

Galilean State Estimation for Inertial Navigation Systems with Unknown Time Delay

Giulio Delama

Control of Networked Systems Group,
University of Klagenfurt,
9020 Klagenfurt, Austria
Email: giulio.delama@ieee.org

Martin Scheiber

Control of Networked Systems Group,
University of Klagenfurt,
9020 Klagenfurt, Austria
Email: martin.scheiber@ieee.org

Yixiao Ge

System Theory and Robotics Group,
Australian National University,
ACT, 2601, Australia
Email: yixiao.ge@anu.edu.au

Tarek Hamel

I3S, CNRS, Université Côte d'Azur,
and Institut Universitaire de France (IUF),
06900 Sophia Antipolis, France
Email: thamel@i3s.unice.fr

Stephan Weiss

Control of Networked Systems Group,
University of Klagenfurt,
9020 Klagenfurt, Austria
Email: stephan.weiss@ieee.org

Robert Mahony

System Theory and Robotics Group,
Australian National University,
ACT, 2601, Australia
Email: robert.mahony@anu.edu.au

Abstract—Many Inertial Navigation Systems (INS) use Global Navigation Satellite System (GNSS) position as the primary measurement to drive filter performance and bound error growth. However, commercial-grade GNSS receivers introduce unknown measurement delays ranging from 50 ms to 300 ms depending on sensor quality and operating mode. Such time delays can significantly degrade INS performance unless they are explicitly compensated for. Existing algorithms commonly estimate this delay offline, run the filter concurrently with GNSS measurements using buffered Inertial Measurement Unit (IMU) data, and predict the current state by forward-integrating buffered inertial measurements via IMU preintegration. The state-of-the-art online method is an Extended Kalman Filter (EKF) that explicitly models the time delay as a state parameter, which defines the preintegration duration. This paper introduces a novel geometric framework for modeling time-delayed INS, in which Galilean symmetry is leveraged to provide a joint representation of space and time for consistent state estimation. An Equivariant Filter (EqF) is derived for the coupled estimation of navigation states and time delay. Validation is performed on two fixed-wing Uncrewed Aerial Vehicles (UAV) with GNSS time lags of 90 ms and 120 ms. The test flights last two to three minutes. Simulations further investigate delays up to 500 ms and provide a statistical comparison against the state-of-the-art EKF. Results show that the EqF preserves accuracy and consistency, while the EKF lacks consistency and its performance degrades significantly with increasing measurement delays.

I. INTRODUCTION AND RELATED WORK

Inertial Navigation Systems (INS) estimate the position, velocity, and orientation of a vehicle by fusing high-rate accelerometer and gyroscope measurements from an Inertial Measurement Unit (IMU) with data from other sensors, such as Global Navigation Satellite Systems (GNSS), magnetometers, barometers, visual- or range-based motion sensors. These systems are widely used in robotic, aerospace, marine, and automotive applications. However, delays in sensor measurements due to sensor processing, communication latency, or asynchronous sampling introduce measurement uncertainty that compromises INS accuracy. Measurements are subject to

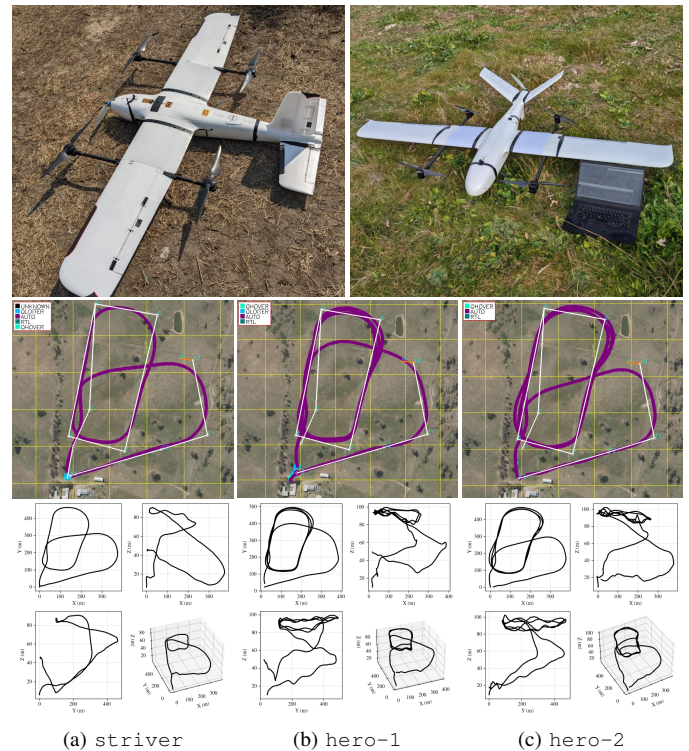


Fig. 1. Overview of the real-world flight experiments. The top row shows the two fixed-wing UAVs used for data collection, the Makeflyeasy *Striver* (left) and *Hero* (right). The remaining panels show the trajectories of the three dataset sequences (a) *striver*, (b) *hero-1*, and (c) *hero-2*, visualized in map view (middle row) and as 3D trajectory reconstructions (bottom rows).

both *lag*, the systematic delay between the actual measurement and the moment it becomes available for processing, and *jitter*, the smaller, variable fluctuations in arrival time caused by factors such as bus congestion or communication load. For example, position measurements from commercial-grade GNSS typically have lags in the range of 50 ms - 300 ms

associated with the coherent integration time of the receiver, and jitter in the order of 20 ms - 30 ms associated with thread priority in the avionics system. Although algorithms exist to estimate and compensate for jitter, accurately determining the time lag is more challenging. Even small delays can introduce biases and inconsistencies in the estimator, emphasizing the need for INS algorithms that account for these timing offsets.

Several delay-aware state estimation methods have been developed. Classical optimization-based techniques estimate both navigation states and sensor delays by iteratively minimizing measurement residuals over batch data. Examples include a general multi-sensor temporal calibration by Kelly and Sukhatme [13], and camera-IMU systems from Li and Mourikis [19], Kelly et al. [14], and Song et al. [24]. On the other hand, Larsen et al. [18] and Nilsson et al. [22] introduced Extended Kalman Filters (EKF) that incorporate delayed measurements by performing extrapolation and interpolation to manage asynchronous updates. Fatehi and Huang [2] studied EKF approaches for multi-rate fusion with irregular sampling and variable measurement delays, highlighting the importance of algorithms that can fuse slow-rate, delayed data with fast measurements in a statistically principled way. Frei et al. [8] developed a robust navigation filter for fusing delayed, asynchronous measurements from multiple sensors in spacecraft rendezvous, modifying the EKF update to fuse delayed data effectively in practice. Khosravian et al. [16] proposed an observer-predictor structure for invariant systems with delayed outputs, using geometric properties to combine delayed measurements with state predictions. More recently, van Goor et al. [26] developed a nonlinear INS observer that compensates for temporal offsets in position measurements using recursively computed delay matrices, achieving almost-global asymptotic and locally exponential stability. These methods focus on processing asynchronous and delayed data within a given filter, rather than explicitly estimating unknown delays online.

Kim et al. [17], Guo et al. [11], and Mina et al. [21] proposed EKF-based methods to estimate measurement time delays online, jointly updating navigation states and the time-delay state during recursive propagation. However, while these approaches can improve accuracy in specific applications, Kelly et al. [15] show that including online time-delay estimation in recursive filters introduces fundamental structural issues that lead to bias and inconsistency, which cannot be fully mitigated by tuning filter noise variances. To the authors' knowledge, there is no existing filter-based online method that does not suffer from the inherent inconsistencies and instabilities identified by Kelly et al. [15].

Recent research has shown the advantages of exploiting equivariant symmetries in state estimation. The Equivariant Filter (EqF) by van Goor et al. [25] has been successfully applied to a wide range of navigation problems. Relevant to this work, Ge et al. [9] extended the EqF to general discrete-time systems, while Fornasier et al. [3, 4, 5, 6, 7] leveraged it for biased-INS, establishing equivariant symmetries that couple navigation states and IMU biases. Scheiber et al. [23] applied it to multi-GNSS navigation. By preserving the

intrinsic geometry of the state space, the EqF outperforms the classical EKF, providing a larger basin of attraction with faster convergence, improved accuracy, and consistency.

In this paper, we leverage *Galilean symmetry* to develop a geometric framework that enables consistent online estimation of navigation states and unknown time delays in INS. Galilean symmetry captures the intrinsic spatio-temporal symmetry of the inertial navigation problem and naturally provides a formulation that couples temporal offset with position and velocity states. Recently, Kelly [12] formalized the *Galilean matrix Lie group* in a technical report, while Delama et al. [1] leveraged Galilean symmetry to model discrete-time INS and develop a novel *IMU preintegration* method improving consistency and reducing linearization errors. Mahony et al. [20] provided a robotics-oriented exposition applying the Galilean group to inertial navigation, kinematics, and sensor fusion under *temporal uncertainty*. The Galilean formulation allows us to formalize the INS state estimation problem for a general scenario with delayed measurements that have both lag and jitter. We use this framework to derive an EqF that jointly estimates navigation states and time delays for GNSS-IMU navigation. Real-world flight experiments with two fixed-wing Uncrewed Aerial Vehicles (UAV) are carried out to compare the performance of the proposed EqF against EKFs with and without offline delay compensation and an EKF with online delay estimation. For GNSS delays of 90 ms and 120 ms, the EqF accurately estimates both navigation states and delays online, outperforming the EKF with delay compensation (where the delay estimate is computed using offline batch optimization). To provide further statistical analysis, we conduct Monte Carlo simulations with delays up to 500 ms, showing that the EqF maintains accurate and consistent estimation, whereas the state-of-the-art EKF with online time-delay estimation lacks consistency and exhibits unstable behavior.

The contributions of this paper can be outlined as follows:

- A geometric framework based on Galilean symmetry for modeling inertial motion under gravity for multiple rigid bodies with relative temporal offsets, allowing spatio-temporal state estimation and enabling a principled representation of INS with delayed measurements (Sec. II).
- Application of this framework to a minimal IMU-GNSS setup with delayed position measurements and derivation of an EqF for online joint estimation of navigation states and unknown GNSS time delays (Sec. III).
- Experimental evaluation of the proposed EqF with real-world UAV flights, and with Monte Carlo simulations providing a statistical comparison against a state-of-the-art EKF and showing improved consistency and accuracy across a range of GNSS time delays (Sec. IV).

II. GALILEAN SPATIO-TEMPORAL STATE ESTIMATION

This section presents a geometric framework for modeling inertial motion with temporal offsets using *Galilean symmetry*. Notation and basic concepts for representing spatio-temporal transformations between inertial frames are given in Sec. II-A, and delayed inertial frames for rigid bodies under gravity are

modeled in Sec. II-B, distinguishing *intra-body* and *cross-body* transformations. Their *continuous-time kinematics* are derived in Sec. II-C, and Sec. II-D outlines how the framework can be applied to state estimation with delayed measurements.

A. Notation and Mathematical Preliminaries

Vectors are denoted by bold lowercase letters $\mathbf{x}$, matrices by bold capital letters $\mathbf{X}$, and group elements by regular letters X or x . The n -dimensional identity matrix is $\mathbf{I}_n \in \mathbb{R}^{n \times n}$, and the $n \times m$ zero matrix is $\mathbf{0}_{n \times m} \in \mathbb{R}^{n \times m}$. Coordinate frames are denoted by capital letters inside curly brackets $\{X\}$. In this study, frames of reference at different times are considered. A subscript t on a frame $\{X_t\}$ specifies the associated time. Calligraphic symbols $\mathcal{M}$ denote manifolds, e.g., $\mathcal{SO}(3)$, while bold symbols $\mathbf{G}$ denote Lie groups, e.g., $\mathbf{SO}(3)$.

Inertial frames are reference frames subject to Newton's laws of motion. They are defined by spatial coordinates (orientation, position, and velocity) and a temporal coordinate (time). The set of all inertial frames is related by *Galilean symmetry*, capturing the invariance of Newtonian mechanics under frame transformations. A *Galilean frame* (or *transformation*) is a map that expresses the coordinates of one inertial frame in terms of another. Specifically, for two inertial frames $\{A\}$ and $\{B\}$, the Galilean frame ${}^A\mathbf{F}_B \in \mathcal{Gal}(3)$ expressed in homogeneous coordinates is represented as

$${}^A\mathbf{F}_B = \begin{bmatrix} {}^A\mathbf{R}_B & {}^A\mathbf{v}_B & {}^A\mathbf{p}_B \\ \mathbf{0}_{1 \times 3} & 1 & {}^A t_B \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5 \times 5}, \quad (1)$$

where ${}^A\mathbf{R}_B \in \mathcal{SO}(3)$ is a rotation matrix encoding the relative orientation, ${}^A\mathbf{v}_B \in \mathbb{R}^3$ the relative *inertial* velocity, ${}^A\mathbf{p}_B \in \mathbb{R}^3$ the relative position, and ${}^A t_B \in \mathbb{R}$ the time offset between the frames. These homogeneous matrices define the *Galilean matrix Lie group* $\mathcal{Gal}(3)$ that provides a principled framework for representing coupled spatial and temporal relationships between inertial frames. An element of the Galilean group $X \in \mathcal{Gal}(3)$ and of its Lie algebra $U \in \mathfrak{gal}(3)$ are given by

$$X = \begin{bmatrix} A & a & b \\ \mathbf{0}_{1 \times 3} & 1 & c \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5 \times 5}, \quad U = \begin{bmatrix} \Omega & v & r \\ \mathbf{0}_{1 \times 3} & 0 & \tau \\ \mathbf{0}_{1 \times 3} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{5 \times 5}, \quad (2)$$

where $A \in \mathbf{SO}(3)$, $\Omega \in \mathfrak{so}(3)$, $a, b, v, r \in \mathbb{R}^3$, and $c, \tau \in \mathbb{R}$.

The Galilean group identity element is $I \equiv \mathbf{I}_5$, and the inverse is $X^{-1} = (A^{-1}, -A^{-1}a, -A^{-1}(b - ca), -c)$. The *wedge operator* $(\cdot)^\wedge : \mathbb{R}^n \rightarrow \mathfrak{g}$ maps a vector $\mathbf{x} \in \mathbb{R}^n$ to the Lie algebra element $\mathbf{x}^\wedge \in \mathfrak{g}$, with inverse $(\cdot)^\vee : \mathfrak{g} \rightarrow \mathbb{R}^n$, where $n = \dim(\mathfrak{g})$. The action of Galilean symmetry on a Galilean frame $\mathbf{F} = (\mathbf{R}, \mathbf{v}, \mathbf{p}, t)$ is given by the matrix multiplication

$$X\mathbf{F} = \begin{bmatrix} A\mathbf{R} & A\mathbf{v} + a & A\mathbf{p} + a t + b \\ \mathbf{0}_{1 \times 3} & 1 & t + c \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix}. \quad (3)$$

B. Galilean Symmetry for Modeling INS with Delay

Accurate modeling of inertial motion requires careful choice of reference frames. In many robotics applications, the natural

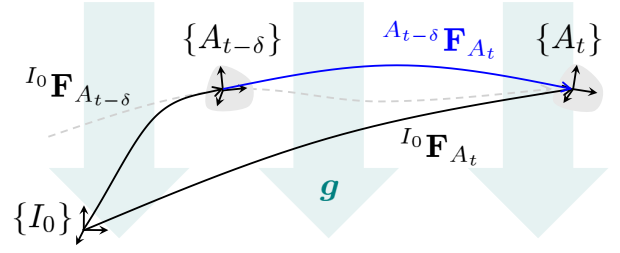


Fig. 2. Galilean transformations for a rigid body A in a gravity field. $\{I_0\}$ is an inertial frame of reference, defined as an arbitrary free-falling Galilean frame at zero time. The transformations ${}^{I_0}\mathbf{F}_{A_{t-\delta}}$ and ${}^{I_0}\mathbf{F}_{A_t}$ map *spatial* and *temporal* coordinates from the body frame at times $t - \delta$ and t to the inertial frame. The blue arrow ${}^{A_{t-\delta}}\mathbf{F}_{A_t}$ illustrates the *intra-body* transformation in the temporal interval δ . Intra-body Galilean frames are independent of the choice of inertial frame $\{I_0\}$, and are a key concept for the geometric spatial-temporal estimation framework presented in this paper.

reference frame is generally fixed to the Earth's surface, a non-inertial frame due to gravity and the Earth's rotation. While the Earth's rotation can often be neglected in many robotics problems, gravity introduces a continuous acceleration $\mathbf{g} \in \mathbb{R}^3$ that must be modeled. In this context, Galilean symmetry provides a principled and rigorous way to understand the effects of gravity in modeling relative non-inertial motion [20].

Consider an inertial frame at zero time $\{I_0\} \equiv \mathbf{I}_5$. Assume a positive time delay $\delta \in \mathbb{R}^+$, and let $\{A_t\}$ and $\{A_{t-\delta}\}$ denote the coordinate frames attached to a rigid body A at times t and $t - \delta$ respectively, as shown in Fig. 2. The coordinates of $\{A_t\}$ with respect to $\{I_0\}$ are described by the Galilean frame

$${}^{I_0}\mathbf{F}_{A_t} = \begin{bmatrix} {}^{I_0}\mathbf{R}_{A_t} & {}^{I_0}\mathbf{v}_{A_t} & {}^{I_0}\mathbf{p}_{A_t} \\ \mathbf{0}_{1 \times 3} & 1 & {}^{I_0}t_{A_t} \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix}, \quad (4)$$

where ${}^{I_0}\mathbf{R}_{A_t} \in \mathcal{SO}(3)$ encodes the orientation, ${}^{I_0}\mathbf{v}_{A_t} \in \mathbb{R}^3$ the inertial velocity, ${}^{I_0}\mathbf{p}_{A_t} \in \mathbb{R}^3$ the position, and ${}^{I_0}t_{A_t} = t$ the time offset. The same structure applies to the delayed time $t - \delta$ by replacing the time index accordingly.

We distinguish *intra-body* transformations, defined for the same rigid-body frame at different times, from *cross-body* transformations, describing the spatio-temporal relation between inertial frames attached to different rigid bodies. Define the *intra-body* Galilean frame ${}^{A_{t-\delta}}\mathbf{F}_{A_t} = {}^{I_0}\mathbf{F}_{A_{t-\delta}}^{-1} {}^{I_0}\mathbf{F}_{A_t}$ as

$${}^{A_{t-\delta}}\mathbf{F}_{A_t} = \begin{bmatrix} {}^{A_{t-\delta}}\mathbf{R}_{A_t} & {}^{A_{t-\delta}}\mathbf{v}_{A_t} & {}^{A_{t-\delta}}\mathbf{p}_{A_t} \\ \mathbf{0}_{1 \times 3} & 1 & \delta \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix}, \quad (5)$$

encoding the relative orientation, velocity, position, and time of $\{A\}$ over the time interval δ . Note that, by construction, the temporal offset ${}^{A_{t-\delta}}t_{A_t} = t - (t - \delta) = \delta$ is the *time delay*.

Consider a second rigid body B , as shown in Fig. 3. The intra-body transformation ${}^{B_{t-\delta}}\mathbf{F}_{B_t} = {}^{I_0}\mathbf{F}_{B_{t-\delta}}^{-1} {}^{I_0}\mathbf{F}_{B_t}$ is defined as in (5) by replacing the frame indices accordingly. Define the *cross-body* Galilean frame ${}^{A_{t-\delta}}\mathbf{F}_{B_t} = {}^{I_0}\mathbf{F}_{A_{t-\delta}}^{-1} {}^{I_0}\mathbf{F}_{B_t}$ as

$${}^{A_{t-\delta}}\mathbf{F}_{B_t} = \begin{bmatrix} {}^{A_{t-\delta}}\mathbf{R}_{B_t} & {}^{A_{t-\delta}}\mathbf{v}_{B_t} & {}^{A_{t-\delta}}\mathbf{p}_{B_t} \\ \mathbf{0}_{1 \times 3} & 1 & \delta \\ \mathbf{0}_{1 \times 3} & 0 & 1 \end{bmatrix}, \quad (6)$$

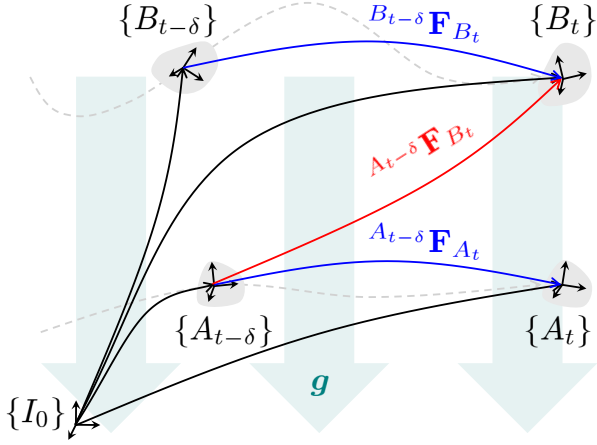


Fig. 3. Galilean transformations between two rigid bodies A and B in a gravity field. This setup extends the single-body case of Fig. 2 to account for the relative motion between two different rigid bodies. The blue arrows represent the *intra-body* transformations $A_{t-\delta}\mathbf{F}_{A_t}$ and $B_{t-\delta}\mathbf{F}_{B_t}$ over the temporal interval δ . The red arrow illustrates the *cross-body* transformation $A_{t-\delta}\mathbf{F}_{B_t}$, mapping coordinates from body B at current time t to body A at an earlier time $t - \delta$. The key idea is to filter for the cross-body Galilean frame, enabling joint estimation of spatial navigation states and time delay δ .

with $A_{t-\delta}t_{B_t} = t - (t - \delta) = \delta$. Leveraging Galilean symmetry ensures that the dynamics of each body are interpreted consistently with respect to an inertial reference frame, and provides a principled framework for representing coupled spatial and temporal transformations. Designing a filter for cross-body Galilean frames enables joint estimation of spatial and temporal states in multi-body INS with delays.

C. Kinematics of Galilean Frames

We consider the continuous-time kinematics of Galilean frames under inertial excitation. Given IMU measurements of angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and linear acceleration $\mathbf{a} \in \mathbb{R}^3$, define the input matrix $\mathbf{U}(\boldsymbol{\omega}, \mathbf{a}) \in \mathbb{R}^{5 \times 5}$ and a constant $\mathbf{N} \in \mathbb{R}^{5 \times 5}$ as

$$\mathbf{U}(\boldsymbol{\omega}, \mathbf{a}) := \begin{bmatrix} \boldsymbol{\omega}^\wedge & \mathbf{a} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 1} \end{bmatrix}, \quad \mathbf{N} := \begin{bmatrix} \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 4} & 1 \\ \mathbf{0}_{1 \times 4} & 0 \end{bmatrix}. \quad (7)$$

Let $\mathbf{U}_{A_t} = \mathbf{U}(\boldsymbol{\omega}_{A_t}, \mathbf{a}_{A_t})$ and $\mathbf{U}_{A_{t-\delta}} = \mathbf{U}(\boldsymbol{\omega}_{A_{t-\delta}}, \mathbf{a}_{A_{t-\delta}})$ be the inertial input matrices for rigid body A at time t and $t - \delta$. The corresponding Galilean frames evolve according to

$$I_0 \dot{\mathbf{F}}_{A_{t-\delta}} = I_0 \mathbf{F}_{A_{t-\delta}} (\mathbf{U}_{A_{t-\delta}} + \mathbf{N}), \quad (8)$$

$$I_0 \dot{\mathbf{F}}_{A_t} = I_0 \mathbf{F}_{A_t} (\mathbf{U}_{A_t} + \mathbf{N}). \quad (9)$$

Note that the role of $\mathbf{N}$ is to model the *time* evolution $\dot{t} = 1$. The kinematics of the intra-body frame $A_{t-\delta}\mathbf{F}_{A_t}$ follow as

$$A_{t-\delta} \dot{\mathbf{F}}_{A_t} = \frac{d}{dt} \left(I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \mathbf{F}_{A_t} \right) \quad (10)$$

$$= -I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \dot{\mathbf{F}}_{A_{t-\delta}} I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \mathbf{F}_{A_t} + I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \dot{\mathbf{F}}_{A_t} \quad (11)$$

$$= -(\mathbf{U}_{A_{t-\delta}} + \mathbf{N})^{A_{t-\delta}} \mathbf{F}_{A_t} + A_{t-\delta} \mathbf{F}_{A_t} (\mathbf{U}_{A_t} + \mathbf{N}), \quad (12)$$

and the same goes for $B_{t-\delta}\mathbf{F}_{B_t}$ by replacing frame indices. An analogous computation yields the kinematics of the cross-body frame $A_{t-\delta}\mathbf{F}_{B_t}$, even though its kinematics depend on

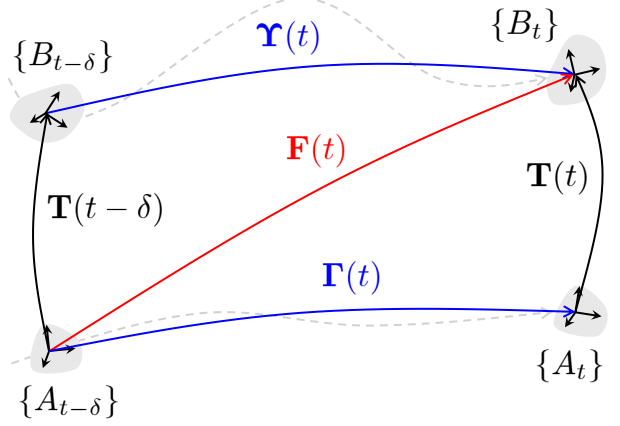


Fig. 4. Representation of the time-varying Galilean frames between bodies A and B illustrated in Fig. 3. The complex frame-dependent notation is replaced with compact notation: $\Gamma(t)$ and $\Upsilon(t)$ denote the *intra-body* transformations (blue) for bodies A and B respectively, while $\mathbf{F}(t)$ denotes the *cross-body* transformation (red) between the two rigid bodies. $\mathbf{T}(t)$ and $\mathbf{T}(t - \delta)$ denote *isochronous* Galilean frames, equivalent to extended poses, which encode the spatial transformation between inertial frames at the same time instant.

inputs at two different times, $t - \delta$ and t :

$$A_{t-\delta} \dot{\mathbf{F}}_{B_t} = \frac{d}{dt} \left(I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \mathbf{F}_{B_t} \right) \quad (13)$$

$$= -I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \dot{\mathbf{F}}_{A_{t-\delta}} I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \mathbf{F}_{B_t} + I_0 \mathbf{F}_{A_{t-\delta}}^{-1} I_0 \dot{\mathbf{F}}_{B_t} \quad (14)$$

$$= -(\mathbf{U}_{A_{t-\delta}} + \mathbf{N})^{A_{t-\delta}} \mathbf{F}_{B_t} + A_{t-\delta} \mathbf{F}_{B_t} (\mathbf{U}_{B_t} + \mathbf{N}), \quad (15)$$

where $\mathbf{U}_{B_t} = \mathbf{U}(\boldsymbol{\omega}_{B_t}, \mathbf{a}_{B_t})$.

D. Application to Spatio-Temporal State Estimation

We provide insights on how to use the proposed framework to derive a simple state observer for the relative motion and time delay of two rigid bodies with delayed relative pose measurements. The notation used is illustrated in Fig. 4.

Define the system's state $\mathbf{F}(t) \in \mathcal{Gal}(3)$ as the cross-body Galilean frame (6), with kinematics given by (15). The state observer is $\hat{\mathbf{F}} = (\hat{R}, \hat{v}, \hat{p}, \hat{\delta}) \in \mathcal{Gal}(3)$, with kinematics

$$\dot{\hat{\mathbf{F}}} = -\mathbf{U}_{A_{t-\delta}} \hat{\mathbf{F}} + \hat{\mathbf{F}} \mathbf{U}_{B_t} + \hat{\mathbf{F}} \mathbf{N} - \mathbf{N} \hat{\mathbf{F}} + \Delta \hat{\mathbf{F}}, \quad (16)$$

where $\Delta \in \mathfrak{gal}(3)$ is an innovation term driving $\hat{\mathbf{F}}(t) \rightarrow \mathbf{F}(t)$. The time-varying inertial inputs $\mathbf{U}_{A_{t-\delta}}, \mathbf{U}_{B_t} \in \mathfrak{gal}(3)$ and the constant $\mathbf{N} \in \mathfrak{gal}(3)$ are defined according to (7) and represented as elements of the Lie algebra of $\mathcal{Gal}(3)$. Note that the observer kinematics (16) is group-affine and preserves the Lie group structure of the state space. The extended poses between the two rigid bodies $\mathbf{T}(t - \delta), \mathbf{T}(t) \in \mathcal{Gal}(3)$ at times $t - \delta$ and t are represented as *isochronous* Galilean frames, i.e., their temporal component is zero, and can be modeled as

$$\mathbf{T}(t - \delta) = \mathbf{F}(t) \Upsilon(t)^{-1}, \quad \mathbf{T}(t) = \Gamma(t)^{-1} \mathbf{F}(t). \quad (17)$$

Note that modeling a delayed pose measurement or retrieving the current relative pose from the state requires the two intra-body transformations $\Gamma(t), \Upsilon(t) \in \mathcal{Gal}(3)$. These Galilean frames depend on the time delay δ and buffered IMU measurements, and are termed *preintegration matrices*. The idea

of using Galilean symmetry to provide a compact formulation of IMU preintegration has been recently introduced by Delama et al. [1]. These terms are treated as known functions of the time delay, denoted as $\Gamma(\delta), \Upsilon(\delta) \in \text{Gal}(3)$. For a fixed delay δ , the preintegration matrices can be recursively computed from buffered IMU data. *Without loss of generality, the definition is given for $\Upsilon(\delta)$, with $\Gamma(\delta)$ defined analogously by replacing the frame indices with A .* Given buffered IMU measurements $\{U_{B_{t-\delta}}, \dots, U_{B_t}\}$, the preintegration matrix is

$$\Upsilon(\delta) = \prod_{k=(t-\delta)}^t \exp((U_{B_k} + N)dt), \quad (18)$$

where $dt \in \mathbb{R}$ is the IMU time step. Recomputing the preintegration matrix from scratch each time the time delay changes during estimation is possible, but highly inefficient. Alternatively, it is possible to compute it once and propagate it as long as the delay converges to a steady value. The discrete-time kinematics of Υ is obtained by discretizing (12) as

$$\Upsilon_{i+1} = -\exp((U_{B_{t-\delta}} + N)dt)\Upsilon_i \exp((U_{B_t} + N)dt), \quad (19)$$

where the subscript i denotes the index of the observer iterative loop. However, this propagation introduces drift due to noise in the IMU measurements, and if the delay changes, the preintegration matrix must be recomputed. In practice, a more robust approach is to keep a *sliding buffer* of preintegrated IMU matrices over a fixed time window, e.g., 0.5 s. At each iteration i , a new identity element is appended, and the oldest entry is removed if the buffer exceeds its maximum length. Then, all entries in the buffer are propagated in batch as

$$\Upsilon_k \leftarrow \Upsilon_k \exp((U_{B_t} + N)dt), \quad (20)$$

where the subscript k identifies the preintegration matrix at the k -th position in the sliding window. The function $\Upsilon(\delta)$ then retrieves the preintegration matrix corresponding to the requested delay, using interpolation if needed. This strategy allows efficient handling of variable delays without recomputing preintegration or introducing drift, increasing memory usage, but keeping the computational cost constant. Using the preintegration matrices, the observer compares the *measured* and *predicted* relative poses to construct a residual

$$\mathbf{r} = f(\mathbf{T}_m(t - \delta), \hat{\mathbf{F}}) \in \mathbb{R}^m, \quad (21)$$

which, for an extended pose measurement, can take the form

$$\mathbf{r} = \log(\mathbf{T}_m(t - \delta)\Upsilon(\delta)\hat{\mathbf{F}}^{-1})^\vee, \quad (22)$$

where $\log : \text{Gal}(3) \rightarrow \mathfrak{gal}(3)$ maps the group error to the Lie algebra. The residual $\mathbf{r}$ is used to drive the innovation Δ , providing a feedback correction that updates the observer toward the true state. This correction is computed in the Lie algebra using a gain matrix as $\Delta = (\mathbf{K} \mathbf{r})^\wedge$. Finally, the relative extended pose estimate at the current time is retrieved:

$$\hat{\mathbf{T}}(t) = \Gamma(\hat{\delta})^{-1} \hat{\mathbf{F}}. \quad (23)$$

III. EQUIVARIANT FILTER FOR DELAYED GNSS-INS

This section derives a *discrete-time* EqF for biased-INS with delayed GNSS measurements using the geometric framework introduced in Sec. II. *The filter is designed to jointly estimate the system's position, velocity, and attitude, as well as the unknown GNSS time delay and IMU biases.* The system is formalized in Sec. III-A and the linearized error dynamics for the propagation are derived in Sec. III-B. Finally, the update for delayed GNSS position measurements is formulated in Sec. III-C. A practical implementation is outlined in Alg. 1.

A. Biased-INS with Delayed Position Measurements

Consider a mobile robot equipped with an IMU that delivers *biased* angular velocity and linear acceleration measurements, and a GNSS sensor providing delayed 3D position measurements affected by an *unknown* time delay $\delta \in \mathbb{R}^+$. The system is modeled using the geometric framework in Fig. 4. In this scenario, the coordinate frame $\{A\}$ is fixed to the Earth rather than attached to a second rigid body, and is subject to a constant input representing the Earth's angular velocity and a linear acceleration counteracting gravity. Consider the system

$$\dot{\xi} = f(\xi, u) \iff \begin{cases} \dot{\mathbf{F}} = -\mathbf{g}_N^\wedge \mathbf{F} + \mathbf{F}(\mathbf{w}_N - \mathbf{b})^\wedge \\ \dot{\mathbf{b}} = \boldsymbol{\tau} \end{cases}, \quad (24)$$

where the state is denoted as $\xi = (\mathbf{F}, \mathbf{b}) \in \mathcal{M} := \text{Gal}(3) \times \mathbb{R}^{10}$ and the input is represented as $u = (\mathbf{w}_N, \boldsymbol{\tau}) \in \mathbb{L} \subset \mathbb{R}^{20}$. In particular, $\mathbf{F} = (\mathbf{R}, \mathbf{v}, \mathbf{p}, \delta) \in \text{Gal}(3)$ denotes the *core states*, including spatial coordinates and the temporal offset δ , and $\mathbf{b} = (\mathbf{b}_\omega, \mathbf{b}_a, \mathbf{b}_v, b_\rho) \in \mathbb{R}^{10}$ denotes the *bias states*, including the IMU biases $\mathbf{b}_\omega, \mathbf{b}_a \in \mathbb{R}^3$ and two *virtual* biases $\mathbf{b}_v \in \mathbb{R}^3$ and $b_\rho \in \mathbb{R}$ introduced by Delama et al. [1] to model the underlying symmetry. $\boldsymbol{\tau} = (\boldsymbol{\tau}_\omega, \boldsymbol{\tau}_a, \boldsymbol{\tau}_v, \tau_\rho) \in \mathbb{R}^{10}$ models the bias random-walk. Finally, $\mathbf{g}_N = (\boldsymbol{\omega}_E, -\mathbf{g}, \mathbf{0}_{3 \times 1}, 1) \in \mathbb{R}^{10}$ is a *constant* vector capturing the effect of the Earth's rotation $\boldsymbol{\omega}_E \in \mathbb{R}^3$ and gravity, and $\mathbf{w}_N = (\boldsymbol{\omega}, \mathbf{a}, \mathbf{0}_{3 \times 1}, 1) \in \mathbb{R}^{10}$ is the time-varying extended *biased inertial input* at current time. The wedge operator $(\cdot)^\wedge : \mathbb{R}^{10} \rightarrow \mathfrak{gal}(3) \subset \mathbb{R}^{5 \times 5}$ is used to map vectors to matrices. Note that the terms $\mathbf{g}_N^\wedge = \mathbf{U}(\boldsymbol{\omega}_E, -\mathbf{g}) + \mathbf{N}$ and $\mathbf{w}_N^\wedge = \mathbf{U}(\boldsymbol{\omega}, \mathbf{a}) + \mathbf{N}$ derive from (7), and the system kinematics depend only on the input at the *current IMU time*.

By assuming a constant input $u_i \in \mathbb{L}$ between consecutive timestamps t_i and t_{i+1} , and defining the IMU time step $dt = t_{i+1} - t_i$, the exact discretization of (24) yields

$$\begin{cases} \mathbf{F}_{i+1} = \exp(-\mathbf{g}_N^\wedge dt) \mathbf{F}_i \exp((\mathbf{w}_{N_i} - \mathbf{b}_i)^\wedge dt) \\ \mathbf{b}_{i+1} = \mathbf{b}_i + \boldsymbol{\tau}_i dt \end{cases}, \quad (25)$$

where the subscript i denotes the index of the iterative loop.

B. Lifted System and Error Dynamics

The EqF is derived from a *lifted system*, in which the original system is represented via a Lie group symmetry with an equivariant structure. This formulation defines the state dynamics in a structured way and leverages the *equivariant error* with its favorable linearization properties (van Goor et al. [25]). Define the symmetry group $\mathbf{G} := \text{Gal}(3) \ltimes \mathfrak{gal}(3)$ called

left-trivialized tangent group of the Galilean group, proposed by Delama et al. [1]. This group is the semi-direct product between $\mathbf{Gal}(3)$ and its Lie algebra. For $X = (A, a) \in \mathbf{G}$ and $Y = (B, b) \in \mathbf{G}$, the group operation and the inverse are

$$XY = (AB, a + \text{Ad}_A[b]), \quad X^{-1} = (A^{-1}, -\text{Ad}_{A^{-1}}[a]). \quad (26)$$

The lifted system state is $X = (X_F, X_b) \in \mathbf{G}$, with kinematics

$$X_{i+1} = X_i \Lambda(\phi(X_i, \xi), u_i), \quad (27)$$

where $\xi \in \mathcal{M}$ is an arbitrarily chosen *state origin*. The mapping $\phi: \mathbf{G} \times \mathcal{M} \rightarrow \mathcal{M}$ defines the *state action* and is given by

$$\phi(X, \xi) := (\mathbf{F} X_F, \text{Ad}_{X_F^{-1}}^\vee(\mathbf{b} - X_b^\vee)). \quad (28)$$

The mapping $\Lambda: \mathcal{M} \times \mathbb{L} \rightarrow \mathbf{G}$ is the *discrete lift*, defined as

$$\Lambda(\xi, u) := (\Lambda_F(\xi, u), \Lambda_b(\xi, u)), \quad (29)$$

where $\Lambda_F \in \mathbf{Gal}(3)$ and $\Lambda_b \in \mathbf{gal}(3)$ take the form

$$\Lambda_F(\xi, u) = \exp(-\text{Ad}_{\mathbf{F}^{-1}}[\mathbf{g}_N^\wedge] \text{dt}) \exp((\mathbf{w}_N - \mathbf{b})^\wedge \text{dt}), \quad (30)$$

$$\Lambda_b(\xi, u) = \mathbf{b}^\wedge - \text{Ad}_{\Lambda_F(\xi, u)}[\mathbf{b}^\wedge + \boldsymbol{\tau}^\wedge \text{dt}]. \quad (31)$$

The EqF is formulated around the *equivariant error*, which is intrinsically defined on the manifold $\mathcal{M}$ and evolves according to coordinate-free dynamics. This global error is linearized about the state origin ξ to derive the filter propagation and update laws that drive filter convergence. It is defined as

$$e := \phi(\hat{X}^{-1}, \xi) \in \mathcal{M}, \quad (32)$$

where $\hat{X} = (\hat{X}_F, \hat{X}_b) \in \mathbf{G}$ denotes the *state estimate* on the symmetry group, and $\xi = (\mathbf{F}, \mathbf{b}) \in \mathcal{M}$ represents the *true state* on the manifold. Define a local parametrization $\vartheta: \mathcal{M} \rightarrow \mathbb{R}^{20}$ for the error around ξ using *normal coordinates* as

$$\varepsilon = \vartheta(e) = \log(\phi_\xi^{-1}(e))^\vee = \log(\phi_\xi^{-1}(\phi(\hat{X}^{-1}, \xi)))^\vee, \quad (33)$$

where $\log(\cdot): \mathbf{G} \rightarrow \mathfrak{g}$ denotes the logarithm of $\mathbf{G}$.

By assuming a *noisy* input $\tilde{u} = u + \eta \in \mathbb{L}$ and fixing $\xi = I$, we derive the linearized error dynamics $\varepsilon_{i+1} \approx \mathbf{A} \varepsilon_i + \mathbf{B} \eta_i$ about $\varepsilon = \mathbf{0}$ and $\eta = \mathbf{0}$. The discrete-time *state* matrix $\mathbf{A} \in \mathbb{R}^{20 \times 20}$ is

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_1 \mathbf{J}_L(\dot{\mathbf{w}}_N \text{dt}) \\ \mathbf{0}_{10 \times 10} & \mathbf{A}_2 \end{bmatrix}, \quad (34)$$

where $\dot{\mathbf{w}}_N = \text{Ad}_{\hat{X}_F}^\vee \mathbf{w}_N + \hat{X}_b^\vee$, and $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{10 \times 10}$ are

$$\mathbf{A}_1 = \text{Ad}_{\exp(\mathbf{g}_N^\wedge \text{dt})}^\vee, \quad \mathbf{A}_2 = \mathbf{A}_1 \text{Ad}_{\exp(\dot{\mathbf{w}}_N^\wedge \text{dt})}^\vee. \quad (35)$$

The discrete-time *input noise* matrix $\mathbf{B} \in \mathbb{R}^{20 \times 20}$ is

$$\mathbf{B} = \begin{bmatrix} -\mathbf{A}_1 \mathbf{J}_L(\dot{\mathbf{w}}_N \text{dt}) \text{Ad}_{\hat{X}_F}^\vee \text{dt} & \mathbf{0}_{10 \times 10} \\ \mathbf{0}_{10 \times 10} & \mathbf{A}_2 \text{Ad}_{\hat{X}_F}^\vee \text{dt} \end{bmatrix}. \quad (36)$$

Algorithm 1 Equivariant Filter with Online Delay Estimation

Parameters: $\xi = (\mathbf{I}_5, \mathbf{0}_{10 \times 1})$, ξ_{init} , Σ_{init} , $\mathbf{Q}$, $\mathbf{R}$, p_0

Initialization: $\hat{X} = \phi_\xi^{-1}(\xi_{init})$, $\hat{\Sigma} = \Sigma_{init}$

Measurements: IMU $\rightarrow (\boldsymbol{\omega}, \mathbf{a})$, GNSS $\rightarrow \mathbf{p}(t - \delta)$

Output: $\hat{X}$, $\hat{\Sigma}$, $\hat{\mathbf{T}}(t) = \Gamma(\hat{\delta})^{-1} \hat{X}_F$

PROPAGATION: require an input meas. $u = (\mathbf{w}_N, \boldsymbol{\tau})$

$$1: \hat{\Sigma} \leftarrow \mathbf{A} \hat{\Sigma} \mathbf{A}^\top + \frac{1}{\text{dt}_k} \mathbf{B} \mathbf{Q} \mathbf{B}^\top \quad (34) \quad (36)$$

$$2: \hat{X} \leftarrow \hat{X} \Lambda(\phi(\hat{X}, \xi), \tilde{u}) \quad (27)$$

$$3: \Upsilon_k \leftarrow \Upsilon_k \exp((\mathbf{w}_N - \hat{\mathbf{b}})^\wedge \text{dt}) \quad \forall k \quad (20)$$

UPDATE: require an output meas. $y_m = \mathbf{p}(t - \delta)$

$$1: \mathbf{r} \leftarrow y_m - h(\phi(\hat{X}, \xi)) \quad (39)$$

$$2: \mathbf{K} \leftarrow \hat{\Sigma} \mathbf{C}^\top (\mathbf{C} \hat{\Sigma} \mathbf{C}^\top + \mathbf{D} \mathbf{R} \mathbf{D}^\top)^{-1} \quad (40)$$

$$3: \Delta \leftarrow \text{d}\phi_\xi^\dagger \circ \text{d}\vartheta^{-1} \circ \mathbf{K} \mathbf{r} \quad *$$

$$4: \hat{\Sigma} \leftarrow (\mathbf{I}_{20} - \mathbf{K} \mathbf{C}) \hat{\Sigma} \quad **$$

$$5: \hat{\Sigma} \leftarrow \mathbf{J}_L(\Delta^\vee) \hat{\Sigma} \mathbf{J}_L(\Delta^\vee)^\top \quad **$$

$$6: \hat{X} \leftarrow \exp(\Delta) \hat{X}$$

*Note that $\text{d}\phi_\xi^\dagger \circ \text{d}\vartheta^{-1}[\cdot] = (\cdot)^\wedge$. **This is the *reset step* (Ge et al. [10]).

C. Delayed Position Measurement Update

A delayed position measurement $y_m \in \mathbb{R}^3$ is modeled as

$$\begin{pmatrix} y_m \\ 0 \\ 1 \end{pmatrix} = \mathbf{T}(t - \delta) \bar{\mathbf{y}}_0 + \bar{\boldsymbol{\mu}} = \mathbf{F}(t) \Upsilon(t)^{-1} \bar{\mathbf{y}}_0 + \bar{\boldsymbol{\mu}}, \quad (37)$$

with $\bar{\mathbf{y}}_0 = (\mathbf{p}_0, 0, 1) \in \mathbb{R}^5$ and $\bar{\boldsymbol{\mu}} = (\boldsymbol{\mu}, \mathbf{0}_{2 \times 1}) \in \mathbb{R}^5$, where $\mathbf{p}_0 \in \mathbb{R}^3$ is the GNSS receiver position in the IMU frame and the noise term $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \Sigma_\mu)$ denotes the GNSS-position uncertainty. Define the measurement function $h: \mathcal{M} \rightarrow \mathbb{R}^3$ as

$$h(\xi) = \mathcal{P}(\mathbf{F} \Upsilon(\delta)^{-1} \bar{\mathbf{y}}_0), \quad (38)$$

where $\mathcal{P}(\mathbf{x}) = \mathbf{x}_{1:3}$ extracts the first three elements of a vector, and $\Upsilon(\delta) \in \mathbf{Gal}(3)$ is the IMU *preintegration matrix* detailed in Sec. II-D, with the difference that here the buffered IMU data must be corrected using the corresponding bias estimate.

Upon receiving a GNSS position y_m , a residual $\mathbf{r} \in \mathbb{R}^3$ is defined by comparing the measurement with its prediction as

$$\mathbf{r} = y_m - h(\phi(\hat{X}, \xi)). \quad (39)$$

Linearizing $\mathbf{r} \approx \mathbf{C} \varepsilon + \mathbf{D} \boldsymbol{\mu}$ yields $\mathbf{D} = \mathbf{I}_3$ and $\mathbf{C} \in \mathbb{R}^{3 \times 20}$ as

$$\mathbf{C} = [\mathbf{C}_1 \quad \mathbf{0}_{3 \times 3} \quad \mathbf{I}_3 \quad \mathbf{C}_2 \quad \mathbf{0}_{3 \times 10}], \quad (40)$$

with $\mathbf{C}_1 \in \mathbb{R}^{3 \times 3}$ and $\mathbf{C}_2 \in \mathbb{R}^{3 \times 1}$ defined as

$$\mathbf{C}_1 = -(\hat{R} p_{\Upsilon^{-1}} - \hat{v} \hat{\delta} + \hat{p})^\wedge, \quad (41)$$

$$\mathbf{C}_2 = \hat{v} - \hat{R}(R_{\Upsilon^{-1}} \boldsymbol{\omega}_{t-\hat{\delta}}^\wedge \mathbf{p}_0 + v_{\Upsilon^{-1}}), \quad (42)$$

where $\hat{X}_F = (\hat{R}, \hat{v}, \hat{p}, \hat{\delta}) \in \mathbf{Gal}(3)$ is the current state estimate, $\Upsilon(\hat{\delta})^{-1} = (R_{\Upsilon^{-1}}, v_{\Upsilon^{-1}}, p_{\Upsilon^{-1}}, -\hat{\delta}) \in \mathbf{Gal}(3)$ is the inverse of the preintegration matrix (18), and $\boldsymbol{\omega}_{t-\hat{\delta}} \in \mathbb{R}^3$ is the IMU angular velocity at time $t - \hat{\delta}$, retrieved from the buffer. The discrete-time EqF is complete and summarized in Alg. 1.

TABLE I
SUMMARY OF REAL-WORLD EXPERIMENTS WITH RMSES FOR EKF AND EQF UNDER DIFFERENT TIME-DELAY SETTINGS.

Experiment Metadata					Rotation RMSE (deg)				Velocity RMSE (m/s)				Position RMSE (m)			
Dataset Sequence	Length (km)	$v_{\max}$ (m/s)	$a_{\max}$ (m/s ²)	δ (ms)	EKF (w/o delay)	EKF (offline)	EKF (online)	EqF (online)	EKF (w/o delay)	EKF (offline)	EKF (online)	EqF (online)	EKF (w/o delay)	EKF (offline)	EKF (online)	EqF (online)
striver	2.10	34.82	51.31	120	5.54	3.49	<u>3.55</u>	3.62	0.75	0.54	0.44	<u>0.45</u>	2.58	0.46	<u>0.58</u>	0.63
hero-1	4.15	38.15	18.94	90	3.22	<u>2.32</u>	2.39	2.31	0.67	0.76	0.73	0.27	2.62	<u>0.64</u>	0.67	0.50
hero-2	4.18	35.10	25.93	90	3.35	2.62	<u>2.76</u>	2.84	0.76	0.46	<u>0.41</u>	0.38	2.51	0.61	0.66	<u>0.62</u>

$v_{\max}$: max speed. $a_{\max}$: max acceleration. δ : delay reference estimated offline. *w/o delay*: delay fixed to 0 ms. *offline*: delay fixed to δ . *online*: delay estimated online. **Best results are in bold and second-best results are underlined.**

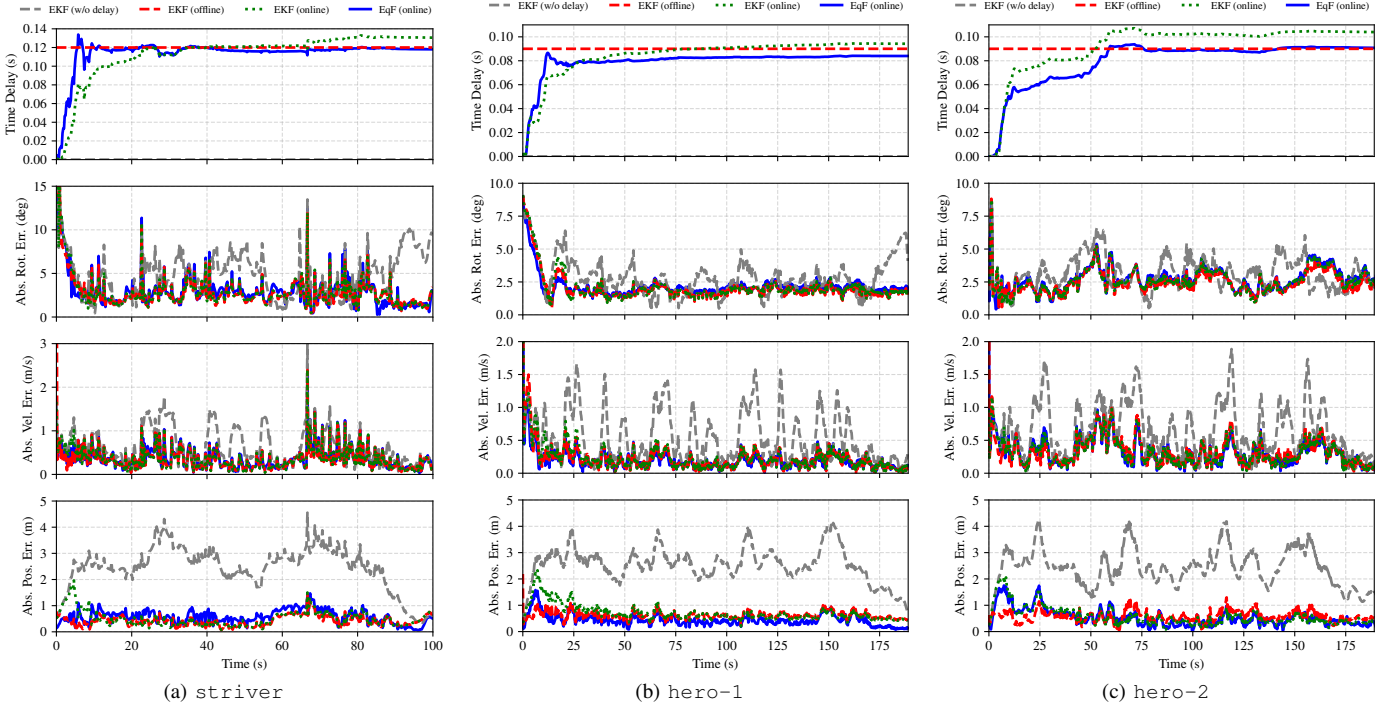


Fig. 5. Real-world comparison of the proposed EqF (solid blue) against three EKF baselines: a standard EKF without delay compensation (dashed grey), an EKF with offline delay estimate (dashed red), and an EKF with online time-delay estimation (dotted green). The top row illustrates the *online time-delay estimation*: (a) *striver* shows rapid, near-perfect convergence of our EqF to the 120ms delay reference, while EKF (online) shows a slower transient and a worse asymptotic; (b) *hero-1* shows a small ~8ms residual bias of the EqF, suggesting the EqF estimate may be more accurate than the nominal 90ms delay, consistent with its superior performance in all other metrics in this specific dataset sequence; (c) *hero-2* shows a slower transient of both EqF and EKF (online), but ultimately the EqF converges to the 90ms delay reference. Overall, the proposed EqF consistently outperforms the EKF without delay compensation across all sequences, and matches or exceeds the accuracy of the EKF with offline and online delay compensation. Experiments metadata and RMSE values are reported in Tab. I. For this magnitude of time delay, the real-world results are consistent with the simulation results illustrated in Fig. 6a.

IV. EXPERIMENTS AND RESULTS

The proposed EqF for GNSS-aided INS with delayed position measurements is validated using both real-world experiments and simulations, and compared against state-of-the-art EKF baselines. We evaluate the performance using standard metrics for estimation accuracy and filter consistency. The *Absolute Rotation Error* (ARE), *Absolute Position Error* (APE), and *Absolute Velocity Error* (AVE) measure the differences between estimated and ground-truth states in the global frame. These metrics are computed as the Euclidean norm of the corresponding error vectors, providing a scalar measure of the accuracy of rotation, position, and velocity estimation over time. In addition, we utilize the *Absolute Delay Error* (ADE) for time delays. For a trajectory with M time steps, let $(\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i)$ denote the ground truth rotation, position, and velocity at time t_i , and $(\hat{\mathbf{R}}_i, \hat{\mathbf{p}}_i, \hat{\mathbf{v}}_i)$ the corresponding filter estimates. Let δ_i and $\hat{\delta}_i$ be the ground truth and estimated time

delays. The errors are computed for each timestamp as

$$\text{ARE}_i = \|\log(\mathbf{R}_i^\top \hat{\mathbf{R}}_i)\| \in \mathbb{R}, \quad \text{AVE}_i = \|\mathbf{v}_i - \hat{\mathbf{v}}_i\| \in \mathbb{R}, \quad (43)$$

$$\text{APE}_i = \|\mathbf{p}_i - \hat{\mathbf{p}}_i\| \in \mathbb{R}, \quad \text{ADE}_i = |\delta_i - \hat{\delta}_i| \in \mathbb{R}. \quad (44)$$

To summarize these errors over the trajectory of length M , we report the *Root Mean Square Error* (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=0}^{M-1} E_i^2} \in \mathbb{R}, \quad (45)$$

where E_i denotes either ARE_i , AVE_i , or APE_i . To evaluate filter consistency, we compute the *Normalized Estimation Error Squared* (NEES), which measures the difference between estimation errors and the predicted covariance. The full-state NEES is calculated for each timestamp as

$$\text{NEES}_i = \frac{1}{N} \boldsymbol{\varepsilon}_i^\top \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\varepsilon}_i, \quad (46)$$

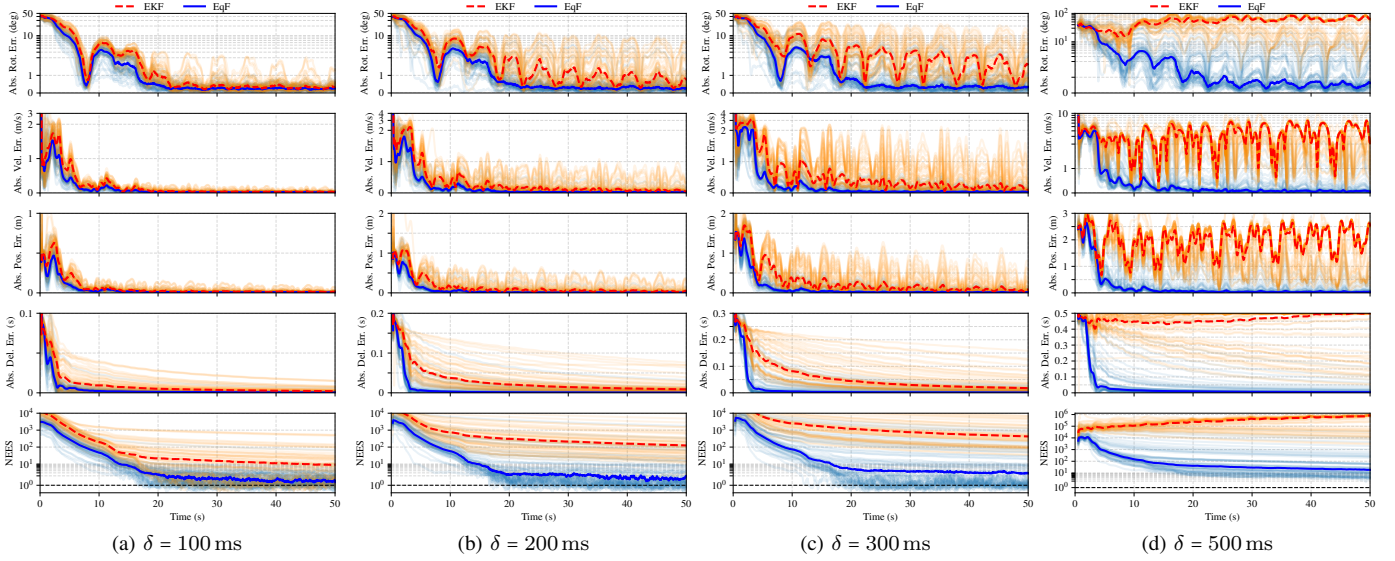


Fig. 6. Results of a Monte Carlo simulation with 50 different realizations of randomized initial states, process noise, and measurement noise, comparing EqF and EKF with *online time-delay estimation* for increasing measurement delays: (a) $\delta = 100$ ms, (b) $\delta = 200$ ms, (c) $\delta = 300$ ms, and (d) $\delta = 500$ ms. From top to bottom, the rows show the absolute rotation, velocity, position, and time-delay errors, followed by the NEES. Semi-transparent lines correspond to individual runs, while thick lines indicate the median across all runs. As the delay increases, the EKF exhibits growing oscillations, loss of consistency, and increased estimation error, ultimately diverging in the (d) case. In contrast, the EqF maintains stable, low-error, and consistent performance across all delays.

where $\varepsilon_i \in \mathbb{R}^N$ is the state estimation error in local coordinates at time t_i , and $\Sigma_i \in \mathbb{R}^{N \times N}$ is the corresponding covariance.

The trajectories used in both simulation and real-world experiments are designed to ensure sufficient excitation of the platform, including variations in linear acceleration and rotational motion. Such excitation is desired to ensure that the time delay is observable, in accordance with classical observability results for the biased-INS. In particular, degenerate cases such as the robot being stationary or moving at a constant velocity along a straight line make the delay unobservable.

A. Real-World UAV Flights

The experimental validation is conducted using data collected from two fixed-wing UAV platforms, *striver* and *hero*, as illustrated in Fig. 1. The dataset consists of three flights: one performed by the *striver* and two by the *hero*. In these scenarios, the GNSS measurements are subject to unknown ground-truth time delays. A delay reference for each experiment is determined via an offline batch optimization that minimizes the state residuals over the full dataset, while the ground-truth trajectories are obtained from an on-board calibrated EKF that fuses measurements from additional sensors, including a magnetometer, barometer, and GNSS velocity.

We evaluate the proposed EqF by comparing its performance against three EKF baselines: *EKF (w/o delay)*, i.e., a standard EKF without delay compensation, *EKF (offline)*, i.e., an EKF utilizing a fixed offline-estimated delay compensation, and *EKF (online)*, i.e., a state-of-the-art EKF with online delay estimation. The latter is derived following the same design approach as Kelly et al. [15], except that we use the extended pose formulation and the preintegration terms that are known from [1, 16]. The EKF state is expressed

as $\xi_{EKF} = (\mathbf{T}, \mathbf{b}, \delta) \in \mathcal{SE}_2(3) \times \mathbb{R}^6 \times \mathbb{R}$, where $\mathbf{T}$ denotes the extended pose at the current time t and the time delay δ is treated as an independent scalar. Since the EKF does not exploit the Galilean frame structure that geometrically couples spatial and temporal states, the propagation differs from that of the EqF: there is no coupling between spatial and temporal state errors, and each component evolves independently in a standard Euclidean framework. Moreover, the measurement function of the EKF is defined as $h(\xi_{EKF}) = \mathcal{P}(\Gamma(\delta)\mathbf{T}\Upsilon(\delta)^{-1}\bar{\mathbf{y}}_0)$, which differs from (38). It is worth noting that although we use preintegration terms Γ and Υ introduced in Sec. II-D to reduce prediction error in the EKF, the residual linearization differs from that of the EqF, leading to different output matrices in the filter update.

The results presented in Fig. 6 show that the EqF successfully tracks time delays online across all three flight sequences, and the corresponding trajectory errors demonstrate that incorporating the delay significantly reduces state errors compared to the standard EKF without delay compensation. Across the sequences, the EqF shows improved transient performance compared to the EKF with online delay estimation, and it achieves accuracy comparable to the EKF with offline-estimated delay compensation, while operating fully online. The quantitative results, including experiment metadata and RMSE values, are summarized in Tab. I. Note that the RMSE values are calculated over the entire trajectory, including the transient phase of the online delay estimation. This introduces a slight disadvantage for EqF and EKF (online), since for these filters the delay is estimated in real time rather than being pre-calibrated as in EKF (offline). Overall, the real-world experiments indicate that the proposed EqF can reliably operate fully online under unknown GNSS time delays.

B. Simulation Experiments

To complement the real-world experiments, we run simulations to provide a statistical comparison of the proposed EqF against the state-of-the-art EKF with online delay estimation, i.e., EKF (online). We ran Monte Carlo simulations using an analytically generated trajectory consisting of circular motion with superimposed horizontal and vertical waves and varying attitude to excite all rotational axes. Four delay scenarios were considered: 100 ms, 200 ms, 300 ms, and 500 ms. Synthetic IMU measurements were generated at 200 Hz, and GNSS position measurements at 20 Hz, with a fixed time delay introduced according to the scenario. For each scenario, 50 Monte Carlo realizations were performed by randomizing the initial states, process noise, and measurement noise. The simulation results, reported in Fig. 6, highlight a clear distinction between the EqF and the EKF when the time delay is estimated online. Even for small measurement delays, the EKF exhibits inconsistencies, reflected in elevated NEES values, whereas the EqF maintains consistent performance across all delays. This behavior of the EKF aligns with the findings of Kelly et al. [15] showing that incorporating online time-delay estimation in classical robotic state-estimation filters introduces fundamental structural issues, leading to bias and inconsistency. In fact, as the delay increases, the EKF exhibits growing oscillations, loss of consistency, and increased estimation errors, ultimately diverging in the 500 ms delay scenario. In contrast, the EqF built on the proposed geometric framework maintains accuracy and consistency, exhibiting full convergence in all tests.

V. CONCLUSION

This paper presented a geometric approach for modeling inertial motion with temporal delays based on Galilean symmetry. The resulting mathematical framework enables spatial-temporal state estimation and offers a principled representation of INS with measurement delays. Using this formulation, we derived an EqF that simultaneously estimates navigation states and unknown GNSS time lags. We validated the approach with real-world flights of two UAVs experiencing GNSS delays of 90 ms and 120 ms. The EqF accurately estimated both navigation states and time delays online, achieving RMSE performance comparable to an EKF with offline-estimated delay compensation, but without requiring tedious offline calibration. Simulations provided further statistical comparison of our EqF against a state-of-the-art EKF with online time-delay estimation. We investigated delays up to 500 ms and observed that the EqF maintains accurate and consistent estimates, while the EKF remains inconsistent and exhibits growing errors with increasing delays, confirming previous findings [15]. These results demonstrate that exploiting Galilean symmetry within an EqF formulation provides a viable approach for joint navigation and delay estimation in GNSS-aided INS.

ACKNOWLEDGMENTS

The authors thank the ArduPilot community and in particular Andrew Tridgell for conducting the experimental flights. They also thank Alessandro Fornasier for valuable discussions.

This work was supported by BMK under the grant agreement 913968 (SAMURAI), by the ARO W911NF-21-2-0245, and by the Franco-Australian International Research Project “Advancing Autonomy for Unmanned Robotic Systems” (IRP ARS). The views and conclusions contained in this document are those of the authors only. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes.

REFERENCES

- [1] Giulio Delama, Alessandro Fornasier, Robert Mahony, and Stephan Weiss. Equivariant IMU Preintegration With Biases: A Galilean Group Approach. *IEEE Robotics and Automation Letters*, 10(1):724–731, 2025. ISSN 23773766. doi: 10.1109/LRA.2024.3511424.
- [2] Alireza Fatehi and Biao Huang. Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay. *Journal of Process Control*, 53:15–25, 5 2017. ISSN 0959-1524. doi: 10.1016/J.PROCONT.2017.02.010.
- [3] Alessandro Fornasier, Yonhon Ng, Christian Brommer, Christoph Bohm, Robert Mahony, and Stephan Weiss. Overcoming Bias: Equivariant Filter Design for Biased Attitude Estimation With Online Calibration. *IEEE Robotics and Automation Letters*, 7(4):12118–12125, 10 2022. ISSN 23773766. doi: 10.1109/LRA.2022.3210867.
- [4] Alessandro Fornasier, Yonhon Ng, Robert Mahony, and Stephan Weiss. Equivariant Filter Design for Inertial Navigation Systems with Input Measurement Biases. *2022 International Conference on Robotics and Automation (ICRA)*, pages 4333–4339, 5 2022. doi: 10.1109/ICRA46639.2022.9811778.
- [5] Alessandro Fornasier, Pieter van Goor, Eren Allak, Robert Mahony, and Stephan Weiss. MSCEqF: A Multi State Constraint Equivariant Filter for Vision-aided Inertial Navigation. *IEEE Robotics and Automation Letters*, 1 2023. ISSN 23773766. doi: 10.1109/LRA.2023.3335775.
- [6] Alessandro Fornasier, Yixiao Ge, Pieter Van Goor, Martin Scheiber, Andrew Tridgell, Robert Mahony, and Stephan Weiss. An Equivariant Approach to Robust State Estimation for the ArduPilot Autopilot System. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 11956–11962, 2024. ISSN 10504729. doi: 10.1109/ICRA57147.2024.10611108.
- [7] Alessandro Fornasier, Yixiao Ge, Pieter van Goor, Robert Mahony, and Stephan Weiss. Equivariant symmetries for inertial navigation systems. *Automatica*, 181:112495, 2025. ISSN 0005-1098. doi: 10.1016/J.AUTOMATICA.2025.112495.
- [8] Heike Frei, Matthias Burri, Florian Rems, and Eicke Alexander Risse. A robust navigation filter fusing delayed measurements from multiple sensors and its application to spacecraft rendezvous. *Advances in Space Research*, 72(7):2874–2900, 10 2023. ISSN 0273-1177. doi: 10.1016/J.ASR.2022.10.025.

- [9] Yixiao Ge, Pieter Van Goor, and Robert Mahony. Equivariant Filter Design for Discrete-time Systems. *Proceedings of the IEEE Conference on Decision and Control*, 2022-December:1243–1250, 2022. ISSN 25762370. doi: 10.1109/CDC51059.2022.9992342.
- [10] Yixiao Ge, Giulio Delama, Martin Scheiber, Alessandro Fornasier, Pieter van Goor, Stephan Weiss, and Robert Mahony. The difference between the left and right invariant extended Kalman filter. *Control Engineering Practice*, 167:106656, 2026. ISSN 0967-0661. doi: <https://doi.org/10.1016/j.conengprac.2025.106656>.
- [11] Yihan Guo, Oliviero Vouch, Simone Zocca, Alex Minetto, and Fabio Dovis. Enhanced EKF-Based Time Calibration for GNSS/UWB Tight Integration. *IEEE Sensors Journal*, 23(1):552–566, 1 2023. ISSN 15581748. doi: 10.1109/JSEN.2022.3223974.
- [12] Jonathan Kelly. All About the Galilean Group SGal(3). *arXiv:2312.07555*, 12 2023.
- [13] Jonathan Kelly and Gaurav S. Sukhatme. A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors. *Springer Tracts in Advanced Robotics*, 79:195–209, 2014. ISSN 1610-742X. doi: 10.1007/978-3-642-28572-1_14.
- [14] Jonathan Kelly, Nicholas Roy, and Gaurav S. Sukhatme. Determining the time delay between inertial and visual sensor measurements. *IEEE Transactions on Robotics*, 30(6):1514–1523, 12 2014. ISSN 15523098. doi: 10.1109/TRO.2014.2343073.
- [15] Jonathan Kelly, Christopher Grebe, and Matthew Giamou. A Question of Time: Revisiting the Use of Recursive Filtering for Temporal Calibration of Multisensor Systems. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 11 2021. doi: 10.1109/MFI52462.2021.9591176.
- [16] Alireza Khosravian, Jochen Trumpf, Robert Mahony, and Tarek Hamel. State estimation for invariant systems on Lie groups with delayed output measurements. *Automatica*, 68:254–265, 6 2016. ISSN 0005-1098. doi: 10.1016/J.AUTOMATICA.2016.01.024.
- [17] Changseung Kim, Geunsik Bae, Woojae Shin, Sen Wang, and Hyondong Oh. EKF-Based Radar-Inertial Odometry With Online Temporal Calibration. *IEEE Robotics and Automation Letters*, 10(7):7230–7237, 2025. ISSN 23773766. doi: 10.1109/LRA.2025.3575320.
- [18] Thomas Dall Larsen, Nils A. Andersen, Ole Ravn, and Niels Kjolstad Poulsen. Incorporation of time delayed measurements in a discrete-time Kalman filter. *Proceedings of the IEEE Conference on Decision and Control*, 4:3972–3977, 1998. ISSN 01912216. doi: 10.1109/CDC.1998.761918.
- [19] Mingyang Li and Anastasios I. Mourikis. Online Temporal Calibration for Camera–IMU Systems: Theory and Algorithms. *The International Journal of Robotics Research*, 33(7):947–964, 2014. doi: 10.1177/0278364913515286.
- [20] Robert Mahony, Jonathan Kelly, and Stephan Weiss. Galilean Symmetry in Robotics. *arXiv:2510.10468*, 10 2025.
- [21] Tara Mina, Lindsey Marinello, and John Christian. Remarks on stochastic cloning and delayed-state filtering. *arXiv:2508.21260*, 8 2025.
- [22] John Olof Nilsson, Isaac Skog, and Peter Handel. Joint state and measurement time-delay estimation of nonlinear state space systems. *10th International Conference on Information Sciences, Signal Processing and their Applications, ISSPA 2010*, pages 324–328, 2010. doi: 10.1109/ISSPA.2010.5605534.
- [23] Martin Scheiber, Alessandro Fornasier, Christian Bommert, and Stephan Weiss. Revisiting Multi-GNSS Navigation for UAVs – An Equivariant Filtering Approach. In *2023 21st International Conference on Advanced Robotics (ICAR)*, pages 134–141. IEEE, 12 2023. ISBN 979-8-3503-4229-1. doi: 10.1109/ICAR58858.2023.10406552.
- [24] Junlin Song, Antoine Richard, and Miguel Olivares-Mendez. Unleashing the Power of Discrete-Time State Representation: Ultrafast Target-based IMU-Camera Spatial-Temporal Calibration. *arXiv:2509.12846*, 2025.
- [25] Pieter van Goor, Tarek Hamel, and Robert Mahony. Equivariant Filter (EqF). *IEEE Transactions on Automatic Control*, 68(6):3501–3512, 6 2022. ISSN 15582523. doi: 10.1109/TAC.2022.3194094.
- [26] Pieter van Goor, Punjaya Wickramasinghe, Matthew Hampsey, and Robert Mahony. Constructive synchronous observer design for inertial navigation with delayed GNSS measurements. *European Journal of Control*, 80: 101047, 11 2024. ISSN 0947-3580. doi: 10.1016/J.EJCON.2024.101047.