

Observable GNSS-IMU Sliding Window Filtering Using Differential Flatness

Jacob C. Johnson, Randal W. Beard

Abstract—Measurements from the global navigation satellite system (GNSS) and inertial measurement units (IMU) complement one another. Global position is observed from GNSS at low frequencies, and attitude and angular velocity are estimated from IMU data at high frequencies. However, in order to observe global attitude, there must be sufficient motion to excite each axis of the IMU, which may not be possible for large vehicles with constrained dynamics. We propose to model the trajectory estimate on the flat output space of a motion model whose heading is constrained to be in the direction of motion. This mitigates the need to observe heading from the measurements. In this method we use a continuous-time spline and optimize the control points such that the flat output trajectory fits the available GNSS and IMU measurements. We validate the proposed method with simulated data and show that it achieves a higher accuracy and lower solve time than continuous-time estimation on the configuration manifold SE(3).

I. INTRODUCTION

The problem of determining the state (e.g. position, velocity, attitude, etc.) of a dynamic system given a set of measurements from noisy sensors is critical to any navigation scenario that involves autonomous or semi-autonomous agents. The global navigation satellite system (GNSS) was developed for this purpose. Any mobile device equipped with a GNSS receiver can obtain accurate global positioning data in real time. However, GNSS positioning has several limitations: the measurements come at low frequencies (usually less than 5 Hz), are subject to high amounts of noise, and, because the measurements are range-based, do not provide any information about the attitude of the system. The high noise of GNSS positioning can be significantly reduced by using carrier-phase differential corrections, in which case the relative receiver position can be recovered with centimeter-level accuracy. To observe attitude, an inertial measurement unit (IMU) is often used. IMUs measure the angular rate and specific acceleration of the system at high frequencies (above 100 Hz). However, these measurements are plagued by time-varying biases which are not observable using the IMU alone. By fusing IMU with GNSS measurements, it is possible to determine both the position and orientation of the system.

This work has been funded by the Center for Autonomous Air Mobility and Sensing (CAAMS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award No. 2139551, along with significant contributions from CAAMS industry members.

J. C. Johnson is a graduate research assistant at Brigham Young University, Provo, UT, 84602 USA. (e-mail: jjohns99@byu.edu).

R. W. Beard is with the Electrical and Computer Engineering Department, Brigham Young University, Provo, UT 84058 USA. (e-mail: beard@byu.edu).

However, it is well-known that the full global attitude of a GNSS-IMU system is only observable if there is sufficient excitation of the gyroscope and accelerometer, which requires maneuvering the system with a non-constant acceleration and angular rate [1]. These types of motions are not always possible for large vehicles like passenger aircraft and watercraft, and are not convenient (or even acceptable) in many scenarios. A magnetometer can be used to observe global heading, but due to the common presence of external magnetic disturbances, they may not provide sufficient reliability.

Most often, GNSS and IMU measurements are fused using some nonlinear variant of the Kalman filter (KF) [2]–[4]. However, the KF is prone to inconsistencies caused by linearization when applied to nonlinear estimation problems. As an alternative to the KF, discrete batch optimization approaches that optimize the discrete state at a set of measurement times in a sliding time window have been proposed [5], [6]. These use Newton-based optimization to iteratively relinearize the nonlinear model during estimation, thus they are less susceptible to linearization errors. However, these methods require integrating the measurements from high-frequency sensors to keep the number of optimization variables low [7]. Continuous-time batch methods [8] can also be used, which allow for direct incorporation of measurements from multiple asynchronous, high frequency sensors without increasing the number of optimization variables. Typically, these methods use continuous splines or Gaussian processes defined directly on the configuration manifold SE(3) to parameterize the system trajectory.

All of these methods will fail to estimate the full pose of the GNSS-IMU system if there is insufficient motion throughout the trajectory because the global heading and IMU biases will not be observable. However, in continuous-time estimation it is possible to constrain the estimated trajectory such that heading is always in the direction of motion [9]. There are a large class of vehicles whose dynamics have this nonholonomic constraint, including fixed-wing aircraft, wheeled ground vehicles, boats, and some underwater vehicles. If the global heading is constrained, then there is no need to observe it, and the full pose of the system can be estimated using only GNSS and IMU without a changing acceleration or angular rate.

We propose to apply this motion constraint to enable fully-observable GNSS-IMU estimation using differential flatness (DF)-based continuous-time trajectory estimation. In this method, we define the trajectory of the system as a continuous spline in the flat output space of the yaw-constrained motion model and recover the full pose and twist

of the system using DF. DF has often been used in the trajectory planning and control of these types of vehicles (e.g., [10]), but to our knowledge has not been used in trajectory estimation apart from our previous work [9], which introduced DF-based continuous-time estimation generally but did not apply it to a three-dimensional model. This paper is an application of our previous work, and provides the following new contributions:

- a GNSS-IMU fusion method that uses DF-based trajectory estimation to constrain the estimated heading and make the problem fully-observable, and
- simulation results that validate the proposed method and show that it outperforms continuous-time estimation on SE(3) both in terms of accuracy and computation time.

The rest of this paper is outlined as follows. Section II introduces DF-based continuous-time estimation, Section III explains the yaw-constrained motion model that we use, and Section IV shows how this model is applied to the GNSS-IMU estimation problem. Section V presents some simulation results, and concluding remarks are given in Section VI.

II. CONTINUOUS-TIME ESTIMATION USING DIFFERENTIAL FLATNESS

A. Spline-based Estimation

The goal of continuous-time estimation is to determine the continuous trajectory $\mathbf{x}(t)$ taken by a system given a set of N_s measurements $\{\mathbf{z}_s^i\}_{i=1, \dots, N_s}$ acquired from one or more sensors indexed by $s \in S$. The sensor measurements are modeled by

$$\mathbf{z}_s^i = h_s(\mathbf{x}(t_s^i)) + \boldsymbol{\eta}, \quad (1)$$

where t_s^i is the measurement timestamp and $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s)$ is additive Gaussian noise. Because the trajectory can be sampled at any time, the sensors can be asynchronous and run at high frequencies. Define a discrete set of estimation parameters $\{\bar{\mathbf{x}}_m \in \mathcal{E}\}_{m=0, \dots, M}$ in the estimation parameter space $\mathcal{E}$ and a *state interpolation function*

$$\mathbf{x}(t) = \iota(t, \{\bar{\mathbf{x}}_m\}) \quad (2)$$

from which the estimated state can be sampled at any time t . Continuous-time estimation seeks to find the optimal estimation parameters that cause the state trajectory to fit the acquired measurements by solving the problem

$$\arg \min_{\{\bar{\mathbf{x}}_m\}} \sum_{s \in S} \sum_{i=1}^N \|\mathbf{z}_s^i - h_s(\iota(t_s^i, \{\bar{\mathbf{x}}_m\}))\|_{\boldsymbol{\Sigma}_s^{-1}}^2. \quad (3)$$

In this paper, we will use splines to represent continuous trajectories. The estimation parameters are the control points of the spline and the estimation parameter space $\mathcal{E}$ is the configuration manifold of the system. We will consider splines on Euclidean spaces as well as splines on Lie groups.

1) *Euclidean splines*: The shape of a spline in the space $\mathbb{R}^d$ (where d is the number of degrees of freedom of the system) is uniquely determined by a set of control points $\{\bar{\mathbf{p}}_m \in \mathbb{R}^d\}_{m=0, \dots, M}$ and corresponding knot points $\{t_n \in \mathbb{R}\}_{n=0, \dots, M+k}$, where k is the spline order that determines the continuity of the spline¹. The spline can be sampled at any time t using

$$\mathbf{p}(t) = \boldsymbol{\Phi}_n(t) \bar{\mathbf{p}}_n \in \mathbb{R}^d, \quad (4)$$

where n is chosen such that $t \in [t_n, t_{n+1}]$,

$$\bar{\mathbf{p}}_n = [\bar{\mathbf{p}}_{n-k+1}^\top \quad \bar{\mathbf{p}}_{n-k+2}^\top \quad \cdots \quad \bar{\mathbf{p}}_n^\top]^\top, \quad (5)$$

and $\boldsymbol{\Phi}_n(t)$ is a basis matrix that encodes the continuity constraints of the spline. See [11] for more details and a derivation of (4).

2) *Lie groups*: Suppose instead that the configuration manifold is a Lie group G . Define the control and knot points as before, except let the control points be $\bar{g}_m \in G$. Then the spline is sampled using [8]

$$g(t) = \bar{g}_{n-k+1} \prod_{j=1}^M \text{Exp}(b_j(t) \boldsymbol{\Omega}_{n+j-k+1}), \quad (6)$$

$$\boldsymbol{\Omega}_i = \text{Log}(\bar{g}_{i-1}^{-1} \bar{g}_i),$$

where $b_j(t) \in \mathbb{R}$ are basis functions that ensure the spline is $\mathcal{C}^{k-2}$ continuous, and $\text{Exp}(\cdot)$ and $\text{Log}(\cdot)$ are respectively the exponential and logarithmic map of G .

The state interpolation function (2) is the composition of (4) or (6) and their time derivatives. The time derivative $\dot{g}(t)$ represents an infinitesimal perturbation to $g(t)$ expressed in the left tangent space. We do not show how to evaluate this here to due space constraints, but a derivation is given in [12].

B. Differential Flatness

A system that is differentially flat can have all of its states $\mathbf{x} \in \mathcal{X}$ and inputs $\mathbf{u} \in \mathcal{U}$ expressed as functions of a flat output $\mathbf{y} \in \mathcal{Y}$ and its time derivatives [13], i.e.,

$$\mathbf{x}(t) = f_x(\mathbf{y}(t), \dot{\mathbf{y}}(t), \dots), \quad (7)$$

$$\mathbf{u}(t) = f_u(\mathbf{y}(t), \dot{\mathbf{y}}(t), \dots). \quad (8)$$

The flat output space $\mathcal{Y}$ often has lower dimension than the configuration manifold of the system [14], thus it is advantageous to perform estimation on this space rather than the configuration manifold [9]. If the spline is defined in the flat output space, then the state interpolation function (2) is the composition of (7) and (4) or (6) and their time derivatives, depending on whether the flat output space $\mathcal{Y}$ is Euclidean. The estimation problem is still represented by (3), except that the estimation parameter space is now the flat output space. Figure 1 outlines the steps required to evaluate the predicted measurements using differential flatness.

¹A spline of order k is $\mathcal{C}^{k-2}$ continuously differentiable.



Fig. 1: Measurement evaluation using differential flatness.

III. YAW-CONSTRAINED MOTION MODEL

Suppose that a GNSS-IMU sensor is moving through the world as it collects measurements. Let $\mathcal{B}$ denote a coordinate frame that is rigidly attached to the IMU, and $\mathcal{I}$ denote an inertial coordinate frame. The pose of the IMU is

$$\mathbf{T}_{\mathcal{I}}^{\mathcal{B}} = \begin{bmatrix} \mathbf{R}_{\mathcal{I}}^{\mathcal{B}} & -\mathbf{R}_{\mathcal{I}}^{\mathcal{B}} \mathbf{p}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}} \\ \mathbf{0} & 1 \end{bmatrix} \in \text{SE}(3), \quad (9)$$

where $\mathbf{R}_{\mathcal{I}}^{\mathcal{B}} \in \text{SO}(3)$ is the rotation from the inertial frame to the IMU frame and $\mathbf{p}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}} \in \mathbb{R}^3$ is the position of the IMU expressed in the inertial frame. We will consider a class of vehicle dynamics that constrains the heading angle to be fixed in the direction of its velocity $\mathbf{v}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}} = \dot{\mathbf{p}}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}$ (this is similar to a unicycle dynamic model, but extended to three dimensions), i.e.,²

$$\psi = \text{atan2} \left(\frac{\dot{p}_y}{\dot{p}_x} \right), \quad (10)$$

where $\mathbf{p}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}} = [p_x \ p_y \ p_z]^\top$. The roll and pitch angles are left unconstrained. The full rotation $\mathbf{R}_{\mathcal{I}}^{\mathcal{B}}$ is recovered from the roll, pitch, and heading using

$$\mathbf{R}_{\mathcal{I}}^{\mathcal{B}} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix}, \quad (11)$$

and the angular velocity is given by

$$\boldsymbol{\omega}_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}} = \begin{bmatrix} \dot{\phi} - s_\theta \dot{\psi} \\ c_\phi \dot{\theta} + s_\phi c_\theta \dot{\psi} \\ -s_\phi \dot{\theta} + c_\phi c_\theta \dot{\psi} \end{bmatrix}, \quad (12)$$

where $c_\alpha = \cos(\alpha)$ and $s_\alpha = \sin(\alpha)$.

Define the state of the system as $\mathbf{x} = \{\mathbf{T}_{\mathcal{I}}^{\mathcal{B}}, \mathbf{v}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}, \boldsymbol{\omega}_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}}, \mathbf{a}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}\}$, where $\mathbf{a}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}} = \ddot{\mathbf{p}}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}$. Then this system is differentially flat with flat output $\mathbf{y} = \{\mathbf{p}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}, \phi, \theta\}$, where $f_x(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) \triangleq f_x(\ddot{\mathbf{y}})$ is given by (9)-(12), and $\ddot{\mathbf{y}} = \{\ddot{y}, \ddot{\phi}, \ddot{\theta}\}$.

While the nonholonomic constraint (10) may seem restricting, we note that there are a number of systems that have this behavior. For example, most ground vehicles are only capable of traveling in the direction of their orientation due to friction between the tires and the ground. A few other examples include fixed-wing aircraft and boats, in the case that wind or water current is negligible.

IV. GNSS-IMU ESTIMATION

We are interested in fusing position measurements from a GNSS receiver and inertial measurements from an IMU in

²Equation (10) assumes that the $\mathcal{B}$ axes are aligned such that the forward axis points in the direction of motion. This is not generally the case. However, it should be possible to determine the rotation from the $\mathcal{B}$ frame to the frame where the nonholonomic constraint applies. This problem is beyond the scope of this paper.

a way that renders the full state $\mathbf{x}$ of the system observable. In principle, the global rotation $\mathbf{R}_{\mathcal{I}}^{\mathcal{B}}$ of the system is only observable if there is sufficient excitation of the IMU [1]. However, for large vehicles with relatively slow dynamics, maneuvering to excite the IMU may not be possible or desirable. Often a magnetometer may be used to observe the global attitude, but magnetometers are affected by perturbations from static and dynamic magnetic fields near the sensor, thus they cannot be used reliably in many scenarios. Using the differentially flat model presented in Section III means that the global heading is constrained by the position trajectory, thus there is no need to observe it with the sensors. This will allow us to estimate the full pose of the system without needing to excessively maneuver the IMU or use a magnetometer.

A. Measurements

Position measurements from the GNSS receiver are modeled by

$$\mathbf{z}_p^i = h_p(\mathbf{x}(t_p^i)) + \boldsymbol{\eta}_p \triangleq \mathbf{p}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}(t_p^i) + \boldsymbol{\eta}_p, \quad (13)$$

where t_p^i is the timestamp of the i -th measurement (where $i = 1, \dots, N_p$) and $\boldsymbol{\eta}_p \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$. Here we have assumed that the GNSS receiver origin is coincident with the origin of the IMU frame, and that the GNSS and IMU clocks are synchronized. Additionally, we have assumed that enough satellites are in view to fully estimate the position and clock bias, and that the position estimate is not corrupted with multipath interference. While these assumptions may be optimistic, they are commonly used in GNSS-based localization.

The IMU consists of a low-grade MEMS gyroscope and accelerometer. The gyroscope measurements are modeled as

$$\mathbf{z}_g^i = h_g(\mathbf{x}(t_g^i)) + \boldsymbol{\eta}_g \triangleq \boldsymbol{\omega}_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}}(t_g^i) + \mathbf{b}_g(t_g^i) + \boldsymbol{\eta}_g, \quad (14)$$

where t_g^i is the timestamp of the i -th gyroscope measurement (where $i = 1, \dots, N_g$), $\mathbf{b}_g$ is a bias term that walks in time, and $\boldsymbol{\eta}_g \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_g)$. The accelerometer measurements are modeled as

$$\begin{aligned} \mathbf{z}_a^i &= h_a(\mathbf{x}(t_a^i)) + \boldsymbol{\eta}_a \\ &\triangleq \mathbf{R}_{\mathcal{I}}^{\mathcal{B}}(t_a^i) \left(\mathbf{a}_{\mathcal{B}/\mathcal{I}}^{\mathcal{I}}(t_a^i) - g\mathbf{e}_3 \right) + \mathbf{b}_a(t_a^i) + \boldsymbol{\eta}_a, \end{aligned} \quad (15)$$

where g is the gravitational constant, $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$, t_a^i is the timestamp of the i -th accelerometer measurement (where $i = 1, \dots, N_a$), $\mathbf{b}_a$ is an additive bias term that walks in time, and $\boldsymbol{\eta}_a \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_a)$. The gyroscope and accelerometer biases are assumed to follow a Gaussian random walk process, i.e.,

$$\begin{aligned} \dot{\mathbf{b}}_g &= \boldsymbol{\eta}_{gb}, \\ \dot{\mathbf{b}}_a &= \boldsymbol{\eta}_{ab}, \end{aligned} \quad (16)$$

where $\boldsymbol{\eta}_{gb} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{gb})$ and $\boldsymbol{\eta}_{ab} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{ab})$.

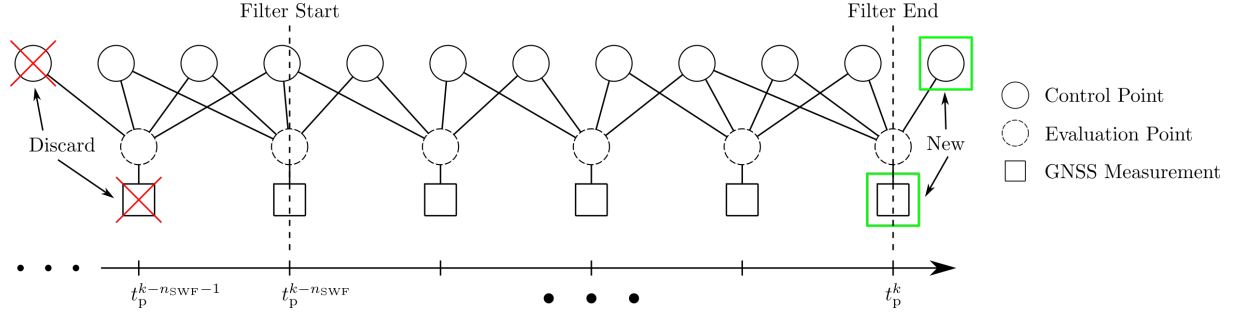


Fig. 2: Factor graph of the continuous-time GNSS-IMU sliding window filter when a GNSS measurement is received (IMU measurements are not shown).

B. Sliding Window Filtering

We will estimate the pose of the system over time from the GNSS and IMU measurements using a spline in the flat output space described in Section III. The position $\mathbf{p}_{B/I}^T$ is Euclidean, thus we will use a Euclidean spline to represent its evolution in time. However, the pitch θ and roll ϕ belong to a space that is diffeomorphic to the Lie group $\text{SO}(2)$. Thus we will use splines on $\text{SO}(2)$ to represent these parameters, and interpolation will be done using equation (6). We will make the knot points of the position and angle splines identical, so that the control points of the three splines can be considered to correspond to one another, and we will define $\bar{\mathbf{y}}_m = \{\bar{\mathbf{p}}_m, \bar{\phi}_m, \bar{\theta}_m\}$ to be the composition of the m -th control point of the position, roll, and pitch splines. In addition, we will need to estimate the gyroscope and accelerometer biases. Because the dynamics of the biases are generally quite slow, we will represent these as constants between consecutive GNSS timestamps, $\mathbf{b}_{\text{gb}}^{k,k-1}$, $\mathbf{b}_{\text{ab}}^{k,k-1}$. Consecutive bias terms will be constrained by the cost terms

$$\mathbf{r}_{\text{gb}}^{k,k-1} = \left\| \mathbf{b}_{\text{gb}}^{k,k-1} - \mathbf{b}_{\text{gb}}^{k-1,k-2} \right\|_{(\delta t_p^2 \Sigma_{\text{gb}})^{-1}}, \quad (17)$$

where $\delta t_p = t_p^k - t_p^{k-1}$. The prior terms $\mathbf{r}_{\text{ab}}^{k,k-1}$ for the accelerometer bias variables are represented similarly. Then the cost function to be minimized is

$$\begin{aligned} \arg \min_{\{\bar{\mathbf{y}}_m\}, \{\mathbf{b}_{\text{gb}}^{k,k-1}\}, \{\mathbf{b}_{\text{ab}}^{k,k-1}\}} & \sum_{s \in \{p, g, a\}} \sum_{i=1}^{N_s} \left\| \mathbf{z}_s^i - h_s(\hat{\mathbf{x}}(t_s^i)) \right\|_{\Sigma_s^{-1}} \\ & + \sum_{k=3}^{N_p} (r_{\text{gb}}^{k,k-1} + r_{\text{ab}}^{k,k-1}), \end{aligned} \quad (18)$$

where $\hat{\mathbf{x}}(t_s^i) = f_x(t_s^i, \{\bar{\mathbf{y}}_m\})$. In order to solve this problem we will need to know the Jacobians of the measurement models with respect to the control points $\bar{\mathbf{y}}_m$. These Jacobians are presented in Appendix A.

The number of measurements that are obtained grows linearly with time. To keep the computational cost of estimation bounded, we will use sliding window filtering, where only the most recent measurements are considered. In particular, we will keep the most recent n_{SWF} GNSS measurements and all IMU measurements that were received since the last GNSS measurement in the filter, and discard old measure-

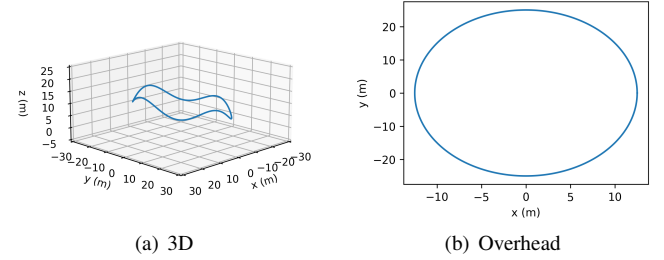


Fig. 3: Simulated sinusoidal trajectory.

ments as new measurements are received³. Figure 2 shows an example factor graph of the continuous-time GNSS-IMU sliding window filter.

V. SIMULATION RESULTS

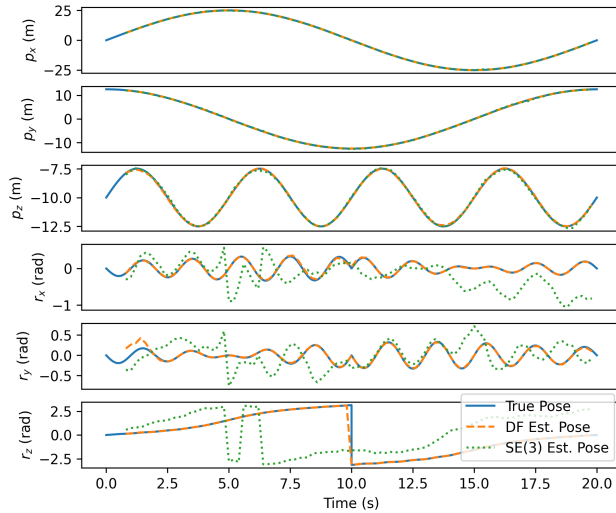
We simulated the GNSS-IMU estimation problem using the following parameters:

$$\begin{aligned} \Sigma_p &= \text{diag}(2.5 \times 10^{-3}, 2.5 \times 10^{-3}, 1 \times 10^{-2}) \text{ m}^2, \\ \Sigma_g &= 1 \times 10^{-6} \mathbf{I} (\text{rad/s})^2, \quad \Sigma_a = 2.5 \times 10^{-5} \mathbf{I} (\text{m/s}^2)^2, \\ \Sigma_{\text{gb}} &= 2.5 \times 10^{-5} \mathbf{I} (\text{rad/s}^2)^2, \quad \Sigma_{\text{ab}} = 2.5 \times 10^{-5} \mathbf{I} (\text{m/s}^3)^2, \\ g &= 9.805 \text{ m/s}^2. \end{aligned} \quad (19)$$

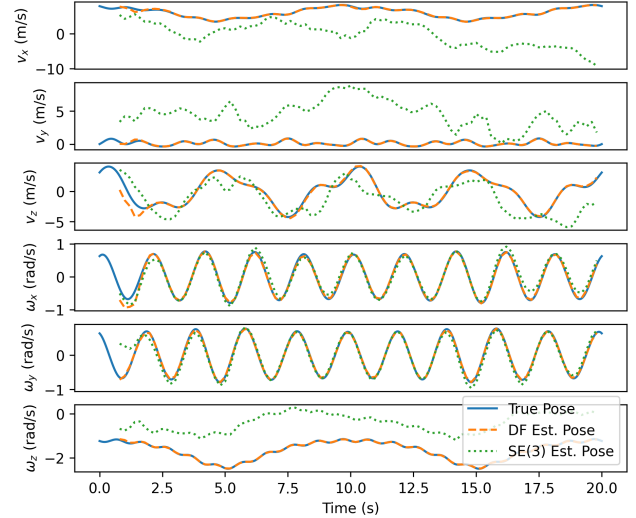
The GNSS noise is typical of measurements that have been corrected using differential signals from a base station receiver. GNSS measurements were simulated at 5 Hz and IMU measurements were simulated at 100 Hz. For estimation, we used a spline of order $k = 6$ and uniform spacing between knot points $\delta t = 0.3$ s, and we set $n_{\text{SWF}} = 20$, which means that the most recent 4 seconds of data was optimized in the sliding window filter. The filter was run whenever a new GNSS measurement was available (every 0.2 s). We simulated a 20 s sinusoidal trajectory that constrained the heading to be in the direction of the velocity vector, shown in Figure 3. To solve problem (3), we used Levenberg-Marquardt optimization implemented with the Ceres solver [15].

We compared our differential flatness-based method to spline continuous-time estimation on $\text{SE}(3)$, where the heading was left unconstrained. The $\text{SE}(3)$ spline also used $k = 6$

³In order to keep the filter consistent, these measurements and the control points connected to them should be properly marginalized before discarding. We consider this to be beyond the scope of the paper.



(a) Pose



(b) Twist

Fig. 4: True versus estimated pose and twist for the DF-based method and for a spline on SE(3). The rotation plots represent $\text{Log}(\mathbf{R}_T^B)$.

Spline Type	Pos. RMSE (m)	Rot. RMSE (rad)	Twist RMSE
Flat Output	0.0565	0.0258	0.1206
SE(3)	0.1688	0.5307	1.5677

TABLE I: Trajectory Error Comparison

and $\delta t = 0.3$ s. Plots of the estimated pose and twist versus truth are shown in Figure 4, and the position, rotation, and twist root-mean-squared error (RMSE) for both methods are shown in Table I. The DF-based method significantly outperformed the spline on SE(3) in terms of trajectory quality. This is because the trajectory did not provide enough acceleration for the full global rotation to become observable from the IMU measurements. The DF-based method, however, did not need to observe the global heading because of the nonholonomic constraint applied to the trajectory estimate, thus it was able to obtain high rotational accuracy.

Figure 5 shows box plots of the time required to solve the estimation problem for each step of the estimator. The red line is at 0.2 s, which is the maximum solve time allowed for real-time performance. All but two of the steps offered faster-than-real-time performance for the DF-based method, whereas only a quarter of the SE(3) steps solved in less than 0.2 s. This shows that the DF-based method is capable of running in real-time (with an update rate of 5 Hz), whereas the SE(3) spline method is far too slow. One reason that the SE(3) spline required so much time to solve is that computing time derivatives and Jacobians of splines on Lie groups is computationally intensive. Using DF allows us to avoid using a spline on SE(3), which is why our method solved much faster. This is an additional benefit of using DF-based estimation in the GNSS-IMU fusion problem.

VI. CONCLUSION

In this paper we presented a DF-based continuous-time solution to the GNSS-IMU estimation problem. We introduced a model that constrains the heading of the system to be in

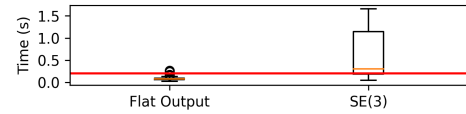


Fig. 5: Solve time distributions for each sliding window filter run, pictured for DF-based estimation (left) and SE(3) spline estimation (right).

the direction of motion, allowing the full pose of the system to be observed even with limited motion. We validated this method in simulation and showed that it can estimate the full pose of the system even when there is a lack of sufficient motion to observe the global heading. These results indicate that our method outperforms continuous-time estimation on SE(3) in terms of both accuracy and computation time.

The results in this paper are largely preliminary. There are several additional steps that should be taken to fully validate the proposed method. First, it is unlikely that the IMU axes will be completely aligned with the coordinate system in which the nonholonomic constraint applies. The rotation between these two coordinate systems will need to be calibrated prior to online estimation. Second, control points and measurements that leave the sliding window should be marginalized rather than discarded to avoid overconfidence. Third, while our results show that this method is capable of real-time performance, the update rate is far lower than that of e.g., a Kalman filter would be. Improvements in computation time would be highly beneficial. Finally, this method will need to be validated online using true hardware data collected onboard a yaw-constrained vehicle. In the case that the vehicle dynamics do not exactly obey the constraint (10) (for example if a ground vehicle experiences wheel slip or a fixed-wing aircraft undergoes a crabbing motion) the quality of the trajectory estimate will degrade. More experiments must be done to understand how much degradation will take place and whether the proposed method is suitable in these scenarios.

A. Jacobians

In this Appendix we present the Jacobians required to solve (3) using Levenberg-Marquardt optimization. This is trivial for the GNSS model,

$$\frac{\partial \mathbf{z}_p}{\partial \mathbf{p}_m} = \Phi_n(t) [\cdots \mathbf{0} \quad \mathbf{I} \quad \mathbf{0} \quad \cdots]^\top \quad (20)$$

if $m \in [n - k + 1, n]$, $\mathbf{0}$ otherwise. Here $\mathbf{I}$ goes in the $m - (n - k + 1)$ -th spot, and $\Phi_n(t)$ was defined in (4).

The Jacobians of the IMU measurement models are not so straightforward. We start with the gyroscope. We have

$$\begin{aligned} \frac{\partial \mathbf{z}_g}{\partial \phi_m} &= \frac{\partial \omega_{B/I}^B}{\partial \phi} \frac{\partial \phi}{\partial \phi_m} + \frac{\partial \omega_{B/I}^B}{\partial \dot{\phi}} \frac{\partial \dot{\phi}}{\partial \phi_m}, \\ \frac{\partial \mathbf{z}_g}{\partial \theta_m} &= \frac{\partial \omega_{B/I}^B}{\partial \theta} \frac{\partial \theta}{\partial \theta_m} + \frac{\partial \omega_{B/I}^B}{\partial \dot{\theta}} \frac{\partial \dot{\theta}}{\partial \theta_m}. \end{aligned} \quad (21)$$

Let $\xi = [\phi \quad \theta \quad \psi]^\top$. Then

$$\begin{aligned} \frac{\partial \omega_{B/I}^B}{\partial \xi} &= \begin{bmatrix} 0 & -c_\theta \dot{\psi} & 0 \\ -s_\phi \dot{\theta} + c_\phi c_\theta \dot{\psi} & -s_\phi s_\theta \dot{\psi} & 0 \\ -c_\phi \dot{\theta} - s_\phi c_\theta \dot{\psi} & -c_\phi s_\theta \dot{\psi} & 0 \end{bmatrix}, \\ \frac{\partial \omega_{B/I}^B}{\partial \dot{\xi}} &= \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}. \end{aligned} \quad (22)$$

The Jacobians of $\phi, \theta, \dot{\phi}$, and $\dot{\theta}$ with respect to their spline control points are obtained similarly to (20). We also need

$$\frac{\partial \mathbf{z}_g}{\partial \mathbf{p}_m} = \frac{\partial \omega_{B/I}^B}{\partial \dot{\psi}} \left(\frac{\partial \dot{\psi}}{\partial \dot{\mathbf{p}}_{B/I}^T} \frac{\partial \dot{\mathbf{p}}_{B/I}^T}{\partial \mathbf{p}_m} + \frac{\partial \dot{\psi}}{\partial \ddot{\mathbf{p}}_{B/I}^T} \frac{\partial \ddot{\mathbf{p}}_{B/I}^T}{\partial \mathbf{p}_m} \right). \quad (23)$$

Noting that

$$\dot{\psi} = -\frac{1}{\|\dot{\mathbf{p}}\|} \dot{\mathbf{p}}^\top \mathbf{1}^\wedge \ddot{\mathbf{p}}, \quad (24)$$

where $\dot{\mathbf{p}} = [p_x \quad p_y]^\top$ and $\mathbf{1}^\wedge = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, we can compute

$$\begin{aligned} \frac{\partial \dot{\psi}}{\partial \dot{\mathbf{p}}_{B/I}^T} &= \left[\frac{1}{\|\dot{\mathbf{p}}\|^2} \ddot{\mathbf{p}}^\top \mathbf{1}^\wedge - \frac{2}{\|\dot{\mathbf{p}}\|^4} (\ddot{\mathbf{p}}^\top \mathbf{1}^\wedge \dot{\mathbf{p}}) \dot{\mathbf{p}}^\top \quad \mathbf{0} \right] \\ \frac{\partial \dot{\psi}}{\partial \ddot{\mathbf{p}}_{B/I}^T} &= \left[-\frac{1}{\|\dot{\mathbf{p}}\|^2} \dot{\mathbf{p}}^\top \mathbf{1}^\wedge \quad \mathbf{0} \right]. \end{aligned} \quad (25)$$

Finally we need the Jacobians of the accelerometer model. We have

$$\frac{\partial \mathbf{z}_a}{\partial \phi_m} = -(\mathbf{R}_I^B \mathbf{a}_{B/I}^T - g \mathbf{e}_3)^\wedge \frac{\partial \mathbf{R}_I^B}{\partial \phi} \frac{\partial \phi}{\partial \phi_m}, \quad (26)$$

where $(\cdot)^\wedge$ is the wedge map of $\text{SO}(3)$. The Jacobians for θ are obtained similarly. Noting that

$$\mathbf{R}_I^B = \text{Exp}(-\phi \mathbf{e}_1) \text{Exp}(-\theta \mathbf{e}_2) \text{Exp}(-\psi \mathbf{e}_3), \quad (27)$$

it is possible to find that

$$\frac{\partial \mathbf{R}_I^B}{\partial \xi} = -[\mathbf{e}_1 \quad \text{Exp}(-\phi \mathbf{e}_1) \mathbf{e}_2 \quad \text{Exp}(-\phi \mathbf{e}_1) \text{Exp}(-\theta \mathbf{e}_2) \mathbf{e}_3], \quad (28)$$

where we use the method described in [16] to represent Jacobians of Lie group elements. Finally,

$$\begin{aligned} \frac{\partial \mathbf{z}_a}{\partial \mathbf{p}_m} &= -(\mathbf{R}_I^B \mathbf{a}_{B/I}^T - g \mathbf{e}_3)^\wedge \frac{\partial \mathbf{R}_I^B}{\partial \psi} \frac{\partial \psi}{\partial \dot{\mathbf{p}}_{B/I}^T} \frac{\partial \dot{\mathbf{p}}_{B/I}^T}{\partial \mathbf{p}_m} \\ &\quad + \mathbf{R}_I^B \frac{\ddot{\mathbf{p}}_{B/I}^T}{\partial \mathbf{p}_m}, \end{aligned} \quad (29)$$

where

$$\frac{\partial \psi}{\partial \dot{\mathbf{p}}_{B/I}^T} = \left[\frac{1}{\|\dot{\mathbf{p}}\|^2} \dot{\mathbf{p}}^\top \mathbf{1}^\wedge \quad \mathbf{0} \right]. \quad (30)$$

REFERENCES

- [1] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. Speyer, "Observability of error states in GPS/INS integration," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 731–743, 2005.
- [2] C. Hide, T. Moore, and M. Smith, "Adaptive Kalman filtering algorithms for integrating GPS and low cost INS," in *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556)*, 2004, pp. 227–233.
- [3] P. Zhang, J. Gu, E. Milios, and P. Huynh, "Navigation with IMU/GPS/digital compass with unscented Kalman filter," in *IEEE International Conference Mechatronics and Automation*, 2005, vol. 3, 2005, pp. 1497–1502 Vol. 3.
- [4] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S156625350400065X>
- [5] F. Gierbach, J. D. Hol, G. Bellusci, and M. Diehl, "Optimization-based sensor fusion of GNSS and IMU using a moving horizon approach," *Sensors*, vol. 17, no. 5, 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/5/1159>
- [6] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter," *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 315–331, 2021. [Online]. Available: <https://navi.on.org/content/68/2/315>
- [7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [8] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," *International Journal of Computer Vision*, vol. 113, no. 3, pp. 208–219, 2015.
- [9] J. C. Johnson, J. G. Mangelson, and R. W. Beard, "Continuous-time trajectory estimation for differentially flat systems," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 145–151, 2023.
- [10] E. Tal, G. Ryou, and S. Karaman, "Aerobatic trajectory generation for a vtol fixed-wing aircraft using differential flatness," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4805–4819, 2023.
- [11] K. Qin, "General matrix representations for B-splines," in *Proceedings Pacific Graphics' 98. Sixth Pacific Conference on Computer Graphics and Applications (Cat. No. 98EX208)*. IEEE, 1998, pp. 37–43.
- [12] J. Johnson, J. Mangelson, T. Barfoot, and R. Beard, "Continuous-time trajectory estimation: A comparative study between Gaussian process and spline-based approaches," *arXiv preprint arXiv:2402.00399*, 2024.
- [13] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "On differentially flat nonlinear systems," *IFAC Symposia Series*, no. 7, pp. 159–163, 1993.
- [14] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC*, vol. 57-1, pp. 349–357, 1995.
- [15] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: <https://github.com/ceres-solver/ceres-solver>
- [16] J. Sola, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.