Multirotor Motion Enhancement using Propeller Speed Measurements

Heba Abdelnasser[1], Mohammad Heggo[1], Oscar Pang[1], Mirko Kovac[1,2], and Julie A. McCann[1]

[1]Imperial College London, United Kingdom
[2]Laboratory of Sustainability Robotics, Swiss Federal Laboratories for Materials Science and Technology (EMPA), Switzerland
{*h.awad19,m.heggo,k.pang,m.kovac,j.mccann*}@*imperial.ac.uk*

Abstract— **Multirotor autopilots often depend on open-loop control without the feedback of propeller speeds, although they are a critical factor in determining motion characteristics. This paper proposes a system that leverages actual propeller speeds as direct feedback to the autopilot to improve the state estimation and dynamics of the multirotor. Software-in-the-Loop (SITL) and Hardware-in-the-Loop (HITL) simulations with real data, in different scenarios, are conducted to demonstrate the impact of combining propeller speeds with typical drone sensors. The results show that the drone becomes more stable with lower trajectory errors. Further, a noticeable reduction in the vehicle position median error while following a trajectory is shown, and a considerable increase in the flying duration time before crashing in case of a motor fault. These results highlight the potential of adding propeller speed feedback to increase the autopilot's controllability which enhances drone performance in sensitive applications.**

I. Introduction

Multirotors are a popular category of flying vehicles; they are the dominant form factor for Unmanned Air Vehicles (UAVs) due to their robustness and simple structure, which can be implemented using low-cost and lightweight components. Multirotors have various indoor and outdoor applications where they can move in challenging or even impossible locations for humans to reach. These include search and rescue missions for victims, traffic surveillance, agriculture monitoring and soil analysis, infrastructure health and cracks inspection, cinematography, and warehouse work [1]. Their increased application, especially in urban or human-populated areas like workplaces, requires increased safety and stability restrictions.

In a multirotor autopilot, the position and attitude are typically adjusted through the rotors' angular speed control inputs which have a fundamental role in determining the motion properties. However, the autopilot usually estimates the vehicle state relative to its surroundings, using onboard sensors, without direct feedback on the rotors' behaviour. In fact, actual propeller rotation speed can be deviated from the reference speed by different environmental, mechanical and electrical parameters including wind velocity and direction, battery voltage, and internal motor resistance and inductance which makes it more challenging for the autopilot to match the actual and desired positions of the multicopter.

In this work, using a quadcopter model, direct feedback of the propellers' actual speed is used to make the motion more stable and safer even under adverse conditions. The quadcopter propellers' actual speeds are fed back to autopilot state estimation and control to enhance the quadcopter motion. In state estimation, the position and orientation are estimated using the quadcopter dynamic model with the measured propeller speeds as inputs. An Extended Kalman Filter (EKF) [2] is then applied to fuse the estimated position and orientation with the inertial quadcopter sensors' measurements. The measured propeller speeds are used to improve the autopilot's total torque and thrust control by embedding the propellers' speed errors to corresponding thrust and/or torque directions in the autopilot's controller stack.

The rest of the paper is organized as follows. Section II presents an overview of the quadcopter dynamics while Section III introduces the proposed system. Section IV evaluates the proposed system. A discussion and directions for future work are proposed in Section V. Section VI concludes the paper.

II. Quadrotor's Dynamic Model

Before we describe our system architecture, we present the fundamental quadrotor dynamics. Usually two frames are defined for the quadcopter structure model: inertial and body frames (Figure 1). The inertial frame is defined by the Earth's axes, where the negative direction of z-axis is the direction of gravity. The body frame is defined by the quadcopter body orientation, where the z-axis is the rotors' thrust direction. The quadcopter's dynamic transition [3] can be represented by the following equation:

$$m\ddot{p}_I = mg + R_{IB}F_B + F_{a,I} \tag{1}$$

where $p_I = [x, y, z]^T$ is the position of the quadcopter in the inertial frame. $g = [0, 0, g_e]^T$ is the gravity vector where g_e is the gravitational constant. $F_B = [0, 0, F]^T$ is the external force vector in the body frame. R_{IB} is the rotation matrix that allows transformation from body to inertial frame to represent the quadcopter attitude. $F_{a,I}$ is the aerodynamic force modeled as first order drag. Thus, the required force to move a quadcopter of mass m with acceleration $\ddot{p}_I$ in the inertial frame, is equal to the gravity force mg added to the inertial frame total thrust of the rotors $R_{IB}F_B$ and the aerodynamic force $F_{a,I}$. The dynamic equation denoting quadcopter rotation can be described with the external torque on the vehicle in the body frame:

$$I\dot{\omega}_B + \omega_B \times I\omega_B = M_B + M_{a,B} \tag{2}$$

where $\omega_B = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity of the quadcopter body frame with respect to the inertial frame. I is the quadcopter inertia. $M_B = [\tau_x, \tau_y, \tau_z]^T$ is the total torque

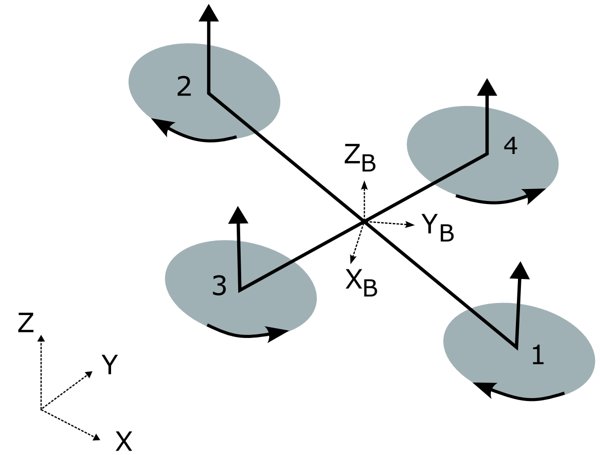


Fig. 1: Diagram of the quadrotor model illustrating the world and body frames.

in the body frame. $M_{a,B}$ is first order drag aerodynamic moments.

The total thrust F and torques $M_B = [\tau_x, \tau_y, \tau_z]$ are generated by the control model via control input to the four rotors.

$$[F, \tau_x, \tau_y, \tau_z] = Hu \tag{3}$$

where $u \in \mathbb{R}^4$ represents thrust generated by each spinning rotor. $H \in \mathbb{R}^{4\times 4}$ is a projection matrix that maps total thrust and moment forces to thrust generated by each rotor.

III. System

In this section, we introduce the proposed system modules. Propeller feedback pose tracker module tracks the quadcopter position and orientation using the propellers speeds readings. State estimation module fuses propellers speeds tracked pose with the inertial sensors estimation using an EKF to estimate the quadcopter state. Enhanced quadcopter control module fuses propeller speeds feedback in the controller to acquire the desired force and torque of the quadcopter.

A. Propeller Feedback Pose Tracker

The tracker algorithm estimates linear and angular vehicle acceleration after measuring the speeds of the four propellers using the dynamic model of the quadcopter. The actual total thrust and torques are first calculated, which are then used to obtain the linear and angular acceleration. The vehicle's four propellers produce four thrust forces $f_{1..4}$ that pull it upward. f_i is related to the i^{th} propeller spinning speed as $f_i = f_{max}|\omega^i|$, where $|\omega^i|$ is the normalized spinning speed of rotor i and f_{max} is the maximum force delivered by one propeller when the motor is running at full speed. The total thrust applied on the body frame is $F_B = [0, 0, -\sum_{i=1}^{4} f_i]^T$.

The actual torque applied on quadcopter's body can be derived from the four propeller spinning speeds as follows,

$$M_B = \begin{bmatrix} l(-f_1 + f_2 + f_3 - f_4) \\ l(f_1 - f_2 + f_3 - f_4) \\ \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{bmatrix} \tag{4}$$

where τ_i is the torque of rotor i and is related to the propeller spinning speed as $\tau_i = \tau_{max}|\omega^i|$ and l is the distance between the rotor and the center of mass of the quadcopter. τ_{max} is the maximum torque delivered by one propeller when the motor is running at full speed.

The estimated thrust F_B and torque M_B are used to track the linear and angular acceleration of the quadcopter using rearranged forms of the Equations 1 and 2. The motion equations in the discrete integration form can be described as follows,

$$p_I^{t+1} = p_I^t + v_I^t \Delta t \tag{5}$$

$$v_I^{t+1} = v_I^t + (g + \frac{1}{m} R_{IB}(q) F_B^t + \frac{1}{m} F_{a,I}^t)\Delta t \tag{6}$$

$$q^{t+1} = q^t + \frac{1}{2} q^t \otimes \omega_B^t \Delta t \tag{7}$$

$$\omega_B^{t+1} = \omega_B^t + I^{-1}(M_B^t + M_{a,B}^t - \omega_B^t \times I\omega_B^t)\Delta t \tag{8}$$

where q is the quaternion that represents the vehicle attitude. $R_{IB}(q)$ is the rotation matrix that is derived from q that allows transformation from body to inertial frame.

B. State Estimation Design

The state estimator provides the position and orientation of the quadcopter using the propellers pose tracker with the onboard inertial sensors. This is implemented using an EKF scheme that depends on update step using input signal and observations. General kinematics equations are usually more convenient to be used for the update step. These equations are for any rigid body moving in the inertial frame 3D space and does not depend on the characteristics of the body. Thus, we depend on the general rigid body equations updated with inertial sensors measurements. The tracked position and orientation from the propellers speed measurements are used as an observation. Pose information (quaternion, velocity and position) is captured in the first 10 states of the EKF $x = [q_0, q_1, q_2, q_3, V_x, V_y, V_z, P_x, P_y, P_z]^T$, where the quaternion defines the angular position of the body frame relative to the navigation frame. Using inertial sensors measurements, the pose is updated as follows,

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t+1} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}_t \begin{bmatrix} \Delta q_0 & \Delta q_1 & \Delta q_2 & \Delta q_3 \\ -\Delta q_1 & \Delta q_0 & -\Delta q_3 & \Delta q_2 \\ -\Delta q_2 & \Delta q_3 & \Delta q_0 & -\Delta q_1 \\ -\Delta q_3 & -\Delta q_2 & \Delta q_1 & \Delta q_0 \end{bmatrix} \tag{9}$$

$$v_{t+1} = v_t + R_{IB}(q_t)\Delta v + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \Delta t \tag{10}$$

$$p_{t+1} = p_t + v_t \Delta t \tag{11}$$

where $\Delta q_{0..4}$ is the rotation from quaternion at time frame t to $t+1$ that is derived from IMU angle shift measurements. $R_{IB}(q_t)$ is the rotation matrix from body to inertial frame derived from quaternions at time t. ΔV is the change in IMU velocity measurements.

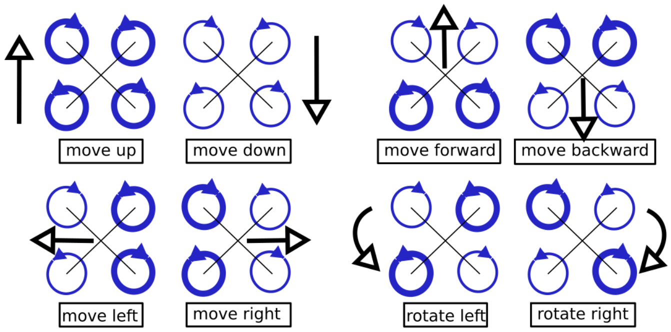


Fig. 2: Relation between quadocpter different movements and propellers spinning direction.

C. Enhanced Quadcopter Control

The speed measurements are further used in autopilot control. Autopilot control is module that aims to adjust the pose of the UAV to match the setpoint. Usually the quadcopter autopilot control is a hierarchical mix of P and PID controllers with mainly two loops [4]. The outer loop is for position control that takes estimated and desired positions as inputs. The output is the desired attitude and commanded thrust that are required to move the vehicle to the desired position. The internal loop is for controlling the attitude that takes the estimated and desired attitude as inputs and outputs the desired torques required for the desired attitude. Besides the estimated pose feedback, we add a feedback on the actual propeller speed measurements. The difference between commanded and actual propellers speeds defines the propeller speed error is represented as follows,

$$\omega_e = \begin{bmatrix} \omega_e^x \\ \omega_e^y \\ \omega_e^z \\ \omega_e^F \end{bmatrix} = \mathbb{M} \begin{bmatrix} \omega_{com_1} - \omega_{act_1} \\ \omega_{com_2} - \omega_{act_2} \\ \omega_{com_3} - \omega_{act_3} \\ \omega_{com_4} - \omega_{act_4} \end{bmatrix} \tag{12}$$

where ω_{com_i} is the commanded speed of propeller i and ω_{act_i} is the actual speed of propeller i. ω_e^x, ω_e^y, ω_e^z and ω_e^F represent the error between the actual and commanded speeds that affect the roll, pitch and yaw torques, and thrust, respectively. $\mathbb{M}$ is a mapping matrix that maps the propeller speed errors to the torques and thrust forces directions (Figure 2 shows the relation between quadocpter movement and generated torques and force which can be translated to $\mathbb{M}$) and is defined as follows,

$$\mathbb{M} = \begin{bmatrix} -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & +1 & +1 & +1 \end{bmatrix} \tag{13}$$

The position controller originally adjusts the total thrust based on the error between the estimated and desired positions, while the attitude controller compensates the torque based on the error between the estimated and desired rotations. Figure 3 shows the quadcopter controls after adding proportional controller for propeller speeds error feedback shown in equation 12.

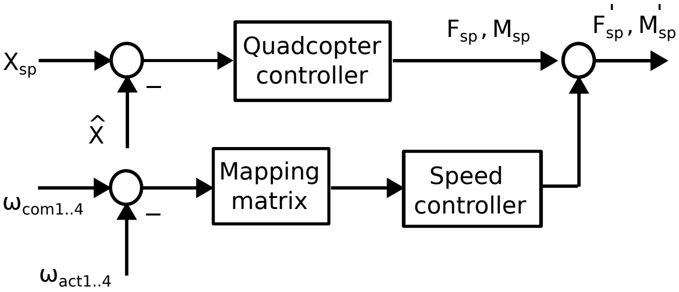


Fig. 3: Quadcopter enhanced control with a feedback of actual propeller speeds.

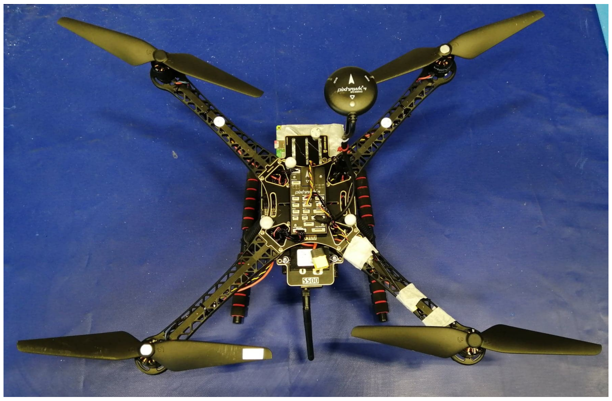


Fig. 4: Holybro S500, the used quadcopter in the HITL setup.

IV. Evaluation

The proposed system is evaluated using SITL and HITL simulations in different operating scenarios under normal flight and rotor failure situations. The SITL simulations are executed on the Gazebo simulator that supports sensor and motor simulation. The adopted autopilot code is the powerful open source PX4 autopilot system[1]. The HITL simulations are performed using Holybro S500 quadcopter (Figure 4) with Pixhawk 4 autopilot running PX4 firmware. It is equipped with four 2216-980KV brushless motor and 9-inch propellers. A Tachometer is used to measure the propeller speed.

A. SITL Simulation

To emulate real-world quadcopter environments, Gazebo wind and magnetic disturbances are added to the simulation model.

1) Trajectory tracking performance: The autopilot performance is evaluated with the two proposed modules as per Section III. Figure 6 shows three examples of testing trajectories at fixed altitude in the simulation that are used to verify the impact of adding the actual propeller speed feedback to the autopilot. It can be noticed that there is a considerable improvement in the quadcopter motion accuracy while following the predefined trajectories. This can be further elucidated in Figure 7 showing the CDFs of the quadcopter position and orientation errors after following

[1] https://px4.io/

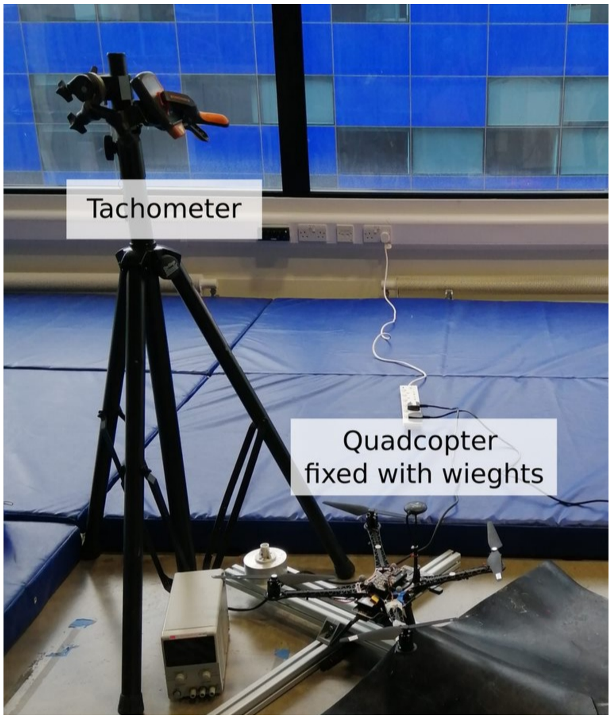


Fig. 5: Fixed quadcopter HITL scenario setup.

the trajectories in Figure 6 successive times. With typical quadcopter autopilot sensors, the median position error is $0.7m$ with a standard deviation of 0.9 m. The median position error decreases to 0.1 m with a standard deviation of 0.13 m when the propeller speed measurement is added to the autopilot state estimation and control stack. The median attitude quaternion distance error using typical quadcopter sensors is 2 with a standard deviation of 26.4 which is significantly reduced to 0.85 with a standard deviation of 3 after adding the propeller rotation speed measurements as the direct propeller speed feedback improves the quadcopter stability that explained with Figure 8. Figure 8 shows the impact of the propeller speed fusion on the roll and pitch angles when the quadcopter follows its trajectory 6b. It can be noticed that the quadcopter moves with lower tilt after adding the speeds feedback, which enhances its flying stability.

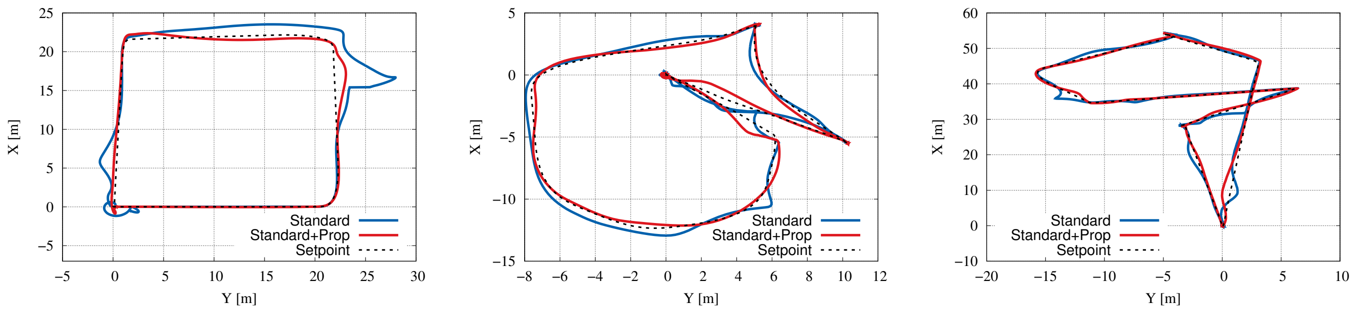
2) Rotor failure time tolerance: In this evaluation scenario, the quadcopter survival ability is studied in the presence of motor fault [5] to show the impact of the propeller speed feedback fusion, compared to the case where the quadcopter relies only on its other onboard sensors. The longer the quadcopter can maintain control of its attitude under motor fault, the higher the probability it can land safely or regain normal attitude, which is useful when the external disturbances to motor is temporary. This is examined as shown in Figure 9, where in Figure 9a, the quadcopter follows a trajectory where a 50% fault is forced on two opposite rotors for 6 seconds. This fault is applied at 30 seconds time instance after the quadcopter starts its mission. Note that the provided propeller speed measurements feedback method enables the quadcopter to safely complete its mission avoiding a possible crash that happens when the quadcopter just relies on its onboard IMU measurements. This is intuitive as the quadcopter control has a faster response to the motor failure when immediate feedback regarding the lack of motor speed is provided. Figure 9b shows the quadcopter's body roll and pitch angles while following the trajectory shown in Figure 9a. The quadcopter is more stable during the motor failure time (shaded area in Figure 9b) with the propeller speed feedback. Figure 9c shows the relationship between the percentage of applied motor failure and the time to crash duration. Generally, the time to crash decreases with increased motor failure percentage. Importantly, for the same percentage of motor failure, the time to crash is longer when the propeller speed feedback is added to the autopilot. It is worth mentioning that when applying a total failure to all motors, the quadcopter crashes within the same 3.5 seconds time in both cases as keeping balance in this case is challengy which needs especial model to handle [6].

B. HITL Simulations

We use speed measurements during a flight mission while the quadcopter is in a fixed position (Figure 5) to validate the impact of the propeller speed's direct feedback fusion with real-world flight data. The flight controller hardware is installed as per an updated HITL setup, so the tachometer reads real spinning propellers while the inertial sensors' readings are acquired from the Gazebo simulator. Figure 10 shows improvement impact of using propeller actual speed measurements in the state estimation and control of the autopilot on a mission trajectory. Figure 11 shows trajectories after a fault is deliberately applied on two opposite rotors after 15 seconds of taking off. Propeller speed feedback fusion enables the quadcopter to safely landing after complete its mission.

V. Discussion and Outlook

The results demonstrate the potential of using propeller speed as direct feedback in the multirotor's autopilot to enhance motion accuracy and stability. The next step is to investigate a suitable technique that can monitor propeller speeds while flying. From the literature, several approaches have been proposed to monitor industrial motors' health and detect their failure by measuring the rotor speed, which can be categorized as contact and contactless techniques. Contact techniques such as Tachometers [7] and Tagbeat [8] can measure the rotational speed by attaching a reflective tape on the rotating target object. Yet, they can only work in line-of-sight scenarios and are sensitive to environmental conditions. On the other hand, contactless techniques such as acoustic-based [9], laser-based [10], and magnetic-based [11] systems work without attaching any device to the rotating target object. However, they are required to be close (e.g., a few centimeters) to the rotating target or again be in the line-of-sight of the target. Recently, the low-cost/power RF-sensing technology has been used to provide long-distance, contact-free rotation speed measurements [12], [13].

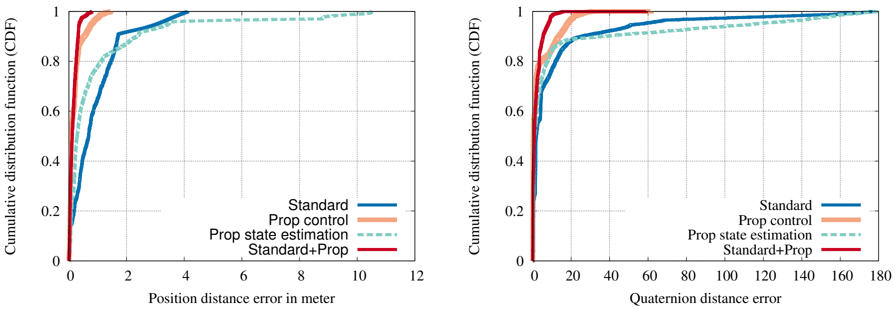


(a)

(b)

(c)

Fig. 6: Sample trajectories used to test the impact of adding actual propeller rotation speed to the autopilot. Typical used inertial sensors in PX4 are labeled as ’Standard’, while ’Standard+Prop’ denotes the case when propellers speeds feedback is added.



(a) Position

(b) Attitude

Fig. 7: The CDF of position and attitude error using typical quadcopter sensors with PX4 autopilot, and after fusing propeller speed feedback in the state estimation (Prop state estimation) and autopilot control stack (Prop control). The impact of adding the combination of two modules is labeled as Standard+Prop.

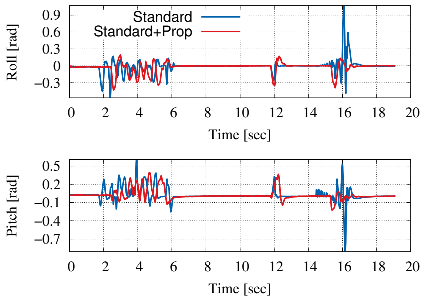
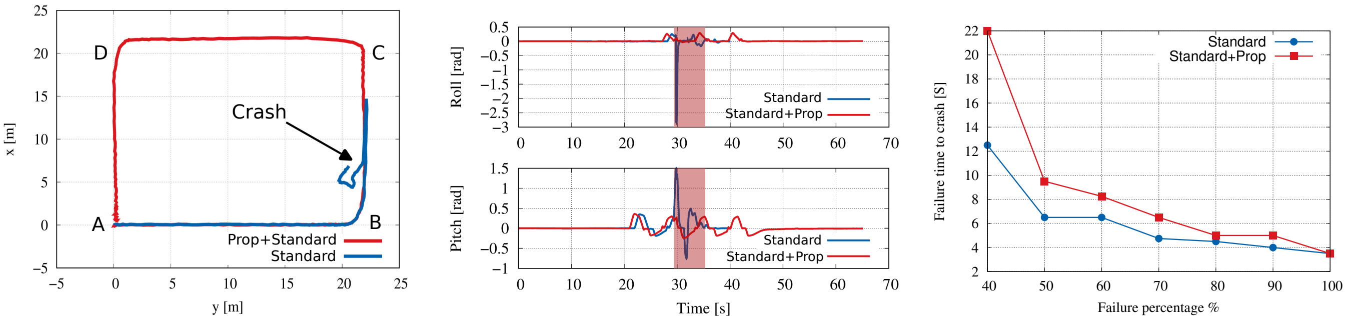


Fig. 8: Roll and pitch angles estimation over time with standard quadcopter sensors and after combining the speeds feedback.



(a) 2D trajectories of the quadcopter with applied fault in two rotors for around 6 seconds between B and C. A to D represent setpoints.

(b) Roll and pitch angle estimates of trajectories in (a). The highlighted area is during applying failure.

(c) The relation between failure percentage of two rotors and failure time to crash.

Fig. 9: The impact of propeller speed feedback fusion on the quadcopter performance with two opposite rotors under failure.

We are using our previously proposed system [13] to investigate the RF-sensing capabilities in monitoring the propeller speed of the quadcopter in Figure 4. Figure 12 shows the commanded PWM versus the actual RPM measurements using the RF and Tachometer sensors. The commanded PWMs are varied by changing the throttle level of a handheld radio remote control. The commanded PWM values for the range of 0 to 100 percent throttle level on the remote control are from 984 to 2006, which covers RPM range from 2000 to 16000. The figure demonstrates that RF-sensing can accurately measure the propeller speeds within a wide speed range which highlights its ability to achieve accurate propeller speeds monitoring.

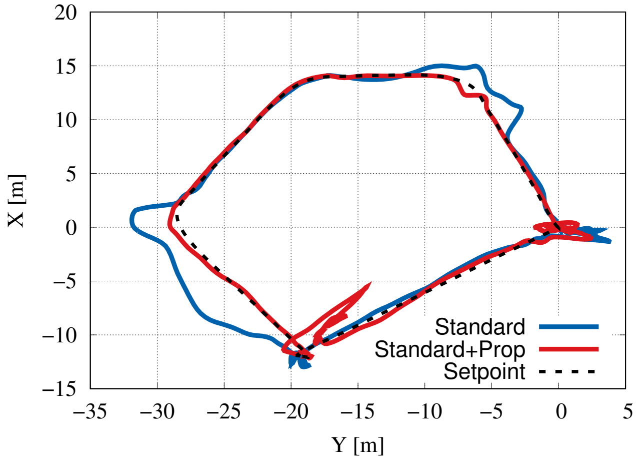


Fig. 10: Mission flight to validate the impact of the propellers speeds feedback on the motion while following a trajectory.

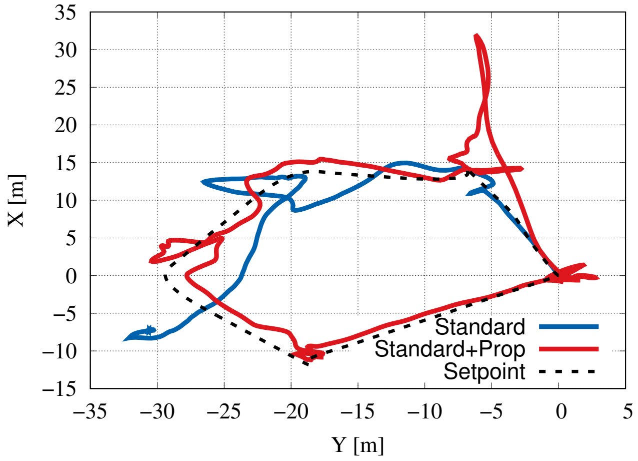


Fig. 11: Mission flight to validate the impact of the propellers speeds feedback in case of rotors partial failure.

VI. Conclusion

In this paper, we propose a system that leverages actual propeller speed measurements to enhance quadcopter motion accuracy. The propeller speeds are used to track the quadcopter's position and orientation through its dynamic model. The measured position and orientation are then fed to the state estimation module to be fused with the other typical quadcopter sensors measurements. The speed measurements are also used as feedback to the autopilot control system to compensate for the torque and thrust error caused by propeller speed drift.

SITL and HITL simulations have been conducted to evaluate the proposed technique in different situations. The results show that adding the actual propeller speed to the autopilot state estimation and control considerably improves quadcopter motion accuracy and stability during motor faults, which is mandatory for sensitive applications.

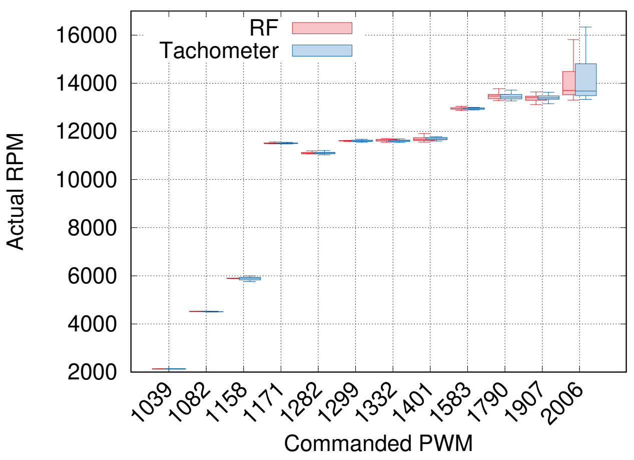


Fig. 12: Commanded PWM versus actual RPM measured by Tachometer and RF while controlling the quadcopter.

References

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, 2017.

[2] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, p. 46, 2004.

[3] Q. Quan, *Introduction to multicopter design and control*. Springer, 2017.

[4] M. Maaruf, M. S. Mahmoud, and A. Ma'arif, "A survey of control methods for quadrotor uav," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 652–665, 2022.

[5] H. Wang, S. Lu, G. Qian, J. Ding, Y. Liu, and Q. Wang, "A two-step strategy for online fault detection of high-resistance connection in bldc motor," *IEEE Transactions on Power Electronics*, vol. 35, no. 3, pp. 3043–3053, 2019.

[6] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 45–52.

[7] R. C. Kavanagh, "Improved digital tachometer with reduced sensitivity to sensor nonideality," *IEEE transactions on industrial electronics*, vol. 47, no. 4, pp. 890–897, 2000.

[8] L. Yang, Y. Li, Q. Lin, H. Jia, X.-Y. Li, and Y. Liu, "Tagbeat: Sensing mechanical vibration period with cots rfid systems," *IEEE/ACM transactions on networking*, vol. 25, no. 6, pp. 3823–3835, 2017.

[9] Z. Li, H. Dai, W. Wang, A. X. Liu, and G. Chen, "Pcias: Precise and contactless measurement of instantaneous angular speed using a smartphone," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–24, 2018.

[10] P. Cheng, M. S. M. Mustafa, and B. Oelmann, "Contactless rotor rpm measurement using laser mouse sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 3, pp. 740–748, 2011.

[11] K. Miyashita, T. Takahashi, and M. Yamanaka, "Features of a magnetic rotary encoder," *IEEE Transactions on Magnetics*, vol. 23, no. 5, pp. 2182–2184, 1987.

[12] C. Lin, C. Ji, J. Xiong, C. Xiang, L. Wang, G. Wu, and Q. Zhang, "Wi-rotate: An instantaneous angular speed measurement system using wifi signals," *IEEE Transactions on Mobile Computing*, 2022.

[13] M. Heggo, L. Bhatia, and J. A. McCann, "Rftacho: Non-intrusive rf monitoring of rotating machines," in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2022, pp. 403–414.