Strathmore
UNIVERSITY

SCHOOL OF COMPUTING AND ENGINEERING SCIENCES
BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONICS ENGINEERING
2nd SEMESTER 2025/2026
**BEE 5214: Multivariate Control Lab_4**

DATE: February, 2026

## 1. Title:

**Simulation and practical implementation of CKF + PI Control of a Quadcopter (X Configuration)**

## 2. Objective:

2.1 Implement the quadcopter digitized EOM and allocator for quadcopter X configuration.

2.2 Use fixed step forward Euler to simulate CKF + PI controller for (i) altitude-only hover and (ii) cascaded altitude + attitude for the quadcopter.

2.3 Implement fixed step forward Euler CKF + PI controller for (i) altitude-only hover and (ii) cascaded altitude + attitude for the quadcopter.

## 3. Learning Outcomes:

By completing this laboratory, students will be able to:

3.1 Build body/world dynamics and the mixer $A_X$ from geometry.

3.2 Assemble an X quadcopter

3.3 Simulate attitude CKF + PI loops with anti-windup and saturation

3.4 Simulate attitude and altitude CKF + PI loops with anti-windup and saturation

3.5 Understand bandwidth separation between inner attitude and outer altitude loops for the quadcopter.

3.6 Implement attitude CKF + PI loops with anti-windup and saturation

## 4.0 Homework

Print the quadcopter motor mounting chassis and arms and assemble the quadcopter as shown in Fig. 4.2.

**Quadcopter (X)**

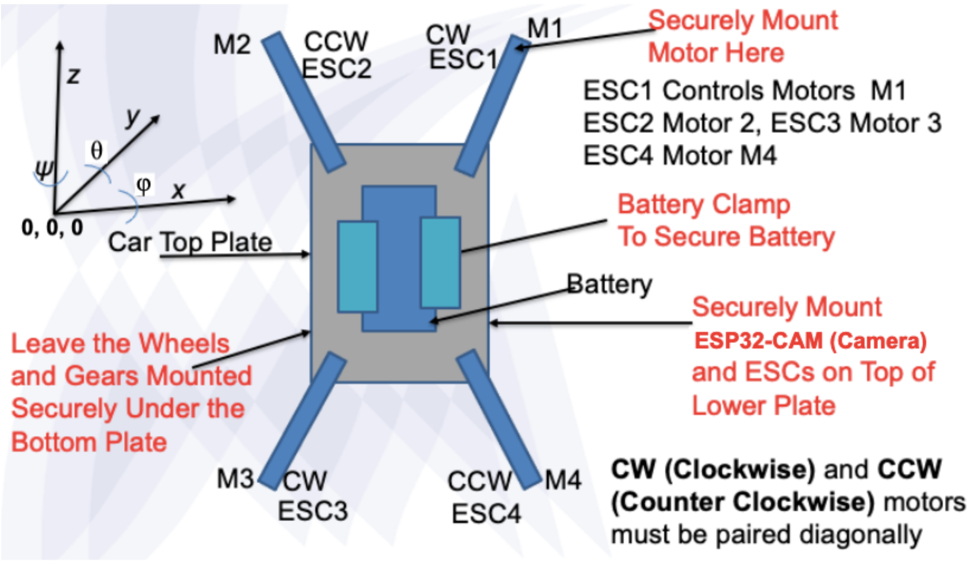


Fig. 4.1 Quadcopter geometry

Fig. 4.2 X Quadcopter assembly

**60 Marks**

## 4.1 X Quadcopter EOM

Flying Car (Drone) X Configuration

**Body Frame - Refer to Fig. 4.1**

- $x$ → East     $y \rightarrow$ North     $z \rightarrow$ Up
- Roll about $x$     Pitch about $y$     Yaw about $z$

**Rigid body states**

Position (3) $[x, y, z]$     Velocity (3) $[v_x, v_y, v_z]$

Motor

**Motor Input Voltages (4)** $[V_1, V_2, V_3, V_4]$ **Motor Currents (4)** $[i_1, i_2, i_3, i_4]$

**Motor Speeds (4)** $[\omega_1, \omega_2, \omega_3, \omega_4]$

Flying Car (Drone) Model Parameters and Difference Equations
Drone mass M
Motor constants $J_m$ $b_m$, $k_t$, $k_b$, L, R, τ
Drone angular momentum in x, y, z reference frame $I_x$ (roll) $I_y$ (pitch) $I_z$ (Yaw)
Drone velocity damping coefficeints in x, y, z reference frame $k_{dx}$ , $k_{dy}$ , $k_{dz}$
Drone propeller viscous damping coefficients $k_\theta$, $k_\varphi$, $k_\psi$
The drone assembly shown in Fig. 4.2 may have upto 24 state variables;

(i) Drone position and velocity in x, y, z coordinates - 6 state variables,
(ii) Drone angular position and speed in x, y and z axis – 6 state variables
(iii) Drone motor input voltage, current, propeller speed
- 3 state variables for each motor making a total of 12 state variables for 4 motors.

| **Motor $M_1$ /Propeller Parameters** | |
|---|---|
| $R_i$ Winding resstance | 0.1Ω |
| $L_i$ Winding inductance | 0.002H |
| $k_{bi}$ Back EMF Constant | 0.01 Vs/rad |
| $k_{ti}$ Torque Constant | 0.01 N/A |
| $b_{mi}$ Motor damping coefficient. | 0.00009Ns/rad |
| $J_{mi}$ Motor/Propeller inertia | 0.0004 kgm$^2$ |
| $k_{Qi}$ Propeller aerodynamic drag coefficient = $C_{Q1}\rho A_1 R_1^3$ | 0.000002 Ns$^2$/rad$^2$ |
| $k_{Ti}$ Propeller trust coefficient = $C_{Ti}\rho A_i R_i^2$ | 0.000018 Ns$^2$/rad$^2$ |
| $k_{dx} = k_{dy}$ Velocity damping | 0.15 Ns/m |
| $k_{dz}$ Velocity damping | 0.25 Ns/m |
| $k_\theta = k_\varphi$ Viscous damping | 0.002 Ns/rad |
| $k_\psi$ Viscous damping | 0.003 |

| **Motor $M_1$ /Propeller Parameters** | |
|---|---|
| $C_{Ti}$ Propeller thrust constant | |
| $C_{Qi}$ Propeller drag constant | |
| $R_i$ Propeller radius | |
| $A_i = \pi R_i^2$ = Propeller disc area | |
| ρ Air density | |
| $l$ Drone arm | 0.1m |
| Varibles | |
| $i_1$ Motor current. | 20A (Max) |
| $\omega_i$ Propeller speed | |
| $V_i$ Input voltage | 12V (Max) |
| m Drone mass | 1.5 kg |
| $I_x = I_y$ | 0.02 kgm$^2$ |
| $I_z$ Angular momentum | 0.04 kgm$^2$ |
| $T_s$ Sampling time | 0.004s |
| g Gravity acceleration | 9.81 ms$^{-2}$ |

**Motor Electrical Dynamics**

$$L_i \frac{di_i}{dt} = -R_i i_i - k_{bi}\omega_i + V_i$$

**Motor Mechanical Dynamics**

$$J_{mi}\dot{\omega}_i = k_{ti} i_i - b_{mi}\omega_i - k_{Qi}\omega_i^2$$

**Thrust and Torques**

$$T = \sum_{i=1}^{4} k_{Ti}\omega_i^2$$

X-configuration torques:

$$\tau_\phi = \frac{l}{\sqrt{2}}(k_{T2}\omega_2^2 + k_{T3}\omega_3^2 - k_{T1}\omega_1^2 - k_{T4}\omega_4^2)$$
$$\tau_\theta = \frac{l}{\sqrt{2}}(k_{T1}\omega_1^2 + k_{T2}\omega_2^2 - k_{T3}\omega_3^2 - k_{T2}\omega_4^2)$$
$$\tau_\psi = k_{Q1}\omega_1^2 - k_{Q2}\omega_2^2 + k_{Q3}\omega_3^2 - k_{Q4}\omega_4^2$$

**Translational Dynamics** $\dot{x} = v_x$ $\dot{y} = v_y$ $\dot{z} = v_z$

$$m\dot{v}_x = T(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - k_{dx}v_x$$
$$m\dot{v}_y = T(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) - k_{dy}v_y$$
$$m\dot{v}_z = T(\cos\phi\cos\theta) - mg - k_{dz}v_z$$

**Rotational Dynamics (gyroscopic)**

$$I_x\dot{p} = \tau_\phi + (I_y - I_z)qr \text{ - } k_\varphi p$$
$$I_y\dot{q} = \tau_\theta + (I_z - I_x)pr \text{ - } k_\theta q$$
$$I_z\dot{r} = \tau_\psi + (I_x - I_y)pq - k_\psi r$$

**Euler Kinematics**

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$

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

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

Hover equilibrium: $\phi = \theta = 0$ $p = q = r = 0$ $v_x = v_y = v_z = 0.$ $\omega_i = \omega_h$(hover speed)

Hover condition: $4k_T\omega_h^2 = mg$ $\omega_h = \frac{1}{2}\sqrt{\frac{mg}{k_T}}$

**State vector:**

$$X = [i_1,\ i_2, i_3, i_4, \omega_1, \omega_2, \omega_3, \omega_4,\ x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r]^T$$

**Control inputs (Motor voltages):**

$$U = [V_1,\ V_2,\ V_2,\ V_2]^T$$

## 4.2 Forward Euler Discretization

Motor Dynamics

**Electrical:**

$$i_{i,k+1} = i_{i,k} + \frac{T_s}{L_i}(-R_i i_{i,k} - k_{bi}\omega_{i,k} + V_{i,k})$$

**Mechanical:**

$$\omega_{i,k+1} = \omega_{i,k} + \frac{T_s}{J_{mi}}(k_{ti} i_{i,k} - b_{mi}\omega_{i,k} - k_{Qi}\omega_{i,k}^2)$$

**Translational Dynamics**

$$v_{x,k+1} = v_{x,k} + \frac{T_s}{m}[T_k(\cos\phi_k\sin\theta_k\cos\psi_k + \sin\phi_k\sin\psi_k) - k_{dx}v_{x,k}]$$
$$v_{y,k+1} = v_{y,k} + \frac{T_s}{m}[T_k(\cos\phi_k\sin\theta_k\sin\psi_k - \sin\phi_k\cos\psi_k) - k_{dy}v_{y,k}]$$
$$v_{z,k+1} = v_{z,k} + \frac{T_s}{m}[T_k(\cos\phi_k\cos\theta_k) - mg - k_{dz}v_{z,k}]$$

**Position update:**

$$x_{k+1} = x_k + T_s v_{x,k},\quad y_{k+1} = y_k + T_s v_{y,k},\quad z_{k+1} = z_k + T_s v_{z,k}$$

**Body Rotational Dynamics**

$$p_{k+1} = p_k + \frac{T_s}{I_x}[\tau_{\phi,k} + (I_y - I_z)q_k r_k] \text{ - } k_\varphi\, p_k$$
$$q_{k+1} = q_k + \frac{T_s}{I_y}[\tau_{\theta,k} + (I_z - I_x)p_k r_k] \text{ - } k_\theta\, q_k$$
$$r_{k+1} = r_k + \frac{T_s}{I_z}[\tau_{\psi,k} + (I_x - I_y)p_k q_k] \text{ - } k_\psi\, r_k$$

**Euler Angle Kinematics**

$$\phi_{k+1} = \phi_k + T_s[p_k + q_k \sin\phi_k \tan\theta_k + r_k \cos\phi_k \tan\theta_k]$$

$$\theta_{k+1} = \theta_k + T_s[q_k \cos\phi_k - r \sin\phi_k]$$

$$\psi_{k+1} = \psi_k + T_s \left[\frac{q_k \sin\phi_k + r \cos\phi_k}{\cos\theta_k}\right]$$

## 4.3 Write Python code to simulate and plot CKF-PI Altitude/Hover

Initial Conditions:

All states initialized to zero

Motors initialized to hover speed:

$$\omega_h = \frac{1}{2}\sqrt{\frac{mg}{k_T}}$$

$\omega_h$ > 450 rad/s

Total Thrust

$$T_k = \sum_{i=1}^{4} k_T\, \omega_{i,k}^2$$

**PI Attitude Control**

$$u_\phi = K_{p\phi} e_\phi + K_{i\phi} \int e_\phi dt$$

$$u_\theta = K_{p\theta} e_\theta + K_{i\theta} \int e_\theta dt$$

$$u_\psi = K_{p\psi} e_\psi + K_{i\psi} \int e_\psi dt$$

Where:

$$e_\phi = \phi_{ref} - \hat{\phi}$$

CKF-estimated states are used.

**Motor Voltage Mixing (X Configuration)**

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = V_{hover} + M^{-1} \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \\ 0 \end{bmatrix}$$

**Cubature Kalman Filter (CKF)**

**State Model**

$$x_{k+1} = f(x_k, u_k) + w_k$$

**Measurement Model**

$$y_k = [z \quad \phi \quad \theta \quad \psi]^T + v_k$$

**CKF Algorithm**

For $n$ states:

(i) Generate $2n$ cubature points:

$$\xi_i = \sqrt{n}\, e_i$$

(iii) Time update:

- Propagate cubature points through nonlinear dynamics.
- Compute predicted mean and covariance.

(iv) Measurement update:

(v) Propagate cubature points through measurement equation.

(vi) Compute Kalman gain.

**Required Plots**

**(i)** Motor voltages $V_1$–$V_4$

**(ii)** Motor currents $i_1$–$i_4$

**(iii)** Propeller speeds $\omega_1$–$\omega_4$

**(iv)** Height $z(t)$

(v) Roll, pitch, yaw response

**40 Marks**

**4.4 Write Python to simulate CKF + PI Attitude + Altitude Control with velocity ($v_x$)**

**Altitude PI Controller**

$$u_z = K_{pz} e_z + K_{iz} \int e_z dt$$

$$e_z = z_{ref} - \hat{z}$$

**Thrust Command**

$$T_{cmd} = mg + u_z$$

Desired hover speed:

$$\omega_{cmd} = \sqrt{\frac{T_{cmd}}{4k_T}}$$

**Forward Velocity Generation**

$$\theta_{ref} = K_v\left(v_{x,ref} - \hat{v}_x\right)$$

Cascaded control structure:

Altitude → Thrust → Velocity → Pitch → Torque → Motor Voltages

**Required Plots**

(i) Motor voltages

(ii) Motor currents

(iii) Propeller speeds

(iv) Height $z(t)$

(v) Forward velocity $v_x(t)$

(vi) Pitch angle response

**Tune attitude and altitude PI controller gains to achieve**

| Performance Metric | Requirement |
|---|---|
| Height overshoot | $< 10\%$ |
| Settling time | $<$ 3 s |
| Forward velocity tracking | $<$ 5% error |
| Attitude steady state error | $\approx 0$ |

**50 Marks**

# 5 Practical

## 5.1 Implement the CKF-PI Altitude/Hover Controller

Convert the and run the Python code written in 4.3 with actual propeller speed measurement on the drone assembled in 4.0 and shown in Fig. 4.2

Take a video of your drone hovering.

40 Marks

## 6. Lab Report

20 Marks

6.1 Use these guidelines
https://drive.google.com/drive/u/2/folders/11saKDjuLBwEa5uQg5ynakZ3JinBQyGKl
to write and upload your individual lab report detailing your journey through items 1 to 6.

### 6.2.1 Questions

(i) Why is CKF preferred over EKF for this system

(ii) What happens if motor dynamics are ignored?

(iii) How does Euler discretization affect stability?

(iv) What happens if sampling time increases?

**References**


1. S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed., Wiley.
2. K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, Princet