MODELLING AND NONLINEAR CONTROL OF A QUADCOPTER FOR STABILIZATION

IE 415 Control Of Autonomous System

Dishank Thakkar

Prof : Sujay D Kadam

INTRODUCTION

A quadcopter, also known as a quadrotor, is an unmanned aerial vehicle propelled by four rotors in cross-configuration. A quadcopter's dynamics are extremely nonlinear, it is an underactuated system with six degrees of freedom and four control inputs which are the rotor velocities.

With the growing number and complexity of quadcopter applications, it is crucial to enhance control techniques to ensure better performance and adaptability to diverse scenarios. This presentation focuses on modeling the quadcopter system and implementing nonlinear control methods to achieve stability and precise trajectory tracking.

OBJECTIVES OF THE STUDY

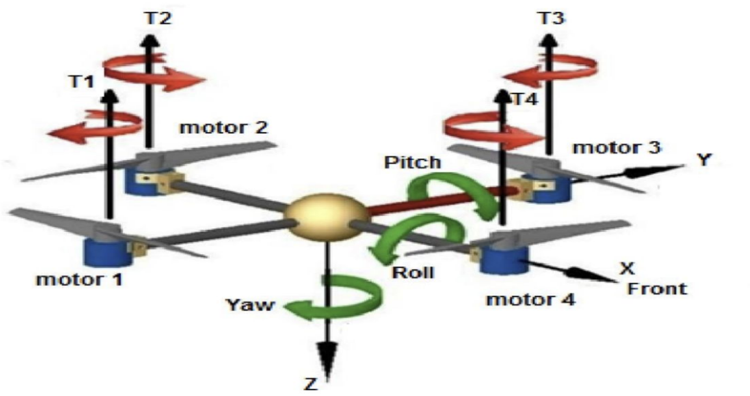
This project focuses on developing a mathematical model of a quadcopter system and applying nonlinear control techniques to achieve stabilization and trajectory tracking. The objectives include:

- Develop the mathematical model of a quadcopter system dynamics.
- Develop a PID control algorithm for the derived nonlinear quadcopter system dynamics.
- Simulate and perform an analysis of the implemented control technique on the quadcopter system for stabilization and trajectory tracking.

SYSTEM MODELLING

The quadcopter system is complex and in order to control it, the quadcopter is modelled on the following assumptions:

- The structure is rigid
- The structure is axis symmetrical
- The Centre of Gravity and the body fixed frame origin coincide
- The propellers are rigid
- Thrust and drag are proportional to the square of the propeller's speed



SYSTEM MODELLING

Rotational Dynamics

$$I\dot{\omega}B + [\omega B \times (I\omega B)] + \Gamma = \tau_B$$

$$\Gamma = J_r \omega_B \omega_r \; where \; \omega r = -\omega 1 + \omega 2 - \omega 3 + \omega 4$$

$$\tau_B = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l k_t(-\omega_2^2 + \omega_4^2) \\ l k_t(-\omega_1^2 + \omega_3^2) \\ k_b(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix}$$

SYSTEM MODELLING

Translational Dynamics

$$m\dot{v}_1 = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT_B + F_D$$

$$F_D = -k_d\, v_l$$

$$T = \sum_{i=1}^{4} F_i = k_t \sum_{i=1}^{4} \omega_i^2$$

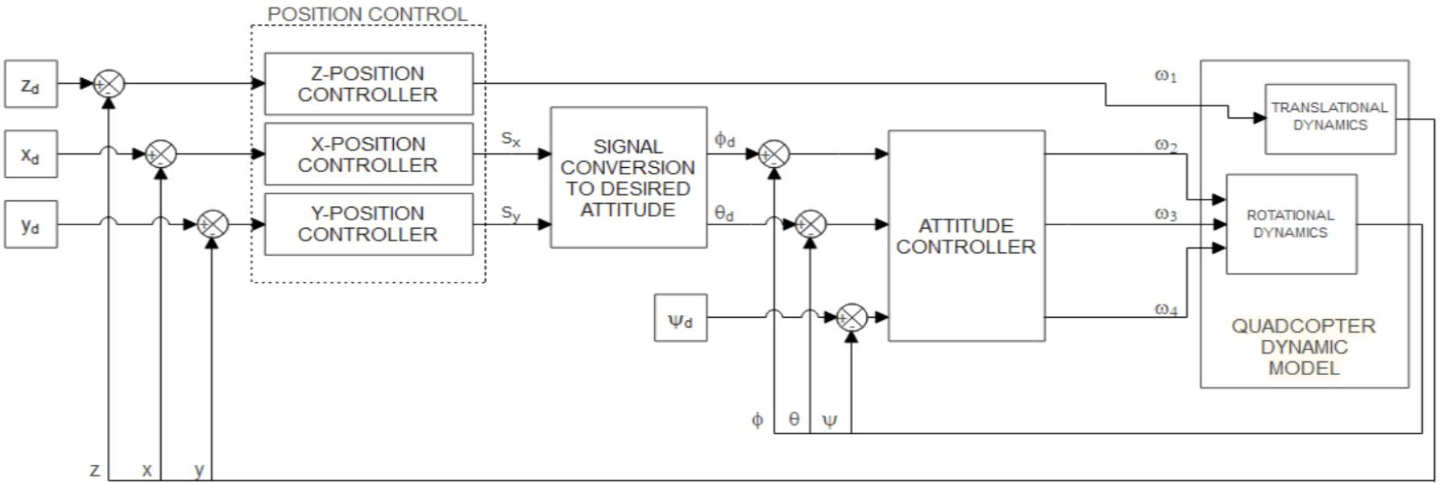
State Space

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\frac{(I_{yy}-I_{zz})qr}{I_{xx}}\\ \frac{(I_{zz}-I_{xx})pr}{I_{yy}}\\ \frac{(I_{xx}-I_{yy})pq}{I_{zz}}\end{bmatrix} - J_r\begin{bmatrix}\frac{q}{I_{xx}}\\ \frac{-p}{I_{yy}}\\ 0\end{bmatrix}\omega_r + \begin{bmatrix}\frac{\tau_\phi}{I_{xx}}\\ \frac{\tau_\theta}{I_{yy}}\\ \frac{\tau_\psi}{I_{zz}}\end{bmatrix}$$

For Linear Dynamics:-

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = -g\begin{bmatrix}0\\0\\1\end{bmatrix} + \frac{T}{m}\begin{bmatrix}c(\psi)s(\theta)c(\phi)+s(\psi)s(\phi)\\ s(\psi)s(\theta)c(\phi)+c(\psi)s(\phi)\\ c(\theta)c(\phi)\end{bmatrix} - \frac{1}{m}\begin{bmatrix}k_{dx} & 0 & 0\\ 0 & k_{dy} & 0\\ 0 & 0 & k_{dz}\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}$$

CONTROL ARCHITECTURE AND DEVELOPMENT



CONTROL ARCHITECTURE AND DEVELOPMENT

PID CONTROL

$$e(t) = r(t) – y(t)$$

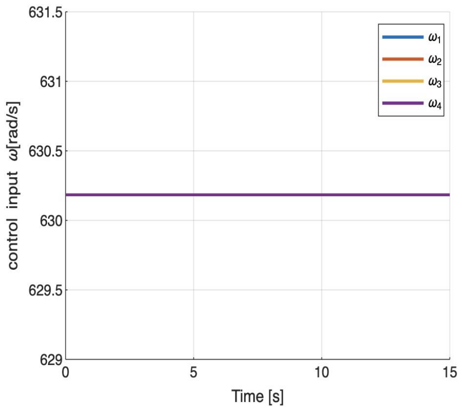
$$u(t) = K_p e(t) + K_I \int e(\tau)d\tau + K_D d/dt(e(t))$$

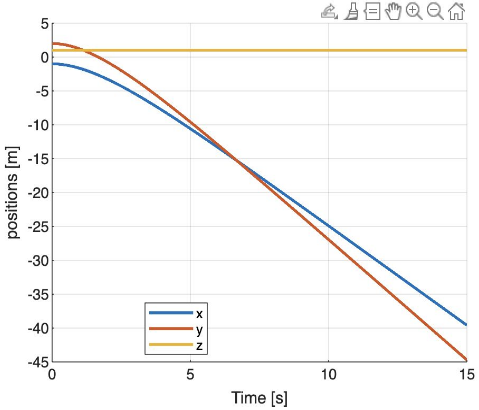
$$\tau = I \times u(t)$$

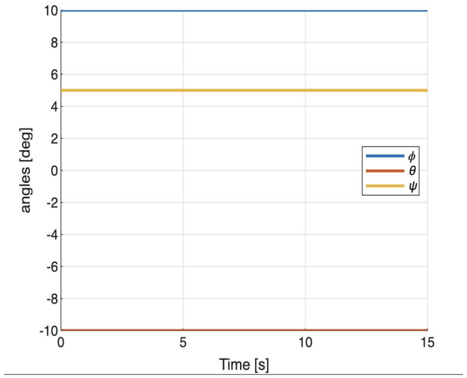
$$\begin{bmatrix}\tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix}=\begin{bmatrix}I_{xx}\left(k_{P\phi}(\phi_{des}-\phi)+k_{D\phi}(\dot{\phi}_{des}-\dot{\phi})\right)\\ I_{yy}\left(k_{P\theta}(\theta_{des}-\theta)+k_{D\theta}(\dot{\theta}_{des}-\dot{\theta})\right)\\ I_{zz}\left(k_{P\psi}(\psi_{des}-\psi)+k_{D\psi}(\dot{\psi}_{des}-\dot{\psi})\right)\end{bmatrix}$$

$$\tau_B=\begin{bmatrix}\tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix}=\begin{bmatrix}lk_t(-\omega_2^2+\omega_4^2)\\ lk_t(-\omega_1^2+\omega_3^2)\\ k_b(-\omega_1^2+\omega_2^2-\omega_3^2+\omega_4^2)\end{bmatrix}$$

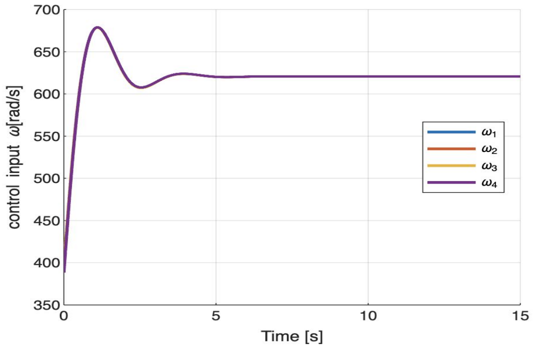
Simulations and Results

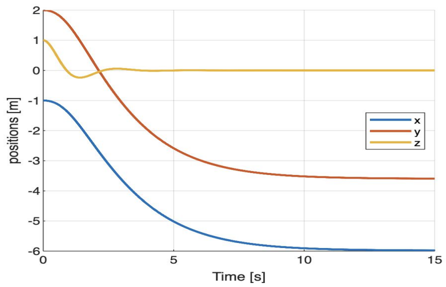


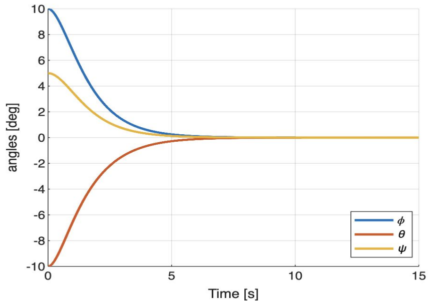




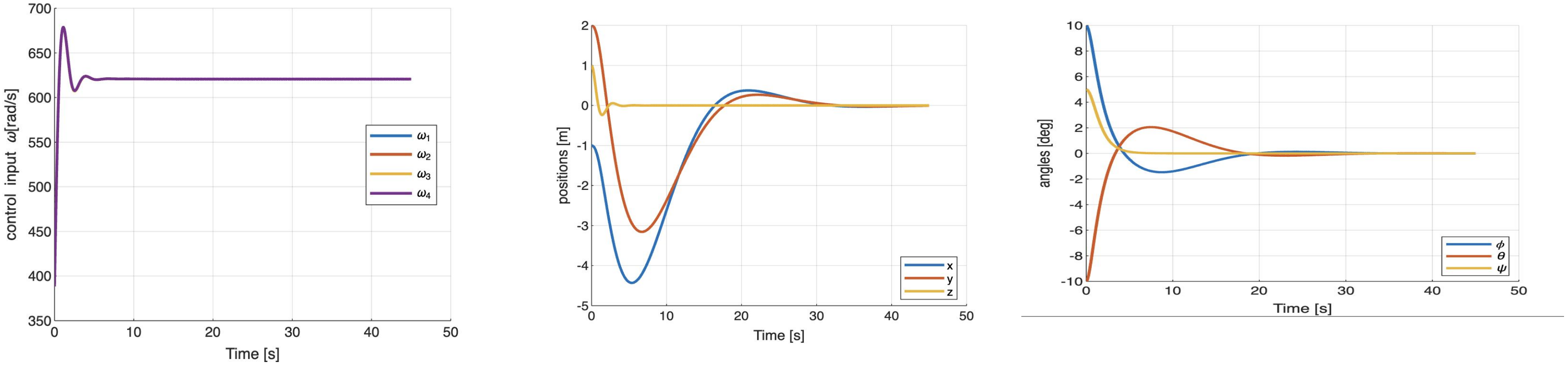
Simulation without PD Control



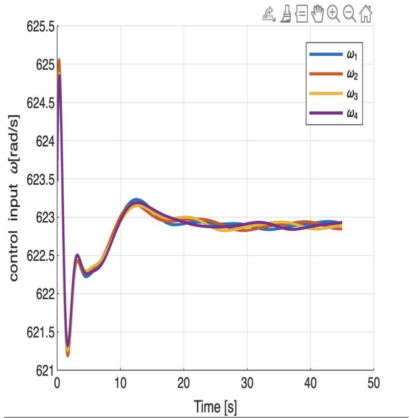


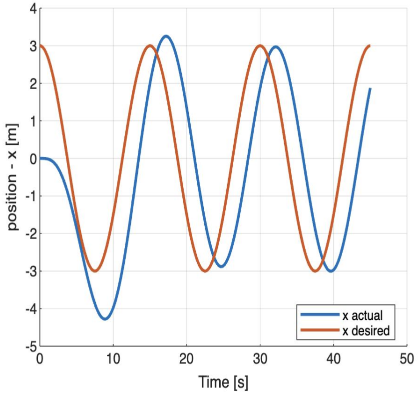


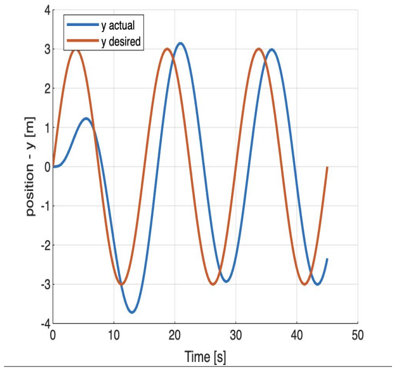
Simulation with PD Control

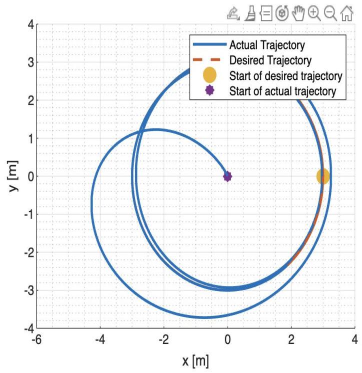


Stabilization with Zero Mapping of Position States

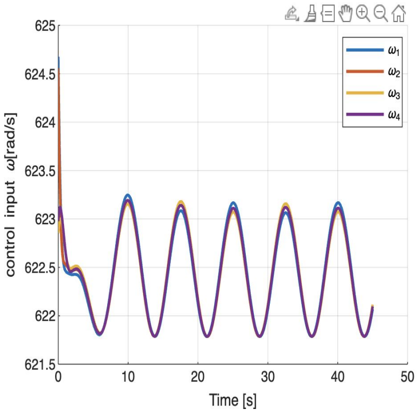


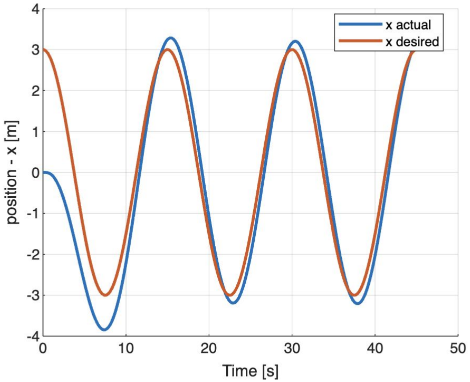


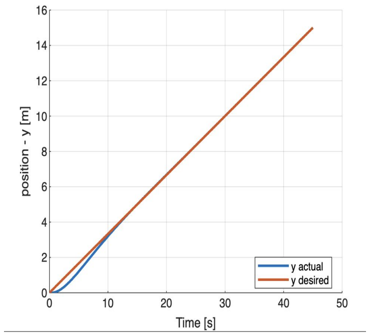


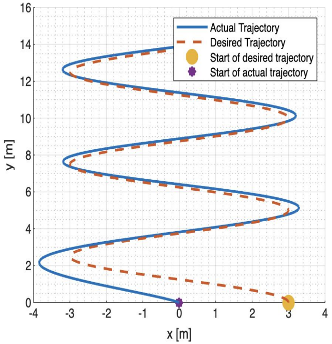


Trajectory Tracking with PD Control (Circular Trajectory)









Trajectory Tracking with PD Control (Spiral Trajectory)

Thank You