# Factor Graph Optimization for Dynamic Parameter Estimation

**Montiel Abello*** mabello@andrew.cmu.edu

**Eric Dexheimer*** edexheim@andrew.cmu.edu

**Sudharshan Suresh*** sudhars1@andrew.cmu.edu

## Abstract

Parameter identification is essential for the development of accurate control algorithms and domain-specific actions. Agile flight for micro-aerial vehicles (MAVs) can be achieved by modeling the dynamics from commanded inputs to vehicle response. Since the effects on MAV control are highly nonlinear, special care must be taken to estimate the parameters. However, expensive hardware tests are not practical, and automatic estimation from flight data would greatly speed up the development process. In this paper, we implement a factor graph optimization for joint estimation of MAV poses and dynamic parameters. We demonstrate improved trajectory estimation and convergence towards ground truth parameters, but robust estimation of the dynamics remains a direction for future work.

## 1 Introduction and Related Work

Agile robot control requires accurate knowledge of vehicle dynamics. This is particularly important for Micro Aerial Vehicles (MAVs), which perform high-speed dynamic maneuvers. Identifying system parameters has twofold importance: **(i)** they may vary temporally, due to extended use and harsh environmental effects and **(ii)** obtaining the ground truth values may be difficult, for example, wind tunnel experiments are considerably expensive.

This work explores jointly estimating these parameters from a minimal set of recorded data, while also optimizing for robot pose. We address this problem by formulating parameter estimation as an inference task in a nonlinear least squares framework. Given noisy sensor data and rough initial estimates, we investigate the convergence to the dynamic parameters. Automatically solving for these parameters assists long-term autonomy and avoids extensive hardware tests.

Mahony et al. illustrate the fundamental principles and guiding dynamic models of MAVs [1]. Chowdhary et al. experiment with both the extended and unscented Kalman filters to estimate dynamic parameters for aerial vehicles [2]. In contrast, Burri et al. formulate parameter recovery as a batch nonlinear least squares problem [3]. They solve for the MAVs center of gravity offset, inertial parameters, and aerodynamic coefficients. Our work closely follows this method, instead in the context of a factor graph optimization. Recent work in this domain has developed efficient solvers such as GTSAM [4] and iSAM [5].

## 2 MAV Dynamics Model

Our dynamics model follows that of [1] and [3]. We further simplify the model by assuming that the odometry measurements are in the same frame as the center of gravity, so the offset between them does not appear in the equations. We also assume that the mass is known since this is usually easy to measure in practice. Lastly, the transformations between each rotor hub and the body frame are assumed to be known. In this section, the subscript of a vector indicates which rotor it belongs to. For the MAV model used in this project, we have $i = 1 \dots 6$.

We begin with a vector of commanded rotor angular velocities given by $\mathbf{n}$. Each rotor produces a thrust force and a moment in the z-direction. Both terms are proportional to the square of the rotor angular velocities, and scaled by a thrust and moment coefficient, respectively:

$$\mathbf{T}_i = c_T \mathbf{n}_i^2 \mathbf{e}_z \tag{1}$$

$$\mathbf{M}_{hub,i} = c_M \mathbf{n}_i^2 \mathbf{e}_z \tag{2}$$

In addition, as in [1], we model a combined induced drag and flapping blade effect. This produces a force perpendicular to the thrust force and proportional to the linear velocity of the rotor hub. Since we assume no wind, this velocity is:

$$\mathbf{v}_{hub,i} = \mathbf{R}_{A_i B}(\mathbf{v}_B + \boldsymbol{\omega}_B \times \mathbf{t}_{BA_i}) \tag{3}$$

Note that the terms discussed thus far are in the individual rotor hub frames, so they must be transformed into a common body frame. The total force on the body from one of

*The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA

the rotors is then:

$$\mathbf{F}_i = \mathbf{R}_{BA_i} \left( \mathbf{T}_i - c_T \mathbf{n}_i^2 \begin{bmatrix} c_D & 0 & 0 \\ 0 & c_D & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{v}_{hub,i} \right) \quad (4)$$

The moment on the MAV not only includes the individual transformed moments, but also the moments induced by the force from the rotors, as they are offset from the center of gravity:

$$\mathbf{M}_i = \mathbf{R}_{BA_i} \mathbf{M}_{hub,i} + \mathbf{F}_i \times \mathbf{t}_{BA_i} \quad (5)$$

The total force and moment is then the sum:

$$\mathbf{F}_{tot} = \sum_i \mathbf{F}_i \quad (6)$$

$$\mathbf{M}_{tot} = \sum_i \mathbf{M}_i \quad (7)$$

Now that we have the force and moment acting on the body, we can use the Newton-Euler equations to solve for the linear and angular accelerations:

$$\dot{\mathbf{v}}_B = \frac{1}{m} \mathbf{F}_{tot} - \mathbf{R}_{BW} \mathbf{g} - \boldsymbol{\omega}_B \times \mathbf{v}_B \quad (8)$$

$$\dot{\boldsymbol{\omega}}_B = \mathbf{I}^{-1} \left( \mathbf{M}_{tot} - \boldsymbol{\omega}_B \times \mathbf{I} \boldsymbol{\omega}_B \right) \quad (9)$$

We can use the above equations to predict the motion of the robot between two timesteps. Note that the pose of the robot in the world frame is needed, as this gives the direction of gravity in the body frame. Thus, pose is also an important variable that needs to be estimated for accurate dynamics.

In the dynamic model, we wish to estimate the thrust coefficient $c_T$, the moment coefficient $c_M$, the combined drag coefficient $c_D$, and the moment of inertia, which we assume is diagonal, $I_{xx}, I_{yy}, I_{zz}$. It is also apparent that some of the terms are strongly coupled, as the combined drag is proportional to the thrust and drag coefficients. In addition, the moment coefficient $c_M$ and $I_{zz}$ have been noted to have similar effects during flight, so these can be difficult to separate [3]. Since these equations are highly nonlinear in the poses and the dynamic parameters, we need to solve for them in a joint optimization.

# 3 Factor Graph Optimization

Previous work [2] applied unscented Kalman filters to this joint estimation problem. This windowed approach linearizes update equations using the current state estimate. While efficient, this can give inaccurate results as early measurements are integrated with incorrect linearization points and can not be updated. Modern inference problems employ nonlinear least squares optimization to utilize all information at each iteration for more accurate state estimation.

We formulate the joint optimization of vehicle trajectory and dynamic parameters as a nonlinear least squares problem. This problem can be represented with the *factor graph* shown in Figure 3.

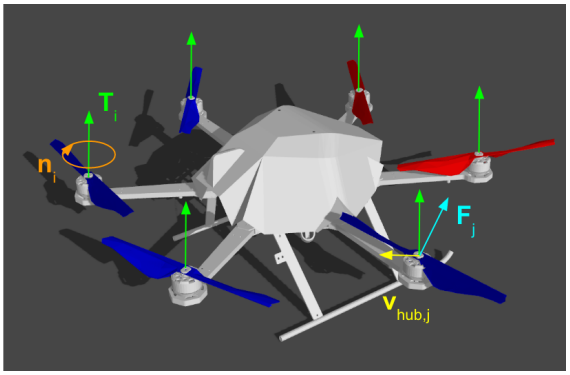


Figure 1: Illustration of rotor angular speed, thrust force, and total rotor force with combined drag effect

## 3.1 State Variables

State variables are unknown quantities we wish to estimate and form the nodes of the graph, represented with large circles. Our state consists of vehicle poses at time step $i$ represented in world coordinates, denoted $\mathbf{x}_{WB,i} \in SE(3)$, and the dynamic parameters which are represented by a vector $D \in \mathbb{R}^6$.

## 3.2 Constraints

Constraints on state variables are represented by edges in the graph. We employ *prior* constraints represented with small circles to model input information. These constraints have low uncertainty in comparison to odometry and dynamic factor constraints. The first type of prior constrains the initial vehicle pose to a known transformation $T_0 \in SE(3)$:

$$\mathbf{x}_{WB,0} = T_0 \oplus \mathbf{w}_p \quad (10)$$

$$\mathbf{w}_p \sim \mathcal{N}(0, R_p) \quad (11)$$

We model vehicle velocities and rotor angular velocities as priors also. Body velocities are represented by twists in the body coordinate frame, $\xi_{B,i} = [\boldsymbol{\omega}_{B,i}, \mathbf{v}_{B,i}] \in se(3)$. The six rotor angular velocities at time step $i$ are represented by $n_i \in \mathbb{R}^6$.

These priors are used in the dynamic parameter factor to compute $\dot{\mathbf{v}}_B$ and $\dot{\boldsymbol{\omega}}_B$, which are integrated to estimate $\xi_{B,i+1}$:

$$p_{i+1} = p_i + \mathbf{v}_{B,i} \Delta t + \frac{1}{2} \dot{\mathbf{v}}_{B,i} \Delta t^2 \quad (12)$$

$$R_{i+1} = R_i \exp \left( \boldsymbol{\omega}_{B,i} \Delta t + \frac{1}{2} \dot{\boldsymbol{\omega}}_{B,i} \Delta t^2 \right) \quad (13)$$

We simulate odometry constraints with noisy relative pose measurements in the body frame, $T_i \in SE(3)$. These are used to constrain $\xi_{B,i+1}$ also:

$$\mathbf{x}_{WB,i+1} = T_i \mathbf{x}_{WB,i} \quad (14)$$

### 3.3 Nonlinear Least Squares

Defining the state to be $\Theta = \{\mathbf{x}_{WB,0}, \mathbf{x}_{WB,n} \ldots, D\}$, the probabilistically optimum state $\Theta^*$ is estimated by solving the following *maximum a posteriori* (MAP) inference problem:

$$\Theta^* = \arg\max_{\Theta} \sum_i ||h_i(X_i) - z_i||^2_{\Sigma_i} \tag{15}$$

where $X_i$ is the $i$th variable node, $z_i$ is a measurement constraining this variable, $h_i$ is a function mapping the state to the measurement space, and $\Sigma_i$ represents the uncertainty of this measurement.

The state variable estimate at iteration $k$, $X_i^k$ is use to compute Jacobian matrices $A_i$ and error terms $b_i$ for each edge in the factor graph:

$$A_i = \Sigma_i^{-1/2} \frac{\partial h_i(X_i)}{\partial X_i}\bigg|_{X_i^k} \tag{16}$$

$$b_i = \Sigma_i^{-1/2}(z_i - h_i(X_i^k)) \tag{17}$$

These are used to form matrices $A$ and $b$, which are used to iteratively compute state updates until convergence.

$$\Delta^* = \arg\min_{\Delta} ||A\Delta - b||_2^2 \tag{18}$$

$$\Theta_{i+1} = \Theta_i + \Delta^* \tag{19}$$

The measurement functions for odometry ($h_O$) and dynamic parameter ($h_D$) constraints both compute the next pose from the previous

$$\tilde{\mathbf{x}}_{WB,i+1} = h_O(\mathbf{x}_{WB,i}, T_i) \tag{20}$$

$$\tilde{\mathbf{x}}_{WB,i+1} = h_D(\mathbf{x}_{WB,i}, \xi_i, \mathbf{n}, \mathbf{D}) \tag{21}$$

In both cases, $A_i$ is computed with the Jacobian of $h$ and $b_i$ is the difference between predicted pose at $i+1$, and the current estimate of $\mathbf{x}_{WB,i+1}$.

## 4 Results

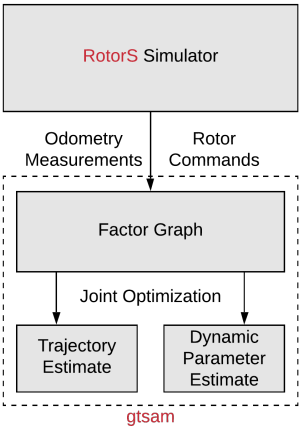


Figure 2: Software implementation for simulation of dynamic parameters and sensor measurements, and factor graph optimization.

We test our method in simulation, using the `RotorS` simulator [6] and the `AscTec Firefly` drone. We record a 100 sec dataset of aggressive maneuvers in the simulation environment. We store timestamped information - odometry data and rotor angular velocities. We corrupt the relative odometry between poses with significant additive white Gaussian noise. Noise is added at each timestep in all six degrees-of-freedom with standard deviation $\sigma = 0.5 \times 10^{-3}$. We assume we have no good information about the dynamic parameters, and initialize them all close to zero.

Our system subscribes to the sensor data ROS topics and builds a factor graph using GTSAM [4]. We run a single batch optimization using a Gauss-Newton optimizer and evaluate the estimated poses and dynamic parameters. In Fig. 4, we compare our estimated trajectory with ground truth and noisy odometry. We see our estimated trajectory adheres better to the ground truth, and this is confirmed by our quantitative assessments in Table 1. The absolute trajectory error (ATE) and relative pose error (RPE) are decreased in our method.

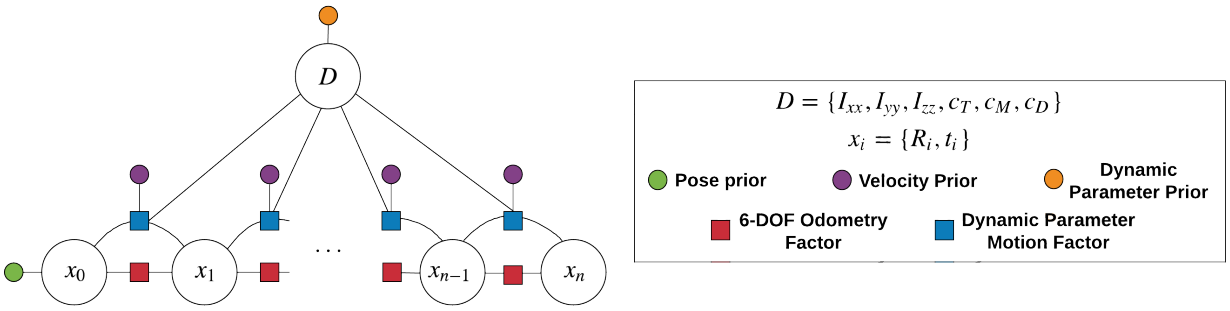


Figure 3: Factor graph used for joint estimation of MAV trajectory and dynamic parameters

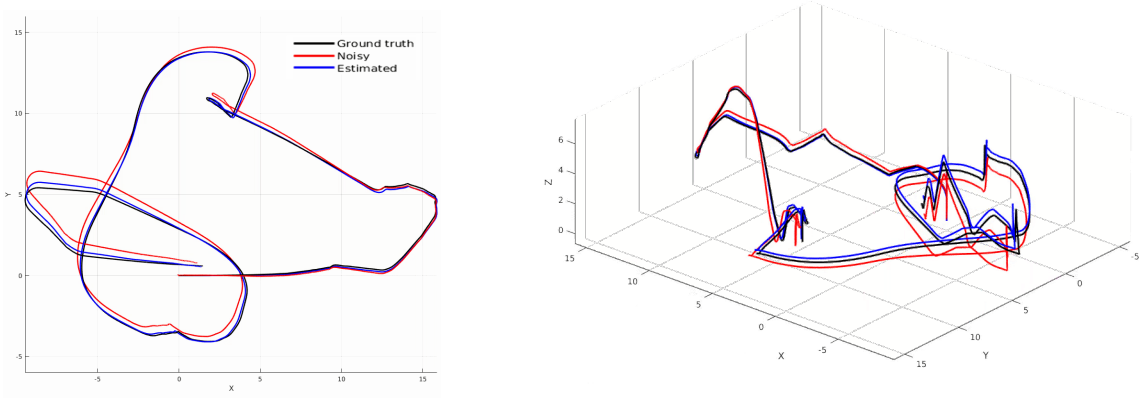


Figure 4: Trajectory plot showing ground truth, noisy odometry and the optimized poses. Visually, we see better performance in our method.

| ATE (m) | | RPE (m) | |
|---|---|---|---|
| *Odom* | *Our Method* | *Odom* | *Our Method* |
| 0.2538 | 0.1909 | 0.7671 | 0.3738 |

Table 1: The absolute trajectory error (ATE) and relative pose error (RPE) of the noisy odometry vs. our estimated trajectory.

| DynParams | Ixx | Iyy | Izz | cT | cM | cD |
|---|---|---|---|---|---|---|
| *Initialization* | 0.0018 | 0.0016 | 0.0512 | 8.54858e-6 | *0.000001* | *0.0433* |
| *Estimated* | 0.0193 | 0.0251 | 0.0771 | 9.1319e-6 | 5.9149e-9 | 0.0213 |
| *Ground truth* | 0.0347 | 0.0458 | 0.0977 | 8.54858e-6 | - | - |

Table 2: Dynamic parameter values, comparing initialized state with final estimate and ground truth.

Further, we analyze the estimated dynamic parameters in Table 2. We note that inertial parameters can be roughly recovered from our flight dataset. Some of the parameters cannot be easily decoupled, as we see in the case of $c_T$ (which we initialize to ground truth). The simulator does not give us information about $c_M$ and $c_D$.

## 5 Discussion

We demonstrate that trajectory estimation can be improved while concurrently converging towards ground truth dynamic parameters. We achieve convergence to an improved trajectory and dynamic parameter values, from a larger initial error as compared to [3], and verify their finding that this joint optimization is highly sensitive to the initialization of dynamic parameter values and assumed noise distributions of state variables.

Future work would be to further improve the convergence of the parameters, or develop an initialization method, similar to the EKF used in [3]. We could also expand our dynamics model to include odometry that is not centered at the MAVs body frame, which would allow us to estimate parameters on real hardware using a vision or LiDAR-based state estimate. In addition, since the factor graph formulation estimates uncertainty of the parameters as well, it would be interesting to apply control algorithms that account for these uncertainties. Lastly, since specific motions are required to decouple the parameters, the optimization could be altered to only step in well-constrained directions, which can be done using singular value decomposition.

## References

[1] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE robotics & automation magazine*, 19(3):20–32, 2012.

[2] Girish Chowdhary and Ravindra Jategaonkar. Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter. *Aerospace science and technology*, 14(2):106–117, 2010.

[3] Michael Burri, Michael Bloesch, Zachary Taylor, Roland Siegwart, and Juan Nieto. A framework for maximum likelihood parameter identification applied on MAVs. *Journal of Field Robotics*, 35(1):5–22, 2018.

[4] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[5] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[6] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. RotorS—a modular gazebo MAV simulator framework. In *Robot Operating System (ROS)*, pages 595–625. Springer, 2016.