

iCalib: Inertial Aided Multi-Sensor Calibration

Yulin Yang¹, Woosik Lee¹, Philip Osteen², Patrick Geneva¹, Xingxing Zuo³ and Guoquan Huang¹

Abstract—An array of heterogeneous sensors available on most autonomous robots are widely used for state estimation and environmental perception, requiring accurate calibration of intrinsic, spatial, and temporal sensing parameters. In this paper, we present a holistic non-linear least squares (NLS) based multi-sensor calibration system, which exploits high-rate inertial navigation and can handle a multitude of asynchronous sensors commonly found on robots (e.g., IMU, cameras, LiDARs, and wheel odometry) while requiring only initial calibration guesses. We perform extensive simulations to verify both the system’s accuracy, consistency, and performance in the ideal motion case. In particular, we perform in-depth study on how planar motion can cause calibration degeneracy and how it directly impacts our ability to calibrate parameters, which is of practical significance for unmanned ground vehicles (UGVs). To demonstrate the practical challenges in performing calibration in these degenerate scenarios, we further evaluate the proposed system on a series of real-world datasets and compare against existing state-of-the-art calibration toolboxes.

I. INTRODUCTION AND RELATED WORK

In an autonomous navigation system, inertial measurement units (IMUs), multiple cameras, range sensors (e.g., LiDARs), and wheel encoders are widely used together to estimate the robot’s 6 degree-of-freedom (DoF) pose (position and orientation) through perceiving the environment and sensing its own kinematic motion. For instance, IMUs and wheel encoders can directly measure the robot’s motion (e.g., angular and linear velocity, and linear acceleration), which can provide accurate state estimation if fused with cameras [1], [2] or LiDARs [3], [4]. In addition, texture-rich images from cameras and accurate 3D points from LiDARs can be jointly used for obstacle detection and environment reconstruction [5]. To properly fuse measurements from these heterogeneous multi-modal sensors in order to accomplish the above tasks, it is essential to accurately calibrate both the spatial and temporal relationships for these sensors.

The spatial (extrinsic) calibration problem refers to estimating the 6 DoF rigid transformation between pairs of sensors (e.g., IMU-camera, camera-LiDAR, and camera-wheel), while the temporal calibration is to find the time offset between sensor measurement clock frame of references which can occur due to asynchronous hardware triggering or

TABLE I: Comparison of different multi-sensor calibration algorithms in the literature. t_{off} refers to time offset calibration between sensors.

Algorithm	IMU	Camera	LiDAR	Wheel	t_{off}	Target
iCalib	×	×	×	×	×	×
Kalibr [6]	×	×			×	×
COC [7]		×		×		×
MSG-CAL [8]		×	×			×
LI-Calib [9]	×		×			
MIMC [10]	×	×			×	
LIC [4]	×	×	×		×	
VIWO [2]	×	×		×	×	

raw data transmission times. Both are required to accurately fuse multi-sensor data for downstream autonomy processes (e.g., navigation, perception).

To date, substantial research efforts [6]–[9], [11]–[15], summarized in Table I, have been taken to address spatial and/or temporal calibration. We here briefly review a few notable examples. As one of the state-of-the-art methods, Kalibr [6] is a multi-camera, multi-IMU, and IMU-camera calibration system based on continuous-time batch estimation. Both camera and IMU intrinsics, spatial-temporal parameters between IMU and cameras can be calibrated. Lv et al. [9] adopted a B-Spline based continuous-time trajectory formulation for fusing IMU and LiDAR readings with extrinsic calibration. The point-to-surfel correspondences are utilized to enhance the calibration accuracy, hence no specific target is needed. Cam-Odom-Cal (COC) [7] uses a two-stage pipeline that handles both intrinsic and extrinsic calibration for multiple cameras and a wheel odometer. The camera intrinsics are first calibrated using a chessboard, and the extrinsics between the cameras and wheel odometry are calibrated via ego-motion alignment. However, both methods [7], [9] do not consider the temporal offset between sensors.

There also has been great interest in camera-LiDAR calibrations. For example, Owens et al. [8] proposed MSG-CAL, a general framework for estimating rigid transformation between multiple cameras and range sensors. Geometric features (points, lines, or planes) extracted from sensor data are used to jointly optimize a cost function yielding extrinsic calibration between sensors. Geiger et al. [11] proposed a single-shot camera-LiDAR extrinsic calibration. However, many chessboards are needed in practice and should be placed carefully to ensure that the optimization problem is well-posed for all degrees of freedom. Similarly, Zhou et al. [12] proposed a camera-LiDAR extrinsic calibration by using line-to-plane correspondences with one calibration board. Although camera-LiDAR extrinsics can be acquired through these works [8], [11], [12], none of these algorithms support temporal calibration. They rely on overlapping FoV among cameras and LiDARs and are unable to handle proprioceptive sensors (e.g., IMU and wheel odometry) which

This work was partially supported by the University of Delaware (UD) College of Engineering, the NSF (IIS-1924897), and the ARL (W911NF-19-2-0226, W911NF-20-2-0098, JWS 10-051-003). Yang was partially supported by the University Doctoral Fellowship and Geneva was also partially supported by the Delaware Space Grant College and Fellowship Program (NASA Grant NNX15AII9H).

¹The authors are with Robot Perception and Navigation Group, University of Delaware, Newark, DE 19716, USA. {yuyang, woosik, pgeneva, ghuang}@udel.edu

²The author is with CCDC Army Research Laboratory (ARL), Adelphi, MD 20783, USA {philip.r.osteen.civ}@mail.mil

³The author is with Institute of Cyber-System and Control, Zhejiang University, Hangzhou, China. {xingxingzuo}@zju.edu.cn

require motion of the platform. A particular challenge for most existing works is the calibration of under-actuated robots (e.g., unmanned ground vehicles (UGVs), autonomous driving cars), which as explored later in this work, presents a myriad of issues when trying to calibrate all parameters.

While the aforementioned works are all based on batch nonlinear least squares (NLS) optimization for *offline* calibration, there exists significant work focusing on *online* calibration based on Bayes filtering and on identifying degenerate motions which may cause sensing parameters to become unobservable. For instance, Eckenhoff et al. [10] developed a multi-state constraint Kalman filter (MSCKF) based multi-IMU multi-camera system with full online calibration, including camera intrinsics, IMU-camera spatial-temporal parameters, and IMU-IMU spatial-temporal parameters. Zuo et al. [3], [4] proposed a LiDAR-IMU-camera fusion framework with full online calibration and identified degenerate motion cases for LiDAR-IMU calibration when leveraging plane features. Lee et al. [2] introduced the visual-inertial wheel odometry (VIWO) which performs online spatial-temporal calibration between IMU-camera and IMU-wheel odometry, and wheel intrinsics, and examines the system observability as well. On the other hand, the authors of [16]–[18] have particularly studied degenerate motion for camera-IMU systems and IMU intrinsic calibration which has led to the identification of many conditions where calibration of all parameters is unlikely. Note that all these methods require high-fidelity initial guesses for calibrations, while this work assumes only some initial guesses available and seeks to find the globally optimal solution via batch NLS.

Specifically, in this work, we develop a general unified multi-sensor calibration algorithm within the NLS estimation framework to find optimal calibration of the camera intrinsics, spatial-temporal parameters for camera-IMU, LiDAR-IMU, and wheel-IMU asynchronous sensors:

- We develop a holistic multi-sensor calibration system leveraging high-rate inertial motion (termed iCalib) for asynchronous IMU, cameras, LiDAR and wheel odometry sensors, and show that under general motions the proposed system is able to accurately calibrate the spacial-temporal parameters between all the sensors.
- We develop two different interpolation schemes based on IMU kinematics or bounding poses to fuse heterogeneous measurements, thus enabling versatile spatial-temporal calibration.
- We perform extensive Monte-Carlo simulation to validate the proposed system and demonstrate the calibration degeneracy issues both in simulation and real-world experiments. The competing performance of the proposed system is shown by comparing to the current state-of-the-art toolboxes Kalibr [6] and MSG-CAL [8].

II. PROBLEM FORMULATION

A. State Vector

The state vector of our proposed inertial aided calibration (iCalib) system consists of the inertial navigation states \mathbf{x}_I , a set of environmental features \mathbf{x}_F , a set of sensors' spatial-

temporal and intrinsic calibration parameters \mathbf{x}_{calib} .

$$\mathbf{x} = \left[\mathbf{x}_{I_0}^\top \quad \dots \quad \mathbf{x}_{I_k}^\top \quad \mathbf{x}_F^\top \quad \mathbf{x}_{calib}^\top \right]^\top \quad (1)$$

$$\mathbf{x}_{I_k} = \left[{}^G \bar{q}^\top \quad {}^G \mathbf{p}_{I_k}^\top \quad {}^G \mathbf{v}_{I_k}^\top \quad \mathbf{b}_{\omega_k}^\top \quad \mathbf{b}_{a_k}^\top \right]^\top \quad (2)$$

$$\mathbf{x}_F = \left[{}^G \mathbf{p}_{f_1}^\top \quad \dots \quad {}^G \mathbf{p}_{f_m}^\top \quad {}^G \mathbf{p}_{\pi_1}^\top \quad \dots \quad {}^G \mathbf{p}_{\pi_l}^\top \right]^\top \quad (3)$$

$$\mathbf{x}_{calib} = \left[\mathbf{x}_{CI_1}^\top \quad \dots \quad \mathbf{x}_{CI_s}^\top \quad \mathbf{x}_{LI}^\top \quad \mathbf{x}_{WI}^\top \right]^\top \quad (4)$$

$$\mathbf{x}_{CI} = \left[{}^I_C \bar{q}^\top \quad {}^I_C \mathbf{p}_C^\top \quad t_{CI} \quad \zeta^\top \right]^\top \quad (5)$$

$$\mathbf{x}_{LI} = \left[{}^I_L \bar{q}^\top \quad {}^I_L \mathbf{p}_L^\top \quad t_{LI} \right]^\top \quad (6)$$

$$\mathbf{x}_{WI} = \left[{}^I_W \bar{q}^\top \quad {}^I_W \mathbf{p}_W^\top \quad t_{WI} \right]^\top \quad (7)$$

where ${}^G \bar{q}$ is the unit quaternion parameterizing the rotation $\mathbf{R}_{(I_k)}^G = {}^G \mathbf{R}$ from the IMU local frame $\{I_k\}$ to the global frame $\{G\}$ at time t_k [19], \mathbf{b}_ω and \mathbf{b}_a are the gyroscope and accelerometer biases, ${}^G \mathbf{v}_{I_k}$ and ${}^G \mathbf{p}_{I_k}$ are the velocity and position of the IMU, and \mathbf{p}_f and \mathbf{p}_π are point and plane features, respectively. The map of environmental landmarks \mathbf{x}_F contains m global 3D position \mathbf{p}_f for point features and l global closet-point (CP) \mathbf{p}_π for plane features [20], [21].

The calibration vector \mathbf{x}_{calib} contains the s number of camera-IMU calibration \mathbf{x}_{CI} , LiDAR-IMU spatial-temporal calibration \mathbf{x}_{LI} , and the wheel-IMU spatial-temporal calibration \mathbf{x}_{WI} . \mathbf{x}_{CI} consists of camera intrinsics (focal length, camera center, and distortion parameters) and the camera-IMU spatial-temporal parameters from each camera to IMU. Note that the current system can be readily extended to include multi-unsynchronized LiDARs. The time offset t_{SI} are defined as: $t_I = t_S + t_{SI}$. where t_I refers to IMU clock while t_S refers to sensor S 's clock with t_{SI} denoted as the time offset between these two timelines. Note that $S \in \{C, L, W\}$, denoting camera, LiDAR or wheel, respectively.

Inertial states, \mathbf{x}_{I_k} , can be cloned at an arbitrary frequency, but in practice we specify the camera with the highest frequency to be the “base” of the system. All measurements will be written as a function in respect to the true IMU time that each state is estimating (including the base of the system), and thus will need to handle this timeoffset between each sensor's measurement and this state time.

B. Kinematic-based Interpolation

Since our state, Eq. (1), contains inertial states only at specific times, we need to be able to write our asynchronous measurements as a function of these poses. We can related any time to the nearest inertial state by leveraging the IMU kinematic, ${}^G \mathbf{v}_I$ and ${}^I \boldsymbol{\omega}$, to directly compute the expected pose at our measurement time. For example, the measurement from sensor S at time $t_{S,in}$ requires getting the pose $\{{}_{S,in}^G \mathbf{R}, {}_{S,in}^G \mathbf{p}_{S,in}\}$. We can interpolate to this pose by first finding how much to interpolate by in the sensor clock frame $\delta t = t_{S,in} + t_{SI} - t_{I_k}$, where we have used the time offset between the measurement clock and the IMU clock frame which our inertial poses occur at. We can then define the following equations:

$${}_{S,in}^G \mathbf{R} = {}_{I_k}^G \mathbf{R} \text{Exp}({}^{I_k} \boldsymbol{\omega} \delta t) {}_{S}^I \mathbf{R} \quad (8)$$

$${}^G\mathbf{p}_{S_{in}} = {}^G\mathbf{p}_{I_k} + {}^G\mathbf{v}_{I_k}\delta t + {}^G_{S_{in}}\mathbf{R}^I\mathbf{p}_S \quad (9)$$

where $\mathbf{Exp}(\cdot)$ represents the exponential operation for $\text{SO}(3)$ [22]. It is important to note here that this interpolation both allows us to find the derivative in respect to both the temporal, t_{SI} , and spacial, $\{S_{in}\mathbf{R}, I\mathbf{p}_S\}$ parameters. This interpolation is only a good approximation when near the pose we are interpolating from, thus we only leverage this for pose measurements which occur near the inertial state time (e.g., our base camera sensor measurements).

C. Pose-based Interpolation

Another more conventional pose interpolation scheme is to find a pose between two bounding inertial states. We leverage this for LiDAR and non-base cameras which are more likely to have large amounts of interpolation. To find the sensor pose $\{S_{in}\mathbf{R}, G\mathbf{p}_{S_{in}}\}$ with the two bounding IMU states \mathbf{x}_{I_k} and $\mathbf{x}_{I_{k+1}}$ we can define the following:

$${}^G_{S_{in}}\mathbf{R} = {}^G_{I_k}\mathbf{R}\mathbf{Exp}(\lambda\mathbf{Log}({}^I_{I_{k+1}}\mathbf{R})){}^I_S\mathbf{R} \quad (10)$$

$${}^G\mathbf{p}_{S_{in}} = {}^G\mathbf{p}_{I_k} + {}^G_{I_k}\mathbf{R}(\lambda{}^I_{I_{k+1}}\mathbf{p}_{I_{k+1}}) + {}^G_{S_{in}}\mathbf{R}^I\mathbf{p}_S \quad (11)$$

$$\lambda = (t_{S,in} + t_{SI} - t_{I_k}) / (t_{I_{k+1}} - t_{I_k}) \quad (12)$$

where $t_{S,in}$ and t_{SI} are the measurement time and sensor time offset, $\{S_{in}\mathbf{R}, I\mathbf{p}_S\}$ represents the sensor rigid transformation to IMU frame, and $\mathbf{Log}(\cdot)$ represents the log operation for $\text{SO}(3)$ [22]. Same as before, this interpolation allows us to find derivatives in respect to these calibration parameters and thus perform calibration. This interpolation scheme is used for the majority of measurements which do not occur near the IMU pose time like the base camera.

D. Nonlinear Least Squares Optimization

We next overview the optimization problem we wish to solve and in the following section discuss the individual cost function definitions. Given sensor S measurements with additive white Gaussian noise, we have:

$$\mathbf{z}_S = \mathbf{h}_S(\mathbf{x}) + \mathbf{n}_S, \mathbf{n}_S \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_S) \quad (13)$$

Then, we can formulate the NLS as:

$$\min_{\mathbf{x}} \sum \|\mathbf{z}_S - \mathbf{h}_S(\mathbf{x})\|_{\mathbf{R}_S}^2 \quad (14)$$

An initial guess $\hat{\mathbf{x}}^\ominus$ is needed to start the optimization. After solving incremental state correction $\delta\mathbf{x}$, we can refine the state estimates by $\hat{\mathbf{x}}^\oplus = \hat{\mathbf{x}}^\ominus \boxplus \delta\mathbf{x}$, where \boxplus represents the state manifold update [22]. In summary, we have the following NLS equivalent to maximum likelihood estimation (MLE):

$$\min_{\mathbf{x}} \sum \mathbb{C}_I + \sum \mathbb{C}_C + \sum \mathbb{C}_L + \sum \mathbb{C}_W \quad (15)$$

Solving this problem via iterative algorithms results in the optimal IMU states, the visual point features, LiDAR plane features and all the calibration parameters needed.

III. INERTIAL AIDED MULTI-SENSOR CALIBRATION

A. IMU Measurements

The IMU cost function is modeled after ACI^2 [23], where the IMU directly reads the angular velocity ω_m and linear acceleration \mathbf{a}_m and is described as:

$$\omega_m = I\omega + \mathbf{b}_g + \mathbf{n}_g, \mathbf{a}_m = I\mathbf{a} + \mathbf{b}_a + \mathbf{n}_a - I_G\mathbf{R}^G\mathbf{g} \quad (16)$$

where $G\mathbf{g} = [0 \ 0 \ -9.81]^\top$, and \mathbf{n}_g and \mathbf{n}_a are continuous-time Gaussian noises that contaminate the IMU readings. The IMU dynamic model can thus be defined as [19], [24]:

$$\begin{aligned} G\dot{\bar{q}} &= \frac{1}{2}\Omega(\omega)G\bar{q}, \quad G\dot{\mathbf{p}}_I = G\mathbf{v}_I \\ G\dot{\mathbf{v}}_I &= G_I\mathbf{R}^I\mathbf{a}, \quad \dot{\mathbf{b}}_g = \mathbf{n}_{wg}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{wa} \end{aligned} \quad (17)$$

where ω and \mathbf{a} denote the local angular velocity and linear acceleration, \mathbf{n}_{wg} and \mathbf{n}_{wa} are the white Gaussian noises driving the gyroscope and accelerometer biases. $\Omega(\omega) := \begin{bmatrix} 0 & -\omega^\top \\ \omega & -[\omega] \end{bmatrix}$ and $[\cdot]$ represents a skew symmetric matrix. Between two base camera time t_k and t_{k+1} we integrate the IMU measurements as:

$$\Delta\bar{q} \triangleq \mathbf{h}_q(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) = {}^I_k\bar{q} \otimes {}^G_{I_{k+1}}\bar{q}$$

$$\Delta\mathbf{p} \triangleq \mathbf{h}_p(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) = {}^G_{I_k}\mathbf{R}^\top ({}^G\mathbf{p}_{I_{k+1}} - {}^G\mathbf{p}_{I_k} - {}^G\mathbf{v}_{I_k}\Delta t - \frac{1}{2}{}^G\mathbf{g}\Delta t^2)$$

$$\Delta\mathbf{v} \triangleq \mathbf{h}_v(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) = {}^G_{I_k}\mathbf{R}^\top ({}^G\mathbf{v}_{I_{k+1}} - {}^G\mathbf{v}_{I_k} - {}^G\mathbf{g}\Delta t)$$

$$\Delta\mathbf{b}_g \triangleq \mathbf{b}_{g_{k+1}} - \mathbf{b}_{g_k}, \quad \Delta\mathbf{b}_a \triangleq \mathbf{b}_{a_{k+1}} - \mathbf{b}_{a_k}$$

where $\Delta t = t_{k+1} - t_k$. Hence, the IMU measurements following ACI^2 can be constructed as:

$$\underbrace{\begin{bmatrix} \Delta\bar{q} \\ \Delta\mathbf{p} \\ \Delta\mathbf{v} \\ \Delta\mathbf{b}_g \\ \Delta\mathbf{b}_a \end{bmatrix}}_{\mathbf{z}_{I_{k+1}}} \simeq \underbrace{\begin{bmatrix} \mathbf{h}_q(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) \otimes \bar{q}^{-1}(\mathbf{H}_{b_g}^q \Delta\mathbf{b}_{g_k}) \\ \mathbf{h}_p(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) - \mathbf{H}_{b_g}^p \Delta\mathbf{b}_{g_k} - \mathbf{H}_{b_a}^p \Delta\mathbf{b}_{a_k} \\ \mathbf{h}_v(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}}) - \mathbf{H}_{b_g}^v \Delta\mathbf{b}_{g_k} - \mathbf{H}_{b_a}^v \Delta\mathbf{b}_{a_k} \\ \mathbf{b}_{g_{k+1}} - \mathbf{b}_{g_k} \\ \mathbf{b}_{a_{k+1}} - \mathbf{b}_{a_k} \end{bmatrix}}_{\mathbf{h}_I(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}})} + \mathbf{n}_{I_{k+1}} \quad (18)$$

where $\mathbf{n}_{I_{k+1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{I_{k+1}})$ represents the integrated IMU noise [23], $\Delta\mathbf{b}_g$ and $\Delta\mathbf{b}_a$ are the bias linearization correction terms, and $\mathbf{z}_{I_{k+1}}$ is the integrated IMU measurements connecting state \mathbf{x}_{I_k} and state $\mathbf{x}_{I_{k+1}}$, and \mathbf{H}_y^z represents the Jacobians of state z to variable y . Finally, the corresponding IMU cost function can be written as:

$$\mathbb{C}_{I_{k+1}} \triangleq \|\mathbf{z}_{I_{k+1}} - \mathbf{h}_I(\mathbf{x}_{I_k}, \mathbf{x}_{I_{k+1}})\|_{\mathbf{Q}_{I_{k+1}}^{-1}}^2 \quad (19)$$

B. Visual Point Measurements

Assuming a visual feature that has been observed by a camera, we can write the visual-bearing measurements (i.e., pixel coordinates [1]) as the following:

$$\mathbf{z}_C = \mathbf{h}_d(\mathbf{h}_p(\mathbf{h}_t({}^G_{C_{in}}\mathbf{R}, {}^G\mathbf{p}_{C_{in}}, {}^G\mathbf{p}_f)), \zeta) + \mathbf{n}_C \quad (20)$$

$$\triangleq \mathbf{h}_C({}^G_{I_{in}}\mathbf{R}, {}^G\mathbf{p}_{I_{in}}, \mathbf{x}_{CI}, {}^G\mathbf{p}_f) + \mathbf{n}_C \quad (21)$$

where \mathbf{z}_C is the raw uv pixel coordinate, $\mathbf{n}_C \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_C)$ the raw pixel noise, ${}^G\mathbf{p}_f$ the estimated landmark position in $\{G\}$, and $\{I_{in}\mathbf{R}, G\mathbf{p}_{I_{in}}\}$ denotes the current interpolated pose in $\{G\}$. The measurement functions \mathbf{h}_d , \mathbf{h}_p and \mathbf{h}_t correspond to the intrinsic distortion, projection, and transformation functions and the corresponding measurement Jacobians can be computed through a simple chainrule. \mathbf{h}_C represents the combination of these functions \mathbf{h}_d , \mathbf{h}_p , and \mathbf{h}_t . We also note here that an additional chainrule needs to be taken in respect to the spacial and temporal sensor parameters that the interpolated poses $\{I_{in}\mathbf{R}, G\mathbf{p}_{I_{in}}\}$ contain. Hence, the visual point cost can be formulated as:

$$\mathbb{C}_C \triangleq \|\mathbf{z}_C - \mathbf{h}_C({}^G_{I_{in}}\mathbf{R}, {}^G\mathbf{p}_{I_{in}}, \mathbf{x}_{CI}, {}^G\mathbf{p}_f)\|_{\mathbf{Q}_C^{-1}}^2 \quad (22)$$

TABLE II: Simulation parameters used. Note that if there are multiple sensors they are separated with a backslash.

Parameter	Value	Parameter	Value
IMU Freq. (hz)	300	Max Cam Pts	100
Cam Freq. (hz)	20/10/10	Cam Time offset (ms)	10/-10/0
LiDAR Freq. (hz)	8	LiDAR Time offset (ms)	-10
Wheel Freq. (hz)	70	Wheel Time offset (ms)	0
LiDAR Rings	16	LiDAR Range Noise (m)	0.03
Pixel Proj. (px)	1	Wheel. White Noise	0.03

C. LiDAR Plane Measurements

In structural environments, there are quite a few plane features that can be extracted. Leveraging the closest point (CP) plane representation [20], [21] and the sliding window based plane feature extraction [4], we can construct the point-on-plane cost as:

$$\mathbf{z}_L \triangleq \mathbf{h}_L({}^G_{I_{in}} \mathbf{R}, {}^G \mathbf{p}_{I_{in}}, \mathbf{x}_{LI}, {}^G \mathbf{p}_\pi, \mathbf{n}_L) \quad (23)$$

$$= \frac{{}^{L_{in}} \mathbf{p}_\pi^\top} {\|{}^{L_{in}} \mathbf{p}_\pi\|} ({}^{L_{in}} \mathbf{p}_{fm} - \mathbf{n}_L) - \|{}^{L_{in}} \mathbf{p}_\pi\| \quad (24)$$

where ${}^{L_{in}} \mathbf{p}_{fm}$ represents the measured 3D LiDAR points on the plane ${}^G \mathbf{p}_\pi$ with $\mathbf{n}_L \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_L)$ denoting the LiDAR measurement noise. The CP plane in interpolated LiDAR frame $\{L_{in}\}$, ${}^{L_{in}} \mathbf{p}_\pi = {}^{L_{in}} d_\pi {}^{L_{in}} \mathbf{n}_\pi$, can be obtained as:

$$\begin{bmatrix} {}^{L_{in}} \mathbf{n}_\pi \\ {}^{L_{in}} d_\pi \end{bmatrix} = \begin{bmatrix} {}^{L_{in}} \mathbf{R} & \mathbf{0}_{3 \times 1} \\ -{}^G \mathbf{p}_{L_{in}}^\top & 1 \end{bmatrix} \begin{bmatrix} {}^G \mathbf{n}_\pi \\ {}^G d_\pi \end{bmatrix} \quad (25)$$

Hence, the cost function for LiDAR plane measurements can be written as:

$$\mathcal{C}_L \triangleq \|\mathbf{z}_L - \mathbf{h}_L({}^G_{I_{in}} \mathbf{R}, {}^G \mathbf{p}_{I_{in}}, \mathbf{x}_{LI}, {}^G \mathbf{p}_\pi, \mathbf{0})\|_{\mathbf{Q}_L}^2 \quad (26)$$

D. Wheel Odometer Measurements

We can additionally consider a ground vehicle equipped with a differential drive system, where we can integrate left and right wheels velocity readings from t_k to t_{k+1} and obtain a 2D relative pose measurement (yaw and x, y translation in wheel frame) as:

$$\begin{aligned} \mathbf{z}_W &\triangleq \mathbf{h}_W(\mathbf{x}_{I_{in,k}}, \mathbf{x}_{I_{in,k+1}}, \mathbf{x}_{WI}) + \mathbf{n}_W \\ &= \begin{bmatrix} \mathbf{e}_3^\top \mathbf{Log}({}_I^W \mathbf{R}_G^{I_{in,k+1}} \mathbf{R}_{I_{in,k}}^G \mathbf{R}_W^I) \\ \Lambda({}_I^W \mathbf{R}_G^{I_{in,k}} \mathbf{R}({}^G \mathbf{p}_{I_{in,k+1}} + {}^G_{I_{in,k+1}} \mathbf{R}^I \mathbf{p}_W - {}^G \mathbf{p}_{I_{in,k}}) + {}^W \mathbf{p}_I) \end{bmatrix} + \mathbf{n}_W \end{aligned} \quad (27)$$

where $\mathbf{n}_W \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_W)$ is constructed from the integration process [2], $\Lambda = [\mathbf{e}_1 \ \mathbf{e}_2]^\top$, \mathbf{e}_i is the i -th standard unit basis vector, and we have performed our integration between two interpolated poses which are a function of time. The cost function can be written as:

$$\mathcal{C}_W \triangleq \|\mathbf{z}_W - \mathbf{h}_W(\mathbf{x}_{I_{in,k}}, \mathbf{x}_{I_{in,k+1}}, \mathbf{x}_{WI})\|_{\mathbf{Q}_W}^2 \quad (28)$$

IV. MONTE-CARLO SIMULATIONS

To verify the proposed iCalib, extensive Monte-Carlo simulations are performed leveraging our previously-developed simulator [1], [2], [25] to simulate asynchronous IMU, 3D LiDAR, 2D wheel odometry, and cameras. Table II details the exact simulation parameters used to generate relatively smooth 3D trajectory in a structural room environment with plane and point features (see Fig. 1).

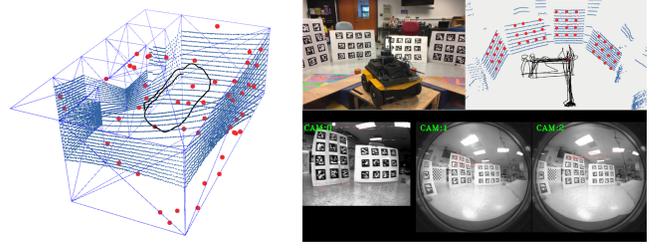


Fig. 1: Left: Simulated structure environment with planes. Visual feature points (red), LiDAR scan (blue) and trajectory (black) can be seen. Right: A ramp is built to excite 3D motion with AuRco tags are spread around as visual landmarks. Example camera images are shown.

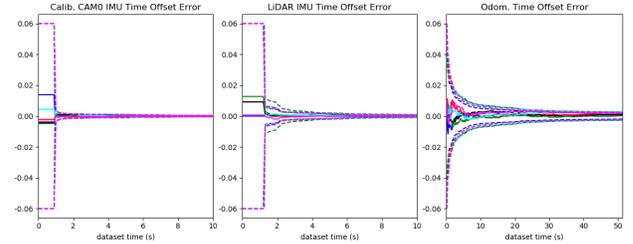


Fig. 2: Temporal calibration errors of IMU-CAM, IMU-LiDAR, IMU-Wheel and corresponding 3σ bound. Note that only the results from the base camera are presented here.

We consider a single IMU, three cameras, one LiDAR and wheel odometer, with the first camera acting as the base sensor frequency for inertial state creation, after which all measurement are constructed as optimized as in Section II-D. The calibration results from 6 Monte-Carlo runs for the proposed system with different random noises are presented in Figs. 2 and 3. For 3D motion trajectory shown in Fig. 2 and the top two rows of Fig. 3, the camera intrinsics, spatial-temporal calibration for IMU-camera, IMU-LiDAR, and IMU-wheel calibration parameters converged nicely demonstrating the ability of the proposed system to perform accurate spacial-temporal calibration between all sensors.

We next looked at how we are able to calibrate under planar motion which is common for ground vehicles. Previous works [2], [4], [16] have shown that planar motion will cause the translation along the rotation axis to be unobservable, as well as the roll and pitch of IMU-wheel calibration [2]. The results presented in the bottom row of Fig. 3 support this analysis. When we simulate 2D planer motion with the same estimator we are unable to calibrate the following: 1) the y component of IMU-camera translation (the rotation is along y axis in camera frame), 2) the z components of IMU-LiDAR, 3) IMU-wheel translation (the rotation is along z axis in LiDAR and wheel frame), and 4) the roll and pitch of IMU-wheel orientation. This verifies that in practical applications where planar datasets are only able to be collected for a robot, typical of ground vehicles, one should *not* expect calibration to be performed successfully. One can try to address this through using non-planar motion, as explored in the following real-world experiments, or through cross sensor constrains which we plan to explore in future works. It is important that while we show, for the first time, that these issues exist when performing calibration of *all* sensor parameters, they are still present when performing pair-wise

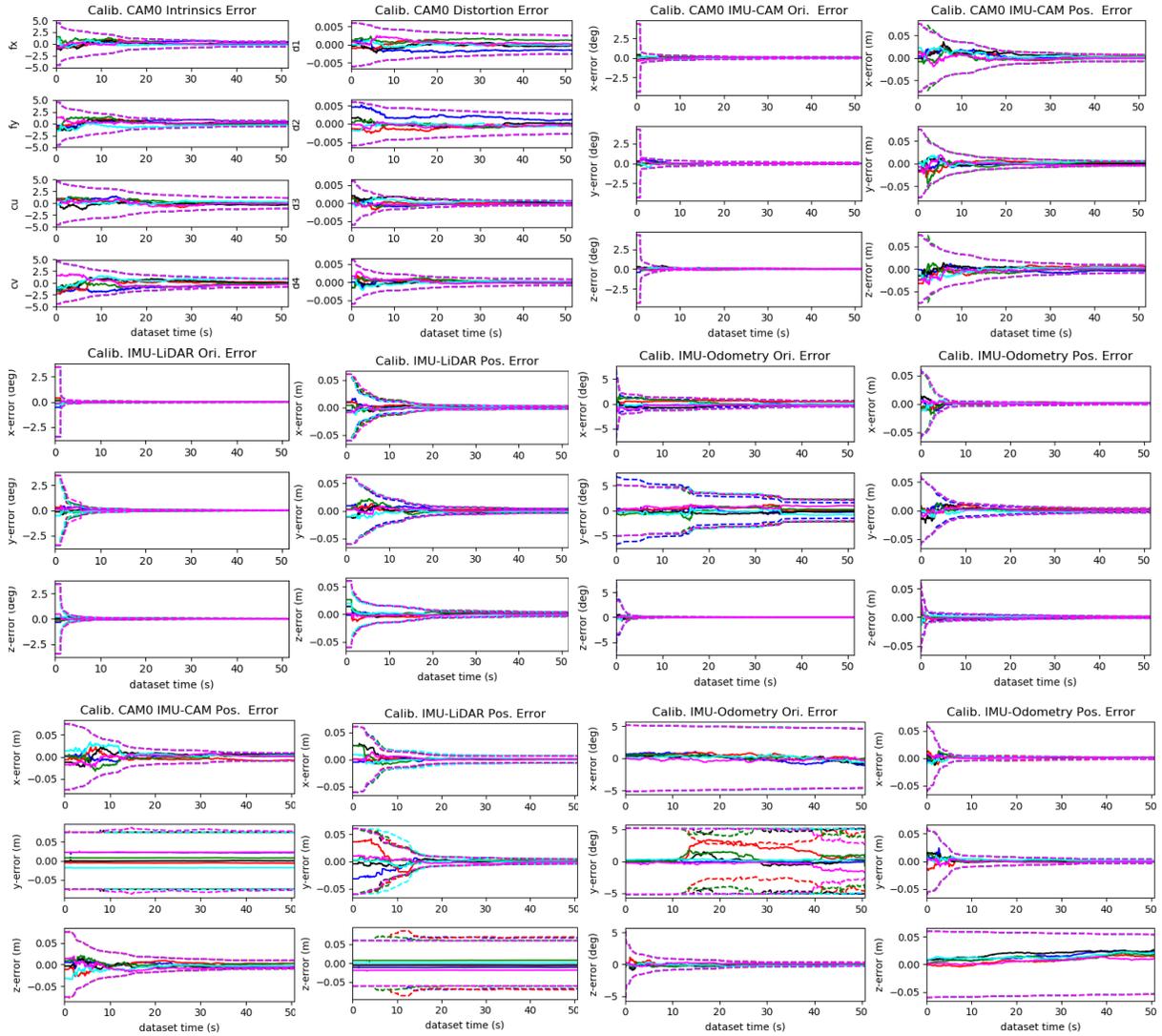


Fig. 3: Calibration errors for IMU-CAM, IMU-LiDAR, IMU-Wheel and corresponding 3σ bounds for six different Monte-Carlo runs. In order to save the space, only the results from base camera are presented. The bottom row of plots show the degenerate planar motion case.

calibration on planar datasets.

V. REAL-WORLD EXPERIMENTS

We leverage a Jackal robot from Clearpath¹ equipped with a FLIR blackfly camera, microstrain 3DMGX3-25 IMU, Intel Realsense T265 tracking camera and Velodyne VLP 16 as our experimental platform. We collected several datasets from the environment shown in Fig. 1. A set of experiments were first performed in a degenerate planar environment, but we found that this was extremely unstable system due to the introduction of unobservable calibration directions. We thus introduced a 15-20cm tall ramp which was driven over to try to excite more 3D motion of the Jackal. ArUco tags were used as visual landmarks that allowed for re-observation. The system was run until convergence with the initial time offset guesses being set to zero, thus minimizing interpolate errors of the base sensor. Three scenarios are tested to validate the proposed system: 1) IMU+Camera+LiDAR with

MSG-CAL [8] as the reference, 2) IMU+3Cameras with Kalibr [6] as the reference and 3) IMU+Camera+Wheel with manually measured reference values. Both MSG-CAL and Kalibr calibrations were performed in ideal conditions (e.g., Kalibr had full 6 DoF excitation) and thus shouldn't have unobservable calibration parameters.

A. IMU + 1 Camera + LiDAR

We average 4 calibration dataset results from MSG-CAL and use the average as reference. Shown in Fig. 4 (top right two column), we transfer the proposed system's converged CAM-IMU and LiDAR-IMU transformations to a relative CAM-LiDAR calibration to allow for direct comparison with MSG-CAL. To allow comparison, we compute the orientation quaternion error relative to a hand measured datum. It can be seen that there is less than 0.5 degrees between orientation errors and 1cm difference between translation with time offsets converging to stable values (bottom left two figures). Note that the z component 3σ is larger at

¹<https://clearpathrobotics.com/>

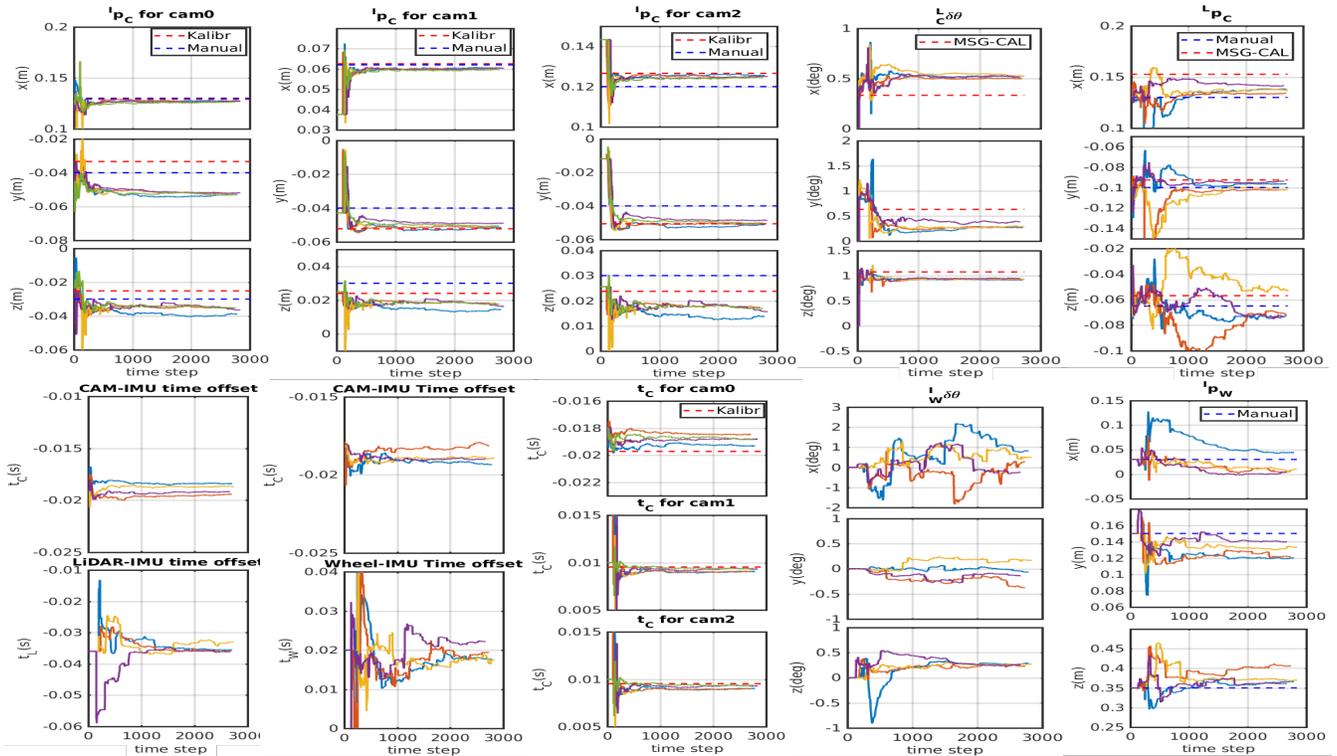


Fig. 4: The calibration results for the proposed system with IMU + 3CAMs, IMU+CAM+LiDAR and IMU+CAM+Wheel. Red dotted lines represent the values from existing calibration libraries (Kalibr or MSG-CAL) while the blue dotted line represent hand-measured values. Note that for each experiment, 6 bags are used and the best 4 results are presented.

$\pm 1.5\text{cm}$ and thus even though repeat runs converge with larger variance, they fall within this conservative bound. It can also be seen that the convergence of the z component is at a slower rate when compared to the x and y . This large variance and slow convergence indicates that even though the robot is driven over a ramp, to induce non-planar motion in the z direction, there is limited information gain and thus the calibration of the z component still suffers from unobservable issues. While the small ramp helped with stability during solving, these results show that it was unable to provide an informative 3D platform motion trajectory, showing the limitation of calibration for ground vehicles.

B. IMU + 3 Cameras

Shown in Fig. 4 (top left three columns), while all three camera extrinsics are able to converge to the hand measured and Kalibr references, the z component is still worst compared to the x and y directions. The time offsets, Fig. 4 (bottom middle column), are able to quickly converge and orientations (which are omitted for space) differences are very small at 0.5 degrees. Same as in the first real-world experiment, the difference in the z component is likely due to insufficient platform motion thus confirming limitations in performing calibration.

C. IMU + 1 Camera + Wheel Odometer

Shown in the right bottom of Fig. 4 (left bottom second column), the time offset of base camera converges and wheel odometer stabilizes to a stable value. The yaw of the IMU-wheel rotation, Fig. 4 (right bottom two columns), converge within 0.2 degrees of the manually measured value, while

the x and y translation components converge with differences of up to 2cm. However, the convergence of the roll, pitch, and z translation of the IMU-wheel calibration are extremely unstable and inaccurate due to both the combination of unobservability and information lost due to the 2D pose projection in Eq. (27). This shows that practical calibration of wheel odometry extrinsics is unlikely and unstable without the addition of cross-sensor constraints which we plan to explore in future works.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an NLS-based inertial aided multi-sensor calibration framework (iCalib), which can calibrate camera intrinsics refinement and spatial-temporal parameters for IMU, camera, LiDAR, and wheel odometry. An IMU kinematics and relative pose based interpolation schemes are leveraged to enable time offset estimation between all sensors. We have performed extensive simulations with 3D trajectories to validate the proposed iCalib and we also study the degeneracy for calibration when the robot undergoes 2D planar motion. We have also tested the proposed system on a real-world UGV dataset. The calibration results are compared to the state-of-the-art calibration toolboxes Kalibr and MSG-CAL. We have shown that there exist significant challenges when calibrating under constrained motions, which limits its widespread deployment in practice and demands more future efforts. For example, in the future we will explore cross-sensor constraints (e.g., visual point on LiDAR plane, and visual points on ground) to address the stability of constrained motion calibration.

REFERENCES

- [1] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins
- [2] W. Lee, K. Eickenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-inertial-wheel odometry with online calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [3] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: Lidar-inertial-camera odometry," Macau, China, Nov. 2019, (accepted).
- [4] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," *arXiv preprint arXiv:2008.07196*, 2020.
- [5] Z. Zhang, A. Suleiman, L. Carlone, V. Sze, and S. Karaman, "Visual-inertial odometry on chip: An algorithm-and-hardware co-design approach," in *Robotics: Science and Systems*, 2017.
- [6] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1280–1286.
- [7] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1793–1800.
- [8] J. L. Owens, P. R. Osteen, and K. Daniilidis, "Msg-cal: Multi-sensor graph-based calibration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3660–3667.
- [9] J. Lv, J. Xu, K. Hu, Y. Liu, and X. Zuo, "Targetless calibration of lidar-imu system based on continuous-time batch estimation," 2020.
- [10] K. Eickenhoff, P. Geneva, and G. Huang, "Mimc-vins: A versatile and resilient multi-imu multi-camera visual-inertial navigation system," *arXiv preprint arXiv:2006.15699*, 2020.
- [11] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3936–3943.
- [12] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *Proceedings of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, October 2018.
- [13] M. Nowicki, "Spatiotemporal calibration of camera and 3d laser scanner," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6451–6458, 2020.
- [14] S. Mishra, P. R. Osteen, G. Pandey, and S. Saripalli, "Experimental evaluation of 3d-lidar camera extrinsic calibration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [15] W. Lee, C. Won, and J. a. Lim, "Unified calibration for multi-camera multi-lidar systems using a single checkerboard," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [16] Y. Yang, P. Geneva, K. Eickenhoff, and G. Huang, "Degenerate motion analysis for aided INS with online spatial and temporal calibration," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 2, pp. 2070–2077, 2019.
- [17] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online imu intrinsic calibration: Is it necessary?" in *Proc. of Robotics: Science and Systems (RSS)*, Corvallis, Or, 2020.
- [18] T. Schneider, M. Li, C. Cadena, J. Nieto, and R. Siegwart, "Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3846–3860, 2019.
- [19] J. Solà, "Quaternion kinematics for the error-state kalman filter," 2017.
- [20] P. Geneva, K. Eickenhoff, Y. Yang, and G. Huang, "LIPS: Lidar-inertial 3d plane slam," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, Oct. 1-5, 2018.
- [21] Y. Yang and G. Huang, "Observability analysis of aided ins with heterogeneous features of points, lines and planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 399–418, Dec. 2019.
- [22] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.
- [23] Y. Yang, B. P. W. Babu, C. Chen, G. Huang, and L. Ren, "Analytic combined imu integrator for visual-inertial navigation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.
- [24] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.
- [25] P. Geneva, N. Merrill, Y. Yang, C. Chen, W. Lee, , and G. Huang, "Versatile 3d multi-sensor fusion for lightweight 2d localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.