2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

An Autonomous Quadrotor System for Robust High-Speed Flight Through Cluttered Environments Without GPS

Marc Rigter[1], Benjamin Morrell[1], Robert G. Reid[2], Gene B. Merewether[2], Theodore Tzanetos[2], Vinay Rajur[3], KC Wong[1], Larry H. Matthies[2]

Abstract— **Robust autonomous flight without GPS is key to many emerging drone applications, such as delivery, search and rescue, and warehouse inspection. These and other applications require accurate trajectory tracking through cluttered static environments, where GPS can be unreliable, while high-speed, agile, flight can increase efficiency. We describe the hardware and software of a quadrotor system that meets these requirements with onboard processing: a custom 300 mm wide quadrotor that uses two wide-field-of-view cameras for visual-inertial motion tracking and relocalization to a prior map. Collision-free trajectories are planned offline and tracked online with a custom tracking controller. This controller includes compensation for drag and variability in propeller performance, enabling accurate trajectory tracking, even at high speeds where aerodynamic effects are significant. We describe a system identification approach that identifies quadrotor-specific parameters via maximum likelihood estimation from flight data. Results from flight experiments are presented, which 1) validate the system identification method, 2) show that our controller with aerodynamic compensation reduces tracking error by more than 50% in both horizontal flights at up to 8.5 m/s and vertical flights at up to 3.1 m/s compared to the state-of-the-art, and 3) demonstrate our system tracking complex, aggressive, trajectories.**

I. Introduction

Quadrotor capabilities have increased rapidly over the last decade, and as a result, they are being used in an increasing variety of applications. Compact drones that can take off vertically and hover while carrying sensors and operating with minimal infrastructure open the doors to many use cases such as last-mile delivery, search and rescue, and warehouse inspection and inventory checking.

Most quadrotor systems in use today require a human pilot, limiting their use in persistent, remote, or pervasive operations. There remains challenges to develop safe, reliable, and capable autonomous navigation systems, which will enable broader application of autonomous quadrotors, particularly indoors. Some of the key challenges remaining are: 1) providing accurate and robust GPS-denied localization with a low-weight, low-power, localization system, 2) accurately tracking trajectories to fly safely and at high speeds in obstacle-rich static environments, 3) integrating all navigation capabilities on one compact system.

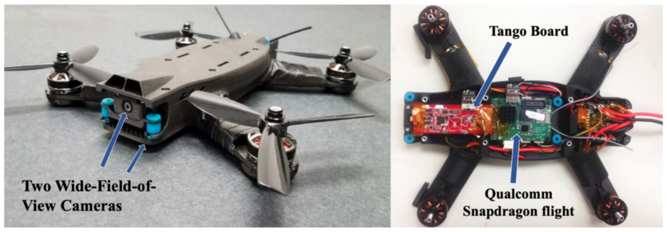


Fig. 1: Our custom 300 mm wide quadrotor designed for high-speed autonomous flight in cluttered environments without GPS. Left: Air frame without battery installed (the second camera faces downwards). Right: Internal layout showing the two embedded processing boards.

In this paper, we describe a compact (<700 g, 300 mm wide) quadrotor that navigates with a pair of fish-eye cameras: one forward-facing and another downward-facing (Fig. 1). These cameras, along with an Inertial Measurement Unit (IMU), are used for visual-inertial odometry (VIO) in unknown environments and localization in previously-mapped environments. Trajectory planning is performed offline with consideration for pre-mapped obstacles. Aerodynamic compensation is included in the controller for accurate trajectory tracking at high speeds. Parameter identification from flight data is used to estimate model parameters for the controller.

Contributions include 1) our quadrotor system design, 2) a method for estimating model parameters from flight data across several flights, and 3) our custom trajectory tracking controller with compensation for aerodynamic parameters. The system is demonstrated in a set of over 100 flight tests, a subset of which are presented here, with accelerations up to 10.8 m/s^2 and speeds of up to 10.5 m/s indoors. Our system is demonstrated in this video: `https://youtu.be/pdtRcDSvK04` while further flight experiments are described in [1] and [2].

II. Related Work

This review is focused on quadrotor systems capable of autonomous flight near obstacles, and control architectures with aerodynamic compensation for high-speed flight. It also

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and was sponsored in part by Google.

[1]M. Rigter, B. Morrell, and K. Wong are with the School of Aerospace, Mechanical and Mechatronics Engineering at The University of Sydney, Australia (email: `mrigter@robots.ox.ac.uk`, `benjamin.morrell@jpl.nasa.gov`, `kc.wong@sydney.edu.au`).

[2]R. Reid, G. Merewether, T. Tzanetos, and L. Matthies are with the Jet Propulsion Laboratory, California Institute of Technology, USA (email: `rob@reid.ai`, {`gene.b.merewether`, `theodore.tzanetos`, `lhm`} `@jpl.nasa.gov`).

[3]V. Rajur is with the College of Computing, Georgia Institute of Technology, USA (email: `vrajur3@gatech.edu`).

presents a brief overview of quadrotor system identification methods.

A. Quadrotor Systems

In previous work, autonomous navigation without GPS has been demonstrated with cameras and IMUs utilizing Visual-Inertial Odometry (VIO) algorithms [3], [4], [5], [6], [7], [8], [9], lidar odometry [10] and a combination of visual, inertial and lidar odometry [11].

A forward-facing stereo camera pair is the most common sensor configuration used for VIO [4], [6], [5], [8], because they are passive, relatively lightweight and require minimal power. Stereo pairs are occasionally combined with a downward facing camera providing velocity estimates from optical flow [12]. Lidar scanners grant exceptional capability but require additional payload capacities to carry the sensor (typically 0.3-0.7 kg) sensor [11], [10].

1) Computational Architecture: Computational architectures generally have a two processor setup, with a dedicated flight computer, such as the Pixhawk (used by [3], [7], [5]), and a CPU to run all autonomy algorithms, such as an Intel NUC i7 (used by [4], [5], [6], [8], [10], [11], [13]), or an Odroid (used by [3], [7]). These highly capable processors enable mapping and planning algorithms to be run onboard the quadrotor, a key requirement for navigation in an unknown environment.

2) Size and Weight: With more sensors, and large computers onboard, the quadrotors become heavy, with systems well over 1 kg, and over 500 mm wide [3], [4], [5], [6], [8], [12], [10], [11]. A recent exception is a 950 g system that won the 2018 IROS drone race [9]. The system we describe here is under 700 g, with a width of 300 mm, and uses only a pair of cameras and IMUs as sensors. We use a similar, two-processor architecture as described above. However, we use a smartphone-class primary processor, and an additional dedicated processor for VIO and localization. Our architecture results in a compact and lightweight system.

3) Speed: While many of systems without GPS described in this section are capable of online planning and mapping, they fly at slow speeds, up to a maximum of 1.5 m/s. For higher-speed flight without GPS, robust VIO is a key component, as evident in [4] and [5], which demonstrate flights of up to 15 m/s in large open spaces. Lidar odometry can enable comparable speeds around obstacles, with 7.8 m/s and 10 m/s shown in [10] and [11] respectively for industrial and forest environments. Our system can achieve similar speeds (10.5 m/s), flying near obstacles, with a robust VIO system and an accurate trajectory tracking controller.

B. Control

To track a planned trajectory, a hierarchical control architecture is typical, with an outer loop position controller, giving a desired thrust vector (magnitude and orientation), and an inner loop attitude controller that computes the commanded moments to achieve the desired orientation. The controller by Lee [14] is widely implemented with strong results [15], [16].

1) Aerodynamic Modelling and Control: To compute the motor outputs required to achieve the desired force and torque, most controllers assume a simplified dynamics model. This model assumes that the quadrotor is near hover [15], [17]. Under this assumption, the thrust is proportional to the square of propeller angular velocity, and drag is neglected. As speeds increase, this model becomes less accurate as secondary aerodynamic effects become significant, and the performance of the controller deteriorates [18]. Recent work has started to consider compensation for these effects.

Drag effects on a quadrotor can be divided into two main components: drag on the propellers, and parasitic drag on the airframe. It has been shown that the propeller drag effect can be approximately modeled as linear, with one lumped parameter [17]. Improvements to horizontal controller tracking have been demonstrated by compensating for this during flight of up to 4 m/s [19], [20]. A further assumption made by Faessler et al. [18] to model propeller drag is that the thrust is constant. However, this means that the model is not suitable for aggressive flight where thrust may vary substantially.

All of these studies neglect parasitic drag in the context of quadrotors, under the assumption that at low speeds it is insignificant compared with the propeller drag [21]. In contrast, we separately model propeller and parasitic drag and compensate for both effects in our controller. We demonstrate that the relevant parameters can be consistently estimated from flight data at speeds of up to 10.5 m/s.

More elaborate models for thrust are based on Blade Element Momentum (BEM) theory. Svacha et al. [20] used the BEM thrust model in their controller and noted improved vertical tracking. However, there was no change in horizontal tracking, likely because the maximum test speed was only 4 m/s, where the effect on horizontal tracking is minimal. We observe tracking improvements during horizontal flight at up to 8.5 m/s.

2) System Identification: The dynamic models discussed previously contain many parameters which must be estimated for a given system. Estimating these parameters from flight data is the preferred option as this reduces human workload and the need for test equipment which may inaccurately reproduce flight conditions. A preference is taken here for batch estimation over filtering, to provide sufficient data to observe all parameters of interest, and for time-domain methods over frequency-domain methods, to work with a non-linear model of quadrotor dynamics.

Prediction-error system identification methods minimize the error between the actual measurements and the measurements predicted by the model for a batch of data. An example is least-squares estimation, which has been used successfully for quadrotor parameter identification [20]. Burri et al. [22] estimate quadrotor model parameters with maximum likelihood estimation which provides a principled approach to account for measurement uncertainty. We take a similar approach, but expand on their work by accounting for additional aerodynamic effects and testing under high-speed flight conditions.

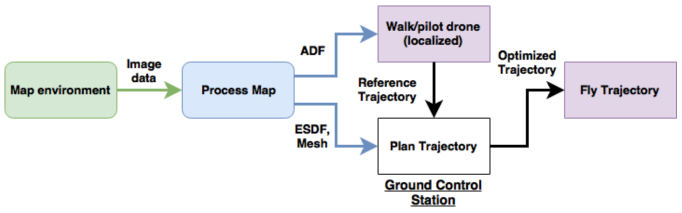


Fig. 2: Concept of operations. Purple boxes are processes running on the quadrotor.

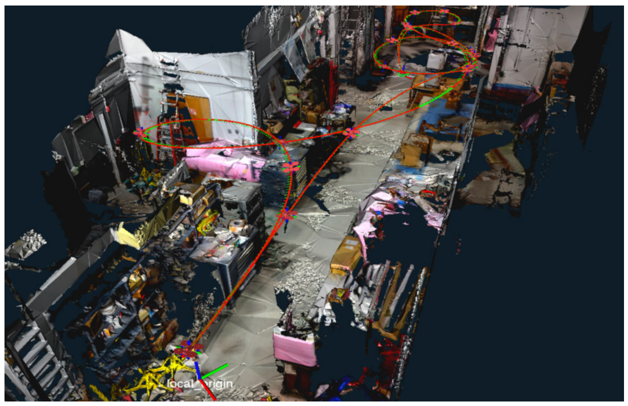


Fig. 3: Ground Control Station trajectory visualization.

III. System Overview

We present an autonomous quadrotor system capable of high-speed flight near obstacles in a static environment. The concept of operations for the system is outlined in Fig. 2.

The first step is to build a 3D map of the environment, collecting data with either the drone's dual wide field of view cameras, or an RGBD camera such as the Asus Zenphone Tango device. This data is processed offline to generate several representations of the environment: an Area Descriptor File (ADF) for localization, a Euclidean Signed Distance Field (ESDF) [23] for obstacle representation, and a textured 3D mesh for visualization. With the ADF uploaded onto the quadrotor, localization algorithms can be run, giving accurate state information on the quadrotor. The quadrotor can then be piloted, or hand-walked around a desired reference-trajectory, while the positions are recorded to give a dense set of waypoints. On the Ground Control Station (GCS), the reference trajectory is simplified to get a small set of waypoints, using the Ramer-Douglas-Peucker algorithm [24], through which a trajectory is optimized.

The ESDF obstacle representation is used in the trajectory optimization to produce collision-free trajectories. The trajectory planner is based on the work of Bry [16]. For planner details, and comparison with other planners, refer to [1].

A 3D graphical user interface on the GCS allows the operator to edit waypoints and edit the ESDF, using the python bindings for the open source Voxblox library [23]. The interface allows the operator to rapidly iterate planning and checking of the trajectory, visualized with the 3D Mesh (see Fig. 3). Once the operator is satisfied with the trajectory, it is sent to the quadrotor, with feedback on the position being sent back to be displayed on the GCS. The code for the GCS is open sourced and available here: `https://github.com/genemerewether/torq`.

For the remainder of this document, the following conventions are used. Equations assume a quadrotor "X" configuration, where the distance between adjacent propeller hubs is $2l$ as illustrated in Fig. 4. The coordinate frame convention used is x-forward, y-left, z-up. The prescripts B and G signify whether vectors are expressed in the body, or global frame, if not clear from context. For example, ${}_B\vec{r}$ is a vector expressed in the body frame. Unit vectors in the directions of the axes of a coordinate frame are denoted as $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$. The orientation of the quadrotor is described by the rotation matrix, $R_{GB} = R$ such that ${}_G\vec{r} = R\,{}_B\vec{r}$.

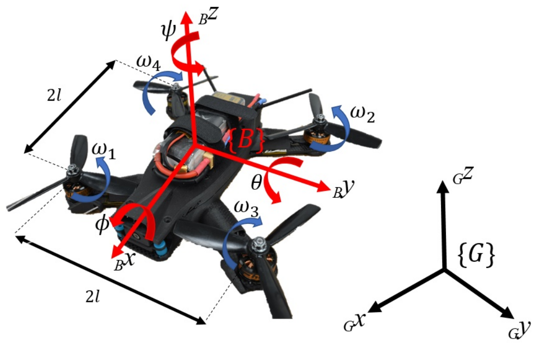


Fig. 4: Free body diagram of a quadrotor illustrating the conventions used in this paper.

A high-level overview of the system will first be presented, before key subsystems are explained in more detail.

A. High Level Architecture

Fig. 5 outlines the key components of the system. The quadrotor has onboard processors and sensors to perform localization, state estimation, control, and interfacing with the GCS. Trajectories are planned on the GCS and sent to the quadrotor over Wi-Fi with ROS messages. The same link is used to send back state information from the localization module to be displayed on the GCS.

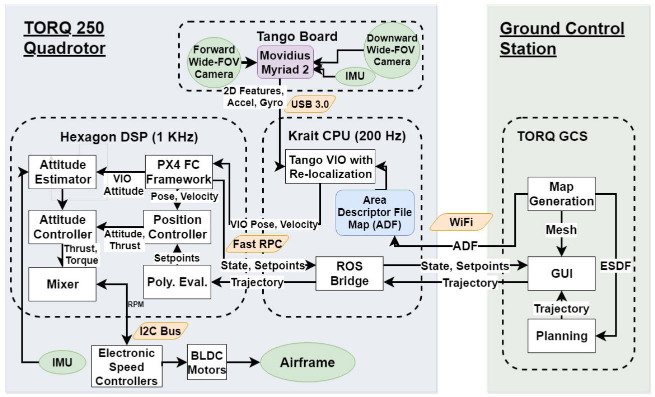


Fig. 5: High Level System Architecture.

The quadrotor was built with a carbon-fiber base airframe measuring 177mm between adjacent hubs and with an "X" configuration. The cameras and all processors are mounted on a vibration-isolated electronics sled (see Fig. 1). The electronics sled, custom cowling, and camera mounts were 3D printed with carbon-fiber infused filament.

B. Onboard Computing

At the core of the real-time computing architecture is the Qualcomm Snapdragon 801 Flight board [25], as shown in Fig. 5. The Snapdragon Flight incorporates a quad-core Krait CPU running at 2.26 GHz and a Hexagon Digital Signal Processor (DSP) core running 801 MHz.

The CPU runs the Linaro Linux distribution and serves as the primary interface with the quadrotor: providing connection to the GCS over a Wi-Fi link, and communicating with the Tango Board over USB 3.0. The PX4 [26] flight software is distributed between the CPU and DSP, with all the real-time control loops running at 1 kHz on the DSP; these include position and attitude controllers, thrust mixers, set-point evaluation and attitude estimators.

The Tango Board provides precise timestamping of the dual camera images and IMU data, and performs front-end VIO processing on a Movidius processor. The VIO algorithm [27], [28] runs in soft-realtime on the CPU and sends pose and velocity estimates to the DSP at 200 Hz over the uORB protocol. The DSP fuses this VIO information with measurements from a second onboard IMU to provide real-time attitude estimation at 1 kHz.

C. Dynamic Model

To implement a model-based controller which works well at high speeds, a model of the quadrotor incorporating aerodynamics is required. The state of the quadrotor, $\mathbf{x}$, is defined by the position, velocity, atttitude (unit quaternion), and angular velocity: $\mathbf{x} = [{}_G\vec{x}, {}_G\vec{v}, q_{GB}, {}_B\vec{\Omega}]^T$.

The thrust produced by a propeller is modeled by the BEM theory model for propeller thrust. Under the assumption that the horizontal advance ratio is small as in [29], the BEM equations reduce to

$$T_i = k_1\omega_i^2 + k_2\omega_i({}_Bv_z + v_{ind}), \tag{1}$$

$$T_i = 2\rho A v_{ind}\sqrt{({}_Bv_z + v_{ind})^2 + {}_Bv_{xy}^2}, \tag{2}$$

where k_1 and k_2 are parameters related to the propeller geometry, ρ is the air density, A is the propeller disk area, v_{ind} is the velocity induced on the air normal to the propeller plane, and ${}_Bv_{xy}$ is the is the component of velocity in the x-y body plane. The angular velocity and thrust produced by the i^{th} propeller are ω_i and T_i. The moments on the vehicle are

$$\begin{bmatrix} {}_B\tau_x \\ {}_B\tau_y \\ {}_B\tau_z \end{bmatrix} = \begin{bmatrix} -c_{t,1}(l+\Delta y) & c_{t,2}(l-\Delta y) & c_{t,3}(l-\Delta y) & -c_{t,4}(l+\Delta y) \\ -c_{t,1}(l-\Delta x) & c_{t,2}(l+\Delta x) & -c_{t,3}(l-\Delta x) & c_{t,4}(l+\Delta x) \\ -c_q & -c_q & c_q & c_q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \tag{3}$$

where Δx and Δy are offsets of the centre of gravity from the geometric centre of the quadrotor, $c_{t,i} = \frac{T_i}{\omega_i^2}$ is a dynamic thrust coefficient, and c_q is a constant torque coefficient. A centre of mass offset in the z direction is not included in the model because this does not influence the torque calculated. Propeller drag, ${}_B\vec{d}_h$, is characterized by a lumped coefficient, $c_{d,h}$, according to the standard simplified model [17]:

$${}_B\vec{d}_h = -\Big(\sum_{i=1}^{4} T_i\Big)\begin{bmatrix} c_{d,h} & 0 & 0 \\ 0 & c_{d,h} & 0 \\ 0 & 0 & 0 \end{bmatrix} {}_B\vec{v}. \tag{4}$$

We also incorporate parasitic drag on the airframe into the model which we parameterize with two drag coefficients, $c_{d,xy}$ and $c_{d,z}$:

$${}_B\vec{d}_p = -\begin{bmatrix} c_{d,xy} & 0 & 0 \\ 0 & c_{d,xy} & 0 \\ 0 & 0 & c_{d,z} \end{bmatrix} \begin{bmatrix} {}_Bv_x|{}_Bv_x| \\ {}_Bv_y|{}_Bv_y| \\ {}_Bv_z|{}_Bv_z| \end{bmatrix}. \tag{5}$$

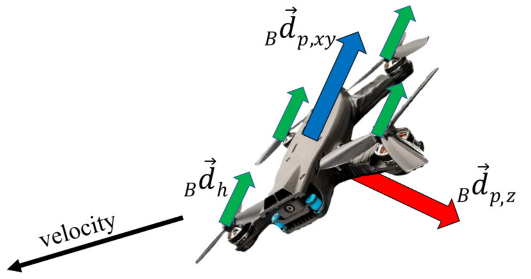


Fig. 6: Visual representation of parameterized model of drag effects on a quadrotor.

Two separate parasitic drag coefficients are used, as parasitic drag is proportional to surface area, and a quadrotor typically has a much larger cross-sectional area perpendicular to the z axis than the x and y axes. The drag effects modeled are represented by the diagram in Fig. 6. The quadrotor is considered to be a rigid body, with translational dynamics governed by

$$m\,{}_G\dot{\vec{v}} = -mg\,{}_G\vec{e}_3 + R\Big(\sum_{i=1}^{4} T_i\Big)\,{}_B\vec{e}_3 + R\,{}_B\vec{d}_p + R\,{}_B\vec{d}_h. \tag{6}$$

The gyroscopic effect of the propellers is neglected so that the rotational dynamics are described by

$$\mathbf{I}\,{}_B\dot{\vec{\Omega}} = {}_B\vec{\tau} - {}_B\dot{\vec{\Omega}} \times \mathbf{I}\,{}_B\dot{\vec{\Omega}}, \tag{7}$$

$$\dot{q_{GB}} = \frac{1}{2} q_{GB} \otimes q_{\vec{\Omega}}, \tag{8}$$

where $\otimes$ denotes quaternion multiplication, $q_{\vec{\Omega}}$ is a quaternion representing the angular velocity, and the inertia matrix, $\mathbf{I}$, is diagonal under the assumption that the quadrotor is symmetric.

D. Control

The control software builds from the PX4 flight software, with modifications to the controller for high-speed tracking of trajectories, aerodynamic considerations, and to incorporate SI units for the quadrotor mass and inertia for intuitive controller tuning. The control architecture is outlined in Fig. 5, starting with a polynomial evaluator giving set-points to the outer position controller.

The position and attitude controllers are modified from Lee et al. [14], with the changes outlined below.

1) Position Controller: Position is controlled by a proportional-derivative (PD) controller, which acts on the error between the desired and actual position and velocity. The desired acceleration and compensation for drag are also fed-forward:

$$\begin{aligned} {}_G\vec{T}_{sp} = -K_p(\vec{x} - \vec{x}_{sp}) - K_d(\dot{\vec{x}} - \dot{\vec{x}}_{sp}) \\ + mg\vec{e}_3 + m\ddot{\vec{x}}_{sp} - R\,{}_B\vec{d}_p - R\,{}_B\vec{d}_h. \end{aligned} \quad (9)$$

The vectors are expressed in the global frame where not specified, and the sp subscript denotes the desired set-points. The K terms are gains, m is the quadrotor mass, and $\vec{e}_3$ is a unit vector aligned with the global z axis. The output from the position controller, ${}_G\vec{T}_{sp}$, is the desired thrust vector.

The desired thrust vector orientation and a yaw set-point are passed through the standard differential flatness transform [15] to produce a quaternion set-point. For an analysis of different differential transform methods, and their suitability for aggressive flight, refer to [2].

2) Attitude Controller: The attitude controller follows Lee et al. [14], but without feed-forward terms to produce the desired torque, ${}_B\vec{\tau}_{sp}$. Hence, the attitude controller is a PD controller acting on the angular error, and angular rate error.

3) Thrust mixer: The thrust mixer computes the angular velocity for each motor, ω_i, to deliver the desired thrust magnitude and torque ($|{}_G\vec{T}_{sp}|$ and ${}_B\vec{\tau}_{sp}$). To perform the mixing step, (1), (2), and (3) are solved iteratively to compute ω_i such that the desired total thrust and torque are delivered. Solving (1) and (2) online is computationally expensive, so in practice a lookup table of $c_t({}_Bv_z, \omega)$ is computed offline. An initial guess is made for ω_i, and the appropriate $c_{t,i}$ are retrieved from the lookup table. This allows (3) to be solved for an updated ω_i. After repeating this process, the solution converges in a few iterations to the appropriate propeller commands which are sent to the motors. We do not account for propeller latency and assume that the motor response is instantaneous.

4) Actuation: The quadrotor is equipped with four 2300 kV Luminier brushless motors fitted with triblade propellers. Custom Electronic Speed Controllers (ESCs) are used and run high-rate RPM feedback for close tracking of the commanded RPM from the mixer.

E. Localization

Google's "Tango" Motion Tracking and Area Learning algorithms (VIO and relocalization, respectively) [27], [28] were customized and integrated into the Snapdragon Flight Board. Using timestamped camera and IMU data, VIO runs on the Tango Board CPU where it estimates the position, orientation and velocities of the quadrotor. Continuous global relocalization is performed in parallel to VIO by matching 2D visual features to the ADF. The system degrades gracefully to an "open loop" VIO-only mode when the quadrotor flies out of the ADF, however corrections to VIO drift can be observed when the quadrotor re-enters and relocalizes to the ADF after closing large loops. Refer to [27], [28] for more details on Tango's algorithms; their performance in our quadrotor system is demonstrated in Section V-A.

IV. Parameter Identification

We adapt the maximum likelihood framework of Burri et al. [22], and estimate model parameters from flight data. Per this approach, we incorporate calibration of our pose sensor into this identification process. We assume that the pose sensor has some offset from the center of the quadrotor, ${}_B\vec{r}_{BS}$, and is rotated with respect to the quadrotor by q_{BS}. Thus, the vector of unknown parameters which must be estimated is:

$$\theta = [I_{xx}, I_{yy}, I_{zz}, C_q, \Delta x, \Delta y, k_1, k_2, c_{d,h}, c_{d,xy}, c_{d,z}, {}_B\vec{r}_{BS}, q_{BS}]^T.$$

In this framework, the state estimate at each time step, $\hat{\mathbf{x}}_\mathbf{k}$, is included in the vector of unknowns $\mathbf{y} = [\theta^T, \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, ..., \hat{\mathbf{x}}_m]^T$. We find

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \mathbf{r}(\mathbf{y})^T \mathbf{W} \mathbf{r}(\mathbf{y}), \quad (10)$$

where $\mathbf{r}(\mathbf{y})$ is a vector of residuals, and $\mathbf{W}$ is the inverse of the covariance of the residuals. There are two types of residuals included in the vector, $\mathbf{r}(\mathbf{y})$. Measurement residuals are computed by: $\mathbf{r}_{z,k} = \mathbf{z}_k - \mathbb{E}[\mathbf{z}_k]$, where $\mathbf{z}_k$ is the actual pose measurement from the VIO system and $\mathbb{E}[\mathbf{z}_k]$ is the expected measurement given the pose estimate, $\hat{\mathbf{x}}_k$, sensor calibration from θ, and measurement noise, $\mathbf{w}_z$. The model residuals are given by: $\mathbf{r}_{x,k} = \hat{\mathbf{x}}_k - \mathbb{E}[\mathbf{x}_k]$, where $\mathbb{E}[\mathbf{x}_k]$ is the expected state at time t_k given $\hat{\mathbf{x}}_{k-1}$ and integrating the model dynamics with process noise, $\mathbf{w}_x$, from t_{k-1} to t_k using propeller velocities measured by the ESCs. Gaussian process and measurement noise are assumed. The covariance of $\mathbf{x}_k$ and $\mathbf{z}_k$ under the noise assumption is calculated, and this is equal to the covariance of the corresponding residuals. For further details, refer to [22]. Equation 10 is minimized via gradient descent, resulting in a maximum likelihood estimate of the state and the unknown parameters. An estimate of the uncertainty in the parameter estimates is calculated using the standard error estimator from [30].

A. Multi-stage Approach

To attain suitable flight data to estimate all parameters, all of the dynamics of the quadrotor must be persistently excited. The trajectory must include high linear speeds to observe drag, and high angular acceleration to estimate inertia. Flying a single trajectory which achieves this is difficult in practice, particularly with limited space. As an alternative, three separate trajectories are flown with each activating separate aspects of the quadrotor dynamics. In each flight, only the active parameters are estimated. The three different trajectories used are:

- Flight 1: Vertical flight
- Flight 2: High-speed horizontal flight
- Flight 3: High angular-acceleration flight.

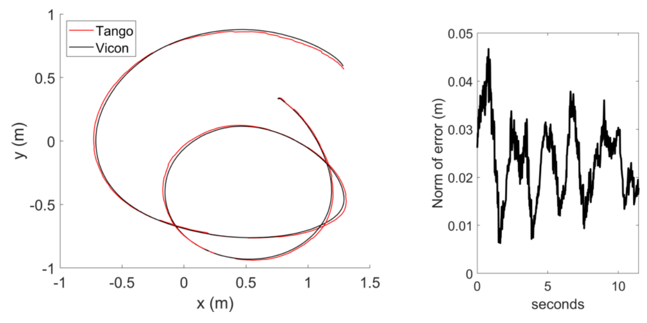
Unknown parameters are initialized coarsely to have an appropriate order of magnitude. During Flight 1, the thrust model is estimated under the assumption that airframe parasitic drag is negligible at the modest vertical speeds. In Flight 2, the drag parameters are estimated. In Flights 1 and 2, there is very little rotation, so the coarsely initialized inertia and torque coefficient have little effect. The high angular acceleration of Flight 3 is used to estimate the inertia and the torque coefficient. The center of mass offset is always observable and may change; therefore it is re-estimated in each flight.

V. Results

We evaluate our system with a series of flight experiments. First, the robustness of the visual localization system is verified by a comparison to pose measurements from the Vicon motion capture system. It is then shown that the model parameters can be consistently estimated from flight data. The controller performance is evaluated, and we show that compensating for aerodynamic effects results in significant improvement in trajectory tracking at high speed. Finally, the system is tested on an aggressive indoor flight with multiple maneuvers to demonstrate that the system is capable of robustly tracking aggressive trajectories.

A. Localization Testing

To assess the localization performance of the VIO system, we perform a flight in a Vicon motion capture system, assuming that the Vicon measurement is ground truth (see Fig. 7a). The mean absolute error of the position estimate during the flight was 23.3 mm relative to the Vicon measurement (Fig. 7b). This localization performance is sufficiently accurate for indoor flight within cluttered environments.



(a) Position estimate (b) Tracking error

Fig. 7: Tango position tracking performance. a) Tango position estimate plotted with the Vicon ground truth. b) Localization error (mean of 23.3 mm for this flight).

B. Parameter Identification

Flight 1 of the parameter identification flights consisted of vertical flight at up to ±3.5 m/s. Flight 2 consisted of horizontal flight at up to 10.5 m/s. Flight 3 consisted of simultaneous rotations about all three axes at up to 6.5 rad/s. The results of the parameter estimation process are summarized in Table I, and compared to ground truth measurements where possible.

The ground truth inertia is from a CAD model of the quadrotor containing the masses of each component. The ground truth torque coefficient of the propellers is measured on a static test stand. The true sensor offset in the x and y directions was measured directly.

TABLE I: Parameter identification results.

Parameter	Estimate	Std. dev.	Ground Truth
I_{xx} (kgm^2)	0.00306	0.00017	0.00241
I_{yy} (kgm^2)	0.00442	0.00023	0.00342
I_{zz} (kgm^2)	0.00441	0.00031	0.00522
Δx (m)	0.00140	0.00030	-
Δy (m)	1.77×10^{-4}	1.7×10^{-4}	-
c_q (Nms^2)	3.40×10^{-10}	4.9×10^{-11}	2.93×10^{-10}
k_1	3.42×10^{-8}	1.8×10^{-9}	-
k_2	-1.62×10^{-5}	2.0×10^{-6}	-
$c_{d,h}$ (sm^{-1})	0.0175	0.0037	-
$c_{d,xy}$ (Ns^{2}m^{-2})	0.0055	0.0033	-
$c_{d,z}$ (Ns^{2}m^{-2})	0.0185	0.0067	-
$q_{BS,0}$	0.9975	0.00040	-
$q_{BS,1}$	-0.0241	0.0031	-
$q_{BS,2}$	-0.0122	0.0023	-
$q_{BS,3}$	0.0646	0.0062	-
$r_{BS,x}$ (m)	0.0901	0.0021	0.088
$r_{BS,y}$ (m)	0.00630	0.0025	0.004
$r_{BS,z}$ (m)	0.0142	0.0028	-

A thrust stand is used to measure the static thrust coefficient. We compare this with the static thrust coefficient predicted by the identified k_1 and k_2 parameters and find that the value measured on the thrust stand is 8% higher. This result is likely because in real flight tests the struts holding each of the motors obstructs airflow behind the propellers, reducing thrust. In the test stand setup, these struts are not present. These observations emphasize one of the advantages of estimating parameters from flight data: it is difficult to replicate flight conditions in a static test setup.

Results from additional experiments, detailed in [31], verified that the center of mass, and sensor offsets are estimated consistently and accurately. The drag coefficient estimates could not be verified in the absence of wind tunnel testing, but are of the expected order of magnitude. Notably, the parasitic drag coefficient acting along the z body axis is approximately three times greater than that acting in the $x-y$ body plane. This result is expected due to the greater surface area of the quadrotor perpendicular to the z axis.

To illustrate the consistency of parameter identification results, the estimates of $c_{d,z}$, the z parasitic drag coefficient, are plotted in Fig. 8 over eight separate trials of the parameter identification process. The variability of the parameter estimates is consistent with the uncertainty estimates.

The x and y inertias were consistently over-estimated by

20-30%, similar to the results of Burri et al.. The authors of the previous research attribute this to incorrect acceleration measurements due to damping of the sensor module. To the contrary, we suggest that the explanation is likely unmodeled aerodynamic torques. In flight-test results not presented here but detailed in [31], we observed that the torque model does not generalize well to all maneuvers, indicating that unmodeled torque effects are acting on the quadrotor. Investigating a more elaborate torque model could be a direction for future research. Another direction could be to use machine learning techniques to model the dynamics without strictly imposing the form of the model.

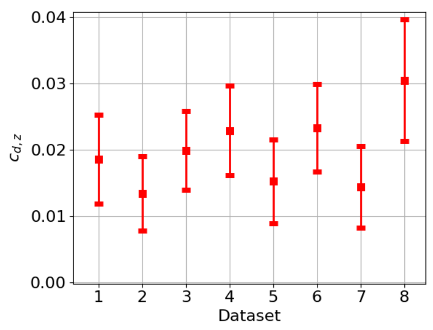


Fig. 8: Parasitic drag coefficient $c_{d,z}$ estimates over eight trials. Error bars are $1-\sigma$ parameter uncertainty estimates.

C. Controller Tracking

We test several versions of the controller to investigate the efficacy of the refinements made. The PD with feed-forward controller with no drag compensation and a static thrust coefficient is denoted $PD\ FF$. The controller incorporating the drag compensation is denoted $PD\ FF + DRG$, and the controller incorporating drag compensation and the refined thrust model is denoted $PD\ FF + THR + DRG$. The horizontal tracking performance of each of the controller variations is assessed with multiple runs of a trajectory with a peak velocity of 8.5 m/s, and peak acceleration of 10.8 m/s^2. The trajectory and associated tracking errors for each run are plotted in Fig. 9. For the $PD\ FF$ controller, the mean absolute tracking error across all flights is 10.5 cm. This error is reduced to 6.5 cm for the $PD\ FF + DRG$ controller, and further to 5.1 cm for $PD\ FF + THR + DRG$, resulting in a 54% reduction in tracking error.

A vertical trajectory with velocities between $\pm$ 3.1 m/s is flown to assess the vertical tracking. The trajectory and tracking errors are plotted in Figure 10. For the $PD\ FF$ controller, the mean vertical tracking error across all flights is 2.76 cm. The error is reduced to 1.84 cm for the $PD\ FF + DRG$ controller, and further to 1.32 cm for $PD\ FF + THR + DRG$, resulting in a 52% reduction in tracking error.

D. General demonstration

We demonstrate the ability of our system to autonomously track a trajectory containing a variety of maneuvers with

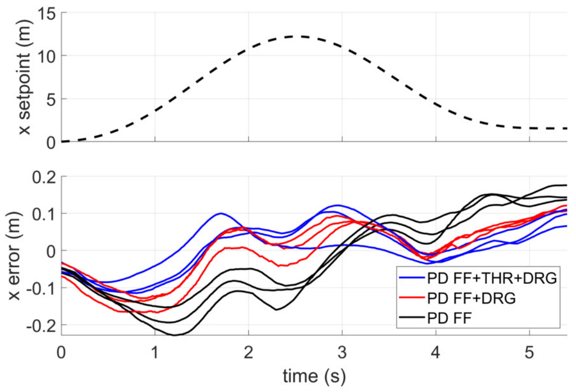


Fig. 9: Horizontal tracking error comparison between controller variants.

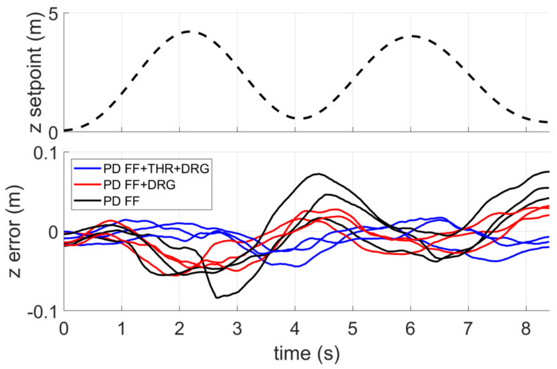
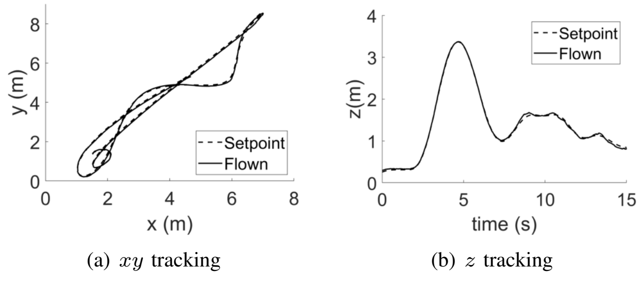


Fig. 10: Vertical tracking error comparison between controller variants.

accuracy, and at high speed. A general trajectory is flown (Fig. 11) with vertical flight up to 2 m/s, horizontal flight up to 5.7 m/s, roll and pitch rates up to 2 rad/s, and yaw rates up to 5.5 rad/s. The mean absolute tracking error during this flight is 4.9 cm, with mean vertical tracking error 2.4 cm, demonstrating that the controller can track an aggressive trajectory with a variety of maneuvers closely, suitable for flight in cluttered environments. In total, our system has been flown robustly for more than 100 flights. For more results demonstrating the robust performance of our system in indoor environments, please refer to [1].



(a) xy tracking (b) z tracking

Fig. 11: Position tracking for general trajectory.

VI. Conclusion

We have presented the system design of a compact quadrotor system, under 700 g and 300 mm wide, that is capable of autonomous flight at speeds of up to 10.5 m/s, and accelerations of 10.8 m/s^2, in a pre-mapped static environment. Localization is performed with two monochrome fisheye cameras, one front-facing, and one downwards-facing, with a dedicated processor and IMU running the Tango VIO code. Custom ESCs are used to implement RPM feedback control of the motors. Trajectories are planned offline in a 3D Ground Control station and are sent to the quadrotor to track. Accurate trajectory tracking at high speeds is achieved by including drag compensation and variable propeller performance in the controller. A multi-stage maximum likelihood parameter identification method is used to estimate the values of the parameters needed for the controller. The improved controller shows more than a 50% improvement in trajectory tracking over the state-of-the-art for horizontal flights at up to 8.5 m/s and vertical flights at up to 3.1 m/s. In future work, online mapping and replanning for obstacle avoidance are required to further increase system capability.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored in part by Google. Rigter and Morrell were visiting scholars supported by The University of Sydney, Australia. The authors wish to thank Russell Smith, Dillon Azzam and Thomas Paynter for their support.

References

[1] B. Morrell, R. Thakker, G. Merewether, R. Reid, M. Rigter, T. Tzanetos, and G. Chamitoff, "Comparison of trajectory optimization algorithms for high-speed quadrotor flight near obstacles," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4399–4406, 2018.

[2] B. Morrell, M. Rigter, G. Merewether, R. Reid, R. Thakker, T. Tzanetos, V. Rajur, and G. Chamitoff, "Differential flatness transformations for aggressive quadrotor flight," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on Robotics and Automation*, no. 1817. Brisbane, Australia: IEEE, 2018, pp. 5204–5210.

[3] Z. Fang, S. Yang, S. Jain, G. Dubey, S. Roth, S. Maeta, S. Nuske, Y. Zhang, and S. Scherer, "Robust autonomous flight in constrained and visually degraded shipboard environments," *Journal of Field Robotics*, vol. 34, no. 1, pp. 25–52, 2017.

[4] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *arXiv preprint arXiv:1712.00036*, 2017.

[5] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in gps-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[6] R. Ait-Jellal and A. Zell, "Outdoor obstacle avoidance based on hybrid visual stereo slam for an autonomous quadrotor mav," in *2017 European Conference on Mobile Robots (ECMR)*, Sept 2017, pp. 1–8.

[7] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1722–1729.

[8] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online UAV replanning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 5332–5339.

[9] E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, "Deep drone racing: Learning agile flight in dynamic environments," *arXiv preprint arXiv:1806.08548*, 2018.

[10] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, "Fast autonomous flight in warehouses for inventory applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3121–3128, 2018.

[11] J. Zhang, R. G. Chadha, V. Velivela, and S. Singh, "P-cap: Pre-computed alternative paths to enable aggressive aerial maneuvers in cluttered environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2018.

[12] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoing-based framework for the 2016 iros autonomous drone racing challenge," *Journal of Field Robotics*, vol. 35, no. 1, pp. 146–166, 2018.

[13] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles," *arXiv preprint arXiv:1710.00604*, 2017.

[14] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 5420–5425.

[15] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[16] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 969–1002, 2015.

[17] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles," *IEEE Robotics and Automation magazine*, vol. 20, no. 32, 2012.

[18] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[19] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, "Nonlinear control of vtol uavs incorporating flapping dynamics," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2419–2425.

[20] J. Svacha, K. Mohta, and V. Kumar, "Improving quadrotor trajectory tracking by compensating for aerodynamic effects," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 860–866.

[21] M. Bangura, R. Mahony *et al.*, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Australasian conference on robotics and automation*, 2012, pp. 1–10.

[22] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, "A framework for maximum likelihood parameter identification applied on MAVs," *Journal of Field Robotics*, pp. n/a–n/a, 2017.

[23] H. Oleynikova, Z. Taylor, M. Fehr, J. Nieto, and R. Siegwart, "Voxblox: Incremental 3d Euclidean Signed Distance Fields for On-Board MAV Planning," *arXiv:1611.03631 [cs]*, Nov. 2016, arXiv: 1611.03631. [Online]. Available: http://arxiv.org/abs/1611.03631

[24] J. E. Hershberger and J. Snoeyink, *Speeding up the Douglas-Peucker line-simplification algorithm*. University of British Columbia, Department of Computer Science, 1992.

[25] Qualcomm, "Snapdragon Flight 801 Processor," https://developer.qualcomm.com/hardware/snapdragon-flight, 2017, [Online].

[26] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[27] E. Marder-Eppstein, "Project tango," in *ACM SIGGRAPH 2016 Real-Time Live!*, ser. SIGGRAPH '16. New York, NY, USA: ACM, 2016, pp. 40:25–40:25. [Online]. Available: http://doi.acm.org/10.1145/2933540.2933550

[28] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 165–172.

[29] M. Bangura, "Aerodynamics and control of quadrotors," 2017.

[30] A. F. Hayes and L. Cai, "Using heteroskedasticity-consistent standard error estimators in ols regression: An introduction and software implementation," *Behavior research methods*, vol. 39, no. 4, pp. 709–722, 2007.

[31] M. Rigter, "High performance quadrotor system identification and control," Bachelor of Engineering Thesis, The University of Sydney, NSW, Australia, 2018.