

State Estimation for Aggressive Flight in GPS-Denied Environments Using Onboard Sensing

Adam Bry, Abraham Bachrach, Nicholas Roy
 {abry, abachrac, nickroy}@mit.edu

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract—In this paper we present a state estimation method based on an inertial measurement unit (IMU) and a planar laser range finder suitable for use in real-time on a fixed-wing micro air vehicle (MAV). The algorithm is capable of maintaining accurate state estimates during aggressive flight in unstructured 3D environments without the use of an external positioning system. Our localization algorithm is based on an extension of the Gaussian Particle Filter. We partition the state according to measurement independence relationships and then calculate a pseudo-linear update which allows us to use 20x fewer particles than a naive implementation to achieve similar accuracy in the state estimate. We also propose a multi-step forward fitting method to identify the noise parameters of the IMU and compare results with and without accurate position measurements. Our process and measurement models integrate naturally with an exponential coordinates representation of the attitude uncertainty. We demonstrate our algorithms experimentally on a fixed-wing vehicle flying in a challenging indoor environment.

I. INTRODUCTION

Developing micro air vehicles that approach the maneuverability and speed of birds flying through urban environments poses a number of challenges for robotics researchers in terms of planning, control, and state estimation. Recent work has demonstrated systems that can perform impressive acrobatics [5] and other control feats [15], [16], however such systems are completely reliant on extremely accurate state estimates provided by external camera arrays. In contrast, vehicles that are capable of flight using state estimates computed from onboard sensor data are either confined to wide open areas without obstacles, or slow-moving hovering vehicles such as quadrotors [2], [9].

The wide disparity between what is possible in terms of agile flight with an external positioning system and what has been demonstrated with onboard sensing suggests that state estimation from onboard sensors is indeed a significant challenge in extending the capabilities of MAVs in real world environments. In addition to providing good estimates of the system mean, the state estimation algorithm should also accurately represent uncertainty so that control and planning algorithms can be appropriately cautious around obstacles and other state constraints [3].

This paper presents a state estimation method that is suitable for use in real-time on a fixed wing MAV maneuvering through a cluttered environment. Our system leverages an inertial measurement unit (gyros and accelerometers) and a planar laser range finder in a filtering framework that provides the accuracy, robustness, and computational efficiency required to localize a MAV within a known 3D occupancy map.



Fig. 1. Fixed wing experimental platform flying indoors localizing using an onboard laser range scanner and inertial measurement unit.

In order to efficiently project the nonlinear laser measurement update of the vehicle position back through the state estimate, we integrate the laser range-finder measurement as a pseudo-measurement on a partition of the state space. The pseudo measurement is computed from a Gaussian Particle Filter (GPF) state update [13]. This technique drastically reduces the number of particles required compared to a vanilla implementation of a GPF, which in itself provides a marked improvement over a conventional particle filter [19]. Our algorithm enables realtime performance in the face of the computational limitations of the flight computer. We quantitatively validate our algorithm on a dataset collected by manually maneuvering the sensing components in a motion capture environment. Finally, we demonstrate the effectiveness of our approach experimentally on a fixed wing vehicle being piloted in a challenging GPS-denied environment.

The process model that accompanies the GPF measurement update is based on an exponential-coordinates extended Kalman filter that is driven by inertial measurements. We also propose a technique for estimating the uncertainty parameters of the IMU, namely gyro and accelerometer noise variances, that is based on a multi-step projection of the noise compared with smoothed state estimates. When the algorithm is used with accurate position and orientation measurements the noise variances converge. When the method is used with inaccurate position-only measurements we still see convergence, but also show that with noisier measurements, the optimization is more sensitive to initialization.

II. PROBLEM STATEMENT

Assuming the MAV to be a rigid body and neglecting higher-order effects such as propeller speed and time-varying airflow over the vehicle, the state of a MAV is given by its position and orientation and the associated linear and angular velocities. For control purposes it is convenient to represent the velocities in body coordinates. Thus the goal of the filter is to estimate the quantities $[\omega_b^T \ v_b^T \ R \ \Delta^T]^T$ where $\omega_b = [p \ q \ r]^T$ is the angular velocity in body coordinates, $v_b = [u \ v \ w]^T$ is the linear velocity in body coordinates, R is the rigid body orientation rotation matrix, and $\Delta = [\Delta_x \ \Delta_y \ \Delta_z]^T$ is the translation vector from the origin in global coordinates to the origin of the body frame, expressed in global coordinates.

We assume a set of inertial measurements consisting of 3-axis acceleration, 3-axis angular rate measurements, and exteroceptive measurements consisting of planar laser range scans. Further, we assume we have access to a 3D map of the environment represented as an occupancy grid.

III. IMU PROCESS MODEL

Our state estimation algorithm uses an Extended Kalman Filter (EKF) to estimate a Gaussian distribution over system states. The EKF process model is based on a discrete time, nonlinear discrete transition function:

$$x_{t+1} = f(x_t, u_t, w_t) \quad (1)$$

where x_t is the system state vector, u_t is the input vector to the system, and w_t is a random disturbance drawn from a normal distribution $N(0, Q)$. The EKF tracks the state at time t as a Gaussian distribution with mean μ_t and covariance Σ_t . These first two moments are propagated forward according to:

$$\bar{\mu}_{t+1} = f(\mu_t, u_t, 0) \quad (2)$$

$$\bar{\Sigma}_{t+1} = A_t \Sigma_t A_t^T + W_t Q W_t^T \quad (3)$$

where $\bar{\mu}$ and $\bar{\Sigma}$ denote predicted quantities before a measurement update has occurred, and A_t and W_t are the appropriate partial derivatives of f .

A. Exponential Coordinates Attitude Uncertainty

We track orientation uncertainty in perturbation rotations in the body frame. If the true orientation is given by the rotation matrix R , we can write $R = \hat{R}R(\chi)$ where \hat{R} is the estimated orientation and $R(\chi) = e^{\chi^\wedge}$ is the error rotation matrix. $\chi \in \mathfrak{R}^3$ is the perturbation rotation about the body axes. We use the \wedge symbol to the right of a vector to denote the skew symmetric matrix formed as:

$$\chi^\wedge = \begin{bmatrix} 0 & -\chi_3 & \chi_2 \\ \chi_3 & 0 & -\chi_1 \\ -\chi_2 & \chi_1 & 0 \end{bmatrix} \quad (4)$$

Taking the matrix exponential of a skew symmetric matrix returns a rotation matrix corresponding to a rotation of $|\chi|$ about the axis defined by χ where χ is referred to as the exponential coordinates of rotation.

In our expression for the true orientation, $R(\chi)$ post multiplies \hat{R} which puts the perturbations in the body frame.

Since the error is parameterized by χ , the covariance can be tracked in a 3×3 matrix Σ_χ . The covariance can be thought of as cones of uncertainty surrounding the body frame axes defined by the columns of \hat{R} . A sketch of this uncertainty is shown in figure 2 for the covariance (in degrees):

$$\Sigma_\chi = \begin{bmatrix} 3^2 & 0 & 0 \\ 0 & 5^2 & 0 \\ 0 & 0 & 15^2 \end{bmatrix} \quad (5)$$

This choice of coordinates for the filter error is desirable since fundamentally rigid body orientation, denoted mathematically as the special orthogonal group (SO3), has three degrees of freedom. While any three-element representation is provably singular for some orientation, more commonly-used parameterizations (i.e., quaternions or rotation matrices) will have constraints between the elements of the representation. Thus a linearized filter covariance over the parameters will not be full rank. Numerical errors pose the constant threat of creating negative eigenvalues, and thus causing the estimator to diverge. Furthermore, an efficient linearized measurement update as is commonly-used in Gaussian filters does not respect the constraints and thus does not map onto SO3. A renormalization scheme could be used after every update, but at any given time the representation can be arbitrarily poor [20].

As we will see, the attitude uncertainty representation is agnostic to the actual underlying orientation integration and tracking. Quaternions and rotation matrices are easy to update based on using χ in the estimator state vector μ .

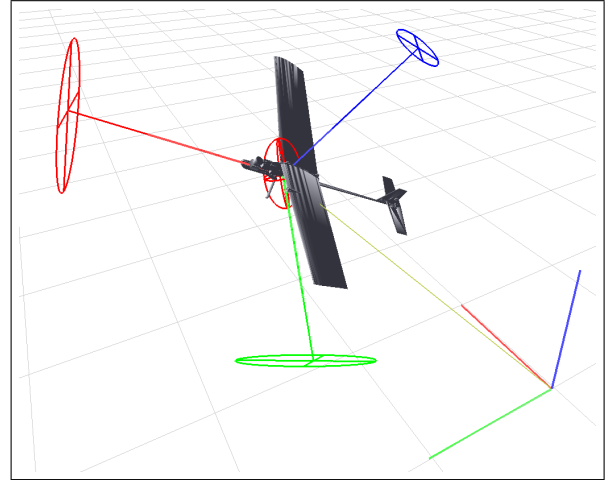


Fig. 2. This figure shows the uncertainty representation in body axes. We see that high variance on the z axis perturbation maps into large motions for the x and y bases. In our implementation we use a Forward-Left-Up (body), East-North-Up (global) (FLU, ENU) coordinate system as opposed to the traditional aerospace frame of Forward-Right-Down, North-East-Down.

B. Process Equations

The equations of motion for a rigid body are given by:

$$\dot{\omega}_b = J^{-1}(-\omega_b \times J\omega_b + \tau_b) \quad (6)$$

$$\dot{v}_b = -\omega_b \times v_b + R^T \bar{g} + a_b \quad (7)$$

$$\dot{R} = R\omega_b^\wedge \quad (8)$$

$$\dot{\Delta} = Rv_b, \quad (9)$$

where τ_b is the torque applied to the body and a_b is the acceleration in body coordinates. Since the IMU provides accurate measurements of ω_b and a_b , we follow the commonly-used technique of omitting ω_b from the state, neglecting equation 6, and treating the IMU measurements as inputs to the filter.

For the quantities used in equation 2 we have

$$x = \begin{bmatrix} v_b & \chi & \Delta \end{bmatrix} \quad (10)$$

$$u = \begin{bmatrix} u_{\text{gyro}} & u_{\text{accel}} \end{bmatrix} \quad (11)$$

$$w = \begin{bmatrix} w_{\text{gyro}} & w_{\text{accel}} \end{bmatrix} \quad (12)$$

Combining this state representation with equations 7-9.

$$f_c(x_t, u_t, w_t) = \begin{bmatrix} \dot{v}_b \\ \dot{R} \\ \dot{\Delta} \end{bmatrix} \quad (13)$$

$$= \begin{bmatrix} -\omega_b \times v_b + R^T \bar{g} + g u_{\text{accel}} \\ R u_{\text{gyro}}^\wedge \\ R v_b \end{bmatrix} \quad (14)$$

Taking the appropriate partial derivatives we get:

$$\frac{\partial \dot{v}_b}{\partial x} = \begin{bmatrix} -\omega_b^\wedge & (R^T \bar{g})^\wedge & 0 \end{bmatrix} \quad (15)$$

$$\frac{\partial \dot{\chi}}{\partial x} = \begin{bmatrix} 0 & -\omega_b^\wedge & 0 \end{bmatrix} \quad (16)$$

$$\frac{\partial \dot{\Delta}}{\partial x} = \begin{bmatrix} R & -R v_b^\wedge & 0 \end{bmatrix} \quad (17)$$

for a continuous dynamics linearization of:

$$A_c = \frac{\partial f}{\partial x} = \begin{bmatrix} -\omega_b^\wedge & (R^T \bar{g})^\wedge & 0 \\ 0 & -\omega_b^\wedge & 0 \\ R & -R v_b^\wedge & 0 \end{bmatrix} \quad (18)$$

and for the input vector we have:

$$\frac{\partial \dot{v}_b}{\partial u} = \begin{bmatrix} v_b^\wedge & gI \end{bmatrix} \quad (19)$$

$$\frac{\partial \dot{\chi}}{\partial u} = \begin{bmatrix} I & 0 \end{bmatrix} \quad (20)$$

$$\frac{\partial \dot{\Delta}}{\partial u} = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad (21)$$

$$W_c = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial \dot{v}_b}{\partial u} \\ \frac{\partial \dot{\chi}}{\partial u} \\ \frac{\partial \dot{\Delta}}{\partial u} \end{bmatrix}. \quad (22)$$

While more sophisticated approximations could be used, we construct the discrete quantities for the filter f , A_t , and W_t using Euler integration:

$$f(x_t, u_t, 0) = x_t + f_c(x_t, u_t, 0)dt \quad (23)$$

$$A_t = I + A_c dt \quad (24)$$

$$W_t = W_c dt. \quad (25)$$

We integrate the attitude separately as

$$R_{t+1} = R_t R(u_{\text{gyro}}^\wedge) \quad (26)$$

IV. LASER MEASUREMENT UPDATE

While the EKF is effective for propagating the first two moments of the nonlinear dynamics through our IMU equations of motion, it is not well-suited to integrating laser measurements from unstructured 3D environments. Using such sensors directly in an EKF requires the extraction and correspondence of features such as corners, and line segments from the sensor measurements, an error prone process that limits the applicability of the algorithms to environments with specific structure [7]. In contrast Monte-Carlo techniques are widely used in laser-based localization algorithms because they allow for the lidar range measurements model to be used directly in the measurement function [19].

While particle filters are efficient enough for effective use in localizing a 2D mobile robot, they require too many particles to be used for the estimation of a 3D MAV. Fortunately, we can obtain the best aspects of both algorithms, and a significant speedup can be realized by employing a hybrid filter that uses an IMU-driven EKF process model with pseudo-measurements computed from Gaussian Particle Filter (GPF) laser measurement updates [13].

A. Gaussian Particle Filters

In its standard form, the GPF maintains a Gaussian distribution over the state space given a measurement history given by $P(x_t | z_{0:t}) = N(x_t; \mu_t, \Sigma_t)$. However, at each iteration of the filter, particles are used to incorporate nonlinear process and measurement models. To compute $P(x_{t+1} | z_{0:t})$ a set of samples $\{x_t^{(j)}\}_{j=1}^M$ is drawn from $N(\mu_t, \Sigma_t)$ and the samples are then propagated through the process model $f(x_t, u_t, w_t)$. To perform the measurement update the samples are weighted according to the measurement model $w_t^{(j)} = P(z_t | x_t^{(j)})$. The updated Gaussian at the end of an iteration of the filter is then obtained as the weighted mean and covariance of the samples

$$\mu_{t+1} = \frac{\sum_{j=1}^M w_t^{(j)} x_t^{(j)}}{w_t^{(j)}} \quad (27)$$

$$\Sigma_{t+1} = \frac{\sum_{j=1}^M w_t^{(j)} (x_t^{(j)} - \mu_{t+1})(x_t^{(j)} - \mu_{t+1})^T}{w_t^{(j)}}. \quad (28)$$

Assuming the underlying system is linear-Gaussian, the filter is shown to approximate the true distribution arbitrarily well with a large number of samples. The GPF filter differs from a standard particle filter by maintaining a unimodal Gaussian distribution over the posterior state instead of the arbitrary (possibly multi-modal) distribution represented by the set of particles in a conventional particle filter.

A straightforward implementation of the GPF for state estimation using a laser on a MAV is impractical and inefficient for two reasons:

- 1) IMU dynamics are well-approximated by linearization as evidenced by the widespread use of EKFs in GPS-IMU state estimation. Thus, a particle process model adds significant computational burden and sampling error, without significantly improving the estimate of the posterior density.

- 2) The IMU filter maintains additional states to track velocity and IMU biases, however the laser measurements are only a function of the position and orientation, parameterized by Δ and χ in our formulation. In fact, most of the orientation information in the measurement exists in the plane of the laser contained in χ_z .

To address the first issue we only use the GPF to perform the measurement update, and instead of propagating samples through the measurement function, we sample directly from the prior distribution returned by the EKF after the process step, $N(\bar{\mu}, \bar{\Sigma})$. To address the second issue above we explicitly partition the state according to independence relationships in the measurement function. We perform a standard GPF measurement update on the partitioned state and use this to compute a pseudo-measurement which is then used to update the full state.

B. Partitioned State Update

The state is partitioned as,

$$x_t = \begin{bmatrix} x_t^m & x_t^p \end{bmatrix}, \quad (29)$$

where $x_t^m \in \mathbb{R}^k$ is the part of the state that affects the measurement, and $x_t^p \in \mathbb{R}^{n-k}$ is independent from the measurement. More formally we assume our measurement function has the form

$$z_t = h(x_t^m, v_t), \quad (30)$$

permitting the independence factorization

$$P(z_t | x_t^p, x_t^m) = P(z_t | x_t^m). \quad (31)$$

We can similarly partition the covariance

$$\bar{\Sigma}_t = \begin{bmatrix} \bar{\Sigma}_t^{(m^2)} & \bar{\Sigma}_t^{(mp)} \\ \bar{\Sigma}_t^{(pm)} & \bar{\Sigma}_t^{(p^2)} \end{bmatrix}. \quad (32)$$

To perform the measurement update we draw samples $\{x_t^{m(j)}\}_{j=1}^M$ from $N(\bar{\mu}_t^m, \bar{\Sigma}_t^m)$. The samples are weighted with the measurement function in equation 31. From these weighted samples we can compute an update for $P(x_t^m | z_0 : z_t)$ using the conventional GPF weighted mean and covariance as in equations 27 and 28. The key idea is to then use the GPF update on the state variables that affect the measurement to propagate a Kalman update to the rest of the state.

To perform a Kalman measurement update we need to know the measurement value z_t , the covariance of the measurement R , and the observation matrix C . Firstly, we set C as a selector matrix for the measurement part of the state

$$C = \begin{bmatrix} I_k & 0_{n-k} \end{bmatrix}. \quad (33)$$

A measurement update on x_t^m would proceed as:

$$K^m = \bar{\Sigma}_t^m (C^m)^T (C^m \bar{\Sigma}_t^m (C^m)^T + R)^{-1} \quad (34)$$

$$\mu_t^m = \bar{\mu}_t^m + K^m (z_t - C^m \bar{\mu}_t^m) \quad (35)$$

$$\Sigma_t^m = (I - K^m C^m) \bar{\Sigma}_t^m \quad (36)$$

Plugging in the identity matrix for C^m , the above equations can be solved for R_t

$$\Sigma_t^m = \bar{\Sigma}_t^m - \bar{\Sigma}_t^m (C^m)^T (C^m \bar{\Sigma}_t^m (C^m)^T + R_t)^{-1} \bar{\Sigma}_t^m \quad (37)$$

$$R_t = (\bar{\Sigma}_t^m - \bar{\Sigma}_t^m (C^m)^T (C^m \bar{\Sigma}_t^m (C^m)^T + R_t)^{-1} \bar{\Sigma}_t^m)^{-1} - \bar{\Sigma}_t^m \quad (38)$$

$$= (\Sigma_t^m - \bar{\Sigma}_t^m)^{-1} \quad (39)$$

where we make use of the matrix inversion lemma between equations 38 and 39.

Using R_t we can now solve for the Kalman gain that would have produced the same change between our prior and posterior covariance using equation 34 and then recover the actual measurement that would have produced the same change in the mean of prior vs. posterior distributions:

$$z_t = K^{m^{-1}} (\mu_t^m - \bar{\mu}_t^m) + \bar{\mu}_t^m. \quad (40)$$

A Kalman gain for the entire state can then be computed using R_t and z_t , and a standard Kalman measurement update performed.

The posterior distribution quantities $\mu_t^{m^{-1}}$ and $\Sigma_t^{m^{-1}}$ are readily available from the GPF measurement update on the measurement part of the state vector. Naively one might use the Gaussian prior from which the samples were drawn to evaluate equations 39 and 40. However, the quantities we care about, R_t and z_t , are obviously sensitive to the difference between the prior and posterior mean and covariance. With a finite number of samples there will be some error between the distribution described by the sample set $\{x_t^{m(j)}\}_{j=1}^M$ and the Gaussian prior. This error will carry over to the weighted sample set which approximates the posterior. We can compensate by using the mean and covariance of the prior sample distribution instead of our analytic expressions for $\bar{\mu}_t^m$ and $\bar{\Sigma}_t^m$. In practice, this substitution makes an enormous difference, particularly with low numbers of particles (which is highly desirable in a real-time application).

Finally, we note that the solutions for R_t and z_t hinge on the invertibility of C^m which is a proxy for the invertibility of our measurement function h in equation 30 with respect to x_t^m . It can be difficult to know *a priori* if the measurement is well conditioned or invertible. If it is not (i.e., if the measurement does not actually contain information about some piece of x_t^m) then the R_t matrix may not be positive-definite, leading to a filter divergence. Thus it is necessary in practice to perform an eigenvalue decomposition on R_t and set any negative eigenvalues to a large constant (implying no information gain along the corresponding eigenvector) and then reconstruct the matrix. This step also protects the algorithm from negative eigenvalues entering through sampling errors.

C. Laser Localization

The likelihood evaluation proceeds according to standard techniques used in 2D localization. We blur the a 3D occupancy map stored as an OctoMap [21] using a Gaussian kernel around occupied cells. To perform particle measurement updates we project the current scan into the map using the sampled particle state, and sum the log-likelihood of the reached cells before exponentiating to obtain a probability with which to weight the particles.

An interesting question is the appropriate partitioning of the state vector for the updates described in the previous section. The use of planar LIDARs to localize in the plane is ubiquitous, suggesting that when working in 3D, laser range scans should at least contain information about the xy plane and χ_z (orientation about the yaw axis of the vehicle). However, in general, a planar slice of a 3D environment may contain some information about the full orientation, but populating the 6D pose space parameterized by χ and Δ with particles may produce limited extra information relative to the computational cost incurred, especially because the direct formulation for our filter based on exponential coordinates, is capable of inferring attitude from accurate position (xyz) measurements. We investigate different choices for x^m in our experiments.

V. IDENTIFYING THE PROCESS NOISE PARAMETERS

Due to the symmetry of the inertial sensors in the IMU, we assume the process noise covariance Q is a diagonal matrix populated as

$$Q = \begin{bmatrix} q_{\text{gyro}}I_3 & 0 \\ 0 & q_{\text{accel}}I_3 \end{bmatrix} \quad (41)$$

and q_{accel} and q_{gyro} are the parameters we wish to identify. Two issues lead to difficulty with finding these values. First, the way the noise projects onto the state changes with the time-varying W_t matrix such that the Q matrix cannot be recovered in closed form simply by summing the outer product of sampled error. More importantly we cannot depend on the availability of ground truth measurements of the measured quantities, since even accurate positioning systems do not directly measure acceleration and angular rate. Further, the behavior of the sensor may be different under actual flight conditions due to vibration and loading effects and thus the values obtained in a static test may not hold.

Nonetheless it is desirable that the model parameters, and especially the process noise parameters, be accurate. For planning purposes we must be able to predict distributions over future states to guarantee safe trajectories. Within the context of state estimation and Monte-Carlo localization, as we describe in section IV-C, it is important that an accurate covariance of the state estimate be maintained when sensor data is sparse or absent, such that the state estimate can be properly distributed to obtain measurements when they become available.

While we do not have access to ground truth acceleration and angular rate with which to estimate the noise parameters, we can post-process data using a Kalman smoothing algorithm to obtain a state history $X = [\hat{x}_0 \ \hat{x}_1 \ \dots \ \hat{x}_T]$ with the error associated with each smoothed state estimate given by

$$\Gamma_t = E [(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T] \quad (42)$$

The key idea in our approach is in projecting the process noise forward over multiple time steps such that the process noise dominates the error in the smoothed estimate, thus allowing us to treat the smoothed estimate as ground truth. This works because the IMU process equations are neutrally stable and thus the perturbing noise results in unbounded

growth in covariance without position updates. The error on the smoothed estimate (with position updates), on the other hand, must be bounded (even if the smoothing occurs with incorrect noise parameters) since the system is observable. Additionally, by projecting the noise forward over multiple steps, the parameters we identify will be suitable for use in planning algorithms that require open-loop predictions [3] and the parameters will work with intermittent measurement functions as may be the case for laser localization in sparse environments.

Using the linearized dynamics from the EKF we can project the filter covariance forward N steps by repeatedly applying equations 3.3. Neglecting the error on the smoothed estimate, we obtain the expression:

$$E [(\hat{x}_{t+N} - \hat{x}_t)(\hat{x}_{t+N} - \hat{x}_t)^T] = \Sigma_{t,N} \quad (43)$$

$$= \sum_{i=0}^{N-1} G_{t+i,N} Q G_{t+i,N}^T \quad (44)$$

where $G_{t,N} = \prod_{j=t+1}^{t+N-1} A_j W_t$. This is an important quantity for our noise identification algorithm because it maps the noise at each time step onto the state vector at time $t+N$. We can see that for identifying characteristics of the process noise, A_t must be neutrally stable and W_t must have full column rank. If A_t is highly unstable, the $\Sigma_{t,N}$ will be overly sensitive to the noise values w_i for small i , whereas if A_t is highly stable, $\Sigma_{t,N}$ will be dominated by larger values of i and thus the forward projection offers little benefit. However, many robotic systems, including our IMU dynamics, exhibit approximately neutrally stable behavior.

For neutrally stable systems, as N gets large we expect $\Sigma_t \gg \Gamma_t$. We can then divide up the dataset X to get $M = T/N$ samples from prediction distributions obtained by subtracting the state at time $t_{\text{end}} = iN + N - 1$ from the state at time $t_{\text{begin}} = iN$ for $i \in [0, M - 1]$. This gives us M samples $y_i = x_{t_{\text{end}}} - x_{t_{\text{begin}}}$ drawn from distributions $N(0, \Sigma_{t_{\text{begin}}, N}) = P(x_{t_{\text{end}}} | x_{t_{\text{begin}}})$. We have a joint likelihood function for our data given the parameters of Q as:

$$P(Y|x_0, Q) = \prod_{i=0}^{M-1} P(x_{iN+N-1} | x_{iN}, Q). \quad (45)$$

We would like to maximize this probability for which we use the log-likelihood function,

$$l(Y|x_0, Q) = -\frac{1}{2} \sum_{i=0}^{M-1} \log |\Sigma_i| + y_i^T \Sigma_i^{-1} y_i. \quad (46)$$

From an intuitive standpoint we are optimizing for the q parameters that would produce the observed drift away from the smoothed estimate given by the samples y_i .

We setup and solve the optimization using standard nonlinear programming techniques. Specifically we use the interior point method implemented in Matlab to solve for the maximum likelihood values of q_{gyro} and q_{accel} . These new values are then used to obtain the Kalman smoothed trajectory, and the process is repeated until convergence.

To identify the noise parameters of the IMU we flew our experimental vehicle (described below) outdoors with a low cost uBlox GPS unit. We also collected a dataset in

TABLE I
NOISE PARAMETER VALUES

Source	Gyro Noise (deg/s)	Accelerometer Noise (g)
Vicon Optimization	0.35	0.0042
GPS Optimization	0.34	0.0182
Manufacturer	0.2	0.005

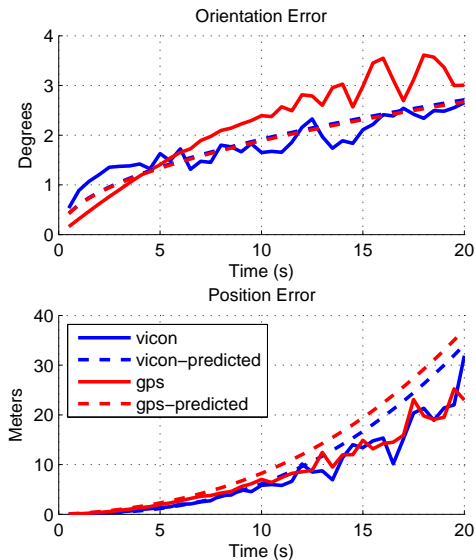


Fig. 3. This figure shows the predicted normed error and the actual normed deviation from the smoothed estimates as a function of lookahead time for the optimization run on both the vicon and GPS datasets. With the optimized values we can accurately predict uncertainty for both estimation and planning purposes.

a high accuracy indoor motion capture system. Optimized noise parameters for a lookahead time of 20 seconds are shown in table I with the manufacturer specified values for comparison.

The optimization on the vicon dataset converges quickly and consistently. However, when the optimization is performed on the GPS dataset the optimization is more sensitive to initial conditions and window size. The vicon system measures attitude directly, thus the smoothed attitude estimate is dominated by the actual measurement. With the GPS dataset, attitude must be inferred from position updates which means the attitude estimate will be more strongly correlated with the IMU noise, thus making it more difficult to find the underlying noise parameter. Additionally, the GPS measurements are subject to time-varying bias which is not modeled in our filter. Nonetheless, the optimization for vicon and GPS converge to nearly identical values for the gyro noise at a 20 second window. The relative sensitivity to the window size for the GPS optimization can be seen in figure 4.

The noise parameters in table I were used to generate the predicted error lines in figure 3. We can see that the predicted error for GPS and vicon are very close for orientation as we would expect from the tabular values. In position, the deviation is also small which is surprising given the large difference in optimized accelerometer noise values. The reason for this is that the positional uncertainty is largely a function of angular uncertainty resulting in the gravity vector being misinterpreted as lateral acceleration. This highlights another difficulty in teasing apart the relative affect of the

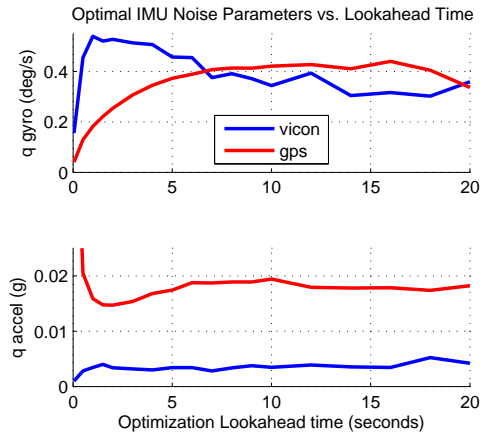


Fig. 4. This figure shows values for q_{gyro} and q_{accel} obtained by optimizing equation 46 for different lookahead times (values of N scaled by sampling frequency) for both GPS and vicon. For small time the optimal noise parameters obtained with GPS are dominated by the error in the smoothed estimates, Γ_t , but we see for large N consistent values are reached. The vicon dataset is less susceptible to this issue. It is interesting to note that as lookahead time increases fewer “samples” are available from a dataset of fixed size, and thus the computed noise values have higher variance, implying some optimal lookahead window to identify the parameters.

noise parameters.

VI. EXPERIMENTAL RESULTS

Our experimental platform consists of a custom built fixed-wing vehicle carrying a payload of a Hokuyo UTM-30LX laser rangefinder, a Microstrain 3DM-GX3-25 IMU, and a 1.6GHz Intel Atom base flight computer. We conducted a number of flight tests in the indoor environment shown in Figure 5(a). While we did not have access to any sort of ground truth state estimates, we were able to test our algorithms on real flight data. The accuracy of our state estimates are validated qualitatively by looking at the accurate reconstruction of the 3D environment by reprojecting the laser points using our state estimates. One such 3D point cloud is shown in Figure 5(b). To get a better feel for the experiments, we invite the interested reader to view the videos of the experiment available on our website: <http://groups.csail.mit.edu/rrg/icra12-agile-flight>

To quantify the error of the state estimator, we aggressively maneuvered the sensing payload in a high accuracy vicon motion capture studio. While the motion of the sensing payload will certainly be very different when the vehicle is flying, the data allows us to evaluate our algorithms with a ground truth comparison. These ground truth state estimates allow us to evaluate the properties of our state estimation algorithm. Results for different number of particles and different partitions of the state vector are summarized in figure 6. We can see that by not partitioning the state and performing standard GPF updates we incur significant computational cost in terms of number of particles to achieve the same level of accuracy. This increase in the number of particles is to be expected given that we are using particles to capture the same correlations that are well captured analytically by the Kalman pseudo-measurement update.

The experiments demonstrate the ability of our algorithm to maintain accurate state estimates in the face of fast motion, with linear velocities up to 9m/s, and angular rates of up to

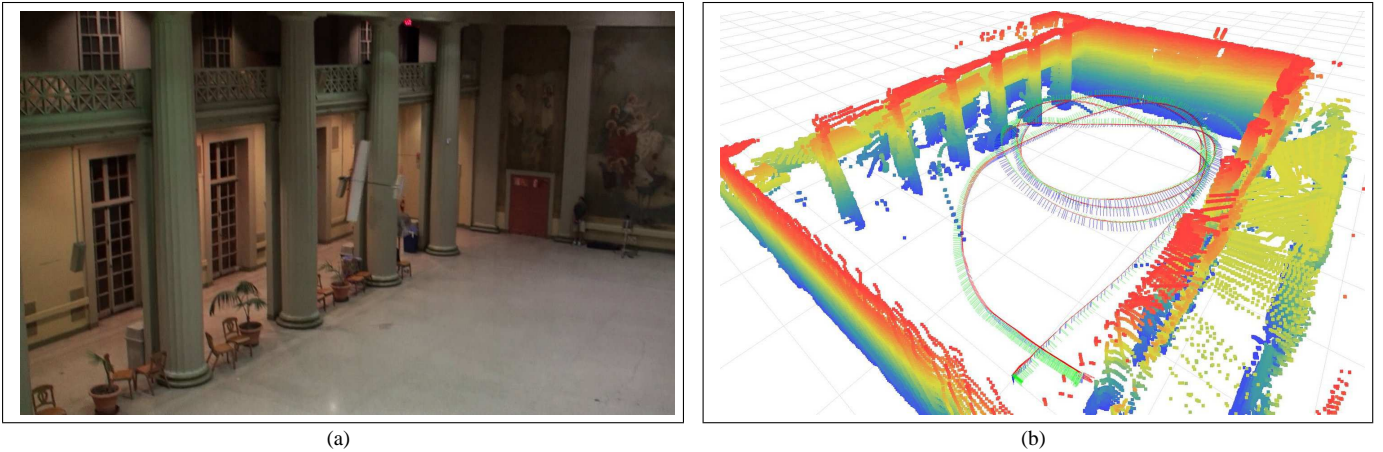


Fig. 5. A picture of the indoor space (a) where we flew our fixed wing vehicle. The space is roughly 12 meters by 20 meters and our vehicle flies between 6 and 10 m/s, thus aggressive maneuvering and tight turning is required to stay airborne. The trajectory flown by the vehicle is shown by the red, green, and blue axes in (b). The quality of the state estimates is evident in the (height colored) point cloud rendered using the state estimates of our algorithm. The floor and ceiling were cropped for visual clarity.

360 degrees per second. While a naive implementation of the GPF measurement update correctly estimates the state of the vehicle with a sufficient number of particles, the required number of particles is dramatically larger than for the partitioned state version. The naive GPF implementation would not be able to run in realtime on board the vehicle given the computation power available.

VII. RELATED WORK

State estimation using Kalman filtering techniques has been extensively studied for vehicles flying outdoors where GPS is available. A relevant example of such a state estimation scheme developed by Kingston et al. [12] involves two Kalman filters where roll and pitch are determined by a filter driven by gyro readings as system inputs while the accelerometer measurements are treated as a measurement of the gravity vector, assuming unaccelerated flight. A separate filter estimates position and yaw using GPS measurements.

This approach is representative of many IMU-based estimators that assume zero acceleration and thus use the accelerometer reading as a direct measurement of attitude (many commercially available IMUs implement similar techniques on board using a complementary filter). While this approach has practical appeal and has been successfully used on a number of MAVs, the zero acceleration assumption does not hold for general flight maneuvering and thus the accuracy of the state estimate degrades quickly during aggressive flight.

Van der Merwe et al. use a sigma-point unscented Kalman filter (UKF) for state estimation on an autonomous helicopter[20]. The filter utilizes another typical approach whereby the accelerometer and gyro measurements are directly integrated to obtain position and orientation and are thus treated as noise perturbed inputs to the filter. Our filter utilizes this scheme in our process model, however we use an EKF with exponential coordinates based attitude representation instead of the quaternions used by Van der Merwe et al.

Techniques to identify the noise parameters relevant for the Kalman filter emerged not long after the original filter, however the most powerful analytical techniques assume

steady state behavior of a linear time invariant system and are thus unsuitable for the time varying system that results from linearizing a nonlinear system [14]. More recent work optimizes the likelihood of a ground-truth projection of the state over the noise parameters but thus requires the system be fitted with a sensor capable of providing ground-truth for training. [1]. Our algorithm does not require the use of additional sensors, or external ground truth.

Laser rangefinders combined with particle filter based localization is widely used in ground robotic systems [19]. While planar lidars are commonly used to estimate motion in the 2D plane, they have also proved useful for localization in 3D environments. Prior work in our group [2], as well as others [18], [6] leveraged a 2D laser rangefinder to perform SLAM from a quadrotor in GPS-denied environments. The systems employ 2D scan-matching algorithms to estimate the position and heading, and redirect a few of the beams in a laser scan to estimate the height. While the systems have demonstrated very good performance in a number of realistic environments, they must make relatively strong assumptions about the motion of the vehicle, and the shape of the environment. Namely, they require walls that are least locally vertical, and a mostly flat floor for height estimation. As a result, the algorithms do not extend to the aggressive flight regime targeted in this paper. Scherer et al. use laser rangefinders to build occupancy maps, and avoid obstacles while flying fast through obstacles [17], however they rely on accurate GPS measurements for state estimation, and do not focus on state estimation.

In addition to the laser based systems for GPS-denied flight, there has been a significant amount of research on vision based control of air vehicles. This includes both fixed wing vehicles [11], as well as larger scale helicopters [4], [10], [8]. While vision based approaches warrant further study, the authors do not address the challenge of agile flight. This is likely to be particularly challenging for vision sensors due to the induced motion blur, combined with the computational complexity of vision algorithms.

Recently, Hesch et al. [7] developed a system that is similar in spirit to ours to localize a laser scanner and INS for localizing people walking around in indoor environments.

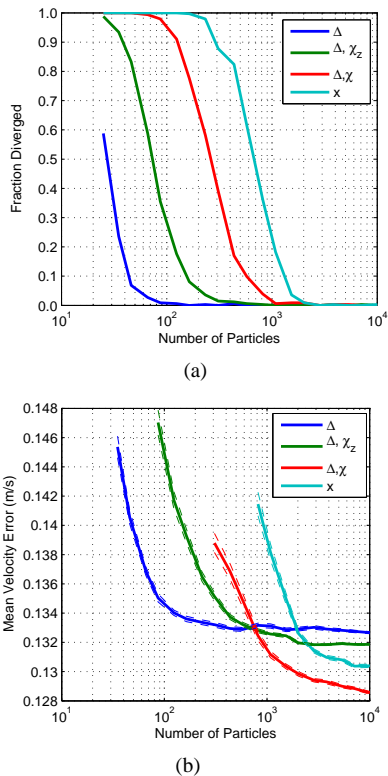


Fig. 6. This figure shows the percentage of trials where the filter diverged (a) and the mean velocity error versus the number of particles used in the GPF (b) for different state partitions in the measurement. As expected, the more states we use in the measurement function the more particles are required to obtain satisfactory estimates. In a naive implementation where the full state is used in the measurement and thus a standard GPF update performed, we require 2000 particles to get similar performance to a measurement update in Δ using only 100 particles. Thus our algorithm yields an effective 20x speedup.

They make a number of simplifying assumptions such as zero velocity updates, that are not possible for a micro air vehicle. Furthermore, they model the environment as a set of planar structures aligned with one of 3 principle axes, which severely limits the types of environments in which their approach is applicable. Our system uses a general occupancy grid representation which provides much greater flexibility of environments.

VIII. CONCLUSION

In this paper we presented a state estimation algorithm for a fixed wing vehicle based on an IMU and laser range scanner. Our algorithm provides a novel extension of the Gaussian particle filter and an exponential coordinates linearization of the IMU dynamics equations. We have demonstrated the performance of our algorithms on two challenging datasets. The quantitative analysis in motion capture clearly shows the advantages of our extensions to the Gaussian particle filter algorithm, while the accurate map generated during the flight tests demonstrate the absolute accuracy of our algorithms.

Integrating the state estimation algorithm with planning and control algorithms to perform closed-loop flight indoors remains for future work. We are particularly interested in using the state estimates in our previously developed partially observable planning frameworks.

In addition to the planning and control extensions, investigation of other sensing modalities such as vision are of great interest. We believe that the filtering framework developed for the laser rangefinder will extend to incorporate additional measurement types, thereby further improving the capabilities of our system.

IX. ACKNOWLEDGEMENTS

This work was supported by ONR under MURI N00014-09-1-1052, MURI N00014-10-1-0936, and Science of Autonomy grant N00014-10-1-0936. Their support is gratefully acknowledged.

REFERENCES

- [1] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun. Discriminative training of Kalman filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [2] A. Bachrach, S. Prentice, R. He, and N. Roy. RANGE: Robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.
- [3] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *IEEE Int. Conf. Robotics and Automation*, May 2011.
- [4] G. Buskey, J. Roberts, P. Corke, and G. Wyeth. Helicopter automation a using a low-cost sensing system. *Computing Control Engineering Journal*, 15(2):8 – 9, April-May 2004.
- [5] A. Coates, P. Abbeel, and A.Y. Ng. Learning for control from multiple demonstrations. In *Int. Conference of Machine Learning*, 2008.
- [6] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *IEEE Int. Conf. Robotics and Automation*, pages 2878–2883, May 2009.
- [7] J.A. Hesch, F.M. Mirzaei, G.L. Mariottini, and S.I. Roumeliotis. A laser-aided inertial navigation system L-INS for human localization in unknown indoor environments. In *IEEE Int. Conf. Robotics and Automation*, pages 5376–5382. IEEE.
- [8] Stefan Hrabar and Gaurav Sukhatme. Vision-based navigation through urban canyons. *Journal of Field Robotics*, 26(5):431–452, 2009.
- [9] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Int. Symp. Robotics Research*, Flagstaff, Arizona, USA, Aug. 2011.
- [10] J. Kelly, S. Saripalli, and G. Sukhatme. Combined visual and inertial navigation for an unmanned aerial vehicle. pages 255–264. 2008.
- [11] J. Kim and S. Sukkarieh. SLAM aided GPS/INS navigation in GPS denied and unknown environments. In *The 2004 International Symposium on GNSS/GPS, Sydney*, pages 6–8, 2004.
- [12] D.B. Kingston and R.W. Beard. Real-time attitude and position estimation for small UAVs using low-cost sensors. *American Institute of Aeronautics and Astronautics*, 2000.
- [13] J. H. Kotecha and P. M. Djuric. Gaussian particle filtering. *IEEE Trans. Signal Processing*, 51(10):2592–2601, October 2003.
- [14] R. Mehra. On the identification of variances and adaptive kalman filtering. *IEEE Trans. Automatic Control*, 15(2):175 – 184, apr 1970.
- [15] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE Int. Conf. Robotics and Automation*, May 2011.
- [16] J.W. Roberts, R. Cory, and R. Tedrake. On the controllability of fixed-wing perching. In *American Control Conference*, pages 2018–2023. IEEE, 2009.
- [17] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma. Flying fast and low among obstacles: Methodology and experiments. *Int. Journal of Robotics Research*, 27(5):549–574, May 2008.
- [18] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *IEEE Int. Conf. Robotics and Automation*, pages 20–25. IEEE.
- [19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence Journal*, 2001.
- [20] R. van der Merwe and E. Wan. Sigma-point Kalman filters for integrated navigation. In *Proc. Institute of Navigation (ION)*, Dayton, OH, June 2004.
- [21] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010.