Abstract—Input uncertainty, e.g., noise on the on-board camera and inertial measurement unit, in vision-based control of unmanned aerial vehicles (UAVs) is an inevitable problem. In order to handle input uncertainties as well as further analyze the interaction between the input and the antecedent fuzzy sets (FSs) of non-singleton fuzzy logic controllers (NSFLCs), an input uncertainty sensitivity enhanced NSFLC has been developed in robot operating system (ROS) using the C++ programming language. Based on recent advances in non-singleton inference, the centroid of the intersection of the input and antecedent FSs (Cen-NSFLC) is utilized to calculate the firing strength of each rule instead of the maximum of the intersection used in traditional NSFLC (Tra-NSFLC). An 8-shaped trajectory, consisting of straight and curved lines, is used for the real-time validation of the proposed controllers for a trajectory following problem. An accurate monocular keyframe-based visual-inertial simultaneous localization and mapping (SLAM) approach is used to estimate the position of the quadrotor UAV in GPS-denied unknown environments. The performance of the Cen-NSFLC is compared with a conventional proportional integral derivative (PID) controller, a singleton FLC (SFLC) and a Tra-NSFLC. All controllers are evaluated for different flight speeds, thus introducing different levels of uncertainty into the control problem. Visual-inertial SLAM-based real time quadrotor UAV flight tests demonstrate that not only does the Cen-NSFLC achieve the best control performance among the four controllers, but it also shows better control performance when compared to their singleton counterparts. Considering the bias in the use of model based controllers, e.g. PID, for the control of UAVs, this paper advocates an alternative method, namely Cen-NSFLCs, in uncertain working environments.

Index Terms—Fuzzy logic controller (FLC), unmanned aerial vehicle (UAV), non-singleton FLC (NSFLC), input uncertainty sensitivity enhanced NSFLC, monocular visual-inertial SLAM.

I. INTRODUCTION

RECENTLY, quadrotor unmanned aerial vehicles (UAVs) have been applied for a wide range of indoor and outdoor civilian applications, e.g., traffic surveillance [1], search and rescue [2], image velocimetry [3], orchard monitoring [4], and fast speed applications, push the working conditions towards the nonlinear region, resulting in more uncertainties and fast and aggressive maneuvers under uncertain and noisy working conditions. Fuzzy logic controllers (FLCs) have been extensively used for the control of nonlinear systems due to their capability of handling uncertainties and delivering adequate control without the requirement for the precise mathematical model of the system which is often either unavailable or highly time-consuming to obtain. Although there are several fuzzy control implementations for the navigation of UAVs [13]–[16], most of them are based on singleton FLCs (SFLCs), which focus on high-level navigation instead of exploring the effect of input uncertainty on the UAV control performance. On the other hand, it is reported in the literature that non-singleton FLCs (NSFLCs) give more promising results when compared to their singleton counterparts for nonlinear servo systems [17], chaotic time series prediction [18], and UAV control [19], where nonlinearities and uncertainties are more visible in the reconstruction [5], wildlife protection [6], perch and stare [7], forest management [8] and person following [9]. The main advantages of these quadrotor UAVs are their small size, low cost, vertical take-off and landing capability as well as their easy maintenance. However, demand for almost perfect design of flight controllers for such aerial vehicles in particular on the operation boundaries remains a challenging task due to several factors, such as inherent underactuation characteristics, coupled translation-rotation dynamics, gyroscopic moments, non-linear dynamic models, aerodynamic damping and onboard mechanical vibration as well as internal (e.g., lack of modeling and inaccuracy of onboard sensors) or external (e.g., illumination variations and blurred areas on onboard captured images) uncertainties. In the literature, conventional controllers, e.g., proportional-integral-derivative (PID) [10], linear quadratic regulator [11], and sliding-mode control [12], have been utilized for the control of quadrotor UAVs to achieve fully autonomous flights. When a linear approximation of the nonlinear dynamic model of the quadrotor UAV is employed to represent the system, lack of modeling might disrupt control performance. Furthermore, the assumption of having small attitude angles must be satisfied during the quadrotor UAV flights because of the same assumption during the linearization. However, certain applications, e.g. aggressive maneuvers and fast speed applications, push the working conditions towards the nonlinear region, resulting in more uncertainties in the control of the quadrotor UAV. Therefore, an advanced model-free control approach is required to improve the control performance and maneuverability of the quadrotor UAVs for fast and aggressive maneuvers under uncertain and noisy working conditions.

Changhong Fu, Andriy Sarabakha, Erdal Kayacan are with School of Mechanical and Aerospace Engineering, Nanyang Technological University (NTU), 50 Nanyang Avenue, Singapore 639798. (e-mail: changhongfu@ntu.edu.sg, andriy001@e.ntu.edu.sg, erdal@ntu.edu.sg).

Changhong Fu and Andriy Sarabakha are with ST Engineering-NTU Corp Laboratory, 50 Nanyang Avenue, Singapore 639798.

Christian Wagner, Robert John and Jonathan M. Garibaldi are with Lab for Uncertainty in Data and Decision Making (LUCID), School of Computer Science, University of Nottingham, Nottingham, United Kingdom. (e-mail: christian.wagner@nottingham.ac.uk, robert.john@nottingham.ac.uk, jon.garibaldi@nottingham.ac.uk).

Christian Wagner is with Institute of Computing and Cyber Systems, Michigan Technological University, Houghton, Michigan, USA.
system.

Although both SFLCs and NSFLCs use the same style of fuzzy rule base, inference engine and defuzzifier, there is a different fuzzifier in the NSFLC which treats the inputs as fuzzy sets (FSs) to deal with input uncertainties better. In this paper, we aim to explore the potential of our recently introduced non-singleton FLC (Cen-NSFLC) [20, 21], where the firing strength of each rule is calculated by using the centroid of the intersection between the input and antecedent fuzzy sets (FSs) rather than the maximum of their intersection applied in traditional NSFLCs (Tra-NSFLCs), to handle uncertainties better, thereby improving the trajectory tracking accuracy in GPS-denied unknown environment. Although the Tra-NSFLC is capable of handling uncertainties by capturing them from control inputs, it does not offer fine-grained uncertainty information tracking, i.e., it is not highly sensitive to the shape of the input of FSs, leading to significant loss of information in the intersection of the input and antecedent models. In [20], [21], the novel approach to NSFLCs has shown promising results in the problems of Mackey-Glass and Lorenz chaotic time-series predictions with different levels of injected noise. An earlier version of this paper presented in [19] conducted extensive quadrotor UAV flight tests in a Java-based simulation environment. Different fuzzifiers were employed for the NSFLCs, and different levels of noise were embedded as the inputs of the SFLCs, Tra-NSFLCs and Cen-NSFLCs to evaluate the hovering performances of the quadrotor UAVs. Additional works in this paper contain the implementation of all controllers in C++ within the robot operating system (ROS) in real time; substantial demonstrations; as well as detailed explanations and analysis for real quadrotor UAV flight experiments and real-world uncertainty affecting real world sensors. In addition, the conventional PID controller has been used to compare and contrast the control performances of the aforementioned FLCs.

Another motivation of this paper is to investigate whether the Cen-NSFLC can better cope with the input uncertainties from monocular keyframe-based visual-inertial simultaneous localization and mapping (SLAM). In the literature, motion capture systems are utilized as an external sensor to estimate the position of the quadrotor UAVs [22], but this approach is very expensive and works only in a limited indoor space.

A large number of onboard sensors are available for the quadrotor UAVs. The GPS device is well researched for outdoor tasks to navigate quadrotor UAVs [23], however the GPS signal is unreliable in urban canyons or dense forest, and it is completely lost in indoor environments. The laser range finder is applied as an alternative sensor to provide real-time robust vision-based position estimation for quadrotor UAVs [27]. However, the visual position estimation algorithms highly depend on feature tracking performances, i.e., inaccurate and uncertain position estimations will be generated under conditions of large illumination changes, fast rotations and translations and low feature detection.

To summarize, the main contributions of this study are:

• To the best of our knowledge, this is the first time that the Cen-NSFLC is implemented for any real-world control problem;
• To the best of our knowledge, the Cen-NSFLC is utilized to work with the monocular keyframe-based visual-inertial SLAM in the long-term navigation of quadrotor UAVs for the first time;
• The control performances of conventional PIDs, SFLCs, Tra-NSFLCs, and Cen-NSFLCs are compared in terms of their trajectory tracking accuracy in a real-time UAV application;
• Since different flight speeds generate different uncertainties in the control system in a visual-inertial SLAM application, different input uncertainty levels are explored with different flight speeds;
• Several real-time implementation guidelines are presented to design a NSFLC for the control of quadrotor UAVs.

The rest of this paper is organized as follows: Section II presents the FLC, the traditional NSFLC and the input uncertainty sensitivity enhanced NSFLC, i.e., the Cen-NSFLC. Section III introduces the dynamic model of quadrotor UAVs used in the real-time tests. In Section IV monocular keyframe-based visual-inertial SLAM is explained. Section V presents the real flight results. Finally, some conclusions are drawn from the study, and future directions are given in Section VI.

II. INPUT UNCERTAINTY SENSITIVITY ENHANCED NSFLC (CEN-NSFLC)

A. Fuzzy Logic Controller (FLC)

Figure 1 shows the general structure of a FLC [28], which includes three parts: (1) fuzzifier part (red block); (2) inference engine part (blue block); it combines fuzzified inputs with IF-THEN rules using a t-norm to derive the firing strength for each rule from the rule base (purple block); and (3) defuzzifier part (green block). The inference engine and defuzzifier parts are the same in both NSFLCs and SFLCs. However, the difference between SFLCs and NSFLCs is the handling of the crisp inputs in the fuzzifier part.

1) Singleton FLC: For the SFLC, its singleton fuzzifier maps a crisp input \( x \) into a fuzzy set \( X \) with support \( x' \), i.e.:

\[
\mu_X(x) = \begin{cases} 
1, & x = x' \\
0, & x \neq x'.
\end{cases}
\]

(1)
e.g., strength. Assume that the Tra-NSFLC contains only two rules, the input and antecedent FSs is utilized to calculate the firing strength. Each input is fuzzified as a fuzzy set (FS) by the fuzzifier (red dashed rectangle of Fig. 3). Figure 4 shows two different input FSs, i.e., $X_1$ and $X_2$, which are intersected with an antecedent $A_1$. Although the actual input FSs are different, the firing levels calculated by the Tra-NSFLC are the same in both cases, i.e., $\mu_{X_1}(e_{max}) = \mu_{X_2}(e_{max}) = a$. 

Finally, the output FS $Y$ is defuzzified, the defuzzification result is the output of the NSFLC, i.e., control command $\phi^*$. The input-output mapping is represented by:

$$\mu_Y(y) = \max[\mu_{Y_1}(y), \mu_{Y_2}(y)]$$

where,

$$\mu_{Y_1}(y) = \min[\mu_{C_1}(y), \min[\mu_{e_1}, \mu_{de_1}, \mu_{e_1}]]$$

$$\mu_{e_1} = \max[\mu_{X_1}(e) * \mu_{A_1}^1(e)]$$

$$\mu_{de_1} = \max[\mu_{X_{de}}(de) * \mu_{A_1}^1(de)]$$

$$\mu_{e_1}^f = \max[\mu_{X_{ef}}(\int e) * \mu_{A_1}^1(\int e)]$$

where $\mu_{X_1}(*) * \mu_{A_1}(*)$ is the intersection of $X_1$ and $A_1$. For $\mu_{Y_2}(y)$, the equations are similar.

### C. Input Uncertainty Sensitivity Enhanced NSFLC (Cen-NSFLC)

As introduced in subsection II-B for Tra-NSFLCs, the calculation of the firing strength is taking the maximum of the intersection of the input FS and antecedent FS, as shown in the red dashed rectangle of Fig. 3. Figure 4 shows two different input FSs, $X_1$ and $X_2$, which are intersected with an antecedent $A_1$. Although the actual input FSs are different, the firing levels calculated by the Tra-NSFLC are the same in both cases, i.e., $\mu_{X_1}(e_{max}) = \mu_{X_2}(e_{max}) = a$. 

In Fig. 3, the mapping between the inputs and output of the Tra-NSFLC is illustrated. Taking the $x$-position controller (shown in Fig. 5) for example, it has three different inputs: position error $e$, the integral of error $\int e$, i.e., accumulated past error, and the derivative of error $de$, i.e., predicted future error. Each input is fuzzified as a fuzzy set (FS) by the fuzzifier (red block in Fig. 1). The fuzzified result is shown in Fig. 3 as the red Gaussian distribution. The maximum of the intersection of the input and antecedent FSs is utilized to calculate the firing strength. Assume that the Tra-NSFLC contains only two rules, i.e.,

- IF $e$ is $A_1$ AND $de$ is $A_2$ THEN $y$ is $C_1$
- IF $e$ is $A_2$ AND $de$ is $A_3$ THEN $y$ is $C_2$

Each rule has specified an AND relationship between the mappings of the three input variables, the minimum of the three is used as the combined firing strength of each rule. Then the output of each rule is the consequent FS at the aforementioned combined firing level. For the two rules, the maximum of both output FSs is used as the output FS $Y$. 

1) Singleton FLC: For the SFLC, its non-singleton fuzzifier does not model input uncertainty. To better cope with noisy, imprecise input measurements, this work employs a non-singleton fuzzifier.

2) Non-Singleton FLC: For the NSFLC, its non-singleton fuzzifier, in our work, maps a crisp input $x$ into a Gaussian membership function, as shown in Fig. 1.

$$\mu_X(x) = \exp \left[ -\frac{(x-x')^2}{2\sigma_{F}^2} \right], \quad (2)$$

where $\sigma_F$ is the spread.
Thus, two different inputs or more specifically, inputs with a different associated uncertainty distribution has result in the same firing level, thereby obtaining the same output from the Tra-NSFLC. In [20, 21], the centroid of the intersection of an input and an antecedent is introduced to enhance the input uncertainty capture capability for the NSFLSs, i.e., Cen-NSFLSs. It is demonstrated for time-series prediction problems with promising results. In this work, a Cen-NSFLC is utilized in a discrete system.

The centroid of the intersection of an input $X_*$ and an antecedent $A_i^1$, i.e. centroid of $X_* \cap A_i^1$, the new input-output mapping is:

$$\mu_Y(y) = \max[\mu_Y^1(y), \mu_Y^2(y)]$$

(5)

where $n$ is the number of discretization levels ($n=100$ in our work) utilized in a discrete system.

The centroid of the intersection of an input $X_*$ and an antecedent $A_i^1$, i.e. centroid of $X_* \cap A_i^1$, the new input-output mapping is:

$$\mu_Y(y) = \max[\mu_Y^1(y), \mu_Y^2(y)]$$

where,

$$\mu_Y^1(y) = \min[\mu_{C^1}(y), \min[\mu_{A_{de}^1}(x_{cen}(X_e \cap A_i^1))]$$,

$$\mu_Y^2(y) = \mu_{X_e \cap A_{de}^1}(x_{cen}(X_e \cap A_i^1))$$,

$$\mu_Y^3(y) = \mu_{X_f \cap A_{de}^1}(x_{cen}(X_f \cap A_i^1))$$.

For $\mu_Y^2(y)$, the equations are similar. The above formulas represent the firing level of an antecedent is its membership degree at the centroid of the intersection with the input set.

Remark 3: We would like to emphasize the differences of the FLC, SFLC, Tra-NSFLC and Cen-NSFLC. The general structure of an FLC is shown in Fig. 1. According to the different types of fuzzifiers, i.e., singleton and non-singleton fuzzifiers, the FLC is divided into two types: singleton FLC (SFLC) and non-singleton FLC (NSFLC). In a traditional NSFLC (Tra-NSFLC), the maximum of the intersection of the input and antecedent fuzzy sets is utilized to calculate the firing strength. On the other hand, the firing strength is computed using the centroid of the intersection of the input and antecedent fuzzy sets in the Cen-NSFLC. In the literature, the Cen-NSFLC has so far not been applied for the long-term navigation of real quadrotor UAVs. Our Cen-NSFLC implementation is able to send the control commands to real quadrotor UAVs at 100 Hz, which is adequate for the control response during fast UAV trajectory following applications.
III. QUADROTOR UAV DYNAMICS AND CONTROL SCHEME

A. Quadrotor Dynamics

For describing the rigid body dynamics of the quadrotor UAV, two coordinate systems are employed: the inertial reference frame \( F_I = \{ \bar{x}_I, \bar{y}_I, \bar{z}_I \} \) and body-fixed reference frame \( F_B = \{ \bar{x}_B, \bar{y}_B, \bar{z}_B \} \). The origin of the body reference frame is located at the center of mass of the quadrotor UAV. The axes \( \bar{x}_B \) and \( \bar{y}_B \) lie in the plane defined by the centers of the four rotors and respectively point toward the right and forward of the quadrotor UAV, as shown in the red dashed rectangle of Fig. 5.

The control of translational and rotational motions of the quadrotor UAV are achieved by changing the thrust of each rotor \( f_i, i = 1, \ldots, 4 \), in various combinations. The thrust from an individual rotor is varied by changing its angular speed \( \omega_i \), \( i = 1, \ldots, 4 \). Then, the control vector \( c \) of the quadrotor UAV is considered as follows:

\[
e = \begin{bmatrix} T \tau_\phi \tau_\theta \tau_\psi \end{bmatrix}^T, \tag{6}
\]

where \( T \) is the total thrust, \( \tau_\phi, \tau_\theta \) and \( \tau_\psi \) are externally applied moments known as rolling, pitching and yawing moments, respectively. Under these considerations, the relation between \( e \) and \( \omega \), becomes [29]:

\[
\begin{align*}
T &= \begin{bmatrix} \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 \end{bmatrix} \\
\tau_\phi &= \frac{\sqrt{2}}{4b} l \left( \omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2 \right) \\
\tau_\theta &= \frac{\sqrt{2}}{4b} l \left( -\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2 \right) \\
\tau_\psi &= \frac{d}{\omega_1^2} \left( -\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2 \right),
\end{align*} \tag{7}
\]

where \( b \) is the propeller drag coefficient, \( d \) is the propeller pitch and \( l \) is the arm length of the quadrotor UAV.

The absolute position of a quadrotor UAV is described by the three Cartesian coordinates \( (x, y, z) \) of its center of mass in the inertial reference frame and its attitude by the three Euler angles \( (\phi, \theta, \psi) \). These three angles are respectively called roll, pitch, and yaw. The time derivative of the quadrotor UAV position \( (x, y, z) \) gives the absolute velocity of the quadrotor UAV’s center of mass expressed in \( F_I \), i.e., \( v = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T = \begin{bmatrix} u & v & w \end{bmatrix}^T \). Similarly, the time derivative of the attitude provides the angular velocity in \( F_I \), i.e., \( \omega = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \).\( \omega_B = [p \ q \ r]^T \) is the body angular rates. The quadrotor dynamical model is given below:

\[
\begin{align*}
\dot{x} &= u \\
\dot{y} &= v \\
\dot{z} &= w \\
\dot{\phi} &= p + s_\phi t_\theta q + c_\phi s_\theta r \\
\dot{\theta} &= c_\phi q - s_\phi r \\
\dot{\psi} &= \frac{s_\phi}{c_\phi} q + \frac{c_\phi}{c_\phi} r
\end{align*}
\]

where \( c_\phi, s_\phi \) and \( t_\phi \) denote \( \cos \phi, \sin \phi \) and \( \tan \phi \), \( m \) is the quadrotor mass, \( g \) is the gravity acceleration \( (g = 9.81 \text{m/s}^2) \), \( I_x, I_y, \) and \( I_z \) are moments of inertia, respectively.

Remark 4: The aforementioned quadrotor dynamic equations are coupled, nonlinear and the system to be controlled is underactuated. In addition, the input uncertainty, e.g., noise on the camera and inertial measurement unit (IMU), in vision-based control of UAV is an inevitable problem. All these features have motivated us to use a fuzzy logic controller, instead of a model-based linear controller, which is able to handle nonlinear systems with uncertainties.

B. Control Scheme

The overall structure of the closed-loop control of the quadrotor UAV is illustrated in Fig. 5. It mainly consists of three modules, i.e., the quadrotor UAV platform (Parrot AR.DRONE 2.0 Elite Edition, equipped with a front-looking monocular RGB camera, a 3-axis accelerometer and a 3-axis gyroscope. The quadrotor UAV publishes captured image frames at 30Hz with a resolution of 640×360 pixels, gyroscope measurements and the estimated horizontal velocity at 200 Hz), monocular keyframe-based visual-inertial SLAM module, and position controller.

Let \( \mathbf{p}^* = [x^*, y^*, z^*, \psi^*] \) be the desired position defined by the user end, \( \hat{\mathbf{p}} = [\hat{x}, \hat{y}, \hat{z}, \hat{\psi}] \) is the estimated position from the keyframe-based visual-inertial SLAM algorithm during the quadrotor UAV flight, the position error \( \mathbf{e}_p = \mathbf{p}^* - \hat{\mathbf{p}} = [x^* - \hat{x}, y^* - \hat{y}, z^* - \hat{z}, \psi^* - \hat{\psi}] \). The position controller computes
the desired control command $u^* = [\phi^*, \theta^*, v_z^*, v_x^*]^T$ for the quadrotor UAVs in order to reach the desired position $p^*$.

**Remark 5.** The roll angle, pitch angle, vertical velocity and yaw rotational speed in the desired control command are normalized to $[-1, 1]$, i.e., $\phi^*, \theta^*, v_z^*, v_x^* \in [-1, 1]$.

IV. MONOCULAR KEYFRAME-BASED VISUAL-INERTIAL SLAM

Recently, monocular keyframe-based visual simultaneous localization and mapping (SLAM) has become a key technology for different types of robots, especially for the UAVs, to estimate their positions. In the literature, the most representative monocular keyframe-based SLAM approach is feature-based parallel tracking and mapping (PTAM) [30]. It is the first work to present the idea of splitting visual tracking and mapping into parallel threads. It has been demonstrated to be successful in different real-time applications. In our work, an efficient local geometric filter [9], which effectively handles outlier feature correspondences based on a forward-backward pairwise dissimilarity measure $E$, is used to improve the visual feature tracking thread. The $E$ for every pair of feature correspondences, i.e., $c_i = (x_{i-1}^k, x_i^k)$ and $c_j = (x_{j-1}^k, x_j^k)$, $i \neq j$, is defined as below:

$$E(c_i^k, c_j^k) = \frac{1}{2} \left[ E(c_i^k, c_j^k|H_k) + E(c_j^k, c_i^k|H_k^{-1}) \right], \quad (9)$$

where,

$$E(c_i^k, c_j^k|H_k) = \left\| (x_i^k - x_j^k) - H_k(x_{i-1}^k - x_{j-1}^k) \right\|,$n

$$E(c_j^k, c_i^k|H_k^{-1}) = \left\| (x_i^k - x_j^k) - H_k^{-1}(x_j^k - x_i^k) \right\|.$$

where $\|\|$ is the Euclidean distance, $x_i^k$ is the $i$th feature location on the $k$th image frame, $H_k$ is a homography transformation estimated by the feature correspondences between image frame $I_{k-1}$ and $I_k$, and $H_k^{-1}$ is the inversion of this homography transformation. Then a hierarchical agglomerative clustering approach [32] is utilized to filter out outlier correspondences based on an effective single-link approach with the forward-backward pairwise dissimilarity measure $E$, thereby reducing the ambiguity correspondences and filter the erroneous correspondences. Figure 6 shows some onboard captured images with feature (green point) tracking results.

The SLAM algorithm used in this study belongs to the feature-based SLAM approach. In practice, the visual SLAM algorithm highly depends on feature tracking performance. Specifically, inaccurate and uncertain position estimations will be generated under the condition of larger illumination changes, faster rotations and translations, and fewer feature detection. Moreover, monocular vision cannot determine the real scale of the environment and camera motion alone, which is essential in robot control. In this work, the onboard inertial measurement unit is used as the proprioceptive device for resolving the scale ambiguity to achieve the monocular keyframe-based visual-inertial SLAM.

**Remark 6.** This is the first work to implement the monocular keyframe-based visual-inertial SLAM for working with the Cen-NSFLCs. All degrees-of-freedom have been controlled in real-world UAV flight experiments.
A. Monocular Keyframe-based Visual-Inertial SLAM Performance

Since the focus of this paper is the evaluation of different levels of input uncertainty affecting the control inputs, we review the monocular keyframe-based visual-inertial SLAM (and its properties) as the key input generating technique. The relationship between the flight speed and uncertainty level is shown in Fig. 7a. To evaluate the SLAM performance, the root mean squared error (RMSE) between the ground truth (G) and the SLAM estimation (E) is used. Figure 7a shows the average SLAM performance results with different UAV flight speeds.

As can be seen from Fig. 7a, the SLAM algorithm obtains the best position estimation result during the UAV hovering flight tests. As the UAV flight speed is increasing, the position estimation accuracy is decreasing. The average RMSEGEs of Tests 2, 3 and 4 have increased 0.139m, 0.16m and 0.199m compared to the UAV hovering flight tests. Figure 7a also shows that increased flight speed results in higher position input uncertainty. Similarly, experiments showed that variation amount in illumination, rotation and translation speeds, reduction in detected features - are all positively correlated to increasing uncertainty/noise levels in the position estimation inputs. For example, the quadrotor UAV during the hovering flights always looks at the same scene, i.e., same illumination and number of detected features, without capturing the blurred image frames.

Figure 7b, 7c and 7d shows x, y and z translation estimations (red color) of two rounds of the trajectory following application which is controlled by the Cen-NSFLC with a maximum flight speed of 2m/s. The ground truths (black color) of the x, y and z translations from the OptiTrack system are used for performance comparisons. As can be seen from Fig. 7b, 7c and 7d, although faster flights result in more challenging pose estimations in the monocular keyframe-based visual-inertial SLAM, the pose estimations can match the ground truths fairly well. Therefore, the SLAM estimations are accurate enough to be used as the control inputs in the long-term navigation of the real-world quadrotor UAV. Moreover, the quadrotor UAV locations labeled in Fig. 7b, 7c and 7d are also shown with external views in Fig. 8.

Remark 7: In order to elaborate the performance of the presented FLCs in this paper, different levels of input uncertainty conditions have been generated due to different UAV flight speeds. Specifically, larger UAV flight speeds will result in faster rotations and translations, fewer feature detection as well as bigger illumination changes in SLAM. Due to the fact that the SLAM algorithm highly depends on the feature tracking performance, the challenging condition of faster rotations will result in inaccurate and uncertain position estimations. In other words, the higher speed for the UAV, the more uncertainties in the localization, therefore more uncertainties in the input of the presented FLCs.

B. Control Performance

The control performances of the PID, SFLC, Tra-NSFLC and Cen-NSFLC are evaluated based on the RMSE between the ground truth (G) and the reference trajectory (R), i.e., RMSEGE. Fig. 9 shows the control performance results, i.e., average RMSEGEs calculated from one hundred flight tests.

From Fig. 9, it can be observed that the Cen-NSFLC consistently obtains the best performance across all speed levels. The control performances of the FLCs are better than those of the conventional PID controller, while both NSFLCs can obtain superior control performance compared to the SFLC, and the Cen-NSFLC outperforms the Tra-NSFLC.

Figure 10 shows the control performances of all the controllers in one round of the trajectory following application with the maximum flight speed of 2m/s. As can be seen from these three figures, all the controllers can navigate the quadrotor UAVs to follow the online generated trajectory, but the control performance ranking is the Cen-NSFLC, Tra-NSFLC, SFLC, and PID controller. Although the Euclidean errors of the SFLC and Tra-NSFLC in some parts of trajectory are less than the one of the Cen-NSFLC, the overall control performance of the Cen-NSFLC is better than the ones of the
Remark 8: The main aim of this study is to elaborate input uncertainty dealing capability of different FLCs. For this goal, the following strategy is followed: It is a well-known fact that, in SLAM, different UAV flight speeds result in different inaccurate and uncertain position estimations. Therefore, different levels of uncertainties are sent to the inputs of the presented FLCs. In addition to test handling uncertainties capability in the FLC/NSFLC, several types of maneuverable flights, i.e., ascending and descending straight lines and curves, have been defined to evaluate the robustness of the controller. In real-time UAV application, in which UAV trajectories are generated online, when a PID controller is tuned with respect to curve line trajectories, it provides oscillatory responses for straight lines. On the other hand, a PID controller tuned with respect to straight line trajectories provides larger steady-state error for curve lines. We would like to emphasize that the trial-and-error approach has been used to achieve as close as to the optimal performance for all types of trajectories, i.e., straight and curve line trajectories and also for different speed values for PID controller. Unlike PID, the presented FLCs are nonlinear controllers, and they provide better performances in particular for uncertain working environments.

VI. Conclusions and Future Work

This study explores the real-world real-time trajectory following problem of quadrotor UAVs under different levels of input uncertainty. A comprehensive evaluation of real UAV control performance has been conducted in the context of varying flight speeds with monocular keyframe-based visual-inertial SLAM as the primary navigation input source. Overall, four individual controllers were compared: the conventional PID controller and three different types of FLCs, i.e., a SFLC and two NSFLCs: Tra-NSFLC and the novel Cen-NSFLC, also providing the first real-world application of the Cen-NSFLC framework. The key objective of this work was not to identify the best control performance possible, but to compare the relative performance of the different controllers under different levels of input uncertainty from the real-world onboard sensors, i.e., the camera and IMU. The flight experiments conducted in this paper show that the control performances of the FLCs are better than those of the conventional PID controller, that the NSFLCs can obtain superior control performance (exhibiting better noise rejection) compared to the SFLC, and the Cen-NSFLC outperforms the Tra-NSFLC, especially at the higher flight speeds. Importantly, significant effort was made to allow for comparable levels of controller design effort for all controllers, i.e. while all controllers could be tuned further, we are confident we have been able to provide a fair basis for their comparison based on the parameters selected. It is also worth noting that the input uncertainty model employed for the NSFLCs (a Gaussian distribution associated with a crisp input) is a simplistic model which for example does not capture manufacture-published uncertainty levels of the sensors. As the best-performing Cen-NSFLC is designed to extract maximum information form the input uncertainty model, we expect that even small amounts of refinement in the input uncertainty models will result in further improved performance.

As part of future work, we will focus on exploring the possibility of making the presented NSFLCs adaptive while using them for navigating quadrotor UAV. Adaptive FLCs may help to improve the trajectory following accuracy when an external performance measure is available \([34]\), e.g., after the quadrotor UAV starts to conduct the second circular line turn, thereby obtaining the minimum Euclidean errors throughout the whole trajectory. On the other hand, improving the input uncertainty model may in itself result in improved performance as alluded to above. Finally, we are planning to apply the outcomes of this work to the design of type-2 fuzzy controllers.
to explore if type-2 NSFLCs can deliver further improved performance.

ACKNOWLEDGMENT

The research was partially supported by the ST Engineering - NTU Corporate Lab through the NRF corporate lab@university scheme. This work was also partially funded by the RCUK’s EP/M02315X/1 From Human Data to Personal Experience grant.

REFERENCES


