

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DEVELOPING SUITABLE ALGORITHMS FOR
TELEOPERATION BETWEEN ROBOTIC MECHANISMS
HAVING DIFFERENT DYNAMICAL AND KINEMATIC
PROPERTIES**

AHMET SOYYİĞİT
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING**

GEBZE
2019

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DEVELOPING SUITABLE ALGORITHMS
FOR TELEOPERATION BETWEEN
ROBOTIC MECHANISMS HAVING
DIFFERENT DYNAMICAL AND
KINEMATIC PROPERTIES**

AHMET SOYYIĞIT
**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING**

**THESIS SUPERVISOR
PROF. ERKAN ZERGEROĞLU**

**GEBZE
2019**

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FARKLI KİNEMATİK VE DİNAMİK
ÖZELLİKLERE SAHİP ROBOTLAR ARASI
TELEOPERASYON İÇİN ELVERİŞLİ
ALGORİTMALAR TASARLANMASI

AHMET SOYYİĞİT
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
PROF. DR. ERKAN ZERGEROĞLU

GEBZE

2019

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 12/06/2019 tarih ve 2019/26 sayılı kararıyla oluşturulan jüri tarafından 20/06/2019 tarihinde tez savunma sınavı yapılan Ahmet SOYYİĞİT'in tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Prof. Dr. Erkan ZERGEROĞLU



ÜYE

: Dr. Öğretim Üyesi Yakup GENÇ



ÜYE

: Dr. Öğretim Üyesi Janset DAŞDEMİR



ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

This work focuses on remote manipulation and teleoperation of a humanoid robotic system. Humanoid robots are well suited to perform tasks on environments that are designed for humans when compared to robotic manipulators mounted on top of a moving platform. Especially when the robot has to work in an unstructured and/or dynamic environment as in the case of disaster response operations.

In this thesis, we propose a real-time haptic teleoperation system, designed to operate a compact humanoid robot with multiple modes. On the operator side, the operator only uses a desktop personal computer and 5 DOF haptic device in the shape of the wand. The operator doesn't need to wear anything uncomfortable during operation. The simplicity of the system enables the operator to further impose his dexterity on the remote robotic system. The main component of the remote side of the system is a relatively small sized humanoid robot (Nao from Softbank Robotics). Although the humanoid remote robot has limited capabilities compared to an actual sized human, we were able to perform basic remote operations with ease such as walking and object manipulation. Depending on the active teleoperation mode and the state of the robot, the remote system sends force feedback signals to the host system which are then applied to the operator through the wand. The main contribution of our study compared to the current state of the art in the literature is providing a multimodal haptic teleoperating system for operating humanoid robots with ease.

Key Words: Robot, Humanoid, Real-time, Haptic feedback, Multimodal.

ÖZET

Bu çalışmada, bir insansı robotik sistemin uzaktan yönetilmesi ve bu şekilde manipülasyon yapılması konularına odaklanılmıştır. İnsansı robotlar, insanlar için tasarlanmış ortamlarda görev yapmak için, üstüne robotik manipulator monte edilmiş hareketli platformlardan daha uygundur. Özellikle robotun yapılandırılmamış ve/veya dinamik ortamlarda çalışması gerektiği afete müdahale gibi durumlarda, bu uygunluk kendini iyice belli etmektedir.

Tez kapsamında, kompakt bir insansı robotun farklı biçimlerde gerçek zamanlı ve dokunsal geri bildirimli olarak uzaktan yönetilmesini sağlayan bir sistem önerilmiştir. Sistemin operatör tarafı sadece masaüstü bilgisayar ve ona bağlanmış çubuk şeklindeki 5 serbestlik dereceli dokunsal geri bildirimli cihazdan oluşmaktadır. Önerilen sistem ile, operatörün herhangi bir giyilebilir ve/veya konforlu olmayan cihazı kullanmaya ihtiyaç duymadan önerilen sistemi rahatça kullanabilmesi, onun uzaktaki robotik sistemi daha hünerli yönetebilmesini sağlamıştır. Uzaktaki sistemin ana bileşeni nisbeten küçük boyutlu bir insansı robottur (Softbank Robotics tarafından üretilmiş Nao). Uzaktan yönetilen insansı robotun kabiliyetlerinin, aslolan büyüklükteki bir insana kıyasla sınırlı olmasına rağmen, basit seviyedeki yarı-otonom operasyonlar rahatlıkla yapılabilmektedir. Çalışma esnasında aktif olan uzaktan yönetim biçimi ve robotin o anki durumuna göre gelen dokunsal geri bildirim sinyalleri operatör tarafında işlenerek kullanıcıya çubuk üzerinden uygulanmıştır. Literatürdeki son çalışmalara kıyasla bizim çalışmamızın katkısı, insansı robotların dokunsal geri bildirimli ve çok biçimli olarak uzaktan yönetilmesinin rahatça gerçekleştirilmesini sağlayan bir sistem sağlamasıdır.

Anahtar Kelimeler: Robot, İnsansı, Gerçek-zamanlı, Dokunsal geri bildirim, Çok biçimli.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Prof. Erkan Zergeroğlu for his guidance, advice, and support through my graduate and undergraduate studies. He has taught me countless lessons and steered me in the right direction whenever I got lost.

Secondly, many thanks to the people who helped the experiments with their time and effort, Samet Gül, Enes Almahdi, Abdullah Bakırcı, Nur Şadan Soyyiğit and Muhammed Ali Dede. Also, Uygur Akalın, who has developed the GUI part of the camera feedback panel of the system during his internship.

Lastly, I deeply thank my family for their patience, encouragement, and trust. Their support has been extremely valuable throughout my life and made this thesis study possible.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	x
LIST of FIGURES	xi
LIST of TABLES	xiii
1. INTRODUCTION	1
2. RELATED WORK	4
3. PROPOSED SYSTEM	7
3.1. Quanser 5 DOF Haptic Wand and Control	8
3.2. Humanoid Robot Nao	10
3.3. Zenom, Real-Time HIL and Simulation Environment	12
3.4. Integration and Calculations	13
3.4.1 Converting Zenom from Xenomai to PREEMPT-RT	14
3.4.2 Converting Haptic Wand Drivers and Libraries	14
3.4.3 Adding Camera Feedback to Zenom GUI	14
3.4.4 Implementing the Main Teleoperation Program	15
3.4.5 Kinematics and Manipulability Calculations	23
4. DEMONSTRATIONS AND DISCUSSIONS	28
4.1. Validating Kinematic Arm Models by Testing	28
4.2. Testing “Head” Mode of The System	30
4.3. Testing “Right Arm” Mode of The System	31
4.4. Testing “Left Arm” Mode of The System	39
4.5. Testing “Both Arms” Mode of The System	45
4.6. Testing “Walk Toward” Mode of The System	52
4.7. Overall Test of The System for a Pick and Place Task	54
5. CONCLUSION	58

REFERENCES	60
BIOGRAPHY	63

LIST of ABBREVIATIONS and ACRONYMS

<u>Abbreviations</u>	<u>Explanations</u>
<u>and Acronyms</u>	
OPC	: The Operator Computer
ROS	: Robotic Operating System
DH	: Denavit-Hartenberg
RTB	: Robotics Toolbox
GPS	: Global Positioning System
IMU	: Inertial Measurement Unit
GUI	: Graphical User Interface
PCI	: Peripheral Component Interconnect
SDK	: Software Development Kit
HIL	: Hardware in the Loop
POSIX	: Portable Operating System Interface for Unix
RTDM	: Real-Time Driver Model

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
1.1: Some examples of telerobotics applications, a) Underwater robot, b) Surgery robot, c) Manipulator, d) Rover robot.	2
3.1: Overall teleoperation system schema.	7
3.2: The Quanser 5 DOF Haptic Wand used in our teleoperation system, a) Released state, b) Holding the wand.	8
3.3: Quanser 5 DOF Haptic Wand movement axes. Rotating around the Z-axis (q_1) is not sensible by encoders.	9
3.4: Nao humanoid robot V5 H 25 that we used in our demonstrations.	10
3.5: Nao H25 joints and sensors.	11
3.6: Zenom graphical interface after executing a project.	13
3.7: Camera feedback dialog of Zenom. Up and Bottom buttons are used to select the camera.	15
3.8: Nao Control Panel of Zenom.	16
3.9: FRAME_TORSO of Nao.	18
3.10: Head joints of Nao.	19
3.11: Some poses with their target positions during “Both arms” mode	21
3.12: Bumpers of NAO’s left foot.	22
3.13: Assigned motion axes for NAO’s left arm.	24
3.14: Assigned motion axes for NAO’s right arm.	25
4.1: Deviation of calculated and actual poses of the right arm.	29
4.2: Deviation of calculated and actual poses of the left arm.	30
4.3: The angles of the head and the Wand during the first demonstration.	31
4.4: The angles of the head and the Wand during the second demonstration.	31
4.5: Position of the right hand and the Wand during the first demonstration.	32
4.6: Position of the right hand and the Wand during the second demonstration.	33

4.7:	Orientation of the right hand and the Wand during the first demonstration.	34
4.8:	Orientation of the right hand and the Wand during the second.	35
4.9:	Manipulability – force data during “Right arm” mode.	37
4.10:	Manipulability ellipsoids of the right arm.	38
4.11:	Position of the left hand and the Wand during the first demonstration.	39
4.12:	Position of the left hand and the Wand during the second demonstration.	40
4.13:	Orientation of the left hand and the Wand during the first demonstration.	41
4.14:	Orientation of the left hand and the Wand during the second demonstration.	42
4.15:	Manipulability – force data during “Left arm” mode.	43
4.16:	Manipulability ellipsoids of the left arm.	44
4.17:	Position of both hands and the Wand during the first demonstration.	46
4.18:	Position of both hands and the Wand during the second demonstration.	47
4.19:	Orientation of both hands and the Wand during the first demonstration.	48
4.20:	Orientation of both hands and the Wand during the second demonstration.	49
4.21:	Manipulability – force data of the system during “Both arms” mode.	51
4.22:	Position - force data of the system during the first demonstration while the active mode is “Walk toward”.	52
4.23:	Position - force data of the system during the second demonstration while the active mode is “Walk toward”.	53
4.24:	The experiment setup.	54
4.25:	Target inclined hole of the experiment.	55

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
3.1: Quanser 5 DOF Haptic Wand workspace boundaries.	9
3.2: Standard Denavit-Hartenberg parameters of NAO's left arm.	24
3.3: Standard Denavit-Hartenberg parameters of NAO's right arm.	25
4.1: General information about subjects.	56
4.2: Time of completion for each step of the task.	56
4.3: Attempts and failures of the subjects.	57
4.4: Comments of the subjects after experiments.	57

1. INTRODUCTION

In this study, we propose a teleoperation system, specially designed to operate a compact humanoid robot with multiple modes. On the operator side, the operator only uses a desktop mounted 5 DOF haptic device in the shape of a wand. The ease of using a simple system without needing to wear any additional device enables the user to further impose his dexterity on the remote robotic system.

On the remote system, the main component is a relatively small sized humanoid robot (Nao from Softbank Robotics). Though the humanoid remote robot has limited capabilities compared to an actual size human we were able to perform basic remote semi-autonomous operations with ease. Depending on the active teleoperation mode and the state of the robot, the remote system imposes force feedback signals to the host system which are then applied to the operator through the wand.

The main contribution of our study compared to the current state of the art in the literature is providing a multimodal haptic teleoperation system for operating humanoid robots with ease.

As in the case of Chernobyl nuclear plant disaster in 1986, where at least three men have sacrificed their lives in order to prevent an explosion by diving underwater and releasing a critical pressure valve by being exposed to a high dose of radiation [1], the use of a teleoperated robotic device might be saved lives. Though recent advancements on robotics made it possible to build robots that can walk, fly and even dive underwater, in an unstructured environment build specifically for humans, the operation of a robotic device with full autonomy, to our best knowledge, is still in its early research stage. The use of remote operation of a robotic device, on the other hand, enables the human operator to perform the task from a safe distance while still using his/her dexterity through the robotic device.

Teleoperation, or remote control, is the operation of a machine or a system from a distance via the inputs generated from a human operator. The field of teleoperation has many applications like controlling a crane or an unmanned air vehicle. In order to enable the remote user to get inside of the remote environment *telepresence* is used. An example of telepresence is speaking to somebody over a video chat by the internet. The combination of telepresence and teleoperation, in robotic literature is referred as *telerobotics*.

interacting directly with physical objects, plays an important role in telerobotic applications.

The teleoperation interface can also be a robotic system having various degrees of freedom. In these systems, the feedback is usually imposed as a force to the end effector of the device. This force is imposed by using electrical motors of the device which gives the sensation of the surface stiffness, may warn the user about something, or assist the operator during teleoperation.

In this thesis study, we present a telerobotic system that allows its users to accomplish manipulation and roaming tasks in an indoor environment with ease. The system consists of a 5 DOF haptic interface (the Haptic Wand) from Quanser Inc. [7] as the operator interface and a compact humanoid robot [8] as the remote manipulator.

The proposed system is easy to use and comfortable as the operator does not have to wear any uncomfortable equipment. The teleoperation system ensures the user to make different motions (modes of teleoperation) specifically (i) moving the head, (ii) moving the left or right arm, (iii) moving both arms simultaneously to grab and turn a valve or a wheel and (iv) walking. The force feedback feature of the haptic device is used in different ways, which are: to facilitate pausing and mode switching, to warn the user when the robot is close being singular and to inform the user about the obstacle that blocks the walking path.

The rest of the thesis is organized as follows: the related work about haptic teleoperation of humanoid robots are presented in the second chapter. The third chapter contains the details of the design and implementation of the proposed system. Discussions on the performance and drawbacks of the proposed system illustrated via demonstrations are presented in the chapter. Finally, in the last chapter, we deliver our conclusions on this thesis with our implications and discussing what can be done in the future.

2. RELATED WORK

A humanoid robot can be teleoperated with different techniques and devices. The user may use a haptic interface, a joystick, a master arm, or just mouse and keyboard of the computer. The sensors of the teleoperated robot may vary depending on the budget. In some studies, extra hardware is mounted on the robot to increase perception. An example is the study of Zhang et al. [9]. To improve the poor vision feedback, they implemented a virtual scene of the environment including the robot. Their system constructs the scene and displays it in real-time by processing motion data of the robot. The motion capture system with infrared and normal cameras watches the robot in order to generate the motion data. And the operator uses a joystick and a master arm to operate the robot while watching the virtual scene in the computer display.

A popular technique for teleoperation is capturing motions of the whole human body and commanding the humanoid robot to imitate these motions. The motions can be captured with devices that may require to be worn by the operator or not. Rodriguez et al. [10] developed two different ROS packages for teleoperating humanoid robot Nao. These are speech-based and gesture-based teleoperation packages. The second one captures human motions by Microsoft Kinect [11] and generates these commands: stand up, crouch, stop, move forward/backward/left/right and turn left/right. The same package can also be used to teleoperate arms but because OpenNI (library for processing Kinect data) cannot detect yaw of the operator's arm, this operation is limited with other axes. Almetwally et al. [12] introduced a technique to fully imitate motions of the human body in real time by using the Microsoft Kinect Depth Camera. The advantage of using Microsoft Kinect is the operator doesn't need to wear or hold anything for teleoperation. The robot they used was also a Nao with ROS, and their main contribution was fully imitating the human walking motion. Koenemann et al. [13] presented a system that ensures a Nao humanoid robot to imitate the captured hand, foot, hip and shoulder positions and center of mass of the operator. The positions are captured in real-time with a device named Xsens MVN [14] by attaching it to the human body. To reach the target poses, they apply inverse kinematics to find the joint angles for all for kinematic chains of Nao (arms and legs) and modify the joint angles to ensure stability by using the center of mass of the human. With this method, they

managed to teleoperate the robot even in single support mode, which was their main contribution.

Now, let's mention the related work about haptic teleoperation of humanoid robots in chronological order. Peer et al. [15] built a system for teleoperating a humanoid robot, HRP-2 [16], using a 7 DOF haptic interface. Their goal was to collaborate with a human on the remote side using a humanoid robot for lifting and transporting an object together. The teleoperation distance was intercontinental which arose stability problems. They managed to design a control architecture with force-position exchange accounting for the admittance type of the haptic input and telerobot, which both are position-based admittance controlled. They continued their work in [17] by extending the former single-arm bilateral teleoperation setting to include bimanual manipulation and walking. This was done by making the haptic interface mobile and introducing a new task-based control framework. They managed to wide the area of motions with the new system.

Renon et al. [18] investigated the results of teleoperating a virtual iCub [19] humanoid robot with haptic feedback. They used a 3 DOF joystick with force feedback feature to sense the touching of the virtual robot when it touches a virtual object. To be able to calculate target arm angles of the robot with inverse kinematics, they calculated DH parameters and created the kinematic arm models using MATLAB RTB [20], just like we did in our thesis.

Ju et al. [21] telemanipulated a 7 DOF arm of a Baxter [22] humanoid with a 6 DOF SensAble Omni [23] haptic device in their proposed system. They implemented two different methods on their system, directly mapping angles of the haptic device and closed-loop inverse kinematic. Force feedback was used to sense the motion of telemanipulator arm at the operator side and to assist position-position mode.

Barros et al. [24] proposed a system for teaching how to walk to a humanoid robot by using teleoperation. They used two 6 DOF SensAble Omni haptic devices to represent legs of the robot. Hence, the operator needs to use both of his/her hands during teleoperation. Instead of using a real one, they simulated the lower half of the robot body in a virtual platform called V-REP [25] and used force feedback to maintain balance during teleoperation.

Fritsche et al. [26] presented an intuitive and affordable teleoperation system by using Microsoft Kinect, SensorGlove and Oculus Rift [27]. With their system, the operator can telemanipulate the iCub humanoid with the first-person view, in other

words, seeing through the eyes of the robot. Haptic feedback is applied to the operator through SensorGlove by measuring pressure made to tactile sensors on the hands of the robot. The robot could mimic the whole-body motions of the operator by measuring with Microsoft Kinect. They tested their system on a pick-and-place task and stated their system can be used with ease.

Lofaro et al. [28] built three different apparatus for remote control of three different humanoid robots. These apparatuses were kinematically scaled systems of the corresponding humanoid robot. Therefore, the operator doesn't need to wear suits which makes their system flexible and easy to use. The system imposes haptic feedback to extend the immersion to tactile spaces. Their motivation to build these apparatuses was to provide a teleoperation system that ensures minimally trained individuals to control complex humanoid robots to accomplish tasks that require dexterity.

Graziano et al. [29] presented a teleoperation system with a wearable suit for upper body motion tracking designed by them. The suit has IMU's and processors that calculate and send pose of the wearing person over Wi-Fi. Also, the operator holds haptic devices with his/her hands that exert force feedback over the index-thumb fingers to sense gripping. To test their system, they telemanipulated a Baxter humanoid robot to perform pick and place operations.

Panzirsch et al. [30] investigated the performance of the human operators when teleoperating a complex humanoid robot using an underactuated haptic device for telehealthcare applications. They explored the effects of two different force feedback architectures that produce force according to the environment, one using sensor-based force information while other is not. Also, they investigated the effects of different workspace scaling factors. These were all tested with twelve participants.

Abi-Farraj et al. [31] proposed a new method for imposing haptic feedback that gives cues about the stability of the torque-controlled humanoid robot during telemanipulation with arms. With these cues, the operator is directed through successful task completion while accounting for the different constraints of the system. Overall, the designed task-relevant haptic interface bridges the gap between the task at the hand and the balance of the robot.

Our contribution to the literature is haptic teleoperating a humanoid robot to accomplish multiple types of tasks with using just a single haptic device similar to a joystick, therefore, easy to use.

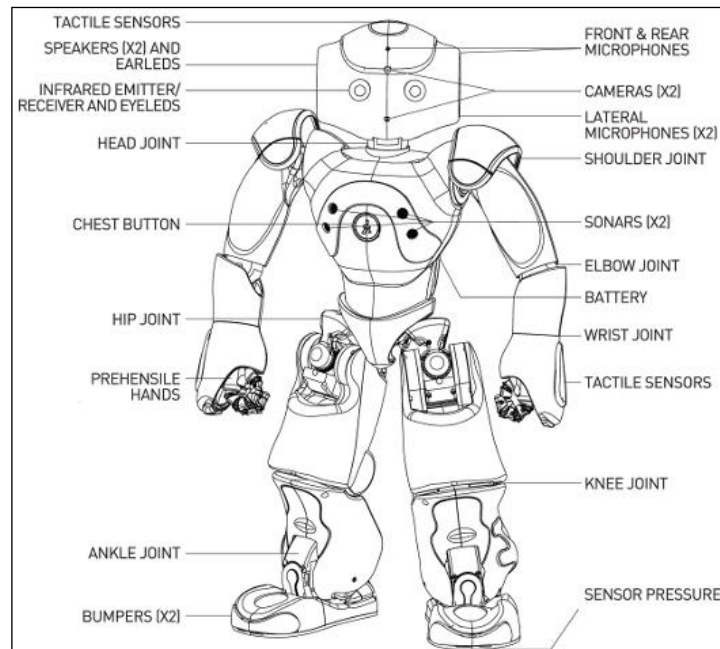


Figure 3.5: Nao H25 joints and sensors.

Nao has a framework named NAOqi that has a mediator process running on the robot for controlling the robot from outside or inside. We reach this process by programming with the NAOqi SDK. The SDK can be used to implement modules for extending robot capabilities or to access available core modules. Some of the modules that the robot has built-in are motion, vision and memory modules. For example, to move Nao's joints from our computer programmatically, we need to send a message to the motion module over the NAOqi process which is done by calling appropriate SDK functions inside our program. Communication with all other modules is established in the same way over the network. Many of the SDK functions have alternatives for blocking and non-blocking operation. If called from a remote computer, these functions involve network communication which may cause unpredictable delays and shouldn't be called in a real-time task. To overcome this problem, we found a solution that will be explained in the integration section.

NAOqi SDK has support for many programming languages, but it has full support for C++ and Python. If C++ or Python SDK is used, implementation of programs that can run inside the robot becomes possible. Also, the SDK for C++ and Python has the greatest coverage of module libraries and the documentation has examples only written in these languages.

3.4.1. Converting Zenom from Xenomai to PREEMPT-RT

The real-time subsystem of the Zenom was implemented using Xenomai as it can deliver hard-real-time. This subsystem includes the periodic task and inter-process communication mechanisms that were wrapped as C++ classes. Using Xenomai to deliver real-time has some drawbacks, however. First, to compile and run the Zenom and Zenom projects, Xenomai must be installed on the system which is much harder than installing PREEMPT-RT and has less support. Second, inside the program, Xenomai system calls must be used to achieve real-time, therefore making programming harder. And third, a separate RTDM driver needed to be written for every external device that will be used inside the real-time program if it is needed to access that device with high frequencies. But in a PREEMPT-RT patched system, no additional libraries or drivers are needed to deliver real-time. Just using POSIX threads and giving real-time priority to them are enough. Although the PREEMPT-RT doesn't guarantee latencies as low as Xenomai, it is fair enough for many control applications including our teleoperation system. For these reasons, we converted the real-time subsystem of the Zenom to use POSIX system calls, and now the Zenom can be compiled and run on all Linux distributions, even if it doesn't have PREEMPT-RT patch.

3.4.2. Converting Haptic Wand Drivers and Libraries

As we converted the real-time subsystem of the Zenom, we did the same for Q8 RTDM driver to make it usable. We also changed the userspace library of the haptic wand to use POSIX system calls instead of Xenomai calls. This way the library and the driver can be used without Xenomai, in all Linux distributions. In our tests, control applications that use Q8 HIL card executed smoothly on the new PREEMPT-RT patched Linux system up to 1 kHz frequency while the system is loaded.

3.4.3. Adding Camera Feedback to Zenom GUI

The operator needs to see from the eyes of the robot as he/she commands it from a distance. For this reason, we added a new Qt dialog to Zenom that receives and shows camera images from the network. This dialog can be opened by selecting the view

No force: While the red button on the Wand is pressed, there won't be any force imposed to the Wand and the operator can move it freely as long as the conditions for imposing force feedback does not satisfy.

Stay where you left: When the operator releases the red button on the Wand, the system generates forces to hold it at the last position where the red button is pressed as long as the conditions for imposing feedback does not satisfy. This state is for the operator to relax, think and do other things if needed by his hand.

Force feedback: If the conditions for imposing force feedback become satisfied, the Wand will be pushed to the initial position. The purpose of this will be explained in the description of teleoperation modes.

In our code, a C++ function is used to generate force through a specified direction in 3D space. This direction is indicated by a vector that is given to the function as a parameter. The beginning point of this vector is the center of the Wand. The norm of the vector is commensurate with the power of the force. To move the wand from its current pose to any other pose, a trajectory is needed. We create this trajectory by using a linear interpolator. And to determine the force vector that will push the wand to next point generated by linear interpolator, we use a proportional-derivative type controller that calculates the force vector and give it to the force generator function. The force generated by the controller is diminished at the beginning of force feedback and increased in time to normal because applying a high force at the beginning causes control problems. To summarize, we calculate the position of the Wand by forward kinematics, determine the next target point by linear interpolator, calculate the force vector to the next point the with controller and generate forces. These are all done in every period of the main program.

With our system, the operator can move the Nao six different ways, and these define the modes of the teleoperation. Only one of these modes can be active at any time. Prior to describing modes, we would like to define some transformation and poses that are used for calculations during teleoperation.

The kinematic calculations of arm motions are done by using FRAME_TORSO as the reference frame. We will call this frame T_{FT} . The definition of FRAME_TORSO in NAO documentation [8] is: "It is attached to the robot's torso reference, so moves with the robot as he walks and changes orientation as he leans. This space is useful when you have very local tasks, that makes sense in the orientation of the torso frame."

$$T_{TLH}(t) = \begin{bmatrix} 1 & 0 & 0 & x/2 \\ 0 & \cos(w_x) & -\sin(w_x) & y/2 \\ 0 & \sin(w_x) & \cos(w_x) & z/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_{ILH} \quad (3.2)$$

Both Arms: In this mode, both hands are moved simultaneously. They translate by the same amount and rotate around the x-axis of the transformation in the middle of the hands at that time. The rotation movement is very similar to turning an industrial valve or a steering wheel. The inverse of z axes of target transformations points to the middle transformation after calculation. Target transformations of the hands are calculated by using $P_{DW}(t)$ and target distance between hands, $2d$. To calculate the target transformations, the middle transformation that hands will rotate around it must be calculated for each pose. Assuming p_{IMT} has the components $[x_1, y_1, z_1]$ and $P_{DW}(t)$ is consisting of $[x_2, y_2, z_2, w_x, w_y]$, The middle transformation is defined as:

$$T_M(t) = \begin{bmatrix} 1 & 0 & 0 & x_1 + x_2/2 \\ 0 & 1 & 0 & y_1 + y_2/2 \\ 0 & 0 & 1 & z_1 + z_2/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_{FT} \quad (3.3)$$

As the hands rotate around the x-axis of the $T_M(t)$, the target transformation of left and right hand is calculated as:

$$T_{LT}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-w_x - (\pi/2)) & -\sin(-w_x - (\pi/2)) & d * \cos w_x \\ 0 & \sin(-w_x - (\pi/2)) & \cos(-w_x - (\pi/2)) & -d * \sin w_x \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_M(t) \quad (3.4)$$

$$T_{RT}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-w_x + (\pi/2)) & -\sin(-w_x + (\pi/2)) & -d * \cos w_x \\ 0 & \sin(-w_x + (\pi/2)) & \cos(-w_x + (\pi/2)) & d * \sin w_x \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_M(t) \quad (3.5)$$

The hands are translated to their target positions by the amount of d to symmetric positions and rotated with respect to $[w_x]$ component of $P_{DW}(t)$. Also, the $[w_y]$ component makes Nao duck the same way, enabling to reach objects are much lower. Six different targets and executed poses of both arms are given in Figure 3.11 as an example. The green dots at the left and right of each pose represents the target position.

Stop: In this mode, the robot does not make any movement. This mode is used for pausing teleoperation.

While the kinematic calculations and the input/output is done periodically in the real-time task of our main program, we also need to communicate with the robot over the network to send commands and receive sensor information. This communication has an overhead that may block the caller function for an unpredictable time especially if wireless communication is used. To address this problem, we created a second task with a real-time priority that is responsible for network communication, called a network task. Therefore, the commands are sent and information is received from the robot in this task. When the periodic task wishes to communicate with the robot, it puts the pointer of a lambda function and its parameters in a queue guarded by a mutex. The lambda functions include function calls of the communication functions. If the mutex is locked, the periodic task doesn't wait for it to be released as it needs to finish other jobs (controlling the Wand) before the deadline and tries again in the next period. In other words, it try-locks the mutex. Otherwise, it won't be possible to control the Wand because of the delay of blocking mutex waiting call. If the mutex is released at that time, it acquires the mutex and add the lambda function pointers to the queue, releases the mutex, notifies the network task and continues executing. Then the network task wakes up, acquires the mutex, and calls the communication functions until the queue becomes empty. This sleep-wakeup mechanism is done by using a condition variable.

3.4.5. Kinematics and Manipulability calculations

To calculate the manipulability of an arm, the Jacobian matrix needs to be calculated first and NAOqi does not offer a function for this purpose. To address this problem, we used the Robotics Toolbox for MATLAB to calculate the Jacobian matrices of the arms. As the kinematic models of the arms are needed to calculate the Jacobian matrices, we calculated the standard Denavit-Hartenberg parameters of the arms and defined the arm models as serial links. We verified the correctness of our models by comparing transformation matrices generated by forward kinematics of our models and the `getTransform()` function from the NAOqi framework. They calculate the same transformation matrices for any given joint angles with a very low error for some cases.

$$T_{BaseLArm} = \begin{bmatrix} \cos(-\frac{\pi}{2}) & -\sin(-\frac{\pi}{2}) & 0 & 0 \\ \sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & 0 & SOY \\ 0 & 0 & 1 & SOZ \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(-\frac{\pi}{2}) & 0 & \sin(-\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\frac{\pi}{2}) & 0 & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$T_{BaseRArm} = \begin{bmatrix} \cos(-\frac{\pi}{2}) & -\sin(-\frac{\pi}{2}) & 0 & 0 \\ \sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & 0 & SOY \\ 0 & 0 & 1 & SOZ \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(-\frac{\pi}{2}) & 0 & \sin(-\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\frac{\pi}{2}) & 0 & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

After the base transformation is multiplied with the transformation matrix of the arm model, an extra rotation around the y axis is needed to reach the hand as tool transformation. The final, forward kinematics transformations of the arms are given below.

$$T_{RArm}(q) = \begin{bmatrix} \cos(-\frac{\pi}{2}) & 0 & \sin(-\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\frac{\pi}{2}) & 0 & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_{RArmDHModel}(q) * T_{BaseRArm} \quad (3.8)$$

$$T_{LArm}(q) = \begin{bmatrix} \cos(-\frac{\pi}{2}) & 0 & \sin(-\frac{\pi}{2}) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\frac{\pi}{2}) & 0 & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * T_{LArmDHModel}(q) * T_{BaseLArm} \quad (3.9)$$

After calculating kinematic models, we generated the C code of the Jacobian matrices of both arms by using RTB code generation feature. The generated C functions take arm angles and returns the Jacobian matrix. We integrated them into our main program and implemented further matrix calculations with the Eigen C++ library.

To calculate the Asada's manipulability in our main program periodically, we calculate the Jacobian of an arm with the generated functions. Then, we multiply the transpose of Jacobian with itself to acquire $A = J^T * J$. In the actual calculation, the Jacobian is multiplied with its inverse. The reason we did the reverse is that it gives the same eigenvalues except for the sixth one which is always zero and unmeaningful as the arm has 5 DOF. We find the five eigenvalues of the matrix A with MATLAB's

eig function and take the square root them. These five values are named $\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, \sqrt{\lambda_4}, \sqrt{\lambda_5}$ and correspond to the five-movement axes. However, we don't know which eigenvalue correspond t which axis because there isn't any order between them. Therefore, we measure the general manipulability of an arm with the formula below:

$$Manipulability = \frac{\max(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, \sqrt{\lambda_4}, \sqrt{\lambda_5})}{\min(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}, \sqrt{\lambda_4}, \sqrt{\lambda_5})} \quad (3.10)$$

When this value grows larger, that means the arm is much closer to becoming singular. If it becomes greater than a predefined value, the main program imposes feedback on the wand for a short duration less than a second to inform the operator that the robot is close to being singular, so the arms should stop going further in that direction. These were already explained before, but we wished to repeat.

4. DEMONSTRATIONS AND DISCUSSION

The operator computer that we used in our demonstrations has an Intel(R) Core (TM) i7-4790 processor and PREEMPT-RT patched Ubuntu 18.04 operating system with Linux kernel version 4.14.52-rt34. And, the model of Nao was V5 with H25 body type. The connection between OPC and Nao is done via a local ethernet network.

First, we tested the real-time behavior of the PREEMPT-RT converted Zenom, Q8 driver and haptic wand libraries on the operator computer. While the Zenom was executing a project in 1 KHz that moves the wand along a square wave trajectory in five axes, we ran a stress-ng [35] instance with parameters “-all 0” which runs various stress tests on all CPU’s of OPC. Execution continued smoothly for four minutes with five overruns. There is another short demonstration of this test in the video file named `zenom_preemptrt_test.mp4` inside this thesis that can be watched.

To show the results of our implementation, we made two different demonstrations. In the first demonstration, we logged the arm data only when the arms did their movement, in other words, not while the arms are moving. Our aim was to calculate the error between the actual and target arm poses. This is not feasible for teleoperation because it increases the delay. And, in the second demonstration, we executed our teleoperation system as it should be, and logged the data regardless of movement. The logged data from both demonstrations are plotted in this section.

The frequency of data logging was 5 Hz and all modes of our teleoperation system are tested for nearly thirty seconds in both experiments. All position and orientation values of the arms assume `FRAME_ROBOT` as basis frame (except manipulabilities, their basis frame depend on eigenvectors). The demonstrations can be watched from `teleop_test1.mp4` and `teleop_test2.mp4` video files.

4.1. Validating Kinematic Arm Models by Testing

In Figure 4.1, the difference between the calculated poses with our right arm kinematic model and the actual ones are illustrated. The poses were generated using the logged arm angles between seconds 45 and 75 of the first demonstration. The “Right arm” mode was being tested at that time. Actual poses and arm angles are received from the robot directly with `getTransform()` and `getAngles()` functions of `NAOqi`. Calculated poses were generated by passing the arm angles and our kinematic

we can see the force imposed on the wand when overall manipulability of the right arm becomes higher than the manipulability threshold. Please remind that the higher values of manipulability mean the arm is closer to becoming singular, and vice versa. After imposing a feedback force for a short duration, the system changes to cooldown state that can be seen in the feedback cooldown plot. During cooldown, the system does not impose feedback forces even if the manipulability becomes higher than the threshold. The cooldown is added to our system because of the delay between OPC and Nao.

Forces are also imposed when the button on the wand becomes released to hold the wand at the last position where the button was pressed. That type of forces can be seen at the beginning and the end of the last two plots of Figure 4.9. These plots show how the force is imposed on the wand according to manipulability measure while teleoperating right arm during the second experiment. When the system starts to impose forces, the target force is not immediately imposed with full power because doing that causes control problems and abnormal sounds to come from the motors.

In the second and third manipulability plots of Figure 4.9, the values define the manipulability of each axis individually over time. These values are the magnitude of semi-axes of manipulability ellipsoids that can be seen in Figure 4.10. The colors of the axes are cyan for X, green for Y and blue for Z. We calculated them by using the formula we learned from the implementation of `vellipse` function of RTB. The formula first calculates the Jacobian matrix and took its translation part, which is the first three row of the Jacobian. Then it multiplies the translation part of the Jacobian with its transpose. After that, it calculates the square root of the matrix. Norm of each row of the resulting matrix is equal to the magnitude of semi-axes of manipulability ellipsoid. For rotation, same calculations are done with using the rotation part of the Jacobian which is fourth and fifth rows.

From the second graph Figure 4.9, we can say the manipulability of X and Z axes increase and decrease mostly at the same time and their values are very close. On the other hand, the magnitude of Y-axis of manipulability ellipsoid which points through the forward mostly (by saying forward, we mean the X-axis of `FRAME_ROBOT`) decreases when the manipulability of X or Z axes increase and vice versa. Y-axis has the least manipulability all over the time which can also be seen in the manipulability ellipsoid in Figure 4.10. The ellipsoids are plotted at six different

For the manipulability of orientations in the third plot of Figure 4.15, we can say the manipulability of roll and yaw axes increase and decrease mostly at the same time and their values are very close. On the other hand, manipulability of pitch axis decreases when the manipulability of roll or yaw axes increase and vice versa. Pitch axis has the least manipulability all over time.

4.5. Testing “Both arms” Mode of The System

During the test of this mode, we were focused on the rotational valve turning movement. In the first experiment, we used the inverse kinematics solver of RTB instead of NAOqi’s solver for “Both arms” mode. Our purpose was to see the approximate solutions when all the target axes are absolute which is not possible when using NAOqi’s solver because it only produces a result when the target transformation is reachable with a little error tolerance. But the solver of RTB generates arm angles with a much greater error tolerance, therefore all axes can be absolute.

The error of position in Figures 4.17 and Figure 4.18 are not much different from the errors in single arm modes because the translation calculation works the same way. However, the error decreased in Figure 4.18 compared to Figure 4.17 because the solver of NAOqi ignores target pitch and yaw and that makes the target transformation much more reachable.

The improvement from the first experiment to the second can also be seen by looking at the roll axis data in Figure 4.19 and Figure 4.20. The maximum error of the roll axis has diminished from 1 to 0.5 radians. Keep in mind the delay in the second experiment, which affects the error. The delay between target and actual orientation can go up to 1.4 seconds (it was 0.6 and 0.8 seconds in right and left arm modes respectively) as the calculation overhead of “Both arms” is more than the single arm modes. The main reason for the higher delay is the main program calculates the manipulability of both arms and target transformation calculation has more overhead.

and 83, the manipulability measure of the right arm surpasses the threshold. Although their target and actual transformations are symmetrical to each other, their manipulability values do not stick at the close values all the time. This is especially seen in configurations where the target roll becomes far from zero, where the rotational movement is happening. And the manipulabilities are very close to each other when the target roll is 0, that can be seen in the seconds between 55 and 60. The reason for this situation is the configurations produced by inverse kinematics solver are not symmetric for the joints shoulder pitch, shoulder roll, and elbow roll. Thus, the situation forces us to check the manipulability of both arms every iteration if one of them is lower than the threshold.

In the second and third manipulability plots of Figure 4.21, the values indicate the magnitude of semi-axes of manipulability ellipsoids of right and left arm (axes ending with R belong the right arm and ending with L belong to left arm in the graph) over time. When we look at the manipulabilities in the second plot, we see the values of X for both arms are close to each other. The same thing can also be said for Y-axis. However, the values are symmetric for Z-axes. This is because the rotational movement mostly affects the Z axis of the manipulability ellipsoid. As the roll is the most important axis for determining how the rotational movement will take place, we can see its impact on Z by looking at the third and second plot together. In the third plot, the manipulability values of roll axes of left and arm are symmetric. The roll value might be a clue of which arm is going to surpass the threshold, but we cannot verify this hypothesis by looking the plots and have to calculate the manipulability of both arms every iteration.

We can see the continuous force that holds the wand at the last pose where the button was last pressed in the middle of the force and torque plots in Figure 4.21. These forces are also seen at the beginning and the end of time in Figure 4.21 which shows how the force is imposed on the wand according to manipulability measure while teleoperating both arms during the second experiment. The impact power of the feedback force is lesser than the holding force because its purpose is just informing the user, not to move the Wand to a target pose.

- Question 3: How was the responsiveness of the system and could you adapt yourself to the latency?
- Question 4: How was the camera feedback?
- Question 5: Was the system easy to use and comfort or did you become exhausted after using?

Five different subjects tested our system and the results are given in the Tables 4-1, 4-2, 4-3 and 4-4. The expert user completed the experiment in three and a half minutes without any failure.

Table 4.1: General information about subjects.

Subjects About	1	2	3	4	5
Age	28	26	26	20	35
Sex	Male	Male	Male	Female	Male
Occupation	Researcher	Student	Student	Student	Researcher

Table 4.2: Time of completion for each step of the task.

Subjects Time of completion	1	2	3	4	5
Step 1	00:50	01:15	00:08	00:38	00:35
Step 2	02:35	02:55	00:50	01:30	02:15
Step 3	05:47	06:39	02:10	03:44	05:24
Step 4	06:00	07:10	03:43	04:15	07:75

Table 4.3: Attempts and failures of the subjects.

Subjects	1	2	3	4	5
Finished at n th attempt	1	5	2	2	2
Failures at the last recorded test of the subject	None	Dropped the object once	Robot fell once, dropped the object once	None	None

Table 4.4: Comments of the subjects after experiments.

Subjects Answers	1	2	3	4	5
Question 1	Helpful	Helpful	Not very helpful	Helpful	A little helpful
Question 2	Easy and pleasant	Easy, would be better with 2 haptics	Easy	Medium	Easy for young, hard for old people
Question 3	Makes a little hard but can be adapted	Good enough and adapted without any problem	Not much latency, adapted	Latency is not bad, like 1 second, adapted	Latency is very low, adapted
Question 4	Field of view is not enough	Field of view is not enough	Field of view is not enough	Well enough to detect target, field of view is not enough	Field of view is not enough
Question 5	Easy and comfortable	Easy and comfortable	Easy and comfortable	Easy and comfortable	Exhausting

5. CONCLUSION

In this study, we have successfully designed and implemented teleoperation of a humanoid robot via a haptic feedback device . As opposed to the other studies in robotic literature, the proposed system, does not require extra wearable hardware to manipulate the humanoid robot. On the host side, where the human operator is placed in an office like environment, the operator uses a haptic device and a computer mouse. And on the remote side a humanoid robot, in our case Nao from Softbank Robotics . The main contribution of our study compared to the current state of the art in the literature is providing a multimodal haptic tele-operating system for operating humanoid robots with ease. Experiments showed that learning how to use the system and adapting is relatively easy.

The performance of the proposed system is evaluated for each teleoperation mode separately by interpreting data taken from demonstrations. Results showed that the system was successful in reaching target arm poses unless precise movements are required.

The main lesson learned, is that the arms of Nao which has 5 DOF are kinematically insufficient for the tasks we aimed to accomplish. This insufficiency was exposed during the test of valve turning movement with both hands, as the high complexity of the movement. While the operator controlling the arms, the system imposes force feedback by using a rendering algorithm based on manipulability, which is well suited for amending kinematic insufficiency.

In addition to individual test of every mode, overall test of the system is done for a pick and place task with five different subjects. Their comments and test performance are recorded. According to their comments, the system is easy to use and comfortable in terms of human computer interaction.

Future work can be improving responsiveness of the system by re-implementing the communication layer with strict real-time constraints. The PD-type controller of the Wand can be improved or redesigned for better accuracy. An inverse kinematics solver for arms can be implemented from scratch instead of using the one coming with NAOqi to reach the intermediate steps and to avoid unnecessary computations. In addition to these, subjects of the overall test also said what can be done in the future. Some of these are re-implementing mode change system to use keyboard strokes

instead of mouse clicking, adding a virtual wall to define the limits of arm movements and stopping all the motions when the red button on the Wand become released.

REFERENCES

- [1] Web 1, (2019), https://en.wikipedia.org/wiki/Chernobyl_disaster, (Accessed: 16/04/2019).
- [2] Web 2, (2019), <https://www.saabseaeye.com/news/jaguar-attack>, (Accessed: 16/04/2019).
- [3] Web 3, (2019), <https://www.davincisurgery.com/da-vinci-systems/about-da-vinci-systems>, (Accessed 16 April 2019).
- [4] Web 4, (2019), <https://www.shadowrobot.com/telerobots/>, (Accessed: 16/04/2019).
- [5] Web 5, (2019), https://www.nasa.gov/mission_pages/station/research/experiments/945.html, (Accessed: 16/04/2019).
- [6] Web 6, (2019), <https://www.darpa.mil/program/darpa-robotics-challenge>, (Accessed: 17/04/2019).
- [7] Web 7, (2019), http://quanser-update.azurewebsites.net/quarc/documentation/quanser_5dof_wand_blocks.html, (Accessed: 18/04/2019).
- [8] Web 8, (2019), http://doc.aldebaran.com/2-1/home_nao.html, (Accessed: 18/04/2019).
- [9] Zhang L., Huang Q., Lu Y., Xiao T., Yang J., Keerio M., (2006), "A Visual Tele-operation System for the Humanoid Robot BHR-02", IEEE/RSJ International Conference on Intelligent Robots and Systems, 1110–1115, Beijing, China, 9-15 October.
- [10] Rodriguez I., Astigarraga A., Jauregi E., Ruiz T., Lazkano E., (2014), "Humanizing NAO robot teleoperation using ROS", 14th IEEE-RAS International Conference on Humanoid Robots, 179-186, Madrid, Spain, 18-20 November.
- [11] Web 9, (2019), <https://developer.microsoft.com/en-us/windows/kinect>, (Accessed: 25/04/2019).
- [12] Almetwally I., Mallem M., (2013), "Real-time tele-operation and tele-walking of humanoid Robot Nao using Kinect Depth Camera", 10th IEEE International Conference on Network, Sensing and Control, 463-466, Evry, France, 10-12 April.
- [13] Koenemann J., Burget F., Bennewitz M., (2014), "Real-time imitation of human whole-body motions by humanoids", IEEE International Conference on Robotics and Automation, 2806-2812, Hong Kong, China, 31 May -7 June.

- [14] Web 10, (2019), <https://www.xsens.com/products/xsens-mvn-animate>, (Accessed: 25/04/2019).
- [15] Peer A., Hirche S., Weber C., Krause I., Buss M., Miossec S., Evrard P., Stasse O., Neo E. S., Kheddar A., Yokoi K., (2008), "Intercontinental multimodal tele-cooperation using a humanoid robot", IEEE/RSJ International Conference on Intelligent Robots and Systems, 405-411, Nice, France, 22-26 September.
- [16] Web 11, (2019), http://global.kawada.jp/mechatronics/hrp2.html?utm_source=robots-ieee-org, (Accessed: 25/04/2019).
- [17] Evrard P., Mansard N., Stasse O., Kheddar A., Schauß T., Weber C., Peer A., Buss M., (2009), "Intercontinental, multimodal, wide-range tele-cooperation using a humanoid robot", IEEE/RSJ International Conference on Intelligent Robots and Systems, 5635-5640, St. Louis, USA, 10-15 October.
- [18] Renon P., Yang C., Ma H., Cui R., (2013), "Haptic Interaction Between Human and Virtual iCub Robot Using Novint Falcon with CHAI3D and MATLAB", Proceedings of the 32nd Chinese Control Conference, 6045-6050, Xi'an, China, 26-28 July.
- [19] Web 12, (2019), <http://www.icub.org/>, (Accessed: 25/04/2019).
- [20] Corke P., (2017), "Robotics, Vision & Control", Springer.
- [21] Ju Z., Yang C., Li Z., Cheng L., Ma H., (2014), "Teleoperation of humanoid baxter robot using haptic feedback", International Conference on Multisensor Fusion and Information Integration for Intelligent Systems, 1-6, Beijing, China, 28-29 September.
- [22] Web 13, (2019), [https://en.wikipedia.org/wiki/Baxter_\(robot\)](https://en.wikipedia.org/wiki/Baxter_(robot)), (Accessed: 25/04/2019).
- [23] Web 14, (2019), <https://www.3dsystems.com/haptics-devices/touch>, (Accessed: 25/04/2019).
- [24] Barros J. J. O., Santos V. M. F. d., Silva F. M. T. P. d., (2015), "Bimanual Haptics for Humanoid Robot Teleoperation Using ROS and V-REP", IEEE International Conference on Autonomous Robot Systems and Competitions, 174-179, Vila Real, Portugal, 8-10 April.
- [25] Web 15, (2019), <http://www.coppeliarobotics.com/>, (Accessed: 25/04/2019).
- [26] Fritsche L., Unverzag F., Peters J., Calandra R., (2015), "First-person tele-operation of a humanoid robot", IEEE-RAS 15th International Conference on Humanoid Robots, 997-1002, Seoul, South Korea, 3-5 November.
- [27] Web 16, (2019), <https://www.oculus.com/rift/>, (Accessed: 25/04/2019).

- [28] Lofaro D. M., Bula M., Early P., Eide E., Javid M., (2015), "ARCHR — Apparatus for Remote Control of Humanoid Robots", IEEE-RAS 15th International Conference on Humanoid Robots, 229-236, Seoul, South Korea, 3-5 November.
- [29] Graziano A., Tripicchio P., Avizzano C. A., Ruffaldi E., (2016), "A wireless haptic data suit for controlling humanoid robots", Proceedings of ISR 2016: 47st International Symposium on Robotics, 334-341. Munich, Germany, 21-22 June.
- [30] Panzirsch M., Weber B., Rubio L., Coloma S., Ferre M., Artigas J., (2017), "Tele-healthcare with humanoid robots: A user study on the evaluation of force feedback effects", IEEE World Haptics Conference (WHC), 245-250, Munich, Germany, 6-9 June.
- [31] Abi-Farraj F., Henze B., Werner A., Panzirsch M., Ott C., Roa M. A., (2018), "Humanoid Teleoperation Using Task-Relevant Haptic Feedback", IEEE/RSJ International Conference on Intelligent Robots and Systems, 5010-5017, Madrid, Spain, 1-5 October.
- [32] Karaküçük H., (2014), "Arduino control with the Zenom simulation environment", Master Thesis, Gebze Technical University.
- [33] Ay C., (2015), "The 5 DOF haptic wand control with the Zenom simulation environment", Master Thesis, Gebze Technical University.
- [34] Ay C., Karaküçük H., Zergeroğlu E., (2014), "Design and Implementation of Linux/Xenomai Based Real-time Subsystem: Zenom", Automatic Control Turkish National Committee (TOK), Kocaeli, Turkey, 11-13 September.
- [35] Web 17, (2019), <https://wiki.ubuntu.com/Kernel/Reference/stress-ng>, (Accessed: 30/04/2019).
- [36] Kaleli E., (2011), "Design and Implementation of Real-time Linux-Xenomai Driver for Quanser I/O Card", Master Thesis, Gebze Institute of Technology.

BIOGRAPHY

Ahmet Soyyigit was born in 1993 in Istanbul, Turkey. He earned a Bachelor of Science degree in Computer Engineering from Gebze Technical University in 2016. Afterward, he started pursuing his graduate research in the field of robotics and real-time system at the Institute of Natural and Applied Sciences of the same university. He is also a teaching assistant in Gebze Technical University Computer Engineering Department.