IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Thrust Microstepping via Acceleration Feedback in Quadrotor Control for Aerial Grasping of Dynamic Payload

Ashish Kumar†, Laxmidhar Behera†, *Senior Member, IEEE*

***Abstract*—In this work, we propose an end-to-end Thrust Microstepping and Decoupled Control (TMDC) of quadrotors. TMDC focuses on precise off-centered aerial grasping of payloads dynamically, which are attached rigidly to the UAV body via a gripper contrary to the swinging payload. The dynamic payload grasping quickly changes UAV's mass, inertia etc, causing instability while performing a grasping operation in-air. We identify that to handle unknown payload grasping, the role of thrust controller is crucial. Hence, we focus on thrust control without involving system parameters such as mass etc. TMDC is based on our novel Thrust Microstepping via Acceleration Feedback (`TMAF`) thrust controller and Decoupled Motion Control (`DMC`). `TMAF` precisely estimates the desired thrust even at smaller loop rates while `DMC` decouples the horizontal and vertical motion to counteract disturbances in the case of dynamic payloads. We prove the controller's efficacy via exhaustive experiments in practically interesting and adverse real-world cases, such as fully onboard state estimation without any positioning sensor, narrow and indoor flying workspaces with intense wind turbulence, heavy payloads, non-uniform loop rates, etc. Our TMDC outperforms recent direct acceleration feedback thrust controller (DA) and geometric tracking control (GT) in flying stably for aerial grasping and achieves RMSE below 0.04m in contrast to 0.15m of DA and 0.16m of GT.**

***Index Terms*—Aerial Systems: Applications; Grasping; Thrust Control, Decoupled Motion Control, Dynamic Payloads**

Code: https://github.com/ashishkumar822/TMDC **Video:** See attachment.

I. Introduction

AERIAL manipulation is a highly challenging research area and a convoluted design practice. It attracts a variety of applications, such as transporting rigidly attached or swinging payloads, gripper-based in-air grasping, etc. This paper focuses on the control of an aerial grasping system for small items, i.e. a quadrotor endowed with a gripper having a three-fingered jaw end-effector (Fig. 1). Notably, developing the control system for such a system is increasingly difficult due to its inherent challenges, as discussed below.

Foremost is the necessity that unlike swinging payloads [1], aerial grasping requires approaching the target laterally. This gives rise to *off-centred* end-effector configuration as it is the natural setup to grasp an item mid-air. This causes a permanent shift in the centre of gravity (CoG). Then after grasping, the grasped item further shifts the CoG dynamically, leading to dynamic payload situations full of critical challenges such as; *First*, according to our experiments, for grasping an item, the UAV needs to approach the target slowly and requires high

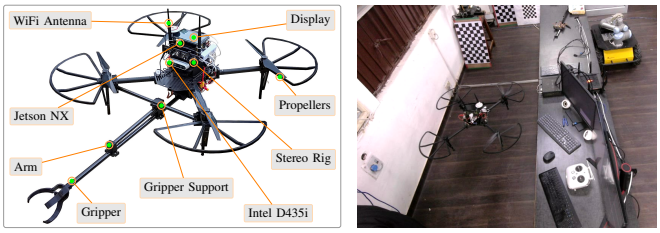

Figure 1: Left: Our aerial manipulator. Right: TMDC controlling a very large sized quadrotor (1.30m×0.90m×0.45m) within a workspace clearance of 10cm at a very small altitude of 0.5m, a quite adverse and dangerous practice considering the large sized quadrotor.

stability in height as well as in lateral direction. Without that, it is extremely likely to miss the target since the gripper is a small entity, and even a small external disturbance can fail the grasp.

Second, due to the fast-changing system dynamics (mass, CoG, etc) during payload attach (grasp) and detach (release) events, the platform is most vulnerable. A centered payload only exerts a downward force, however, in addition to the downward force, an off-center load (grasped item) exerts an intensive torque along the roll or pitch axis depending upon the gripper mounting, thus causing severe unwanted motion of the UAV, resulting in a crash. Notice that this is different from swinging payloads, which are attached beforehand. In contrast, in our case, the payload is attached dynamically autonomously and is held into the gripper tightly.

Further, from a practical standpoint, dealing with the non-uniform execution rates of the control system is another critical challenge. This challenge arises because, in aerial grasping, several compute-intensive sub-systems co-exist, such as deep neural networks and visual SLAM, which occupy the computing resources intensively, thus preventing uniform loop rates. Since a control system is sensitive to the loop rates, an execution delay of even 5-10ms can degrade its performance [2], thus preventing the control system from handling intense disturbances, e.g. torque caused by grasping an off-center item.

Furthermore, precise take-off-hover-land in the presence of dynamic payloads, inaccurate system parameters, imbalanced platform due to the off-center grippers, battery discharging, etc., make the overall problem quite convoluted due to an imprecise thrust estimation. Thus, the control system design for aerial grasping is currently a state-of-the-art problem.

In this work, we underline that these challenges are closely related to the thrust controller, which is a crucial component of the quadrotor control. Typically, the quadrotor control can be sectioned into an outer loop (position controller) and an

†EE, Indian Institute of Technology (IIT), Kanpur, India. {krashish,lbehera}@iitk.ac.in

inner loop (attitude controller). These loops strictly depend on the system parameters, such as mass and propeller coefficient, and are severely affected by their incorrect values.

The position controller has a position feedback loop that computes acceleration from a desired 3D position and a thrust controller that computes the required thrust from this acceleration. The thrust controller is a partially open-loop controller, known as model inversion [3]. Due to its open-loop nature and dependency on the accurate values of mass and gravity, it becomes a bottleneck in aerial grasping since dynamic payloads change the mass and CoG quickly, and unmodeled external disturbances degrade its performance.

Hence most works focus on improving the position feedback loop, i.e., LQR [4], MPC [5], [6], INDI [7] that requires even the motor rotational speeds in addition to the system parameters. In contrast, a recent approach [8] replaces the open-loop thrust controller with Direct Acceleration Feedback (DA) thrust controller. However, it requires high-frequency loop rates and does not incorporate velocity control, a key requirement in aerial grasping to perform visual servoing.

While there exist a number of works for cable swinging payloads [9], [10], [1]; they take model-based approach, e.g. spring-mass model [9] to address their challenges which are different from ours and do not transfer to our case as they target a different problem. Moreover, they focus largely on position control but use the traditional thrust control model. On the contrary, our interest is to develop a thrust controller to address off-center aerial grasping challenges.

Apart from the above, the control system's performance significantly depends on the feedback signal's accuracy and rate. Therefore, the works dealing with dynamic payloads report the control performance with very small weights by obtaining position feedback from millimeter-accurate VICON and keeping a controlled workspace. For instance, [11] shows control of a mini helicopter for centered payload case with a tiny payload by using VICON feedback. Even DA [8] is tested with VICON. In contrast, real-world aerial grasping is far more complex since it relies only on the onboard positioning system, which is often quite slow and imprecise. Therefore, comparative evaluation of control systems is rarely available in the literature and is also a difficult task.

In light of the above discussion, in this paper, we focus on aerial grasping using an off-center gripper. To the best of our knowledge, an end-to-end controller specifically targeting off-centered aerial grasping is not visible in the literature. To this end, we make the following contributions:

Firstly, our major contribution is a novel thrust controller called Thrust Microstepping via Acceleration Feedback (`TMAF`). It is highly responsive even at smaller loop rates (e.g. 30Hz) against fast-changing system parameters and can easily tackle battery discharging and deterioration of its current deliverance that directly affects the rotor thrust (Sec. III).

Secondly, we develop an intuitive Decoupled Motion Control (`DMC`) which decouples the horizontal and vertical motion of the UAV to handle dynamic payloads (Sec. IV-B).

Finally, we club `TMAF` and `DMC` into an end-to-end control system called Thrust Microstepped Decoupled Control (TMDC) for aerial grasping. TMDC can handle non-uniform loop rates, offers precise position and velocity control while showing highly stable flights during grasping of unknown payload (Sec. IV).

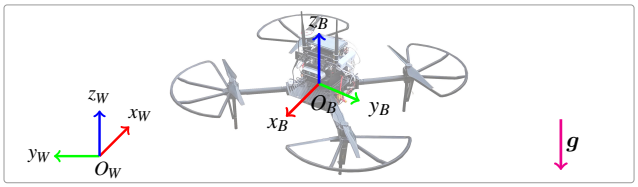

Figure 2: Quadrotor reference frames.

We benchmark TMDC in quite practical and adverse settings, e.g. obtaining position feedback from onboard SLAM that is slow and imprecise relative to VICON, and controlling a large quadrotor (1.30m×0.90m×0.45m) at very low altitude (below 50cm), in narrow indoor workspace with merely 10cm of clearance (Fig. 1). These are quite dangerous and difficult objectives due to the risk of collision and casualty, ceiling and ground effects [12], and intense wind turbulence, however, these are common situations in aerial grasping. Noticeably, TMDC handles them precisely, indicating the most representative accomplishment of this work (Sec. VI).

II. Quadrotor System and Nomenclature

Our quadrotor platform is shown in Fig. 2. We define a *world frame* $\mathscr{F}_W$ having axes $\{\boldsymbol{x}_W, \boldsymbol{y}_W, \boldsymbol{z}_W\}$ that is always fixed at the take-off location. The gravity vector $\boldsymbol{g} \in \mathbb{R}^3$ is aligned opposite to the $\boldsymbol{z}_W$ axis. While the quadrotor configuration is described via a *body frame* $\mathscr{F}_B$, which coincides with the Center-of-Gravity (CoG) of the quadrotor, and has axes $\{\boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$. Both the frames $\mathscr{F}_W$ and $\mathscr{F}_B$ are FLU or *Front*-*Left*-*Up*, denoting their $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ axes.

The position and orientation of $\mathscr{F}_B$ relative to $\mathscr{F}_W$ is given by a vector $\boldsymbol{p}_B = [x, y, z]^T \in \mathbb{R}^3$ and a rotation matrix $\boldsymbol{R}_B \in SO3$. The quadrotor's velocity, acceleration, and angular velocity, are represented as $\boldsymbol{v}_B = [v_x, v_y, v_z]^T \in \mathbb{R}^3$, $\boldsymbol{a}_B = [a_x, a_y, a_z]^T \in \mathbb{R}^3$, $\boldsymbol{\omega}_B = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3$, expressed in $\mathscr{F}_W$, except that $\boldsymbol{\omega}_B$ is in $\mathscr{F}_B$. *Note:* The *boldface* denotes a vector.

The quadrotor motion is governed by the *thrust intensity* $f_B \in \mathbb{R}$, and *torques* $\boldsymbol{\tau}_B = [\tau_x, \tau_y, \tau_z]^T \in \mathbb{R}^3$. f_B is applied parallel to $\boldsymbol{z}_B$ whereas $\boldsymbol{\tau}_B$ operates in $\mathscr{F}_B$ which controls the roll-pitch-yaw of the quadrotor in $\mathscr{F}_B$. Practically, controlling the quadrotor's position/velocity and heading (ψ) is of great interest, which depends on $\{f_B, \phi, \theta, \omega_z\}$. Here, ϕ and θ are defined in $\mathscr{F}_B$ whereas ψ in $\mathscr{F}_W$. The position controller generates $\{f_B^\star, \phi^\star, \theta^\star\}$ commands based on the desired state, i.e. desired position/velocity and heading ($\psi^\star$). The '$\star$' superscript denotes the desired value of any variable.

These commands are sent to the low-level attitude controller, which essentially controls the angular velocities of the platform to generate $\boldsymbol{\tau}_B^\star$ for achieving the desired $\{\phi^\star, \theta^\star, \psi^\star\}$. Finally, based on the desired thrust and the torques, motor speed allocation is performed [3], which in turn is realized via Electronic-Speed-Controller (ESC).

In the quadrotor's position control process, the thrust controller or *Model Inversion* (Eq. 1) plays a crucial role.

$$\boldsymbol{f}_B^\star = m(\boldsymbol{a}_B^\star + \boldsymbol{g}) - \boldsymbol{f}_e \tag{1}$$

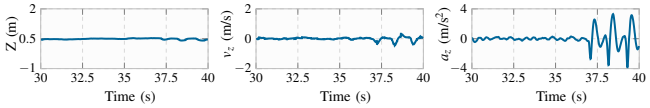

Figure 3: Effect of a sudden force of 12N in z_W onto the z, v_z, a_z of the quadrotor. Noticeably, the acceleration signal captures a detailed profile of this event, while position and velocity signals do not.

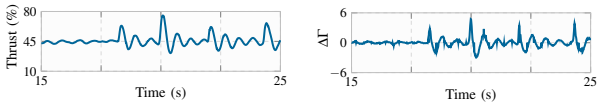

Figure 4: Thrust intensity ($f_B^\star$) and microsteps ($\Delta\Gamma$) output of TMAF.

where, $\boldsymbol{f}_e \in \mathbb{R}^3$ is an external disturbance (wind draft, sudden jerk, imperfect modelling), m is the UAV mass, and $\boldsymbol{a}_B^\star$ is the desired acceleration produced by the position feedback loop. Since motor thrust is applicable only along $\boldsymbol{z}_B$, a desired orientation of the quadrotor is also required to attain $\boldsymbol{f}_B^\star$. It is computed via geometric tracking controller [13] as follows:

$$\boldsymbol{z}_B^\star = \boldsymbol{f}_B^\star / \|\boldsymbol{f}_B^\star\| \tag{2}$$

By using the Eq. 1 and 2, the desired thrust intensity $f_B^\star = \boldsymbol{z}_B^T \boldsymbol{f}_B^\star$, and $\{\phi^\star, \theta^\star\}$ can be computed [13], whereas the desired heading $\psi^\star$ is obtained as input from the user.

As visible in Eq. 1 that the model inversion depends on the mass, gravity and disturbance modeling to estimate $f_B^\star$ accurately, it may lead to incorrect thrust estimation and, thus, undesired response in dynamic payload conditions of aerial grasping. Hence our problem of interest is to develop a mass and gravity-independent thrust controller which can also handle low-frequency loop rates and non-uniform execution.

III. Thrust Microstepping

We propose Thrust Microstepping via Acceleration Feedback (TMAF) to estimate the desired thrust. The microstepping is inspired by the microstepping technique for fine-grained control of the stepper motors. Whereas the use of acceleration feedback is based on our observations that during payload attach, detach, or disturbance events, the acceleration signal changes rapidly, even at low-frequency sampling and small variations in the position or velocity signal (Fig. 3). We divide TMAF into two steps, as described below.

1) Microstepper Controller: In this step, we generate a microstep $\boldsymbol{\Delta\Gamma}$ based on the position feedback loop's output $\boldsymbol{a}_B^\star$. For that, we define the following control law:

$$\boldsymbol{\Delta\Gamma} = \alpha \boldsymbol{e_a} + \beta \dot{\boldsymbol{e}}_{\boldsymbol{a}} \tag{3}$$

where, $\boldsymbol{e_a} = \boldsymbol{a}_B^\star - \boldsymbol{a}_B$ is the acceleration error, and $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are the controllers gains. Notice that, this controller essentially tracks $\boldsymbol{a}_B^\star$ and fully captures the variations in the acceleration signal. When $\boldsymbol{a}_B$ undergoes large changes, $\boldsymbol{e_a}$ becomes large. Thus, $\boldsymbol{\Delta\Gamma}$ changes accordingly to handle the changes.

2) Microstepped Thrust Estimation: This step computes the total instantaneous thrust $\boldsymbol{\Gamma}_t$ or $\boldsymbol{f}_B^\star$ by accumulating past $\boldsymbol{\Delta\Gamma}$'s in a recursively i.e $\boldsymbol{\Gamma}_t = \boldsymbol{\Gamma}_{t-1} + \boldsymbol{\Delta\Gamma}_t$, where, $\boldsymbol{\Gamma}_{t-1} = \sum_{i=0}^{t-1} \boldsymbol{\Delta\Gamma}_i$. Now, rewriting the previous expression as follows:

$$\boldsymbol{f}_B^\star \equiv \boldsymbol{\Gamma}_t = \boldsymbol{\Gamma}_0 + \sum_{i=1}^{t} \boldsymbol{\Delta\Gamma}_i \implies \boldsymbol{f}_B^\star = \boldsymbol{f}_{B_{t-1}}^\star + \boldsymbol{\Delta\Gamma}_t \tag{4}$$

where, $\boldsymbol{\Gamma}_0$=0, and t=0 indicates flight commencement. For the convergence proof, please refer to Appendix A.

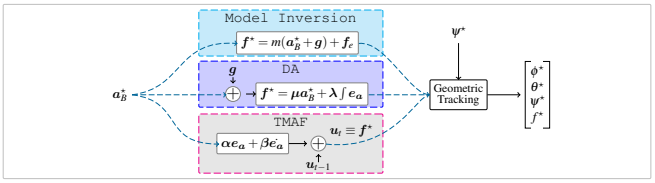

Figure 5: Thrust Microstepping via Acceleration Feedback (TMAF) *vs* Direct Acceleration (DA) Feedback [8] and Model Inversion [3].

The accumulation in Eq. 4 should not be confused with the integrator in the PID control because '*dt*' is absent here. This enables TMAF to exploit rapid acceleration changes to counter sudden disturbances instantly (Fig. 4, Fig. 17). Now $\boldsymbol{f}_B^\star = \{f_x^\star, f_y^\star, f_z^\star\}$ can be used to obtain $\{f_B^\star, \phi^\star, \theta^\star\}$ via Eq. 2.

A. Structural Analysis of TMAF and State-of-the-art

Model inversion (MI) (Eq. 1) and DA [8] are the most suitable approaches to compare with TMAF. For better analysis, we write formulations of the three controllers below:

$$\text{MI} \equiv \boldsymbol{f}_B^\star = m(\boldsymbol{a}_B^\star + \boldsymbol{g}) - \boldsymbol{f}_e \tag{5}$$

$$\text{DA} \equiv \boldsymbol{f}_B^\star = \boldsymbol{\mu} \boldsymbol{a}_B^\star + \boldsymbol{\lambda} \int \boldsymbol{e_a} dt \tag{6}$$

$$\text{TMAF} \equiv \boldsymbol{f}_B^\star = \alpha \boldsymbol{e_a} + \beta \dot{\boldsymbol{e}}_{\boldsymbol{a}} + \boldsymbol{f}_{B_{t-1}}^\star \tag{7}$$

where $\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ are the tunable parameters, $\boldsymbol{e_a} = \boldsymbol{a}_B^\star - \boldsymbol{a}_B$, and $\boldsymbol{f}_B^\star$ is the desired thrust vector to attain $\boldsymbol{a}_B^\star$. If comparing Eq. 6 with Eq. 5, it can be noticed that $\boldsymbol{\mu}$ absorbs the mass and gravity, while $\boldsymbol{\lambda}$ handles the uncertainties and external disturbances [8]. Whereas our TMAF is substantially different from the model inversion and DA, as shown in Fig. 5.

1) Absence of Mass and Gravity: Similar to model inversion, DA also needs gravity compensation that is added to $\boldsymbol{a}_B^\star$ (Fig. 5). Whereas TMAF is completely unaware of the system mass and gravity. This makes TMAF free of uncertainty in these parameters, allowing TMAF to offer precise control.

2) Absence of Error Integral: DA (Eq. 6) is a Proportional-Integral (PI) controller. At low-frequency loop rates, the integral term turns DA slower due to the low pass filtering effects and also needs integral windup. The low-frequency loop rates also prevent using larger values of $\boldsymbol{\lambda}$ in DA to achieve faster response, because, it causes system oscillations due to the enhanced integrator's dominance [8]. Therefore, DA should be run at high loop rates ($>$ 250Hz). However, in this case, the accelerometer noise jumps-in, which adversely affects the DA performance, thus mandating minimum-lag filtering of the acceleration measurements [8].

In contrast, TMAF only has difference terms (Eq. 3) which alleviates runtime bias estimation, and the absence of an integral term turns TMAF highly responsive while avoiding the integral windup. Therefore, TMAF can seamlessly work even at a low frequency which naturally acts as a low pass filter [8] on the accelerometer signal (Sec. VI-F). Thus unlike DA, our TMAF does not need acceleration filtering, saving overhead when using TMAF in microcontrollers.

Moreover, the Eq. 3 consists of an acceleration differential term (Jerk), making TMAF instantly reactive to sudden disturbances. This term may have high-frequency noise, but the low-frequency operation implicitly handles that.

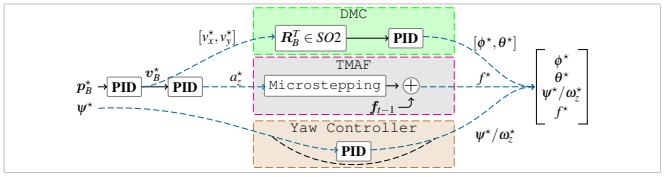

Figure 6: High-level design of TMDC.

3) Rapid Acceleration Changes: The absence of integral term allows TMAF to exploit rapid changes in the acceleration. This is why TMAF shows stiffness against sudden disturbances (See Sec. VI-I4 and video). In contrast, MI (Eq. 1) can not harness them due to its open-loop nature, and DA, by nature, suppresses them due to its integral action over e_a (Eq. 6).

B. TMAF in Aerial Manipulation

In aerial grasping, the dynamic payloads alter the system mass, and the battery discharging and ageing reduce the current output. Since the system parameters are still the same, the above events reduce rotor thrust via Eq. 1. To compensate, the thrust should be corrected accordingly. However, in model inversion [3], it is done manually via hit-and-trial methods and is kept fixed during the flight (50% in [14]). While in DA, the value of $\boldsymbol{\mu}$ needs to be adjusted.

In contrast, TMAF captures these events via accelerometer readings since inaccurate thrust would cause undesired motion, and adjusts the thrust via microstepping without requiring manual tuning (Fig. 13 and See video).

IV. Thrust Microstepped Decoupled Control

In UAVs, the position control is developed specifically for a problem under consideration. In the same way, we also develop an end-to-end control design TMDC (Fig. 6) for off-center aerial grasping that offers different control modes in different directions, e.g. position control in $\boldsymbol{z}_W$ while velocity control in $\boldsymbol{x}_W$ or $\boldsymbol{y}_W$. It is a crucial functionality to execute aerial grasping operations precisely. TMDC achieves that via TMAF and Decoupled Motion Control (DMC) to generate $\{f_B^\star, \phi^\star, \theta^\star, \psi^\star\}$ precisely from the input needs (different control modes, input state) instead of Eq. 1 and 2.

In this work, we have focused on PID-based loops, which can be replaced by any other controller as future improvements, e.g. non-linear control, such as a robust or sliding mode, etc.

A. Position Feedback Loop

The position feedback loop of TMDC produces the $\boldsymbol{a}_B^\star$ commands. It has two cascaded controllers, motivated by the need for position control in aerial grasping to perform precise navigation (via position commands) and visual servoing (via velocity commands) during grasping operations.

1) Position Controller: It outputs virtual velocity commands $\boldsymbol{v}_B^\star$, and the control law is defined as follows:

$$\textbf{PID}_{\boldsymbol{p}} \equiv \boldsymbol{v}_B^\star = \boldsymbol{K}_p^p \boldsymbol{e_p} + \boldsymbol{K}_i^p \int \boldsymbol{e_p} dt + \boldsymbol{K}_d^p \dot{\boldsymbol{e}}_{\boldsymbol{p}} \tag{8}$$

where, $\boldsymbol{e_p} = \boldsymbol{p}_B^\star - \boldsymbol{p}_B$ is position error, $\boldsymbol{K}^p \equiv \{\boldsymbol{K}_p^p, \boldsymbol{K}_i^p, \boldsymbol{K}_d^p\}$. $\boldsymbol{K}_p^p = \text{diag}(k_{p_x}^p, k_{p_y}^p, k_{p_z}^p)$, $\boldsymbol{K}_i^p = \text{diag}(k_{i_x}^p, k_{i_y}^p, k_{i_z}^p)$, and $\boldsymbol{K}_d^p = \text{diag}(k_{d_x}^p, k_{d_y}^p, k_{d_z}^p)$ are PID gains in the $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ axes.

2) Velocity Controller: It generates virtual commands based on $\boldsymbol{v}_B^\star$ such that $\boldsymbol{v}_B \to \boldsymbol{v}_B^\star$. The control law is as follows:

$$\textbf{PID}_{\boldsymbol{v}} \equiv \boldsymbol{q}_B = \boldsymbol{K}_p^v \boldsymbol{e_v} + \boldsymbol{K}_i^v \int \boldsymbol{e_v} dt + \boldsymbol{K}_d^v \dot{\boldsymbol{e}}_{\boldsymbol{v}} \tag{9}$$

where $\boldsymbol{e_v} = \boldsymbol{v}_B^\star - \boldsymbol{v}_B$ is velocity error, $\boldsymbol{K}^v \equiv \{\boldsymbol{K}_p^v, \boldsymbol{K}_i^v, \boldsymbol{K}_d^v\}$. $\boldsymbol{K}_p^v = \text{diag}(k_{p_x}^v, k_{p_y}^v, k_{p_z}^v)$, $\boldsymbol{K}_i^v = \text{diag}(k_{i_x}^v, k_{i_y}^v, k_{i_z}^v)$, and $\boldsymbol{K}_d^v = \text{diag}(k_{d_x}^v, k_{d_y}^v, k_{d_z}^v)$ are the PID gains.

The output of $\textbf{PID}_{\boldsymbol{v}}$ i.e. $\boldsymbol{q}_B = \{q_x, q_y, q_z\} \equiv \boldsymbol{a}_B^\star$ is fed to TMAF which produces the desired thrust intensity and direction i.e. $\{f_B^\star, \phi^\star, \theta^\star, \psi^\star\}$ similar to MI or DA.

In this paper, we are also interested in decoupled control of quadrotor, which utilizes TMAF only in the z direction and utilizes the output of $\textbf{PID}_{\boldsymbol{v}}$ differently, as discussed below.

B. Decoupled Motion Controller (DMC)

Inspired by the 2D motion, we propose a decoupled control of the quadrotor which boils down to controlling the vertical direction ($\{z, v_z, a_z\}$), horizontal direction ($\{x, y, v_x, v_y, a_x, a_y\}$), and heading ($\psi$) independently. Since $\{\phi, \theta\}$ governs the horizontal motion i.e. in $\{x, y\}$, we assume the quadrotor as a 2D-agent navigating in xy-plane with z controlled by TMAF, and x, y by DMC, with heading control done separately.

In DMC, we propose to compute $\{\phi^\star, \theta^\star\}$ directly without using forces. This is in contrast to the existing works, such as geometric tracking controller [13] that uses forces in all directions to obtain $\{\phi^\star, \theta^\star\}$, whereas DMC uses force only in z direction. This design helps DMC to counter unmodeled disturbances induced in the lateral directions caused by the off-centered dynamic payloads. This approach is advantageous for aerial grasping to develop complex state machines [2].

1) Direction $\{\phi^\star, \theta^\star\}$: DMC has two independent controllers, one for each of $\boldsymbol{x}_W$ and $\boldsymbol{y}_W$, producing $\{\phi^\star, \theta^\star\}$. Since ϕ, θ induce motion in x_B and y_B, the $\{v_x^\star, v_y^\star\}$ need to be mapped into the body frame due to the presence of heading (ψ). We denote the body frame velocity as v_D which is fed to $\textbf{PID}_{\boldsymbol{v}}$ to obtain $\boldsymbol{q} = \{q_x, q_y\}$ that essentially represents $\{\phi^\star, \theta^\star\}$. The above operation is summarized below:

$$v_D^\star = \boldsymbol{R}_B^T(\psi) v_B^\star \tag{10}$$

$$[\phi^\star, \theta^\star]^T \equiv \boldsymbol{q} = \textbf{PID}(\boldsymbol{v}_D^\star) \tag{11}$$

where $\boldsymbol{R}_B^T \in SO2$, transforms $v_B^\star$ from $\mathscr{F}_W$ to $\mathscr{F}_B$, and $[\phi^\star, \theta^\star]^T$ steers the quadrotor from $[p_x, p_y]$ to $[p_x^\star, p_y^\star]$.

2) Thrust Intensity: In DMC, the thrust estimation is done only in $\boldsymbol{z}_W$ to control $\{z, v_z, a_z\}$. It is done via TMAF which outputs $f_z^\star$. Then we compute $f_B^\star$ from $f_z^\star$ as follows:

$$f_B^\star = f_z^\star / (\boldsymbol{z}_B . \hat{k}) \tag{12}$$

where, $\hat{k}$ is the unit vector along z_W. This equation is a simple manipulation of $f_B^\star = \boldsymbol{z}_B^T \boldsymbol{f}_B^\star$, given that thrust can only be applied along $\boldsymbol{z}_B$. Here $\boldsymbol{z}_B . \boldsymbol{z}_W = 0$ should be avoided, which is generally done by limiting the angles in this range, otherwise f_z becomes zero since ϕ or $\theta = 90^o$, thus causing free-fall of the quadrotor, unless ϕ and θ ensure $\boldsymbol{z}_B . \boldsymbol{z}_W \neq 0$.

3) Yaw Controller: GT [13] uses the heading $\psi^\star$ (obtained from the user or planner) to calculate $\{\phi^\star, \theta^\star\}$. Whereas, we control $\psi^\star$ independently via a PID controller that produces $\omega_z^\star$ based on ψ^*. The control law is given as:

$$\textbf{PID}_{\omega} \equiv \omega_z^\star = k_p e_\psi + k_i \int e_\psi dt + k_d \dot{e}_\psi \tag{13}$$

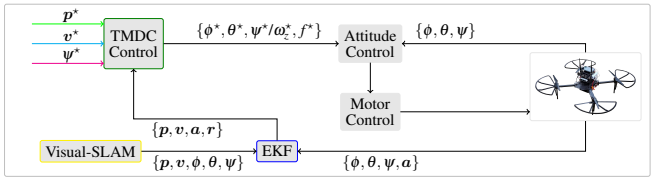

Figure 7: Connectivity between TMDC and other feedback components.

Table I. TMDC gains

Gain	PID	PID	TMAF	$\mathbf{PID}_\omega$
$\boldsymbol{K}_p$	diag(3.0, 3.0, 1.0)	diag(3.0, 3.0, 1.5)	diag(0.6, 0.6, 0.6)	0.4
$\boldsymbol{K}_i$	diag(0.1, 0.1, 0.1)	diag(0.5, 0.5, 0.25)	–	0.01
$\boldsymbol{K}_d$	diag(0.8, 0.8, 0.1)	diag(0.5, 0.5, 0.20)	diag(0.02, 0.02, 0.02)	0.8

Table II. UAV platform specifications.

Attribute	Specification	Attribute	Specification
Size	0.90m× 0.90m× 0.45m	Operating Voltage	22V
Size w/ Gripper	1.5m × 0.90m × 0.45m	Hover time	20 minutes
Weight	2.5Kg	Hover time w/ Gripper	15 minutes
Weight w/ Gripper	3.4Kg	Onboard Computer	NVIDIA Jetson-NX
Rotor diameter	0.46m	Communication	UART @921600 Baud
Rotor-to-Rotor distance	0.35m	Stereo-Rig	2× @432 × 240, 30Hz

where, $e_\psi = \psi^\star - \psi$. The $\omega_z^\star$ is sent to the angular rate controller to achieve $\psi \to \psi^\star$. If required, $\psi^\star$ can also be sent to the attitude controller.

V. Implementation

1) Loop Rates: We obtain position feedback via onboard visual-SLAM at 30Hz, and raw acceleration from the onboard IMU at 80Hz. They are fused using EKF to estimate the quadrotor state. The $\mathbf{PID}_p, \mathbf{PID}_v$ and TMAF runs at 30Hz, 60Hz and 80Hz respectively. The overall information flow between the different controllers is shown in Fig. 7.

2) Derivative Filtering: Since derivative controller input can be noisy, we use a cosine-weighted moving average filter over a window of 4 samples. The weights are given by $w_i = \cos(i\zeta)\ \forall\ i \in [0,4), \quad \zeta = \zeta_m/4, \zeta_m = 80^o$, and $i = 0$ refers to the latest sample. The same applies to the "jerk" term in Eq. 3 of TMAF, which we compute via Euler backward difference on the accelerometer measurements.

3) Tuning: TMDC has six PIDs and a TMAF controller, which we tune to obtain a behaviour similar to a critically damped system. We use `runtime-reconfiguration` and `rqt_reconfigure` utility of ROS, which allows changing the gains at runtime, reducing the tuning effort only to $\sim 5-10$ minutes. For tuning, we first act upon PID in z_W, then in x_W and y_W, which are identical. Due to the cascaded nature of TMDC, we first tune TMAF gains, then $\mathbf{PID}_v$ and finally $\mathbf{PID}_p$ and $\mathbf{PID}_\omega$. Table I shows the tuned values, which can be an initial guess for a different platform. Our TMDC is quite resistant to the variance in its gains. Hence, various controllers of TMDC need not be perfectly tuned to achieve a desirable control performance (Sec. VI-G).

VI. Experiments

We rigorously analyze TMDC on a quadrotor (Fig. 1, Table II) in several adverse scenarios: (*i*) different flying workspaces, (*ii*) autonomous take-off, hover, and land, (*iii*) step-response, (*iv*) navigation via trajectory execution, (*v*) battery discharge, (*vi*) non-uniform loop rates and gain variance, and (*vii*) center and off-center dynamic payloads. Finally, we compare TMAF with DA [8] and DMC with geometric tracking [13] comprehensively. See the video for a real-world demonstration of these experiments.

Table III. Takeoff-Hover-Land accuracy, average of 10 trails.

Workspace	RMSE (m)						
	Take-off		Hover			Land	
	x	y	x	y	z	x	y
`Constrained_ws`	0.04	0.03	0.04	0.03	0.02	0.03	0.04
`Open_ws`	0.06	0.06	0.03	0.05	0.02	0.03	0.05
`Open_ws+Fan`	0.03	0.02	0.02	0.07	0.02	0.03	0.05
`Open_ws+Gripper`	0.06	0.07	0.02	0.06	0.02	0.03	0.09
`Open_ws+Gripper+Payload`	0.07	0.08	0.03	0.07	0.02	0.02	0.07

A. Constrained Flying Workspaces with Adverse Settings

We define five workspaces: `Constrained_ws`, `Open_ws`, `Open_ws+Fan`, `Open_ws+Gripper`, and `Open_ws+Gripper+Payload`. The `Constrained_ws` has only 10cm gap on both sides of the UAV in y_W. The `Open_ws` has 2.50m ceiling height and 2m gap on all sides.

We analyze the take-off, hover, and land profiles for each workspace to understand TMDC's behaviour (See Fig. 8). Interestingly, regardless of the workspace, TMDC exhibits consistent performance for each control input, i.e. $\{x^\star, y^\star, z^\star\}$. The `Constrained_ws` poses a challenge in front of TMDC to stabilize the quadrotor due to a small flying area, intense rotor draft, and turbulence originating due to the large size of the quadrotor. This demands precise thrust controlling. For this reason, for `Constrained_ws`, the position feedback loop of TMDC produces noisy desired acceleration ($a_z^\star$), and hence the desired thrust, which indicates frequent corrections. This enables TMDC to achieve stable hovering in this workspace while maintaining high accuracy in the lateral direction despite the narrow clearance (10cm).

Similarly, in `Open_ws+Fan`, the TMAF adjusts the thrust to counter the external disturbances precisely while maintaining high hovering accuracy. This workspace has the highest thrust needed because the overhead fan tries to push the UAV downwards (Fig. 9). The other two workspaces have relatively lower aggressive microstepping due to eased constraints.

B. Takeoff–Hover–Landing Quantitative Performance

Based on TMDC, we also develop high-level controllers for smooth takeoff and landing, crucial for flight autonomy, while hovering is achieved directly via requesting the set-point.

1) Takeoff: For takeoff, we use position control in x_W, y_W and velocity control in z_W with $x^\star = y^\star = 0$ and $v_z^\star = 0.20$m/s, allowing precise vertical takeoff without being affected by the ground-effects [12] (see Sec. VI-I5). From Fig. 8, it is noticeable that TMDC exhibits the desired response, owing to TMAF in z and DMC in $x-y$, and visible in Table III.

2) Hovering: We use position control in all the axes with $x^\star = y^\star = 0$, $z^\star = 0.5$m. Fig. 8 indicates a highly accurate hovering response of TMDC, also verifiable via Table III.

3) Landing: Similar to takeoff, we set $x^\star = y^\star = 0$, $v_z^\star = -0.10$m/s. Fig. 8 indicates a small error between landing and takeoff position, which is a desirable attribute to execute landing in constrained workspaces precisely (Table III).

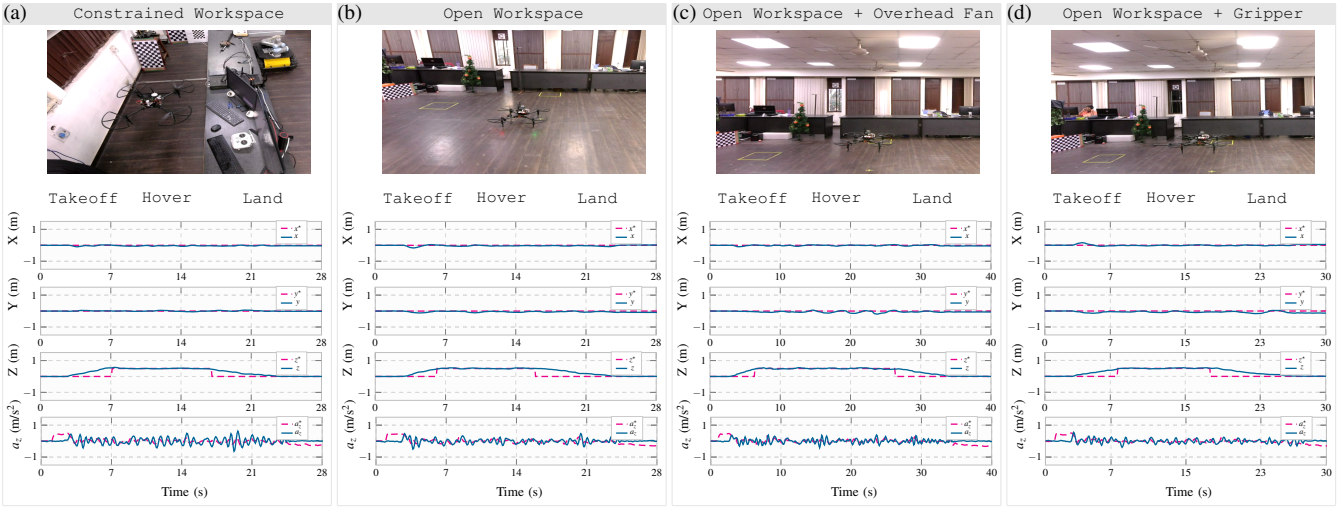

Figure 8: Takeoff-Hover-Land performance of TMDC in different flying workspaces having challenging situations.

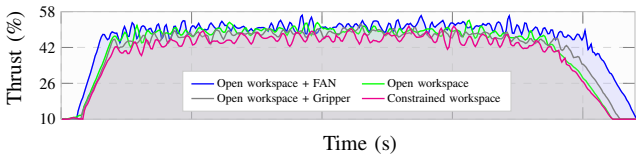

Figure 9: Thrust profile of TMDC in various workspaces.

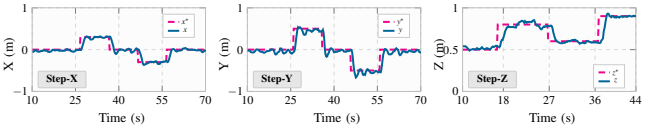

Figure 10: Step responses in $\mathcal{F}_W$, conducted as different experiments.

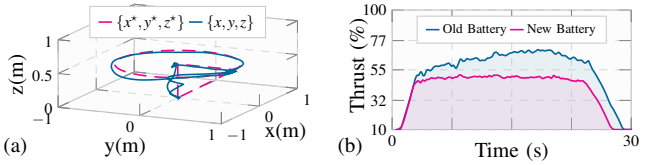

Figure 11: (a) TMDC for trajectory execution @radii=0.7m, average of 10 trails, and (b) Thrust output of TMAF for new *vs* old battery.

C. Step Response

We also analyse the step response of TMDC in all directions (Fig. 10). It can be seen that TMDC converges quickly to the new set-points, and stays there precisely. A slight overshoot is also visible during set-point changes, which was caused to counter heavy turbulence indoors due to the large UAV size.

D. Trajectory Execution or Non-Stationary Configuration

We also evaluate TMDC for trajectory execution. Fig. 11a shows circular trajectory execution response, averaged over 10 trails; 5 with gripper and 5 without gripper. This analysis shows that apart from hovering, TMAF outputs accurate thrust even during navigation, trajectory execution or non-stationary configuration while handling unknown payloads.

E. Battery Discharge & Battery Life

We analyze TMAF behaviour for two fully-charged batteries: a new, and an old one having decreased current supplying capability and heating issues. Fig. 11b shows our analysis.

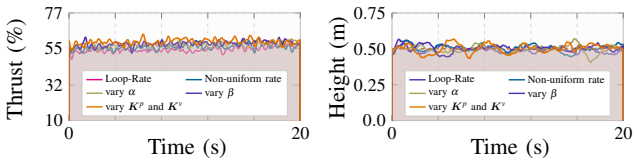

Figure 12: Ablation study of TMDC in various scenarios: (*i*) Scaling loop-rates $(0.33-1.25)$, (*ii*) non-uniform loop-rates (a jitter of ± 5 ms), (*iii-iv*) scaling α, β of TMAF $(0.75-1.25)$, and (*v*) scaling gains of $\mathbf{PID}_{\boldsymbol{p}}$, $\mathbf{PID}_{\boldsymbol{v}}$ $(0.33-1.25)$ in the position feedback loop.

The high thrust value in the case of the old battery indicates that motors need more power to generate a thrust equivalent to the new battery because the old battery is incapable of supplying as much power as supplied by the newer one for the same amount of thrust. TMAF handles this issue easily via microstepped thrust updates (Eq. 4), enabling robustness in TMDC against novel unmodeled situations.

F. Loop Rate Scaling and Non-uniform Loop Rates

Fig. 12 shows the analysis. Our observations align with our claims that TMDC can sustain with low frequency and non-uniform execution rates. We have plotted the average value of the experiment conducted in a range of values (Fig. 12).

G. Effect of Gain Tuning Variance

To analyze the effect of gains, we vary the parameters of TMAF i.e. α and β, and observe the hovering performance. We see that TMAF is robust to the large variance in its gains, establishing that despite the cascaded control structure, TMDC need not be tuned perfectly. See Fig. 12.

H. Payload Conditions

1) Static Off-Center Payload: Fig. 8d shows this experiment. It must be noted that TMDC was tuned without the arm, which weighs 900gm (36% of the UAV weight). It verifies TMDC sustaining large payload variance.

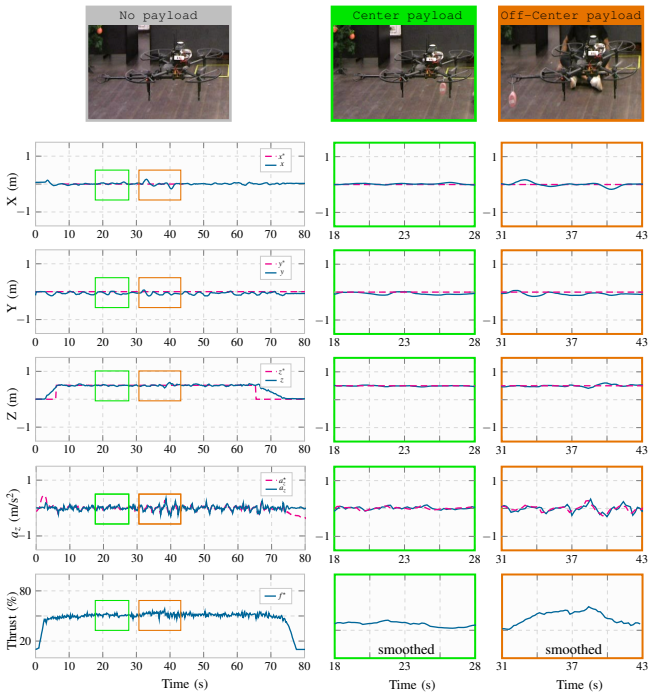

Figure 13: Takeoff-Hover-Land performance of TMDC with dynamic payload. The insets show payload attach and detach duration.

2) Dynamic Payload: Handling large and unknown dynamic payloads is the major accomplishment of TMDC. The Fig. 13 shows the results. Noticeably, the UAV hovers with high accuracy even after attaching the payloads dynamically.

With a center payload (700 gm) and the gripper, the UAV's weight increases by 64%, which is quite a large variance. Nonetheless, TMDC handles this scenario by increasing the thrust automatically, thanks to the TMAF thrust controller.

In contrast, the effect of off-center payload is much more intense since CoG gets shifted, causing sudden unwanted motions in $\boldsymbol{x}_W$ at the attach and detach events. However, the TMAF and DMC suppress them precisely. Payload analysis with the state-of-the-art controllers is discussed next.

I. TMAF vs DA

Due to the code's unavailability, we implement DA [8]. Both TMAF and DA run at the same frequency (Sec. V) and are tuned to exhibit their best performance. We analyze the hovering performance of these two controllers in many practical and challenging scenarios, as discussed below.

1) Center and Off-center Dynamic Payloads: We compare our TMAF+DMC with DA and Geometric Tracking (GT) [13] when the gripper is attached to the UAV. We use two weights w1 and w2, and attach them in five different settings (Fig. 14).

It can be noticed that DA struggles to maintain height during payload attach and detach events. Moreover, it gets severely disturbed in the case of off-center payload, as seen in the x-axis performance. Now we use GT in our TMDC, which also has severe displacements in the x-axis. On the other hand, our TMDC with the proposed decoupled motion control DMC accurately outputs roll and pitch angles based on the system behaviour in the lateral directions during disturbance to handle

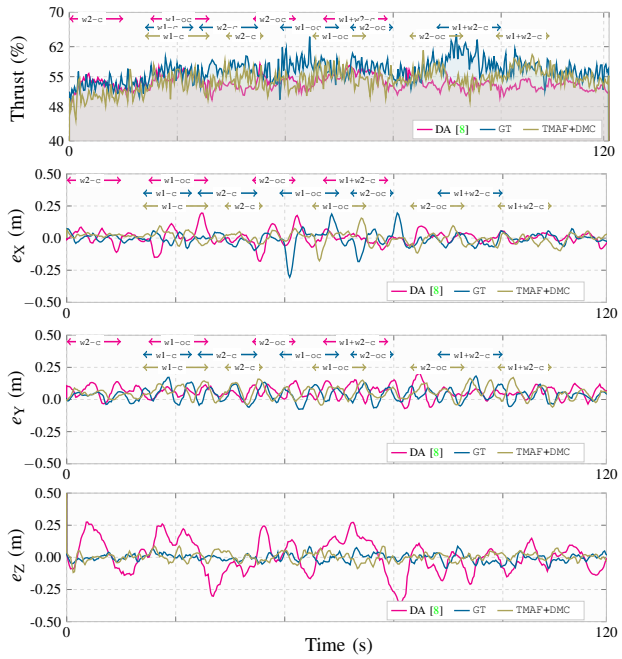

Figure 14: TMAF+DMC *vs* DA [8] and Geometric Tracking [13] at dynamic center and off-center payloads. The arrows '↔', '↔', '↔' indicates payload attach to detach duration. w1=700gm, w2=250gm, 'c' and 'oc' denotes center and off-center respectively. Metrically, DA archives an RMSE of (0.15m, 0.10m, 0.20m) and GT achieves an RMSE of (0.16m, 0.09m, 0.05m) which is quite higher than the RMSE of TMAF+DMC (0.04m, 0.05m, 0.02m).

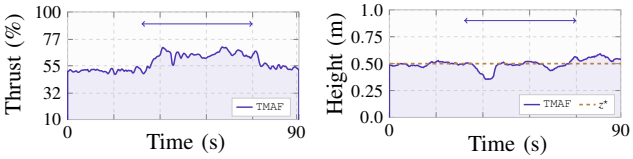

Figure 15: TMDC hovering at very heavy payload of 1.2Kg and gripper of 0.9Kg. The '↔' indicates the payload attached duration.

off-center payload, as evident from the minimal displacement in x, which justifies the importance of decoupled control.

2) Extra Heavy Payload: To go even further, we dynamically attach a heavier weight of 1.2kg (Fig. 15), which introduces a variance of 84%. Noticeably, TMAF accommodates the new weight seamlessly, as evident from the increased thrust value from $\sim$ 55% to 70%. We could easily attach more weight; however, we avoided that to prevent motor heating due to a constant higher power on the motors.

3) Hovering with Battery Discharge and Off-center Gripper: Fig. 16 shows the analysis in three different cases: (*i*) flying with 100% battery, (*ii*) flying with 50% battery, and (*i*) flying with 100% battery with off-center gripper weighing 900 gms. It can be readily noticed that DA fails to maintain height when the gripper is attached to the platform whose mass is unknown to DA, and also, as battery discharges, DA does not output the required thrust, needing to adjust its $\boldsymbol{\mu}$ (Eq. 6). While our TMAF is free from such issues.

4) Disturbance Rejection : Fig. 17 shows the analysis. We introduce a sudden intensive force of $\sim$ 15N on the quadrotor. Noticeably, DA gets disturbed from the set point with a large variance. While TMAF offers stiffness against the disturbance and maintains the height by quickly adjusting the thrust.

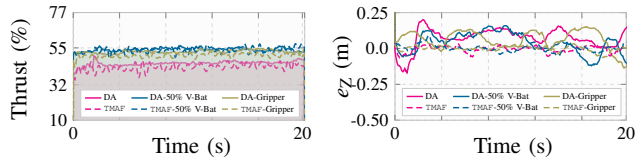

Figure 16: Hovering performance TMDC *vs* DA [8] in different settings. Without gripper, DA achieves an RMSE of 0.16m but only 0.03m with TMDC. While for 50% battery discharge, we observe RMSE of 0.15m with DA and 0.02m with TMDC. Finally, with gripper attached, DA shows an RMSE of 0.15m and TMDC of 0.02m.

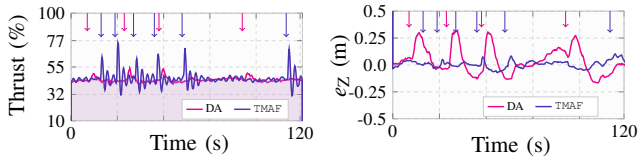

Figure 17: TMDC *vs* DA [8] at an external disturbance of 15N during hovering. The arrows ↓ and ↓ indicate disturbance introduction. In this case, DA shows a peak offset of 0.25m whereas TMDC a peak offset only of 0.07m.

5) Ground Effect: Fig. 18 shows the analysis. We keep the hovering height to 0.30m, less than the rotor diameter (0.46m). It can be seen that TMAF exhibits accurate hovering performance with near the ground by micro-adjusting the thrust, while DA does not maintain the desired height.

VII. Conclusion

In this paper, we present Thrust Microstepping and Decoupled Control (TMDC) for quadrotor targeting aerial grasping challenges, such as center and off-center dynamic payloads, battery discharge and wind drafts. TMDC has two novel components. First is Thrust Microstepping via Acceleration Feedback (TMAF) thrust controller which handles unmodeled disturbances and large variances in system mass, payloads, even at low-frequency loop rates. Second is Decoupled Motion Control (DMC), which decouples the motion control in the horizontal and vertical directions to precisely counteract the disturbances induced during an off-center payload attachment by directly estimating desired attitude instead of using forces. TMAF outperforms recent direct acceleration feedback thrust controller (DA), and DMC outperforms geometric tracking control for attitude estimation. Overall, TMDC exhibits stable flights in many real-world adverse situations such as constrained flying workspace, solely onboard position feedback, limited computing power, non-uniform loop rates, etc., verified via rigorous experimental analysis.

Appendix A

We prove the asymptotic stability of our thrust controller. Thus, rewriting the Eq. 1 and taking its time derivative:

$$a_B = \frac{f_B}{m} + \frac{f_e}{m} - g, \qquad \dot{a}_B = \frac{\dot{f}_B}{m} + d \tag{14}$$

where, $d = -\frac{\dot{m} f_B}{m^2} + \frac{\dot{f}_e}{m} - \frac{\dot{m} f_e}{m^2}$, $\dot{f}_B = \Delta f_B / T$ using Euler's backward difference, T is the sampling time. Hence, $\dot{a}_B$ becomes $(1/mT)\Delta f_B + d$. Using $\dot{a}_B$ and $\dot{f}_B$ into $e_a = a_B^* - a_B$ leads to:

$$\begin{aligned} \dot{e}_a &= \dot{a}_B^\star - \frac{\alpha e_a + \beta \dot{e}_a}{mT} - d = -e_a \frac{\alpha}{\beta + mT} + \dot{a}_B^\star \frac{mT}{\beta + mT} - d \frac{mT}{\beta + mT} \\ &= -k(e_a - \bar{e}_a) \end{aligned} \tag{15}$$

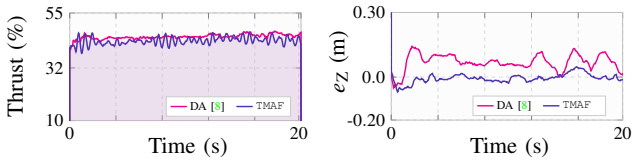

Figure 18: Hovering performance of TMDC *vs* DA [8] at ground effect. TMDC shows an RMSE offset of 0.02m and peak offset of 0.11m only in comparison to the significant offset of 0.12m with DA and peak offset of 0.30m.

where $k = \frac{\alpha}{\beta + mT}$ and $\bar{e}_a = \dot{a}_B^\star \frac{mT}{\alpha} - d\frac{mT}{\alpha}$. It can be inferred from Eq. 15 that the acceleration error e_a converges to $\bar{e}_a$. Generally, in aerial grasping, mass, external disturbance, or $\dot{a}_B^\star$ does not vary continuously and quickly after the payload attach and the detach events. Hence in the expression of $\bar{e}_a$, it can be assumed that $\dot{a}_B^\star = \dot{m} = \dot{f}_e \approx 0$. This implies $\bar{e}_a \to 0 \implies e_a \to 0$, indicating the asymptotic stability of the thrust controller. This, in turn, results in the convergence of the position feedback PID loops (Sec. IV-A) to the set point.

References

[1] B. Xian and S. Yang, "Robust tracking control of a quadrotor unmanned aerial vehicle-suspended payload system," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2653–2663, 2020.

[2] A. Kumar, M. Vohra, R. Prakash, and L. Behera, "Towards deep learning assisted autonomous UAVs for manipulation tasks in gps-denied environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1613–1620, IEEE, 2020.

[3] R. Mahony, V. Kumar, and P. Corke, "Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine, September 2012*.

[4] P. Foehn and D. Scaramuzza, "Onboard state dependent LQR for agile quadrotors," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6566–6572, IEEE, 2018.

[5] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, precision, and payloads: Adaptive nonlinear MPC for quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2021.

[6] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear MPC for quadrotor fault-tolerant control," *IEEE Robotics and Autom. Letters*, 2022.

[7] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2020.

[8] M. Hamandi, M. Tognon, and A. Franchi, "Direct acceleration feedback control of quadrotor aerial vehicles," in *2020 IEEE International Conf. on Robotics and Automation (ICRA)*, pp. 5335–5341, IEEE, 2020.

[9] P. Kotaru, G. Wu, and K. Sreenath, "Dynamics and control of a quadrotor with a payload suspended through an elastic cable," in *2017 american control conference (ACC)*, pp. 3906–3913, IEEE, 2017.

[10] V. P. Tran, F. Santoso, M. A. Garratt, and I. R. Petersen, "Adaptive second-order strictly negative imaginary controllers based on the interval type-2 fuzzy self-tuning systems for a hovering quadrotor with uncertainties," *IEEE/ASME trans. on mechatronics*, vol. 25, no. 1, pp. 11–20, 2019.

[11] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control," *Autonomous Robots*, vol. 33, no. 1, pp. 129–142, 2012.

[12] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pp. 289–302, Springer, 2013.

[13] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE conference on decision and control (CDC)*, pp. 5420–5425, IEEE, 2010.

[14] https://docs.px4.io/master/en/flight_stack/controller_diagrams.html, "Pixhawk control loops,"