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

Adam Bry, Abraham Bachrach, Nicholas Roy

***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 maintaing 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 25x fewer particles than a naive implementation to achieve similar accuracy in the state estimate. Using a multi-step forward fitting method we are able to identify the noise parameters of the IMU leading to high quality predictions of the uncertainty associated with the process model. 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 on board sensors is indeed a significant challenge in extending the capabilities of MAVs in real world environments. In addition to providing accurate estimates, the state estimation system must also accurately identify its uncertainty in order to enable planning with applicable partially observable motion planning algorithms [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 grid map.

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

Our process model is a based on an exponential-coordinates extended Kalman filter that is driven by inertial measurements. Unlike conventional system identification techniques that estimate model parameters from sensor data labeled with ground truth states, the model parameters of the IMU process model are estimated using an algorithm that does not require access to ground truth data. 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 computed from a Gaussian Particle Filter state update. This drastically reduces the number of particles required, thereby enabling realtime performance in the face of the computational limitations of the flight computer. We demonstrate the effectiveness of our approach experimentally on a fixed wing vehicle being piloted in a challenging GPS-denied environment.

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 $\begin{bmatrix} \omega_b^T & v_b^T & R & \Delta^T \end{bmatrix}^T$ where $\omega_b = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the angular velocity in body

coordinates, $v_b = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is the linear velocity in body coordinates, R is the rigid body orientation rotation matrix, and $\Delta = \begin{bmatrix} \Delta_x & \Delta_y & \Delta_z \end{bmatrix}^T$ 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 IMU measurements consisting of 3-axis acceleration, 3-axis angular rate measurements, and planar laser range scans. Further, we assume we have access to a 3D map of the environment represented as an occupancy grid. In addition, for planning purposes, we require a covariance over the estimated quantities Σ_{vehicle} such that $\Sigma_{\text{vehicle}} \approx \Sigma_{\text{actual}}$ where Σ_{actual} is the true covariance.

III. IMU Process Model

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

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

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:

$$\bar{\mu}_{t+1} = f(\mu_t, u_t, 0) \tag{2}$$
$$\bar{\Sigma}_{t+1} = A_t \Sigma_t A_t^T + W_t Q W_t^T \tag{3}$$

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 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. $\chi \in \Re^3$ is the perturbation rotation about the body axes. We use the $^\wedge$ notation to denote the skew symmetric matrix formed as:

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

Taking the 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 and 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):

$$\Sigma_\chi = \begin{bmatrix} 12.8117^2 & 0 & 16.2057^2 \\ 0 & 5.7296^2 & 0 \\ 16.2057^2 & 0 & 25.6235^2 \end{bmatrix} \tag{5}$$

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 would pose the constant threat of creating negative eigenvalues, and thus blowing up the estimator. Furthermore, an efficient linearized measurement update as is commonly used in Guassian 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.

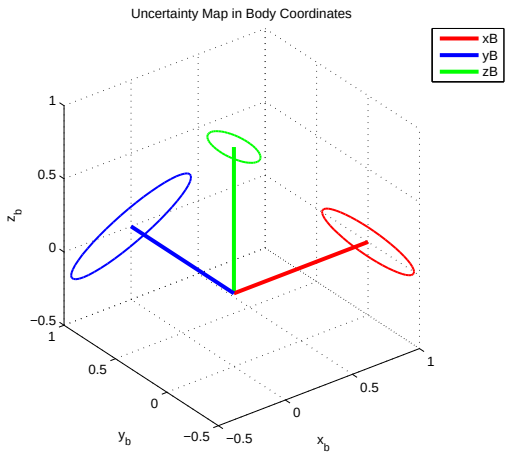


Fig. 2. This figure shows the uncertainty representation in body axes. We see that high variance on the z axis perturbation maps into large motions for the x and y bases. The correlation between x and z uncertainty maps onto the y basis vector.

IV. Process Equations

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

$$\dot{\omega}_b = J^{-1}(-\omega_b \times J\omega_b + \tau_b) \tag{6}$$
$$\dot{v}_b = -\omega_b \times v_b + R^T \bar{g} + a_b \tag{7}$$
$$\dot{R} = R\omega_b^\wedge \tag{8}$$
$$\dot{\Delta} = Rv_b, \tag{9}$$

where τ_b is the torque applied to the body and a_b is the acceleration in body coordinates. Since the IMU returns noisy 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

$$x = \begin{bmatrix} v_b & \chi & \Delta \end{bmatrix} \tag{10}$$
$$u = \begin{bmatrix} u_{\text{gyro}} & u_{\text{accel}} \end{bmatrix} \tag{11}$$
$$w = \begin{bmatrix} w_{\text{gyro}} & w_{\text{accel}} \end{bmatrix} \tag{12}$$
$$\tag{13}$$

Combining this state representation with equations 7-9.

$$f_c(x_t, u_t, w_t) = \begin{bmatrix} \dot{v}_b \\ \dot{R} \\ \dot{\Delta} \end{bmatrix} \tag{14}$$
$$= \begin{bmatrix} -\omega_b \times v_b + R^T \bar{g} + g u_{\text{accel}} \\ R u_{\text{gyro}}^{\wedge} \\ R v_b \end{bmatrix} \tag{15}$$

Taking the appropriate partial derivatives we get:

$$\frac{\partial \dot{v}_b}{\partial x} = \begin{bmatrix} -\omega_b^{\wedge} & (R^T \bar{g})^{\wedge} & 0 \end{bmatrix} \tag{16}$$
$$\frac{\partial \dot{\chi}}{\partial x} = \begin{bmatrix} 0 & -\omega_b^{\wedge} & 0 \end{bmatrix} \tag{17}$$
$$\frac{\partial \dot{\Delta}}{\partial x} = \begin{bmatrix} R & -R v_b^{\wedge} & 0 \end{bmatrix} \tag{18}$$

for a continuous dynamics linearization of:

$$A_c = \frac{\partial f}{\partial x} = \begin{bmatrix} -\omega_b^{\wedge} & (R^T \bar{g})^{\wedge} & 0 \\ 0 & -\omega_b^{\wedge} & 0 \\ R & -R v_b^{\wedge} & 0 \end{bmatrix} \tag{19}$$

and for the input vector we have:

$$\frac{\partial \dot{v}_b}{\partial u} = \begin{bmatrix} v_b^{\wedge} & gI \end{bmatrix} \tag{20}$$
$$\frac{\partial \dot{\chi}}{\partial u} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{21}$$
$$\frac{\partial \dot{\Delta}}{\partial u} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{22}$$
$$W_c = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial \dot{v}_b}{\partial u} \\ \frac{\partial \dot{\chi}}{\partial u} \\ \frac{\partial \dot{\Delta}}{\partial u} \end{bmatrix}. \tag{23}$$

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

$$f(x_t, u_t, 0) = x_t + f_c(x_t, u_t, 0)dt \tag{24}$$
$$A_t = I + A_c dt \tag{25}$$
$$W_t = W_c dt. \tag{26}$$

We integrate the attitude separately as

$$R_{t+1} = R_t R(u_{\text{gyro}}^{\wedge}) \tag{27}$$

V. Identifying the Process Noise Parameters

The process noise covariance Q is a diagonal matrix populated as

$$Q = \begin{bmatrix} q_{\text{gyro}} I_3 & 0 \\ 0 & q_{\text{accel}} I_3 \end{bmatrix} \tag{28}$$

and q_{accel} and q_{gyro} are the parameters we wish to identify. Two issues lead to difficulties with the conventional approach to system identification techniques, where the vehicle trajectory is tracked using an external measurement system, and random deviations from the reference trajectory are modeled as process noise parameters. 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 squaring the deviations. More importantly we cannot depend on the availability of ground truth measurements of the vehicle's deviation from a reference trajectory since it is generally very difficult to obtain realistic flight data for MAVs with labeled ground truth as size and weight restrictions prevent the use of highly accurate GPS systems, and only the smallest and lightest fixed wing vehicles (which are incapable of carrying a meaningful sensor payload) can fly in motion capture systems.

Nonetheless it is critical 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 V-B, it is critical that an accurate covariance of the state estimate be maintained when sensor data is sparse or absent, such that the state estimate can be can be properly distributed to obtain measurements when they become available.

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

$$\Gamma_t = E\left[(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T\right] \tag{29}$$

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. 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 type predictions [3] and the parameters will work with intermittent measurement functions. It is important to note that while the Kalman smoothing algorithm returns a covariance history as well as a state history, it may be arbitrarily far from the actual error Γ_t since the system is nonlinear and the smoother is run with the wrong noise parameters since the true parameters are unknown. However, as long as the system is observable over the smoothing window the error will be bounded.

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:

$$E\left[(\hat{x}_{t+N} - \hat{x}_t)(\hat{x}_{t+N} - \hat{x}_t))^T\right] = \Sigma_{t,N} \tag{30}$$

$$= \sum_{i=0}^{N-1} G_{t+i,N} Q G_{t+i,N}^T \tag{31}$$

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 as it maps the noise at each time step onto the state vector at time $t+N$. 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 $\Sigma_t >> \Gamma_t$. We can then divide up the data set 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:

$$P(Y|x_0, Q) = \prod_{i=0}^{M-1} P(x_{iN+N-1} | x_{iN}, Q). \tag{32}$$

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

$$l(Y|x_0, Q) = -\frac{1}{2} \sum_{i=0}^{M-1} \log|\Sigma_i| + y_i^T \Sigma_i y_i. \tag{33}$$

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}.

A. Partioned GPF 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. In contrast Monte-Carlo techniques are widely used in laser-based localization algorithms [19]. We use the Gaussian Particle Filter (GPF) to perform laser-based measurement updates [13].

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

$$\mu_{t+1} = \frac{\sum_{j=1}^M w_t^{(j)} x_t^{(j)}}{w_t^{(j)}} \tag{34}$$

$$\Sigma_{t+1} = \frac{\sum_{j=1}^M w_t^{(j)} (x_t^{(j)} - \mu_{t+1})(x_t^{(j)} - \mu_{t+1})^T}{w_t^{(j)}}. \tag{35}$$

Assuming the underlying system is linear-Gaussian, the filter is shown to approximate the true distribution arbitrarily well with a large number of samples.

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, and thus using a particle process model adds significant computational burden while also introducing error through sampling.
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.

The state is partitioned as,

$$x_t = \begin{bmatrix} x_t^m & x_t^p \end{bmatrix}, \tag{36}$$

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

$$z_t = h(x_t^m, v_t), \tag{37}$$

permitting the independence factorization

$$P(z_t|x_t^p, x^m) = P(z|x^m). \tag{38}$$

We can similarly partition the covariance

$$\bar{\Sigma}_t = \begin{bmatrix} \bar{\Sigma}_t^{(m^2)} & \bar{\Sigma}_t^{(mp)} \\ \bar{\Sigma}_t^{(pm)} & \bar{\Sigma}_t^{(p^2)} \end{bmatrix}. \tag{39}$$

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 38. 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 34 and 35. 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

$$C = \begin{bmatrix} I_k & 0_{n-k} \end{bmatrix}. \tag{40}$$

A measurement update on x^m would proceed as:

$$K^m = \bar{\Sigma}_t^m (C^m)^T (C^m \bar{\Sigma}_t (C^m)^T + R)^{-1} \tag{41}$$

$$\mu_t^m = \bar{\mu}_t^m + K^m (z_t - C^m \bar{\mu}_t^m) \tag{42}$$

$$\Sigma_t^m = (I - K^m C^m) \bar{\Sigma}_t^m \tag{43}$$

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

$$\Sigma_t^m = \bar{\Sigma}_t^m - \bar{\Sigma}_t^m (C^m)^T (C^m \bar{\Sigma}_t (C^m)^T + R_t)^{-1} \bar{\Sigma}_t^m \tag{44}$$

$$R_t = (\bar{\Sigma}_t^{m^{-1}} - \bar{\Sigma}_t^{m^{-1}} \Sigma_t^m \bar{\Sigma}_t^{m^{-1}})^{-1} - \bar{\Sigma}_t^m \tag{45}$$

$$= (\Sigma_t^{m^{-1}} - \bar{\Sigma}_t^{m^{-1}})^{-1} \tag{46}$$

where we make use of the matrix inversion lemma between equations 45 and 46.

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 41 and then recover the actual measurement that would have produced the same change in the mean of prior vs. posterior distributions:

$$z_t = K^{m^{-1}} (\mu_t^m - \bar{\mu}_t^m) + \bar{\mu}_t^m. \tag{47}$$

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 46 and 47. 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 37 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 doesn't actually contain information about some piece of x_t^m) then our R_t value 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 related errors.

B. Laser Localization

We generate 3D voxel maps of environments using orthogonal Hokuyo lidars mounted on a rolling tripod. A scan matching algorithm developed in prior work [2] by the authors runs using the horizontally mounted lidar while the vertical lidar sweeps out a 3D point cloud of the environment. The point cloud is then projected into a 3D occupancy map computed using OctoMap [21].

The likelihood evaluation proceeds according to standard techniques used in 2D localization. We blur the map using a Gaussian kernel around occupied cells. To perform particle measurement updates we project the scan from that time-step into the map and sum the log-likelihood of the reached cells before exponentiating to obtain a probability with which to weight the particles.

The exponential coordinates framework of the filter provides an easy framework for particle generation. To sample in orientation we sample $\{\chi_t^{m(j)}\}_{j=1}^M$ from $N(0, \bar{\Sigma}_{\chi t}^m)$ and then exponentiate and post multiply our predicted orientation quaternion $q_t^{m(j)} = \bar{q}_t q(\chi_t^{m(j)})$. An interesting question is the appropriate partitioning of the state vector for the updates described in section V-A.

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.

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. To identify the noise parameters of the IMU during realistic flying, we

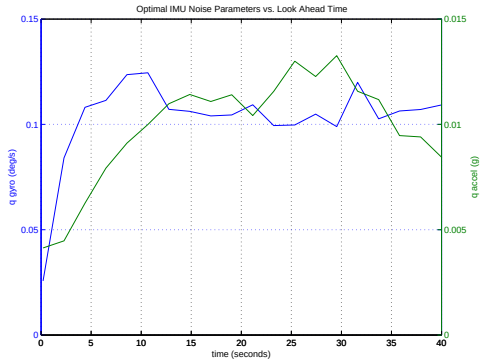


Fig. 3. This figure shows values for q_{gyro} and q_{accel} obtained by optimizing equation 33 for different look ahead values of N. For small N the optimal noise parameters are dominated by the error in the smoothed estimates, Γ_t, but we see for large N consistent values are reached. It is interesting to note that as N increases fewer "samples" are available from a data set of fixed size, and thus more variance appears in the computed noise values, implying some optimal lookahead window to identify the parameters.

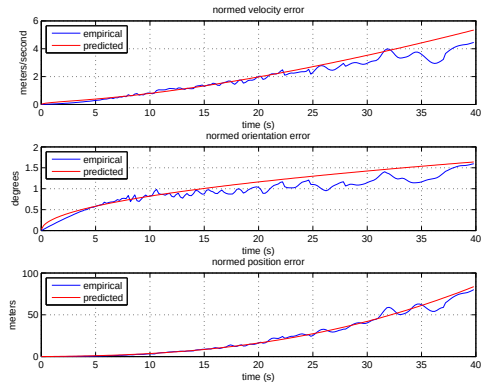


Fig. 4. This figure shows the predicted normed error and the actual normed deviation from the smoothed estimates as a function of lookahead N. With the optimized values we can accurately predict uncertainty for both estimation and planning purposes.

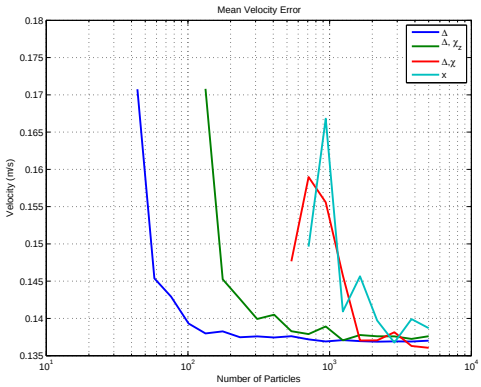


Fig. 5. This figure shows the mean velocity error verses the number of particles used in the GPF for different state partitions in the measurement. As expected, the more states we use in the measurement function the more particles are required to obtain satisfactory estimates. In a naive implementation where the full state is used in the measurement and thus a standard GPF update performed, we require 2500 particles to get similar performance to a measurement update in Δ using only 100 particles. Thus our algorithm yields an effective 25x speedup.

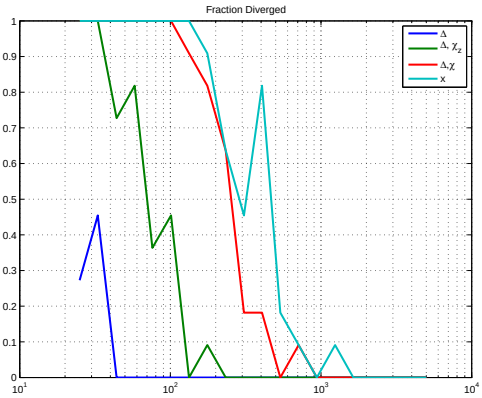


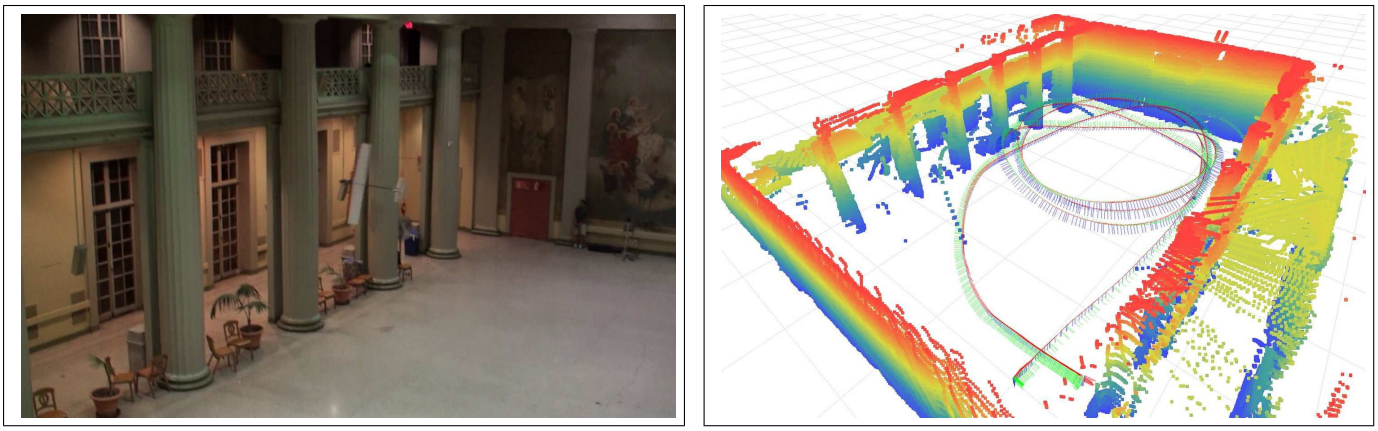
Fig. 6. This figure shows the percentage of trials where the filter diverged as a function of the number of particles for different state partitions. Generally, the more of the state included in the measurement, the more particles are required to prevent divergence.

flew the vehicle outdoors with a low cost uBlox GPS unit. Optimized noise parameters as a function of the lookahead window N are depicted in figure 3. For small N the correlation and error in the smooth estimate corrupts the optimized noise parameters, but as expected for larger N we can recover the true values. Using the optimized values to predict the uncertainty we obtain the results in figure 4.

We conducted a number of flight tests in the indoor environment shown in Figure 7(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 were then 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 7(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. The true trajectory vs. an example state estimate are shown in figure 8. These ground truth state estimates allow us to evaluate the properties of our state estimation algorithm. Results for different number of particles and different state partitioning are summarized in figure 5. 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 makes sense given that we're using particles to capture the same correlations that are well captured analytically by the pseudo Kalman measurement update.

The experiments demonstrate the ability of our algorithm to maintain accurate state estimates in the face of fast motion. 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



(a) (b)

Fig. 7. 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 airborn. 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.

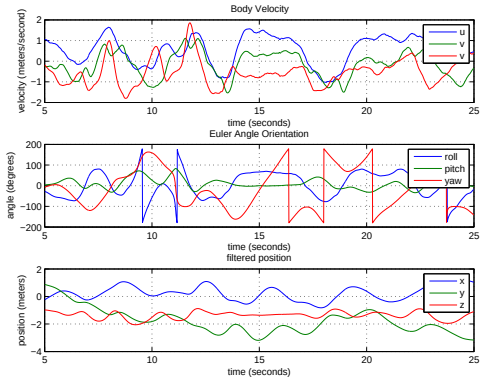


Fig. 8. This figure shows "ground truth" components of the state versus time for "hand flown" data in a VICON motion capture system.

particles is dramatically larger than for the partitioned state version. The naive GPF implementation would not be able to run in real time onboard 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 is determined by a filter driven by gyro measurements as inputs and where 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. 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, which 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 is based on 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. We believe that the presented method for identifying the noise parameters of the IMU will be very useful for planning purposes.

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.

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. Rangerobust 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.