

A New Quadrotor Manipulation System: Modeling and Point-to-Point Task Space Control

Ahmed Khalifa* and Mohamed Fanni

Abstract: This article introduces a novel quadrotor manipulation system that consists of 2-link manipulator with unique topology attached to the bottom of a quadrotor. This new system presents a solution for the limitations found in the current quadrotor manipulation system. Unlike the current system, the proposed system enables the end effector to achieve an arbitrary position and orientation with minimum possible number of actuators/links. System kinematics and dynamics are derived. A new closed form inverse kinematics algorithm is presented such that a task space motion controller can be implemented. To study the feasibility of the proposed system, a quadrotor with high enough payload, to add the 2-link manipulator, is designed and constructed. The system controller is designed based on three control techniques: Feedback Linearization based PID (FL-PID), Direct Fuzzy Logic Control (DFLC), and Fuzzy Model Reference Learning Control (FMRLC). Simulation framework is implemented in MATLAB/SIMULINK based on real system parameters to emulate a realistic setup. These controllers are tested under the effect of picking/placing a payload and changing the operating region. Results enlighten the system feasibility, the superior performance of the FMRLC, and the efficiency of the proposed inverse kinematics algorithm.

Keywords: Aerial manipulation, Dynamics, 2-link manipulator, Feedback Linearization, Fuzzy Logic Control, Fuzzy Model Reference Learning Control, Kinematics.

1. INTRODUCTION

Quadrotor is one of the Unmanned Aerial Vehicles (UAVs) which offer possibilities of speed and access to regions that are otherwise inaccessible to ground robotic vehicles. Quadrotor vehicles possess certain essential characteristics, such as small size and cost, Vertical Take Off and Landing (VTOL), and impressive maneuverability, which highlight their potential for use in search and rescue applications [1]. However, most research on UAVs has typically been limited to monitoring and surveillance.

Operation of the quadrotor can be illustrated using Fig. 1. The quadrotor has four rotors mounted in cross configuration. The propeller rotates by angular velocity (Ω_j with $j = 1, 2, 3, 4$), and it produces a thrust force (F_j) and drag moment (M_j) which are directly proportional to Ω_j^2 . Thus, by varying the speeds of each rotor, the flight of the quadrotor is controlled. In order to rotate around x -direction (ϕ), thrust force of rotor 4 must be larger than that of rotor 2 (i.e. $F_4 > F_2$), and consequently, the quadrotor will move in y -direction. Similarly for rotation around y -direction (θ), $F_3 > F_1$, and thus, motion in x -direction will be occurred. As a result, the quadrotor is a

4 DOF system (i.e. it can do 4 independent motions).

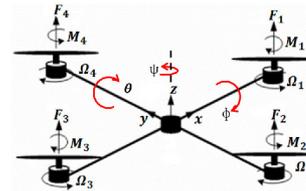


Fig. 1. Schematic diagram of a typical Quadrotor

Due to the superior mobility of the quadrotor systems, much interest is given to utilize them for aerial manipulation. Previous research on aerial manipulation can be divided into three approaches. In the first approach, a gripper/tool is mounted at the bottom of an UAV to transport a payload or interact with existing structures [2, 3]. Accordingly, in these systems, not only the attitude of the payload/tool is restricted to the attitude of the UAV, but also the accessible range of the end-effector is confined due to the fixed configuration of the gripper/tool with respect to the UAV body and blades. Consequently, the resulting aerial system has 4-DOF; three translational DOF and one rotational DOF (Yaw),

Ahmed Khalifa and Mohamed Fanni are with the Department of Mechatronics and Robotics Engineering, Egypt-Japan University of Science and Technology, New Borg El-Arab City, Alexandria, Egypt (e-mail: {mohamed.fanni, ahmed.khalifa}@ejust.edu.eg).

Mohamed Fanni is on leave from Department of Production Engineering and Mechanical Design, Mansoura University, Mansoura, Egypt.

* Corresponding author.

i.e., the gripper/tool cannot possess pitch or roll rotation without moving horizontally. The second approach is to suspend a payload with cables [4]. However, this approach has a drawback that the movement of the payload cannot be always regulated directly because the manipulation is achieved using a cable which cannot always move the payload as desired.

To overcome these limitations, a third approach is developed in which an aerial vehicle is equipped with a robotic manipulator. For example, in [5], a test bed including two 4-DOF robot arms and a crane emulating an aerial robot is proposed. When employing the robotic manipulator, the dynamics of the robotic manipulator is highly coupled with that of the aerial vehicle, which should be carefully considered in the controller design. Very few reports exist in the literature that investigate the combination of aerial vehicle with robotic manipulator. Kinematic and dynamic models of the quadrotor combined with arbitrary multi-DOF robot arm are derived using the Euler-Lagrangian formalism in [6]. Moreover, the effects of manipulator, two 4-DOF arms, on the quadrotor are simulated based on the dynamic model which considers a quadrotor and robotic arms separately. In [7], a quadrotor with 1-DOF robotic arm is presented. In [8], a quadrotor with light-weight manipulators, three 2-DOF arms, are tested, although the movement of the manipulators is not explicitly considered during the design of the PID controller. In [9], an aerial manipulation using a quadrotor with a 2-DOF robotic arm is presented but with certain topology that disable the system from making an arbitrary position and orientation of the end-effector. In this system, the axes of the manipulator joints are parallel to each other and parallel to one in-plane axis of the quadrotor. Thus, the system cannot achieve orientation around the second in-plane axis of the quadrotor without moving horizontally. In [10, 11], the authors propose a new aerial manipulation system that consists of 2-link manipulator with unique topology, with two revolute joints whose axes are perpendicular to each other and the axis of the first joint is parallel to one in-plane axis of the quadrotor. Thus, the end-effector is able to reach arbitrary position and orientation without moving horizontally.

In this paper, a quadrotor is selected and a manipulator with two revolute joints is designed. The whole system is connected. Kinematic and dynamic models of system are derived. Furthermore, novel solution to the problem of motion control in the task space of the quadrotor manipulation system are introduced. This inverse kinematics analysis is utilized in order to achieve point-to-point task space control of the proposed quadrotor manipulation system based on three control techniques.

2. DESIGN OF THE PROPOSED SYSTEM

In this section, a complete design of the proposed system is carried out. There are three aims of making such design. The first aim is to prove that the proposed system is realizable. The second aim is to get the geometrical and inertia parameters from the CAD model of the designed system, and use them in the designing and simulating the control system. The third aim is to build the system and make identification experiments to get the aerodynamic parameters needed for simulation as well as the mapping between the control signal and Pulse Width Modulation (PWM) needed for real implementation.

The structure of the proposed system is shown in Fig. 2. The axis of the first revolute joint (z_0) that is fixed with respect to the quadrotor is parallel to the body x -axis of the quadrotor (see Fig. 3). The axis of the second joint (z_1) will be parallel to the body y -axis of quadrotor when the first link is vertical. So, the pitching and rolling rotation of the end effector is now possible independently on the horizontal motion of the quadrotor. With this new system, the capability of manipulating objects with arbitrary location and orientation is achieved because the DOF are increased from 4 to 6.

The proposed quadrotor manipulation system consists mainly from two parts; the quadrotor and the manipulator. During the reset of this section, design of each of them is presented such that they can combined and perform the required task.



Fig. 2. 3D CAD model of the New Quadrotor Manipulation System

2.1. The Two-Link Manipulator

The design of this manipulator is based on light weight and enough workspace under the quadrotor.

Our target is to design a light and simple 2 DOF manipulator that can carry as much as possible of a payload.

The arm components are selected, purchased and assembled such that the total weight of arm is 200 g and can carry a payload of 200 g. The arm components are:

- Three servo motors: HS-422 for gripper, HS-5485HB for joint 1, and HS-422 for joint 2.
- Serial servo controller (SSC-32): Interface between the main control unit and the servo motors.
- Motor accessories: Aluminum Tubing - 1.50 in, Aluminum Multi-Purpose Servo Bracket Two Pack,

Aluminum Tubing Connector Hub, and Aluminum Long "C" Servo Bracket with Ball Bearings Two Pack.

2.2. Quadrotor

The quadrotor components are selected, purchased and assembled such that it can carry payload of 500 g (larger than the total arm weight including the payload value). The quadrotor components are:

- Airframe: Mechanical structure of an aircraft that supports all the components, "ST450 metal folding".
- Rotor Assembly: Propeller (EPP1045), Electric Motor (930KV ST2812), and the Electronic Speed Controller (Lulin 30 A). It can produce approximately 400 g of thrust force.
- Microcontroller Unit: Implementation of stabilization control algorithms, Arduino Mega 2560.
- Wireless Communication: Two XBee modules are going to be used: one for the quadrotor and another in the ground station computer that will handle all telemetry for system identification and control purposes, Zigbee Pro- 63 mW PCB Antenna Series2.
- Sensors: Providing information like aircraft attitude, acceleration, altitude, global position. Inertial Measurement Unit (10 DOF Multiwii ZMR), Sonar unit (SRF04), and GPS unit (SKM53).
- Battery: Lithium Polymer Battery is used to power both the electronics components and the motors.

3. KINEMATIC ANALYSIS

Fig. 3 presents a sketch of the Quadrotor-Manipulator System with the relevant frames. The frames are assumed to satisfy the Denavit-Hartenberg (DH) convention [12].

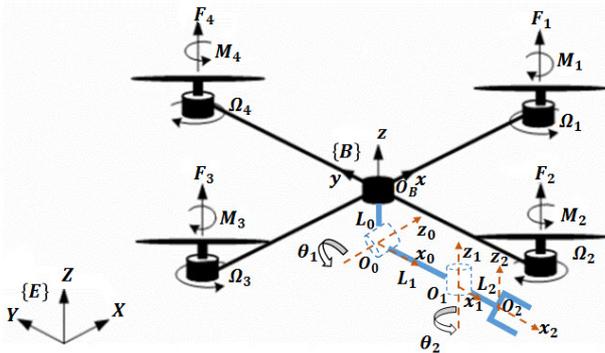


Fig. 3. Schematic of Quadrotor Manipulation System frames

The rotational kinematics of the quadrotor is represented through Euler angles. The can be completely described by its position and orientation with respect to reference earth-fixed and inertial frame $\{E\}$, $O_I-X Y$

Z . Let define $\eta_1 = [X, Y, Z]^T$ as the vector of the body position coordinates in the earth-fixed reference frame. The vector $\dot{\eta}_1$ is the corresponding time derivative. If one defines $v_1 = [u, v, w]^T$ as the linear velocity of the origin of the body-fixed frame $\{B\}$, $O_B-x y z$, whose origin is coincident with the center of mass (CM), with respect to the origin of the earth-fixed frame expressed in the body-fixed frame, the following relation between the defined linear velocities holds:

$$v_1 = R_I^B \dot{\eta}_1 \quad (1)$$

where R_I^B is the rotation matrix expressing the transformation from the inertial frame to the body-fixed frame and it is given by

$$R_I^B = \begin{bmatrix} C_\psi C_\theta & S_\psi C_\theta & -S_\theta \\ -S_\psi C_\phi + S_\psi S_\theta C_\psi & C_\psi C_\phi + S_\psi S_\theta S_\psi & C_\theta S_\phi \\ S_\psi S_\phi + C_\psi S_\theta C_\phi & -C_\psi S_\phi + S_\psi S_\theta C_\phi & C_\theta C_\phi \end{bmatrix} \quad (2)$$

Let define $\eta_2 = [\phi, \theta, \psi]^T$ as the vector of body Euler-angle coordinates in an earth-fixed reference frame. Those are commonly named roll, pitch and yaw angles and corresponds to the elementary rotation around X , Y and Z in fixed frame. The vector $\dot{\eta}_2$ is the corresponding time derivative (expressed in the inertial frame). Let define $v_2 = [p, q, r]^T$ as body-fixed angular velocity. The vector $\dot{\eta}_2$ is related to the body-fixed angular velocity by a proper Jacobian matrix:

$$v_2 = J_v \dot{\eta}_2 \quad (3)$$

The matrix J_v can be expressed in terms of Euler angles as:

$$J_v = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \quad (4)$$

Note that C and S are short notations for $\cos(\cdot)$ and $\sin(\cdot)$, respectively. The DH parameters for the 2-Link Manipulator are derived and presented in [10].

The position and orientation of the end effector relative to the body-fixed frame is easily obtained by multiplying the following homogeneous transformation matrices A_0^B , A_1^0 , A_2^1 that can be found from the DH method [12]. where θ_1 and θ_2 are the manipulator joints' angles.

3.1. Forward Kinematics

Let define the position and orientation of the end effector expressed in the inertial frame, as $\eta_{ee_1} = [x_{ee}, y_{ee}, z_{ee}]^T$ and $\eta_{ee_2} = [\phi_{ee}, \theta_{ee}, \psi_{ee}]^T$ respectively.

The forward kinematics problem consists of determining the operational coordinates (η_{ee_1} and η_{ee_2}) of the end effector, as a function of the quadrotor movements (X, Y, Z , and ψ) as well as the motion of the manipulator's joints (θ_1 and θ_2). This problem is solved by computing the homogeneous transformation matrix composed of relative translations and rotations.

The transformation matrix from the body frame to the inertial frame A_B^I which is:

$$A_B^I = R_B^I * transl(X, Y, Z) \quad (5)$$

where R_B^I is 4×4 matrix, and $transl(X, Y, Z)$ is 4×4 matrix that describes the translation of X , Y and Z in the inertial coordinates. The total transformation matrix that relates the end effector frame to the inertial frame is T_2^I , which is given by:

$$T_2^I = A_B^I A_0^B A_1^0 A_2^1 \quad (6)$$

Define the general form for this transformation matrix as a function of end effector variables (η_{ee_1} and η_{ee_2}), as following:

$$T_{ee} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_{ee} \\ r_{21} & r_{22} & r_{23} & y_{ee} \\ r_{31} & r_{32} & r_{33} & z_{ee} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Equating (6) and (7), an expression for the parameters of T_{ee} (r_{ij} , x_{ee} , y_{ee} , and z_{ee} ; $i, j = 1, 2, 3$) can be found, from which values of the end effector variables can be determined. Euler angles of the end effector (ϕ_{ee} , θ_{ee} and ψ_{ee}) can be computed from the rotation matrix of T_{ee} as in [13].

3.2. Inverse Kinematics

The inverse kinematics problem consists of determining the quadrotor movements (X , Y , Z , and ψ) as well as the motion of the manipulator's joints (θ_1 and θ_2) as function of operational coordinates (η_{ee_1} and η_{ee_2}) of the end effector.

The inverse kinematics solution is essential for the robot's control, since it allows to compute the required quadrotor movements and manipulator joints angles to move the end effector to a desired position and orientation.

The rotations of the end effector can be parameterized by using several methods one of them, that is chosen, is the Euler angles [13].

Equation (6) can be expressed, after putting $\phi = \theta = 0$, since the target is to find inverse kinematics for reset position, as following:

$$T_2^I = \begin{bmatrix} C_\psi S_{\theta_2} + C_{\theta_1} C_{\theta_2} S_\psi & C_\psi C_{\theta_2} - C_{\theta_1} S_\psi S_{\theta_2} & S_\psi S_{\theta_2} & X + L_1 C_{\theta_1} S_\psi + L_2 C_\psi S_{\theta_2} + L_2 C_{\theta_1} C_{\theta_2} S_\psi \\ S_\psi S_{\theta_2} - C_\psi C_{\theta_1} C_{\theta_2} & C_{\theta_2} S_\psi + C_\psi C_{\theta_1} S_{\theta_2} & -C_\psi S_{\theta_2} & Y - L_1 C_\psi C_{\theta_1} + L_2 S_\psi S_{\theta_2} - L_2 C_\psi C_{\theta_1} C_{\theta_2} \\ -C_{\theta_2} S_{\theta_1} & S_{\theta_1} S_{\theta_2} & C_{\theta_1} & Z - L_0 - L_1 S_{\theta_1} - L_2 C_{\theta_2} S_{\theta_1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

From (8) and (7), the inverse kinematics of the system can be derived. According to the structure of (8), the inverse orientation is carried out first followed by the inverse position. The inverse orientation has three cases as following: **CASE 1:** Suppose that not both of r_{13} and r_{23} are zero. Then from (8), we deduce that $\sin(\theta_1) \neq 0$ and $r_{33} \neq \pm 1$. Then, $\cos(\theta_1) = r_{33}$ and $\sin(\theta_1) = \pm \sqrt{1 - r_{33}^2}$ and thus,

$$\theta_1 = atan2(\pm \sqrt{1 - r_{33}^2}, r_{33}) \quad (9)$$

and

$$\psi = atan2(\pm r_{13}, \mp r_{23}) \quad (10)$$

$$\theta_2 = atan2(\pm r_{32}, \mp r_{31}) \quad (11)$$

Thus, there are two solutions depending on the sign chosen for $\sin(\theta_1)$. If $r_{13} = r_{23} = 0$, then the fact that T_e is orthogonal implies that $r_{33} = \pm 1$.

CASE 2: If $r_{13} = r_{23} = 0$ and $r_{33} = 1$, then $\cos(\theta_1) = 1$ and $\sin(\theta_1) = 0$, so that $\theta_1 = 0$. In this case, from the rotation matrix of (8), the sum $\theta_2 + \psi$ can be determined as:

$$\theta_2 + \psi = atan2(r_{11}, r_{12}) \quad (12)$$

We can assume any value for ψ and get θ_2 . Therefore, there are infinite number of solutions.

CASE 3: If $r_{13} = r_{23} = 0$ and $r_{33} = -1$, then $\cos(\theta_1) = -1$ and $\sin(\theta_1) = 0$, so that $\theta_1 = \pi$. In this case, from (8), $\theta_2 - \psi$ can be determined as:

$$\theta_2 - \psi = atan2(r_{11}, r_{12}) \quad (13)$$

One can assume any value for ψ and get θ_2 . Therefore, there are infinite number of solutions. In cases 2 and 3, one may put $\psi = 0$ and get θ_2 . Finally, the inverse position is determined from:

$$X = x_{ee} - (L_1 C_{\theta_1} S_\psi + L_2 C_\psi S_{\theta_2} + L_2 C_{\theta_1} C_{\theta_2} S_\psi) \quad (14)$$

$$Y = y_{ee} - (-L_1 C_\psi C_{\theta_1} + L_2 S_\psi S_{\theta_2} - L_2 C_\psi C_{\theta_1} C_{\theta_2}) \quad (15)$$

$$Z = z_{ee} - (-L_0 - L_1 S_{\theta_1} - L_2 C_{\theta_2} S_{\theta_1}) \quad (16)$$

4. DYNAMIC ANALYSIS

In Fig. 4, a block diagram that shows the effect of adding a manipulator to a quadrotor is presented.

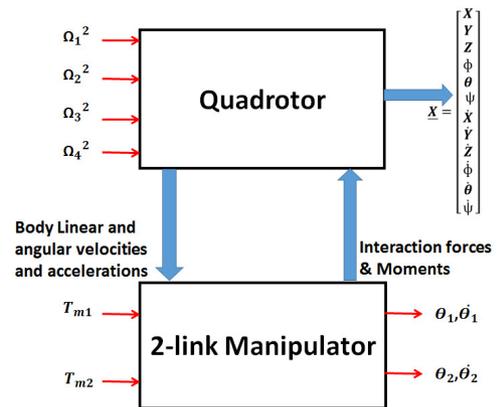


Fig. 4. Effects of adding a manipulator to the quadrotor

For the manipulator dynamics, Recursive Newton Euler method [14] is used to derive the equations of motion. Since the quadrotor is considered to be the base of the manipulator, the initial linear and angular velocities and accelerations, used in Newton Euler algorithm, are that of the quadrotor expressed in body frame. Applying the Newton Euler algorithm to the manipulator considering that the link (with length L_0) that is fixed to the quadrotor is the base link, manipulator's equations of motion can be obtained, in addition to, the forces and moments, from manipulator, that affect the quadrotor.

Let us define for each link i , the following variables:

ω_i^i , angular velocity of frame i expressed in frame i , $\dot{\omega}_i^i$, angular acceleration of frame i , v_i^i , linear velocity of the origin of frame i , $\dot{v}_{c_i}^i$, linear acceleration of the center of mass of link i , \dot{v}_i^i , linear acceleration of the origin of frame i , r_i^i , the vector from the origin of frame $(i-1)$ to the origin of link i , $r_{c_i}^i$, the vector from the origin of frame $(i-1)$ to the center of mass of link i , g^I , the vector of gravity expressed in inertial frame I , $z_{(i-1)}^{(i-1)}$, is a unit vector pointing along the i^{th} joint axis and expressed in the $(i-1)^{th}$ link coordinate system, $R_i^{(i-1)}$, rotation matrix from frame i to frame $(i-1)$, I_i^i , the inertia matrix of link i about its center of mass coordinate frame, and $f_{(i,i-1)}^i / n_{(i,i-1)}^i$ is the resulting force/moment exerted on link i by link $(i-1)$ at point $O_{(i-1)}$, where $i = 1, 2$.

For the link 0:

$$\omega_0^0 = R_I^0 v_2 \quad (17)$$

$$\dot{\omega}_0^0 = R_I^0 \dot{v}_2 \quad (18)$$

$$\begin{aligned} v_0^0 &= R_B^0 v_1 + \omega_0^0 \times r_0^0, \\ r_0^0 &= [0, L_0, 0]^T \end{aligned} \quad (19)$$

$$\dot{v}_0^0 = R_B^0 \dot{v}_1 + \dot{\omega}_0^0 \times r_0^0 + \omega_0^0 \times (\omega_0^0 \times r_0^0) \quad (20)$$

For link i ($i = 1, 2$), one can calculate the following variables:

$$\omega_i^i = R_{i-1}^i (\omega_{i-1}^{i-1} + \dot{\theta}_i z_{i-1}^{i-1}) \quad (21)$$

$$\dot{\omega}_i^i = R_{i-1}^i (\dot{\omega}_{i-1}^{i-1} + \ddot{\theta}_i z_{i-1}^{i-1} + \omega_{i-1}^{i-1} \times \dot{\theta}_i z_{i-1}^{i-1}) \quad (22)$$

$$v_i^i = R_{i-1}^i v_{i-1}^{i-1} + \omega_i^i \times r_i^i \quad (23)$$

$$\dot{v}_i^i = R_{i-1}^i \dot{v}_{i-1}^{i-1} + \dot{\omega}_i^i \times r_i^i + \omega_i^i \times (\omega_i^i \times r_i^i) \quad (24)$$

$$\dot{v}_{c_i}^i = \dot{v}_i^i + \dot{\omega}_i^i \times r_{c_i}^i + \omega_i^i \times (\omega_i^i \times r_{c_i}^i) \quad (25)$$

The inertial forces and moments acting on link i are given by:

$$F_i^i = -m_i \dot{v}_{c_i}^i \quad (26)$$

$$N_i^i = -I_i^i \dot{\omega}_i^i - \omega_i^i \times I_i^i \omega_i^i \quad (27)$$

The total forces and moments acting on link i are given by:

$$f_{i,i-1}^i = f_{i+1,i}^i - m_i g_i - F_i^i \quad (28)$$

$$n_{i,i-1}^i = n_{i+1,i}^i + (f_i^i + r_{c_i}^i) \times f_{i,i-1}^i - r_{c_i}^i \times f_{i+1,i}^i - N_i^i \quad (29)$$

$$f_{i,i-1}^{i-1} = R_i^{i-1} f_{i,i-1}^i \quad (30)$$

$$n_{i,i-1}^{i-1} = R_i^{i-1} n_{i,i-1}^i \quad (31)$$

where $z_0^0 = [0, 0, 1]^T$, $r_1^1 = [L_1, 0, 0]^T$, $r_{c_1}^1 = [-L_1/2, 0, 0]^T$, $z_1^1 = [0, 0, 1]^T$, $r_2^2 = [L_2, 0, 0]^T$, and $r_{c_2}^2 = [-L_2/2, 0, 0]^T$. The gravity vector expressed in frames 1 and 2 are:

$$\begin{aligned} g^2 &= R_I^2 g^I, \\ g^I &= [0, 0, -g]^T \end{aligned} \quad (32)$$

$$g^1 = R_I^1 g^I \quad (33)$$

where,

$$R_I^1 = R_0^1 R_B^0 R_I^B \quad (34)$$

$$R_I^2 = R_1^2 R_I^1 \quad (35)$$

where m_i and L_i are the mass and length of link i .

The torques acting on joints 1 and 2 are finally given by:

$$T_{m_1} = (n_{1,0}^0)^T z_0^0 + b_1 \dot{\theta}_1 \quad (36)$$

$$T_{m_2} = (n_{2,1}^1)^T z_1^1 + b_2 \dot{\theta}_2 \quad (37)$$

where, b_1 and b_2 are friction coefficients.

The interaction forces and moments of the manipulator acting on the quadrotor expressed in body frame, $F_{m,q}^B$ and $M_{m,q}^B$ are given as follows:

$$\begin{bmatrix} F_{m,q}^B \\ M_{m,q}^B \end{bmatrix} = \begin{bmatrix} R_0^B & 0_{3 \times 3} \\ \text{skew}(P_{B0}^B)R_0^B & R_0^B \end{bmatrix} \begin{bmatrix} f_{1,0}^0 \\ n_{1,0}^0 \end{bmatrix} \quad (38)$$

where, $\text{skew}(\cdot)$ is skew symmetric matrix [12] of $P_{B0}^B = [0, 0, -L_0]^T$, which is the position vector of the origin O_0 relative to frame B . The interaction forces expressed in the inertial frame are:

$$F_{m,q}^I = R_B^I F_{m,q}^B \quad (39)$$

Applying a payload of mass m_p (see Fig. 5) will change link 2's parameters such as mass moments of inertia, total mass of this link, and center of gravity of this link as following:

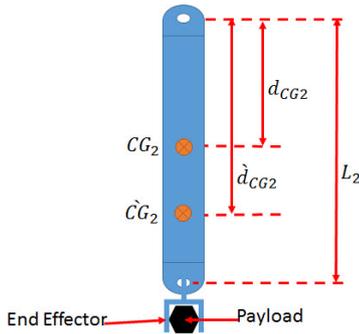


Fig. 5. Schematic diagram of link 2 after adding the payload

$$\dot{I}_2^2 = I_2^2 + m_2(d_{CG_2} - d_{CG_2})^2 + m_p(L_2 - d_{CG_2})^2 \quad (40)$$

$$d_{CG_2} = \frac{m_2 d_{CG_2} + m_p L_2}{m_2 + m_p} \quad (41)$$

$$\dot{m}_2 = m_2 + m_p \quad (42)$$

where CG_2 is the point of center of gravity of link 2, I_2^2 is the inertia matrix of link 2 about its center of mass coordinate frame, m_2 is mass of link 2, and (\cdot) refers to the value of the parameter after adding the payload.

Changing the point of center of gravity of link 2 will change the $r_{c_2}^2$ (vector from the origin of frame $\{1\}$ to the center of mass of link 2) to be

$$r_{c_2}^2 = [-(L_2 - d_{CG_2}), 0, 0]^T \quad (43)$$

Substituting from (40, 42 and 43) in the dynamic equations of the system, one can study the effect of carrying the payload (m_p).

The equations of motion of the manipulator are:

$$M_1 \ddot{\theta}_1 = T_{m_1} + N_1 \quad (44)$$

$$M_2 \ddot{\theta}_2 = T_{m_2} + N_2 \quad (45)$$

where, T_{m_1} and T_{m_2} are the manipulator-actuators' torques. M_1 , M_2 , N_1 , and N_2 are nonlinear terms and they are functions in the system states $(\eta_2, v_2, \dot{v}_2, \dot{v}_1, \theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$.

The equations of motion of the quadrotor after adding the forces/moments applied by the manipulator are:

$$m \ddot{X} = T(C_\psi S_\theta C_\phi + S_\psi S_\phi) + F_{m,q_x}^I \quad (46)$$

$$m \ddot{Y} = T(S_\psi S_\theta C_\phi - C_\psi S_\phi) + F_{m,q_y}^I \quad (47)$$

$$m \ddot{Z} = -mg + TC_\theta C_\phi + F_{m,q_z}^I \quad (48)$$

$$I_x \ddot{\phi} = \dot{\theta} \dot{\phi} (I_y - I_z) - I_r \dot{\theta} \Omega + T_{a_1} + M_{m,q_\phi}^B \quad (49)$$

$$I_y \ddot{\theta} = \dot{\psi} \dot{\phi} (I_z - I_x) + I_r \dot{\phi} \Omega + T_{a_2} + M_{m,q_\theta}^B \quad (50)$$

$$I_z \ddot{\psi} = \dot{\theta} \dot{\phi} (I_x - I_y) + T_{a_3} + M_{m,q_\psi}^B \quad (51)$$

where F_{m,q_x}^I , F_{m,q_y}^I , and F_{m,q_z}^I are the interaction forces from the manipulator to the quadrotor in X, Y , and Z directions defined in the inertial frame, and M_{m,q_ϕ}^B , M_{m,q_θ}^B , and M_{m,q_ψ}^B are the interaction moments from the manipulator to the quadrotor around X, Y , and Z directions defined in the body frame.

The variables in the above equations are defined as following: m is the mass of the quadrotor. T is the total thrust applied to the quadrotor from all four rotors. τ_{a_1} , τ_{a_2} , and τ_{a_3} are the three input moments about the three body axes, These moments are the rolling, pitching, yawing moment about x -, y -, and z -axis of the body frame respectively, and they are given as:

$$\begin{bmatrix} T \\ \tau_{a_1} \\ \tau_{a_2} \\ \tau_{a_3} \end{bmatrix} = \underbrace{\begin{bmatrix} K_{F_1} & K_{F_2} & K_{F_3} & K_{F_4} \\ 0 & -dK_{F_2} & 0 & dK_{F_4} \\ -dK_{F_1} & 0 & dK_{F_3} & 0 \\ -K_{M_1} & K_{M_2} & -K_{M_3} & K_{M_4} \end{bmatrix}}_G \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (52)$$

d is the distance between the quadrotor center of mass and the rotation axis of the propeller. K_{F_j} and K_{M_j} are the thrust and drag coefficients of rotor j . $\bar{\Omega}$ is given by:

$$\bar{\Omega} = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \quad (53)$$

I_r is the rotor inertia. I_f is the inertia matrix of the vehicle around its body-frame assuming that the vehicle is symmetric about x -, y - and z -axis.

In order to test the feasibility of the proposed system, a simulation environment will be built. Thus, there is a need to find the real parameters of the system to make the simulation results more accurate and reliable. The authors in [15] describe a methodology to identify all the parameters of the proposed system. The identified parameters include the structure parameters and rotor assembly parameters (K_F and K_M). These parameters will be used in the system simulation and controller design later. A CAD model is developed using SOLIDWORKS to calculate the mass moments of inertia and all the missing geometrical parameters. Three simple test rigs are built and used to identify the relationship between the motor input Pulse Width Modulation (PWM) signal and the angular velocities, the thrust forces, and drag moments of the rotors. A simple algorithm is implemented to an IMU for estimating the attitude and altitude of the quadrotor. Experimental set up is built to verify and test the accuracy of these estimation and the identification techniques. This is achieved by testing a controller designed based on feedback linearization method to stabilize the quadrotor attitude. The experimental results show that the controller succeeds to achieve attitude stabilization and hence prove the accuracy of the estimated parameters. The identified parameters are given in Table 1.

Table 1. System parameters

Par.	Value	Unit	Par.	Value	Unit
m	1	kg	L_2	85×10^{-3}	m
d	223.5×10^{-3}	m	m_0	30×10^{-3}	kg
I_x	13.215×10^{-3}	$N.m.s^2$	m_1	55×10^{-3}	kg
I_y	12.522×10^{-3}	$N.m.s^2$	m_2	112×10^{-3}	kg
I_z	23.527×10^{-3}	$N.m.s^2$	I_r	33.216×10^{-6}	$N.m.s^2$
L_0	30×10^{-3}	m	L_1	70×10^{-3}	m
K_{F1}	1.667×10^{-5}	$kg.m.rad^{-2}$	K_{F2}	1.285×10^{-5}	$kg.m.rad^{-2}$
K_{F3}	1.711×10^{-5}	$kg.m.rad^{-2}$	K_{F4}	1.556×10^{-5}	$kg.m.rad^{-2}$
K_{M1}	3.965×10^{-7}	$kg.m^2.rad^{-2}$	K_{M2}	2.847×10^{-7}	$kg.m^2.rad^{-2}$
K_{M3}	4.404×10^{-7}	$kg.m^2.rad^{-2}$	K_{M4}	3.170×10^{-7}	$kg.m^2.rad^{-2}$

5. CONTROLLER DESIGN

Quadrotor is an under-actuated system, because it has four inputs (angular velocities of its four rotors) and six variables to be controlled. By observing the operation of the quadrotor, one can find that the movement in X -direction is based on the pitch rotation, θ . Also, the movement in Y -direction is based on the roll rotation, ϕ . Therefore, motion along X - and Y -axes will be controlled

through controlling θ and ϕ .

Fig. 6 presents a block diagram of the proposed control system. The desired values for the end effector's position (x_{ee_d} , y_{ee_d} and z_{ee_d}) and orientation (ϕ_{ee_d} , θ_{ee_d} and ψ_{ee_d}) are converted to the desired values of the quadrotor (X_d , Y_d , Z_d and ψ_d) through the inverse kinematics that are derived in Sect. 3.. Next, these values are applied to a trajectory generation algorithm which will be explained later. After that, the controller block receives the desired values and the feedback signals from the system and provides the control signals (T , τ_{a1} , τ_{a2} , τ_{a3} , T_{m1} and T_{m2}). The inverse of matrix G of the control mixer, in Fig. 6, is used to transform the assigned thrust force and moments of the quadrotor (the control signals) from the controller block into assigned angular velocities of the four rotors.

Finally, The actual values of the quadrotor and joints are converted to the actual values of the end effector variables through the forward kinematics which are derived in Sect. 3..

The control design criteria are to achieve system stability and zero position error, for the movements in X , Y , Z , and ψ directions as well as for joints' angles θ_1 and θ_2 and consequently for the end effector variables (η_{ee1} and η_{ee2}), under the effect of:

- Picking/placing a payload.
- Changing the operating region of the system.

Noting that in the task space, a position tracking is implemented, and in the joint space, trajectory tracking is required.

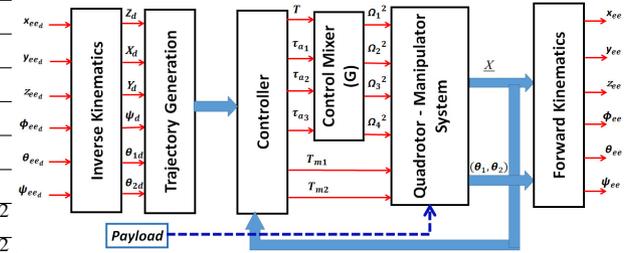


Fig. 6. Block diagram of the control system

5.1. Trajectory Generation

Quintic Polynomial Trajectories [12] are used as the reference trajectories for X , Y , Z , ψ , θ_1 , and θ_2 . Those types of trajectories have sinusoidal acceleration which is better in order to avoid vibrational modes.

The desired values of end effector position and orientation (Multi-region of operation and point-to-point control) are used to generate the desired trajectories for X , Y , Z , ϕ , θ and ψ using the inverse kinematics and then the algorithm for generating the trajectories.

6. FEEDBACK LINEARIZATION CONTROLLER

This section discusses the control system design based on the technique of feedback linearization [12, 16]. Feedback linearization transforms the nonlinear system dynamics into a linear system. Then the control laws are chosen so that zero tracking errors are achieved.

Fig. 7 presents the block diagram of this control technique. In this Figure, the nonholonomic constraints are used to determine the desired trajectories of θ and ϕ from the desired trajectories of X , Y , Z , ψ , θ_1 , and θ_2 and their derivatives. Then feedback linearization controllers are used to obtain a zero tracking errors for θ , ϕ , Z , ψ , θ_1 and θ_2 .

The nonholonomic constraints define the coupling between various states of the system. They are used to determine the desired trajectories of θ and ϕ . From the equations of the translation dynamics (46-48), one can extract the expressions of these high order nonholonomic constraints:

$$\sin(\phi) = \frac{(\ddot{X} - F_{m,q_x}^I)S_\psi - (\ddot{Y} - F_{m,q_y}^I)C_\psi}{\sqrt{(\ddot{X} - F_{m,q_x}^I)^2 + (\ddot{Y} - F_{m,q_y}^I)^2 + (\ddot{Z} + g - F_{m,q_z}^I)^2}} \quad (54)$$

$$\tan(\theta) = \frac{(\ddot{X} - F_{m,q_x}^I)C_\psi + (\ddot{Y} - F_{m,q_y}^I)S_\psi}{\ddot{Z} + g - F_{m,q_z}^I} \quad (55)$$

where F_{m,q_x}^I , F_{m,q_y}^I , and F_{m,q_z}^I are functions of the system states and their derivatives.

Putting subscript d to all variables in (54) and (55), then ϕ_d and θ_d can be obtained numerically.

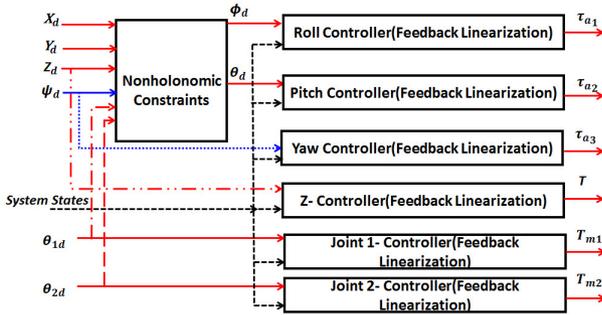


Fig. 7. Details of the controller block in case of FL-PID

Z-Controller can be developed by expressing the equation of motion in Z-direction in the following form:

$$(m\ddot{Z} + mg - F_{m,q_z}^I)/(C_\phi C_\theta) = T \quad (56)$$

The following control input will cancel out the nonlinearities in the system;

$$T = (mu_z + mg - F_{m,q_z}^I)/(C_\phi C_\theta) \quad (57)$$

where,

$$u_z = \ddot{Z}_d + K_{p_z}e_z + K_{d_z}\dot{e}_z + K_{i_z}\int_0^t e_z dt \quad (58)$$

This control law leads to the exponential stable dynamics

$$\ddot{e}_z + K_{d_z}\dot{e}_z + K_{p_z}e_z + K_{i_z}\int_0^t e_z dt = 0 \quad (59)$$

which implies that the error, $e_z \rightarrow 0$. K_p , K_d and K_i are the controller parameters.

For ϕ , θ , ψ , θ_1 and θ_2 controllers, similar control laws are chosen.

The system equations of motion and the control laws are simulated using MATLAB/SIMULINK program. The controller parameters of the feedback linearization controller are given in Table 2. Those parameters are tuned to get the required system performance. The controller are tested to stabilize and track the desired trajectories under the effect of picking a payload of value 150 g at instant 15 s and placing it at instant 65 s.

Table 2. FL-PID parameters

Par.	Value	Par.	Value
$[K_{p_z}K_{d_z}K_{i_z}]$	[16, 8, 0.01]	$[K_{p_\psi}K_{d_\psi}K_{i_\psi}]$	[16, 8, 0.01]
$[K_{p_\phi}K_{d_\phi}K_{i_\phi}]$	[100, 8, 10]	$[K_{p_{\theta 1}}K_{d_{\theta 1}}K_{i_{\theta 1}}]$	[16, 8, 0.01]
$[K_{p_\theta}K_{d_\theta}K_{i_\theta}]$	[100, 8, 10]	$[K_{p_{\theta 2}}K_{d_{\theta 2}}K_{i_{\theta 2}}]$	[16, 8, 0.01]

The simulation results are presented in Fig. 10. These results show that the controller design based on feedback linearization can track the desired trajectories before picking the payload, but at the instant of picking and then holding the payload, it fails to track the desired trajectories and the system becomes unstable even if the payload is released. These results show that FL-PID provides a good trajectory tracking capabilities but it fails to make system stable against adding the payload. In addition, due to the high nonlinearities and the complex dynamics in the system, the control laws are very complex and difficult to be implemented onboard (implementation in real time). Therefore, there is a need for an adaptive control technique to overcome the mentioned problems with lower complexity.

7. DIRECT FUZZY LOGIC CONTROL

Recently, fuzzy logic control [17–19] has become an alternative to conventional control algorithms to deal with complex processes and combine the advantages of classical controllers and human operator experience.

An intelligent controller, based on Direct Fuzzy Logic Control (DFLC), for a quadrotor was designed and

presented in [20, 21]. In this work, a modification of this technique is done and used to control the quadrotor-manipulator system to achieve the required objectives.

In Fig. 8, three fuzzy controllers are designed to control the quadrotor's roll (ϕ), pitch(θ) and yaw(ψ) angles, denoted by FLC_ϕ , FLC_θ , and FLC_ψ , respectively, with the former two serving as attitude stabilizers. Three fuzzy controllers, FLC_x , FLC_y and FLC_z , are designed to control the quadrotor's position. In addition, two fuzzy controllers FLC_{θ_1} and FLC_{θ_2} are designed to control the two joints' angles of the manipulator.

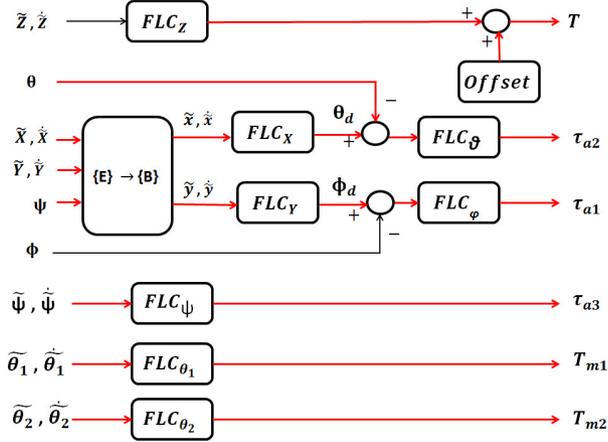


Fig. 8. Details of the controller block in case of DFCL

All eight fuzzy controllers have similar inputs that are: The error $e = (\cdot) = (\cdot)_d - (\cdot)$, which is the difference between the desired signal $(\cdot)_d$ and its measured value (\cdot) . This input is normalized to the interval $[-1, +1]$. The error rate c , which is normalized to the interval $[-3, +3]$.

In this control strategy, the desired pitch and roll angles, θ_d and ϕ_d , are not explicitly provided to the controller. Instead, they are continuously calculated by controllers FLC_x and FLC_y in such a way that they stabilize the quadrotor's attitude. First, one can convert the error and its rate of X and Y that is defined in the inertial frame into their corresponding values defined in the body frame. This conversion is done using the transformation matrix defined in (2) assuming small angles (ϕ and θ) as following:

$$\dot{\tilde{x}} = \dot{\tilde{X}} \cos \psi + \dot{\tilde{Y}} \sin(\psi) \quad (60)$$

$$\dot{\tilde{y}} = \dot{\tilde{X}} \sin(\psi) - \dot{\tilde{Y}} \cos \psi \quad (61)$$

The input and output membership functions of each FLC are tuned and chosen to be 3 symmetric and triangular. The input and output scaling factors for the error, change of error, and fuzzy output (K_{e_i} , K_{c_i} , and K_{u_i} ; $i = x, y, z, \phi, \theta, \psi, \theta_1, \theta_2$) of each FLC are tuned such that required performance is obtained.

The rule base of each FLC block is the same and is designed to provide a PD-like fuzzy controller. A Mamdani fuzzy inference method is used with a min-max operator for the aggregation and the center of gravity method for defuzzification.

There is a need to add an 'Offset' value to the control signal from the $FLC_z(T)$ in order to counter balance the weight of the quadrotor. This value has to be tuned.

It is important to note that the fuzzy controllers are designed in light of the knowledge acquired on the system's behavior and from its dynamic model. This property sets the fuzzy controllers apart from conventional controllers which depend on the plant's mathematical model [20].

Parameters of the DFCL are given in Table 3. Those parameters are tuned to get the required system performance.

The controller are tested to stabilize and track the desired trajectories under the effect of picking a payload of value 150 g at instant 15 s and placing it at instant 65 s.

The simulation results are presented in Fig. 10. These results show that DFCL is able to track the desired trajectories before, during picking, and holding the payload. However, the DFCL fails to track the desired trajectories during changing the region of operation (operating point) because it needs to retune its scaling factors. Thus, the DFCL succeeds to make system stable against adding the payload. However, it fails to provides a good trajectory tracking capabilities with different operation regions. In addition, it suffers from the necessity of calibrating and determining the offset value which is affected by payload value and cannot be estimated accurately. Moreover, considering the complexity of the controller implementation in real time, DFCL is fairly simple. Therefore, their is a need for high performance and more robust adaptive control technique to overcome these problems.

Table 3. DFCL parameters

Par.	Value	Par.	Value
$[K_{e_x} K_{c_x} K_{u_x}]$	[.007, .05, 5]	$[K_{e_y} K_{c_y} K_{u_y}]$	[.007, .05, 5]
$[K_{e_z} K_{c_z} K_{u_z}]$	[1, .3, 16.5]	$[K_{e_\psi} K_{c_\psi} K_{u_\psi}]$	[1, .5, 0.2]
$[K_{e_\phi} K_{c_\phi} K_{u_\phi}]$	[.5, .5, 9]	$[K_{e_{\theta_1}} K_{c_{\theta_1}} K_{u_{\theta_1}}]$	[2, .05, 4]
$[K_{e_\theta} K_{c_\theta} K_{u_\theta}]$	[.5, .5, 10]	$[K_{e_{\theta_2}} K_{c_{\theta_2}} K_{u_{\theta_2}}]$	[5, .3, 0.3]
<i>Offset</i>	7.85 N		

8. FUZZY MODEL REFERENCE LEARNING CONTROL

In this section, an adaptive fuzzy logic control based on "Fuzzy Model Reference Learning Controller" (FMRLC) is designed to control the proposed quadrotor manipulation system. This control technique is presented

in details in [17, 23–26].

The main drawback of fuzzy controllers is the large amount of parameters to be tuned. Also, the designed DFCL needs to retune its parameters in each operation region. Moreover, the fuzzy controller constructed for the nominal plant may perform inadequately if significant and unpredictable plant parameter variations, or environmental disturbances occur [22, 23].

In this work, a learning control algorithm is used to resolve some of these fuzzy controller design issues. This algorithm employs a reference model to provide a performance feedback for tuning a fuzzy controller's knowledge-base.

The control system design is the same as in Fig. 8 by replacing each of the FLC_z , FLC_ϕ , FLC_θ , FLC_ψ , FLC_{θ_1} and FLC_{θ_2} block with the block shown in Fig. 9. However, there is no need for the offset value that is used in Fig. 8 because the FMRLC can compensate the quadrotor weight. The blocks of FLC_x and FLC_y are still the same because there is no need for adaptation here, since these blocks are used to map the relation between the error in X and Y directions into the required roll and pitch motions.

The functional block diagram for the FMRLC is shown in Fig. 9.

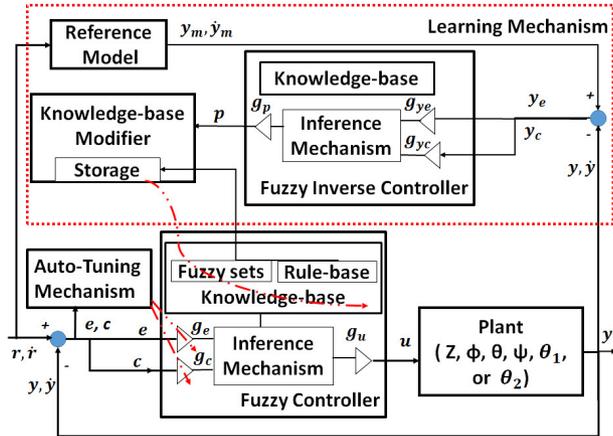


Fig. 9. Functional block diagram for the FMRLC

8.1. The Fuzzy Controller

The plant in Fig. 9 has output y (which can be Z , ϕ , θ , ψ , θ_1 or θ_2), and an input u (which can be T , τ_{a_1} , τ_{a_2} , τ_{a_3} , T_{m_1} or T_{m_2}). The scaling controller gains g_e , g_c , and g_u for the error, e , change in error, c , and controller output, u , are used respectively, such that the universe of discourse of all inputs and outputs are the same and equal to $[-1, 1]$.

The membership functions are chosen to be 11 symmetric triangular-shaped functions for each controller input. The fuzzy sets for the fuzzy controller output are assumed to be symmetric and triangular-shaped with a base width of 0.4, and all centered at zero on the normalized universe of discourse. They are what the

FMRLC will automatically tune through the learning mechanism. Thus, the initial rule base elements are set to zeros.

The centers of the input membership functions are tuned using the auto-tuning mechanism shown in Fig. 9. Mamdani fuzzy inference method is used with a min-max operator for the aggregation. The standard center of gravity is used as a defuzzification technique.

8.2. The Reference Model

The reference model is used to quantify the desired performance. A 1st order model is chosen as the reference model:

$$\frac{y_m(s)}{r(s)} = \frac{1}{\tau_{c_i}S + 1} \quad (62)$$

where $y_m(s)$ is the output response of the reference system, and $r(s)$ is the desired value of the plant. τ_{c_i} ($i = z, \phi, \theta, \psi, \theta_1, \text{ and } \theta_2$) is the time constant of the reference model. The performance of the overall system is controlled with respect to the reference model by the learning mechanism by generating an error signal:

$$y_e = y_m - y \quad (63)$$

8.3. The Learning Mechanism

The learning mechanism tunes the rule-base of the direct fuzzy controller so that the closed-loop system behaves like the reference model. Based on y_e , the learning mechanism will take the required action. It consists of two parts, fuzzy inverse model and knowledge-base modifier.

The Fuzzy Inverse Model

The fuzzy inverse model performs the function of mapping y_e to changes in the process inputs p to make y_e tends to zero.

The input to the fuzzy inverse model includes the error (y_e) and change in error (y_c). Also, it has scaling gains, g_{y_e} , g_{y_c} and g_p .

Each input and output is represented by 11 symmetric and triangular-shaped membership functions. The rule-base for the fuzzy controller has rules of the form:

$$\text{IF } y_e \text{ is } Y_e^k \text{ AND } y_c \text{ is } Y_c^s \text{ THEN } p \text{ is } P^m$$

where Y_e^k/Y_c^s denote the $k^{\text{th}}/s^{\text{th}}$ linguistic value associated with y_e/y_c and P^m denotes the consequent linguistic value associated with p .

Denoting the center of the output membership function for this rule $c_{k,s}$. The rule base array shown in Table 4 is used for the fuzzy inverse model. The entries of the table are the center values of output membership functions $c_{k,s}$. Mamdani fuzzy inference method is used with a min-max operator for the aggregation and the standard center of gravity is used as defuzzification technique.

Table 4. Knowledge-base array for the Fuzzy Inverse Model

$c_{k,s}$		Y_c^s										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
Y_e^k	-5	-1	-1	-1	-1	-1	-1	-0.8	-0.6	-0.4	-0.2	0
	-4	-1	-1	-1	-1	-1	-0.8	-0.6	-0.4	-0.2	0	0.2
	-3	-1	-1	-1	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4
	-2	-1	-1	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6
	-1	-1	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8
	0	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1
	1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1	1
	2	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1	1	1
	3	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1	1	1	1
	4	-0.2	0	0.2	0.4	0.6	0.8	1	1	1	1	1
5	0	0.2	0.4	0.6	0.8	1	1	1	1	1	1	

The Knowledge-Base Modifier

Based on the information about the necessary changes, p , the knowledge-base modifier changes the rule-base of the fuzzy controller so that the previously applied control action will be modified by the amount p .

Assume the previously computed control signal $u(t-1)$, and assume that it contributed to the present good or bad system performance. By modifying the fuzzy controller's knowledge-base, one may force the fuzzy controller to produce a desired output $u(t-1) + p(t)$, which should be put in at time $(t-1)$ to make y_e smaller. Then, in the next time, if similar values for the error and change in error are obtained, then the input to the plant will be one that will reduce the error between the reference model and plant output.

Knowledge-base modification is performed by shifting the centers b_m of the membership functions of the output that are associated with the fuzzy controller rules that contributed to the previous control action $u(t-1)$. It works as following:

- Define "active set" of rules at time $(t-1)$ to be all the rules in the fuzzy controller whose membership value is:

$$\mu_i(e(t-1), c(t-1)) > 0 \quad (64)$$

- For all rules in the active set, use (65) to modify the output membership function centers.

$$b_m(t) = b_m(t-1) + p(t) \quad (65)$$

Rules that are not in the active set do not have their output membership functions modified.

8.4. Auto-Tuning Mechanism

With the auto-tuning for the input scaling gains of the fuzzy controller, the centers of the input membership functions are tuned such that the control surface is

properly focused on the region that describes the system activity.

An auto-tuning mechanism is used in [24] to tune g_e and g_c gains online as following:

Let the maximum of each fuzzy controller inputs (e, c) over a time interval of the last T_a seconds be denoted by $\max_{T_a}\{e\}$ and $\max_{T_a}\{c\}$. Then this maximum value is defined as the gain of each input e and c so that,

$$g_e = \frac{1}{\max_{T_a}\{e\}} \quad \text{and} \quad g_c = \frac{1}{\max_{T_a}\{c\}} \quad (66)$$

Parameters of the FMRLC are given in Table 5. The controller are tested to stabilize and track the desired trajectories under the effect of picking a payload of value 150 g at instant 15 s and placing it at instant 65 s. The simulation results are presented in Fig. 10. These results show that FMRLC is able to track the desired trajectories (with different operating regions) before, during picking, holding, and placing the payload with zero tracking error.

Table 5. FMRLC Parameters

Par./Val.	Z	ϕ	θ	ψ	θ_1	θ_2
$g_{e-initial}$	0.2	2	2	0.3	1/60	1/60
$g_{c-initial}$	1/10	1	1	1/30	1/1000	1/1000
g_u	16.5	0.93	0.93	0.19	0.63	0.32
g_{y_e}	1/60	1/1	1/1	1/1	1/2	1/1.5
g_{y_c}	1/15	1/1	1/1	1/1	1/2	1/1.5
g_p	3	0.0029	0.0029	0.0019	0.0063	9.6e-4
$\tau_c(s)$	0.03	0.01	0.01	0.01	0.1	0.1
$T_a(s)$	0.1	0.05	0.05	0.05	0.1	0.1

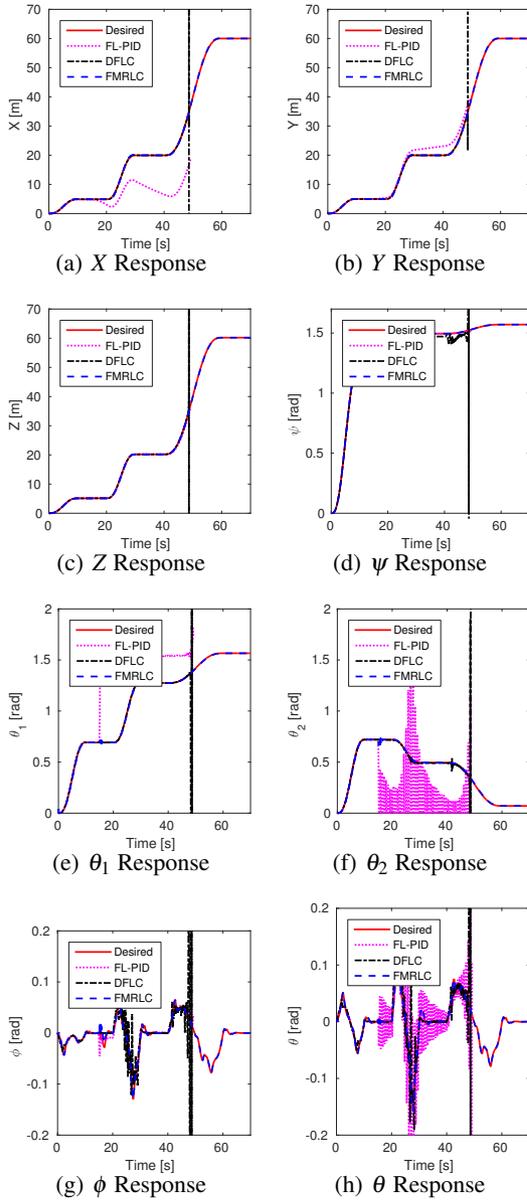


Fig. 10. Actual response of FL-PID, DFLL, and FMRLC techniques in the quadrotor/joint space

The end effector position and orientation can be found from the forward kinematics (see Fig. 11).

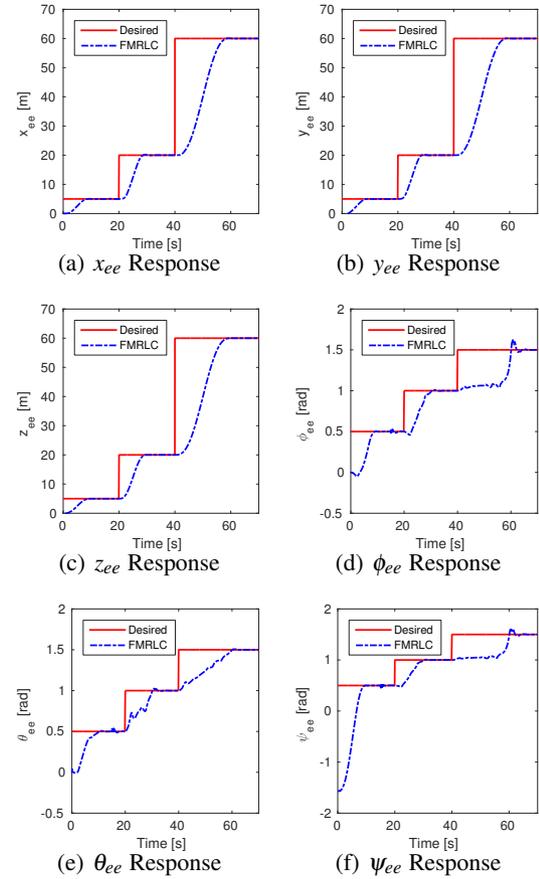


Fig. 11. Actual response of FMRLC technique in the task space

From the above discussion and results, the following items can be concluded about the performance of FMRLC technique:

- FMRLC technique succeeds to make system stable against adding/releasing the payload with high accuracy, in addition to, provides a good trajectory tracking capabilities with different operation regions.
- Considering the complexity of the controller implementation in real time, FMRLC is moderate. It is more complicated than DFLL and simpler than feedback linearization.
- Therefore, FMRLC technique is able to achieve the performance objective of the system.

9. CONCLUSIONS

Design, kinematics, dynamics and control of a novel aerial manipulation system, with unique topology, are presented. The new proposed system adds new features to the current aerial manipulation systems. New inverse kinematics is derived and utilized to achieve point-to-point task space control. The system controller was designed based on FL-PID, DFLL, and FMRLC techniques. The system equations of motion and the

control laws are simulated using MATLAB/SIMULINK program. The FL-PID fails to make system stable against adding the payload, and it suffers from complex real time implementation. The DFCLC technique succeeds to make system stable against adding/releasing the payload. However, it fails to provides a good trajectory tracking capabilities with different operation regions. The FMRLC succeeds to make system stable against adding/releasing the payload with high accuracy. In addition, it provides a good trajectory tracking capabilities with different operation regions. Simulations results show the effectiveness of the inverse kinematics analysis and indicate that the FMRLC has a superior performance compared with that of both FL-PID and DFCLC. As a future work, measurement system will be implemented to accurately measure the system states. Furthermore, the proposed robot and the motion control algorithms will be tested experimentally.

REFERENCES

- [1] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor unmanned aerial vehicles," in *Southeastcon, 2012 Proceedings of IEEE*, pp. 1–6, IEEE, 2012.
- [2] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2668–2673, IEEE, 2011.
- [3] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.
- [4] F. A. Goodarzi, D. Lee, *et al.*, "Geometric control of a quadrotor uav transporting a payload connected via flexible cable," *International Journal of Control, Automation and Systems*, vol. 13, no. 6, pp. 1486–1498, 2015.
- [5] C. M. Korpela, T. W. Danko, and P. Y. Oh, "Mm-uav: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 93–101, 2012.
- [6] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a uav with a robotic arm," in *10th International IFAC Symposium on Robot Control*, pp. 704–709, 2012.
- [7] J. Escareno, M. Rakotondrabe, G. Flores, and R. Lozano, "Rotorcraft mav having an onboard manipulator: Longitudinal modeling and robust control," in *Control Conference (ECC), 2013 European*, pp. 3258–3263, IEEE, 2013.
- [8] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 227–240, 2013.
- [9] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two dof robotic arm," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 4990–4995, IEEE, 2013.
- [10] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael, "Modeling and control of a new quadrotor manipulation system," in *2012 IEEE/RAS International Conference on Innovative Engineering Systems*, pp. 109–114, IEEE, 2012.
- [11] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael, "Adaptive intelligent controller design for a new quadrotor manipulation system," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pp. 1666–1671, IEEE, 2013.
- [12] J. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2000.
- [13] G. G. Slabaugh, "Computing euler angles from a rotation matrix," *Retrieved on August*, vol. 6, p. 2000, 1999.
- [14] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. Wiley-Interscience, 1999.
- [15] M. Elsamanty, A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael, "Methodology for identifying quadrotor parameters, attitude estimation and control," in *Advanced Intelligent Mechatronics (AIM), 2013 IEEE/ASME International Conference on*, pp. 1343–1348, IEEE, 2013.
- [16] J.-J. E. SLOTTINE and W. LI, *Applied Nonlinear Control*. Prentice-Hall, 1991.
- [17] K. M. Passino and S. Yurkovich, *Fuzzy control*. Citeseer, 1998.
- [18] Z. Kovačić and S. Bogdan, *Fuzzy controller design: theory and applications*, vol. 19. CRC Press, 2006.
- [19] Y. H. Choi and S. J. Yoo, "A simple fuzzy-approximation-based adaptive control of uncertain unmanned helicopters," *International Journal of Control, Automation and Systems*, vol. 14, no. 1, pp. 340–349, 2016.
- [20] S. A. Raza and W. Gueaieb, "Intelligent flight control of an autonomous quadrotor," *InTech*, 2010.
- [21] B. Erginer and E. Altuğ, "Design and implementation of a hybrid fuzzy logic controller for a quadrotor vtol vehicle," *International Journal of Control, Automation and Systems*, vol. 10, no. 1, pp. 61–70, 2012.
- [22] Yang, Jin and Su, Hang and Li, Zhijun and Ao, Di and Song, Rong, "Adaptive control with a fuzzy tuner for cable-based rehabilitation robot," *International Journal of Control, Automation and Systems*, vol. 14, no. 3, pp. 865–875, 2016.
- [23] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," in *1992 First IEEE Conference on Control Applications*, pp. 686–691, IEEE, 1992.
- [24] W. A. Kwong and K. M. Passino, "Dynamically focused fuzzy learning control," *Part B: Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 1, pp. 53–74, 1996.
- [25] S. Blažič, I. Škrjanc, and D. Matko, "Globally stable direct fuzzy model reference adaptive control," *Fuzzy Sets and Systems*, vol. 139, no. 1, pp. 3–33, 2003.
- [26] P. LI and F.-J. JIN, "Adaptive fuzzy control for unknown nonlinear systems with perturbed dead-zone inputs," *Acta Automatica Sinica*, vol. 36, no. 4, pp. 573–579, 2010.



Ahmed Khalifa received the B.S. degree in Industrial Electronics and Control Engineering from Monofiya University, Egypt, in 2009, and the M.Sc. degree in Mechatronics and Robotics Engineering from Egypt-Japan University of Science and Technology (EJUST), Egypt, in 2013. He is currently pursuing the Ph.D. degree in Mechatronics and Robotics Engineering

at EJUST.



Mohamed Fanni received the B.E. and M.Sc. degrees in mechanical engineering from Faculty of Engineering of both Cairo University and Mansoura University, Egypt, in 1981 and 1986, respectively and the Ph.D. degree in engineering from Karlsruhe University, Germany, 1993. He is an Associate Professor with Innovation, Graduate School of Engineering Science,

EJUST, Alexandria, on leave from Production Engineering & Mechanical Design Department, Faculty of Engineering, Mansoura University, Egypt. His major research interests include robotics engineering, automatic control, and Mechanical Design. His current research focuses on Design & Control of Mechatronic Systems, Surgical Manipulators, Teleoperation systems and Flying/Walking Robots.