EMG Controlled Exoskeleton Leg Mechanism

23.07.2019

Projects

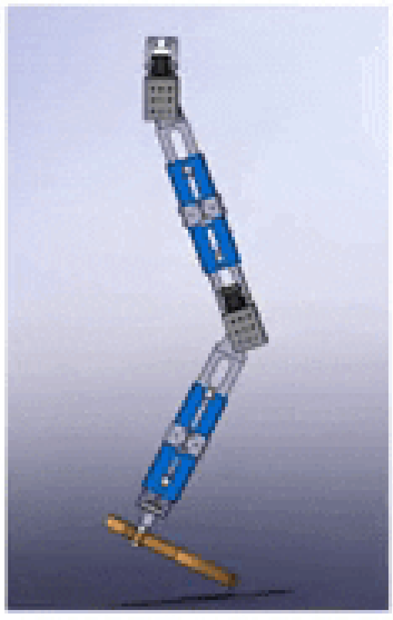
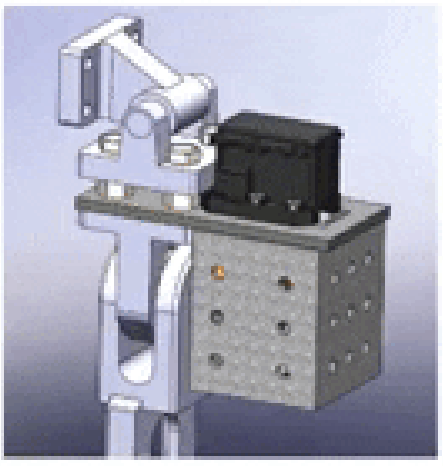
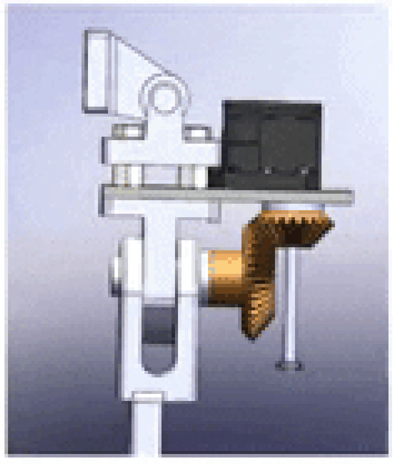
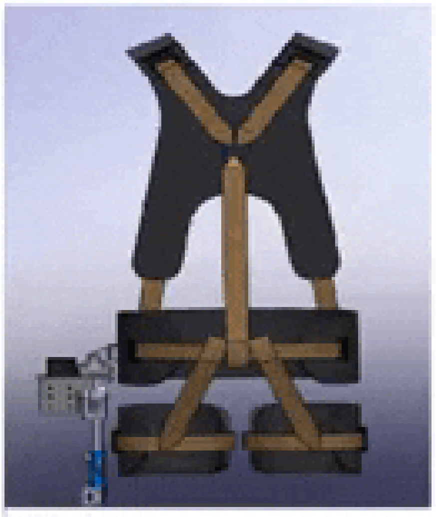
Note: This article is an abbreviation of my graduation thesis report. Contact me for more information.

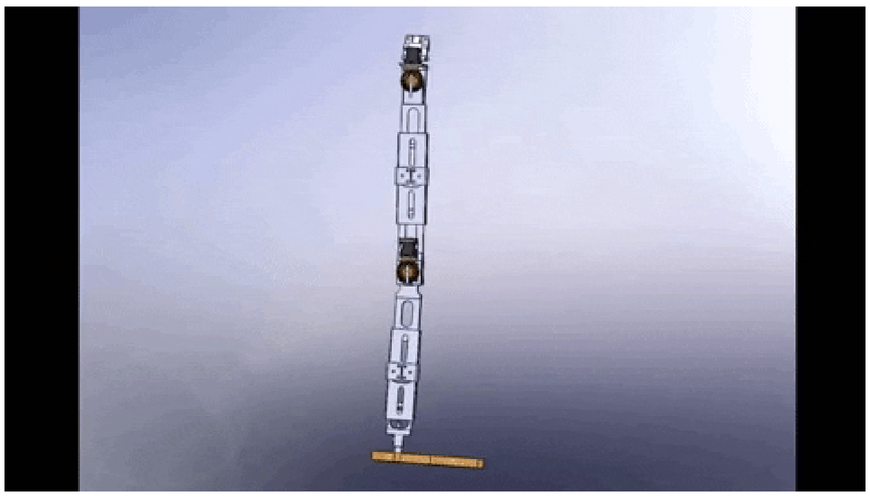
1. PROJECT GOAL & MOTIVATION

The aim of this project is to build a stable system that is able to understand the intention of the user by processing the electromyography (EMG) signals taken from the user's arm and help the movement being performed. The functional relationship between the EMG signals and the exoskeleton leg mechanism is intended to contribute to the user movement. Basically, the natural movement of the human leg is to be imitated in a functional way.

The primary motivation for the project process was to increase the living standards of elderly people and/or people who have lower limb-related disabilities. Besides, the exoskeleton mechanism can be used to improve the working conditions of industrial production line workers and to minimize occupational deformations and diseases.

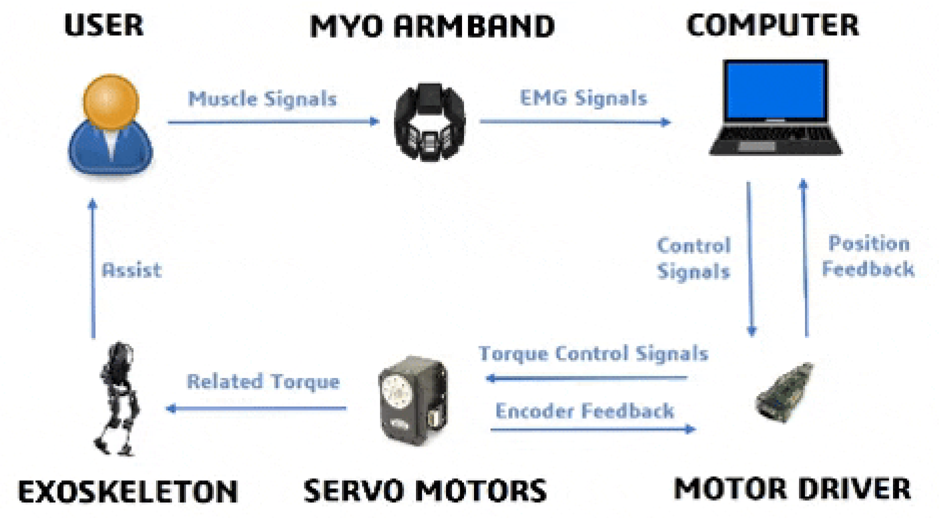
2. MECHANICAL DESIGN



3. HOW SYSTEM WORKS

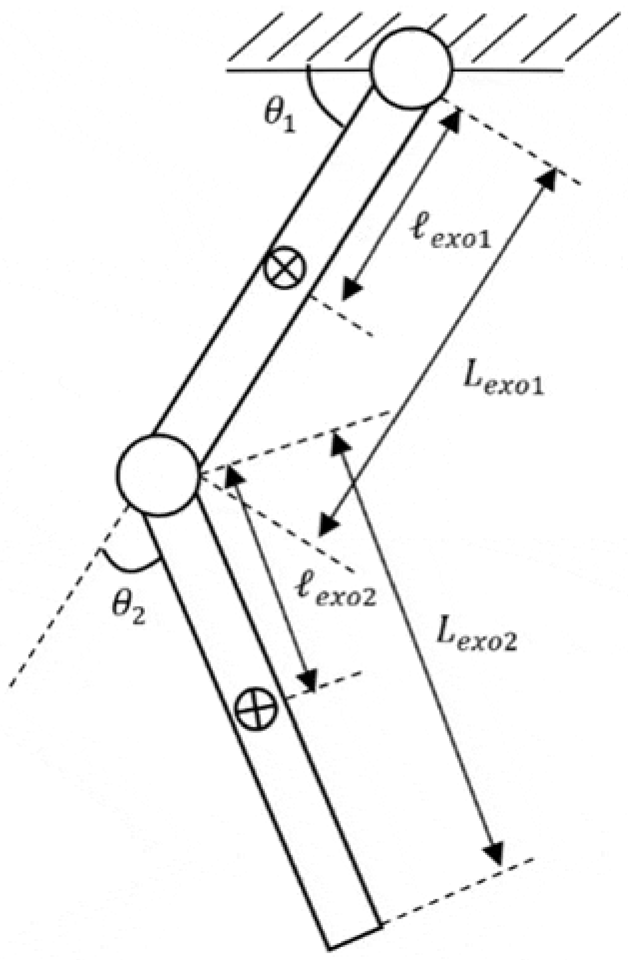
In the figure below, the detailed working principle of the system is shown.



Muscle signals are received from the user's arm with the help of Myo Armband which contains 8 different EMG electrodes, then the signals are sent to a computer where an artificial neural network and a motor driving algorithms have been embedded. At this point, with the help of the neural network algorithm the system classifies the received EMG signals as flexion, extension, fist and relaxation, then, related actuation signals are sent to the motor driver circuit which will drive the servo motors located in hip and knee joints so that the specified action (normal speed walk, slow walk, stop, etc.) takes place. By the feedback mechanism, the rotary encoder provides instant angle-position data of the leg mechanism

4. EQUATIONS OF MOTION

In this section the mathematical model which shows the behaviour of the system will be given. Note that, the rigid double pendulum model shown in the figure below will be used to represent the leg mechanism.



The closed-form equations of motion of a double pendulum are derived using Lagrange's equations of motion. Each link of the pendulum is treated as a rigid body, defined by its mass and the position of the center of mass with respect to its corresponding joint; for example, *r1* is the distance of the center of mass of link 1 with respect to joint 1.

Here is the obtained equation of motion in the following vector form. Note that, M(q) is inertia matrix, V(q,q') is Coriolis/centripetal vector, and G(q) is gravity vector.

$$M(q)\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}+\begin{bmatrix}-m_2L_1\ell_2(2\dot{\theta}_1\dot{\theta}_2+\dot{\theta}_2^2)\sin\theta_2\\ m_2L_1\ell_2\dot{\theta}_1^2\sin\theta_2\end{bmatrix}+\begin{bmatrix}-(m_1\ell_1+m_2L_1)g\cos\theta_1-m_2g\ell_2\cos(\theta_1+\theta_2)\\ -m_2g\ell_2\cos(\theta_1+\theta_2)\end{bmatrix}=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}$$

where

$$M(q)=\begin{bmatrix}m_1\ell_1^2+m_2L_1^2+I_{cm_1}+m_2\ell_2^2+2m_2L_1\ell_2\cos\theta_2 & m_2\ell_2^2+m_2L_1\ell_2\cos\theta_2\\ m_2\ell_2^2+m_2L_1\ell_2\cos\theta_2 & m_2\ell_2^2+I_{cm_2}\end{bmatrix} \tag{3}$$

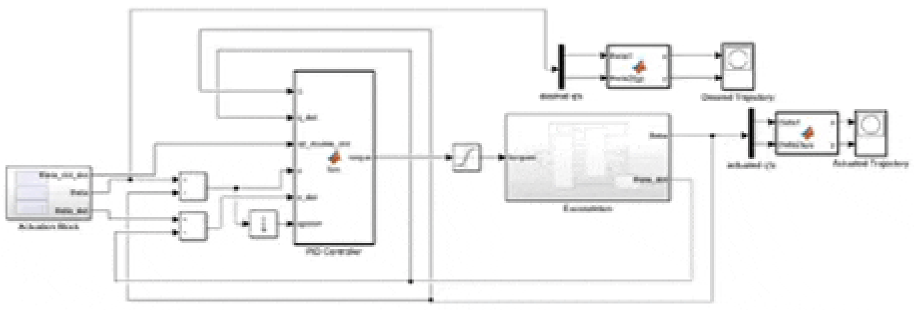
Therefore, the equation can be constituted in the standard forms as [12]:

$$M(q)\ddot{q}+V(q,\dot{q})+G(q)=\tau \tag{4}$$

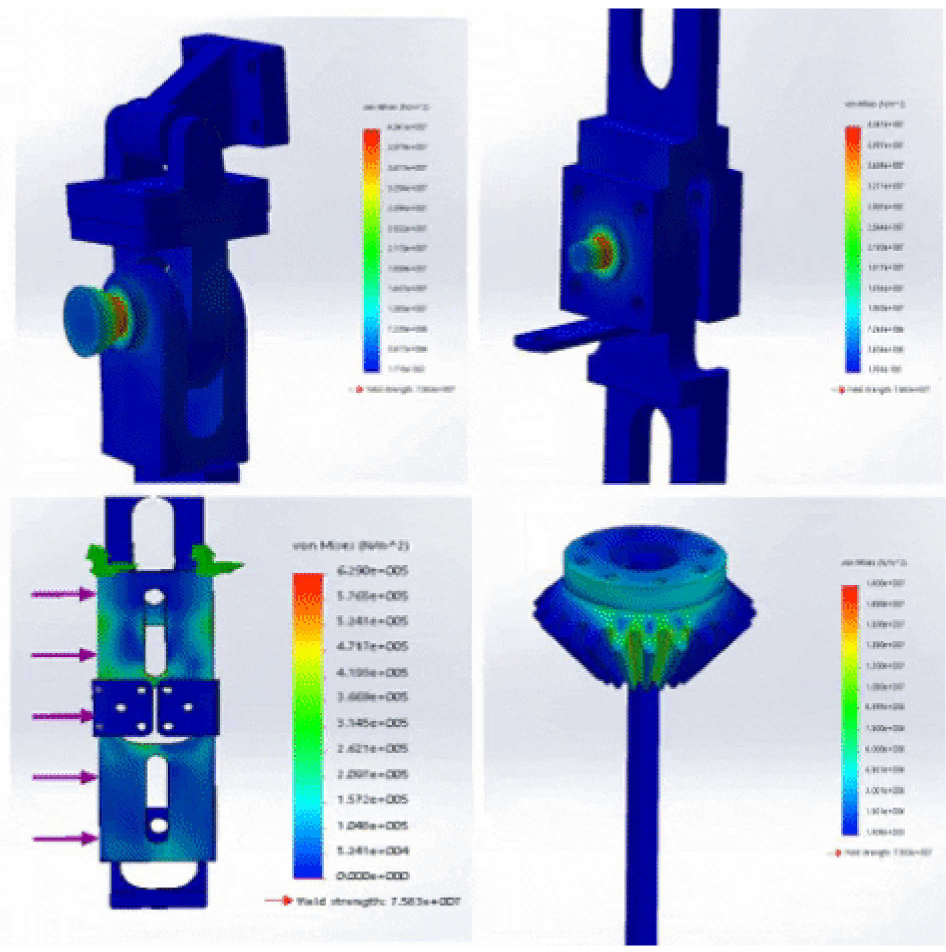
where *M(q)* is the inertia matrix (symmetric), *V(q,q')* is the Coriolis/centripetal vector, and *G(q)* is the gravity vector. The constituent elements of given vectors are mass (*m1, m2*), total length (*L1, L2*), distance of center of mass from the proximal end (*l1, l2*), and inertia values (*Icm1, Icm2*) of upper and lower parts, respectively, in the exoskeleton system.

The obtained system equations are transferred to MATLAB program and thereby, mathematical system model for the exoskeleton system is created in Simulink environment. After that, a standard PID controller with feedback linearization is utilized.

5. ENTIRE SYSTEM SIMULINK MODEL

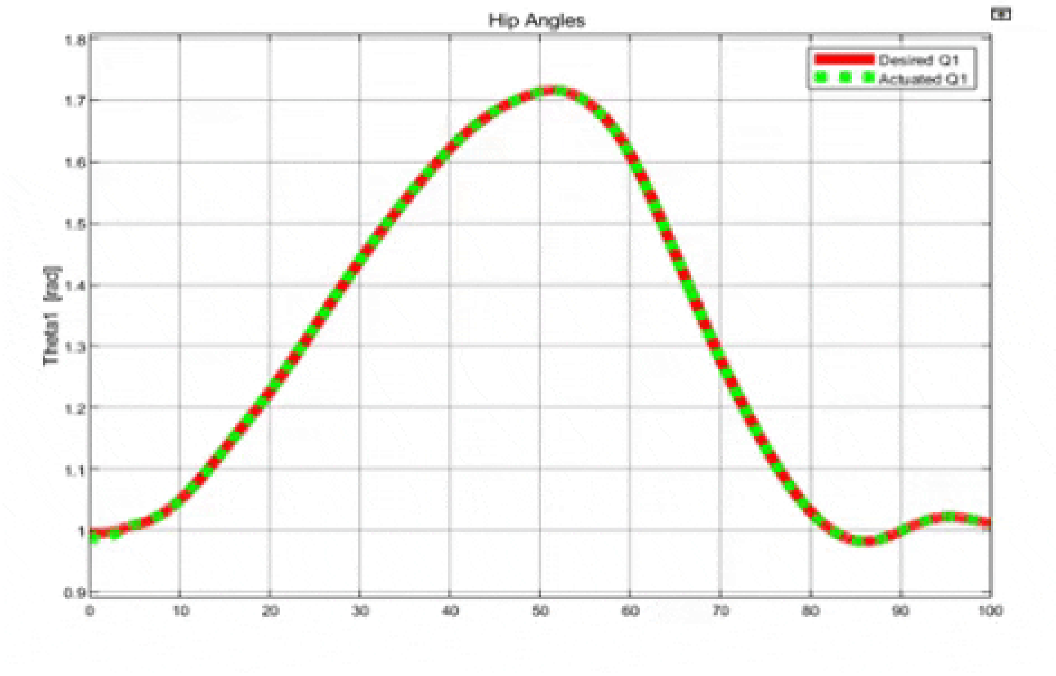


6. STRESS ANALYSIS

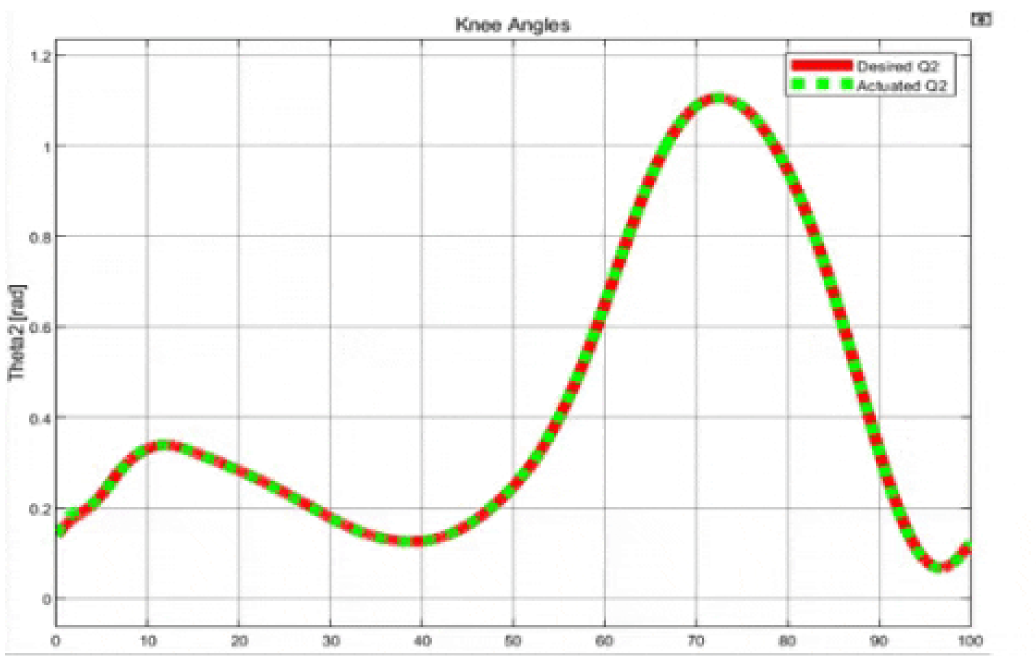


7. SIMULATION RESULTS

Desired vs Actuated Hip Joint Angles



Desired vs Actuated Knee Joint Angles

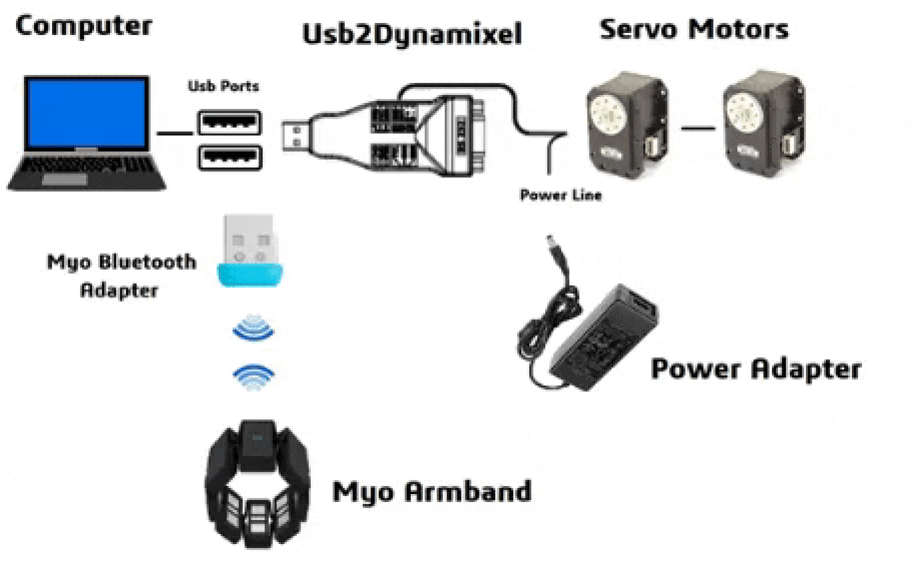


8. MECHANICAL SYSTEM

The body material has been cast from ETIAL 160 as blocks in the shape of the rectangular prism. Then they have been reformed via CNC milling and drilling. The entire mechanical system is shown in the figure below.

9. ELECTRONIC SYSTEM

Connections between electronic components are illustrated in the figure below.



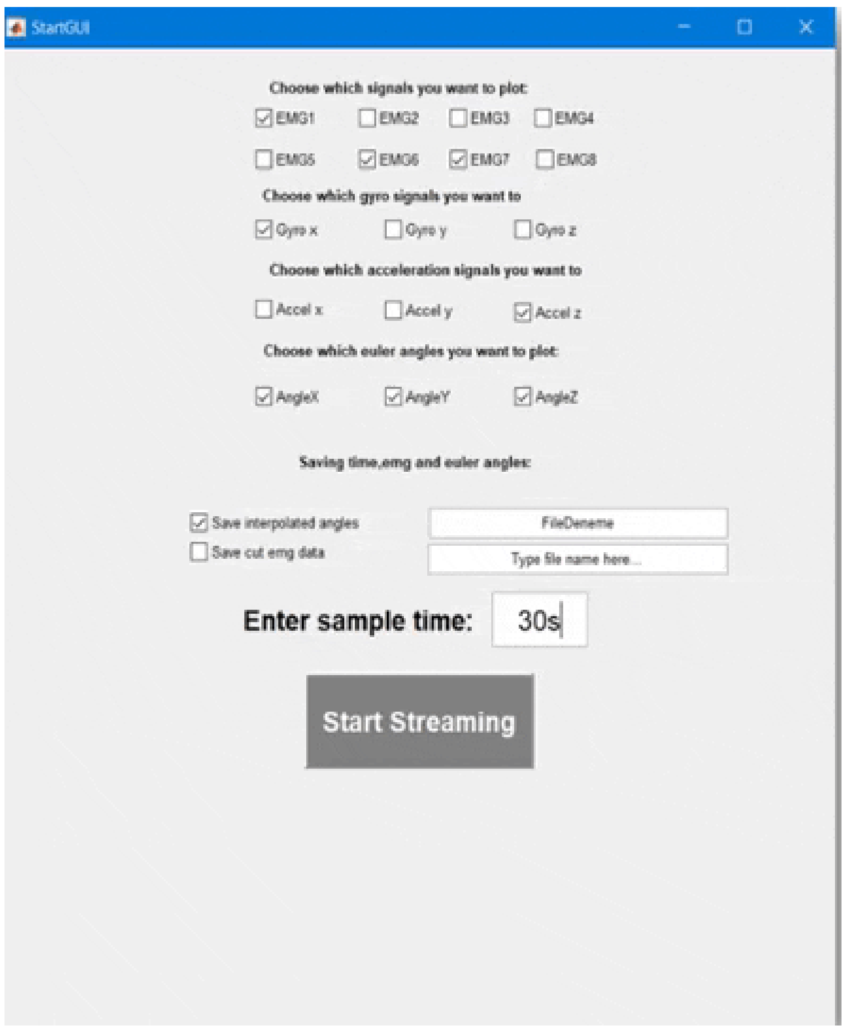
10. SOFTWARE IMPLEMENTATION

This section is divided into 5 subheadings namely **Data Acquisition from Myo Armband**, **Signal Pre-processing**, **Channel Selection**, **Neural Network Structure, Motor Driving**. Firstly, in order to obtain the training set needed for the neural network application, a MATLAB based graphical user interface has built. Then acquired siganls are pre-processed. Later the neural network algorithm is implemented for the real-time classification of different hand gestures based on received EMG data.

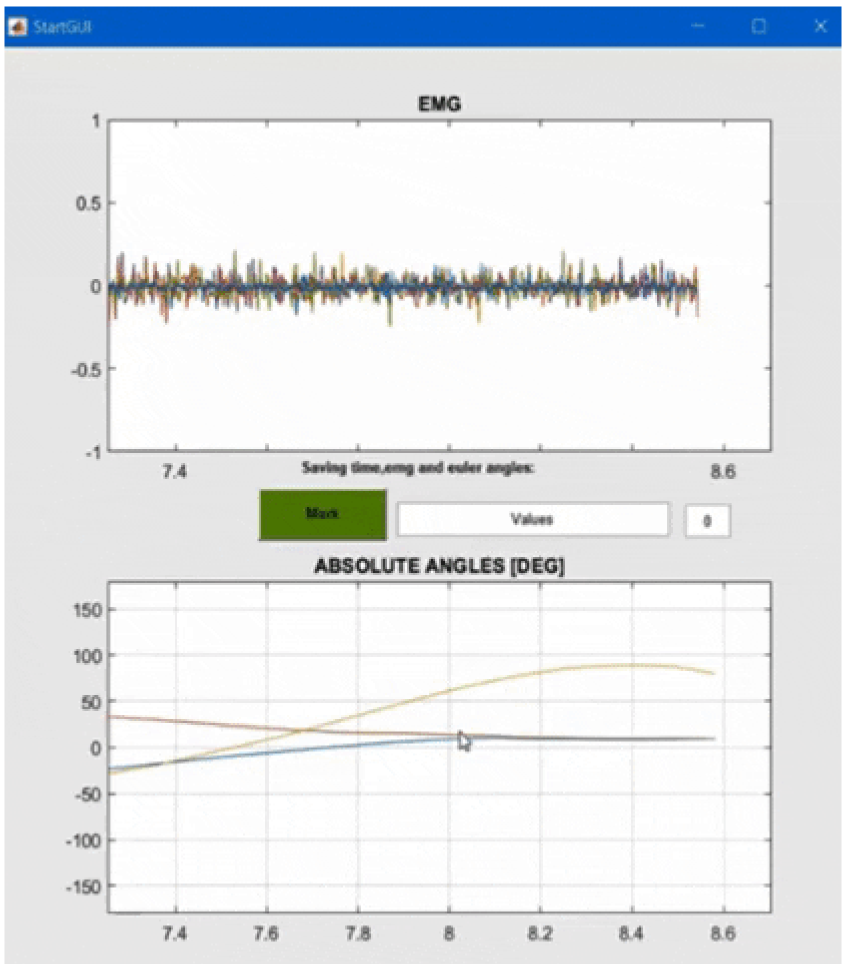
a. Data Acquisition from Myo Armband

In this study, Myo Arm Band (Thalmic Lab) was used to obtain EMG data from the surface. Myo Armband has 8 stainless steel EMG sensors on it. The device sends the EMG signals with 8-bit resolution and 200 Hz sampling frequency via Bluetooth. In addition to these, Myo Armband has 3 axis gyroscope and 3 axis accelerometer which are working with 50 Hz sampling frequencies.

A MATLAB based graphical user interface is built to acquire real time data from the Myo Armband. The GUI will be used to create a training set for the neural network algorithm.**.** In the figure below, a screenshot of the MATLAB-GUI is shown:



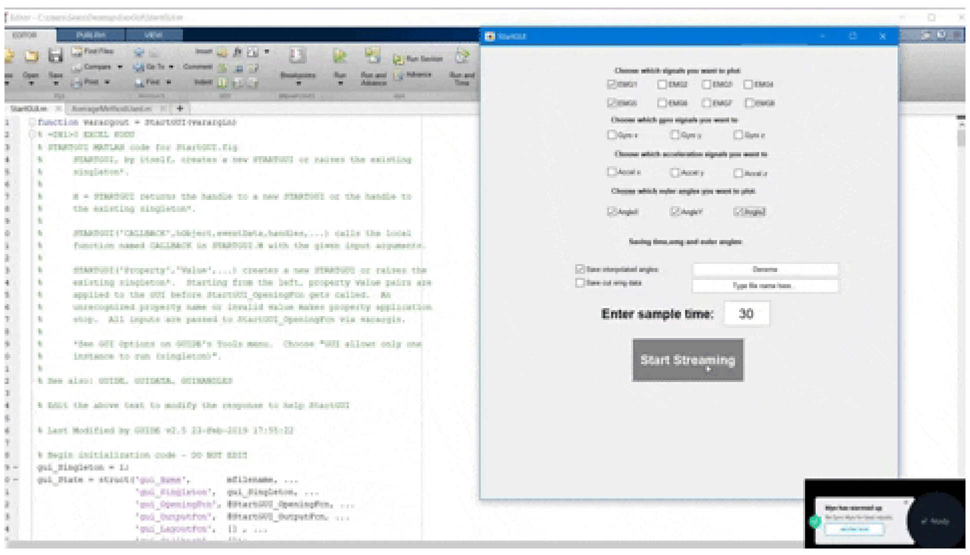
User can select which data he wants to plot just by clicking the related checkboxes. The user can plot and save separately all of the EMG data received from the 8 electrodes of the Myo Armband. Also, acceleration, gyroscope and euler angles data can be plotted or saved in the same way. Note that the euler angles are formed via integration of gyroscope data. In the figure below, screenshot of the interface is shown while real-time data is streaming.



The green mark button is used to mark specific data points, hereby signal preprocessing will be easier after saving the data. Note that, euler angles data has not been used in the scope of this study but in future studies, it is planned to determine the user intention using Myo Armbands placed on the user's leg and this euler angles data is expected to be useful in this process.

In short, the data acquisition interface has been used to obtain data needed to train a neural network. Also, by obtaining euler angles, movement trajectories of any user can be constituted. Later, obtained movement trajectories can be stored in the device memory and mimic by the leg mechanism to assist the user. Besides current position of the user can be determined by using these euler angles data.

`Data Acquisition GUI - MATLAB`

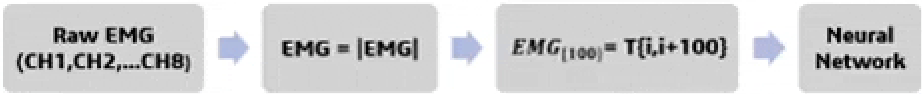


b. Signal Pre-processing

EMG signal contains a lot of noise for various reasons such as: surface temperature, thickness between muscle and skin, the disruption of the signal of the motor units. This is why EMG signals are subjected to certain pre-processing steps to achieve a robust result.

Unlike many other studies, no filtering process is applied in this study, which increases the speed of real-time operations as raw data will be used.

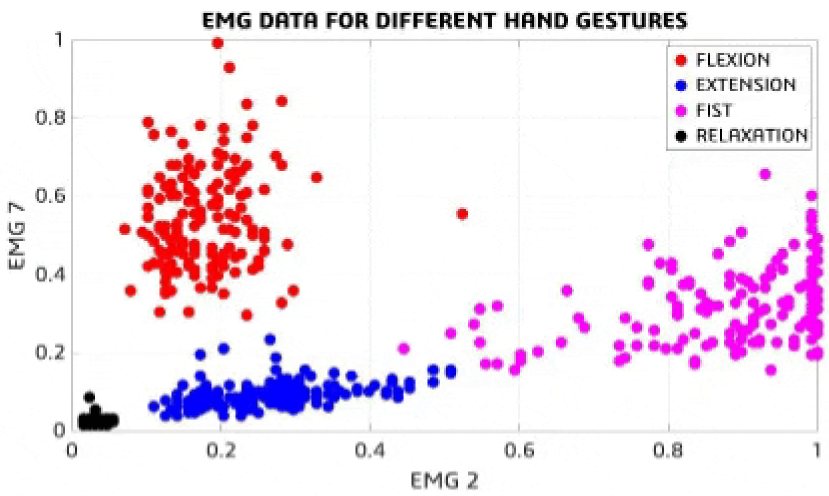
Pre-processing steps are given in Figure below, respectively.



Firstly, the magnitude of the received EMG values were calculated by taking the absolute values. Then, to remove the effect of each value from the system, the maximum values of the absolute values were taken for every 100 ms. The final values were fed to the neural network.

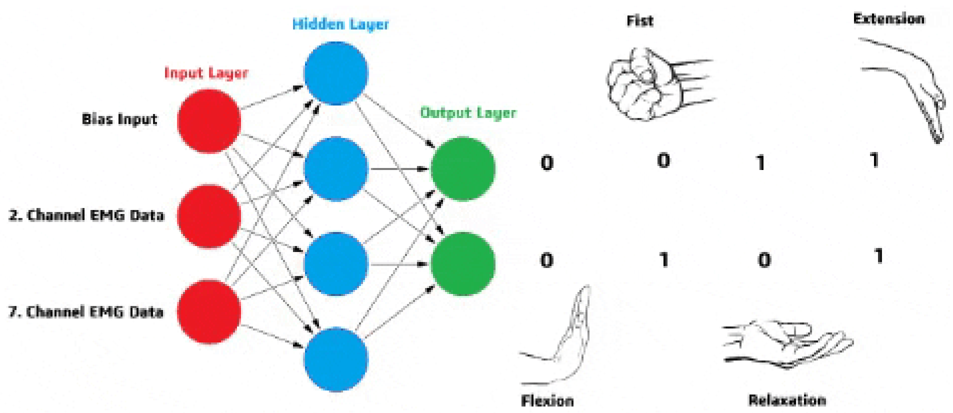
c. Channel Selection

In order to ensure stable and robust separation of hand movements and to increase speed for real-time applications, not all 8-channel EMG has been processed. The selection of the basic channels that can discriminate is carried out with cluster diagrams. The most discrete channels are designated as (2 and 7). Figure X shows the graph of this diagram. The chart contains a total of 600 data, 150 for each hand gesture. These data collected from a single person were used as a training set.



d. Neural Network Structure

Artificial neural networks are one of the main tools used in machine learning. As the “neural” part of their name suggests, they are brain-inspired systems which are intended to replicate the way that we humans learn. Neural networks consist of input and output layers, as well as (in most cases) a hidden layer consisting of units that transform the input into something that the output layer can use. They are excellent tools for finding patterns which are far too complex or numerous for a human programmer to extract and teach the machine to recognize.

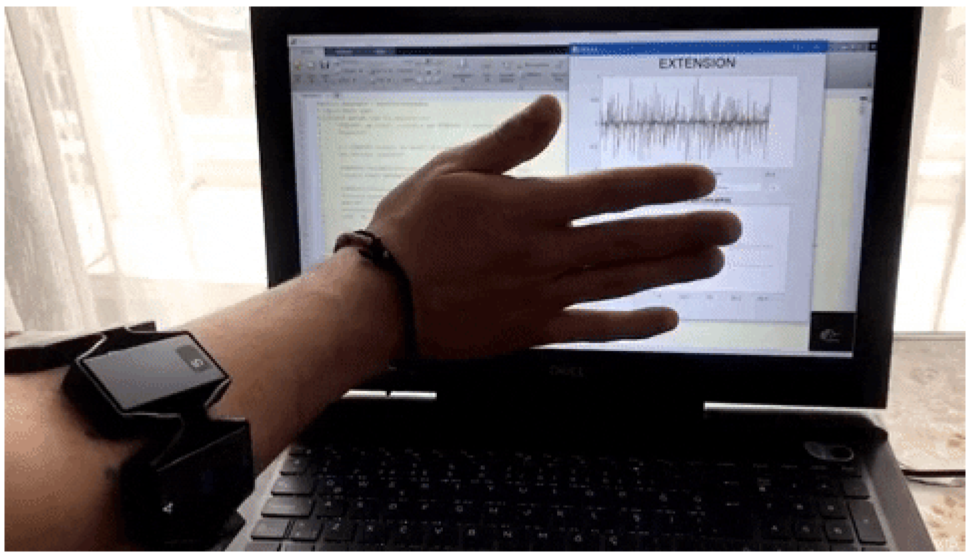


The corresponding neural network model is composed of 3 layers: the input layer, the hidden layer and the output layer. The input layer is fed by preprocessed EMG data. The sigmoid activation function used for all neurons. Thus the neuron outputs are between 0 and 1. The sigmoid activation function is given in equation below:

$$f(z) = \frac{1}{1 + \exp(-z)}$$

The values at the output of the neural network are trained in the value range [0,1]. Two output unit is used to classify 4 hand gestures, and training was performed according to these outputs. Levenberg-Marquardt algorithm was used as the training algorithm instead of back-propagation learning. Contrary to backpropagation learning, this algorithm can be used to train the neural network very quickly and optimally express the situations by changing the parameters.

Hand Gesture Classification via Neural Network Algorithm



e. Motor Driving

To drive motors and actuate the system. Robotis had launched a software development kit named Dynamixel SDK to write needed codes in various environments and languages.

The ROBOTIS Dynamixel SDK is a software development library that provides Dynamixel control functions for packet communication. The API is designed for Dynamixel actuators and Dynamixel-based platforms. It assumes that you are familiar with C/C++ programming. This e-Manual provides comprehensive information about ROBOTIS products and applications.

Dynamixel SDK also contains example codes available for each programming language and operating system. For this particular project, it is decided to use MATLAB. Keep in mind that, as an input, hip and knee paths obtained in the previous section are used.

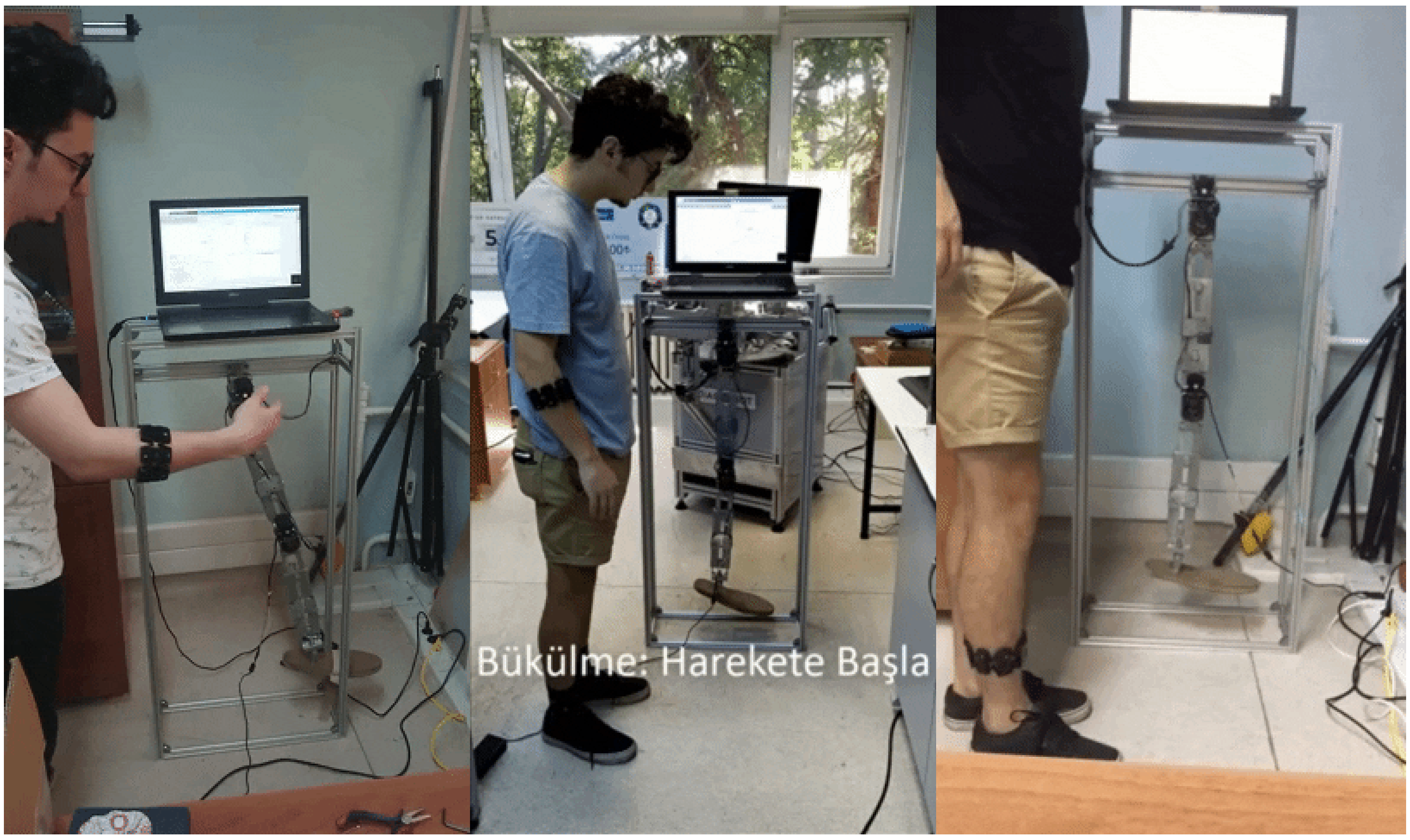
11. SYSTEM INTEGRATION

After the mechanical, electronic hardware along with the software implementation of the project tested to eliminate possible malfunctions, the systems are integrated to mechanism’s current stage called the demo mode.

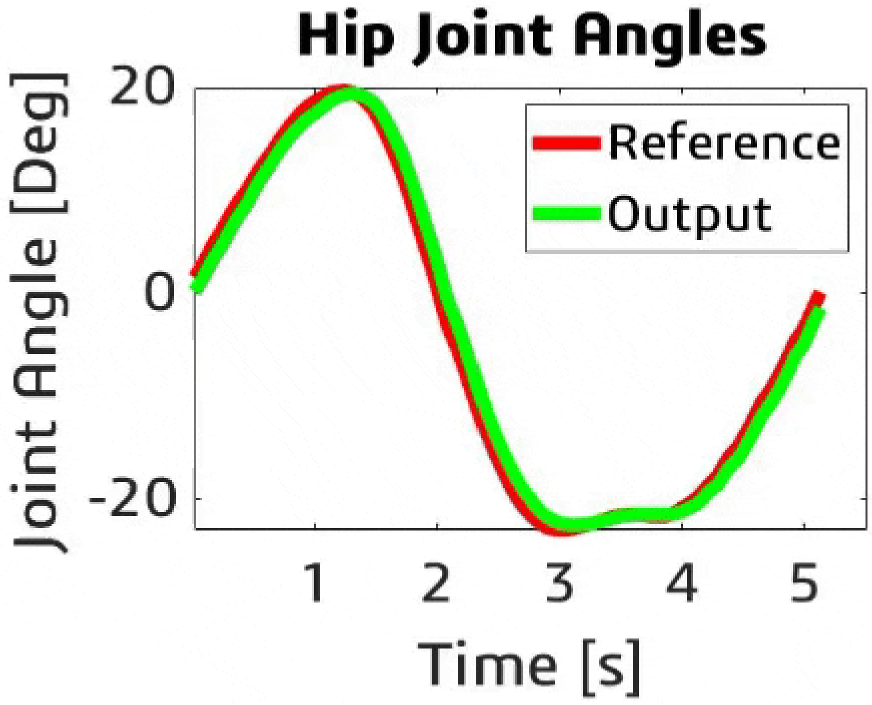
A 20 x 20 sigma chassis is produced to hold the mechanism and the mechanical body is hung on to it. Electronic components that contained needed software are placed in their places, and due to their physical locations on the exoskeleton, needed extension cables are done by manipulating the jumper and 3-pin cables. All of the electronic components except motors are placed on a wooden board attached to the top of the chassis for the ease.

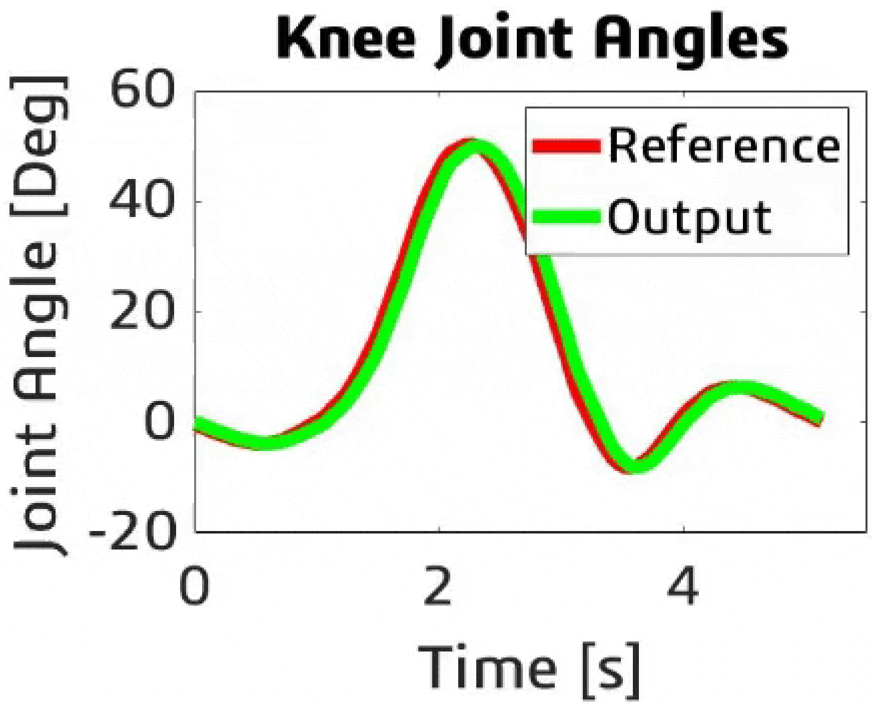
Later on, when the mechanism becomes a mobile industrial product, instead of a computer a Raspberry Pi 2 will be used and components will be placed inside of the manufactured waist mechanism. Also, it should not be forgotten that in the mobilized version of the exoskeleton, the power cells will be replaced with the Li-Po batteries with corresponding voltage and current values.

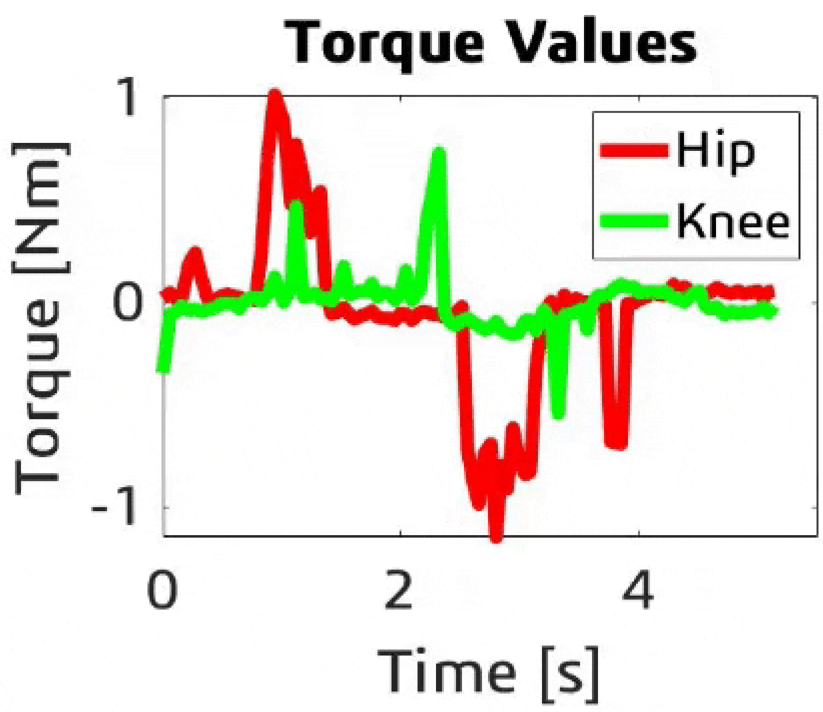
Demo Exoskeleton System - Test Step

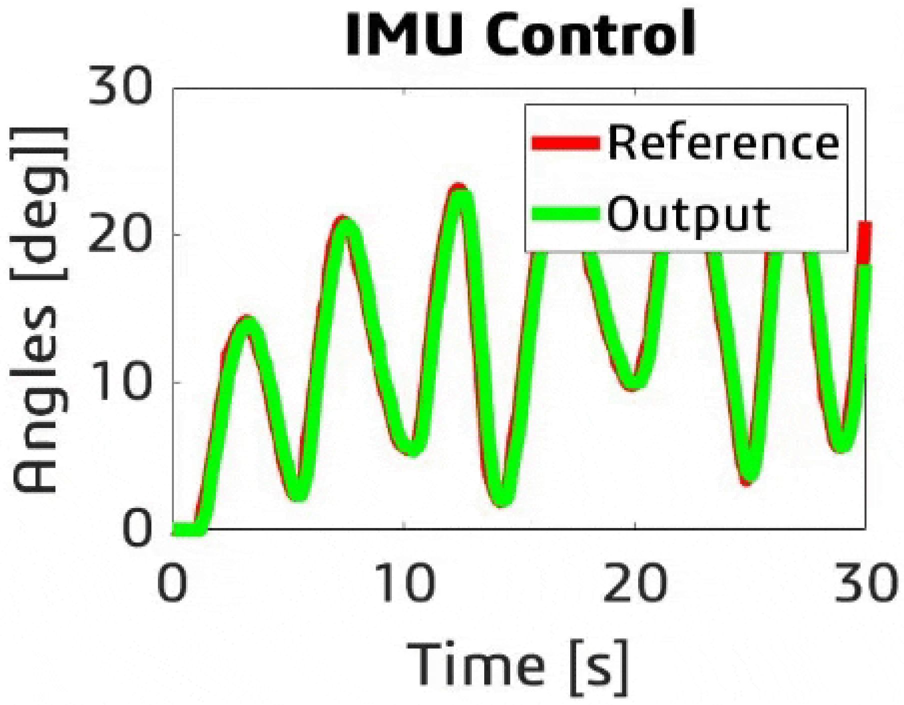


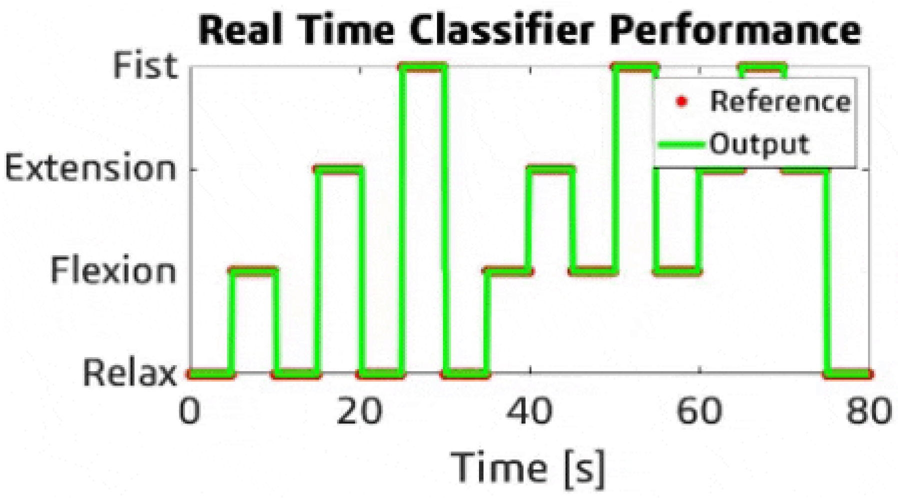
12. RESULTS


Hip Joint Angles
Joint Angle [Deg]
20
0
-20
1
2
3
4
5
Time [s]
Reference
Output


Knee Joint Angles
Joint Angle [Deg]
60
40
20
0
-20
0
2
4
Time [s]
Reference
Output


Torque Values
Torque [Nm]
1
0
-1
0
2
4
Time [s]
Hip
Knee


IMU Control
Angles [deg]]
Time [s]
Reference
Output
0
10
20
30


Real Time Classifier Performance
Fist
Extension
Flexion
Relax
Reference
Output
0
20
40
60
80
Time [s]

More blogs
