End-Effector Position Tracking Control for an Aerial Manipulator with a Lightweight Manipulator

Huazi Cao (caohuazi@westlake.edu.cn)
Westlake University
Lixin Wang
Beihang University

Research Article

Keywords: aerial manipulator, adaptive NN control, RBFNN, position tracking control, lightweight manipulator

Posted Date: August 23rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-804614/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
End-effector position tracking control for an aerial manipulator with a lightweight manipulator

Huazi Cao · Lixin Wang

Received: date / Accepted: date

Abstract An end-effector position tracking control task for an aerial manipulator is usually constituted by two subtasks. The first is motion control, and the second is coordinate control so that the end-effector of the aerial manipulator can precisely track the given trajectory. This paper proposes a novel end-effector position tracking control approach for the aerial manipulator with a lightweight manipulator to achieve these two subtasks. The motion control of the aerial manipulator is solved by a partially coupled approach and divided into a multirotor controller and a manipulator controller. The multirotor controller is designed by the adaptive neural network control, while joints of the manipulator are steered by PID controllers. By resorting to radial basis function neural networks with adaptive weight estimation laws, the dynamic coupling between the multirotor and the manipulator can be compensated in real time. With the support of Lyapunov stability criteria, it is proved that the desired trajectories can be boundedly tracked by the multirotor under the proposed controller. Then, a new coordinate control method is proposed based on the linear model predictive control method. This method ensures that the solution satisfies physical limits of the aerial manipulator and can be executed in real time. Simulations demonstrate that the proposed motion controller significantly outperforms a baseline nonlinear motion controller in the simulation cases. Besides, comparisons among the proposed coordinate control method and traditional methods are simulated to demonstrate effectiveness and performance.

Keywords aerial manipulator · adaptive NN control · RBFNN · position tracking control · lightweight manipulator

1 Introduction
Aerial manipulator integrates capabilities of the multirotor and the manipulator, which can extend the scope of potential applications. Since the payload of the multirotor is limited, the aerial manipulator with a lightweight manipulator has received much attention in recent years. The lightweight manipulator generally equips light servos (e.g., Dynamixel servos [1], Herkulex DRS-0101 [2] and DRS-0602 [3]). Limitations of these servos affect the implementation of control methods in terms of control capabilities, available feedback signals and update rate. In the aerial manipulator, the end-effector is the end part of the manipulator that interacts with the environment. The role of the end-effector position tracking control is significant. Totally, it consists of two subtasks. The first is motion control, which is to steer the multirotor and the manipulator from the current positions to the desired positions under the dynamic coupling. The second is coordinate control, which is to coordinate motions of the multirotor and the manipulator such that the end-effector can precisely track the desired trajectory.

These existing methods to design the motion controller of the aerial manipulator can be classified into decoupled, coupled and partially coupled methods (see [4] and [5] for recent surveys). Decoupled methods treat the multirotor and the manipulator as two subsystems
that are independently steered by utilizing the existing control algorithms \cite{6,8}. Decoupled methods ignore the influence of the manipulator over the dynamics of the multirotor, which motivates the development of lightweight manipulators \cite{9}. However, as soon as the motion of the manipulator is more demanding on accelerations, decoupled methods may achieve large tracking errors. Coupled methods treat the multirotor and the manipulator as a unique entity \cite{10}. The design of a coupled controller relies on the full dynamic model of the aerial manipulator. Therefore, coupled methods perform better in compensating dynamic couplings and allow better performances in terms of accuracy and stability. Based on this viewpoint, the backstepping control \cite{11}, adaptive control \cite{1}, the multi-stage model predictive control \cite{12} have been studied to develop the coupled motion controller for the aerial manipulator. Coupled methods become complicated when the number of joints increases. These methods require torque controlled motors that are unreliable for aerial manipulators usually built with conventional servo actuators \cite{8}. Besides, update rates of light servo actuators are usually lower than 100 Hz \cite{13}, which is another barrier to implementing coupled methods. Partially coupled methods treat the multirotor and the manipulator as two subsystems, but the controllers exploit the information of the multirotor and the manipulator to compensate the dynamic coupling \cite{14}. The work in \cite{8} designed a sliding battery box to compensate the dynamic coupling based on the estimate results. The performance of this method is limited by the sliding velocity of the battery box. Besides, this method also needs extra mechanical structure. Motivated by these limitations, researchers have proposed some approaches to compensate the dynamic coupling by the capability of the multirotor. For example, a multi-layer architecture has been proposed in \cite{8} and a variable parameter integral backstepping controller has been designed in \cite{15}. Works in \cite{16} and \cite{17} introduced disturbance observers into the motion controller to compensate the dynamic couplings. However, these observers rely on the mathematical model of the dynamic coupling.

The coordinate control of the aerial manipulator is usually treated as an inverse kinematics problem. For example, the works in \cite{11} and \cite{18} introduced the closed-loop inverse kinematics (CLIK) method into the coordinate control of the aerial manipulator. To improve the performance of the coordinate control, the null space-based method and compound behaviors are used in solving the coordinate control problem \cite{19}. Besides, a multiple task-priority inverse kinematics algorithm is introduced to solve the coordinate controller problem for a dual-arm aerial manipulator \cite{20}. However, these methods solve the coordinate control problem with respect to the current state, and they are not able to consider the future state. The work in \cite{21} introduced a nonlinear model predictive control (NMPC) method and treated the coordinate control problem as a constraint optimization problem. With the predictive model, the solution of the NMPC method can solve the optimization problem with respect to the current information and the prediction of the future. Due to the high computational cost of the NMPC method, it takes quite some time to calculate the solution, which might be a barrier to real-time applications.

To overcome the difficulties of the motion control, we propose a novel partially coupled motion controller based on the adaptive neural network (NN) control method that is widely used in solving uncertain nonlinear system control problems \cite{22,23}. The adaptive radial basis function neural networks (RBFNNs) improve the accuracy of estimation with online training \cite{24}. The proposed method compensates the dynamic coupling in real time and does not rely on the accuracy model of the dynamic coupling. To overcome the difficulties of the coordinate control, we propose a novel coordinate controller based on the linear model predictive control (MPC) method. The physical limitations of the aerial manipulator is encoded as constraints of a quadratic program (QP) problem, which ensures that the solution satisfies these limitations. Besides, this method can solve the coordinate problem in real time with the current information and the prediction of the future.

As stated above, the proposed end-effector position tracking controller has a two-layer architecture. The inner layer is the proposed motion controller and the outer layer is the proposed coordinate controller. The main contributions of this paper are fourfold. First, a partially coupled motion controller based on the aerial control method is proposed to solve the motion control of the aerial manipulator. To the best of our knowledge, this is the first time that the adaptive NN control method has been applied on the motion control of the aerial manipulator. Second, dynamic couplings of the aerial manipulator are estimated by adaptive RBFNNs via the proposed adaptive weight estimation law. The weights of the adaptive RBFNNs can be trained online. Third, stability analysis proves that tracking errors of the closed-loop multirotor system converge to a small neighborhood of the origin with assuming that the dynamic couplings are continuous. Fourth, we address the coordinate control of the aerial manipulator as a QP problem. And, a linear MPC method is applied to calculate the solution with respect to the given task. Compared with the NMPC method, the linear MPC needs.
End-effector position tracking control for an aerial manipulator with a lightweight manipulator

less computing resources. It can be solved efficiently by existing QP solvers (e.g., CVXGEN [25]), which means it can be executed in some real-time applications. Compared with the CLIK methods, the proposed coordinate method shows better effectiveness and performance. To the best of our knowledge, this is the first time that the linear MPC method has been applied on the coordinate control of the aerial manipulator.

The remainder of this paper is structured as follows. Plant description and preliminaries are given in Section 2. Section 3 proposes the adaptive NN control method to design the multirotor controller. The linear MPC method has been applied on the coordinate control problem of the aerial manipulator in Section 4. Section 5 shows the proposed control structure for the end-effector position tracking control. Then, the simulation results for the proposed method are given in Section 6. Conclusions are drawn in Section 7.

2 Plant description and preliminaries

This section presents a mathematical model of the aerial manipulator and some preliminary results that will be used throughout this paper.

2.1 Kinematics

The aerial manipulator studied in this paper consists of a multirotor and a manipulator (see Fig. 1). The manipulator consists of a number of rigid links connected with joints. The end-effector is the last link of the manipulator.

![Configuration of the coordinates for the aerial manipulator.](image)

Fig. 1: Configuration of the coordinates for the aerial manipulator.

Let $\Sigma_I$ denote the inertial frame. $p_i \in \mathbb{R}^3$ represents the position of the $i$-th link’s mass center in $\Sigma_I$. $p = [x, y, z]^T \in \mathbb{R}^3$ denotes the position of the multirotor in $\Sigma_I$. Let $\Sigma_b$ denote the body frame of the multirotor. Let $p_b^i \in \mathbb{R}^3$ represents the position of the $i$-th link’s mass center in the body frame $\Sigma_b$. Then, we have

$$p_i = p + Rp_b^i,$$  

where $R \in SO(3)$ is the rotation matrix mapping vectors expressed in $\Sigma_b$ into the vectors expressed in $\Sigma_I$. $p_b^i$ can be obtained by Denavit-Hartenberg convention [26].

Let $p_e^i \in \mathbb{R}^3$ denote the position of the end-effector in $\Sigma_b$. $q = [q_1, q_2, \cdots, q_n]^T \in \mathbb{R}^n$ denotes the joint vector of the manipulator. Therefore, the position of the end-effector expressed in inertial frame $\Sigma_I$ is given as

$$p_e = p + R p_b^e,$$  

where $p_e \in \mathbb{R}^3$ denote the position of the end-effector in $\Sigma_I$. $p_b^e$ is a function of the manipulator’s joint position vector $q$. Deriving (2) with respect to time yields

$$\dot{p}_e = \dot{p} - (R p_b^e) \times \omega + R P_b^e \dot{q},$$  

where $[\cdot]_\times$ denotes the skew-symmetric matrix, $\omega \in \mathbb{R}^3$ is the angular velocity vector of the multirotor in $\Sigma_b$, $P_b^e \in \mathbb{R}^{3 \times n}$ is the partial derivative matrix of $p_b^e$ with respect to $q$.

2.2 Dynamics

Let $\Theta = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ denote Euler angle vector of the multirotor. The dynamic model of the aerial manipulator is derived by the Euler–Lagrange method [26]. It can be written as follows [27]:

$$B(\xi) \ddot{\xi} + C(\xi, \dot{\xi}) + g(\xi) = f_s,$$  

where $\xi = [p^T, \Theta^T, q^T]^T$, $B(\xi) \in \mathbb{R}^{n \times n}$ is the symmetric and positive definite inertia matrix, $C(\xi, \dot{\xi}) \in \mathbb{R}^{n \times n}$ is the Coriolis matrix, $g(\xi) \in \mathbb{R}^n$ is the gravity term, $f_s \in \mathbb{R}^n$ is the generalized force vector.

Dynamic couplings are determined by the motions of the multirotor and the manipulator [4]. However, servo actuators equipped by the lightweight manipulator generally can not provide enough signals to calculate the dynamic couplings [13]. From the practical point of view, the dynamic model [1] is divided into the multirotor and the manipulator models. The multirotor model with dynamic couplings can be written as follows [28]:

$$\begin{cases} \ddot{p} = v, \\
\ddot{p} = \frac{m_b}{m_b} \dot{m}_b g - \frac{f}{m_b} + \Delta v
\end{cases}$$  

(5)
\[
\begin{align*}
\dot{\Theta} &= Q^{-1}\omega, \\
\dot{\omega} &= -I_b^{-1}\omega \times I_b\omega + I_b^{-1}\tau + I_b^{-1}g_s(\xi) + \Delta\omega,
\end{align*}
\]  

(6)

where \(m_b\) represents the mass of the multirotor, \(m_1\) represents the mass of the \(i\)-th link, \(g \in \mathbb{R}^3\) is the acceleration vector of gravity, \(f \in \mathbb{R}^3\) is the control input vector of the position loop, \(g_s(\xi) \in \mathbb{R}^3\) denotes the torque induced by the displacement of the manipulator on the multirotor, \(\Delta\omega\) represents the mass of the multirotor, \(\Delta\omega\) represents the torque induced by the displacement of the manipulator on the multirotor, \(\Delta_s\) and \(\Delta_w\) respectively represent the weight vector, and the torque induced by the velocity and the acceleration of the manipulator,

\[
Q = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi \cos \theta & \cos \phi \end{bmatrix}.
\]  

(7)

The structure of the RBFNN is given in Fig. 2. Gaussian functions are chosen to design basis functions, which can be written as

\[
s_i(\eta) = \exp\left(-\frac{(\eta - a_i)^T(\eta - a_i)}{b_i^2}\right), \quad i = 1, 2, \ldots, N.
\]

2.3 Radial basis function neural network

The RBFNN is usually used for modeling nonlinear systems since it can approximate any continuous function over a compact set.

Lemma 1 \((29)\) For any continuous function \(\Phi(\eta)\): \(S \rightarrow \mathbb{R}\), there exist \(\sigma, \eta > 0\) and a RBFNN \(W^T s(\eta)\) such that

\[
\Phi(\eta) = w^* T s(\eta) + \epsilon(\eta), \quad |\epsilon| \leq \sigma, 
\]  

(10)

where \(\eta \in S \subset \mathbb{R}^n\) represents the neural network input vector, \(S\) represents a compact set, \(\mathbf{w} = [w_1, w_2, \ldots, w_N] \in \mathbb{R}^N\) represent the weight vector, \(w^*\) is the ideal weight vector, \(s = [s_1(\eta), s_2(\eta), \ldots, s_N(\eta)] \in \mathbb{R}^N\) is the basis function vector of the RBFNN, \(N\) represents the node number of the neural network.

The structure of the RBFNN is given in Fig. 2. Gaussian functions are chosen to design basis functions, which can be written as

\[
s_i(\eta) = \exp\left(-\frac{(\eta - a_i)^T(\eta - a_i)}{b_i^2}\right), \quad i = 1, 2, \ldots, N.
\]

3 Adaptive NN controller design

In this section, the adaptive NN control method is introduced in designing the position control and the attitude control of the multirotor to compensate dynamic couplings of the aerial manipulator.

3.1 Position control

Let \(p^d \in \mathbb{R}^3\) denote the desired position. The position error of the multirotor is then defined as \(\tilde{p} = p - p^d\). To depict the desired velocity of the multirotor, we define \(v_r = p^d - K_p \tilde{p}\),

(11)

where \(K_p \in \mathbb{R}^{3 \times 3}\) is a positive diagonal matrix. Then, the velocity error of the multirotor is defined as \(\tilde{v} = \tilde{p} - v_r\).

Let \(\eta_p = [\Theta^T, \omega^T, q^T]^T\) denote the input vector of the position RBFNNs. Let \(W_p \in \mathbb{R}^{N \times 3}\) and \(S_p(\eta_p) \in \mathbb{R}^{N \times 3}\) respectively represent the weight matrix and the basis function matrix of the position RBFNNs. The position controller of the multirotor is defined as

\[
f = m_b \left(\frac{m_b + \sum_{i=1}^{n} m_i}{m_b}\right) g - \dot{\tilde{v}} + W_p^T S_p(\eta_p) \\
+ K_v \tilde{v} + \tilde{p},
\]  

(14)

where \(K_v \in \mathbb{R}^{3 \times 3}\) is a positive diagonal matrix. The adaptive weight estimation law of the position RBFNNs is given as

\[
\dot{w}_{p, i} = \Gamma_p (s_{p,i}(\eta_p)) \tilde{v}_i - \mu_p w_{p,i}, \quad i = 1, 2, 3,
\]  

(15)
where \( \mathbf{w}_{p,i} \in \mathbb{R}^N \) is the \( i \)-th column vector of \( \mathbf{W}_p \), \( s_{p,i}(\eta_p) \in \mathbb{R}^N \) is the \( i \)-th column vector of \( \mathbf{S}_{p}(\eta_p) \), \( \dot{\mathbf{v}} \) is the \( i \)-th element of \( \dot{\mathbf{v}} \), \( \Gamma_p \in \mathbb{R}^{N \times N} \) is a positive diagonal matrix, \( \mu_p \in (0, 1) \) represents the learning rate.

**Theorem 1** Assuming that \( \Delta_v \) is continuous, if the position controller is designed as \( (14) \) with the adaptive weight estimation law \( (15) \), then the position error \( \ddot{\mathbf{p}} \) is bounded.

*Proof* Let \( \ddot{\mathbf{w}}_{p,i} = \mathbf{w}_{p,i} - \mathbf{w}_{p,i}^{*} \) denote the error of the RBFNN weight vectors. A candidate Lyapunov functions is given as

\[
V_p(r_p) = \frac{1}{2} \ddot{\mathbf{p}}^T \ddot{\mathbf{p}} + \frac{1}{2} \ddot{\mathbf{v}}^T \ddot{\mathbf{v}} + \frac{3}{2} \sum_{i=1}^{3} \ddot{\mathbf{w}}_{p,i}^T \Gamma_p^{-1} \ddot{\mathbf{w}}_{p,i},
\]

where \( r_p = [\ddot{p}^T, \ddot{v}^T, \ddot{w}_{p,1}^T, \ddot{w}_{p,2}^T, \ddot{w}_{p,3}^T]^T \). Let \( \lambda_{p,m} \) and \( \lambda_{p,M} \) denote the minimum and the maximum eigenvalues of \( \Gamma_p^{-1} \). Then, we have

\[
\frac{k_{p,m}}{2} \| r_p \|^2 \leq V_p(r_p) \leq \frac{k_{p,M}}{2} \| r_p \|^2,
\]

where \( k_{p,m} = \min(1, \lambda_{p,m}) \) and \( k_{p,M} = \max(1, \lambda_{p,M}) \). Combining \( (3) \), \( (13) \) and \( (14) \), the time derivative of \( V_p \) is given as

\[
\dot{V}_p = \ddot{p}^T \ddot{p} + \ddot{v}^T \ddot{v} + \sum_{i=1}^{3} \ddot{w}_{p,i}^T \Gamma_p^{-1} \ddot{w}_{p,i} - \ddot{p}^T \mathbf{K}_p \ddot{p} - \ddot{v}^T \mathbf{K}_v \ddot{v} - \mu_p \sum_{i=1}^{3} \ddot{w}_{p,i}^T \mathbf{w}_{p,i}
\]

Substituting \( (10) \) and \( (15) \) into \( (18) \) yields

\[
\dot{V}_p = -\ddot{p}^T \mathbf{K}_p \ddot{p} - \ddot{v}^T \mathbf{K}_v \ddot{v} + \ddot{v}^T (\Delta_v - \mathbf{W}_p \mathbf{S}_p(\eta_p) - \ddot{\mathbf{p}} - \ddot{\mathbf{K}}_v \ddot{\mathbf{v}})
\]

Using Young’s inequality, we have

\[
\ddot{v}^T \mathbf{e}_p \leq \frac{1}{2} \| \ddot{v} \|^2 + \frac{1}{2} \| \mathbf{e}_p \|^2.
\]

Substituting \( (20) \) into \( (19) \), we can obtain

\[
\dot{V}_p \leq -\ddot{p}^T \mathbf{K}_p \ddot{p} - \ddot{v}^T \mathbf{K}_v \ddot{v} - \frac{\mu_p}{2} \sum_{i=1}^{3} \ddot{w}_{p,i}^T \ddot{w}_{p,i}
\]

where \( k_1 = \min(\lambda_{p,m}, \lambda_{K_v,m} - 0.5, \mu_p/2) \), \( \lambda_{K_v,m} \) is the minimum eigenvalue of \( \mathbf{K}_v \), \( \lambda_{K_v,m} \) is the minimum eigenvalue of \( \mathbf{K}_v \). According to \( (17) \), \( (22) \) is rewritten as

\[
\dot{V}_p \leq -\frac{k_1}{k_{p,M}} V_p + \sigma_p,
\]

where \( \sigma_p = \frac{1}{2} \| \mathbf{e}_p \|^2 + \frac{\mu_p}{2} \sum_{i=1}^{3} \mathbf{w}_{p,i}^T \mathbf{w}_{p,i} \). Because \( \mathbf{e}_p \) is bounded, we can obtain \( \sigma_p \leq \sigma_{p,M} \). Using the comparison method \( (30) \), we can obtain

\[
V_p(t) \leq \exp(-\frac{k_1}{k_{p,M}} t) \left( V_p(0) - \frac{\sigma_{p,M} k_{p,M}}{2 k_1} \right) + \frac{\sigma_{p,M} k_{p,M}}{2 k_1}.
\]

Then, we can obtain

\[
\lim_{t \to \infty} \| r_p \| \leq \frac{2}{k_{p,M}} \sqrt{\frac{\sigma_{p,M} k_{p,M}}{2 k_1}}.
\]

According to the definition of \( r_p \), it can be obtained that

\[
\lim_{t \to \infty} \| \mathbf{p} \| \leq \frac{2}{k_{p,M}} \sqrt{\frac{\sigma_{p,M} k_{p,M}}{2 k_1}},
\]

which means that the position error is bounded with the dynamic couplings of the aerial manipulator.

Consequently, the desired total thrust \( f \) and desired attitude angles can be computed as

\[
f = \| \mathbf{f} \|.
\]

(27)

\[
\begin{align*}
\phi_i &= \sin^{-1} \left( \frac{f_1 \sin \psi_i - f_2 \cos \psi_i}{f} \right), \\
\theta_i &= \tan^{-1} \left( \frac{f_1 \cos \psi_i + f_2 \sin \psi_i}{f_3} \right),
\end{align*}
\]

(28)

where \( f_i \) is the \( i \)-th element of \( \mathbf{f} \).
3.2 Attitude control

Let $\mathbf{\Theta}^d \in \mathbb{R}^3$ denote the desired attitude of the multirotor. Let $\mathbf{\Theta} = \mathbf{\Theta}^d - \mathbf{\Theta} \in \mathbb{R}^3$ denote the attitude error of the multirotor. To depict the desired angular velocity of the multirotor, we define

$$\omega_r = Q(\mathbf{\Theta}^d - K_\Theta \mathbf{\Theta}),$$

where $K_\Theta \in \mathbb{R}^{3 \times 3}$ is a positive diagonal matrix. Then, the angular velocity error of the multirotor is defined as $\hat{\omega} = \omega - \omega_r$.

Let $\eta_a = [\Theta^T, \omega^T, q]^T$ denote the input vector of the attitude RBFNNs. Let $W_a \in \mathbb{R}^{N \times 3}$ and $S_a(\eta_a) \in \mathbb{R}^{N \times 3}$ respectively represent the weight matrix and the basis function matrix of the attitude RBFNNs. The attitude controller of the multirotor is given as

$$\tau = I_3 (\hat{\omega} - \hat{\mathbf{K}} \hat{\omega} - Q^{-T} \hat{\mathbf{\Theta}} - W_a^T S_a(\eta_a)) + \omega \times I_3 \hat{\omega} - g_s,$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ is a positive diagonal matrix. The adaptive weight estimation law of the attitude RBFNNs is given as

$$\dot{w}_{a,i} = \Gamma_a (s_{a,i}(\eta_a) \hat{\omega}_i - \mu_a w_{a,i}), \quad i = 1, 2, 3,$$

where $w_{a,i} \in \mathbb{R}^N$ is the $i$-th column vector of $W_a$, $s_{a,i}(\eta_a) \in \mathbb{R}^N$ is the $i$-th column vector of $S_a(\eta_a)$, $\hat{\omega}_i$ is the $i$-th element of $\hat{\omega}$, $\Gamma_a \in \mathbb{R}^{N \times N}$ is a positive diagonal matrix, $\mu_a \in (0, 1)$ represents the learning rate.

**Theorem 2** Assuming that $\Delta_a$ is continuous, if the position controller is designed as \[ \text{(30)} \] with the adaptive weight estimation law \[ \text{(31)} \], then the attitude error $\hat{\mathbf{\Theta}}$ is bounded.

**Proof** Let $\dot{w}_{a,i} = w_{a,i} - w_{a,i}^*$ denote the error of the attitude RBFNNs weight vectors. A candidate function is given as

$$V_a(r_a) = \frac{1}{2} \hat{\mathbf{\Theta}}^T \hat{\mathbf{\Theta}} + \frac{1}{2} \hat{\omega}^T \hat{\omega} + \frac{3}{2} \sum_{i=1}^{3} \dot{w}_{a,i} \Gamma_a^{-1} \dot{w}_{a,i},$$

where $r_a = [\hat{\mathbf{\Theta}}^T, \hat{\omega}^T, \hat{\mathbf{w}}_{a,1}^T, \hat{\mathbf{w}}_{a,2}^T, \hat{\mathbf{w}}_{a,3}^T]^T$. Let $\lambda_{a,m}$ and $\lambda_{a,M}$ respectively denote the minimum and the maximum eigenvalues of $\Gamma_a^{-1}$. Then, we have

$$\frac{1}{2} k_{a,m} \| r_a \|^2 \leq V_a(r_a) \leq \frac{1}{2} k_{a,M} \| r_a \|^2,$$

where $k_{a,m} = \min(1, \lambda_{a,m})$ and $k_{a,M} = \max(1, \lambda_{a,M})$. Combining \[ \text{(30)} \], \[ \text{(29)} \] and \[ \text{(30)} \], the time derivative of $V_a$ is given as

$$\dot{V}_a = - \hat{\mathbf{\Theta}}^T K_\Theta \hat{\mathbf{\Theta}} - \hat{\omega}^T K_\omega \hat{\omega}$$

$$+ \hat{\omega} (\Delta_a - W_a) S_a(\eta_a) + \sum_{i=1}^{3} \dot{w}_{a,i} \Gamma_a^{-1} \dot{w}_{a,i},$$

Substituting \[ \text{(10)} \] and \[ \text{(31)} \] into \[ \text{(34)} \] yields

$$\dot{V}_a = - \hat{\mathbf{\Theta}}^T K_\Theta \hat{\mathbf{\Theta}} - \hat{\omega}^T K_\omega \hat{\omega}$$

$$+ \hat{\omega}^T (\dot{W}_a S_a(\eta_a) + \epsilon_a) + \sum_{i=1}^{3} \dot{w}_{a,i} \Gamma_a^{-1} \dot{w}_{a,i}$$

$$= - \hat{\mathbf{\Theta}}^T K_\Theta \hat{\mathbf{\Theta}} - \hat{\omega}^T K_\omega \hat{\omega} + \hat{\omega}^T \epsilon_a,$$

By the completion of squares, we have

$$- \dot{w}_{a,i} w_{a,i} \leq - \frac{\dot{w}_{a,i}^T \dot{w}_{a,i}}{2} + \frac{\epsilon_a^T \epsilon_a}{2}.$$

Using Young’s inequality, we have

$$\hat{\omega}^T \epsilon_a \leq \frac{1}{2} \| \epsilon_a \|^2 + \frac{1}{2} \| \dot{r}_a \|^2.$$

Substituting \[ \text{(29)} \] into \[ \text{(35)} \], we can obtain

$$\dot{V}_a \leq - \hat{\mathbf{\Theta}}^T K_\Theta \hat{\mathbf{\Theta}} - \hat{\omega}^T K_\omega \hat{\omega} - \mu_a \sum_{i=1}^{3} \dot{w}_{a,i}^T w_{a,i}^*$$

$$+ \hat{\omega}^T \epsilon_a + \frac{\mu_a}{2} \sum_{i=1}^{3} \dot{w}_{a,i}^T w_{a,i}^*,$$

$$\leq - k_2 \| r_a \|^2 + \frac{1}{2} \| \epsilon_a \|^2 + \frac{\mu_a}{2} \sum_{i=1}^{3} \dot{w}_{a,i}^T w_{a,i}^*,$$

where $k_2 = \min(\lambda_{K_{a,m}}, \lambda_{K_{a,m}} - 0.5, \mu_a/2)$. $\lambda_{K_{a,m}}$ is the minimum eigenvalue of $K_\Theta$, $\lambda_{K_{a,m}}$ is the minimum eigenvalue of $K_\omega$. According to \[ \text{(33)} \], \[ \text{(38)} \] is rewritten as

$$\dot{V}_a \leq \frac{k_2}{k_{a,M}} V_a + \sigma_a,$$

where $\sigma_a = \frac{1}{2} \| \epsilon_a \|^2 + \frac{\mu_a}{2} \sum_{i=1}^{3} \dot{w}_{a,i}^T w_{a,i}^*$. Because $\epsilon_a$ is bounded, we can obtain $\sigma_a \leq \sigma_{a,M}$. Using the comparison method \[ \text{(30)} \], we can obtain

$$V_a(t) \leq \exp(-2 \frac{k_{a,M}}{k_{a,M}} t) \left( V_a(0) - \frac{\sigma_{a,M}}{2k_1} k_{a,M} \right) + \frac{\sigma_{a,M}}{2k_1} k_{a,M}.$$  

(40)

Then, we can obtain

$$\lim_{t \to \infty} \| r_a \| \leq \frac{2}{k_{a,M} k_{a,M}} \sqrt{\frac{\sigma_{a,M} k_{a,M}}{2k_1}}.$$  

(41)

According to the definition of $r_a$, it can be obtained that

$$\lim_{t \to \infty} \| \Theta \| \leq \frac{2}{k_{a,M} k_{a,M}} \sqrt{\frac{\sigma_{a,M} k_{a,M}}{2k_1}},$$  

(42)

which means that the attitude error is bounded with the dynamic couplings of the aerial manipulator.
4 Coordinate control

In this section, we propose a coordinate control method to coordinate motions of the multirotor and the manipulator based on the linear MPC method.

4.1 Coordinate control problem

The coordinate control is to calculate desired motions of the multirotor and the manipulation with the given task. The differential kinematics equation (3) describes the relationship between the position of the end-effector and states of the system. In the aerial manipulator system, the multirotor is underactuated. Therefore, not all aerial manipulator states can be controlled. To address this issue, we define \( \mathbf{h} = [p^T, q^T]^T \) as the controlled states vector. The coordinate control problem for the aerial manipulator can be formulated as to find reference values for the control vector \( \mathbf{h} \). Then, we have

\[
\mathbf{p}_c = \mathbf{J}\dot{\mathbf{h}} - (\mathbf{R}\mathbf{p}_c^d) \times \mathbf{\omega},
\]

where \( \mathbf{J} = [\mathbf{E}_3, \mathbf{R}\mathbf{p}_c^d] \) is the actuated Jacobian matrix of the aerial manipulator, \( \mathbf{E}_3 \) is an identity matrix of order 3 \( \times \) 3.

Besides, the aerial manipulator has physical limits, e.g., joint position limits, velocity limits and acceleration limits. These limits are also taken into account in solving the coordinate control problem. Generally, the physical limits of the aerial manipulator can be written as

\[
\mathbf{h}_m \leq \mathbf{h} \leq \mathbf{h}_M,
\]

\[
\tilde{\mathbf{h}}_m \leq \mathbf{h} \leq \tilde{\mathbf{h}}_M,
\]

where the subscribe \([\cdot]_m\) represents the lower boundary of the corresponding variable, the subscribe \([\cdot]_M\) represents the upper boundary of the corresponding variable.

4.2 Linear MPC Method

The linear MPC method is introduced in solving the coordinate control problem of the aerial manipulator. The MPC is an advanced control technique used to control a plant while satisfying a set of constraints. It determines an optimal control input locally at each control update time, usually via a predictive model, by propagating input and state sequences forwards in time.

The state of the aerial manipulator predictive model is defined as \( \mathbf{x} = [\mathbf{h}, \dot{\mathbf{h}}]^T \). Let \( \mathbf{y} \) denote the velocity vector of actuated joints, i.e., \( \mathbf{y} = \dot{\mathbf{h}} \). Define control input as \( \mathbf{u} = \dot{\mathbf{h}} \). Therefore, the predictive model with constant step size \( \Delta t \) can be formulated as

\[
\mathbf{x}_{t+1} = \begin{bmatrix} E_{3+n} & \Delta t E_{3+n} \\ 0_{3+n} & E_{3+n} \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 0_{3+n} \\ \Delta t E_{3+n} \end{bmatrix} \mathbf{u}_t,
\]

\[
y_{t+1} = \begin{bmatrix} 0_{3+n} \\ E_{3+n} \end{bmatrix} \mathbf{x}_{t+1},
\]

where \( 0_{3+n} \) is a zero matrix of order \((3 + n) \times (3 + n)\), \( E_{3+n} \) is an identity matrix of order \((3 + n) \times (3 + n)\), \( \Delta t \) is the time step of the predictive model.

Define position error of the end-effector as \( \mathbf{e}_c = \mathbf{p}_c - \mathbf{p}_c^d \), where \( \mathbf{p}_c^d \) is the desired position of the end-effector. The coordinate control problem is mathematically formulated as a finite prediction horizon QP problem:

\[
\min_U \sum_{i=1}^{n_p} \| \mathbf{J}_e \mathbf{y}_{k+i|k} - (\mathbf{R}\mathbf{p}_c^d) \times \mathbf{\omega} - \mathbf{\hat{\sigma}}_{k+i} \|^2
\]

\[
+ \sum_{i=1}^{n_c} \mathbf{u}_{k+i-1|k}^T \mathbf{W}_u \mathbf{u}_{k+i-1|k}
\]

s.t. \( \mathbf{x}_{k+i+1|k} = \mathbf{A}\mathbf{x}_{k+i|k} + \mathbf{B}\mathbf{u}_{k+i|k} \)

\( \mathbf{y}_{k+i+1|k} = \mathbf{C}\mathbf{x}_{k+i+1|k} \)

\( \mathbf{x}_{k+i+1|k} \in \mathcal{U}, \quad i = 0, 1, \ldots, n_p - 1 \)

where \( n_p \) is the prediction horizon of the MPC method, \( n_c \) denotes the control horizon, \((\cdot)_{k+i|k}\) denotes predictive result at time \( t_{k+i} \), \( \mathbf{J}_e \) is the actuated Jacobian matrix at time \( t_k \), \( \mathbf{W}_u \) is a positive definite weighting matrix that prioritizes the different joints, \( \mathcal{U} = [\mathbf{u}_{k+i-1|k}^T, \ldots, \mathbf{u}_{k+1|k}^T, \mathbf{u}_{k+i-1|k}^T]^T \) denotes the predictive control vector, \( \mathbf{\hat{\sigma}}_{k+i} \) is virtual control vector at time \( t_{k+i} \Delta t \) and can be calculated by

\[
\mathbf{\hat{\sigma}}_{k+i} = \mathbf{\hat{p}}_{c,k+i}^d - \mathbf{K}_e (\mathbf{p}_{c,k} - \mathbf{\hat{p}}_{c,k+i}^d),
\]

where \( \mathbf{\hat{p}}_{c,k+i}^d \) is the desired position of end-effector at time \( t_{k+i} \), \( \mathbf{p}_{c,k} \) is the position of the end-effector at time \( t_k \).

An advantage of the above formulation is that it can be solved efficiently by existing numerical QP solving methods (e.g., CVXGEN [25]). Due to introducing a predictive model, the solution also ensures that the constraints will be satisfied in the future. At each control update step \( t_k \), the QP problem is solved with an input value to obtain \( \mathbf{U} \). Although the control horizon of this algorithm \( n_c \geq 1 \), only the first control action of the sequence is implemented to the multirotor controller and the manipulator controller. Then, we can obtain the reference values for the multirotor (\( \mathbf{p}_c^d, \mathbf{\hat{p}}_c^d, \mathbf{\hat{p}}_c^d \)) and the manipulator (\( \mathbf{q}_c^d, \mathbf{\hat{q}}_c^d, \mathbf{\hat{q}}_c^d \)).
5 Control architecture

The whole control system consists of two layers (see Fig. 3). The inner layer is the proposed motion controller, and the outer layer is the proposed coordinate controller. Since the motion controller is partially decoupled, the motion controller can be divided into the multirotor and the manipulator controllers. The inputs of the system are the desired position of the end-effector $p_d^e$ and the yaw angle of the multirotor $\psi_d$. Under the giving $p_d^e$, the coordinate controller proposed in section 4 can calculate the multirotor position reference $p_d^m$ and the manipulator joint reference $q_d^m$.

The adaptive NN multirotor controller proposed in section 3 consists of the position controller and attitude controller. The lightweight manipulator is attached to the multirotor body. Generally, all joints of the manipulator are controlled by servo actuators. Therefore, the manipulator controller discussed in this paper consists of several servo controllers that can steer joints of the manipulator from current angles to desired angles. The PID method is used for designing the servo controllers. Fig. 4 illustrates the structure of the PID servo controller.

6 Simulation

An aerial manipulator which consists of a DJI F450 multirotor and a manipulator is simulated (see Fig. 5). Then, the joint position vector is written as $q = [q_1, q_2]^T$.

The aerial manipulator is simulated in Simscape which provides a multibody simulation environment for 3D mechanical systems in MATLAB [31]. Table 1 shows the masses, lengths, and physical bounds of the aerial manipulator.

6.1 Validation for the motion controller

The greatest challenge for the motion controller is the dynamic coupling between the multirotor and the manipulator. Therefore, simulations are introduced to illustrate the effectiveness of the multirotor controller when the manipulator is moving. In the simulations, the movement of the manipulator is set as Fig. 6.

To show the performance of the proposed controller method, we compared the proposed controller with a baseline controller. The baseline controller is similar to [14] and [40] with $W_q S_p(\eta_p) = 0$ and $W_q S_\Theta(\eta_\Theta) = 0$. The controller parameters in simulations are selected as $K_p = \text{diag}(1, 1, 1, 1, 1, 1)$, $K_v = \text{diag}(1, 1, 1, 1, 1, 1), K_\Theta = \text{diag}(1, 1, 1, 1, 1, 1), K_\psi = \text{diag}(1, 1, 1, 1, 1, 1).$ $K_\phi = \text{diag}(12, 18, 12), K_\psi = \text{diag}(8, 8, 8).$ The adaptive RBFNN parameters are chosen as $N = 25$, $K_p = 0.5E_{25}$, $A_k = 8E_{25}$, where $E_{25}$ is an identity matrix of order $25 \times 25$.

A position tracking task is simulated to validate the proposed multirotor controller. From Fig. 7 one can conclude that the proposed method performs much better than the baseline controller. The mean position error vector of the baseline controller is $(0.39 \text{ m}, 0.27 \text{ m}, 0.36 \text{ m})$. The mean position error vector of the proposed controller is $(0.15 \text{ m}, 0.17 \text{ m}, 0.09 \text{ m})$. These results show the performance and robustness of the proposed method in the simulation.

6.2 Validation for the coordinate controller

Three cases are simulated to illustrate the effectiveness of the coordinate controller. The first case demonstrates the proposed method when the end-effector moves from a start point to an end point. The second case demonstrates the proposed method when the end-effector moves from a start point to an end point. The third case demonstrates the proposed method when the end-effector tracks a 3D trajectory. The results of these simulation cases are shown as follows.

**Case 1:** In this case, the end-effector of the aerial manipulator moves from a start point to an end point. The end point is set as $p^e = [0.35, 0, -0.20]_m$. The start point is set as $[-0.01, 0, 0.38]_m$. Fig. 8 shows the result end-effector trajectories of the three methods. The trajectory lengths and arriving times of results are given in Tab. 2. From Fig. 8 and Tab. 2 one can obtain that the results of the proposed method are more smooth than the results of the other two methods. The result trajectory length of the proposed method is 0.29 m. It is shorter than the trajectory lengths of the CLIK method and the damped CLIK method. The duration time from start point to end point of the proposed method is 11.68 s. It is also less than the duration times of the CLIK method and the damped CLIK method.

**Case 2:** In this case, the end-effector of the aerial manipulator tracks a 3D trajectory. The 3D trajectory...
of the end-effector’s position is set as
\[
\begin{align*}
    x_e^d &= 2.38 - 2 \cos 0.1t, \\
    y_e^d &= 2 \sin 0.1t, \\
    z_e^d &= -h_e^d = -0.02t.
\end{align*}
\] (50)
$\|p_e - p_e^d\|$ as the index of the 3D tracking performance. The indexes obtained by the proposed method, the CLIK method and the damped CLIK method are 0.04 m, 0.24 m and 0.35 m, respectively.

**Fig. 7:** Simulation results of the validation for the multirotor controller.

**Case 3:** In this case, the end-effector of the aerial manipulator tracks several plane trajectories. Fig. 10 shows the results of this case. The desired position of the end-effector $p_e^d$ is given as Fig. 10. The desired position of the multirotor $(x^d, y^d, z^d)$ and the desired joint
angles of the manipulator \((q^d_1, q^d_2)\) are obtained by the proposed coordinate control method.

As stated above, these simulations illustrate the effectiveness and performance of the proposed motion controller and the proposed coordinate controller.

7 Conclusion

This paper proposed a new approach to design the end-effector position tracking controller of the aerial manipulator with a lightweight manipulator. The whole control system consists of two layers. The inner layer is the motion controller, and the outer layer is the coordinate controller. In addition, this article presented several interesting results.

According to the characters of the lightweight manipulator, a partially coupled motion controller is designed by dividing the motion controller into the multirotor and manipulator controllers. The multirotor controller is designed by the adaptive NN method, and the joints of the manipulator are controlled by the PID controllers. Dynamic couplings of the aerial manipulator are estimated by RBFNNs with an adaptive weight estimation law. The weights of the RBFNNs are trained online. With assuming that the dynamic couplings are continuous, it is proved that the desired multirotor trajectories can be boundedly tracked under the proposed adaptive NN controller. The proposed motion controller is applicable to the aerial manipulator with a lightweight manipulator since this controller does not require torque controlled motors with high update rates and does not rely on the mathematical model of the dynamic couplings. Simulations show that the proposed method performs better than the baseline controller.

The coordinate controller is designed by the linear MPC method. The coordinate control problem is formulated as a QP problem with a finite time horizon. The physical limits of the aerial manipulator are considered as constraints of the QP problem. With the predictive model, the proposed controller can ensure that the solution satisfies the physical limits. Since the MPC method obtains the control signals by minimizing the cost function, solutions of this method show high performance in the simulation cases. Compared with the CLIK method and the damped CLIK method, the solution of this method performs better in terms of trajectory length and duration time. Compared with the NMPC method, this method spends less computation cost and shows more possibility to be executed in real-time applications.

There are several important topics for future research. For example, the multirotor control method presented in this paper may be further utilized as a controller by taking the manipulator model into account. It would be interesting and practical to solve these problems in future research.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank Dr. Shiyu Zhao, at School of Engineering, Westlake University, for his help and support.

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.
References

1. H. Lee, H. J. Kim, Estimation, control, and planning for autonomous aerial transportation, IEEE Transactions on Industrial Electronics 64 (4) (2016) 3369–3379.

2. A. Suarez, G. Heredia, A. Ollero, Physical-virtual impedance control in ultralightweight and compliant dual-arm aerial manipulators, IEEE Robotics and Automation Letters 3 (3) (2018) 2553–2560.

3. A. Suarez, A. Jimenez-Cano, V. Vega, G. Heredia, A. Rodriguez-Castaño, A. Ollero, Lightweight and human-size dual arm aerial manipulator, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 1778–1784.

4. X. Ding, P. Guo, K. Xu, Y. Yu, A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems, Chinese Journal of Aeronautics 32 (01) (2019) 203–217.

5. A. Ollero, M. Tognon, A. Suarez, D. Lee, A. Franchi, Past, present, and future of aerial robotic manipulators, IEEE Transactions on Robotics.

6. X. Meng, Y. He, J. Han, Survey on aerial manipulator: System, modeling, and control, Robotics 38 (7) (2020) 1288–1317.

7. V. Lippiello, F. Ruggiero, Exploiting redundancy in cartesian impedance control of uavs equipped with a robotic arm, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2012, pp. 3768–3773.

8. F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Pérez, V. Lippiello, A. Ollero, B. Siciliano, A multilayer control for multicopter uavs equipped with a servo robot arm, in: 2015 IEEE international conference on robotics and automation (ICRA), IEEE, 2015, pp. 4014–4020.

9. A. Suarez, A. E. Jimenez-Cano, V. M. Vega, G. Heredia, A. Rodriguez-Castaño, A. Ollero, Design of a lightweight dual arm system for aerial manipulation, Mechatronics 50 (2018) 30–44.

10. F. Ruggiero, V. Lippiello, A. Ollero, Aerial manipulation: A literature review, IEEE Robotics and Automation Letters 3 (3) (2018) 1957–1964.

11. H. Yang, D. Lee, Dynamics and control of quadrotor with robotic manipulator, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2014, pp. 5544–5549.

12. S. A. Emami, A. Banazadeh, Simultaneous trajectory tracking and aerial manipulation using a multi-stage model predictive control, Aerospace Science and Technology 112 (2021) 106573.

13. A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue, A. Ollero, Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 992–997.

14. G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, A. Ollero, Control of a multicopter outdoor aerial manipulator, in: 2014 IEEE/RSJ international conference on intelligent robots and systems, IEEE, 2014, pp. 3417–3422.

15. A. Jimenez-Cano, J. Braga, G. Heredia, A. Ollero, Aerial manipulator for structure inspection by contact from the underside, in: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2015, pp. 1879–1884.

16. G. Zhang, Y. He, B. Dai, F. Gu, J. Han, G. Liu, Robust control of an aerial manipulator based on a variable inertia parameters model, IEEE Transactions on Industrial Electronics 67 (11) (2019) 9515–9525.

17. F. Ruggiero, J. Cacace, H. Sadeghian, V. Lippiello, Impedance control of vtol uavs with a momentum-based external generalized forces estimator, in: 2014 IEEE international conference on robotics and automation (ICRA), IEEE, 2014, pp. 2093–2099.

18. K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, A. Ollero, Behavioral control of unmanned aerial vehicle manipulator systems, Autonomous Robots 41 (5) (2017) 1203–1220.

19. G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, A. Ollero, Coordinated control of aerial robotic manipulators: theory and experiments, IEEE Transactions on Control Systems Technology 26 (4) (2017) 1406–1413.

20. E. Cataldi, F. Real, A. Suárez, P. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, A. Ollero, Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 2960–2966.

21. D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, J. Andrade-Cetto, Nonlinear model predictive control for aerial manipulation, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 87–93.

22. A. Errachdi, M. Benrejeb, Performance comparison of neural network training approaches in indirect adaptive control, International Journal of Control, Automation and Systems 16 (3) (2018) 1448–1458.

23. X. Yang, X. Zheng, Adaptive nn backstepping control design for a 3-dof helicopter: Theory and experiments, IEEE Transactions on Industrial Electronics 67 (5) (2020) 3967–3979.

24. S. S. Ge, C. Wang, Direct adaptive nn control of a class of nonlinear systems, IEEE Transactions on Neural Networks 13 (1) (2002) 214–221.

25. J. Mattingley, B. Stephen, Cvxgen: a code generator for embedded convex optimization 13 (1) (2011) 1–27.

26. B. Siciliano, O. Khatib, Springer handbook of robotics, springer, 2016.

27. V. Lippiello, F. Ruggiero, Cartesian impedance control of a uav with a robotic arm, IFAC Proceedings Volumes 45 (22) (2012) 704–709.

28. Z. A. Ali, X. Li, Controlling of an under-actuated quadrotor uav equipped with a manipulator, IEEE Access 8 (2020) 34664–34674.

29. J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, Neural computation 3 (2) (1991) 246–257.

30. H. K. Khalil, J. W. Grizzle, Nonlinear systems, Vol. 3, Prentice hall Upper Saddle River, NJ, 2002.

31. S. Miller, J. Wendlandt, Real-time simulation of physical systems using simscape, MATLAB News and Notes (2010) 1–13.

32. M. G. Carmichael, D. Liu, K. J. Waldron, A framework for singularity-robust manipulator control during physical human-robot interaction, The International Journal of Robotics Research 36 (5-7) (2017) 861–876.