

Received November 2, 2018, accepted November 12, 2018, date of publication November 19, 2018,  
 date of current version December 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2882241

# Vision-Based Tracking Control of Quadrotor With Backstepping Sliding Mode Control

BINGFENG ZHAO<sup>1</sup>, YANG TANG<sup>✉1</sup>, (Member, IEEE), CHUNPING WU<sup>2</sup>,  
 AND WEI DU<sup>✉1</sup>, (Member, IEEE)

<sup>1</sup>Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

<sup>2</sup>School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding authors: Yang Tang (yangtang@ecust.edu.cn) and Chunping Wu (13918352054@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0809302, in part by the National Natural Science Foundation of China under Grant 61751305 and Grant 61673176, in part by the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017, and in part by the Alexander von Humboldt Foundation of Germany.

**ABSTRACT** Vision-based quadrotor will be a good carrier for big data. This paper investigates the quadrotor tracking control by designing an adaptive sliding mode controller based on the backstepping technique with the advantages of simplicity in design and ease of application. A sliding mode controller is first developed to ensure fast convergence speed with the desired reference, and then the backstepping technique is used until the desired reference trajectory is achieved and finally the appropriate control laws are obtained. In order to achieve the precise and fast localization of a quadrotor, a popular visual odometry algorithm is applied to gathering good position information required in motion estimation. We employ Kalman filter for sensor data fusion and state estimation. Gazebo is applied by creating a 3D dynamic environment to recreate the complex environment potentially encountered in the real world.

**INDEX TERMS** Backstepping, quadrotor, Kalman filter, semi-direct monocular visual odometry, sliding mode control, tracking control.

## I. INTRODUCTION

Quadrotors are a good carrier of machine intelligence, which is increasingly entering the role that has always been dominated by human intelligence [1]. Data enabling technologies play an important role in modern scientific discovery and technological advancement [2]. A quadrotor with several sensors may produce a lot of data, so it is also a good carrier for big data [3]. Quadrotors have recently received a growing interest due to their immense applications such as surveillance, the pipeline inspection, traffic and fire monitoring, search and rescue missions [4]–[7]. A quadrotor is usually equipped with fewer rotors than degrees-of-freedom, i.e., the so-called under-actuated property, where only the translational motion can be controlled with a thrust produced along a single body-fixed axis [8]. Due to the under-actuated property, the position stabilization and tracking control of the under-actuated quadrotor are more challenging.

In fact, the tracking control issue has been the focus of a vast amount of literature over the past years, along with a myriad of successful controllers [8]–[14]. In [12] and [13]

for instance, different control schemes are developed based on disturbance estimations, which can make the tracking performance of the considered systems better. For recent work, an adaptive approximation-based regulation control method is proposed in [14]. As is known to all, the backstepping technique is an efficient nonlinear feedback control strategy for solving the robust control problem of quadrotor, along with the non-matching uncertainties existing in the quadrotor dynamics models [15], [16]. Combining with the sliding mode control method and the backstepping technique [17], the design of control laws can be simplified, which can also increase the robustness of the quadrotor. Hence, in this paper, we investigate the tracking control of under-actuated quadrotor by designing the adaptive sliding mode controller based on the backstepping technique in the presence of external disturbances.

To obtain the measurements of quadrotor for the control law, these under-actuated vehicles are generally equipped with Inertial Measurement Units (IMU) and Global Positioning Systems (GPS). The orientation and angular velocity

signals can be estimated based on IMU, while the signals from the GPS can be used to estimate the position and linear velocity [18]. However, when the GPS are unreliable or ineffective in applications in urban and indoor environments, a massive amount of work on small-scale quadrotor has been witnessed by equipping with vision sensors [19], [20], which can provide versatile visual information. Therefore, we apply monocular semi-direct visual odometry (SVO) algorithm for obtaining the position information of the quadrotor [21]. More specifically, the algorithm combines the high accuracy and quick speed of direct methods. The feature-based methods, where the robustness and faster flight maneuvers are increased by the high frame-rate visual odometry.

In addition, a simulator is more advantageous for modeling outdoor environments which can provide the realistic sensor feedback, despite the stage can simulate the dynamic state of quadrotor in indoor environments. Hence, the development of Gazebo [22], [23] has been greatly driven due to the extensively applications of some vehicles in outdoor motions, including robotic vehicles and quadrotors [24]. Gazebo is mainly used to accurately reproduce the dynamic environments that quadrotors may encounter. When Gazebo is applied, the simulated quadrotor has mass, position, velocity, friction and other attributes so that they can behave realistically. By using Gazebo, in this paper, we verify the proposed adaptive sliding mode controller based on the backstepping technique, and test the monocular SVO algorithm.

There are several nice studies in tracking control. In [14], a novel adaptive approximation based regulation control is proposed, in which a backpropagating constraints-based trajectory tracking control (BCTTC) scheme is used for tracking control. In [25], the robust integral of the signum of the error (RISE) method and the adaptive control method based on immersion and invariance (I&I) are used for quadrotor tracking control. However, image information is not used for localization in those papers. There are several vision-based tracking control methods in [26]–[29]. In [26], a fuzzy PID controller is designed, which can tune the parameters of the PID controller according to different conditions. In [27], an object localization and tracking strategy is designed from monocular image sequences, and putting tracking information into a dynamic Kalman model. However, in this paper, the camera is placed at a fixed place of the environment, not on the quadrotor. To a certain extent, it is more like using global information. In [28], a vision-based tracking method is proposed for a micro-aerial vehicle. The camera of the quadrotor is mounted downward facing the ground. However, it detects the ground vehicle by obvious features and then calculates the position of UAV. Hence it can't be used in a low-textured background. In [29], an image-based visual servoing (IBVS) scheme is used to track image features in two-dimensional image space. However, it suffers from the problem of rich features in the environment. With the help of SVO, the position of quadrotor can be estimated without priori features and conspicuous features to achieve the control objective in our proposed paper. In [30], in order to meet the

challenge with little no-priori knowledge of the environment, a system is designed and a software architecture is proposed. A stereo camera is used and gyroscope data is considered for the computation of SVO. The method can improve the accuracy but it brings additional hardware cost. The architecture is similar to ours. However, we use a monocular camera for localization, due to the comparable accuracy and the reduction of the mechanical complexity. As for tracking control method, a typical method is given in [31], which can be considered as a PD controller. We put forward a sliding mode controller for better performance. In both simulation and Gazebo platform, we can see that our method has better performance than the method described in [31].

The contributions of this paper can be summarized as: (a) The monocular SVO algorithm is applied to obtaining accurate location information by motion estimation and mapping thread. Some modification was applied to accommodating our application. (b) A sliding mode controller is proposed with the backstepping method, and a proportion-integral-derivative (PID) controller is used for comparison in simulation. (c) Gazebo is applied by creating a 3D dynamic environment to test the two tracking controllers. The remainder of this paper is organized as follows. Some notations and models are given in Section II. System structure will be illustrated in Section III. Semi-direct visual odometry is introduced briefly in Section IV. In Section V, we use Kalman filter for state estimation. We propose a sliding mode controller for tracking in Section VI. A practical experiment and conclusion are respectively derived in Sections VII and VIII.

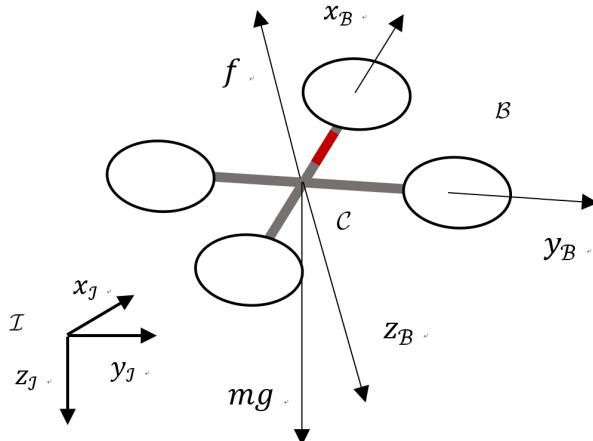
## II. PRELIMINARIES

### A. NOTATION

In this paper, we use calligraphic letters for coordinate frames (e.g.  $\mathcal{A}$ ).  $\mathbb{R}$  is the set of real numbers.  $\mathbb{R}^n$  is a vector with  $n$  real items. A rigid body transformation from coordinate  $\mathcal{B}$  to  $\mathcal{A}$  is denoted by  $T_{\mathcal{B}}^{\mathcal{A}} \in \text{SE}(3)$ , where  $\text{SE}(3)$  is Special Euclidean group in a three-dimension space. Any transformation  $T$  can be taken apart as the rotational part  $R \in \text{SO}(3)$  and the translation part  $t \in \mathbb{R}^3$ .  $\text{SO}(3) = \{R \in \mathbb{R}^{3 \times 3} | R^T R = I, \det R = 1\}$  means Special Orthogonal group in three-dimension space, and the corresponding Lie algebra is  $\mathfrak{so}(3)$ . Because we use monocular camera for acquiring images,  $\text{Sim}(3)$ , which means Similar transformation group, is used to indicate the up-to-scale transformation.

### B. QUADROTOR MODEL

Quadrotor has two pairs of orthogonal propellers. The pair of rotors on the same side has the same direction of rotation. One of the pairs spins clockwise while the other counter-clockwise. By adjusting the spinning rate of each motor, the torque of quadrotor can be changed. The torque is generated by the difference in spinning rate of motors. The change of torque can further drive the attitude of the drone, and thus we can achieve the tracking control of trajectory. A quadrotor is an under-actuated system because the six degrees of



**FIGURE 1.** The relationship among body frame, inertial frame and camera frame.

freedom in the drone space are controlled by the spinning rate of the four motors. Three frames are considered in this paper, i.e., body frame  $\mathcal{B}$  oriented forward right down, inertial frame  $\mathcal{I}$  oriented north east down (NED) and camera frame  $\mathcal{C}$ . The relationships among the three frames are illustrated in Figure 1.

A body frame is represented as  $\mathcal{B}$  while an inertial frame is  $\mathcal{I}$ . The origin of the body-fixed frame is located at the center of the mass of the quadrotor. Let  $x \in \mathbb{R}^3$  and  $v \in \mathbb{R}^3$  respectively denote the position and velocity of a quadrotor in frame  $\mathcal{I}$ .

The kinematic and dynamics equations of the rigid body are as follows:

$$\dot{x} = v, \quad (1a)$$

$$\dot{R} = R\hat{\Omega}, \quad (1b)$$

$$\dot{v} = mge_3 - fRe_3, \quad (1c)$$

$$J\dot{\Omega} = -\Omega \times J\Omega + \tau, \quad (1d)$$

where the attitude is denoted by the rotation matrix  $R$  and three Euler angles, the roll  $\phi$ , pitch  $\theta$ , and yaw  $\psi$ .  $f$  is the thrust in body frame  $\mathcal{B}$ .  $\tau$  is the moment controlled by attitude controller.  $\Omega$  means the angular rate in  $\mathcal{B}$  frame.  $mg$  is the gravity vector.  $e_3 = [0 \ 0 \ 1]$ ,  $\hat{\cdot}$  denotes the map from  $\mathbb{R}^3$  to  $\mathfrak{so}(3)$ , e.g.  $\hat{xy} = x \times y$ , for any  $x, y \in \mathbb{R}^3$ .

For vision-based control, a camera is mounted on the bottom of the quadrotor, which can shoot the scene beneath the body frame when the quadrotor flies. There is a transformation from camera frame  $\mathcal{C}$  to body frame  $\mathcal{B}$ , which is denoted by  $T_{\mathcal{B}}^{\mathcal{C}} \in \text{SE}(3)$ . The intensity image from camera  $\mathcal{C}$  at time  $k$  is denoted by  $I_k^{\mathcal{C}} : \Omega^{\mathcal{C}} \subset \mathbb{R}^2 \mapsto \mathbb{R}$ , where  $u \in \Omega^{\mathcal{C}}$  is the image pixel coordinate. A point in 3D space  $\rho \in \mathbb{R}^3$  maps to image pixel position  $u \in \mathbb{R}^2$  through the camera projection model. The relationship between  $u$  and  $\rho$  is  $u = \pi(\rho)$ , and the back-projection model is  $\rho = \pi^{-1}(u)$ .  $\mathcal{R}_k^{\mathcal{C}} \subset \Omega$  denotes that the set of pixels for which depth can be estimated at the  $k$ th time in camera  $\mathcal{C}$ . 3D-space point in world frame  $\rho_{\mathcal{I}}$  can

be transformed to camera frame  $\mathcal{C}$ :

$$\rho_{\mathcal{C}} = T_{\mathcal{C}}^{\mathcal{I}} \rho_{\mathcal{I}}. \quad (2)$$

### III. SYSTEM STRUCTURE

The system structure is shown in Figure 2. In measurement step, visual odometry is used to obtain position data. Kalman filter block gathers the information from sensors and estimates the state value. In tracking controller block, PID or sliding mode controller was used to drive the quadrotor to the desired position. We use the attitude controller and the rotor rate allocation scheme is illustrated in [31].

According to [31], there are two control inputs:  $f$  is the total thrust and  $\tau$  is the total moment. They are chosen as follows:

$$f = -(k_x e_x - k_v e_v - mge_3 + m\ddot{x}_d)Re_3, \quad (3)$$

$$\tau = -k_R e_R - k_{\Omega} e_{\Omega} + \Omega \times J\Omega, \quad (4)$$

where  $e_x$  is the error between current position and desired position, the error between current and desired velocity is  $e_v$ .  $k_x$  and  $k_v$  are proportionality coefficients of position and velocity error, respectively. This method can be seen as PD tracking control. The controller to be designed is a continuous-time controller while the Kalman filter is a discrete-time process, hence the system is a hybrid system [32]. The colored parts will be introduced in the following sections.

### IV. VISUAL ODOMETRY

In this paper, we use the semi-direct visual odometry (SVO) framework proposed in [19] and [21] for state estimation. SVO combines the merits of both feature-based methods and direct methods. This approach creates two threads, one is for motion estimation and the other is for mapping. In the motion estimation thread, motion between two recent frames is estimated by image alignment. For accuracy improvement, a direct method on patches is used for refinement of subpixel feature. In mapping thread, feature depth is estimated using inverse depth filter. Here is an overview for this method:

#### A. MOTION ESTIMATION

We want to estimate the motion between two sequential frames  $k-1$  and  $k$ , and  $T_{kk-1} = T_{\mathcal{B}_{k-1}}^{\mathcal{B}_k}$  is the transformation between the two frames in the body frame  $\mathcal{B}$ , so we minimize the photometric error:

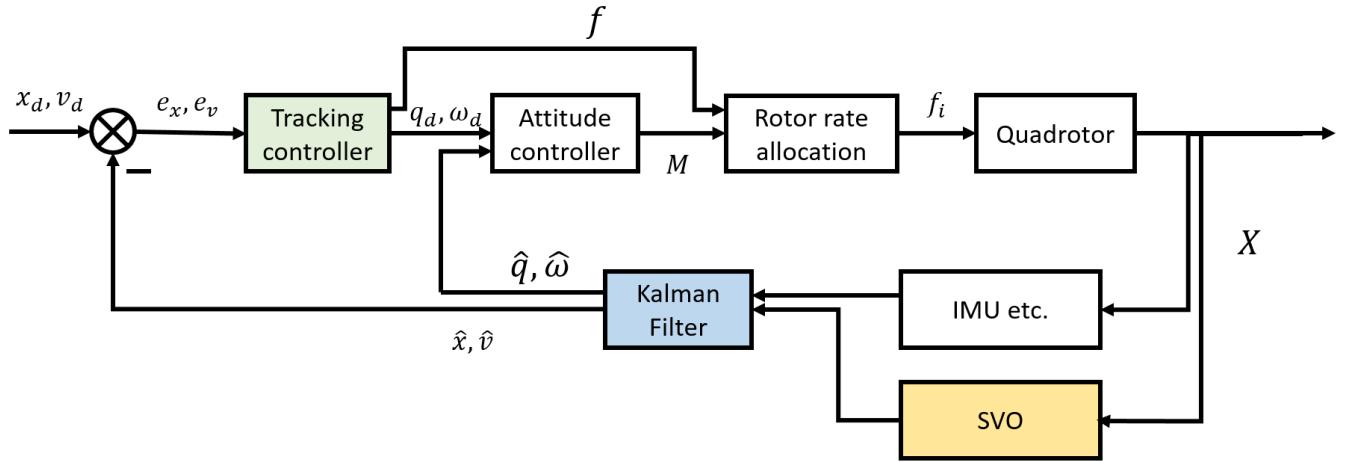
$$T_{kk-1}^* = \arg \min_{T_{kk-1}} \sum_{u \in \mathcal{R}_{k-1}^{\mathcal{C}}} \frac{1}{2} \|r_{I_u^{\mathcal{C}}}(T_{kk-1})\|_{\Sigma_I}^2, \quad (5)$$

where,  $r_{I_u^{\mathcal{C}}}$  is the photometric error, which means the intensity difference in sequential images  $I_k^{\mathcal{C}}$  and  $I_{k-1}^{\mathcal{C}}$  of each pixel that observes the same point in 3D space in the camera frame  $\rho_{\mathcal{C}}$ .

$$r_{I_u^{\mathcal{C}}}(T_{kk-1}) := I_k^{\mathcal{C}}(\pi(T_{\mathcal{C}\mathcal{B}} T_{kk-1} \rho_{\mathcal{C}})) - I_{k-1}^{\mathcal{C}}(\pi(T_{\mathcal{C}\mathcal{B}} \rho_{\mathcal{C}})), \quad (6)$$

the corresponding 3D position of point  $\rho_{\mathcal{C}}$  for a pixel with known depth, expressed in the previous frame  $\mathcal{B}_{k-1}$ , can be calculated by back-projection:

$$\rho_{\mathcal{C}} = T_{\mathcal{B}\mathcal{C}} \pi_{\rho}^{-1}(u), \forall u \in \mathcal{R}_{k-1}^{\mathcal{C}}. \quad (7)$$



**FIGURE 2.** Block diagram of system structure.

A non-linear least squares problem in (5) is solved by sparse image alignment. Next the geometry constraint last step will be loosened to decrease the error of 3D points and the camera pose. We detect the edge and the corner features to represent points that have strong gradient in the image.  $u$  is the projected feature point position in the image pixel coordinate.

Here two types of features are considered. The first type is corner, we calculate the correction  $\delta u^*$  by minimizing the photometric cost:

$$\delta u^* = \arg \min_{\delta u} \sum_{\Delta u \in \mathcal{P}} \frac{1}{2} \|I_k^C(u' + \delta u + \Delta u) - I_r^C(u + A\Delta u)\|^2, \quad (8)$$

where,

$$u'^* = u' + \delta u^*, \quad u' = \pi(T_{CB}T_{kr}T_{BC}\pi_\rho^{-1}(u)), \quad (9)$$

and  $\Delta u$  is used to sum over the patch  $\mathcal{P}$ , the value of  $\Delta u$  in every iteration.  $A$  is affine warping matrix, which is used by the reference  $r$  patch. Lucas-Kanade algorithm [33] was used to solve the alignment problem.

The second type is edge feature, we find a scale correction value along the direction edge normal direction  $n$  to acquire the relevant features position  $u^*$  in the latest frame:

$$u'^* = u' + \delta u^* \cdot n, \quad (10)$$

where

$$\delta u^* = \arg \min_{\delta u} \sum_{\Delta u \in \mathcal{P}} \frac{1}{2} \|I_k^C(u' + \delta u \cdot n + \Delta u) - I_r^C(u + A\Delta u)\|^2. \quad (11)$$

In the previous step, we established the feature relative with subpixel accuracy. However, the feature alignment breaks the epipolar constraints and deduces a reprojection error  $\delta u$ . So we need to correct the camera poses and landmark

points  $\chi = \{T_{kW}, \rho_i\}$  by optimizing the sum of re-projection errors:

$$\begin{aligned} \chi^* = \arg \min_{\chi} & \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}_k^C} \frac{1}{2} \|u_i'^* - \pi(T_{CB}T_k\pi_\rho(\rho_i))\|^2 \\ & + \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{L}_k^E} \frac{1}{2} \|n_i^T(u_i'^* - \pi(T_{CB}T_k\pi_\rho(\rho_i)))\|^2, \end{aligned} \quad (12)$$

where  $\mathcal{K}$  is the set of all keyframes in the map,  $\mathcal{L}_k^C$  and  $\mathcal{L}_k^E$  are all corner feature landmarks and all edge feature set, respectively.

## B. MAPPING

Here we will describe how the mapping thread calculates the depth of the most recently discovered feature. Assuming that the camera pose can be known from the previous motion estimation step. A recursive Bayesian depth filter is proposed for estimating the depth of single pixel.

We denote that  $d$  is the depth of single pixel, and we assume that the inverse depth of single pixel obeys the sum of two distribution according to [34]: one is the inverse depth  $\rho$ , which follows a normal distribution, in which the mean value is the real inverse depth according to  $\rho_i$  [35]; whereas the other is the inlier probability  $\gamma$ . So a measurement can be modeled with a mixture of a Gaussian and a Uniform model distribution: one is inlier measurement of mean value, which denotes the true inverse depth  $\rho_i$ . The other is a uniform distribution, the maximum value is  $\rho_i^{\max}$  and the minimum value is  $\rho_i^{\min}$ :

$$p(\tilde{\rho}_i^k | \rho_i, \gamma_i) = \gamma_i \mathcal{N}(\tilde{\rho}_i^k | \rho_i, \tau_i^2) + (1 - \gamma_i) \mathcal{U}(\tilde{\rho}_i^k | \rho_i^{\min}, \rho_i^{\max}), \quad (13)$$

where  $\gamma_i$  denotes the inlier probability.  $\mathcal{N}(\tilde{\rho}_i^k | \rho_i, \tau_i^2)$  denotes that  $\tilde{\rho}_i^k$  follows a normal distribution with mean  $\rho_i$  and variance  $\tau_i^2$ .  $\mathcal{U}(\tilde{\rho}_i^k | \rho_i^{\min}, \rho_i^{\max})$  denotes that  $\tilde{\rho}_i^k$  follows a uniform distribution, the minimum is  $\rho_i^{\min}$  and the maximum is  $\rho_i^{\max}$ .

By assuming a different variance of one pixel in the image plane, a good measurement can be calculated by the geometric method, and  $\tau_i^2$  is the variance of the measurement. If only a small range around the current depth estimate is searched on the epipolar line, the depth estimate will have high performance.

Assuming all observations are independent, we get Bayesian estimation for  $\rho$  using the measurements from  $\tilde{\rho}_{r+1}$  to  $\tilde{\rho}_k$ :

$$p(\rho, \gamma | \tilde{\rho}_{r+1}, \dots, \tilde{\rho}_k) = p(\rho, \gamma) \prod_k p(\tilde{\rho}_k | \rho, \gamma) \cdot \text{const}, \quad (14)$$

where  $p(\tilde{\rho}_k | \rho, \gamma)$  is a prior on the true inverse depth. For iterative calculation of the posterior estimation, [35] shows that (14) can be recognized as a Gaussian  $\times$  Beta distribution. The Beta distribution is denoted as follows:

$$q(\rho, \gamma | a_k, b_k, \mu_k, \sigma_k^2) = \text{Beta}(\gamma | a_k, b_k) \mathcal{N}(\rho | \mu_k, \sigma_k^2), \quad (15)$$

where Beta(\*) means \* is of Beta distribution.  $a_k$  and  $b_k$  are the parameters of a Beta distribution. Beta  $\times$  Gaussian is an approximating distribution, this distribution can minimize the Kullback-Leibler divergence from the true posterior (14). After the  $k$ th observation, the update result is as follows:

$$p(\rho, \gamma | \tilde{\rho}_{r+1}, \dots, \tilde{\rho}_k) \approx q(d, \gamma | a_{k-1}, b_{k-1}, \mu_{k-1}, \sigma_{k-1}^2) \cdot p(\tilde{\rho}_k | d, \gamma) \cdot \text{const}, \quad (16)$$

and by matching the first and second moments of  $\hat{d}$  and  $\gamma$ , we can approximate the true posterior with a Beta  $\times$  Gaussian distribution illustrated in (16). The parameters selection and update formulas for  $a_k$ ,  $b_k$ ,  $\mu_k$  and  $\sigma_k^2$  can also be found in [34].

### C. IMPLEMENTATION DETAILS

Our application is that the camera is mounted downward to the ground and the depth information is almost the same. Although [19] illustrates the application on multi-camera system, here we only consider one camera mounted on the quadrotor.

In order to bootstrap the algorithm, the pose of the first two key frames is used to establish an initial map. We assume that the first two key frames are in a local planar. Then we estimate a homography matrix, which is fit for our application. The initial map is triangulated from the first two frames.

An image pyramid of the five levels is created and half-sampling was applied to the image. In the coarsest level, the image was sampled only a few pixels, which can reduce the calculation cost, significantly. The photometric cost is minimized at the coarsest pyramid level. If the error does not converge, features in lower pyramid level will be first calculated, then all the levels of pyramid can be computed.

In mapping thread, for the sake of simplifying the calculation, the image is divided into blocks of fixed size.

By a proper bootstrap result, original point can be established. By the motion estimation threads, transformation between keyframes can be found. With the help of mapping

thread, accumulative errors in first thread can be reduced. We can finally get the pose up-to-scale:

$$T_S^C = sR_S^C + t_S^C, \quad (17)$$

where  $S$  denotes the SVO frame. We assume that our displacement is not large, so the error between real scene and SVO frame is small. Under this assumption, there is an up-to-scale transformation  $\{s, R_S^I, t_S^I\} \in \text{Sim}(3)$  from SVO frame  $S$  to inertial frame  $I$ .

### V. KALMAN FILTER

For quadrotor pose estimation, we define the state vector of dynamic model as follows:

$$X = [q_I^T \ x_I^T \ v_I^T \ \Omega_B^T \ a_B^T]^T, \quad (18)$$

where  $q_B^{I,T}$  denotes the rotation quaternion from body frame  $B$  to inertial frame  $I$ .  $x_I$  and  $v_I$  are position and velocity of three axes in inertial frame  $I$ ,  $\omega_B$  and  $a_B$  are angular rate and linear acceleration in body frame  $B$ , respectively.

A discrete dynamic model can be:

$$X_k = F_k X_{k-1} + \gamma_k, \quad (19)$$

where  $F_k$  is a block diagonal matrix,  $k$  is the sample step, and  $\gamma_k$  is the Gaussian white noise vector with zero-mean, and the covariance is  $Q_k$ , i.e.,

$$E[\gamma_i] = q_i, \quad E[(\gamma_i - q_i)(\gamma_i - q_i)^T] = Q_i \delta_{ij}, \quad (20)$$

where  $E(*)$  means the expectation of \*;  $q_i$  and  $Q_i$  are true mean and true covariance, respectively;  $\delta$  is Kronecker delta function. The measurement process of SVO estimation can be as follows:

$$z_k = G(X_k) + v_k, \quad (21)$$

with the SVO position estimation result:

$$z_k = [x_{S_k} \ y_{S_k} \ z_{S_k}]^T, \quad (22)$$

and

$$G(X_k) = [0_{4 \times 3}^T \ (sR_S^I x_{k,I} + t_S^I)^T \ 0_{3 \times 3}^T \ 0_{3 \times 3}^T \ 0_{3 \times 3}^T]^T. \quad (23)$$

Because SVO is bootstrapped by a monocular camera, there is a factor  $s$  that denotes the scale from SVO estimation position to real scene.  $R_S^I \in \text{SO}(3)$  is rotation matrix from body frame to SVO frame  $S$ .  $t_S^I$  is the displacement vector from inertial frame  $I$  to SVO frame  $S$ . Scale  $s$  can be estimated by the height measurement,  $R_S^I$  and  $t_S^I$  can be estimated in SVO bootstrapping step.  $v_k$  denotes the SVO measurement noise, which is recognized as a zero-mean Gaussian distribution with covariance  $R_k$ , i.e.,

$$E[v_i] = r_i, \quad E[(v_i - r_i)(v_i - r_i)^T] = R_i \delta_{ij}, \quad (24)$$

where  $r_i$  and  $R_i$  are the true mean value and true covariance of SVO measurement, respectively.

Let  $\hat{X}_k$  be a priori state estimation at step  $k$ , which is predicted by the measurement at the end of step  $k-1$ , and

let  $\hat{X}_{k,k}$  be the posteriori state estimation at step  $k$  given by measurement  $z_k$ . Then the priori value, the posteriori estimation errors and their corresponding covariances are  $e_k = X_k - \hat{X}_{k,k}$ ,  $P_{k,k} = E[e_k e_k^T]$ ,  $e_{k,k-1} = X_k - \hat{X}_{k,k-1}$  and  $P_{k,k-1} = E[e_{k,k-1} e_{k,k-1}^T]$ , respectively. We can know from the Kalman filter algorithm in which there are two major parts:

Prediction step:

$$\hat{X}_{k,k-1} = F\hat{X}_{k-1,k-1}, \quad (25)$$

$$\hat{P}_{k,k-1} = FP_{k-1,k-1}F^T + Q_{k-1}. \quad (26)$$

Correction step:

$$\hat{X}_{k,k} = \hat{X}_{k,k-1} + K_k(z_k - G(\hat{X}_{k,k-1})), \quad (27)$$

$$P_{k,k} = P_{k,k-1} - K_k H_k P_{k,k-1}, \quad (28)$$

$$K_k = P_{k,k-1} H_k^T R_k + H_k P_{k,k-1} H_k^{T-1}, \quad (29)$$

where  $K_k$  denotes the Kalman gain matrix at step  $k$ . The measurement noise and process noise covariances are  $Q_k$  and  $R_k$ . They are usually considered as constant matrix which can be calibrated in experiment. In the prediction step, we get angular rate  $\Omega_B$  and linear acceleration data  $a_B$  in body frame from gyroscope and accelerometer. In the correction step, magnetometer is the orientation data provider, we can get rotation matrix  $R_I^B$  and quaternion  $q_I$  from it. For more details one can refer to [36].

## VI. SLIDING MODE CONTROLLER

Consider the nonlinear system model below:

$$\begin{aligned} \dot{x}_1 &= Ax_2, \\ \dot{x}_2 &= m(x_2) + Bu + d, \end{aligned} \quad (30)$$

where  $x_1, x_2, u \in \mathbb{R}^n$  are vectors with  $n$  dimension,  $A, B \in \mathbb{R}^{n \times n}$  are  $n$  dimension square constant matrices.  $m(x_2)$  denotes the part which is easy to model.  $d$  is the uncertain interference, which cannot be modeled in  $m(x_2)$ .

In the actual scenario, there are many uncertainties in quadrotor control, such as the influence of wind, flight vibration and velocity fluctuation. For the sake of designing an accurate and handy model, we make one constraint for  $d$ : we assume that  $d$  has an unknown upper bound.  $\bar{d}$  is a unknown positive number, which is not less than the upper bound of  $d$ .

In traditional sliding model control, the chattering is difficult to eliminated, which may cause discontinuous virtual control. At the same time, the rapid change of input will damage the real system. Therefore, to solve the problem above, a continuous sliding mode control with norm switching function is utilized.

Here we consider two sliding surfaces:

$$z_1 = x_1 - x_{1d}, \quad (31)$$

$$z_2 = x_2 - x_{2d}. \quad (32)$$

In (31) and (32),  $z_1, z_2 \in \mathbb{R}^3$  are the tracking errors of  $x_1$  and  $x_2$ .  $x_{1d}$  is the desired trajectory of the physical system, and  $x_{2d}$  is the virtual control signal, which is given by control

system. We can get dynamic equations from (30) and (31) as follows:

$$\dot{z}_1 = Ax_2 - \dot{x}_{1d}. \quad (33)$$

Splitting  $x_2$  into  $(z_2 + x_{2d})$ , we get:

$$\dot{z}_1 = Az_2 - Ax_{2d} - \dot{x}_{1d}, \quad (34)$$

here, we construct the first Lyapunov function:

$$V_1 = \frac{1}{2}z_1^T z_1. \quad (35)$$

It can be easily checked that  $V_1 \geq 0$ . It's differential is as follows:

$$\begin{aligned} \dot{V}_1 &= z_1^T \dot{z}_1 \\ &= z_1^T (Ax_2 - \dot{x}_{1d}) \\ &= z_1^T (A(z_2 + x_{2d} - \dot{x}_{1d})) \\ &= z_1^T Az_2 + z_1^T (Ax_{2d} - \dot{x}_{1d}). \end{aligned} \quad (36)$$

Here we assign  $x_{2d}$ :

$$x_{2d} = -A^{-1}(k_1 z_1 - \dot{x}_{1d}). \quad (37)$$

We can get the equation from (36) and (37):

$$\dot{V}_1 = -k_1 z_1^T z_1 + z_1^T A z_2, \quad (38)$$

where the positive constant  $k_1$  can be adjusted according the specific model. We can guarantee that the first term of the right side of (38) is non-positive, next we try to deal with the second term  $z_1^T A z_2$ . The second Lyapunov function is as follows:

$$V_2 = \frac{1}{2}z_2^T z_2. \quad (39)$$

It is obvious that  $V_2 \geq 0$ . We compute the differential of  $V_2$ :

$$\begin{aligned} \dot{V}_2 &= z_2^T z_2 \\ &= z_2^T (m(x_2) + Bu + d - \dot{x}_{2d}). \end{aligned} \quad (40)$$

We assign a positive constant parameter  $k_2$ , we get

$$u = B^{-1}[-m(x_2) - A^T z_1 - k_2 z_2 - \Delta + \dot{x}_{2d}], \quad (41)$$

here,

$$\Delta = d \frac{z_2}{\|z_2\|}. \quad (42)$$

Substituting (41) into (40), we get:

$$\begin{aligned} \dot{V}_2 &= z_2^T (-A^T z_1 - k_2 z_2 + d - \Delta) \\ &= -z_2^T A^T z_1 - k_2 z_2^T z_2 + z_2^T (d - \Delta). \end{aligned} \quad (43)$$

According to the assumption that the disturbance  $d$  is bounded, we set another unknown bounded term  $\rho$  to compensate the uncertain disturbance, and the upper bound of  $\rho$  is  $\hat{\rho}$ . Combining with (38) and (43), we get:

$$\begin{aligned} \dot{V}_1 + \dot{V}_2 &= -k_1 z_1^T z_1 - k_2 z_2^T z_2 + z_2^T (d - \Delta) \\ &\leq -k_1 z_1^T z_1 - k_2 z_2^T z_2 + \|z_2\| \rho - z_2^T \Delta, \end{aligned} \quad (44)$$

here we assign that:

$$\Delta = \hat{\rho} \frac{z_2}{\|z_2\|}. \quad (45)$$

Then (44) becomes:

$$\dot{V}_1 + \dot{V}_2 = -k_1 \|z_1\|^2 - k_2 \|z_2\|^2 + z_2^T (\rho - \hat{\rho}). \quad (46)$$

Now we design the third Lyapunov function as follows:

$$V_3 = \frac{1}{2r} (\rho - \hat{\rho})^2. \quad (47)$$

It is obvious that  $V_3 \geq 0$ . The differential of  $V_3$  is:

$$\dot{V}_3 = -\frac{1}{r} (\rho - \hat{\rho}) \dot{\hat{\rho}}. \quad (48)$$

With the adaptive regulation law, we get the estimation:

$$\dot{\hat{\rho}} = r \|z_2\| \geq 0. \quad (49)$$

Combining with the three Lyapunov functions, we can get:

$$\begin{aligned} \dot{V} &= \dot{V}_1 + \dot{V}_2 + \dot{V}_3 \\ &= -k_1 \|z_1\|^2 - k_2 \|z_2\|^2 + \|z_2\| (\rho - \hat{\rho}) - \frac{1}{r} (\rho - \hat{\rho}) \dot{\hat{\rho}} \leq 0. \end{aligned} \quad (50)$$

Since  $V_1, V_2, V_3$  are nonnegative and  $\dot{V}$  is nonpositive, then the above equality holds if and only if  $z_1 = 0$  and  $z_2 = 0$ . So the system is stable according to Lyapunov stability theorem.

For our proposed controller, the controller can be described as follows:

$$u = B^{-1} [-m(v) - A^{-1} e_x - k_2 e_v - \Delta + \dot{x}_{2d}] \quad (51)$$

$$f = (u - m g e_3 + m \ddot{x}_d) R e_3, \quad (52)$$

$$\tau = -k_R e_R - k_\Omega e_\Omega + \Omega \times J \Omega, \quad (53)$$

$x_{2d}$  is given by (37). After the variable substitution we get:

$$x_{2d} = -A^{-1} (-k_1 e_x + v_d). \quad (54)$$

here,  $v_d$  is the desired velocity of the quadrotor.  $\Delta$  can be described as sign function of  $e_v$ :

$$\Delta = \hat{\rho} \frac{e_v}{\|e_v\|} = \hat{\rho} * \text{sgn}(e_v). \quad (55)$$

Compared with the traditional PID controller, we can find that additional compensation items  $m(v)$  and  $\Delta$  are added to the controller.  $\dot{x}_{2d}$  is added for the sliding mode controller.

## VII. SIMULATIONS AND EXPERIMENTS

The adaptive sliding mode controller proposed in Section VI and traditional PID controller will be tested in the following simulation. Firstly, we will illustrate the simulation results of the two controllers with the help of MATLAB [39], Gazebo simulations are applied to compare the performance of two controllers later.

## A. SIMULATION RESULTS

In both traditional PID and the proposed sliding mode controller simulations, the parameters in model (30) are:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (56a)$$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (56b)$$

$$m(x_2) = \frac{1}{2} C_D(\theta) \rho_a x_2^2 S. \quad (56c)$$

Here,  $m(x_2)$  means the aerodynamic fraction according to [37],  $\rho_a$  is the air density,  $C_D(\theta)$  is the whole vehicle drag coefficient related to  $\theta$ , we assume that  $\theta$  maintains a small number, so we calculate the maximum of  $C_D$ .  $S$  is the maximum cross-sectional area of quadrotor, we assume that  $S$  is no less than  $0.03m^2$ . External disturbance  $d = 0.1 \sin(t)$  is added to the model. We add a random noise to the measurement of position, the amplitude of noise is 0.02. Considering the hardware limitation, we add an extra constraint  $\|u\| \leq 3$ , which means that the input acceleration has a maximum value. The attitude control loop and the motor power distribution are essential for a standard control system. The output of the tracking controller is the input of attitude controller, we make an assumption that the pitch and the roll angle is small, so the accelerations along axis  $x$  and  $y$  are approximately proportional to the pitch and roll, respectively.

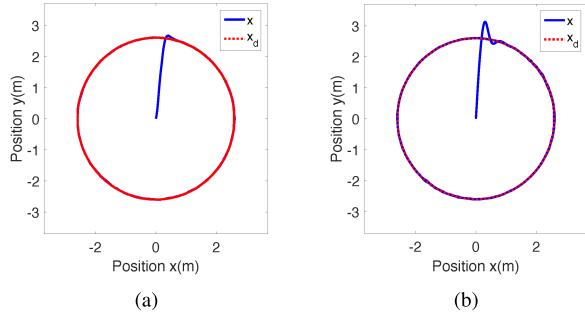
For the controller we proposed in (41), the parameters are chosen as  $k_1 = 0.7$ ,  $k_2 = 1.2$ . As for PID controller, we choose  $k_p = 2.8$  and  $k_d = 1.5$ . The trajectory overshoot is larger when we choose a large  $k_p$ , the response speed becomes slow when we decrease  $k_p$ . The trajectory chattering becomes serious when  $k_d$  is too small. A large  $k_d$  will bring a slow response result. According to the above considerations, we choose  $k_p$  as 2.8 and  $k_d$  as 1.5 in the comparison.

In initial conditions, quadrotor is placed at the origin point  $[0 \ 0 \ 0]^T$ . The desired trajectory is designed as  $x_{1d} = [2.6 \sin(0.05t) \ 2.6 \cos(0.05t) \ 3]^T$ . A traditional proportion-integral-derivative (PID) controller is proposed for comparison. Figures 3-4 show the simulation results of the traditional PID and the sliding mode controller.

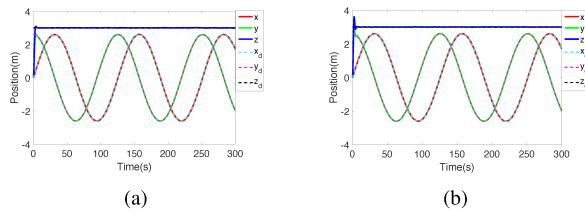
The simulation result shows that the designed controller in this paper can track the desired trajectory accurately. Proposed sliding mode controller has better performance than the traditional PID controller, both in reaction speed and tracking accuracy. It is clear that the proposed sliding mode controller has less overshoot, and a faster responding speed than the traditional PID controller does. The tracking result is quite similar.

## B. GAZEBO SIMULATION RESULTS

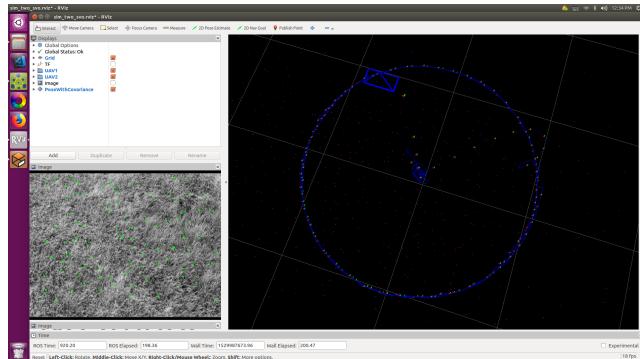
Our platform is tested on a PC with Intel Core i7-7700 microprocessor, running at 3.60 GHz with 8G RAM. The operation system is Ubuntu 16.04 LTS. We choose ROS [38], the version of which is kinetic, for message communication



**FIGURE 3.** Tracking results in simulation. (a) The proposed SMC trajectory in 2D plane. (b) The traditional PID controller Trajectory in 2D plane.  $x_d$  is the desired trajectory and  $x$  is the simulation result.



**FIGURE 4.** Tracking evolution results in simulation. (a) SMC tracking. (b) PID tracking. The dashlines in both pictures are the desired value and blue solid lines mean the simulation result.

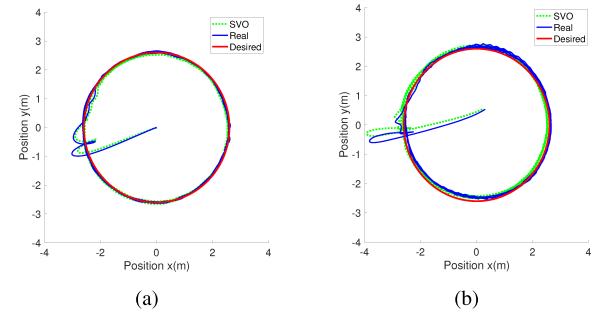


**FIGURE 5.** ROS simulation scene, where camera pose is indicated by blue cubes.

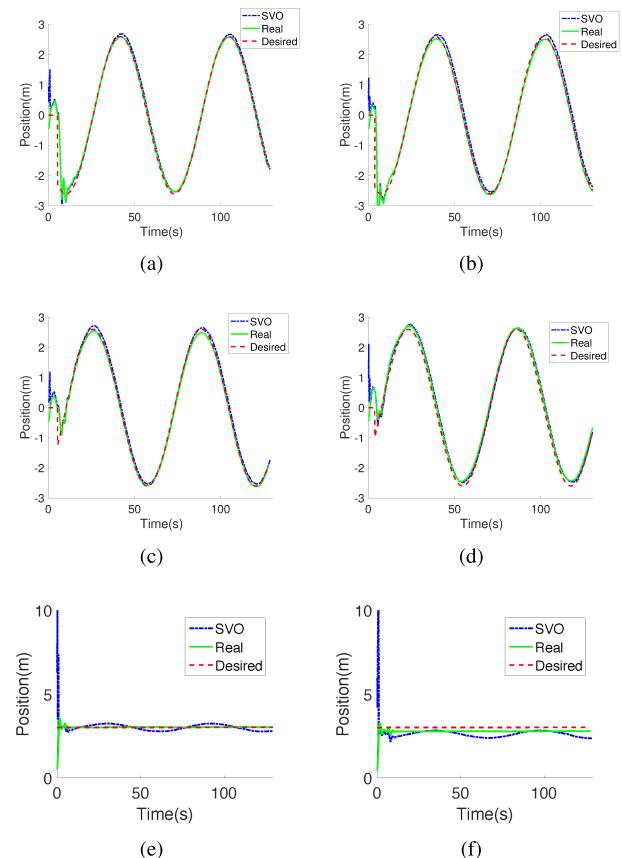
and Gazebo for sensor data capturing. We use rotor-simulator framework [22] for simulation. The simulation block diagram is shown in Figure 5.

We record the desired trajectory, position in camera frame, and real position of virtual quadrotor. We test both the traditional PID controller and the proposed sliding mode controller on this platform. Real trajectory, desired trajectory and position in SVO frame in  $x-y$  plane are illustrated in Figure 6. Figure 7 depicts more detail evolution about the tracking result.

From the two figures above, we can draw a conclusion that quadrotor can get its position by SVO, and there is no GPS signal in the simulation. The control scheme is as follows: First, the quadrotor takes off when the remote controller sends a takeoff command, then a fixed thrust is given, soon the quadrotor flies in the air. When it goes to the height



**FIGURE 6.** Tracking result in ROS environment. (a) SMC tracking in  $x-y$  plane. (b) PID tracking in  $x-y$  plane. The green dashlines in both picture are the estimation positions of SVO. The red solid lines mean the desired trajectories and blue solid lines mean real positions in Gazebo simulator.



**FIGURE 7.** Tracking results in ROS environment. (a), (c), (e) are the tracking results with the proposed sliding-mode controller in  $x$ ,  $y$  and  $z$  directions, respectively. The tracking results of PID controller in  $x$ ,  $y$  and  $z$  directions are illustrated in (b), (d) and (f), respectively. The green dash-lines in pictures above are the estimation position of SVO. The red solid lines mean the desired trajectories and blue solid lines mean real position in Gazebo simulator.

of 0.5 meters, which is measured by barometer (laser may be a good choice as well), SVO system records the first two frames and starts bootstrapping. After the position in SVO frame is estimated, the quadrotor will be at the hover condition. Because the SVO we used here is a monocular vision odometry, the measurement of  $x$  and  $y$  is up-to-scale. Luckily,

as height information can be obtained from barometer, we can estimate the scale from it. Then we get the position data from SVO. The desired trajectory is a circle, the center of which is at  $[0 \ 0 \ 3]^T$ . When the tracking starts, quadrotor first moves towards the desired trajectory and will be captured by the desired trajectory. Here, real trajectory, desired trajectory and position in SVO frame in  $x - y$  plane are illustrated in Figure 6. The tracking result is shown in Figure 7.

When it is moving fast, many features will be lost, which may lead the localization to failure. Therefore, we slow down the speed of the quadrotor as compared to that in the simulation. It is clear that SVO is an effective visual odometry for positioning, eventually, for both controllers can achieve trajectory tracking. The most significant difference is that the sliding mode controller has less overshoot than PID controller does, which is similar to the first simulation. In every channel, the proposed sliding mode controller performs better performance than traditional PID controller.

## VIII. CONCLUSION

The monocular SVO algorithm is applied to achieve the precise and fast location of quadrotor. A Kalman filter is applied for state estimation and data fusion. An adaptive sliding mode controller is designed using the backstepping technique. Compared with the traditional PID controller, the proposed sliding mode controller in both MATLAB and Gazebo simulations show better performance.

The results obtained will be extended to a real quadrotor in future work. IMU information can be used in feature tracking and position estimation in order to improve the accuracy of feature position. Because of the slow velocity both in numerical simulation and Gazebo simulation, more aggressive motion can be considered in future work. Furthermore, visual formation based on this method will be studied in the future.

## ACKNOWLEDGMENT

I would like to express my gratitude to Ms. Dandan Zhang and Mr. Xin Jin.

## REFERENCES

- [1] D. Kennedy and S. P. Philbin, "The imperative need to develop guidelines to manage human versus machine intelligence," *Frontiers Eng. Manage.*, vol. 5, no. 2, pp. 182–194, 2018.
- [2] J. J. Shi, S. Zeng, and X. Meng, "Intelligent data analytics is here to change engineering management," *Frontiers Eng. Manage.*, vol. 4, no. 1, pp. 41–48, 2017.
- [3] X. Jin, B. W. Wah, X. Cheng, and Y. Wang, "Significance and challenges of big data research," *Big Data Res.*, vol. 2, no. 2, pp. 59–64, 2015.
- [4] Z. Zuo, "Adaptive trajectory tracking control design with command filtered compensation for a quadrotor," *J. Vib. Control*, vol. 19, no. 1, pp. 94–108, Jan. 2013.
- [5] D. Cabecinhas, R. Cunha, and C. Silvestre, "A globally stabilizing path following controller for rotorcraft with wind disturbance rejection," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 708–714, Mar. 2015.
- [6] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 50–66, Jan. 2010.
- [7] A. Roberts and A. Tayebi, "Adaptive position tracking of VTOL UAVs," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 129–142, Feb. 2011.
- [8] A. Abdessameud and F. Janabi-Sharifi, "Image-based tracking control of VTOL unmanned aerial vehicles," *Automatica*, vol. 53, pp. 111–119, Mar. 2015.
- [9] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.*, vol. 4, no. 11, pp. 2343–2355, Nov. 2010.
- [10] N. Wang, S.-F. Su, M. Han, and W.-H. Chen, "Backpropagating constraints-based trajectory tracking control of a quadrotor with constrained actuator dynamics and complex unknowns," *IEEE Trans. Syst., Man, Cybern. Syst.*, pp. 1–16, 2018.
- [11] N. Wang, S.-F. Su, J. Yin, Z. Zheng, and M. J. Er, "Global asymptotic model-free trajectory-independent tracking control of an uncertain marine vehicle: An adaptive universe-based fuzzy control approach," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1613–1625, Jun. 2018.
- [12] H. Liu, D. Li, Z. Zuo, and Y. Zhong, "Robust three-loop trajectory tracking control for quadrotors with multiple uncertainties," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2264–2274, Apr. 2016.
- [13] H. Liu, W. Zhao, Z. Zuo, and Y. Zhong, "Robust control for quadrotors with multiple time-varying uncertainties and delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1303–1312, Feb. 2017.
- [14] N. Wang, J.-C. Sun, M. Han, Z. Zheng, and M. J. Er, "Adaptive approximation-based regulation control for a class of uncertain nonlinear systems without feedback linearizability," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3747–3760, Aug. 2018.
- [15] N. Zhu, N.-M. Qi, and C.-M. Qin, "Adaptive sliding mode controller design for BTT missile based on backstepping control," *Control Decis.*, vol. 31, no. 3, pp. 769–773, 2010.
- [16] J. Li and D. M. Xu, "Adaptive sliding mode controller for nonlinear systems with mismatched uncertainties based on adaptive backstepping scheme," *Control Decis.*, vol. 14, no. 1, pp. 46–50, 1999.
- [17] A. Stansky, J. K. Hedrick, and P. P. Yip, "The use of sliding modes to simplify the backstepping control method," in *Proc. Amer. Control Conf.*, Albuquerque, NM, USA, Jun. 1997, pp. 1703–1708.
- [18] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Lincoln, MA, USA: Ganga-Jamuna Press, 2006.
- [19] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
- [20] P. Ondruška, P. Kohli, and S. Izadi, "MobileFusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 11, pp. 1251–1258, Nov. 2015.
- [21] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 15–22.
- [22] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A modular gazebo MAV simulator framework," in *Robot Operating System*, vol. 1, K. Anis, Ed. Cham, Switzerland: Springer, 2016, pp. 595–625.
- [23] N. P. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep. 2004, pp. 2149–2154.
- [24] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robot.*, vol. 33, no. 1, pp. 3–46, Jan. 2016.
- [25] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, May 2015.
- [26] W. Si, H. She, and Z. Wang, "Fuzzy PID controller for UAV tracking moving target," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 3023–3027.
- [27] Y. Wu, Y. Sui, and G. Wang, "Vision-based real-time aerial object localization and tracking for UAV sensing system," *IEEE Access*, vol. 5, pp. 23969–23978, 2017.
- [28] T. Hoang, E. Bayasgalan, Z. Wang, G. Tsechpenakis, and D. Panagou, "Vision-based target tracking and autonomous landing of a quadrotor on a ground vehicle," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5580–5585.
- [29] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 971–976.
- [30] K. Mohta et al., "Fast, autonomous flight in GPS-denied and cluttered environments," *J. Field Robot.*, vol. 35, no. 1, pp. 101–120, 2018.

- [31] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on  $SE(3)$ ," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Dec. 2010, pp. 5420–5425.
- [32] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, no. 2, pp. 28–93, Apr. 2009.
- [33] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.
- [34] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 932–945, Oct. 2008.
- [35] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *Image Vis. Comput.*, vol. 29, no. 7, pp. 434–441, 2011.
- [36] C. Luo, S. I. McClean, G. Parr, L. Teacy, and R. De Nardi, "UAV position estimation and collision avoidance using the extended Kalman filter," *IEEE Trans. Veh. Technol.*, vol. 62, no. 6, pp. 2749–2762, Jul. 2013.
- [37] D. Shi, X. Dai, X. Zhang, and Q. Quan, "A practical performance evaluation method for electric multicopters," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1337–1348, Jun. 2017.
- [38] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw. Conf.*, vol. 3, no. 2, 2009, p. 5.
- [39] The MathWorks Inc. *MathWorks—Maker of MATLAB and Simulink—MATLAB & Simulink*. Accessed: May 5, 2018. [Online]. Available: <http://www.mathworks.com/>



**CHUNPING WU** is currently pursuing the Ph.D. degree with the Mechanical and Power Engineering College, Shanghai Jiao Tong University, Shanghai, China. She is mainly engaged in transformation of research and technology achievement in industrial fields including intelligent manufacturing, industry Internet, artificial intelligence, Internet of Things, and automation control. With both specialized knowledge and business experience, she has been working in planning formulation, policy research, and standardization promotion of intelligent manufacturing and advancement of major projects in Shanghai Economic and Information Commission for years, making Shanghai a forefront of intelligent manufacturing across whole country.



**BINGFENG ZHAO** received the bachelor's degree from the East China University of Science and Technology, where he is currently pursuing the master's degree. His research interests are focused on applications in quadrotor formation control with vision information.



**YANG TANG** (M'11) received the B.S. and Ph.D. degrees in electrical engineering from Donghua University, Shanghai, China, in 2006 and 2010, respectively. From 2008 to 2010, he was a Research Associate with The Hong Kong Polytechnic University, Hong Kong. From 2011 to 2015, he was a Post-Doctoral Researcher with the Humboldt University of Berlin, Berlin, Germany, and also with the Potsdam Institute for Climate Impact Research, Potsdam, Germany. Since 2015, he has been a Professor with the East China University of Science and Technology, Shanghai. His current research interests include multi-agent systems/complex networks, cyber-physical systems, hybrid dynamical systems, and artificial intelligence and their applications. He was a recipient of the Alexander von Humboldt Fellowship and the ISI Highly Cited Researchers Award in Computer Science by Clarivate Analytics in 2017. He is an Associate Editor of *Scientific Reports*, the *Journal of the Franklin Institute*, *Neurocomputing*, the *Proceedings of the Institution of Mechanical Engineers, Part I-The Journal of Systems and Control Engineering*, and the *International Journal of Control Automation, and Systems*, and a Leading Guest Editor of the *Journal of the Franklin Institute*.



**WEI DU** (M'16) received the B.S. and M.S. degrees in electrical engineering from Donghua University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2016. He is currently an Associate Professor with the East China University of Science and Technology, Shanghai, China. His current research interests include evolutionary computation, especially differential evolution, evolutionary multi-objective optimization, robust evolutionary multi-objective optimization, and their applications.