



Shahid Chamran  
University of Ahvaz

# Journal of Applied and Computational Mechanics



## Research Paper

# Lyapunov-Based Tracking Control of a Bi-Rotor

Mohammad Ganji-Nahoji<sup>✉</sup>, Ali Keymasi-Khalaji<sup>✉</sup>

Department of Mechanical Engineering, Faculty of Engineering, Kharazmi University, Tehran, P.O. Box 15719-14911, Iran,  
Email: mohammadganji18@gmail.com (M.G.N.); keymasi@khu.ac.ir (A.K.K.)

Received December 13 2023; Revised February 27 2024; Accepted for publication February 27 2024.

Corresponding author: A. Keymasi-Khalaji (keymasi@khu.ac.ir)

© 2024 Published by Shahid Chamran University of Ahvaz

**Abstract.** This paper discusses the modeling and trajectory tracking control of a Bi-rotor. A novel class of vertical flight robots that can perform vertical takeoff, landing, and passenger transportation. Bi-rotor aircraft utilize a simplified mechanism compared to helicopters while maintaining the ability to perform complex maneuvers. With six degrees of freedom and four actuators, including two tilt actuators for steering and two propellers for thrust generation, they are classified as underactuated systems. The trajectory tracking controller employs a combination of feedback linearization and backstepping control methods, with an inner loop controlling the Euler angles and an outer loop regulating the Bi-rotor position and calculating desired angles for trajectory tracking. Control algorithms in the limited existing literature often rely on simplified mathematical models, which tend to overlook crucial nonlinear coupling terms. However, neglecting these terms can have significant implications for the dynamic behavior of the system. The dynamic modeling of the Bi-rotor aircraft was validated using the ADAMS software and integrated with the Simulink environment in MATLAB software. The obtained results represent the effectiveness of the proposed algorithm for the control of the Bi-rotor.

**Keywords:** Bi-rotor aircraft; Backstepping control; Feedback linearization; Tilt actuators; ADAMS software.

## 1. Introduction

In recent years, there has been a significant increase in the importance of highly maneuverable vertical flight robots, particularly in air transportation in confined spaces [1]. Advancements in science and technology have led to the utilization of these robots not only for entertainment and filming purposes but also for tasks such as postal delivery, food transportation, providing first aid to injured people [2], deploying life-saving tubes for drowning individuals, locating hiding criminals during pursuits, and military applications [3]. However, these advancements require air vehicles to be more compact for a given payload, making it increasingly challenging to maintain proper control as their size decreases [4].

There are various classes of UAVs capable of carrying different types of payloads, yet despite these advancements, there are still numerous opportunities for further development in this field [5]. Multi-rotor aircraft have attracted considerable attention with their original structure and arrangement, and the most common ones are currently four, six, and eight rotor aircraft, which must have different propeller speeds to move [6]. With the advancement of configurations that allow for high maneuverability to easily pass through limited space environments while effectively reducing the unit time power demand, a Bi-rotor, whose dynamic system was capable of tilting, was presented [7]. Sanchez et al. [8] proposed one of the first studies in the literature that implements the concept of a Tiltrotor vehicle to develop a mini UAV, and this article presents the modeling of the system through the Newton-Euler formulation and a control scheme that uses bounded smooth functions to perform hovering. In [9] a complex configuration is proposed and implemented for hover flight. This configuration utilizes the gyroscopic effect provided by tilting rotors. However, it should be noted that controlling the pitch angle is difficult due to the weak contribution of gyroscopic-based torque in handling pitch dynamics. In [10], Kendall et al. presented a backstepping algorithm for a mathematical model inspired by the Gress [9] mechanism, whose simulation results lead to the stabilization of a Bi-rotor vehicle.

The key to achieving stable flight for the Bi-rotor is the design of its attitude controller. The accuracy of the model heavily relies on the control of nonlinear systems [11].

Researchers have proposed various PID control methods for controlling the position and orientation of Bi-rotors [12]. Despite their usefulness, such techniques have their limitations. For instance, in [11], the Euler method was used to linearize the dynamics of a Bi-rotor.

Various nonlinear flight control systems have been proposed and developed to address linear controller design's theoretical constraints and disadvantages. These advanced systems exhibit superior performance in terms of stability and robustness and are capable of handling complex scenarios beyond the scope of traditional linear control techniques. The nonlinear controllers are designed to overcome the fundamental limitations of linear models, which are rooted in the assumption of small perturbations and linearity of the system dynamics. By adopting a nonlinear approach [13], these systems can capture the full range of system



dynamics and exhibit superior performance characteristics.

Several nonlinear control methods, such as feedback linearization [14], backstepping [15, 16], dynamic inversion, predictive control [17], and sliding mode control [18, 19], have been proposed for controlling the motion of UAVs, such as quadrotors. However, regarding trajectory tracking for Bi-rotors, nonlinear control has only been reported in backstepping [20], and the presented methods have only focused on linear control. Within the limited scope of existing research, mathematical models frequently employ simplified representations that overlook critical nonlinear coupling terms. However, the omission of these fundamental terms within the models can have profound and far-reaching implications for the dynamic behavior of the system.

The objective of the current work is to develop a comprehensive model of a Bi-rotor drone that accurately captures its intricate behaviors. Another objective is to design effective control methods for regulating the motion of the UAV, ensuring reliable and precise performance. The research aims to employ nonlinear control techniques, such as feedback linearization, backstepping, and the combined backstepping and feedback linearization (CBFL) control method, to advance attitude and position control in Bi-rotor drones.

The main achievements of this paper are: (1) The current work presents a validated comprehensive dynamic model for the Bi-rotor aircraft using the ADAMS software integrated with the Simulink environment in MATLAB software. (2) The proposed methodology effectively regulates the motion of the UAV, offering reliable and precise performance. (3) The stability of the proposed control approach is analyzed using the Lyapunov method. (4) By employing nonlinear control techniques, the research contributes to the advancement of attitude and position control in Bi-rotor drones, specifically through feedback linearization, backstepping, and the combined backstepping and feedback linearization (CBFL) control method. (5) The study's findings have significant implications for the field of aerial robotics and can find potential applications across various industries.

In conclusion, this article presents a comprehensive model of a Bi-rotor drone that accounts for its intricate behaviors, with a validated dynamic model for the Bi-rotor aircraft using the ADAMS software integrated with the Simulink environment in MATLAB software. The proposed methodology effectively regulates the motion of the UAV, ensuring reliable and precise performance. The study's findings have significant implications for the field of aerial robotics and offer potential applications across various industries. By employing nonlinear control techniques such as feedback linearization, backstepping, and the combined backstepping and feedback linearization (CBFL) control method, including the powerful CBFL method previously mentioned in [21], this research contributes to the advancement of attitude and position control in Bi-rotor drones.

## 2. Dynamic Modeling

A Bi-rotor consists of two propellers and two tilting actuators. By rotating the propellers, thrust forces, and pitch/roll torques are generated. The propellers will rotate in opposite directions at equal speeds to eliminate the resulting yaw torque. The servo motors are responsible for determining the angle of the propellers at each moment to adjust the direction of the generated forces for robot control.

Bi-rotors can maneuver in longitudinal and lateral motions using two strategies. In the first strategy, similar to a helicopter, they move by changing the propellers' angle with the tilting actuators' help. Alternatively, similar to a quadrotor in the second strategy, they achieve longitudinal and lateral motion by creating speed differences among the propellers. Generally, Bi-rotors combine both methods to perform the best maneuver in the shortest possible time. In the following, the attention is directed toward the most pertinent nonlinear modeling and the primary factors that exert influence on the Bi-rotor, such as aerodynamic thrust force, drag torque, and gyroscopic effects. The presence of tilt actuators necessitates a separate treatment for the modeling of other vertical-flight robots. Initially, the degrees of freedom of the Bi-rotor are analyzed, followed by the derivation of the dynamic model of the Bi-rotor in the form of governing equations, based on the system assumptions.

In general, an entirely unconstrained rigid body in space is embodied by a Bi-rotor, characterized by six degrees of freedom. These encompass three translational motions (longitudinal, lateral, vertical) and rotations around each of these motions. The aircraft's position and orientation are comprehensively defined by these degrees of freedom.

For the modeling of the Bi-rotor, four Cartesian coordinate systems are used:

1. Inertial Frame: It is characterized by a fixed orientation and serves as the absolute frame of reference.
2. Body-attached Frame: It is linked to the robot and its orientation is contingent upon the Bi-rotor's attitude at any given moment.
3. Propeller Tilt Frames: Two coordinate systems are considered for the tilt of each propeller around the tilt axis.

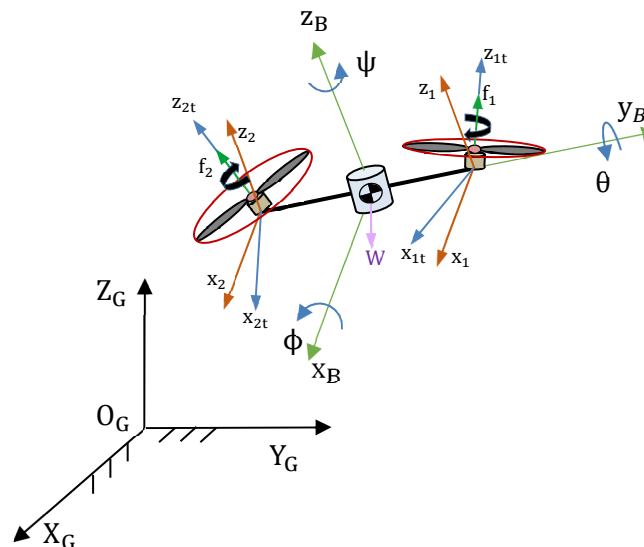


Fig. 1. The Bi-rotor and coordinate systems.



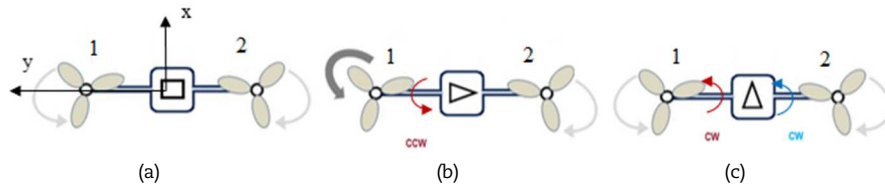


Fig. 2. Bi-rotor motion maneuvers.

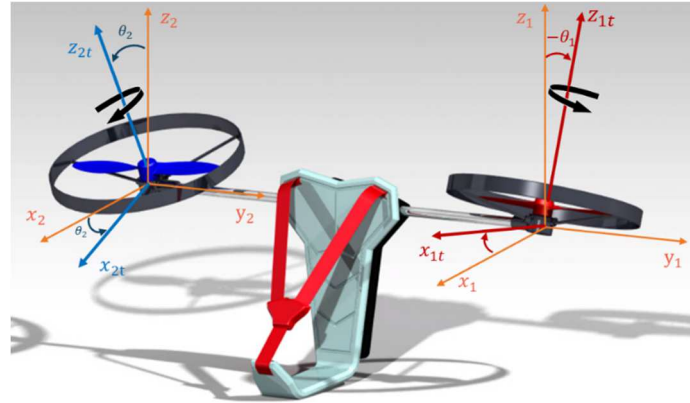


Fig. 3. Rotation of the propellers of the Bi-rotor.

The configuration of the system is illustrated in Fig. 1. In this figure,  $X_G Y_G Z_G$  is the inertial coordinate system,  $X_B Y_B Z_B$  is the local coordinate system,  $X_{t1} Y_{t1} Z_{t1}$  and  $X_{t2} Y_{t2} Z_{t2}$  are coordinate systems for the tilt mechanisms of each propeller. Where  $W$  represents the weight force,  $f_1$  and  $f_2$  denote the thrust forces generated by the propellers.

Based on the frame and arrangement of the actuators in the Bi-rotor's motion, the following descriptions are applicable:

- To achieve forward motion (Fig. 2(a)), the Bi-rotor employs a rotational speed of propellers (1) and (2). Additionally, both propellers tilt in a clockwise direction around the Y-axis. This alteration in the angle of the thrust forces propels the Bi-rotor forward.
- In order to initiate a rightward movement (Fig. 2(b)), propeller (1) rotates at a higher speed compared to propeller (2). This discrepancy in rotational speed results in a change in the generated force, causing the robot to move in the (-Y) direction. To counteract the increased generated torque, propeller (1) tilts counterclockwise around the Y-axis.
- During hovering (Fig. 2(c)), the propellers rotate at a consistent speed in opposite directions. This synchronized rotation generates a thrust force that precisely balances the weight of the robot. However, since the robot's center of mass is not aligned with the Y-axis, the propellers are tilted at fixed angles. These equal and opposite tilts effectively counteract the torques produced around the Y-axis, ensuring that the propeller speed remains constant.

By taking into account the two thrust forces of propellers and two tilt actuators as the four inputs, and considering the six degrees of freedom for the Bi-rotor, the six dynamic equations are computed using the Newton-Euler method. The Euler transformation matrix [22] is expressed as follows.

$$R_B^G = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \varphi - \sin \psi \cos \varphi & \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \varphi & \sin \psi \sin \theta \cos \varphi - \cos \psi \sin \varphi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (1)$$

For each of the tilt actuators of the Bi-rotor, a coordinate system will be considered, in which the forces and torques generated will be transformed to the local coordinate systems using the transfer matrices (2) and (3) and ultimately transferred to the absolute coordinate system using the Euler transformation matrix. Figure 3 illustrates the Bi-rotor as well as the direction of propellers' rotation. In this figure,  $X_i Y_i Z_i$  represents the coordinate systems before rotation and  $X_{it} Y_{it} Z_{it}$  represents the rotated coordinate systems.

Rotation matrices for each propeller are as:

$$R_{T1}^B(\theta) = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 \\ 0 & 1 & 0 \\ -\sin \theta_1 & 0 & \cos \theta_1 \end{bmatrix} \quad (2)$$

$$R_{T2}^B(\theta) = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ 0 & 1 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \quad (3)$$

The expression of the dynamics of a rigid body subjected to external forces and torques applied to its center of mass can be achieved by utilizing the Newton-Euler equations for three-dimensional space. These equations are formulated as:

$$\begin{bmatrix} F^G \\ M^B \end{bmatrix} = \begin{bmatrix} m & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} a^G \\ \alpha^B \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^B \times I \omega^B \end{bmatrix} \quad (4)$$



where  $F^G$  is the sum of forces acting on the Bi-rotor from the actuators and external forces, expressed in three-dimensional space as follows:

$$F^G = \begin{bmatrix} F_x^G & F_y^G & F_z^G \end{bmatrix}^T \quad (5)$$

Furthermore,  $a^G$  represents the acceleration of the Bi-rotor's center of mass, given by the following form:

$$a^G = [\ddot{x} \quad \ddot{y} \quad \ddot{z}]^T \quad (6)$$

Also  $M^B$  represents the torques applied to the Bi-rotor in the local coordinate system as follows:

$$M^B = \begin{bmatrix} M_x^B & M_y^B & M_z^B \end{bmatrix}^T \quad (7)$$

The inertia matrix of the Bi-rotor is as follows:

$$I = \begin{bmatrix} I_x & 0 & I_{xz} \\ 0 & I_y & 0 \\ I_{xz} & 0 & I_z \end{bmatrix} \quad (8)$$

where  $I_x, I_y, I_z$  and  $I_{xz}$  are the moments of inertia of the Bi-rotor in the local coordinate system. In addition, the angular velocity and angular acceleration of the Bi-rotor in the local frame are defined as follows:

$$\omega^B = [\dot{\varphi} \quad \dot{\theta} \quad \dot{\psi}]^T \quad (9)$$

$$\alpha^B = [\ddot{\varphi} \quad \ddot{\theta} \quad \ddot{\psi}]^T \quad (10)$$

A rotating propeller generates a thrust force in the positive  $z_B$  direction of the local coordinate system. The magnitude of this force is calculated according to equation (11) for Bi-rotor propellers.

$$f_i = b\omega_i^2 \quad (i = 1 \rightarrow 2) \quad (11)$$

where  $f_i$  represents the thrust force,  $b$  is the thrust coefficient, and  $\omega_i$  is the rotational speed of each propeller. The thrust coefficient is unique for each propeller which is calculated based on the following equation:

$$b = c_T \rho A R^2 \quad (12)$$

In this equation,  $\rho$  represents the air density,  $R$  is the radius of the propellers,  $A$  is the area of the propeller disc (wetted area), and  $c_T$  is the thrust coefficient. The thrust coefficient is a dimensionless parameter determined based on the airfoil profile of the propeller which is calculated using the blade element momentum theory [23].

Considering that the generalized forces are required to be expressed in the main frame, the consideration is given to the total thrust forces in the  $x_B$  direction ( $u_1$ ) and the total thrust forces in the  $z_B$  direction ( $u_2$ ). To convert these forces into the inertial frame, the transformation matrices, as described by equations (1), (2), and (3), are utilized as follows:

$$F_t = R_B^G \begin{bmatrix} R_{T1}^B & 0 \\ 0 & R_{T2}^B \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} (\sin\varphi \sin\psi + \cos\varphi \cos\psi \sin\theta)u_2 + \cos\theta \cos\psi u_1 \\ (\cos\varphi \sin\theta \sin\psi - \cos\psi \sin\varphi)u_2 + \cos\theta \sin\psi u_1 \\ \cos\varphi \cos\theta u_2 - \sin\theta u_1 \end{bmatrix} \quad (13)$$

The robot's center of mass is subjected to the gravitational force, which is applied in the negative  $z_1$  direction in the fixed coordinate system. The expression for the gravitational force is as follows:

$$F_g = [0 \quad 0 \quad -mg]^T \quad (14)$$

The aerodynamic forces are the forces that arise from the rotation of the propellers in the Bi-rotor and act in opposition to the thrust force, thereby impeding the motion of the Bi-rotor. These forces can be defined as follows:

$$F_f = \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (15)$$

The aerodynamic coefficients ( $\gamma_x, \gamma_y, \gamma_z$ ) are utilized in the formulation. The resultant force equations in the Bi-rotor, which are obtained by summing equations (13), (14), and (15), are expressed as follows:

$$F^I = F_t + F_g - F_f \quad (16)$$

As a result, the dynamic equations are as follows:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \begin{bmatrix} (\sin\varphi \sin\psi + \cos\varphi \cos\psi \sin\theta)u_2 + \cos\theta \cos\psi u_1 \\ (\cos\varphi \sin\theta \sin\psi - \cos\psi \sin\varphi)u_2 + \cos\theta \sin\psi u_1 \\ \cos\varphi \cos\theta u_2 - \sin\theta u_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} - \begin{bmatrix} \gamma_x & 0 & 0 \\ 0 & \gamma_y & 0 \\ 0 & 0 & \gamma_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (17)$$



The drag torque of each propeller is generated in the opposite direction of its rotation. The magnitude of this torque for each propeller can be calculated as:

$$M_i = d\omega_i^2 \quad (i = 1 \rightarrow 2) \quad (18)$$

where  $M_i$  is the drag torque of each propeller and  $d$  is the drag coefficient. The drag coefficient is calculated for each propeller as:

$$b = c_D \rho A R^3 \quad (19)$$

In this equation,  $\rho$  represents the air density,  $R$  is the radius of the propellers,  $A$  is the surface area of the propeller (wetted area), and  $c_D$  is the drag coefficient. The drag coefficient, similar to the thrust coefficient, is a dimensionless number determined based on the airfoil profile of the propeller using the momentum theory and blade element theory.

In the equations governing the propeller torque, the torques exerted by the propellers around the propeller axes in the local coordinates are influenced by the control inputs that include:

1. The drag torque arising from the rotation of the propellers is applied in the opposite direction of the propeller rotation axis. It occurs due to the resistance encountered during propeller rotation and is represented by equation (20).

2. The torque resulting from the thrust forces produced by each propeller is induced due to the distance between the forces and the robot's center of mass. It is expressed by equations (21).

$$M_p = R_B^G R_{T1}^B \begin{bmatrix} 0 \\ 0 \\ -M_1 \end{bmatrix} + R_B^G R_{T2}^B \begin{bmatrix} 0 \\ 0 \\ -M_2 \end{bmatrix} \quad (20)$$

$$M_s = \begin{bmatrix} \mathbf{x}_c \\ l \\ -z_c \end{bmatrix} \times F_1 + \begin{bmatrix} \mathbf{x}_c \\ -l \\ -z_c \end{bmatrix} \times F_2 \quad (21)$$

The total torque due to the rotation of the propellers is as:

$$M_t = M_p + M_s = \begin{bmatrix} blc_1\omega_1^2 - blc_2\omega_2^2 - ds_1\omega_1^2 + ds_2\omega_2^2 \\ u_1z_c - u_2x_c \\ -dc_1\omega_1^2 + dc_2\omega_2^2 - bls_1\omega_1^2 + bls_2\omega_2^2 \end{bmatrix} = \begin{bmatrix} u_4 \\ u_1z_c - u_2x_c \\ u_3 \end{bmatrix} \quad (22)$$

The gyroscopic effect in the Bi-rotor arises from the simultaneous generation of torques caused by the rotation of the propellers and the rotation of the Bi-rotor's body around its local coordinate axes. A torque is created, and its magnitude and direction are calculated as follows:

$$M_g = (\dot{\phi}\vec{i} + \dot{\theta}\vec{j} + \dot{\psi}\vec{k}) J_r \times \left( R_{T1}^B \begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix} + R_{T2}^B \begin{bmatrix} 0 \\ 0 \\ -\omega_2 \end{bmatrix} \right) = \begin{bmatrix} J_r \sigma_z \dot{\theta} \\ -J_r (\dot{\phi} \sigma_z + \dot{\psi} \sigma_x) \\ J_r \sigma_x \dot{\theta} \end{bmatrix} \quad (23)$$

The inertia of the rotors, denoted as  $J_r$ , is involved in the calculations. The vector sums of the propeller rotational velocities in the  $x$  and  $z$  directions of the local coordinate system, represented by  $\sigma_z$  and  $\sigma_x$ , respectively, are determined by considering their rotational directions, as specified as:

$$\sigma_x = w_1 \sin(t_1) - w_2 \sin(t_2) \quad (24)$$

$$\sigma_z = w_1 \cos(t_1) - w_2 \cos(t_2) \quad (25)$$

Similar to the force equations, certain torques oppose the motion of the Bi-rotor. These aerodynamic torques are defined within the rotor torque equations as follows:

$$M_f = \begin{bmatrix} \gamma_\phi & 0 & 0 \\ 0 & \gamma_\theta & 0 \\ 0 & 0 & \gamma_\psi \end{bmatrix} \begin{bmatrix} \dot{\phi}^2 \\ \dot{\theta}^2 \\ \dot{\psi}^2 \end{bmatrix} \quad (26)$$

where  $(\gamma_\phi, \gamma_\theta, \gamma_\psi)$  are the coefficients of aerodynamic friction.

Taking into account all the mentioned effects and using equations (22), (23), and (26), the rotational equations of motion are calculated as follows:

$$\begin{bmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T = I^{-1} (M_t + M_g - M_f) \quad (27)$$

Therefore,

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\theta}\dot{\psi}(I_y - I_z) \\ \dot{\phi}\dot{\psi}(I_z - I_x) \\ \dot{\phi}\dot{\theta}(I_x - I_y) \end{bmatrix} - \begin{bmatrix} \dot{\phi}\dot{\theta}I_{xz} \\ (\dot{\psi}^2 - \dot{\phi}^2)I_{xz} \\ -\dot{\psi}\dot{\theta}I_{xz} \end{bmatrix} + \begin{bmatrix} u_4 \\ u_1z_c - u_2x_c \\ u_3 \end{bmatrix} + \begin{bmatrix} J_r \sigma_z \dot{\theta} \\ -J_r (\dot{\phi} \sigma_z + \dot{\psi} \sigma_x) \\ J_r \sigma_x \dot{\theta} \end{bmatrix} - \begin{bmatrix} \gamma_\phi & 0 & 0 \\ 0 & \gamma_\theta & 0 \\ 0 & 0 & \gamma_\psi \end{bmatrix} \begin{bmatrix} \dot{\phi}^2 \\ \dot{\theta}^2 \\ \dot{\psi}^2 \end{bmatrix} \quad (28)$$



Finally, the dynamic equations of the Bi-rotor are expressed as follows:

$$\begin{aligned}
 \ddot{y} &= \frac{1}{m} ((\cos \varphi \sin \theta \sin \psi - \cos \psi \sin \varphi) u_2 + \cos \theta \sin \psi u_1) - \frac{\gamma_y}{m} \dot{y} \\
 \ddot{z} &= \frac{1}{m} (\cos \varphi \cos \theta u_2 - \sin \theta u_1) - g - \frac{\gamma_z}{m} \dot{z} \\
 \ddot{\varphi} &= \frac{J_r \dot{\theta} (I_z \sigma_z + I_{xz} \sigma_x) - \dot{\theta} \dot{\varphi} I_{xz} (I_x - I_y + I_z) - \dot{\theta} \dot{\psi} (I_z^2 - I_x I_y + I_{xz}^2)}{I_x I_z - I_{xz}^2} + \frac{I_{xz} \gamma_\psi \dot{\psi}^2 - I_z \gamma_\varphi \dot{\varphi}^2 + I_x u_4 - I_{xz} u_3}{I_x I_z - I_{xz}^2} \\
 \ddot{\theta} &= \dot{\varphi} \dot{\psi} \frac{(I_z - I_x)}{I_y} - (\dot{\psi}^2 - \dot{\varphi}^2) \frac{I_{xz}}{I_y} + \frac{J_r}{I_y} (\dot{\psi} \sigma_x - \dot{\varphi} \sigma_z) + \frac{u_1 z_c - u_2 x_c}{I_y} - \frac{\gamma_\theta}{I_y} \dot{\theta}^2 \\
 \ddot{\psi} &= \frac{J_r \dot{\theta} (I_x \sigma_x + I_{xz} \sigma_z) + \dot{\theta} \dot{\psi} I_{xz} (I_x - I_y + I_z) + \dot{\theta} \dot{\varphi} (I_x^2 - I_x I_y + I_{xz}^2)}{I_x I_z - I_{xz}^2} + \frac{I_{xz} \gamma_\varphi \dot{\varphi}^2 - I_x \gamma_\psi \dot{\psi}^2 - (I_{xz} u_4 - I_x u_3)}{I_x I_z - I_{xz}^2}
 \end{aligned} \quad (29)$$

In the mathematical modeling of the system, four control inputs are determined; the total thrust forces in the x-direction of the local coordinate system ( $u_1$ ), the total thrust forces in the z-direction ( $u_2$ ), the torques applied to the  $x_B$  and  $z_B$  axes by the propellers, represented by  $u_i$  ( $i = 3 \rightarrow 4$ ). Equation (30) represents the relationships between the considered inputs and the rotational speeds of the propellers can be obtained as:

$$[u_i]_{4 \times 1} = A [\omega_i^2]_{4 \times 1} \quad (30)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & b & b \\ b & b & 0 & 0 \\ -d & d & -bl & bl \\ bl & -bl & -d & d \end{bmatrix} \begin{bmatrix} c_1 \omega_1^2 \\ c_2 \omega_2^2 \\ s_1 \omega_1^2 \\ s_2 \omega_2^2 \end{bmatrix} \quad (31)$$

where  $l$  is the distance between the center of gravity of the Bi-rotor and the axis of each rotor.

To calculate the rotor speeds and their rotational angles based on the control inputs, the inverse matrix equation (31) is used as follows:

$$A^{-1} = \begin{bmatrix} 0 & \frac{1}{2b} & -\frac{d}{2(b^2 l^2 + d^2)} & \frac{bl}{2(b^2 l^2 + d^2)} \\ 0 & \frac{1}{2b} & \frac{d}{2(b^2 l^2 + d^2)} & -\frac{bl}{2(b^2 l^2 + d^2)} \\ \frac{1}{2b} & 0 & -\frac{bl}{2(b^2 l^2 + d^2)} & -\frac{d}{2(b^2 l^2 + d^2)} \\ \frac{1}{2b} & 0 & \frac{bl}{2(b^2 l^2 + d^2)} & \frac{d}{2(b^2 l^2 + d^2)} \end{bmatrix} \quad (32)$$

Based on equations (31) and (32), the rotor speeds and the rotational angles are calculated as follows:

$$\omega_i = \sqrt{\frac{R_{i,1}^2 + R_{(i+2),1}^2}{b}} \quad (i = 1 \rightarrow 2) \quad (33)$$

$$\theta_i = \text{atan2}(R_{(i+2),1}^2, R_{i,1}^2) \quad (i = 1 \rightarrow 2) \quad (34)$$

In equations (33) and (34),  $R_i$  can be calculated as follows:

$$[R_i]_{4 \times 1} = A^{-1} [u_i]_{4 \times 1} \quad (35)$$

### 3. State Space Representation

The independent state variables of the system, which include position, Euler angles, and their derivatives with respect to time, are defined as  $X = [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z} \ \varphi \ \dot{\varphi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T$ .

The system's state vector is as follows:

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \quad (36)$$

By substituting the state variables into the system's dynamic equations (29), the system's state space can be written as follows:

$$\dot{X} = F(X) + G(X)U \quad (37)$$

where,



$$F(X) = \begin{bmatrix} x_2 \\ -\frac{\gamma_x}{m} x_2 \\ x_4 \\ -\frac{\gamma_x}{m} x_4 \\ x_6 \\ -g - \frac{\gamma_x}{m} x_6 \\ x_8 \\ a_5 x_8 x_{10} - a_6 x_{10} x_{12} + a_3 x_{10} \sigma_x + a_4 x_{10} \sigma_z - a_1 \gamma_f x_8^2 + a_2 \gamma_\psi x_{12}^2 \\ x_{10} \\ -a_8 (x_{12}^2 - x_8^2) + a_7 x_8 x_{12} - a_9 (x_8 \sigma_z - x_{12} \sigma_x) - a_{10} \gamma_\theta x_{10}^2 \\ x_{12} \\ a_{13} x_8 x_{10} - a_5 x_{10} x_{12} - a_3 x_{10} \sigma_z - a_{12} x_{10} \sigma_x + a_2 \gamma_f x_8^2 - a_{11} \gamma_\psi x_{12}^2 \end{bmatrix} \quad (38)$$

$$G(X) = [G_1 \ G_2 \ G_3 \ G_4]_{12 \times 4} \quad (39)$$

which includes the following elements:

$$G_1 = [0 \ g_4 \ 0 \ g_5 \ 0 \ -g_6 \ 0 \ 0 \ 0 \ a_{10} z_c \ 0 \ 0]^T \quad (40)$$

$$G_2 = [0 \ g_1 \ 0 \ g_2 \ 0 \ g_3 \ 0 \ 0 \ 0 \ -a_{10} x_c \ 0 \ 0]^T \quad (41)$$

$$G_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad (42)$$

$$G_4 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T \quad (43)$$

in which,

$$\begin{aligned} g_1 &= \left( \frac{1}{m} \right) (\cos(x_7) \cos(x_{11}) \sin(x_9) + \sin(x_7) \sin(x_{11})) \\ g_2 &= \left( \frac{1}{m} \right) (\cos(x_7) \sin(x_9) \sin(x_{11}) - \cos(x_{11}) \sin(x_7)) \\ g_3 &= \left( \frac{1}{m} \right) \cos(x_7) \cos(x_9) \\ g_4 &= \left( \frac{1}{m} \right) \cos(x_9) \cos(x_{11}) \\ g_5 &= \left( \frac{1}{m} \right) \cos(x_9) \sin(x_{11}) \\ g_6 &= \left( \frac{1}{m} \right) \sin(x_9) \end{aligned} \quad (44)$$

Furthermore, the control input matrix is as follows:

$$U = [u_1 \ u_2 \ u_5 \ u_6]^T \quad (45)$$

in which,

$$\begin{bmatrix} u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} -a_2 & a_1 \\ a_{11} & -a_2 \end{bmatrix} \begin{bmatrix} u_3 \\ u_4 \end{bmatrix} \quad (46)$$

The following matrix represents the parameters in the state-space equations of the system in terms of the moments of inertia of the Bi-rotor as:

$$\begin{aligned} a_1 &= \frac{-I_z}{a}; a_2 = \frac{-I_{xz}}{a}; a_3 = \frac{-I_{xz} J_r}{a}; a_4 = \frac{-I_z J_r}{a}; a_5 = \frac{I_{xz} (I_x - I_y - I_z)}{a}; a_6 = -\frac{(I_{xz}^2 + I_z^2 - I_y I_z)}{a}; a_7 = -\frac{(I_x - I_z)}{I_y}; \\ a_8 &= \frac{I_{xz}}{I_y}; a_9 = \frac{J_r}{I_y}; a_{10} = \frac{1}{I_y}; a_{11} = -\frac{I_x}{a}; a_{12} = -\frac{I_x J_r}{a}; a_{13} = -\frac{(I_x^2 - I_y I_x + I_{xz}^2)}{a}; a = I_x I_z - I_{xz}^2 \end{aligned} \quad (47)$$



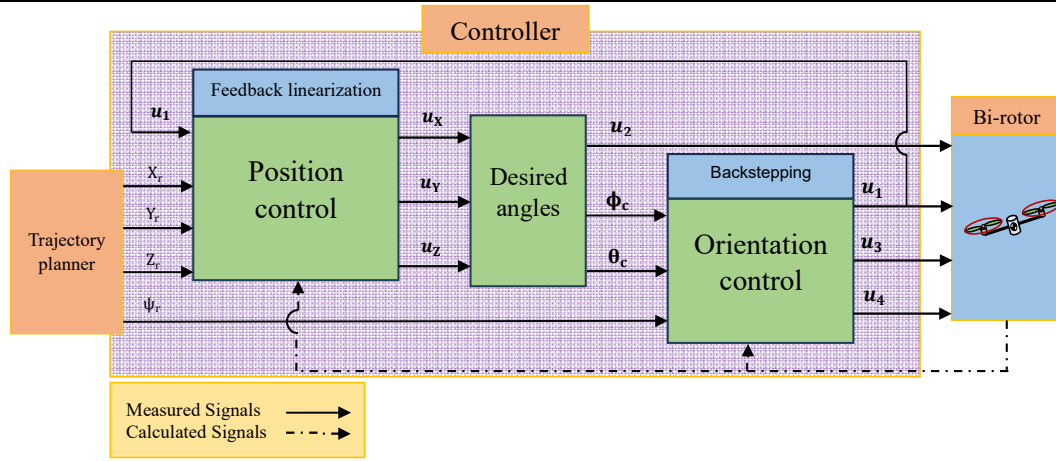


Fig. 4. Bi-rotor control diagram.

#### 4. Control Algorithm

The Bi-rotor system is characterized as underactuated, with four control inputs and six degrees of freedom. This implies that only four degrees of freedom, namely spatial positions ( $x, y, z$ ), and yaw angle ( $\psi$ ), can be directly controlled. As shown in Fig. 4, the trajectory planner block first determines the desired reference trajectories. Then, the desired values are passed to the position control and attitude control units. For simplicity, virtual control inputs ( $u_x, u_y, u_z$ ) are chosen to control the  $x, y$ , and  $z$  variables. By using these virtual inputs, the control input  $u_2$  is computed. Subsequently, the values of  $\varphi_c$  and  $\theta_c$  are determined in the reference attitude determination unit. The input values ( $u_1, u_3, u_4$ ) are determined based on the obtained attitude angles in the Bi-rotor attitude controller. In this way, the position of the Bi-rotor is controlled through the attitude control of the Bi-rotor. However, virtual control inputs are needed to connect these two parts and control the robot's position.

As shown in Fig. 4, three virtual inputs ( $u_x, u_y, u_z$ ) are calculated using the feedback linearization approach, while the other three control inputs ( $u_1, u_3, u_4$ ) are generated using the backstepping control method.

##### 4.1. Calculation of roll and pitch angles

As mentioned, only four degrees of freedom of the Bi-rotor, including ( $x, y, z, \psi$ ), are controlled using the system's four control inputs. Therefore, for controlling  $x, y$ , and  $z$ , which do not have separate control inputs according to the Bi-rotor's dynamic equations, three virtual inputs are considered as follows:

$$u_x = (\cos\varphi \sin\theta \cos\psi + \sin\varphi \sin\psi) u_2 \quad (48)$$

$$u_y = (\cos\varphi \sin\theta \sin\psi - \sin\varphi \cos\psi) u_2 \quad (49)$$

$$u_z = \cos\varphi \cos\theta u_2 \quad (50)$$

Based on equations (49), (50), and (51), we can write:

$$\tan\theta_c = \frac{u_x \cos\psi + u_y \sin\psi}{u_z} \quad (51)$$

$$\tan\varphi_c = \frac{\cos\varphi_{9s} (u_x \sin\psi + u_y \cos\psi)}{u_z} \quad (52)$$

As a result, from equations (52) and (53), we have:

$$\theta_c = \tan^{-1} \left( \frac{u_x \cos\psi + u_y \sin\psi}{u_z} \right) \quad (53)$$

$$\varphi_c = \tan^{-1} \left( \frac{\cos\varphi_{9s} (u_x \sin\psi + u_y \cos\psi)}{u_z} \right) \quad (54)$$

Based on the virtual inputs of the system, the control input  $u_2$ , which is required to control the variable  $z$ , it depends on the command angles ( $\varphi_c, \theta_c$ ). Therefore based on (51), to determine the control input  $u_2$  using virtual inputs, we have:

$$u_2 = \frac{u_z}{\cos\varphi_c \cos\theta_c} \quad (55)$$

##### 4.2. The CBFL control method

The Combined Backstepping and Feedback Linearization (CBFL) control method is derived by combining two control methods, namely backstepping and feedback linearization control. The combined backstepping and feedback linearization control approach offers enhanced performance, expanded control authority, improved tracking performance, and increased robustness compared to using backstepping or feedback linearization control algorithms alone. It combines the strengths of both techniques to achieve



superior control outcomes in terms of stability, trajectory tracking, and robustness against uncertainties and disturbances. By combining these two methods, improved results are attained in terms of convergence to the reference trajectory and the minimization of steady-state errors. In this method, three virtual inputs ( $u_x, u_y, u_z$ ) are used to stabilize the variables ( $x, y, z$ ) using the feedback linearizing control method. The control input  $u_2$  is obtained based on  $u_z$ . Additionally, three other control inputs ( $u_1, u_5, u_6$ ) are calculated using the backstepping control method to stabilize the variables ( $\varphi, \theta, \psi$ ).

### 4.3. Stability proof

#### 4.3.1. Stability of the outer control loop

In contrast to linear control methods, nonlinear control methods encounter multiple constraints and complexities. The process of linearization in the dynamic equations of a nonlinear system leads to deviations from its actual model. However, the approach based on a linearizing feedback, which draws inspiration from linear control methods, involves the transformation of nonlinear differential equations into a linear differential equation accompanied by a nonlinear algebraic equation. By applying feedback linearization control, the system can be controlled without the need for linearization in the system equations. The dynamic equations of the systems, excluding uncertainties and disturbances, are expressed as follows:

$$\ddot{X} = F(X, \dot{X}) + G(X, \dot{X})U \quad (56)$$

Proposition: Control law (58) stabilizes the dynamic system (57) around the reference trajectories.

$$U = \frac{1}{G(X, \dot{X})} \left\{ \ddot{X}_r + k_D(\dot{X}_r - \dot{X}) + k_P(X_r - X) + k_I \int_0^t (X_r - X) d\tau - F(X, \dot{X}) \right\} \quad (57)$$

where  $X_r$  is the vector of desired system trajectories,  $t$  is the flight duration, and ( $k_P, k_D, k_I$ ) are the proportional, derivative, and integral control gains, which are always assumed to be positive.

Proof: In equation (57),  $\ddot{X}$  is considered as follows:

$$\ddot{X} = \ddot{X}_r + k_D(\dot{X}_r - \dot{X}) + k_P(X_r - X) - F(X, \dot{X}) \quad (58)$$

The error function is defined as  $e = X_r - X$ , its first derivative with respect to time as  $\dot{e} = \dot{X}_r - \dot{X}$ , and its second derivative as  $\ddot{e} = \ddot{X}_r - \ddot{X}$ . Therefore equation (57) can be transformed into the following equation:

$$\ddot{e} + k_D \dot{e} + k_P e + k_I \int_0^t e d\tau = 0 \quad (59)$$

By taking the time derivative of equation (60), the following equation can be obtained:

$$\ddot{e} + k_D \dot{e} + k_P e = 0 \quad (60)$$

The characteristic equation of this expression can be written as:

$$\lambda^3 + k_D \lambda^2 + k_P \lambda + k_I = 0 \quad (61)$$

All the closed-loop system poles will be chosen on the left of the imaginary axis using appropriate control gains. Therefore, the asymptotic stability of the closed-loop system for its variables under the feedback linearization control method is proven using the Lyapunov method.

The control inputs accordingly are calculated as follows:

$$u_x = -u_1 \cos x_9 \cos x_{11} + m(\dot{x}_{2c} + k_{Dx} \dot{e}_x + k_{Px} e_x) \quad (62)$$

$$u_y = -u_1 \cos x_9 \sin x_{11} + m(\dot{x}_{4c} + k_{Dy} \dot{e}_y + k_{Py} e_y) \quad (63)$$

$$u_z = u_1 \sin x_9 + mg + m(\dot{x}_{6c} + k_{Dz} \dot{e}_z + k_{Pz} e_z) \quad (64)$$

### 4.4. Inner loop control

In the inner loop control of the system, the backstepping method is employed to stabilize the system variables around desired trajectories. Based on the introduced control strategy, the state variables of the system ( $\phi, \theta, \psi$ ) are stabilized around ( $\varphi_c, \theta_c, \psi_r$ ) using stable system inputs,

$$\dot{x}_{2i-1} = x_{2i} \quad (65)$$

$$\dot{x}_{2i-1} = f_{2i}(X) + g_{2i}(X)u_k \quad (66)$$

where  $i = 3 \rightarrow 6$  and  $k \in \{1, 3, 4\}$ .

Theorem: Control law (68) stabilizes the dynamic system (29) around the reference trajectories,

$$u_k = \frac{1}{g_{2i}(X)} [\dot{x}_{2ir} + e_{2i-1} - f_{2i}(X) - c_{2i-1}(e_{2i} + c_{2i-1}e_{2i-1}) - c_{2i}e_{2i}] \quad (67)$$

where  $e_i$  ( $i = 1 \rightarrow 12$ ) are error signals and  $c_i$  ( $i = 1 \rightarrow 12$ ) are positive control gains.



Proof: Firstly, the error functions for the control of the odd-numbered state variables of the system are defined as follows:

$$e_{2i-1} = x_{(2i-1)r} - x_{2i-1} \quad (68)$$

where  $x_{(2i-1)r}$  ( $i = 1 \rightarrow 6$ ) are the desired signals. The candidate Lyapunov function to stabilize these error signals is defined as follows:

$$W_i = \frac{1}{2} e_{2i-1}^2 \quad (69)$$

The first derivative of this error function with respect to time is calculated as follows:

$$\dot{W}_i = e_{2i-1} \dot{e}_{2i-1} = e_{2i-1} (\dot{x}_{(2i-1)r} - \dot{x}_{2i-1}) \quad (70)$$

By utilizing equation (66), equation (71) can be expressed as follows:

$$\dot{W}_i = e_{(2i-1)} (\dot{x}_{(2i-1)r} - x_{2i}) \quad (71)$$

To achieve asymptotic stability of the closed-loop system, we can choose:

$$x_{2i} = \dot{x}_{(2i-1)r} + c_{2i-1} e_{2i-1} \quad (72)$$

Therefore  $\dot{W}_i$  can be calculated as:

$$\dot{W}_i = -c_{2i-1} e_{2i-1}^2 < 0 \quad (73)$$

which is a negative definite function. As mentioned, the values  $c_{2i-1}$  ( $i = 1 \rightarrow 6$ ) are considered to be positive.

By using this equation, another error function can be defined for controlling the even-numbered state variables of the system as follows:

$$e_{2i} = x_{2i} - \dot{x}_{(2i-1)r} - c_{2i-1} e_{2i-1} \quad (74)$$

according to the system state space equations  $\dot{x}_{(2i-1)r} = x_{2ir}$ . Therefore the derivative of equation (75) can be obtained as follows:

$$\dot{e}_{2i} = \dot{x}_{2i} - \dot{x}_{2ir} - c_{2i-1} \dot{e}_{2i-1} \quad (75)$$

Furthermore, the derivative of equation (69) can be expressed as follows:

$$\dot{e}_{2i-1} = \dot{x}_{(2i-1)r} - x_{2i} \quad (76)$$

By substituting equation (77) into equation (75), we can write:

$$e_{2i} = -\dot{e}_{2i-1} - c_{2i-1} e_{2i-1} \quad (77)$$

The second Lyapunov candidate function is defined as follows:

$$V_i = \frac{1}{2} (e_{2i-1}^2 + e_{2i}^2) \quad (78)$$

Its derivative is calculated as follows:

$$\dot{V}_i = e_{2i-1} \dot{e}_{2i-1} + e_{2i} \dot{e}_{2i} \quad (79)$$

By substituting (76) and (78) into (80), we obtain:

$$\dot{V}_i = e_{2i-1} (-e_{2i} - c_{2i-1} e_{2i-1}) + e_{2i} (\dot{x}_{2i} - \dot{x}_{2ir} - c_{2i-1} \dot{e}_{2i-1}) \quad (80)$$

Furthermore, by substituting equations (67) and (78) into equation (81), the following expression is obtained:

$$\dot{V}_i = e_{2i-1} (-e_{2i} - c_{2i-1} e_{2i-1}) + e_{2i} \{f_{2i}(X) + g_{2i}(X)u_k - \dot{x}_{2ir} - c_{2i-1} (-e_{2i} - c_{2i-1} e_{2i-1})\} \quad (81)$$

To ensure the stability of the state variables around the reference trajectories,  $\dot{V}_i$  should be negative definite.

$$V_i = -c_{2i-1} e_{2i-1}^2 - c_{2i} e_{2i}^2 \quad (82)$$

Furthermore, three other control inputs ( $u_1, u_5, u_6$ ) by combining equations (75) and (76) are calculated as follows:

$$u_5 = -a_5 x_8 x_{10} + a_6 x_{10} x_{12} - a_3 x_{10} \sigma_x - a_4 x_{10} \sigma_z + a_1 \gamma_\varphi x_8^2 - a_2 \gamma_\psi x_{12}^2 + \Delta_\varphi \quad (83)$$



$$\Delta_{\varphi} = e_7 - c_7 (e_8 + c_7 e_7) - c_8 e_8 \quad (84)$$

$$u_1 = \frac{1}{a_{10} z_c} \left( a_{10} u_2 x_c + a_8 (x_{12}^2 - x_8^2) - a_7 x_8 x_{12} + a_9 (x_8 \sigma_z - x_{12} \sigma_x) + a_{10} \gamma_{\theta} x_{10}^2 + \Delta_{\theta} \right) \quad (85)$$

$$\Delta_{\theta} = e_9 - c_9 (e_{10} + c_9 e_9) - c_{10} e_{10} \quad (86)$$

$$u_6 = -a_{13} x_8 x_{10} + a_5 x_{10} x_{12} + a_3 x_{10} \sigma_z + a_{12} x_{10} \sigma_x - a_2 \gamma_{\varphi} x_8^2 + a_{11} \gamma_{\psi} x_{12}^2 + \Delta_{\psi} \quad (87)$$

$$\Delta_{\psi} = e_{11} - c_{11} (e_{12} + c_{11} e_{11}) - c_{12} e_{12} \quad (88)$$

## 5. Obtained Results

The performance of the designed controllers for trajectory tracking of the Bi-rotor was evaluated using MATLAB. In order to validate the dynamic modeling of the Bi-rotor, the robot, along with the actuators and system inputs, was modeled in ADAMS software. It was then utilized to track the reference trajectory. In this section, the performance of the Bi-rotor system under the backstepping and CBFL control methods are compared, with the control strategy shown in Fig. 4 being used for the CBFL control method. The system parameters and the information related to the controllers are gathered in Table 1. The control parameters are given in Table 2.

### 5.1. ADAMS/MATLAB CO-simulation

The co-simulation with Adams serves as a valuable tool to validate the modeling and control approach. While the study has limitations, such as simplified assumptions, potential model inaccuracies, idealized controller tuning, and the absence of hardware-specific factors, co-simulation with Adams offers several advantages. It enables comprehensive validation of the dynamic model, considering real-world factors and enhancing the accuracy and realism of the simulation. Adams provides advanced capabilities for evaluating system performance and dynamic behavior. Moreover, co-simulation facilitates by integrating control algorithms with the mechanical aspects of the system. By leveraging the capabilities of Adams, the simulation study can provide valuable insights and validation for the modeling and control approach. Hence, the Adams software is utilized for simulation in order to validate the dynamic modeling of the Bi-rotor robot [24, 25]. For model validation as an initial step, the robot's physics equations can be examined using highly accurate dynamic analysis software. In this article, a 3D model of the Bi-rotor is designed using SolidWorks software for validation purposes. Subsequently, this model is imported into the ADAMS software to determine the mechanical constraints and connections.

Upon transferring the Bi-rotor to the ADAMS software and defining the mechanical constraints and connections for joint simulation between ADAMS and MATLAB, the ADAMS/MATLAB CO-simulation block diagram is generated in the Simulink environment, depicted as Fig. 6.

Table 1. Bi-rotor parameters.

Description	Parameter	Value	Unit
Gravity acceleration	g	9.806	m/s <sup>2</sup>
Bi-rotor Mass	m	5.549	kg
The distance between the rotors and the center of mass of the Bi-rotor along the X <sub>B</sub> axis.	x <sub>c</sub>	7.578×10 <sup>-3</sup>	m
The distance between the rotors and the center of mass of the Bi-rotor along the Y <sub>B</sub> axis.	y <sub>c</sub>	0.782	m
The distance between the rotors and the center of mass of the Bi-rotor along the Z <sub>B</sub> axis.	z <sub>c</sub>	3.039×10 <sup>-2</sup>	m
The moment of inertia of the Bi-rotor around the X <sub>B</sub> axis.	I <sub>x</sub>	1.161	N.m.s <sup>2</sup>
The moment of inertia of the Bi-rotor around the Y <sub>B</sub> axis.	I <sub>y</sub>	0.032	N.m.s <sup>2</sup>
The moment of inertia of the Bi-rotor around the Z <sub>B</sub> axis.	I <sub>z</sub>	1.135	N.m.s <sup>2</sup>
The moment of inertia of the Bi-rotor with respect to the X <sub>B</sub> Z <sub>B</sub> plane.	I <sub>xz</sub>	-2.173×10 <sup>-3</sup>	N.m.s <sup>2</sup>
The moment of inertia of the rotor.	J <sub>r</sub>	4.021×10 <sup>-4</sup>	kg.m/rad <sup>2</sup>
Thrust coefficient	B	1.417×10 <sup>-4</sup>	kg.m/rad <sup>2</sup>
Torque coefficient	D	2.062×10 <sup>-6</sup>	kg.m/rad <sup>2</sup>
Air density	ρ	1.184	kg/m <sup>3</sup>
Propeller radius	R	0.3	m
Propeller area	A	28.3×10 <sup>-2</sup>	m <sup>2</sup>
Lift coefficient	C <sub>T</sub>	0.0047	–
Drag coefficient	C <sub>D</sub>	0.228×10 <sup>-3</sup>	–

Table 2. Control parameters.

Control method	Control gains	Parameter	Value
Backstepping	H = 3		
	K = 2	[c <sub>1</sub> , ..., c <sub>12</sub> ]	[Q, K, Q, K, H, H, 4, 4, H, H]
	Q = 0.5		
CBFL	Proportional	[K <sub>px</sub> , K <sub>py</sub> , K <sub>pz</sub> ]	[2, 2, 2]
	Derivative	[K <sub>dx</sub> , K <sub>dy</sub> , K <sub>dz</sub> ]	[2, 2, 3]
	Integral	[K <sub>ix</sub> , K <sub>iy</sub> , K <sub>iz</sub> ]	[0, 0, 0, 0, 0, 0]
	Stabilizing gains	[c <sub>7</sub> , ..., c <sub>12</sub> ]	[5, 6, 5, 6, 5, 6]



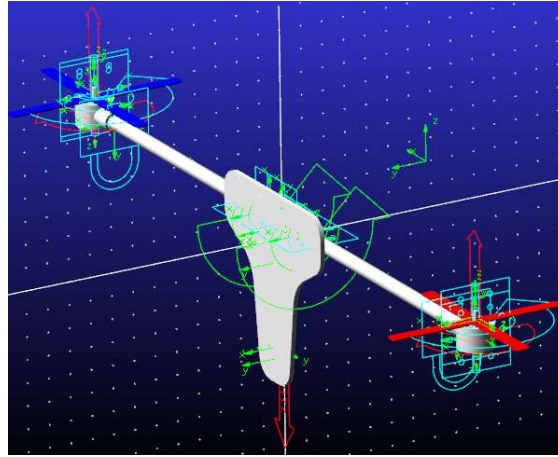


Fig. 5. 3D model of the Bi-rotor in ADAMS software.

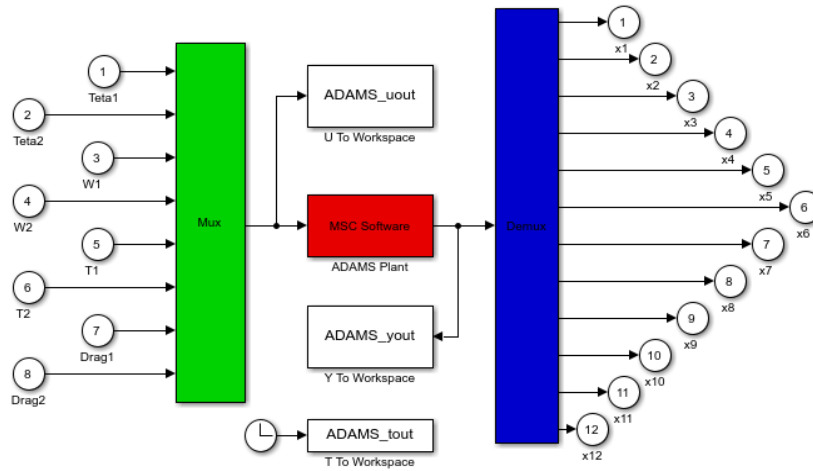


Fig. 6. Block diagram generated for ADAMS/MATLAB CO-simulation.

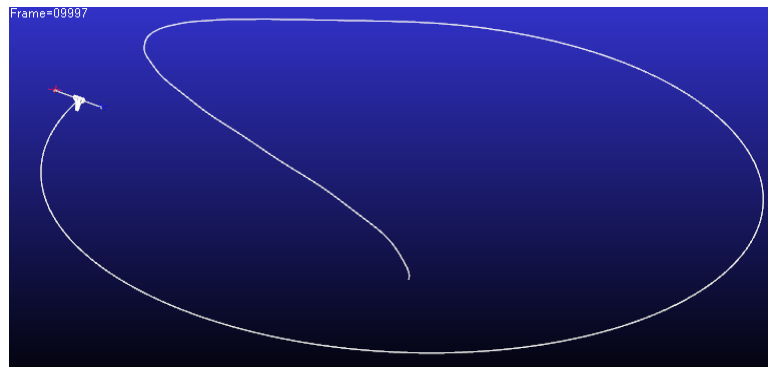


Fig. 7. Circular trajectory tracking.

## 5.2. CO-simulation results

In this section, the joint simulation addresses the tracking of a circular trajectory using the CBFL control method and the reference path equations (90). The flight time for the Bi-rotor to complete one revolution along the circular trajectory is approximately 50 seconds, starting from the moment it takes off from the ground,

$$\begin{aligned} X_r &= 10 \sin \frac{t}{8} \\ Y_r &= 10 \cos \frac{t}{8} \\ Z_r &= 3 \end{aligned} \quad (89)$$

The motion of the Bi-rotor under the CBFL control method and the trajectory followed by the robot are depicted in Fig. 7.

The CBFL control method, using the parameters given in Table 3, is employed in the simulation for circular trajectory tracking. Figure 7 shows that the Bi-rotor accurately follows the reference trajectory.



Table 3. Parameters of the controllers in co-simulation.

Control Method	Control Gain	Parameter	Value
CBFL	Proportional Gains	$[K_{px}, K_{py}, K_{pz}]$	$[2, 2, 2]$
	Derivative Gains	$[K_{dx}, K_{dy}, K_{dz}]$	$[2, 2, 3]$
	Integral Gains	$[K_{ix}, K_{iy}, K_{iz}]$	$[0, 0, 0, 0, 0, 0]$
	Backstepping Gains	$[c_7, \dots, c_{12}]$	$[5, 6, 5, 6, 5, 6]$

The comparison of system errors in tracking a circular trajectory using the Adams software provides a deeper understanding of the system dynamics. Figures 8 and 9 demonstrate that the error signals obtained from the system's mathematical model align closely with the error signals obtained through the Adams software simulation. During the initial phase of motion, the rotor increases its roll angle ( $\varphi$ ) to initiate movement in the y-direction. As a consequence of this roll angle increase, there is a slight oscillation observed in the pitch angle. This oscillation is a result of the interplay between the rotor's roll and pitch motions. To achieve accurate path tracking in the x-direction, the tilt angles experience an initial increase. This response is necessary to ensure that the robot moves along the desired trajectory. By adjusting the tilt angles, the robot can effectively control its position in the x-direction, aligning itself with the circular path. As the robot approaches the desired altitude, corrective measures are taken to maintain stability. The roll and tilt angles are adjusted in the opposite direction to reduce the speed of the robot's motion. This adjustment is crucial in preventing overshooting and ensuring that the robot reaches and maintains the desired altitude with stability. By carefully managing the roll and tilt angles, the system can effectively control its trajectory, minimizing errors in tracking the circular path. These adjustments play a critical role in maintaining stability throughout the motion and ensuring precise path tracking. The comparison between the system's mathematical model and the Adams software simulation highlights the effectiveness of the control strategy in achieving accurate trajectory tracking.

### 5.3. Trajectory tracking control

The comparative analysis of the control methods for the Bi-rotor system provides valuable insights into their performance during circular trajectory tracking. In this section, a Bi-rotor starts its motion from the reference point (0,0,0) on the ground to pursue a circular path at the height of ten meters using the control methods with the reference trajectory (90):

$$\begin{aligned} X_r &= 10 \sin \frac{t}{8} \\ Y_r &= 10 \cos \frac{t}{8} \\ Z_r &= 10 \end{aligned} \quad (90)$$

It is assumed that approximately 50 seconds are allocated for the Bi-rotor to complete one full circle of the circular trajectory, starting from the moment it lifts off the ground. Figure 10 illustrates the Bi-rotor's motion under these control methods.

Figure 11 illustrates the Euler angle variations for the CBFL, backstepping, and feedback linearization control methods. Figure 12 portrays the position errors, which are defined as  $(e_x = x - x_r)$ ,  $(e_y = y - y_r)$ , and  $(e_z = z - z_r)$  for CBFL, backstepping, and feedback linearization control methods.

As depicted, the initial phase of motion is characterized by an increase in the Bi-rotor's tilt angle ( $\varphi$ ) to attain maximum speed. As the Bi-rotor approaches the target altitude, it performs a controlled rotation in the opposite direction around the roll angle. This adjustment is crucial for reducing the speed and ensuring stability at the desired altitude. The CBFL method showcases efficient control in this aspect, enabling quick and precise modifications of the tilt angles. It is observed that the CBFL method exhibits faster and smoother variations in the roll angle compared to the other control methods. This indicates that the CBFL approach can achieve more precise control and stability during the Bi-rotor's motion. Also, the CBFL method demonstrates superior performance, as it exhibits smaller position errors compared to the backstepping and feedback linearization methods. This implies that the CBFL method can accurately track the circular trajectory with minimal deviations. Figure 13 depicts the control inputs of the Bi-rotor under CBFL, backstepping, and feedback linearization control methods.

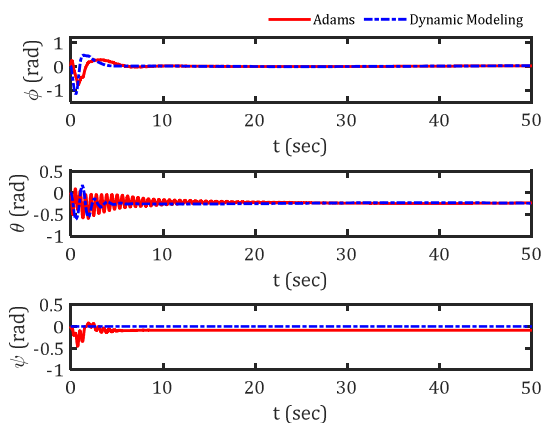


Fig. 8. Comparison of variations in Euler angles.

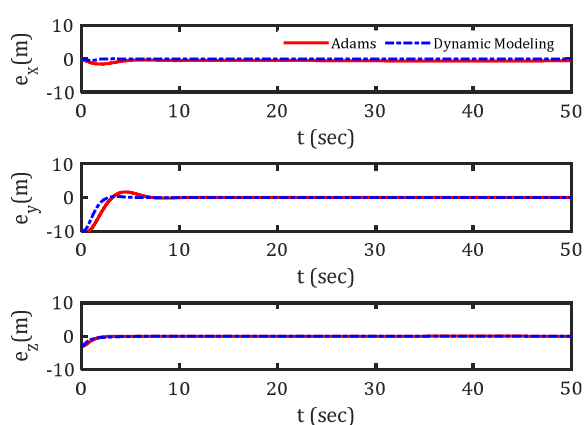


Fig. 9. The comparison of variations in Bi-rotor Euler angles.



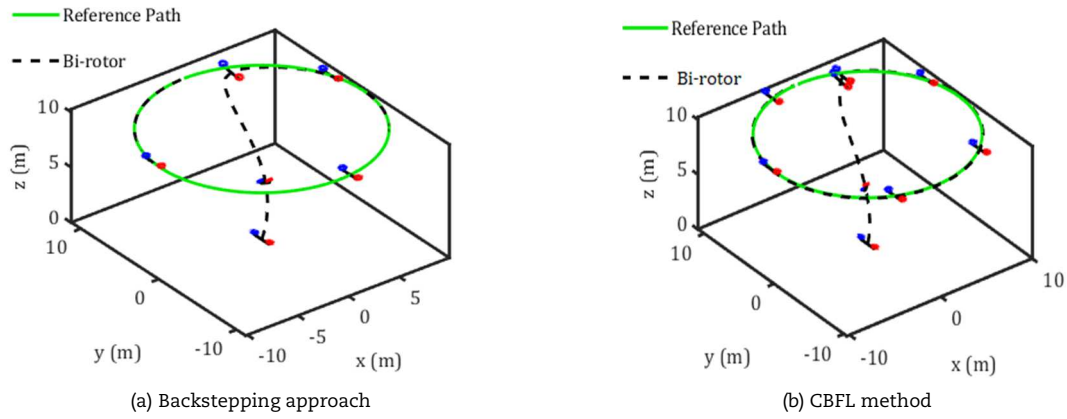


Fig. 10. Circular trajectory tracking for Bi-rotor using CBFL and Backstepping control methods.

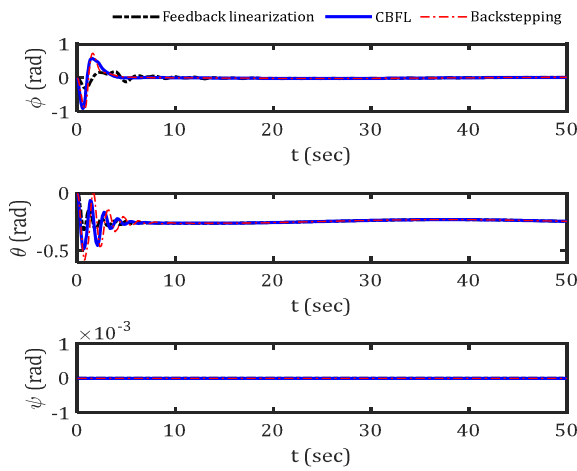


Fig. 11. Comparison of Euler angle variations.

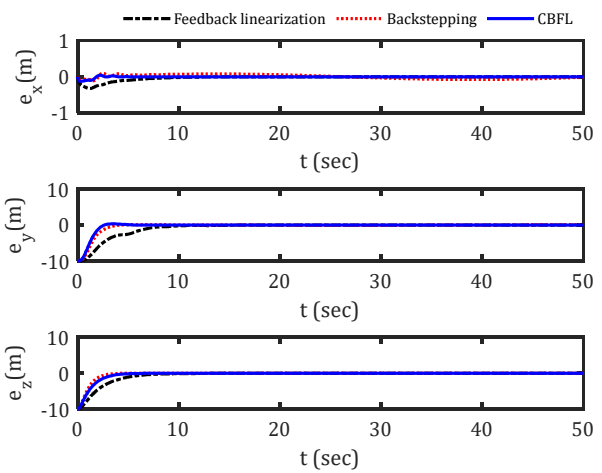


Fig. 12. Comparison of error signals.

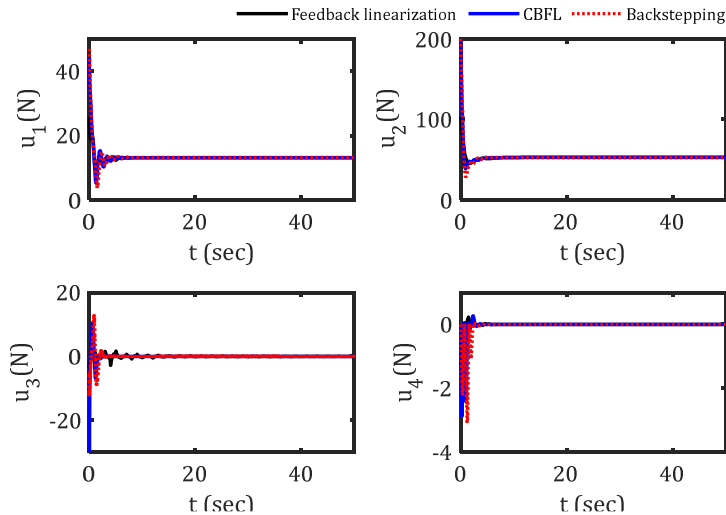


Fig. 13. Comparison of control inputs in Bi-rotor for circular trajectory tracking.

Further analysis in Fig. 13 reveals the control inputs for the Bi-rotor under each control method. The CBFL method exhibits smaller input values compared to the backstepping method, indicating a more efficient utilization of control resources. Despite these smaller inputs, the CBFL method induces faster changes in the Euler angles of the Bi-rotor. This leads to smoother and less oscillatory control inputs, highlighting the effectiveness of the CBFL method in achieving stable and precise control. Figure 14 illustrates the variations in tilt angles for circular trajectory tracking in the CBFL, backstepping, and feedback linearization control methods.

As can be seen, Fig. 14 focuses on the variations in the tilt angles during circular trajectory tracking. The oscillation levels observed in the tilt angles suggest that the CBFL method can effectively minimize these oscillations compared to the backstepping method. The faster response of the CBFL method in adjusting the tilt angles contributes to smoother flight dynamics, reducing undesirable fluctuations during the Bi-rotor's motion.



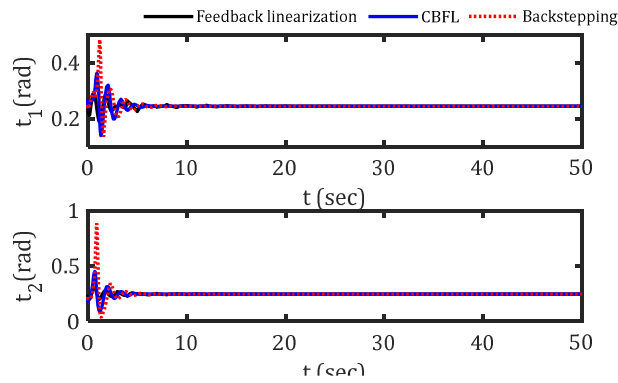


Fig. 14. Variations in tilt angles in circular trajectory tracking.

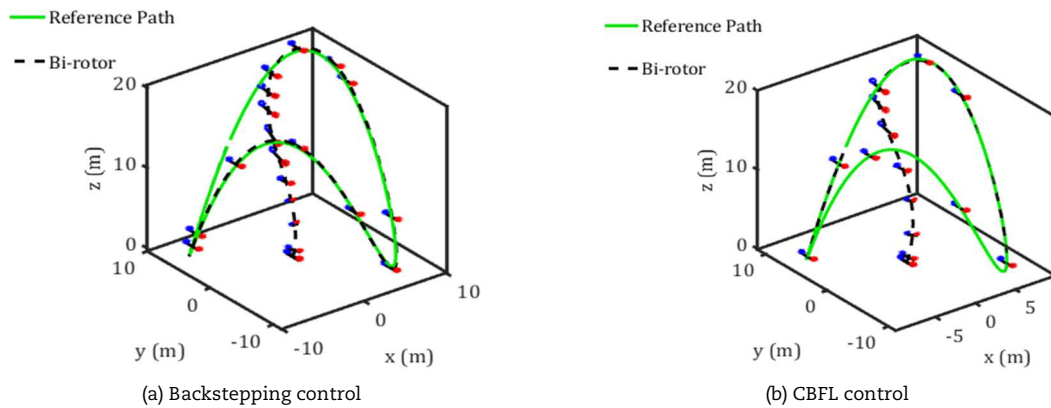


Fig. 15. Tracking paraboloid hyperbolic reference trajectory.

In summary, the comparative analysis of the control methods highlights the advantages of the CBFL approach in circular trajectory tracking for the Bi-rotor system. The CBFL method demonstrates faster, smoother, and more precise control, resulting in reduced position errors, smaller control inputs, and minimized oscillations in the tilt angles. These findings provide valuable insights for optimizing the control strategy of the Bi-rotor system to enhance its performance and stability.

#### 5.4. Tracking paraboloid hyperbolic reference trajectory

In this section, for following a paraboloid hyperbolic trajectory using control methods, including backstepping control, feedback linearization, and a hybrid combination of both approaches (CBFL). The reference path originates from the reference point (0,0,0) on the ground, as defined by equations:

$$\begin{aligned} X_r &= 10 \sin \frac{t}{8} \\ Y_r &= 10 \cos \frac{t}{8} \\ Z_r &= 10 + 10 \sin \frac{t}{4} \end{aligned} \quad (91)$$

Under these control methods, the flight time for the Bi-rotor, from takeoff to completion of one round of the paraboloid hyperbolic path, is approximately 50 seconds. The motion of the Bi-rotor under these control methods is depicted in Fig. 15.

Figure 15 visually depicts the precise guidance of the Bi-rotor toward the reference trajectory. Initially, as the Bi-rotor embarks on its motion, the roll angle ( $\varphi$ ) increases to facilitate movement in the y-direction. Simultaneously, the tilt angles are adjusted to achieve maximum speed in the x-direction. As the Bi-rotor approaches the target altitude, a controlled rotation in the opposite direction around the roll angle is executed. This maneuver serves to reduce speed and ensure stability at the intended height.

To evaluate the performance of the control methods, the position and orientation errors are analyzed in Fig. 16 and Fig. 17, respectively.

Figure 16 provides a comparison of the position errors along the paraboloid hyperbolic path for the Bi-rotor. Notably, the CBFL method demonstrates superior performance, exhibiting reduced steady-state attitude errors when compared to the backstepping and feedback linearization methods. This implies that the CBFL method can achieve more accurate positioning along the paraboloid hyperbolic trajectory, resulting in improved tracking capabilities.

Figure 17 illustrates the variations in Euler angles for the Bi-rotor. It can be observed that the CBFL method outperforms the other control methods, showcasing smoother and more stable variations in the Euler angles. This indicates that the CBFL method can effectively regulate the orientation of the Bi-rotor, ensuring precise control throughout the trajectory.

Figure 18 compares the variations of control inputs over time for all three control methods (backstepping control, feedback linearization, and CBFL) in tracking the paraboloid hyperbolic trajectory.



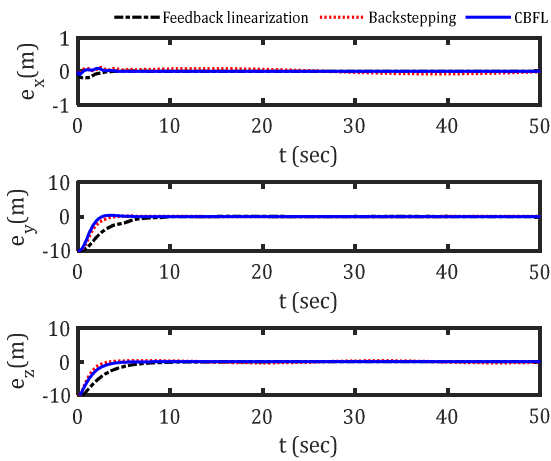


Fig. 16. The comparison of Bi-rotor position errors along the paraboloid hyperbolic path.

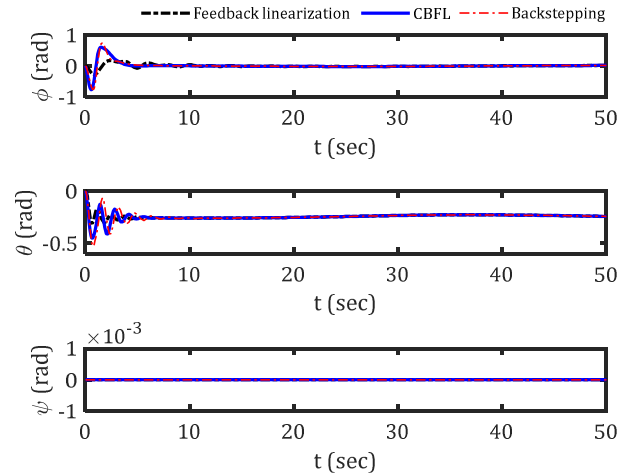


Fig. 17. The comparison of variations in Bi-rotor Euler angles.

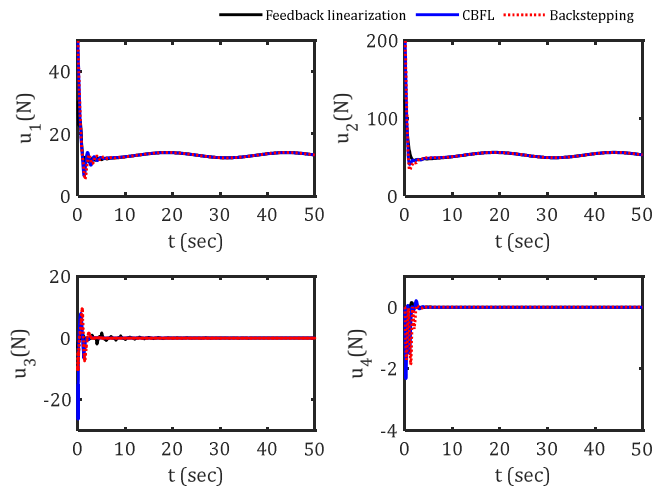


Fig. 18. Control inputs for the Bi-rotor in tracking the paraboloid hyperbolic trajectory.

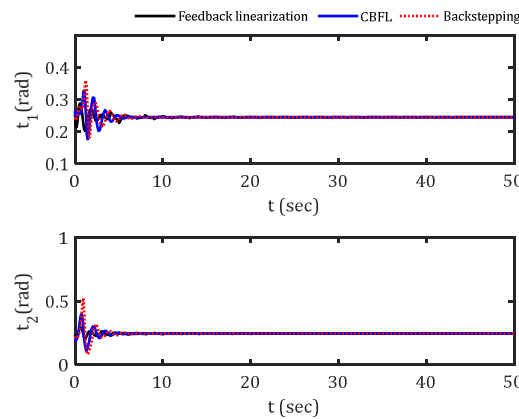


Fig. 19. Bi-rotor tilt angle variations in paraboloid hyperbolic trajectory tracking.

Furthermore, Fig. 18 focuses on the control inputs required by the Bi-rotor for tracking the paraboloid hyperbolic trajectory. Comparing the three control methods (backstepping control, feedback linearization, and CBFL), it is evident that the backstepping control method demands larger control inputs compared to the other two methods. In contrast, the CBFL method demonstrates efficient control with smaller input values. This highlights the superior control capabilities of the CBFL method in guiding the Bi-rotor with less aggressive adjustments.

The time-history of tilt angles for the three control methods is depicted in Fig. 19.

It is apparent that the backstepping control method necessitates larger tilt angle adjustments for the Bi-rotor to align with the desired reference trajectory. On the other hand, the CBFL method exhibits smaller variations in tilt angles, indicating smoother and more stable flight dynamics. A numerical comparison among all three control methods for the Bi-rotor along the paraboloid hyperbolic reference trajectory is presented in Table 4.



**Table 4.** Numerical comparison among all three control methods.

	Backstepping Method	Feedback linearization method	CBFL
$\int_0^{50} u_1^2 dt$	9255	8892	9231
$\int_0^{50} u_2^2 dt$	$1.494 \times 10^5$	$1.431 \times 10^5$	$1.489 \times 10^5$
$\int_0^{50} e_x^2 dt$	0.173	0.056	0.0095
$\int_0^{50} e_y^2 dt$	111.8	197.7	95.22
$\int_0^{50} e_z^2 dt$	84.09	182	95.31
$\int_0^{50} e^2 dt$	196.063	379.756	190.5395

The numerical comparison presented in Table 4 confirms the findings observed in the circular reference trajectory analysis. The CBFL method demonstrates lower positional errors compared to the feedback control method, as similar thrust forces are generated through the propellers. Additionally, the CBFL method experiences fewer oscillations in error variations, further highlighting its effectiveness in trajectory tracking for the Bi-rotor system.

In summary, the comprehensive analysis of the control methods for the Bi-rotor system along the paraboloid hyperbolic trajectory reveals the superiority of the CBFL method. The CBFL method not only reduces position and orientation errors but also requires smaller control inputs and induces smoother tilt angle variations. These findings underscore the effectiveness of the CBFL method in achieving precise trajectory tracking with enhanced stability and control for the Bi-rotor system.

## 6. Conclusions

This paper provided a comprehensive analysis of the modeling, control, stability analysis, and flight capabilities of the Bi-rotor system, with a focus on accurate trajectory tracking. Suitable control methods and gains were applied to achieve precise performance with minimal steady-state errors. The chosen control approaches demonstrated their effectiveness in ensuring accurate trajectory tracking. The dynamic model of the Bi-rotor was validated using a Co-Simulation approach, integrating ADAMS and MATLAB, which rigorously verified its accuracy and reliability. The proposed control strategy not only showcased its effectiveness in trajectory tracking but also held promising potential for addressing other control problems such as obstacle avoidance and control in the presence of external loads. The main achievements of this work included a validated dynamic model, effective motion regulation, stability analysis using the Lyapunov method, and contributions to the advancement of attitude and position control in Bi-rotor drones through nonlinear control techniques. The findings of this study have significant implications for the field of aerial robotics and offer potential applications across various industries.

## Author Contributions

The authors (M. Ganji-Nahoji and A. Keymasi-Khalaji) contributed to this paper equally. All authors shared equal participation in the preparation of the manuscript including design, analysis, discussion, writing, and review.

## Acknowledgments

Not applicable.

## Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

## Funding

The authors received no financial support for the research, authorship, and publication of this article.

## Data Availability Statements

In this study, no datasets were generated, used, or analyzed.

## References


- [1] Moore, M.D., Personal air vehicles: a rural/regional and intra-urban on-demand transportation system, *Journal of the American Institute of Aeronautics and Astronautics*, 2646, 2003, 1-20.
- [2] Ramanathan, V. et al., Warming trends in Asia amplified by brown cloud solar absorption, *Nature*, 448, 2007, 575-578.
- [3] Cavoukian, A., *Privacy and Drones: Unmanned Aerial Vehicles*, Information and Privacy Commissioner of Ontario, Canada, 2012.
- [4] Reeder, J.P., Garren, J.F., Kelly, J.R., A visual flight investigation of hovering and low-speed vtol control requirements low-speed vtol control requirements, in D-2788, NASA, Washington D.C., 1965.
- [5] Marques, P., Da Ronch, A., *Advanced UAV Aerodynamics, Flight Stability and Control: Novel Concepts, Theory and Applications*, Advanced UAV Aerodynamics, Flight Stability and Control: Novel Concepts, Theory and Applications, 1-756, 2016.
- [6] Leishman, J.G., *Principles of Helicopter Aerodynamics*, Cambridge University Press, 2016.
- [7] Gary, G., Natural pitch stabilization of biicopters in hover using lift-propeller gyroscopics, *Journal of Guidance, Control, and Dynamics*, 41(2), 2018, 476-487.
- [8] Sanchez, A., Escareño, J., Garcia, O., Lozano, R., Autonomous Hovering of a Noncyclic Tiltrotor UAV: Modeling, Control and Implementation, *IFAC Proceedings*, 41(2), 2008, 803-808.



- [9] Gary, G., Using Dual Propellers as Gyroscopes for Tilt-Prop Hover Control, in *Biennial International Powered Lift Conference*, Williamsburg, Virginia, 5-7 November 2002.
- [10] Kendoul, F., Fantoni, I., Lozano, R., Modeling and Control of a Small Autonomous Aircraft Having Two Tilting Rotors, *IEEE Transactions on Robotics*, 22(6), 2006, 1297-1302.
- [11] Qimin, Z., Zihe, L., Jieru, Z., Shuguang, Z., Modeling and attitude control of Bi-copter, in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, Beijing, China, 172-176, 2016.
- [12] Albayrak, Ö.B., Ersan, Y., Bagbasi, A.S., Turgut Basaranoglu, A., Arkan, K.B., Design of a Robotic Bicopter, in *2019 7th International Conference on Control, Mechatronics and Automation (ICCM)*, 2019.
- [13] Slotine, J.J.E., Li, W., *Applied Nonlinear Control*, Prentice-Hall, 1991.
- [14] Martins, L., Cardeira, C., Oliveira, P., Feedback Linearization with Zero Dynamics Stabilization for Quadrotor Control, *Journal of Intelligent & Robotic Systems*, 101(1), 2020, 7.
- [15] Amiri, N., Ramirez-Serrano, A., Davies, R.J., Integral Backstepping Control of an Unconventional Dual-Fan Unmanned Aerial Vehicle, *Journal of Intelligent & Robotic Systems*, 69(1), 2013, 147-159.
- [16] Das, A., Lewis, F., Subbarao, K., Backstepping Approach for Controlling a Quadrotor Using Lagrange Form Dynamics, *Journal of Intelligent & Robotic Systems*, 56(1), 2009, 127-151.
- [17] Pereira, J.C., Leite, V.J.S., Raffo, G.V., Nonlinear Model Predictive Control on SE(3) for Quadrotor Aggressive Maneuvers, *Journal of Intelligent & Robotic Systems*, 101(3), 2021, 62.
- [18] González, I., Salazar, S., Lozano, R., Chattering-Free Sliding Mode Altitude Control for a Quad-Rotor Aircraft: Real-Time Application, *Journal of Intelligent & Robotic Systems*, 73(1), 2014, 137-155.
- [19] Razmi, H., Afshinfar, S., Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerospace Science and Technology*, 91, 2019, 12-27.
- [20] Abedini, A., Bataleblu, A.A., Roshanian, J., Robust Backstepping Control of Position and Attitude for a Bi-copter Drone, in *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 17-19 Nov., 2021.
- [21] Keymasi Khalaji, A., Saadat, I., Tracking control of quadrotors in the presence of obstacles based on potential field method, *Amirkabir Journal of Mechanical Engineering*, 53(2), 2021, 1095-1110.
- [22] Spong, M.W., Hutchinson, S., Vidyasagar, M., *Robot modeling and control*. Hoboken, NJ: John Wiley & Sons Hoboken, NJ (in English), 2006.
- [23] Bouabdallah, S., Siegwart, R.Y., Full control of a quadrotor, *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
- [24] Ryan, R.R., ADAMS — Multibody System Analysis Software, in *Multibody Systems Handbook*, W. Schiehlen Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 361-402, 1990.
- [25] Idrissi, M., Annaz, F., Salami, M., Mathematical & Physical Modelling of a Quadrotor UAV, in *2021 7th International Conference on Control, Automation and Robotics (ICCAR)*, 23-26 April, 206-212, 2021.

## ORCID iD

Mohammad Ganji-Nahoji  <https://orcid.org/0009-0009-3267-9031>

Ali Keymasi-Khalaji  <https://orcid.org/0000-0001-5414-6046>



© 2024 Shahid Chamran University of Ahvaz, Ahvaz, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0 license) (<http://creativecommons.org/licenses/by-nc/4.0/>).

**How to cite this article:** Ganji-Nahoji M., Keymasi-Khalaji A. Lyapunov-Based Tracking Control of a Bi-Rotor, *J. Appl. Comput. Mech.*, 10(3), 2024, 503–520. <https://doi.org/10.22055/jacm.2024.45556.4384>

**Publisher's Note** Shahid Chamran University of Ahvaz remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

