery", *Proceedings of the IEEE,* Vol. 94, No. 9, pp.1689-1695, 2006.

9. Guven Y. and D. E., Barkana, "Bone Cutting Trajectory Generation using a Medical User Interface of an Orthopedical Surgical Robotic System", *Proceedings of the Third International Conference on Human System Interaction*, Rzesow, 13-15 May 2010, pp. 325-330, IEEE, New York, 2010.

10. Guven Y. and D. E., Barkana, "Medical User Interface for Orthopedic Surgery Robotic System", *Proceedings of the Fifteenth National Biomedical Engineering Meeting*, Antalya, 21-24 April 2010, pp. 1-5, IEEE, New York, 2010.

11. Guven Y. and D. E., Barkana, "Control Architecture for an Orthopedic Surgical Robotic System OrthoRoby", *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, 20-22 October 2010, pp. 334-339, 2010.

12. Song, S., A., Mor and B., Jaramaz, "HyBAR: Hybrid Bone-attached Robot for Joint Arthroplasty", *International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 5, No. 2, pp. 223-231, 2009.

13. Brandt G., A., Simolong, L., Carrat, P., Merloz, H., Staudte, S., Lavallee, K., Radermacher and G. Rau, "CRIGOS: A Compact Robot for Image-Guided Orthopaedic Surgery", *IEEE Transactions on Information Technology in Biomedicine,* Vol. 3, No. 4, pp. 252-260, 1999.

14. Kwon, D. S., J. J., Lee, Y. S., Yoon, Y. S., Ko, J., Kim, J. H., Chung, C. H., Won and J. H., Kim, "The Mechanism and the Registration Method of a Surgical Robot for Hip Arthroplasty", *IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1889-1894, 2002.

15. Wapler, M., V., Urban, T., Weisener, J., Stallkamp, M., Dürr, A., Hiller, "A Stewart Platform for Precision Surgery", *Transactions of the Institute of Measurement and Control*, Vol. 25, No. 4, pp. 329-334, 2003.

16. Tsai, T. C. and Y. L., Hsu, "Development of a Parallel Surgical Robot with Automatic Bone Drilling Carriage for Stereotactic Neurosurgery", *IEEE International Conference on Systems, Man and Cybernetics*, 10-13 October 2004 Vol. 3, pp. 2156-2161, 2004.

17. Merlet, J. P., *Les Robots Paralleles, Hermes*, Paris, 1997.

18. Plaskos, C., P., Cinquin, S., Lavallée and A. J., Hodgson, "Praxiteles: Miniature Bone-mounted Robot for Minimal Access Total Knee arthroplasty", *The International Journal of Medical Robotics and Computer Assisted Surgery*, Vol.1, No. 4, pp. 67 – 79, 2005.

19. Barkana, D. E., "Design and Implementation of a Control Architecture for a Robot-Assisted Orthopedic Surgery", *The International Journal of Medical Robotics and Computer Assisted Surgery*, Vol. 6, No. 1, pp. 42-56, 2010.

20. Barkana D. E., "Evaluation of Low-Level Controllers for an Orthopedic Surgery Robotic System", *IEEE Transactions on Information Technology in Biomedicine*, Vol.14, No.4, pp. 1128-1135, 2010.

21. Antsaklis, P. J. and X. D., Koutsoukos, "Hybrid systems: Review and recent progress", in T. S. A. G. Balas, *Software-Enabled Control: Information Technologies for Dynamical Systems*, Ed. Piscataway, NJ: IEEE Press, pp. 1–29, 2003.

22. Sezgin, M., and B., Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging 13*, Vol. 146, 2004.

23. Hui, L., Z., Chenghui and Z. Caiming, “Visualization and Surface Rendering Based on Medical Image”, *Computer-Aided Design and Applications*, Vol. 9, No.1, pp. 79-86, 2012.

24. Middleton, G. V., *Data Analysis in the Earth Sciences Using MATLAB*, Prentice Hall, 1999.

25. Persson, P. and G., Strang, “A Simple Mesh Generator in Matlab”, *SIAM Review*, Vol. 46, No.2, pp. 329-345, 2004.

26. Morris, T., *Computer Vision and Image Processing*, Palgrave Macmillan., 2004.

27. Muthukrishnan, R. R., and M. M., Radha, “Edge Detection Techniques For Image Segmentation”, *International Journal Of Computer Science and Information Technology*, Vol. 3, No.6, pp. 259-267, 2011.

28. Mitra, S. K., *Digital Signal Processing: A Computer-Based Approach, 2e with DSP Laboratory using MATLAB*, McGraw-Hill, 2001.