

Development and Preliminary Evaluation of a RANSAC Algorithm for Dynamical Model Identification in the Presence of Unmodeled Dynamics

Cora A. Dimmig^{1,2}, Joseph Moore^{1,2} and Louis L. Whitcomb²

Abstract—This paper reports a novel Random Sample Consensus (RANSAC) algorithm for robust identification of second-order plant dynamical model parameters in the presence of unmodeled plant dynamics and noisy experimental data. Accurate plant dynamical models are essential to model-based control system design and for accurate numerical simulation of plant response. Studies of RANSAC approaches for plant model identification have been extremely limited and have not explored performance improvements in the presence of unmodeled dynamics. The performance of the proposed approach, evaluated in a preliminary simulation study of a planar aerial rotorcraft model, is found to be significantly more robust to the effects of unmodeled vehicle dynamics and outlier noise than conventional least squares parameter identification. We conjecture that the proposed approach may be broadly applicable to robust model parameter identification for a wide variety of plants that exhibit noisy sensor data and/or unmodeled dynamics.

I. INTRODUCTION

Accurate and robust model identification is essential for model-based control system design and accurate numerical simulation of dynamic land, air, space, and undersea vehicles. Robust model identification enables precise, dynamic position control, essential to many applications, including safely maneuvering around people and in complex environments.

State estimates are inherently noisy due to sensor dropout, varying operational conditions, and discrepancies between the true vehicle dynamics and the modeled dynamics, i.e. unmodeled dynamics. Unmodeled dynamics can add severe non-Gaussian uncertainty to the sensor data that many model identification approaches are not equipped to address.

Random Sample Consensus (RANSAC) is an iterative algorithm designed to estimate parameters of a mathematical model from noisy experimental data while rejecting outlier data. RANSAC iteratively evaluates parameters, estimated based on minimal samples of a data set, and outputs the parameters for the model that best matches the data [11]. Dynamical Plant Identification RANSAC (DIRANSAC) was recently developed to apply the RANSAC approach to dynamical model identification [22]. Reported herein is (a) a simulation study motivating the use of RANSAC based approaches in the challenging case of unmodeled dynamics

and (b) Inlier-Biased DIRANSAC (I-DIRANSAC), which adds greater emphasis on inliers to strengthen DIRANSAC.

We formulated I-DIRANSAC to internally utilize any parameter estimation algorithm. As a demonstrative example of the improvement I-DIRANSAC can provide to the internal estimator, we used Ordinary Least Squares (OLS) in this evaluation. OLS is the most common approach used for model identification, but assumes all noise is Gaussian and that large data sets will smooth gross errors. Without these significant assumptions, the goal of I-DIRANSAC is to identify reliable parameters that result in an input-output response most representative of the true vehicle.

To evaluate model identification performance, we used a planar rotorcraft model. We considered a widely used *baseline plant model* for parameter identification and formulated an *extended plant model*, which incorporates commonly ignored mechanical and aerodynamic imbalances, to represent the real-world unknown plant. These models are disjoint to allow for evaluating the algorithms in the presence of unmodeled dynamics. Due to this formulation, trajectory tracking error with the identified parameters is the primary performance metric since there are not “true” parameters to recover due to the intended mismatch in the models.

For evaluation, the identified parameters are employed in model-based closed-loop control, tracking trajectories spanning 4 meters, with data that contains significant outliers and unmodeled dynamics. The controller using I-DIRANSAC-identified parameters exhibits improved trajectory tracking performance of up to 1 meter and 3.5 meters of position error in comparison to the controllers using DIRANSAC-identified and OLS-identified parameters, respectively. This significant of an improvement enables precise control for increased maneuverability and operational safety. Furthermore, the improvement compared to OLS motivates using RANSAC based approaches when vehicles have sensors or complicated dynamics that may result in non-Gaussian noise.

II. RELATED WORK

Dynamical model parameter identification has been widely studied. Classical methods such as least squares, which is most commonly used, and maximum likelihood methods utilize observed behavior to estimate parameters of a fixed model [13], [25]. Adaptive methods utilize sensor data to improve the estimate online [18]. Data with non-Gaussian noise can significantly skew the results from these methods, which is one of the challenges I-DIRANSAC was designed to overcome. Recently numerical and learning techniques have been investigated [2], [4], [28]. These methods often require a large quantity of data, highly accurate simulations, and

We gratefully acknowledge the support of a Johns Hopkins University Laboratory for Computational Sensing and Robotics (LCSR) Distinguished Graduate Fellowship, the Johns Hopkins University Applied Physics Laboratory internal research and development funds, and the National Science Foundation under award IIS-1909182.

¹Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723, USA.

²Department of Mechanical Engineering and Laboratory for Computational Sensing and Robotics, Johns Hopkins University, Baltimore, MD 21218, USA. Email: cdimmig@jhu.edu, Joseph.Moore@jhuapl.edu, llw@jhu.edu

often necessitate local linearity assumptions. I-DIRANSAC utilizes one data set, either from simulation or a real system, and can find a global or local solution for a nonlinear model.

The RANSAC approach to model parameter identification, originally reported in the seminal paper by Fischler and Bolles, is designed to robustly fit model parameters in the presence of noise and outliers commonly encountered in a wide variety of modeling problems involving experimental data [11]. Fundamentally, RANSAC takes a minimal sample of the total data and then expands this sample, where possible, based on inliers to the identified model. RANSAC is an iterative, brute force algorithm, reporting the best solution found after evaluating sufficient random samples. This yields significant improvements for model fitting in the presence of both Gaussian and non-Gaussian noise. RANSAC was introduced in the context of scene analysis and automated cartography and has since been widely used in the field of computer vision for regression problems and estimating transformation models between correlated images [11].

Numerous variations on the original RANSAC algorithm, including Locally optimized RANSAC [8], Maximum Likelihood SAC [27], and Progressive Sample Consensus [7], have been reported to optimize speed, accuracy, and robustness [6]. Recursive RANSAC seeks to estimate the parameters of multiple signals [19], [20]. RANSAC has been used for motion estimation [21] and for geometric shape detection in point clouds [24]. In the application space of dynamical systems, RANSAC has been investigated to improve state estimation as a Kalman smoother [10]. To the best of our knowledge, the first reported RANSAC approach for dynamical model identification is the DIRANSAC algorithm, which was applied to a single degree-of-freedom plant model in the presence of outliers [22]. In the present paper, we report a novel extension of DIRANSAC, informed by previously reported variants of RANSAC, and demonstrate that this approach shows significant identification improvement in the presence of unmodeled dynamics.

III. TECHNICAL APPROACH

We propose a RANSAC algorithm, with a novel technique to inlier biasing, for robust identification of second-order dynamical model parameters with experimental data corrupted by noise and significant unmodeled dynamics. We refer to this algorithm, expressed in Algorithm 1, as Inlier-Biased DIRANSAC (I-DIRANSAC) due to its significant weighting of inliers to the identified models and focus on robust evaluation of the identified parameters. We also present results with a previously unreported version of DIRANSAC that extends the algorithm for use with higher dimensional systems and evaluates performance in the presence of a common form of non-Gaussian noise, i.e. unmodeled dynamics.

The squared model fit error function originally proposed in DIRANSAC [22] is modified for I-DIRANSAC to be a root mean squared (RMS) error to reduce the bias of gross errors. The original acceptance metric in [22] for storing the identified parameters when error is minimal was expanded to simultaneously minimize error and maximize inliers. To further inlier bias, the inliers in samples that pass a broader

inlier acceptance metric (Algorithm 1 line 19) are now used to compute a new, potentially further refined, model. Lastly, the termination criteria have been modified from the original formulation of DIRANSAC [22] to minimize overall evaluation time, adding thresholds for sufficient performance to terminate and a user-defined maximum number of iterations.

A. I-DIRANSAC Algorithm Methodology

In the statement of I-DIRANSAC, Algorithm 1, the number of control inputs, sensor outputs, and model parameters are n_U , n_Y , and n_P , respectively. The total number of observations is N . The main input to I-DIRANSAC is sensor and control input data collected from the dynamical system of interest, such as velocities, accelerations, and commanded thrusts and moments, depending on the vehicle and sensor suite. The algorithm samples a minimal subset of the observations based on a user specified size, s , and computes the design matrices, W , b , defined in (1), for that sample. More observations are added to the sample as needed until the internal parameter estimator solution is well-defined with the resulting design matrices. We utilize minimal samples to reduce the likelihood of outliers and to avoid over-fitting to the data. The parameter hypothesis, $\hat{p}$, for the sample is then computed. If that hypothesis passes a preliminary test verifying the feasibility of the parameters, then the plant is simulated forward using the hypothesized parameters and the control inputs, time step, and initial state from the data. If the error and inlier fraction minimizes the acceptance metric, the new parameters are stored. If there is an improvement in either the model fit error or inlier fraction, the set of inliers, $\mathcal{I}$, is used as the next sample. Once the iteration limit is met or the inlier fraction is above a specified threshold, the parameters with the lowest acceptance metric are returned.

B. I-DIRANSAC Dynamical System Functions

For the dynamical system of interest, we define a function to compute the design matrix and a preliminary test to evaluate the resulting identified parameters. For a system that is linear in terms of the parameters, $p \in \mathbb{R}^{n_P \times 1}$, we can use the “regressor matrix” form [15]. The equations of motion can be expressed as $W_i p = b_i$, where $W_i \in \mathbb{R}^{r \times n_P}$ and $b_i \in \mathbb{R}^{r \times 1}$, for each observation, i . Then in “ComputeDesignMatrix” in Algorithm 1, for a set of s observations, the design matrix, W , and the observational vector b can be defined,

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_s \end{bmatrix} \in \mathbb{R}^{sr \times n_P}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix} \in \mathbb{R}^{sr \times 1}, \quad (1)$$

such that $Wp = b$. The preliminary test of the parameters, specified as “PreliminaryTest” in Algorithm 1, performs a basic check that the parameters are feasible, such as verifying the sign of parameters, e.g. mass must be strictly positive.

C. I-DIRANSAC Parameter Estimation Functions

Any parameter estimation method could be used within I-DIRANSAC for the parameter hypothesis step, such as ordinary least squares (OLS) or total least squares (TLS). This paper presents results using OLS internal to I-DIRANSAC.

Algorithm 1 Inlier-Biased DIRANSAC (I-DIRANSAC)

Input:

- U Matrix of observed plant control inputs $\in \mathbb{R}^{n_U \times N}$
- Y Matrix of observed sensor data $\in \mathbb{R}^{n_Y \times N}$
- dt Time step of the sampled data (sampling frequency)
- ε Vector of inlier thresholds for each signal $\in \mathbb{R}^{n_Y \times 1}$
- ξ Number of signals needed to classify as an inlier
- w Vector of weights for model fit error $\in \mathbb{R}^{n_Y \times 1}$
- ν Vector of acceptance metric weights: $[\nu_e, \nu_{f_I}]$
- s Minimum number of observations used in a sample
- k_m Maximum number of major iterations to perform
- $f_{I,t}$ Inlier fraction threshold for early termination

Output:

- p Final parameter estimate $\in \mathbb{R}^{n_P \times 1}$
- f_I Fraction of inliers for the parameter estimate

```
1:  $e = \infty, f_I = 0, k = 0$ 
2: GetNewSample = true
3: while  $k < k_m$  and  $f_I < f_{I,t}$  do
4:   if GetNewSample then
5:      $S =$  Set of  $(s - 1)$  randomly sampled observations
       from  $U$  and  $Y$ 
6:     while Solution is not well-defined
       and  $\{U, Y\} \neq S$  do
7:        $S = S +$  Randomly sampled observation from  $U$ 
       and  $Y$ 
8:        $W, b \leftarrow$  ComputeDesignMatrix( $S$ )
9:       CheckWellDefined( $W, b$ )
10:    end while
11:     $k = k + 1$ 
12:  end if
13:   $\hat{p} \leftarrow$  ComputeParameterHypothesis( $W, b$ )
14:  GetNewSample = true
15:  if PreliminaryTest( $\hat{p}$ ) then
16:     $Y_{\hat{p}} \leftarrow$  ForwardSimulation( $\hat{p}, dt, U, Y_0$ )
17:     $e_{\hat{p}} \leftarrow$  ComputeModelFitError( $Y_{\hat{p}}, Y, w$ )
18:     $f_{I,\hat{p}}, \mathcal{I} \leftarrow$  ComputeFractionOfInliers( $Y_{\hat{p}}, Y, \varepsilon, \xi$ )
19:    if  $f_{I,\hat{p}} > f_I$  or  $e_{\hat{p}} < e$  then
20:      if AcceptanceMetric( $e_{\hat{p}}, e, f_{I,\hat{p}}, f_I, \nu$ ) then
21:         $e = e_{\hat{p}}$ 
22:         $f_I = f_{I,\hat{p}}$ 
23:         $p = \hat{p}$ 
24:      end if
25:      if  $S \neq \mathcal{I}$  then
26:         $S = \mathcal{I}$ 
27:         $W, b \leftarrow$  ComputeDesignMatrix( $S$ )
28:        CheckWellDefined( $W, b$ )
29:        if Solution is well-defined then
30:          GetNewSample = false
31:        end if
32:      end if
33:    end if
34:  end if
35: end while
36: return  $p, f_I$ 
```

For the internal parameter estimator, the function “CheckWellDefined,” as specified in Algorithm 1, verifies that the parameters are observable. For OLS, the condition for observability is that $W^T W$ is full rank. Since this matrix is Hermitian Positive Semi-Definite, the eigenvalues are real and non-negative. Thus we check that the smallest eigenvalue is above a threshold to verify it is not numerically close to zero. The parameters can then be estimated with the “ComputeParameterHypothesis” function based on the design matrices. For example, the Moore-Penrose inverse computes the standard OLS solution, $p = (W^T W)^{-1} W b$.

D. General I-DIRANSAC Functions

In I-DIRANSAC we define a “ForwardSimulation” function to simulate the dynamical system with the identified parameter estimates. The initial state is set to be the first state in the set of observations, Y_0 , and the plant is simulated forward using the control input data, U , at the data sampling time step, dt . This outputs the simulated sensor data, $Y_{\hat{p}}$ for the given parameter estimate, $\hat{p}$. The model fit error is calculated in the “ComputeModelFitError” function using a weighted root mean squared (RMS) error,

$$e_{\hat{p}} = \sum_{i=1}^{n_Y} \left(w_i \sqrt{\frac{1}{N} \sum_{j=1}^N (Y_{\hat{p},i,j} - Y_{i,j})^2} \right), \quad (2)$$

where the indices i, j indicate the matrix position of the observed or simulated sensor data. The weights, w_i , on each signal can be set to ignore certain signals and/or weight others higher. The fraction of inliers for the proposed model is computed in “ComputeFractionOfInliers.” For each signal i and time step j , the corresponding data point, $Y_{\hat{p},i,j}$, is considered a signal inlier when $|Y_{\hat{p},i,j} - Y_{i,j}| < \varepsilon_i$. An overall estimated state at time step j , $Y_{\hat{p},j}$, which consists of signals $Y_{\hat{p},i,j}$ for i from 1 to n_Y , is considered an inlier ($\in \mathcal{I}$) if it has at least ξ , specified by the user, signal inliers. The overall inlier fraction, $f_{I,\hat{p}}$, is the total number of inliers divided by the total number of estimated states. The inlier thresholds can be set to ignore certain signals in this calculation.

We designed the “AcceptanceMetric” in Algorithm 1 to simultaneously minimize the error and maximize the number of inliers. We define it using the acceptance metric weights for error, ν_e , and fraction of inliers, ν_{f_I} , as follows

$$\nu_e e_{\hat{p}}^2 + \nu_{f_I} \left(\frac{1}{f_{I,\hat{p}}} \right)^2 < \nu_e e^2 + \nu_{f_I} \left(\frac{1}{f_I} \right)^2. \quad (3)$$

This is a binary valued function comparing performance with the current hypothesis, $e_{\hat{p}}, f_{I,\hat{p}}$, versus the stored best estimate, e, f_I . Each side of the inequality is a weighted quadratic cost of the model fit error and inverse of the fraction of inliers. If the current hypothesis minimizes this cost function, then this metric returns “True” and the hypothesis will be stored as the current best estimate.

IV. ROTORCRAFT PLANT MODEL

A. Baseline Plant Model

We selected a 3-degree-of-freedom (DOF) planar rotorcraft model, as depicted in Fig. 1, as a representative

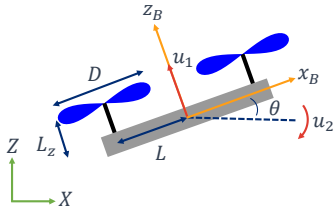


Fig. 1: Diagram of the planar rotorcraft model

dynamical system with which to evaluate the algorithm's performance in simulation. This model represents a 3-DOF version of the 6-DOF model commonly used in practice in quadrotor controller design [16]. We define the system state to be position (x, z) in the world frame and body angle θ in the plane. The distance from the vehicle's center of gravity (CG) to each rotor is L and was treated as a known constant. The control inputs are total thrust, u_1 , defined along the body Z axis, z_B , and total moment, u_2 , defined as a positive moment around the body Y axis, as depicted in Fig. 1. Induced drag due to the backward inclination of aerodynamic force with respect to airfoil motion, as modeled in [16], was additionally considered with coefficient of induced drag, C_D . The vehicle has a total mass, m , with a center of mass located at the body-frame origin, and moment of inertia, I , about the Y axis. By summing the forces and moments on the system in the world frame, the equations of motion are

$$\begin{bmatrix} \ddot{x} \\ \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{m}u_1 \sin \theta \\ \frac{1}{m}u_1 \cos \theta - g \\ \frac{L}{I}u_2 \end{bmatrix} - \frac{C_D}{m}|u_1| \begin{bmatrix} \dot{x} \cos \theta \\ \dot{z} \sin \theta \\ 0 \end{bmatrix}. \quad (4)$$

The plant parameters to be identified for this model are mass, m , moment of inertia, I , and the coefficient of induced drag, C_D . Mass is included as a parameter to identify since the vehicle could carry an object or have frequent instrumentation changes which would vary the overall mass. We can rewrite (4) so that the model parameter vector to be identified, $p = [m, I, C_D]^T$, appears linearly in the form $W_i p = b_i$, from which W and b from (1) can be computed for a set of observations,

$$\begin{bmatrix} \ddot{x} & 0 & |u_1| \dot{x} \cos \theta \\ \ddot{z} + g & 0 & |u_1| \dot{z} \sin \theta \\ 0 & \ddot{\theta} & 0 \end{bmatrix} \begin{bmatrix} m \\ I \\ C_D \end{bmatrix} = \begin{bmatrix} -u_1 \sin \theta \\ u_1 \cos \theta \\ u_2 L \end{bmatrix}. \quad (5)$$

We formulated an extension to this baseline plant model, (4), to include imbalance terms, or frequently unmodeled dynamics, that may be more representative of a real-world rotorcraft. In practice, these effects are commonly ignored in favor of the simpler baseline plant model, (4). However, unmodeled dynamics can cause a periodic disturbance, which presents as severe non-Gaussian noise in the sensor data. Discrepancies between the modeled dynamics and the true vehicle dynamics are inherent to the modeling of real-world systems; all mathematical plant models are approximate. Thus the disturbance caused by unmodeled dynamics must be overcome to enable precise control of a vehicle. Formulation of these disjoint models allows the parameter identification techniques to be evaluated in the presence of unmodeled dynamics. We considered aerodynamic imbalance and mass

imbalance as two common forms of unmodeled dynamics. Aerodynamic imbalance is due to a difference in the coefficient of thrust across propellers [1], [17]. Mass imbalance is due to a difference in mass between the blades of a propeller.

B. Aerodynamic Imbalance

The relation between torque and thrust on each propeller can be formulated as follows, as in [1],

$$T = C_T \rho n^2 D^4, \quad (6)$$

$$Q = 2\pi C_P \rho n^2 D^5, \quad (7)$$

where T is thrust, C_T is the coefficient of thrust, ρ is fluid density, n is propeller rps, D is propeller diameter, Q is torque, and C_P is the coefficient of power. Clearly, Q can be expressed linearly in terms of T

$$Q = \left(\frac{2\pi C_P D}{C_T} \right) T. \quad (8)$$

In the case of the planar rotorcraft, the control input u_1 is the total thrust. Using the total moment, u_2 , the thrust on each propeller, T_1 , T_2 , can be solved for as follows

$$T_1 = \frac{1}{2} \left(u_1 - \frac{u_2}{L} \right), \quad (9)$$

$$T_2 = \frac{1}{2} \left(u_1 + \frac{u_2}{L} \right). \quad (10)$$

Thus, for each propeller i , the propeller rps, n_i , can be solved for using (6)

$$n_i = \sqrt{\frac{T_i}{C_T \rho D^4}}. \quad (11)$$

Consider a thrust due to the imbalance between the blades of a propeller, e.g. slightly different C_T and D for each blade. The difference in these constants results in some imbalance term, γ . For one propeller with two blades, α and β ,

$$\gamma = C_{T_\alpha} D_\alpha^4 - C_{T_\beta} D_\beta^4. \quad (12)$$

The magnitude of gamma controls the amount of vibration or noise in the system due to aerodynamic imbalance. The thrust due to aerodynamic imbalance for propeller i , T_{a_i} , is

$$T_{a_i} = \gamma_i \rho n_i^2. \quad (13)$$

The resulting torque has a varying lever arm due to the rotating propeller in the 2D plane. The projected radius, r_{proj_i} , depends on the true radius $r = \frac{D}{2}$ and is projected into the plane using the relation $r_{proj_i} = r \cos \phi_i$. Here, the angle of projection is $\phi_i = \omega_i t$, where ω_i is the angular velocity and relates to the propeller rps, $\omega_i = 2\pi n_i$. Thus, the projected radius can be expressed as

$$r_{proj_i} = r \cos(2\pi n_i t). \quad (14)$$

Finally, the torque on the system due to the imbalance of each propeller, Q_{a_i} , can be solved for using (8) and (14)

$$Q_{a_i} = \frac{2\pi C_P D_{proj_i}}{C_T} \gamma_i \rho n_i^2 \quad (15)$$

$$= \frac{4\pi C_P}{C_T} \gamma_i \rho n_i^2 r_{proj_i}. \quad (16)$$

These imbalance terms for each propeller are injected into the dynamics through the sum of moments and forces.

C. Mass Imbalance

Consider a propeller to be two point masses, of mass m_α and m_β , each a distance r_{proj_i} from the center of the propeller. The difference in mass between the two blades of the propeller is $\Delta m = m_\alpha - m_\beta$.

The torque due to the centrifugal type force on the rotating propeller and the difference in mass, Δm , causes a rotational mass imbalance. This torque on propeller i can be expressed as follows

$$\tau_{c_i} = \Delta m \omega_i^2 r_{proj_i} L_z, \quad (17)$$

where L_z is the distance along the z_B axis from the center of gravity to the propeller as depicted in Fig. 1.

Another form of mass imbalance is due to the gravitational loading of the masses m_α and m_β on propeller i . The difference in mass, Δm , between each blade of the propeller causes a force, F_{g_i} , and torque, τ_{g_i} , on the system. These can be expressed as

$$F_{g_i} = m_\alpha g - m_\beta g = \Delta m g \quad (18)$$

$$\tau_{g_i} = \Delta m g r_{proj_i}. \quad (19)$$

A sensitivity analysis showed the effect of these terms to be small in comparison to the rotational mass imbalance.

D. Extended Plant Model

Combining the aerodynamic and mass imbalance terms for the system's two propellers with the baseline plant model, (4), yields the extended plant model's dynamics

$$\begin{bmatrix} \ddot{x} \\ \ddot{z} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{m} u_1 \sin \theta - \frac{C_D}{m} |u_1| \dot{x} \cos \theta \\ \frac{1}{m} u_1 \cos \theta - g - \frac{C_D}{m} |u_1| \dot{z} \sin \theta + \sum_{i=1}^2 (T_{a_i} + F_{g_i}) \\ \frac{L}{I} u_2 + \sum_{i=1}^2 (Q_{a_i} + \tau_{c_i} + \tau_{g_i}) \end{bmatrix}. \quad (20)$$

We also considered blade flapping as a form of commonly unmodeled dynamics for a rotorcraft. Blade flapping is due to a higher absolute tip velocity of the advancing blade generating more lift than the retreating blade when a rotor translates horizontally through air [16]. A sensitivity analysis of the dynamics in (20) with blade flapping included showed its overall effect to be negligible for this system.

E. Plant Parameters

We selected the plant parameters based on DJI's Mavic Mini [9] and wind tunnel data [5]. Table I lists the plant parameters used for this evaluation. Feasible magnitudes of the imbalance terms were estimated based on the plant parameters. A 1.5% to 2.5% difference between the blades in both coefficient of thrust, C_T , and diameter, D , yielded an estimate for the bounds of aerodynamic imbalance, γ , as defined in (12). To estimate a feasible range for mass imbalance, we considered a propeller with two blades, each weighing about 0.01 kg [9], with a mass imbalance between the blades of about 0.6% to 1.4%. The resulting ranges of imbalances are shown in Table II. A sensitivity analysis confirmed these estimates to be a reasonable magnitude.

We added two forms of noise to the extended plant model: Gaussian noise and outlier, or salt and pepper, noise. We applied Gaussian noise, based upon the Pixhawk 2's internal gyroscope and accelerometer [3], [26], directly to the acceleration and angular velocity terms and the covariance was integrated for the lower order terms. We added outliers to each signal with a probability π_o . Outliers were sampled from a uniform distribution that ranged from $-a_o$ to a_o . The selected noise parameter ranges are shown in Table III.

TABLE I: Rotorcraft Plant Parameters for the Baseline and Extended Plant Models

Variable	Value	Description
g	9.81 m/s ²	Gravity
m	0.249 kg	Mass
I	0.004 kg m ²	Moment of inertia
C_D	0.025 s/m	Coefficient of induced drag
C_T	0.06	Coefficient of thrust
C_P	0.06	Coefficient of power
D	0.127 m	Propeller diameter
L	0.177 m	Distance rotor to CG along X axis
L_z	0.01 m	Distance rotor to CG along Z axis

TABLE II: Rotorcraft Imbalance Parameter Value Ranges for the Extended Plant Model

Variable	Min Value	Max Value	Imbalance Type
γ	1.0×10^{-6}	1.8×10^{-6}	Aerodynamic (m ⁴)
Δm	6.0×10^{-5}	1.4×10^{-4}	Mass (kg)

TABLE III: Noise Parameter Value Ranges for the Extended Plant Model

Variable	Min Value	Max Value	Description
μ	0	0	Mean of Gaussian noise
$\sigma_{\dot{\theta}}$	0.003	1.0	Covariance of Gaussian noise on angular velocity (rad/s)
$\sigma_{\ddot{x}, \ddot{z}}$	0.04	2.5	Covariance of Gaussian noise on acceleration terms (m/s ²)
a_o	0	100	Maximum absolute magnitude of outliers
π_o	0.05	0.5	Probability of outliers

V. NUMERICAL PERFORMANCE EVALUATION

A. Numerical Simulation Setup

We implemented a numerical simulation of the extended plant model, (20), together with a model-based trajectory tracking controller utilizing the baseline plant model, (4). In the simulations reported herein, we employed three reference trajectory types: a "square" trajectory, a "star" trajectory, and a "sine" trajectory. For a square trajectory, the vehicle starts at the origin and plans to four waypoint XZ positions spaced equidistant from each other and the origin and then back to the first waypoint. Each side of the square is 4 meters. A star trajectory follows the same waypoints as the square trajectory, returns to the origin, and then repeats the square trajectory with a 45 degree rotation. A sine trajectory minimizes the distance between a sine wave reference trajectory with amplitude 2 meters and the vehicle's XZ position.

For trajectory generation, the nonlinear optimization problem is formulated using direct transcription [23]. This problem is solved using the Sparse Nonlinear Optimizer (SNOPT) [12]. To control to the generated trajectory, we use a time-varying linear quadratic regulator (TVLQR) [14].

The simulation process begins with numerically simulating rotorcraft flight with the extended plant model, (20). The system is controlled through a star trajectory to fully excite the model's state space. The plant parameters are identified using this data and the expression of the baseline plant model in (5). It is desired to minimize the difference between the input-output behavior of the plant with the identified parameters and the real plant. Thus, the identified parameters are evaluated by generating square and sine trajectories with the baseline plant model, (4), and then executing the baseline plant based controller on the extended plant model, (20). Root mean squared (RMS) error between the generated reference trajectory and the extended plant's true path is used as a performance metric.

We note that since the extended plant model, (20), is utilized to simulate the rotorcraft dynamics, and the parameter identifiers employ the baseline plant model, (4), the simulated extended plant model data contains signals due to dynamics, e.g. the imbalances, that are unmodeled in the identified baseline plant model. Thus exact identification of "true" parameters cannot be accomplished. This is intentional, as our goal is to investigate the comparative performance of the parameter identifiers in the presence of realistic unmodeled dynamics.

B. I-DIRANSAC Setup

Table IV shows the I-DIRANSAC parameters we used for evaluation with this dynamical system. OLS was used as the parameter estimator internal to I-DIRANSAC. The generated data sets were comprised of about 2500 observations sampled at 100 Hz. Based on these data sets, the minimal sample size, s , was set to 75. As expected, more iterations of the algorithm covers a larger portion of the sample space, which results in a higher potential to identify robust parameters. We empirically determined 1000 iterations to be sufficient, with more iterations showing diminishing returns. In theory, evaluating all possible samples (greater than a googol iterations) could improve the solution further, but would be computationally impractical. We selected the weights, w , for the RMS model fit error in (2), as seen in Table IV, such that only the velocity and acceleration terms would be considered.

With the drag term, the system is an open loop observer of itself and thus velocity will converge even when the initial conditions between the collected data and simulation differ.

TABLE IV: I-DIRANSAC Parameters

Variable	Value	Description
k_m	1000	Maximum number of major iterations
$f_{\mathcal{I},t}$	0.85	Inlier fraction threshold for early termination
s	75	Minimum number of observations in a sample
$\varepsilon_{x,z,\theta}$	0	Inlier threshold for position
$\varepsilon_{\dot{x},\dot{z},\dot{\theta},\ddot{x},\ddot{z},\ddot{\theta}}$	0.5	Inlier threshold for velocity and acceleration
ξ	3	Number of signals within inlier threshold needed to classify an observation as an inlier
$w_{x,z,\theta}$	0	Model fit error weight for position
$w_{\dot{x},\dot{z},\dot{\theta}}$	1	Model fit error weight for velocity
$w_{\ddot{x},\ddot{z},\ddot{\theta}}$	0.1	Model fit error weight for acceleration
ν_e	1	Acceptance metric weight for model fit error
$\nu_{f_{\mathcal{I}}}$	0.06	Acceptance metric weight for inlier fraction

C. Performance Evaluation of I-DIRANSAC

Figures 2 - 5 show summary plots comparing the results of I-DIRANSAC, DIRANSAC, and OLS parameter identification for tracking square and sine trajectories in the presence of non-Gaussian noise: outliers and unmodeled dynamics in the form of aerodynamic and mass imbalances. We varied one independent variable at a time while all other imbalance and noise terms were set to zero. In Figures 2, 3, and 4, the X axis corresponds to the independent variable: outlier magnitude, aerodynamic imbalance, and mass imbalance, respectively. The Y axis is the RMS trajectory tracking error for a particular signal. From left to right these signals are position in X (x), position in Z (z), orientation (θ), velocity in X ($\dot{x}$), velocity in Z ($\dot{z}$), and angular velocity ($\dot{\theta}$). Each data point is the mean of the RMS error across 50 iterations of the specified algorithm. For each iteration new data was collected with the extended plant model, (20). The error bars show one standard deviation in either direction. In Figures 3 and 4, the error bars for OLS have zero length due to the analytic formulation of the imbalance terms causing the data at each iteration to be constant. Thus OLS produces a constant result since there are no other forms of randomization. This is in contrast to the RANSAC based approaches which have inherent randomization. In these figures, lower RMS error denotes better performance.

1) *Robustness to Data Outliers:* Fig. 2 shows trajectory tracking performance evaluated on a square trajectory for a

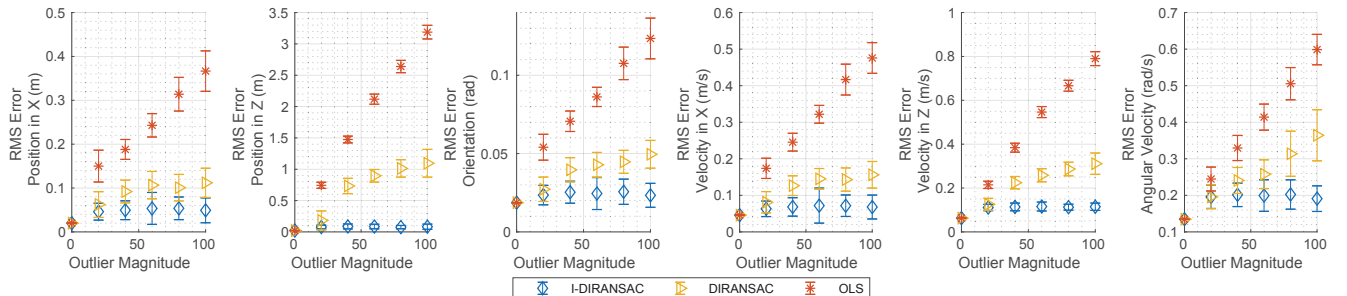


Fig. 2: Trajectory tracking RMS error for various outlier magnitudes evaluated on a square trajectory.

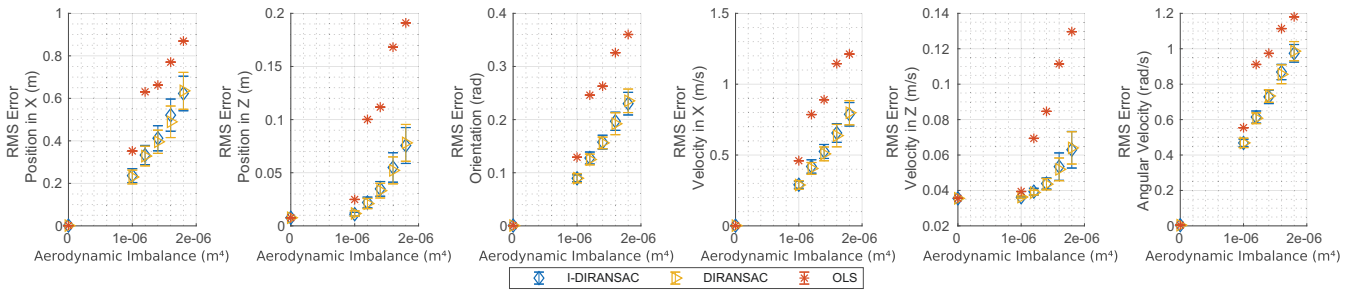


Fig. 3: Trajectory tracking RMS error for various aerodynamic imbalances evaluated on a sine trajectory.

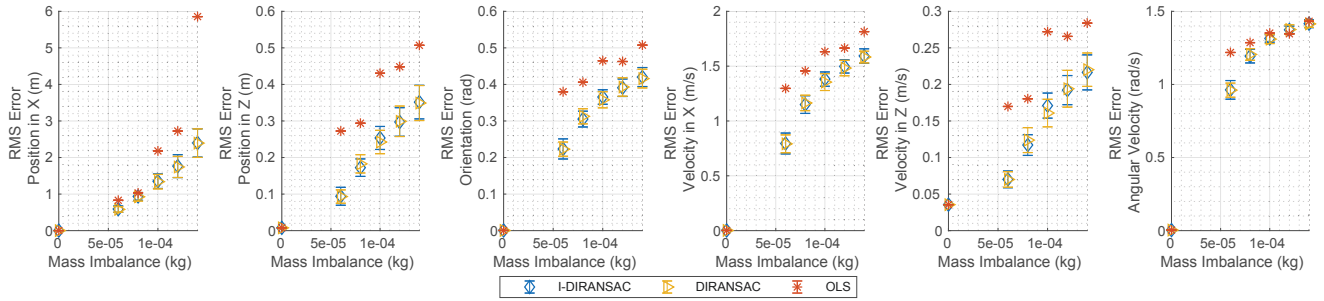


Fig. 4: Trajectory tracking RMS error for various mass imbalances evaluated on a sine trajectory.

magnitude of outliers varying from 0 to 100 with constant probability of 0.1. Clearly, as outlier magnitude increased, the RMS error for trajectory tracking steadily increased when using the parameters identified by OLS and DIRANSAC. In contrast, I-DIRANSAC had consistent performance for each outlier magnitude with minimum RMS error.

The greater focus on model inliers in I-DIRANSAC compared to DIRANSAC allows I-DIRANSAC to find a maximal sample without outliers, thus allowing for parameter estimates with improved trajectory tracking performance over DIRANSAC up to 1 meter in position and up to 3 meters in comparison to OLS, without sacrificing performance in the presence of imbalances.

Fig. 5 shows an example of the trajectory tracking performance evaluated on square and sine trajectories for 50 iterations of each algorithm with an outlier magnitude of 80. The axes represent the XZ plane. For each identified set of parameters, the resulting generated reference trajectory and vehicle tracking performance is plotted. In these figures, minimal deviations between the vehicle and reference trajectories denotes better performance. The parameters identified by OLS and DIRANSAC result in a significant error in Z when tracking the reference trajectories versus the parameters identified by I-DIRANSAC result in robust tracking performance.

2) *Robustness to Unmodeled Dynamics*: RMS trajectory tracking error evaluated on a sine trajectory is shown for aerodynamic imbalance varying from 1×10^{-6} to $1.8 \times 10^{-6} m^4$ in Fig. 3 and mass imbalance varying from 6×10^{-5} to $1.4 \times 10^{-4} kg$ in Fig. 4. The RANSAC based approaches performed comparably and show significant improvement in comparison to OLS in the presence of imbalance terms. These results motivate the use of RANSAC based approaches in cases where there may be mismatches

between the true vehicle dynamics and model dynamics. Fig. 3 shows that in position, the RANSAC based approaches showed an improvement from OLS under the effect of aerodynamic imbalance of up to about 0.3 meters in X. In the presence of mass imbalance, the RANSAC based approaches showed an improvement of up to about 3.5 meters in X, as reflected in Fig. 4.

Tuning of the I-DIRANSAC algorithm parameters could enhance performance further, but this serves as a preliminary motivation for the improvement a RANSAC based approach can provide to the internal parameter estimator, i.e. OLS in this evaluation. Different internal estimators could be used with I-DIRANSAC and could offer greater improvement. These trajectory tracking improvements, particularly in posi-

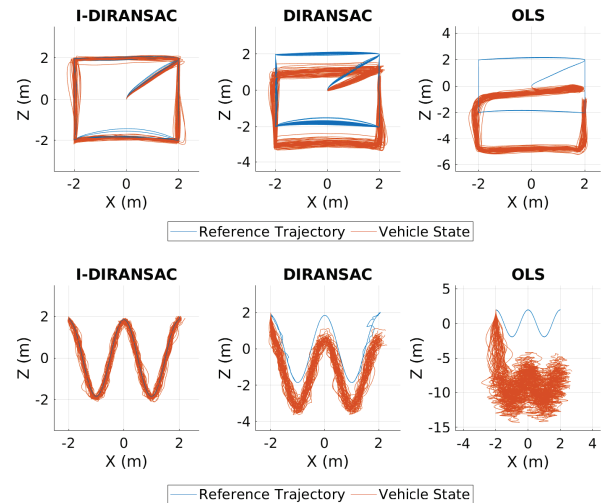


Fig. 5: Trajectory tracking with an outlier magnitude of 80 and outlier probability of 0.1 over 50 iterations.

tion, could make a significant difference in precision control.

We performed this analysis for various levels of Gaussian noise as well which yielded comparable results from I-DIRANSAC, DIRANSAC, and OLS, as expected.

The results using I-DIRANSAC were produced with the new acceptance metric for storing a best parameter solution defined in (3). In general, maximizing the fraction of inliers alone performed best in cases where there were outliers versus minimizing the model fit error alone performed best in cases where there were imbalance terms. The proposed hybrid metric simultaneously maximizes the fraction of inliers, minimizes the model fit error, and rejects cases where there is a significant jump in either. Further improvements could be seen from tuning the weights in this metric depending on the operational scenario and data.

The identified parameters are not included in this analysis of performance in the presence of imbalances since the “true” plant parameters are unknown due to the intended mismatch between the baseline and extended plant models. When evaluating performance for various magnitudes of outliers, as presented in Fig. 2, the true parameters are known since the imbalance terms in the extended plant model, (20), are all set to zero causing the resulting dynamics to be equivalent to the baseline plant model, (4). In this case, the parameters identified by I-DIRANSAC matched the true parameters most closely, which is reflected in the tracking performance in comparison to OLS depicted in Figures 2 and 5.

VI. CONCLUSION

This study reports a novel RANSAC algorithm for dynamical model identification in the presence of inherently non-Gaussian noise arising from unmodeled dynamics and outliers, and a numerical simulation study of the proposed I-DIRANSAC approach and DIRANSAC in comparison to OLS. The numerical simulation study shows that RANSAC based approaches, using OLS as the internal parameter estimator, offer improved robustness to unmodeled dynamics in comparison to traditional OLS. Additionally, I-DIRANSAC offers improved robustness to outliers in comparison to DIRANSAC and OLS. In particular, the position tracking improvements when using I-DIRANSAC in the presence of unmodeled dynamics and non-Gaussian noise could enable significant advancements in precise, dynamic position control. This could be valuable in a wide variety of robotic applications, including for safety in human machine interaction and greater maneuverability in challenging environments. I-DIRANSAC could be utilized with many classes of vehicles; however, vehicles with sensors or complicated dynamics, that often result in data with outliers or other non-Gaussian noise, would see the most benefit.

ACKNOWLEDGMENT

We are grateful to Mr. Eric Nguyen who contributed to the software development of the simulation environment.

REFERENCES

- [1] C. N. Adkins and R. H. Liebeck. Design of optimum propellers. *Journal of Propulsion and Power*, 10(5):676–682, Sept.–Oct. 1994.
- [2] A. Allevato, E. S. Short, M. Pryor, and A. Thomaz. TuneNet: One-shot residual tuning for system identification and sim-to-real robot task transfer. In *Conference on Robot Learning*, pages 445–455. PMLR, 2019.
- [3] ArduPilot Dev Team. The Cube Overview. <https://ardupilot.org/copter/docs/common-the-cube-overview.html>.
- [4] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomlin. Goal-driven dynamics learning via Bayesian optimization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5168–5173. IEEE, 2017.
- [5] J. B. Brandt, R. W. Deters, G. K. Ananda, O. D. Dantsker, and M. S. Selig. UIUC Propeller Data Site. <https://m-selig.ae.illinois.edu/props/propDB.html>.
- [6] S. Choi, T. Kim, and W. Yu. Performance evaluation of RANSAC family. In *Proceedings of the British Machine Vision Conference*, pages 81.1–81.12. BMVA Press, 2009.
- [7] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 220–226. IEEE, 2005.
- [8] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In M. B. and K. G., editors, *Pattern Recognition. DAGM 2003*, Lecture Notes in Computer Science, vol 2781, pages 236–243. Springer, 2003.
- [9] DJI. Mavic Mini Specs. <https://www.dji.com/mavic-mini/specs>.
- [10] S. Farahmand, G. B. Giannakis, and D. Angelosante. Doubly robust smoothing of dynamical processes via outlier sparsity constraints. *IEEE Transactions on Signal Processing*, 59(10):4529–4543, 2011.
- [11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, Jun. 1981.
- [12] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47(1):99–131, 2005.
- [13] P. K. Khosla and T. Kanade. Parameter identification of robot dynamics. In *1985 24th IEEE Conference on Decision and Control*, pages 1754–1760, 1985.
- [14] D. E. Kirk. *Optimal Control Theory: An Introduction*. Courier Corporation, 2004.
- [15] W.-S. Lu and Q.-H. Meng. Regressor formulation of robot dynamics: computation and applications. *IEEE Transactions on Robotics and Automation*, 9(3):323–333, 1993.
- [16] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept. 2012.
- [17] M. McCrink and J. W. Gregory. Blade element momentum modeling of Low-Re small UAS electric propulsion systems. In *33rd AIAA Applied Aerodynamics Conference*, Jun. 2015.
- [18] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems*. Courier Corporation, 2012.
- [19] P. C. Niedfeldt and R. W. Beard. Recursive RANSAC: Multiple signal estimation with outliers. *IFAC Proc.*, 46(23):430–435, 2013.
- [20] P. C. Niedfeldt and R. W. Beard. Convergence and complexity analysis of recursive-RANSAC: A new multiple target tracking algorithm. *IEEE Transactions on Automatic Control*, 61(2):456–461, 2015.
- [21] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
- [22] T. Paine. Robust model identification methods for nonlinear second-order plant models for underwater vehicles. Master’s thesis, The Johns Hopkins University, 2018.
- [23] D. Pardo, L. Möller, M. Neunert, A. W. Winkler, and J. Buchli. Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *IEEE Robotics and Automation Letters*, 1(2):946–953, 2016.
- [24] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for point-cloud shape detection. *Comp. Graphics Forum*, 26(2):214–226, 2007.
- [25] J. Swevers, C. Ganseman, D. B. Tükel, J. De Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997.
- [26] TDK Corporation. *6-Axis MEMS MotionTracking™ Device with Enhanced EIS Support and Sensor Fusion Processing*, Oct. 2017.
- [27] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
- [28] S. Zhu, A. Kimmel, K. E. Bekris, and A. Boularias. Fast model identification via physics engines for data-efficient policy search. *arXiv preprint arXiv:1710.08893*, 2017.