Warsaw University of Technology

Faculty of Electronics and Information Technology

ENUME 2025 – Assignment A #05

Estimation of lower limb joint angles

Tutor: Dr. Jakub Wagner

Authors: Qinghe Zhao, Yi Yang

Warsaw, 18.03.2025

Table of Contents

List of Mathematical Symbol

$lfeo$ – Three-dimensional coordinates (in mm) of the left femur external marker.

$ltio$ – Three-dimensional coordinates (in mm) of the left tibia internal marker.

$lkne$ – Three-dimensional coordinates (in mm) of the left knee marker.

$rfep$ – Three-dimensional coordinates (in mm) of the right femur anterior marker.

$rfeo$ – Three-dimensional coordinates (in mm) of the right femur external marker.

$rkne$ – Three-dimensional coordinates (in mm) of the right knee marker.

z_{LT} – Unit vector in the direction of the left thigh, defined as:

$$z_{LT} = \frac{lfeo - ltio}{\left\|lfeo - ltio\right\|_2}$$

y_{LF} – Unit vector in the left lower limb's local coordinate system, defined as:

$$y_{LF} = \frac{lkne - lfeo}{\left\|lkne - lfeo\right\|_2}$$

z_{RF} – Unit vector in the direction of the right femur, defined as:

$$z_{RF} = \frac{rfep - rfeo}{\left\|rfep - rfeo\right\|_2}$$

y_{RF} – Unit vector in the right lower limb's local coordinate system, defined as:

$$y_{RF} = \frac{rfeo - rkne}{\left\|rfeo - rkne\right\|_2}$$

x_{LF} – Lateral unit vector in the left lower limb's local coordinate system, computed as:

$$x_{LF} = \frac{y_{RF} \times z_{RF}}{\left\|y_{RF} \times z_{RF}\right\|_2}$$

β_{LK} – Left knee adduction angle (in radians), calculated by:

$$\beta_{LK} = \arcsin(z_{LT} \cdot y_{LF})$$

α_{LK} – Left knee flexion angle (in radians), computed by:

$$\alpha_{LK} = \arcsin\left(\frac{z_{LT} \cdot x_{LF}}{\cos(\beta_{LK})}\right)$$

fc – Cutoff frequency of the digital low-pass filter (Hz)

MAE – Mean Absolute Error (in radians), defined as the average absolute difference between the noise-free reference value of α_{LK} and the filtered α_{LK}.

$||\cdot||$ – Euclidean (2-) norm of a vector.

$\arcsin(\ )$ – Inverse sine function.

$\cos(\ )$ – Cosine function.

$\times$ – Cross product operator.

$time$ – Time vector (s).

$noise_{std}$ – Standard deviation of the added noise, set to 40 mm.

Introduction

In modern motion science and biomechanics, motion capture systems like Vicon are widely used to obtain the three-dimensional coordinates of markers on the human body. These data allow the estimation of joint angles, which is essential for gait analysis, performance evaluation, and rehabilitation. However, during actual measurements, instrument precision limitations and environmental interference, introduce noise, which can affect the accuracy of the computed joint angles.

The experiment aims to investigate the effect of low-pass filtering on reducing noise and to determine the optimal cutoff frequency (within the range of 0.1 Hz to 20 Hz) that minimizes the mean absolute error (MAE) in the estimation of the left knee flexion angle. The process involves:

1. Compute the left knee flexion angle using noise-free typical gait data.
2. Add normally distributed noise (40 mm standard deviation) to the original marker coordinates to simulate a lower-accuracy motion capture system.
3. Apply a low-pass filter with various cutoff frequencies to smooth the noisy data.
4. Recalculate the left knee flexion angle based on the filtered data.
5. Compute the mean absolute error by comparing the filtered angles to the original value from step 1.
6. Analyze the mean absolute error cross different frequencies to determine the optimal value.

This report presents the methodology, experimental procedures, results, and the analysis, with all calculations performed in MATLAB, leading to a discussion of the optimal cutoff frequency that minimizes the estimation error.

Methodology and Results of Experiments

1. Computation of Left Knee Flexion Angle from Noise-Free Data

Before introducing noise, the left knee flexion angle α_{LK} is computed using the original, noise-free marker coordinates from *typical_gait.mat.*

The file *typical_gait.mat* also contains a time vector, which provides the acquisition time points corresponding to each row of marker coordinates. The values from this time vector are used in all time-based processing and plotting in this experiment.

The procedure follows these steps:

- **Vectors Computation**

First, we compute all unit vectors needed for calculation of the left knee flexion angle.

Although the formulas are defined for a single time point using 3-element column vectors, the implementation uses MATLAB's vectorized operations `vecnorm`, which apply the computations efficiently across all time points (rows) simultaneously, avoiding the need for manual row-by-row processing.

For the full implementation details, see the section *List of the Developed Program.*

The left thigh unit vector is computed by:

$$z_{LT} = \frac{lfeo - ltio}{||lfeo - ltio||_2}$$

The right femur unit vector is computed by:

$$z_{RF} = \frac{rfep - rfeo}{||rfep - rfeo||_2}$$

The right lower unit vector is computed by:

$$y_{RF} = \frac{rfeo - rkne}{||rfeo - rkne||_2}$$

The left lower limb unit vector is computed by:

$$y_{LF} = \frac{lkne - lfeo}{||lkne - l\,feo||_2}$$

The lateral unit vector for the left limb is computed by:

$$x_{LF_{filt}} = \frac{y_{RF} \times Z_{\ RF}}{||y_{RF} \times Z_{\ RF}||_2}$$

 Angle Computation:

Next, we compute the left knee adduction angle by:

$$\beta_{Lk} = \ \text{arcsin}\ (\ z_{LT} \cdot y_{LF})$$

Finally, we use the previously computed vector z_{LT}, x_{LF} and the angle β_{Lk} to compute the left knee flexion angle:

$$\alpha_{LK} = \ \text{arcsin}(\ \frac{z_{LT} \cdot x_{LF}}{\cos(\beta_{LK})})$$

The variation of α_{LK} over time, based on the noise-free data from *typical_gait.mat* is presented in Fig. 1.

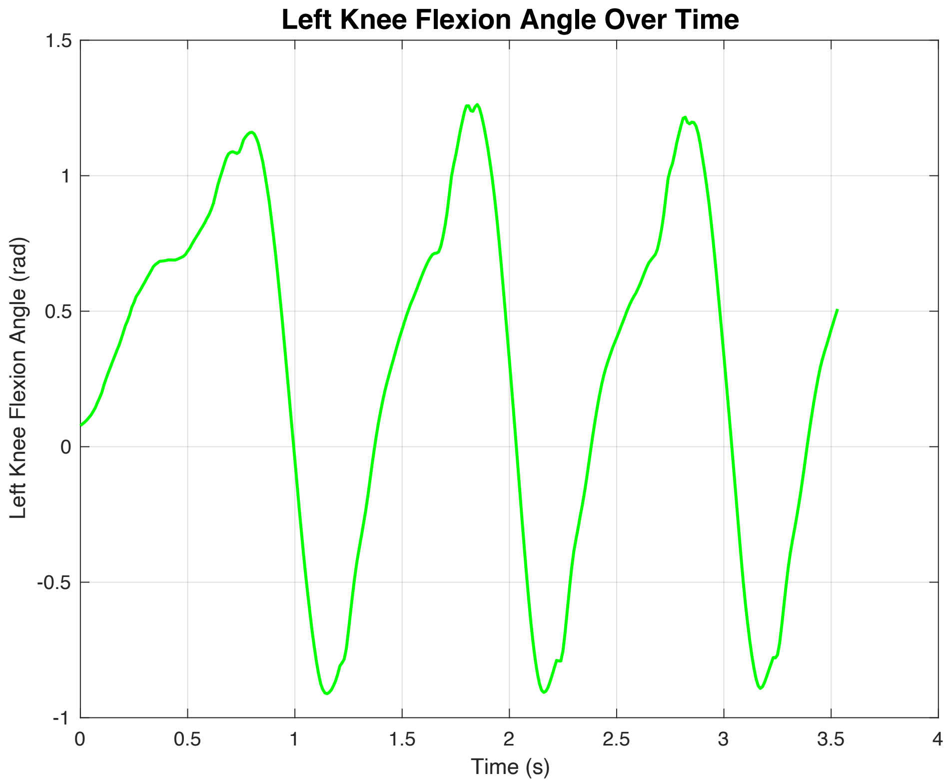

Fig. 1. The variation of α_{LK} over time, based on the noise-free data from *typical_gait.mat*

2. Simulation of Measurement Noise

To simulate the inaccuracies of a lower-precision motion capture system, normally distributed noise is added to the original (noise-free) marker coordinates. In this experiment, noise is generated using MATLAB's `randn` function with a standard deviation of 40 mm. The noise is added element-wise to each marker coordinate.

For example, the noisy left femur external marker data is computed as:

$$lfeo_{noisy} = lfeo + noise_{std} \cdot randn\left(size(lfeo)\right)$$

Similarly, this procedure is applied to all other markers (*ltio, rfep, rfeo, rkne,* and *lkne*), to generate a complete noisy dataset for further analysis.

3. Low-Pass Filtering of Noisy Data

To eliminate the simulated noise, a digital low-pass filter is applied to the noisy data using the MATLAB's `lpfilt` function. The filter is applied individually to each marker's noisy data.

A range of cutoff frequencies is tested to determine the optimal value:

fc_{values} = [0.1, 0.5, 1, 2, 2.5, 3, 3.5, 4, 5, 10, 15, 20] Hz

- 0.1 – 1 Hz: Strong smoothing.
- 2 – 5 Hz: Moderate filtering. (In this experiment majority of values are from this range.)
- 10 – 20 Hz: Minimal filtering.

For each value of fc, the program applies the low-pass filter as follows:

$$lfeo_{filt} = lpfilt\ (\ lfeo_{noisy},\ fc\ )$$

This filtering process is applied similarly to all other markers (*ltio, rfep, rfeo, rkne,* and *lkne*). The filtered coordinates are then used to recalculate the necessary vectors and subsequently the filtered left knee flexion angle.

4. Recalculation of the Left Knee Flexion Angle

After applying the low-pass filter to the noisy marker coordinates, the left knee flexion angle is recalculated. The computational process remains identical to that in Step 1, except that the filtered marker coordinates are used instead of the original noise-free data.

The recalculated vectors and angles are computed as follows:

- The filtered unit vectors $z_{LT_{filt}}$, $y_{LF_{filt}}$, $x_{LF_{filt}}$ are obtained from the filtered marker coordinates ($lfeo_{filt}$, $ltio_{filt}$, etc.).

- The left knee adduction angle is recomputed using:

$$\beta_{LK_{filt}} = \arcsin\left(z_{LT_{filt}} \cdot y_{LF_{filt}} \right)$$

- The left knee flextion angle is then recalculated with:

$$\alpha_{LK_{filt}} = \arcsin\left(\frac{z_{LT_{filt}} \cdot x_{LF_{filt}}}{\cos\left(\beta_{LK_{filt}}\right)} \right)$$

To prevent numerical errors, values are constrained within [-1,1] before applying the $\arcsin(\ \ )$ function.

Fig. 2 – Fig. 13 show the variation of α_{LK} over time, computed from the noisy data after applying low-pass filtering at different cutoff frequencies from the set fc_{values}.

Fig. 2. the variation of α_{LK} over time at $fc = 0.1$

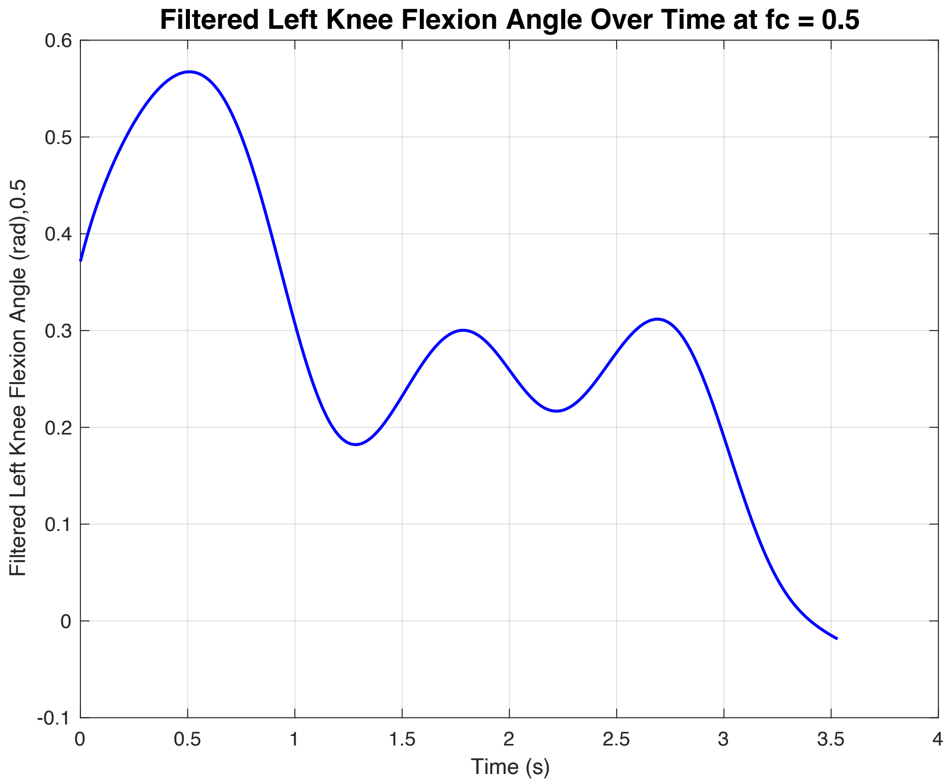

Fig. 3. the variation of α_{LK} over time at $fc = 0.5$

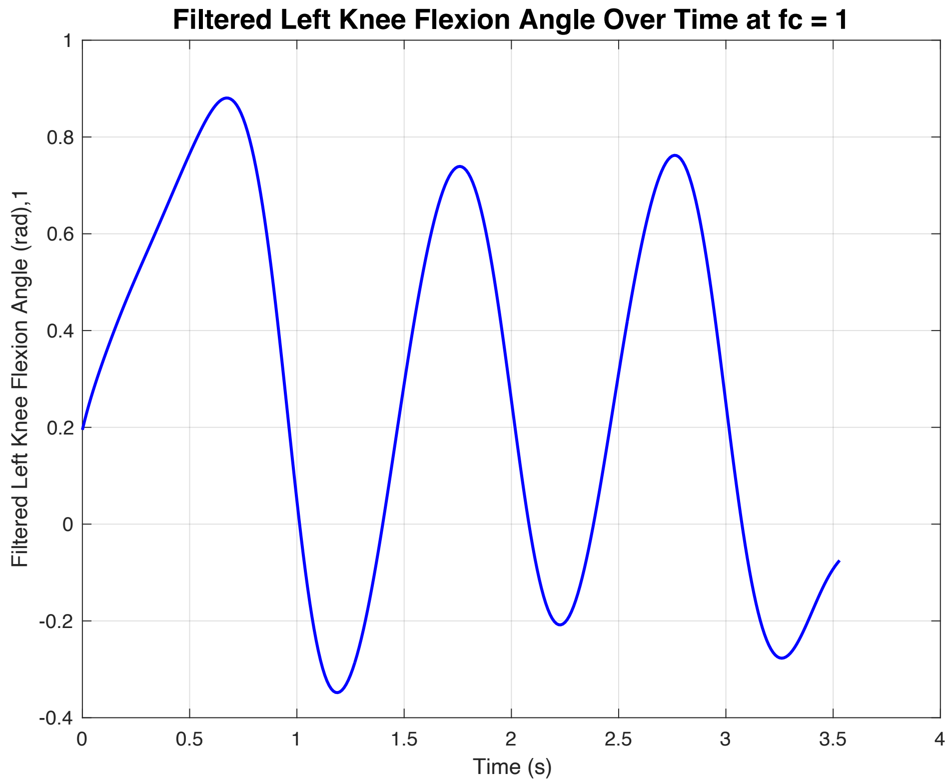

Fig. 4. the variation of α_{LK} over time at $fc = 1.0$

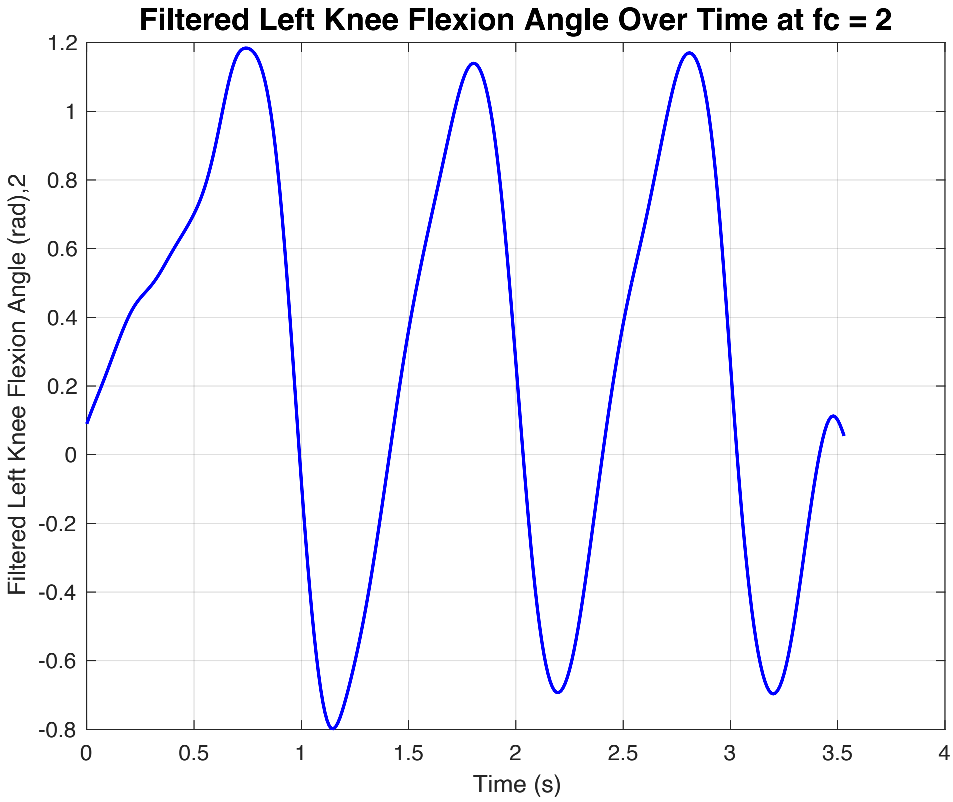

Fig. 5. the variation of α_{LK} over time at $fc = 2.0$

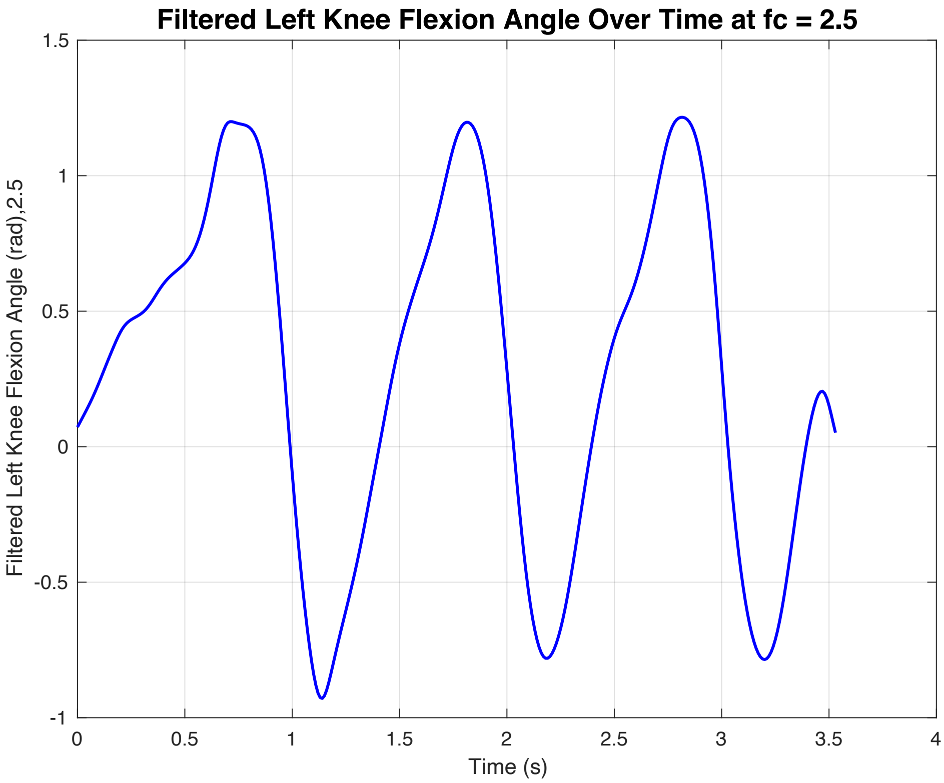

Fig. 6. the variation of α_{LK} over time at $fc = 2.5$

Fig. 7. the variation of α_{LK} over time at $fc = 3.0$

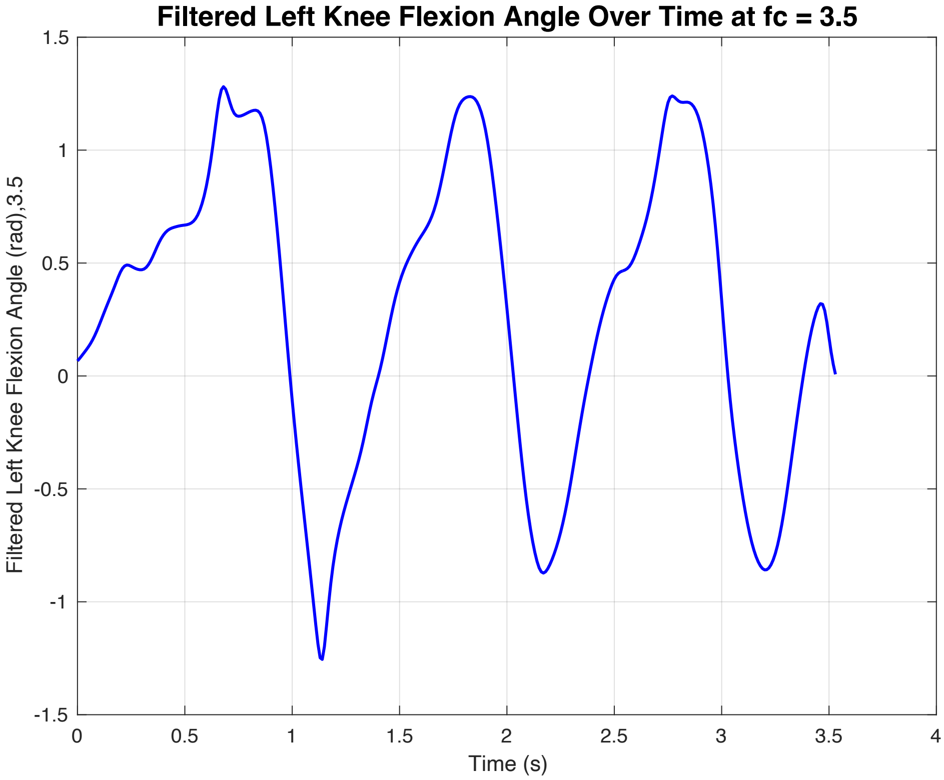

Fig. 8. the variation of α_{LK} over time at $fc = 3.5$

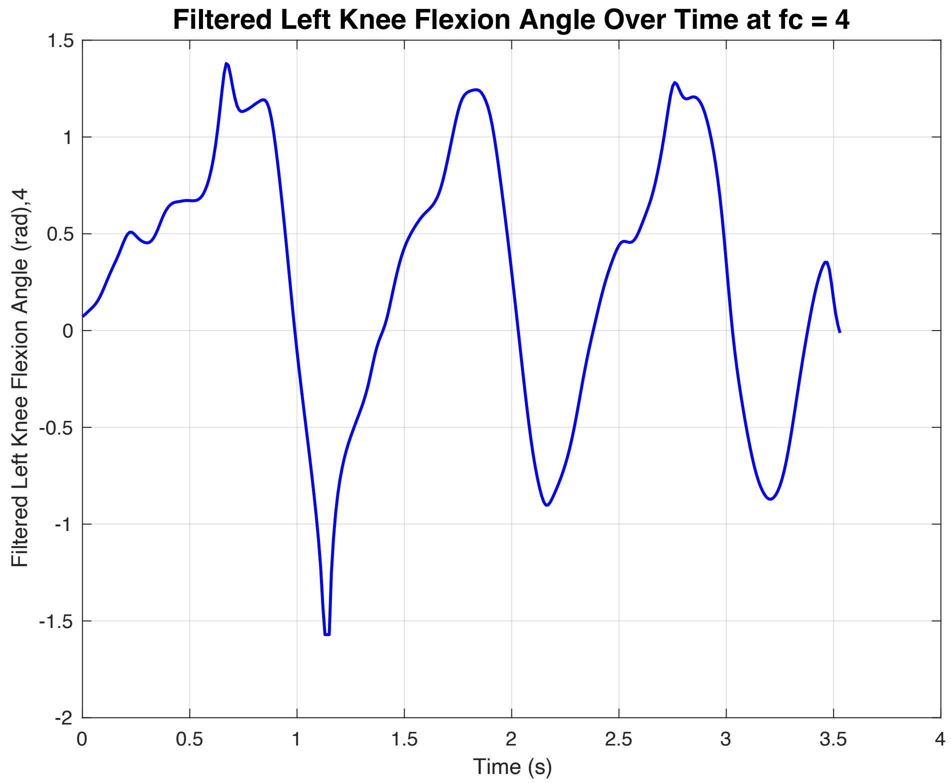

Fig. 9. the variation of α_{LK} over time at $fc = 4.0$

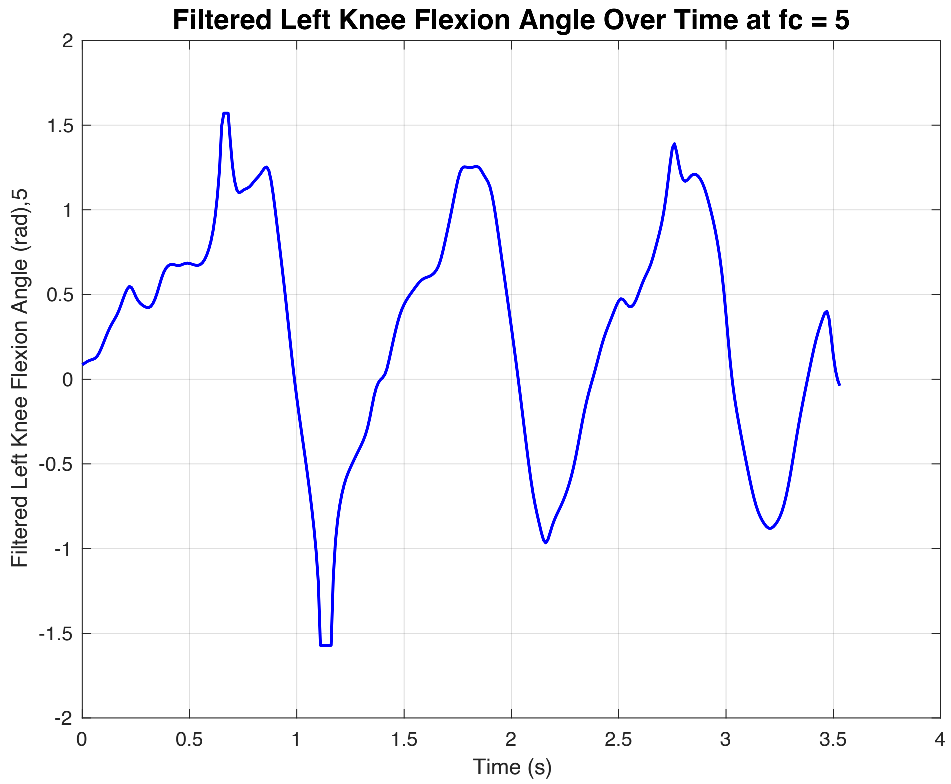

Fig. 10. the variation of α_{LK} over time at $fc = 5.0$

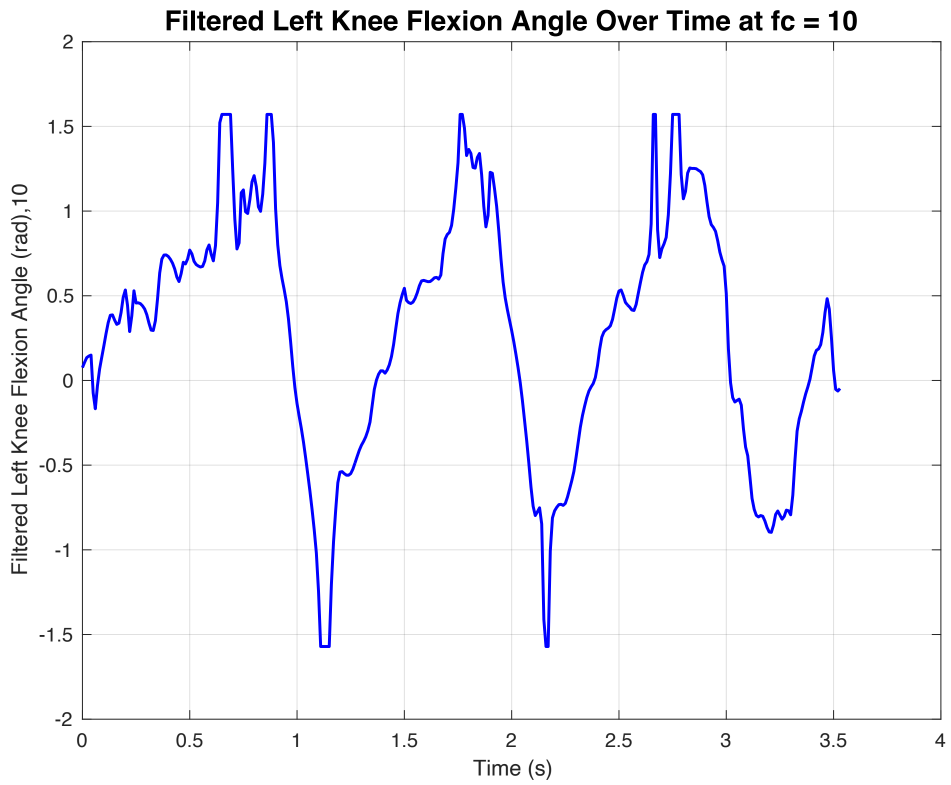

Fig. 11. the variation of α_{LK} over time at $fc = 10.0$

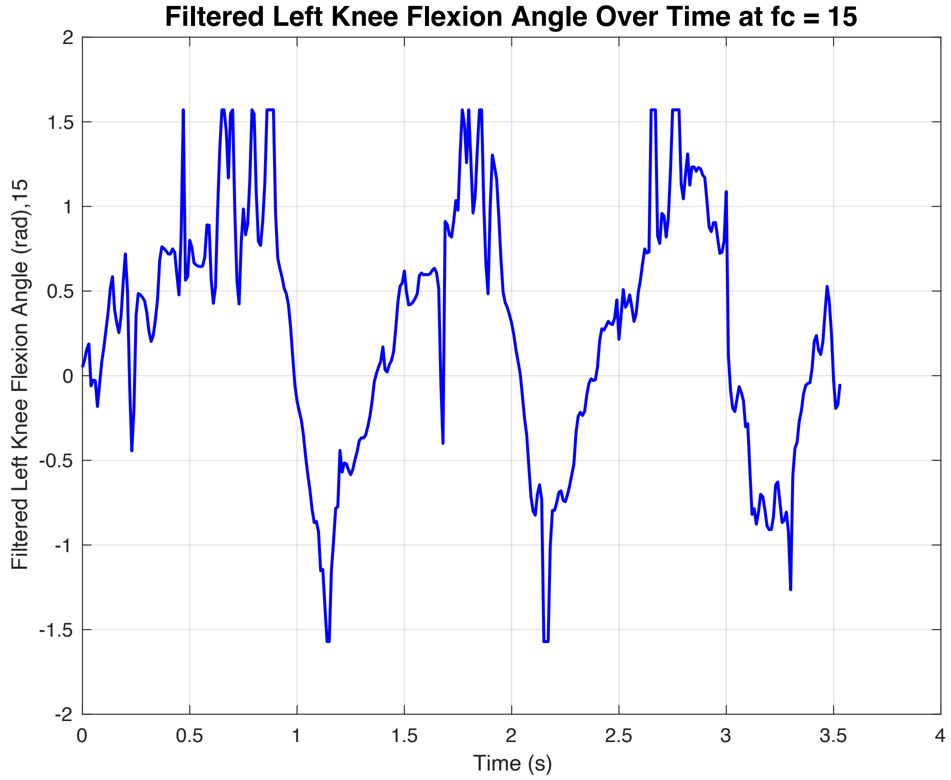

Fig. 12. the variation of α_{LK} over time at $fc = 15.0$

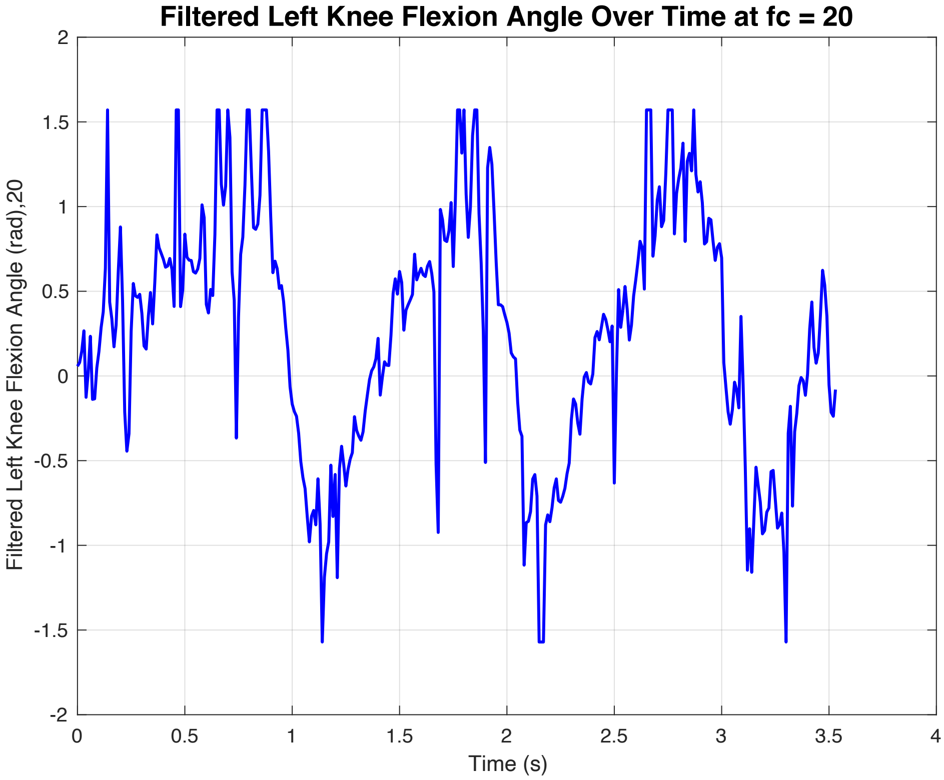

Fig. 13. the variation of α_{LK} over time at $fc = 20.0$

5. Error Evaluation

The performance of the filtering process is quantified by calculating the Mean Absolute Error (MAE) between the original (noise-free) left knee flexion angle α_{LK} and the filtered angle $\alpha_{LK_{filt}}$:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\alpha_{LK}(i) - \alpha_{LK_{filt}}(i)\right|$$

Here, N is the total number of time samples. The MAE is computed for each cutoff frequency. The results are presented in table 1. The visualized relationship between the cutoff frequency and the error is presented in fig.14.

table. 1. The result of computing MAE at different cutoff frequencies

fc (Hz)	MAE (rad)
0.1	0.5667
0.5	0.4778
1.0	0.2543
2.0	0.0894
2.5	0.0635
3.0	0.0591
3.5	0.0631
4.0	0.0716
5.0	0.0852
10.0	0.1276
15.0	0.1739
20.0	0.2204

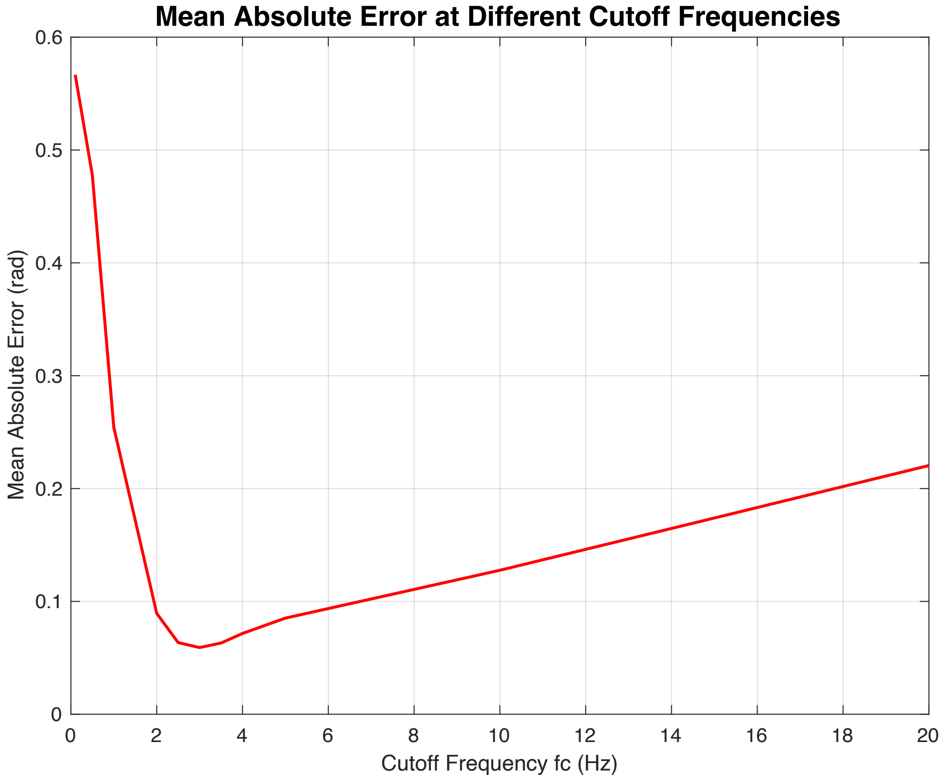

Fig. 14. the visualized relationship between the cutoff frequency and the error

Discussion

The experimental results clearly demonstrate the impact of low-pass filter cutoff frequency on the accuracy of the estimated left knee flexion angle. By comparing the mean absolute error (MAE) values obtained at different cutoff frequencies, several key observations can be made:

1. **Trend of MAE Reduction and Increase:**

   As the cutoff frequency increases from 0.1 Hz to 3.0 Hz, the MAE progressively decreases. After reaching a minimum at 3.0 Hz, the MAE begins to increase gradually as the fc continues to rise toward 20 Hz. This increase is relatively smooth and steady. This U-shaped trend confirms that both under-filtering (high fc) and over-filtering (low fc) negatively affect the estimation, which will be further discussed below.

2. **Excessive Smoothing at Low Cutoff Frequencies:**

   At very low cutoff frequencies (e.g., 0.1 Hz and 0.5 Hz), the filter removes not only the noise but also important signal components. This excessive smoothing distorts the signal, leading to significantly higher errors (e.g., 0.5667 rad and 0.4778 rad), showing that such low frequencies are not suitable for accurate angle estimation.

3. **Insufficient Noise Suppression at High Cutoff Frequencies:**

   Conversely, when the cutoff frequency is set too high (above 3.0 Hz), noise begins to reappear in the filtered signal. Although signal details are better preserved, the residual noise increases the error again (e.g., 0.0716 rad at 4 Hz, with further increases at 5 Hz and above). This indicates that high fc values fail to sufficiently attenuate the noise, resulting in a degradation of the estimation accuracy.

4. **Optimal Cutoff Frequency:**

   The lowest MAE of approximately 0.0591 rad is achieved at 3.0 Hz, suggesting that this frequency provides an optimal balance between noise reduction and signal preservation in this experimental setup. At 3.0 Hz, the filter effectively reduces the noise without compromising the integrity of the signal. This optimal setting is crucial for ensuring accurate estimation of the left knee flexion angle, particularly in applications where measurement precision is critical.

5. **Implications for Practical Applications:**

   These findings highlight the importance of selecting an appropriate cutoff frequency when processing motion capture data. In practical scenarios, where measurement errors are inevitable, the proper tuning of the low-pass filter is essential for minimizing error and ensuring the reliability of joint angle estimations. The experimental results suggest that, for systems similar to the one simulated here, a cutoff frequency near 3 Hz should be considered as a starting point for further calibration and refinement.

6. **Limitations and Future Work:**

   Although the experiment successfully identifies an optimal cutoff frequency under the simulated conditions - specifically, normally distributed noise with a standard deviation of 40 mm added to marker coordinates. It is important to note that real-world scenarios may introduce additional complexities such as non-stationary noise or variations in motion dynamics. Future studies could explore adaptive filtering techniques or investigate the influence of different noise distributions to further improve the robustness of joint angle estimations.

Conclusion

In summary, this experiment investigated the effect of low-pass filter cutoff frequency on the accuracy of estimating the left knee flexion angle using typical gait data. By simulating measurement noise and applying a range of cutoff frequencies, the study identified that a cutoff frequency of approximately 3 Hz yields the minimum mean absolute error (MAE) in the estimation process.

The experimental results confirmed that excessively low cutoff frequencies lead to over-smoothing, which distorts the signal and results in high error values, while excessively high cutoff frequencies fail to effectively suppress the noise. The optimal balance was achieved at 3 Hz, where the filtered signal most closely approximated the noise-free reference, thereby providing an accurate estimation of the left knee flexion angle.

This study highlights the critical importance of appropriately tuning filter parameters when processing motion capture data, and the findings serve as a useful guideline for practical applications where measurement noise is inevitable. Although the current simulation successfully determined an optimal cutoff frequency under controlled conditions, further research is warranted to address additional real-world complexities, such as non-stationary noise and dynamic variations in human motion.

References

[1] Dr. Jakub Wagner, *“Assignment A: Estimation of lower limb joint angles”,* Institute of Radioelectronics and Multimedia Technology, spring semester 2025

[2] Dr. Jakub Wagner, *“Assignment A - #05”* Warsaw University of Technology, Faculty of Electronics and Information Technology Numerical Methods (ENUME), Spring Semester 2025.

[3] R. Z. Morawski, Lecture Notes for the Course *Numerical Methods*, Warsaw University of Technology, Faculty of Electronics and Information Technology, spring semester 2024/25.

[4] *typical_gait.mat*, Warsaw University of Technology, 2025. (Internal dataset)

[5] *lpfilt.m*, Warsaw University of Technology, 2025. (Internal MATLAB function)

Listing of the developed program

```
%  TASK A loads typical_gait.mat, which contains time samples and
%  3D coordinates (in mm) of various markers (e.g., lfeo, ltio,
%  rfep, rfeo, rkne, lkne). The script computes the left knee
%  flexion angle from the noise-free data, simulates a lower-
%  accuracy motion capture  system by adding normally distributed
%  noise to each marker coordinate, and then applies a low-pass
%  digital filter (lpfilt.m) at multiple cutoff frequencies.
%  Finally, it recalculates the flexion angle from the filtered data
%  and compares it to the original angle by computing the mean
%  absolute error (MAE).
%
%  The steps are:
%  1) Load noise-free gait data from typical_gait.mat.
%  2) Compute the left knee flexion angle (alpha_LK) using
%     vector operations based on the Plug-in Gait approach.
%  3) Add zero-mean Gaussian noise (std = 40 mm) to each marker
%     to simulate measurement inaccuracies.
%  4) Filter the noisy data with a low-pass filter (lpfilt.m)
%     using a range of cutoff frequencies (0.1 to 20 Hz).
%  5) Recompute alpha_LK from the filtered data.
%  6) Calculate and plot the mean absolute error (MAE) vs.
%     cutoff frequency to identify the optimal filter setting.
```

File: *taskA.m*

```
clearvars; close all;
load('typical_gait.mat');

%calculate based on data from typical_gait.mat (without error)
z_LT = (lfeo - ltio) ./ vecnorm((lfeo - ltio), 2, 2);
z_RF = (rfep - rfeo) ./ vecnorm((rfep - rfeo), 2, 2);
y_RF = (rfeo - rkne) ./ vecnorm((rfeo - rkne), 2, 2);
y_LF = (lkne - lfeo) ./ vecnorm((lkne - lfeo), 2, 2);
x_LF = cross(y_RF, z_RF) ./ vecnorm(cross(y_RF, z_RF), 2, 2);
beta_LK = asin(dot(z_LT, y_LF, 2));
alpha_LK = asin(dot(z_LT, x_LF, 2) ./ cos(beta_LK));
```

```
%graph (without error)
figure;
plot(time, alpha_LK, 'g', 'LineWidth', 1.5);
grid on;
xlabel('Time (s)');
ylabel('Left Knee Flexion Angle (rad)');
title('Left Knee Flexion Angle Over Time', 'FontSize', 14);

%add noise
%randn - generate random error following normal distribution
%repeat this part?
noise_std = 40;
lfeo_noisy = lfeo + noise_std * randn(size(lfeo));
ltio_noisy = ltio + noise_std * randn(size(ltio));
rfep_noisy = rfep + noise_std * randn(size(rfep));
rfeo_noisy = rfeo + noise_std * randn(size(rfeo));
rkne_noisy = rkne + noise_std * randn(size(rkne));
lkne_noisy = lkne + noise_std * randn(size(lkne));

%smooth datas with filter (using different fc)
fc_values = [0.1, 0.5, 1, 2, 2.5, 3, 3.5, 4, 5, 10, 15, 20];
MAE_values = zeros(size(fc_values));

set(0, 'DefaultFigureWindowStyle', 'normal')

for i = 1 : length(fc_values)
    fc = fc_values(i);

    lfeo_filt = lpfilt(lfeo_noisy, fc);
    ltio_filt = lpfilt(ltio_noisy, fc);
    rfep_filt = lpfilt(rfep_noisy, fc);
    rfeo_filt = lpfilt(rfeo_noisy, fc);
    rkne_filt = lpfilt(rkne_noisy, fc);
    lkne_filt = lpfilt(lkne_noisy, fc);

    z_LT_filt = (lfeo_filt - ltio_filt) ./ vecnorm((lfeo_filt -
   ltio_filt), 2, 2);
    z_RF_filt = (rfep_filt - rfeo_filt) ./ vecnorm((rfep_filt -
   rfeo_filt), 2, 2);
    y_RF_filt = (rfeo_filt - rkne_filt) ./ vecnorm((rfeo_filt -
   rkne_filt), 2, 2);
```

```
    y_LF_filt = (lkne_filt - lfeo_filt) ./ vecnorm((lkne_filt -
    lfeo_filt), 2, 2);
    x_LF_filt = cross(y_RF_filt, z_RF_filt) ./
    vecnorm(cross(y_RF_filt, z_RF_filt), 2, 2);
    beta_LK_filt = asin(dot(z_LT_filt, y_LF_filt, 2));
    val = dot(z_LT_filt, x_LF_filt, 2) ./ cos(beta_LK_filt);
    val = max(min(val, 1), -1);
    alpha_LK_filt = asin(val);

    figure(i+1);
    plot(time, alpha_LK_filt, 'b', 'LineWidth', 1.5);
    grid on;
    xlabel('Time (s)');
    ylabel(['Filtered Left Knee Flexion Angle(rad),' num2str(fc)]);
    title(['Filtered Left Knee Flexion Angle Over Time at fc = ',
    num2str(fc)], 'FontSize', 14);

    %calculate mean absolute error
    MAE_values(i) = mean(abs(alpha_LK - alpha_LK_filt));
    fprintf('fc = %.1f Hz, MAE = %.4f rad\n', fc, MAE_values(i));

end

%plot MAE at different frequencies on graph
figure;
plot(fc_values, MAE_values, 'r', 'LineWidth', 1.5);
grid on;
xlabel('Cutoff Frequency fc (Hz)');
ylabel('Mean Absolute Error (rad)')
title('Mean Absolute Error at Different Cutoff Frequencies',
'FontSize', 14);
```