

Design and Control of a VTOL Aircraft: Fixed-Wing and Quadrotor Integration for Autonomous Flight

Prepared By

Muhammad Ayman	201801356
Ahmed Medhat	201801207
Dena Qatry	201800642
Areeg Ayman	201801902
Mariam Wagdy	201801585

Supervised By

Dr. Mustafa Abdallah

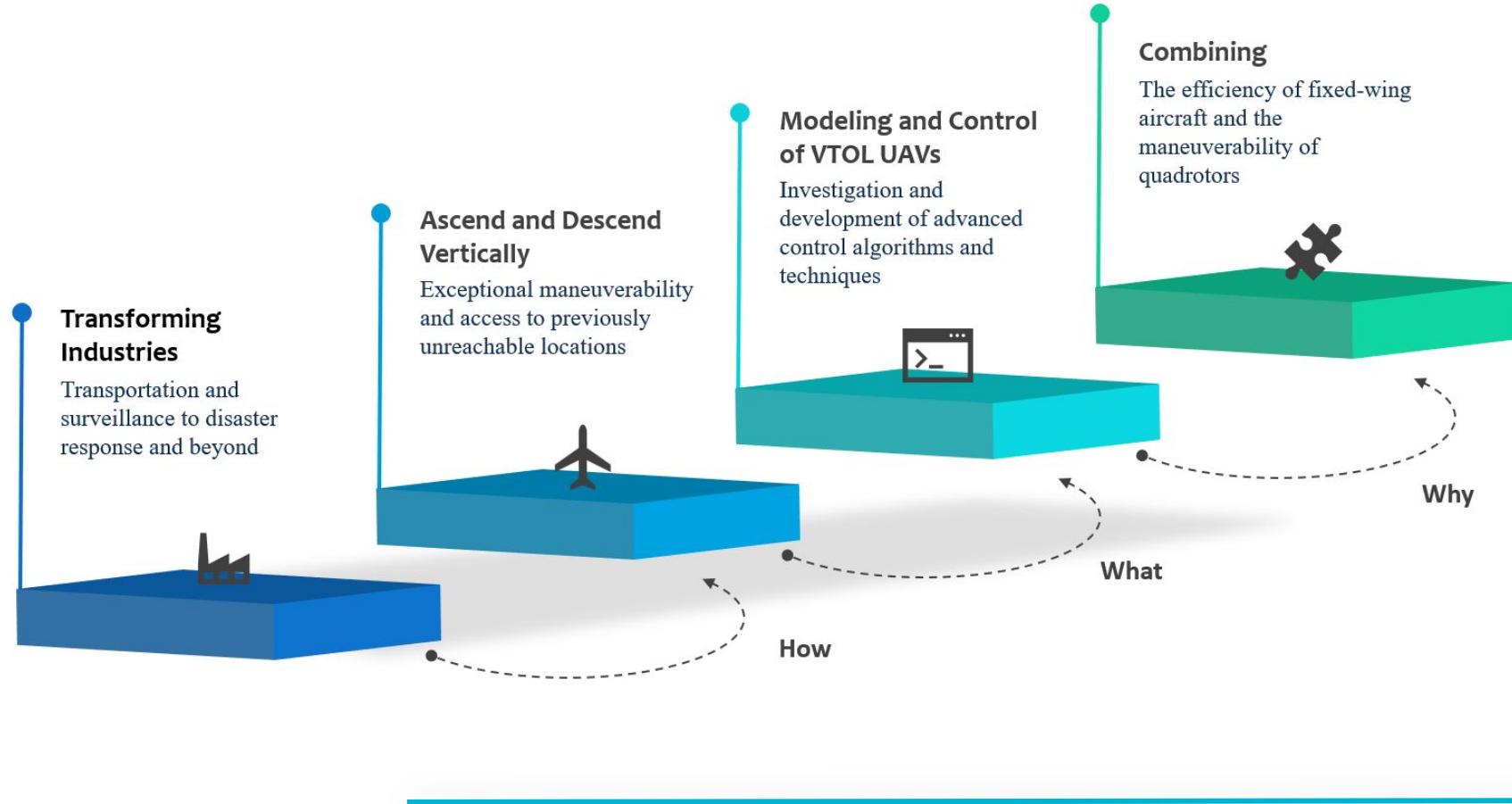
Outline



1. Introduction
2. Aerodynamic Stability Derivatives
3. VTOL Modeling
4. Modeling Verification
5. Autopilot Controller Design and Implementation
6. Conclusion and Future Work

1. Introduction

1. Introduction



2. Aerodynamic Stability Derivatives

2. Aerodynamic Stability Derivatives



Introduction

- Tornado is a MATLAB-based 3D-vortex lattice program.
- Used to calculate aerodynamic coefficients, 3D forces, and stability derivatives.
- Assume small angle of attack.
- Fuselage effect, frictional drag, compressibility, and thickness of the lifting surfaces are not considered.
- Requires detailed geometry and detailed flight state.
- Results for control surface derivatives, aerodynamic derivatives, aerodynamic forces at trim conditions.

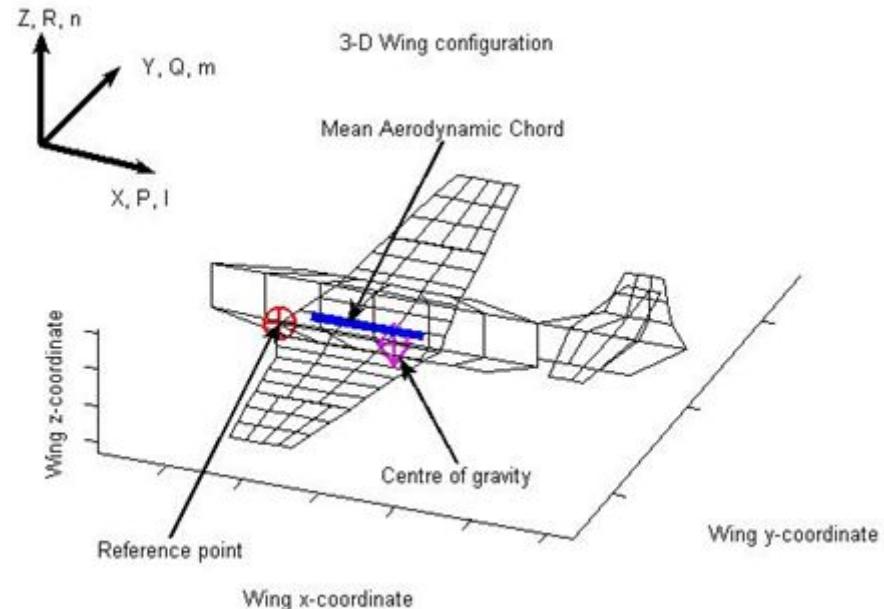


Figure 2.1. Definition of Tornado Coordinate System

2. Aerodynamic Stability Derivatives



Geometry Plots

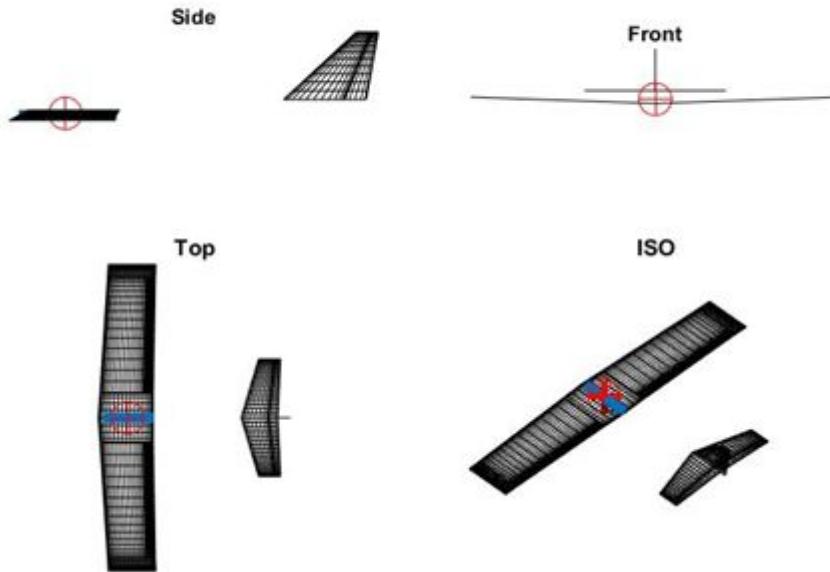


Figure 2.2. Panels Distribution in Tornado

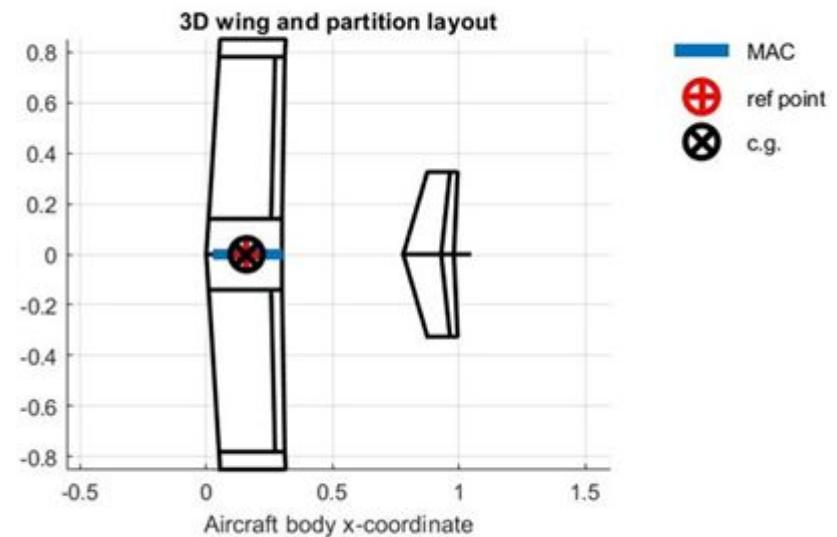


Figure 2.3. Tornado Geometry Plots

2. Aerodynamic Stability Derivatives



Results: Aerodynamic Coefficients Dependency on the Angle of Attack (α)

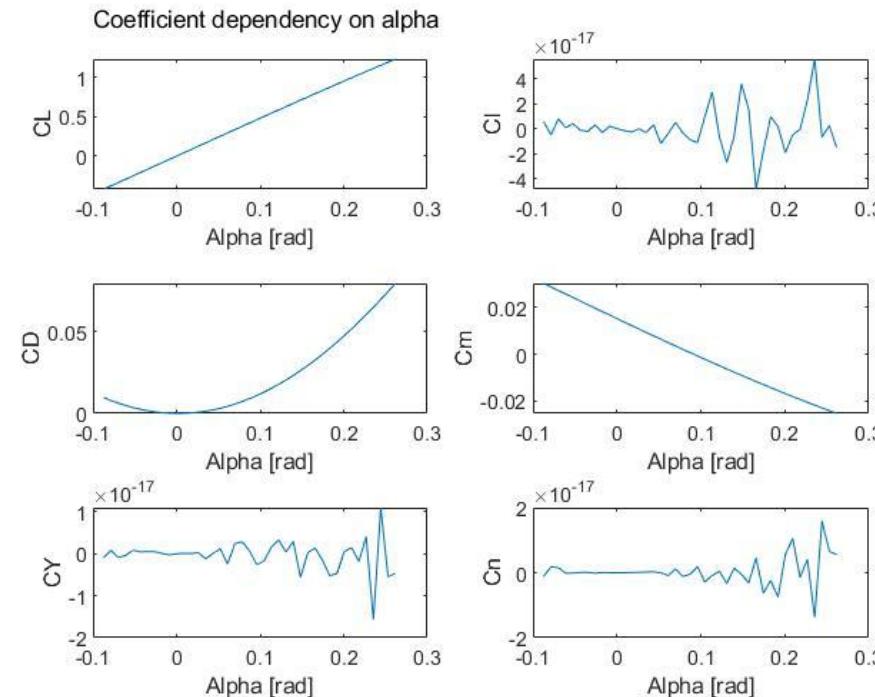


Figure 2.4. The Aerodynamic Coefficients Dependency on the Angle of Attack (α)

2. Aerodynamic Stability Derivatives



Results: Aerodynamic Coefficients Dependency on the Elevator δe

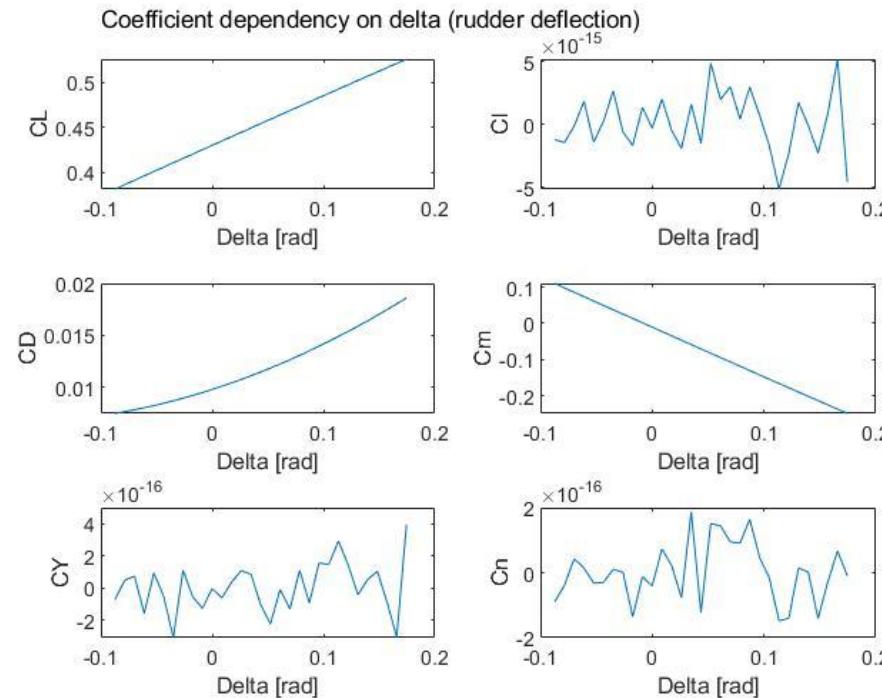


Figure 2.4. The Aerodynamic Coefficients Dependency on the Elevator (δe)

3. VTOL Modeling (Fixed Wing UAV)

I. Introduction



- The "Flight Stability and Automatic Control" reference.
- Aircraft's mass is believed to be fixed and it is presumed to be a rigid body.
- Aircraft's motion is decoupled into longitudinal and lateral modes.
- Assuming a steady flight with small perturbations around the equilibrium point.
- Using first order Taylor series expansion.

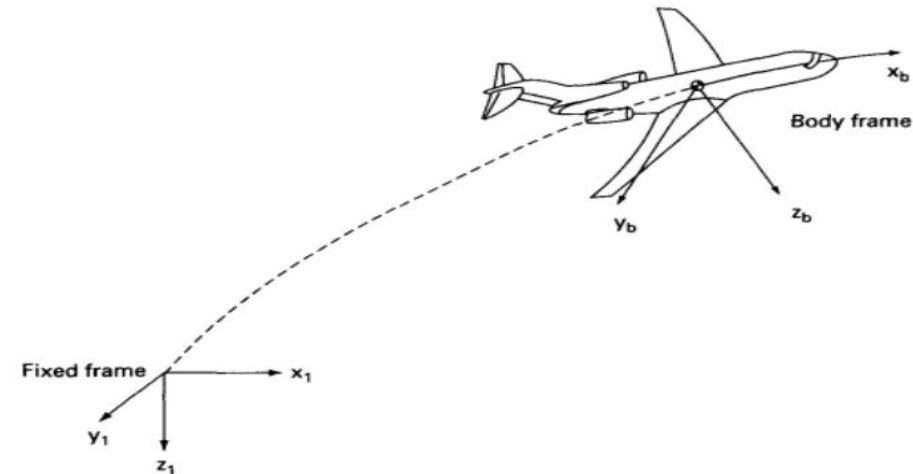


Figure 3.1. Inertial and body frames identification

II. Equations of motion



$$X - mgS\theta = m(u' + qw - rv) \quad \text{Force Equations}$$

$$Y + mgC\theta S\varphi = m(v' + ru - pw)$$

$$Z + mgC\theta C\varphi = m(w' + pv - qu)$$

$$L = I_x p' - I_{xz} r' + qr(I_z - I_y) - I_{xz} pq \quad \text{Moment Equations}$$

$$M = I_y q' + rq(I_x - I_z) + I_{xz}(p^2 - r^2)$$

$$N = -I_{xz} p' + I_z r' + pq(I_y - I_x) + I_{xz} qr$$

II. Equations of motion



$$p = \dot{\varphi} - \dot{\psi}S\theta$$

Body Angular Velocities in Terms of Euler Angles and Euler Rates

$$q = \dot{\theta}C\varphi + \dot{\psi}C\theta S\varphi$$

$$r = \dot{\psi}C\theta C\varphi - \dot{\theta}S\varphi$$

$$\dot{\theta} = qC\varphi - rS\varphi$$

Euler Rates in Terms of Euler Angles and Body Angular Velocities

$$\dot{\varphi} = p + qS\varphi T\theta + rC\varphi T\theta$$

$$\dot{\psi} = (qS\varphi + rC\varphi)sec\theta$$

III. Linear Equations



Longitudinal Equations

$$\begin{aligned} & \left(\frac{d}{dt} - X_u \right) \Delta u - X_w \Delta w + (g C \theta_o) \Delta \theta = X_{\delta_e} \Delta \delta_e + X_{\delta_T} \Delta \delta_T \\ & - Z_u \Delta u + [(1 - Z_w) \frac{d}{dt} - Z_w] \Delta w - [(u_o + Z_q) \frac{d}{dt} - g S \theta_o] \Delta \theta = Z_{\delta_e} \Delta \delta_e + Z_{\delta_T} \Delta \delta_T \\ & - M_u \Delta u - (M_w \frac{d}{dt} + M_w) \Delta w + (\frac{d^2}{dt^2} - M_q \frac{d}{dt}) \Delta \theta = M_{\delta_e} \Delta \delta_e + M_{\delta_T} \Delta \delta_T \end{aligned}$$

Lateral Equations

$$\begin{aligned} & \left(\frac{d}{dt} - Y_v \right) \Delta v - Y_p \Delta p + (u_o - Y_r) \Delta r - (g C \theta_o) \Delta \varphi = Y_{\delta_r} \Delta \delta_r \\ & - L_v \Delta v + (\frac{d}{dt} - L_p) \Delta p - (\frac{I_{yz}}{I_x} \frac{d}{dt} + L_r) \Delta r = L_{\delta_a} \Delta \delta_a + L_{\delta_r} \Delta \delta_r \\ & - N_v \Delta v - (\frac{I_{xz}}{I_z} \frac{d}{dt} + N_p) \Delta p + (\frac{d}{dt} - N_r) \Delta r = N_{\delta_a} \Delta \delta_a + N_{\delta_r} \Delta \delta_r \end{aligned}$$

IV. Parameter Estimation

Longitudinal Mode

$$\dot{x}_{5x1} = A_{5x5}x_{5x1} + B_{5x2}u_{2x1}$$

Where x is a 5-by-1 vector containing the states $\Delta u, \Delta w, \Delta q, \Delta \theta$, and Δh respectively.

$A =$

$$\begin{array}{ccccc}
 x_u & x_w & 0 & -g \cos(\theta_o) & 0 \\
 \frac{Z_u}{1 - Z_{\dot{w}}} & \frac{Z_w}{1 - Z_{\dot{w}}} & \frac{u_o + Z_q}{1 - Z_{\dot{w}}} & \frac{-g \sin(\theta_o)}{1 - Z_{\dot{w}}} & 0 \\
 M_u + \frac{M_{\dot{w}} Z_u}{1 - Z_{\dot{w}}} & M_w + \frac{M_{\dot{w}} Z_w}{1 - Z_{\dot{w}}} & M_q + \frac{M_{\dot{w}}(u_o + Z_u)}{1 - Z_{\dot{w}}} & \frac{-g M_{\dot{w}} \sin(\theta_o)}{1 - Z_{\dot{w}}} & 0 \\
 0 & 0 & 1 & 0 & 0 \\
 -\sin(\theta_o) & -\cos(\theta_o) & 0 & u_o \cos(\theta_o) + w_o \sin(\theta_o) & 0
 \end{array}
 =
 \begin{array}{cccccc}
 -0.03645 & 0.2708 & 0 & -9.7712 & 0 \\
 -1.1133 & -6.3299 & 16.8276 & -0.8839 & 0 \\
 0.26767 & 0.7402 & 12.1758 & 0.2125 & 0 \\
 0 & 0 & 1 & 0 & 0 \\
 -0.090000 & -0.9960 & 0 & 18.0000 & 0
 \end{array}$$

IV. Parameter Estimation



Longitudinal Mode

B =

$$\begin{vmatrix} X_{\delta_e} & X_{\delta_T} \\ \frac{Z_{\delta_e}}{1 - Z_{\dot{\omega}}} & \frac{Z_{\delta_T}}{1 - Z_{\dot{\omega}}} \\ M_{\delta_e} + \frac{M_{\dot{\omega}} Z_{\delta_e}}{1 - Z_{\dot{\omega}}} & M_{\delta_T} + \frac{M_{\dot{\omega}} Z_{\delta_T}}{1 - Z_{\dot{\omega}}} \\ 0 & 0 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 92.3979 \\ -13.1360 & 0 \\ -115.4777 & 3.5190 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$$

IV. Parameter Estimation



Lateral Mode

$$\dot{x}_{5x1} = A_{5x5}x_{5x1} + B_{5x2}u_{2x1}$$

Where x is a 5-by-1 vector containing the states $\Delta\beta, \Delta p, \Delta r, \Delta\phi$, and $\Delta\psi$ respectively.

$A =$

$$\begin{vmatrix}
 \frac{Y_\beta}{u_o} & Y_p & -(u_o - Y_r) & g \cos(\theta_o) & 0 \\
 L_v^* + \frac{I_{xz}}{I_x} N_v^* & L_p^* + \frac{I_{xz}}{I_x} N_p^* & L_r^* + \frac{I_{xz}}{I_x} N_r^* & 0 & 0 \\
 N_v^* + \frac{I_{xz}}{I_z} L_v^* & N_p^* + \frac{I_{xz}}{I_z} L_p^* & N_r^* + \frac{I_{xz}}{I_z} L_r^* & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0
 \end{vmatrix} = \begin{vmatrix}
 -3.8190 & -0.0007535 & -17.7590 & 9.7717 & 0 \\
 7.1050 & -5.1799 & 0.07723 & 0 & 0 \\
 11.2492 & -0.3579 & -0.7479 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0
 \end{vmatrix}$$

IV. Parameter Estimation



Lateral Mode

B =

$$\begin{vmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a}^* + \frac{I_{xz}}{I_x} N_{\delta_a}^* & L_{\delta_r}^* + \frac{I_{xz}}{I_x} N_{\delta_r}^* \\ N_{\delta_a}^* + \frac{I_{xz}}{I_z} L_{\delta_a}^* & N_{\delta_r}^* + \frac{I_{xz}}{I_z} L_{\delta_r}^* \\ 0 & 0 \\ 0 & 0 \end{vmatrix} = \begin{vmatrix} 0 & -2.3832 \\ 80.02653 & 0.8649 \\ 6.1344 & -7.6614 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$$

V. Flying Qualities



The UAV used in this project is a Level 1, Class 1, and a Category C aircraft.

- Clearly suitable flying abilities for the mission flight phase.
- Small, light aircraft.
- Progressive maneuvers and precise flight-path control.

Damping ξ	
<i>Level 1 Flying Qualities</i>	> 0.04
<i>Project's UAV</i>	0.0918

Table 3.1. Long Mode Damping of UAV and Link to Flying Qualities

Damping ξ	
<i>Level 1 Flying Qualities</i>	$0.35 < \xi < 1.3$
<i>Project's UAV</i>	1

Table 3.2. Short Mode Damping of UAV and Link to Flying Qualities

3. VTOL Modeling (Quadrotor)

I. Linear Rate Transformation



Inertial Fixed Frame → Quadrotor Body Frame

This transformation is defined by euler angles as shown (z→y→x):

$$R_{x,y,z}(\phi, \theta, \psi) = R_x(\phi)R_y(\theta)R_z(\psi)$$

Where,

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad R_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad R_z = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Accordingly,

$$d/dt[x_n, y_e, z_d] = R_{x,y,z}(\varphi, \theta, \psi)[u, v, w]$$

II. Angular Rate Transformation



Inertial Fixed Frame → Quadrotor Body Frame

*Transformation relates the quadrotor angular velocities to angular velocities & attitude rates wrt the datum axes.

*The transformation is done by first applying **roll** with angle ϕ and angular velocity $d/dt \phi$, followed by a **pitch** with angle θ and angular velocity $d/dt \theta$, followed by a **yaw** with angle Ψ and angular velocity $d/dt \Psi$.

This results in:

$$p = \dot{\phi} - \dot{\psi} \sin \theta$$

$$q = \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta$$

$$r = \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi$$



$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

III. Linear Acceleration



*The linear acceleration is defined in the inertial fixed frame by Newton's Second Law as:

$$F = m d/dt (V)$$

Where m is the mass and V is the velocity vector $[u, v, w]$.

*As the quadrotor moves & rotates, the velocity vector is rotated. So, the derivative of the velocity vector needs to account for the **rotational speed** as well as the **rate of the change of the velocity components**.

$$F = m d/dt (V) + \omega \times mV$$



$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

III. Linear Acceleration



Force Analysis

*The forces acting on the quadrotor are the lift (thrust) force and weight force.

$$T = k(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$

Accordingly,

$$F_x = 0, F_y = 0, F_z = mg - T$$

Transforming the force vector from inertial frame to body frame:

$$F_b = R_{x,y,z}(\phi, \theta, \psi). F$$



$$\begin{aligned}\dot{u} &= rv - qw - g \sin \theta \\ \dot{v} &= pw - ru + g \cos \theta \sin \phi \\ \dot{w} &= qu - pv + g \cos \phi \cos \theta - \frac{T}{m}\end{aligned}$$

IV. Angular Acceleration



*The rate of change of angular momentum equals the resultant moment.

$$M = d/dt(H)$$

*Hence the quadrotors changes in direction which implies that the angular momentum change in direction:

$$M = d/dt(H) + \omega \times H$$

where,

$$H = I \omega$$

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}$$



$$\begin{aligned} M_x &= \dot{p} I_x + qr(I_z - I_y) \\ M_y &= \dot{q} I_y + pr(I_x - I_z) \\ M_z &= \dot{r} I_z + pq(I_y - I_x) \end{aligned}$$

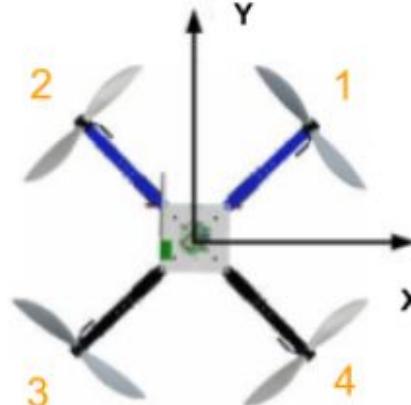
IV. Angular Acceleration Cont'd

*The external moments are just the torques generated by quadrotor motors, that are expressed as:

$$M_x = lk(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

$$M_y = lk(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2)$$

$$M_z = d(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$



*Equating the equations obtained results in the remaining states:

$$\dot{p} = \frac{I_y - I_z}{I_x} qr + \frac{1}{I_x} M_x$$

$$\dot{q} = \frac{I_z - I_x}{I_y} pr + \frac{1}{I_y} M_y$$

$$\dot{r} = \frac{I_x - I_y}{I_z} pq + \frac{1}{I_z} M_z$$

4. Model Verification (Fixed Wing UAV)

I. Kinematics Model (Nonlinear)



Aircraft Uncontrolled Response

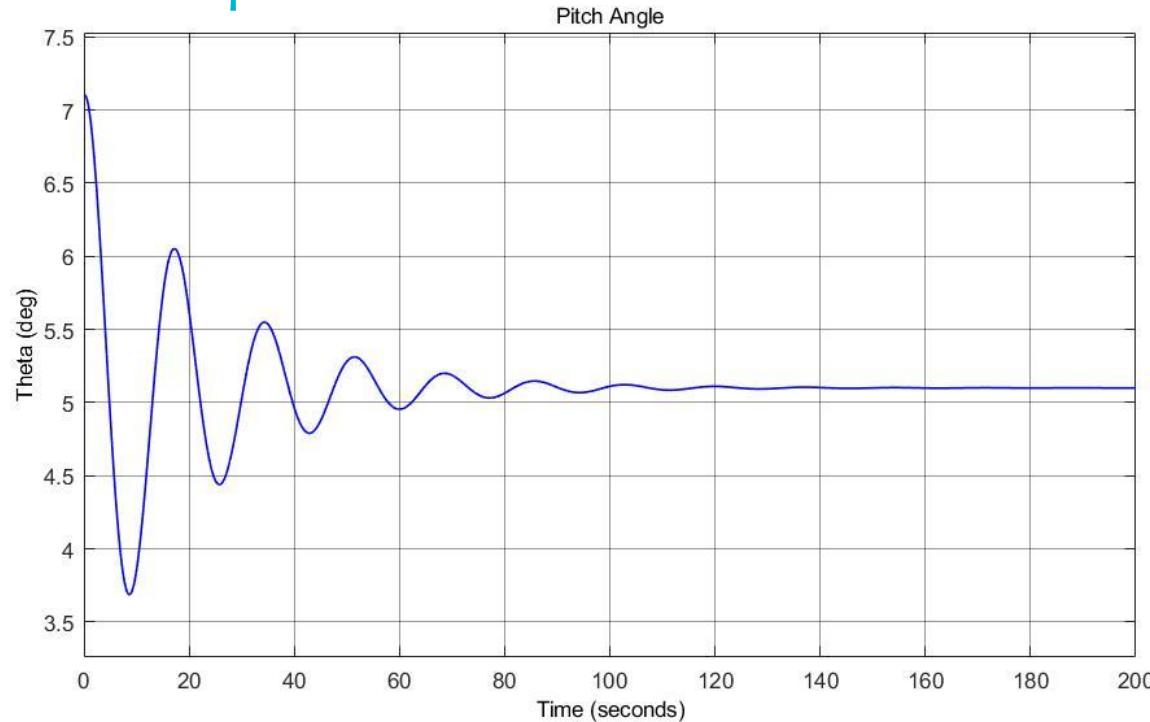


Figure 4.1. Pitch Response due to +2 deg Initial Value in Pitch Angle (θ)

Settling time is 87.26 seconds and the overshoot is 27.78%

I. Kinematics Model (Nonlinear)



Aircraft Uncontrolled Response

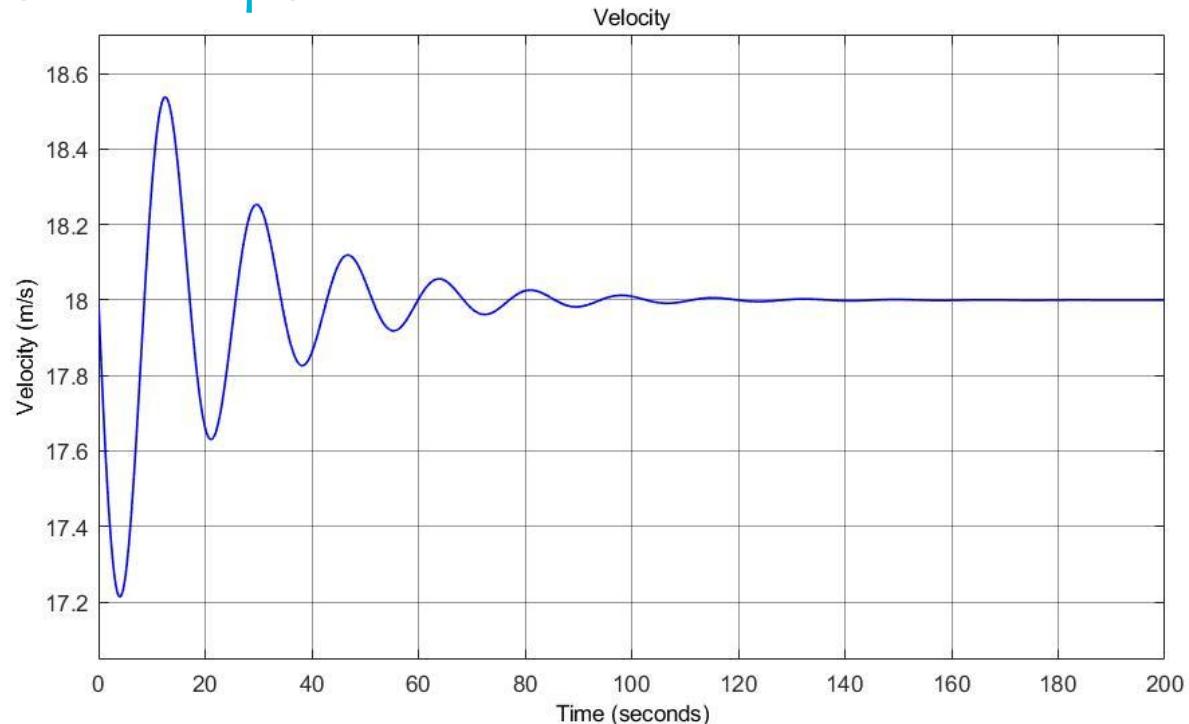


Figure 4.2. Velocity Response due to +2 deg Initial Value in Pitch Angle (θ)

Settling time is 91.90 seconds and the overshoot is 4.39%

I. Kinematics Model (Nonlinear)



Aircraft Uncontrolled Response

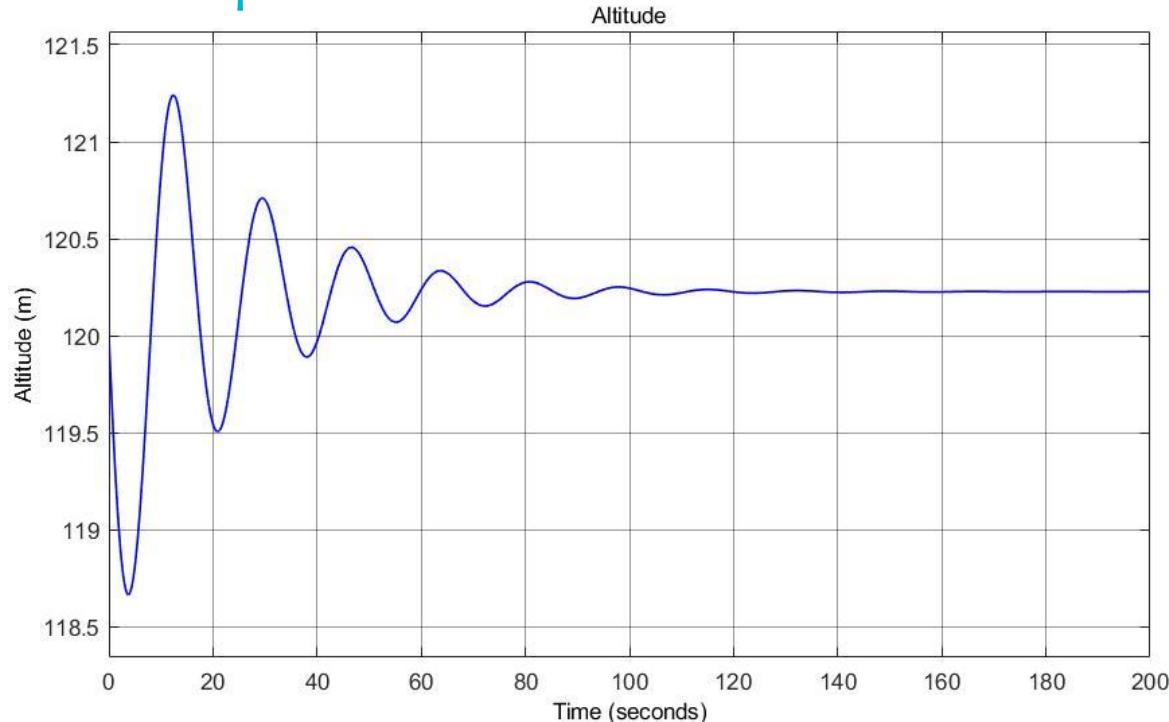


Figure 4.3. Altitude Response due to +2 deg Initial Value in Pitch Angle (θ)

Settling time is 86.62 seconds, the overshoot is 2.67%, and the steady state error is 1.77 meters.

I. Kinematics Model (Nonlinear)



Aircraft Uncontrolled Response

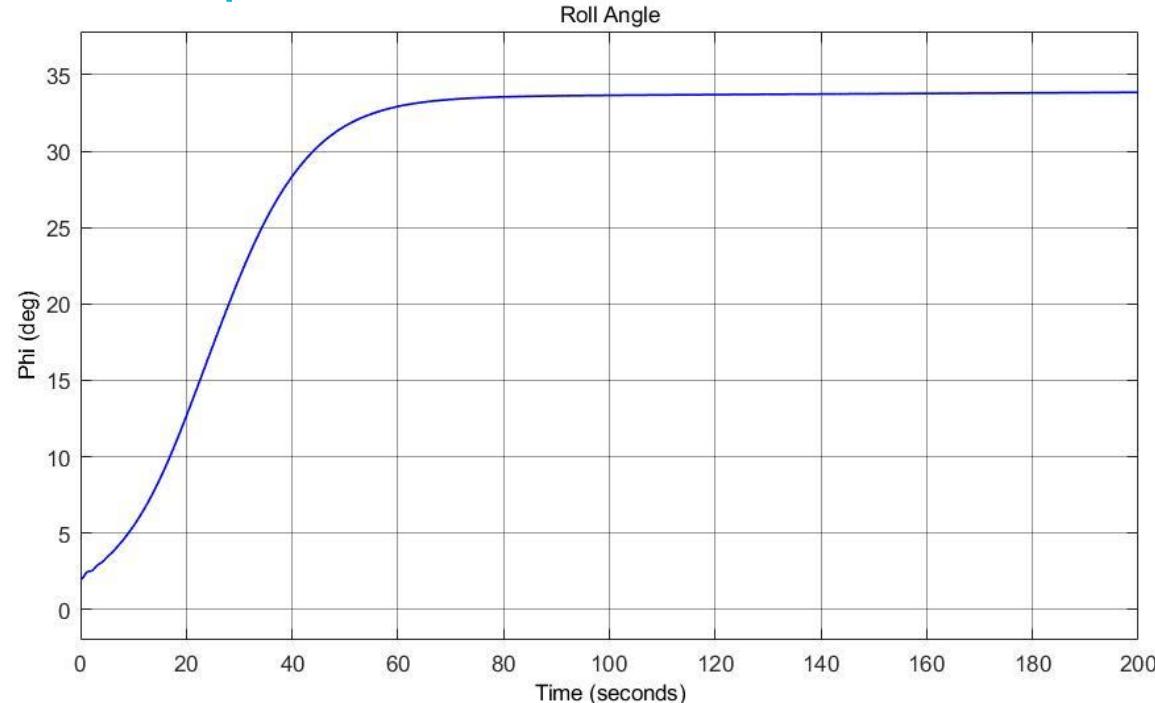


Figure 4.4. Roll Response due to +2 deg Initial Value in Roll Angle (ϕ)

I. Kinematics Model (Nonlinear)



Aircraft Uncontrolled Response

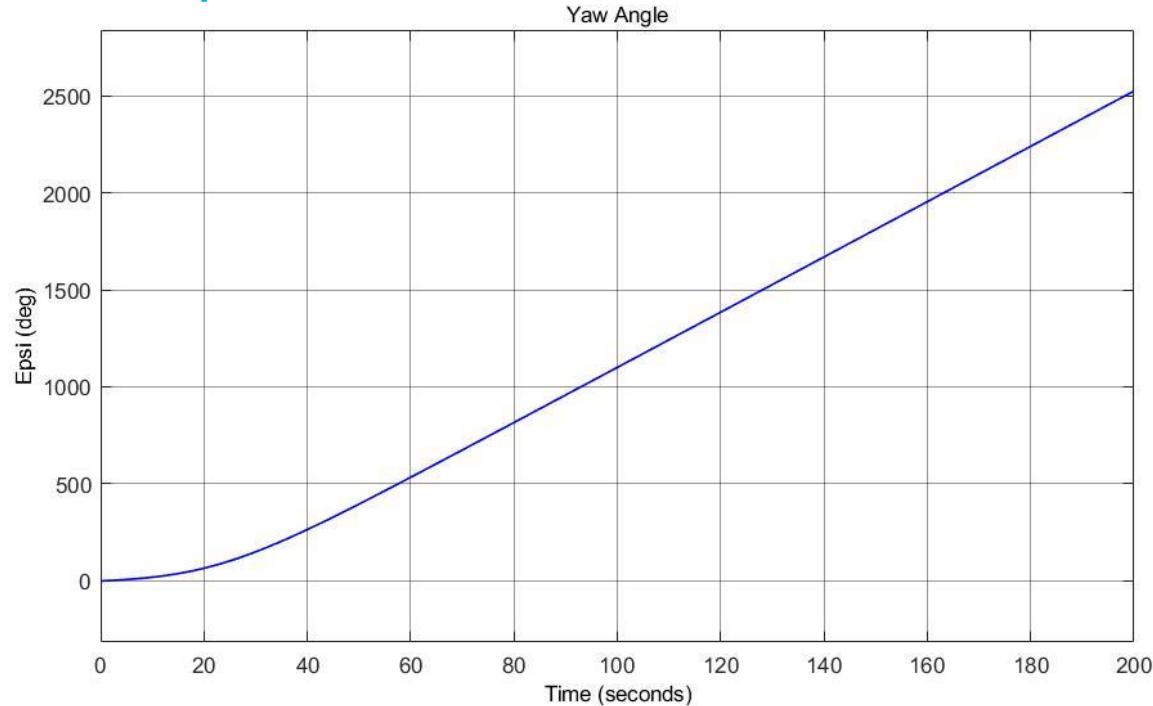


Figure 4.5.. Yaw Response due to +2 deg Initial Value in Roll Angle (ϕ)

II. Linear Model

- Quick approximations for validating the nonlinear model.
- By six simultaneous nonlinear second order differential equations.
- Same response for more disturbances validate the accuracy of the nonlinear model.

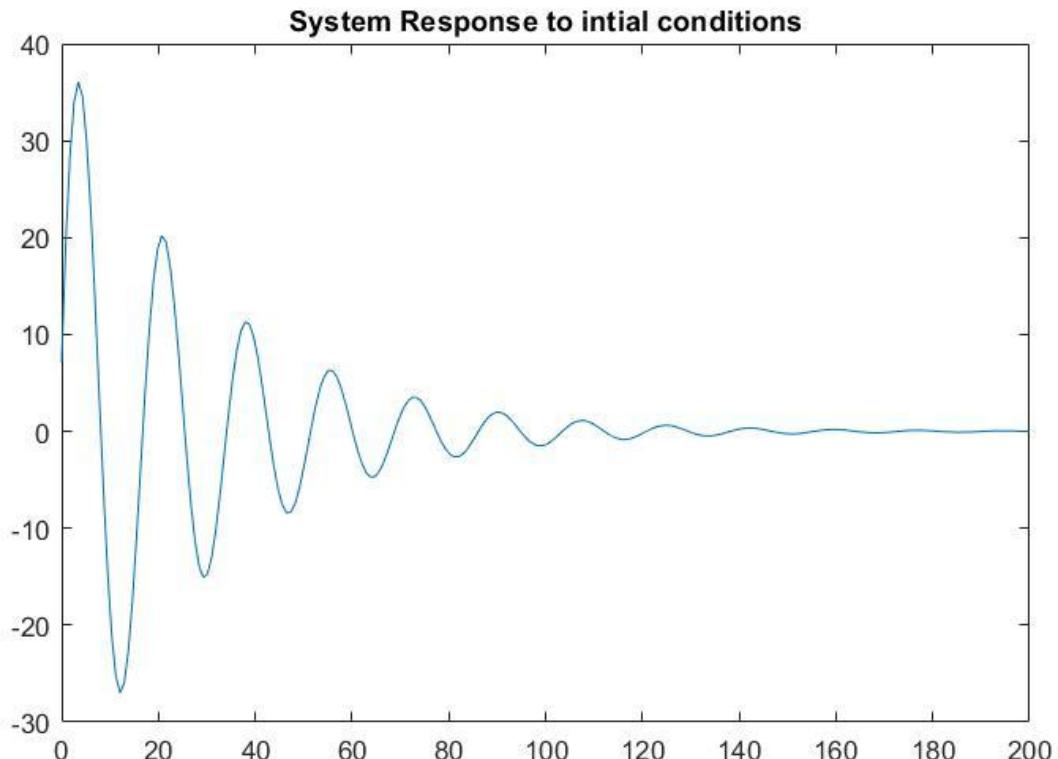


Figure 4.6.. The Linear Model Response to +2 deg Disturbance in Pitch Angle (θ)

4. Model Verification (Quadrotor)

I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Elevating

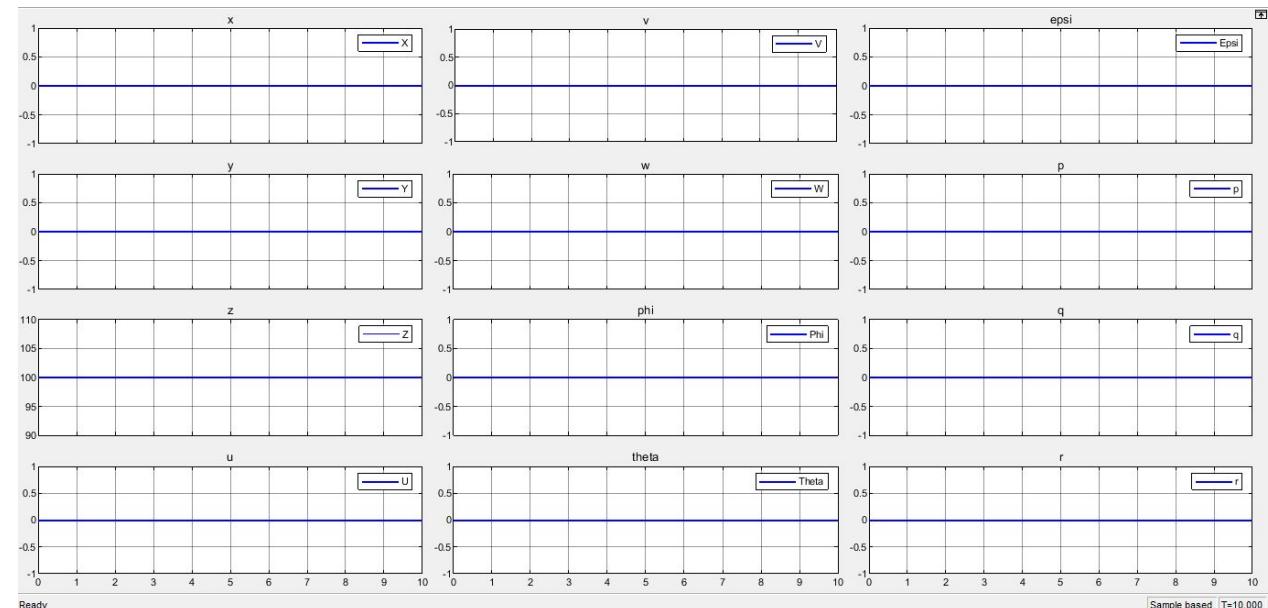
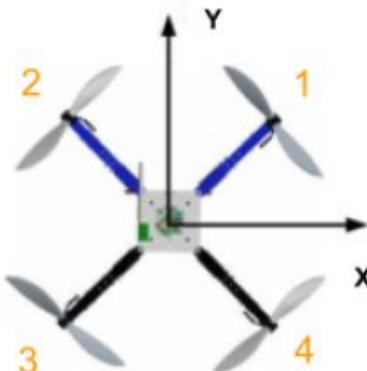
The thrust force needs equal the quadrotor weight

$$T = k(u_1 + u_2 + u_3 + u_4) = mg$$

$$4u = 24 * 9.81$$

$$u = 58.86 \text{ rpm}^2$$

This is the hover speed



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Elevating

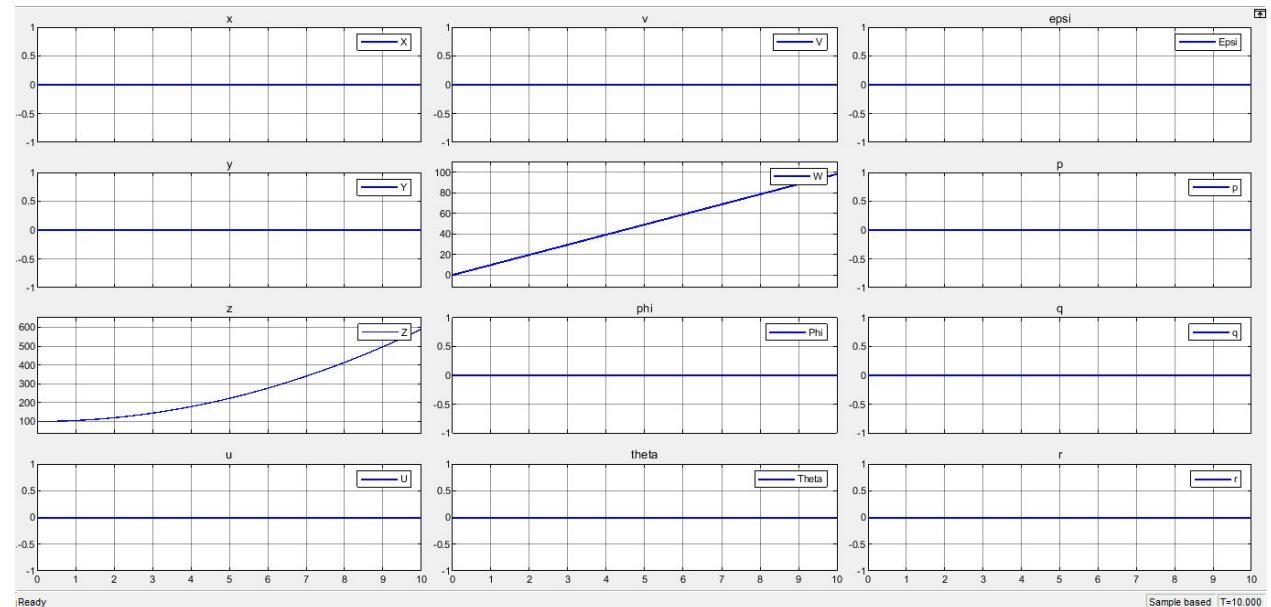
The thrust force needs equal the quadrotor weight

$$T = k(u_1 + u_2 + u_3 + u_4) = mg$$

$$4u = 24 * 9.81$$

$$u = 58.86 \text{ rpm}^2$$

Any speed higher than the hover speed,
elevates the quadrotor



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Pitching While Hovering

For the speeds needed to achieve pitching,

$$T = k(2u_1 + 2u_2) = mg$$

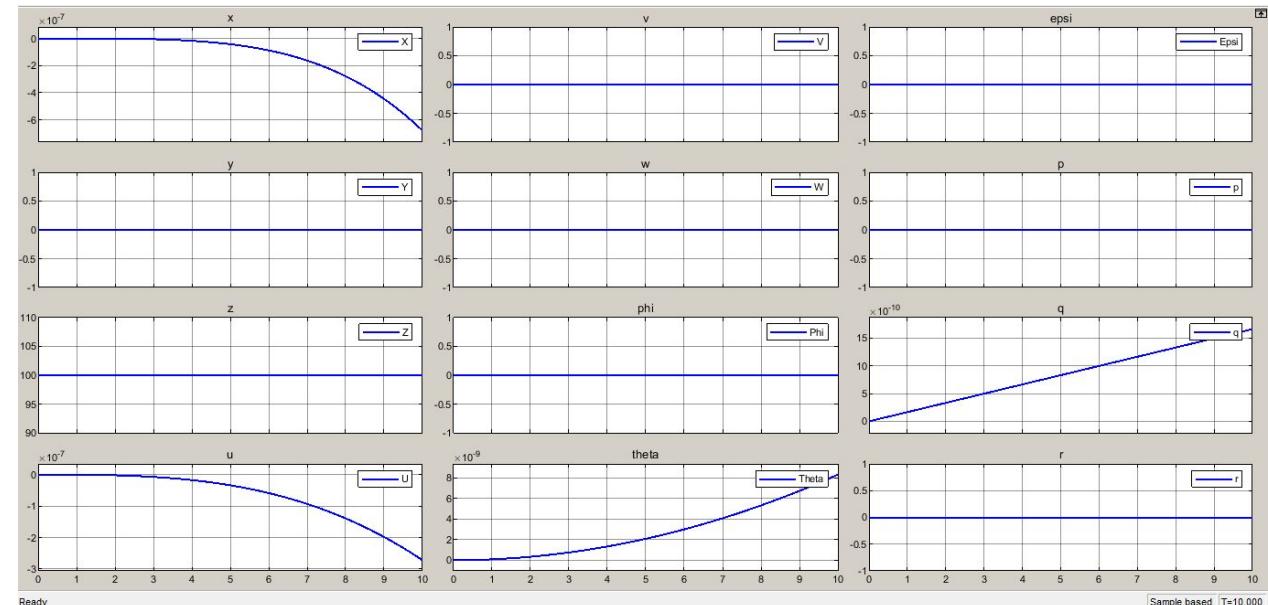
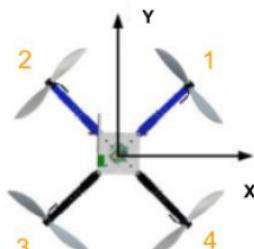
$$\text{Assume } u_1 = 60 \text{ rpm}^2$$

$$u_2 = 57.72 \text{ rpm}^2$$

Motor 1 and 2 = 60 rpm²

Motor 3 and 4 = 57.72 rpm²

Pitches the quadrotor up



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Pitching While Hovering

For the speeds needed to achieve pitching,

$$T = k(2u_1 + 2u_2) = mg$$

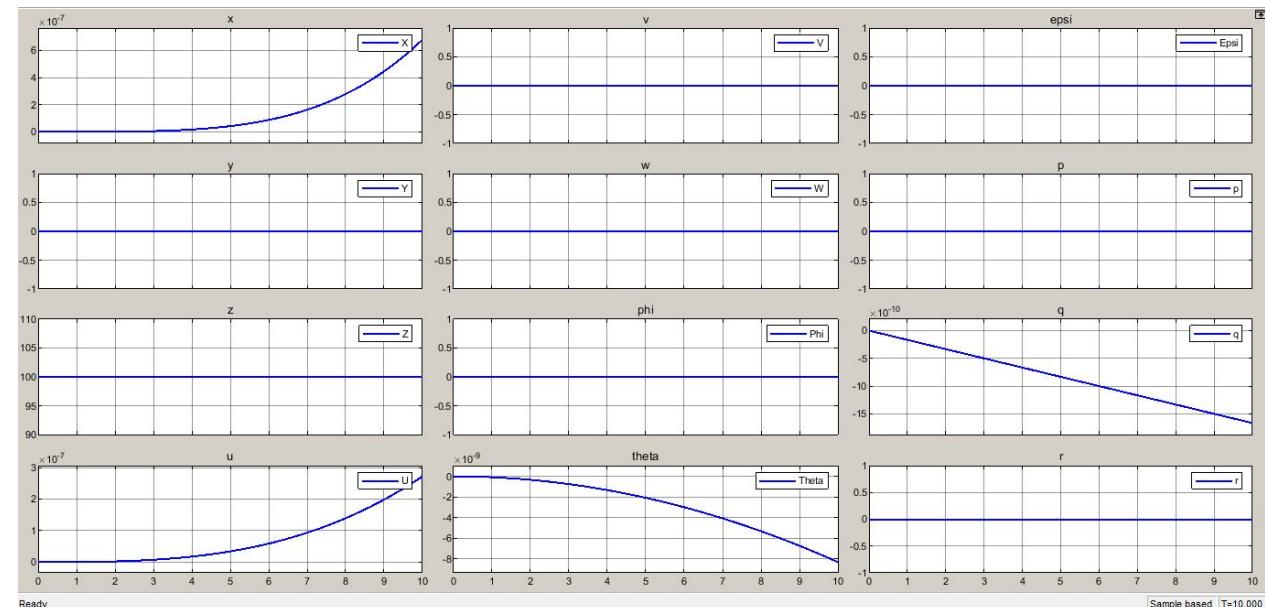
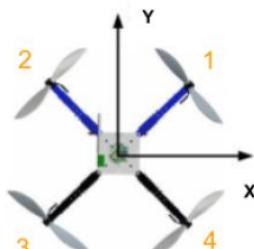
Assume $u_1 = 60 \text{ rpm}^2$

$$u_2 = 57.72 \text{ rpm}^2$$

Motor 1 and 2 = 57.72 rpm²

Motor 3 and 4 = 60 rpm²

Pitches the quadrotor down



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Rolling While Hovering

For the speeds needed to achieve pitching,

$$T = k(2u_1 + 2u_2) = mg$$

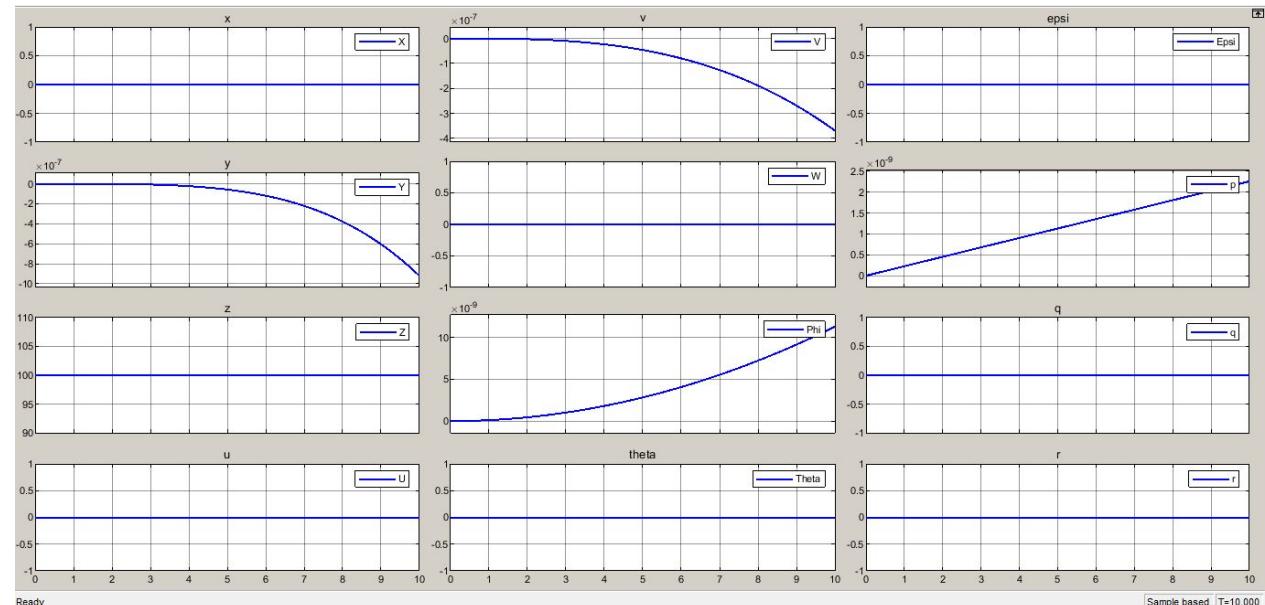
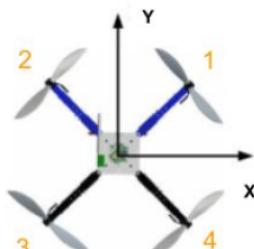
Assume $u_1 = 60 \text{ rpm}^2$

$$u_2 = 57.72 \text{ rpm}^2$$

Motor 1 and 4 = 60 rpm²

Motor 2 and 3 = 57.72 rpm²

Rolls the quadrotor left



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Rolling While Hovering

For the speeds needed to achieve pitching,

$$T = k(2u_1 + 2u_2) = mg$$

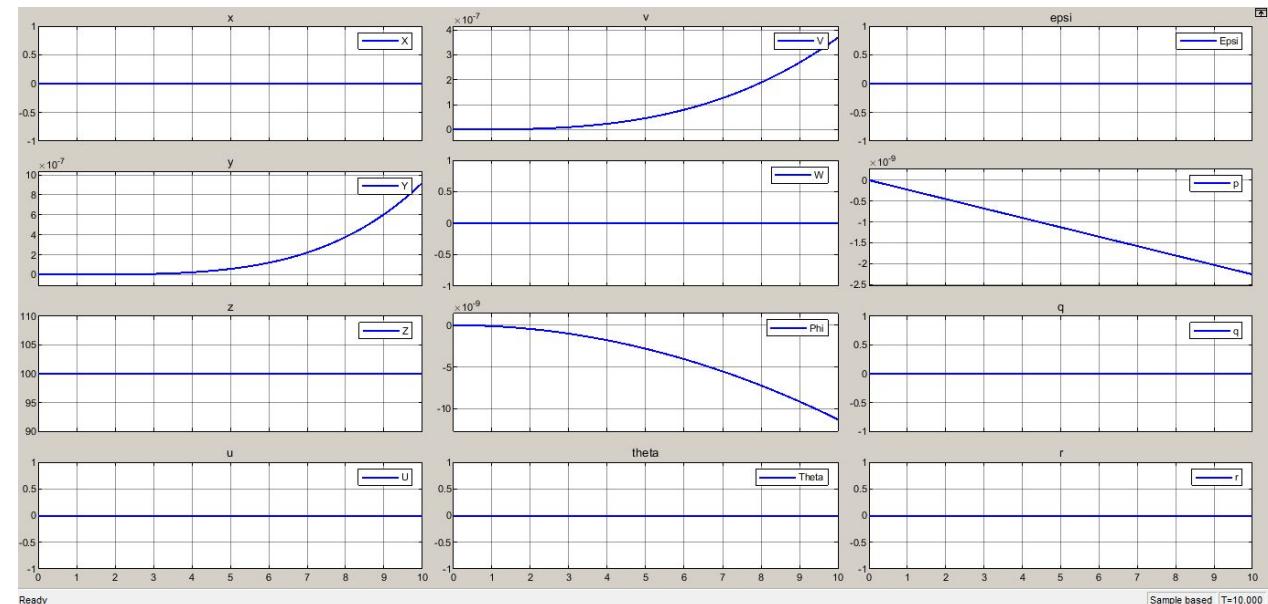
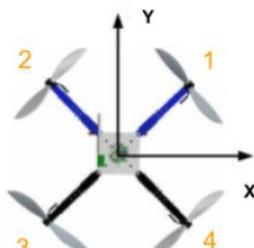
Assume $u_1 = 60 \text{ rpm}^2$

$$u_2 = 57.72 \text{ rpm}^2$$

Motor 1 and 4 = 57.72 rpm²

Motor 2 and 3 = 60 rpm²

Rolls the quadrotor right



I. Quadrotor Elevating, Pitching, Rolling, and Yawing



Yawning While Hovering

For the speeds needed to achieve pitching,

$$T = k(2u_1 + 2u_2) = mg$$

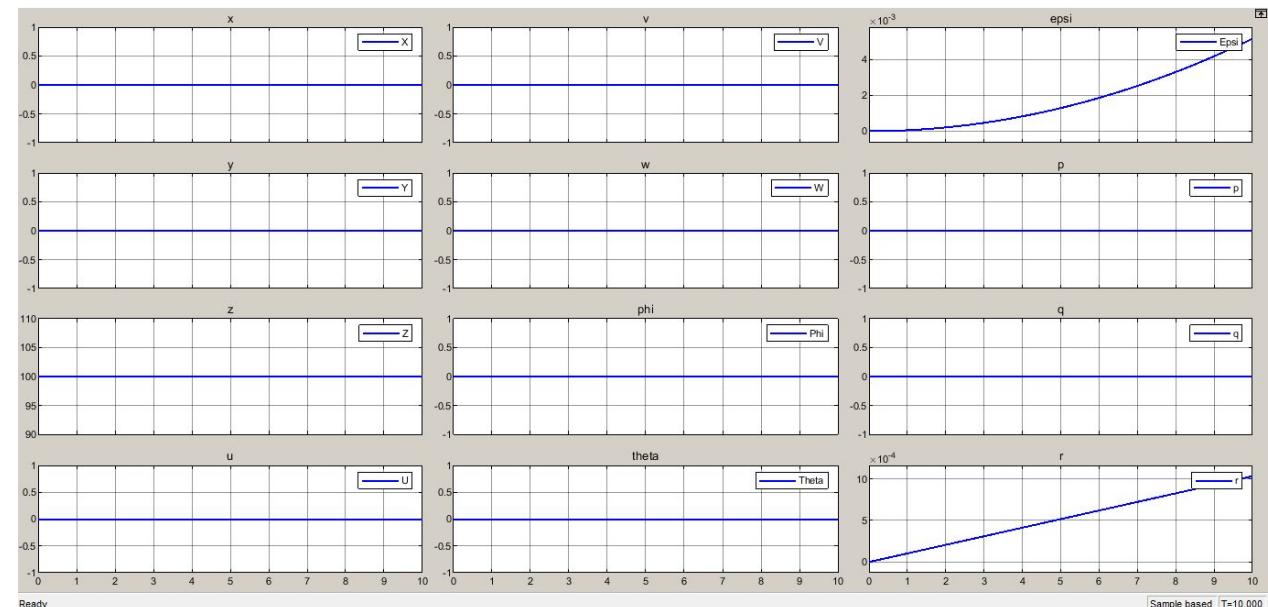
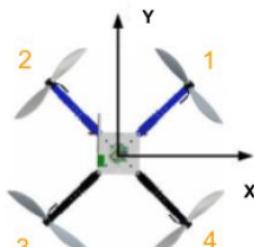
Assume $u_1 = 60 \text{ rpm}^2$

$$u_2 = 57.72 \text{ rpm}^2$$

Motor 1 and 3 = 60 rpm²

Motor 2 and 4 = 57.72 rpm²

Yaws clockwise

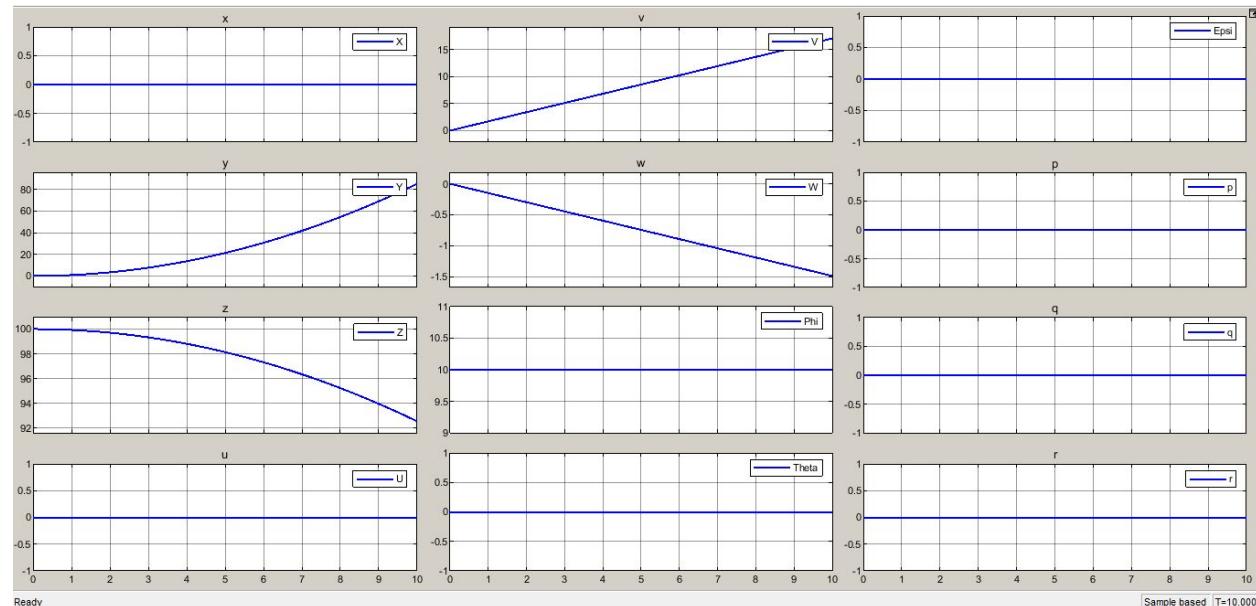


II. Quadrotor Initial Disturbances Response



Initial Disturbances in Phi

Introducing a 10 deg disturbance in the phi angle while hovering at 100 m.

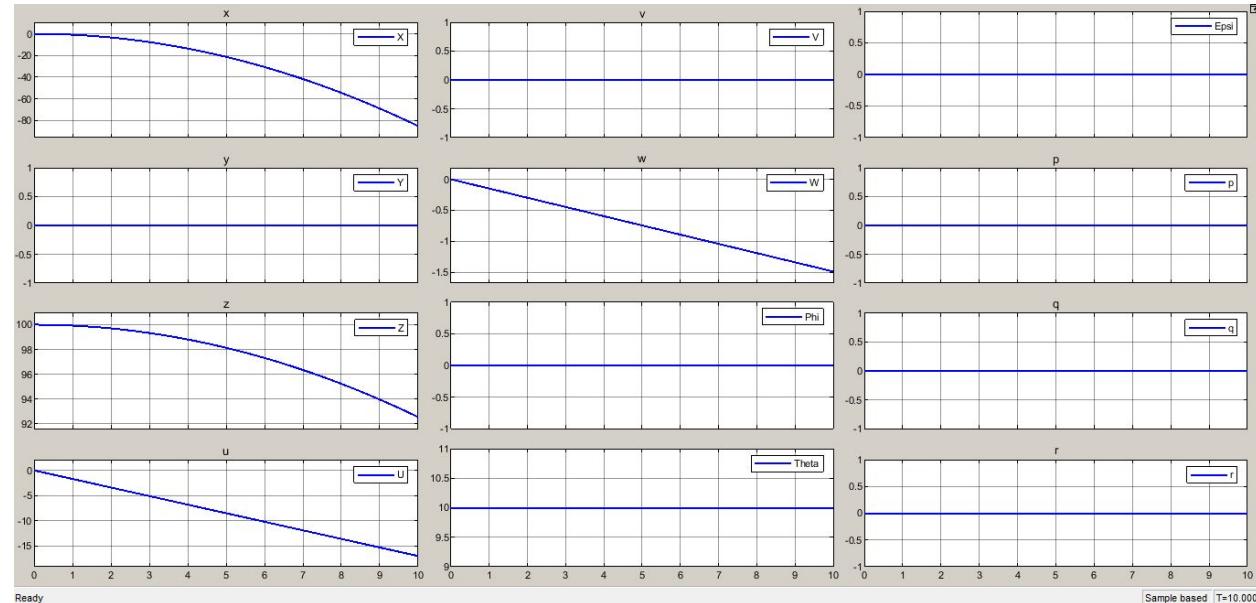


II. Quadrotor Initial Disturbances Response



Initial Disturbances in Theta

Introducing a 10 deg disturbance in the theta angle while hovering at 100 m.

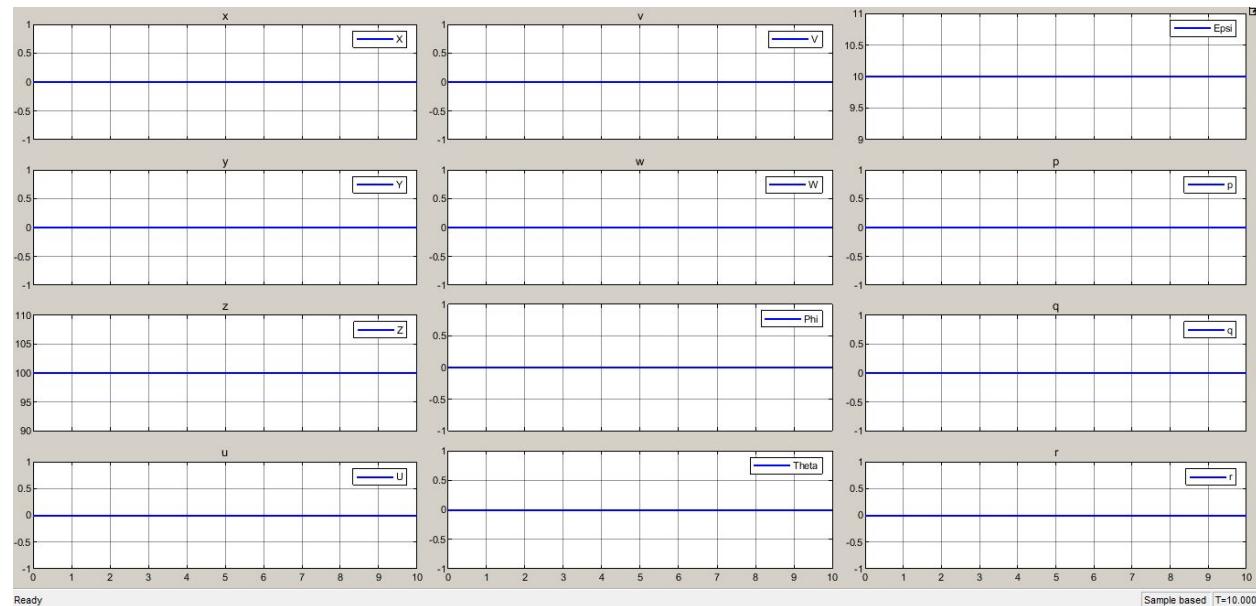


II. Quadrotor Initial Disturbances Response



Initial Disturbances in Epsi

Introducing a 10 deg disturbance in the ϵ_{psi} angle while hovering at 100 m.



5. Autopilot Controller Design and Implementation (Fixed Wing UAV)

I. Total Energy Control System



Introduction

- Developed by Boeing and NASA in the 1980s
- Utilizes energy management principles to track altitude and speed during longitudinal flight
- Successfully used in various optimization problems in engineering.

I. Total Energy Control System



TECS Controller

- Total energy =Kinetic energy +potential energy

$$E_T = \frac{1}{2}mV^2 + mgh$$

- Taking the derivative of the equation w.r.t time

$$\dot{E}_T = mV V` + m g h`$$

I. Total Energy Control System



TECS Controller

Specific Energy rate is

$$\dot{E} = \frac{\dot{E}_T}{mgV} = \frac{V}{g} + \frac{h}{V} = \frac{V}{g} + \sin \gamma$$

For small flight path angle

$$\dot{E} \approx \frac{V}{g} + \gamma$$

I. Total Energy Control System



From the equation of motion of an aircraft in straight level flight

$$T - D = mg\left(\frac{V}{g} + \sin \gamma\right) \approx mg\left(\frac{V}{g} + \gamma\right)$$

Since at Trim Thrust equals Drag, therefore change in thrust equals

$$\Delta T = mg\left(\frac{V}{g} + \gamma\right)$$

I. Total Energy Control System



From the equation of motion of an aircraft in straight level flight

$$T - D = mg\left(\frac{V}{g} + \sin \gamma\right) \approx mg\left(\frac{V}{g} + \gamma\right)$$

Since at Trim Thrust equals Drag, therefore change in thrust equals

$$\Delta T = mg\left(\frac{V}{g} + \gamma\right)$$

I. Total Energy Control System



Elevator Control, being energy-efficient and accurate, is employed to convert kinetic energy into potential energy and vice versa.

This is described by the specific energy balance equation:

$$B^* = \gamma - \frac{V}{g}$$

I. Total Energy Control System

Simulink Implementation

The system is composed of two feedback loops: one for regulating total energy by adjusting the throttle setpoint, and another for maintaining energy balance by setting the pitch angle for attitude control

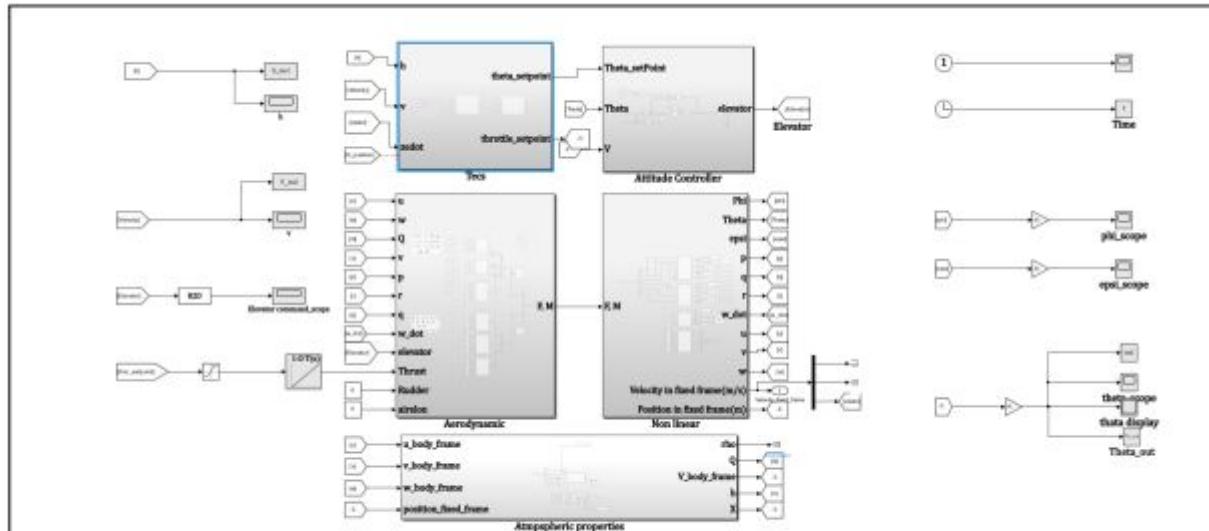


Figure 5.1.2.5 Integration of TECS controller with Fixed wing model

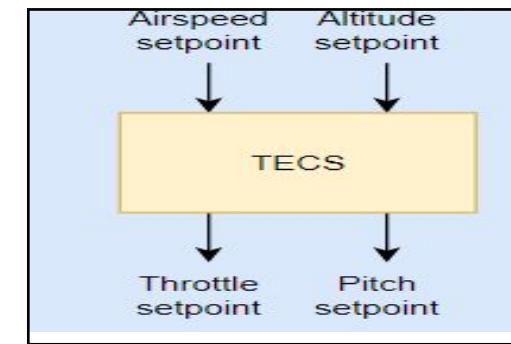


Figure 5.1.2.1 Total Energy control system

I. Total Energy Control System

Simulink Implementation

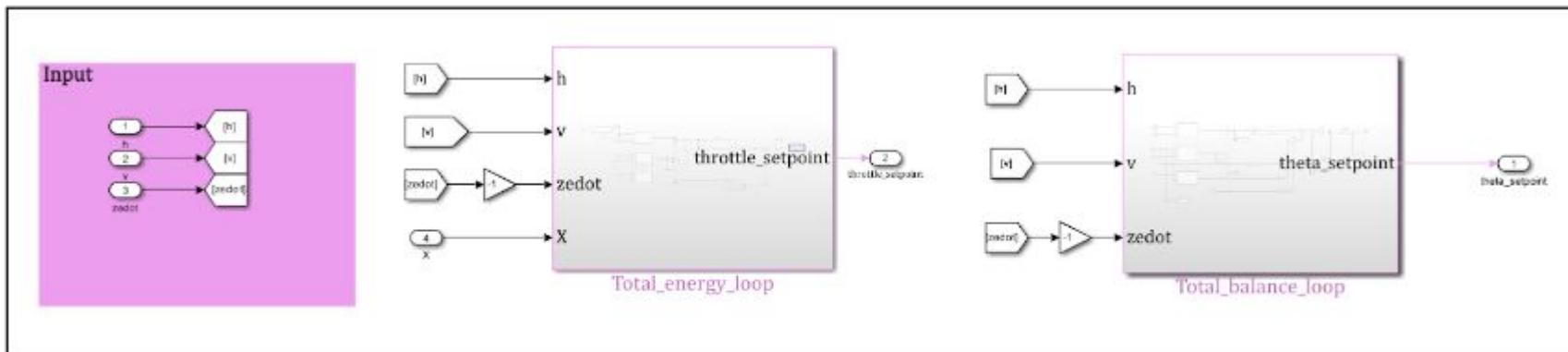


Figure 5.1.2.2:TECS simulink implementation

I. Total Energy Control System



Specific Total Energy Loop

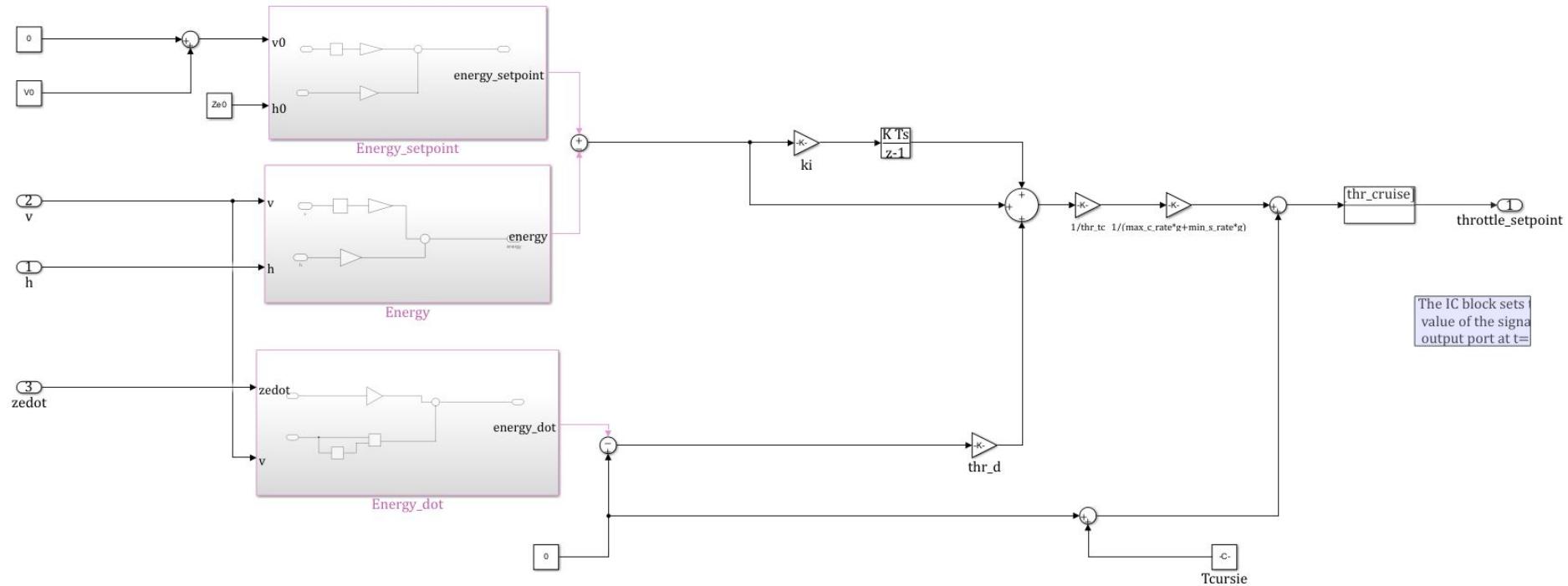


Figure 5.1.2.3: specific total energy loop

I. Total Energy Control System



Specific Energy Balance Loop

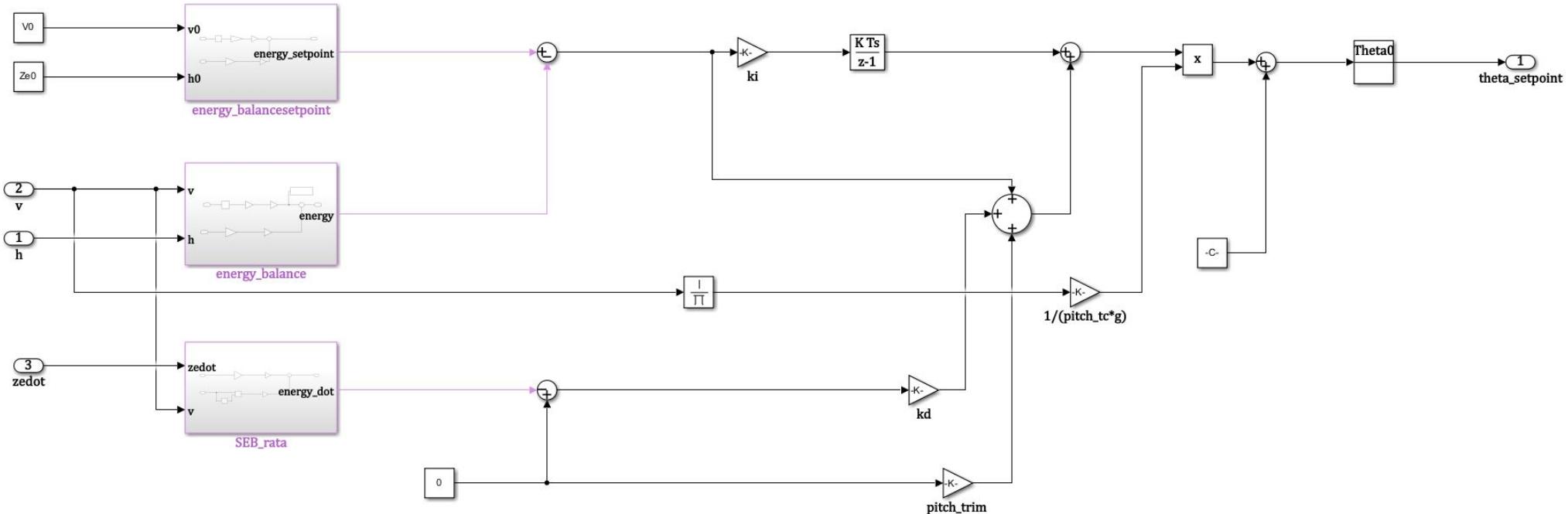


Figure 5.1.2.4: Specific Energy Balance Loop Simulink implementation

I. Total Energy Control System



Gain Tuning

The optimization techniques are :

- 1- Genetic algorithm
- 2-particle swarm algorithm
- 3- Differential Evolution(DE)
- 4-Simulated Annealing (SA)

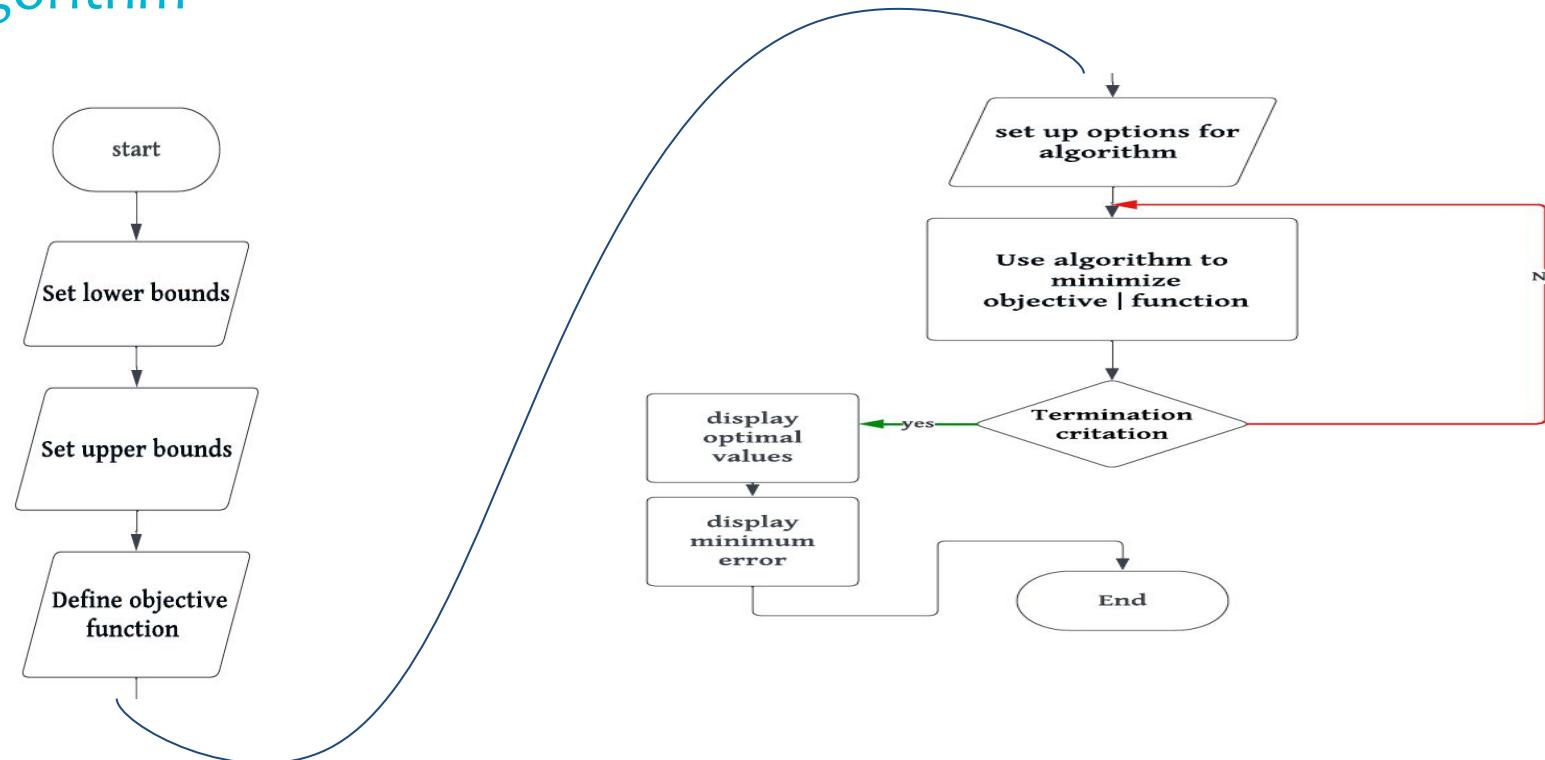
Optimizing on the following gains:

- 1- Pitch loop damping gain (Pitch_d)
- 2- Integrator gain (int_g) (the integrator gain is the same for both loops)
- 3- Pitch loop time constant (Pitch_tc)
- 4- Throttle loop damping gain (Thr_d)
- 5- Throttle loop time constant (Thr_tc)

I. Total Energy Control System



Genetic Algorithm



I. Total Energy Control System



Genetic Algorithm

The output gains of the genetic algorithm were as follows:

Gains	Values
Integrator gains	0.05
Throttle Damping	0
Throttle Time constant	1.5
Pitching Damping	0.1
Pitch Time constant	2.5

Table 5.1.2.1 Value of Gains tuning GA algorithm

I. Total Energy Control System



Genetic Algorithm Response

Pitch Angle Response

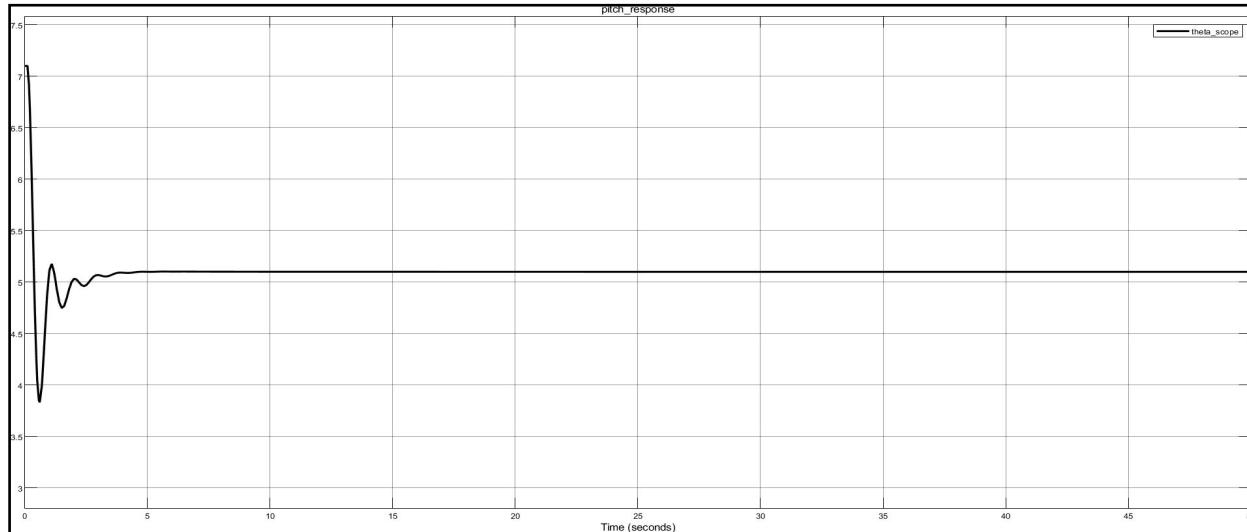


Figure 5.1.2.7.pitch angle response after using GA algorithm to tune gains

I. Total Energy Control System



Genetic Algorithm Response

Altitude Response

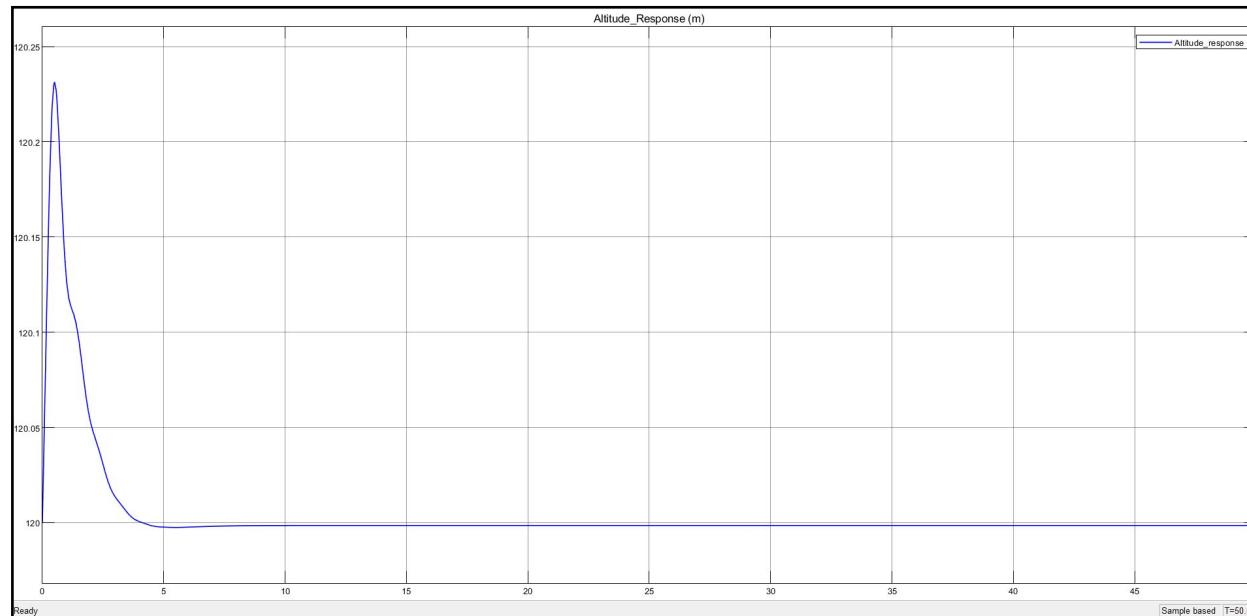


Figure 5.1.2.8. Altitude response after using GA algorithm to tune gains

I. Total Energy Control System



Genetic Algorithm Response

Velocity Response

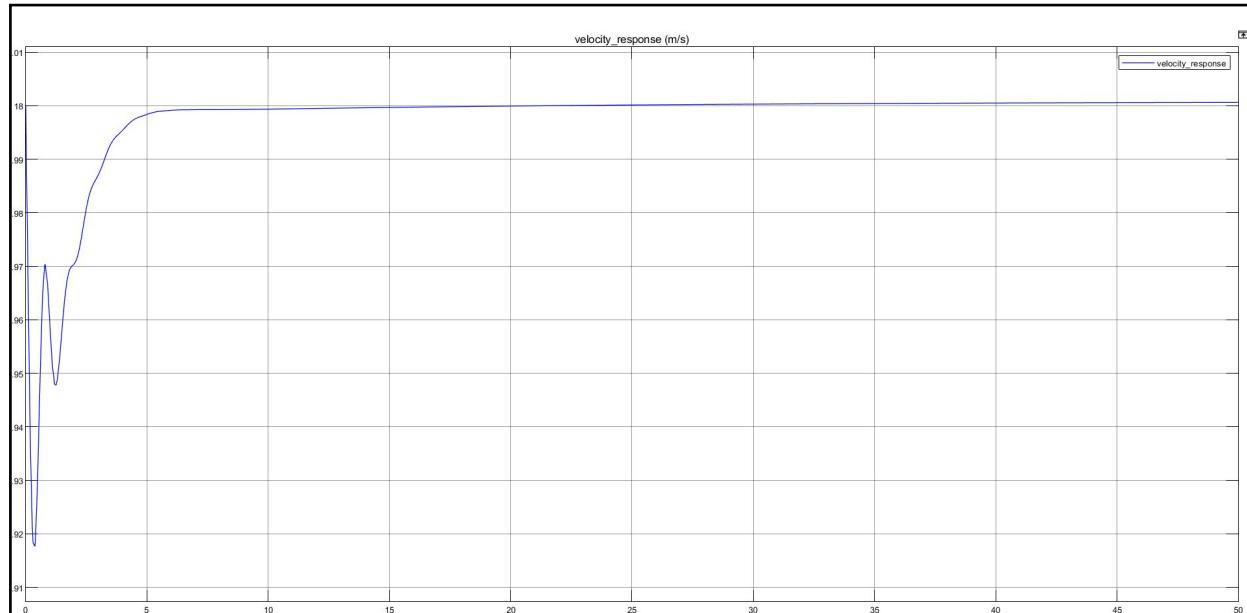
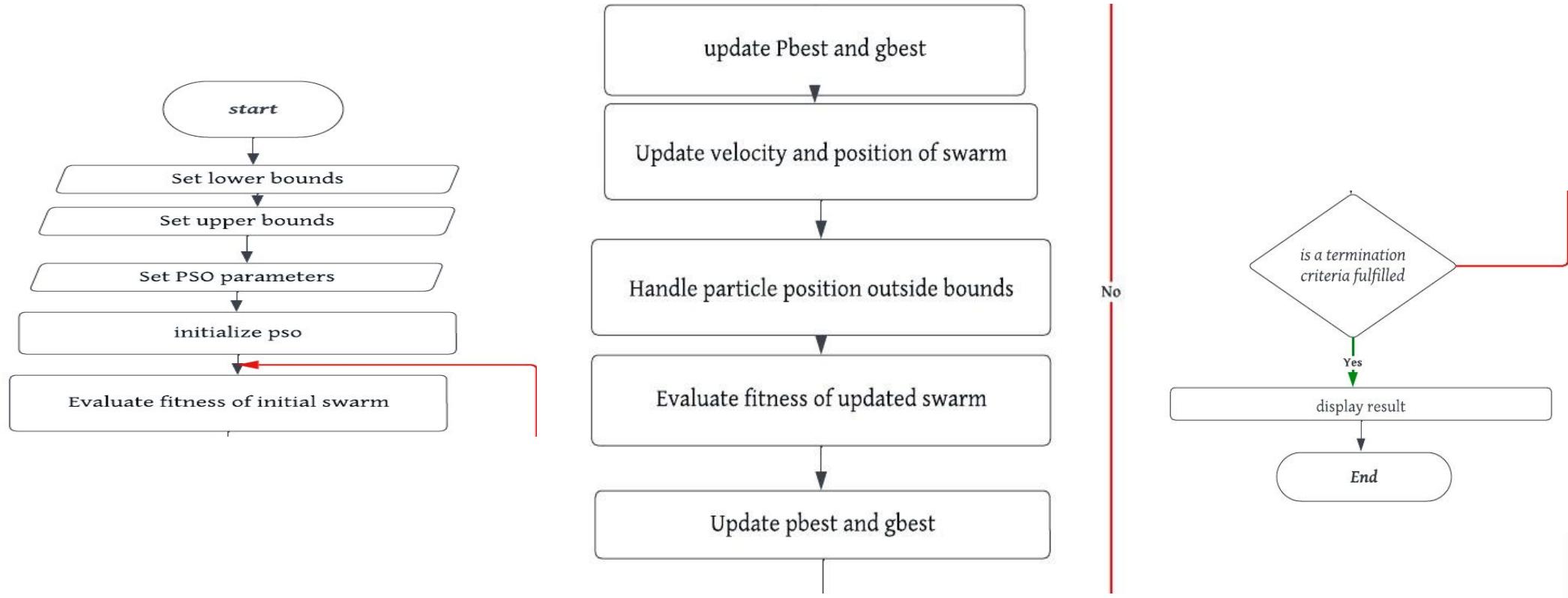


Figure 5.1.2.9. velocity response after using GA algorithm to tune gains

I. Total Energy Control System



Particle Swarm



I. Total Energy Control System



Particle Swarm Gains

The output gains of the PSO algorithm were as follows:

Gains	Values
Integrator gains	0.05
Throttle Damping	0.1
Throttle Time constant	5
Pitching Damping	0.15
Pitch Time constant	2

Table 5.1.2.2 Value of Gains tuning PSO algorithm

I. Total Energy Control System



Particle Swarm Response:

Pitch Angle Response

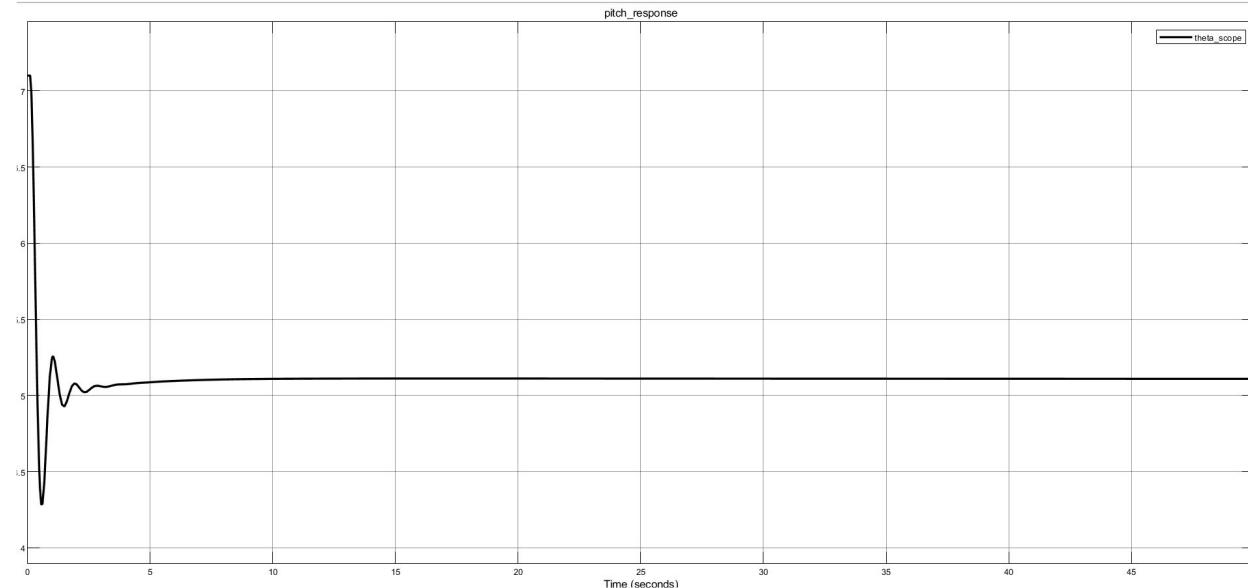


Figure 5.1.2.11. Pitch angle response after using PSO algorithm to tune gains

I. Total Energy Control System



Particle Swarm Response:

Altitude Response

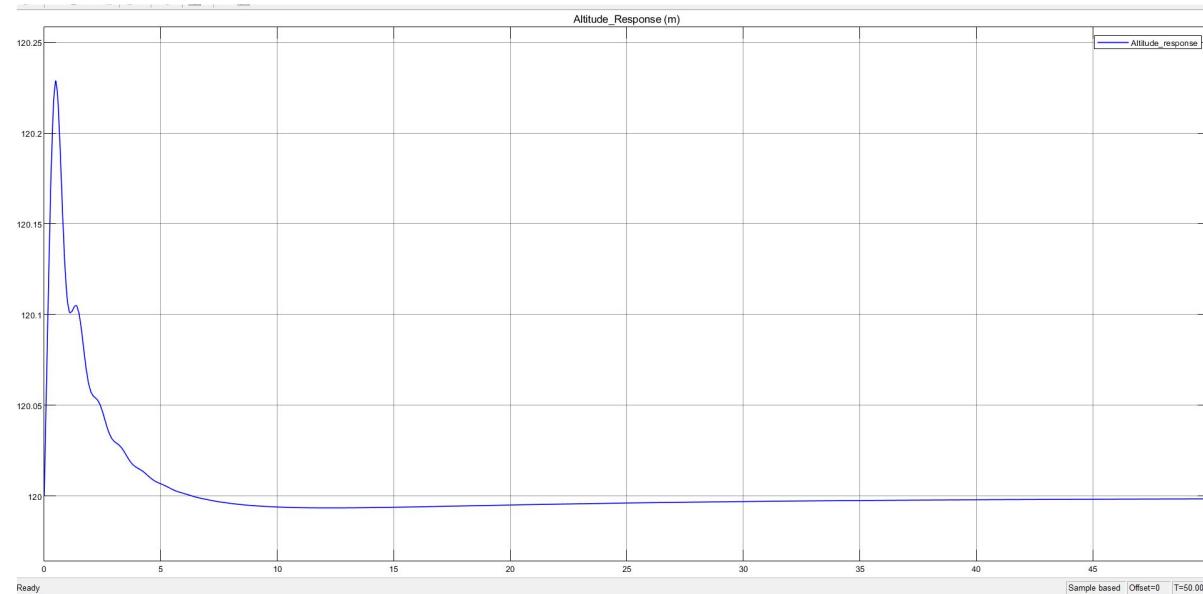


Figure 5.I.2.13. Altitude response after using PSO algorithm to tune gains

I. Total Energy Control System



Particle Swarm Response:

Velocity Response

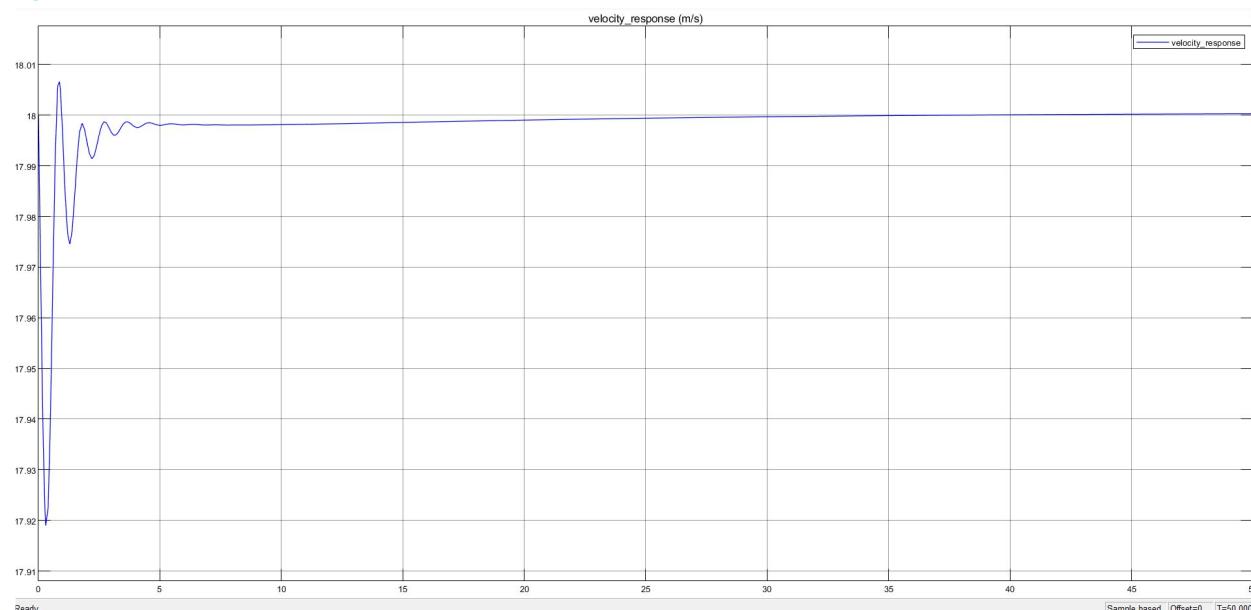
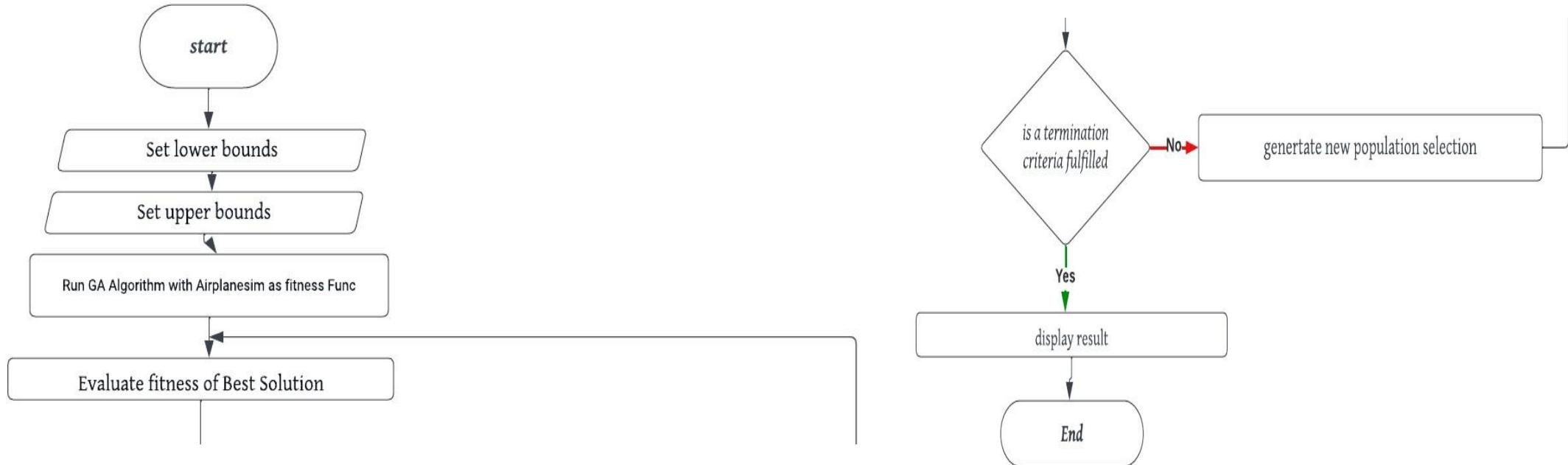


Figure 5.1.2.12. velocity response after using the PSO algorithm to tune gains

I. Total Energy Control System



Simulated Annealing



I. Total Energy Control System



Simulated Annealing Gains

5.1.2.3.3.1. Results of the SA algorithm

The output gains of the SA algorithm were as follows:

Gains	Values
Integrator gains	0
Throttle Damping	.15
Throttle Time constant	3.5
Pitching Damping	.05
Pitch Time constant	3

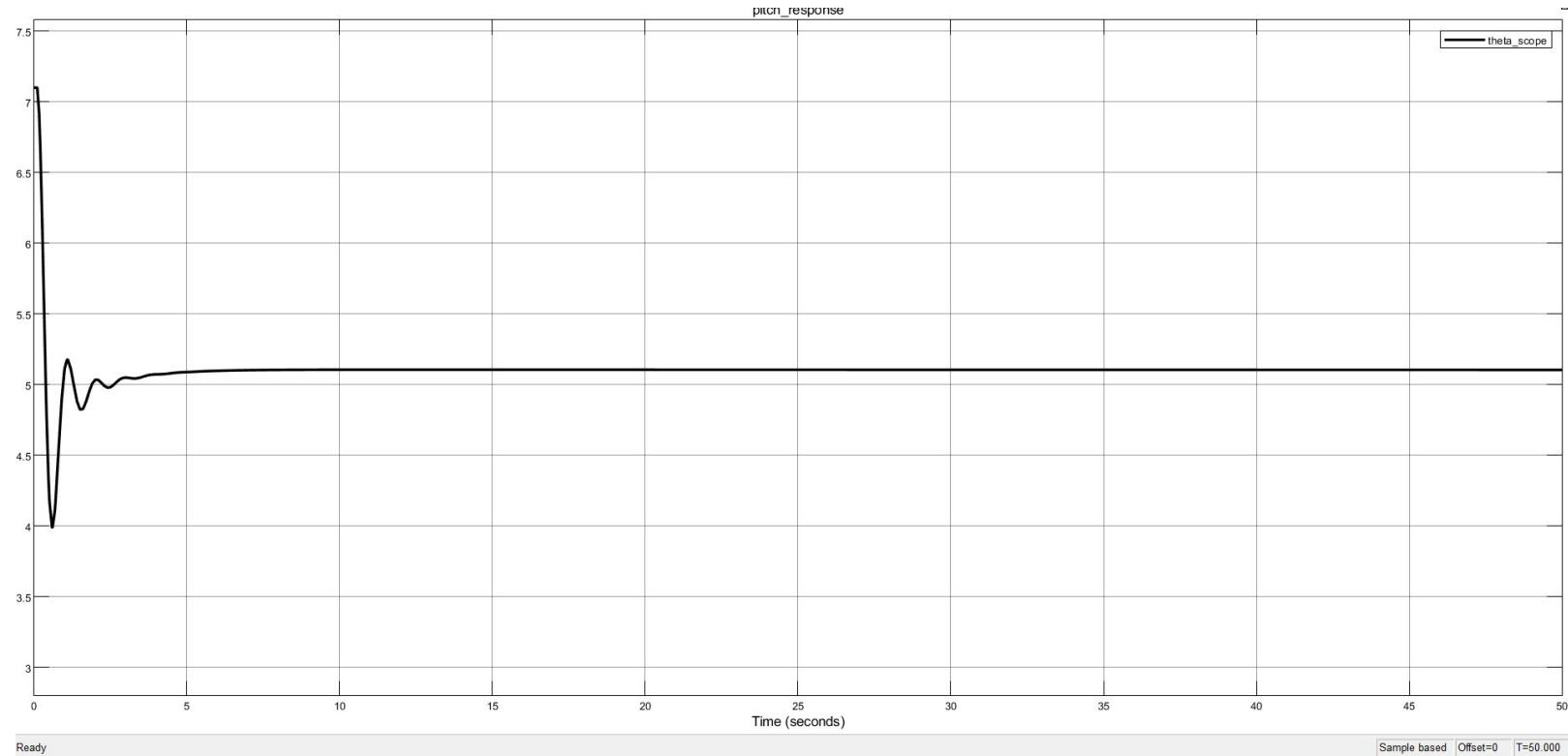
Table 5.1.2.3 Value of Gains tuning SA algorithm

I. Total Energy Control System



Simulated Annealing Response

pitch response



Ready

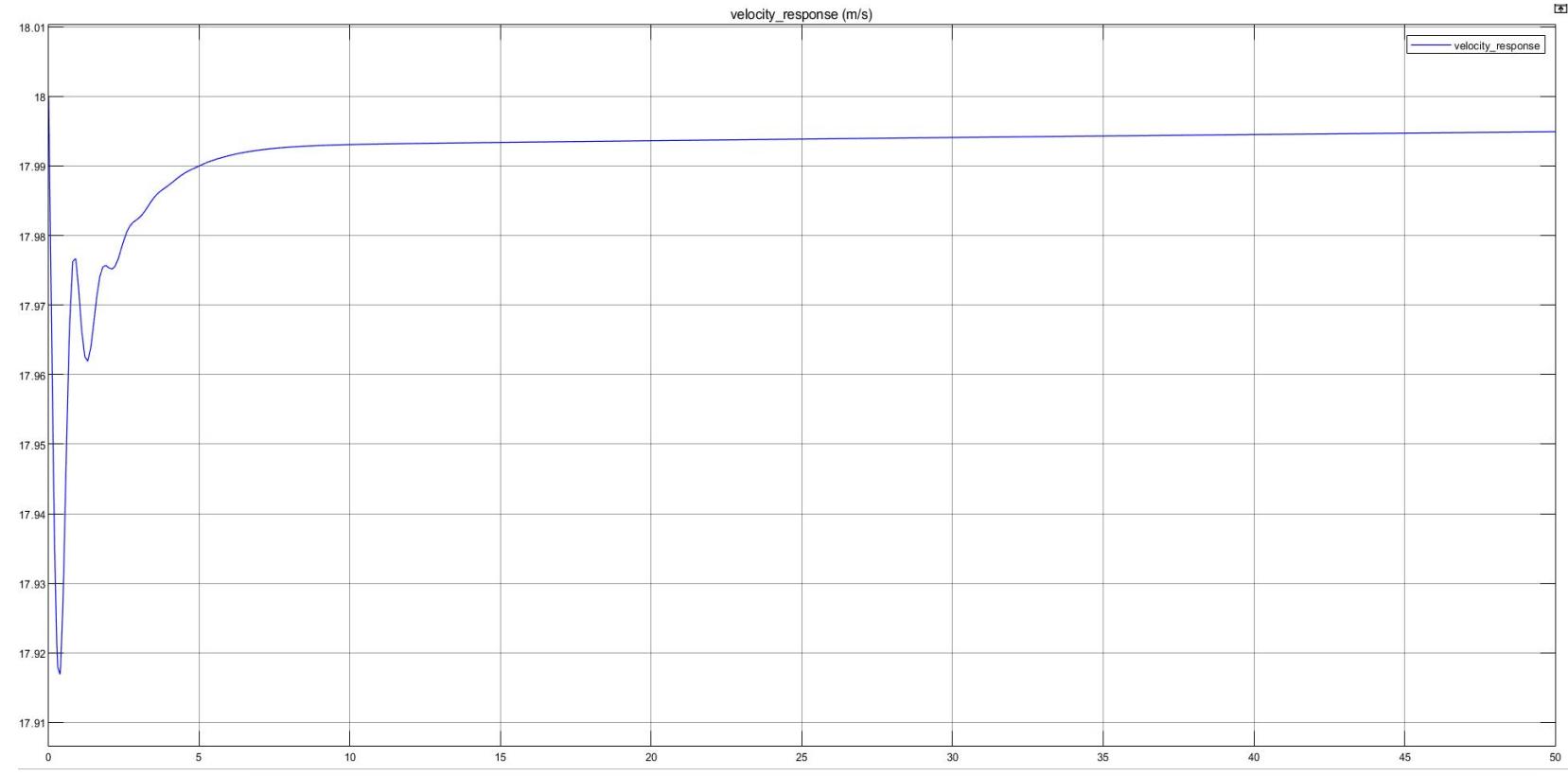
Sample based | Offset=0 | T=50 000

I. Total Energy Control System



Simulated Annealing Response

velocity response

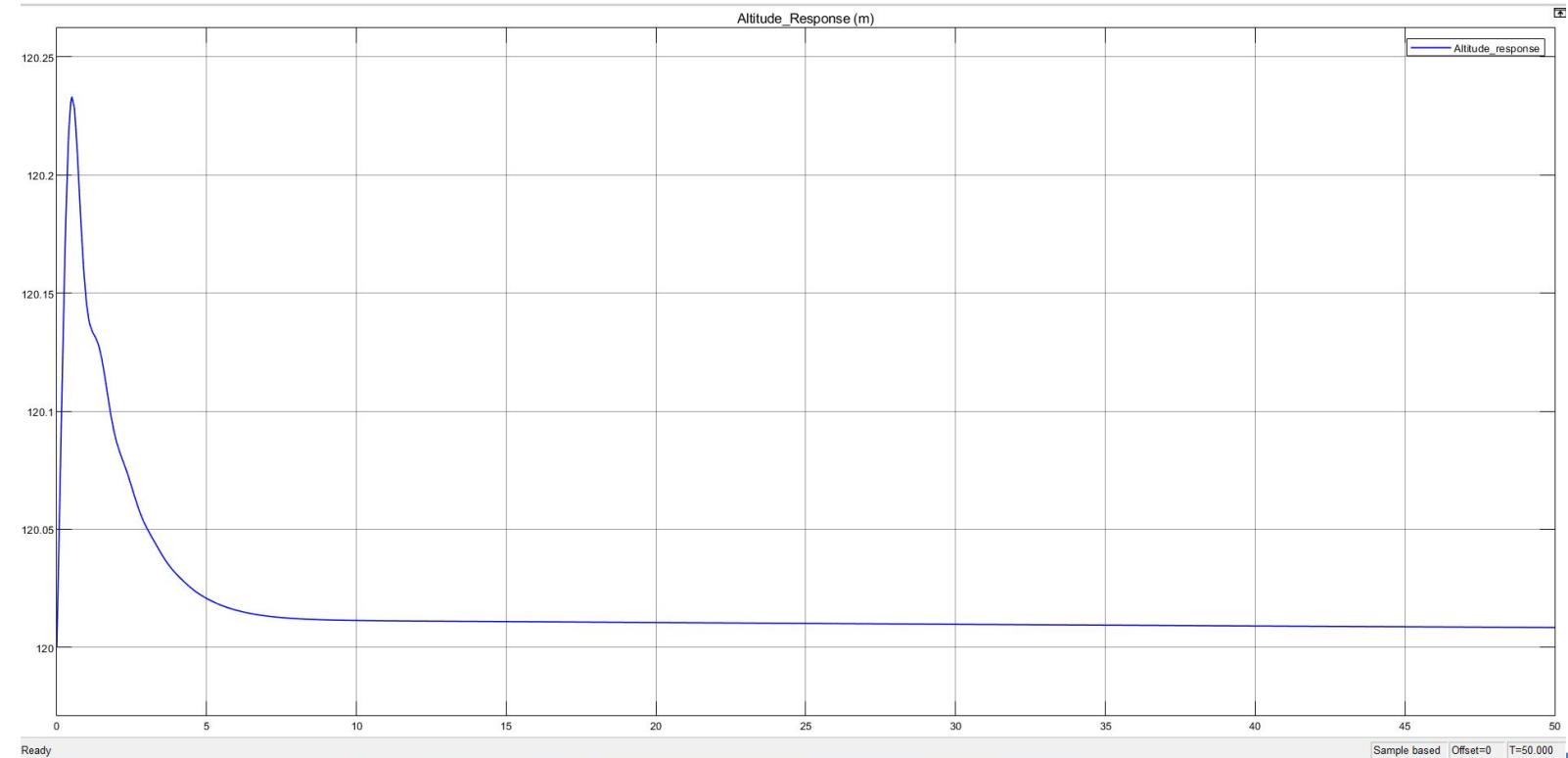


I. Total Energy Control System



Simulated Annealing Response

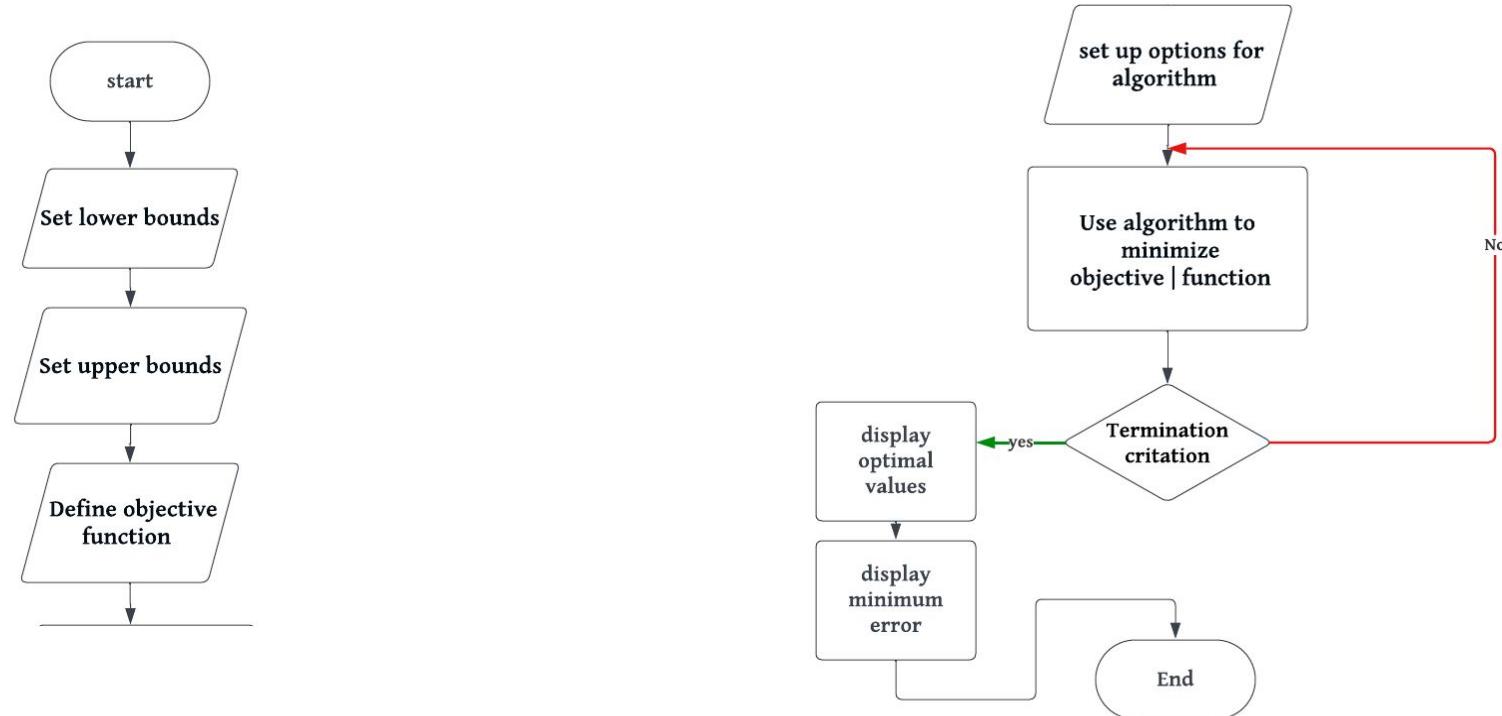
Altitude Response



I. Total Energy Control System



DE Algorithm



I. Total Energy Control System



DE Gains

The output gains of the DE algorithm were as follows:

Gains	Values
Integrator gains	.05
Throttle Damping	0.1
Throttle Time constant	3.5
Pitching Damping	.35
Pitch Time constant	2

Table 5.1.2.4 Value of Gains tuning DE algorithm

I. Total Energy Control System



DE Response

Pitch Angle Response

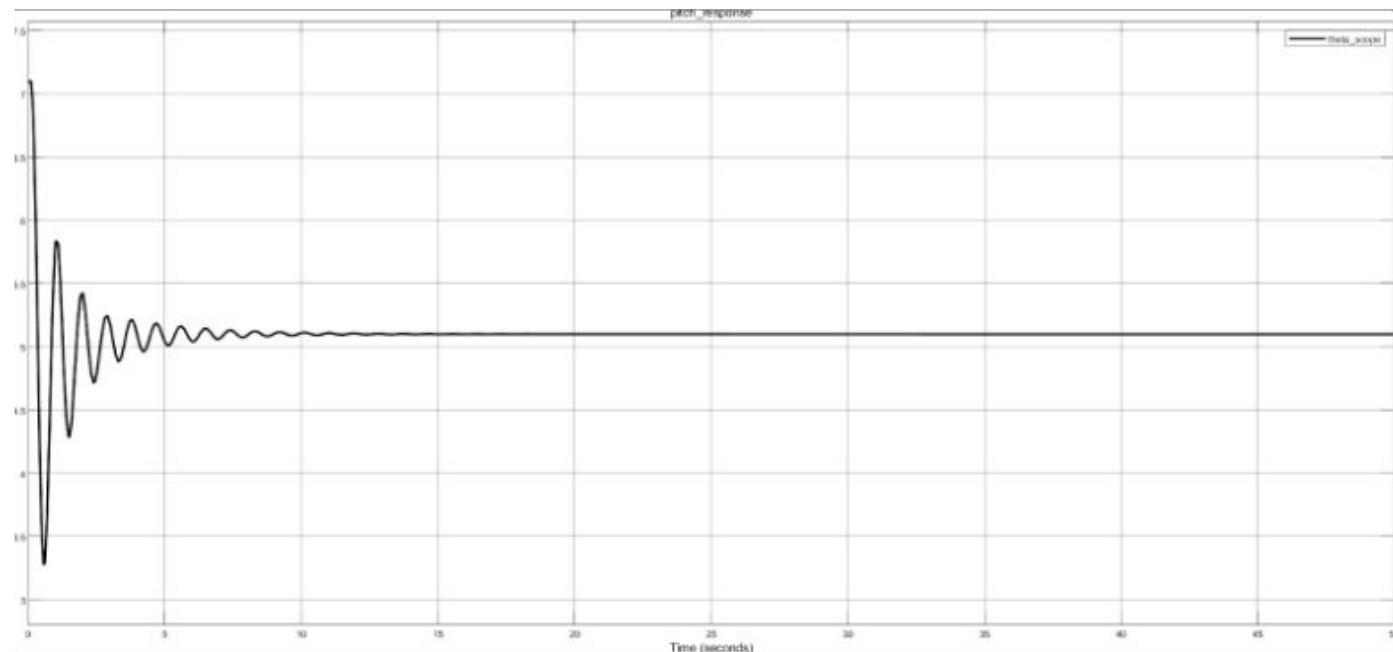


Figure 5.1.2.19. Pitch angle response after using DE algorithm to tune gains!

I. Total Energy Control System



DE Response

Velocity Response

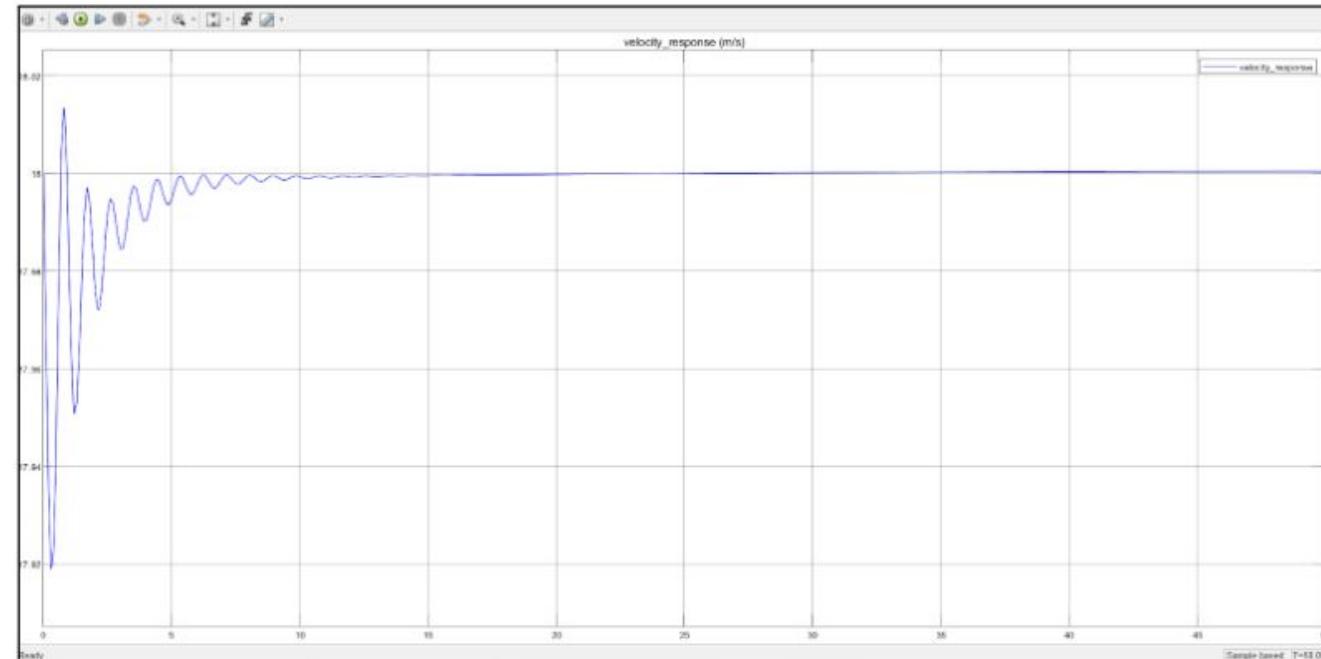


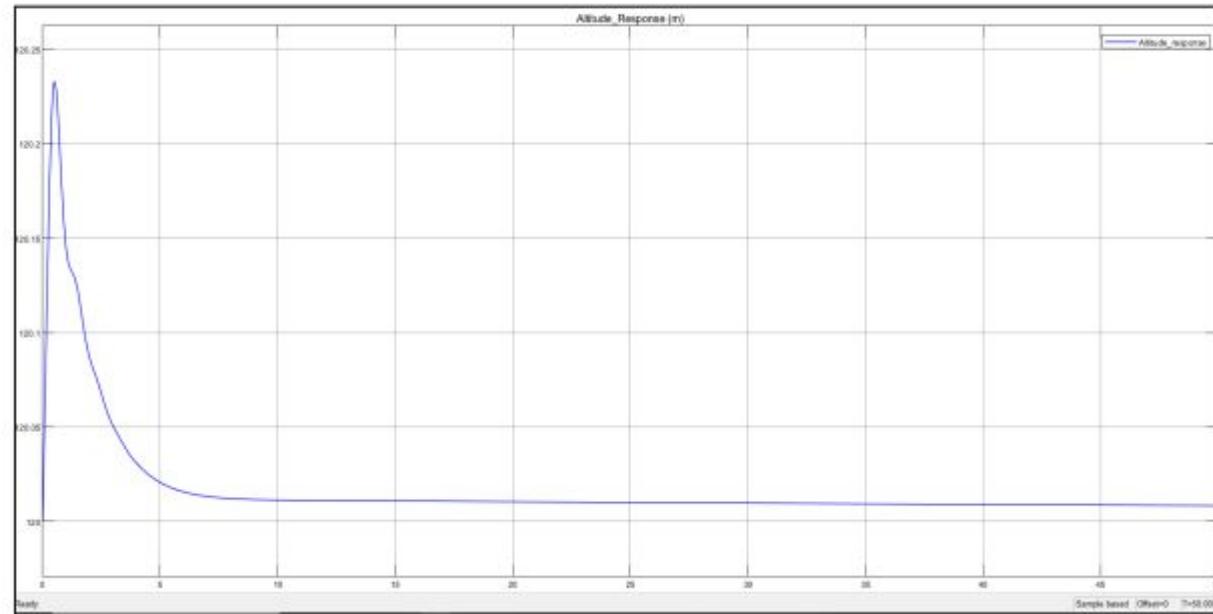
Figure 5.1.2.20. velocity response after using DE algorithm to tune gains

I. Total Energy Control System



DE Response

Altitude Response

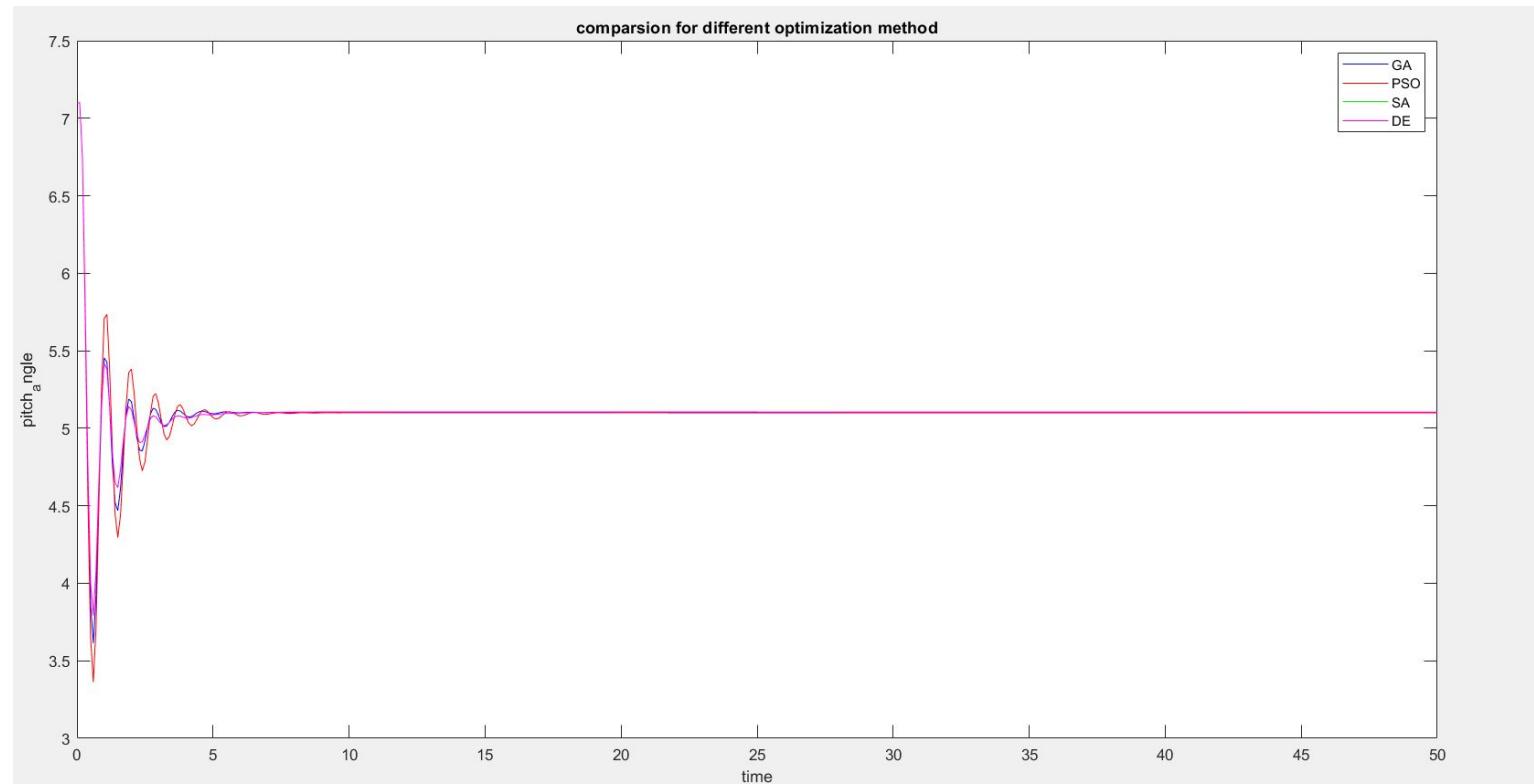


I. Total Energy Control System



Comparison between Four Methods

Pitch Response

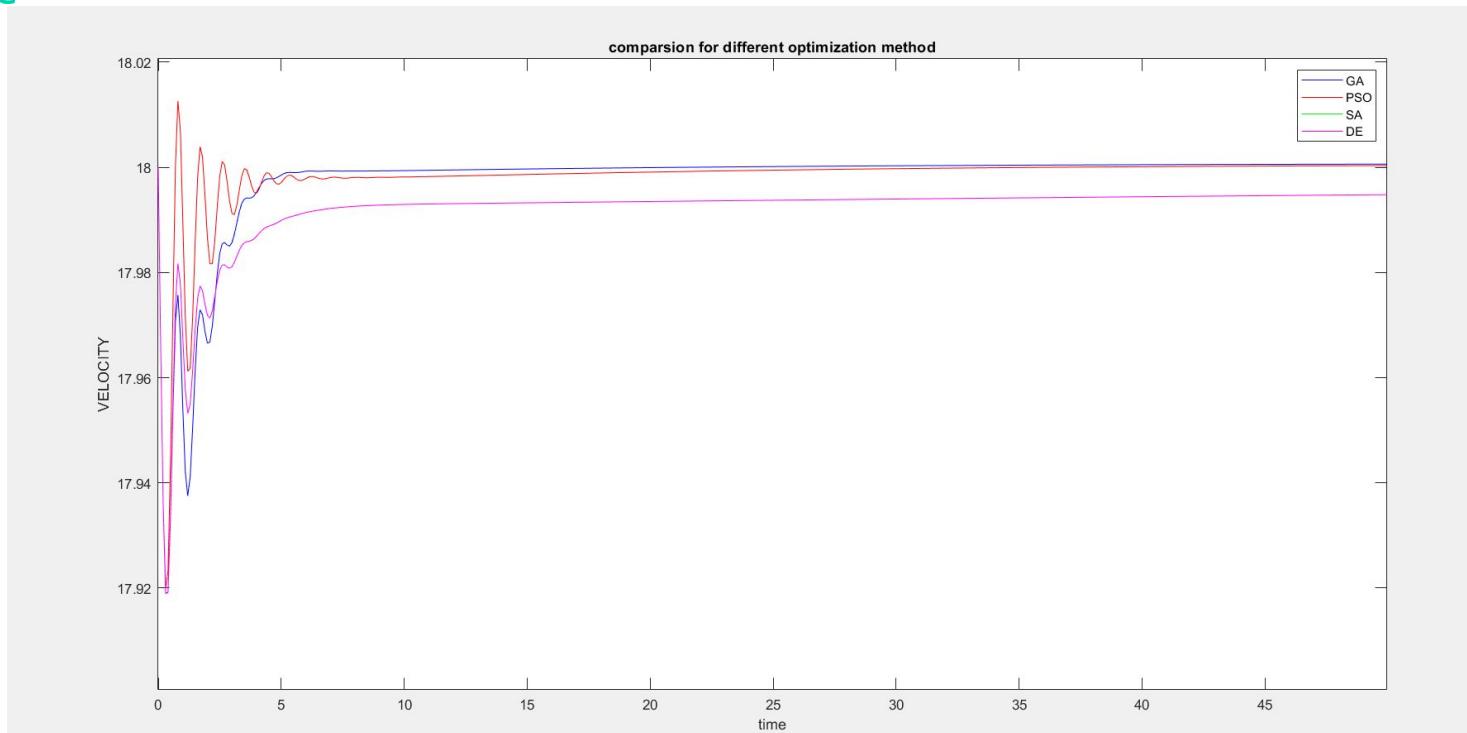


I. Total Energy Control System



Comparison between Four Methods

Velocity Response

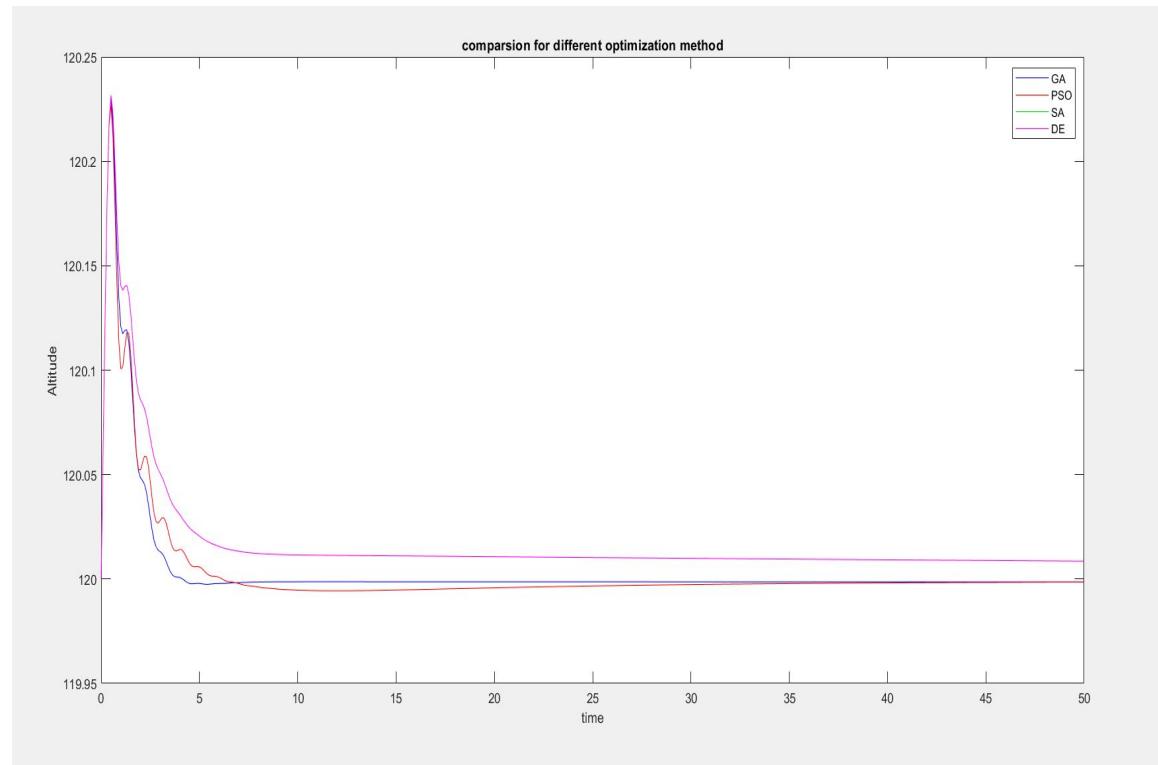


I. Total Energy Control System



Comparison between Four Methods

Altitude Response



II. Linear Quadratic Control (LQR)



Introduction

1. LQR is an optimal control method.
2. Quadratic cost function is used as a regulator.

$$J_{LQR} = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

3. Target is finding the control input u_{LQR} that minimizes the performance index.
4. Q is positive definite or positive semi-definite symmetry matrix, representing state error.
5. R is a positive-definite Hermitian or real symmetric matrix, representing control cost.
6. Optimal stabilizing gain is $K_{LQR} = R^{-1} B^T P$
7. Matrix P is evaluated by solving the Algebraic Riccati Equation

$$A P + A^T P - P B R^{-1} B^T P + Q = 0$$

II. Linear Quadratic Control (LQR)



Eliminating Steady State Error

- A feed-forward integral gain is added to the system.
- Adding p more states to the original states, where ξ represents the error integral of target state.

$$\dot{\xi}_p = C_p x - r_p$$

- Number of error states should be less than or equal number of actuators available (i,e $p \leq m$).
- Augmented system is in form

$$\begin{bmatrix} \dot{x} \\ \dot{\xi}_1 \\ \vdots \\ \dot{\xi}_p \end{bmatrix}_{(n+p) \times 1} = \begin{bmatrix} Ax + Bu \\ y_1 - r_1 \\ \vdots \\ y_p - r_p \end{bmatrix} = \begin{bmatrix} (Ax + Bu)_{n \times 1} \\ (Cx - r)_{p \times 1} \end{bmatrix}$$

- Optimum control input is defined as: $u = -K_p x - K_I \xi$

II. Linear Quadratic Control (LQR)



Implementation

- The LQR gains are calculated via MATLAB
- LQR tuning was done manually
- A range for elevator, aileron, and rudder deflection angles is from -15° to 30° .
- Thrust to have a positive value only (from 0 to ∞).

II. Linear Quadratic Control (LQR)



Longitudinal LQR

$Q =$

$$\begin{matrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.00001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{matrix} \quad R = \begin{matrix} 0.01 & 0 \\ 0 & 0.01 \end{matrix}$$

$K =$

$$\begin{matrix} 0.0701 & -2.0443 & -2.0829 & -171.0664 & -15.2656 & 4.6298 & -7.0703 \\ 31.7301 & -0.0877 & 0.0109 & 1.5850 & 0.2055 & 316.1939 & 0.1035 \end{matrix}$$

II. Linear Quadratic Control (LQR)



Longitudinal LQR

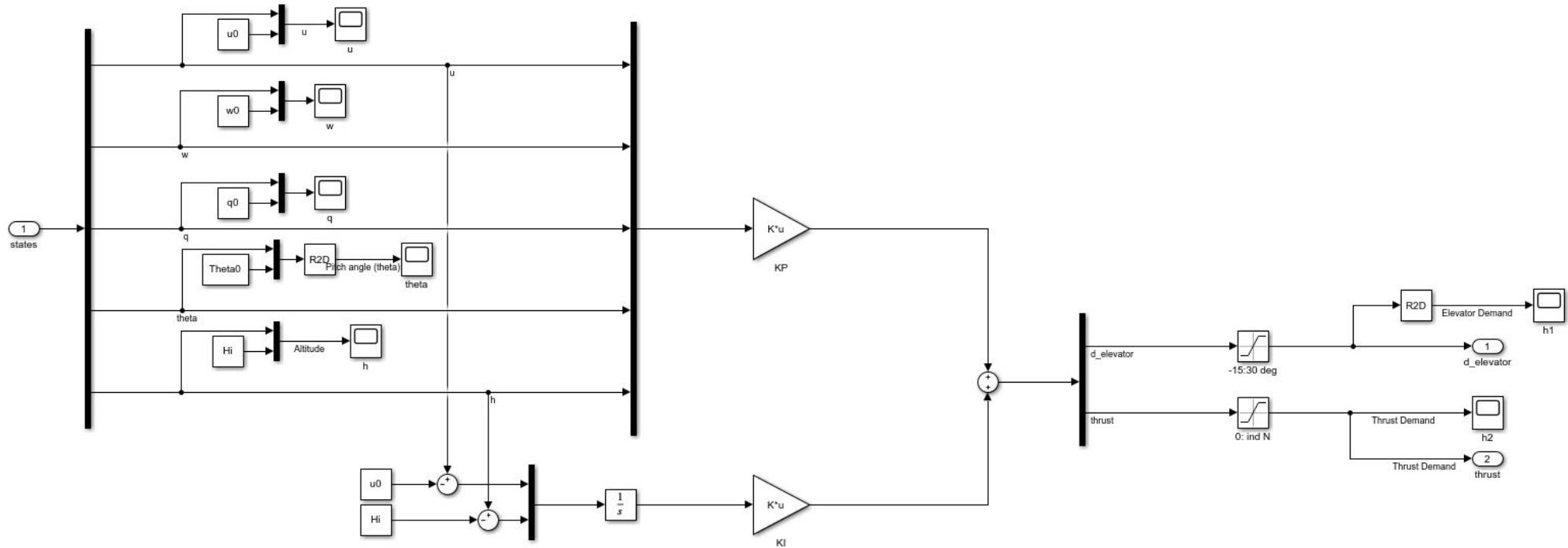
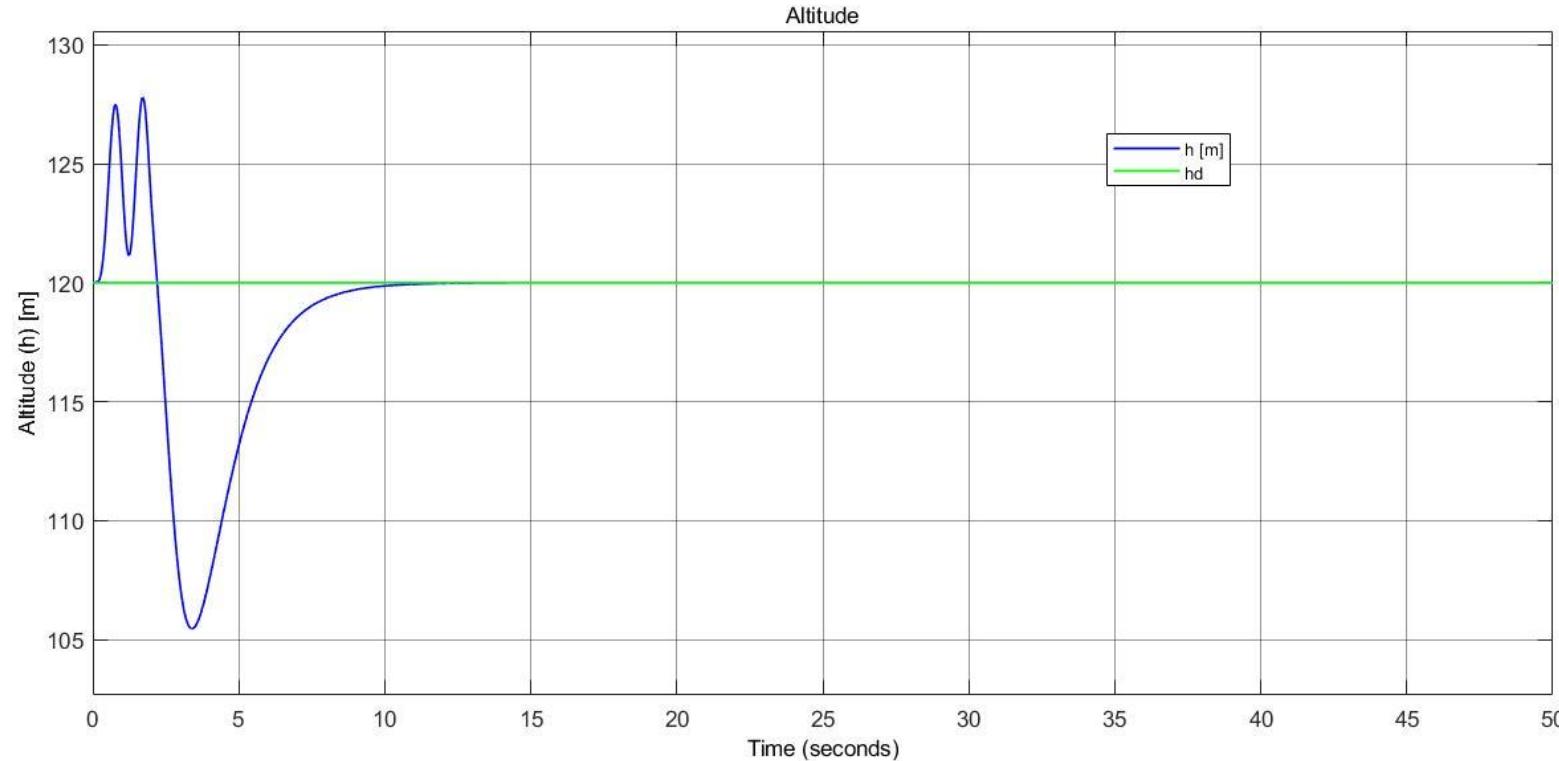


Figure 5.1 Simulink Block for Nonlinear Longitudinal LQR

II. Linear Quadratic Control (LQR)



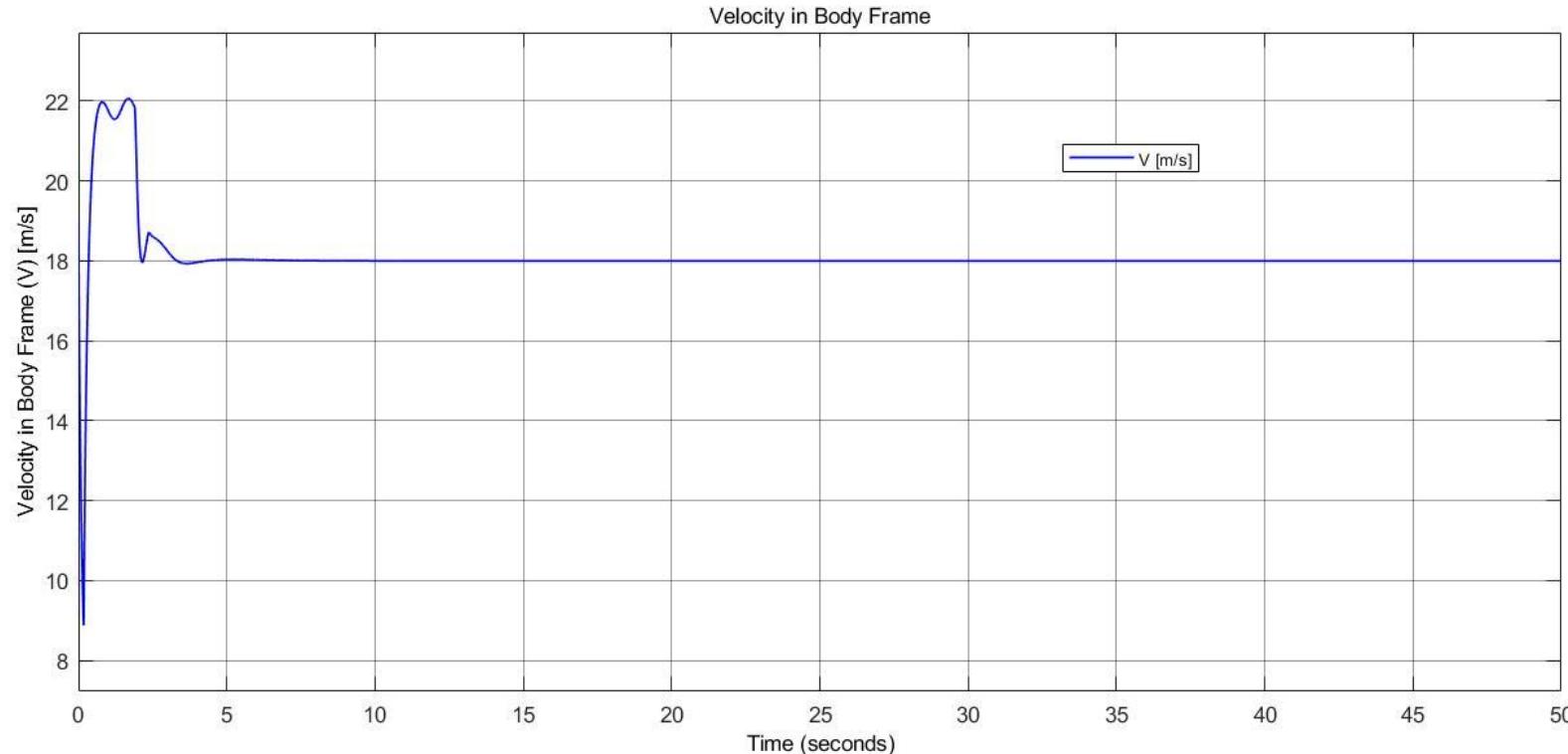
A.1. Altitude Response due to +2 deg Initial Value in Pitch Angle (θ)



II. Linear Quadratic Control (LQR)



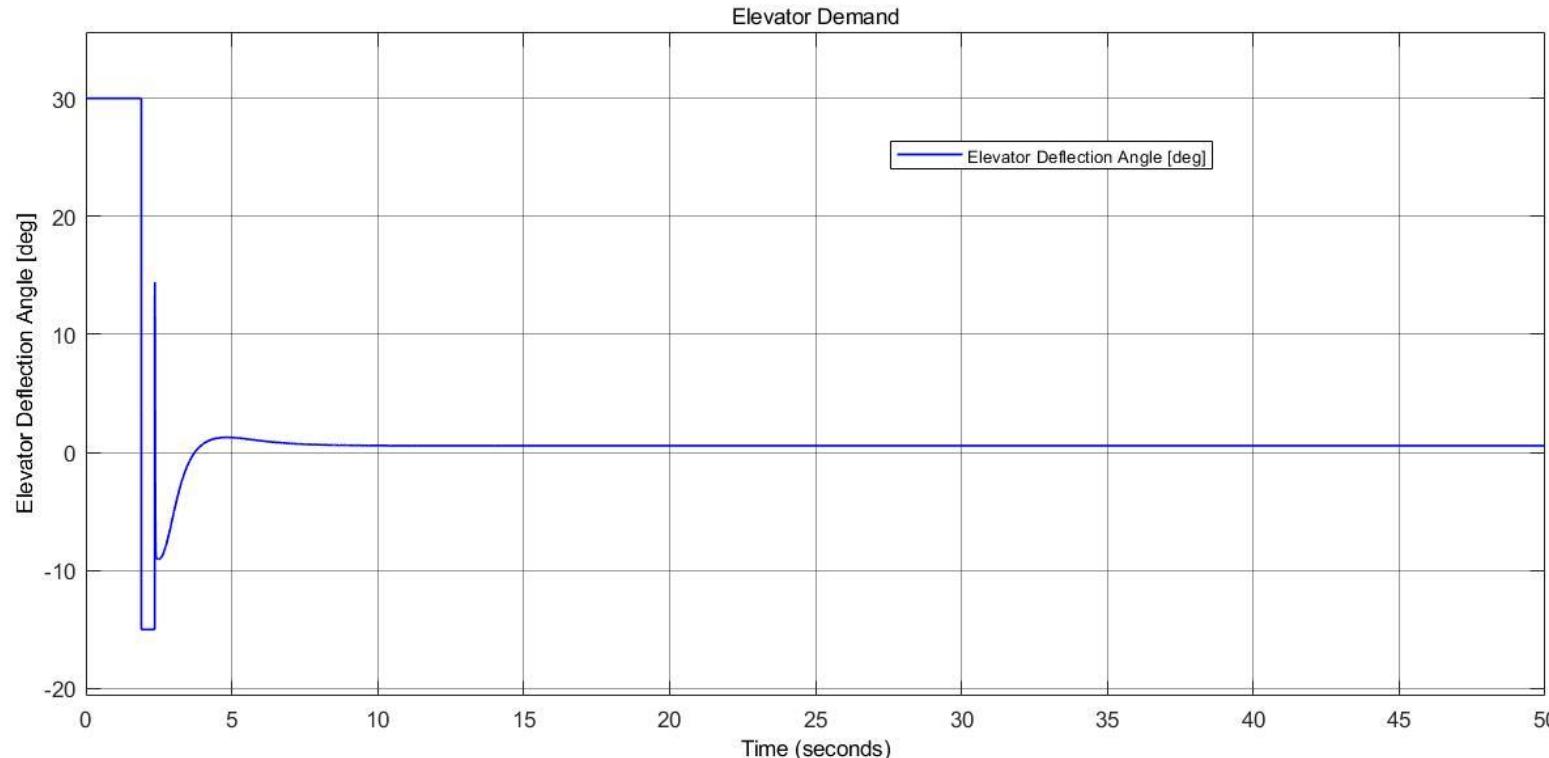
A.2. Velocity Response due to +2 deg Initial Value in Pitch Angle (θ)



II. Linear Quadratic Control (LQR)



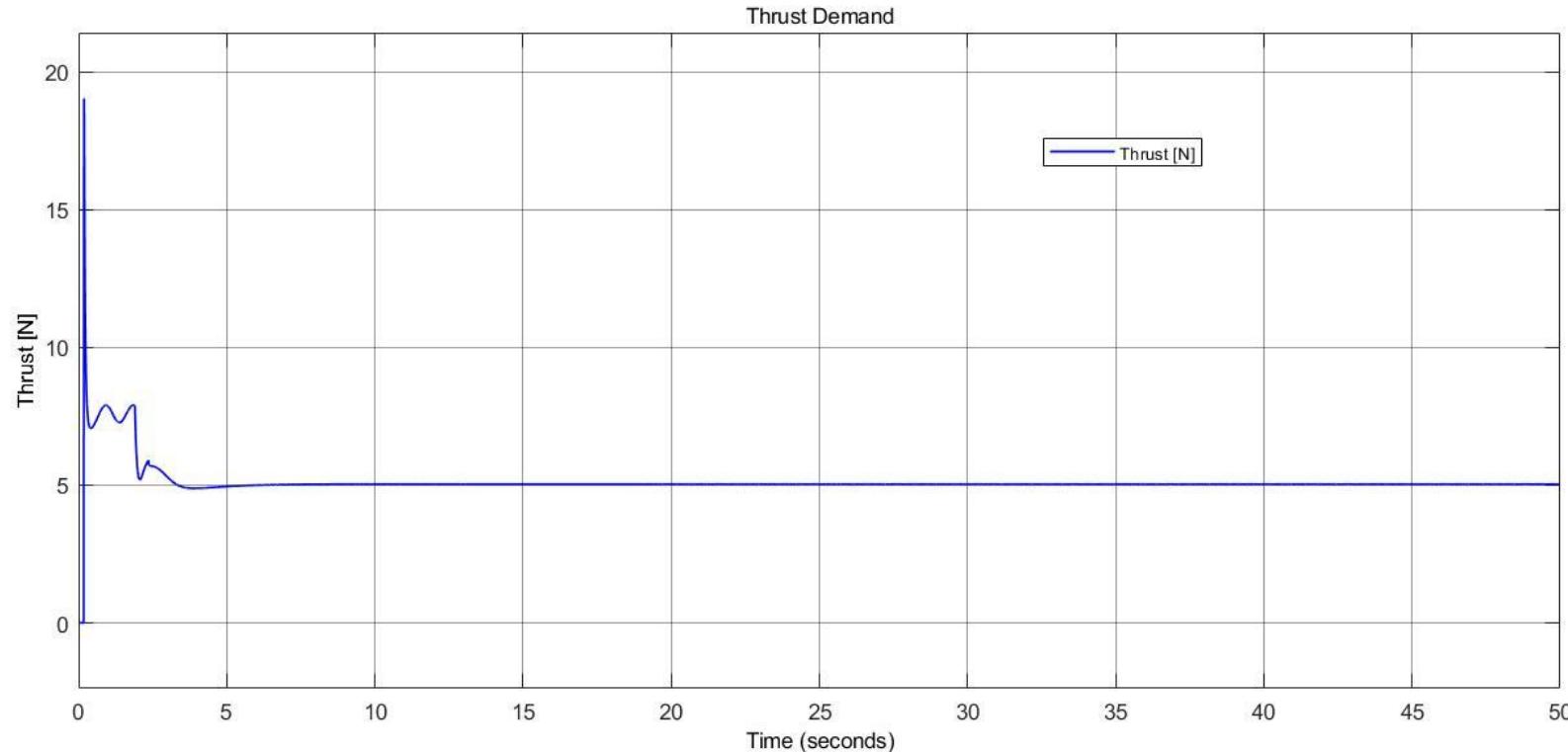
A.3. Elevator Demand due to +2 deg Initial Value in Pitch Angle (θ)



II. Linear Quadratic Control (LQR)



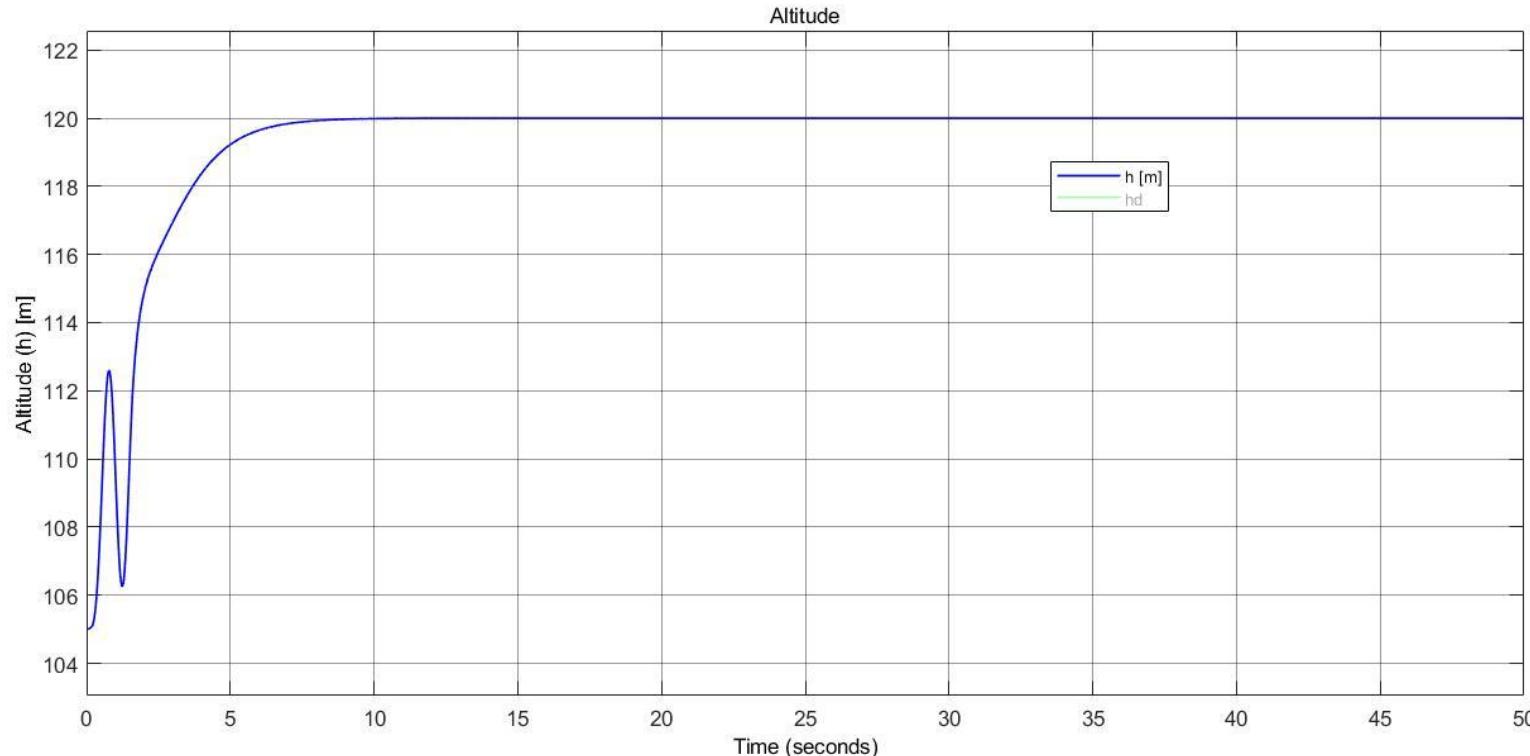
A.4. Thrust Demand due to +2 deg Initial Value in Pitch Angle (θ)



II. Linear Quadratic Control (LQR)



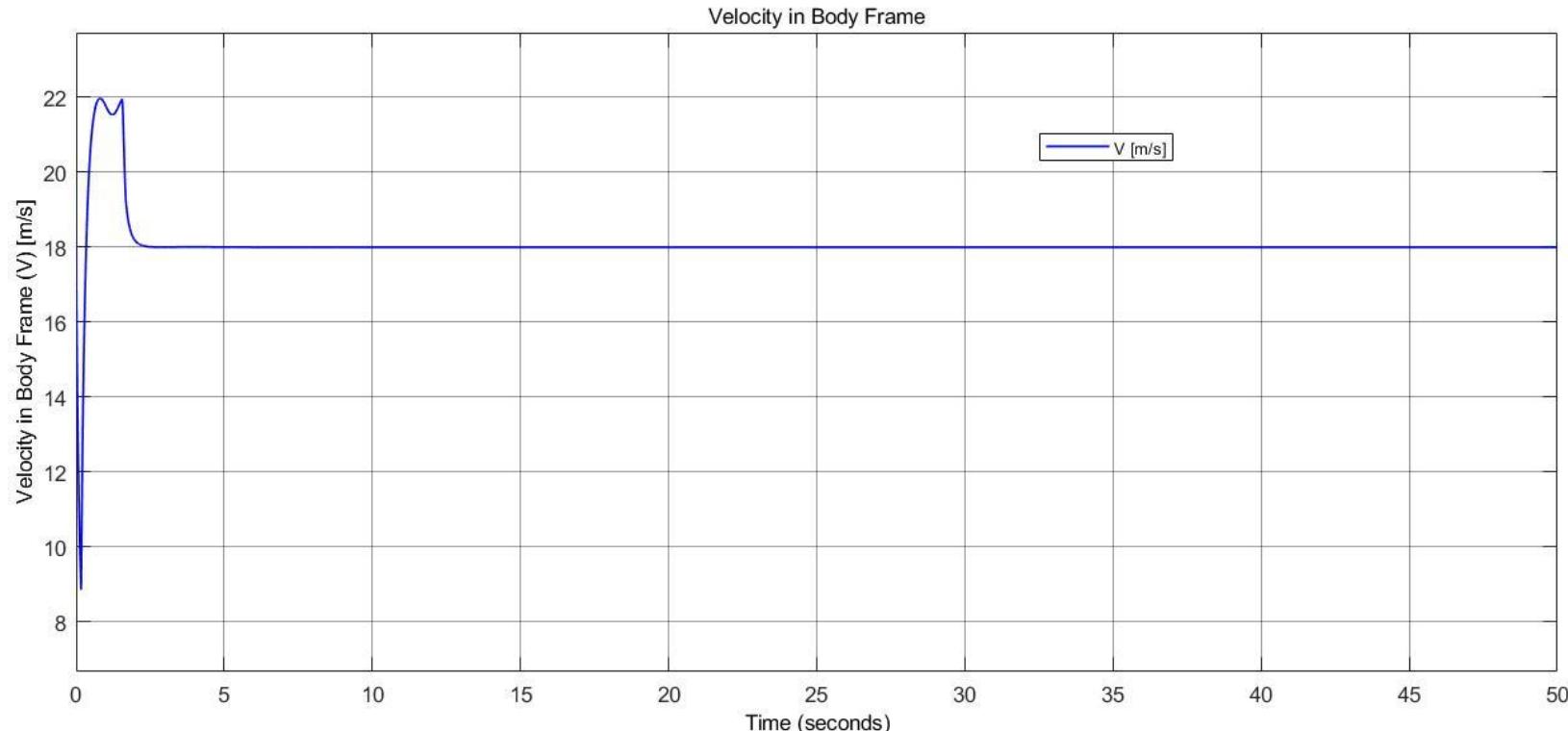
B.1. Altitude Response due to -15 m Initial Value in Altitude (h)



II. Linear Quadratic Control (LQR)



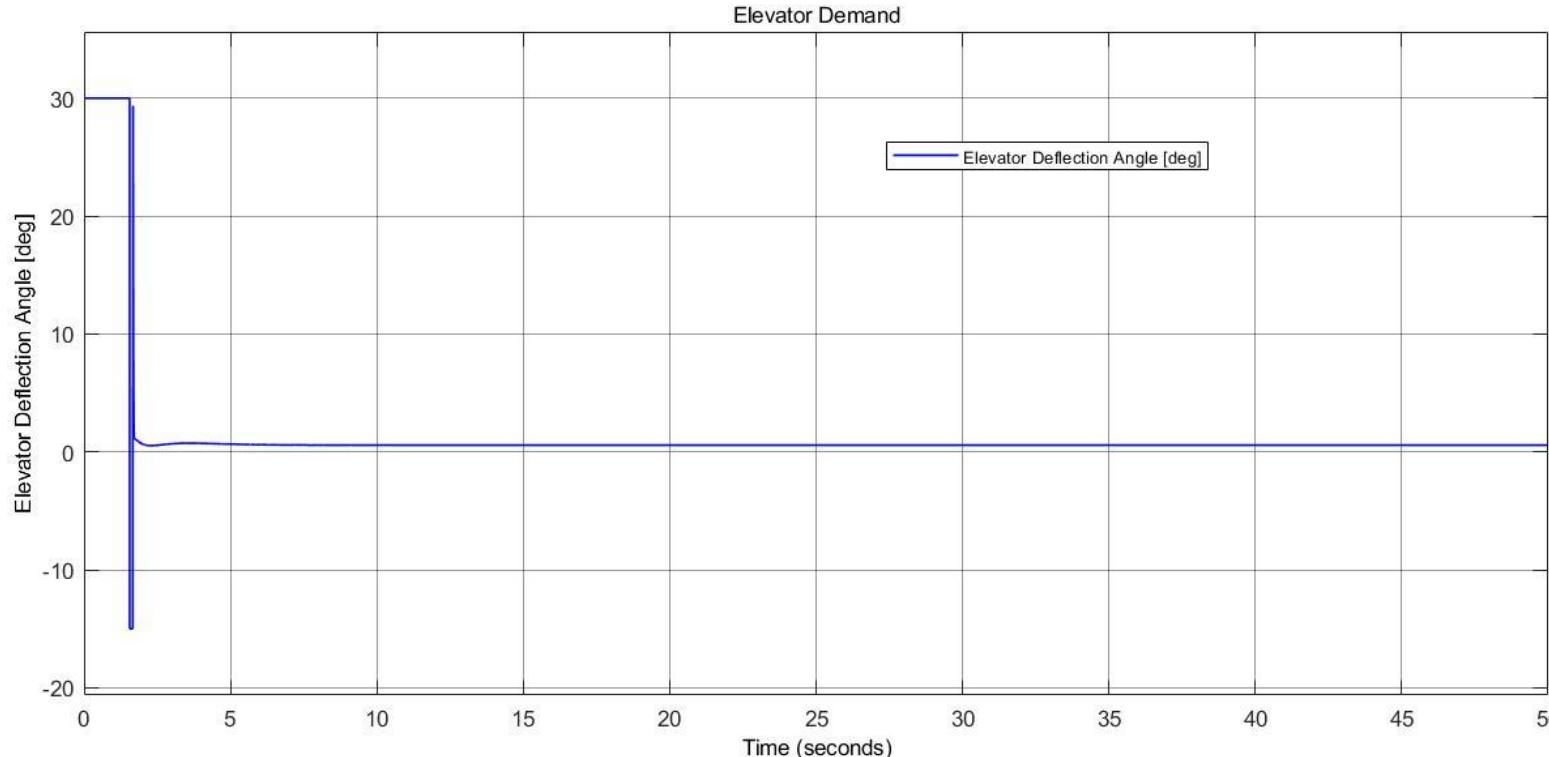
B.2. Velocity Response due to -15 m Initial Value in Altitude (h)



II. Linear Quadratic Control (LQR)



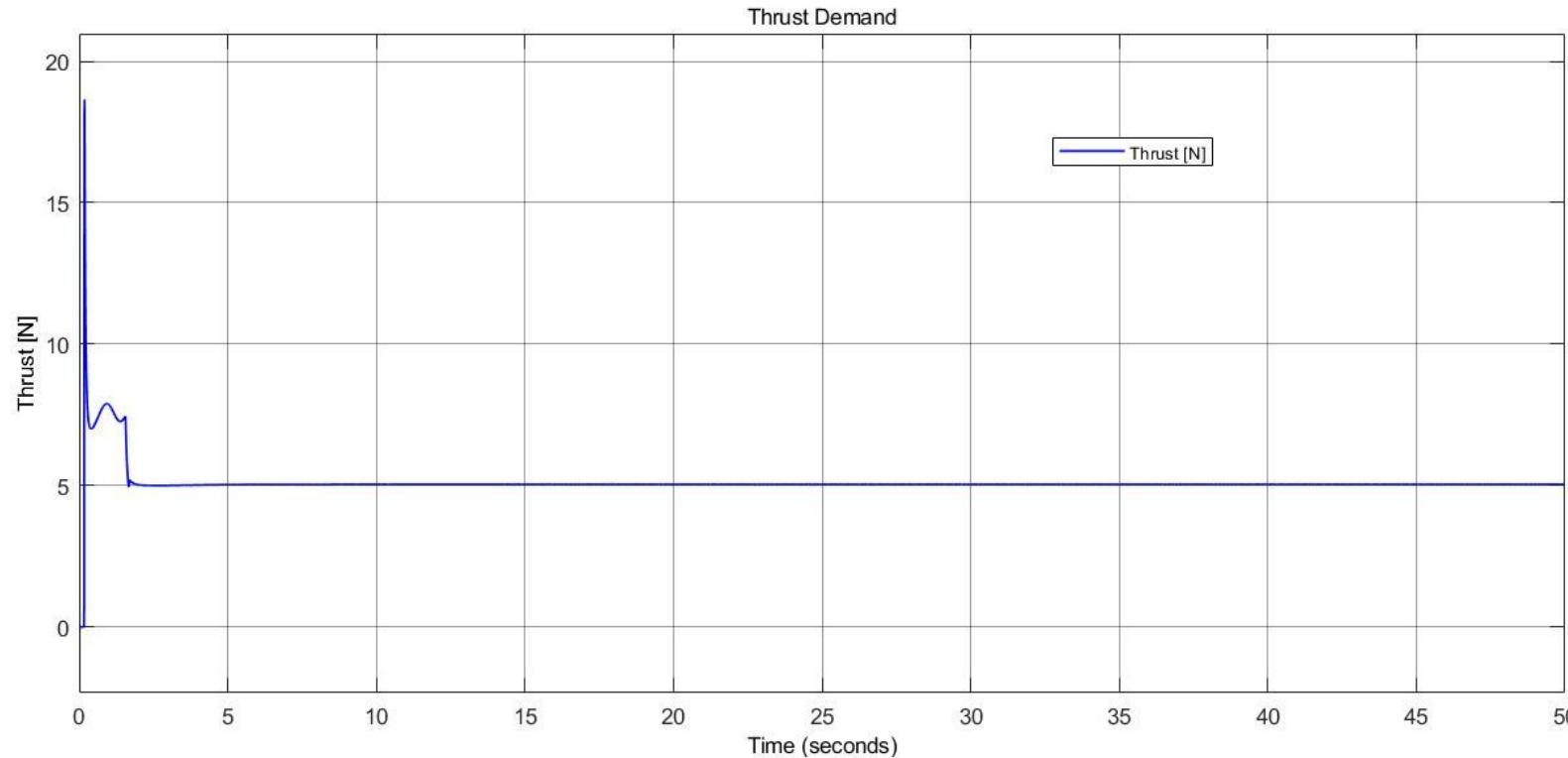
B.3. Elevator Demand due to -15 m Initial Value in Altitude (h)



II. Linear Quadratic Control (LQR)



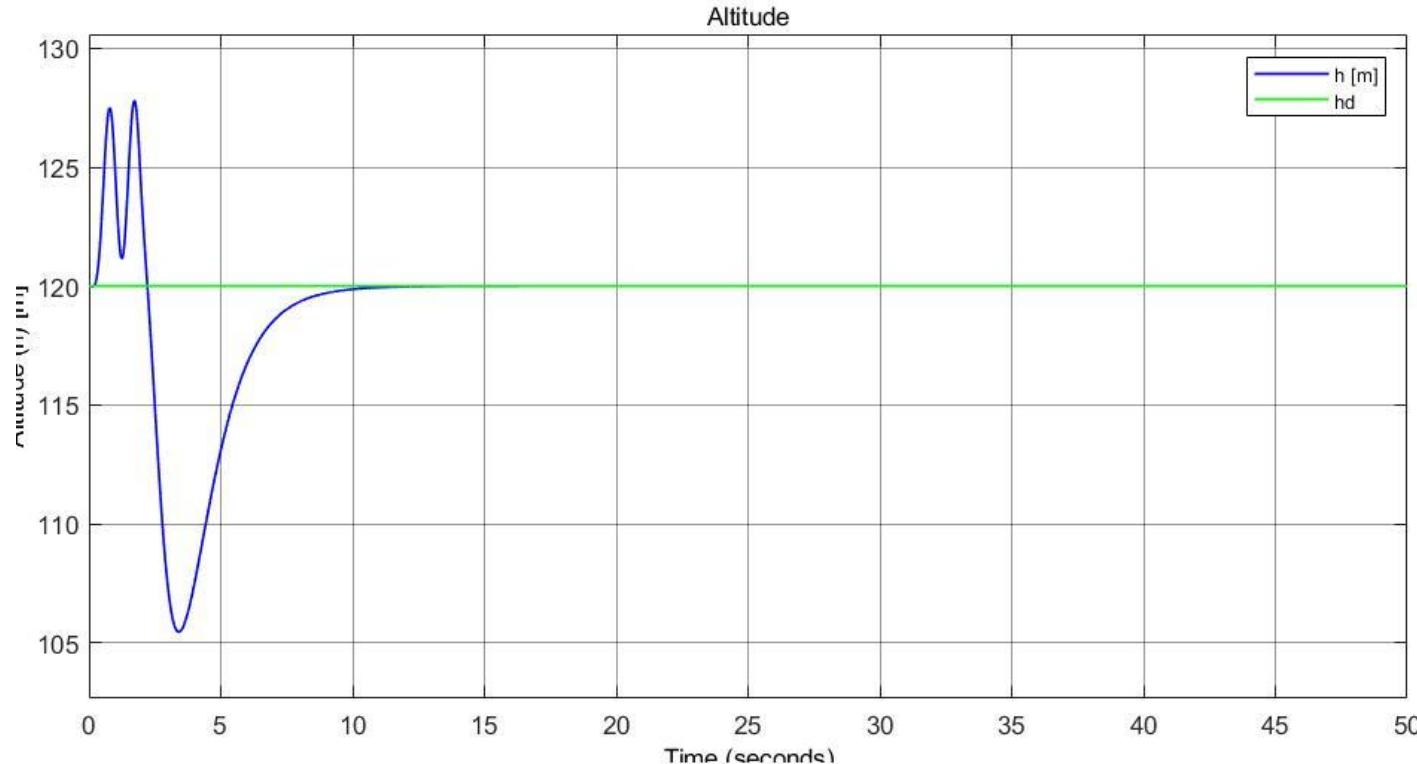
B.4. Thrust Demand due to -15 m Initial Value in Altitude (h)



II. Linear Quadratic Control (LQR)



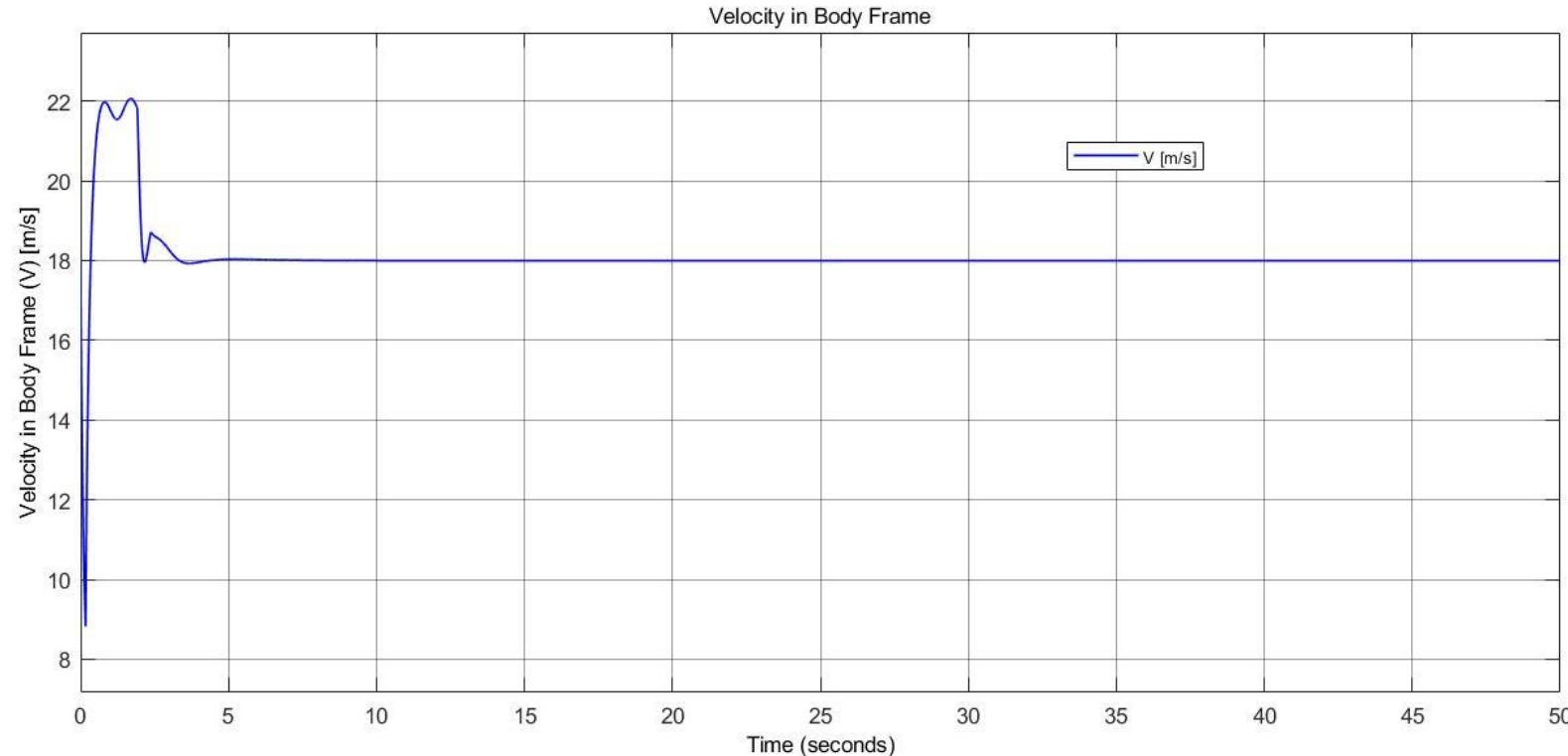
C.1. Altitude Response due to +1 m/s Initial Value in Velocity (u)



II. Linear Quadratic Control (LQR)



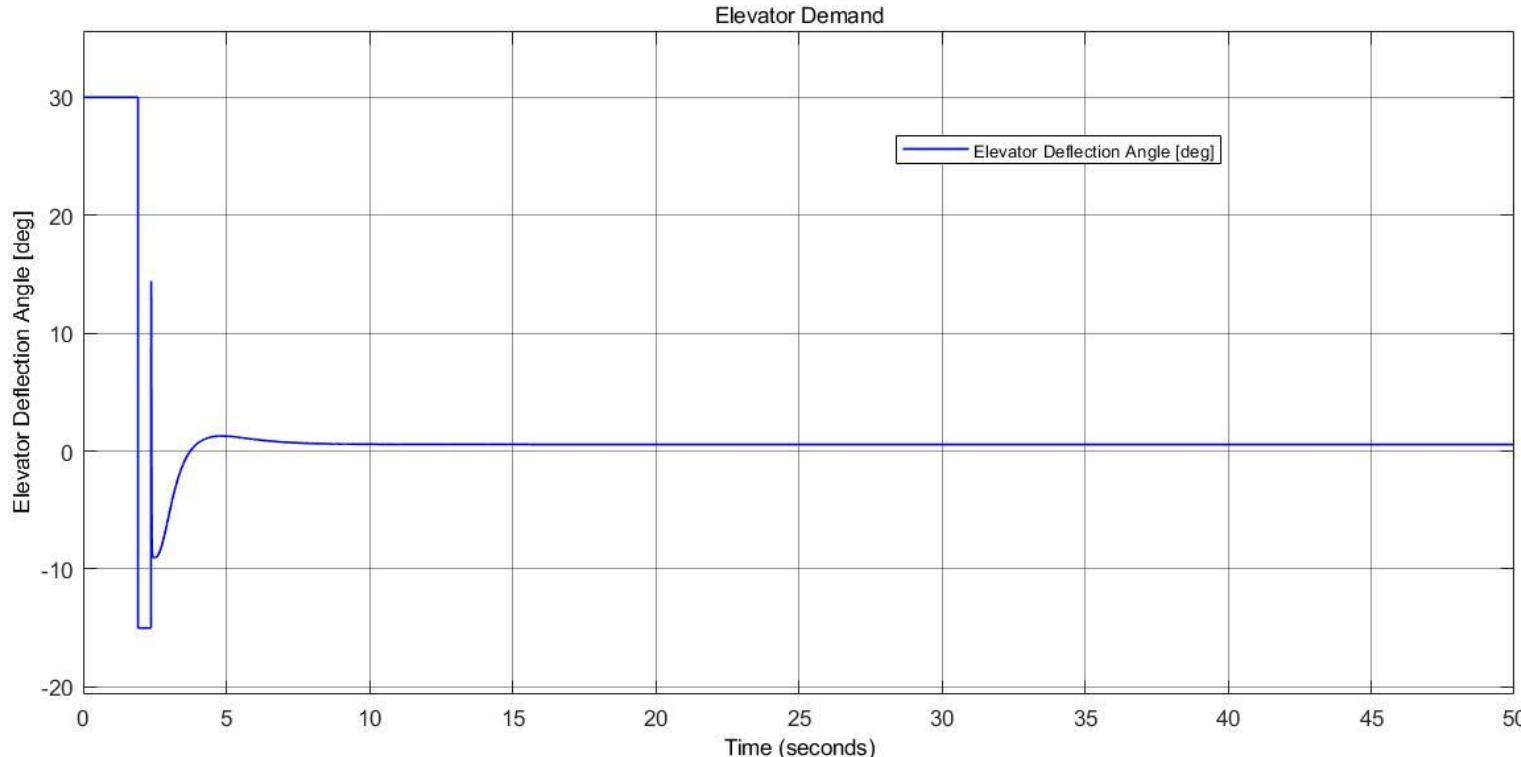
C.2. Velocity Response due to +1 m/s Initial Value in Velocity (u)



II. Linear Quadratic Control (LQR)



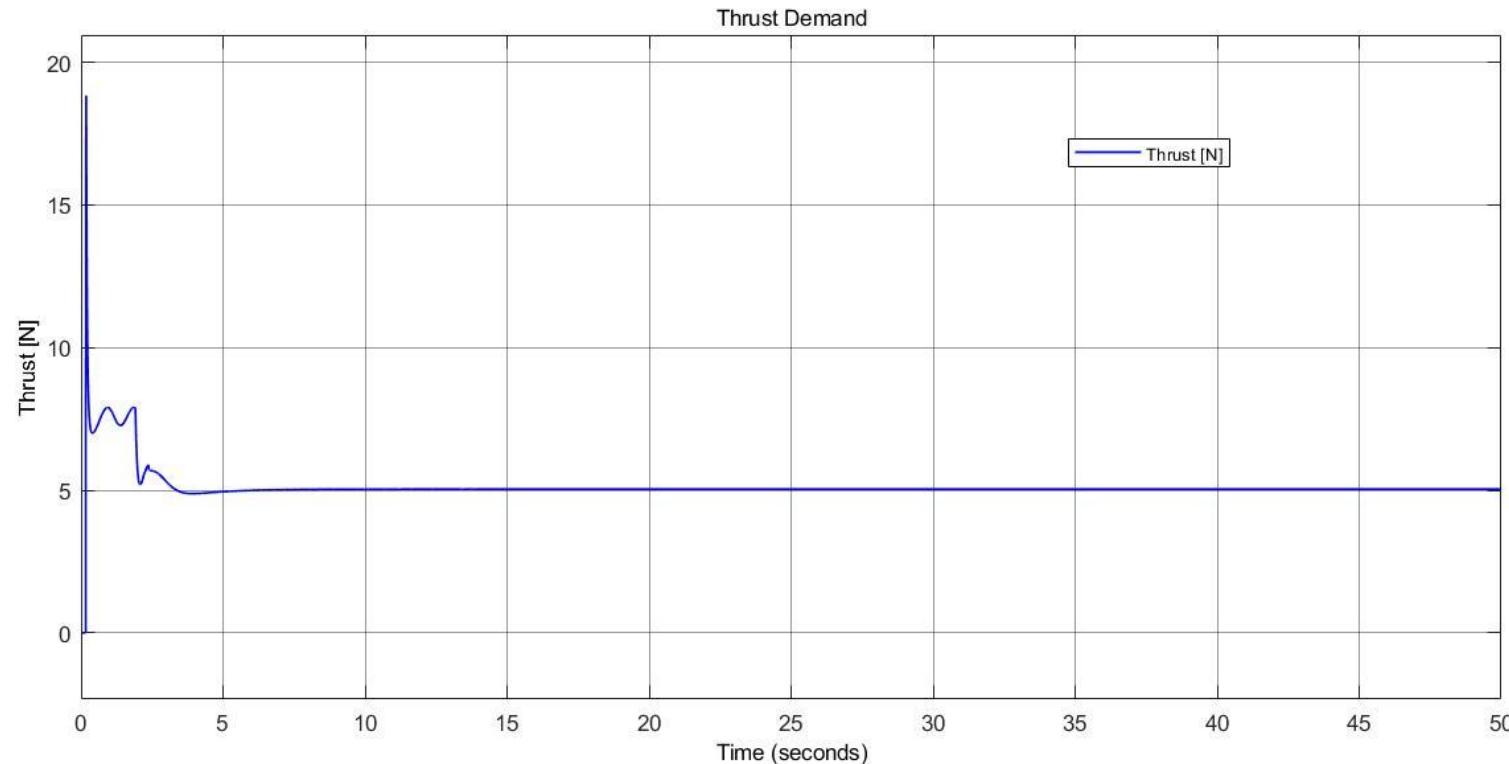
C.3. Elevator Demand due to +1 m/s Initial Value in Velocity (u)



II. Linear Quadratic Control (LQR)



C.4. Thrust Demand due to +1 m/s Initial Value in Velocity (u)



II. Linear Quadratic Control (LQR)



Lateral LQR

$Q =$

$$\begin{matrix} 100000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 120000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1100 \end{matrix}$$

$R =$

$$\begin{matrix} 100 & 0 \\ 0 & 100 \end{matrix}$$

$K =$

$$\begin{matrix} -13.9583 & 2.7390 & 5.8603 & 2.1993 & 0.7443 & -15.0770 & 2.9688 \\ 29.3557 & 1.7473 & -24.3051 & 16.2962 & 0.3562 & 31.1885 & 1.4349 \end{matrix}$$

II. Linear Quadratic Control (LQR)



Lateral LQR

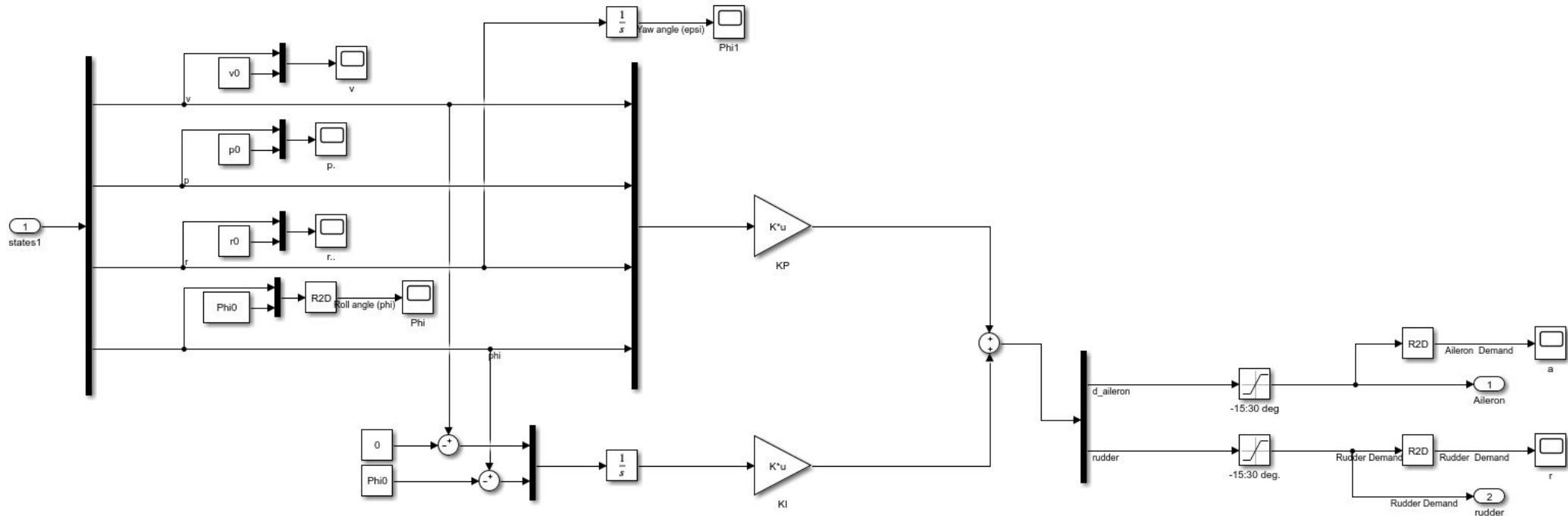
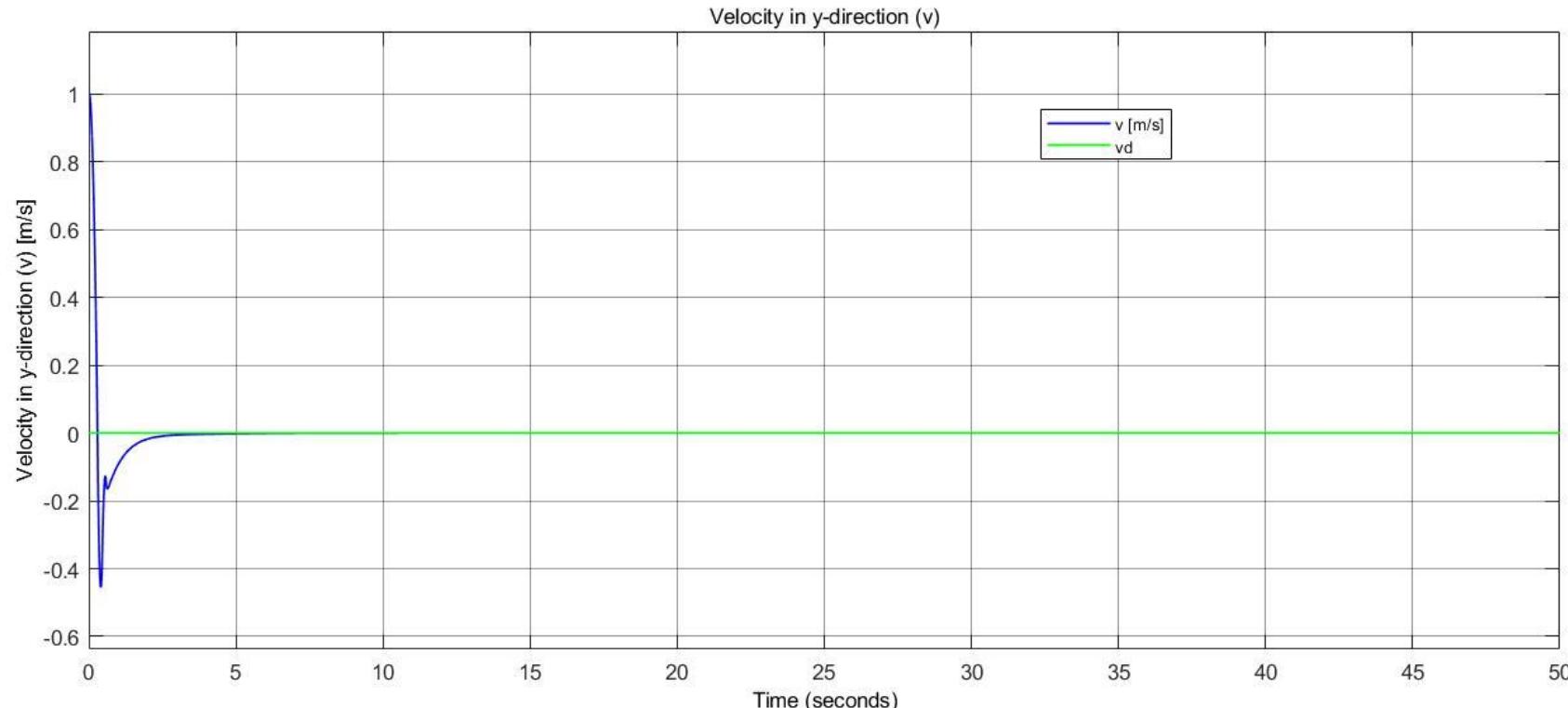


Figure 5.2 Simulink Block for Nonlinear Lateral LQR

II. Linear Quadratic Control (LQR)



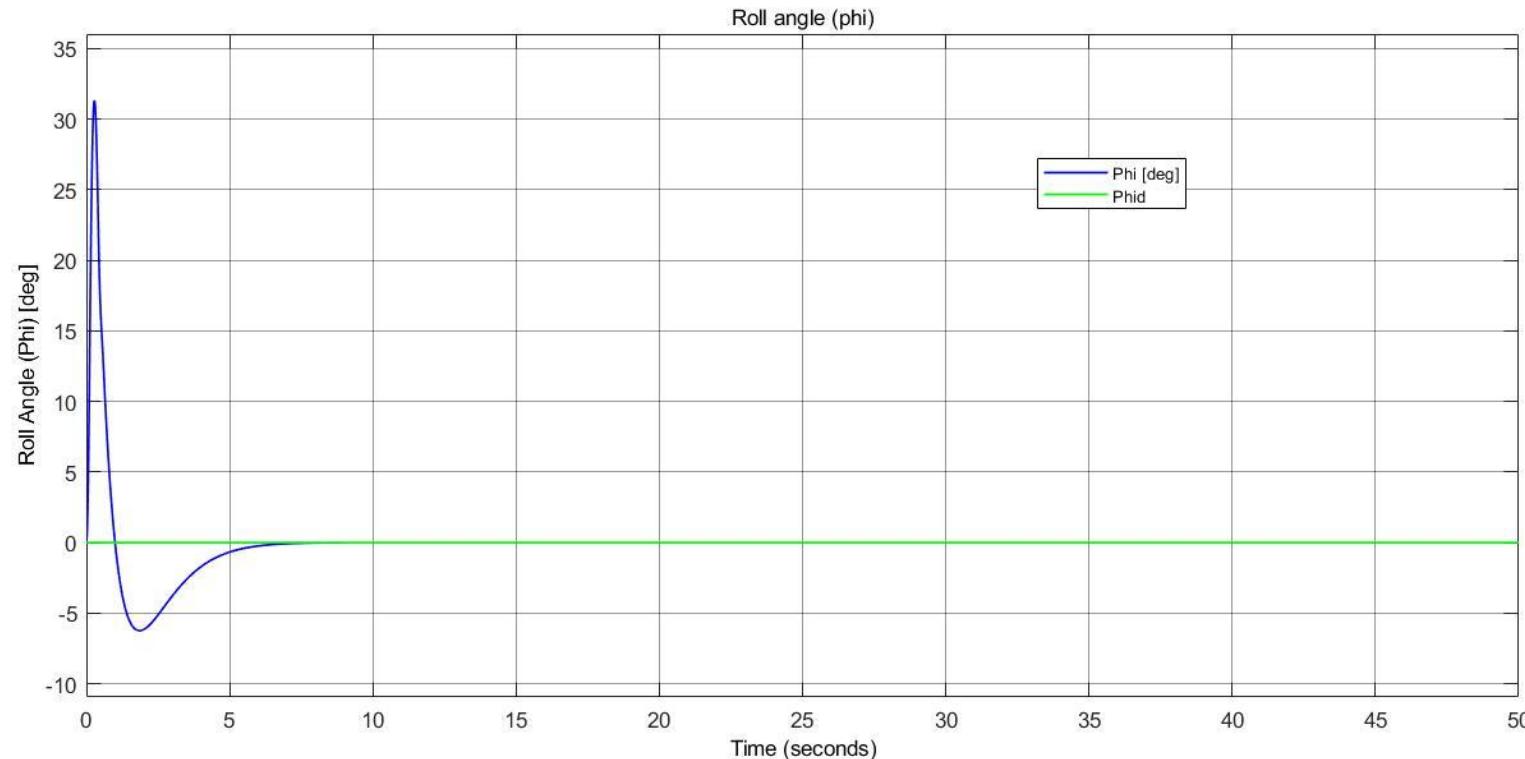
A.1. Velocity Response due to +1 m/s Initial Value in y- velocity (v)



II. Linear Quadratic Control (LQR)



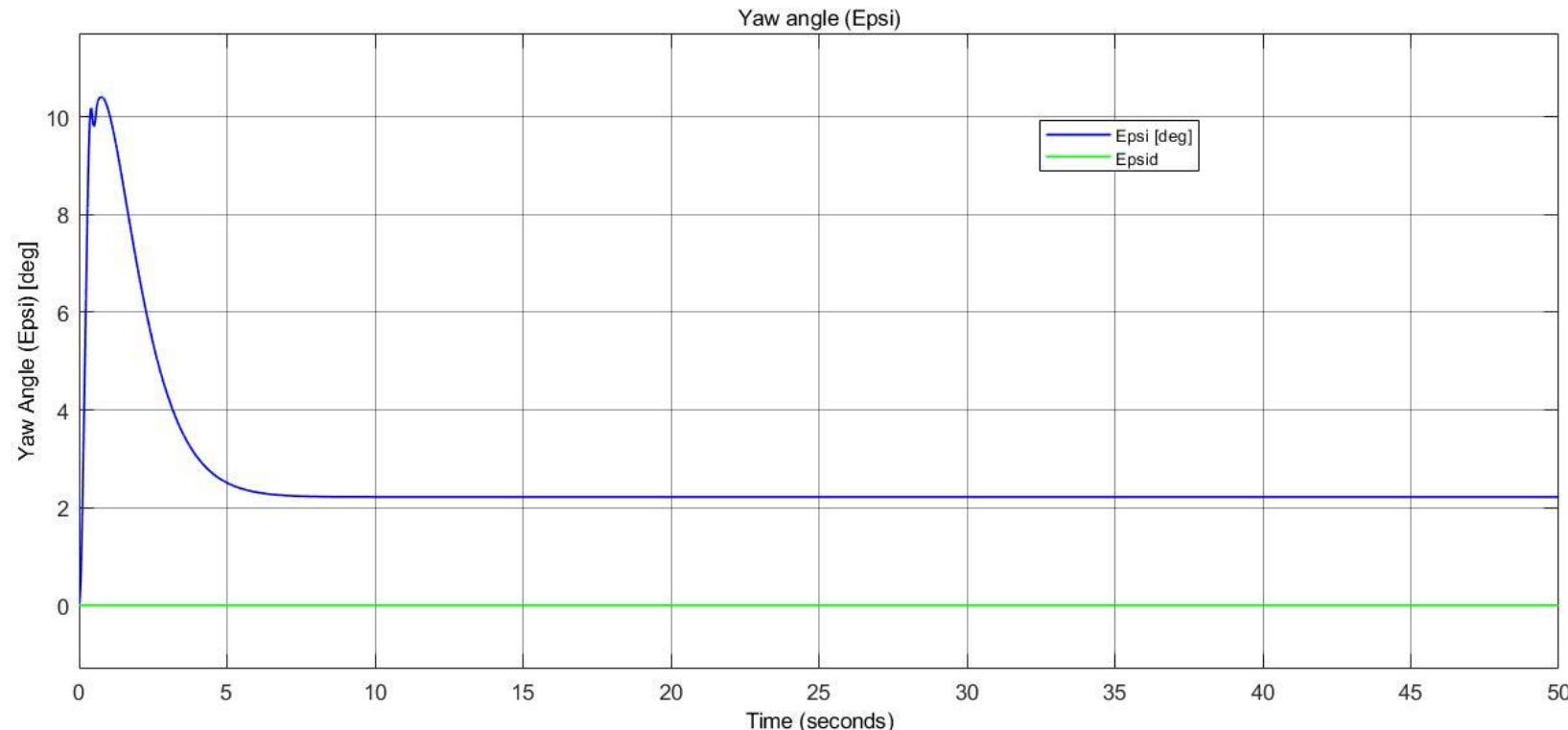
A.2. Roll Angle Response due to +1 m/s Initial Value in y- velocity (v)



II. Linear Quadratic Control (LQR)



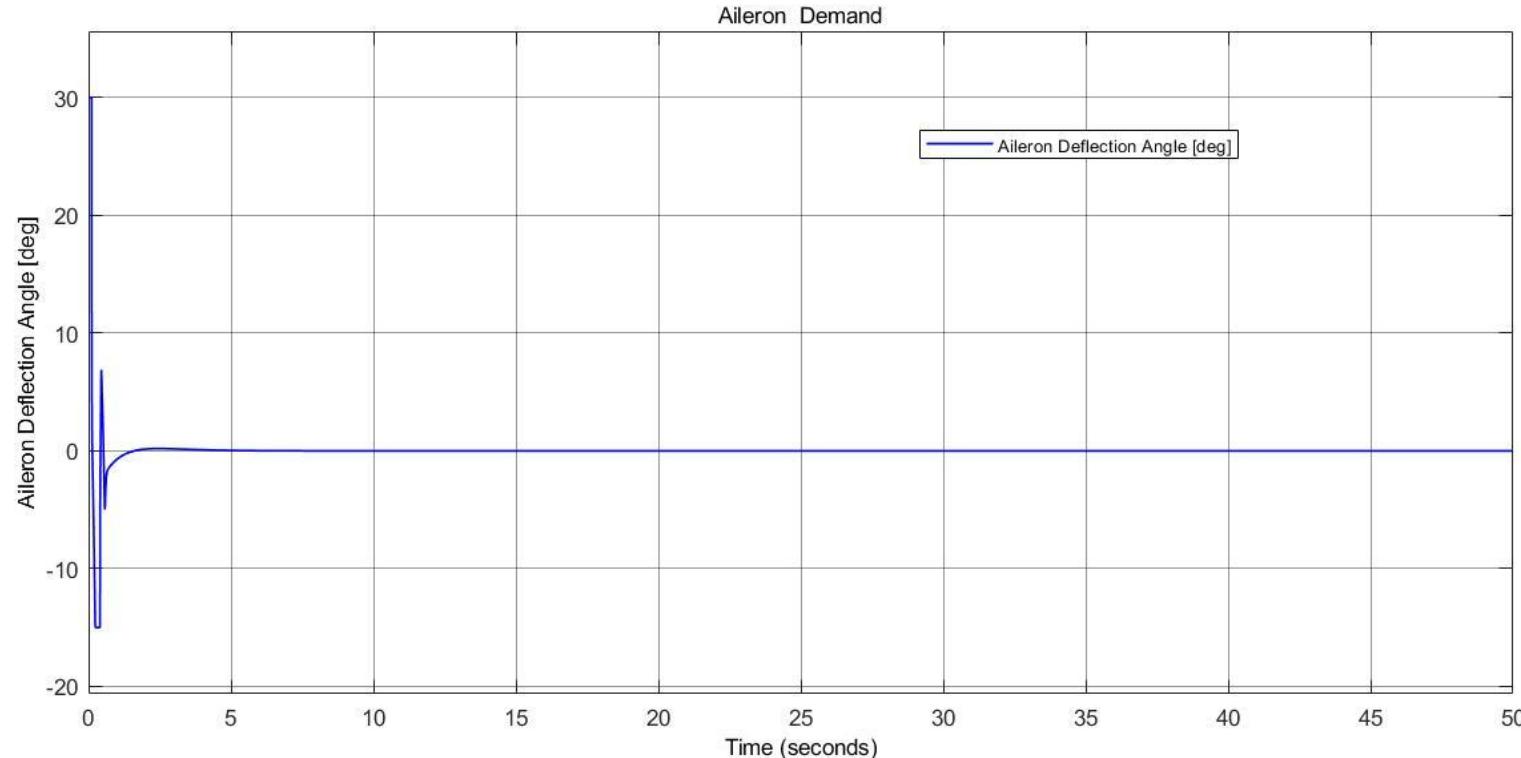
A.3. Yaw Angle Response due to +1 m/s Initial Value in y- velocity (v)



II. Linear Quadratic Control (LQR)



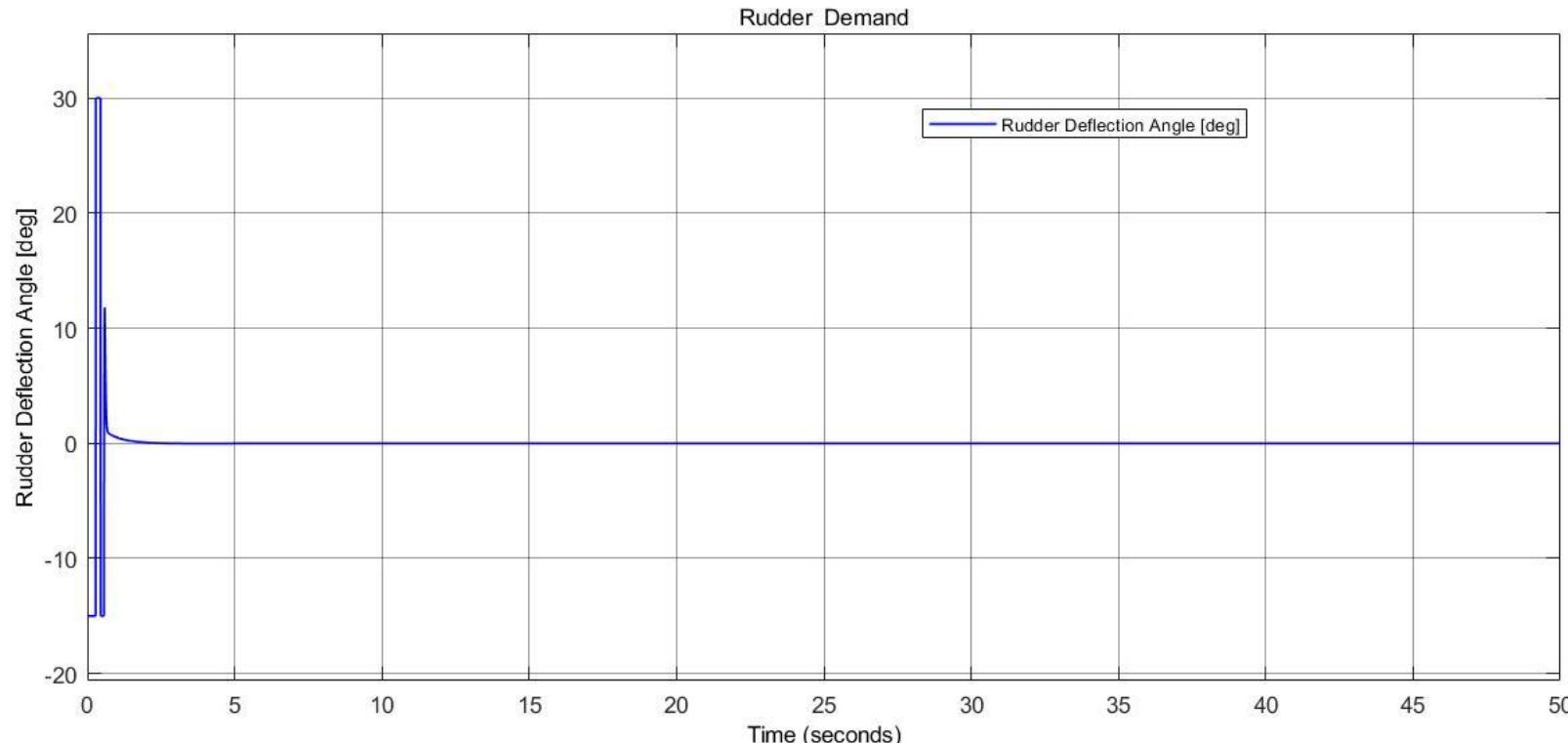
A.4. Aileron Demand due to +1 m/s Initial Value in y- velocity (v)



II. Linear Quadratic Control (LQR)



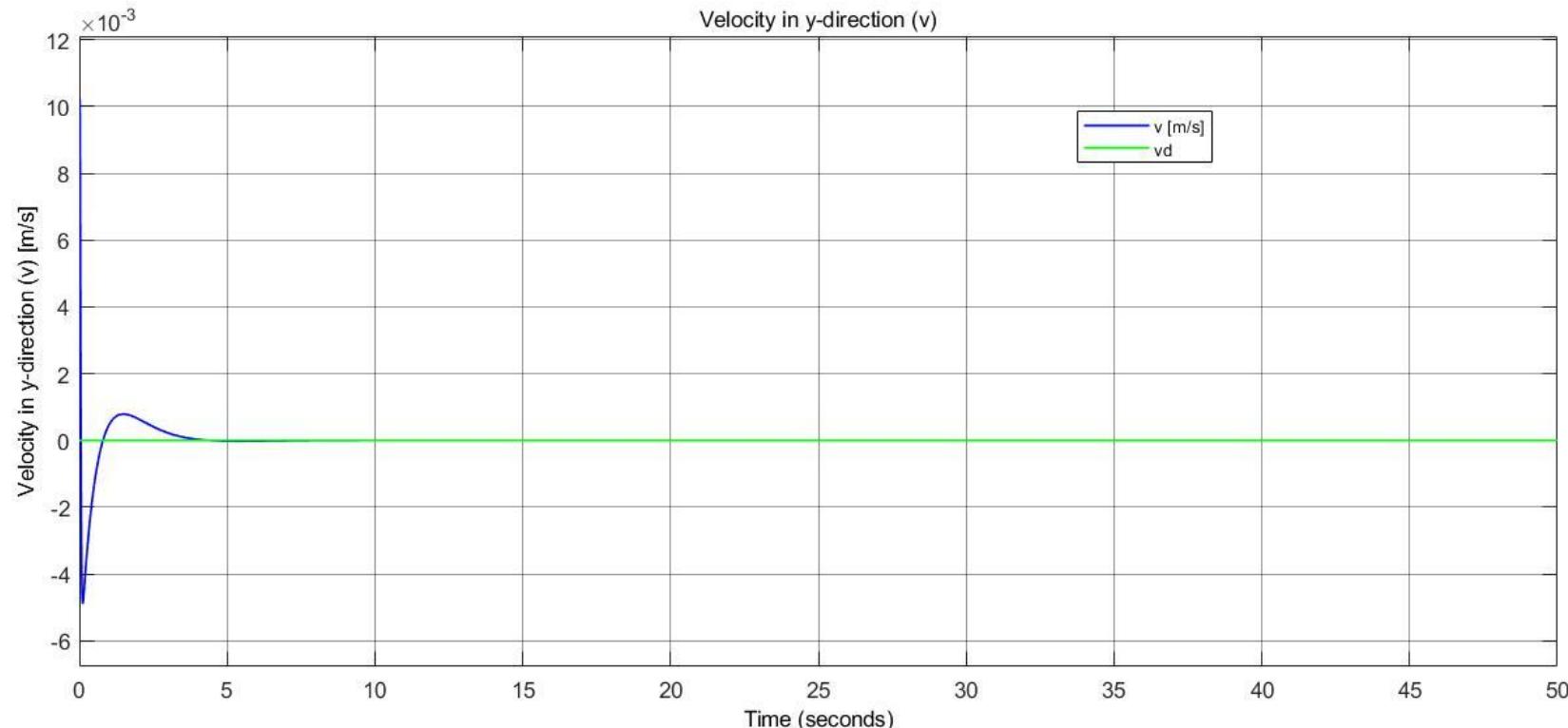
A.5. Rudder Demand due to +1 m/s Initial Value in y- velocity (v)



II. Linear Quadratic Control (LQR)



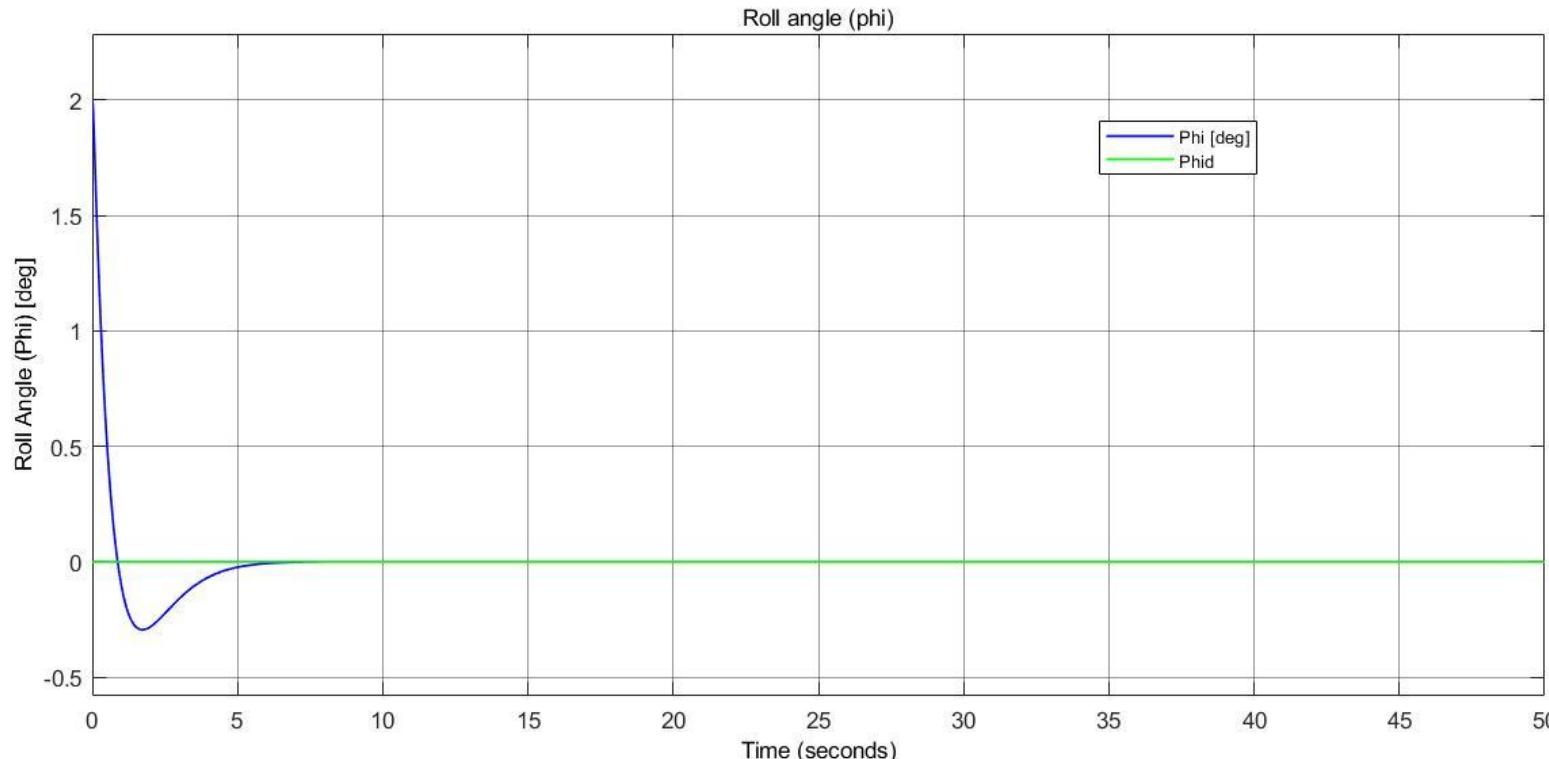
B.1. Velocity Response due to +2 deg Initial Value in Roll Angle (φ)



II. Linear Quadratic Control (LQR)



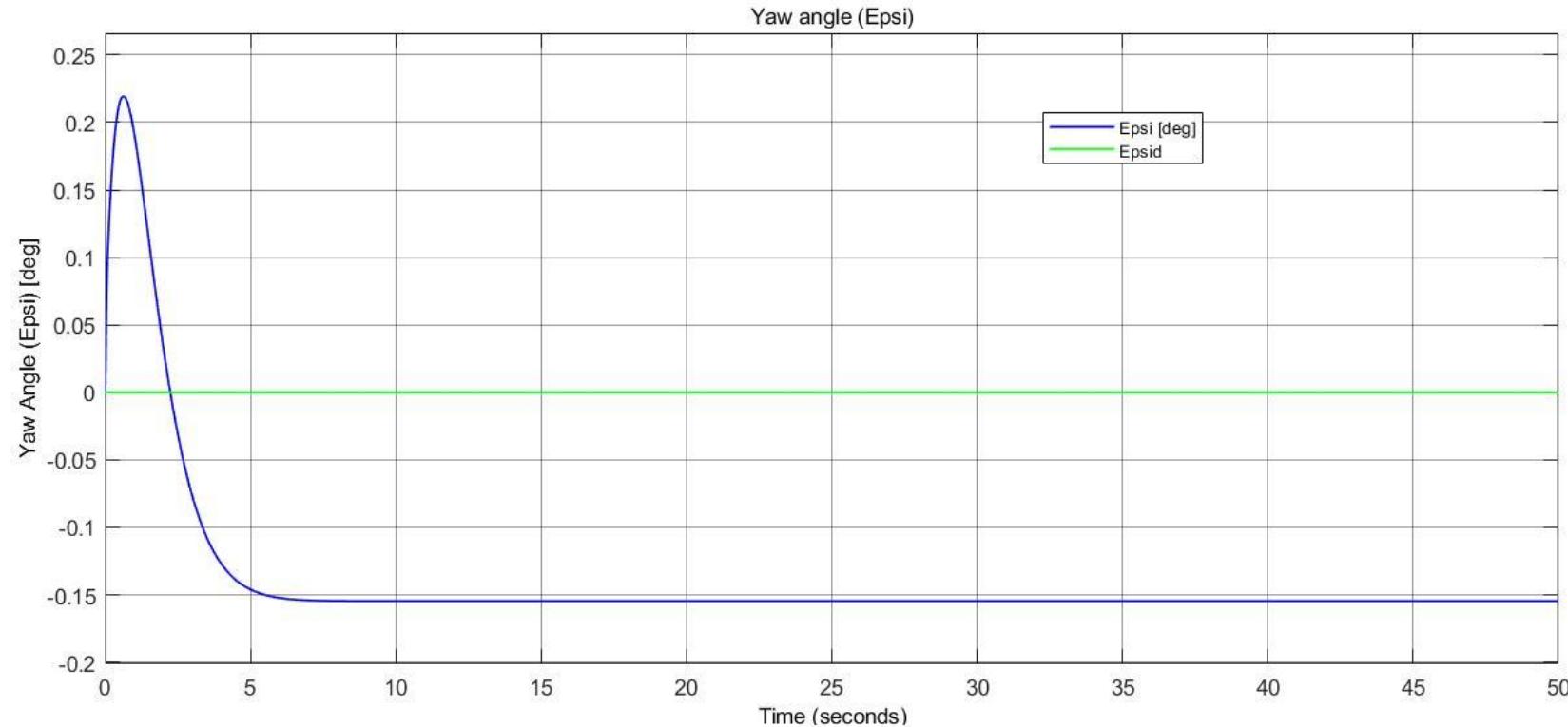
B.2. Roll Angle Response due to +2 deg Initial Value in Roll Angle (φ)



II. Linear Quadratic Control (LQR)



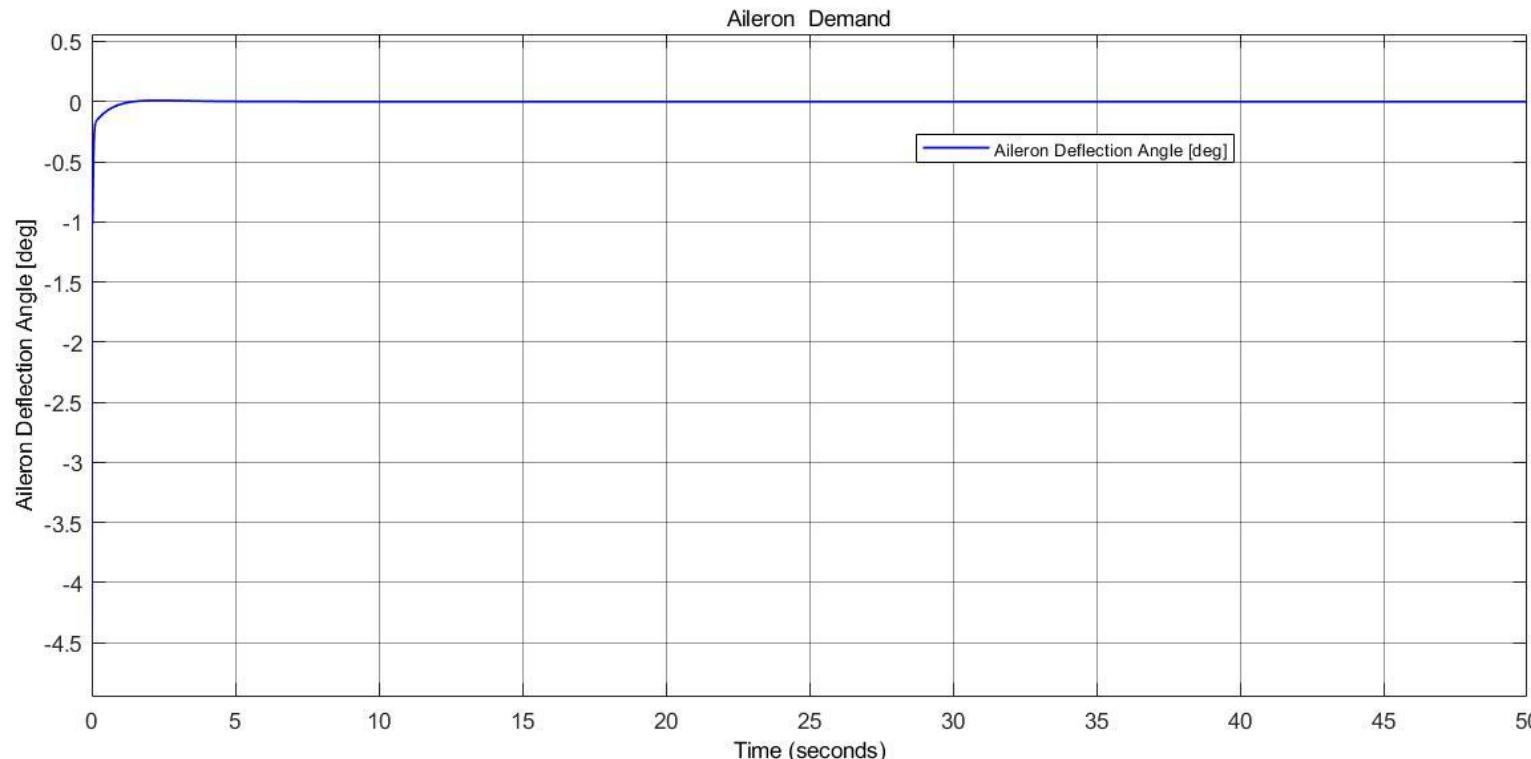
B.3. Yaw Angle Response due to +2 deg Initial Value in Roll Angle (φ)



II. Linear Quadratic Control (LQR)



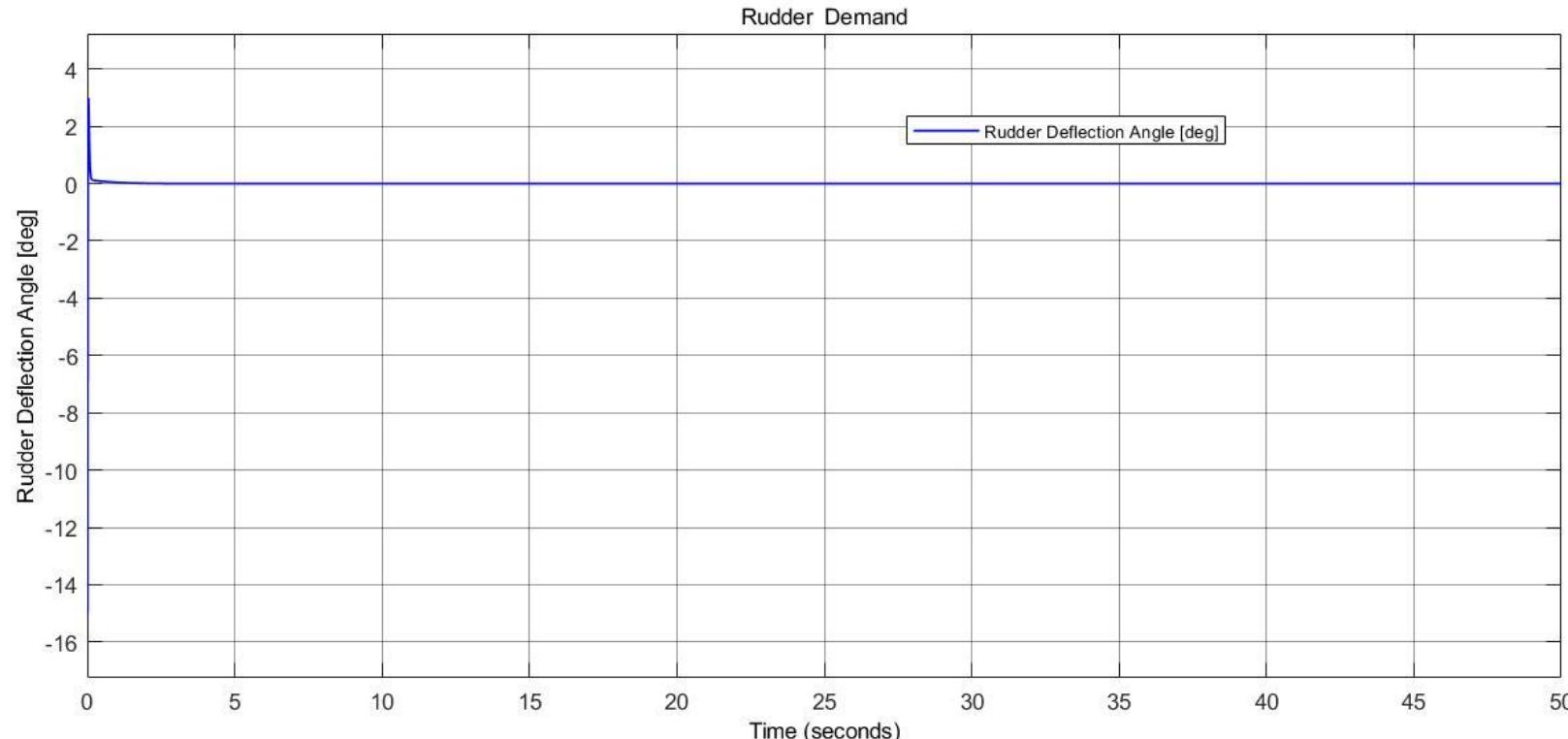
B.4. Aileron Demand due to +2 deg Initial Value in Roll Angle (φ)



II. Linear Quadratic Control (LQR)



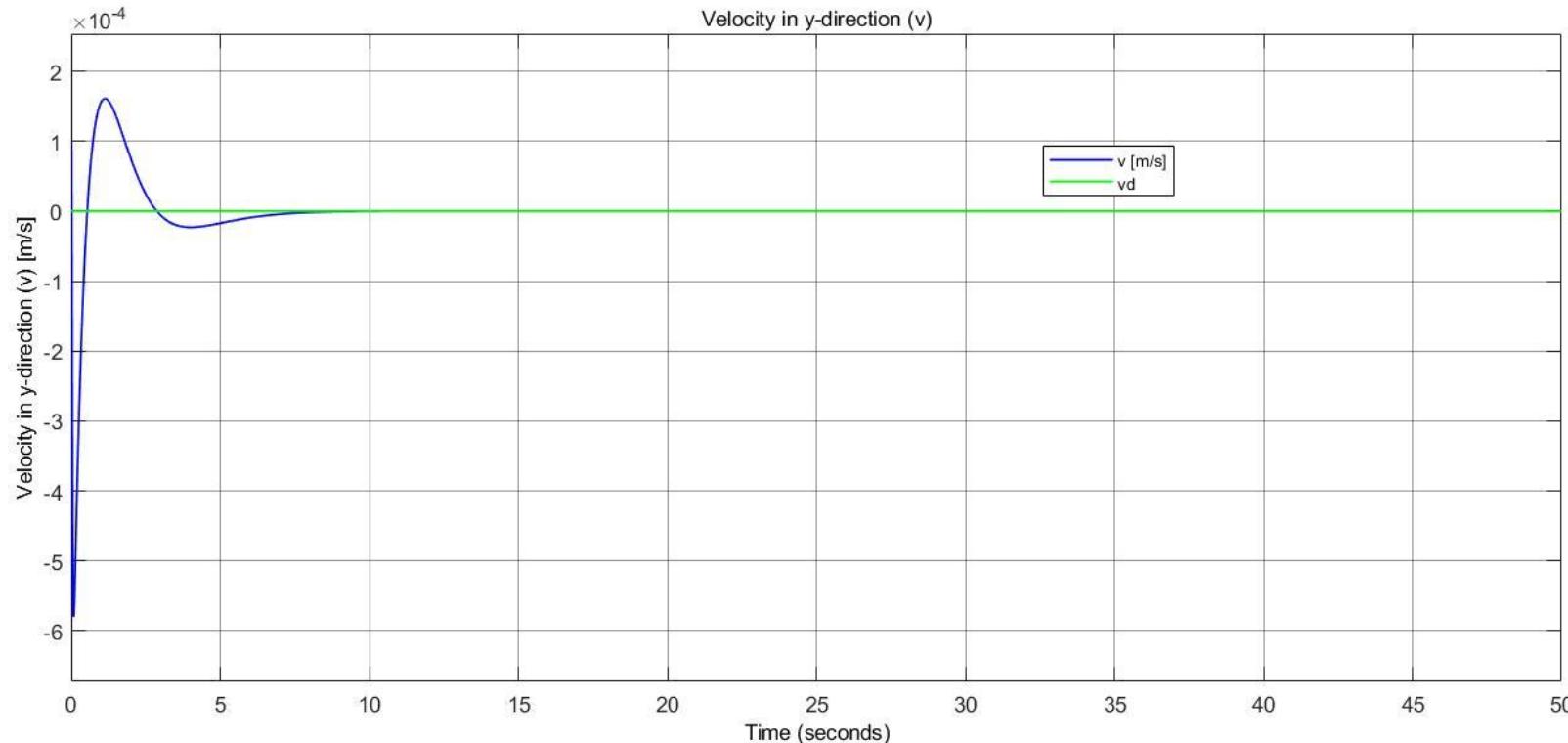
B.5. Rudder Demand due to +2 deg Initial Value in Roll Angle (φ)



II. Linear Quadratic Control (LQR)



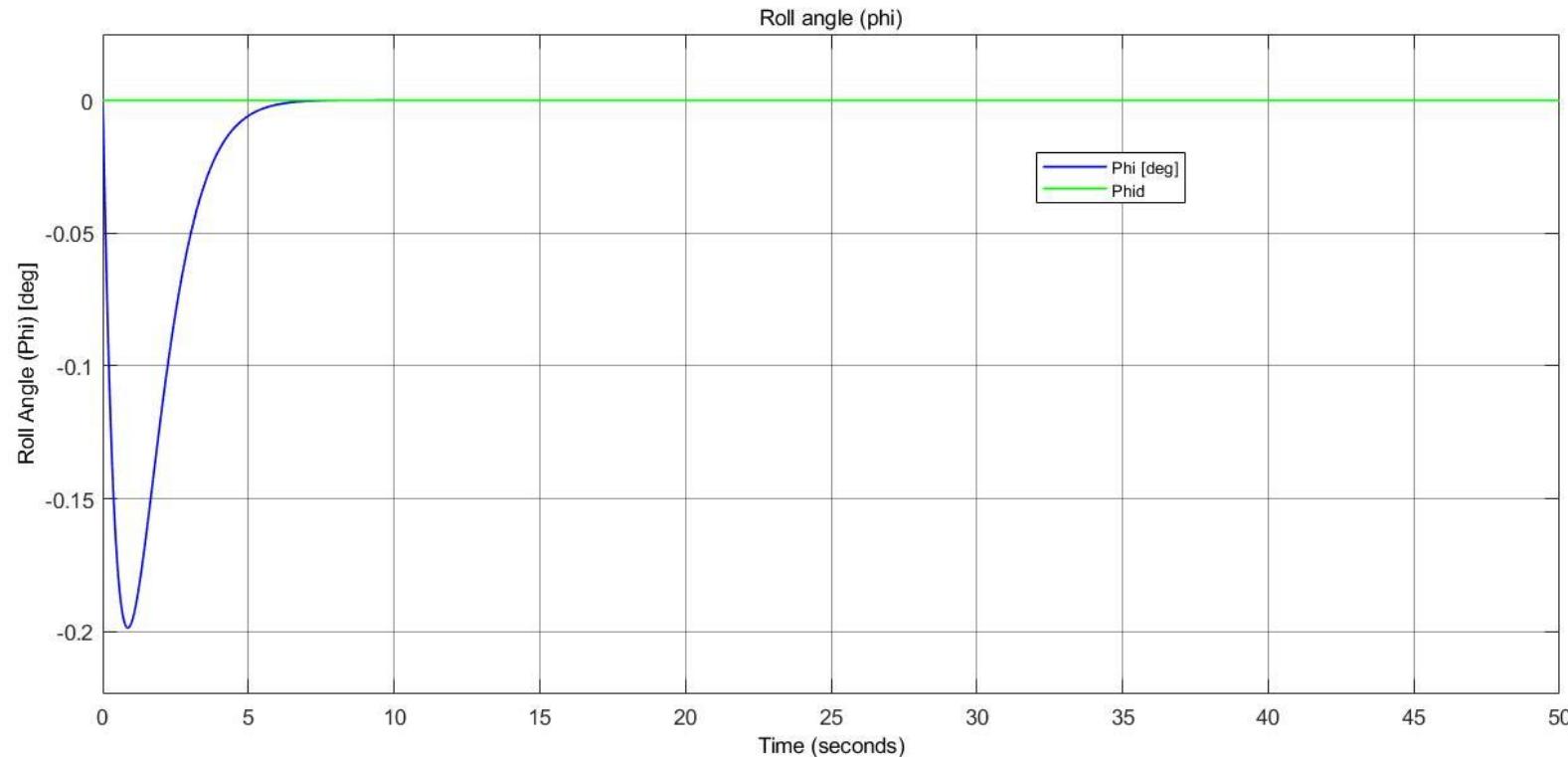
C.1. Velocity Response due to +2 deg Initial Value in Yaw Angle (ψ)



II. Linear Quadratic Control (LQR)



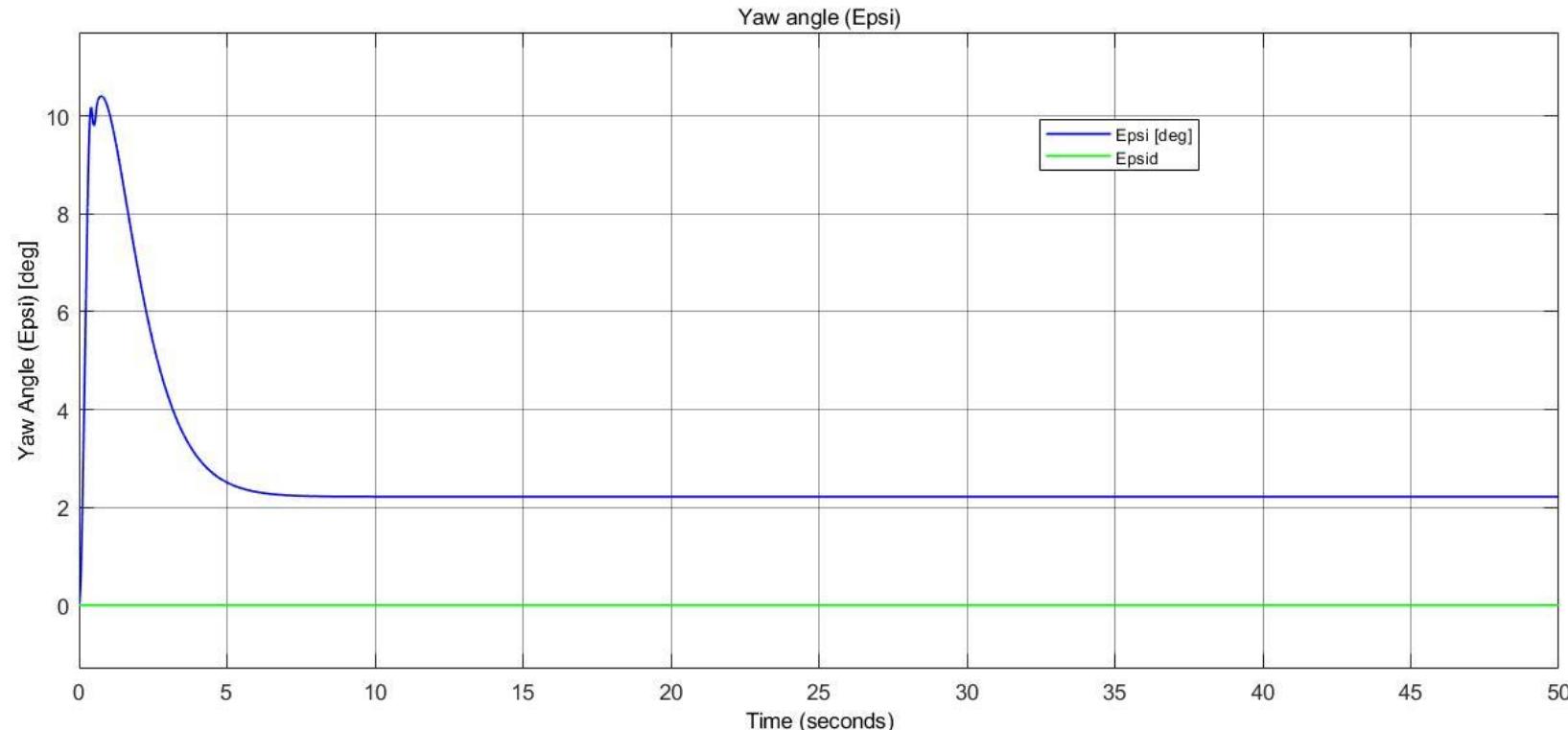
C.2. Roll Angle Response due to +1 m/s Initial Value in Yaw Angle (ψ)



II. Linear Quadratic Control (LQR)



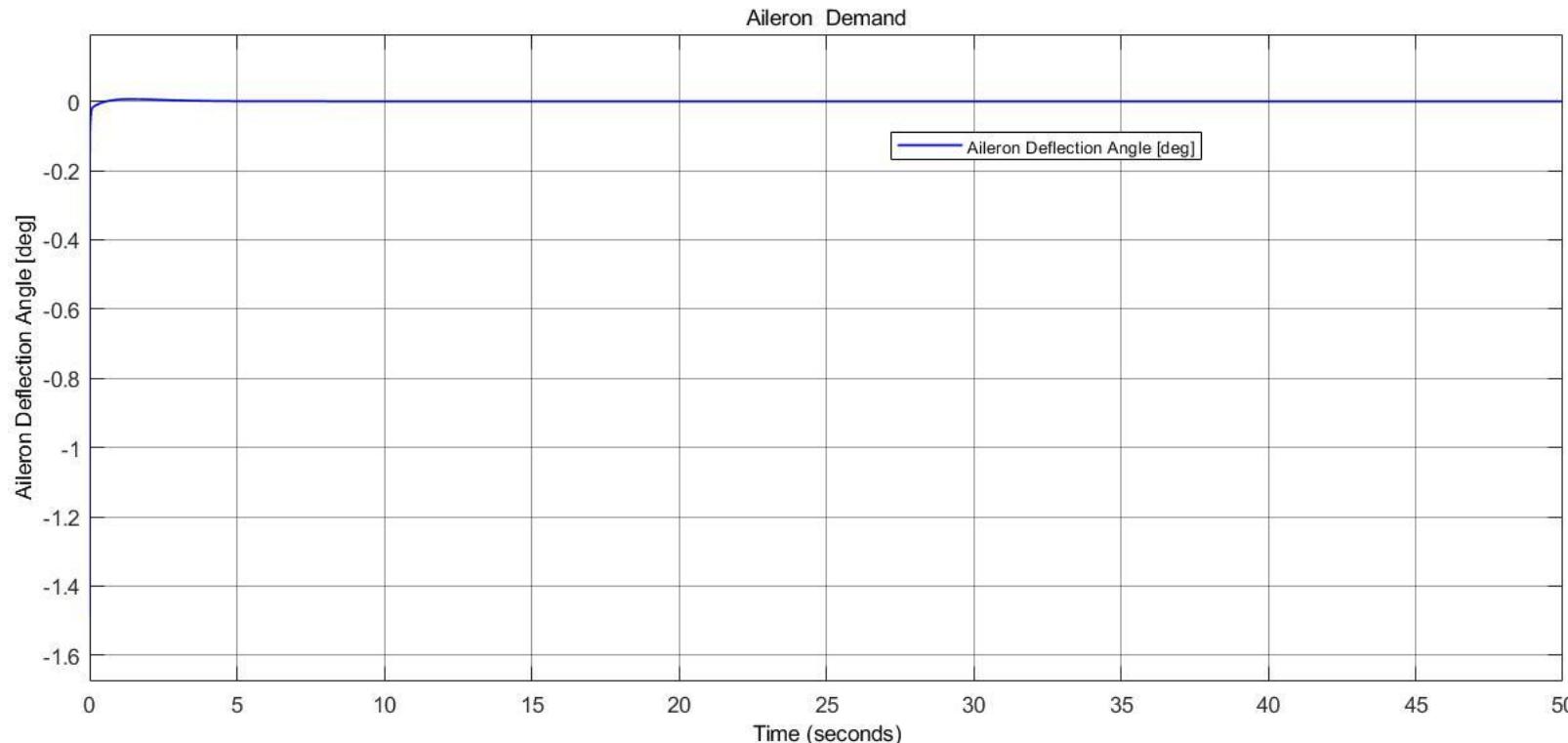
C.3. Yaw Angle Response due to +1 m/s Initial Value in Yaw Angle (ψ)



II. Linear Quadratic Control (LQR)



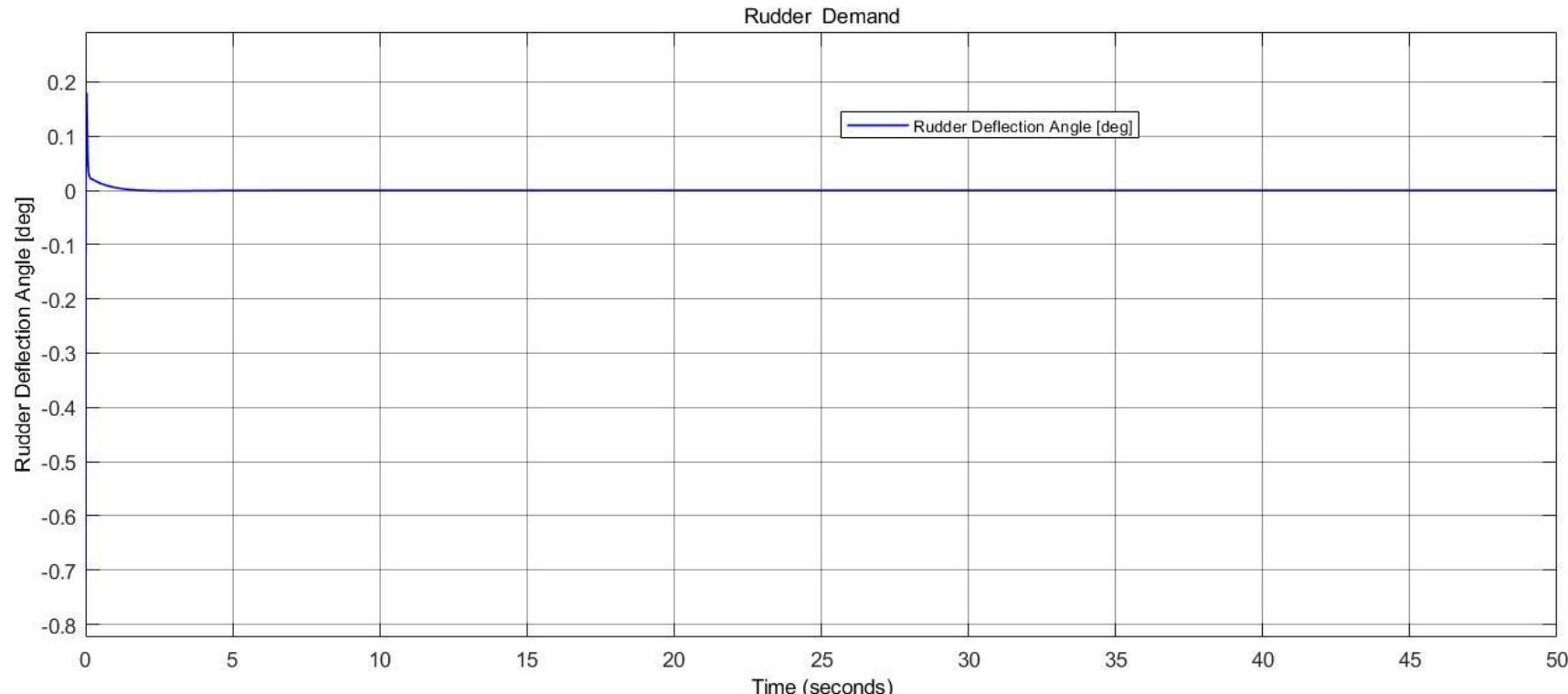
C.4. Aileron Demand due to +1 m/s Initial Value in Yaw Angle (ψ)



II. Linear Quadratic Control (LQR)



C.5. Rudder Demand due to +1 m/s Initial Value in Yaw Angle (ψ)

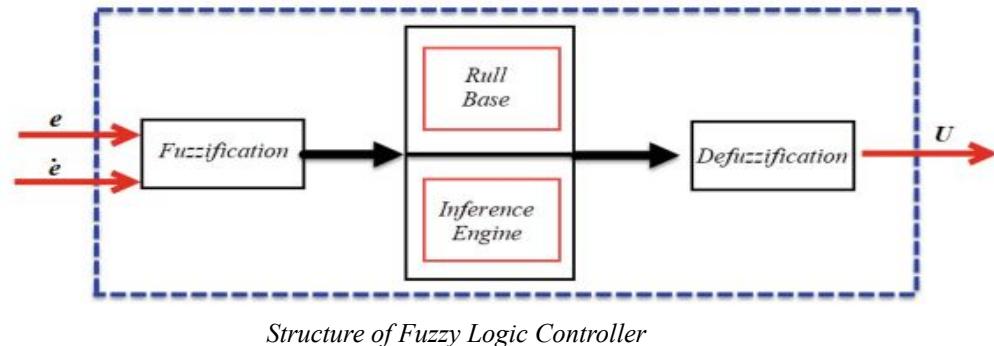


III. Fuzzy Logic Controller (FLC)

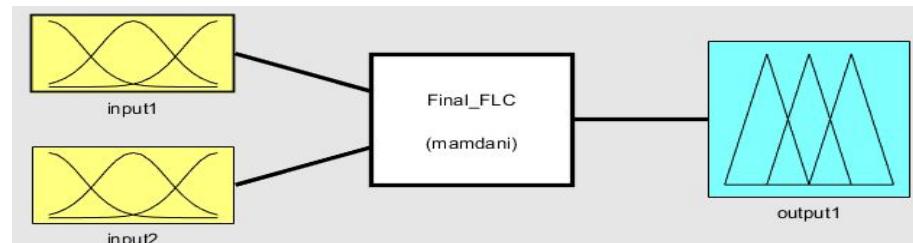
FLC Architecture:

FLC consists of 3 basic sub-groups:

- Fuzzification.
- Fuzzy inference engine (decision logic)
- Defuzzification.



Structure of Fuzzy Logic Controller



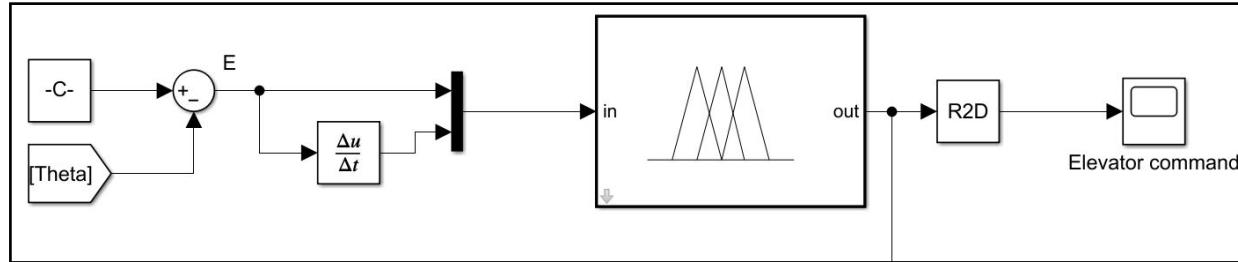
Mamdani fuzzy inference block

The most frequent approach of the fuzzy inference system, the Mamdani model.

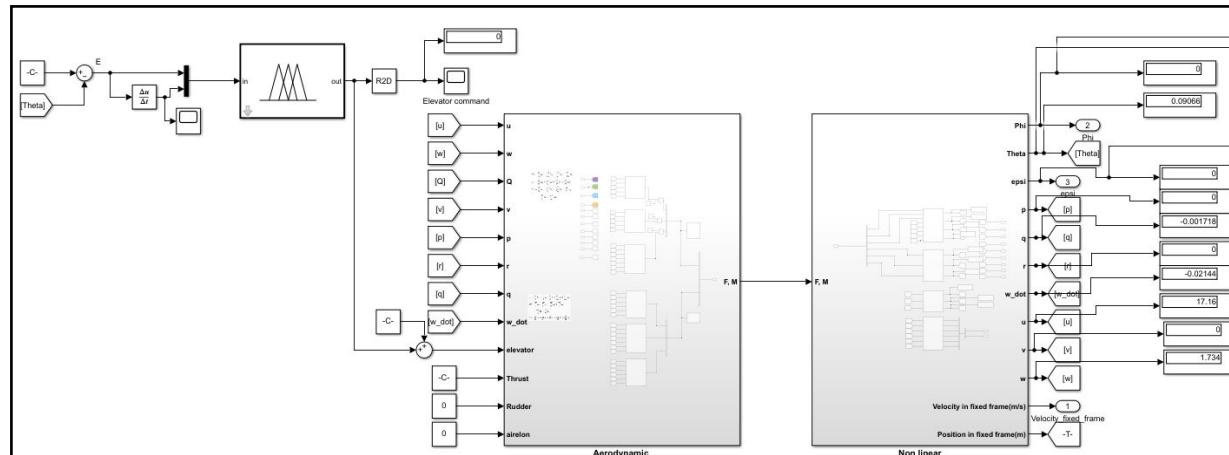
III. Fuzzy Logic Controller (FLC)



Pitch Angle control:



FLC in a feedback loop of the pitch control system.



Integrating the FLC Controller into the Simulink Model.

III. Fuzzy Logic Controller (FLC)

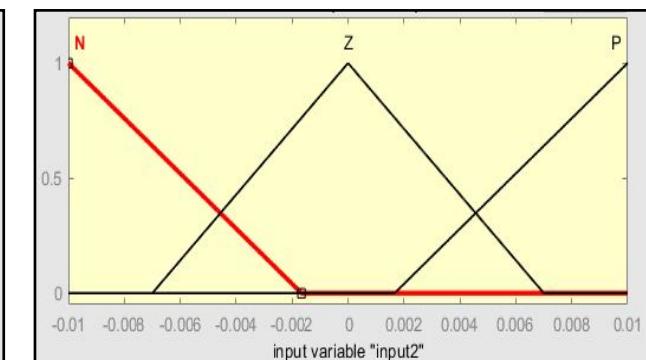
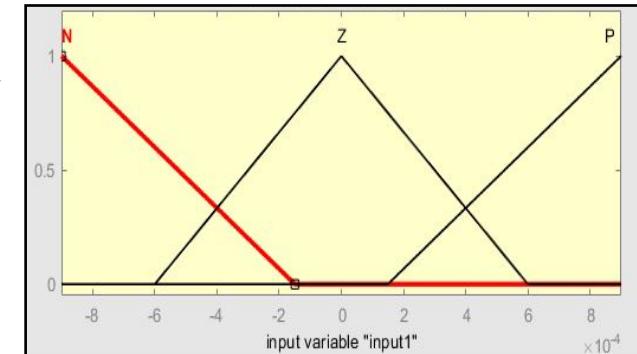
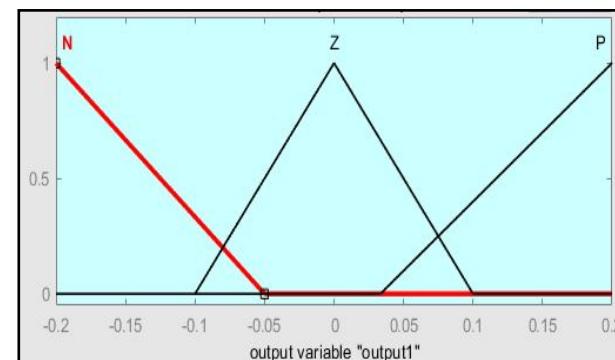
Fuzzification & Defuzzification

-Fuzzification converges the input and output signal into a number of fuzzy represented values

It consists of three types of membership function:

- Negative (N)
- Zero (Z)
- Positive (P).

-The centroid of the area is used as the defuzzification method.



III. Fuzzy Logic Controller (FLC)

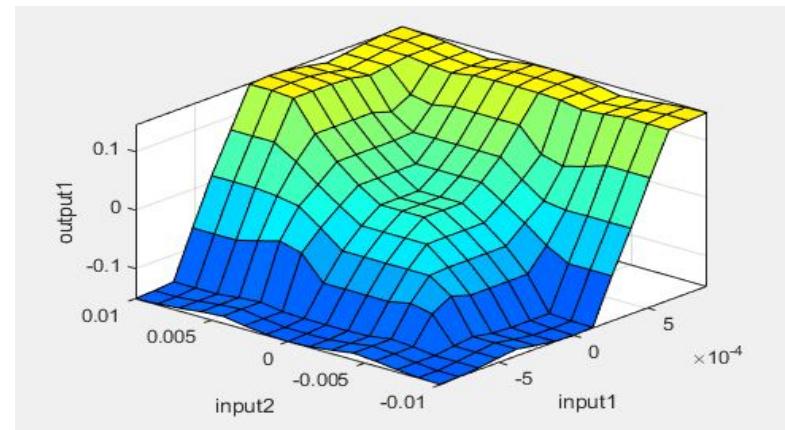
Rule Base:

- There are 2 inputs as fuzzy variables.
- Each variable has 3 MFs, then the FLC for pitch control has a total of 9 rules in the table.

	Error, e	Delta error. de	Delta u, du
1	N	N	N
2	N	Z	N
3	N	P	N
4	Z	N	N
5	Z	Z	Z
6	Z	P	P
7	P	N	P
8	P	Z	P
9	P	P	P

Rules for the fuzzy controller.

The Rule base is implemented in the form of an IF-THEN rule structure.

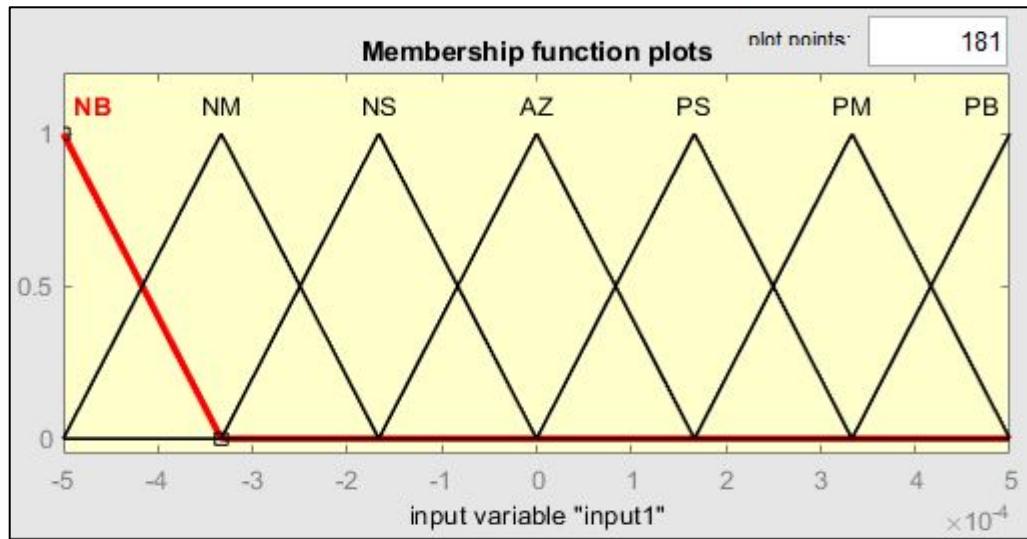


Surface look of the fuzzy logic

III. Fuzzy Logic Controller (FLC)

Rule Base:

-The more MFs used, the more complex the tuning process gets.



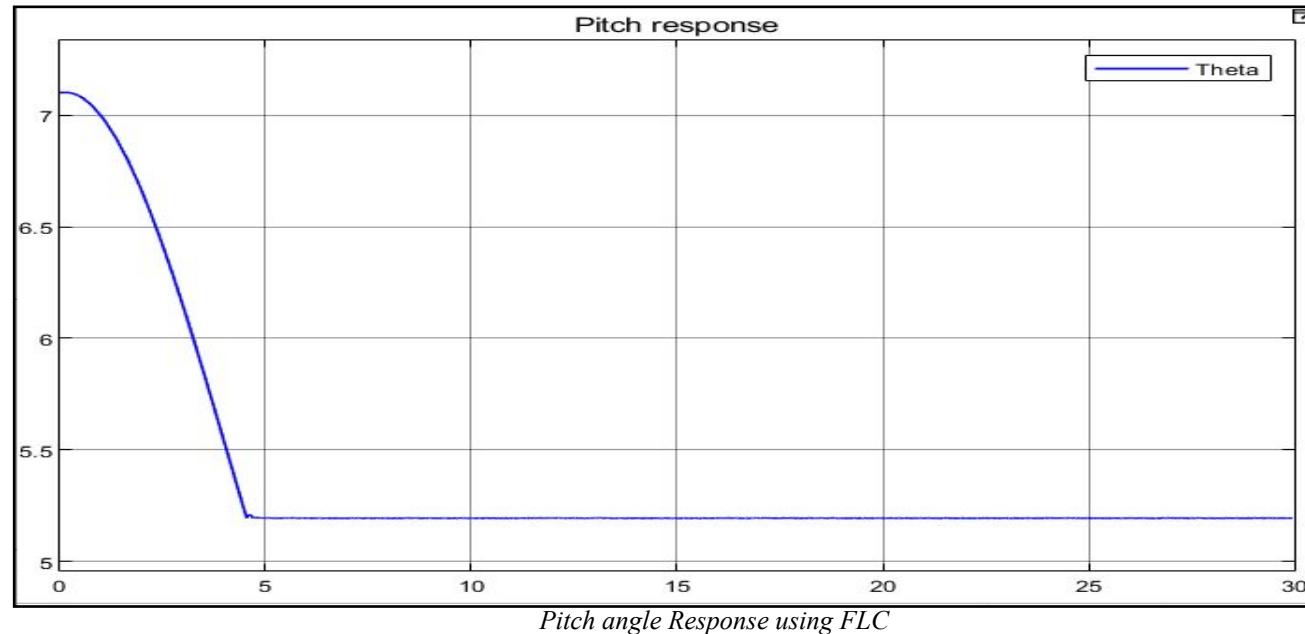
7 membership functions

E		θ						
		NB	NM	NS	AZ	PS	PM	PB
q	NB	NB	NB	NB	NM	NM	PS	PM
	NM	NB	NB	NM	NM	NS	PS	PB
	NS	NB	NB	NM	NS	AZ	PM	PB
	AZ	NB	NM	NS	AZ	PS	PM	PB
	PS	NB	NS	AZ	PS	PM	PM	PB
	PM	NB	NS	AZ	PM	PM	PB	PB
	PB	NM	NS	PS	PM	PM	PB	PB

Fuzzy rules for 7 MFs

III. Fuzzy Logic Controller (FLC)

Pitch Angle Response : After applying a 2 degree initial disturbance in the Pitch Angle



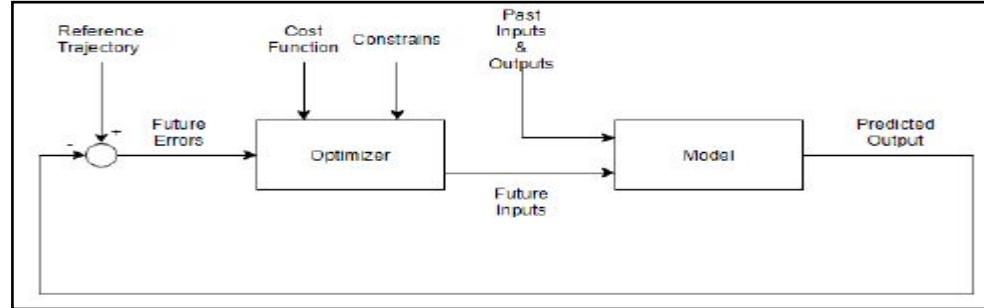
The response has zero overshoot and a settling time of 4.5s.

However, its computational cost is very high.

VI. Model Predictive Control (MPC)

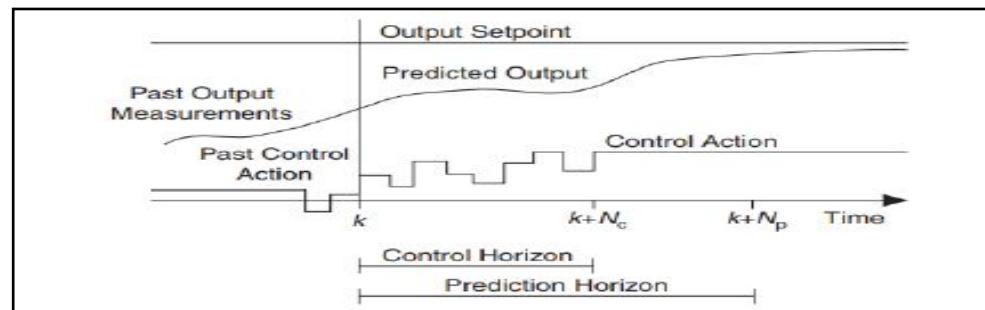


-MPC is an advanced technique that solves an online optimization problem to control a system.



MPC block diagram

-MPC modifies the control action before changing the output setpoint.



Prediction & control horizons

VI. Model Predictive Control (MPC)



Constrained MPC yields a numerical solution. As a result, the constrained MPC problem is described in the discrete state space:

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + Du_k$$

With the cost function as a linear quadratic function like:

Where,

r is the output set-point,

\hat{y} is the predicted process output,

Δu is the predicted change in control value, $\Delta u = u_k - u_{k-1}$,

Q is the output error weight matrix

R is the control weight matrix.

To acquire the optimal future control action, the problem is solved for $\partial J / \partial u = 0$.

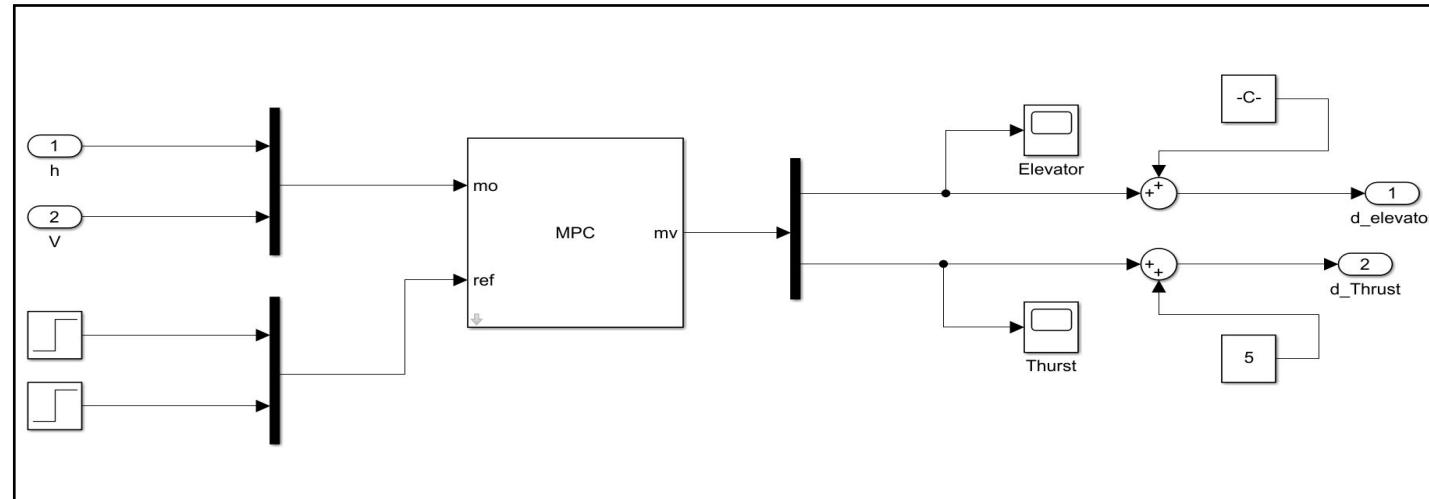
$$J = \sum_{k=0}^{N_p} (\hat{y} - r)^T Q (\hat{y} - r) + \Delta u^T R \Delta u$$

VI. Model Predictive Control (MPC)



MPC For Longitudinal Mode

- The 2 inputs are the altitude and theta.
- The output is the control action of both the elevator and Thrust.



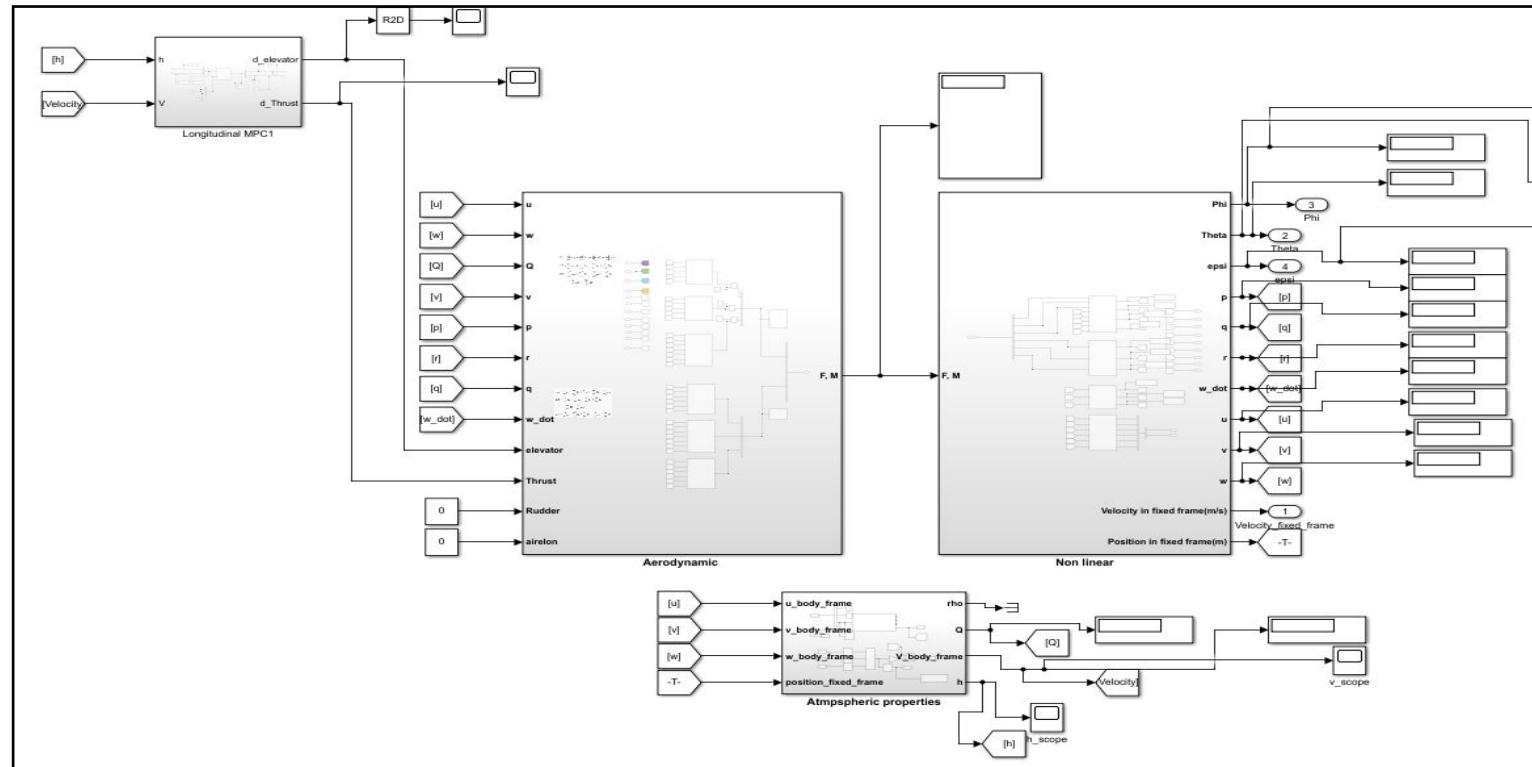
Longitudinal MPC configuration

VI. Model Predictive Control (MPC)



MPC For Longitudinal Mode

Integrating the MPC Controller into the Simulink Model

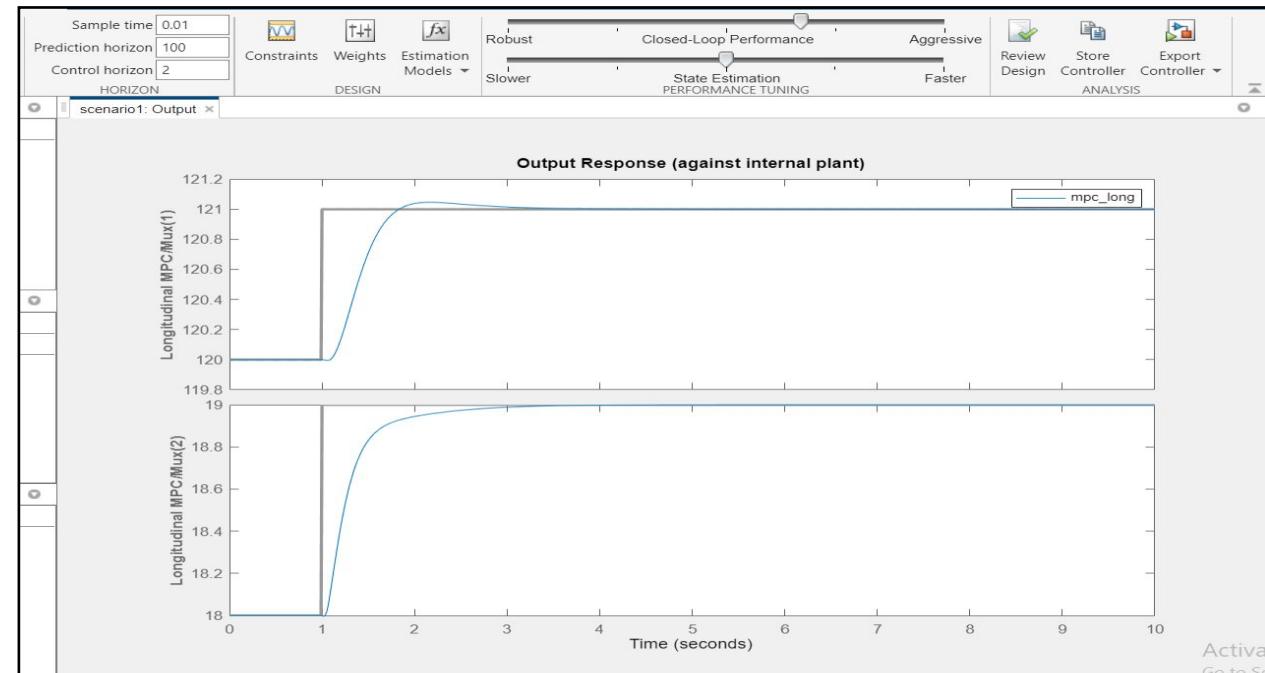


VI. Model Predictive Control (MPC)



Tuning For unconstrained MPC

- A prediction horizon of 100 was used after iterations.
- A control horizon of 2.
- Sample time is 0.01.
- Using the default weights.



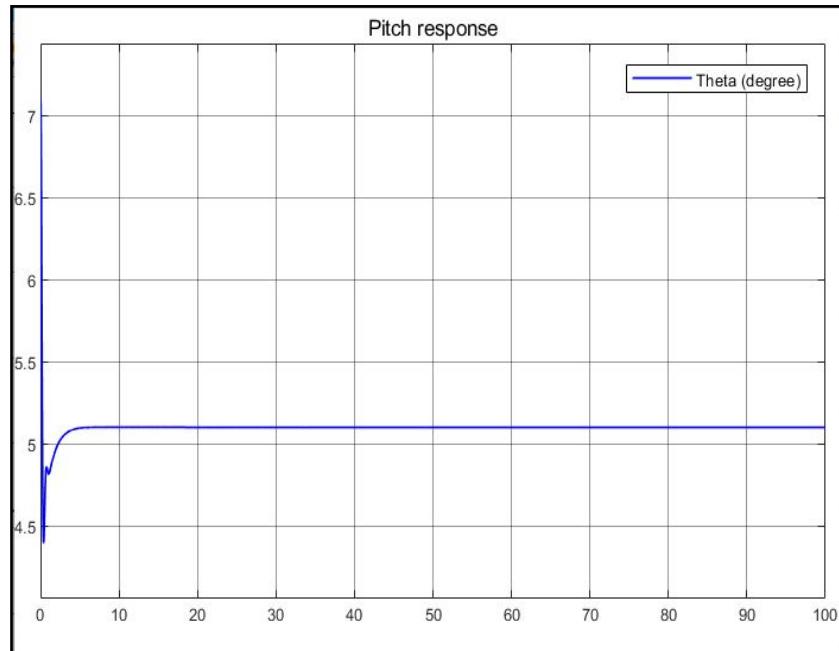
Output Response against the Commanded Response Using Both Elevator & Thrust

VI. Model Predictive Control (MPC)

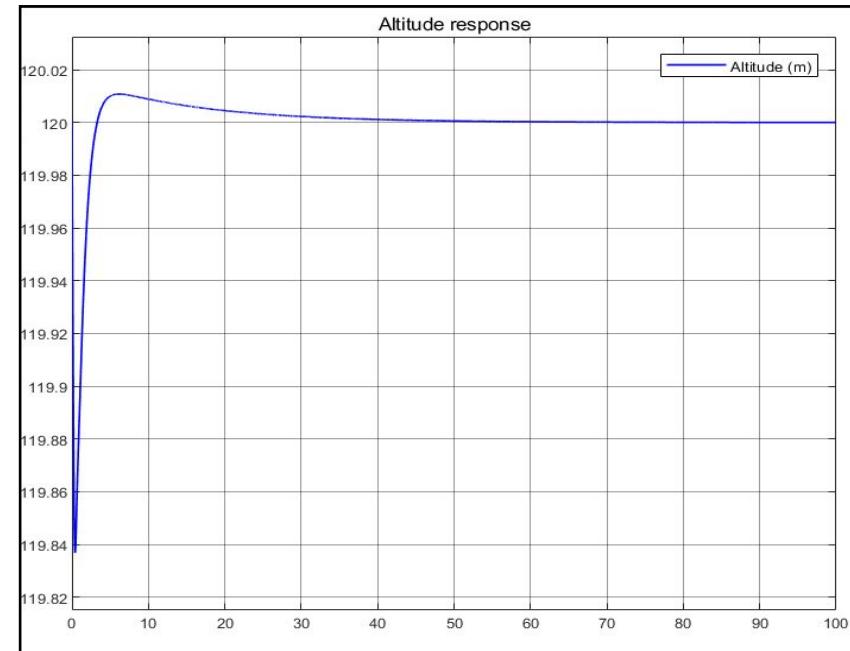


Responses of the unconstrained MPC

After applying 2-degree disturbance in the Pitch angle:



Pitch angle Response using MPC



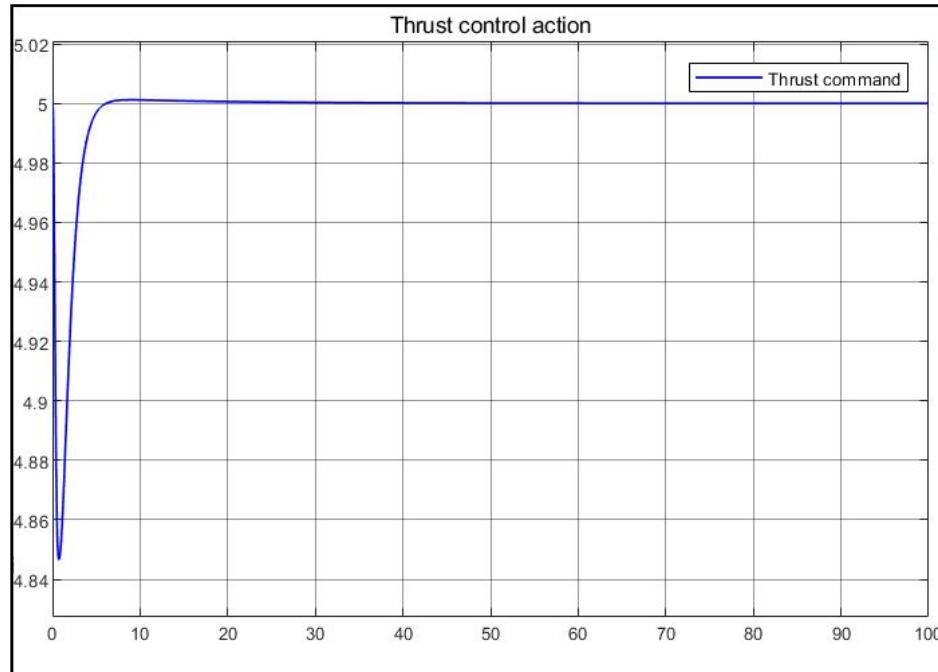
Altitude Response using MPC

VI. Model Predictive Control (MPC)

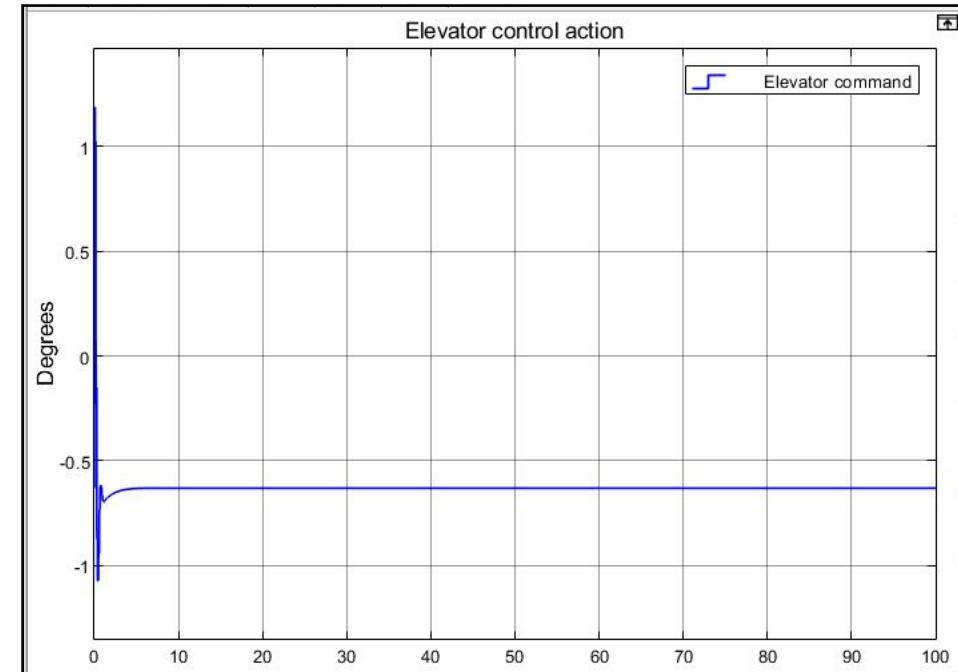


Responses of the unconstrained MPC

After applying 2-degree disturbance in the Pitch angle:



Thrust Action using MPC



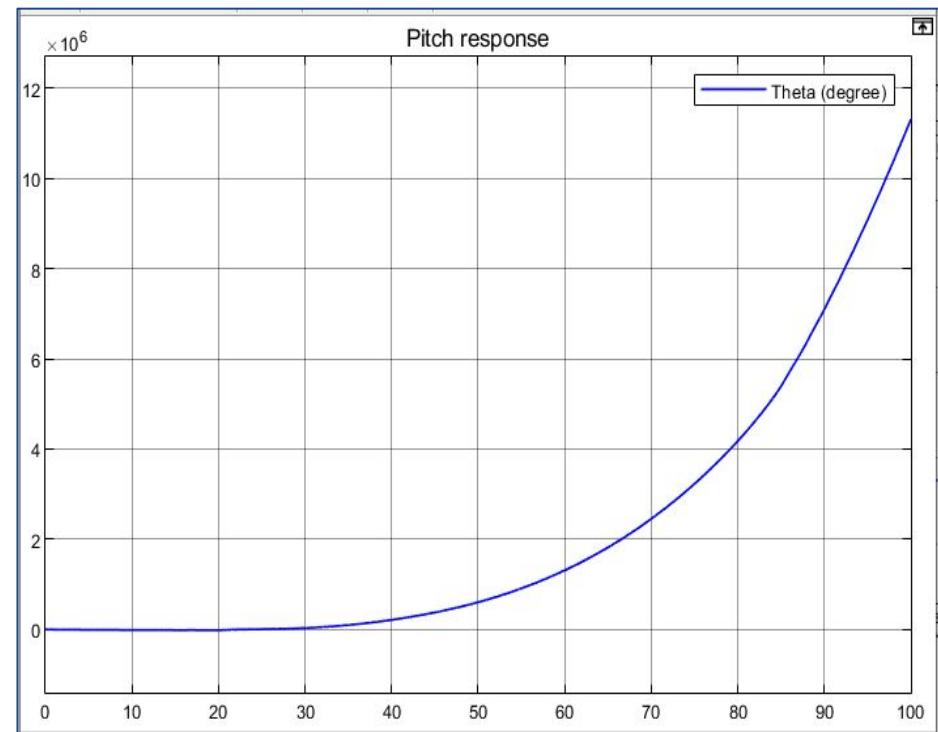
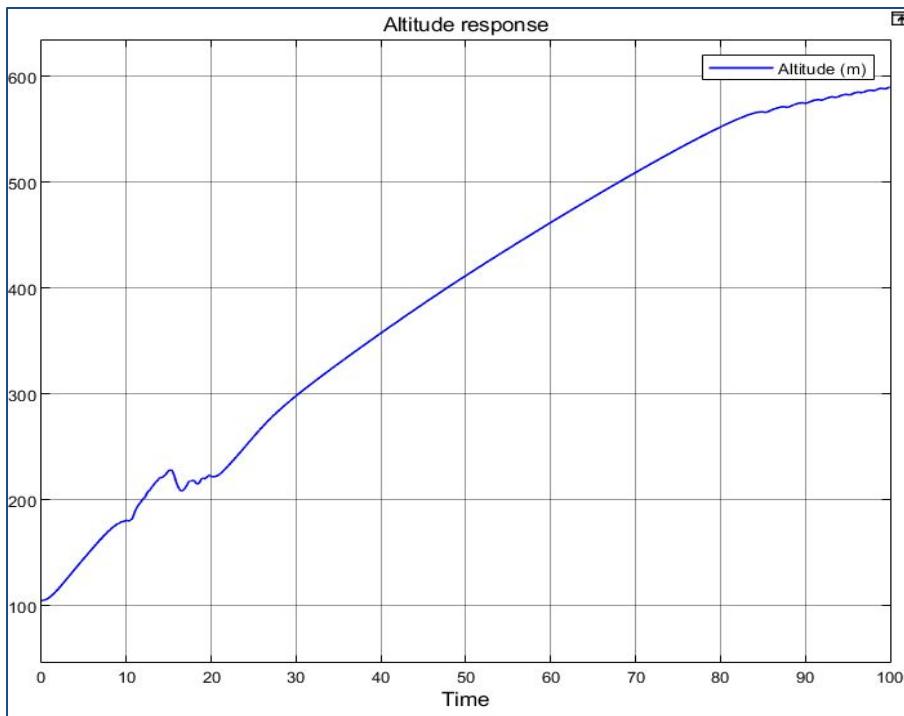
Elevator Action using MPC

VI. Model Predictive Control (MPC)



Responses of the unconstrained MPC

During climb from 105m to the ref setpoint 120m:



VI. Model Predictive Control (MPC)



Tuning For constrained MPC

The max elevator deflection was set to 0.5235(30°) nose up and -0.2967 rad (17°) nose down, while the thrust control range was 0:16 N.

Input and Output Constraints					
Channel	Type	Min	Max	RateMin	RateMax
▼ Inputs					
$u(1)$	MV	-0.26179	0.52359	-Inf	Inf
$u(2)$	MV	-5	11	-Inf	Inf
▼ Outputs					
$y(1)$	MO	-Inf	Inf		
$y(2)$	MO	-Inf	Inf		

MPC Input and output Constraints

VI. Model Predictive Control (MPC)



Tuning For constrained MPC

The following formula developed by (Alhajeri , 2021) is used to get Input rate weights values of 3.344.

$$\rho = \alpha \left(\frac{\theta}{\tau} + 0.94 \right)^{0.15} \sigma^{0.9} K^2$$

Where,

K is the steady state gain, and is set to 1

α and σ are set according to(Alhajeri , 2021) with a recommended setting of (0.84,1)

Θ is the prediction horizon which is 100

τ is the sample time which is 0.01

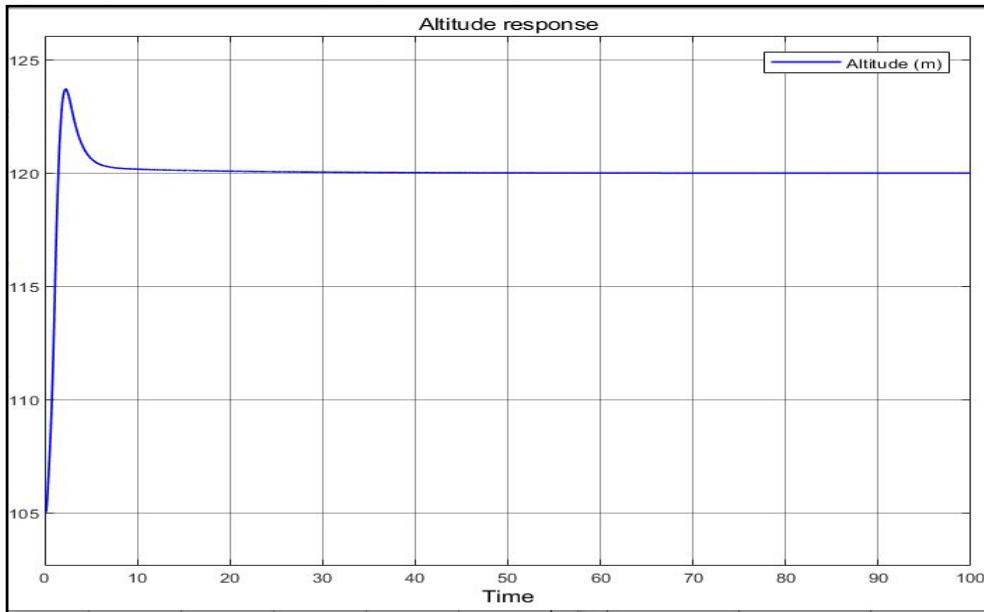
Input Weights (dimensionless)					
	Channel	Type	Weight	Rate Weight	Target
1	u(1)	MV	0	3.344	nominal
2	u(2)	MV	0	3.344	nominal

VI. Model Predictive Control (MPC)

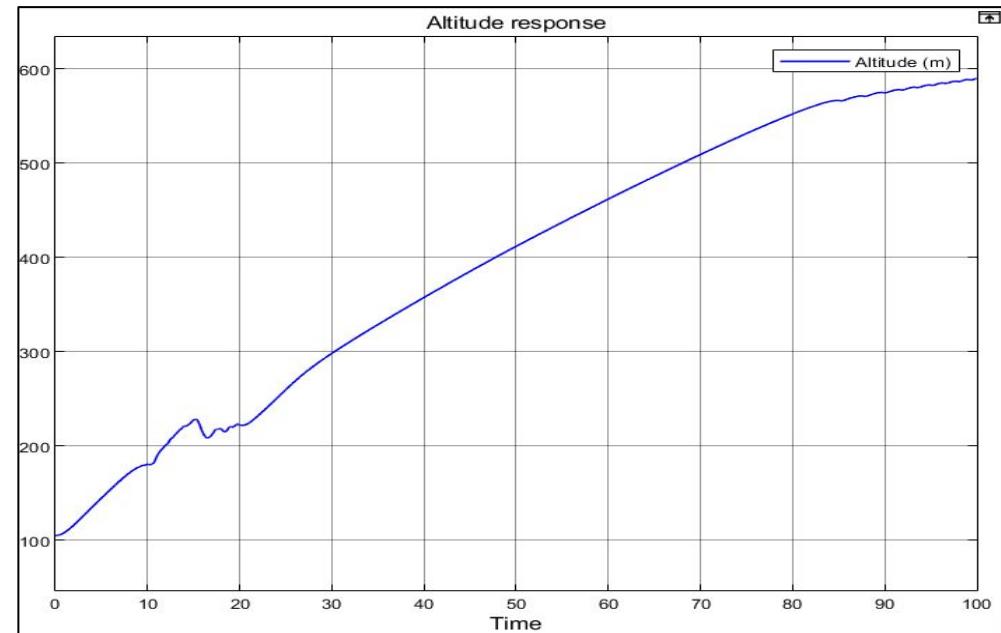


Responses of the MPC

During climb from 105m to the ref setpoint 120m:



Altitude Response using Constrained MPC



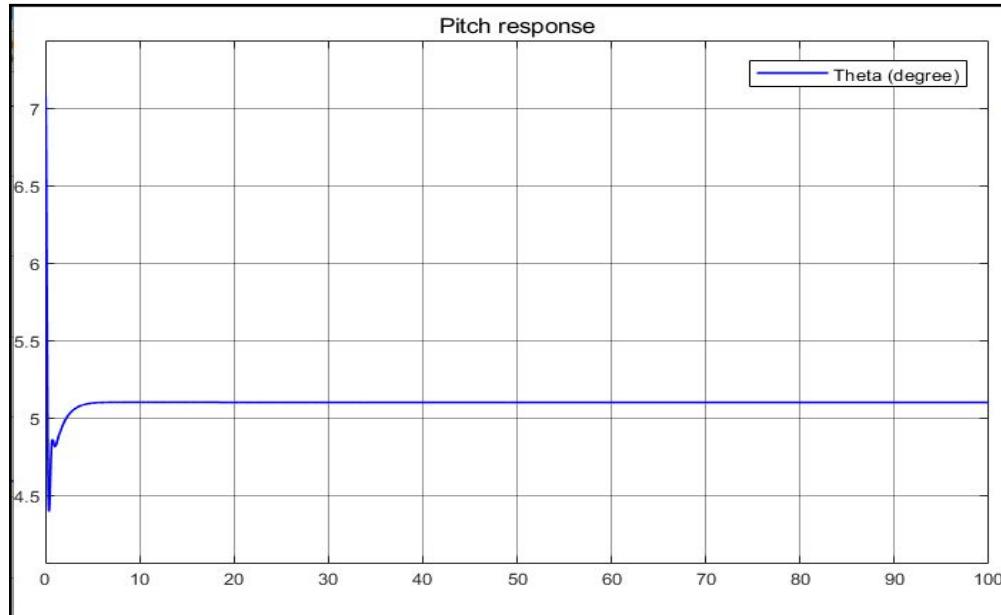
Altitude Response using UnConstrained MPC

VI. Model Predictive Control (MPC)

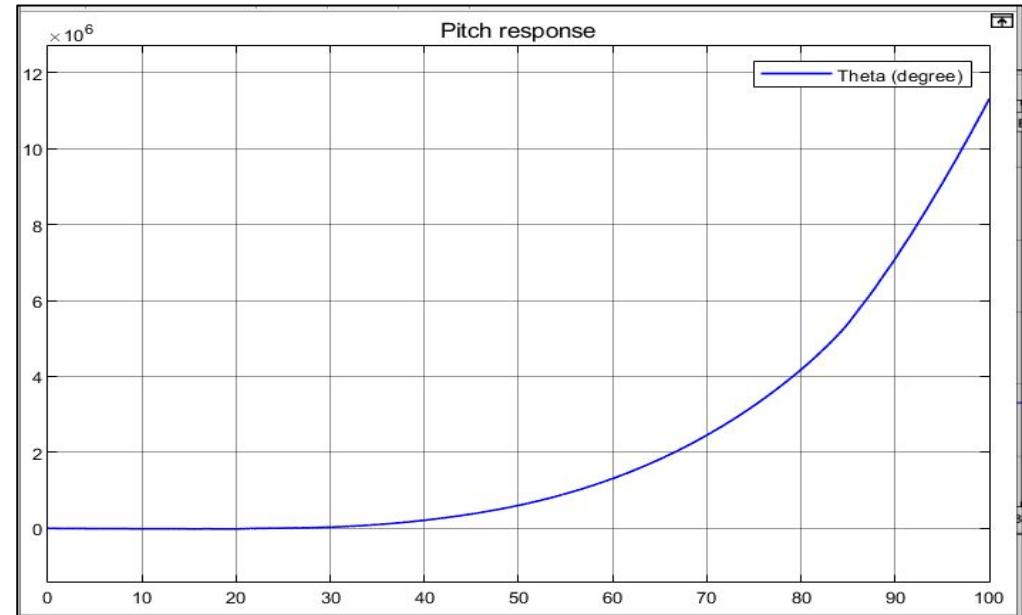


Responses of the MPC

During climb from 105m to the ref setpoint 120m:



Pitch Angle Response using Constrained MPC



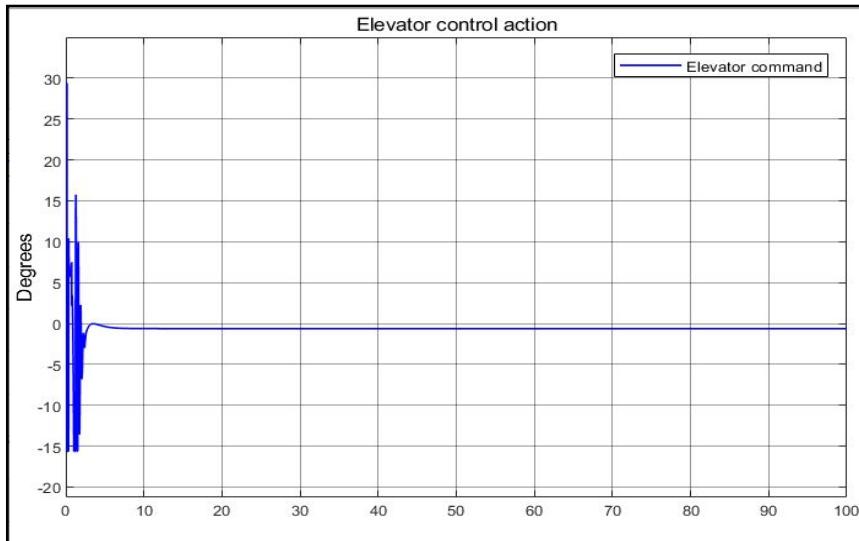
Pitch Angle Response using UnConstrained MPC

VI. Model Predictive Control (MPC)

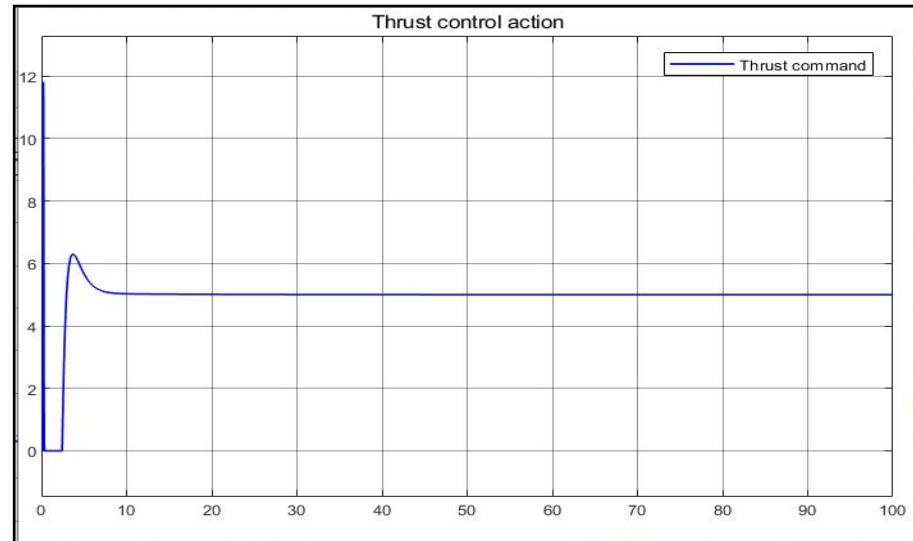


Responses of the MPC

During climb from 105m to the ref setpoint 120m:



Elevator Action using Constrained MPC



Thrust Action using Constrained MPC

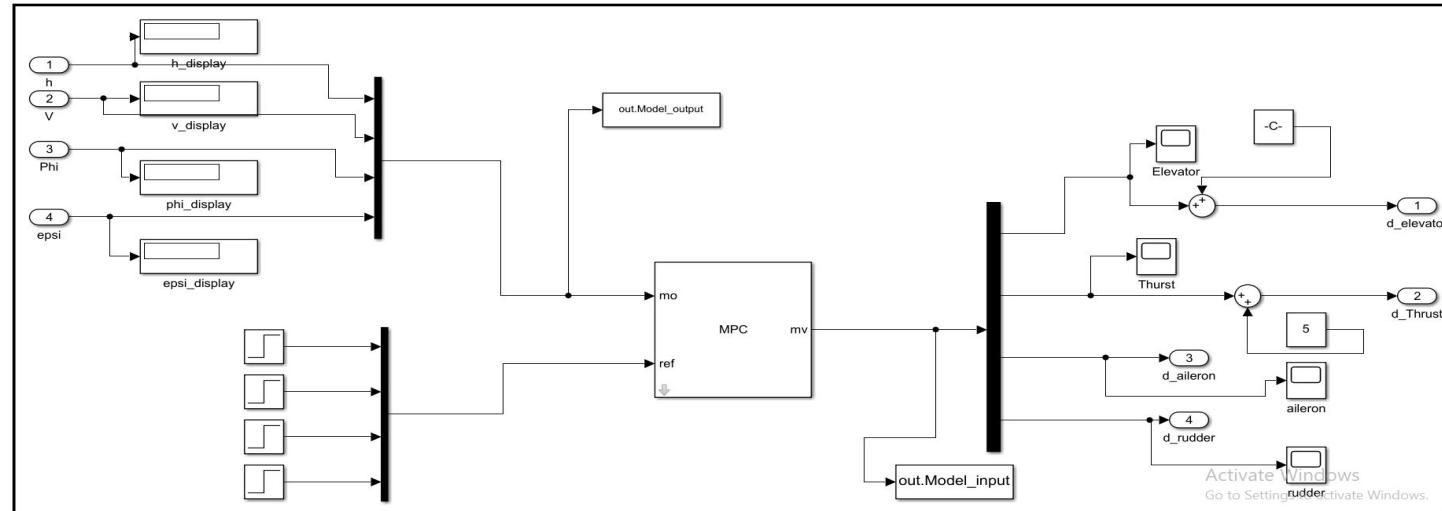
Robustness in the control surfaces for the first few seconds.

VI. Model Predictive Control (MPC)



MPC For Both Longitudinal & Lateral Modes

- The 4 inputs are the altitude, theta, phi and epsi
- The output is the 4 control action of the elevator, Thrust, Rudder and Aileron.

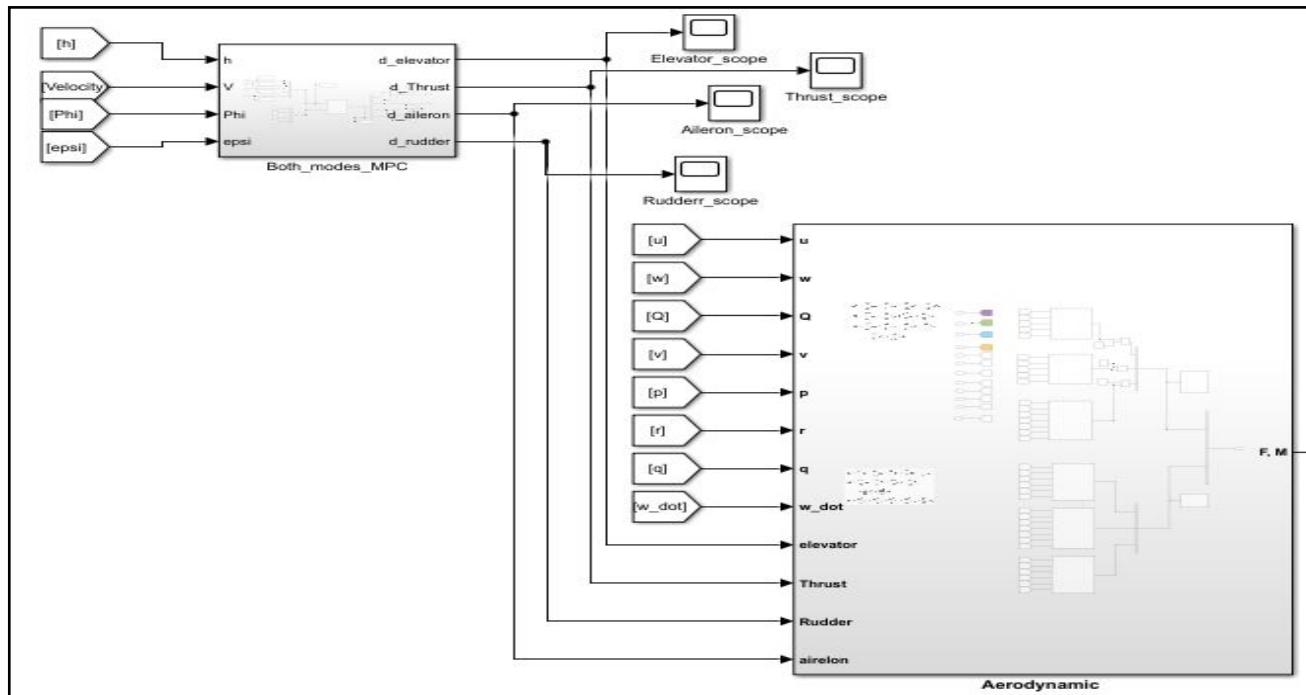


VI. Model Predictive Control (MPC)



MPC For Both Longitudinal & Lateral Modes

Integrating the MPC Controller into the Simulink Model

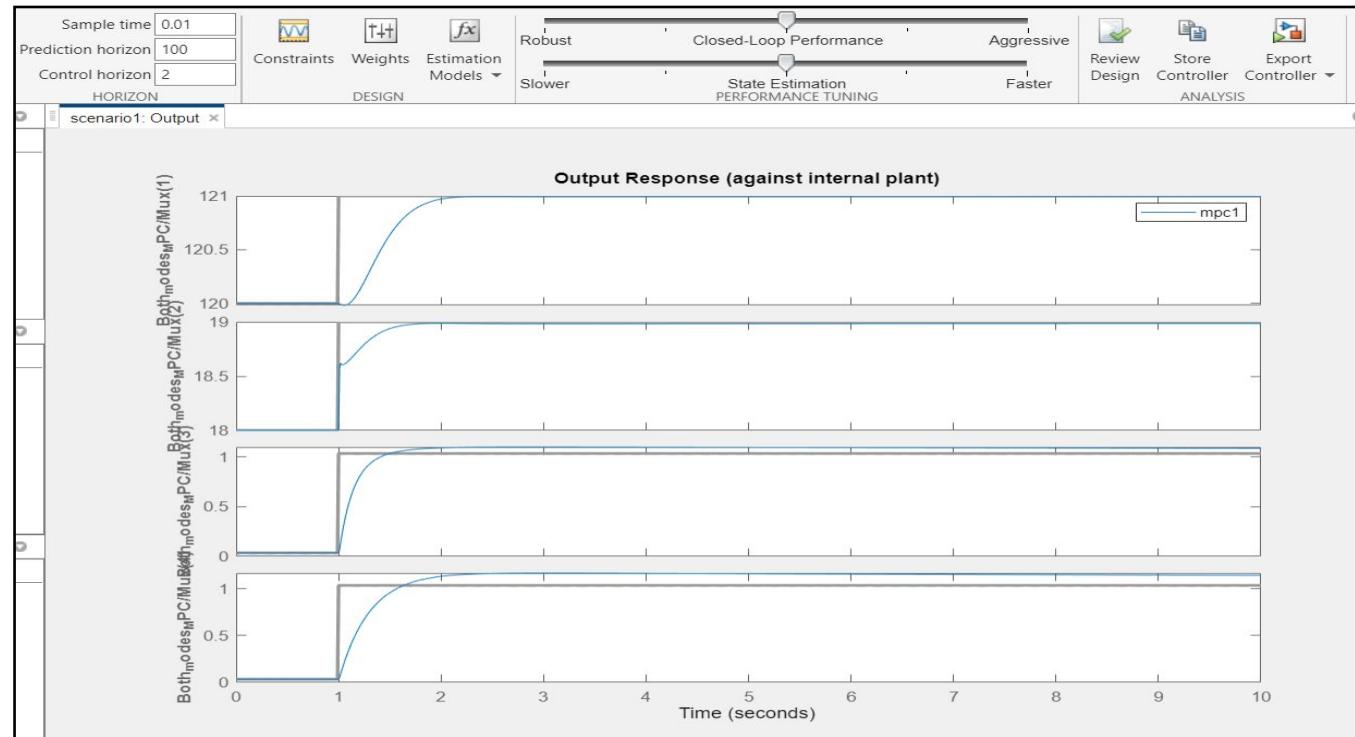


VI. Model Predictive Control (MPC)



Tuning For unconstrained MPC

- A prediction horizon of 100 was used after iterations.
- A control Horizon of 2.
- Sample time is 0.01.
- Using the default weights.

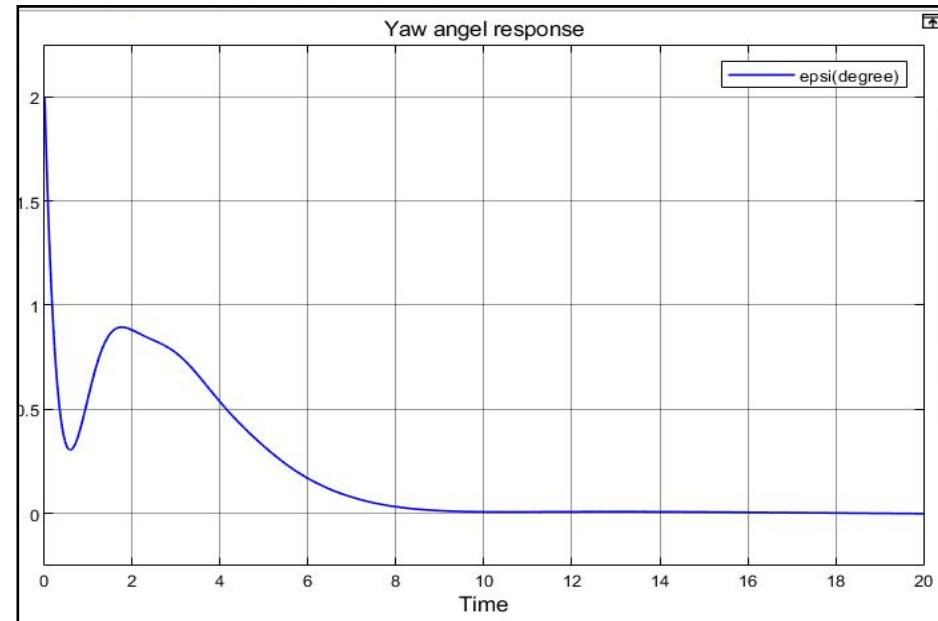
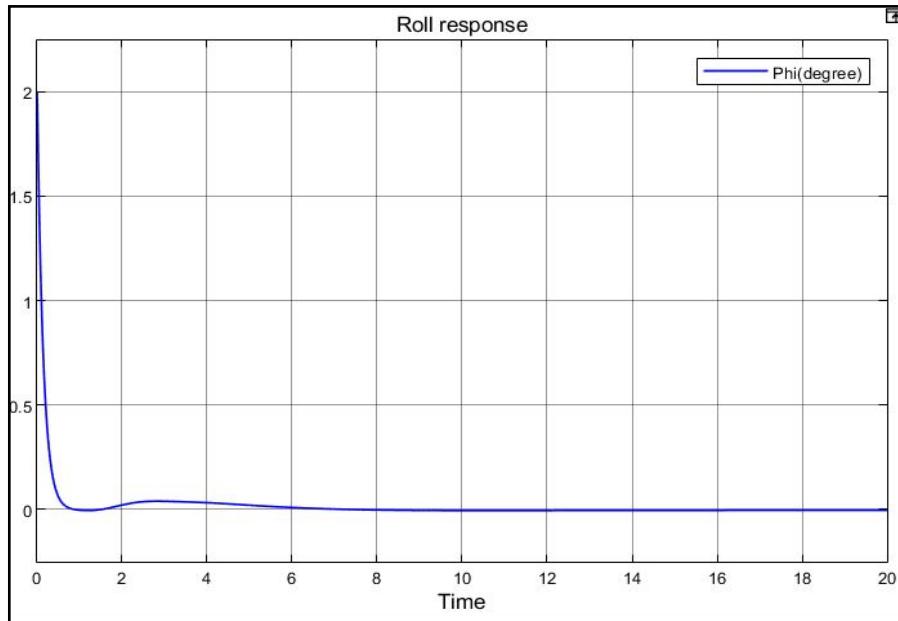


VI. Model Predictive Control (MPC)



Responses of the unconstrained MPC

After applying 2-degree disturbance in the Roll and Yaw angles:



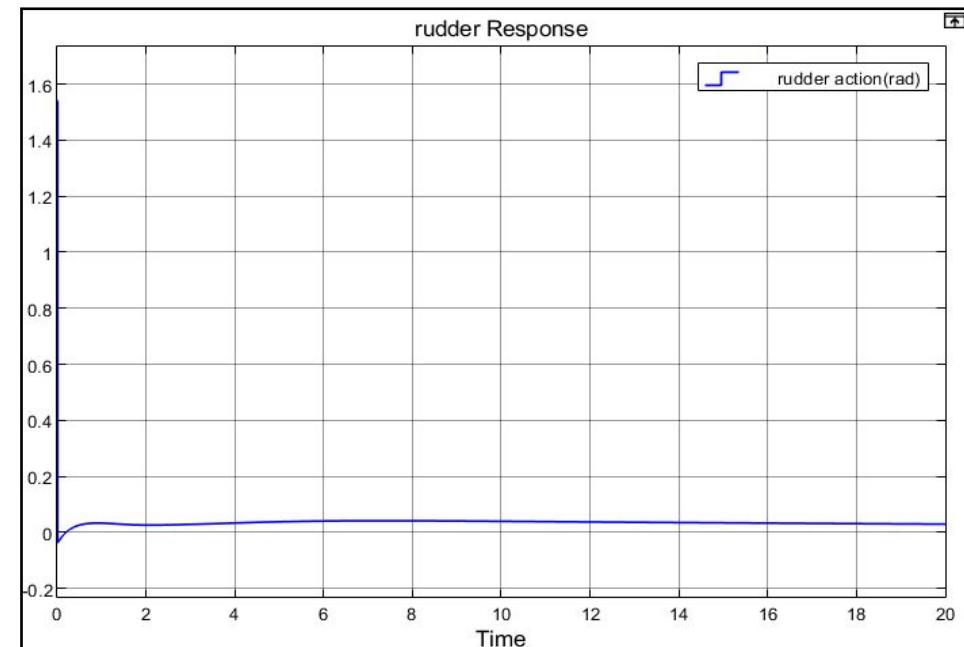
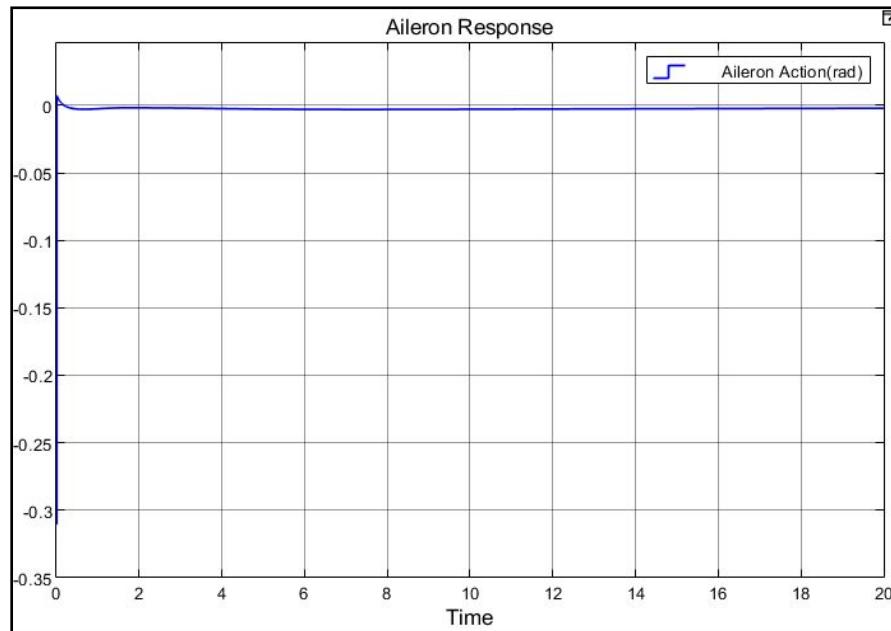
VI. Model Predictive Control (MPC)



Responses of the unconstrained MPC

After applying 2-degree disturbance in the Roll and Yaw angles:

:

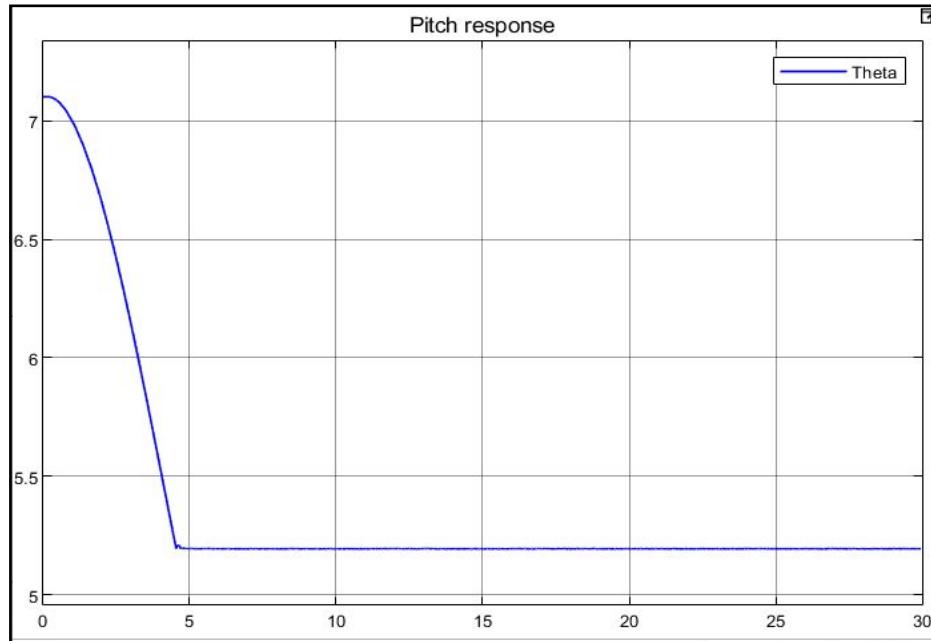


Comparisons between Controllers (Fixed Wing UAV)

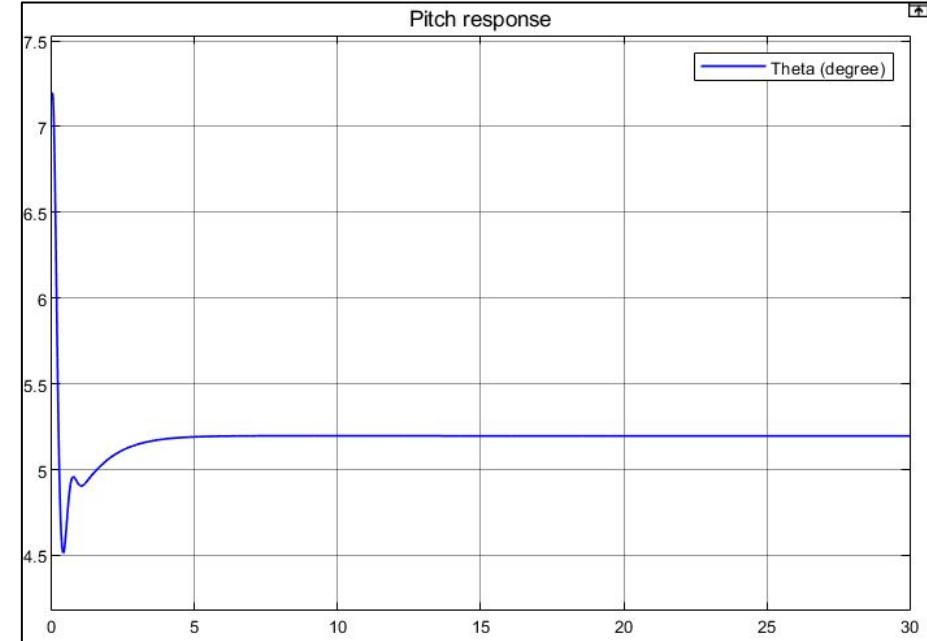
Comparison between FLC & MPC



After applying a 2 degree initial disturbance in the Pitch Angle.



Pitch angle Response using FLC

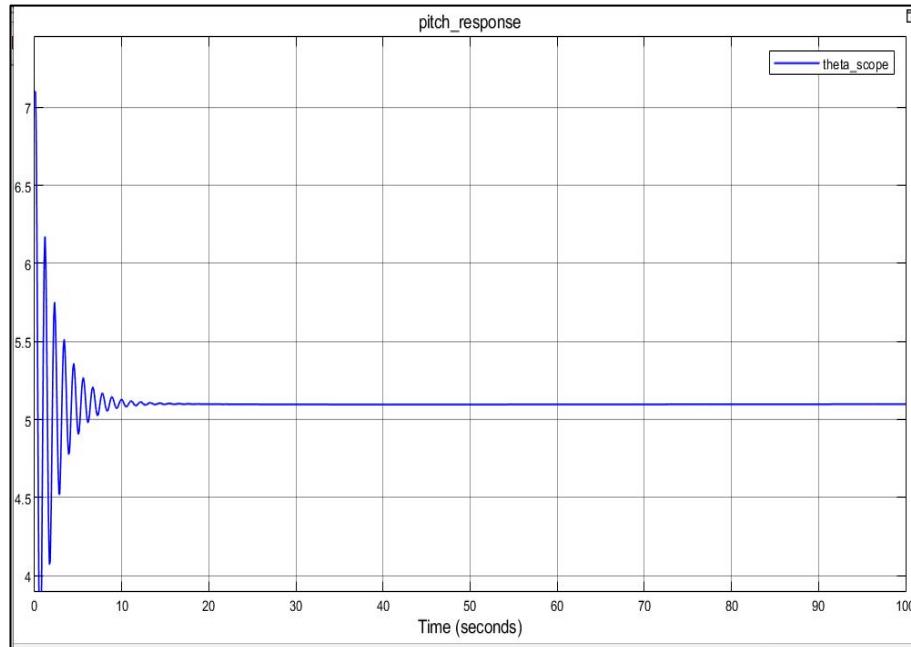


Pitch angle Response using MPC

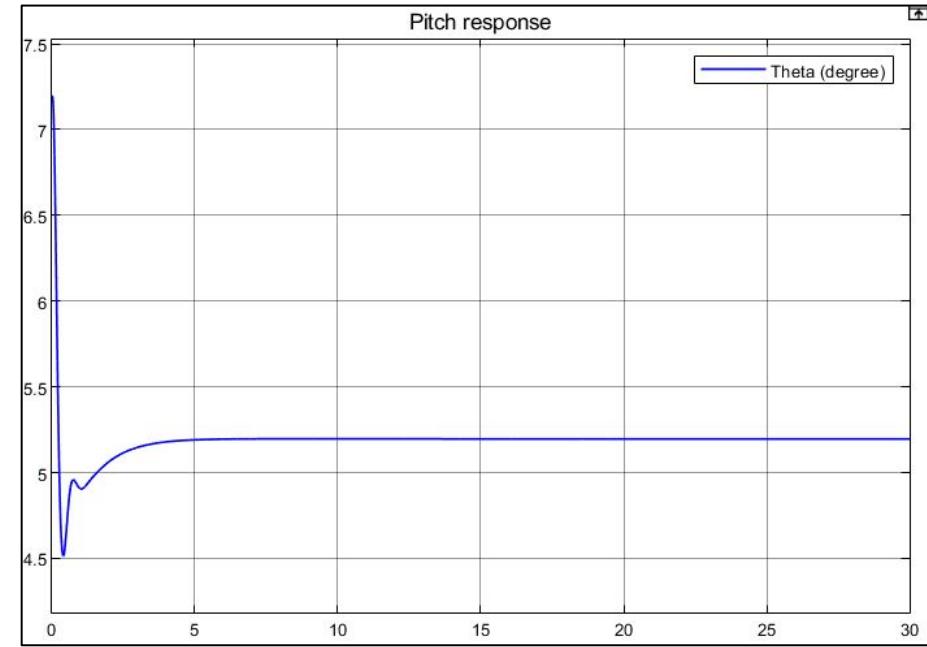
Comparison between TECS & MPC



Pitch angle Response due to 2-degree disturbance in the pitch angle.



Pitch angle Response using TECS



Pitch angle Response using MPC

Comparison between TECS & MPC



Pitch angle Response specifications due to 2-degree disturbance in the pitch angle

```
RiseTime: 0  
TransientTime: 7.5855  
SettlingTime: 20.2314  
SettlingMin: 4.7848  
SettlingMax: 24.3021  
Overshoot: 376.5126  
Undershoot: 0  
Peak: 24.3021  
PeakTime: 0.7000
```

Pitch angle Response using TECS

```
RiseTime: 0  
TransientTime: 3.1513  
SettlingTime: 2.2302  
SettlingMin: 4.3955  
SettlingMax: 7.1000  
Overshoot: 39.2161  
Undershoot: 0  
Peak: 7.1000  
PeakTime: 0.0100
```

Pitch angle Response using MPC

Conclusion: Choosing the controller for the VTOL- Fixed Wing



MPC is selected as the controller for the VTOL Fixed Wing, since:

- In the longitudinal mode, the response specifications are way better than TECS (with the GA gains).
- MPC is a MIMO controller which makes it easier to control both longitudinal and lateral modes at the same time.
- Although the FLC algorithm is the simplest among them all, its computational power is higher, and as the input parameters number increases, the harder the FLC tuning process.

So, MPC has proven to be the better controller in our work.

5. Autopilot Controller Design and Implementation (Quadrotor)

I. Nonlinear MPC



NMPC Setup and Tuning

NMPC setup requires

- Prediction model state function
- Prediction model analytical Jacobian
- Nominal control target (hover)
- Prediction & control horizon
- Time step
- ODE solver
- Control input constraints & their rates
- Cost function
- OVs & MVs weights

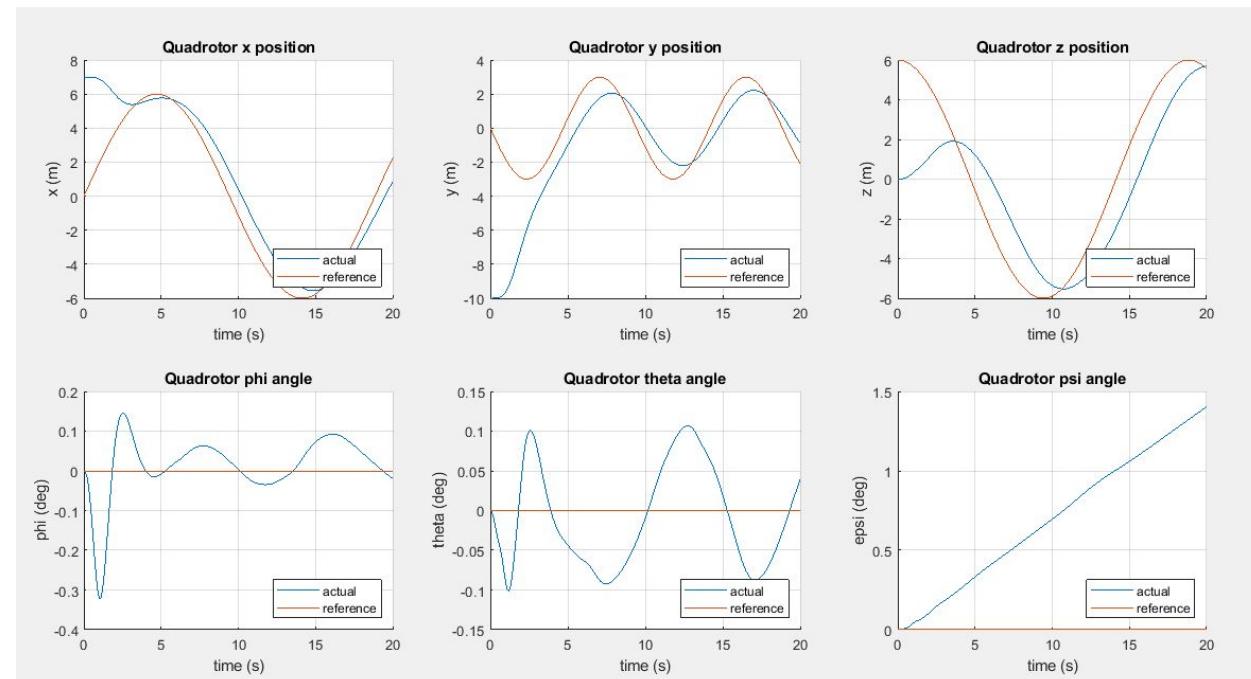
I. Nonlinear MPC



NMPC Responses Case I

Takes the 3 positions & 3 angles only - following a reference trajectory

Delay of approx. 1.33s



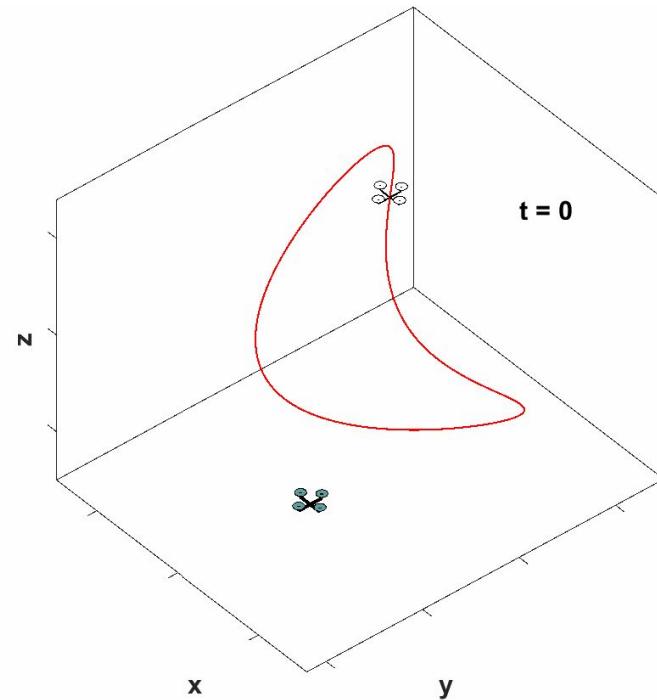
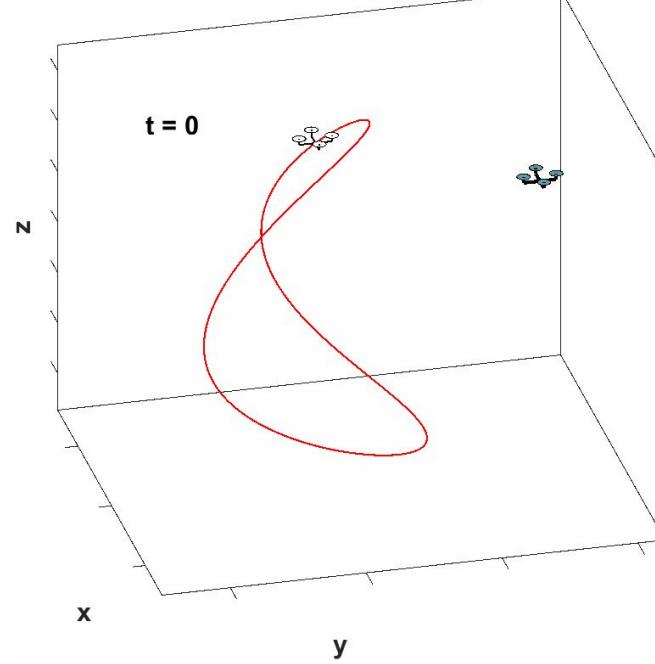
I. Nonlinear MPC



NMPC Responses Case I

Takes the 3 positions & 3 angles only - following a reference trajectory

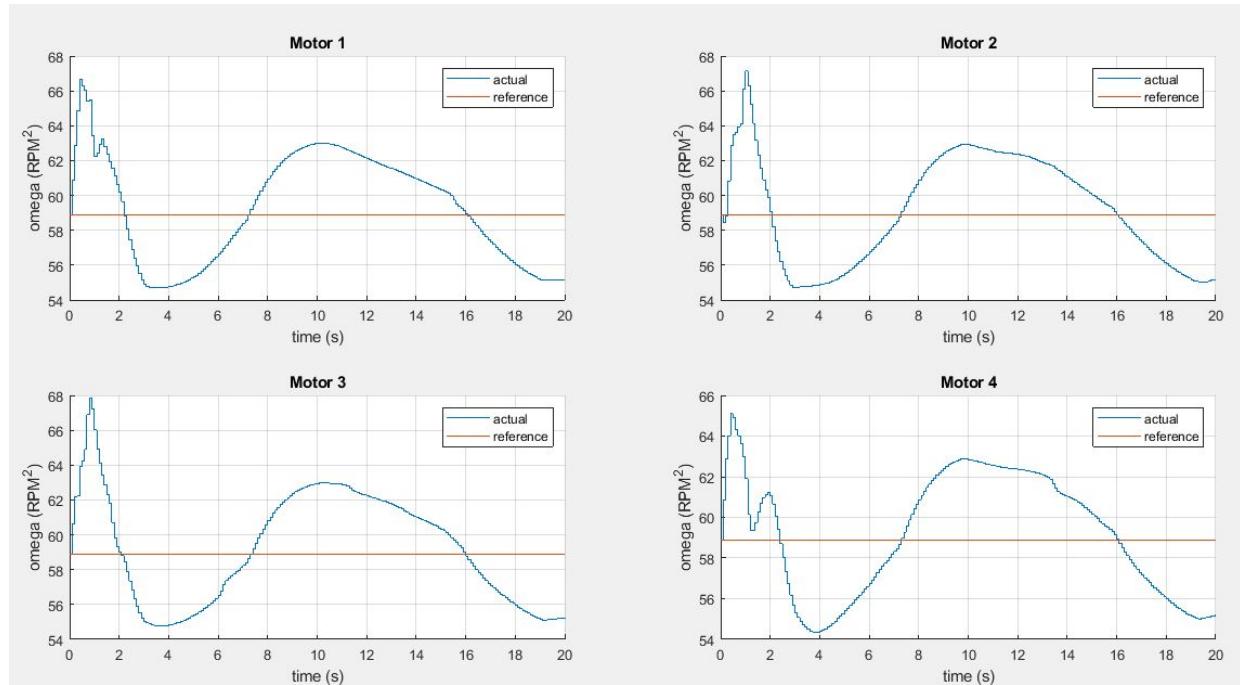
Delay of approx. 1.33s



I. Nonlinear MPC

NMPC Responses Case I

Takes the 3 positions & 3 angles only - following a reference trajectory



I. Nonlinear MPC

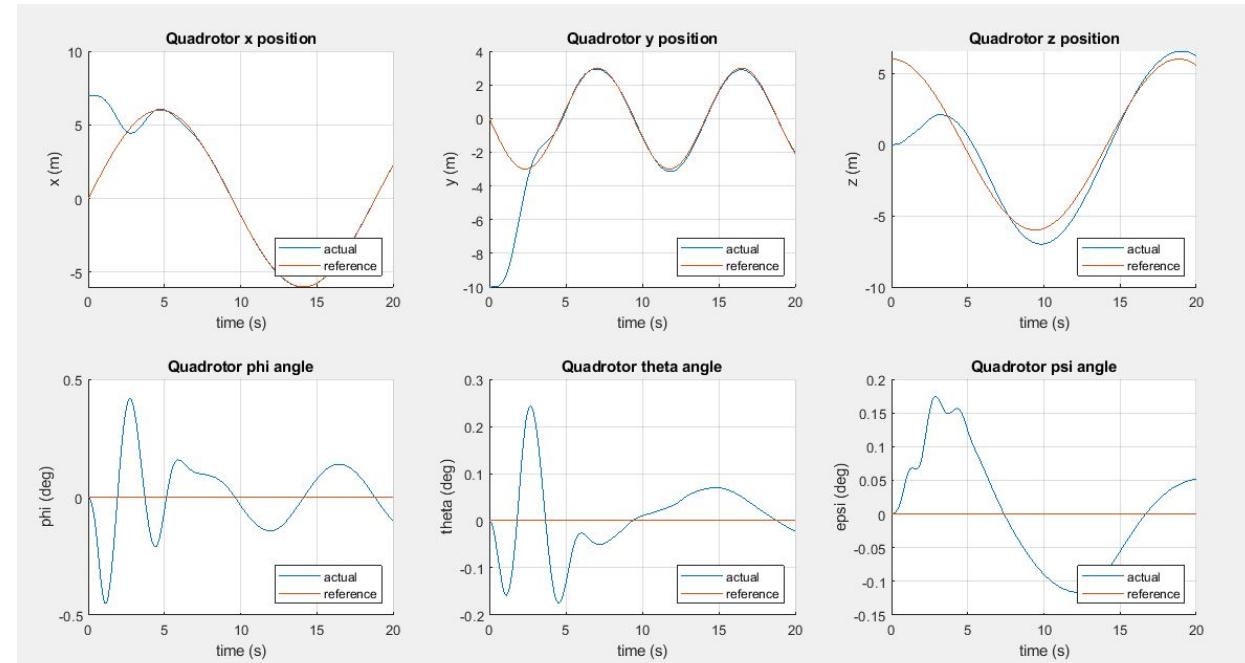


NMPC Responses Case I

Takes the 3 positions, 3 angles, & their rates - following a reference trajectory

Z overshoot of 19.32%

Y overshoot of 4.67%



I. Nonlinear MPC

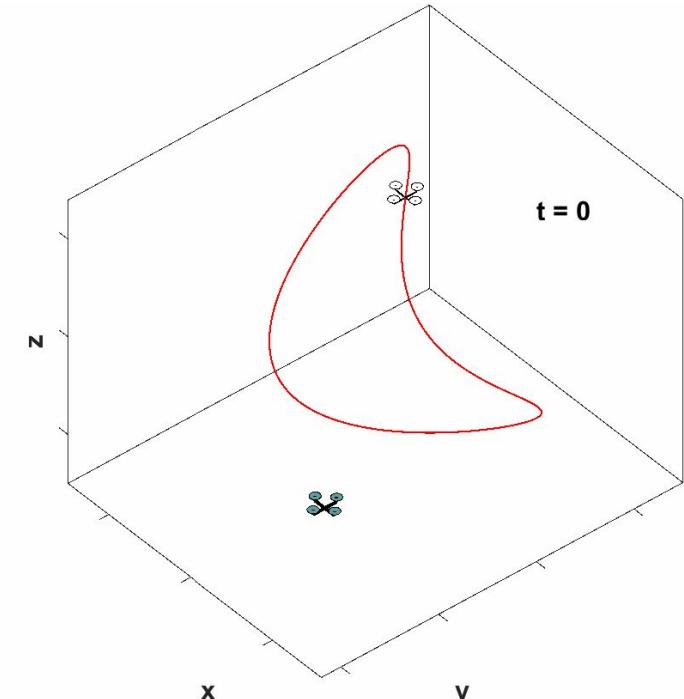
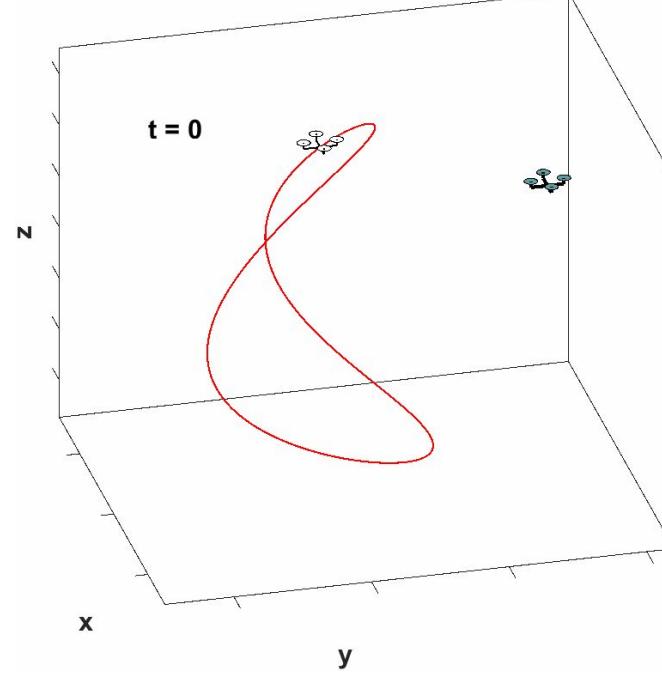


NMPC Responses Case I

Takes the 3 positions, 3 angles, & their rates - following a reference trajectory

Z overshoot of 19.32%

Y overshoot of 4.67%

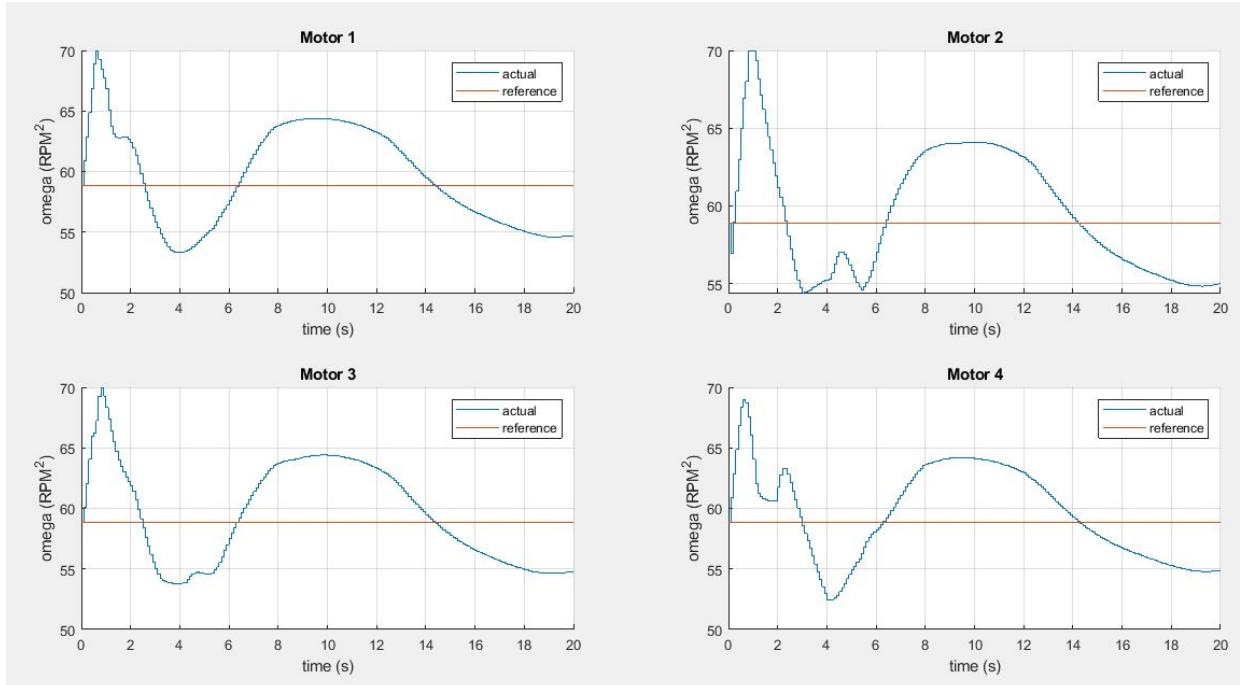


I. Nonlinear MPC



NMPC Responses Case I

Takes the 3 positions & 3 angles only - following a reference trajectory



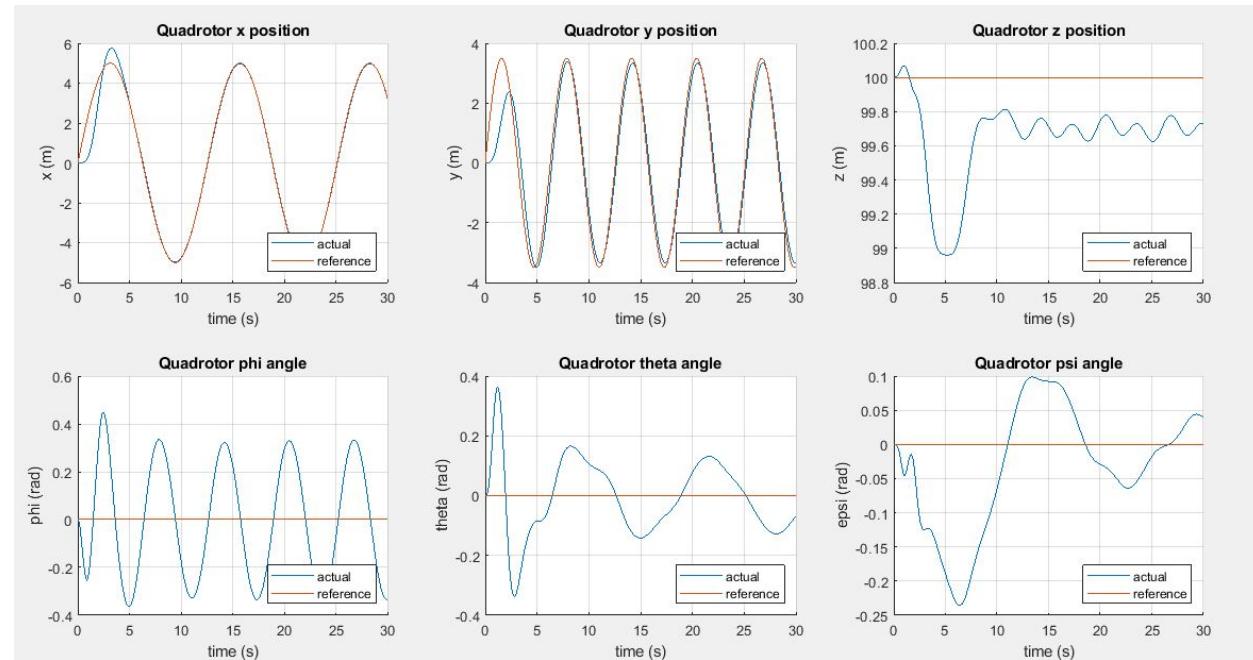
I. Nonlinear MPC

NMPC Following a Reference Trajectory

Following an infinity shaped reference trajectory

X overshoot of 15.02%

Z steady state error of 0.3 m (0.3%),

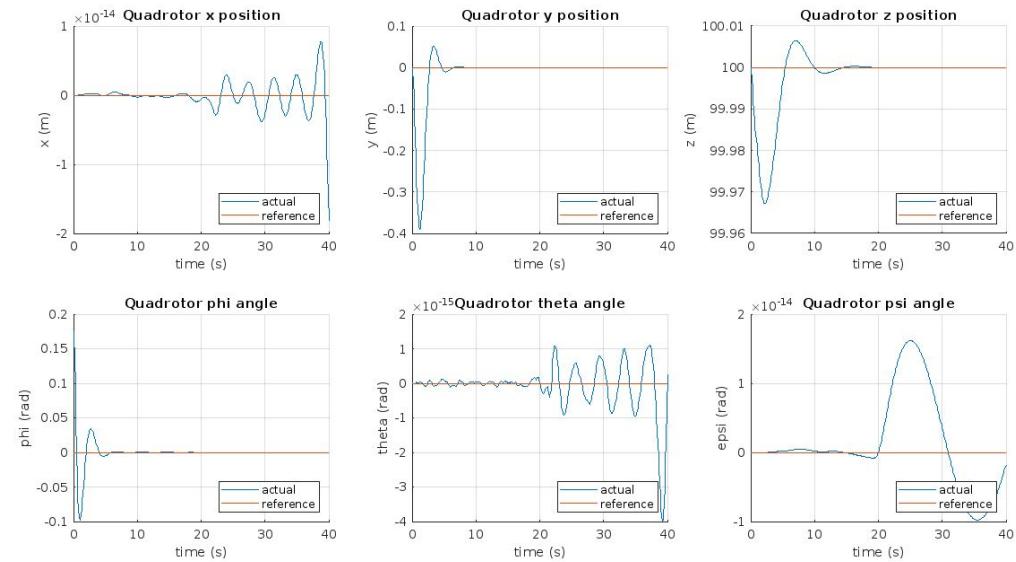
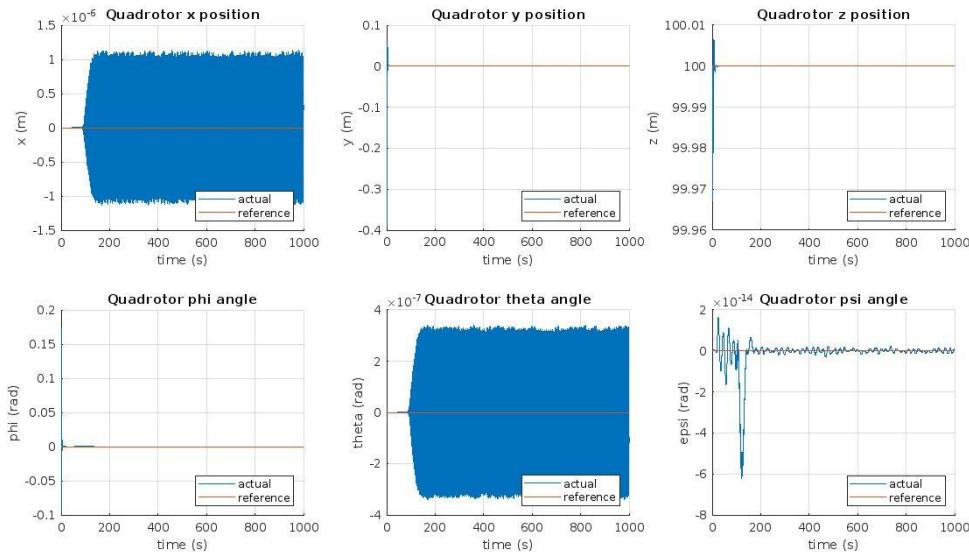


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the phi angle while hovering at 100 m for 40 s & 1000 s.

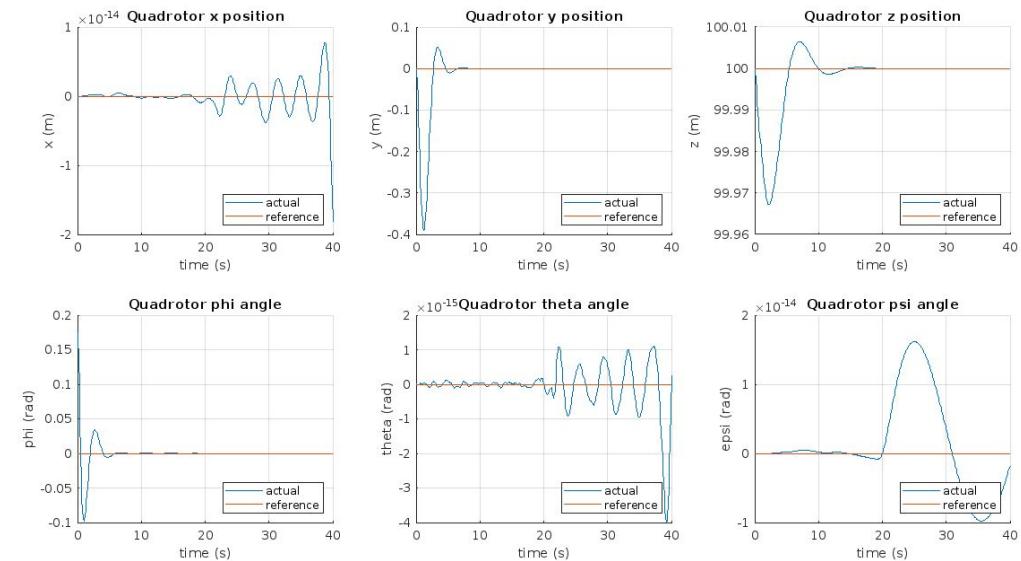
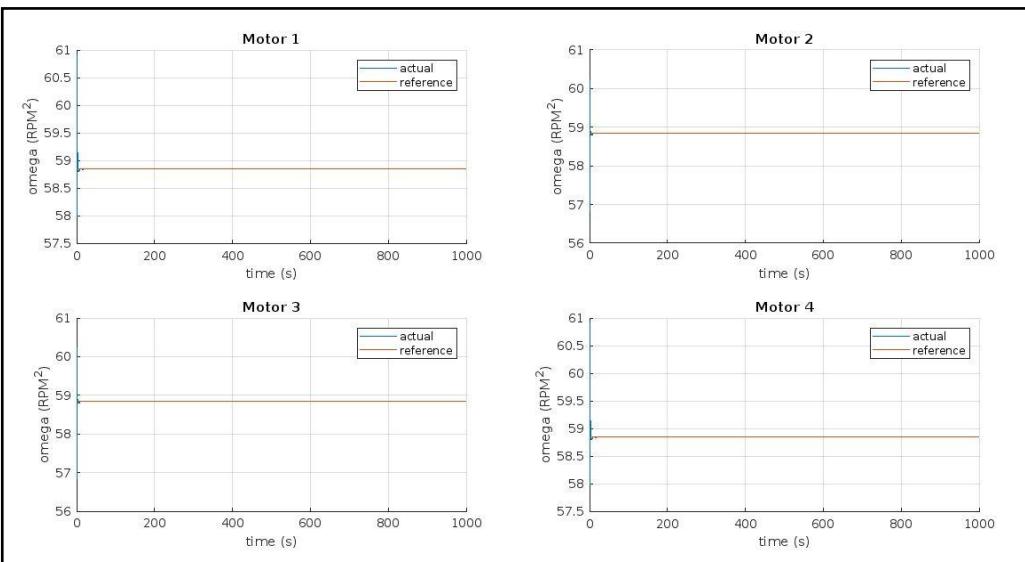


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the phi angle while hovering at 100 m for 40 s & 1000 s.

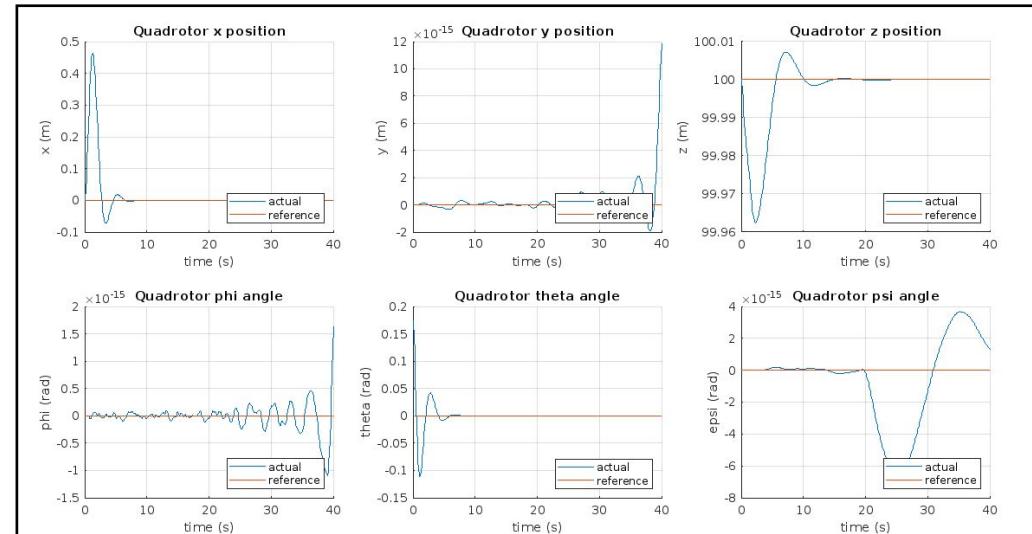
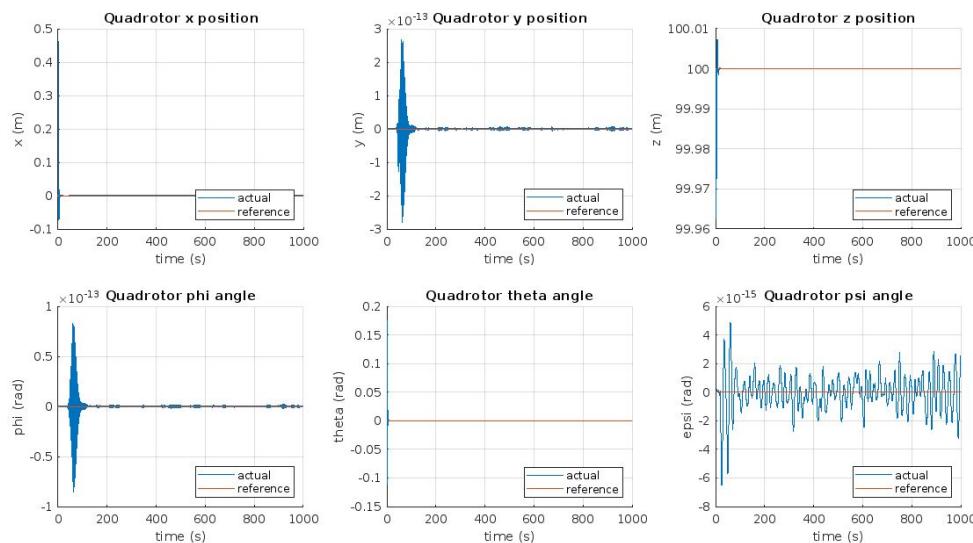


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the theta angle while hovering at 100 m for 40 s & 1000 s.

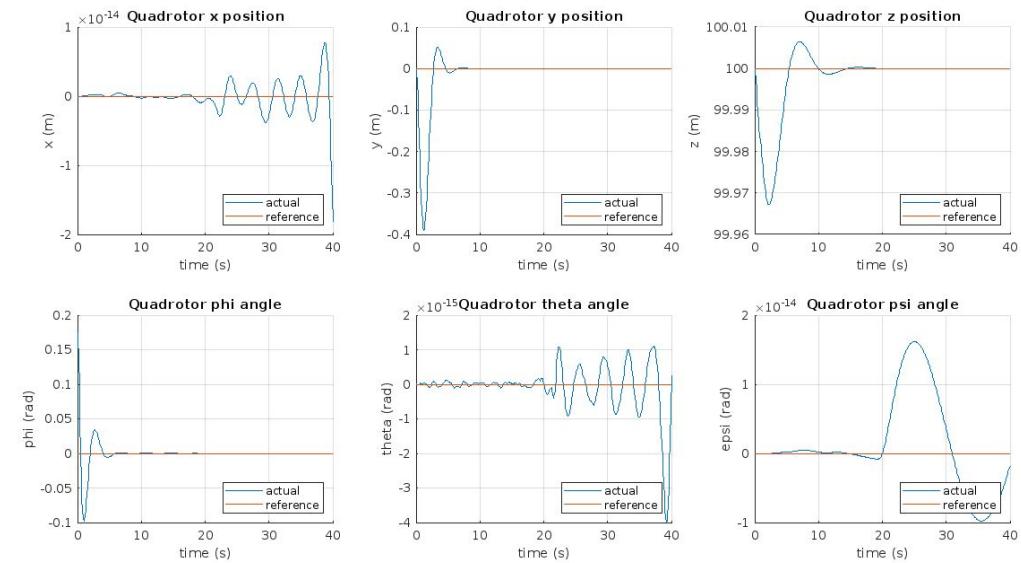
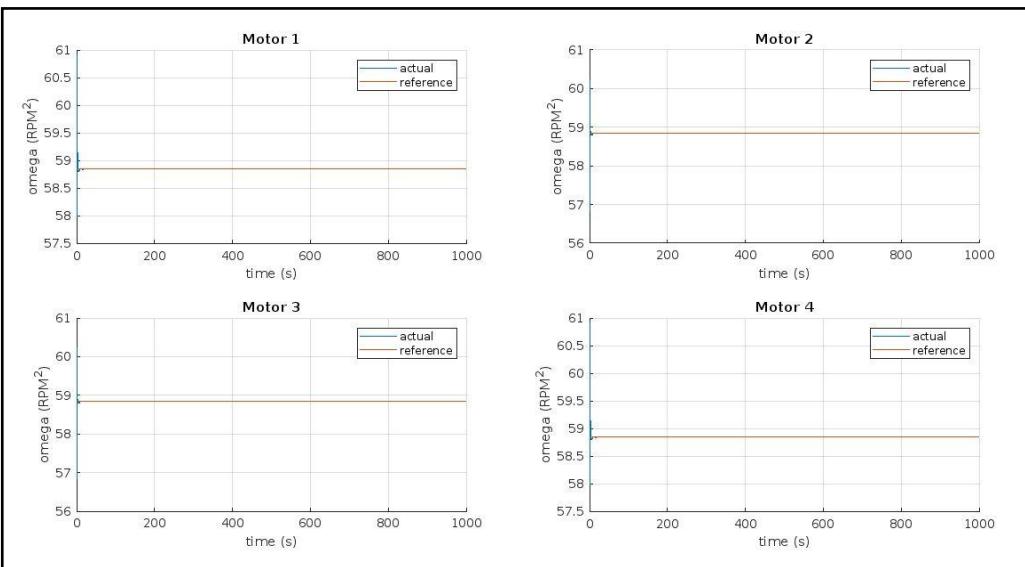


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the theta angle while hovering at 100 m for 40 s & 1000s.

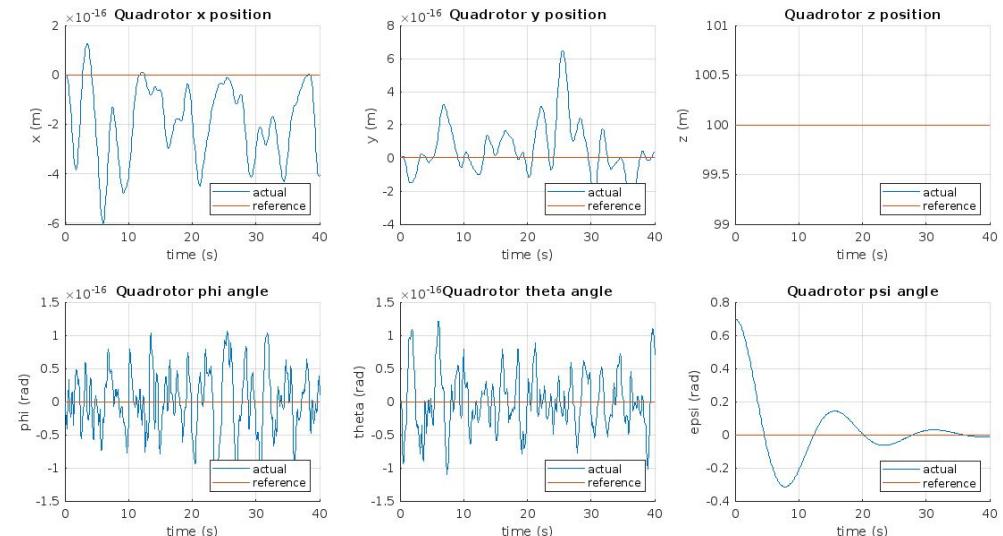
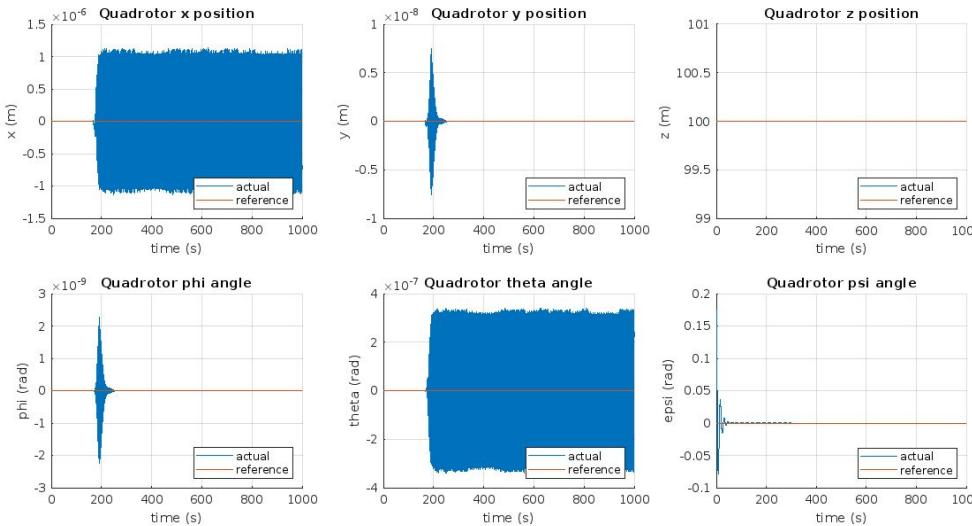


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the $\epsilon\psi$ angle while hovering at 100 m for 40 s & 1000 s.

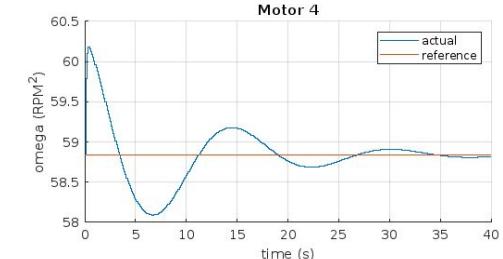
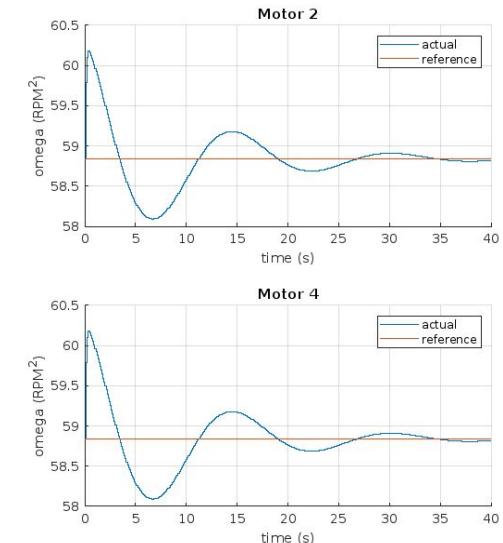
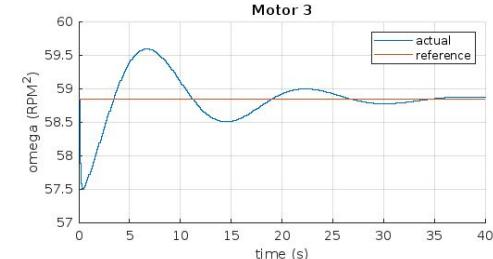
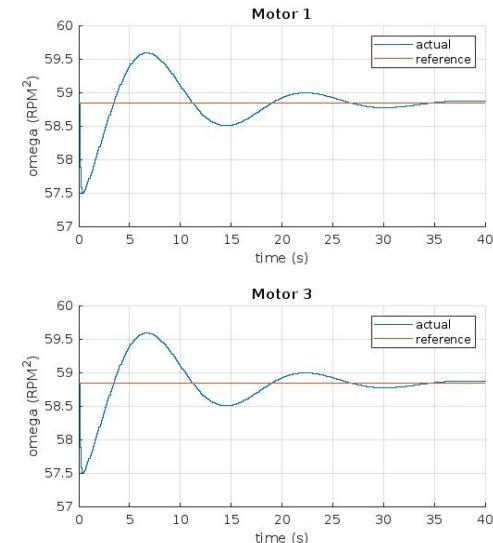
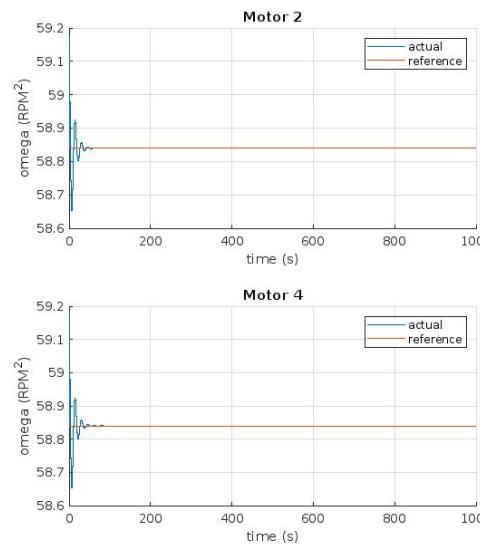
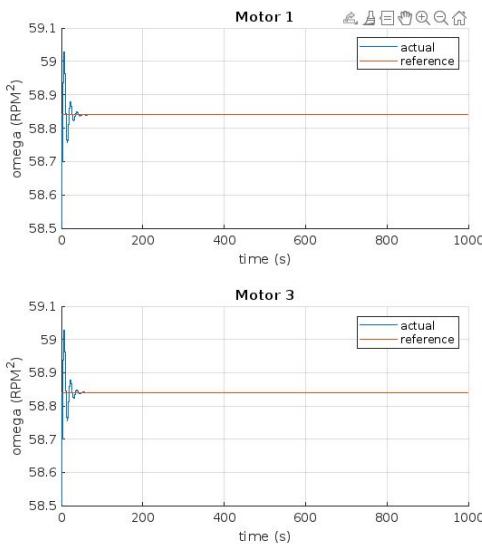


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in the ϵ_{pi} angle while hovering at 100 m for 40 s & 1000s.

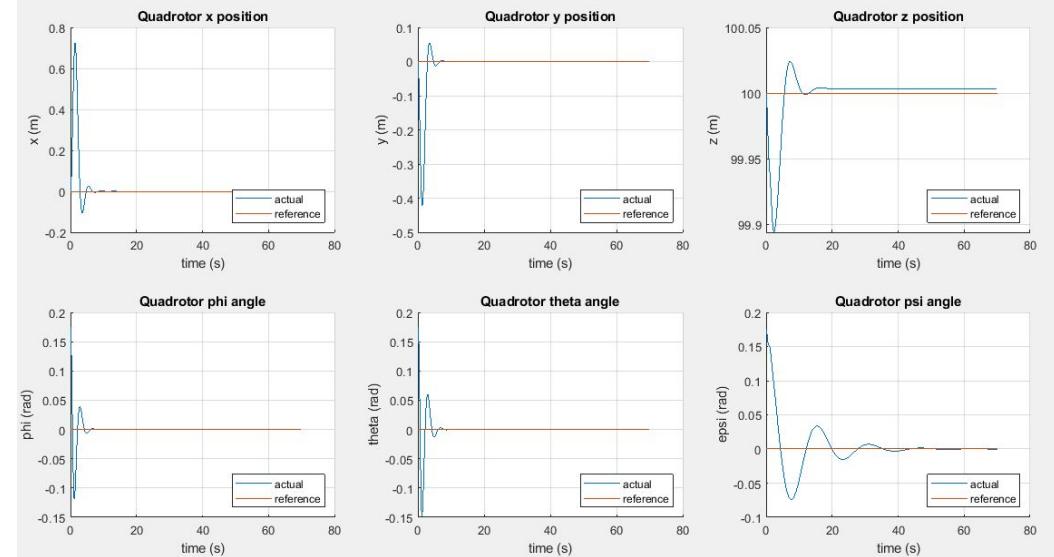
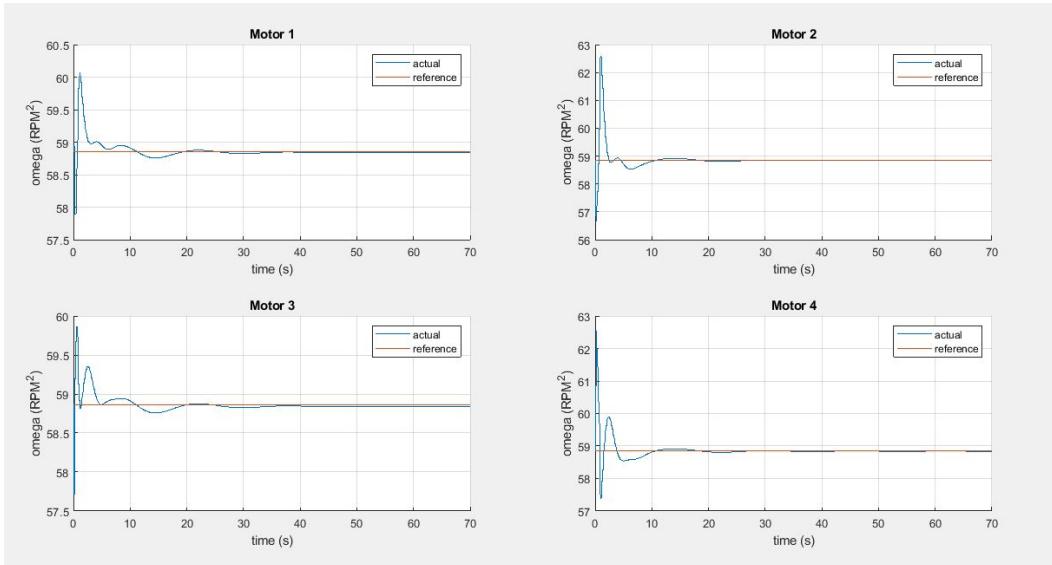


I. Nonlinear MPC



NMPC Initial Disturbance Rejection

10 deg initial disturbance in all three angles while hovering at 100 m for 70 s.



II. PID Controller



Introduction

*PID control is a simple yet powerful method of controlling system processes. Provided the error signal as input $e(t)$, the PID controller manipulates the error signal by evaluating three terms k_p , k_i , k_d .

*The terms are then added together creating the control action $u(t)$:

$$u(t) = k_p * e(t) + k_i * \int_0^t e(\tau) d\tau + k_d * \frac{d}{dt} e(t)$$

* K_p is proportional to the error signal. It applies a corrective action based on the magnitude of the error. It helps the controller to respond quickly to quick **variations** in the error. But, on the other hand, it could lead to overshoot and steady-state errors.

II. PID Controller



- ***Ki** keeps track of the accumulated **past errors**, it provides a control action that gradually **reduces** the accumulated error, which helps in eliminating offset in controlled variables.
- ***Kd** makes use of the error rate of change to anticipate future trends and dampen rapid changes and oscillations.

Parameter	Rise Time	Overshoot	Settling Time	S-S Error
k_p	Decrease	Increase	Small Change	Decrease
k_i	Decrease	Increase	Increase	Decrease
k_d	Small Change	Decrease	Decrease	No change

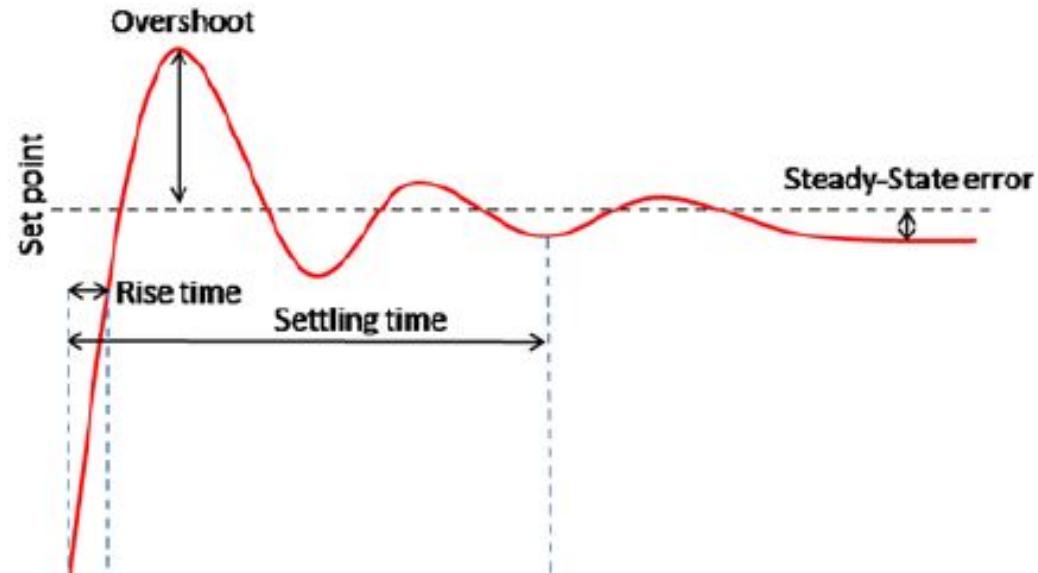
Table 5.2.1. PID Parameters Effect on Performance Characteristics (Nise, 2019).

II. PID Controller

Performance Characteristics

***Rise Time:** Represents the time taken to reach around 90-95% of the desired setpoint. A smaller rise time indicates a fast response.

***Overshoot:** Represents the maximum deviation from the desired setpoint. It's mainly caused due to aggressive control actions.

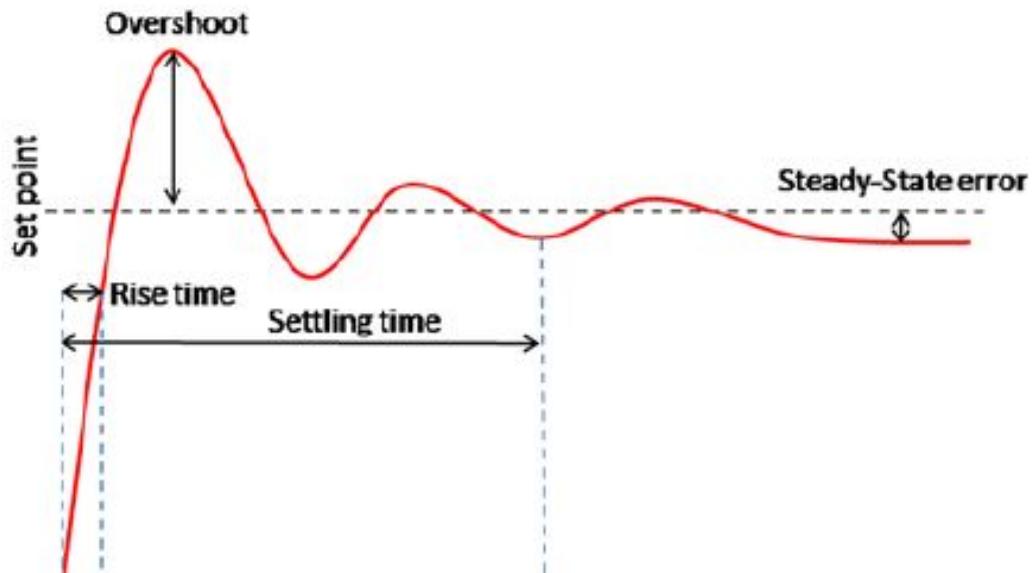


II. PID Controller

Performance Characteristics

***Settling Time:** Represents the time taken by the signal to reach and settle within an acceptable range of the setpoint and remains there indefinitely.

***Steady-state Error:** Represents the deviation that remains after the response reaches its steady state.



II. PID Controller

Control Architecture

*A quadrotor is a highly nonlinear and **underactuated** system, which means that it has fewer control inputs (actuators) than the degrees of freedom that should be controlled. Namely, a quadrotor has only **4 actuators**, meanwhile, it has **6 degrees of freedom**, both translational and rotational.

*A cascaded PID controller consists of several PID controllers that are arranged **hierarchically**, where each controller is responsible for controlling one of the system states.

*This arrangement in essence results in two control loops; **outer control loop** (position controller) and **inner control loop** (attitude controller).

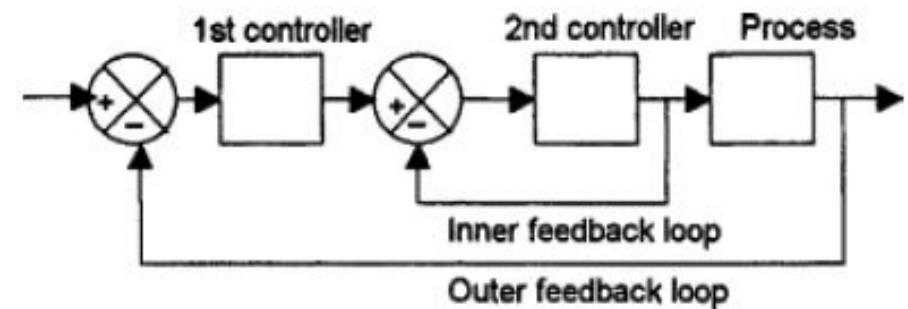


Figure 5.2.6. Cascaded PID Controller (Bolton, 2021).

II. PID Controller



Control Architecture

*Due to the **decoupled nature** of the system dynamics, the system states could be arranged such that the secondary process variables, that influence the primary process variables, are placed in the outer control loop.

*On the other hand, the **primary process variables** are placed in the **inner control loop**. In that sense, all the 6 degrees of freedom of the quadrotor are all controlled with only 4 actuator inputs.

*The outer control loop is responsible for controlling the position (x & y) and altitude (z) of the quadrotor. Meanwhile, the inner control loop is responsible for controlling the quadrotor orientation (attitude) in terms of roll, pitch, and yaw.

II. PID Controller

Simulink Representation

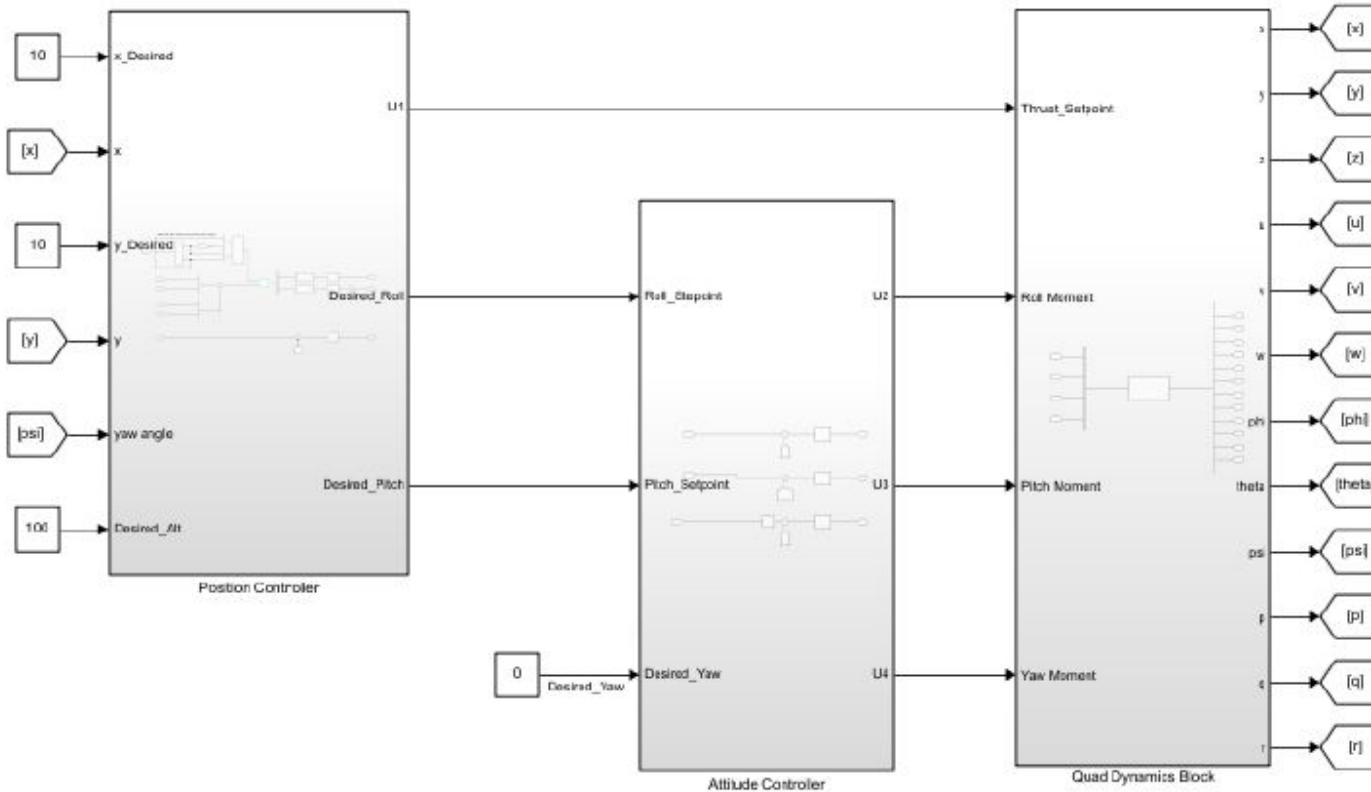


Figure 5.2.7. Simulink PID Controller Block.

II. PID Controller

Position Controller

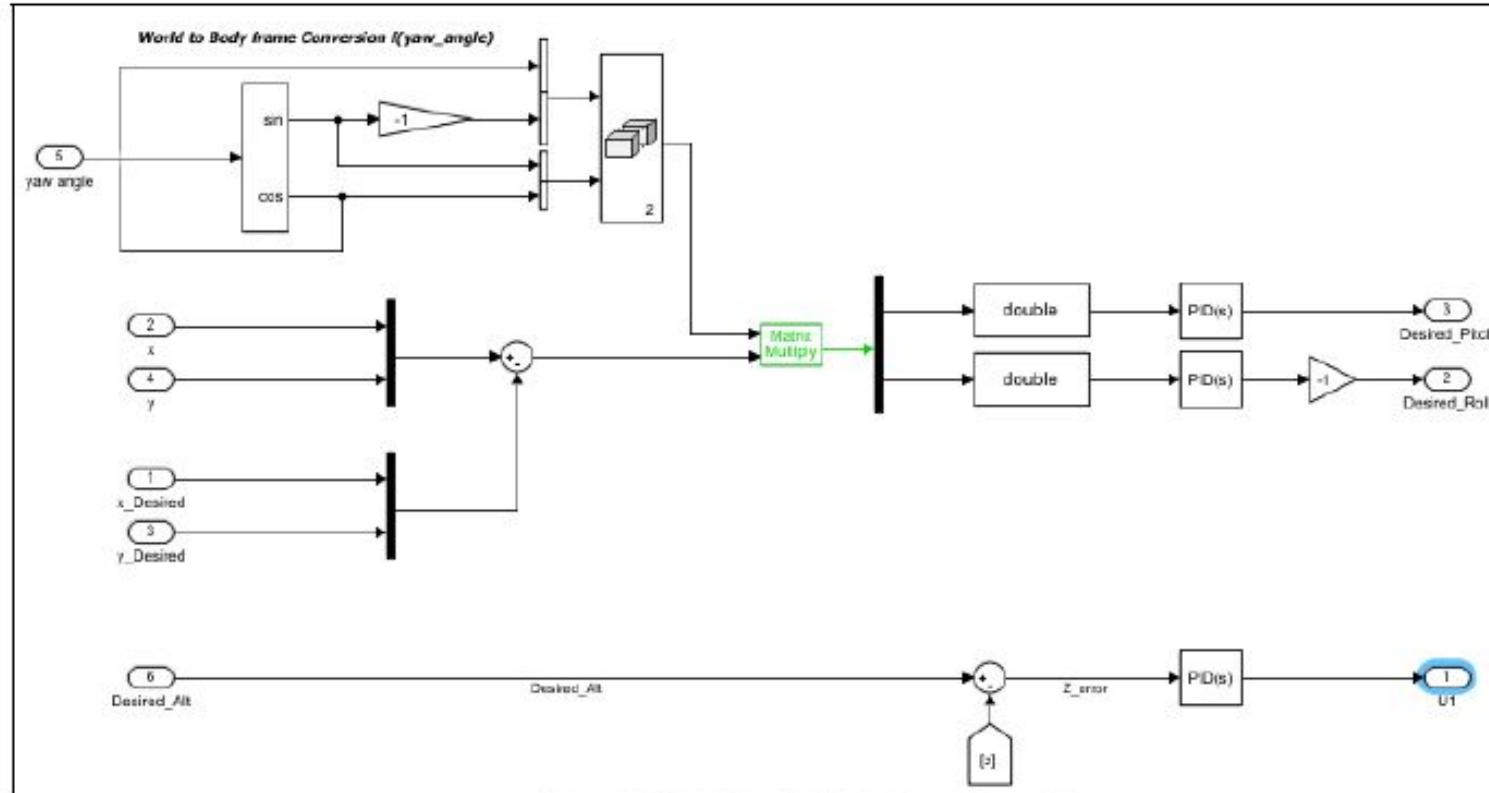


Figure 5.2.8. Simulink Position Controller.

II. PID Controller



Attitude Controller

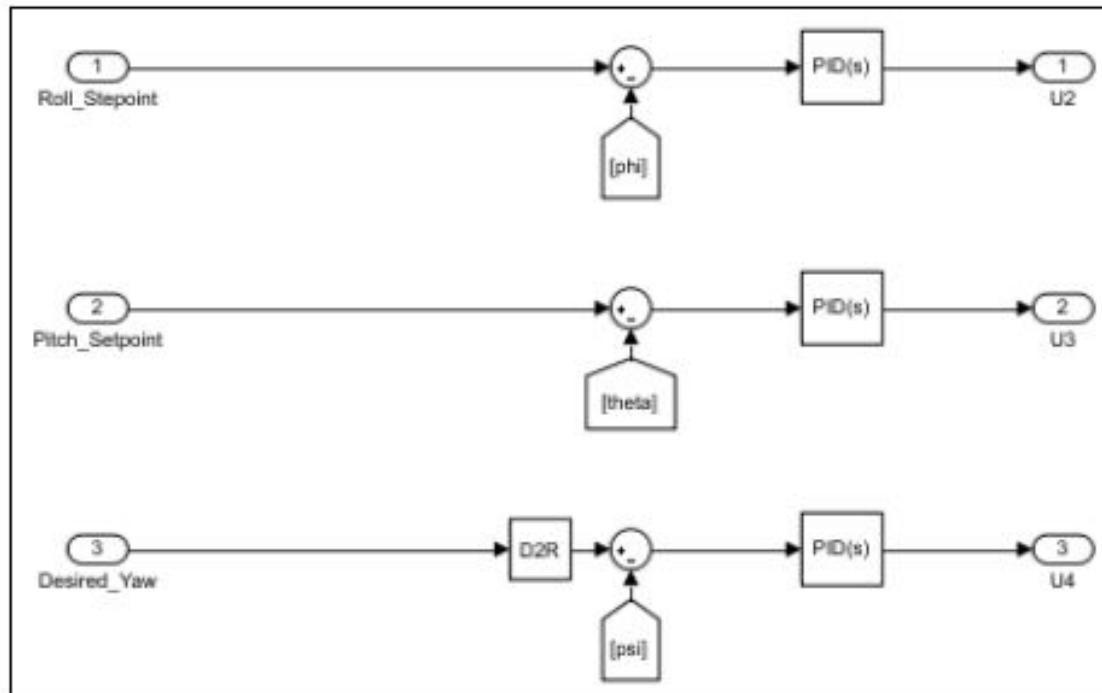


Figure 5.2.9. Simulink Attitude Controller.

II. PID Controller



Controller Tuning

There exist several methods that could be used to tune PID controllers, among which are:

***Manual Tuning:** this approach implies manually tweaking the PID controller gains until an acceptable response is obtained. However, in the case of a quadcopter, there are 6 PID controllers that should be adjusted. Accordingly, such an approach is deemed inefficient and impractical.

***Heuristics & Optimization Algorithms:** methods like Ziegler-Nichols and Cohen-Coon make use of the system response data and provide gain estimates using formulas. Such methods could be useful in achieving an acceptable response but not yet optimum.

II. PID Controller



Optimization Algorithms

On the other hand, optimization algorithms like genetic algorithms can find the optimum or near-optimal PID gains. Their main working mechanism is iteratively minimizing a **performance index (cost function)** specified to the algorithm. While such an approach is more **computationally intensive**, it can effectively handle a complex system with nonlinear dynamics, as in the current case of a quadrotor.

*Proposed Tuning Approach:

Ziegler-Nichols Method was used to obtain a good **initial guess** for PID parameters. Thereafter, the gains were manually tweaked in order to test the response under **small perturbations** in PID parameters. In such a sense, an estimate for the **search area** was identified, which was used together with the initial guess as an input to optimization algorithms in order to fine-tune the controller and reach the best performance possible.

II. PID Controller



Ziegler-Nichols Method

The method is applied to tune the PID controller in the following manner:

1. The integral gain is set to zero.
2. The proportional gain is increased incrementally until sustained oscillations are obtained, set to K_u .
3. The period of oscillation, time from peak to peak, is measured and set to T_u .

Controller	K_p	T_i	T_d	K_i	K_D
PID	$0.6 * K_u$	$T_u/2$	$T_u/8$	$1.2 * K_u/T_u$	$0.075 * K_u T_u$
P	$0.5 * K_u$	-	-	-	-
PI	$0.45 * K_u$	$T_u/1.2$	-	$0.54 * K_u/T_u$	-
PD	$0.8 * K_u$	-	$T_u/8$	-	$0.1 * K_u T_u$

Table 5.2.2. PID ZN Controller Gain Estimation (Meshram & Kanajiya, 2012).

II. PID Controller



Optimization Algorithm

*The choice of which optimization algorithm to use is dependent on the **system characteristics**. In the case of a quadrotor, the system dynamics are **highly nonlinear**. In addition, the optimization problem involves variables with **high-dimensional search space**. Moreover, the fitness (cost) function is **non-differentiable**. Therefore, the gradient is not available for use.

*Finally, the optimization problem is considered **multimodal**, meaning that several optimal or near-optimal solutions exist depending on the chosen search space. All of this narrowed down the options to a few algorithms, among which **Genetic Algorithm** was suitable for our case.

II. PID Controller



GA Working Mechanism

*The algorithm is inspired by the process of **natural selection**, in which the algorithm operates on a set of potential solutions, where individuals undergo genetic operations to **reproduce** better generations.

*Reproduction is meant to produce new generations that carry better characteristics contributing to a better **fitness value** for the objective function.

*In that sense, the algorithm can help find the optimum PID parameters that minimize the performance index.

II. PID Controller



GA Setup

***Search Space:** while a large search space can help the algorithm explore further solutions, potentially improving the performance, it can come at the expense of high **computational effort**. Accordingly, the ZN method was initially used to **narrow down** to a viable search space.

***Population Size:** large population size can also help in better exploration of the search space. However, if it's too high, it may slow down the simulation due to **computational overhead**.

*As more individuals in this case will need to be evaluated and selected in each generation (iteration). Therefore, it's recommended to start from 50 to 200 initially and increase incrementally while monitoring performance.

II. PID Controller



GA Setup

***Max Number of Generations:** it determines how long the algorithm will run before termination. Setting a high number of generations can **potentially** improve the result, as the algorithm will continue searching for solutions. However, it's also **computational overhead**. It's also recommended to start with 100 to 1000 and increase incrementally as well.

***Performance Index:** the choice of the fitness (cost) function contributes greatly to both the convergence of the algorithm and performance improvement. In this case, two cost functions are **proposed**; the least squares method and integral time absolute error.

II. PID Controller



Pre-Simulation

All the controllers were tuned as **PD controllers**, meaning that the integral gain was set to zero in all controllers. This was done for the following reasons:

***Simplicity:** removing the integral gain will reduce the problem size in terms of the number of variables to be tuned. In addition, the model dynamics are already fast enough, so PD is expected to perform well.

***Model Inaccuracies:** the mathematical model of the quadrotor may fail to capture the true dynamics exactly of the quadrotor in real-life operation. As the measurements are subject to sensor noises. The integral term is very sensitive to such uncertainties and may lead to instability.

II. PID Controller



Ziegler-Nichols Results

The method was applied to the model states while maintaining a zero desired yaw angle.

Inner Loop Controller		
<i>State</i>	K_u	T_u
Phi	0.1875	6.93 s
Theta	0.075	7.33 s
psi	0.625	1.6 s

Table 5.2.3. ZN Results for Inner Loop.

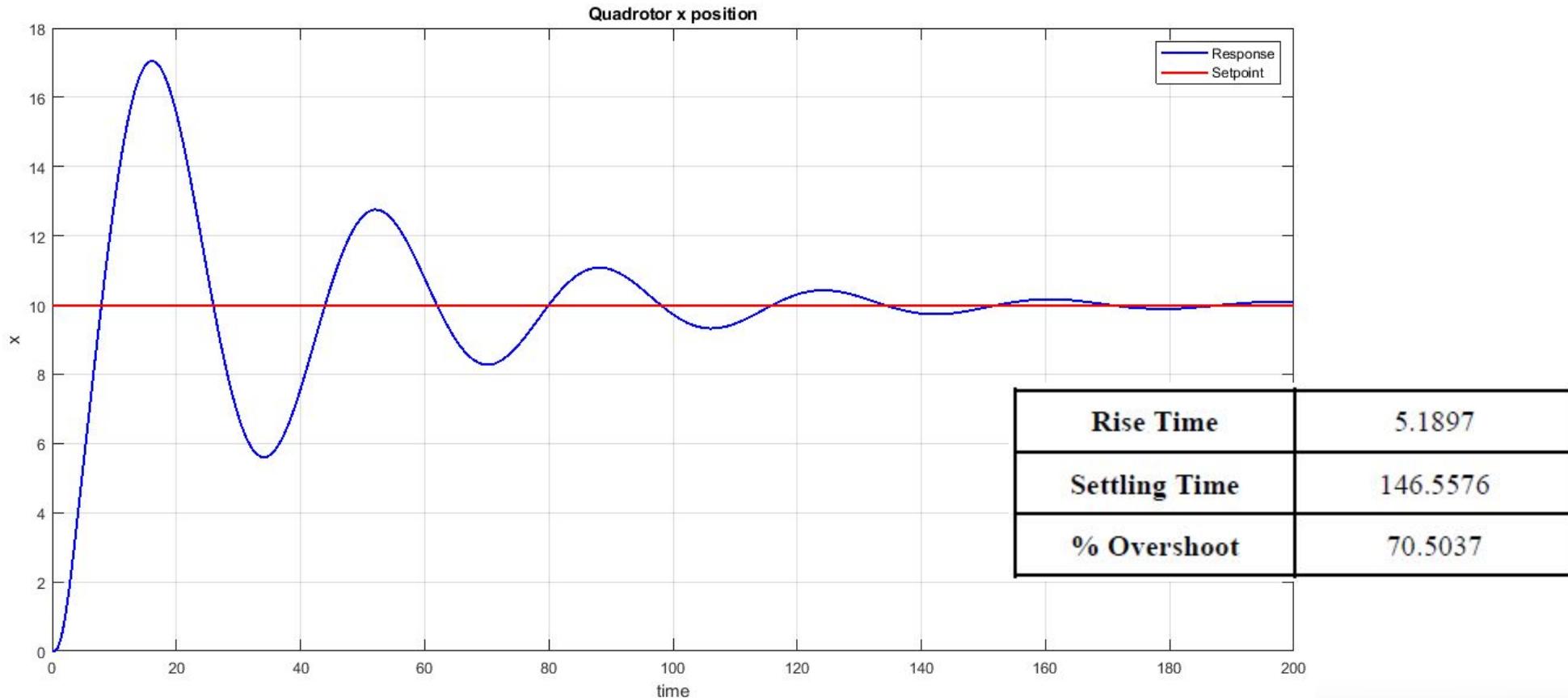
Outer Loop Controller		
<i>State</i>	K_u	T_u
X	$2.5 * 10^{-3}$	16 s
Y	$3.75 * 10^{-3}$	13.3 s
Z	18.75	3.73 s

Table 5.2.4. ZN Results for Outer Loop.

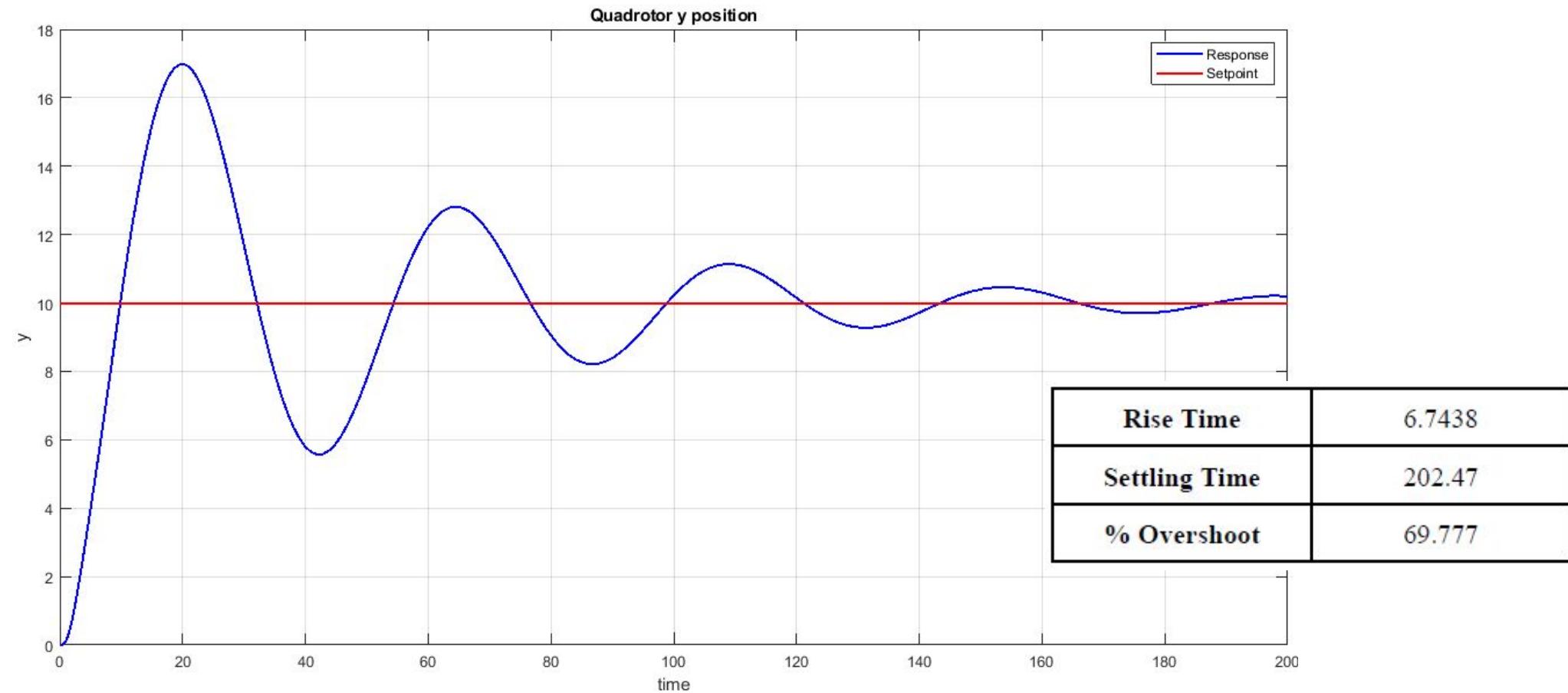
II. PID Controller



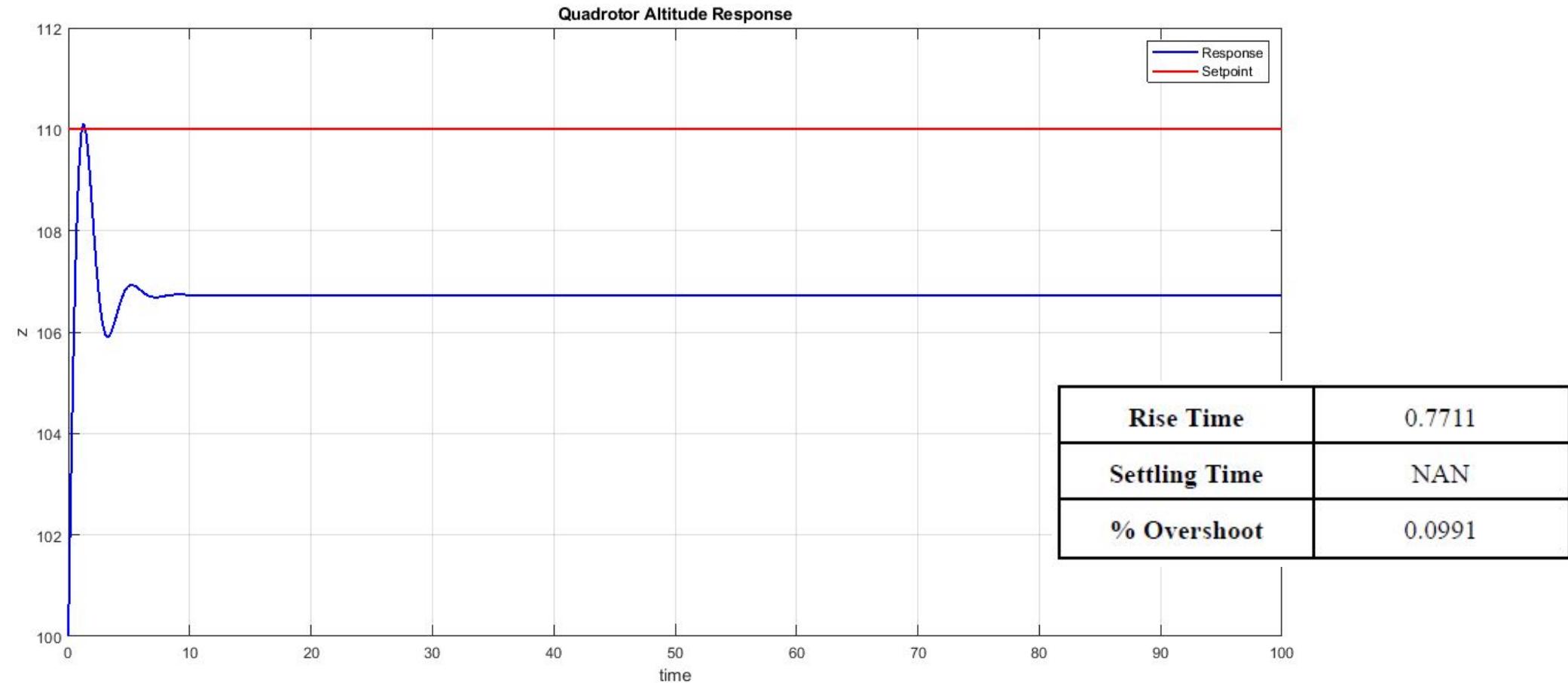
ZN Simulation Step Response Results $(0, 0, 100) \rightarrow (10, 10, 110)$



II. PID Controller



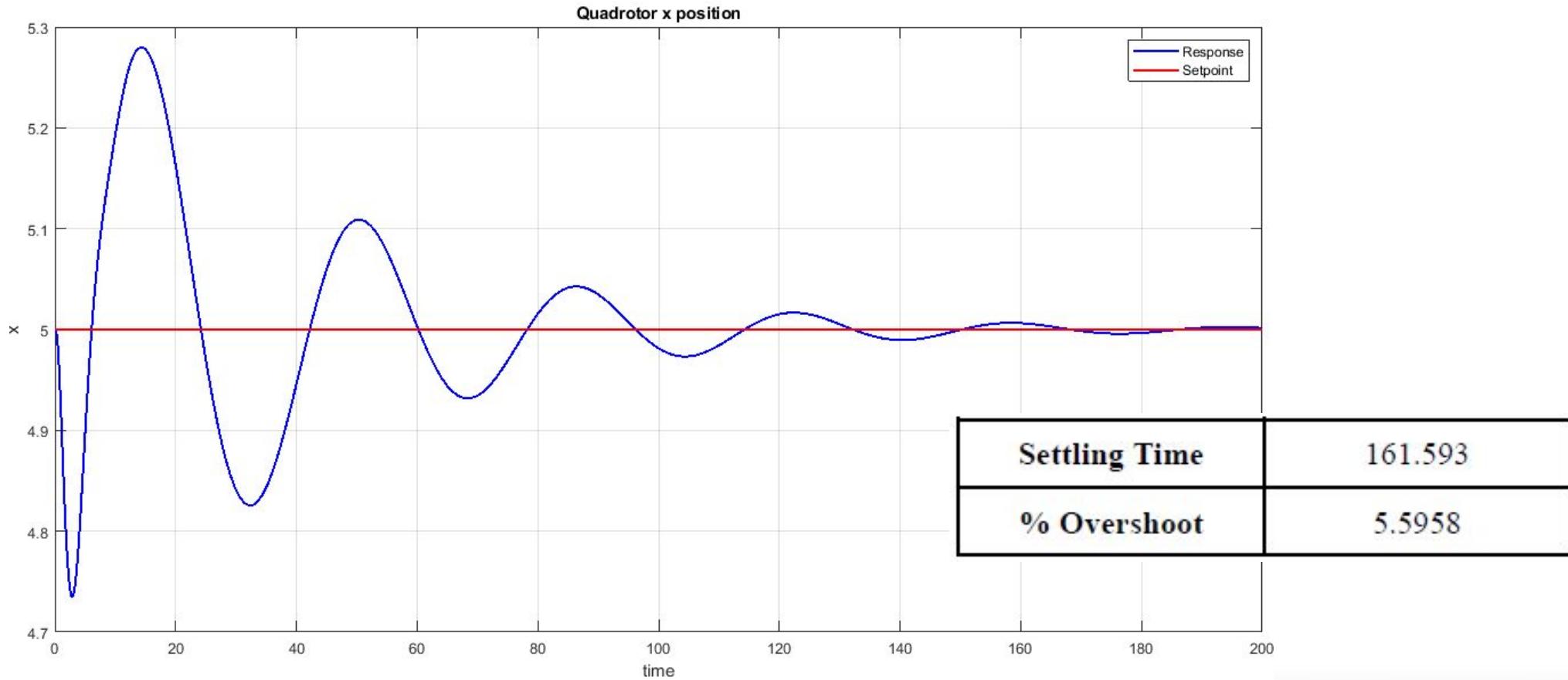
II. PID Controller



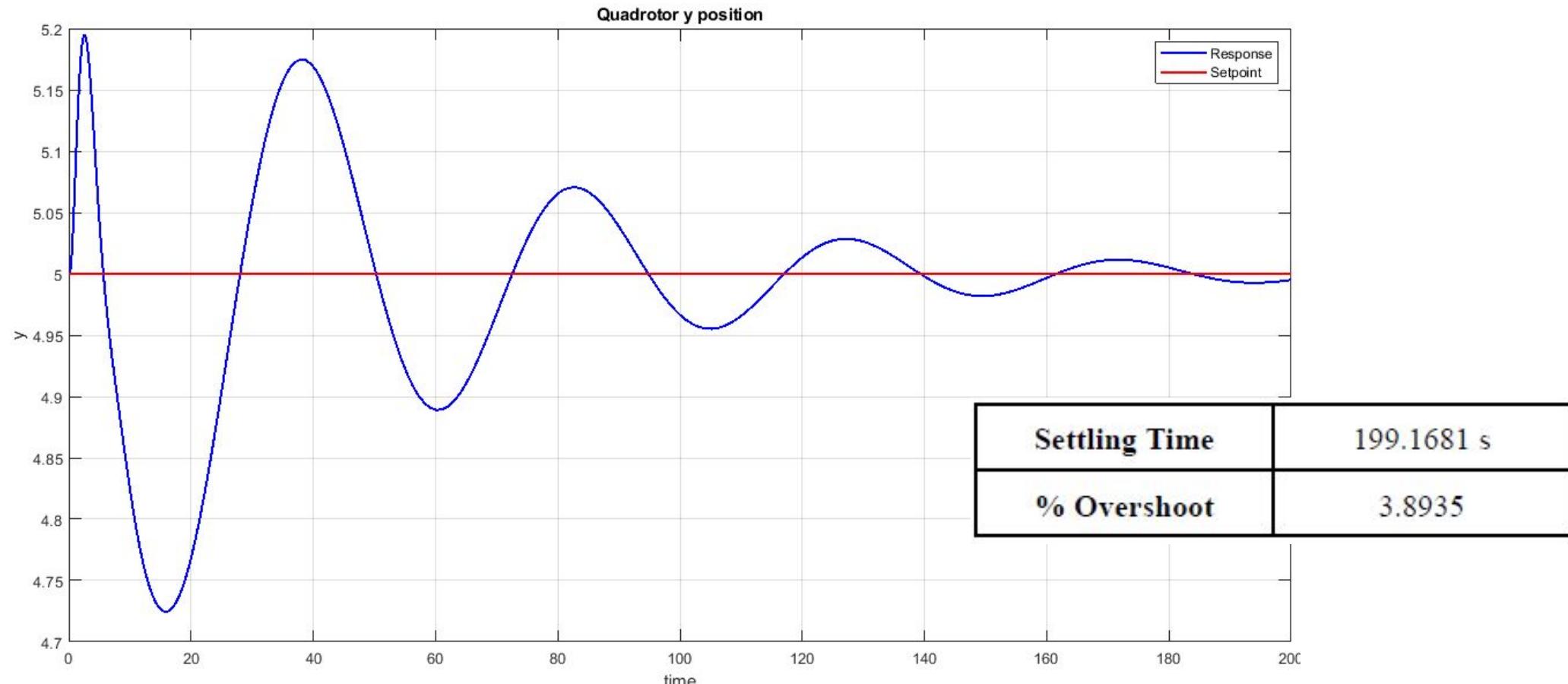
II. PID Controller



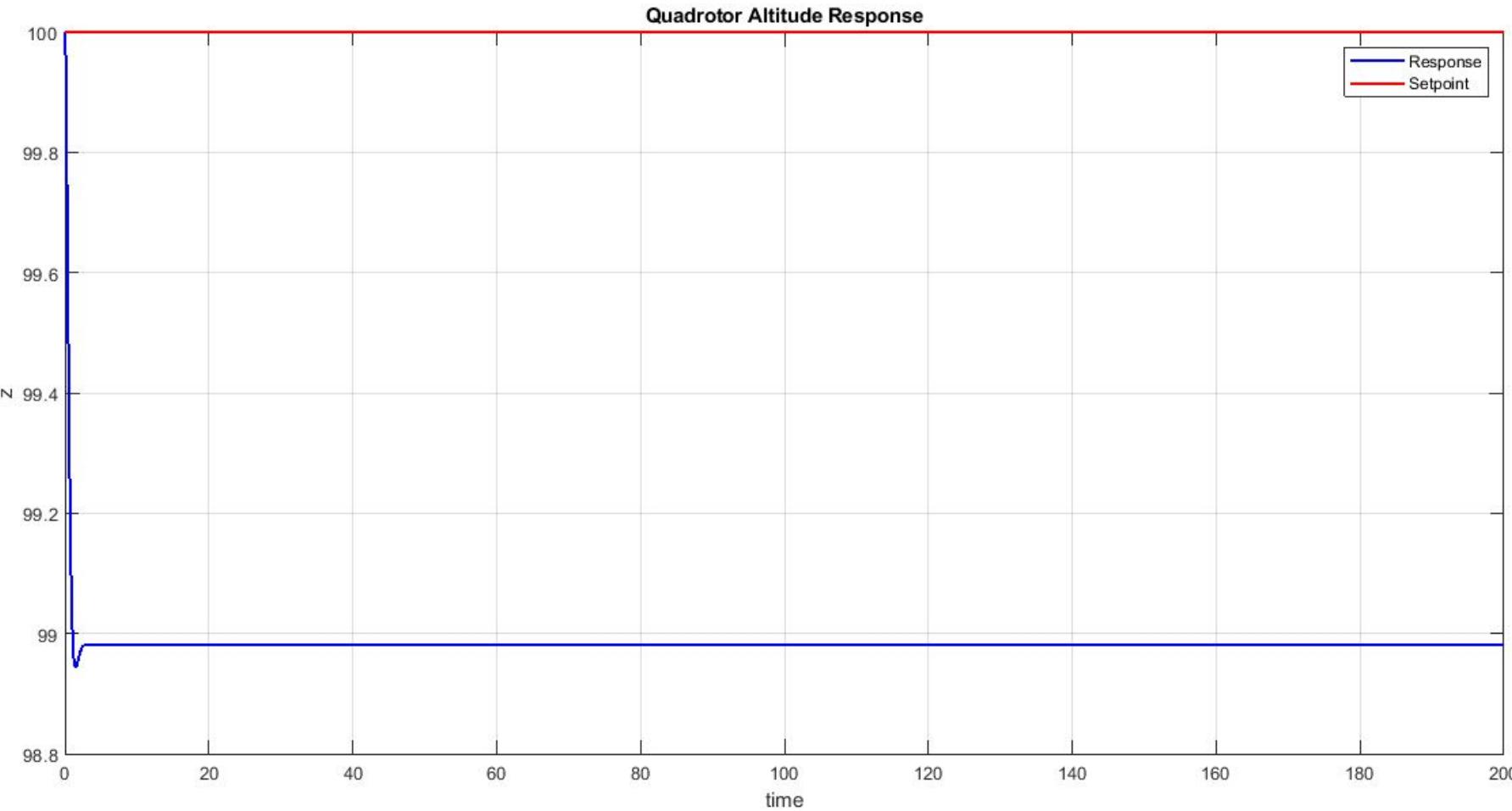
ZN Results ~ Initial Disturbance Rejection 2 degrees in roll & pitch



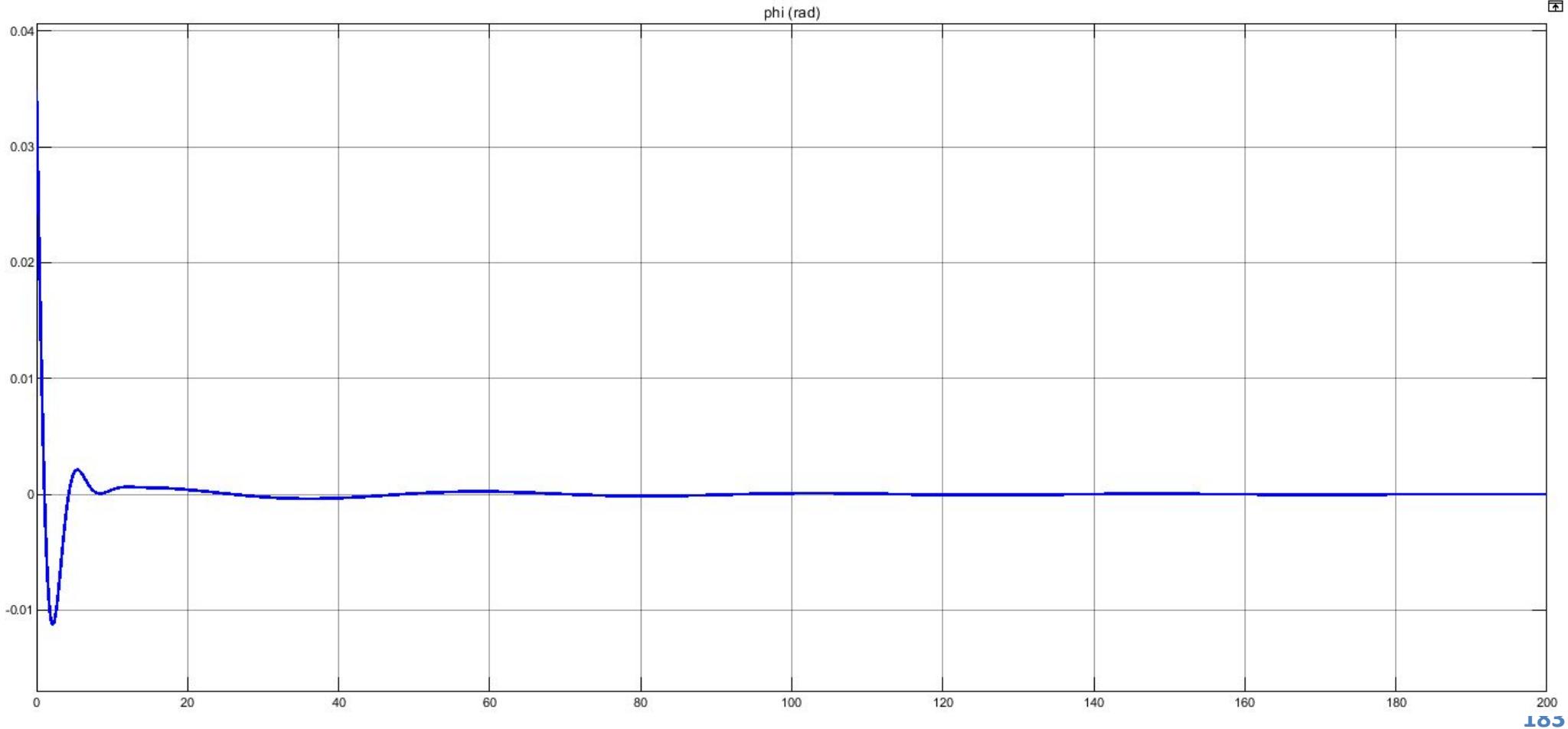
II. PID Controller



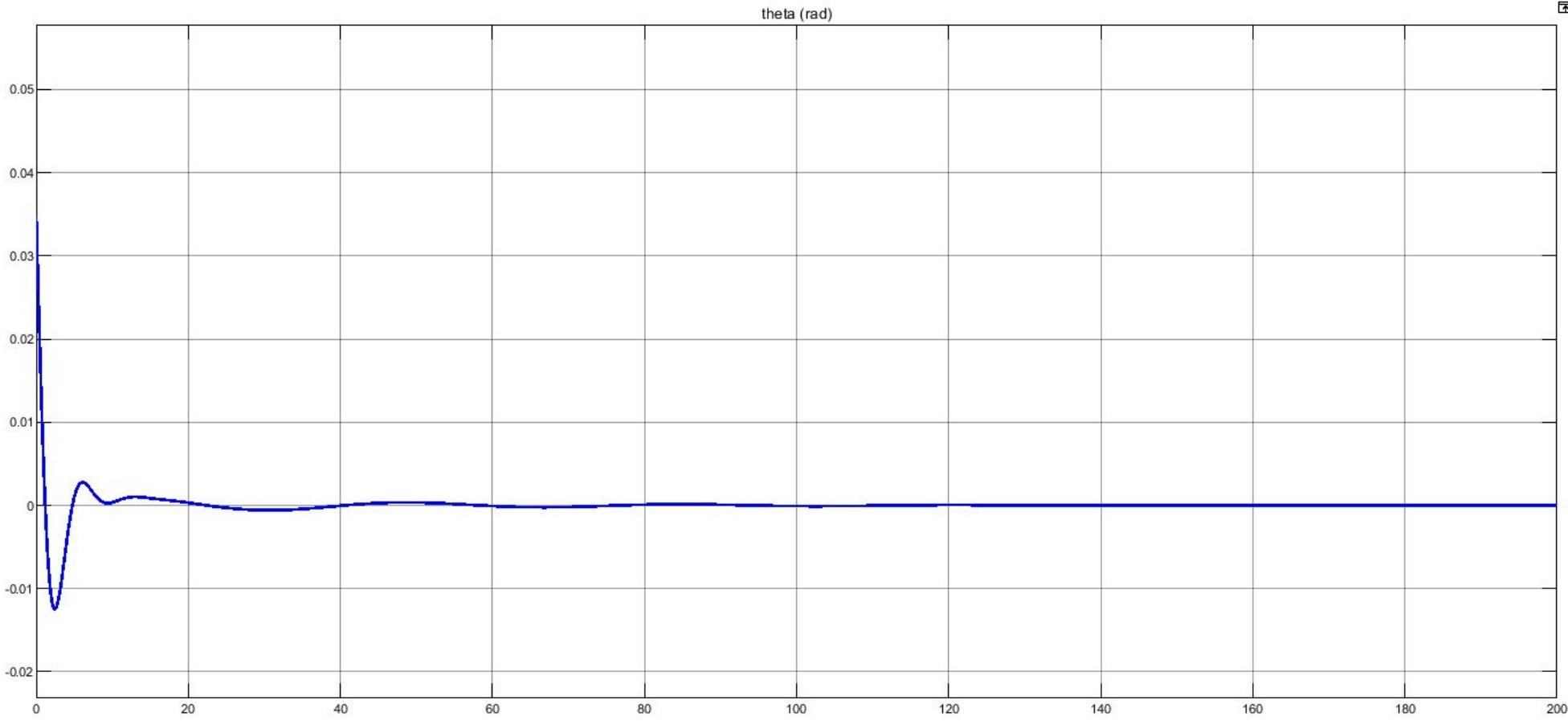
II. PID Controller



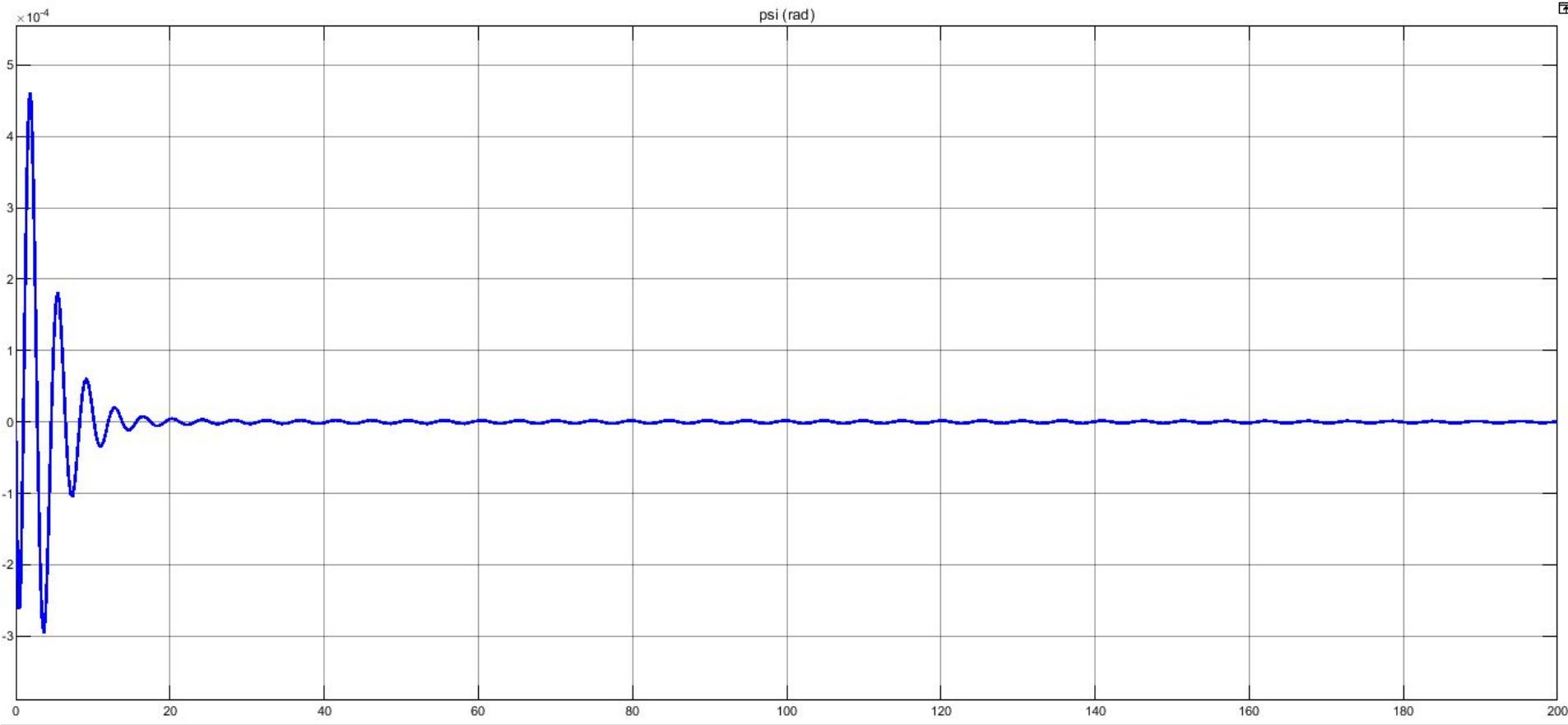
II. PID Controller



II. PID Controller



II. PID Controller



II. PID Controller



Discussion

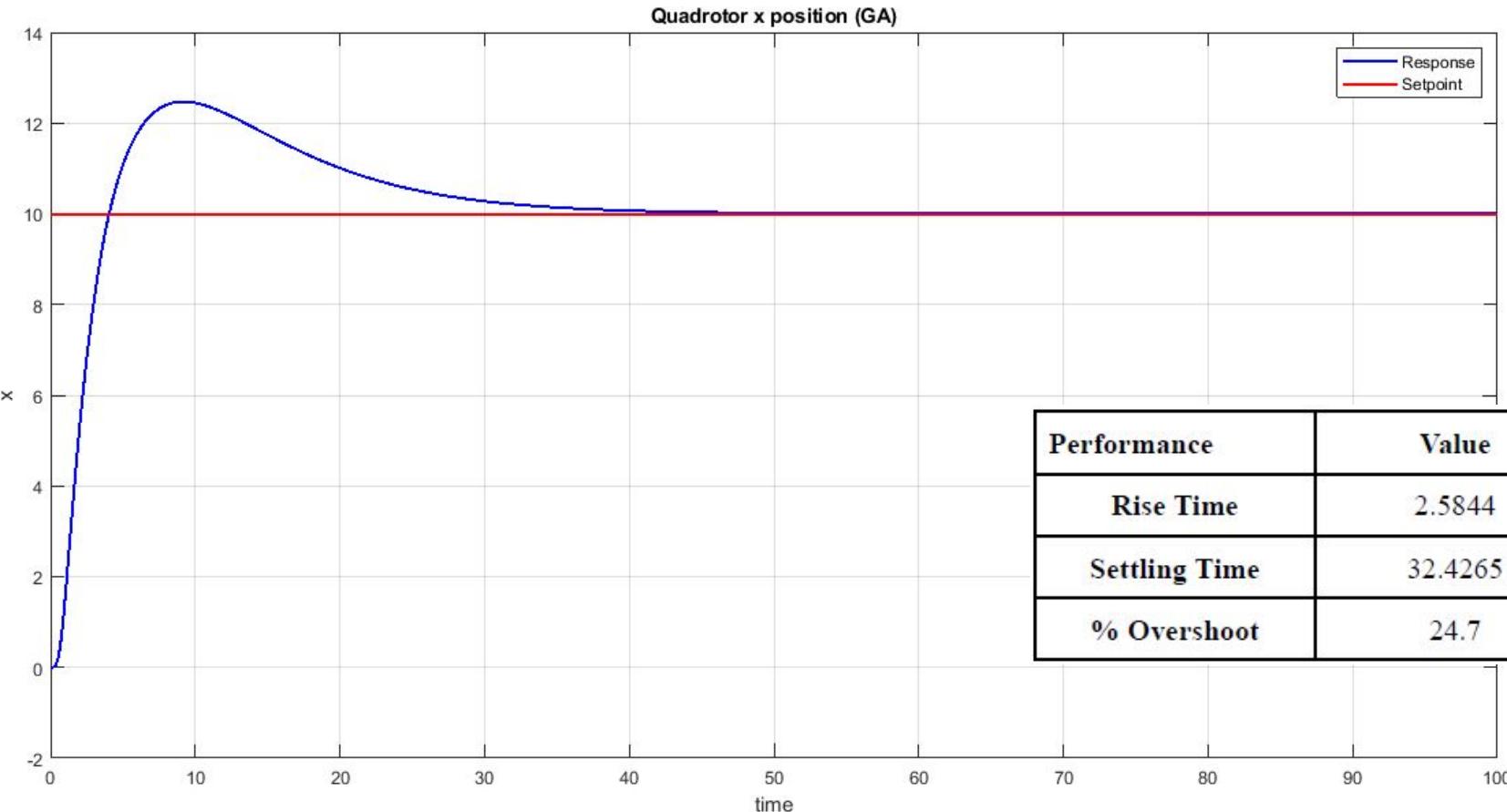
*As for the **step response**, high levels of overshoot and long settling times were spotted. However, the controller managed to settle eventually around the setpoint.

*As for **disturbance rejection**, the overshoot was acceptable, as it oscillates with small values across the setpoint. However, the settling time problem persists in this case as well.

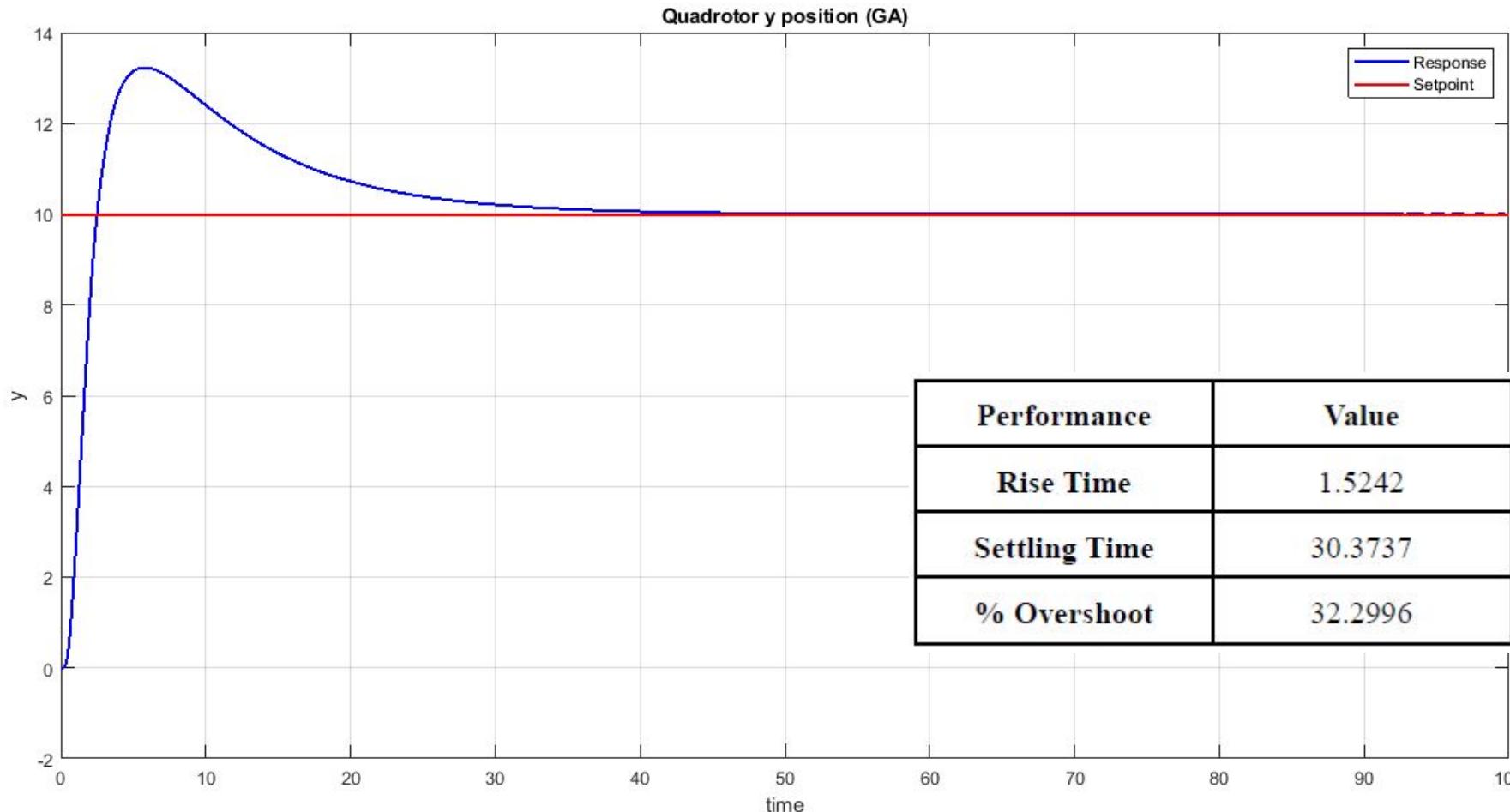
II. PID Controller ~ GA Results



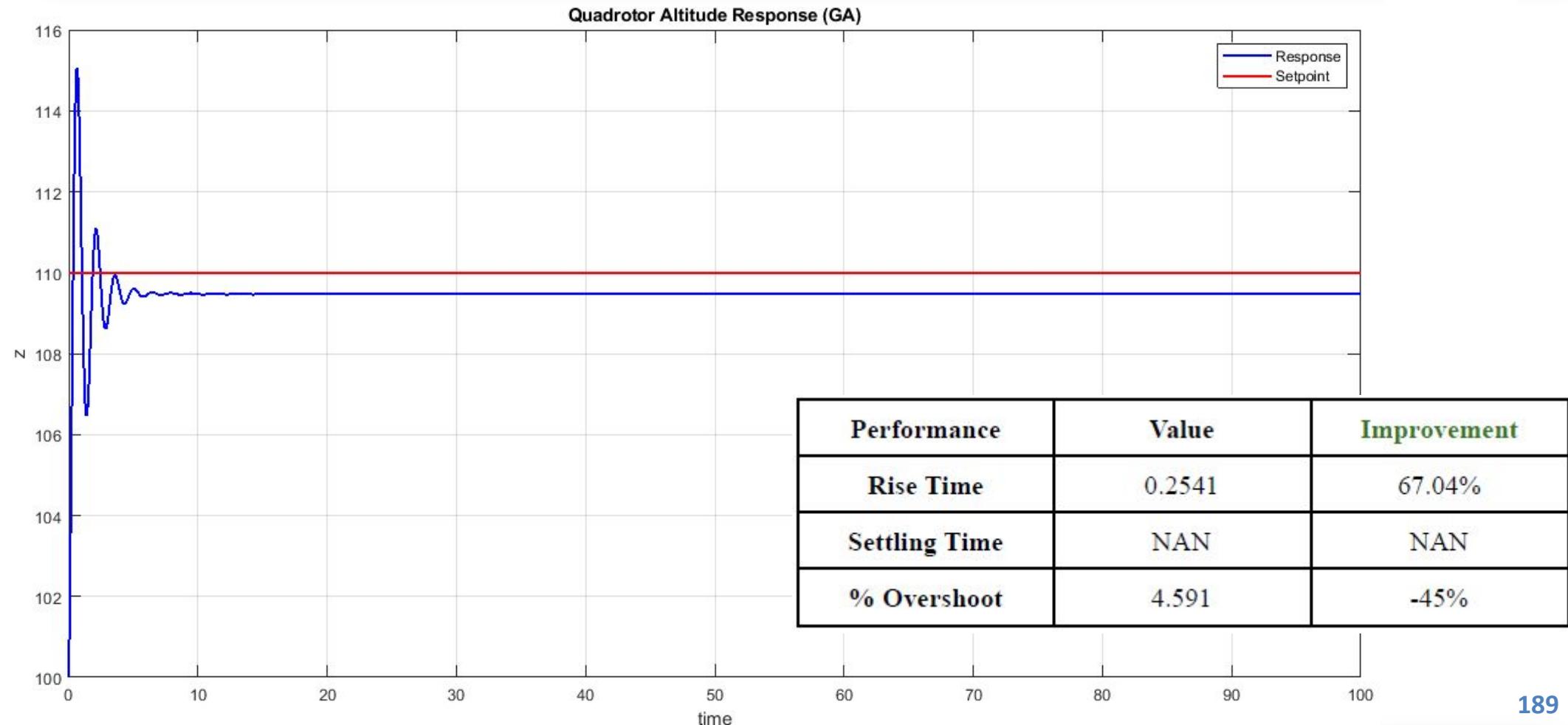
Step Response



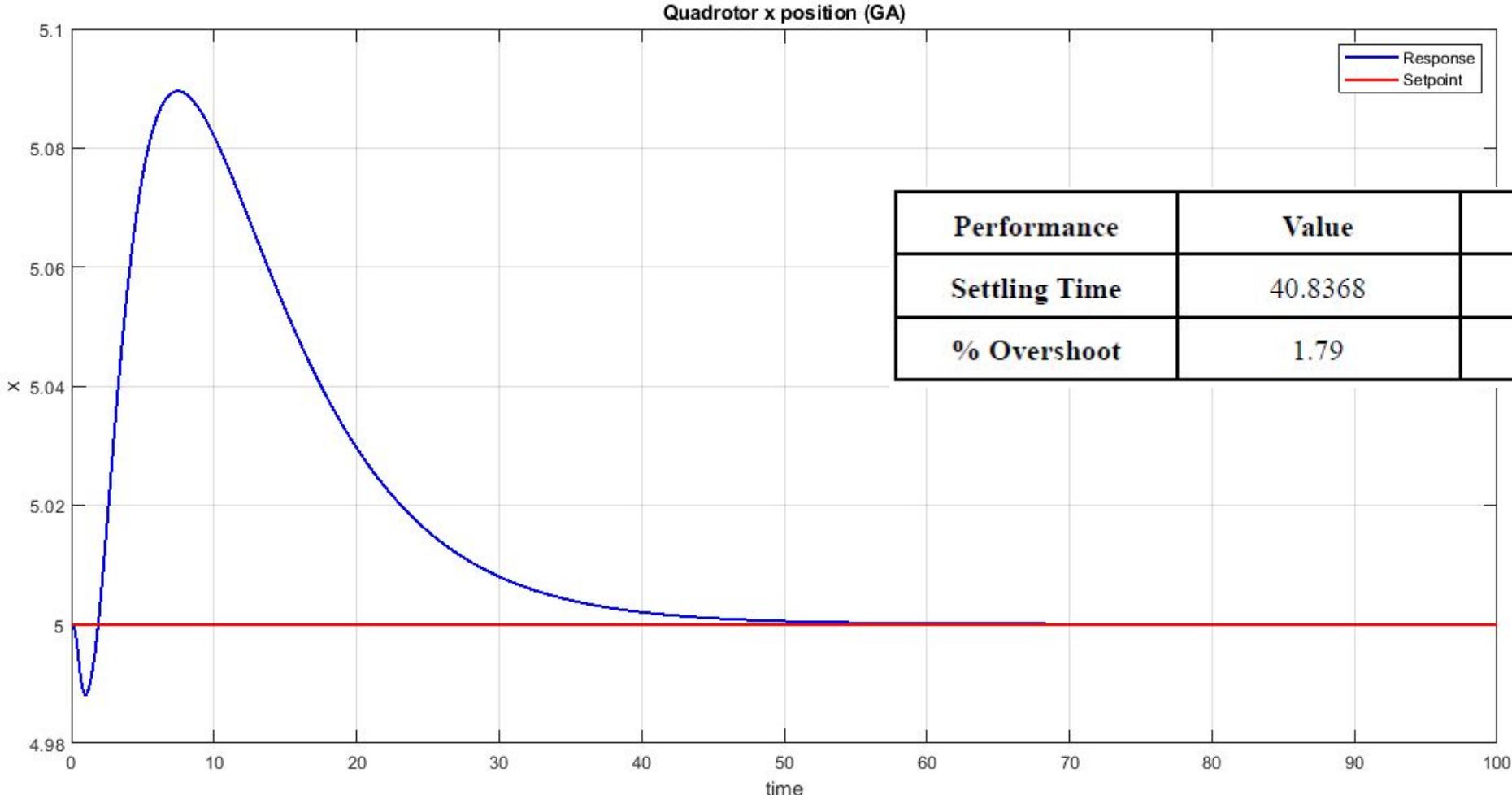
II. PID Controller ~ GA Results



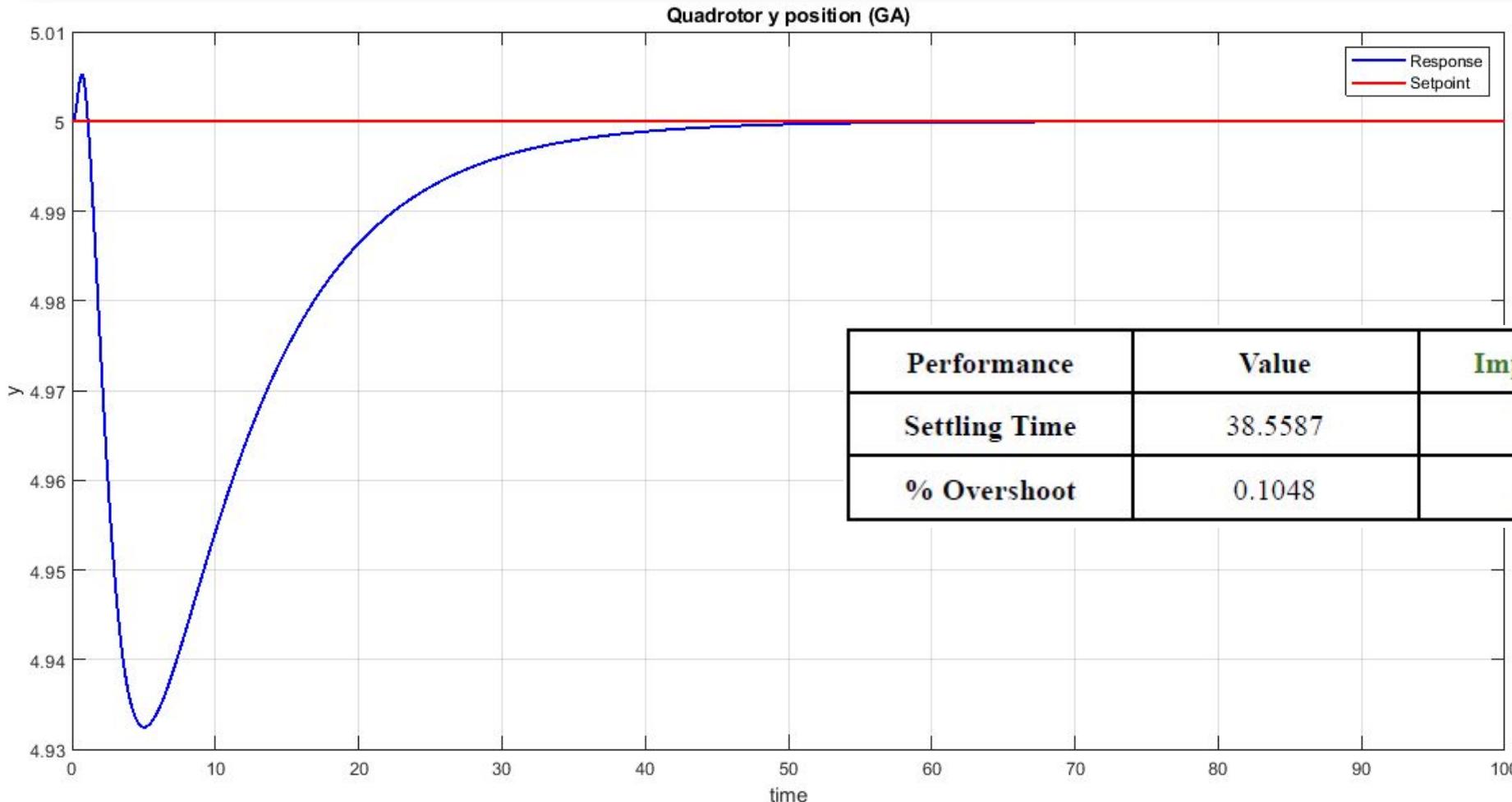
II. PID Controller ~ GA Results



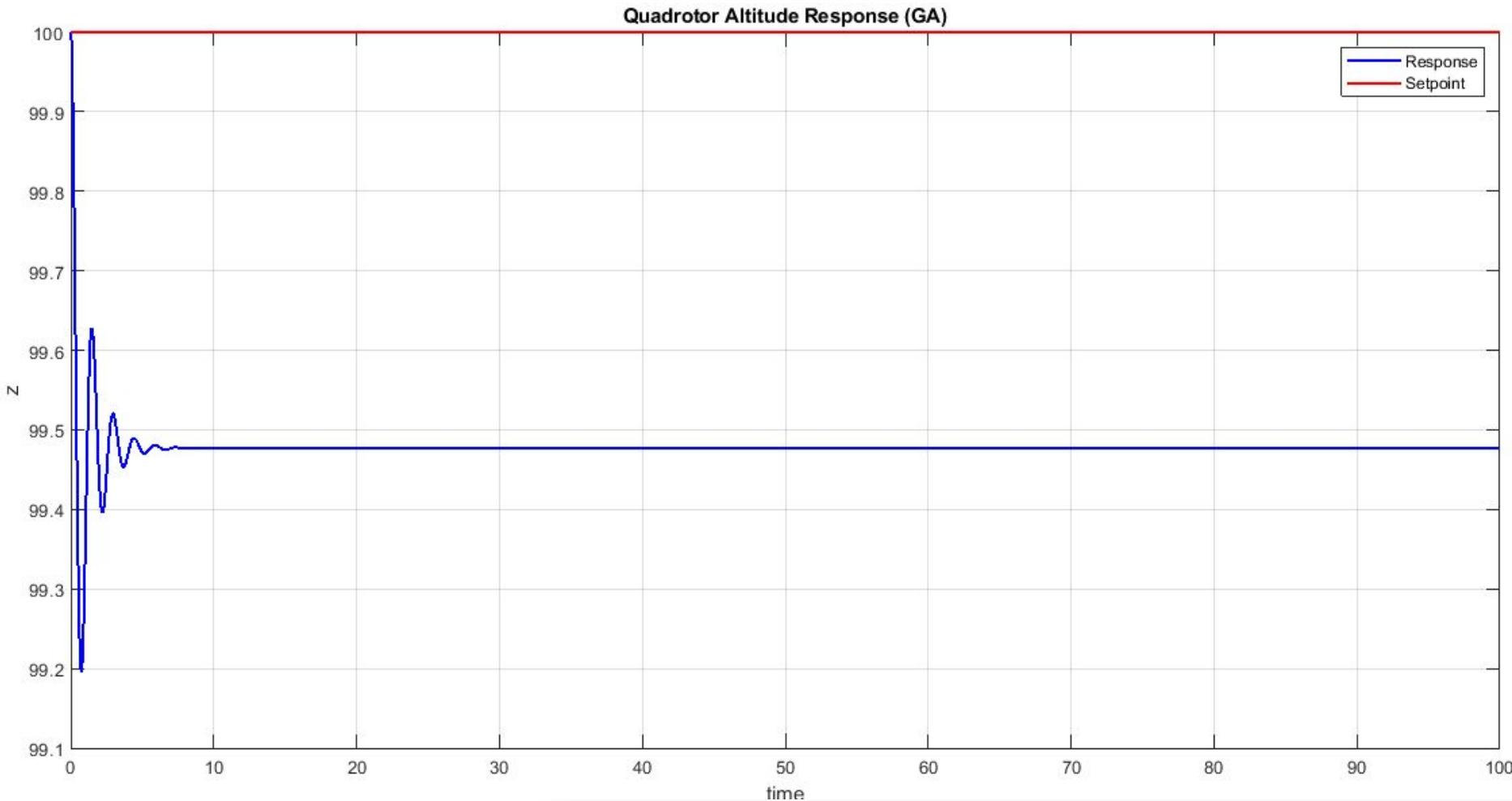
II. GA Results ~ Initial Disturbance Rejection



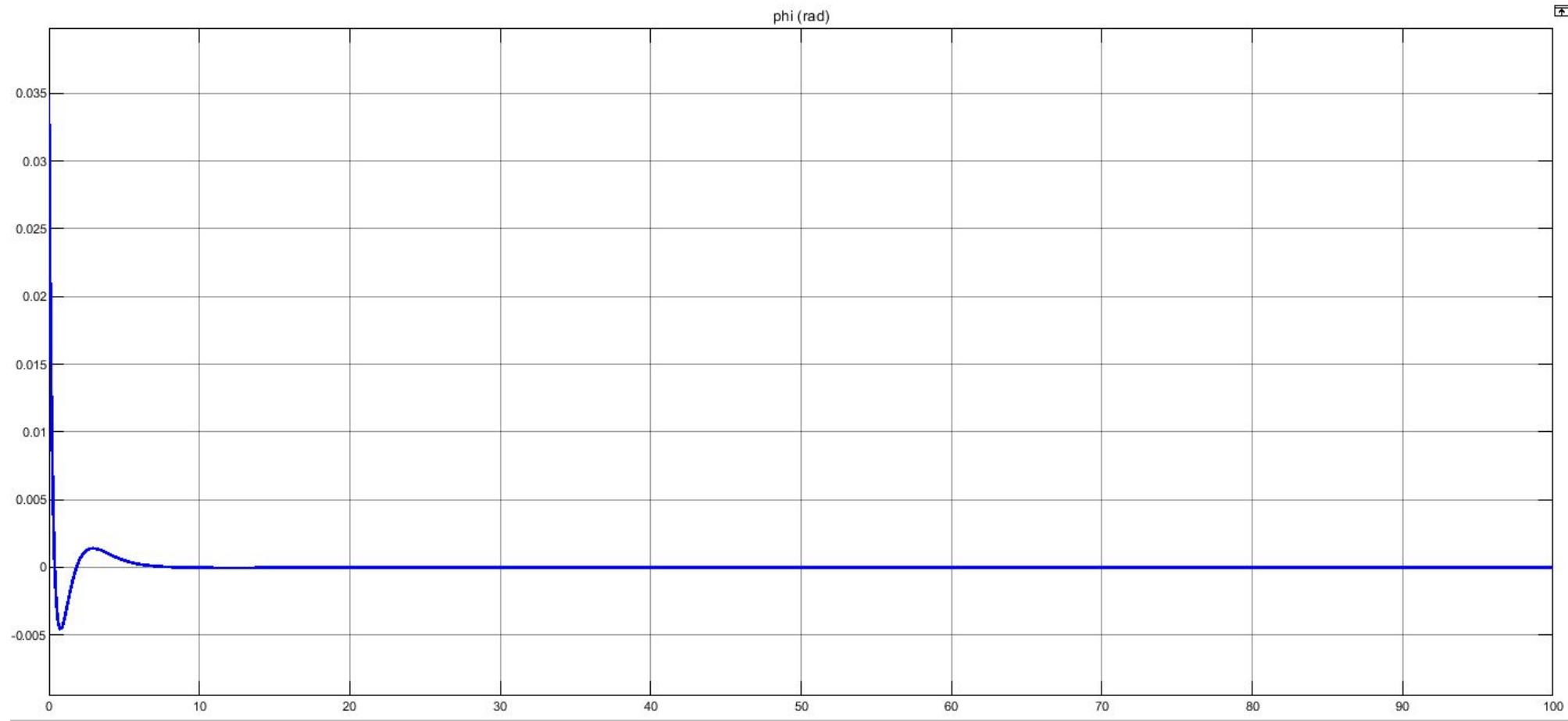
II. GA Results ~ Initial Disturbance Rejection



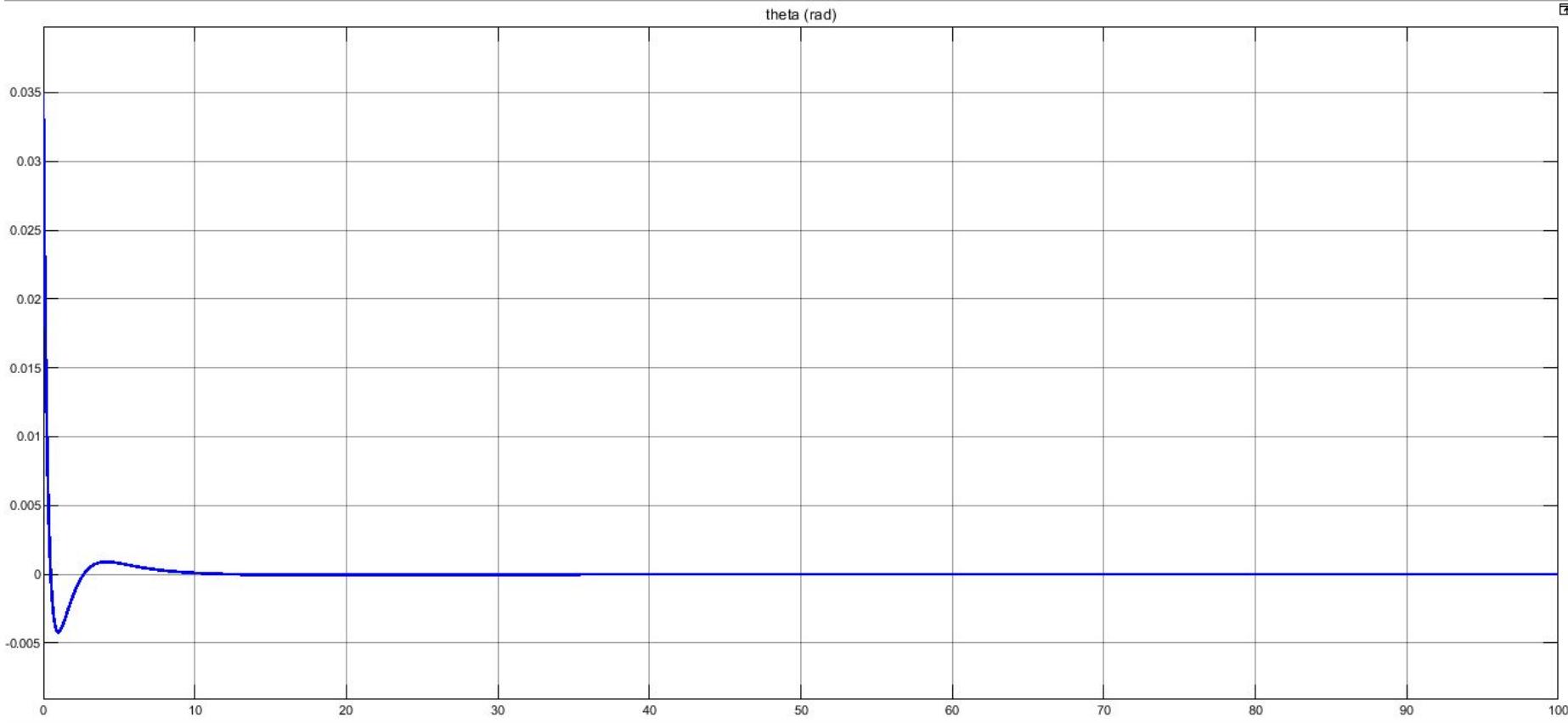
II. GA Results ~ Initial Disturbance Rejection



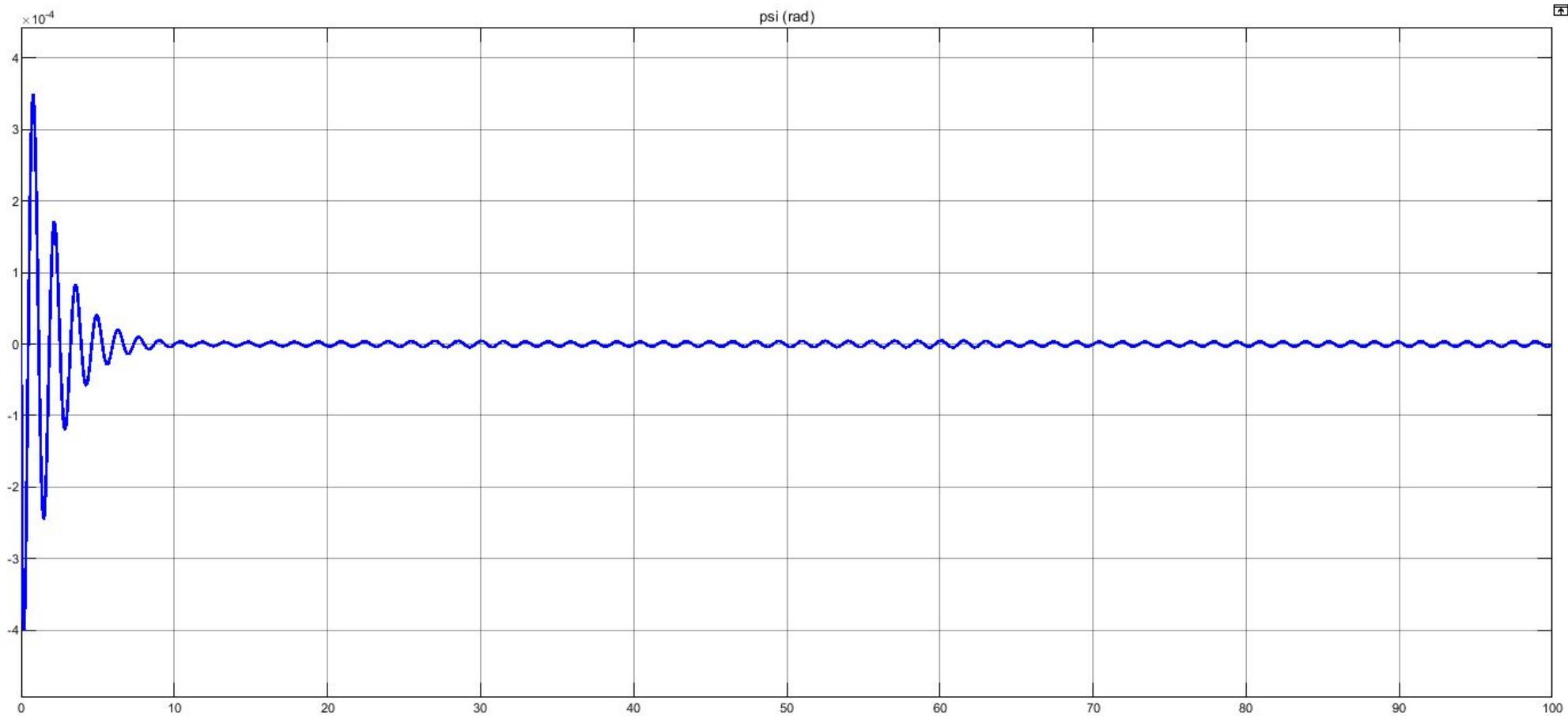
II. GA Results ~ Initial Disturbance Rejection



II. GA Results ~ Initial Disturbance Rejection



II. GA Results ~ Initial Disturbance Rejection



II. PID Controller



Discussion

*In both test scenarios; step response and disturbance rejection, the outer loop response (x & y) positions **improved** greatly and that was reflected in the performance criteria as shown above.

*As for altitude (z), the response did not settle at the setpoint desired exactly, but the error was in the range of 5% of the setpoint. Hence, the controller works best in **XY planar motion**. On the other hand, **difficulties** might be present in 3D motion.

II. PID Controller

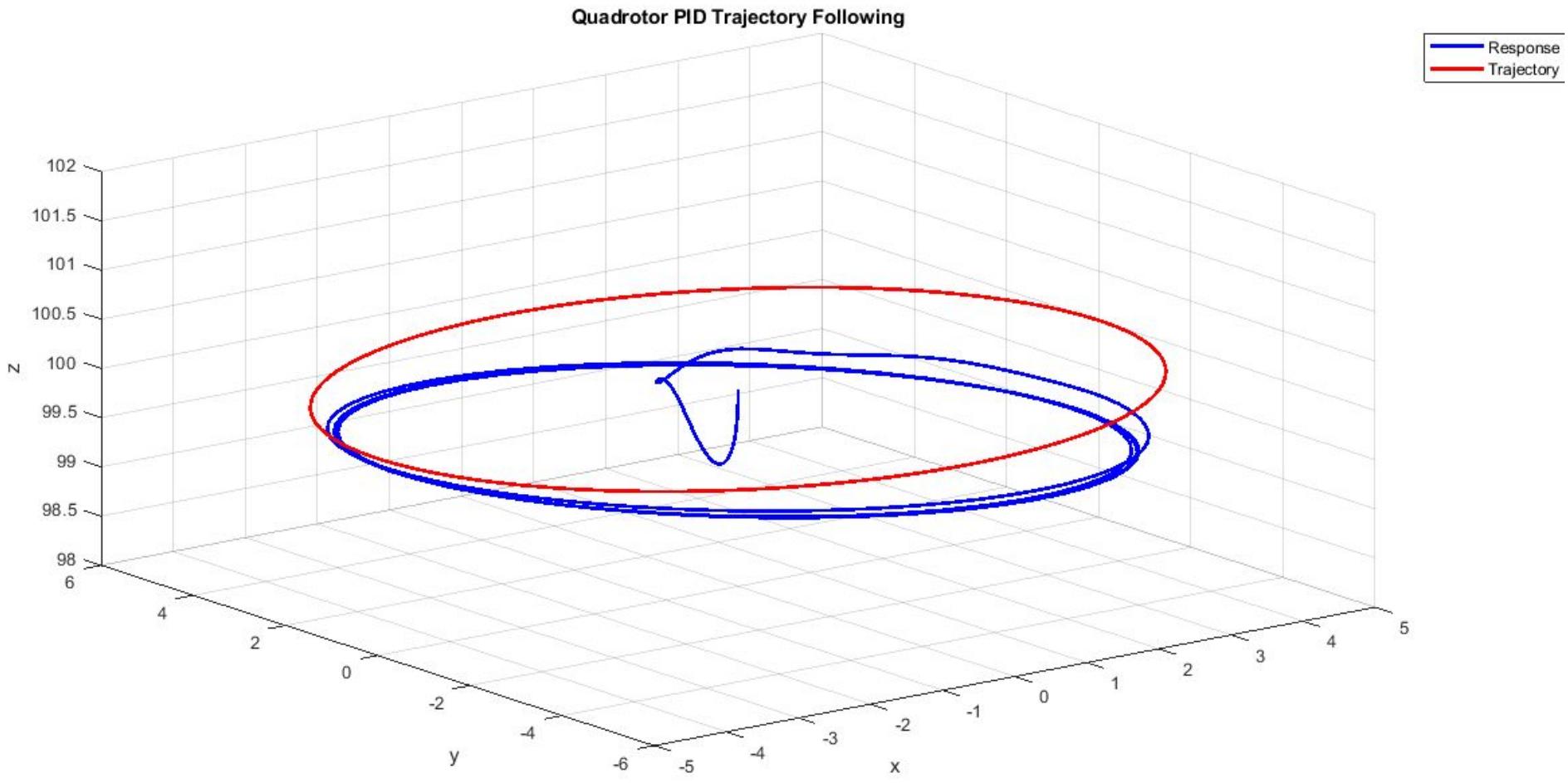


Performance Further Improvement

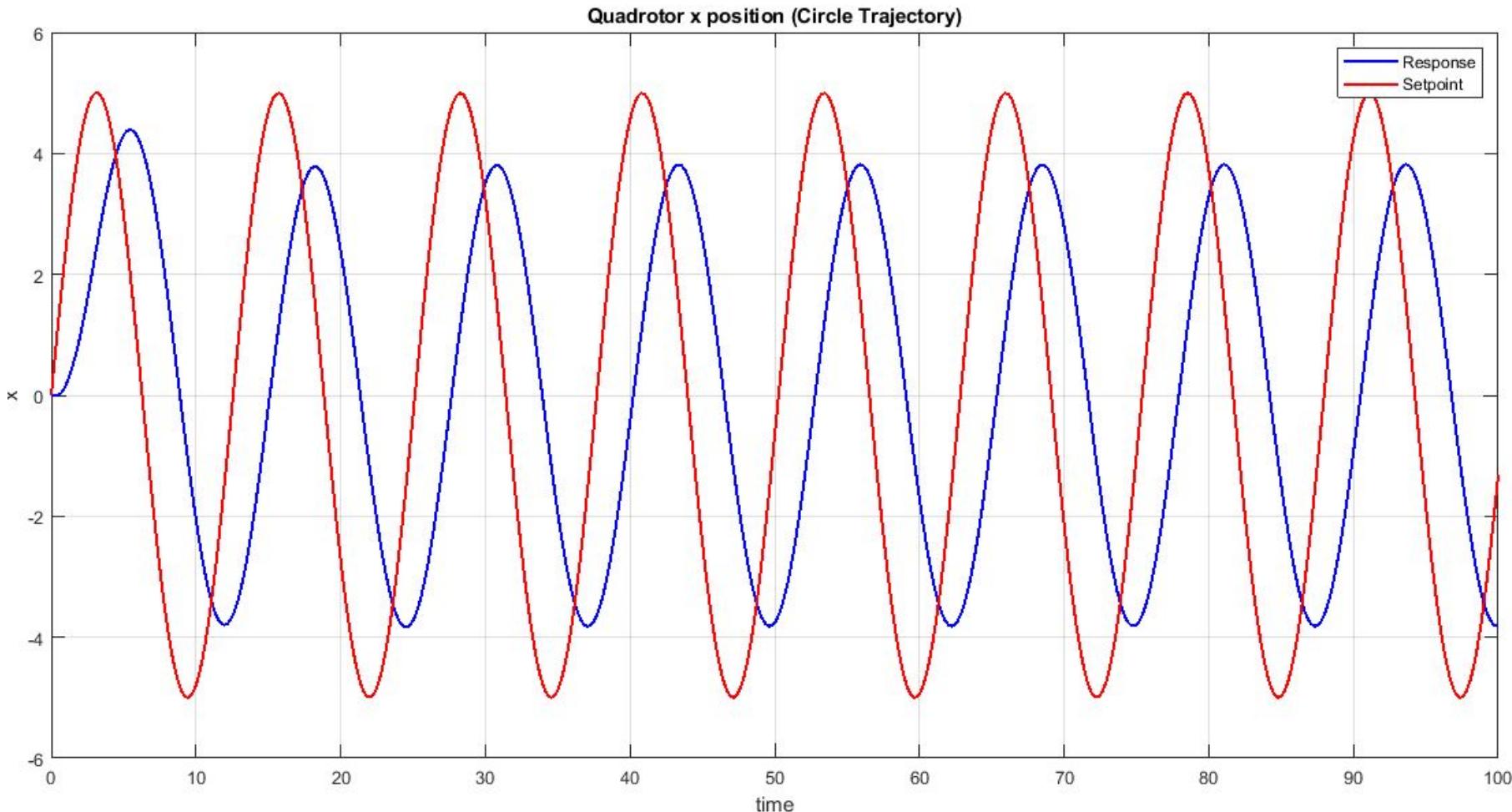
*The obtained results may be refined further to achieve better performance. This could be achieved by expanding the **search area** to include more potentially viable solutions. In addition, **optimization settings** could be adjusted to have more populations and generations. However, all of this will come as **computational overhead**.

*Another approach is using **hybrid algorithms** in MATLAB. Instead of just relying on Genetic Algorithm, the algorithm starts with GA and once it's finished, it passes the obtained value to another optimization method. Many methods are available including: fminsearch, patternsearch, and fminunc.

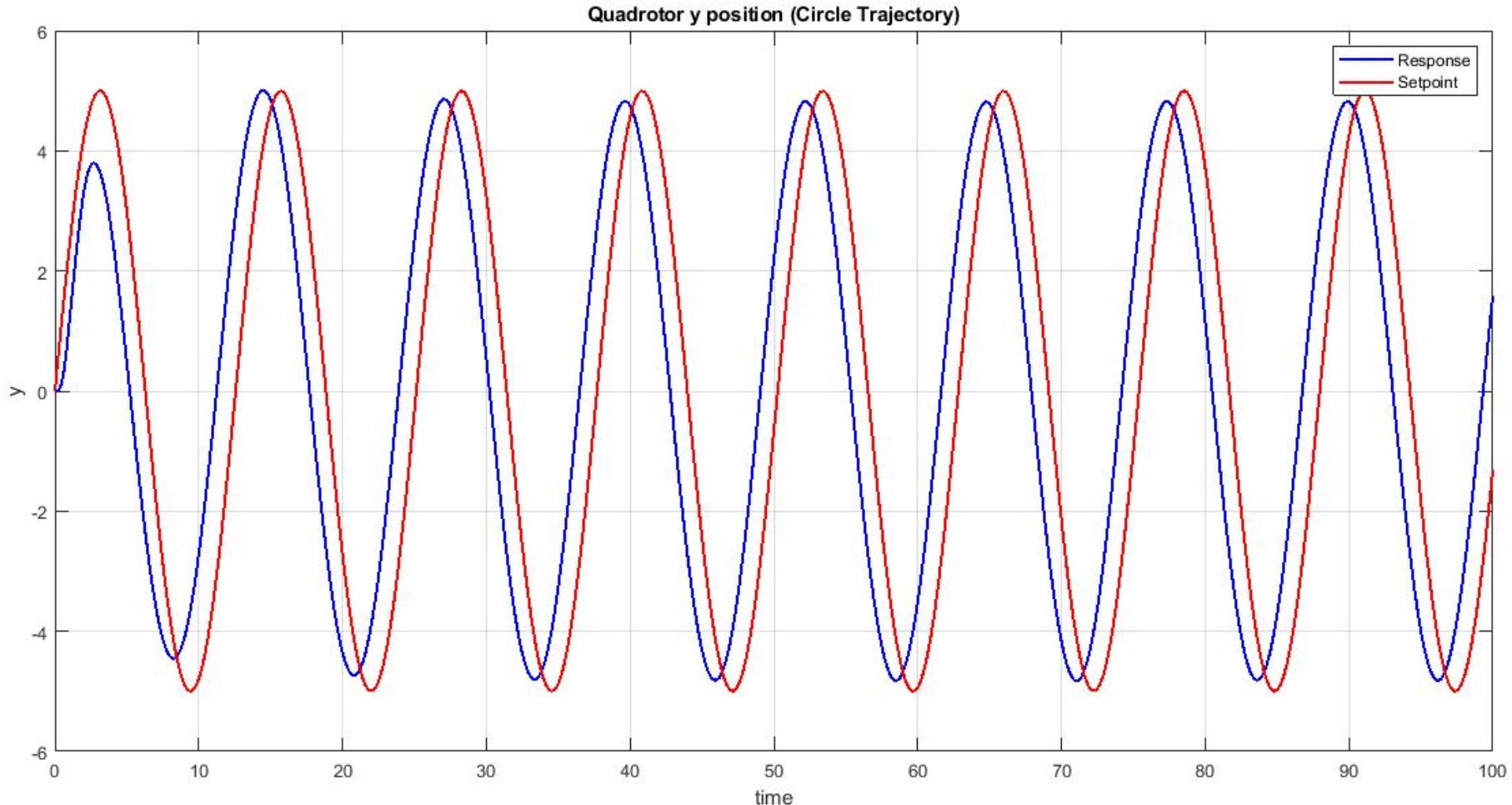
II. GA ~ Trajectory Tracking (Circle, R = 5 m)



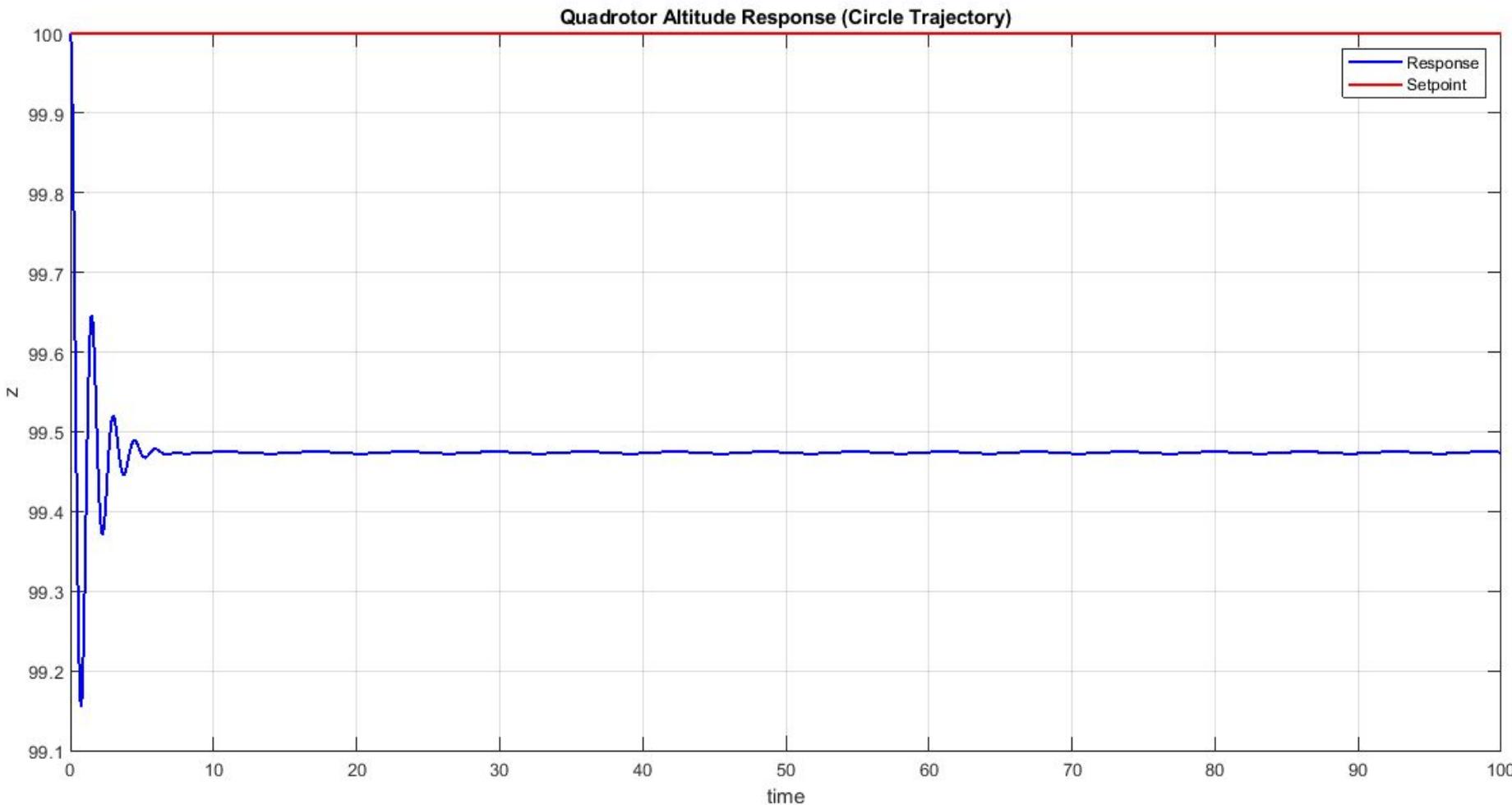
II. GA ~ Trajectory Tracking (Circle, R = 5 m)



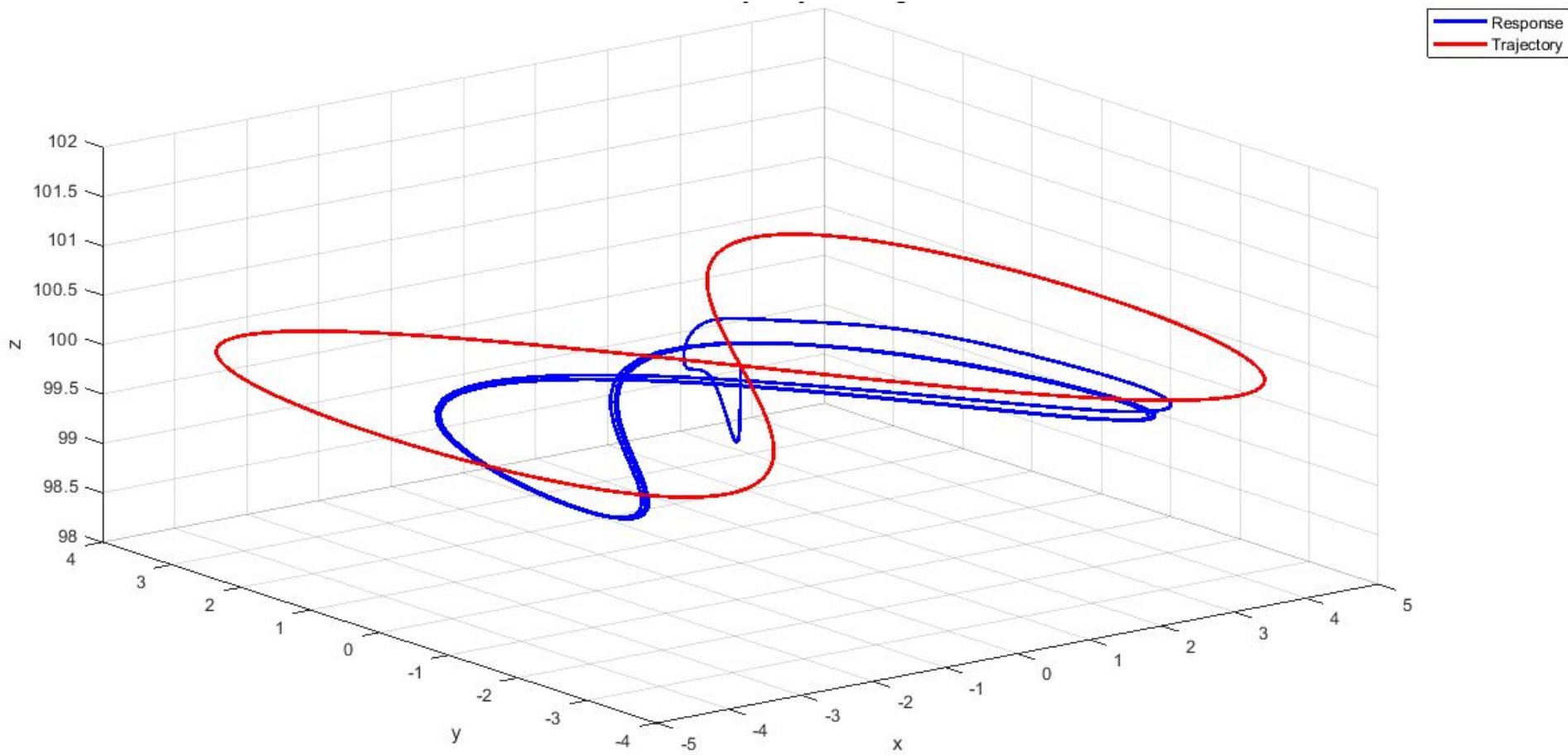
II. GA ~ Trajectory Tracking (Circle, R = 5 m)



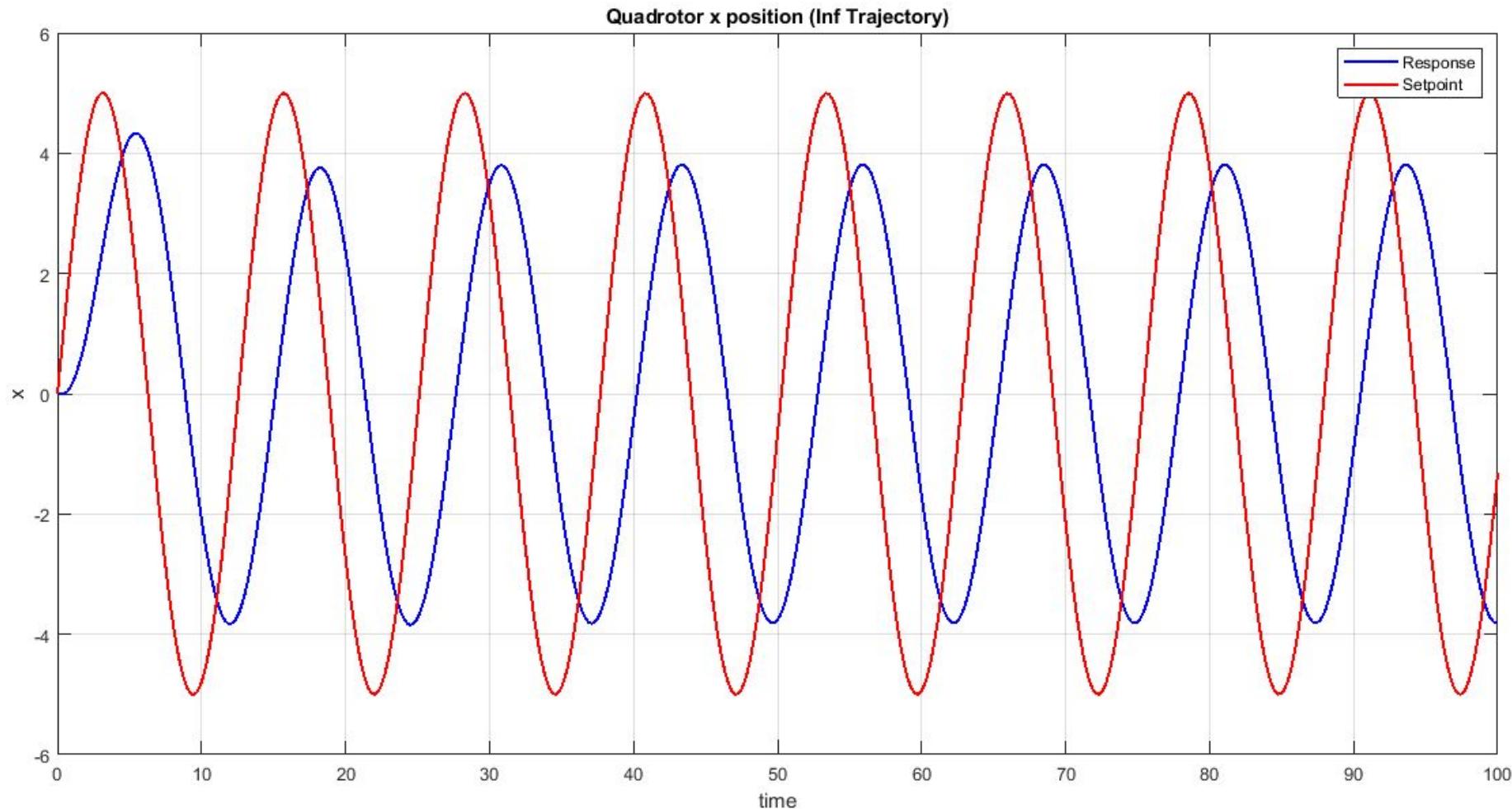
II. GA ~ Trajectory Tracking (Circle, R = 5 m)



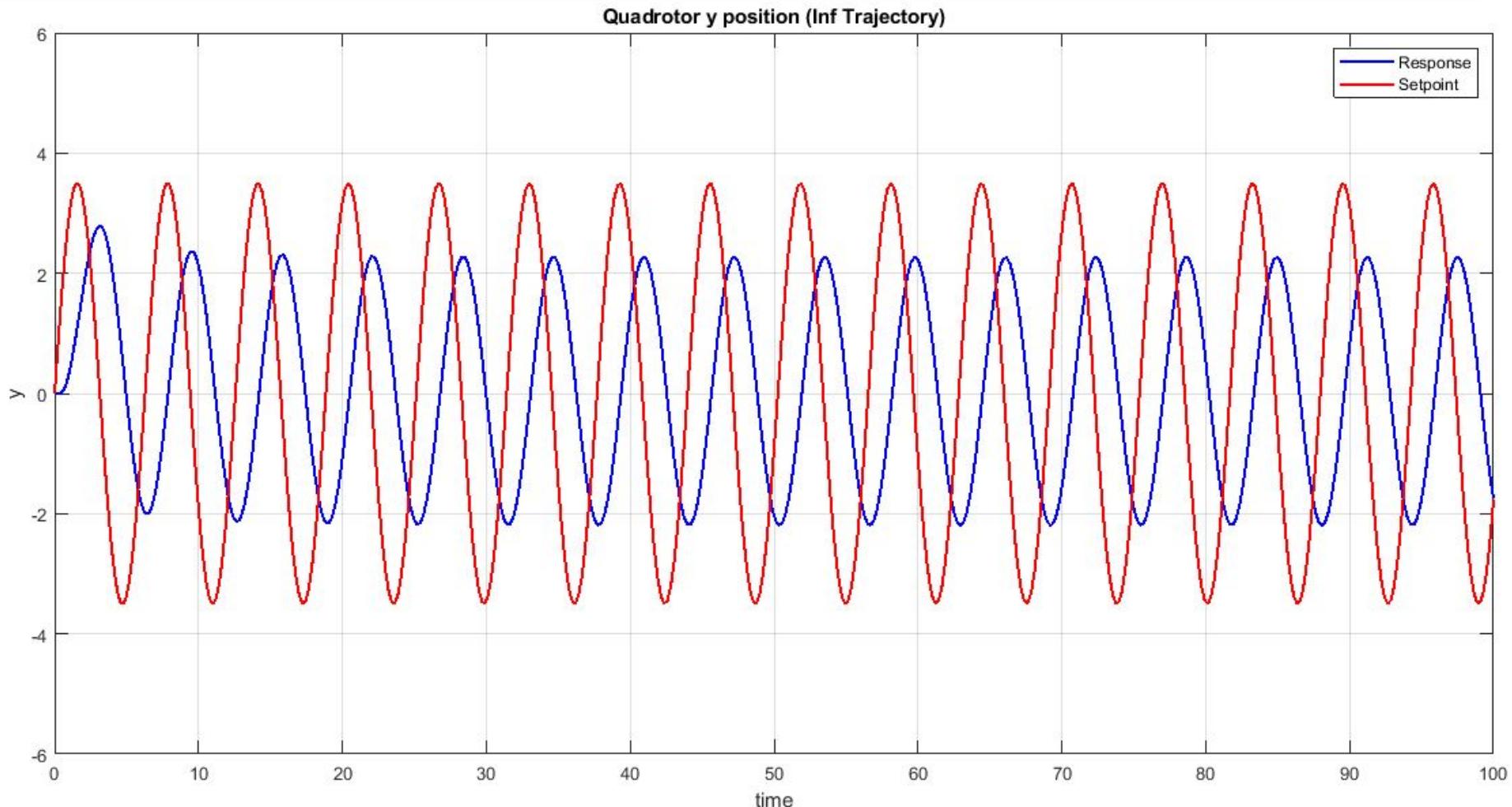
II. Trajectory Tracking (∞ , $5\sin(t)$, $3.5\sin(t/2)$)



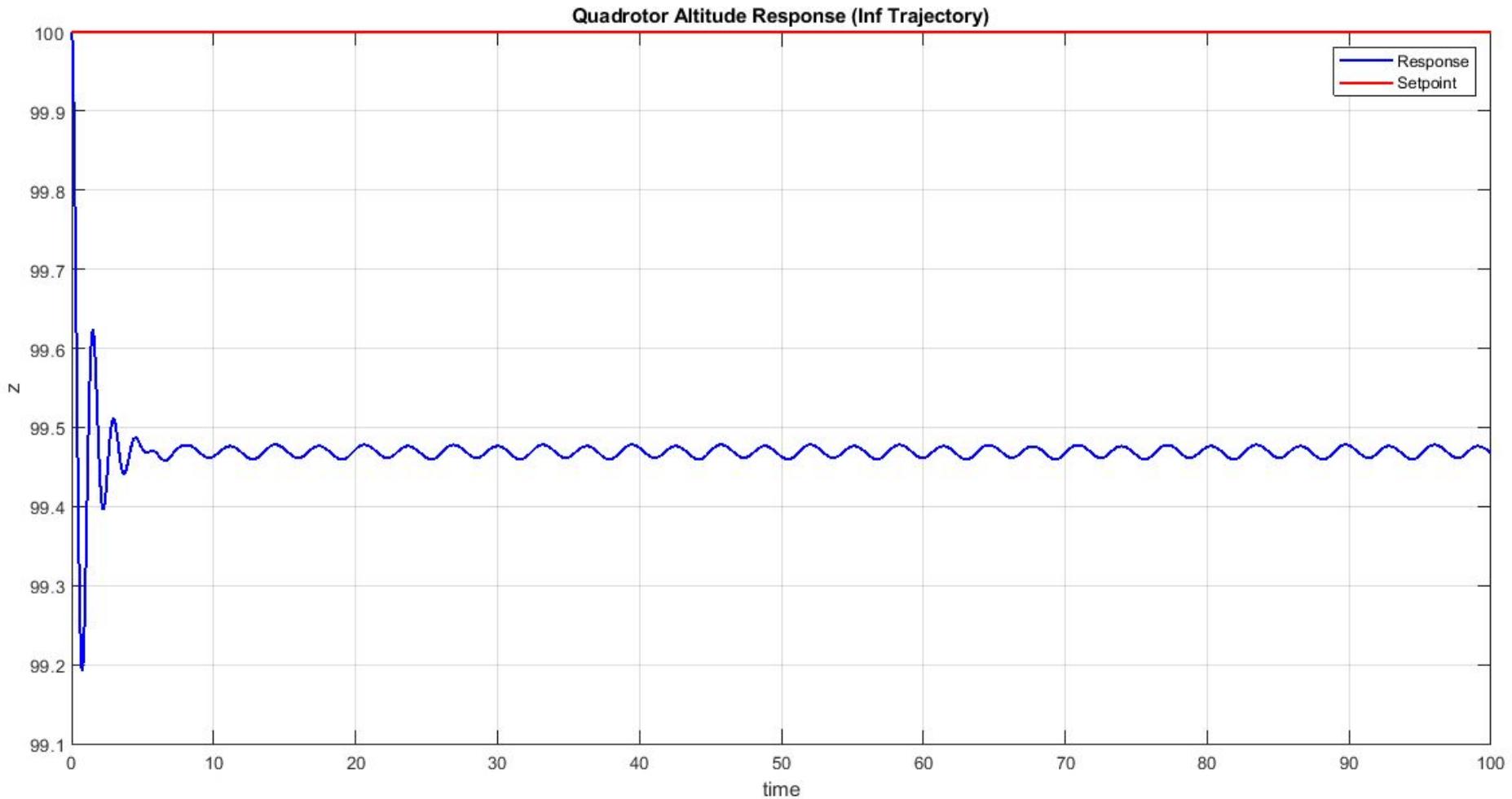
II. Trajectory Tracking (∞ , $5\sin(t)$, $3.5\sin(t/2)$)



II. Trajectory Tracking (∞ , $5\sin(t)$, $3.5\sin(t/2)$)



II. Trajectory Tracking (∞ , $5\sin(t)$, $3.5\sin(t/2)$)



II. PID Controller



Discussion

*The PID controller achieved **satisfactory** results tracking the **circle** trajectory, with errors in x, y, and z of 12%, 4%, and 0.5% respectively. The error in the x position could be reduced by further tuning of the controller so as to achieve better results.

*However, with more aggressive trajectories, the controller was unable to adjust effectively to the desired setpoints.

6. Conclusion and Future Work

I. Fixed Wing UAV



Most of the controllers have achieved stabilization, but they differ in implementation, computational power, and response specifications.

MPC better response with pitch angle disturbance, controlling simultaneously both longitudinal and lateral modes.

LQR tuning process is exhaustive.

FLC requires a high computational power, and the tuning process gets harder with the increase of input parameters

II. Quadrotor



Both controllers achieved stability, a high command of maneuverability, as well as disturbances immunity.

NMPC was better in reference tracking, doing sharp turns smoothly, but failed to fully recover quadrotors initial state when initial disturbances were introduced.

PID was better in fully rejecting disturbance and fully recover quadrotors initial state, ubt failed to perform reference tracking specially when sharp turns presented.



THANK YOU

www.zewailcity.edu.eg

