Mechanical Engineering Department

ME 655 – WS (BME 656-WS)

Wearable Robotics and Sensors

Virtual Lab Assignment III

Submitted by:

Rakesh Gummapu

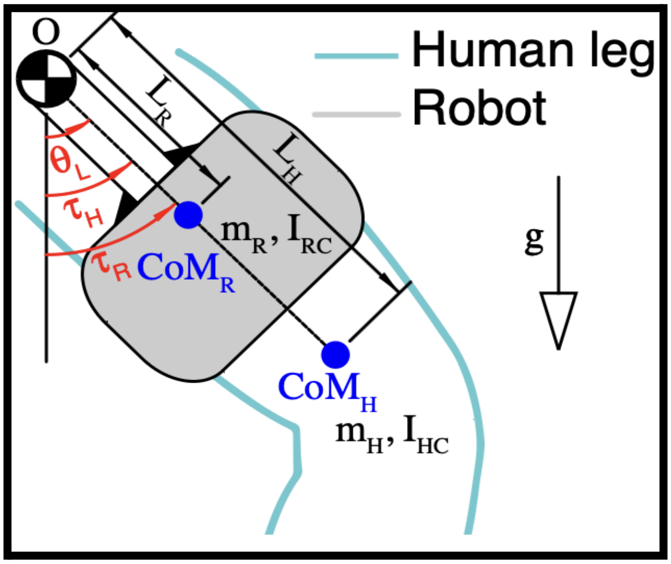
Peera Tienthong

Sahil Hadkar

Graduate Students:

"I pledge that I have abided by the Graduate Student Code of Academic Integrity"

Question:



The single-DOF exoskeleton as shown in the **Fig.1** is designed to assist the motion of the wearer's left hip joint in the sagittal plane during treadmill walking. Control of a 1-DOF hip exoskeleton attached to the left hip, assuming hip-joint O is externally supported and neglecting the ground reaction forces. A rigid frame supports the hip joint O of the robot, which is modeled as an ideal revolute joint (not shown in the picture). The wearer's leg, which is also modeled as a rigid bar rotating around the same revolute joint and has the following parameters: m_H, I_{HC} *and* L_H, is rigidly attached to the exoskeleton. With the aid of a torque generator, the combined effect of the hip extensor and flexor muscles is approximated. As shown in the figure, the robot's actuator is approximated by a rigid bar with mass m_R and a barycentric moment of inertia I_{RC}. The actuator is modeled as a pure torque generator.

Given:

The provided MATLAB/Simulink Files include:

I. Desired Trajectories for the left and the right hip joints (and their time derivatives) over 1 minute of treadmill walking

II. A simple model of the wearer's motor control system (for the left leg), which tracks a desired trajectory ($\theta_{L,des}$, $\dot{\theta}_{L,\,des}$, $\ddot{\theta}_{L,des}$)

Objective:

The main objective of this simulation-based assignment is to develop two assistive controllers for the robot to reduce the torque τ_H required by the wearer to track the desired trajectory by providing assistive τ_R using --

i. Model-based controller via adaptive frequency oscillators

ii. Model-free controller using waveform learning via nonlinear filter

To find:

PART A.

1. To generate the plot of the torque τ_H required to steer the system along the desired trajectory, assuming that the robot is passive $\tau_R = 0$.
2. Writing the inverse dynamics of the overall system and estimating the overall torque (Robot + human leg) required to guide the system along a given trajectory.
3. Designing a pool of *Adaptive Frequency Oscillators* (AFOs).
4. To develop a model-based assistive controller based on the inverse dynamics obtained in question 2.
5. To implement a *Kernel-based nonlinear filter.*
6. To use the nonlinear filter as a model-free assistive controller.

PART B.

7. To approximate the left hip joint angle and angular velocity using the *Complementary Limb Motion Estimation (CLME)* method.

System and Implementation:

1. To generate the plot of the torque τ_H required to steer the system along the desired trajectory, assuming that the robot is passive $\tau_R = 0$.

Code for data and parameters provided for Simulink model:

```
% ME655-WS(BME656-WS) - Virtual Lab Experience 3
% Group 7

clear all
close all
clc

% Create Simulation data
load('ME655_Lect09_VL3_Data.mat')
LH_hip_traj = timeseries([Lhip_pos Lhip_vel Lhip_acc Rhip_pos Rhip_vel Rhip_acc],t);

% initial conditions for human controller
Lhip_pos0=Lhip_pos(1);
Lhip_vel0=Lhip_vel(1);
Tau_h_max=30; %[Nm] max torque human can exert

%% Robot motor control

% Oscillator and filter settings:
eps=12;
nu=.5;
M=6;
lambda=.95;
```

```
N=80;
h=2.5*N;

t_start=5.0; %[s] time of AFO activation
dt_active_control=10; %[s] wait 10s after AFO is active before turning assistance ON
```

Results:

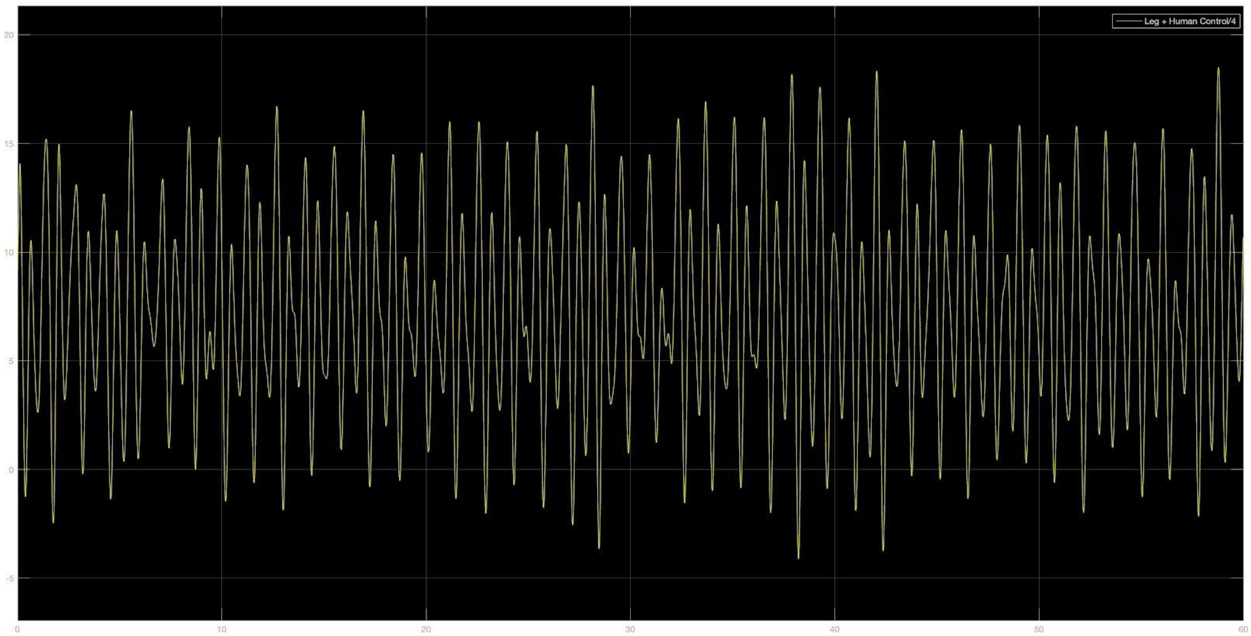


Fig 1.1: Torque trajectory of the left human-hip joint with no assistance from the robot over 60 seconds

Fig 1.2: The desired joint trajectory of the left-hip joint (Yellow) vs. the joint trajectory of the left-hip joint (Blue) over 60 seconds

Fig 1.3: the desired joint trajectory of the left-hip joint (Yellow) vs. the joint trajectory of the left-hip joint (Blue) in zoomed version over 60 seconds

The human motor control system can track the desired trajectory and they are similar.

2. Writing the inverse dynamics of the overall system and estimating the overall torque (Robot + human leg) required to guide the system along a given trajectory.

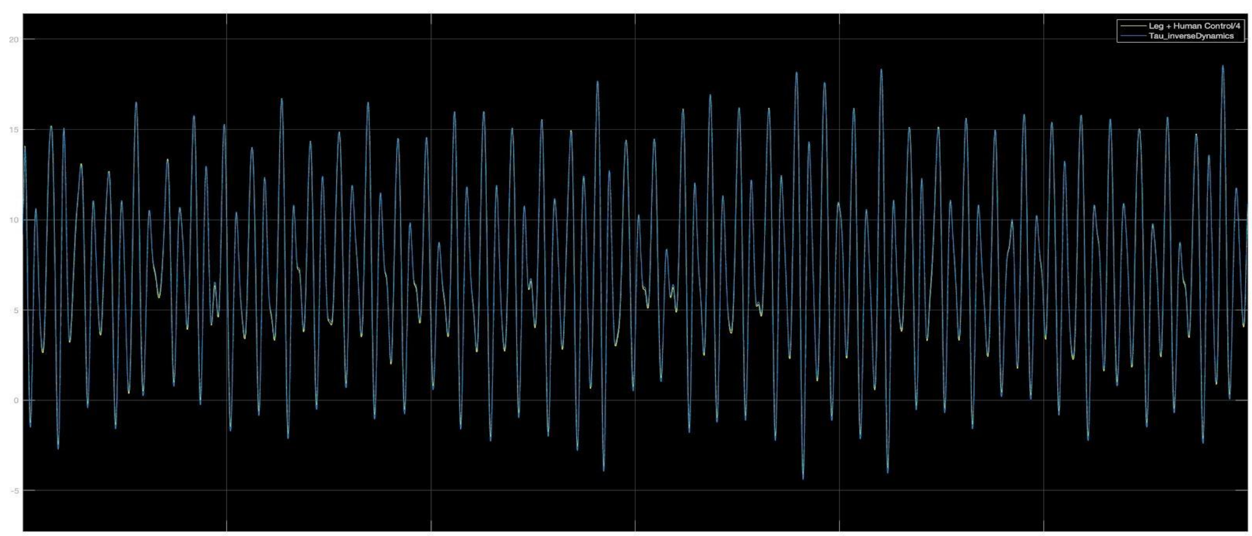


Fig 2.1: torque trajectory obtained from the system's inverse dynamics (Blue) vs. torque trajectory from the provided human-leg control (Yellow) over 60 seconds

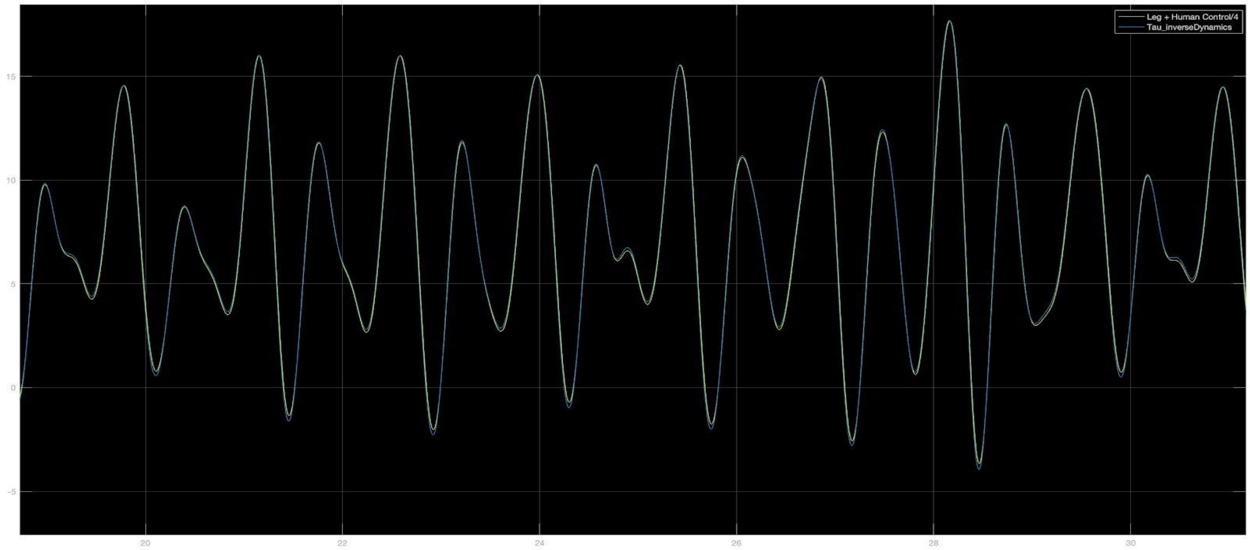
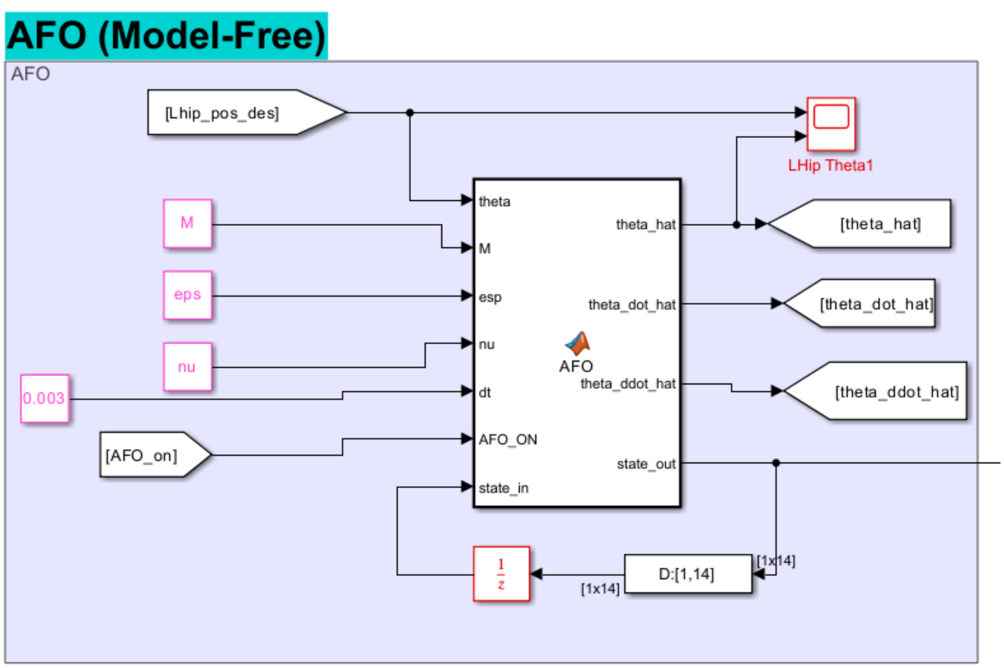


Fig 2.2: torque trajectory obtained from the system's inverse dynamics (Blue) vs. torque trajectory from the provided human-leg control (Yellow) in zoomed version over 60 seconds

The overall torque of the inverse dynamics output matches with the output of the torque trajectory from the provided human leg.

3. Designing a pool of Adaptive Frequency Oscillators (AFOs)

Simulink Model:



By using Simulink, the AFOs is designed and at time t = 5 sec the AFO is activated.

Results:

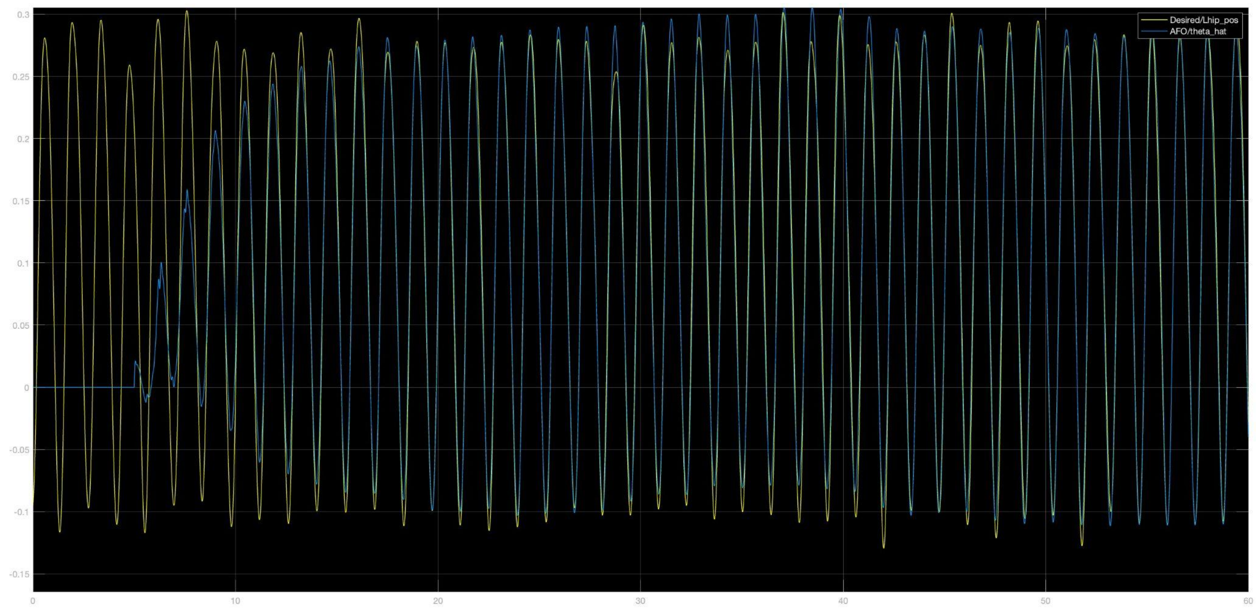


Fig 3.1: left-hip joint trajectory obtained from AFO(Blue) vs. provided left-hip joint trajectory (Yellow) over 60 seconds

AFO is activated at time t=5 sec so, there is a delay in the left hip trajectory for 5 sec and then it gradually increases until convergence.

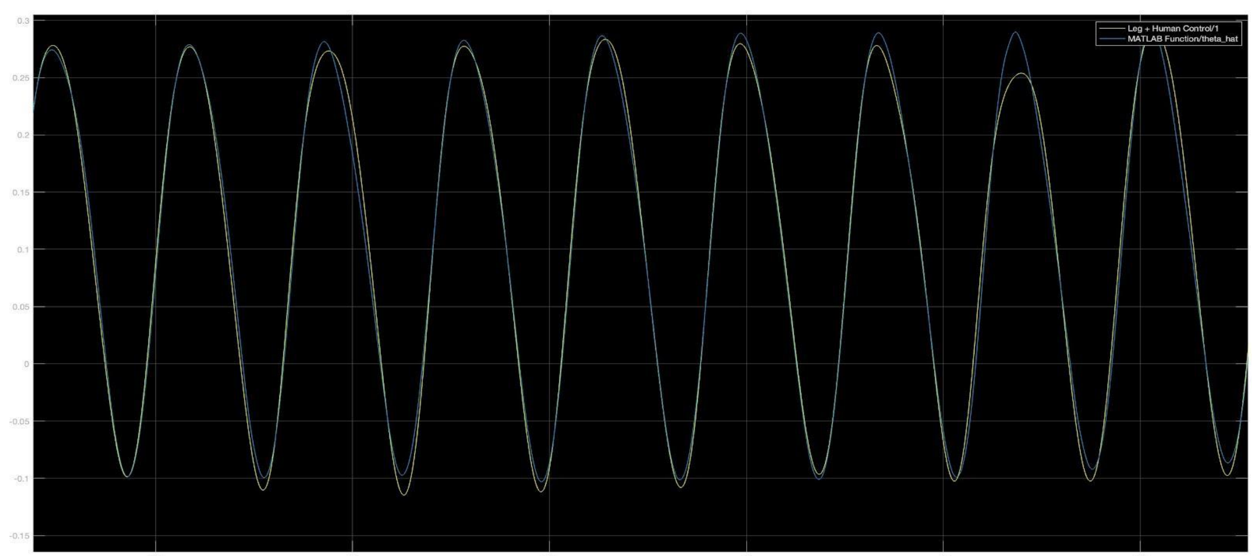


Fig 3.2: left-hip joint trajectory obtained from AFO(Blue) vs. provided left-hip joint trajectory (Yellow) in zoomed version over 60 seconds

4. To develop a model-based assistive controller based on the inverse dynamics obtained in question 2.

Using the outputs from part 3 as inputs to the model based assistive controller to provide the user with an assistive torque τ_R equal to 50% of the torque required.

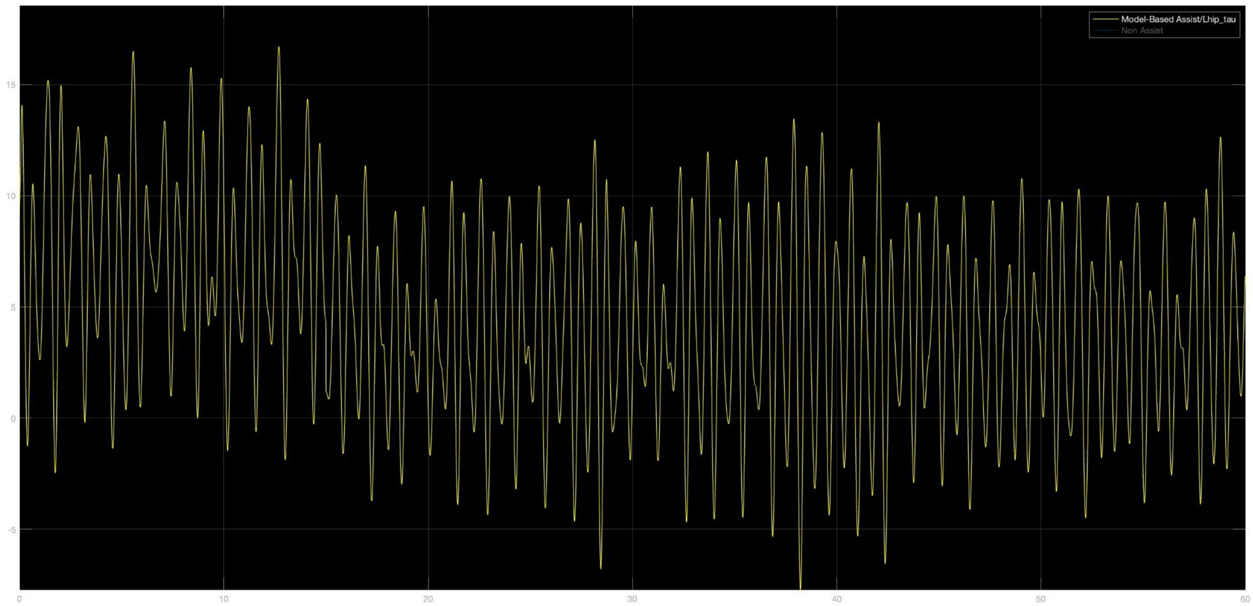


Fig 4.1: torque trajectory of the left-hip control with 50% of the robot assistive torque obtained from the model-based assistive controller over 60 seconds.

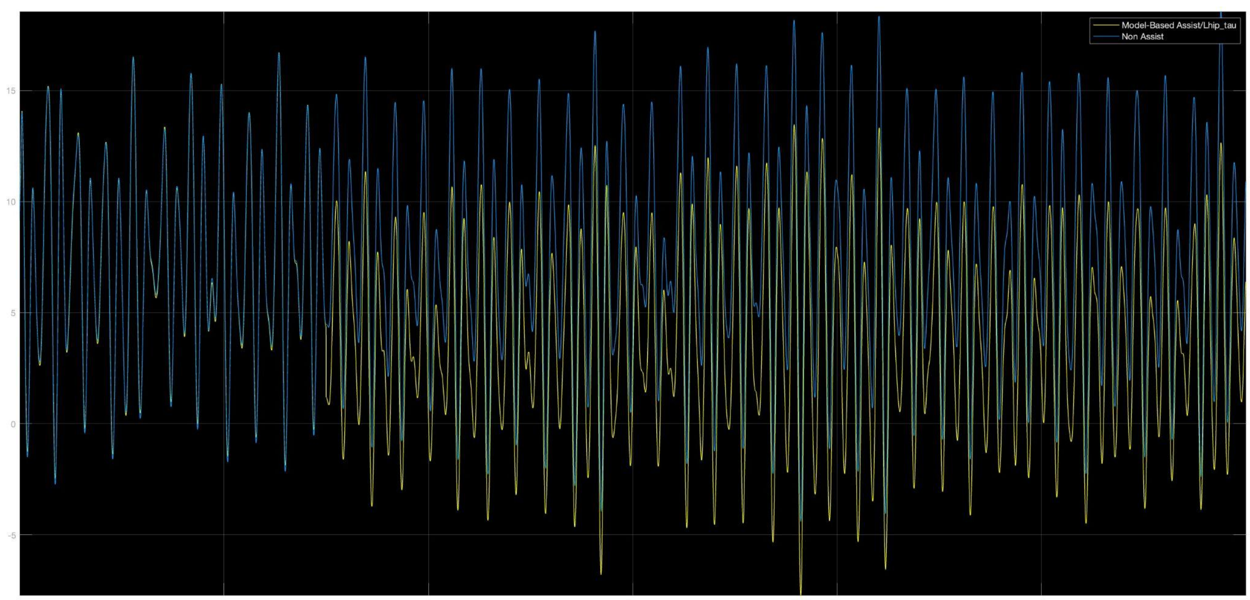


Fig 4.2: torque trajectory of the left-hip control with 50% of the robot assistive torque obtained from the model-based assistive controller (Yellow) vs. torque trajectory of the left-hip control with no assistance (Blue) over 60 seconds

When 50% of the torque is assisted by the assistive robot, the torque required by the leg has decreased due to the assistance of the robot.

5. To implement a *Kernel-based nonlinear filter.*

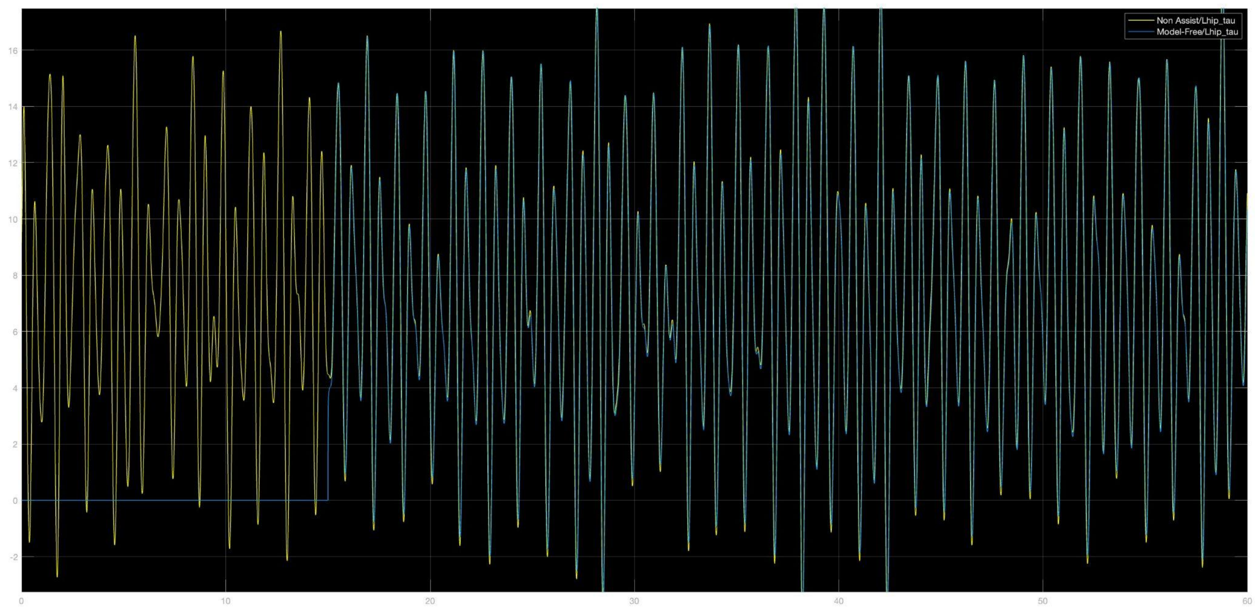
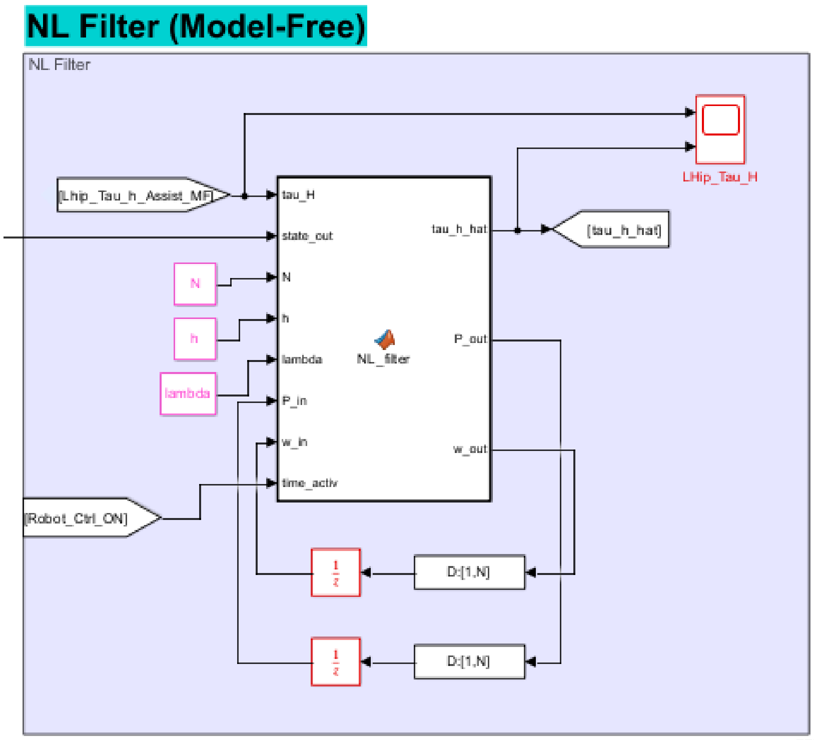


Fig 5.1: torque trajectory of the left-hip control with no assistance obtained from the model-free assistive controller (Blue) vs. torque trajectory of the left-hip control with no assistance (Yellow) over 60 seconds

Fig 5.2: torque trajectory of the left-hip control with no assistance obtained from the model-free assistive controller (Blue) vs. torque trajectory of the left-hip control with no assistance (Yellow) in zoomed version

6. To use the nonlinear filter as a model-free assistive controller.

Simulink Model:



Results:

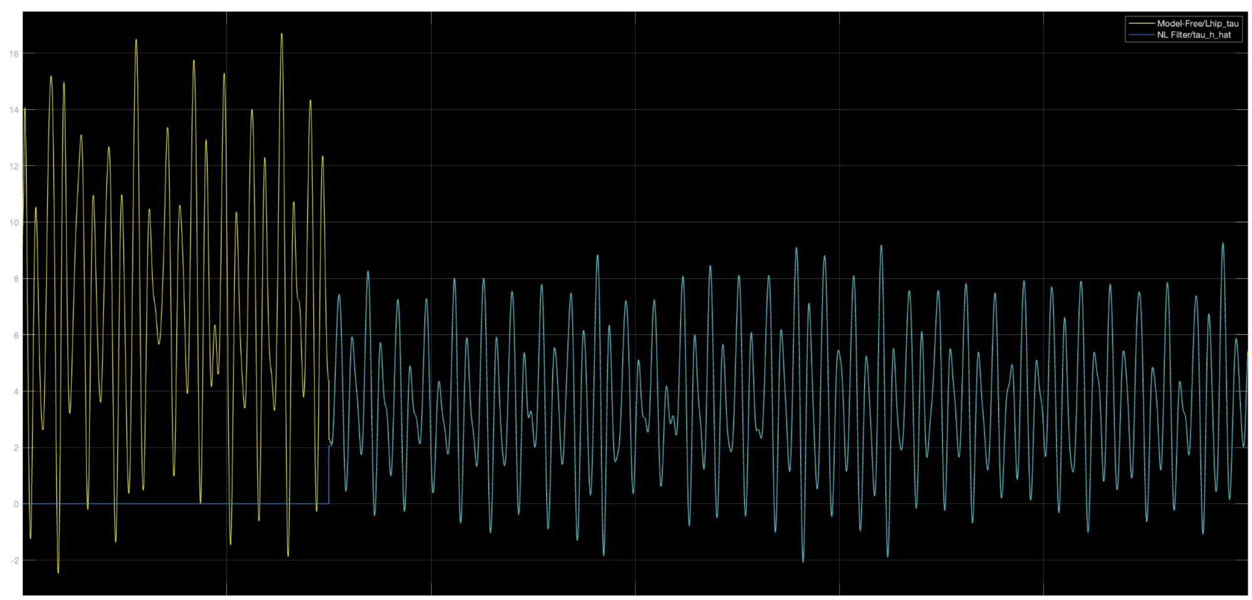


Fig 6.1: estimated torque trajectory of the left-hip control with the robot assistance obtained from the model-free assistive controller (Blue) vs. torque trajectory of the left-hip control with the robot assistance (Yellow) over 60 seconds.

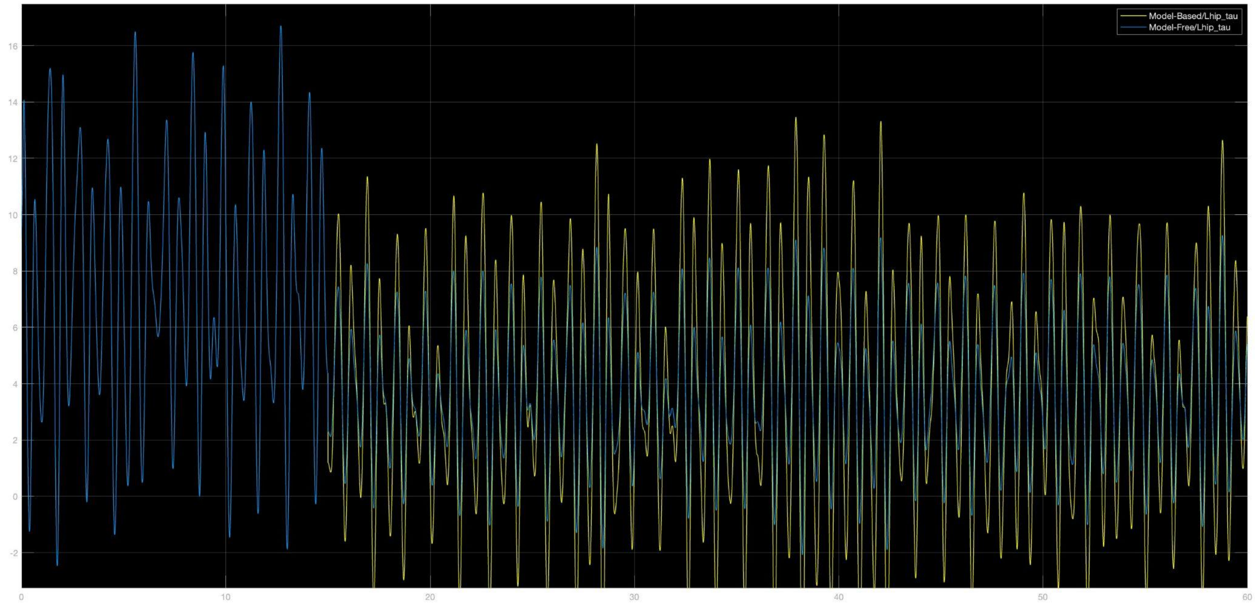


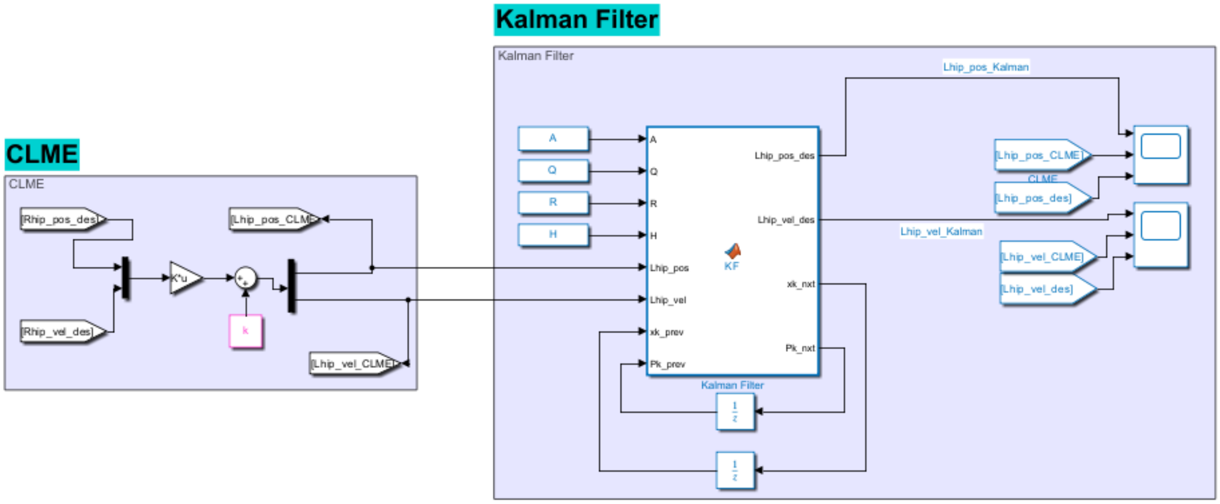
Fig 6.2: estimated torque trajectory of the left-hip control with the robot assistance obtained from the model-free assistive controller (Blue) vs. torque trajectory of the left-hip control with the robot assistance obtained from the model-based assistive controller (Yellow) over 60 seconds

We can see from the above two graphs that model-based help is superior to model-free assistance since the second graph shows that less torque is required on the left hip as compared to the model-free controller. Similarly, when we compare the results of the model-based assistive to the robot-based assistance for 50%, there is a change in torque because robot-based assistance for 50% is not fully assisted and the human must perform more work in moving his leg, so the model-based fully assisted is better than the model-free assistance and 50% assistance. When the KL filter is added the noise is filtered out to give out a good assistance in the gait cycle of the walking phase of the human.

Part B

7. Using the Complementary Limb Motion Estimation (CLME) method and train gain K and offset k and derive the parameters of the Kalman Filter. One of the CLME's uses is gait training for patients with a paretic left leg and a healthy right leg. KF is used to compute coherent estimates from CLME estimates. The constant matrices A, H, Q, and R are determined by the Simulink model. To train the CLME, choose a data set with t = 30, normalize the data, calculate Mhp, Mhh, and C, K,k using only the first 30 seconds of the data set to train Blue (K and k).

Simulink Model:



Code for computing CLME parameters:

```
%calculate Mhp and Mhh
Mhp = nan(size(xh,2),size(xp,2));
Mhh = nan(size(xh,2),size(xh,2));
n = length(xp(:,1));
for i = 1:length(Mhp(:,1))
    for j = 1:length(Mhp(1,:))
        Mhp(i,j) = sum(xh(:,i).*xp(:,j))/(n-1);
    end
end

for i = 1:length(Mhh(:,1))
    for j = 1:length(Mhh(1,:))
        Mhh(i,j) = sum(xh(:,i).*xh(:,j))/(n-1);
    end
end

%calculate C, K, and k
C = (Mhh \ Mhp)';
K = Sp*C*inv(Sh);
k = -K*RhipAvg + LhipAvg;
%given K and k, estimate from BLUE
Lhip_pred = zeros(length(train_indices),2);
Rhip_train = [Rhip_pos_train, Rhip_vel_train];
for i = 1:length(Lhip_pred(:,1))
   Lhip_pred(i,:) = K*Rhip_train(i,:)'+k;
end
```

Code for computing Kalman Filter parameters:

```
%% Kalman Filter

T = t(2)-t(1); %sample period
A = [1 T;0 1]; G = [T^2/2; T];  %process parameters
Q = G*G'*var(Lhip_acc(train_indices(:))); %process noise covariances
```

```
H = eye(2); %measure state directly
R = [var(Lhip_pred(:,1)-Lhip_pos_train) 0;0 var(Lhip_pred(:,2)-Lhip_vel_train)]; %measurement noise covariance
```

Results:

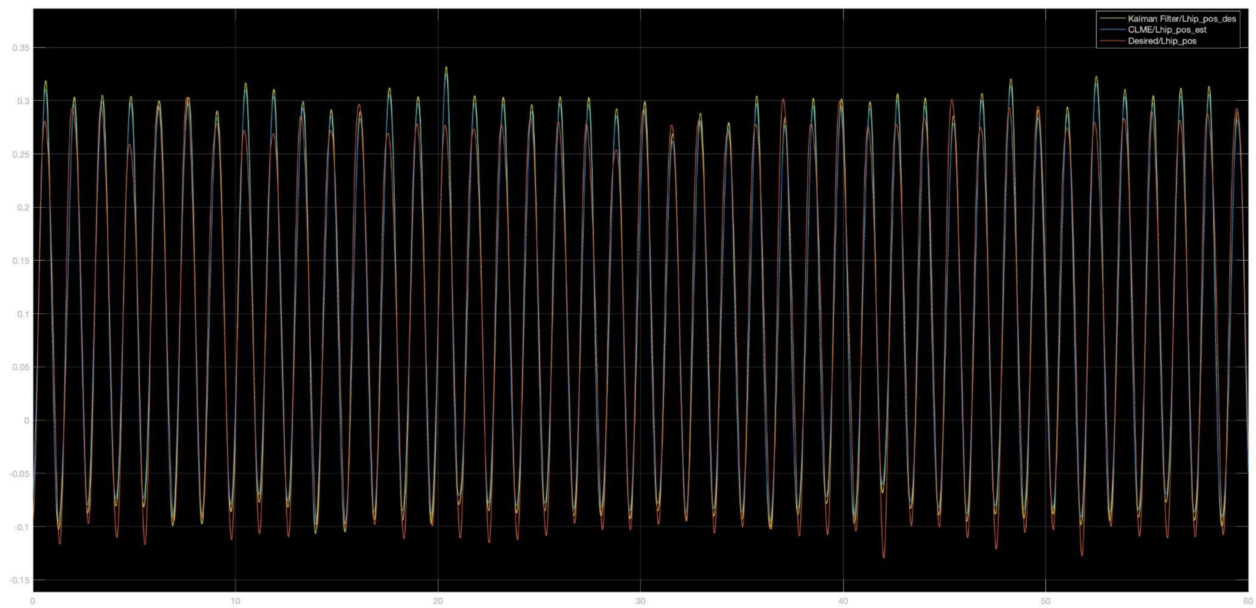


Figure 7.1: Left Hip position, desirable position, and position estimation vs time (yellow represents Kalman filter, blue represents CLME, red represents Desired position)

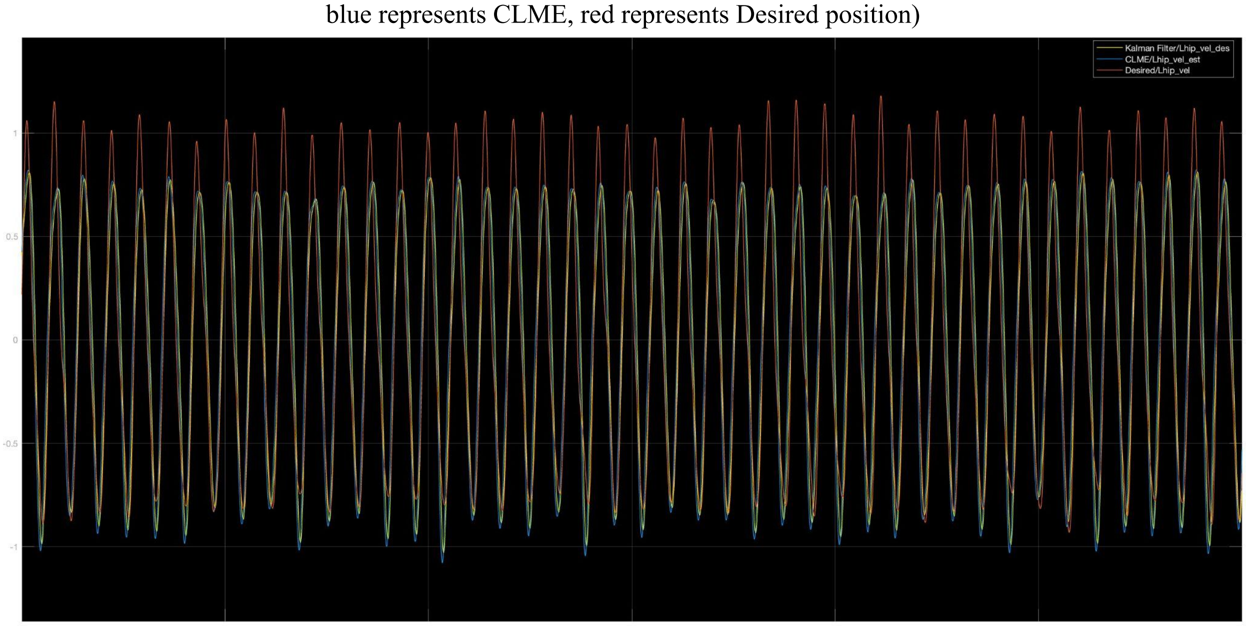


Figure 7.2: Left hip velocity, desired velocity and velocity estimation vs time (yellow represents the Kalman Filter, blue represents the CLME, red represents the desired left hip velocity vs time).

Conclusion: By carrying out the above simulations we have studied the difference between assisted and non-assisted phase of the human leg and model-based and model-free assistance with full and 50% assistance offered by the robot and the use of Kalman filter, NL filters.