Estimation of robot pose and phased-array radio antenna orientation for GNSS-denied navigation using factor graph optimisation

Glen Hjelmerud Mørkbak Sørensen, Torleiv H. Bryne, Kristoffer Gryte and Tor Arne Johansen

***Abstract*—This paper presents two-way factors for phased-array radio systems (PARS) range, azimuth angle, and elevation angle measurements for use in a factor graph optimisation (FGO)-based aided inertial navigation system (INS). This enables simultaneous estimation of robot pose and ground-mounted PARS antenna array orientation. Estimator performance is validated in a loss-of-GNSS scenario using Bluetooth Low Energy (BLE) PARS data collected from a multirotor drone experiment. The performance whenever the antenna orientation is determined offline using the Fast Optimal Matrix Algorithm (FOAM) is used as a benchmark, and we compare it against online estimation performance whenever radio orientation is initialised as the FOAM estimate (i.e., warm start) and identity (i.e., cold start). Outliers in the BLE PARS measurements are mitigated by the aided INS using the Tukey and Geman-McClure M-estimators, and we discuss the tuning of their threshold values and analyse subsequent effects on the FGO estimate in detail by evaluating three select threshold values for each. The results show comparable performance in NED position and PARS antenna orientation estimates for sensible M-estimator threshold values. This demonstrates the method's potential for use in ground-mounted radio orientation calibration, an important enabler for real-time GNSS-denied navigation applications using PARS.**

***Index Terms*—Online calibration, Factor graph optimisation, Matrix Lie groups, phased-array radio systems**

I. Introduction

Access to global navigation satellite systems (GNSS)-based navigation is essential in modern society – we expect these systems to work and provide accurate positioning and timing fix when we need them. At the same time, GNSS signals are extremely vulnerable to intentional interference such as blocking of the signals, i.e., *jamming*, or being tricked into following a different path due to the broadcasting of false signals, i.e., *spoofing* [1], [2]. In the maritime sector, findings from a recent report from the Royal Institute of Navigation found that a sizeable majority of the industry considers this a serious issue, with related problems such as unnecessary dependencies between GNSS receivers and electronic equipment on-board vessels posing safety risks when in or near areas affected by GNSS interference [3]. Due to these vulnerabilities, access to alternative navigation systems has become more important, as unmanned and autonomous systems are increasingly introduced into various aspects of everyday life. One group of such alternative navigation systems are those based on *phased-array radio systems* (PARS), which historically have been based on military- or industry-grade hardware [4], but has also been demonstrated more recently with low-cost Bluetooth Low Energy (BLE) hardware, as seen in e.g., [5] where UAV flight aided by BLE angle-of-arrival (AoA) measurements was demonstrated. In PARS-aided navigation in general, dealing with outliers from e.g., multipath is essential. This may be done using outlier rejection with the natural test, as was done in e.g., [6]. For BLE PARS, where measurements are considerably more noisy than equivalent measurements from military- or industry-grade hardware, dealing with such outliers is even more important.

One central aspect of PARS-based navigation is the calibration of the ground-mounted radio orientation relative to the navigation frame. If we have access to two time synchronised sets of observations in the GNSS- and PARS-frames, then this can be formulated as an *absolute orientation problem* and solved with, e.g., the Fast Optimal Matrix Algorithm (FOAM) algorithm [7], [8]. An approach more suitable for real-time application is presented in [6], [9], where a multiplicative extended Kalman filter is used to estimate the mounting angles of the PARS antenna array during UAV flight with the help of real-time kinematic (RTK)-GNSS measurements.

Factor graph optimisation (FGO) has become popular in robotics in recent years, particularly in simultaneous localisation and mapping (SLAM) applications where methods based on the extended Kalman filter (EKF) quickly become intractable for large problems [10]. Fixed-lag smoothed FGO based on iSAM2 [11] allows for information over a time horizon to be used in state estimation in a manner suitable for real-time applications, making it a popular choice in many robotics applications, see e.g., [12].

In [13], we tested an FGO-based estimation framework on experimental multirotor flight data and showcased the viability of inertial navigation in handover scenarios when switching from RTK position and compass to aiding by BLE angles and RTK range/barometer pressure. In this setup, the drone broadcasted BLE packages that were received by a ground-mounted PARS array that subsequently determined the angle-of-arrival (AoA) azimuth and elevation angles for use in aiding via the Direction Finding (DF) technique [14]. Both transmitter and receiver were equipped with the same BLE radio

The work is supported by the Research Council of Norway through the projects Phased-array radio systems for resilient localization and navigation of autonomous systems in GNSS-denied environments (PARNAV), no. 338789 and Multi-Sensor Data Timing, Synchronization and Fusion for Intelligent Robots (SentiPro), no. 327538.

The authors are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. Corresponding author e-mail: glen.h.m.sorensen@ntnu.no

hardware, but only the receiver was equipped with a PARS antenna. The mounting angles of the BLE radio antenna array on the ground were estimated in post-processing by comparing measured position derived from BLE with measured position from RTK-GNSS. The fact that we only had measurements from a single PARS array motivated our use of M-estimators [15] to *mitigate* rather than reject outliers, i.e., try to get some usable information from noisy measurements instead of simply discarding them.

In this paper, we present the following contributions:

- We present an overview of robust estimation using M-estimators applied to factor graph optimisation, discussing some of the background and motivation for their use in robotics and compare the effect of two select M-estimators - Tukey and Geman-McClure – to that of a least-squares estimator in a motivating example.
- We extend our FGO-based estimation framework to include simultaneous estimation of robot pose and ground radio orientation with two-way BLE PARS factors for radio-fixed measurements by deriving Jacobians for our PARS measurements with respect to a radio orientation increment on the $SO(3)$ matrix Lie group.
- Using BLE measurements alongside RTK-GNSS measurements during initialisation, we compare estimator performance with online estimation of radio orientation against our previous setup with offline estimation of the ground radio mounting angles using the FOAM algorithm.

The estimation performance is evaluated with both cold and warm start, where we assume alignment of the ground radio frame with the navigation frame, and where we initialise the mounting angles using our offline estimate from [13], respectively.

The remainder of the article is organised as follows: Section II present background theory, whilst Section III presents the incremental fixed-lag smoother that serves as the core of an inertial navigation system. We describe M-estimators and their use in robust estimation in factor graph optimisation in Section IV, before presenting our augmented BLE PARS factors in Section V. The experimental setup is described in Section VI, the results are presented in Section VII and discussed in Section VIII, before the work is summarised and concluded in Section IX.

II. Preliminaries

A. Notation and coordinate frames

We denote vectors and matrices using bold-face lowercase and uppercase letters, respectively, e.g., $\boldsymbol{a}$ and $\boldsymbol{A}$. $\boldsymbol{R}_b^a$ represents a rotation matrix that rotates a vector from coordinate frame $\{b\}$ to frame $\{a\}$. We consider four coordinate frames in this paper: the local North-East-Down navigation frame $\{n\}$, the body frame $\{b\}$ and the radio frame $\{r\}$. Hence, $\boldsymbol{p}_{nb}^r$ is the position in $\{b\}$ with respect to $\{n\}$ given in $\{r\}$.

B. Skew-symmetric matrices

For any vector $\boldsymbol{a} = [a_x \;\; a_y \;\; a_z]^\top \in \mathbb{R}^3$, the matrix $[\boldsymbol{a}]_\times \in \mathbb{R}^{3\times 3}$ is its skew-symmetric matrix, i.e., a matrix where $[\boldsymbol{a}]_\times = -[\boldsymbol{a}]_\times^\top$. The skew-symmetric matrix is related to the cross-product via

$$\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b}, \tag{1}$$

where the ordering may be switched using the relation

$$[\boldsymbol{a}]_\times \boldsymbol{b} = -[\boldsymbol{b}]_\times \boldsymbol{a} \Leftrightarrow \boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}. \tag{2}$$

C. Matrix Lie group theory

1) $SO(3)$ matrix Lie group: The $SO(3)$ matrix Lie group is defined as the set of rotation matrices $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ [16]:

$$SO(3) = \left\{ \boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}, \; \det \boldsymbol{R} = 1 \right\}, \tag{3}$$

The hat operator on $SO(3)$ is defined as

$$^\wedge\colon \mathbb{R}^3 \mapsto \mathfrak{so}(3) : \boldsymbol{\xi_o} \mapsto [\boldsymbol{\xi_o}]_\times \tag{4}$$

where $\boldsymbol{\xi}_o$ is the attitude increment. It maps $\boldsymbol{\omega}$ onto the *Lie algebra* $\mathfrak{so}(3)$ corresponding to $SO(3)$. *The capitalised exponential map* [16], $\mathrm{Exp}\,(\boldsymbol{\xi})$, is thus defined as the matrix exponential of the perturbation $\boldsymbol{\xi_o}^\wedge \in \mathfrak{so}(3)$

$$\begin{aligned} \mathrm{Exp}\,(\boldsymbol{\xi_o}) &\triangleq \exp\,(\boldsymbol{\xi_o}^\wedge) \\ &= \boldsymbol{I}_3 + [\boldsymbol{u}]_\times \sin(\theta) + [\boldsymbol{u}]_\times^2 (1 - \cos(\theta)), \end{aligned} \tag{5}$$

and is equivalent to the Rodrigues rotation formula [16].

2) $SE(3)$ matrix Lie group: The $SE(3)$ matrix Lie group is defined as the set of poses $\boldsymbol{T} \in \mathbb{R}^{4\times 4}$ [17]:

$$SE(3) \triangleq \left\{ \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix} \middle| \begin{array}{l} \boldsymbol{R} \in SO(3), \\ \boldsymbol{p} \in \mathbb{R}^3 \end{array} \right\}, \tag{6}$$

where $\boldsymbol{p}$ is a position vector. The hat operator on $SE(3)$ is defined as based on the *local perturbation* $\boldsymbol{\xi}$

$$\boldsymbol{\xi} \triangleq \begin{bmatrix} \boldsymbol{\xi}_o{}^\top & \boldsymbol{\xi}_p{}^\top \end{bmatrix}^\top \in \mathbb{R}^6, \tag{7}$$

where $\boldsymbol{\xi}_p$ is the small translational increment expressed in $\{b\}$:

$$^\wedge\colon \mathbb{R}^6 \mapsto \mathfrak{se}(3) : \boldsymbol{\xi} \mapsto \begin{bmatrix} [\boldsymbol{\xi}_o]_\times & \boldsymbol{\xi}_p \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{8}$$

It maps $\boldsymbol{\xi}$ onto the *Lie algebra* $\mathfrak{se}(3)$ corresponding to $SE(3)$. The exponential map [16], $\mathrm{Exp}\,(\boldsymbol{\xi})$, is thus defined as the matrix exponential of the mapped perturbation $\boldsymbol{\xi}^\wedge \in \mathfrak{se}(3)$

$$\mathrm{Exp}\,(\boldsymbol{\xi}) \triangleq \exp\,(\boldsymbol{\xi}^\wedge) = \exp\left(\begin{bmatrix} \boldsymbol{\xi}_o^\wedge & \boldsymbol{\xi}_p \\ \boldsymbol{0} & 0 \end{bmatrix} \right). \tag{9}$$

3) Useful identities and approximations: Some identities and approximations are central in deriving the Jacobians of measurements with respect to the Lie algebras, which we will present here. Firstly, $(\mathrm{Exp}\,(\boldsymbol{\xi}))^{-1} = \mathrm{Exp}\,(-\boldsymbol{\xi})$ [16]. Secondly, a matrix Lie group pose can be expressed as a function of an estimate and a local perturbation, i.e., $\boldsymbol{T} = \hat{\boldsymbol{T}}\mathrm{Exp}\,(\boldsymbol{\xi})$ [17].

For Jacobian derivations, the first-order approximation of the exponential map of $SE(3)$ is given as

$$\mathrm{Exp}\,(\boldsymbol{\xi}) \approx \begin{bmatrix} \boldsymbol{I} + [\boldsymbol{\xi}_o]_\times & \boldsymbol{\xi}_p \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{10}$$

and is used extensively. We note that the first-order approximation of $SO(3)$ is embedded in the top right.

III. Inertial Navigation System

An aided inertial navigation system (INS) based on the iSAM2 fixed-lag smoother in GTSAM [18] with smoother lag set to $t_s = 2\text{s}$ is used to maintain the state estimate $\hat{\boldsymbol{x}} \triangleq \left(\hat{\boldsymbol{T}}_b^n,\ \hat{\boldsymbol{v}}_{nb}^n,\ \hat{\boldsymbol{b}}_b,\ \hat{\boldsymbol{R}}_r^n\right) \in SE(3) \times \mathbb{R}^3 \times \mathbb{R}^6 \times SO(3)$, where $\boldsymbol{v}_{nb}^n$ is the robot velocity, $\hat{\boldsymbol{b}}_b$ denotes the accelerometer and angular rate sensor biases of the IMU and $\hat{\boldsymbol{R}}_r^n$, which is a new addition compared to the implementation in [13], is the radio orientation we wish to estimate. The prediction between updates is driven by the on-manifold IMU preintegrator found in GTSAM [12], [19], [20] which encodes the information from accelerometer and angular rate sensor measurements between time steps i and j as

$$\begin{aligned} \Delta\tilde{\boldsymbol{R}}_{ij} &= \boldsymbol{R}_i^\top \boldsymbol{R}_j \mathrm{Exp}\left(\boldsymbol{w}_o\right) \\ \Delta\tilde{\boldsymbol{v}}_{ij} &= \boldsymbol{R}_i^\top(\boldsymbol{v}_j - \boldsymbol{v}_i - \boldsymbol{g}^n \Delta t_{ij}) + \boldsymbol{w}_v \\ \Delta\tilde{\boldsymbol{p}}_{ij} &= \boldsymbol{R}_i^\top(\boldsymbol{p}_j - \boldsymbol{p}_i - \boldsymbol{v}_i\Delta t_{ij} - \frac{1}{2}\boldsymbol{g}\Delta {t_{ij}}^2) + \boldsymbol{w}_p, \end{aligned} \tag{11}$$

where $\Delta t_{ij} \triangleq t_j - t_i$ is the preintegration time and $\boldsymbol{g}^n \triangleq [0\ 0\ 9.81]^\top$ is the gravity vector in $\{n\}$. $\boldsymbol{w}_o$ and $\boldsymbol{w}_v$ are angular rate sensor and accelerometer noise, respectively, whilst $\boldsymbol{w}_p$ is the preintegration noise.

For the aiding, all measurements used need to have a Jacobian $\boldsymbol{H}$ with respect to the Lie algebra $\mathfrak{se}(3)$ defined for use in the fixed-lag smoother. During the initial calibration period, RTK-GNSS position and attitude measurements are used along with the BLE angular measurements and RTK range; their measurement models are given by

$$\boldsymbol{z}_{\text{RTK}}^n = \boldsymbol{p}_{nb}^n + \boldsymbol{\varepsilon}, \tag{12a}$$

$$\boldsymbol{h}_{\text{RTK}}(\hat{\boldsymbol{T}}) = \hat{\boldsymbol{p}}_{nb}^n, \tag{12b}$$

$$\boldsymbol{H}_{\text{RTK}} = \frac{\partial \boldsymbol{h}_{\text{RTK}}}{\partial \boldsymbol{\xi}^\wedge} = \begin{bmatrix}\boldsymbol{0}_{3\times3} & \hat{\boldsymbol{R}}_b^n\end{bmatrix}, \tag{12c}$$

and

$$\boldsymbol{z}_{\text{comp}}^n = \boldsymbol{R}_b^n \boldsymbol{l}_{\text{RTK}}^b + \boldsymbol{\varepsilon}, \tag{13a}$$

$$\boldsymbol{h}_{\text{comp}}(\hat{\boldsymbol{T}}) = \hat{\boldsymbol{R}}_b^n \boldsymbol{l}_{\text{RTK}}^b, \tag{13b}$$

$$\boldsymbol{H}_{\text{comp}} = \frac{\partial \boldsymbol{h}_{\text{comp}}}{\partial \boldsymbol{\xi}^\wedge} = \begin{bmatrix}-\hat{\boldsymbol{R}}_b^n [\boldsymbol{l}_{\text{RTK}}^b]_\times & \boldsymbol{0}_{3\times3}\end{bmatrix}, \tag{13c}$$

respectively, where $\boldsymbol{l}_{\text{RTK}}^b$ is the RTK baseline between the antennas on the robot. Whenever aiding measurements are available, they are added to the graph with their respective factors, whilst the current state of the preintegrator is added to the graph using GTSAM's `CombinedImuFactor`. Based on the updated IMU bias estimate, the preintegrator is subsequently reset. The modified BLE factors are covered in detail in Section V.

IV. Robust Estimation Using M-Estimators

M-estimators stem from robust statistics, formalised by [15], with the goal of providing a method of robust estimation with *contaminated distributions*, i.e., samples with outliers present. They are an alternative to *outlier rejection* e.g., the natural test [21, Ch. 7.6.1] where large outliers are simply discarded if the normalised residuals fall outside of a set threshold. An M-estimator is defined by its *cost* function $Q(\tilde{z})$ and corresponding *influence* function $\psi(\tilde{z}) = \frac{\partial Q}{\partial \tilde{z}}$. Furthermore, the *weight* of the M-estimator is given by $w(\tilde{z}) = \frac{\psi(\tilde{z})}{\tilde{z}}$.

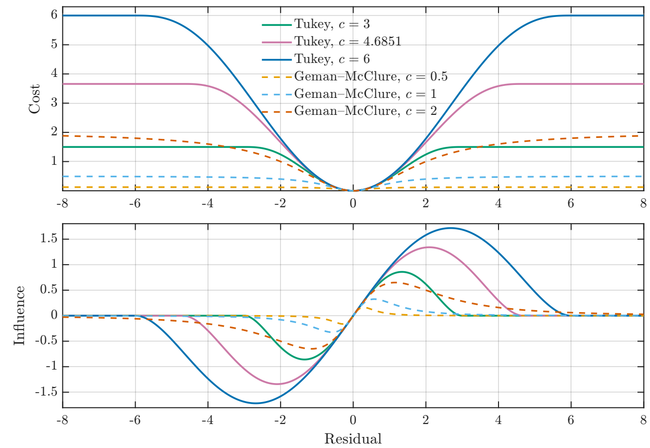

Fig. 1. Comparison of the Tukey (TK) and Geman-McClure (GM) M-estimator cost and influence functions for different threshold values. The middle value for either M-estimator is the default value.

Any M-estimator must adhere to three requirements [22]:

1) Bounded influence $\psi(\tilde{z})$.
2) Cost Q must be convex in $\tilde{z}$.
3) Whenever the curvature of the cost Q is singular, the influence must be non-zero.

If we consider the standard least squares estimator against these criteria, we see that it satisfies criteria 2 and 3, but its influence function is linear in the error and thus is not bounded. As a result, large outliers can greatly deteriorate the computation of sample mean and variance. For further details on M-estimators, see [22].

A. The Tukey M-estimator

The first selected robust error function is Tukey. The cost of the Tukey M-estimator is given by two different functions depending on whether the residual $\tilde{z}$ is within the threshold c:

$$Q_{\mathrm{T}}(\tilde{z}) = \begin{cases} \frac{c^2}{6}\left(1 - [1 - (\tilde{z} - (\tilde{z}/c)^2]^3\right), & \text{for } |\tilde{z}| \leq c \\ \frac{c^2}{6}, & \text{otherwise,} \end{cases} \tag{14}$$

where we remark that for large residuals, the resulting cost when using the Tukey M-estimator is constant. As a result, the influence and weight of such an outlier will be zero. This is visualised in Fig. 1, where the cost and influence of a residual $\tilde{z}$ for three choices of threshold are given and we can clearly see the piece-wise continuous nature of the influence nature of the M-estimator's influence. The threshold value $c = 4.6851$ is the default value for the Tukey M-estimator used in GTSAM.

B. The Geman-McClure M-estimator

The second selected robust error function is Geman-McClure. The cost of the Geman-McClure M-estimator is given by

$$Q_{\mathrm{G}}(\tilde{z}) = \frac{1}{2}\frac{c^2\tilde{z}^2}{c^2 + \tilde{z}^2}, \tag{15}$$

TABLE I

COMPARISON BETWEEN COST AND INFLUENCE OF AN INLIER AND OUTLIER WITH LEAST SQUARES, TUKEY, AND GEMAN-MCCLURE.

	Inlier $\tilde{z} = 0.3$		Outlier $\tilde{z} = 5$	
	Cost	Influence	Cost	Influence
Least squares	0.0450	0.3	12.5	5
Tukey $c = 4.6851$	0.0448	0.2975	3.6584	0
Geman-McClure $c = 1$	0.0413	0.252	0.4808	0.0074

where we remark that unlike the Tukey M-estimator this is a smooth function. This specific variant of Geman-McClure, which is the one implemented in GTSAM, is the *generalised* Geman-McClure M-estimator from [23], and is based on the original definition presented in [22]. The cost and influence of three values of c for Geman-McClure are also visualised in Fig. 1, where $c = 1$ is the default value used in GTSAM.

Here, we see the smooth nature of the influence, as well as a far more aggressive truncation of large outliers when compared to Tukey. This behaviour is suitable for real-time applications, but a potential drawback is the potential lack of convergence if the initial estimate is far from the ground truth.

C. Effect on factor graph estimate

Each measurement will have its own factor in the graph used by the fixed-lag smoother to update its state estimates. By default, no consideration is made to the validity of the factor information. The smoother takes in a sum of whitened residuals (normalised residuals with unit covariance) and tries to minimise the optimisation problem irrespective of actual quality. An example of how the M-estimators behave compared to the standard least-squares estimate with two whitened residual values of 0.3 and 5 is given in Table I. The resulting cost and influence demonstrate what has been covered so far when discussing the M-estimators: their behaviour mimics that of the least squares estimate if residuals are near zero, whilst outliers get aggressively de-weighted in cost and influence. Furthermore, the noticeable traits of Tukey and Geman-McClure are also visible – zero influence for outliers outside threshold for Tukey, and aggressive smoothing in cost for Geman-McClure. Even for the inlier, we remark that Geman-McClure will assign it a noticeable lower influence than Tukey, whose influence is only marginally smaller than the least squares estimate.

V. TWO-WAY PARS FACTORS

In this section, we derive the Jacobians for PARS factors used to estimate both the robot pose and the orientation of the radio antenna array. We also discuss nuances between our implementation and the existing implementations found in GTSAM [18].

A. Relation to existing GTSAM range- and bearing factors

GTSAM has two-way factors for range and bearing, as well as a combined range and bearing factor. With these factors, simultaneous estimation of the robot pose and beacon position or full beacon $SE(3)$ pose may be achieved, which is useful in simultaneous localisation and mapping (SLAM) use cases. However, as far as we can tell, in the latest stable release (v4.2) estimation of only the beacon's orientation relative to the navigation frame does not seem to be possible. However, this may be subject to change in future releases.

Since our goal is calibration of an antenna array orientation where the radio array is co-located with the local navigation frame, these factors do not really meet our requirements. In addition, as was the case in [24], these factors still assume body-fixed measurements, whilst the measurements here are given in the radio antenna array frame, $\{r\}$. Finally, we still wish to consider the azimuth- and elevation angle measurements separately when dealing with outlier rejection. For these reasons, we derive our own factors based on previous work.

B. Jacobians with respect to the robot pose Lie algebra

As in our previous work, we use the PARS measurement model below, found in e.g., [25]:

$$\boldsymbol{z}^r_{\text{PARS}} = \begin{bmatrix} \rho^r \\ \Psi^r \\ \alpha^r \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{p}^r_{rb}\|_2 \\ \arctan2\left(p^r_{rb,y}, p^r_{rb,x}\right) \\ \arctan2\left(-p^r_{rb,z}, \bar{\rho}\right) \end{bmatrix} + \boldsymbol{\varepsilon}. \tag{16}$$

Above, $\boldsymbol{\varepsilon}$ is zero-mean Gaussian noise and ρ^r, Ψ^r, and α^r denote the range, azimuth angle, and elevation angle measurement expressed in $\{r\}$, respectively.

In order to use our PARS measurement model in factor graph optimisation on the $SE(3)$ matrix Lie group, we have to define the Jacobians of the measurements in (16) with respect to the Lie algebra $\mathfrak{se}(3)$. In our previous work, [24] and [13], we assumed known rotation between $\{r\}$ and $\{n\}$ and only had to define the Jacobian with respect to $\boldsymbol{\xi}^\wedge$. The Jacobians were found via the chain rule

$$\boldsymbol{H}_\bullet = \frac{\partial h_\bullet}{\partial \boldsymbol{\xi}^\wedge} = \frac{\partial h_\bullet}{\partial \boldsymbol{p}^n_{rb}} \frac{\partial \boldsymbol{p}^n_{rb}}{\partial \boldsymbol{\xi}^\wedge}, \tag{17}$$

where $h_\bullet$ represents a measurements in (16) (i.e., $z_\bullet = h_\bullet(x) + \varepsilon_\bullet$). The resulting measurement Jacobians becomes:

$$\boldsymbol{H}_\rho = \frac{1}{\|\hat{\boldsymbol{p}}^r_{rb}\|_2} (\hat{\boldsymbol{p}}^r_{rb})^\top \boldsymbol{H}_p \tag{18}$$

$$\boldsymbol{H}_\Psi = \frac{1}{\hat{p}_x^2 + \hat{p}_y^2} \begin{bmatrix} -\hat{p}_y & \hat{p}_x & 0 \end{bmatrix} \boldsymbol{H}_p \tag{19}$$

$$\boldsymbol{H}_\alpha = \frac{1}{\|\hat{\boldsymbol{p}}^r_{rb}\|_2^2} \begin{bmatrix} \frac{p_x p_z}{\sqrt{\hat{p}_x^2 + \hat{p}_y^2}} & \frac{p_y p_z}{\sqrt{\hat{p}_x^2 + \hat{p}_y^2}} & -\sqrt{\hat{p}_x^2 + \hat{p}_y^2} \end{bmatrix} \boldsymbol{H}_p, \tag{20}$$

where $\boldsymbol{H}_p$ must be determined to relate the position to the Lie algebra. This was done in [24] via the first-order approximation:

$$\boldsymbol{p}^r_{rb} \approx \hat{\boldsymbol{p}}^r_{rb} + \frac{\partial \boldsymbol{p}^r_{rb}}{\partial \boldsymbol{\xi}^\wedge} \boldsymbol{\xi} = \hat{\boldsymbol{p}}^r_{rb} + \boldsymbol{H}_p \boldsymbol{\xi}, \tag{21}$$

and $\boldsymbol{H}_p = [\boldsymbol{0}_3 \ (\boldsymbol{R}^n_r)^\top \hat{\boldsymbol{R}}^n_b] \in \mathbb{R}^{3\times 6}$ is found in [13] to be the group Jacobian relating the measurement expressed in $\{r\}$ to $\mathfrak{se}(3)$.

C. Jacobians with respect to the radio orientation Lie algebra

In this paper, we will modify our factors to allow for estimation of the rotation $\boldsymbol{R}_r^n$ between $\{r\}$ and $\{n\}$. To achieve this, we define an additional local perturbation

$$\boldsymbol{\vartheta} \triangleq \begin{bmatrix}\vartheta_x & \vartheta_y & \vartheta_z\end{bmatrix}^\top \in \mathbb{R}^3 \Leftrightarrow \boldsymbol{\vartheta}^\wedge \triangleq [\boldsymbol{\vartheta}]_\times \in \mathfrak{so}(3), \quad (22)$$

which represents the increment in $\{r\}$ such that

$$\boldsymbol{R}_r^n = \hat{\boldsymbol{R}}_r^n \mathrm{Exp}\left(\boldsymbol{\vartheta}\right), \quad (23)$$

for which the Jacobians of the PARS measurement model will be given by

$$\frac{\partial h_\bullet}{\partial \boldsymbol{\vartheta}^\wedge} = \frac{\partial h_\bullet}{\partial \boldsymbol{p}_{rb}^r}\frac{\partial \boldsymbol{p}_{rb}^r}{\partial \boldsymbol{\vartheta}^\wedge}. \quad (24)$$

We recognise that the first term in the expanded expression matches the one in (17). Thus, we only have to determine a first-order approximation

$$\boldsymbol{p}_{rb}^r \approx \hat{\boldsymbol{p}}_{rb}^r + \frac{\partial \boldsymbol{p}_{rb}^r}{\partial \boldsymbol{\xi}^\wedge}\boldsymbol{\xi} + \frac{\partial \boldsymbol{p}_{rb}^r}{\partial \boldsymbol{\vartheta}^\wedge}\boldsymbol{\vartheta}, \quad (25)$$

to determine both Jacobians. The Jacobian with respect to $\boldsymbol{\xi}$ is given in (21), and the Jacobian with respect to $\boldsymbol{\vartheta}$ is found to be

$$\boldsymbol{H}_r \triangleq [(\hat{\boldsymbol{R}}_r^n)^\top \hat{\boldsymbol{p}}_{rb}^n]_\times \in \mathbb{R}^{3\times 3}, \quad (26)$$

with detailed derivations given in Appx. A.

VI. Experimental verification

This section outlines the multirotor drone hardware and software, its followed autopilot path, and the chosen evaluation criteria for analysis. Since we are reusing the experimental measurements from [13] to expand our estimation framework to include online estimation of the PARS mounting angles, we will only briefly summarise the experiment setup here, and refer to that paper for further details.

A. Experiment setup

The multirotor NED position given by both the RTK-GNSS position measurements and the position computed from the BLE angular measurements and RTK-GNSS range are both shown in Fig. 2. The hardware components of the drone and ground station are given in Table II. The drone operates with the Pixhawk Cube Orange autopilot with ArduCopter v4.6.0 and a modified version of the DUNE robotic middleware [26]. The IMU has an update rate of approx. 2000 Hz, the RTK-GNSS 1 Hz, and BLE measurements around 16-17 Hz. The experiment was conducted in a open field west of Trondheim, Norway, far away from any urban centre. The base itself was placed in the testing area and we detected no signs of GNSS interference.

The radio orientation prior is set to the FOAM estimate with uncertainty $0.5°$ during warm start, and to $\hat{\boldsymbol{R}}_r^n = \boldsymbol{I}_3$ with maximum antenna uncertainty $180°$ during cold start.

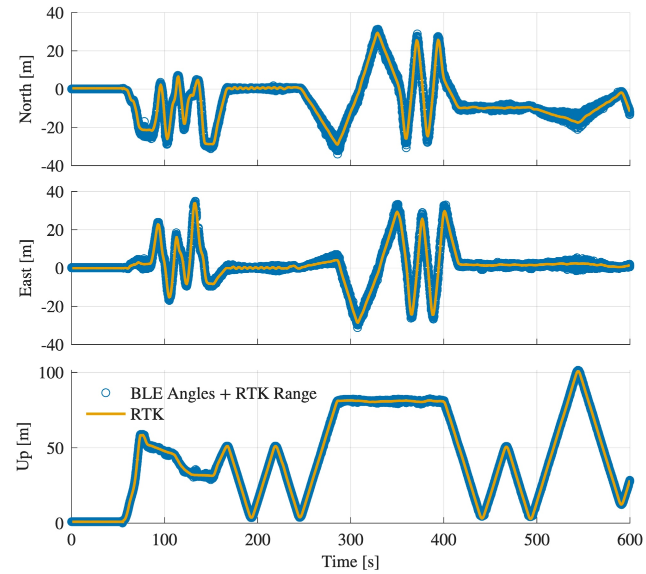

Fig. 2. Drone position from RTK-GNSS and inferred from BLE angular measurements and RTK range.

TABLE II
Overview of the hardware components of the multirotor drone and ground station.

Drone	Ground station
Nordic Semiconductor nRF52833	Nordic Semiconductor nRF52833
	12-element antenna array
SentiSystems Sentiboard v1.3	SentiSystems Sentiboard v1.3
Raspberry Pi 4B	Raspberry Pi 4B
Dual RTK uBlox GNSS receiver	RTK uBlox GNSS receiver
STIM300 IMU	

B. Evaluation criteria

To evaluate the performance of the different M-estimator configurations, the estimation error and corresponding 3σ-bounds, as well as the root mean square error (RMSE) during autopilot flight, are used. Specifically, we compare the multirotor *position* estimation error in the different modes and use the results of the offline calibration in [13] as a benchmark. The results are segmented into four: the initial 200s where RTK and BLE measurements are used alongside one another to calibrate the radio orientation, and then three subsequent segments SEG. 1 - SEG. 3 where RTK is unavailable. The segments are visualised with both text and pastel-colour backgrounds in the estimation error plots. In the offline runs we compare against, only RTK-GNSS measurements were used prior to the handover.

VII. Results

This section presents the results of both warm and cold start of the online calibration of the radio orientation with both the Tukey and Geman-McClure M-estimators. For brevity, the thresholds values are denoted by C1-C3 with numeric values given in their respective tables and plots. The C2-value is the default value used in GTSAM (see Section IV) for both M-estimators and is what we use in the offline benchmark using the rotation matrix estimated with FOAM.

TABLE III
NED POSITION RMSE WITHOUT ROBUST ESTIMATION SCHEMES. LOWEST VALUE SHOWN IN BOLD, SECOND LOWEST IN ITALICS.

		N [m]	E [m]	D [m]		N [m]	E [m]	D [m]
Offline	RTK	**0.344**	**0.312**	**0.238**	SEG. 2	0.932	**0.828**	**0.297**
Online warm		*0.602*	*0.406*	*0.313*		**0.868**	*1.120*	*0.390*
Online cold		0.741	0.462	0.355		*0.889*	1.297	0.431
Offline	SEG. 1	*0.596*	**0.387**	**0.246**	SEG. 3	**0.708**	0.692	**0.512**
Online warm		**0.572**	*0.414*	*0.273*		*0.928*	*0.524*	*0.534*
Online cold		0.616	0.424	0.293		1.043	**0.517**	0.548

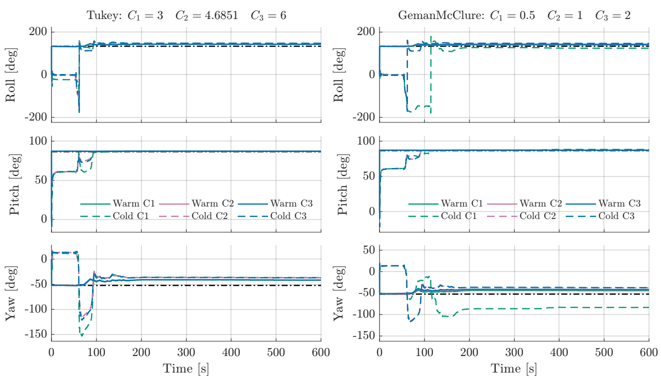

Fig. 3. Online radio orientation mounting angle estimates for Tukey and Geman–McClure. Offline estimate from FOAM represented by a black, dotted line.

A. Mounting angles

The evolution of the estimated mounting angles of the PARS antenna array is presented in Fig. 3, compared against the offline FOAM estimated given by a black, dotted line. One can see that with the exception of Geman-McClure with C1, all configurations converge close together during the period RTK-GNSS is available. We observe that the cold start configurations have notable transients before this, whilst the warm starts initialised to the FOAM estimate do end up at slightly different roll and yaw estimates, with the difference between the online estimate and the FOAM-estimated yaw being more prominent.

B. Baseline without robust estimation schemes

For a baseline comparison, the multirotor NED position RMSE without robust estimation schemes active is shown in Table III. In general, we see that offline performance has the lowest RMSE. There are a few exceptions to this. Namely, the warm start of the online calibration outperforms the offline run in North-position in SEG. 1 and SEG. 2, and cold start outperform both in SEG. 3 in East-position. We see from the inferred position of the BLE PARS in Fig. 2 that there is visible noise, e.g., in the outer ranges of the North-East position relative to the origin, and as the drone is near its maximum altitude. Some outliers as a result of reflections may also be excluded here. Both may contribute to the general performance after handover when compared to the RTK-GNSS-based aiding.

TABLE IV
NED POSITION RMSE FOR TUKEY WITH THRESHOLD VALUES C1 = 3, C2 = 4.6851, C3 = 6. LOWEST VALUE SHOWN IN BOLD, SECOND LOWEST IN ITALICS.

			N [m]		E [m]		D [m]	
		C#	Warm	Cold	Warm	Cold	Warm	Cold
RTK	Offline	C2	**0.344**	**0.344**	**0.312**	**0.312**	**0.238**	**0.238**
	Online	C1	0.606	1.164	*0.438*	0.618	*0.901*	0.477
		C2	*0.605*	*0.750*	0.441	*0.465*	0.902	*0.353*
		C3	*0.605*	0.769	0.441	0.470	0.902	0.358
SEG. 1	Offline	C2	0.596	**0.596**	**0.387**	*0.387*	**0.246**	**0.246**
	Online	C1	**0.569**	0.720	0.425	**0.371**	*0.271*	0.331
		C2	*0.572*	*0.617*	*0.422*	0.413	0.272	*0.293*
		C3	*0.572*	0.619	0.423	0.412	0.273	0.294
SEG. 2	Offline	C2	0.933	0.933	**0.828**	**0.828**	**0.298**	**0.298**
	Online	C1	*0.870*	1.177	*1.105*	*1.095*	*0.387*	*0.419*
		C2	**0.866**	**0.894**	1.117	1.291	0.389	0.430
		C3	**0.866**	*0.902*	1.117	1.283	0.390	0.429
SEG. 3	Offline	C2	**0.710**	**0.710**	0.697	0.697	**0.512**	**0.512**
	Online	C1	*0.919*	1.278	0.536	*0.593*	0.535	0.581
		C2	0.923	*1.047*	*0.530*	**0.523**	*0.534*	*0.549*
		C3	0.924	1.054	**0.528**	**0.523**	*0.534*	0.550

C. Tukey M-estimator

The NED position RMSE results are given in Table IV, whilst the estimation error and corresponding 3σ-bounds are shown in Fig. 4. The general trend is that the online runs are equal or slightly worse with respect to RSME. The exceptions here are in North-position Segs. 1 and 2 where warm-start online configurations perform slightly better, as well as SEG. 1 and SEG. 3 in East-position where in the first case, the cold-start C1-configuration has marginally better performance than other configurations and modes. The estimation error appears fairly consistent for all modes and configurations when considering the estimation error plots, although we do observe some more notable spikes in the North- and East-position estimation error in the RTK initialisation phase and towards the end in SEG. 3. We also observe higher uncertainty in East-position in the online cases during warm-start than during cold-start before the drone takes off, though their change over time after this point is comparable.

D. Geman-McClure M-estimator

The NED position RMSE results are given in Table V, whilst the estimation error and corresponding 3σ-bounds are shown in Fig. 5. We observe the same pattern seen for the Tukey M-estimator with respect to the RMSE, but with the online warm-start performing slightly better in North-position in SEG. 1 and SEG. 2. In SEG. 3, configurations C2 and C3 also perform marginally better than offline, whilst the C1 configuration performs marginally worse. The cold-start C1 configuration performs noticeably worse than the other configurations and its Tukey counterpart in all segments after RTK is lost. This is clearly visible in the estimation error plot

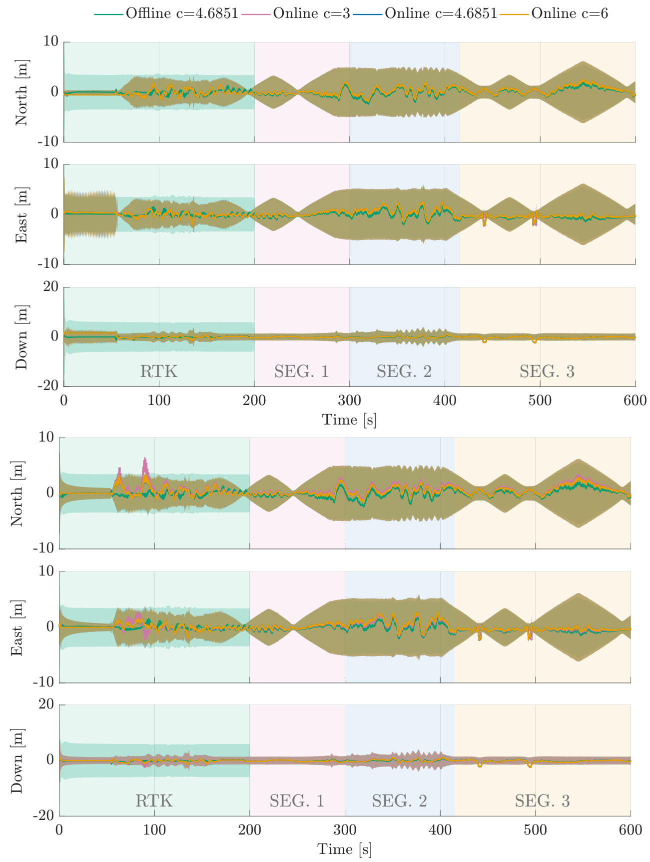

Fig. 4. NED position estimation error and 3σ-bounds (same-coloured dashed line) for the Tukey M-estimator in warm- and cold start of the online radio orientation calibration.

as well, where the estimator struggles to maintain a consistent position estimate throughout the run with this configuration.

VIII. Discussion

Apart from the C1 configuration for Geman-McClure, the differences in RMSE can be said to be negligible. Given that the offline run is based on a solution of the absolute orientation problem (AOP) across the whole trajectory, it is perhaps not unsurprising that it generally outperforms the online cases. This is supported by the fact that the configurations with warm-start – where the radio orientation is initialised to the solution to the AOP – generally perform better than their cold-start counterparts. The difference in yaw mounting angle observed between the offline and online configurations may be a result of noise. The fact that this does not appear to significantly affect the position estimate may be due to the relatively short distances we operate from the antenna array compared to the setup found in [6]. Otherwise, we may have observed a greater effect from the differing antenna yaw angle estimates.

The fact that the C1-configurations are, in general, the worst-performing during cold starts is interesting. Whilst this is not that impactful for Tukey, the consequences for Geman-McClure are apparent. Since both estimators have the same lack of prior information on the antenna orientation during

TABLE V
NED position RMSE for GemanMcClure with threshold values C1 = 0.5, C2 = 1, C3 = 2. Lowest value shown in bold, second lowest in italics.

			N [m]		E [m]		D [m]	
		C#	Warm	Cold	Warm	Cold	Warm	Cold
RTK	Offline	C2	**0.344**	**0.344**	**0.312**	**0.312**	**0.238**	*0.238*
	Online	C1	*0.531*	*0.444*	*0.367*	0.488	0.883	**0.235**
		C2	0.579	0.709	0.423	*0.426*	0.882	0.333
		C3	0.594	0.753	0.397	0.457	*0.306*	0.353
SEG. 1	Offline	C2	0.588	**0.588**	**0.392**	**0.392**	**0.246**	*0.246*
	Online	C1	**0.559**	2.205	*0.399*	6.635	*0.262*	**0.229**
		C2	*0.566*	*0.610*	0.414	*0.411*	0.269	0.292
		C3	0.571	0.616	0.421	*0.411*	0.272	0.293
SEG. 2	Offline	C2	0.937	0.937	**0.828**	**0.828**	**0.300**	**0.300**
	Online	C1	*0.886*	9.332	*1.052*	7.245	*0.377*	0.496
		C2	**0.868**	**0.896**	1.099	*1.280*	0.386	*0.430*
		C3	**0.868**	*0.899*	1.115	1.283	0.390	*0.430*
SEG. 3	Offline	C2	**0.737**	**0.737**	0.760	0.760	**0.519**	*0.519*
	Online	C1	0.938	1.407	0.782	4.067	0.562	**0.501**
		C2	*0.922*	1.059	*0.615*	*0.605*	*0.537*	0.553
		C3	0.929	*1.055*	**0.529**	**0.521**	0.538	0.553

cold start, this difference can most likely be attributed to the difference in behaviour, as seen in the comparison plot Fig. 1. Geman-McClure with C1 is extremely aggressive in de-weighting the influence of even moderate outliers, a feature that could be seen as incompatible with a cold-start radio orientation estimation, given the resulting innovations would subsequently be large as long as the frames are not naturally aligned. Here, perhaps an adaptive loss function as that explored in e.g., [27] could improve performance. Alternatively, during calibration whenever RTK is active, it may be better to simply use the BLE measurements as is to ensure we use the information available in the measurements to their full extent to quickly converge to a decent radio orientation estimate. Increasing the smoother lag from 2s to e.g., 5-10s may also have a positive effect on the estimate, albeit at a computational cost. The short smoother lag was deemed sufficient due to the high rate of BLE PARS measurements, but a longer time horizon could possibly benefit the online estimation. Of course, when targeting real-time applications, care must be put into ensuring that the navigation system can deliver state updates in a timely manner.

With sensible threshold values, the cold-start configurations for both Tukey and Geman-McClure appear to be a good starting point for estimation if an initial calibration run is possible to perform. Afterwards, the estimator can either be configured with this rotation as its defined $\hat{\boldsymbol{R}}^n_r$ either offline or as the initial estimate for warm-start.

IX. Conclusion

In this paper, we augmented our FGO-based estimation framework with two-way PARS factors, enabling us to cal-

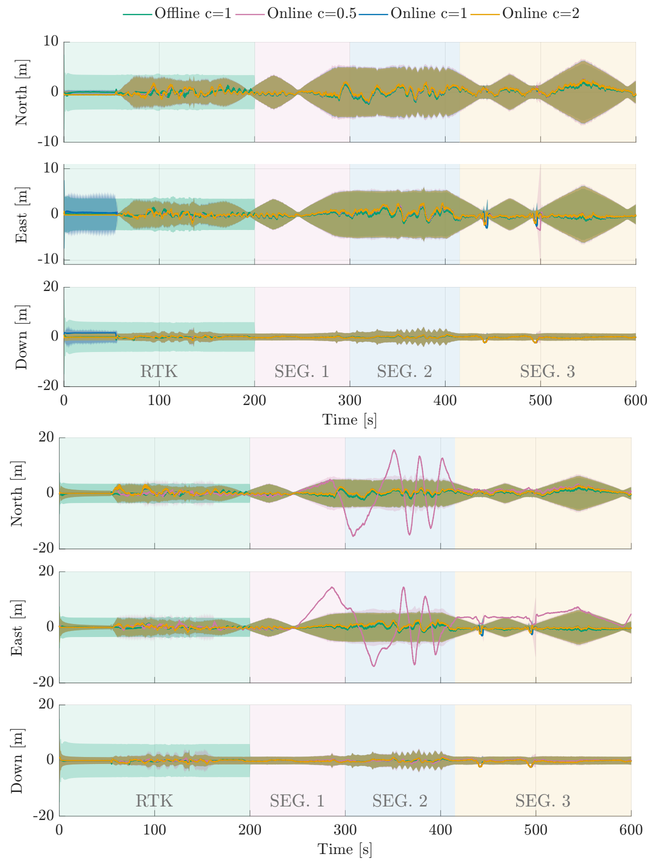

Fig. 5. NED position estimation error and 3σ-bounds (same-coloured dashed line) for the Geman-McClure M-estimator in warm- and cold start of the online radio orientation calibration.

ibrate the radio orientation relative to $\{n\}$, an important step towards real-time operations with the technology. The factors' were evaluated on multirotor flight data using the Tukey- and Geman-McClure M-estimators to mitigate the effect of outliers. We compared the online runs with three configurations with the results using the offline factors with the recommended default threshold. We found that, in general, both cold- and warm-start performed well, with equivalent or marginally worse performance with respect to RMSE. The exception was the most aggressive configurations for both, where the aggressive Geman-McClure settings yielded poorer performance, a result attributed to the nature of the M-estimator for low threshold values being a poor match for a cold-start calibration problem.

Future work includes considering alternative M-estimator loss functions and test calibration in more real-time scenarios for multirotor flights and other relevant robotic applications.

ACKNOWLEDGEMENTS

The authors would like to thank colleagues at NTNU for their help in conducting field experiments with the multirotor drone. Johan Suárez at the Department of Electronic Systems and Pål Kvaløy and Morten Einarsve at the Department of Engineering Cybernetics.

REFERENCES

[1] A. Pinker and C. Smith, "Vulnerability of the GPS Signal to Jamming," *GPS Solutions*, vol. 3, no. 2, pp. 19–27, Oct. 1999.

[2] F. Dovis, Ed., *GNSS interference, threats, and countermeasures*, ser. Artech House GNSS technology and applications series. Boston: Artech House, 2015, oCLC: ocn893017345.

[3] Royal Institute of Navigation, "IMPACTS OF GNSS INTERFERENCE ON MARITIME SAFETY." [Online]. Available: https://rin.org.uk/page/RIN_Maritime_Report

[4] K. Gryte, T. H. Bryne, and T. A. Johansen, "Phased Array Radio Navigation: UAV Field Tests During GNSS Jamming," in *2024 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, Chania, Crete, Greece, Jun. 2024, pp. 586–593.

[5] M. L. Sollie, K. Gryte, T. H. Bryne, and T. A. Johansen, "Automatic Recovery of Fixed-Wing Unmanned Aerial Vehicle Using Bluetooth Angle-of-Arrival Navigation," in *2024 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, Chania, Crete, Greece, Jun. 2024, pp. 390–397.

[6] M. Okuhara, T. H. Bryne, K. Gryte, and T. A. Johansen, "Phased Array Radio Navigation System on UAVs: In-Flight Calibration," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 3, p. 51, Oct. 2023.

[7] F. L. Markley, "Attitude determination using vector observations: A fast optimal matrix algorithm," in *NASA Goddard Space Flight Center*, Feb. 1993, nTRS Author Affiliations: NASA Goddard Space Flight Center NTRS Document ID: 19930015542 NTRS Research Center: Legacy CDMS (CDMS).

[8] M. Lourakis, "An efficient solution to absolute orientation," in *2016 23rd Int. Conf. on Pattern Recognition (ICPR)*. Cancun, Mexico: IEEE, Dec. 2016, pp. 3816–3819.

[9] M. Okuhara, T. H. Bryne, K. Gryte, and T. A. Johansen, "Phased Array Radio Navigation System on UAVs: Real-Time Implementation of In-flight Calibration," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1152–1159, 2023.

[10] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017. [Online]. Available: http://www.nowpublishers.com/article/Details/ROB-043

[11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE Int. Conf. on Robotics and Automation*. Shanghai, China: IEEE, May 2011, pp. 3281–3288. [Online]. Available: http://ieeexplore.ieee.org/document/5979641/

[12] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, Jul. 2015.

[13] G. H. M. Sørensen, T. H. Bryne, K. Gryte, and T. A. Johansen, "Bluetooth Phased-array Aided Inertial Navigation Using Factor Graphs: Experimental Verification," Feb. 2026, arXiv:2602.17407 [eess]. [Online]. Available: http://arxiv.org/abs/2602.17407

[14] Nordic Semiconductor, "Bluetooth Direction Finding – Presentation of technology." [Online]. Available: https://www.nordicsemi.com/Products/Wireless/Bluetooth-Direction-Finding

[15] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[16] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," Dec. 2021, arXiv:1812.01537 [cs]. [Online]. Available: http://arxiv.org/abs/1812.01537

[17] T. D. Barfoot and P. T. Furgale, "Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems," *IEEE Trans. on Robotics*, vol. 30, no. 3, pp. 679–693, Jun. 2014.

[18] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[19] T. Lupton and S. Sukkarieh, "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions," *IEEE Trans. on Robotics*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[20] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," in *2014 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014, pp. 4290–4297.

[21] F. Gustafsson, *Statistical sensor fusion*, 3rd ed. Studentlitteratur, 2010.

[22] Z. Zhang, “Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting,” *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, Jan. 1997.

[23] P. Agarwal, “Robust Graph-Based Localization and Mapping,” PhD Thesis, University of Freiburg, Freiburg, Germany, 2015.

[24] G. H. M. Sørensen, T. H. Bryne, K. Gryte, T. Synnevåg, and T. A. Johansen, “Robust phased-array radio system aided inertial navigation using factor graph optimisation,” in *2025 European Control Conference (ECC)*, Thessaloniki, Greece, Jun. 2025, pp. 1787–1794.

[25] K. Gryte, T. Bryne, and T. Johansen, “Unmanned aircraft flight control aided by phased-array radio navigation,” *Journal of Field Robotics*, vol. 38, Dec. 2020.

[26] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, “The LSTS toolchain for networked vehicle systems,” in *2013 MTS/IEEE OCEANS*. Bergen, Norway: IEEE, Jun. 2013, pp. 1–9.

[27] E. Ahmadi, A. Olama, P. Välisuo, and H. Kuusniemi, “Adaptive Factor Graph-Based Tightly Coupled GNSS/IMU Fusion for Robust Positionin,” Nov. 2025, arXiv:2511.23017 [cs]. [Online]. Available: http://arxiv.org/abs/1812.01537

APPENDIX A
RADIO ORIENTATION INCREMENT JACOBIAN

We start the Jacobian derivation by expanding the position vector in homogeneous coordinates

$$\begin{aligned}\begin{bmatrix}\boldsymbol{p}_{rb}^r\\1\end{bmatrix} &= \begin{bmatrix}(\boldsymbol{R}_r^n)^\top & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\boldsymbol{R}_b^n & 0\\0 & 1\end{bmatrix}\begin{bmatrix}(\boldsymbol{R}_b^n)^\top(\boldsymbol{p}_{nr}^n - \boldsymbol{p}_{nb}^n)\\1\end{bmatrix}\\ &= \begin{bmatrix}(\boldsymbol{R}_r^n)^\top & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\boldsymbol{R}_b^n & 0\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}(\boldsymbol{R}_b^n)^\top & -(\boldsymbol{R}_b^n)^\top\boldsymbol{p}_{nb}^n\\0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_{nr}^n\\1\end{bmatrix}\\ &= \boldsymbol{T}((\boldsymbol{R}_r^n)^\top,\ \boldsymbol{0})\boldsymbol{T}(-\boldsymbol{R}_b^n,\ \boldsymbol{0})\boldsymbol{T}^{-1}(\boldsymbol{R}_b^n,\ \boldsymbol{p}_{nb}^n)\begin{bmatrix}\boldsymbol{p}_{nr}^n\\1\end{bmatrix},\end{aligned}$$

and we use the relations $\boldsymbol{T} = \hat{\boldsymbol{T}}\mathrm{Exp}\left(\boldsymbol{\xi}\right)$ [17] and $(\hat{\boldsymbol{T}}\mathrm{Exp}\left(\boldsymbol{\xi}\right))^\top = \mathrm{Exp}\left(-\boldsymbol{\xi}\right)(\hat{\boldsymbol{T}})^\top$ [16] to express the position as a function of our estimates with small perturbations

$$\begin{aligned}\begin{bmatrix}\boldsymbol{p}_{rb}^r\\1\end{bmatrix} &= \begin{bmatrix}(\hat{\boldsymbol{R}}_r^n\mathrm{Exp}\left(\boldsymbol{\vartheta}\right))^\top & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n\mathrm{Exp}\left(\boldsymbol{\xi}\right) & 0\\0 & 1\end{bmatrix}\\ &\quad\left(\begin{bmatrix}\hat{\boldsymbol{R}}_b^n & \hat{\boldsymbol{p}}_{nb}^n\\0 & 1\end{bmatrix}\mathrm{Exp}\left(\boldsymbol{\xi}\right)\right)^{-1}\begin{bmatrix}\boldsymbol{p}_{nr}^n\\1\end{bmatrix}\\ &= \begin{bmatrix}\mathrm{Exp}\left(-\boldsymbol{\vartheta}\right)\hat{\boldsymbol{R}}_n^r & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n\mathrm{Exp}\left(\boldsymbol{\xi}\right) & 0\\0 & 1\end{bmatrix}\\ &\quad\mathrm{Exp}\left(-\boldsymbol{\xi}\right)\begin{bmatrix}(\hat{\boldsymbol{R}}_b^n)^\top & -(\hat{\boldsymbol{R}}_b^n)^\top\hat{\boldsymbol{p}}_{nb}^n\\0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_{nr}^n\\1\end{bmatrix}.\end{aligned}$$

Then use the first-order approximation of the exponential map:

$$\begin{aligned}\begin{bmatrix}\boldsymbol{p}_{rb}^r\\1\end{bmatrix} &\approx \begin{bmatrix}(\boldsymbol{I}_3 - [\boldsymbol{\vartheta}]_\times)\hat{\boldsymbol{R}}_n^r & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n(\boldsymbol{I}_3 + [\boldsymbol{\xi}_o]_\times) & 0\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}\boldsymbol{I}_3 + [\boldsymbol{\xi}_o]_\times & \boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top(\hat{\boldsymbol{p}}_{nb}^n - \boldsymbol{p}_{nr}^n)\\1\end{bmatrix}\\ &= \begin{bmatrix}(\boldsymbol{I}_3 - [\boldsymbol{\vartheta}]_\times)\hat{\boldsymbol{R}}_n^r & 0\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n + \hat{\boldsymbol{R}}_b^n([\boldsymbol{\xi}_o]_\times)^2 & \hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p + \hat{\boldsymbol{R}}_b^n[\boldsymbol{\xi}_o]_\times\boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top(\hat{\boldsymbol{p}}_{nb}^n - \boldsymbol{p}_{nr}^n)\\1\end{bmatrix}\end{aligned} \tag{27}$$

This can be simplified further by taking into account that squared and cross terms will be very small (approximately 0), such that Eq. (27) can be simplified further

$$\begin{aligned}\begin{bmatrix}\boldsymbol{p}_{rb}^r\\1\end{bmatrix} &\approx \begin{bmatrix}(\boldsymbol{I}_3 - [\boldsymbol{\vartheta}]_\times)\hat{\boldsymbol{R}}_n^r & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n & \hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top(\hat{\boldsymbol{p}}_{nb}^n - \boldsymbol{p}_{nr}^n)\\1\end{bmatrix}\\ &= \begin{bmatrix}\hat{\boldsymbol{R}}_n^r - [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r & 0\\0 & 1\end{bmatrix}\begin{bmatrix}-\hat{\boldsymbol{R}}_b^n & \hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top(\hat{\boldsymbol{p}}_{nb}^n - \boldsymbol{p}_{nr}^n)\\1\end{bmatrix}\\ &= \begin{bmatrix}-(\hat{\boldsymbol{R}}_n^r - [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r)\hat{\boldsymbol{R}}_b^n & (\hat{\boldsymbol{R}}_n^r - [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r)\hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top(\hat{\boldsymbol{p}}_{nb}^n - \boldsymbol{p}_{nr}^n)\\1\end{bmatrix}\end{aligned}$$

Again we have a cross term assumed to be small. Hence,

$$\begin{aligned}\begin{bmatrix}\boldsymbol{p}_{rb}^r\\1\end{bmatrix} &\approx \begin{bmatrix}-\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n + [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n & \hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\0 & 1\end{bmatrix}\\ &\quad\begin{bmatrix}-(\hat{\boldsymbol{R}}_b^n)^\top\hat{\boldsymbol{p}}_{rb}^n\\1\end{bmatrix}\\ &= \begin{bmatrix}(\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n - [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n)(\hat{\boldsymbol{R}}_b^n)^\top\hat{\boldsymbol{p}}_{rb}^n + \hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\1\end{bmatrix}\\ &= \begin{bmatrix}\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{p}}_{rb}^n - [\boldsymbol{\vartheta}]_\times\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{p}}_{rb}^n + \hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\1\end{bmatrix}\\ &= \begin{bmatrix}\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{p}}_{rb}^n + [\hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{p}}_{rb}^n]_\times\boldsymbol{\vartheta} + \hat{\boldsymbol{R}}_n^r\hat{\boldsymbol{R}}_b^n\boldsymbol{\xi}_p\\1\end{bmatrix}\end{aligned}$$

Now we recognise the structure of the desired first-order approximation of the robot position relative to the $\{r\}$ frame:

$$\boldsymbol{p}_{rb}^r \approx \hat{\boldsymbol{p}}_{rb}^r + [(\hat{\boldsymbol{R}}_r^n)^\top\hat{\boldsymbol{p}}_{rb}^n]_\times\boldsymbol{\vartheta} + \begin{bmatrix}0 & (\hat{\boldsymbol{R}}_r^n)^\top\hat{\boldsymbol{R}}_b^n\end{bmatrix}\boldsymbol{\xi}. \tag{28}$$

where

$$\boldsymbol{H}_r \triangleq [(\hat{\boldsymbol{R}}_r^n)^\top\hat{\boldsymbol{p}}_{rb}^n]_\times \in \mathbb{R}^{3\times3}, \tag{29}$$

relates radio orientation between $\{r\}$ and $\{n\}$ to $\mathfrak{so}(3)$, whilst

$$\boldsymbol{H}_p = \begin{bmatrix}0 & (\hat{\boldsymbol{R}}_r^n)^\top\hat{\boldsymbol{R}}_b^n\end{bmatrix} \in \mathbb{R}^{3\times6}, \tag{30}$$

as before relates the robot position expressed in $\{r\}$ to $\mathfrak{se}(3)$.