1. Introduction

In engineering, it is often difficult to control systems that have unknown or changing parameters. In such cases, adaptive control methods are used to help the system stay stable and follow the desired behavior. In this study, a linear mass-spring-damper (MSD) mechanical system is used, and different control strategies are designed to control it even when some system parameters are not known.
This project mainly uses the Backstepping method. This method helps us design the control step by step in a systematic way. One big advantage of Backstepping is that it allows us to prove the stability of the system using a Lyapunov function. It is also flexible, which means we can easily add other methods like adaptive control and artificial neural networks (ANNs) to improve performance. With these additions, the controller can work well even if the system changes or has disturbances.
In the designed controller, a linearly parameterized adaptive rule is used, and an ANN model is added to estimate the unknown parts of the system. The ANN can learn the system's behavior in real time and help the controller react to changes better.
In this study, three different controllers are developed:An adaptive controller,
A linearly parameterized controller, and an ANN-based controller via backstepping approach.

2. Theoretical Background

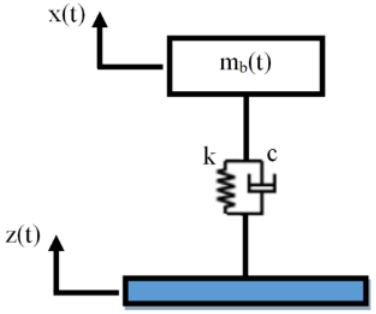

Figure 2.1. 1 dof mechanical system

An active vibration control is applied to a single-degree-of-freedom (1-DoF) mechanical system, consisting of a mass-spring-damper model. The main objective is to suppress the vibrations of the mass caused by external
disturbances z(t).
The dynamic equation of the system is derived using Euler-Lagrange formulation
as:

$$m\frac{d^2x_d(t)}{dt^2}+c\frac{dx_d(t)}{dt}+kx_d(t)=c\frac{dx_z(t)}{dt}+kx_z(t)$$

where $x_d(t)$ is the displacement of the mass, $x_z(t)$ is the output, and c and k represent damping and stiffness coefficients.

3. System Model and Control Strategy

Dynamic Model

m$\ddot{x}_d$(t)+c$\dot{x}_d$(t)+kx $_d$(t)= c$\dot{x}_z$(t)+ kx$_z$(t) (1)
m$\ddot{x}_d$(t)+c($\dot{x}_d$(t)- $\dot{x}_z$(t))+k(x $_d$(t)- x$_z$(t))= τ (2)

Backstepping Approach Parameters

$x_1=\dot{x}_d$ $x_{1d}=\dot{x}_z$ (3)
$x_2= x_d$ $x_{2d}= x_z$ (4)
$\dot{x}_2=x_1$ $\dot{x}_{2d}= X_{1d}$ (5)

So dynamic model is :

m$\dot{x}_1$+c(x $_1$- x$_{1d}$)+k(x $_2$- x$_{2d}$)= τ (6)

Control purpose: $x_2 \rightarrow x_{2d}$ as $t \rightarrow \infty$

Error Definitons

Tracking error: e= X_{2d}- x_2 (7)

Instead of x_1 a virtual control input uv is designed :
$e_v = u_v - x_1$ (9)

Control Analysis and design for u_v :
$u_v = \dot{x}_{2d} + K_v e$ (10)

Closed Loop Error System :
$\dot{e}=-K_v e$ (11)
Where K_v is the constant positive control gain

Lyapunov Like Stability Analysis
$V_1=\frac{1}{2}e^2$ (12)
$\dot{v}_1=-k_v e^2$ (13)

For adaptive , lineeratizaiton and ANN based controller via backstepping given formulas are used.

Adaptive Controller via Backstepping Approach
$\dot{\hat{m}}= e_v(\ddot{x}_{2d}+ k_v e)$ (14)
$\dot{\hat{c}}= e_v (x_1 - x_{1d})$ (15)
$\dot{\hat{k}}= e_v(x_2 - x_{2d})$ (16)

Control input is:
$\tau= \hat{m}\ddot{x}_{2d}+\hat{m} K_v\dot{e}+\hat{c} (x_1- x_{1d})+ \hat{k} (x_2- x_{2d})+K_g e_v$ (17)
Where K_g is the constant positive control gain

ANN Based Controller via Backstepping Approach

$\dot{\hat{w}}= e_v [h(x_v)]^T$ (18)

$x_v= [\ \ddot{x}_{2d}\ \ \dot{e}\ \ X_1\ \ X_{1d}\ \ x_2\ \ X_{2d}\ ]^T$ (19)

Control input is:
$\tau=\hat{w}h(x_v)- K_g\, e_v+e+\delta\varepsilon\, sgn(e_v)$ (20)
Where K_g is the constant positive control gain, ε is a bounded modeling error and ,
$|\varepsilon| \leq \delta\varepsilon \quad \delta\varepsilon \in IR$ (21)

Linear Parametization via Backstepping Approach

$y=[\ \ddot{x}_{2d}+\dot{e}\ \ x_1\ \ x_{1d}\ \ x_2\ \ x_{2d}\ ]$ (22)

$q=\begin{bmatrix} m \\ c \\ k \end{bmatrix}$ (23)

$\dot{\hat{Q}}= = y^T e_v$ (24)

Control input is:
$\tau=y\hat{Q}- K_g\, e_v-e$ (25)
Where K_g is the constant positive control gain

For using this special equations control input design was successfully implemented.

4. Simulation Setup

```
% Parametreler
    m = 300;   % kg
    c = 1000;   % Ns/m
    k = 20000; % N/m
```

Figure4.1 System Parameters

4.1 Simulation Setup for Adaptive Controller via Backstepping Approach

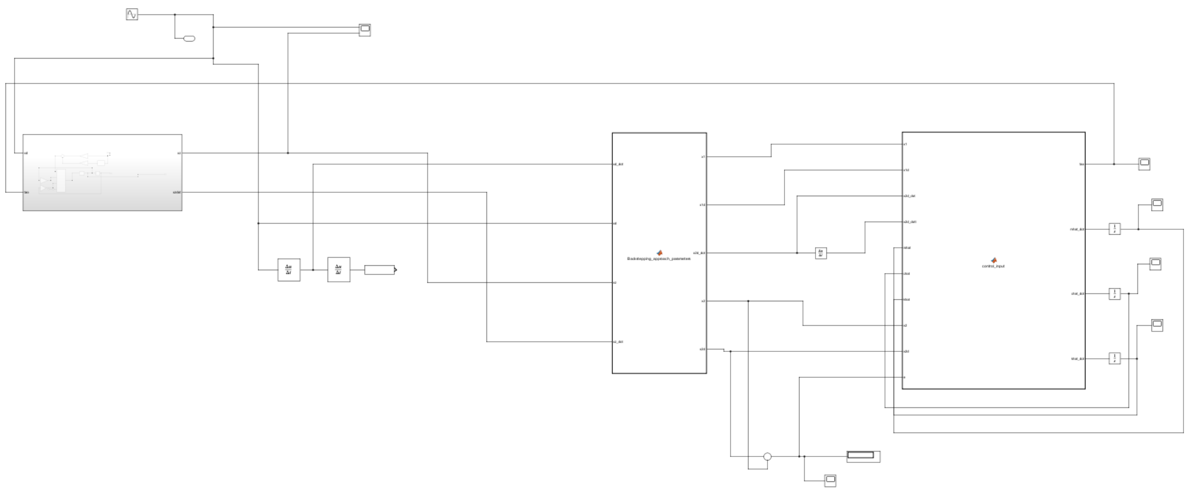

Figure4.1.1 Simulink Model

```
function [x1,x1d,x2d_dot,x2,x2d] = Backstepping_approach_parameters(xd_dot,xd,xz,xz_dot)

x1=xd_dot;
x2=xd;
x2_dot=x1;
x2d=xz;
x1d=xz_dot;
x2d_dot=x1d;
```

```
function [tao,mhat_dot,chat_dot,khat_dot] = control_input(x1,x1d,x2d_dot,x2d_dott,mhat,chat,khat,x2,x2d,e)
    % Parameters
    m = 300;      % kg
    c = 1000;     % Ns/m
    k = 20000;    % N/m
    Kv= 9;
    Kg=10;

    %Error Definitions
    edot=-Kv*e;
    Uv=x2d_dot+Kv*e;
    ev=Uv-x1;
    %Lyapunov like stability analysis

    V1=1/2*e^2;
    V1_dot=-Kv*e^2;

    %Doing Stability Analysis i found mhat_dot,chat_dot and khat_dot

    mhat_dot=ev*(x2d_dott+Kv*edot);
    chat_dot=ev*(x1-x1d);
    khat_dot=ev*(x2-x2d);

    %Control Input tao Design
    tao=mhat*x2d_dott+mhat*Kv*edot+chat*(x1-x1d)+khat*(x2-x2d)+Kg*ev;

end
```

Figure4.1.2 MATLAB function code

4.2 Simulation Setup for Linear Parametization Controller via Backstepping Approach

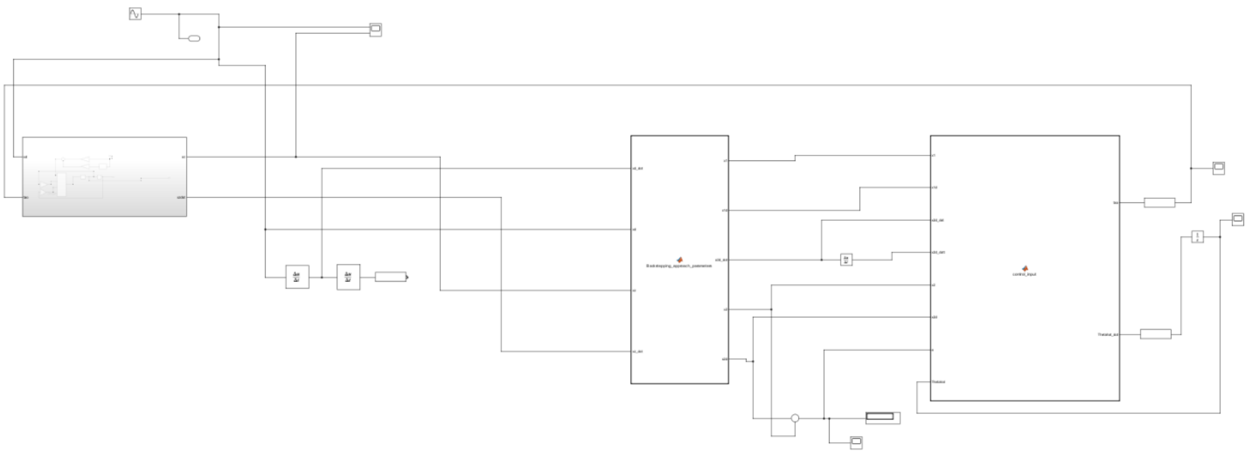

Figure4.2.1 Simulink Model

```
function [tao, Thetahat_dot] = control_input(x1, x1d, x2d_dot, x2d_dott, x2, x2d, e, Thetahat)

    % Gains
    Kv = 3;
    Kg = 10;

    % Error Definitions
    edot = -Kv * e;
    Uv = x2d_dot + Kv * e;
    ev = Uv - x1;

    %Force the Thetahat vector to be a column vector
    Thetahat = Thetahat(:);

    % y contains measurable and known terms and their functions
    y = [x2d_dott + Kv * edot, x1 - x1d, x2 - x2d];
    Thetahat_dot = y' * ev;

    % Control Input tao
    tao = y * Thetahat + Kg * ev + e;

end
```

Figure4.2.2 MATLAB function code

4.2 Simulation Setup for Linear ANN Based Controller via Backstepping Approach

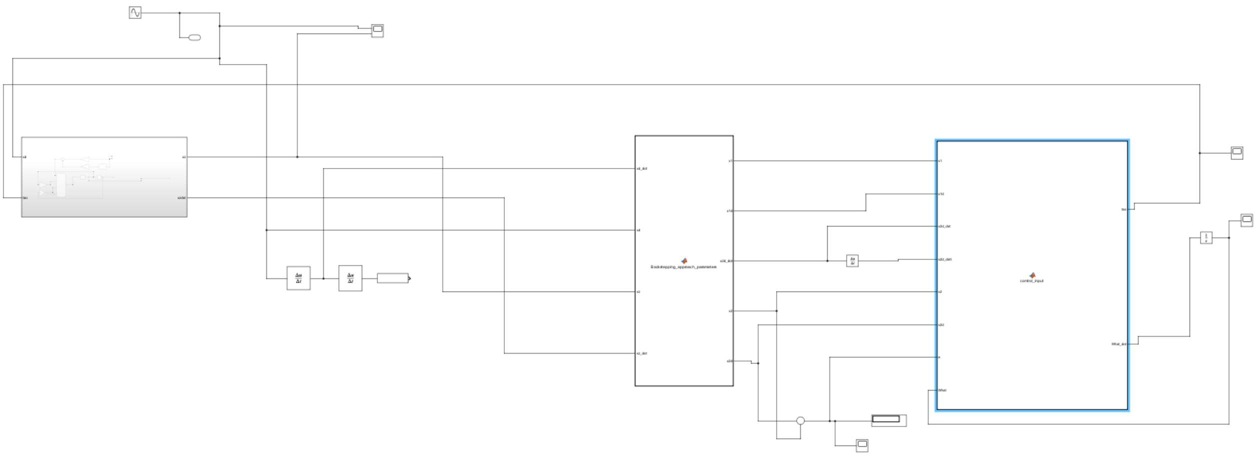

Figure4.3.1 Simulink Model

```
function [tao, What_dot] = control_input(x1, x1d, x2d_dot, x2d_dott, x2, x2d, e, What)

    % Gains
    Kv = 3;
    Kg = 5;
    ro_epilson=2;
    % Error Definitions
    edot = -Kv * e;
    Uv = x2d_dot + Kv * e;
    ev = Uv - x1;

   xv=[x2d_dott;edot;x1;x1d;x2; x2d];
   h=tanh(xv);
   What_dot=ev*h';


   % Control Input tao
    tao = What*h+Kg*ev+e+ro_epilson*sign(ev);

end
```

Figure4.3.2 MATLAB function code

5. *Results and Discussion*

5.1) When sinusoidal input applied to Adaptive Controller

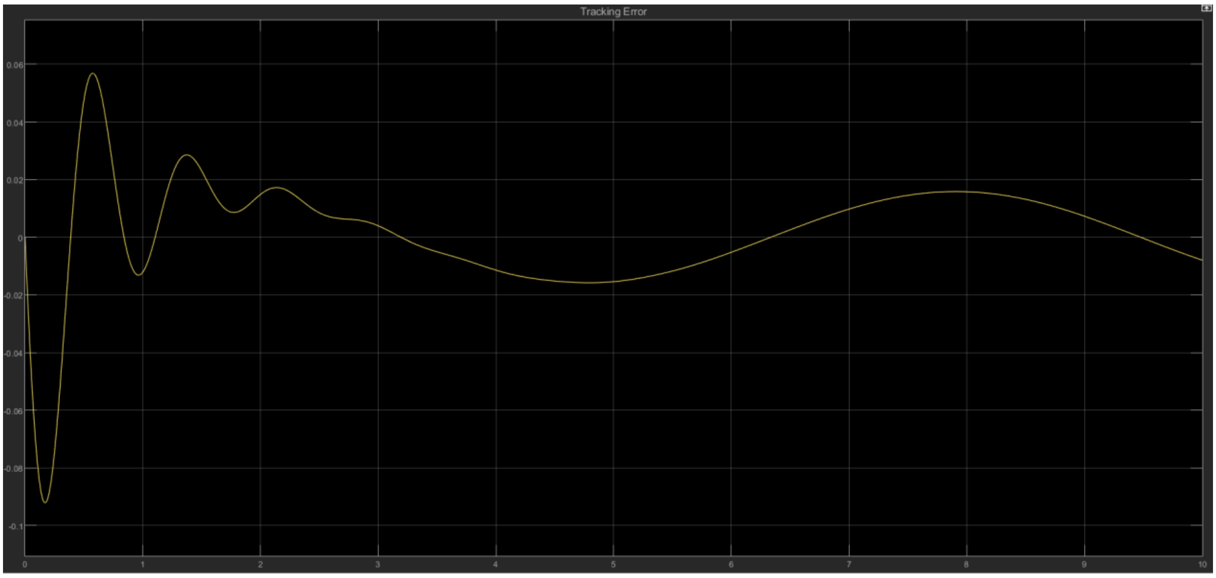

Figure5.1.1 Tracking error graph

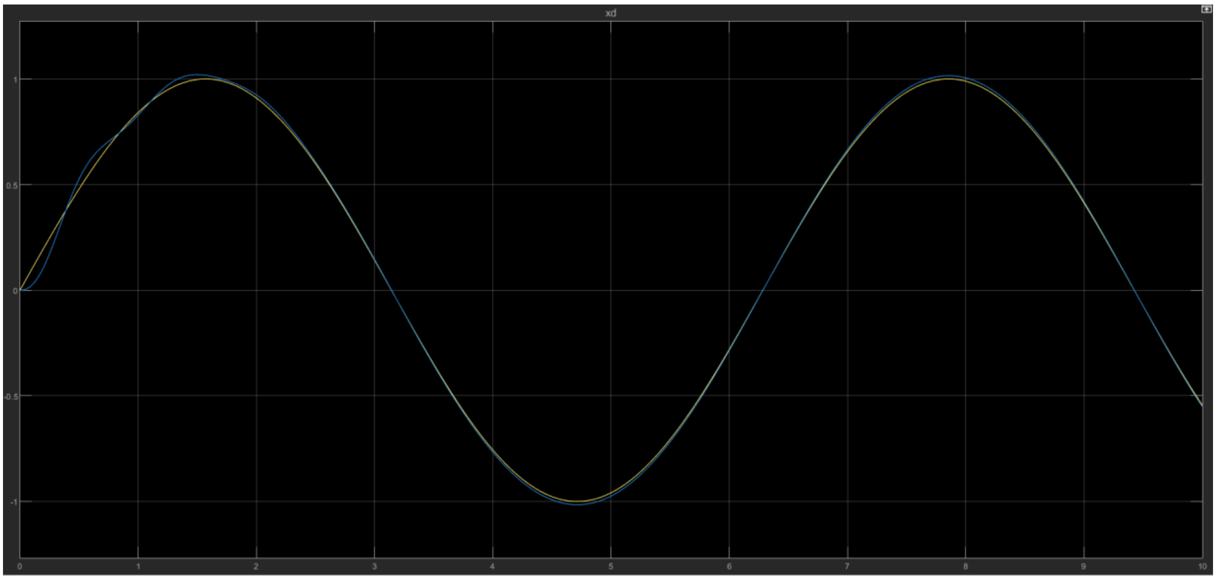

Figure5.1.2 x_z and x_d graph

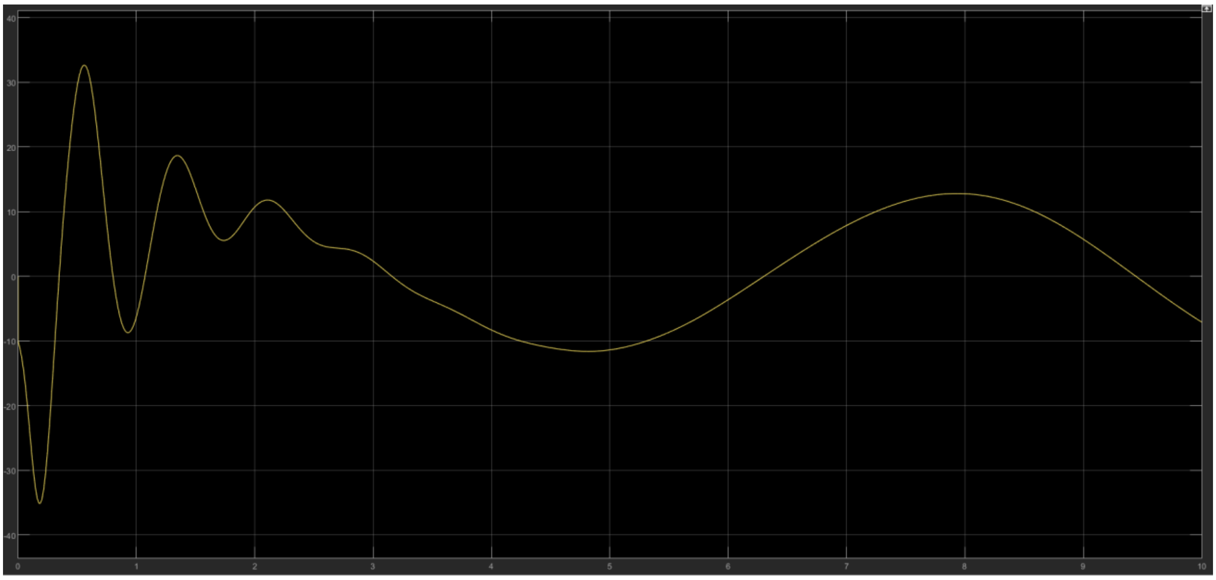

Figure5.1.3 τ graph

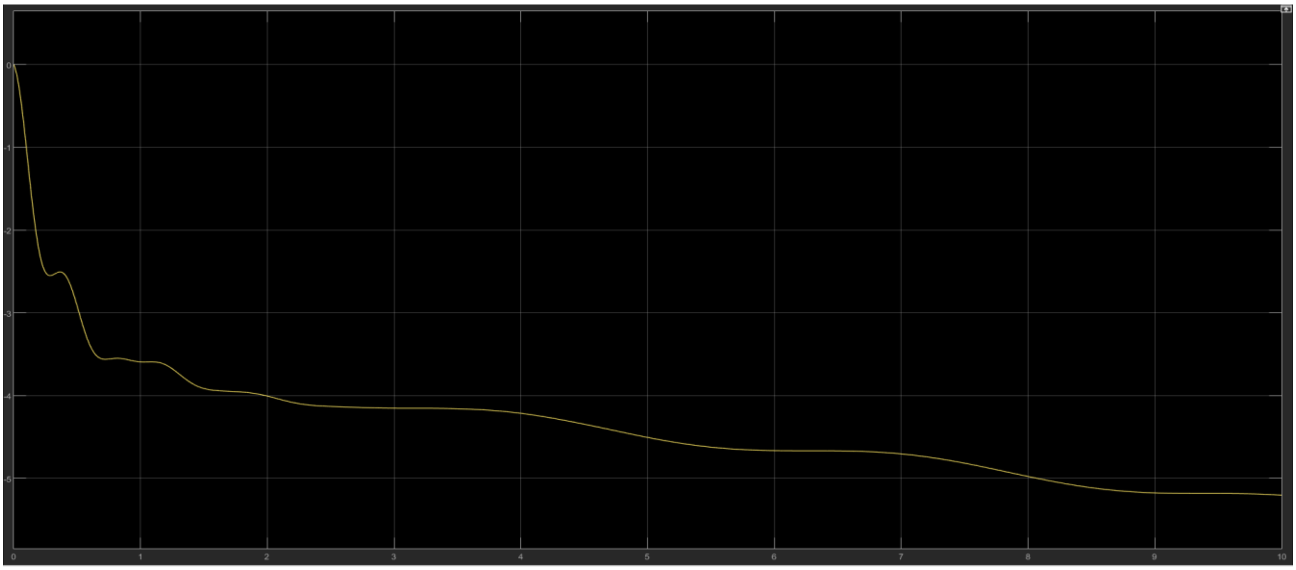

Figure 5.1.4 $\hat{m}$ graph

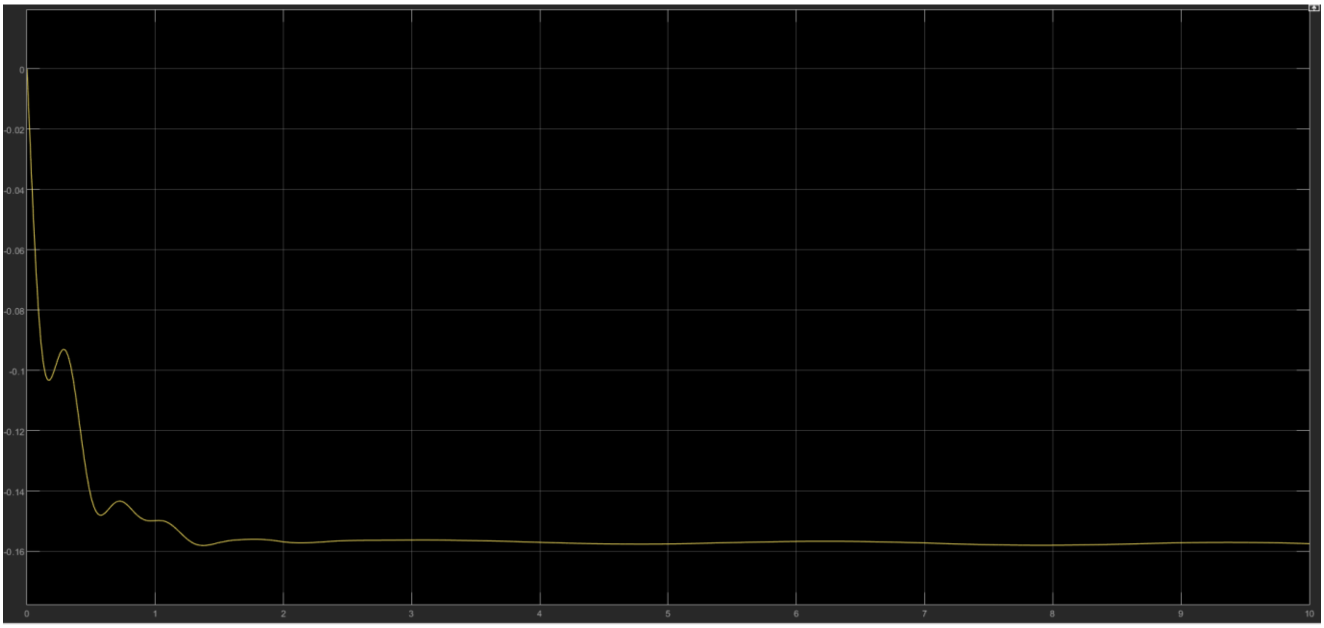

Figure 5.1.5 $\hat{c}$ graph

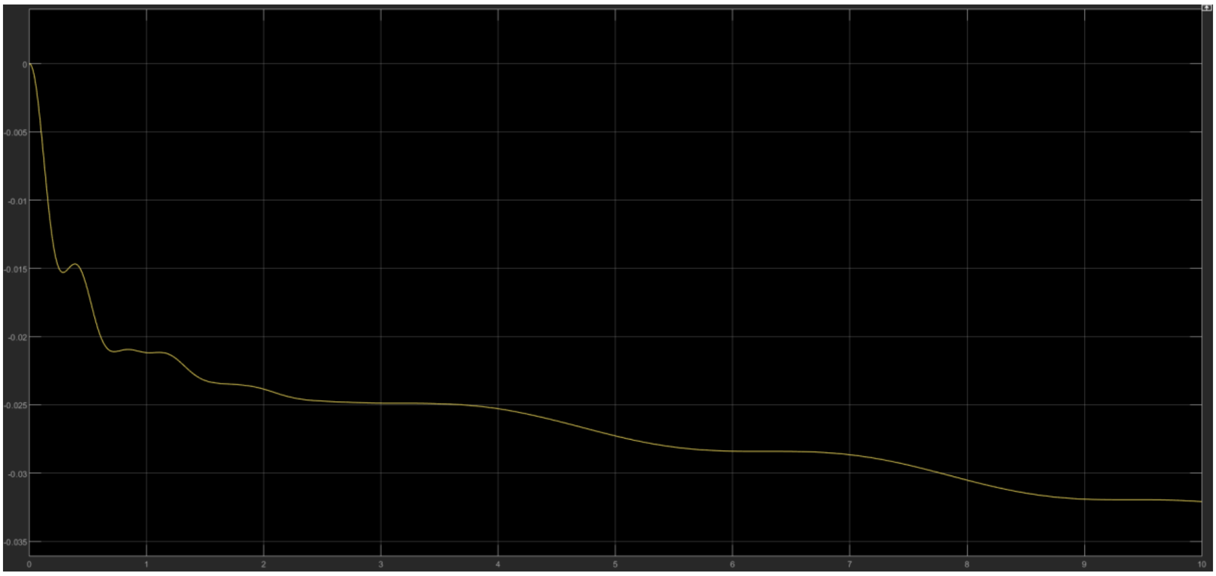

Figure 5.1.6 $\hat{k}$ graph

5.2) When sinusoidal input applied to Linear Parametizaiton Controller

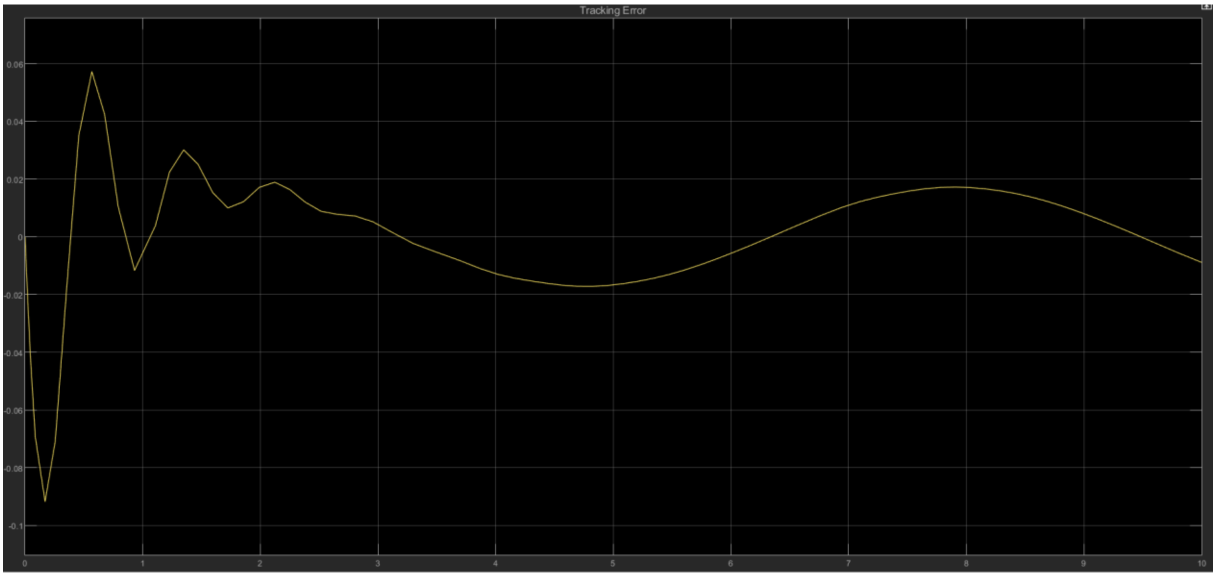

Figure5.2.1 error graph

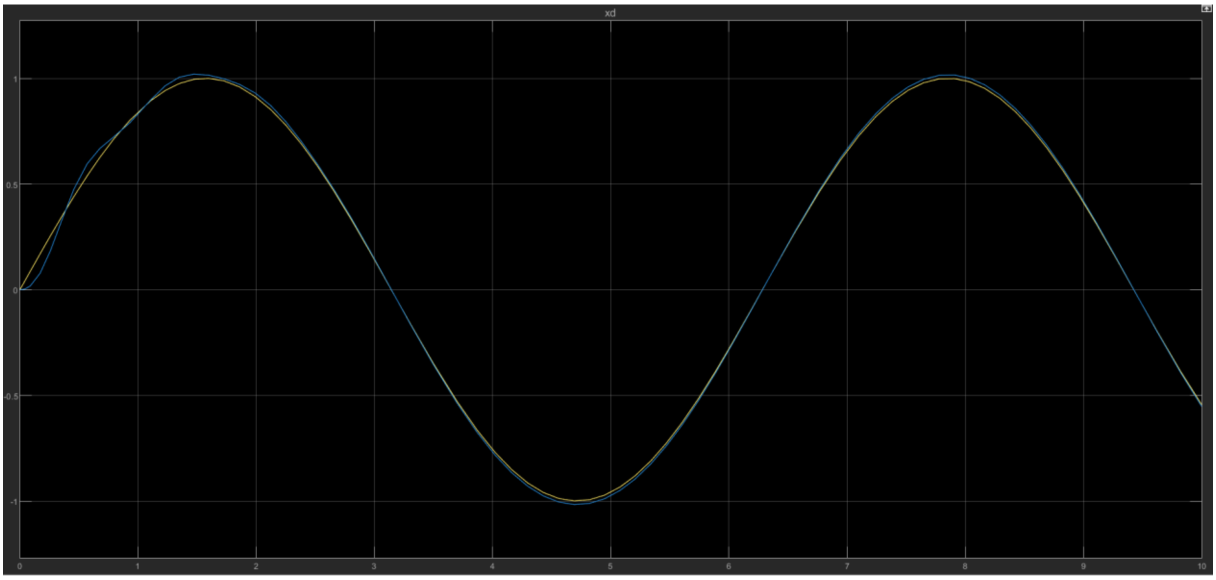

Figure5.2.2 x_z and x_d graph

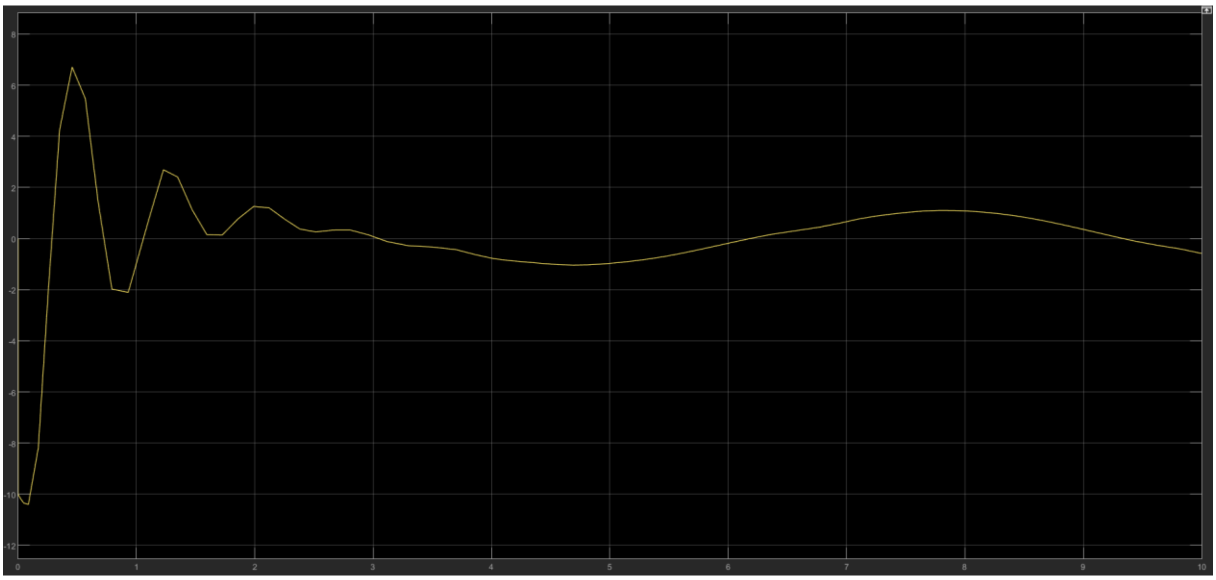

Figure5.2.3 τ graph

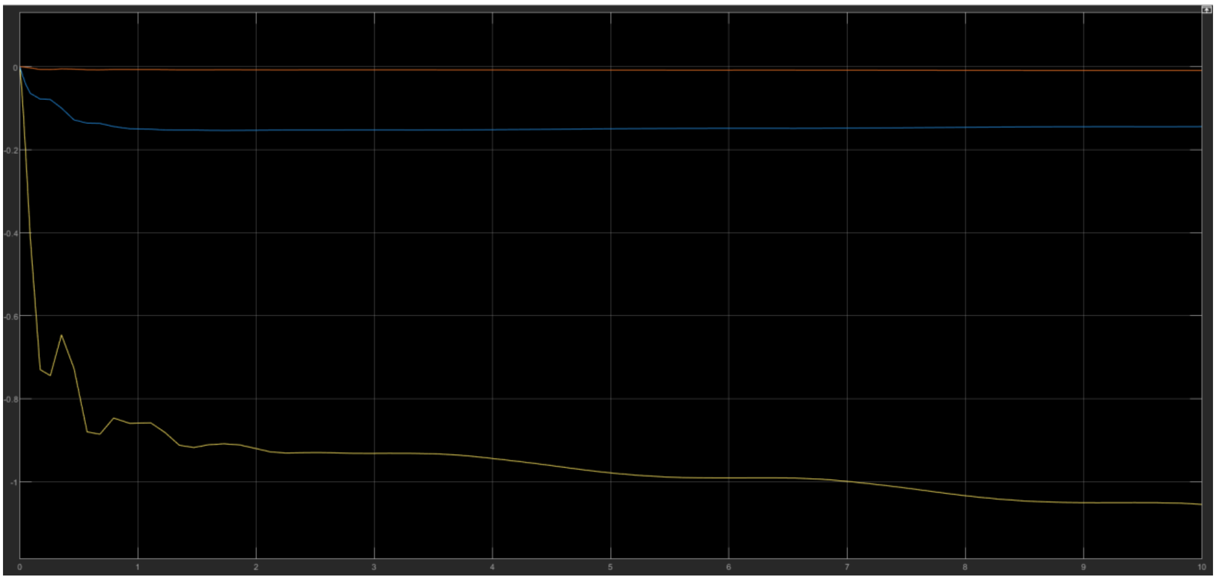

Figure5.2.4 $\hat{Q}$ graph

5.2) When sinusoidal input applied to ANN Based Controller

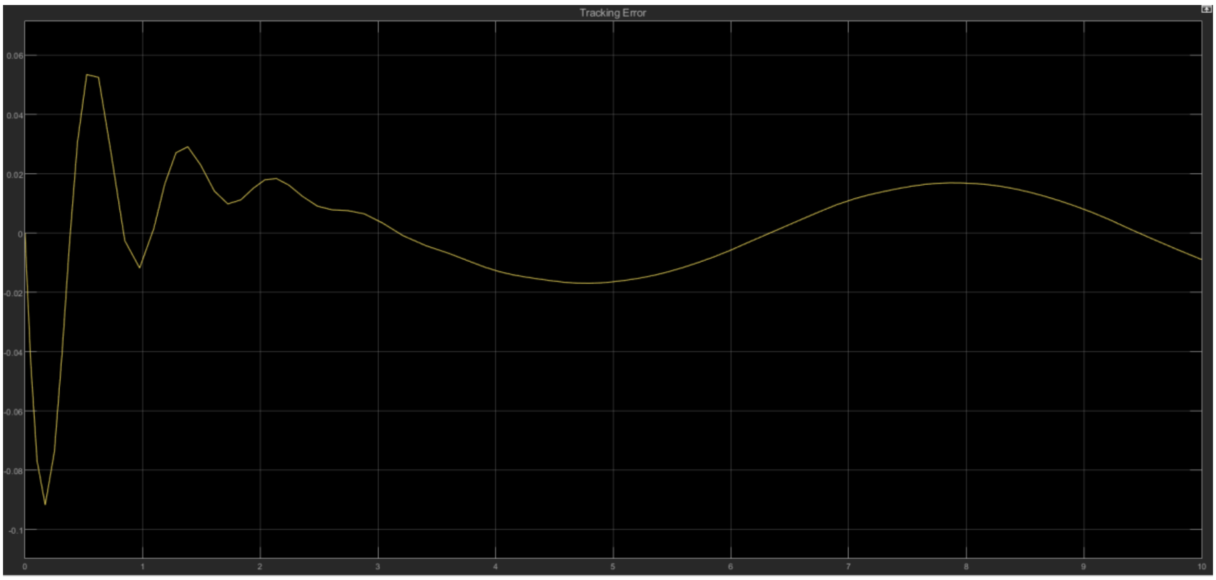

Figure5.3.1 error graph

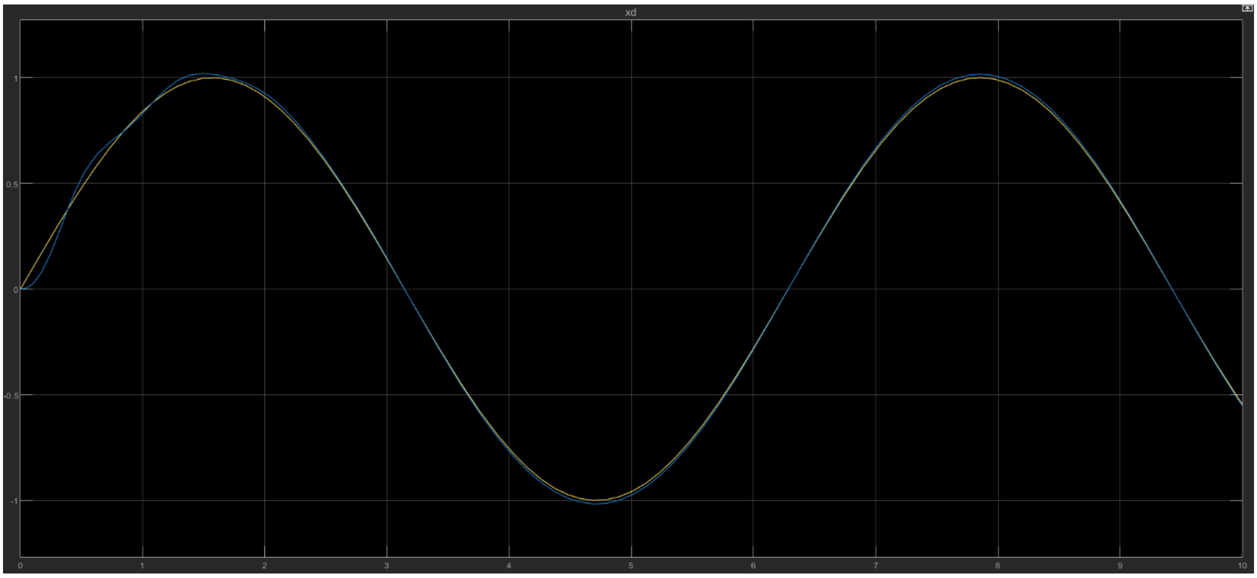

Figure5.3.2 x_z and x_d graph

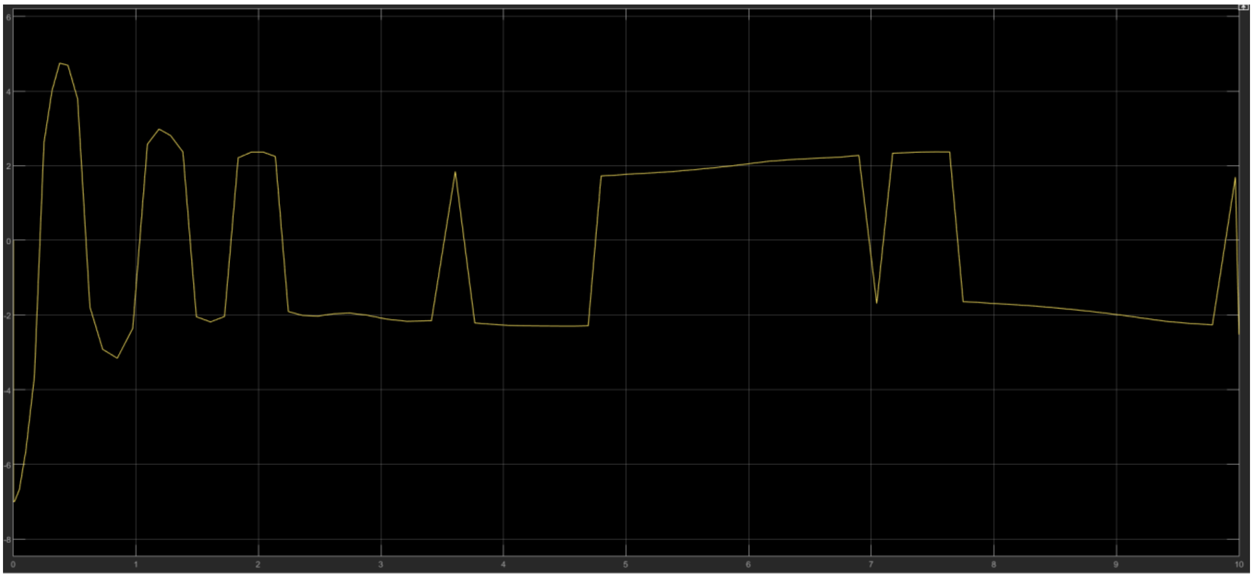

Figure5.3.3 τ graph

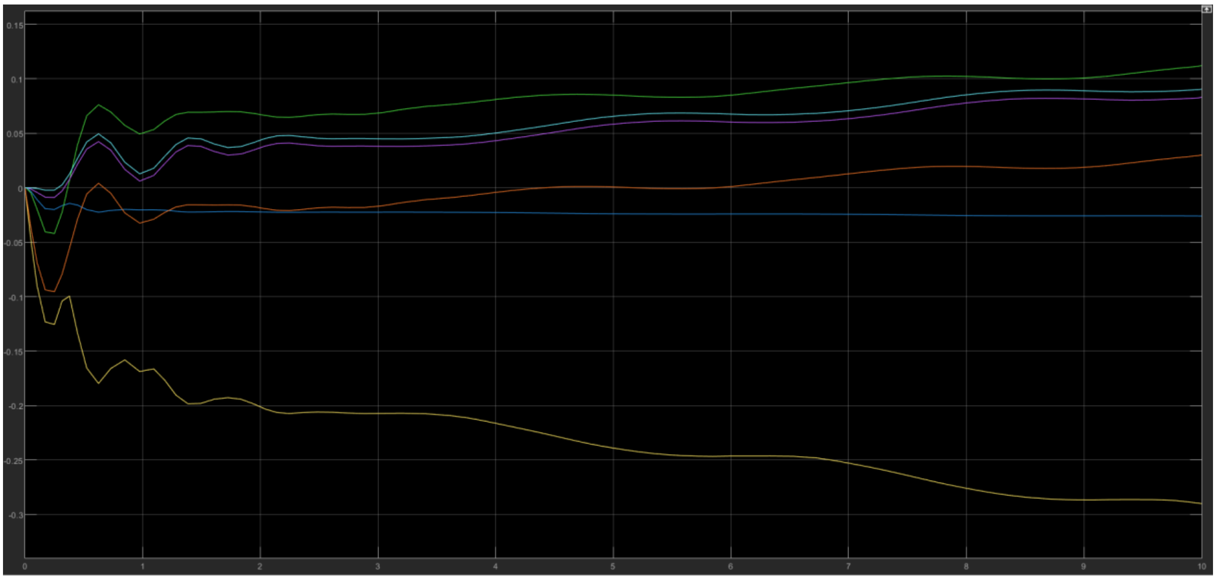

Figure5.3.4 $\hat{w}$ graph

6. Conclusion

The Backstepping approach is a powerful method that enables the systematic and stable design of controllers for nonlinear systems. Its fundamental principle is to analyze the system's dynamic structure step by step and ensure stability at each stage through Lyapunov-based functions. In each control step, a subsystem of the overall system is regulated, and through these steps, the final control input τ\tauτ is constructed. This structure makes it possible to maintain system stability and performance even in the presence of unknown parameters, external disturbances, or modeling inaccuracies. Moreover, the Backstepping framework is highly suitable for integration with complementary strategies such as adaptive control and artificial neural networks (ANNs). In this study, these advantages were leveraged to design and implement three distinct controllers: an adaptive controller, a linearly parameterized controller, and an ANN-based controller.

Simulation results for the adaptive controller showed that the tracking error (Figure 5.1.1) was initially high but rapidly converged toward zero. This indicates that the controller successfully estimated the unknown parameters of the system in real-time and adapted accordingly. In terms of tracking performance, the system output closely followed the reference signal (Figure 5.1.2). The control input τ\tauτ (Figure 5.1.3) exhibited noticeable fluctuations in the early stages of adaptation but stabilized over time. Furthermore, the parameter estimation graphs (Figures 5.1.4–5.1.6) confirmed that the adaptive mechanism accurately learned the dynamic behavior of the system.

In the case of the linearly parameterized controller, the tracking error (Figure 5.2.1) quickly decreased and approached zero, demonstrating effective control under fixed model assumptions. Although this controller lacks a learning mechanism, it performed well in systems with stable parameters. The system output successfully followed the reference signal (Figure 5.2.2), and the control input τ\tauτ (Figure 5.2.3) was smoother and less oscillatory compared to the adaptive controller. This controller is particularly suitable for systems with relatively constant dynamics, where robustness to changing parameters is not a priority.

For the ANN-based controller, the learning process was directly reflected in the simulation results. The error signal (Figure 5.3.1) was initially high, and the control input τ\tauτ (Figure 5.3.3) was unstable due to the ANN's lack of prior knowledge about the system. However, as training progressed, the ANN learned the system behavior, reduced the tracking error significantly, and produced a much more stable control signal. Over time, the system output tracked the reference signal with high precision (Figure 5.3.2). The key advantage of the ANN-based controller is its ability to operate without any prior information about the system, relying entirely on data-driven learning. The weight update behavior of the neural network (Figure 5.3.4) further validated the success of the learning process.

When comparing the control inputs across all controllers, the adaptive controller initially produced high-frequency oscillations that gradually stabilized, the linearly parameterized controller generated the smoothest and most stable control input due to its static nature, and the ANN-based controller produced unstable signals early on but ultimately achieved both stability and precision after the learning phase. These differences illustrate that each control structure presents unique advantages and trade-offs in terms of learning speed, robustness, and signal

smoothness.
In conclusion, the Backstepping approach proves to be a highly effective framework for nonlinear system control. It offers theoretical guarantees of stability through Lyapunov-based analysis, facilitates step-by-step development of control laws, and integrates seamlessly with adaptive and learning-based strategies. The controllers developed in this project demonstrated strong theoretical and simulation-based performance under both known and uncertain conditions. These outcomes highlight why the Backstepping method is widely adopted in engineering applications and emphasize its versatility, adaptability, and robustness in modern control system design.

7. References

[1] BIDIKLI, B. (2019). A Gain Tuning Method for Active Vibration Control of 1 Degree-of-Freedom Mechanical Systems: A Proportional-Integral-Derivative Control Approach.

[2] Slotine, J. J. E., C Li, W. (1991). *Applied nonlinear control* (Vol. 199, No. 1, p. 705). Englewood Cliffs, NJ: Prentice hall.

[3] Khalil, H. K., C Grizzle, J. W. (2002). *Nonlinear systems* (Vol. 3). Upper Saddle River, NJ: Prentice hall.

[4] Turker, T., Gorgun, H., Zergeroglu, E., C Cansever, G. (2007, May). Exact model knowledge and direct adaptive controllers on ball and beam. In *2007 IEEE International Conference on Mechatronics* (pp. 1-6). IEEE.