Final Year Project Report

Development of a Teach-Pendant for a Humanoid Robot with Cartesian and Joint-Space Control Modalities

Zhenis Otarbay, Iliyas Assylgali

A thesis submitted in part fulfilment of the degree of

BSc in Robotics and Mechatronics

Supervisor: Michele Folgheraiter

Department of Robotics and Mechatronics
School of Science and Technology
Nazarbayev University

May 2, 2019

Table of Contents

Abstract

The main goal of this project is to design and build a remote controller (teach-pendant) for the biped robot under development at the department of Robotics and Mechatronics in Nazarbayev University. The system is equipped with a touch-based Graphical User Interface (GUI) from which the robot's joints and the robot's end-effectors can be controlled in the joint and Cartesian space respectively. The chassis of the teach-pendant was designed using SolidWorks and FreeCAD software in order to accommodate a 9-inch display with a touch sensor and control panel, a 5000 mAh battery, a Raspberry pi 3, and an Arduino Nano. The forward and inverse kinematic model are formalized using Denavit-Hartenberg convention and computed using Octave and Matlab software. Octave is used to draw frames, so it is possible to verify if the forward and inverse kinematics calculations are correct and accurate. Matlab is used together with the Peter Corke's Robotics toolbox to compare the solutions with the one calculated by our algorithms. To allow real time calculation of the FK and IK solution and implement the GUI, a Raspberry pi 3 was used. In addition an Arduino Nano was installed to process information from the two pot-encoders and the emergency button and send it to Raspberry pi 3. The algorithms were coded in Python 3 and the the Tkinter" library was used as a GUI builder. Different modules were added to interface with the robot main control systems by using TCP-IP protocol and to interact with the servomotors.

Acknowledgments

At first we would like to thank our supervisor Dr. Michele Folgheraiter for his guidance and patience. Furthermore, we would like to thank Dr. Matteo Rubagotti, Dr. Atakan Varol, Dr. Almas Shintemirov, Dr. Tohid Alizadeh, Dr. Mohammad Mosadeghzad, Dr. Anara Sandygulova, Dr. Berdakh Abibullaev, Ms. Altay Zhakatayev, Ms. Iliyas Tursynbek for helping us and as Professors to teach us many important subjects throughout the four year of study in Nazarbayev University.

Chapter 1: **Motivation and Problem Definition**

The main goal of this project is to design and to build a touch-sensitive wireless teach pendant (TP) for the NU-Biped Robot (see Fig 1.1) developed at the Humanoid Robotics Laboratory of Nazarbayev University. A teach pendant is a remote controller meant to facilitate the monitoring and the operation of a robotic system, in our case the biped robot. The system should allow an intuitive control of the robot both in the joint and Cartesian space. Furthermore, it should allow to shut down safely the system in case of an emergency situation, e.g. when a human enter in collision with the robot or when some of the robot components are malfunctioning.

This project is mainly technological in nature. However, for what regard the solution of the Inverse Kinematics (IK), we adapted for our scope a method found in literature. Different problems have been encountered and solved during the execution of the project. A compact and ergonomic design of the TP that includes all the required components and the internal connections. The implementation of an intuitive GUI that, by using an intuitive touch interface, allows to control the joint position, to calculate and implement the Forward Kinematics (FK) visualization, to solve the IK, and to implement the NU-Biped TP and Robot Controller wireless connection. All the software is meant to operate in real time while the NU-Biped Robot is operating.

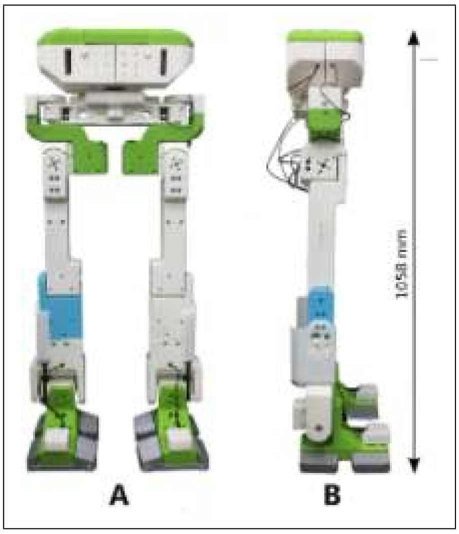


Figure 1.1: NU Biped

Although it can be used for different robots, the TP was developed specifically for the NU-Biped. The robot weight is only 12.4 kg, while its height is about 1 m. There are 6 rotational joints in the left leg of the robot as well as 6 rotational joints in the right leg. The main computational unit on-board is based on a Raspberry Pi 3 and a different micro-controllers, while the central computational unit is connected by using a TCP/IP protocol. According to Folgheraiter et al. [1] the robot actuation and computational units absorb a total nominal power of 409 W while performing dynamic movements, therefore the power to weigh ratio is $\frac{Power}{Weight} = 37.3\frac{W}{kg}$. If we take in consideration also the height, the $\frac{NominalPower*Height}{Weight} = 37.3 * 1.8 = 67.1\frac{W*m}{kg}$.

The rest of this document is organized as follow: Chapter 2 reviews the works and books that we used during this project, Chapter 3 introduces the forward and inverse kinematics of the Biped Robot together with their visual implementation. Chapter 4 describes the implementation of the software and hardware. Chapter 5 brings the main results and the last chapter the conclusions and the future work.

Chapter 2: **Literature Review and Related Work**

Folgheraiter et al. [1] introduces the kinematics and the mechanical design of the newly developed 12 DOFs biped robot. The most important ideas in the paper is how to design a low cost and low weight biped robot by using 3D printing techniques and light weigh materials. This, to reduce the energy consumption and render the robot more safe while interacting with human beings. The dimensions of the NU-biped leg parts are provided together with the FK model. In particular it provides the DH table and the dimension L1, L2, L3, L4 that are used in this work.

Evaluating the proposed solutions, we can state that this robot in terms of power to weight ratio performs better than other state-of-the-art bipeds and that the problems that could appear while building or working with the robot are solvable.

The NU-biped's inverse kinematics solution is similar to that one of the humanoid robot presented in Muhammad et al. [2]. However, there are also some differences in calculating the angles and in the DH parameters table. For example, in the DH table of the NU-biped the links offsets are absent and the 5th joint has a different solution. Furthermore, some angles such as θ_4 and θ_5 are calculated differently in our solution.

In Kajita's et al. [3] a simplified dynamic model of the biped is presented based on the dynamics of an inverted pendulum. Although, not directly related with this work, the work is interesting due to the fact that in the future we can extend the TP interface to include some parameters adjustment of the balancing control system. It will be possible to perform experiments and by using the TP change in real time the behavior of the control system.

The Lung-Wen's book [4] and the Craig's books [5] were used extensively for this work. They introduce the DH convention, give a methodology to compute the FK by using the transformation matrices and suggest some methods to obtain the closed form inverse kinematics solutions. Furthermore, they indicate a possible strategy to select the more convenient solution among the one obtained (e.g. the "closest" solution that avoids the collision with obstacles present in the robot's work-space).

Bruno's book [6] has useful material dedicated to Robotics. It introduces the modeling, the planning and control together with a brief history of robot architecture development.

In the Kenneth's paper [7] are reported some relevant and useful information about the usage of TPs in robotics. It introduces some design ideas and can be used to compare them with our solution.

Sundaram's [8] is about a Bluetooth communication between a laptop and raspberry pi 3. This paper is relevant for us because the connection between the robot hardware and the TP or a laptop is one of the important parts in this project.

Sung-nam's paper [9] is about the design of a GUI for a 12 DOF biped-robot. The GUI makes easy to control the robot. The mention TP uses different micro-controllers such as FPGA and DSP. The architecture of the control system of the biped robot is also described.

Chapter 3: **Kinematic Model**

3.1 Forward Kinematics and Visualization

The FK of the NU-Biped is based on the modified DH-convention due to J. Craig [5]. The DH parameters for the NU-Biped robot are reported in Table 3.2, the links lengths in Table 3.1, while in Fig.3.1 is reported the Kinematic architecture of the robot. In the original model the base frame was the frame 0, while the last one frame 6. In this case, to find the FK we need to find T06.

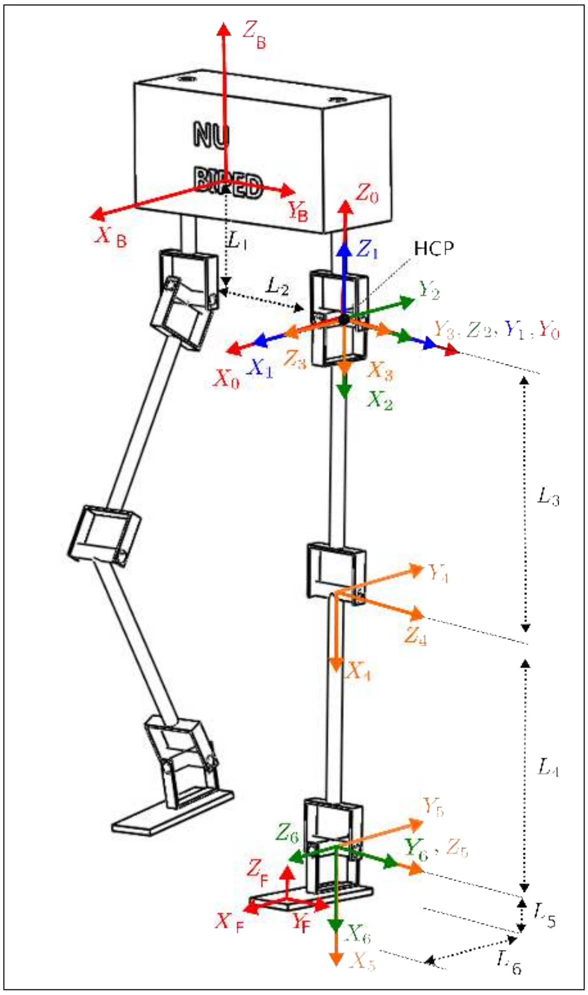


Figure 3.1: Kinematics architecture of the Biped Robot

Table 3.1: Dimensions of the NU-biped.

Lengths	Value (mm)
L_1	200
L_2	154.75
L_3	318.50
L_4	318.50

Table 3.2: DH-table for a single leg of the biped according to the modified Craig-DH convention. Angles are in deg.

Joint	α_{i-1}	a_{i-1}	d_i	θ_i
1(Hip Yaw)	0	-	-	θ_1
2(Hip Pitch)	-90	0	0	$\theta_2 + 90$
3(Hip Roll)	90	0	0	θ_3
4(Knee Pitch)	-90	L_3	0	θ_4
5(Ankle Pitch)	0	L_4	0	θ_5
6(Ankle Roll)	90	0	0	θ_6

Starting from the transformation matrix that represents the pose of frame i with respect to frame i-1

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

for example, the transformation frame 1 with respect to 0:

$$T_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & L_2 \\ 0 & 0 & 1 & -L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we can obtain the FK T_6^0 as :

$$T_6^0 = \prod_{i=1}^{6} = T_i^{i-1}(\theta_i) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

r_{11} to r_{33} are rotation elements while p_x to p_z are position elements of T_0^6

In Fig. 3.1 are also included two additional frames, the one located in the center of the robot upper-body, frame B, and the frame that is displaces in the center of the foot's sole. These additional frames are necessary to define target trajectories for the feet and the body of the robot. Section 3.2 is dedicated to the calculation of the IK of the leg and is based on these additional two frames.

3.1.1 Forward Kinematics Visualization in Octave

To verify the correctness of direct kinematics calculation we plotted each frame by using Octave. For each transformation matrix involved in the FK we extracted the four columns and used them as versors and origin position for the correspondent frame representation. The four vectors are function of the joint angles, therefore the frames representation can be recomputed for each joints configuration. To allow seen graphically all the frames in a particular region, we apply scaling. As it is possible to notice in Fig. 3.1.1, frame 0, frame 1, frame 2 and frame 3 are represented in different colors and with different axes lengths. In the picture the links of the leg are represented with a dashed line.

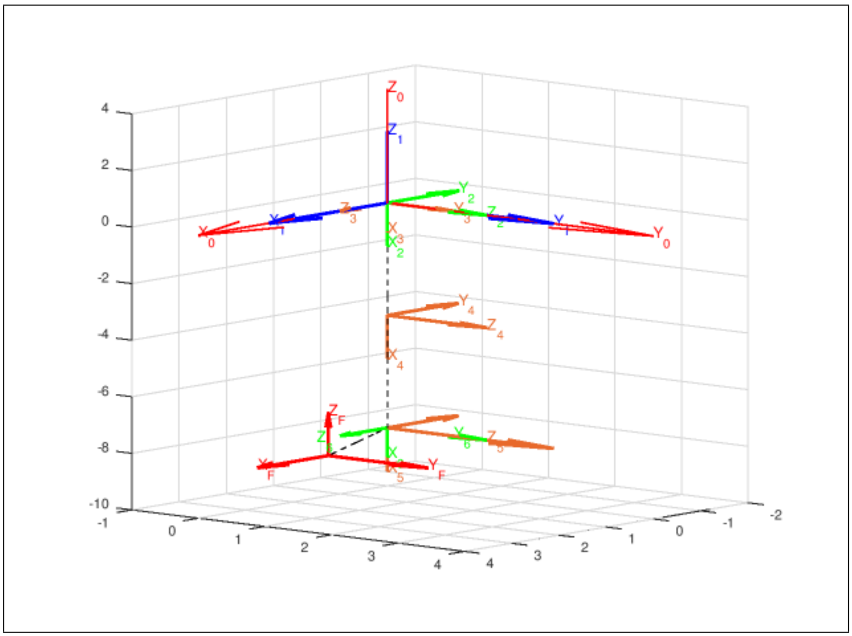


Figure 3.2: Left leg home position in Octave, in dm

While Fig. 3.1.1 represents the frames in their home position, Fig. 3.1.1 represents the frames in the case the knee of the leg is bent -90 deg with respect the z-axis of frame 4.

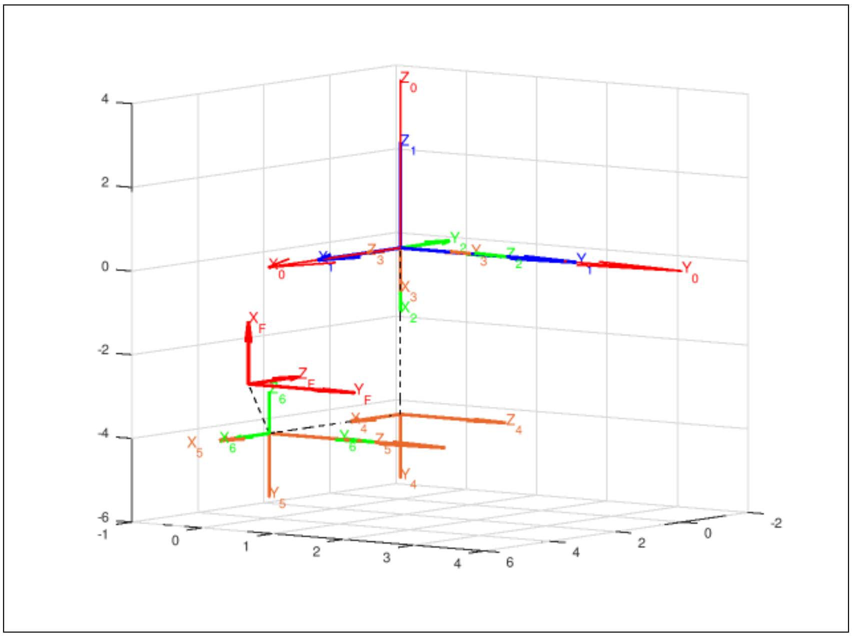


Figure 3.3: Left leg knee bent -90 degrees in Octave, in dm

3.2 Inverse Kinematics

The goal of the inverse kinematics algorithm is to find all the possible joint angles that bring the foot frame 6 to a wanted pose (position and orientation). In general for a 6 DOFs open kinematic chain the inverse kinematic problem is quite difficult to solve due to the fact that the equation we need to solve are transcendental. For the system of 16 equations in Eq. 3.1, of which four are trivial, we may have no solution, one or many solutions according to the pose we want to reach T_{Target} and the particular kinematic architecture.

$$T_{Target} = T_6^0 \tag{3.1}$$

In general, for open kinematic chains that have the last three joint axes intersecting in one point (most of the industrial manipulators verify this property), we can separate the solution in two parts. At first we solve for the first three joints in order to reach the target position and then for the last three to fulfill the requirement on the orientation. Thus, when the solutions are available for $\theta_1, \theta_2, \theta_3$ we can solve the system $T_0^3 * T_{Target} = T_6^3$ for $\theta_4, \theta_5, \theta_6$. In our case, however, not the last three joints but the first three joints (hip) are intersecting. This, requires to invert the kinematic problem as proposed in [2]. By taking the inverse of both left and right sides of Eq. 3.1 we obtain

$$T_{target}^{-1} = T_0^6 \tag{3.2}$$

where

$$T_0^6 =_5^6 T *_4^5 *..._0^1 T. \tag{3.3}$$

that has the following structure

$$\begin{bmatrix} u'_x & v'_x & w'_x & p'_x \\ u'_y & v'_y & w'_y & p'_y \\ u'_z & v'_z & w'_z & p'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ... & ... & ... & f_1(\theta_4, \theta_5, \theta_6) \\ ... & ... & ... & f_2(\theta_4, \theta_5, \theta_6) \\ ... & ... & ... & f_3(\theta_4, \theta_5, \theta_6) \\ ... & ... & ... & 1 \end{bmatrix}. \tag{3.4}$$

If now we take the three equations that form by considering the last column of the left and right side of Eq. 3.4 we obtain a system of equations that depend only on θ_4, θ_5 and θ_6

$$\begin{cases} p'_x = -(L_3 c_{45} + L_4 c_5) * c_6 \\ p'_y = (L_3 c_{45} + L_4 c_5) * s_6 \\ p'_z = -L_3 s_{45} - L_4 s_5 3) \end{cases} \tag{3.5}$$

By squaring and summing up the left and right sides of the three equations we obtain

$$p'^2_y + p'^2_y + p'^2_z = L_3^2 c_{45}^2 + L_4^2 c_5^2 + 2 L_3 L_4 c_{45} c_5 + L_3^2 s_{45}^2 + L_4^2 s_3^2 + 2 L_3 L_4 s_{45} s_5, \tag{3.6}$$

and by using the trigonometric identity $cos(\alpha + \beta) = cos(\alpha)cos(\beta) + sin(\alpha)sin(\beta)$ we obtain

$$p'^2_y + p'^2_y + p'^2_z = L_3 2 + L_4^2 + 2L3L_4 * (c_4) \tag{3.7}$$

which can be solved for c_4

$$c_4 = \frac{px'^2 + p_y'^2 + p_z'^2 - L_3^2 - L_4^2}{2L_3L_4} \tag{3.8}$$

and θt_4 as

$$\boxed{\theta_4\text{=atan2}(\pm\sqrt{1-c_4^2}, c_4)}$$

Squaring and summing up the first and second equations of Eq. 3.5 we obtain

$$c_5(L_3c_4 + L_4) - L_3s_4s_5 = \pm\sqrt{p_y'^2 + p_y'^2}, \tag{3.9}$$

that together with the third equation of Eq. 3.5 and with the substitution $k_1 = (L_3c_4 + L_4)$, $k_2 = L_3s_4$ and $k_3 = \sqrt{p_x'^2 + p_y'^2 + p_z'^2}$ let us to obtain the system

$$\begin{cases} k_1c_5 - k_2s_5 = \pm k_3 \\ \\ k_1s_5 + k_2c_5 = -p_z' \end{cases} \tag{3.10}$$

that by substitution can be solved for s_5 and c_5

$$c_5 = \frac{\pm k_3 + k_2s_5}{k_1}, s_5 = \frac{\frac{\pm k_2k_3}{k_1} - p_z'}{k_1 + \frac{k_2^2}{k_1}} \tag{3.11}$$

and therefore for θ_5

$$\boxed{\theta_5 = atan2(s_5, c_5)}$$

chosen θ_4 there are two correspondent solutions for θ_5.

By dividing the first and second equations of Eq. 3.5 we obtain $\frac{c_6}{s_6} = \frac{p_y'}{p_y'}$ and therefore

$$\boxed{\theta_6 = atan2(p_y', -p_x')}$$

A possible different approach to obtain θ_5 and θ_6 is also reported in the Appendix.

To solve for θ_1, θ_2 and θ_3 we can multiply the left sides of Eq. 3.2 by the inverse of T_6^3 obtaining

$$T_3^6 *_6^0 T_{target}^{-1} = T_0^3 \tag{3.12}$$

where

$$T_3^6 *_6^0 T_{target}^{-1} = \begin{bmatrix} u"_x & v"_x & w"_x & p_x" \\ u"_y & v"_y & w"_y & p_y" \\ u"_z & v"_z & w"_z & p_z" \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

At this point the right side of Eq.3.13 is completely known, while the right side of equation Eq. 3.13 is

$$T_0^3 = \begin{bmatrix} -s_1s_3 - s_2c_1c_3 & -s_1s_2c3 + s_3c_1 & -c_2c_3 & * \\ -s_1c_3 + s_2c_1s_3 & s_1s_2s_3 + c_1c_3 & s_3c_2 & * \\ c_1c_2 & s_1c_2 & -s_2 & * \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

By considering the elements (3,3) of Eq. 3.13 and Eq. 3.14 we can solve easily for θ_2 :

$$\boxed{\theta_2 = arcsin(-w"_z), \theta_2^* = \pi - \theta_2}$$

By considering the elements (3,2) and (3,3) of Eq. 3.13 and Eq. 3.14 and assuming $c_2 \neq 0$ we can solve for θ_1 as

$$\boxed{\theta_1 = atan2(\frac{v"_z}{c_2}, \frac{u"_z}{c_2})}$$

Finally, considering the elements (1,3) and (2,3) of Eq. 3.13 and Eq. 3.14 and assuming $c_2 \neq 0$ we can solve for θ_3 as

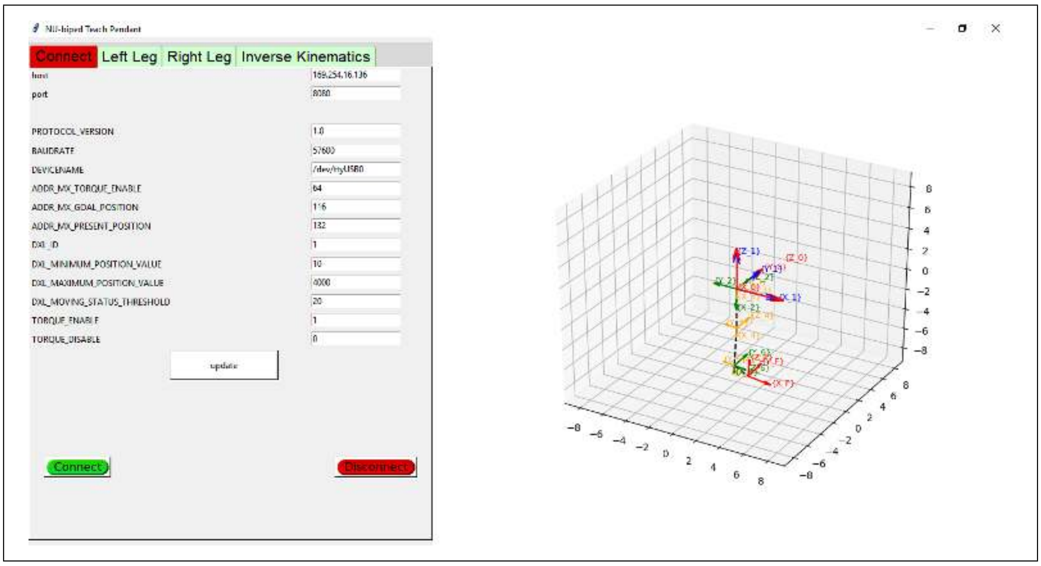
$$\boxed{\theta_3 = atan2(\frac{w"_y}{c_2}, \frac{-w"_x}{c_2})}$$

Overall, we have two solutions for θ_4, two solutions for θ_5, one correspondent solution for θ_6, two solutions for θ_2 and one correspondent solution for both θ_1 and θ_3. Therefore we have 8 possible set of solutions for the IK of the leg.

Chapter 4: **Teach Pendant Implementation**

4.1 Teach Pendant Graphical User Interface

The GUI represents the central part of the teach pendant. It allows to use all the functionalists offered by the software. The GUI was developed in Python 3 and the Tkinter library. As it is shown in Fig. 4.1 it consists in two main parts. The left one including four different window tabs and the right one that represents graphically the pose and the frames attached to each link of the leg.



host	169.254.16.136
port	8080
PROTOCOL_VERSION	1.0
BAUDRATE	57600
DEVICENAME	/dev/ttyUSB0
ADDR_MX_TORQUE_ENABLE	64
ADDR_MX_GOAL_POSITION	116
ADDR_MX_PRESENT_POSITION	132
DXL_ID	1
DXL_MINIMUM_POSITION_VALUE	10
DXL_MAXIMUM_POSITION_VALUE	4000
DXL_MOVING_STATUS_THRESHOLD	20
TORQUE_ENABLE	1
TORQUE_DISABLE	0

Figure 4.1: Teach Pendant Graphical User Interface (Robot Connection).

The first window tab allows the connection through the TCP/IP protocol to the main robot controller, furthermore other parameters of the robot can be changed by setting the values of different fields.

The second window tab, see Fig 4.1, allows to change the position of all the joints of the left leg of the NU-Biped. In total 6 sliders are present that are initially set in the middle position (zero degree). Two control modalities are possible by clicking one of the two buttons at the bottom of the window. In the "Update Manually" modality it is possible to move the sliders, see the effect on the leg kinematic and if the movement is the one wanted send the reference positions to the robot leg. In the second modality, "Update Automatically", the positions are directly send to the robot controller after each movement of the sliders.

The third tab windows is similar to the second one, but operates the joints of the right leg of the biped. Finally, the last tab window (see Fig. 4.1) allows to directly set the pose (position and orientation) of the left and right foot by using the solution of the IK.

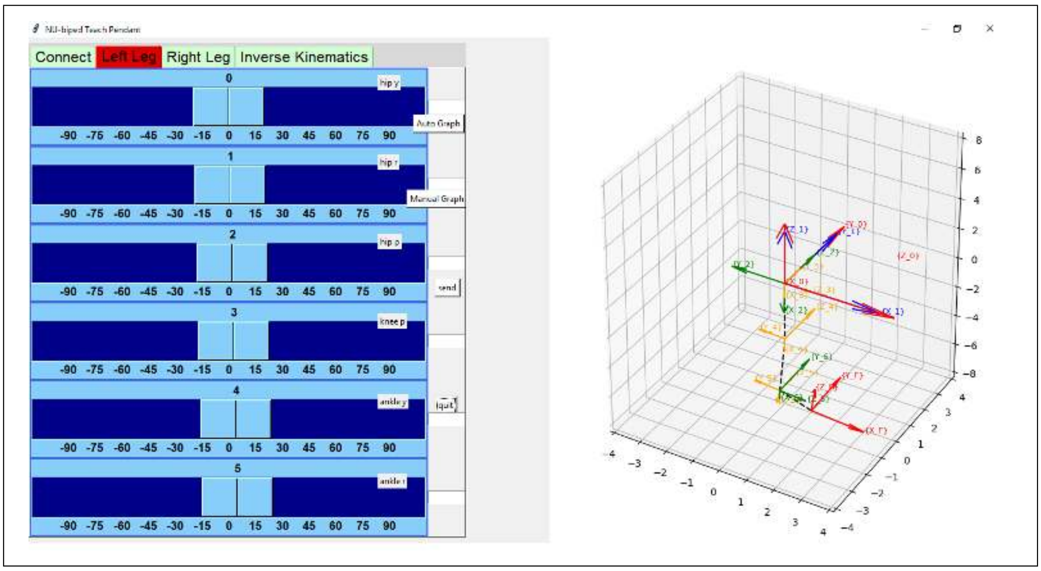
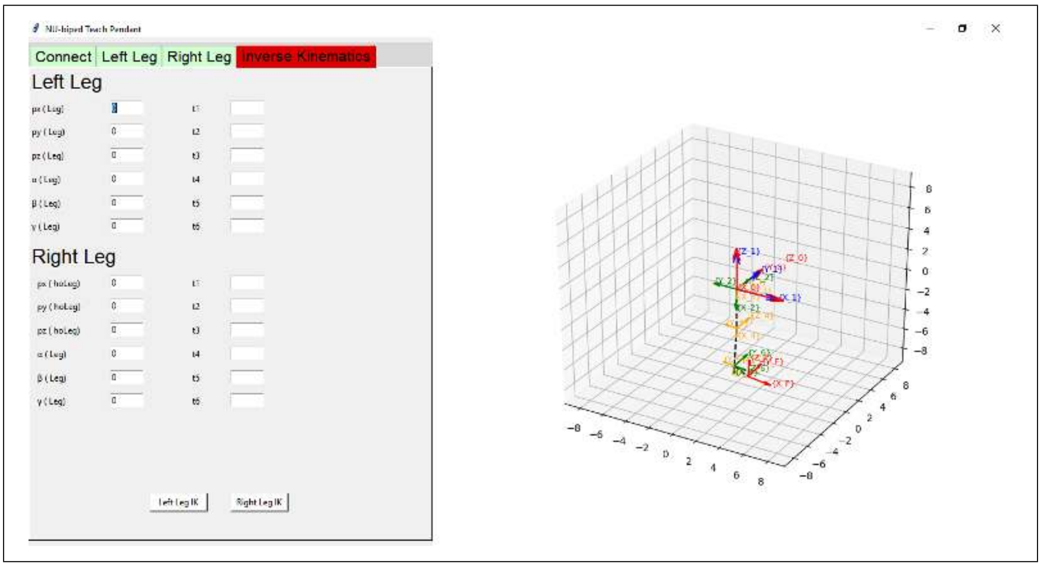


Figure 4.2: Teach Pendant Graphical User Interface (Joint Space Control)



Left Leg

px (Leg) 0 t1
py (Leg) 0 t2
pz (Leg) 0 t3
α (Leg) 0 t4
β (Leg) 0 t5
γ (Leg) 0 t6

Right Leg

px (hoLeg) 0 t1
py (hoLeg) 0 t2
pz (hoLeg) 0 t3
α (Leg) 0 t4
β (Leg) 0 t5
γ (Leg) 0 t6

Figure 4.3: Teach Pendant Graphical User Interface (Inverse Kinematics)

4.2 Software Architecture and Main Hardware Components

The teach pendant Software Architecture includes different software modules implemented in Python:the GUI, the Forward kinematics, the Inverse Kinematics and the TCP/IP connection. They interface through a wireless connection to the main robot controller, which includes also the TCP/IP connection and the robot joints controller.

The main hardware components included in the the teach pendant are showed in Fig. 4.2. The Raspberry pi 3 represents the main computational unit where the software modules run, this is connected through a USB port to the Arduino Nano that is dedicated to read the activation of the emergency button and the change in position of the two pot-encoders. The 9 inches touch screen is connected through a dedicated video driver card to the HDMI connector of the Raspberry pi 3. In addition, a touch controller is connected to a USB port of the Raspberry pi 3. Finally, a plate with a set of buttons is connected to the video driver

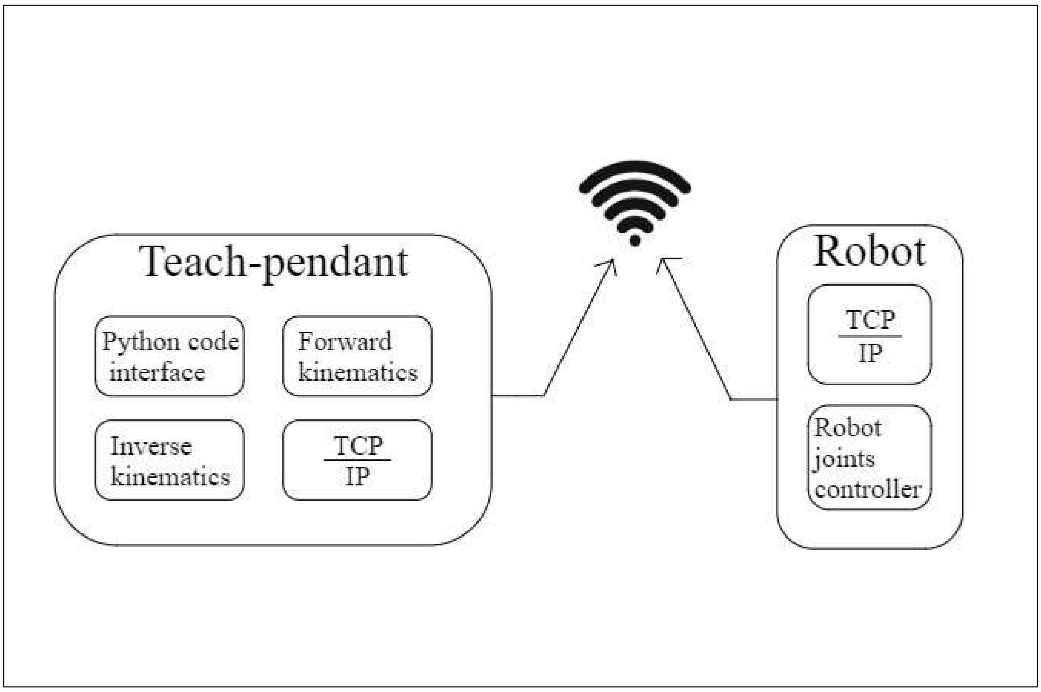


Figure 4.4: Teach Pendant Components and Software Architecture

in order to set the screen brightness etc. The teach pendant includes also an internal 5000 mAh battery pack in order to provide up to two hours of power.

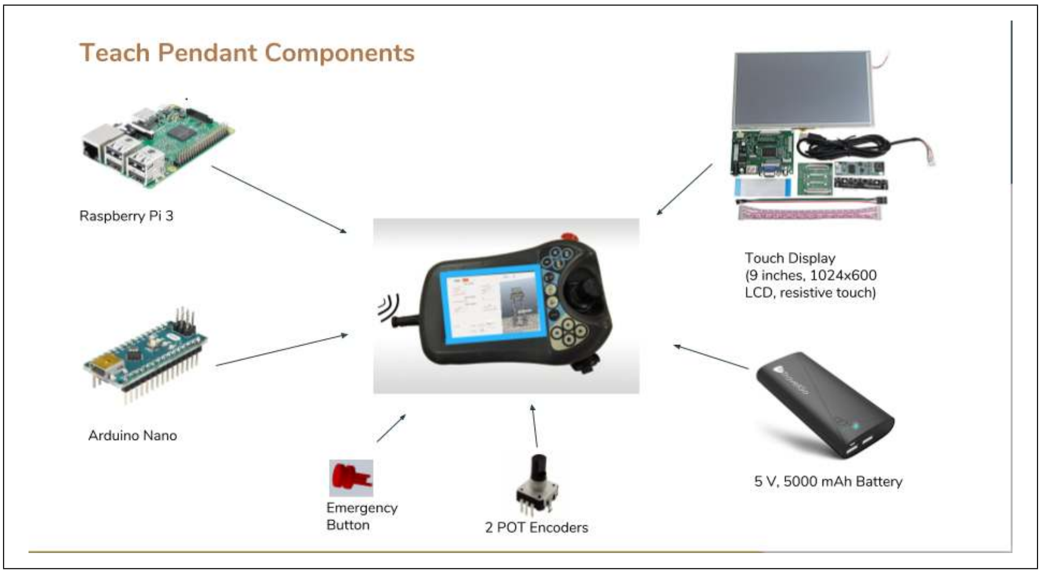


Figure 4.5: Teach Pendant Components

4.3 Teach Pendant Design

The teach pendant chassis was designed using Solid Works and FreeCad software and printed in PLA material with a Ultimaker S5. While designing the teach pendant chassis, our main goal was to accommodate all the required hardware components in the smallest space possible.

At first we did a draft design of the Teach Pendant. After that, we designed the upside and downside parts of the Teach Pendant (see Fig. 4.3). We made the display holders, the clearing holes for the bolts and the battery pack holders. So, we proceed making the inside element places. To verify the correctness of the model in Solidworks we have made a 9 inch display model and downloaded the cad models of the Raspberry pi3 and Arduino Solidworks from the open source GrabCAD.com (only raspberry pi 3 and Arduino were downloaded others are our own models).

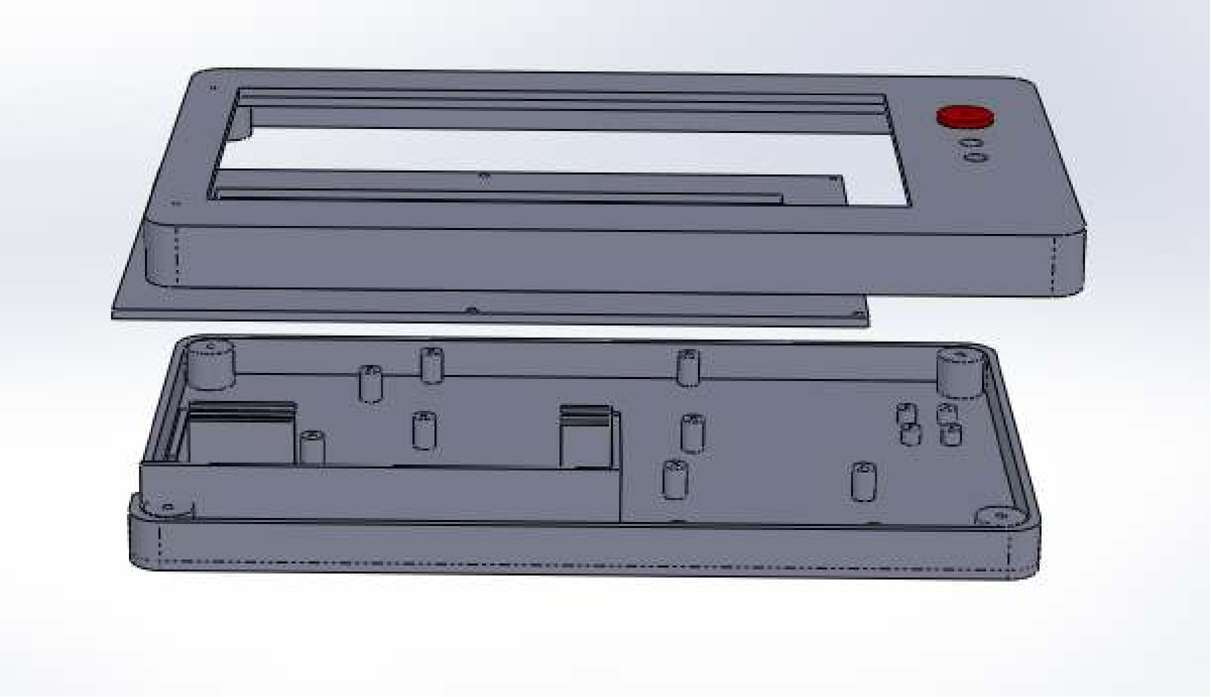


Figure 4.6: Teach Pendant Design (CAD model assembly)

The final CAD model of the teach pendant has a length of 30 cm, its width is 18 cm, and its height is about 4 cm. It has places designed for holding the battery, the raspberry pi 3, the USB port battery charger. 4 clearing holes are for the video driver, 2 clearance holes are instead for the touch sensor.

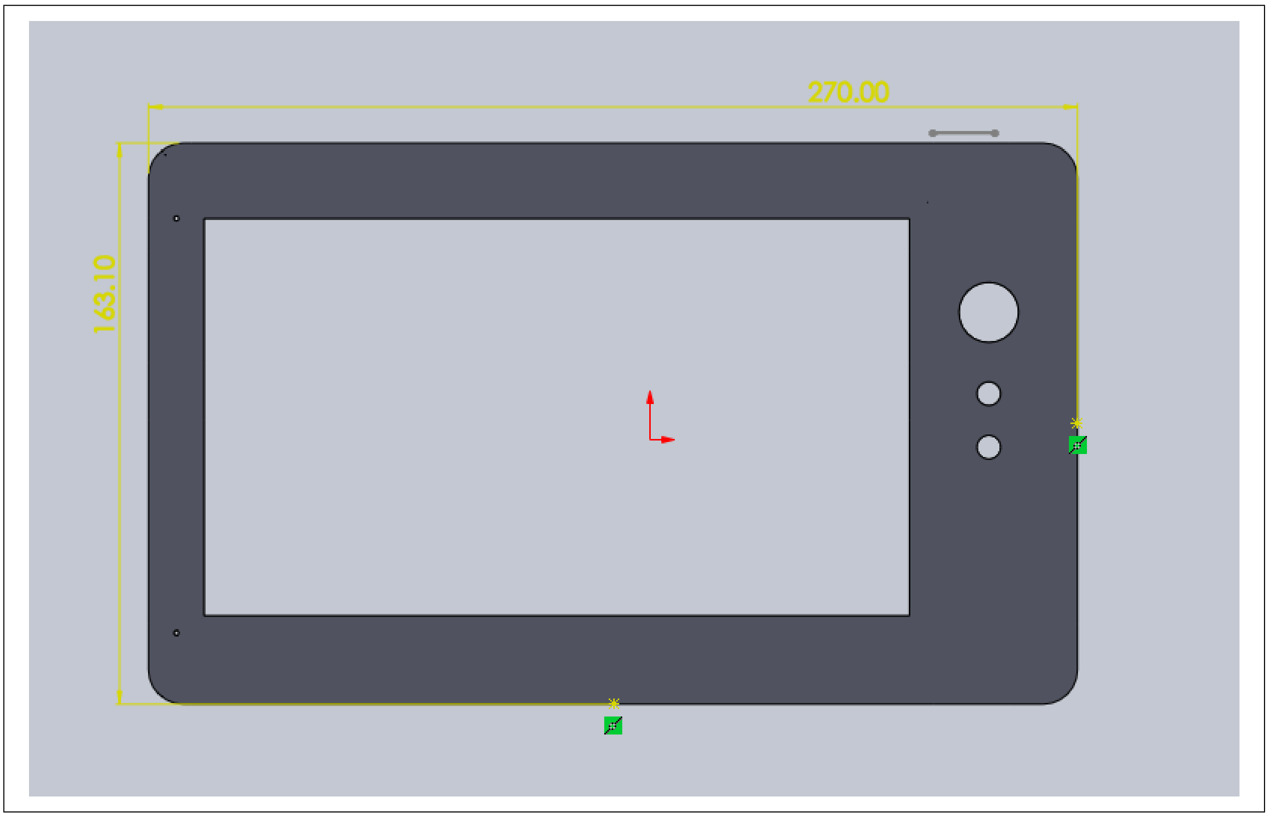


Figure 4.7: Teach Pendant Design (CAD model upside with holder)

Chapter 5: **Evaluation and Results**

When the CAD model was finalized we proceed printing the different parts of the chassis, the emergency button and the different supports necessary for the screen and the electronics boards. To realize the assembly of the teach pendant we displaced the screen and the electronics boards in the upper and bottom part of the chassis. We prepared customized power cables and USB cables in order to save space and to keep the assembly more ordered.

In the Fig. 5.1 it is possible to see the teach pendant prototype disassembled with the components placed in their final location. Thus, the Raspberry pi 3, the Arduino Nano, the touch sensible screen with its display controller in the middle of the bottom part of the chassis, the touch controller plate and the battery. There are also the emergency button and the two encoders in the right upper part of the chassis. In particular, the emergency button has the function to disarm the robot actuation system and the encoders to precisely adjust the sliders position when controlling the robot's joints.

In Fig. 5.2 we can see the final prototype of the teach pendant fully assembled. The device can be easily grasped using the left hand thanks to the hand holder located on the frontal left side.

We also tested the teach pendant software and the connection with the biped robot. We fully verified the calculation and visualization of FK. Also, the IK was partially verified by software but not with the robot's hardware. We implemented the IK so that the program would choose the closest solution to the current posture and ignores the other solutions. In case of negative and positive angles which are at the same distance from the current robot position then the first solution is chosen and the other ignored. For example, compare the Fig 5.3 and Fig. 5.4.

We updated the Teach Pendant's GUI to include the tab for the IK. Now, we can provide both the position and orientation inputs; as a result we get the 6 joint angles and we can also see the frames for these 6 joints on the right side . The first three fields are for the 3 positions and the last 3 fields are for the α, β, γ angles.

Finally, we also tested the teach pendant with the NU-Biped actuation system, in particular we controlled some of the leg joints position in real time.

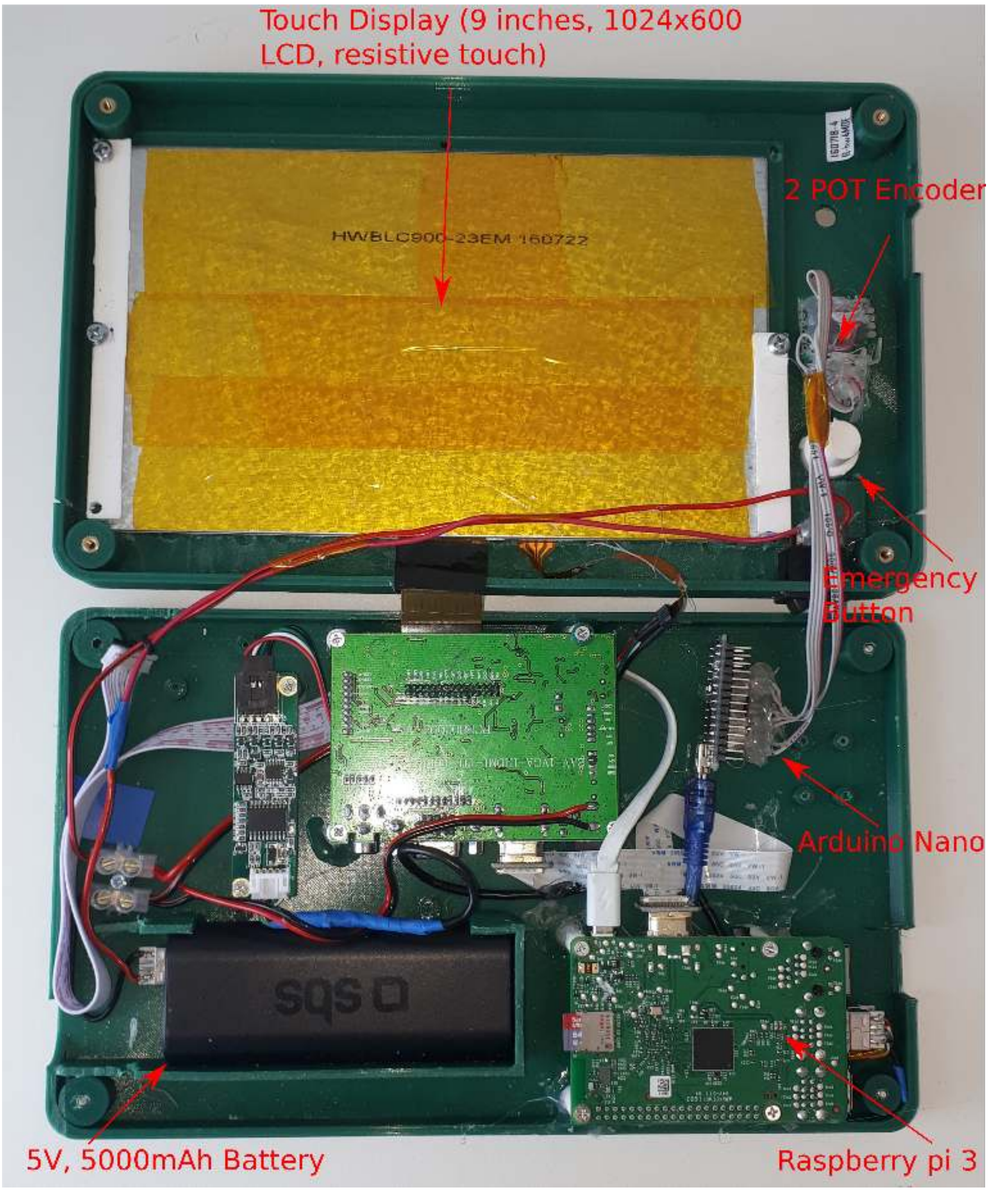


Figure 5.1: NU-Biped Teach Pendant disassembled

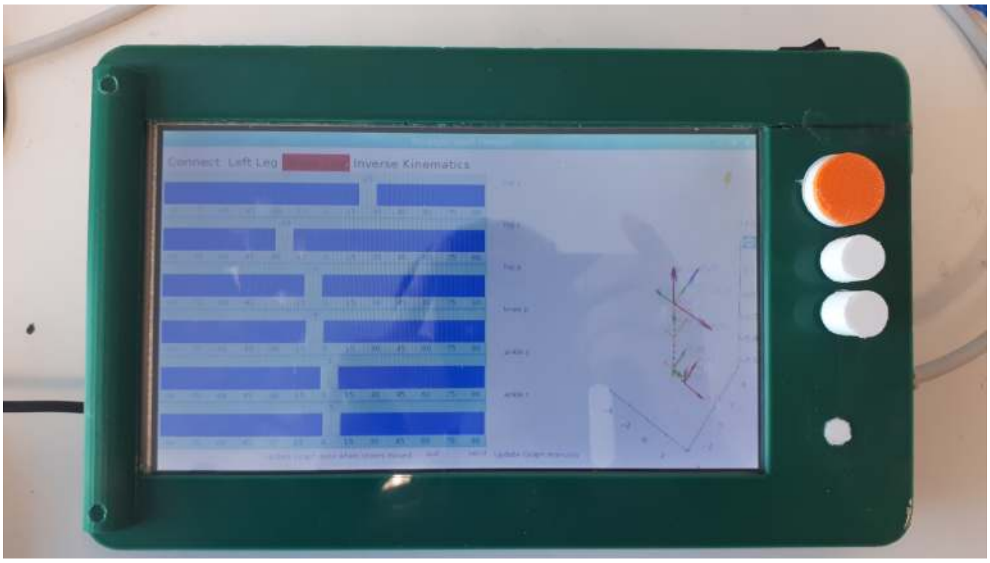
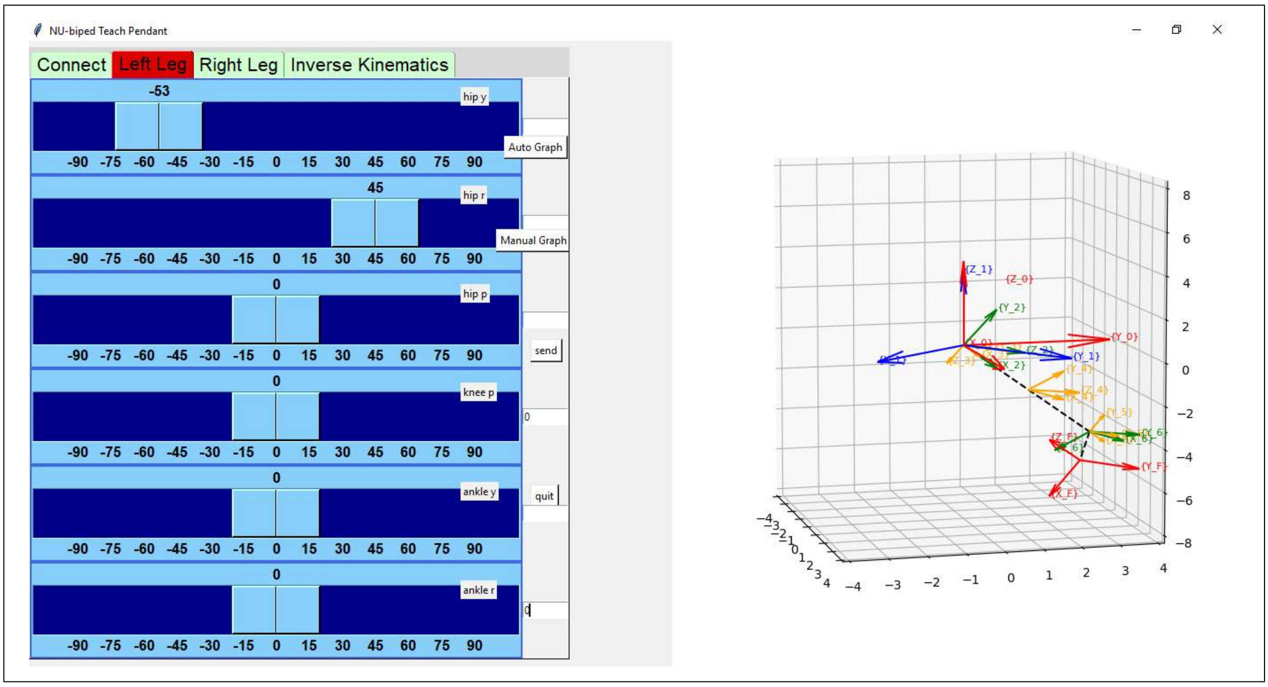


Figure 5.2: NU-Biped Teach Pendant assembled



hip y -53

hip r 45

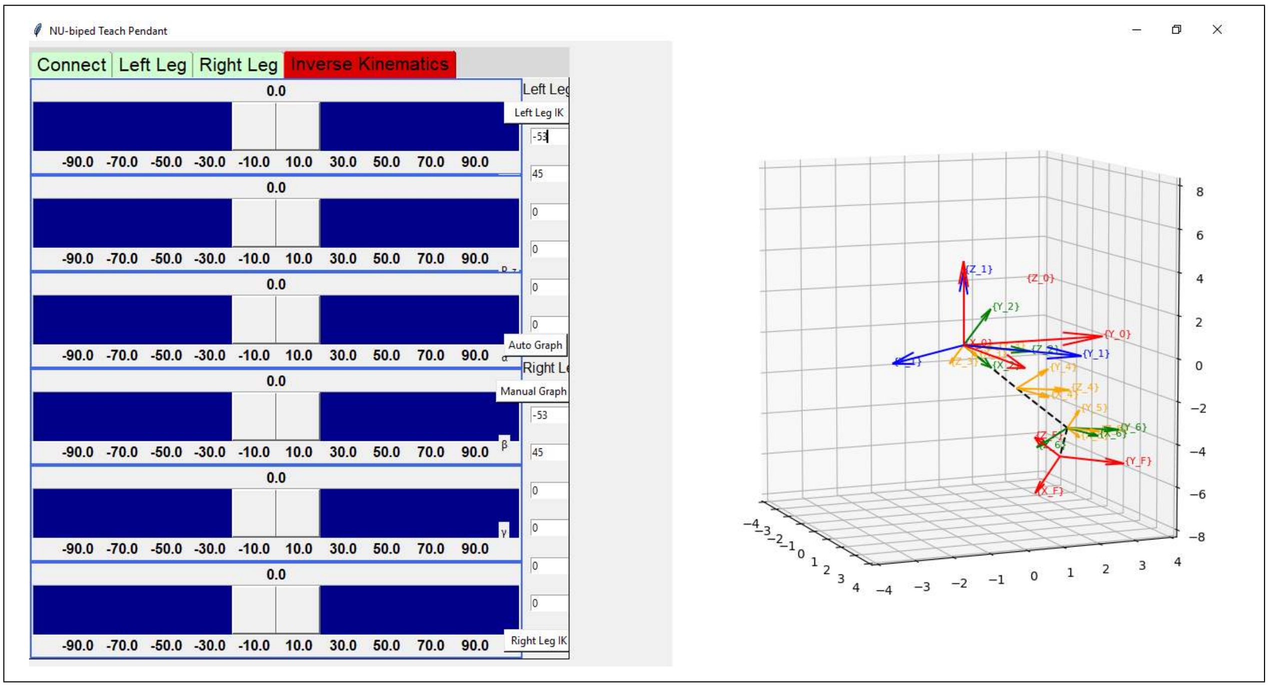
hip p 0

knee p 0

ankle y 0

ankle r 0

Figure 5.3: Visualization of the Forward Kinematics



0.0

0.0

0.0

0.0

0.0

0.0

Left Leg IK: -53, 45, 0, 0, 0, 0

Right Leg IK: -53, 45, 0, 0, 0, 0

Figure 5.4: Starting from Inverse Kinematics we calculate the angles and then do Visualization

Chapter 6: **Conclusion and Future Work**

In conclusion, we have implemented and verified the FK calculation and its visualization for the NU-Biped robot. We have designed and built the prototype of the Teach Pendant. We implemented and tested an intuitive GUI to control the robot, both in the joint and Cartesian space. We verified the Robot and Teach Pendant connection. We tested the teach pendant with a Dynamixel motor that is the same motor as in the hip joint of the robot. Overall, we think that NU-biped teach pendant maybe useful also for other robots. The program can be easily extended to increase the number of joints. The low cost design and realization is one of the notable advantages of our system. The kinematics architecture, forward, and inverse kinematics calculations make the robot control with teach pendant more efficient. The Cartesian and joint space position management using python Tkinter GUI is favourable for its fast time control and comfortable design.

In Future we could include in the Teach Pendant other software modules to send more high level commands to the robot, e.g walk autonomously, avoid obstacles, etc. Furthermore, we can extend the interface to control also the hands of the future NU-Humanoid Robot.

Bibliography

[1] M. Folgheraiter, "Design and modeling of a lightweight and low power consumption full-scale biped robot.," *International Journal of Humanoid Robotics*, vol. 15, no. 5, 2018.

[2] A. M. A. H. A. Park and C. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots.," *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE*, 2010.

[3] K. S. et al., "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation.," *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on.*, vol. 1, 2001.

[4] T. Lung-Wen, "Robot analysis: the mechanics of serial and parallel manipulators," *John Wiley and Sons*, 1999.

[5] C. J. J., "Introduction to robotics: mechanics and control.," *Upper Saddle River, NJ, USA*, vol. 3, 2005.

[6] S. B. et al., "Robotics: modelling, planning and control," *Springer Science* & *Business Media*, 2010.

[7] K. K. W. et al., "Robotic system with teach pendant.," *U.S. Patent*, 6 May 2003.

[8] S. G. S. et al., "Bluetooth communication using a touchscreen interface with the raspberry pi.," *Southeastcon, 2013 Proceedings of IEEE*, 2013.

[9] S.-U. L. Oh, Sung-nam and K. ll Kim, "Bluetooth communication using a touchscreen interface with the raspberry pi.," *International Journal of Control, Automation, and Systems*, vol. 1, no. 2, pp. 252–256, 2003.

**

Appendix A: **IK Alternative Solution**

Another approach to find the θ_5 and θ_6 will be the following: (but the results are the same)

In the above expression atan2() a function works similar atan which returns $\frac{sin()}{cos()}$, but atan2() adjusts the quadrant properly. By squaring equations (1) and (2), adding them we can eliminate θ_6 and end up with:

$(c_5(c_4L_3 + L_4) - s_4s_5L_3)^2 = \pm(p'_x)^2 + (p'_y)^2$ (4)

After expanding the eq. (3) we can get:

$-(s_5(c_4L_3 + L_4) + c_5(s_4L_3)) = p'_z$ (5)

Assume that $c_4L_3 + L_4 = rc_\psi$ and $s_4L_3 = rs_\psi$ after substituting those eqs. into the (4) and (5) we can determine that:

$rc_{5\psi} = \pm\sqrt{((p'_x)^2 + (p'_y)^2)}$ (6)

$rs_{5\psi} = -p'_z$ (7)

here $r = \sqrt{(p'_x)^2 + (p'_y)^2 + (p'_z)^2}$, and also $\psi = atan2(s_4L_3 + L_4)$. By dividing Eq. (6) to (7) we get that $tan(\theta_5 + \psi$, that can at the end give the solution for θ_5.

$$\boxed{\theta_5 = atan2(-p'_z, \pm\sqrt{(p'_x)^2 + p_y^2}) - \psi}$$

By dividing Eq. (2) to (1) we can obtain solution for θ_6:

$$\boxed{\theta_6 = atan2(p'_y, -px')}$$

———————————————— $T_3^6 *_6^0 T_{target}^{-1} =$
$\begin{bmatrix} c_6r'_{1x} - s_6n'_y & c_6r'_{2x} - s_6s'_y & c_6a'_x - s_6a'_y & px_{lhs} \\ s_6r'_{1x} + c_6n'_y & s_6r'_{2x} + c_6s'_y & s_6a'_x + c_6a'_y & py_{lhs} \\ c_1c_2s_{345} + s_1c_{345} & s_1c_2s_{345} & s_2s_{345} & pz_{lhs} \\ 0 & 0 & 0 & 1 \end{bmatrix}$ At this point the right side of Eq.3.13 is completely known

$px_{lhs} = c_6p'_x - s_6p'_y$

$py_{lhs} = s_6p'_x + c_6p'_y$

$pz_{lhs} = -s_{45}L_3 - s_5L_4$

While the right side of equation Eq. 3.12 is

$T' =$

$$\begin{bmatrix} c_1c_2c_{345} - s_1s_{345} - s_1s_{345} & s_1c_2c_{345} + c_1s_{345} & s_2c_{345} & px_{rhs} \\ -c_1s_2 & -s_1s_2 & c_2 & 0 \\ c_1c_2s_{345} + s_1c_{345} & s_1c_2s_{345} - c_1c_{345} & s_2s_{345} & pz_{rhs} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $r_{14} = -c_{45}L_3 - c_5L_4$

$r_{34} = -s_{45}L_3 - s_5L_4$

By equating (2,3) elements of the right hand side and left hand side, it is possible us to find c_2 and s_2, so that using atan2() we can determine θ_2:

$s_6a'_x + c_6a'_y = c_2$

$s_2 = \pm\sqrt{1 - (s_6a'_x + c_6a'_y)^2}$

$$\boxed{\theta_2 = atan2(s_2, c_2)}$$

By equating (2,1) and (2,2) of the left hand side and right hand side of G_2, we can determine θ_1:

$s_6r'_{1x} + c_6n'_y{=}{-}c_1s_2$ (8)

$s_6r'_{2x} + c_6s'_y = -s_1s_2$ (9)

by squaring both sides of (1) we may get:

so that,by dividing these two equations we may obtain:

$$\boxed{\theta_1 = atan2(-s_6s'_x - c_6s'_y, -s_6n'_x - c_6n'_y)}$$

when s_2 is less than 0 , then $\theta_1 = \theta_1 + \pi$

By equating (1,3) and (3,3) elements of the left hand side and right hand side matrices one could obtain:

$c_6a'_xs_6a'_y = s_2c_{345}$ (10)

$a'_z = s_2s_{345}$ (11)

by dividing equations (10) and (11) we may obtain: $t_{345} = atan2(a'_z, c_6a'_x - s_6a'_y)$

and when $s_2 < 0$ then we should make $t_{345} = t_{345} + \pi$. At the end, from t_{345} by subtracting θ_4 and θ_5 we may get:

$$\boxed{\theta_3 = t_{345} - \theta_4 - \theta_5}$$

Appendix B: **SOFTWARE**