Sensor Model Identification via Simultaneous Model Selection and State Variable Determination

Christian Brommer[1], Alessandro Fornasier[1], Jan Steinbrener[1], Stephan Weiss[1]

Abstract—We present a method for the unattended gray-box identification of sensor models commonly used by localization algorithms in the field of robotics. The objective is to determine the most likely sensor model for a time series of unknown measurement data, given an extendable catalog of predefined sensor models. Sensor model definitions may require states for rigid-body calibrations and dedicated reference frames to replicate a measurement based on the robot's localization state. A health metric is introduced, which verifies the outcome of the selection process in order to detect false positives and facilitate reliable decision-making. In a second stage, an initial guess for identified calibration states is generated, and the necessity of sensor world reference frames is evaluated. The identified sensor model with its parameter information is then used to parameterize and initialize a state estimation application, thus ensuring a more accurate and robust integration of new sensor elements. This method is helpful for inexperienced users who want to identify the source and type of a measurement, sensor calibrations, or sensor reference frames. It will also be important in the field of modular multi-agent scenarios and modularized robotic platforms that are augmented by sensor modalities during runtime. Overall, this work aims to provide a simplified integration of sensor modalities to downstream applications and circumvent common pitfalls in the usage and development of localization approaches.

Index Terms—Sensor Fusion, State-Estimation, Automated Sensor Integration, Modularity, Autonomous Navigation

I. INTRODUCTION

Accurate and robust localization is a crucial prerequisite for any application of autonomous robotics. Without accurate localization information, the generation of control inputs will not be possible. Localization algorithms in research and consumer products heavily rely on a number of sensor types to infer the most accurate position and orientation possible. In addition to position and orientation, state-of-the-art robot localization

[1] Control of Networked Systems Group, University of Klagenfurt, Austria. E-Mail: christian.brommer@aau.at, { alessandro.fornasier, jan.steinbrener, stephan.weiss }@ieee.org
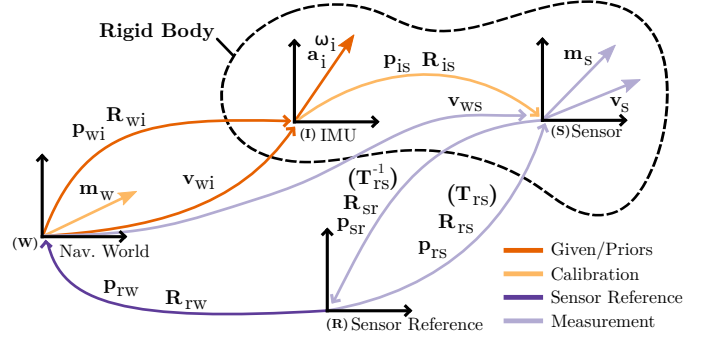Corresponding author: Christian Brommer

Fig. 1: Generalized setup of transformations to express various sensor modalities such as vectors for the magnetic field or the velocity and transformations for 3 and 6-DoF sensor measurements and calibrations. This work aims to process a gray-box sensor signal together with a reliable system state to identify a corresponding sensor model and its properties.

approaches such as Extended Kalman Filters (EKFs) can also estimate additional states such as sensor calibrations (e.g., bias offsets or extrinsic calibration parameters) online. For this reason, essential core states (position, velocity, orientation, etc.) and sensor calibration states $\mathbf{x}$ are incorporated in the sensor model $h(\mathbf{x})$ and the Jacobian for sensor measurement updates $\mathbf{H}(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}$.

Figure 1 shows the essential configuration of the sensor models that are addressed by the presented approach. A full definition of all sensor models is shown in Section IV. The integration of such sensors can be a complex task, and typically, a number of questions need to be answered prior to the integration:

- What type of information does the sensor provide?
- Given a rigid body, what is the extrinsic calibration?
- Is the measurement given in the normal direction ($\mathbf{T}_{sr}$) or the inverse with respect to the robot ($\mathbf{T}_{sr}^{-1}$) (see Fig. 1)?
- Does the measurement model require a reference frame?

The integration of a sensor modality usually requires expert knowledge. In order to generate a sensor model for the update in filter definitions, the main components of the general system, such as given and prior states, calibration states, and reference frames, need to be identified.

To provide additional and quantifiable insight into the magnitude of complications related to the setup of localization and state-estimation solutions, we evaluated the number of relevant issues, discussions, and wikis for the open-source platform GitHub[1] with respect to sensor-modeling and state-estimation.

The keyword "Sensor model" appears in 80k issues, 4k discussions, and 13k wikis, with a significant share of 14.4k issues and 1k discussions related to "sensor model integration". Similarly, "State-Estimation" (30k issues), "sensor-fusion" (8.5k issues), and "sensor calibration" (22k issues). This indicates that setting up localization solutions and integrating sensor modalities, 1) requires extensive documentation - to be read, and 2) is not always straightforward and causes predominant and time-consuming issues. Such issues even motivate setups where raw sensor data or entirely proprietary solutions are preferred for ease of use. The proposed work aims to allow a simplified and automated integration of sensor modules during runtime without prior knowledge about the sensor type or its properties to streamline and robustify the task of integrating a new sensor model. This mitigates possible errors and accelerates the research and development phases.

In previous work [1], we focused on the modularity aspect of a possible application that integrates a sensor modality during runtime. The important aspect is that a sensor module does not need to be *a priori* known to the system. However, this approach requires knowledge of the sensor type and, ideally, an initial guess for the additional calibration states of a sensor module.

Of course, a pragmatic approach is to always know the sensor modalities that are used during the operation of a robot, or having a self-identifying sensor. However, we aim to simplify the process of sensor integration without expert knowledge and future work that does not rely on knowing a sensor type before integration. We illustrate two scenarios where the proposed method could be applied:

a) Automated Robust Sensor Integration: Sensor devices are not always well documented nor easily integrated by an engineer without prior knowledge about state-estimation algorithms and specific sensor modalities. Assuming an inexperienced person needs to integrate the odometry information provided by an Intel Real-Sense sensor, which is rigidly mounted on a UAV.

First, we need to determine the type and direction of the measurements. In this case, the position and rotation measurements could be expressed w.r.t. an arbitrary vision frame chosen by the sensor up on initialization, its inverse, or different directions for the position and rotation information. Given a lack of documentation, arbitrary reference frames, or inexperience, finding the right references and directions can result in time-consuming tabletop experiments.

The next step is to determine the calibration states and reference frame of the sensor. We assume that the UAV uses an IMU as a core sensor, and we need to find the rigid-body calibration of the Real-Sense w.r.t. the IMU frame. Lack of knowledge on how the IMU is mounted, how the odometry frame of the Real-Sense is oriented, and which arbitrary reference frame is used by the Real-Sense can lead to errors when inferring this calibration through estimation by hand or evaluation of numerical measurements of the IMU and the Real-Sense.

We also need to know the reference in which the measurement is expressed and if this differs from the currently used world reference of the localization algorithm. If the reference frame does not differ, then it does not need to be incorporated into the sensor model. In the case of the Real-Sense, the odometry reference is most likely aligned with the gravity vector, but the remaining orientation parameters and the position depend on the pose of the sensor during internal initialization.

These elements can lead to time-consuming integration and error-prone results. With the proposed method, the sensor type and parameters for its setup are inferred automatically without the need of expert knowledge, thus simplifying the system integration.

b) Collaborative State Estimation (CSE): To the contrary of the previous example, for CSE, localization information of one robot is not only given by rigidly attached sensor modalities. Localization information can also be inferred through temporary encounters with other moving robotic platforms, sharing their state estimates in the vicinity.

One possible use-case is a setup with an Unmanned Aerial Vehicle (UAV) and rover, providing a recharging base for the UAV. While charging, the UAV is temporarily attached to the moving rover and can either share its state estimate or individual sensor modalities with the rover. In this case, sensor modalities, respective calibration states, and references do not need to be known a priory. Gray-box sensor information can be transmitted to the rover, which identifies the sensor module, type of measurement, and sensor calibrations. During this encounter, both platforms can benefit and improve their estimated localization.

This example can be transferred to a more tangible industry use case, given a scenario with an autonomous pallet truck that picks up boxes with individual Radio Frequency Identifications (RFIDs) modules that can be located within the warehouse. Given the presented method and a short measurement sequence, the localization algorithm of the pallet truck can incorporate the measurement automatically without further knowledge, and benefit from the additional position measurements, improving accuracy and redundancy.

Of course, a more practical approach is to pre-process and unify the output of collaborating robots and to use this information directly. However, if a robot carries more than one sensor, then sharing individual sensor information and incorporating them with accurately modeled extrinsic calibration states in a tightly coupled fashion can improve robustness, accuracy, and consistency, given that probabilistic correlations are estimated accurately. Depending on the overall complexity of such a modular system, a self-identification approach to sensor models can also be more scalable. In any case, it is helpful if the correct sensor model, and its parameters (i.e., calibrations and references) are inferred automatically, based on the given measurement signal, following the presented method outlined by Figure 2 to efficiently integrate sensor information online into a localization algorithm.

The presented work focuses on the following contributions:
- Gray-Box sensor model identification based on noisy estimates of localization states and sensor measurement signals from an unknown sensor type. The measurement source is unknown, and a catalog of the most common sensor models in the field of robotics is provided.
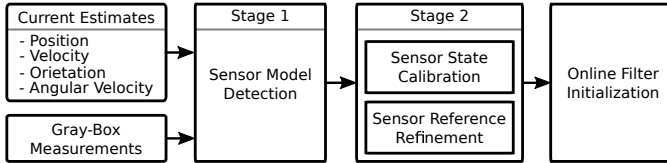- Estimation of sensor calibration states for filter initialization.

Fig. 2: Outline of the stages for the sensor model identification method.

- Evaluation of the requirement for a dedicated sensor measurement reference frame, needed to align the sensor information with the navigation frame of a localization algorithm.
- A health metric for the validation of a selected sensor module, preventing false positives.

Essential steps of the proposed sensor identification method are illustrated in Figure 2 and further detailed in Section IV. The general structure of the method is as follows. The localization system needs to be in a self-sustained state, meaning a global or stable relative localization needs to be available. In the presented examples, this information is provided by an EKF with relatively noise core states (i.e., an IMU-driven propagation with vision sensor updates). Stage 1 is responsible for the detection of the correct sensor model and the correct sensor signal references, e.g., is the sensor measurement expressed w.r.t. the world frame, or is the measurement expressed by the inverse? This is done based on an over-determined system definition and provides a health metric to prevent false positive selections. Stage 2 performs the estimation of the calibration states for the selected sensor model and identifies if a sensor reference frame, different from the world reference frame, is needed. After the sensor model and the existence of additional states are known, the correct Jacobian for the integration of an updated sensor to a filter framework is selected.

In the final step, the sensor model and the parameterization of the initial calibration states are integrated as an additional source of information to a state estimation framework. The sensor self-calibration of the previously presented framework ensures the refinement of the calibration states in the long term. A complete integration of this process was tested with simulated data in Section V-A and with real-world data and two robotic platforms: A UAV use case (see Sec. V-D) and a ground vehicle use case (see Sec. V-E).

II. RELATED WORK

The presented work generally concerns the research on model identification based on given sensor data and partially known system models. A number of methods have been developed and address solutions to this problem in various ways.

A pragmatic approach was summarized by Mitterer [2], where a sensor contains an electronic data sheet, accessible to a host, detailing the properties of a sensor. While this is a great solution for higher-level applications, fundamental non-smart sensors do not provide this functionality, and the determination of an extrinsic calibration still requires a dedicated process.

Analytical methods have been introduced, starting with Fuzzy Modeling. Nagai [3] and Frolik [4] describe the generation of a fuzzy logic sensor model for a number of inputs and a corresponding output, additionally exposing information from possible latent features. Nagai [3] uses uncategorized input information to generate a refined model, tailored to a specific output, such as to predict variables in a chemical process where direct measurements are not possible. The approach is especially interesting because it allows different criteria on the variable sets, such as Gaussian distributions, and the clustering of data points (Fuzzy c-means) to coherent structures by simplification with Kohonen maps. The clustering achieves a simplified set of sensor states and promotes resilience to over-fitting.

However, as mentioned in the introduction, the goal is to generate a model that can be used in a filter setup with online self-calibration. Since this generated model does not provide a direct association with the physical states of a sensor model, an abstract Fuzzy model will not allow human interpretation of physical calibration states and would provide a solution with untraceable states.

A more traceable solution for identifying states or coefficients and their significance was introduced by Tibshirani [5] with the Least Absolute Shrinkage and Selection Operator (LASSO) and later adaptation Least Angle Regression (LARS) by Efron [6]. LASSO poses a regularization in the form of the L1-Norm on a selection of coefficients to an optimization problem, thus promoting sparsity of coefficients. A selection is made by eliminating coefficients with insignificant contributions to the overall solution. To utilize this approach, an over-determined system must be defined and evaluated against a given stream of e.g. sensor signals. This approach was not designed to identify individual states of higher dimensions, and each coefficient is evaluated individually, without definable coherence between states. This renders the usage of the LASSO approach difficult for usage in multi-dimensional geometric problems. It is not guaranteed that a non-required vector in $\mathbb{R}^3$ is eliminated entirely, especially if the signal has noise, allowing for possible over-fitting.

The LASSO operator is one of the main techniques used by the Sparse Identification of Non-linear Dynamics (SINDy) framework by Brunton [7]. SINDy was used for the identification of time-varying physical problems and demonstrated with fluid dynamical problems [8]. It requires a time series of data and over-determined model expressions that are commonly present for the problems that are to be analyzed. This approach was extended by Mangan [9] to incorporate information criteria (Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)) for model verification and to find the most informative model through sequential thresholding. Mangan [9] also discussed the problem that noise-afflicted data could prevent the identification of the correct model with SINDy and the possible requirement of pre-filtered data.

Zheng [10] also extended the work on SINDy by introducing the Sparse Relaxed Regularized Regression (SR3) Framework. This extension introduced a relaxation loss term, which aims to increase the robustness of the state sparsification if inaccurate or noise-afflicted data is given. The sparsification is not performed on the coefficients $\mathbf{x}$ directly but via a second loss term $\mathbf{w}$ which allows the relaxation $\min_{x,w} \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \|\mathbf{w}\|_1 + \|\mathbf{x} - \mathbf{w}\|_2^2$ with system

definition $\mathbf{A}$ and input $\mathbf{b}$. While SINDy can solve multi-dimensional problems, it was not directly designed to incorporate multi-dimensional states such as geometrical transformations.

Other approaches for model identification perform iterative optimization steps. Mixed integer non-linear programming is one technique; it also requires a predefined and meaningful catalog of model components and estimates coefficients and boolean-like or integer variables using a branch and bound technique. Integer variables are selected iteratively (branch), and the value range is limited to a specific range (bound). Variables that do not minimize the problem in the value range are reversed, and the next branch of variable thresholding is followed.

This approach is similar to Bayesian networks, which also utilize a branching technique, with the difference that individual states are added without a decision variable as a weight, and the choice of the addition or existence of a state is made based on the improvement of the system error. Kadane [11] and Khosbayar [12] describe multiple approaches to model selection by utilizing the Bayesian approach. Compared to the optimization-based methods, which require penalties that promote sparsification, e.g., regularization through LASSO and LARS or multiplicative mixed integers, the Bayesian approach includes or rejects model states for each iteration. This results in a similar methodology of sparsifying a sensor model while maintaining accuracy.

A drawback with branching techniques is that the order in which states are added and evaluated can lead to non-deterministic and possibly wrong results depending on the nature of the proposed system models. An additional state that improves the system's accuracy in first iterations does not necessarily belong to the correct sensor model. Even with forward selection and backward elimination methods, the final model selection might minimize the error but remain incorrect. Reasons for false detections can be similarities of the sensor models (see Eq. (22) and (23)) or differences in magnitudes of individual state variables (see Sec. IV). Thus, an incorrect first choice and the addition of a state can be irreversible.

A brute-force method for model identification through Boolean decision-making would be to iterate through a binary table for all posed states one by one. However, depending on the number of possible states, this approach can be too time-consuming/inefficient. Lesage [13] uses a probabilistic approach with randomly drawn Bernoulli samples to find a suitable selection more efficiently, while Xiong [14] updates Boolean definitions based on a thresholding technique and validates decisions based on the iterative system error.

In summary, Fuzzy logic finds abstract states that correlate given input data to output data. However, it does not provide an association to physically meaningful states, which prevents online state estimation in subsequent stages. LASSO regularizes the collective magnitude of a state vector and reduces the number of states (selection) for the solution of a problem. However, it does not allow structured group penalties of multi-dimensional variables, making the selection process inaccurate for geometrical problems. SINDy finds a polynomial for a dynamic system model, and variations to

this approach address further improvements to noise resilience; however, incorporating states that perform multi-dimensional geometrical transformations is not straightforward.

Furthermore, approaches with iterative refinement stages divide the solution to model selection into two stages, the state selection and the state estimation in consecutive steps, where the state selection can change per iteration after state values have been refined. Such approaches can be time-consuming and lead to incorrect state selections if the posed model states are similar and prone to over-fitting in the presence of noise.

To the best of our knowledge, no existing approach or framework addresses the identification of multi-dimensional, geometrically constrained sensor models, nor the detection of false positives as presented by this work. Our approach contributes to the existing literature, by allowing for a direct selection of interpretable sensor models that can incorporate multi-dimensional geometric transformations and a system definition for the optimization of selection operators and refinement of state variables simultaneously. It also establishes a health metric for the identification of false positive selections, and resilience to signal scaling and noise.

III. NOTATION

For a generalized description of sensor models, we are defining frame dependencies as shown by Figure 1. Following the notation $_{A}\mathbf{p}_{BC}$ for a translation of frame C with respect to frame B expressed in frame A. With $_{A}\mathbf{p}_{BC} \equiv \mathbf{p}_{BC}$ if frame $\mathsf{A} = \mathsf{B}$. The quaternion $\mathbf{q}_{AB}$ describes the rotation of frame B with respect to frame A and corresponds to a rotation matrix $\mathbf{R}_{(\mathbf{q}_{AB})} \equiv \mathbf{R}_{AB}$. A variable describing a time sequence is denoted by the prefix t, e.g., $^{t}\mathbf{v}$. Since the transformations in Figure 1 are used for multiple sensors, such states are labeled with the sensor type $^{type}\mathbf{x}$, e.g. $^{P}\mathbf{p}_{is}$ for the translation of the sensor frame w.r.t. the IMU frame used for the position sensor (P) (see labels in Tab. I).

Rotations utilized within the system definition are defined as Rotation matrices in $SO(3)$. Rotations as states which are optimized, or measurements, are defined in the tangent space with conversions $\log(\cdot)^{\vee} : SO(3) \rightarrow \mathbb{R}^{3}$ and $\exp(\boldsymbol{\omega}^{\wedge}) : \mathbb{R}^{3} \rightarrow SO(3)$ with $(\cdot)^{\wedge} \equiv \lfloor \cdot \rfloor_{\times}$ and $(\cdot)^{\vee}$ as the structural decomposition of the skew-symmetric structure. This definition ensures the correct properties of rotations during the optimization process.

IV. METHOD

This section describes how an over-determined system model is defined, which aspects are critical for selection operators, and which constraints are important to render the presented method reliable. As illustrated by Figure 2, the proposed method requires a given sequence of associated (synchronized) data points providing the current state estimate and the measurement by the unknown sensor module as the input. The two data streams can be acquired, e.g., during the operation of the robot and after a sensor was added for online integration, or in a handheld setting for offline processing. After a data sequence was acquired, the data points of the two data streams need to be associated, e.g., based on timestamps.

If a data point of one data stream has no valid association, then this point is discarded. While the data needs to be correctly associated, it does not have to be given in a causal time sequence. After the data points are synchronized, the recorded sequence is fixed and processed in Stage 1.

The known inputs are the essential states required to describe the location of a robot, commonly called *core states*. For this work, the core states are defined by the position $\mathbf{p}_{wi}$, rotation $\mathbf{R}_{(\mathbf{q}_{wi})}$, and velocity $\mathbf{v}_{wi}$ of an IMU-frame i with respect to the navigation world-frame W. Because we are including velocity sensor models, we also require the angular velocity information from the IMU $\boldsymbol{\omega}_i$. This sequence of core states

$$^t\mathbf{x}_c = \begin{bmatrix} ^t\mathbf{p}_{wi}^\mathsf{T} & ^t\mathbf{v}_{wi}^\mathsf{T} & ^t\mathbf{q}_{wi}^\mathsf{T} & ^t\boldsymbol{\omega}_i^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{13 \times n} \qquad (1)$$

is provided online by an IMU-driven EKF filter framework such as presented by [1]. In scenarios where an IMU is not available but data such as angular velocity is required by a sensor model, this information would need to be estimated using available sensor data containing rotational information and e.g. numerical differentiation.

Stage 1, detailed in Section IV-A, performs the main model identification based on a fixed given data sequence and a generalized model definition that adheres to the reference frame and state structure shown by Figure 1. According to this structure, a sensor can have an extrinsic calibration state describing a transformation of the sensor frame w.r.t. the IMU frame ($\mathbf{p}_{is}$, and $\mathbf{R}_{is}$). The local deviation of the magnetic field is described as a Cartesian vector $\mathbf{m}_w$ which does not need to align with any axis of the world frame and depends on the current geolocation [15].

A dedicated sensor reference frame R is introduced because the information provided by a sensor entity does not need to align with the current global navigation world frame W of the estimator, thus requiring transformations $\mathbf{p}_{rw}$ and $\mathbf{R}_{rw}$. An example is given by a vision sensor that is initialized at a later point of a vehicle operation, and its reference is defined relative to the current location.

Sensor measurements can either be expressed w.r.t. the sensor frame S itself, such as the magnetic field $\mathbf{m}_s$ and the body velocity $\mathbf{v}_s$, or as position $\mathbf{p}_{rs}$ and rotation $\mathbf{R}_{rs}$ of the sensor frame w.r.t. the sensor reference frame R, or as the inverse; the sensor reference frame w.r.t. the sensor frame $\mathbf{p}_{sr}$, $\mathbf{R}_{sr}$. A special case is presented if the sensor measurement is the inverse $\mathbf{T}_{rs}^{-1}$ (see Fig. 1). In that case, the rotation $\mathbf{R}_{rw}$ is a free parameter and not constrained by the sensor reference frame S nor the navigation world frame W. Thus, the transformations $\mathbf{p}_{rw}$ and $\mathbf{R}_{rw}$ are co-dependent and infinite valid solutions exist as long as $\mathbf{p}_{rw}$ is chosen according to $\mathbf{R}_{rw}$ to remain coherent. Stage 2, described in Section IV-B, performs refinements and final choices on the calibration states after the general sensor model was chosen.

A. Sensor Identification

The goal of this work is to identify the following sensor models commonly used in robot localization tasks. We use the notation of rotation matrices to express their state variables expressed in the tangent space e.g. $\exp(\boldsymbol{\omega}_{wi}^\wedge) = \mathbf{R}_{wi}$, to provide a concise description of the sensor models. While the symbolic expressions for sensor states are the same for each sensor model in Figure 1, each model has a dedicated state variable for the selection process according to Table I. Only the sensor reference frame is defined as a common, shared, optimization state to provide an additional constraint and to restrain incorrect model selections while the reference frame converges. Model states are given by

$$\mathbf{x}_m = [\mathbf{p}_{rw}^\mathsf{T} \; \boldsymbol{\omega}_{rw}^\mathsf{T} \; {}^p\mathbf{p}_{is}^\mathsf{T} \; {}^{ip}\mathbf{p}_{is}^\mathsf{T} \; {}^{ip}\boldsymbol{\omega}_{is}^\mathsf{T} \; {}^r\boldsymbol{\omega}_{is}^\mathsf{T}$$
$$\qquad {}^{ir}\boldsymbol{\omega}_{is}^\mathsf{T} \; {}^{vel}\mathbf{p}_{is}^\mathsf{T} \; {}^{bv}\mathbf{p}_{is}^\mathsf{T} \; {}^{bv}\boldsymbol{\omega}_{is}^\mathsf{T} \; {}^{mag}\boldsymbol{\omega}_{is}^\mathsf{T} \; \mathbf{m}_w^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{36}. \quad (2)$$

Let $\mathbf{x} = \begin{bmatrix} ^t\mathbf{x}_c^\mathsf{T} & \mathbf{x}_m^\mathsf{T} \end{bmatrix}^\mathsf{T}$ be the full state vector, containing the time sequence of the core states $^t\mathbf{x}_c$ and the self-calibration states of all proposed sensor models $^t\mathbf{x}_m$ for the following sensor model definitions:

Position
$$\mathbf{p}_{rs}(\mathbf{x}) = \mathbf{p}_{rw} + \mathbf{R}_{rw} \left(^t\mathbf{p}_{wi} + {}^t\mathbf{R}_{wi} \; {}^p\mathbf{p}_{is} \right) \qquad (3)$$
Inverse Position
$$\mathbf{p}_{sr}(\mathbf{x}) = -{}^{ip}\mathbf{R}_{is}^\mathsf{T} \; \left(^t\mathbf{R}_{wi}^\mathsf{T} \left(\mathbf{R}_{rw}^\mathsf{T} \; \mathbf{p}_{rw} + {}^t\mathbf{p}_{wi} \right) + {}^{ip}\mathbf{p}_{is} \right) \qquad (4)$$
Rotation
$$\mathbf{R}_{rs}(\mathbf{x}) = \mathbf{R}_{rw} \; {}^t\mathbf{R}_{wi} \; {}^r\mathbf{R}_{is} \qquad (5)$$
Inverse Rotation
$$\mathbf{R}_{sr}(\mathbf{x}) = {}^{ir}\mathbf{R}_{is}^\mathsf{T} \; {}^t\mathbf{R}_{wi}^\mathsf{T} \; \mathbf{R}_{rw}^\mathsf{T} \qquad (6)$$
World Velocity
$$\mathbf{v}_{ws}(\mathbf{x}) = {}^t\mathbf{v}_{wi} + {}^t\mathbf{R}_{wi} \; {}^t\boldsymbol{\omega}_i^\wedge \; {}^{vel}\mathbf{p}_{is} \qquad (7)$$
Body Velocity
$$\mathbf{v}_s(\mathbf{x}) = {}^{bv}\mathbf{R}_{is}^\mathsf{T} \; {}^t\mathbf{R}_{wi}^\mathsf{T} \; {}^t\mathbf{v}_{wi} + {}^{bv}\mathbf{R}_{is}^\mathsf{T} \; {}^t\boldsymbol{\omega}_i^\wedge \; {}^{bv}\mathbf{p}_{is} \qquad (8)$$
Magnetometer
$$\mathbf{m}_s(\mathbf{x}) = {}^{mag}\mathbf{R}_{is}^\mathsf{T} \; {}^t\mathbf{R}_{wi}^\mathsf{T} \; \mathbf{m}_w \qquad (9)$$

Table I also shows the categorical type of states that are similar in purpose between sensor models. As a brief reference for future work; we evaluated a system set up where each sensor state was defined as a shared state between models i.e. ${}^{ip}\mathbf{p}_{is}$ and ${}^p\mathbf{p}_{is}$, the extrinsic calibration of the position and inverse position sensor was defined as one coupled state. The intent was to pose additional constraints in the form of coherence, but the results were limited in terms of robustness. This indicates that the system definition cannot be oversimplified, and individual states need to be granular while the system becomes more versatile. However, the states for the reference frames $\mathbf{p}_{rw}$ and $\mathbf{R}_{rw}$ are shared which benefits robustness.

TABLE I: Sensor model state association and sensor reference state sharing. Fields marked with x indicate a shared state, transformation entries represent a dedicated state for the model in the columns state category, and a free space indicates that a state is not required.

Model	$\mathbf{p}_{is}$	$\mathbf{R}_{is}$	$\mathbf{p}_{rw}$	$\mathbf{R}_{rw}$
Position	${}^p\mathbf{p}_{is}$		x	x
Inverse Position	${}^{ip}\mathbf{p}_{is}$	${}^{ip}\mathbf{R}_{is}$	x	x
Rotation		${}^r\mathbf{R}_{is}$		x
Inverse Rotation		${}^{ir}\mathbf{R}_{is}$		x
World Velocity	${}^{vel}\mathbf{p}_{is}$			
Body Velocity	${}^{bv}\mathbf{p}_{is}$	${}^{bv}\mathbf{R}_{is}$		
Magnetometer		${}^{mag}\mathbf{R}_{is}$		

Before continuing to the selection of states and models,

we need to define the format of states and measurements. Translation states, as well as position $[m]$, velocity $[m/s]$, and magnetometer measurements are defined as direction vectors in the 2-Sphere S^2. Because of this, the unit of the magnetometer measurements are not relevant for the detection process; units may be Tesla, 'counts' as provided by certain System on Chips (SoCs), or any.

Rotational states $\boldsymbol{\omega}$ within the sensor model definition are defined trough their tangent space value $\boldsymbol{\omega}$, and therefore, the parameters for a particular rotation are not unique. This does not pose issues in theory, but for real-world applications, this can result in numerical issues if the values for $\boldsymbol{\omega}$ are of higher magnitude. Thus, given the axis-angle representation, we are limiting the magnitude of the angle for all rotational states in the range of $-\pi \le \boldsymbol{\omega} \le \pi$ by defining individual loss terms.

$$\mathcal{L}_{\text{rot}}(\boldsymbol{\omega}) = \begin{cases} \|\boldsymbol{\omega}\| - \pi, & \text{if } \pi < \|\boldsymbol{\omega}\| \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$\text{with } \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\|\boldsymbol{\omega}\| = \boldsymbol{\omega} = \log(\mathbf{R})^\vee, \quad (11)$$

$\mathbf{R}$ being a rotation matrix, $\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ the axis of rotation and $\|\boldsymbol{\omega}\|$ as the angle of rotation.

Our method utilizes soft Booleans $\mathcal{B} = [0, 1]$ for the selection of sensor models. Ideally, such soft Booleans can vary in a limited range during the optimization process $b_{\text{opt}} \in \mathcal{B}$ and are categorized in a Boolean set in the final stage of the optimization $b_{\text{final}} \in \{0, 1\}$. All selectors b_{opt} are valid candidates when the method is initialized. The soft Boolean selectors converge concurrently during the optimization while their value range is constrained by dedicated loss terms (see Fig. 4 in the experiment section). Given ideal data, the selectors can converge to a single choice. Given realistic and imperfect data, the choice can become more ambiguous. Because of this, we define a decision and health metric to define b_{final} and to ensure that the correct model was selected.

As mentioned in the related work (Sec. II), our goal is to not perform an iterative process such as, branch and bound, or randomization of Boolean code tables. The method described by this work performs an optimization and selection process on all states simultaneously in stage one.

Utilizing the described soft Booleans as selection operators allows for a grouping of multi-dimensional geometrical terms, which is required for the selection of sensor models within the over-determined system definition. While LASSO promotes sparsity for individual state-values $\mathbf{v}_i \in \mathbb{R}^n$ as means of selections i.e., $\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1$, $\boldsymbol{\alpha} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$, the utilization of multiplicative soft Booleans ($b_i \in \mathcal{B}$) extends this approach and allows for the grouping and forming of multi-dimensional states. Let $\mathcal{L}_{\text{B}}$ (b) be a collection of loss-terms defined by Equation (14) to enforce the properties of soft Booleans for the final system definition. This allows for multiplicative selections such as:

$$\boldsymbol{\beta} = [b_1\mathbf{v}_1, \; b_2\mathbf{v}_2, \dots, b_n\mathbf{v}_n] \quad (12)$$

$$\min_{\mathbf{b}} \; y = \sum_i \mathcal{L}_{\text{B}}(b_i). \quad (13)$$

Where b_i is the Boolean selector for the $\mathbf{v}_i$ state. Please keep in mind that this example only shows the regularization as an example with respect to a full problem definition of a system.

We evaluated two concepts: a method for *individual state selections*, which allows multiple selection operators to be true, and a *direct selection approach*, which allows a *one-out-of-many* single choice of pre-defined sensor models.

The concept of *individual state selection* was limited in scalability and robustness for real-world scenarios. The corresponding system definition requires the augmentation of all sensor model states by individual soft Booleans and careful modeling of the inter-dependencies among proposed sensor models. The main reason for the limited robustness is a sensitivity to the initial convergence phase. One of the main problems addressed by this work is that the selection of states and the convergence of state values should coincide. However, this approach was sensitive to signal attenuation and prone to cross-detection of states due to over-fitting, promoted by incorrect utilization of states.

Assuming a measurement was given by a position sensor. The concurring states of a magnetometer model would produce a signal component with a much lower magnitude compared to global position information. A magnetic field vector can freely rotate within the noise band of the position sensor signal and lead to the minimization of the system's residual, but results in an overall sensor model that is not meaningful. This was also evaluated by using Particle Swarm Optimization (PSO) to circumvent such issues, but resulted in untraceable processing time and limited success for ill-conditioned scenarios.

The approach for *pre-defined sensor model selection* did not fall short of the outlined issues and is presented in the following. The method allows a robust selection of dedicated sensor models (Eq. (3)-(9)) as a whole and resolves the weaknesses of the approach for *single state selection* which required individual loss terms for each selection operator, while the *direct selection approach* requires a global set of regularizations to enforce the soft Boolean properties and achieve a *one-out-of-many* selection.

A sensitivity analysis in Section V-A has shown that the presented method produces zero false positives for data with high SNR, indicating that ideally, only one model can be selected for close to ideal data.

While this is true, given simulated and noise-free data with a trajectory that ensures the observability of states, an application with real-world data requires additional steps to achieve a robust result. The first step is to normalize the unknown measurement as well as the individual components of the over-determined system model to the value range $[0, 1]$. This normalization mitigates errors due to over-fitting and sensitivity to signal attenuation, which was shown as an example in the previous section. In particular, the magnetometer sensor model (Eq.(9)) and especially the ubiquity of its state $\mathbf{m}_{\text{w}}$ can freely contribute to the minimization of noise-afflicted residuals. Measurement and model normalization addresses this issue while also improving resilience to noise (see Sec. V).

However, noise-afflicted data and uncertainties of the given core states can still prevent a definite single model selection since a number of sensor models can contribute to a solution partially. Thus, a set of constraints, operating on the selection

operators b, are defined to enforce the convergence to a single selection. Given $b_* \in \mathcal{B}$, $b \in \mathcal{B}^n$, the following loss terms can be combined as

$$\mathcal{L}_{\mathsf{B}}(b) = \mathcal{L}_{\mathsf{norm}}(b) + \mathcal{L}_{\mathsf{pos}}(b) + \mathcal{L}_{\mathsf{std}}(b). \qquad (14)$$

First, enforce that b is constrained to a single overall contribution of models

$$\mathcal{L}_{\mathsf{norm}} = \left| \|b\|_1 - 1 \right|_2, \qquad (15)$$

which does not over-constrain the system and allows for a free convergence of state-values and selectors because a mixture of models is still possible. A second constraint enforces that all selectors are positive by posing a penalty for negative selectors

$$\mathcal{L}_{\mathsf{pos}} = \left\| \{ b \mid b \in \mathsf{b}, \ b < 0 \} \right\|_2. \qquad (16)$$

This constraint is crucial for the rotation models (Eq. 5 and 6) because the measurement of a rotation signal and the model definition provide rotational information expressed in the tangent space. Since, $\log(\mathbf{R}^\mathsf{T})^\vee \equiv -\boldsymbol{\omega}$, the definition of the rotation and inverse rotation model can interfere and possibly cancel each other out while, at the same time, reaching a valid decision for the correct module. A valid solution from the system point of view could otherwise be a selection with $([b_{\mathbf{R}_{\mathsf{sr}}} = 0.5, b_{\mathbf{R}_{\mathsf{rs}}} = -0.5, b_{\mathbf{P}_{\mathsf{sr}}} = 1])$, leading to ambiguities for the decision-making process.

While the normalization already improves issues due to signal attenuation, an additional improvement can be made by enforcing single vector module entities to $\|\mathbf{v}\|_2 = 1$. In the presented case, this only concerns the magnetometer model and renders the constraint to

$$\mathcal{L}_{\mathsf{vec}} = \left| \|\mathbf{m}_{\mathsf{s}}\|_2 - 1 \right|_2. \qquad (17)$$

Single vector entities can highly contribute to over-fitting and false positives, especially during the transient phase. For this reason, this loss term is squared in the final system definition (Eq. 20) to provide steeper gradients for faster convergence to fully avoid possible incorrect convergence of the overall system.

Finally, to enforce a *single* decision, we utilize the regularization introduced by Theorem IV.1 in conjunction with $\mathcal{L}_{\mathsf{norm}}$, which ensures $\|\mathbf{b}_{\mathsf{vec}}\|_1 = 1$ as

$$\mathcal{L}_{\mathsf{std}} = \left\| std(\mathbf{b}_{\mathsf{vec}}) - \frac{1}{\sqrt{N}} \right\|_2, \qquad (18)$$

with $std(\mathbf{x}) = \sqrt{\frac{1}{N-1} \sum_i (x_i - \mu)^2}$ and N as dimension of b.

The reasoning behind the loss-term $\mathcal{L}_{\mathsf{std}}$ and the corresponding theorem IV.1 originated based on the property of maximum variation for a vector $\mathbf{x} \in \mathbb{R}^N$ as follows: If we pose the constraints $\mathcal{L}_{\mathsf{norm}}$ and $\mathcal{L}_{\mathsf{pos}}$ onto a decision vector $\mathbf{x}$ with soft Boolean and non-random elements x_i, then the standard deviation of this vector has an upper limit, equivalent to the maximum variation of $\mathbf{x}$. The proof shows, that the specific maximum variation of vector $\mathbf{x}$ is $\frac{1}{\sqrt{N}}$ and can only be reached if the vector has a single entry of one and zeros otherwise. This property is essential for loss-term $\mathcal{L}_{\mathsf{std}}$ and the presented one-out-of-many decision making approach which enforces a single selection.

Theorem IV.1. *Let* $\mathbf{x} \in \mathbb{R}^N$ *be a vector. Let* $std(\mathbf{x})$ *represent the standard deviation of* $\mathbf{x}$. *Assume* $0 \leq x_i \leq 1 \ \forall \ i = 1, \ldots, N$ *of vector* $\mathbf{x}$, $\|\mathbf{x}\|_1 = 1$, *and* $std(\mathbf{x}) = \frac{1}{\sqrt{N}}$. *Let* M *be the number of non-zero entries* x_i *of* $\mathbf{x}$. *Then, there are only* N *constellations for* $\mathbf{x}$ *with* $M = 1$, *and zero constellations with* $M > 1$.

Proof.

1. We proof by contradiction and therefore assume $\exists \ M > 1 \mid 0 < x_i < 1 \ \forall \ i = 1, \ldots, M$.

2. Given the hypothesis $std(\mathbf{x}) = \frac{1}{\sqrt{N}}$ and $\|\mathbf{x}\|_1 = 1$, we get

$$std(\mathbf{x}) = \frac{1}{\sqrt{N}} = \sqrt{\frac{1}{N-1} \sum_i^N (x_i - \mu)^2},$$

$$\frac{1}{N} = \frac{1}{N-1} \sum_i^N (x_i - \mu)^2,$$

$$\frac{N-1}{N} = \sum_i^N (x_i - \mu)^2$$

3. where $\sum_i^N (x_i - \mu)^2$ can be split in two sets, $S_0 = \{i \mid x_i = 0\}$, $S_\alpha = \{i \mid 0 < x_i < 1\}$ of vector $\mathbf{x}$

$$\frac{N-1}{N} = \sum_{i \in S_\alpha}^M (x_i - \mu)^2 + \sum_{i \in S_0}^{N-M} \mu^2.$$

$$\mu = \frac{1}{N} \sum_i^N x_i = \frac{1}{N} \left(\sum_{i \in S_\alpha}^M x_i + \sum_{i \in S_0}^{N-M} 0 \right) = \frac{1}{N}.$$

4. Substituting for μ, we can show that:

$$\frac{N-1}{N} = \sum_{i \in S_\alpha}^M (x_i - \mu)^2 + \sum_{i \in S_0}^{N-M} \mu^2,$$

$$= \sum_{i \in S_\alpha}^M \left(x_i - \frac{1}{N} \right)^2 + \sum_{i \in S_0}^{N-M} \left(\frac{1}{N} \right)^2$$

$$= \sum_{i \in S_\alpha}^M \left(x_i^2 - 2x_i \frac{1}{N} + \frac{1}{N^2} \right) + \frac{N-M}{N^2}$$

$$= \sum_{i \in S_\alpha}^M \left(x_i^2 \right) - \frac{2}{N} \sum_{i \in S_\alpha}^M x_i + \frac{M}{N^2} + \frac{N-M}{N^2}$$

$$= \sum_{i \in S_\alpha}^M \left(x_i^2 \right) - \frac{2}{N} \sum_{i \in S_\alpha}^M x_i + \frac{1}{N},$$

5. where $\sum_i^M x_i = \|1\|_1 = 1$ was used in the last equation. It follows that

$$\frac{N-1}{N} = \sum_{i \in S_\alpha}^M \left(x_i^2 \right) - \frac{1}{N},$$

$$1 - \frac{1}{N} = \sum_{i \in S_\alpha}^M \left(x_i^2 \right) - \frac{1}{N},$$

$$\sum_{i \in S_\alpha}^M \left(x_i^2 \right) = 1.$$

6. The last equality is true if and only if $M = 1$, proving the theorem.

$\square$

An example can be given for the two dimensional case:

1. Given the following equality of the constraints $\sum_i^N (x_i^2) = \|\mathbf{x}\|_1 = 1$ and an example with $N = 2$

$$\|\mathbf{x}\|_1 = x_1 + x_2 = 1$$
$$x_1 = 1 - x_2.$$

2. Substituting for x_1, and solving the polynomial for x_2 we can show that

$$\sum_i^N (x_i^2) = x_1^2 + x_2^2 = 1$$
$$= (1 - x_2)^2 + x_2^2 = 1,$$
$$0 = x_2^2 - x_2 \Rightarrow x_{2_{1,2}} = \{0, 1\}.$$

3. Solving for x_1 given $\sum_i^N (x_i^2)$ and the solutions for x_2

$$x_{1_{1,2}} = \{1, 0\} \text{ given } x_{2_{1,2}} = \{0, 1\},$$

4. shows mutual exclusive boolean selectors and supports the proof that constraints are only fulfilled for $M = 1$.

A visual proof that the sets of values for the given constraints are only inclusive where the conditions mentioned above are met, is visualized by Figure 3. Experimental results, which show how this loss term improves the soft Boolean convergence, is presented by Figure 4 (Sec. V).

The full and final approach can be addressed with common non-linear solvers such as the Levenberg–Marquardt algorithm and numerically approximated Jacobians. This is beneficial due to the high variability of the system definition and its consecutive Jacobians.

The system definition of this approach is defined by

$$f_{\text{sys}}(\mathbf{x}_\text{c}, \mathbf{x}_\text{m}, \mathbf{b}) = b_{\mathbf{P}_\text{sr}} \, \mathbf{p}_\text{sr}(\mathbf{x}) + b_{\mathbf{P}_\text{rs}} \, \mathbf{p}_\text{rs}(\mathbf{x}) + b_{\mathbf{R}_\text{rs}} \, \boldsymbol{\omega}_\text{rs}(\mathbf{x}) \quad (19)$$
$$+ b_{\mathbf{R}_\text{sr}} \, \boldsymbol{\omega}_\text{sr}(\mathbf{x}) + b_{\mathbf{v}_\text{ws}} \, \mathbf{v}_\text{ws}(\mathbf{x}) + b_{\mathbf{v}_\text{s}} \, \mathbf{v}_\text{s}(\mathbf{x}) + b_{\mathbf{m}_\text{s}} \, \mathbf{m}_\text{s}(\mathbf{x})$$

$$\mathcal{L}_{\text{loss}}(\mathbf{b}) = \lambda_\text{n} \, \mathcal{L}_{\text{norm}}(\mathbf{b}) + \lambda_\text{p} \, \mathcal{L}_{\text{pos}}(\mathbf{b}) \quad (20)$$
$$+ \lambda_\text{s} \, \mathcal{L}_{\text{std}}(\mathbf{b}) + \lambda_\text{v} \, \mathcal{L}_{\text{vec}}(\mathbf{b})^2$$

$$\min_{\mathbf{x}_\text{m}, \mathbf{b}} \ y = \frac{1}{2} \frac{\sum_i \|f_{\text{sys}}(\mathbf{x}_{\text{c}_i}, \mathbf{x}_{\text{m}_i}, \mathbf{b}) - {}^t\mathbf{S}_i\|}{\sqrt{n_{\text{samples}}}} + \mathcal{L}_{\text{loss}}(\mathbf{b}) \quad (21)$$

with the over-determined system definition f_{sys}, previously introduced constraints $\mathcal{L}_{\text{loss}}$, individual weights λ_*, $\mathbf{S}$ as the unknown sensor signal, and n_{samples} as the number of sample points to render the system error invariant to the length of a dataset.

In this case, the norm of $\mathbb{R}^{3 \times n}$ is the Frobenious norm. By normalizing the Frobenious norm based on $\sqrt{n_{\text{samples}}}$, we obtain the RMSE, which is agnostic to the number of samples.

Since the system components are normalized for this approach, the state values of the selected module are unscaled and need to be refined, which is the purpose of stage 2 (Sec. IV-B).

An additional note on module similarities: By examining Eq. (22) and (23), it can be seen that the result for both

expressions is a vector in $\mathbb{R}^3$. An issue can arise if the rotations defined by (23) do not fully minimize the error and express $\log(\mathbf{R}_\text{sr})^\vee = \boldsymbol{\omega}_\text{sr}$ incorrectly. For this scenario, a multiplicative vector could also minimize the error and contribute to the solution due to overfitting or over-determination. In this case, Eq. (22) can double as a solution for a rotation sensor module.

$$\mathbf{m}_\text{s} = \mathbf{R}_\text{is}^\mathsf{T} \, {}^t\mathbf{R}_\text{wi}^\mathsf{T} \, \mathbf{m}_\text{w} \quad (22)$$
$$\log(\mathbf{R}_\text{sr})^\vee = \log(\mathbf{R}_\text{is}^\mathsf{T} \, {}^t\mathbf{R}_\text{wi}^\mathsf{T} \, \mathbf{R}_\text{rw}^\mathsf{T})^\vee \quad (23)$$

This issue can be circumvented by enforcing additional constraints on $\mathbf{m}_\text{w}$. This is the reason why we decided to fully isolate rotation signal identification from all other sensor module expressions. Besides the issue described above, the main reason for this decision is that rotational signals need to be expressed in the tangent space and require additional conversion to $\mathbb{R}^3$ before the detection in stage 1 can be performed. The detection and refinement of calibration states for rotational information are still of interest to define the nature of the measurement expression (normal or inverse) and the calibration states.

Defining the weights for loss terms is often a heuristic approach. However, the weights for this system setup (Eq.(20)) are defined such that all initial loss terms result in approximately the same range of magnitude. Weights for the presented setups are $\lambda_\text{p} = 200, \lambda_\text{n} = 50, \lambda_\text{v} = 100, \lambda_\text{s} = 20$, and the development of weighted loss terms throughout the optimization process is shown by Figure 4.

Finally, the decision-making process is straightforward. Given the final result of the selectors, $\mathbf{b}_{\text{vec}}$, the soft Boolean with the highest magnitude is selected to determine a candidate for the model choice. Since we know that the problem definition is not convex, we need to ensure that the choice is not a false positive before proceeding to the next stage. Thus, a health metric, based on the distance of the soft Boolean with the highest and second highest magnitude, together with the remaining loss terms that enforce the soft Boolean properties, are used to identify false positives. This metric is detailed and validated in the experiment Section V-B.

Overall, this approach reaches a final decision while preventing false positive decisions due to noise and uncertainties of the provided input signals and module cross-detections due to signal attenuation. The scalability of this approach is also improved with respect to the single-state selection because of minimized system inter-dependencies.

B. Sensor State Calibration

This section details the process of stage 2 (see Fig. 2) concerning the estimation of sensor state calibrations and the necessity of reference frames for dedicated sensor modules. In general, the same sensor model definitions that are used for stage 1 are used for stage 2, with the difference that only the previously detected sensor model is selected and processed. While the unknown measurement input and the estimated measurement were normalized in stage 1, e.g., for noise resilience, stage 2 requires the raw system information for the determination of calibration states.
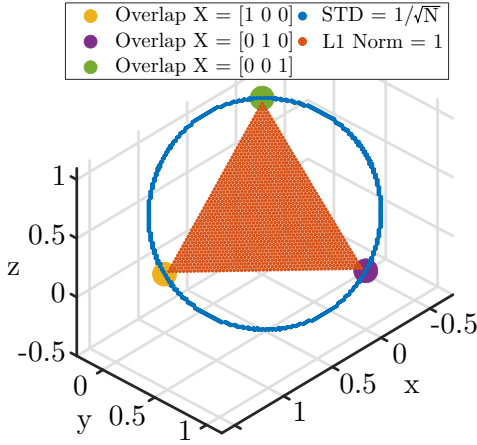
Fig. 3: Visual proof of Theorem IV.1

Since the problem definition is non-convex, the initialization point of optimization parameters is vital for correct convergence. We performed tests that showed that a naive initialization of these states could lead to incorrect results. A more robust initialization and convergence is given if the states are initialized with the final parameters from stage 1. These parameters are obtained with the normalized system error definition in stage 1, possibly leading to higher resilience to local minima due to lower gradients in the optimization landscape. Experiment Section V-C shows corresponding results.

The system error definition for rotations differs for stage 1 and stage 2. As introduced earlier, the rotations for the calculation of the system error are expressed in the tangent space. In this space, the expression of rotations is not unique and reoccurring. Since these parameters are not normalized in stage 2, it needs to be assured that the repetitiveness of values expressing the same rotations is consistent and comparable for the system input and the estimated measurement. Thus, the error between two series of rotations ω_a and ω_b for rotation models in stage 2 is defined as follows:

$$\epsilon_{\text{rot}} = \frac{\sum_i \left\| \log(\exp(\omega_{A_i}^\wedge)\ \exp(\omega_{B_i}^\wedge)^\mathsf{T})^\vee \right\|_2}{N}, \quad (24)$$

this ensures that rotations are unique trough rotation matrices in $SO(3)$ when determining the error.

It also needs to be noted that the states of the models need to be observable. This concerns the reference frames for the rotation and the inverse position sensors. Using Figure 1 as a reference; for rotation modules, the reference frame rotation $\mathbf{R}_{\text{rw}}$ is unobservable and needs to be fixed in order to obtain a valid calibration for $\mathbf{R}_{\text{is}}$. For the inverse position module, it can be seen that the sensor reference frame R is not constrained by a rotation. Thus, the translation and rotation of the reference frame are codependent, and there are multiple combinations for $\mathbf{p}_{\text{rw}}$ and $\mathbf{R}_{\text{rw}}$ that are expressing a valid solution.

The second step within stage 2 is the reference frame determination. Until this point, the model detection and estimation of initial calibration states assumed that sensors require a reference frame. The last step in the model and state identification is to determine if a reference frame is required. This is important information for consecutive steps where the detected sensor model is parameterized and used by a recursive

filter, allowing for sensor self-calibration. If a reference frame is not required, it is beneficial not to include it.

The reference frame is defined by $\mathbf{p}_{\text{rw}}$ and $\mathbf{R}_{\text{rw}}$. We use the LASSO approach and the L1-Norm as a penalty for the reference frame. The following loss is added to the optimization process for reference frame detection, while the sensor to IMU calibration is also subject to optimization,

$$\mathcal{L}_{\text{ref}} = \|\mathbf{p}_{\text{rw}}\|_1 + \|\boldsymbol{\omega}_{\text{rw}}\|_1. \quad (25)$$

$\mathcal{L}_{\text{ref}}$ represents a non-minimizable loss term if a reference frame is required. Estimating the calibration state itself is a property of the individual sensor model definition in Equation (19). This process allows for clear decisions on the existence of reference frames based on the remaining magnitude of the reference states. Experimental results are shown in Section V-C.

V. Experiments

This section investigates the reliability and properties of the presented method. Initial experiments are performed on ten simulated Lissajous figures, perturbed with various noise settings to analyze the sensitivity of the approach (Sec. V-A) and reliability of the introduced health metric (Sec. V-B) on 1750 case scenarios for statistical significance. In addition, the approach is applied within a real-world scenario to prove its applicability (Sec. V-D and V-E).

The simulated trajectories have the following stats: Range in position 0-3.5 m, max. velocity 3.5 m/s, roll and pitch angles within $\pm 30°$, yaw $\pm 30°$, lateral max. acceleration 1.7 m/s^2, and max. angular velocity 3.2 rad/s. Experiments for the sensitivity analysis add specific noise to individual system inputs. Unless stated otherwise, the following noise parameters are used for a realistic setup. For the general approach, we assume that core state information is provided by an estimator. However, for the simulated experiments, we are using ground truth data with additive noise to provide core states with relatively noisy data. This ensures repeatability and consistency throughout all simulated cases. Please keep in mind that these are 1-sigma values. The corresponding 3-sigma, especially for the rotation, is intended to prove that this method also works with high/sufficient noise:

System input: Position 0.1 m, velocity 0.05 m/s, orientation 2°, angular velocity 0.014 $deg/s/\sqrt{hz}$ (continuous) as for the BMI055 MEMS IMU, used e.g. by the PX4 autopilot. *Measurements:* Position 0.3 m, Rotation 3° axis angle rotation, Velocity 0.05 m/s, Magnetometer 6° axis angle rotation.

A. Sensitivity Analysis and Validation

This section details a sensitivity analysis of the proposed method and the system definition to gain insights into robustness and noise resilience. The presented approach is designed to identify a single sensor modality at a time and therefore, the following evaluation is performed for individual sensor models. The failure points and reported false selections are direct outcomes of stage 1 and are not evaluated in conjunction with the presented health metric (Sec.V-B). In the following, Tables II, III, and IV show the results for three different

sensitivity analysis aspects by evaluating successful and failed detections with respect to different noise levels. The setup and objective for each experiment is detailed in the following. However, all experiments use the same ten individual trajectories and the tables detail the number of false detections out of the ten trails each. It is expected that low noise profiles cause no false detections and, e.g., unrealistically high noise perturbations cause false detections. High noise scenarios can also be interpreted as sensor failure and data with a significant number of outliers. In such cases, it is important that failures are detected reliably. A different data segment can then be used for re-evaluation.

This analysis follows two approaches, the first evaluation is done with reasonable noise for the system inputs (core states) and gradually increasing sensor measurement noises. This relates the analysis to real-world applications and intuitively interpretable results. Table II shows the results when increasing the noise levels until a respective sensor type fails. It can be seen that the position and rotation detections hardly fail despite the comparatively high noise levels. This is due to the characteristics of the measurement types, the loss terms that prevent over-fitting and cross-detection, as well as reasonable uncertainty of the core states. As a reference to the health metric in the next section, none of the position and rotation selections, which have been correctly selected, were close to the decision threshold.

TABLE II: Variation of the sensor measurement noise given a reasonable uncertainty of the system states (Sec.V). The table shows the number of false decisions for 10 different trajectories. High noise values are intended to result in false choices to show the sensitivity of the system.

Sensor & Noise				False selections in 10 trails
Noise $[m]$	**3**	**6**	**8**	**10**
Position	0	0	0	2
Inv Position	0	0	0	0
Noise axis angle $[deg]$	**10**	**45**	**90**	**135**
Rotation	0	0	0	0
Inv Rotation	0	0	0	0
Noise $[m/s]$	**0.5**	**1.5**	**2**	**3**
World Velocity	0	0	2	8
Body Velocity	0	0	3	10
Noise axis angle $[deg]$	**10**	**45**	**90**	**135**
Magnetometer	0	0	0	3

The second approach relates the input noise to the variation of the trajectory and thus provides a more comparable insight into the sensitivity of the system. For a fair validation, we need to establish a common ground for the comparison of various trajectories and corresponding changes for the input data. For this, we utilize a definition of the *signal-to-noise ratio* (SNR). Since the SNR is a local ratio between the power of a signal m and the power of its noise σ, the standard SNR metric $\text{SNR} = \frac{m}{\sigma}$ would be impractical given robotics applications because the power of the noise would need to change with the power of the signal.

Sensors in robotics usually have a magnitude-independent noise. Thus, we are using a modified definition of the SNR and use the standard deviation λ of the trajectory (system input) as a metric for the variation of the signal and the power of

the noise $\text{SNR} = \frac{\lambda(\mathbf{x})}{\sigma_n}$ with $\lambda(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_i (x_i - \mu)^2}$. This allows the generation of noise-afflicted data with the same SNR for multiple test trajectories. The noise for each trajectory will be different according to this definition.

While the noise for translations and vectors is additive, the noise for rotations and the magnetic vector $\mathbf{m}_w$ are generated using the definition of the axis angle rotation based on the SNR α which, in the presented case, affects each axis equally

$$\alpha = \|\boldsymbol{\omega}\| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}, \tag{26}$$

$$\alpha^2 = 3\,\omega_i^2 \text{ for equal effect on each axis}, \tag{27}$$

$$\omega_i = \frac{\alpha}{\sqrt{3}}. \tag{28}$$

Table III shows the results of a setup in which all system inputs are noise afflicted with the same SNR to the point of failure, done for all sensor model inputs individually. The results show that most model detection fails at SNR $= 0.5$, which determines that the noise floor is twice the magnitude of the variation of the input signal.

TABLE III: Variation of the trajectory dependent SNR for all system inputs, the known states, and unknown sensor measurements, for a fair comparison across trajectories.

SNR	2	1.5	1	0.5	0.4
Sensor		False selections in 10 Trials			
Position	0	0	0	0	0
Inv Position	0	0	0	1	1
Rotation	0	0	0	2	5
Inv Rotation	0	0	0	2	2
World Velocity	0	0	2	10	10
Body Velocity	0	0	2	9	7
Magnetometer	0	0	0	3	3

Finally, Table IV shows a more elaborate analysis with noise perturbations and all system parameters individually over a range of SNRs and all individual sensor signals as inputs. It can be seen that the majority of detections fail if the noise for the measurement itself is high (SNR < 0.5, column m). Furthermore, the magnetometer and body velocity detection is sensitive to high noise on core state rotation (SNR < 0.5, column r), and the velocity is slightly sensitive to high core state velocity noise (SNR < 0.4, column r).

All tests concerning sensitivity show that position and rotation detections are generally more robust, which is due to their characteristics, as described in the beginning.

TABLE IV: Individual SNR perturbations on all system inputs while retaining respective other inputs at reasonable uncertainty. This experiment aims to visualize the sensitivity of the posed sensor models (rows) towards noise perturbations on core states (columns). System inputs: Position p, velocity v, rotation r, and angular velocity ω as well as measurement input m.

SNR			1					0.5					0.4		
Input	p	v	r	ω	m	p	v	r	ω	m	p	v	r	ω	m
Sensor						Number of false selections in 10 trails									
Position	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Inv Position	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rotation	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Inv Rotation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
World Velocity	0	0	0	0	0	0	3	0	0	3	0	6	0	0	2
Body Velocity	0	0	0	0	0	0	1	7	0	2	0	2	8	0	6
Magnetometer	0	0	0	0	0	0	0	3	0	0	0	0	7	0	0

Both approaches serve a specific purpose; the first approach applies to real-world scenarios and to which point a noise-afflicted sensor signal can be detected, given that the system's state is reasonably accurate. The second approach uses comparable noise settings for all trajectories and individually challenges the overall system to the point of failure and reveals the sensitivity of individual components toward one another.

As an additional validation element, Figure 4 visualizes the effect of the STD loss term and the convergence of the loss functions.

In the following, we are introducing two examples for the velocity sensor identification from the larger analysis shown by Table II. Using the same trajectory (see Fig. 5, 6), we are showing one case, which fails because of the predominant measurement noise setting of $3\,\mathrm{m/s}$ (false selection case indicated by the table), and one successful detection using $1.5\,\mathrm{m/s}$ measurement noise.
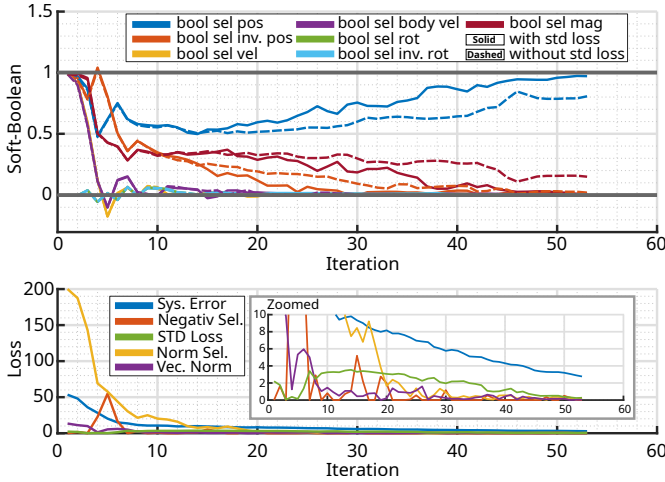

Fig. 4: Experimental comparison and further proof of Theorem IV.1 for the STD loss term, which enforces a single model choice. The data for this example is perturbed with noise to promote a case that tends towards an ambiguous decision. **Top:** Convergence of soft Booleans with (solid) and without (dashed) the STD loss term. It can be seen that a fully accurate decision is not reached if the loss term is not enabled (Position Model: 0.8, Magnetometer: 0.2). This further confirms the exact scenario of possible over-fitting (Sec. IV) if dedicated loss terms are not defined. **Bottom:** Convergence of the system error and all loss terms for the case where the STD loss is used.
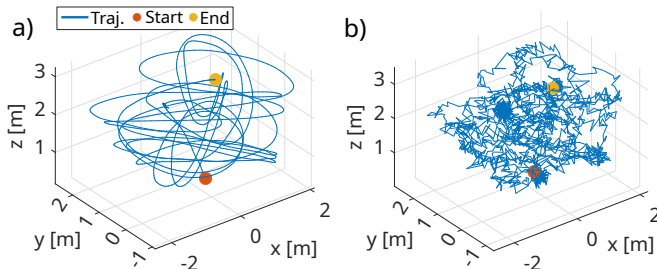

Fig. 5: 3D Lissajous trajectory for one of ten test datasets. Figure a) without noise on the system input (core states) and Figure b) with noise, as defined in the introduction of Sec. V.
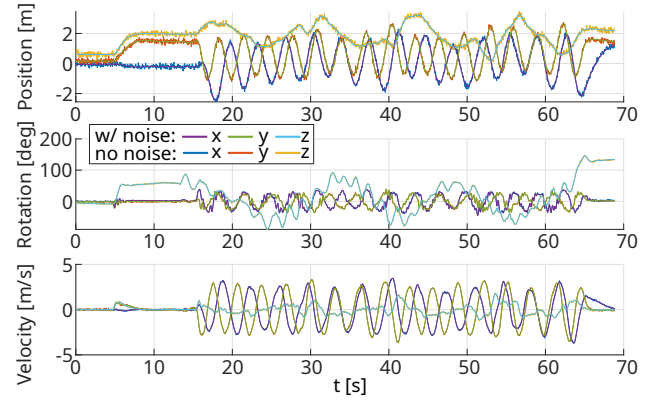

Fig. 6: 2D position, rotation and velocity, corresponding to Figure 5, as the base localization information for the vehicle.
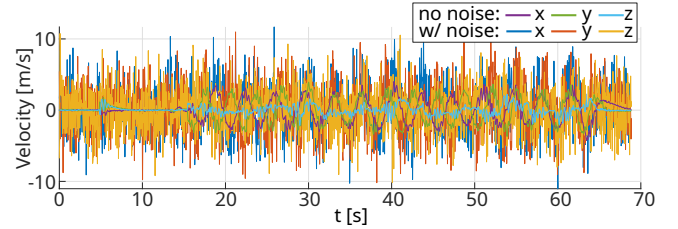

Fig. 7: Ground truth velocity measurement in relation to the noise-perturbed signal ($\sigma = 3\,\mathrm{m/s}$), causing an incorrect model identification.
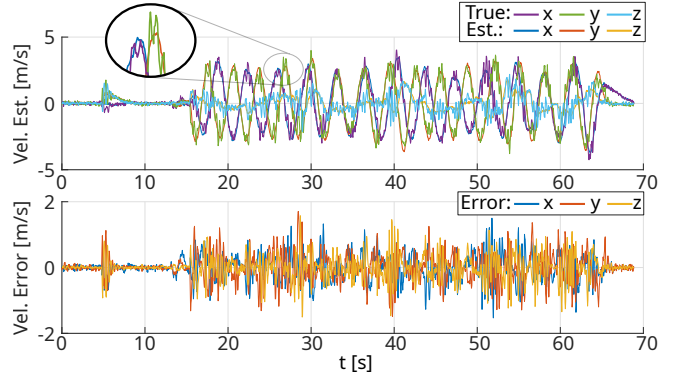

Fig. 8: Reprojection of the incorrectly identified position sensor model onto the ground truth velocity measurement. The estimated model loosely resembles the true signal but shows strong oversimplifications as highlighted by the zoomed section and the error plot.
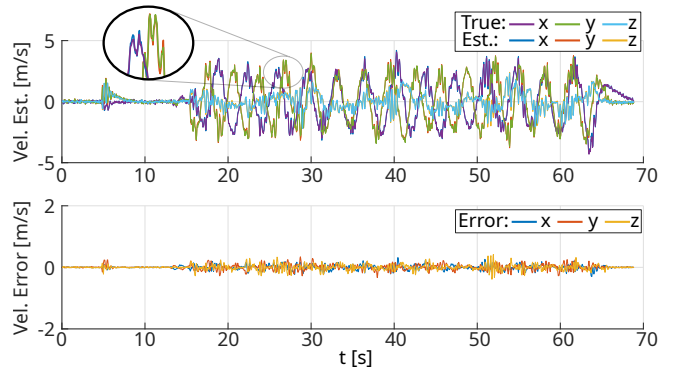

Fig. 9: Reprojection of the correctly identified and parameterized velocity sensor model. The estimated data matches the true signal closely, confirmed by the error plot with a significantly lower magnitude compared to Figure 8.

Figure 7 shows the velocity measurement for the failing case and relates the noise-perturbed measurement to the ground truth signal. In this case, the sensor model was incorrectly identified as a position sensor, Figure 8 shows that the re-projected measurement, based on the model and estimated parameterization resembles the true value to some extent. However, the zoomed element shows that the signal is strongly oversimplified. To the contrary, Figure 9 for the successful detection shows that the estimated signal matches the ground truth of the measurement closely. Comparing the error plots of both cases also shows a significant difference in magnitude.

TABLE V: Results of the two example identification cases shown by Fig. 8 and 9. Section V-B will show that the introduced health metric can detect the false model selection reliably.

	Noise	**Model Sel.**	**Soft Boolean**		**RMSE**		
			Selector	Distance	x	y	z
Case 1	3 m	Pos.	0.548	**0.082**	3.32	3.33	3.39
Case 2	1.5 m	Vel.	0.564	**0.403**	1.51	1.48	1.49

Table V presents the results for both scenarios, soft Boolean values, the selected model, as well as the RMSE for the reprojection of the sensor model with the estimated parameterization. The next section will introduce a metric to identify false positives and show that such failure cases can be detected and rejected reliably. We will show that the distance between the highest and second highest boolean is one of the main parameters for this identification and that the example case presented here, with a false detection and a Boolean distance of 0.082 would be rejected accordingly.

B. Health Metric

Since the problem definition for the presented method is not convex, a metric for detecting false decisions is vital. The choice for a sensor model is made based on the soft Boolean with the highest magnitude. We define an additional health metric that uses the distance between the highest and second-highest soft Boolean as well as the final values of the loss functions, which define the soft Boolean properties.

The distance between the two highest soft Booleans correlates with the certainty of a correct selection. The box plot in Figure 10 illustrates this relation. The left side shows the Boolean distances for correct and incorrect choices based on ground-truth selectors. The distances for the categorical "wrong choice" show a few outliers, but the first quantiles for both distances do not overlap. Please keep in mind that the data used for this box plot is the same as for the results of the sensitivity analysis with perturbed SNR in Table IV. This data was designed to intentionally produce false choices due to, in part, close to unrealistic high noise perturbations on the system inputs.

Given the box plot, a naive choice for a decision threshold would be at $\Delta b > 0.62$ labeled as the pre-loss term decision threshold in Figure 10. However, this includes two false positives and rejects an unnecessarily high number of true positives. Thus, in addition to outlier rejection, we also want to define a metric that promotes a high recall (low false

negatives) and 100% precision (no false positives). In the following, we introduce a metric to pre-condition the base of decision and filter decisions based on remaining loss terms, residual of the model, and the Boolean distance for each selection. This health metric will be deployed by rejecting results for which any individual criteria does not hold and substantially improve the precision ($\frac{\text{true positives}}{\text{true positives}+\text{false positives}}$) and recall ($\frac{\text{true positives}}{\text{false negatives}+\text{true positives}}$) for the decision making process.

The parameter space for the identification of meaningful thresholds for the outlier detection in six dimensions is to
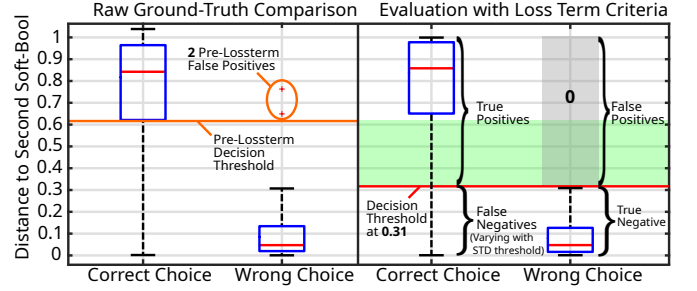

Fig. 10: Box plot for soft Boolean distances Δb (difference between highest and second highest) for correct and wrong model selections based on ground truth. **Left:** Raw samples, 1690 correct and 60 incorrect selections with two false positives for the wrong choice category within 1750 samples. Figure 12 is used to determine thresholds for loss parameters in order to remove such outliers. **Right:** Samples with removed false positives based on remaining loss term limits. The green area shows a decision range with corresponding trade-offs in precision and recall (Fig. 13, 14). The best recall (91%) at 100% precision is reached with the **red** Bool-Decisions threshold $\Delta b < 0.31$, $\mathcal{L}_{\text{norm}} < 0.2$, and $\mathcal{L}_{\text{std}} < 4.1$. **Data:** 1750 samples, from sensitivity analysis (Tab. IV). Based on the health metric, 603 samples are rejected, with no remaining false positives. Please keep in mind that selected samples are intended to fail due to high noise perturbations on the system inputs.
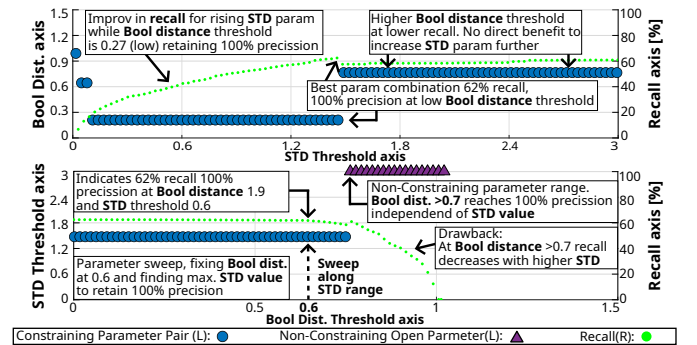

Fig. 11: Detailing of Figure 12, shows the parameter evaluation of Boolean distance threshold against STD threshold (bottom), and STD threshold against Boolean distance (top). The objective is to find the threshold with the lowest Boolean distance and highest STD value to reach the highest recall at 100% precision for the identification of false positives. Figure 12 shows this evaluation schematic for all parameters. Figure 13 and 14 show a traditional precision-recall graph, further refining the most meaningful parameters.
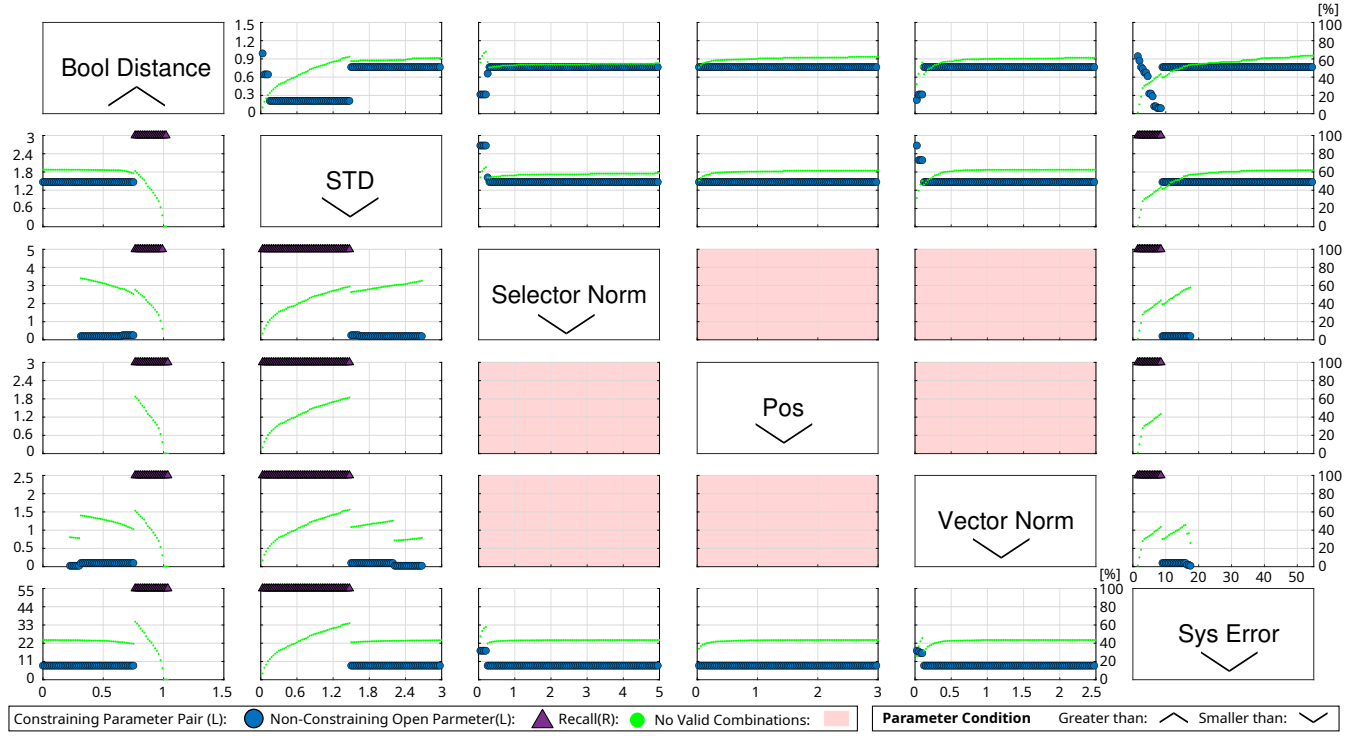
Fig. 12: Matrix scatter plot with an overview of relevant final system parameters given 1750 samples from the previous sensitivity analysis. The graph is used to determine a precursor for rejecting outliers and improving false positives based on parameter correlation and recall. The diagonal labels determine the axis labels. Parameters such as loss terms are rejected if they are smaller than ($\vee$) the threshold, and parameters such as Boolean distances are rejected if they are greater than ($\wedge$) the threshold. The left y-axis **(L)** relates to blue dots, representing a constraining parameter pair, and purple triangles indicating that a parameter pair is not constraining. The right y-axis **(R)** relates to green dots, representing the recall in [%] for the individual parameter pair. Red plots indicate no results for any parameter pair.
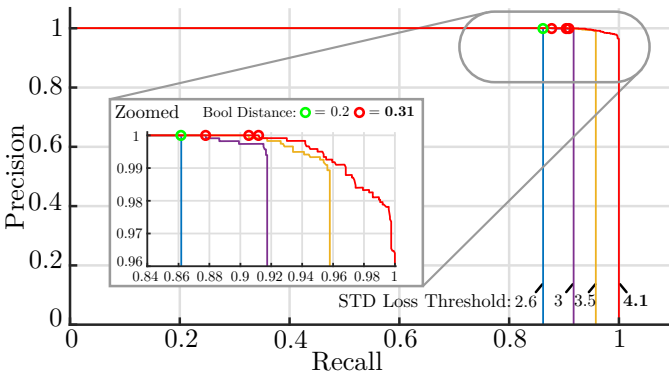

Fig. 13: Precision/Recall, for dedicated $\mathcal{L}_{std}$ over the soft Boolean distance threshold 0-1 and markers at dedicated distances. $\mathcal{L}_{std} < 2.6$ rejects all wrong decisions independently of Δb at a lower recall 86%. The best result for 100% precision is given at $\mathcal{L}_{std} < 4.1$ and a distance of $\Delta b > 0.31$ with 91% recall.

high for numerical evaluations. Because of this, we are using a graphical representation in Figure 12 that uses a scatter plot matrix and the paradigm of precision and recall at its core. Here we can use multiple axes in two dimensions and identify opposing effects of e.g. remaining loss terms. We can observe, which points of the value ranges do not retain 100% precision, and can observe this behavior over all remaining loss terms for (Δb, $\mathcal{L}_{std}$, $\mathcal{L}_{norm}$, $\mathcal{L}_{pos}$, $\mathcal{L}_{vec}$, and f_{sys}).
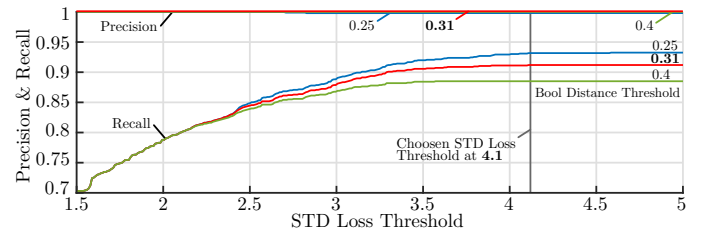

Fig. 14: Precision/Recall, for dedicated soft Boolean distances ranging over $\mathcal{L}_{std}$. The red precision graph for a distance of $\Delta b = 0.31$ remains at 100% precision below $\mathcal{L}_{std} < 4.1$ as the best threshold.

The scatter plot matrix in Figure 12 shows the same raw data as the left side of the box plot (Fig. 10). Figure 11 is intended to detail individual aspects based on the Δb, $\mathcal{L}_{std}$ loss-terms.

The diagonal blocks of this matrix represent the pivot point of a specific parameter. Scatter plots along a row of a respective diagonal entry present the specific parameter on their left y-axis, and plots along the column of this diagonal entry present the same parameter on the x-axis. Each individual row and column have the same axis scaling. Parameters such as the Boolean distance are included if they are greater than a threshold, indicated by $\wedge$, and parameters such as the remaining loss terms are included only if they are below a threshold, indicated by $\vee$.

Markers within the plots represent the combination of

both parameters, which result in the highest recall at 100% precision. The results are determined by fixing parameter values on the x-axis and increasing the parameter value corresponding to the y-axis to the last point where 100% precision is given. Because the parameter for the y-axis does not always pose a constraint in combination with the parameter of the x-axis, triangle markers ▲ are used to indicate the case where no upper limit exists. Fields marked in red indicate that no combination of the respective two parameters allow 100% precision. Outliers can be visually identified by cut-offs, causing longer straight lines, at higher value ranges either on the upper or lower triangular of the full plot matrix.

The purpose of this plot matrix is to identify parameter combinations of interest for further evaluation. Because the marker for parameter pairs do not provide any information about the corresponding recall value, data points ● for individual recall values ranging from 0-100 % according to the right y-axis are added.

We are using loss terms, residuals, and the Boolean distance for this metric. Loss terms close to zero represent the most accurate results, and high loss terms indicate that particular properties and constraints could not be retained. The opposite holds for the Boolean distance. The bigger the distance, the more certain the decision is. Generally, we require thresholds that are as constraining as necessary but as inclusive as possible to retain a high recall.

The matrix plot shows that there is no meaningful combination between the loss-terms $\mathcal{L}_{\mathrm{norm}}$, $\mathcal{L}_{\mathrm{pos}}$, and $\mathcal{L}_{\mathrm{vec}}$ because there is no parameter combination to produce a valid result at 100% precision. Outliers can be visually inferred for $\Delta b = 0.65$, $\mathcal{L}_{\mathrm{std}} = 1.49$, $\mathcal{L}_{\mathrm{norm}} = 0.2$, and $f_{\mathrm{sys}} = 8.57$. The matrix plot indicates that the most meaningful parameters for outlier rejection and maximization of the recall at 100% precision are $\mathcal{L}_{\mathrm{std}}$ and $\mathcal{L}_{\mathrm{norm}}$ because it shows the highest recall at 65%.

We can also see that $\mathcal{L}_{\mathrm{std}}$ together with Δb indicates a more dynamic parameter range that can allow for further improvement. Thus, we are applying the first criteria to the individual results ($\mathcal{L}_{\mathrm{norm}} < 0.2$) and generate precision/recall graphs for Δb, depending on $\mathcal{L}_{\mathrm{std}}$ to further detail their effect on the recall after the precursor was applied.

Figure 13 shows the precision/recall graph for dedicated $\mathcal{L}_{\mathrm{std}}$ thresholds while varying the Boolean distance threshold. It can be seen that a strong limitation of $\mathcal{L}_{\mathrm{std}} < 2.6$ eliminates all incorrect choices at 100% precision, but at the cost of a low recall due to unnecessarily removed true samples. The best recall (91%) while maintaining 100% precision is reached for $\mathcal{L}_{\mathrm{std}} > 4.1$ and $\Delta b = 0.31$.

Figure 14 shows the same data for individual Boolean distances but varying $\mathcal{L}_{\mathrm{std}}$. A recall plateau is reached for higher STD loss thresholds. Please note that $\mathcal{L}_{\mathrm{std}}$ does not decrease the Boolean threshold further. However, it reduces the number of samples in the false negative and true negative categories, thus allowing for an increased recall and a redundant multi-parameter method, providing a robust health metric. The right box plot of Figure 10 shows the statistic for the Boolean distances with the full pre-filtering of samples applied $\mathcal{L}_{\mathrm{norm}} > 0.2, \mathcal{L}_{\mathrm{std}} > 4.1, \Delta b = 0.31$. The green

TABLE VI: Comparison of final calibration and reference states for the position sensor module. The table shows absolute values (e.g. $\mathbf{p}_{\mathrm{ip}}$) and individual errors w.r.t. the true calibration (e.g. $\tilde{\mathbf{p}}_{\mathrm{ip}}$), and the corresponding norm. The evaluation presents the average result based on 10 different trajectories. The comparison shows four cases for cross-evaluation where a reference frame was required or not, and the loss term $\mathcal{L}_{\mathrm{ref}}$ was applied or not. In the case of the reference frames, applying the additional loss provides a more reliable basis for the decision if a reference frame needs to be modeled. $\boldsymbol{\omega}$ represents the rotation in the tangent space and radians and $\|\boldsymbol{\omega}\|$ is the angle error.

	w/o		w/$\mathcal{L}_{\mathrm{ref}}$		w/o		w/$\mathcal{L}_{\mathrm{ref}}$		w/o		w/$\mathcal{L}_{\mathrm{ref}}$	
	$\mathbf{p}_{\mathrm{ip}}$	$\tilde{\mathbf{p}}_{\mathrm{ip}}$	$\mathbf{p}_{\mathrm{ip}}$	$\tilde{\mathbf{p}}_{\mathrm{ip}}$	$\mathbf{p}_{\mathrm{rw}}$	$\tilde{\mathbf{p}}_{\mathrm{rw}}$	$\mathbf{p}_{\mathrm{rw}}$	$\tilde{\mathbf{p}}_{\mathrm{rw}}$	$\boldsymbol{\omega}_{\mathrm{rw}}$	$\tilde{\boldsymbol{\omega}}_{\mathrm{rw}}$	$\boldsymbol{\omega}_{\mathrm{rw}}$	$\tilde{\boldsymbol{\omega}}_{\mathrm{rw}}$
No reference frame required												
x	0.298	0.027	0.300	0.013	0.017	0.057	9e-6	3e-5	0.013	0.025	9e-4	0.002
y	0.497	0.022	0.494	0.014	0.033	0.090	8e-7	2e-6	0.007	0.028	7e-5	0.004
z	1.009	0.034	1.000	0.022	0.028	0.071	9e-8	7e-8	0.003	0.025	0.002	0.005
$\|.\|$	1.164	0.048	1.155	**0.029**	0.047	0.128	1e-7	**3e-5**	0.015	0.045	0.002	**0.008**
With reference frame required												
x	0.294	0.026	0.315	0.025	9.998	0.066	9.915	0.109	0.014	0.038	0.003	0.008
y	0.503	0.026	0.502	0.023	0.002	0.098	0.002	0.005	0.880	0.038	0.879	0.023
z	0.981	0.064	1.032	0.062	0.032	0.121	0.003	0.012	0.003	0.037	0.002	0.013
$\|.\|$	1.141	0.074	1.190	**0.070**	9.998	0.169	9.915	**0.110**	0.880	0.065	0.879	**0.029**

area marks the improvement of the decision range for correct choices, providing zero false positives and corresponding recalls according to Figures 13 and 14.

The presented health metric allows for the detection of all false choices with zero false positives. Should a selection be identified as incorrect, the procedure for stage 1 can be re-run with modified initial values to reach a correct choice within the optimization landscape. Therefore, the presented method can be used with raw data in contrast to [7] and [10], which requires a pre-processing of data and possible loss of information to improve noise resilience.

C. Reference Frame Determination and Calibration States

We will demonstrate the benefit of the reference frame determination based on the position sensor module. Calibration states are defined by the IMU to sensor translation $\mathbf{p}_{\mathrm{is}} = [0.3, 0.5, 1.0]$ and, in case one is given, the reference frame $\mathbf{p}_{\mathrm{rw}} = [10, 0, 0]$ and $\boldsymbol{\omega}_{\mathrm{rw}} = [0, 0.8727, 0]$ corresponding to $[0, 50, 0]$ degree in Euler angles.

The evaluation is done based on four scenarios, with 10 different trajectories and numerically illustrated by Table VI. Two scenarios in which no reference frame is required, once without (w/o) additional constraint and once with $\mathcal{L}_{\mathrm{ref}}$ applied (w/$\mathcal{L}_{\mathrm{ref}}$). The two other scenarios, in contrast do require a reference frame and are also evaluated once without reference frame constraint and once with $\mathcal{L}_{\mathrm{ref}}$.

Table VI shows the improvement in the cases where the additional loss term $\mathcal{L}_{\mathrm{ref}}$ was applied. Refining the reference frame also positively affects the accuracy of the calibration $\mathbf{p}_{\mathrm{is}}$. In case a reference frame exists, the accuracy of this frame is improved. In case no reference frame is required, the basis for this decision becomes more clear due to a marginal remaining norm of $3 \cdot 10^{-5}$.

The error of the final calibration states is evaluated based on the same 10 trajectories as for the reference frame de-

TABLE VII: Average extrinsic calibration errors for all sensor modules based on 10 trajectories. ω represents the rotation in the tangent space in radians and $\|\omega\|$ is the angle error.

	Pos.	Inv. Pos.		Vel.	Body Vel.		Mag.	
	$\tilde{\mathbf{p}}_{ip}$	$\tilde{\mathbf{p}}_{ip}$	$\tilde{\omega}_{ip}$	$\tilde{\mathbf{p}}_{iv}$	$\tilde{\mathbf{p}}_{iv}$	$\tilde{\omega}_{iv}$	$\tilde{\mathbf{m}}_{w}$	$\tilde{\omega}_{im}$
x	0.026	0.076	0.016	0.002	0.008	0.004	0.014	0.023
y	0.021	0.062	0.024	0.004	0.006	0.006	0.011	0.008
z	0.033	0.141	0.012	0.003	0.012	0.009	0.009	0.031
$\|.\|$	0.048	0.172	0.031	0.005	0.013	0.012	0.019	0.039

termination. Table VII shows the averaged errors for the individual calibrations. The absolute calibration errors show sufficient accuracy for the given noise parameters and thus, the initial guess of the calibration states are beneficial for the initialization of e.g., an EKF.

D. Real-World Application (UAV)

The previous sections used simulated noise-affected data with dynamic trajectories to demonstrate and evaluate the presented method on sensor model identification. The following experiment is intended to show the application of this approach to real-world data. For this purpose, we use a larger-scale outdoor-to-indoor transition trajectory from the INSANE datasets [16]. The dataset used a quadcopter that was designed for non-agile flight maneuvers and therefore, the flight segment of the trajectory has only little dynamic movement and no noteworthy orientation excitation. However, the trajectory does have an initial segment where the quadcopter was excited by hand in 6-DoF, and then placed on the ground for a consecutive flight.

The objective is to detect a global velocity sensor, representing a vector element, and a magnetometer as a rotational component. The magnetometer (@80 Hz) is mounted close to the center of the vehicle. The ground truth calibration, given by the dataset, was performed with dedicated rotational excitation. The velocity measurements are provided by a GNSS sensor (@8 Hz), which is mounted off-center on a larger beam at a distance of 0.56 m from the IMU. The localization information of the vehicle is generated, using the MaRS framework with two GNSS sensors as an input. The sensor data for the gray-box identification remained unprocessed and is directly passed to the identification approach.

We chose to perform the experiment for the magnetometer and velocity sensor identification for three segments. One
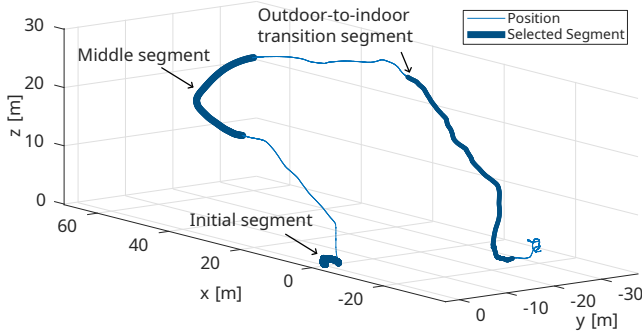

Fig. 15: 3D position of the full trajectory, estimated with the MaRS framework. The bold highlighted sections are the initial and middle segments used for the sensor model identification experiment.
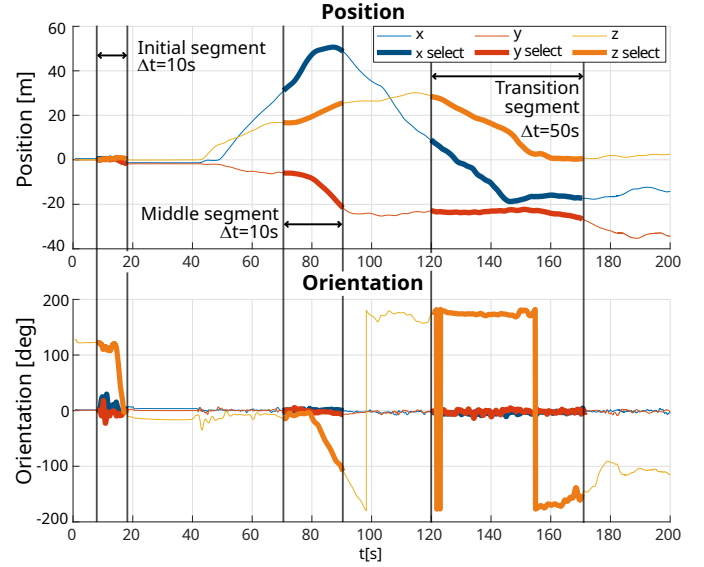

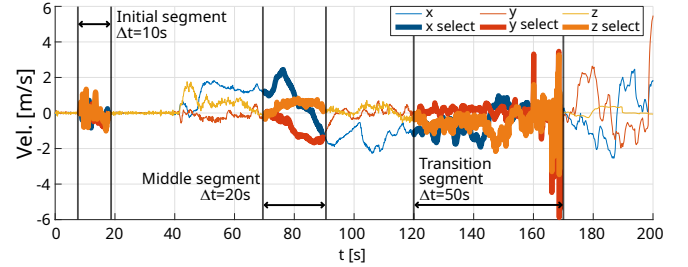Fig. 16: 2D position and rotation estimates of the trajectory shown by Figure 15.

Fig. 17: Unprocessed GNSS velocity measurements with the highlighted section, which are used for the sensor model identification experiment.

for the initial phase, where the vehicle was excited by hand, representing a realistic scenario where an unknown sensor model needs to be identified and initialized on the ground, a second segment, where a sensor model needs to be identified mid-flight, and a more difficult third outdoor-to-indoor transition segment for the velocity model, where the velocity signal degrades towards the end of the trajectory. The graphs in Figure 15 and 16 show the estimated position and orientation for the full trajectory where the selected segments are highlighted. The initial segment is 10 s long (8 s-18 s), with 870 samples for the magnetometer and 80 samples for the velocity sensor. The middle segment is 20 s long (70 s-90 s), with 1756 magnetometer and 160 velocity samples, and the transition segment is 50 s long (120 s-170 s) with 400 velocity samples. The middle segment is chosen to be longer and for a period where a rotation in yaw occurred because we need to increase the information given by the unknown signal, given the moderate flight pattern. As mentioned before, for real-world scenarios, the addition of sensors mid-flight could benefit from intentionally performing short maneuvers that increase the quality of observability for improved detection [17].

Figure 17 and 18 show the raw magnetometer and velocity measurements with the highlighted test segments. The estimated core state and the unknown sensor signal samples are synchronized to the sensor timestamps and used as input to
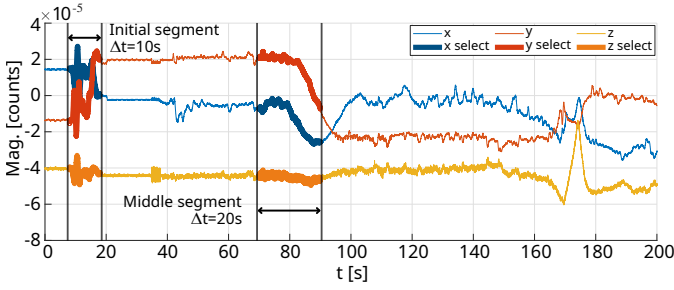
Fig. 18: Unprocessed magnetometer sensor measurements in counts. The highlighted section is used for the sensor model identification experiment.

Soft Boolean	Mag.		Vel.		
	Init.	Mid.	Init.	Mid.	Transit.
Selector	0.971	1.007	0.852	0.998	0.812
Dist. to second	0.945	0.982	0.772	0.998	0.626

TABLE VIII: Value of the soft Boolean selectors and the distance to the second highest boolean for both sensors and trajectory segments. The boolean distance is a measure for the accuracy of the selection as described in Section V-B.

the sensor model identification approach.

The sensor model identification was successful in all four cases. The evaluation of the experiments is done based on a number of parameters: The final soft Booleans for the selection, the error of the calibration states, and the RMSE for the sensor measurement compared to the resembled measurement based on the parameterized sensor model.

The final boolean selector and the distance to the second highest boolean, shown by Table VIII, are unambiguous and true-positives according to the health metric (see Sec. V-B). Tables IX and X show the estimated calibration state of the two sensors after stage two of the identification process (see Fig. 2). The calibration for the magnetometer shows minor errors for the estimation of the rotation for the rigid-body calibration $\mathbf{R}_{im}$, which has an angle error of $2.953°$ for both, the initial and middle part of the trajectory. However, the declination of the estimated magnetic field expressed in the world frame $\mathbf{m}_w$ has a noticeable error of $15°$, which is due to only minimal excitation of the roll and pitch angle for both

$\mathbf{R}_{im}$ [deg]	True	Init.		Mid.	
		Est.	Error	Est.	Error
x	0.260	0	0.260	0	0.260
y	2.264	0	2.264	0	2.264
z	-1.882	0	-1.883	0	-1.882
$\mathbf{m}_w$ [deg]					
Incl.	85.178	81.747	-3.431	92.290	7.112
Decl.	-46.330	-62.471	-16.141	-62.016	-15.686

TABLE IX: Comparison of the true magnetic field vector expressed in the world frame $\mathbf{m}_w$ and the rigid-body rotation $\mathbf{R}_{im}$ to the estimated results, using the parameterized sensor model, identified by the sensor model identification approach.

$\mathbf{p}_{iv}$ [m]	True	Init.		Mid.		Transit.	
		Est.	Error	Est.	Error	Est.	Error
x	0.35	0.356	-0.006	0.449	-0.099	0.866	0.516
y	0.41	0.420	-0.010	0.697	-0.287	-1.775	2.185
z	0	0.212	-0.212	0.288	-0.288	1.065	1.065

TABLE X: Estimated and measured calibration of the rigid body translation between the IMU and the velocity sensor $\mathbf{p}_{iv}$. The error of the z-component is slightly higher due to the zero z-component of the true calibration and an insufficient orientation excitation around the directional vector from the IMU to the sensor frame.

RMSE	Init.	Mid.	Init.	Mid.	Transit.
x	1.365	2.249	0.183	0.098	0.295
y	1.334	2.004	0.162	0.089	0.595
z	0.979	1.286	0.183	0.092	0.529

TABLE XI: The table shows the RMSE for the comparison of the raw magnetometer and velocity measurements and the calculated measurement estimates. Measurement estimates are generated based on the identified sensor model, the parameterization, and given core states. The RMSE is shown for the initial and middle segments of the trajectory.

data segments.

The rigid body translation between the IMU sensor and the velocity sensor $\mathbf{p}_{iv}$ also shows plausible error margins. In general, the error for the z-axis is higher ($\sim 20\,\mathrm{cm}$) for both data segments. This is due to the fact that the true calibration of the sensor is negligible for its z-component. Because of this, a high angular velocity around a directional vector from the IMU to the sensor would be required to correctly estimate this parameter [18]. Yet, the high leaver-arm of $0.56\,\mathrm{m}$ of the x and y-component allows for well-observable x and y-calibration, shown by the low error for the initial phase. On the contrary, the y-component for the mid-flight data segment shows a higher error due to the lack of roll and pitch excitation during flight. Despite the small errors for the calibration states, the overall RMSE, shown by Table XI for the individual data segments, is reasonable.

As mentioned in the introduction, the sensor model identification approach is meant to automate the process of integrating a sensor into a state estimation framework, possibly to the extent where this is done online. Thus, in the following, we are using the MaRS framework, which utilizes two GNSS sensors for the estimate of the core states during the whole sequence of the trajectory, and introduce the previously identified sensor models, in two separate experiments, with the initial guess from the calibration stage at $t = 90\,\mathrm{s}$.

The result for the integration of the magnetometer sensor is shown by Figure 19. This graph illustrates the convergence of the magnetic field vector $\mathbf{m}_w$ and rigid-body rotation $\mathbf{R}_{im}$. It can be seen that both calibration states converge towards the true value. This is an improvement compared to the initial guess but not a convergence as might be expected. There are two reasons for this slight discrepancy. One is the
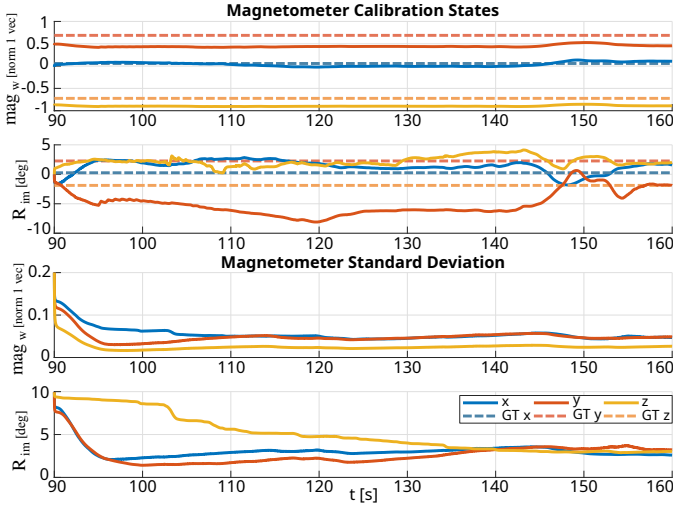
Fig. 19: The graphs show the convergence of the magnetometer calibration states: the reference of the magnetic field expressed in the world frame $\mathbf{mag}_w$, and the rigid body magnetometer sensor orientation with respect to the IMU sensor $\mathbf{R}_{im}$. The magnetometer is added to the MaRS framework during runtime at $t = 90\,\text{s}$ with the initial guess of the sensor model identification approach. This plot shows that the calibration states converge toward the correct values despite an initial estimate with a slightly higher error, as shown by Table IX. This scenario is particularly difficult because the calibration states require more agile rotational movement to converge correctly, and the given trajectory only has moderate movement.
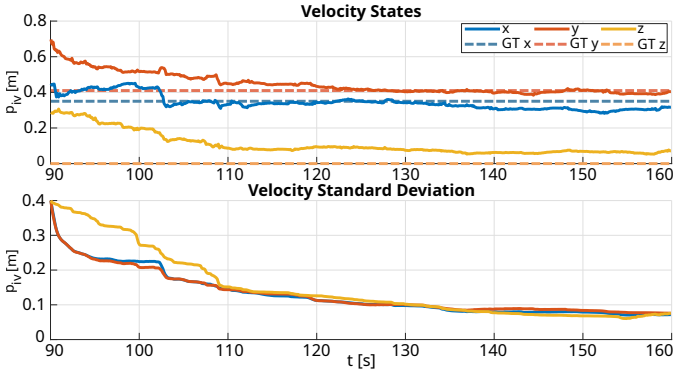

Fig. 20: Convergence of the rigid body velocity calibration states with the position of the sensor with respect to the IMU frame $\mathbf{p}_{iv}$. The sensor is added to the MaRS framework during runtime at $t = 90\,\text{s}$ with the initial guess of the sensor model identification approach. This plot illustrates that the calibration states converge toward the correct values despite a higher error for the initial guess, as shown by Table X.

lack of orientational excitation, which limits the quality of convergence. The second is more predominant and due to the intrinsic calibration of the magnetometer sensor. It needs to be noted that the normalized measurements of a magnetic field vector lie on a unit sphere; however, magnetometer suffer from hard- and soft-iron errors which cause an offset and distortion of this sphere and require dedicated compensation [19]. This compensation is done for the generation of the ground truth calibration of the sensor, but not for the measurement input

to the sensor model identification approach. This is done to maintain a fair comparison because the input measurement is meant to be unknown and unprocessed.

The result of integrating the velocity sensor is shown by Figure 20. The distinct changes in velocity (see Fig. 17) result in a clean convergence of the rigid body calibration $\mathbf{p}_{iv}$, which compensates for the small error of the initial guess given by the sensor model identification approach.

Overall, the given sensor types have been reliably detected in both real-world cases. The boolean selectors show a high value, which indicates a low ambiguity in the selection process despite the fact that the data was prone to observability issues for both sensor types. The estimate for the initial guess of the sensor calibrations showed small errors after completing the sensor parameterization stage. However, the consecutive integration of the parameterized sensor models to a state estimator during runtime showed that the self-calibration of the sensor states further improve the sensor's rigid body calibrations.

This method can be improved by using observability-aware motion generation [20], either during the time of the sensor model identification or the later refinement of the calibration states by the recursive filter. However, this is out of the scope of the presented experiment.

The previous experiments showed an example with undisturbed real-world sensor data. The following experiment will show that the identification approach can also identify a model if real-world data with an increased noise profile is given. For this case, we are using the same dataset and choose the outdoor-to-indoor transition sequence between 120-170 s to identify a velocity sensor.

During this time, the vehicle is moving along the front of a building and enters the building at the end. Thus, the raw velocity measurement (see Fig. 17) shows a higher noise due to shadow-casting and multi-path effects (urban canyoning) and distinct outliers at the end of this sequence because the GNSS signal is dropping out. Therefore, one part of the given measurement sequence is ill-conditioned and the second part is becoming invalid.

Please note that the estimate of the vehicle state is based on a multi-sensor and sensor switching setup including GNSS, VIO, Magnetometer, and an indoor motion capturing system. The interested reader is referred to [16] for the detailed setup of the filter. Thus, the estimate of the vehicle localization can be assumed to be correct for the entire period and only the GNSS data stream becomes invalid at the end.

Table VIII shows that the velocity sensor model is correctly identified, confirmed by the boolean selector. Compared to the results for the initial and middle segment, Table X shows an increased error for the estimation of the calibration states for the transition scenario. This is due to the higher noise floor and lack of orientation, leading to poorly observable calibration states. Table XI also shows an increased RMSE due to the significantly higher noise of the measurement signal that we are comparing against. However, the correct sensor model identification shows that the proposed approach also works with realistically ill-conditioned real-world data. Figure 21 shows the corresponding overlay of the ground truth for the
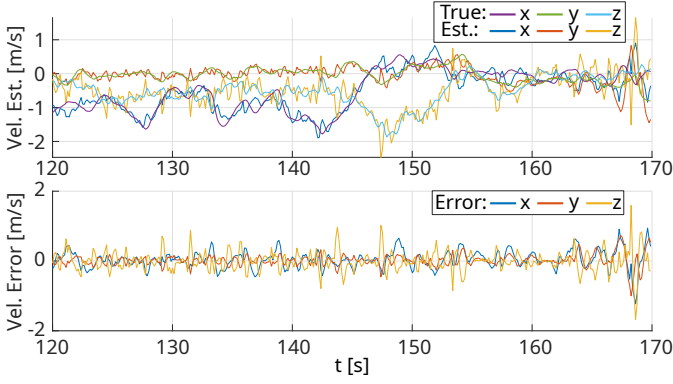
Fig. 21: Reprojection of the correctly identified velocity sensor model and its ground truth as well as the corresponding error plot. Despite the fact that the given real-world measurement sequence has increased noise and partially invalid measurements at the end, the resembled data generally matches the true signal. However, the plots show jitters due to the increased error for the estimated calibration states, which is a result of the erroneous real-world measurement sequence.

measurement signal, and the estimate, based on the identified sensor model and parameters for the calibration states as well as the error over time.

E. Real-World Application (Ground Vehicle)

While the previous section demonstrates the approach to sensor model identification for a UAV use case, this section presents an experiment for a ground vehicle. We are using the same filtering framework (MaRS [1]), which uses an IMU as the primary sensor and therefore renders the state dynamics agnostic to the robotics platform. Thus, the estimates of both vehicle localizations scenarios are similar in characteristics. However, the importance of this experiment lies in the profile and dynamic of the trajectories for the robotic vehicle itself. In this case, the trajectory is moderate in excitation and bound to the ground plane which poses a significant aspect for the validation of the proposed algorithm.

We are using sensor data from a car that drove monotonously (i.e., cruise control) with an average speed of $55\,\mathrm{km/h}$ along a trajectory of $64\,\mathrm{km}$ with mainly straight elements and three dedicated turning points as shown by Figure 22. The car was equipped with an IMU (@$200\,\mathrm{Hz}$), GNSS (@$5\,\mathrm{Hz}$) measuring position and velocity, and a magnetometer (@$40\,\mathrm{Hz}$). The IMU and GNSS information is used for the base of the localization algorithm, using the MaRS framework, which provides the system input to the sensor model identification approach.

For the overall experiment, we are using the magnetometer measurements of the highlighted $\Delta t = 60\,\mathrm{s}$ trajectory segments for the sensor model identification. The estimated position and rotation of the trajectory is highlighted in Figure 23, with corresponding magnetometer measurements shown by Figure 24. All segments have rotational information in yaw and negligible excitation in the roll and pitch direction since the vehicle is bound to the ground and the experiment area was flat. Segment one contains a short turn of $45°$, segments
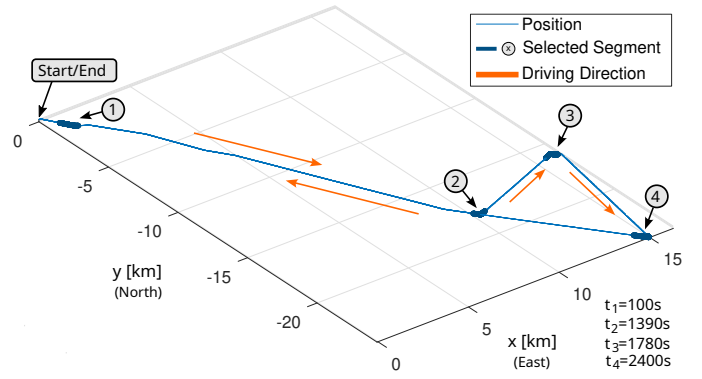

Fig. 22: 3D trajectory of the ground vehicle position estimated with the MaRS framework. Bold segments highlight the four evaluation sequences for the sensor model identification.
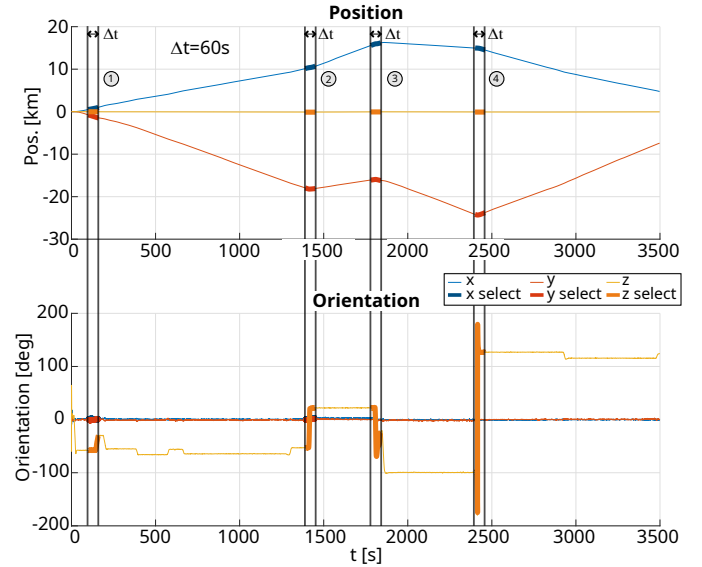

Fig. 23: 2D position and orientation of the estimated ground vehicle trajectory with highlighted sections corresponding to sequences in Figure 22.
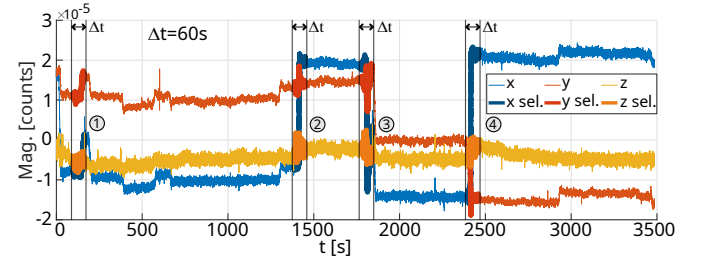

Fig. 24: Raw magnetometer sensor measurement with highlighted sections used for the sensor model identification approach.

two and three have a turn of close to $90°$, and segment four has a turn of $120°$.

These segments are intentionally chosen because they contain rotational information. This is justified by the earlier statement that the properties of the trajectory need to render the states of the sensor observable, which is given by the orientational excitement in the turning points of the trajectory. Otherwise, the self-calibration state $\mathbf{m_w}$ shown in Equation (9)

Soft Boolean	Seq. 1	Seq. 2	Seq. 3	Seq. 4
Selector	1.149	0.684	0.947	0.902
Dist. to second	1.029	0.527	0.974	0.681

TABLE XII: The table shows that the magnetometer sensor model is correctly detected for all four sequences. The magnitude of the soft Boolean selectors show a clear selection, and the distance to the second highest boolean is above the threshold of $\Delta b > 0.31$, which is a measure for the accuracy of the selection and confirms that the selections are not false positives (see Sec. V-B).

		Error of Estimate			
$\mathbf{R}_{im}$ [deg]	True	Seq. 1	Seq. 2	Seq. 3	Seq. 4
x	4	4	4	4	4
y	17	17	17	17	17
z	-1	-1	-1	-1	-1
$\mathbf{m}_w$ [deg]					
Incl.	-16.5	-2.5	12.0	9.8	10.0
Decl.	75.5	-6.2	-14.6	-11.6	21.2

TABLE XIII: Comparison of the true magnetic field vector expressed in the world frame $\mathbf{m}_w$ and the rigid-body rotation $\mathbf{R}_{im}$, to the identified sensor model and estimated calibration states. Estimates of $\mathbf{R}_{im}$ are close to zero, which is due to the minimal orientational excitation of the vehicle and low quality of observability.

	Mag. [μ counts]			
RMSE	Seq. 1	Seq. 2	Seq. 3	Seq. 4
x	1.196	3.531	3.176	4.141
y	1.187	3.513	3.005	4.051
z	0.681	1.213	1.615	1.154

TABLE XIV: The table shows the RMSE for the comparison of the raw magnetometer measurements vs. the calculated measurement estimates, based on Equation 9 and the parameterization determined by the identification process.

is unobservable. We refer the interested reader to previous work in [15] for a detailed analysis.

The results show that the sensor model identification approach correctly identified the magnetometer model based on the given measurement data for all four experiment sequences. The correct identification is validated by table XII showing that the soft Boolean selector provides a clear decision with an unambiguous distance to the second soft boolean.

Please keep in mind that this particular example is meant to show that a sensor model, such as the magnetometer, can also be identified with insufficient rotational excitation in all three axes, posing a low quality in observability of the calibration states. While the identification of the model was correct for the presented experiments, the lack of roll and pitch excitation did not allow to clearly restrain the relation between $\mathbf{R}_{im}$ and $\mathbf{mag}_w$ causing higher errors for the estimated calibration states. This is shown by table XIII and XIV respectively.

The estimates of $\mathbf{mag}_w$ and $\mathbf{R}_{im}$ show an error in the range of $20°$, which may be acceptable for the initialization of a filter if the remaining trajectory would provide sufficient

orientations to allow the convergence of the calibration states. However, since the properties of the trajectory are known by design (i.e. a car on flat grounds), and the requirements on observability for the magnetometer model are known, a high-level decision can be made to only include the sensor upon rotational excitement.

VI. Future Work

In terms of future work, we showed that it is possible to identify sensors based on an extended catalog of known sensor models. This could be extended by future work to explore a more granular approach. Lessons learned while developing the current approach are, that more granular selectors for sensor model components quickly lead to model over-determination and non-applicable sensor models due to observability constraints caused by shared observability of co-dependent states. We evaluated a system setup where each sensor state was defined as a shared state between models, but the results were limited in terms of robustness. In addition, if a sensor model has additive components that can be obscured by the noise floor (SNR), then detection against other model hypotheses is difficult and not robust because they are not distinguishable. While this indicates that the system definition can not be over-simplified, future research could further isolate components of the measurement information to be more decoupled from known measurement models, and design models that include latent, non-intuitive states. The presented approach can also be extended and adopted to address challenges in fields outside of the robotics community e.g. neuroscience, bio-engineering, or logistics.

VII. Conclusion

The introduced approach allows us to infer a sensor model by providing the current state of a vehicle and a corresponding series of sensor measurements. The selection can be validated by the introduced health metric, defined and validated by an extensive cross-examination of final optimization losses and selection parameters, utilizing a scatter-plot matrix to maximize the recall of the metric at a $100\,\%$ precision. The approach also allows us to reliably decide if a reference frame for sensor measurements is required or not. As a result, the removal or introduction of a reference frame as a possible calibration state showed improved accuracy for the remaining calibration parameters in both cases. The inference of an initial guess for state calibrations of the formally unknown measurement signal is sufficiently accurate for the initialization of a localization algorithm, which was proven through the application in a real-world scenario. We did not identify any severe shortcomings. The exposed failure modes only occur in scenarios in which the measurement noise of individual components is unrealistically high.

This method is helpful for inexperienced users who need to identify the source and type of measurement, sensor calibrations, or sensor reference frames. It will also be important for the automated integration of sensor modalities in the field of modular multi-agent scenarios and modularized robotic platforms that are augmented by sensor modalities during runtime, ensuring a more accurate and robust integration of

new sensor elements. Overall, this approach aims to provide a simplified integration of sensor modalities to downstream applications and circumvent common pitfalls in the usage and development of localization approaches.

REFERENCES

[1] C. Brommer, R. Jung, J. Steinbrener, and S. Weiss, "MaRS: A Modular and Robust Sensor-Fusion Framework," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 359–366, Apr. 2021.

[2] T. Mitterer and H. Zangl, "Beyond pure sensing: IEEE 21450 in digitalization of the development cycle of smart transducers," *IEEE Instrumentation & Measurement Magazine*, vol. 23, no. 2, pp. 47–52, Apr. 2020.

[3] E. Y. Nagai and L. V. R. D. Arruda, "Soft Sensor Based on Fuzzy Model Identification," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 127–132, 2005.

[4] J. Frolik, M. Abdelrahman, and P. Kandasamy, "A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 6, pp. 1761–1769, Dec. 2001.

[5] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, Apr. 2004.

[7] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[8] J.-C. Loiseau, B. R. Noack, and S. L. Brunton, "Sparse reduced-order modelling: Sensor-based dynamics to full-state estimation," *Journal of Fluid Mechanics*, vol. 844, pp. 459–490, Jun. 2018.

[9] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2204, Aug. 2017.

[10] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, "A Unified Framework for Sparse Relaxed Regularized Regression: SR3," *IEEE Access*, vol. 7, pp. 1404–1423, 2019.

[11] J. B. Kadane and N. A. Lazar, "Methods and Criteria for Model Selection," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 279–290, Mar. 2004.

[12] A. Khosbayar, J. Valluru, and B. Huang, "Adaptive inference for Bayesian network soft-sensor in the presence of process and sensor drift," *The Canadian Journal of Chemical Engineering*, vol. 100, no. 9, pp. 2119–2134, Sep. 2022.

[13] A. Lesage-Landry, J. A. Taylor, and D. S. Callaway, "Online Convex Optimization With Binary Constraints," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 6164–6170, Dec. 2021.

[14] H. Xiong, M. Yu, L. Liu, F. Zhu, J. Qin, F. Shen, and L. Shao, "A Generalized Method for Binary Optimization: Convergence Analysis and Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[15] C. Brommer, C. Bohm, J. Steinbrener, R. Brockers, and S. Weiss, "Improved State Estimation in Distorted Magnetic Fields," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. Athens, Greece: IEEE, Sep. 2020, pp. 1007–1013.

[16] C. Brommer, A. Fornasier, M. Scheiber, J. Delaune, R. Brockers, J. Steinbrener, and S. Weiss, "The INSANE dataset: Large number of sensors for challenging UAV flights in Mars analog, outdoor, and out-/indoor transition scenarios," *The International Journal of Robotics Research*, Feb. 2024.

[17] J. A. Preiss, K. Hausman, G. S. Sukhatme, and S. Weiss, "Simultaneous self-calibration and navigation using trajectory optimization," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1573–1594, Dec. 2018.

[18] X. Huang, C. Yang, and M. Wen, "A Sliding Window Filter with GNSS-State Constraint for RTK-Visual-Inertial Navigation," *IEEE Transactions on Robotics*, pp. 1–17, 2024.

[19] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira, "Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1293–1306, Apr. 2011.

[20] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-Aware Trajectory Optimization for Self-Calibration With Application to UAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, Jul. 2017.

BIOGRAPHIES

Christian Brommer received the Ph.D. degree in robotics from the University of Klagenfurt, at the Control of Networked Systems group, Austria, in 2024. He is currently a Postdoctoral Fellow with Oak Ridge Associated Universities at the DEVCOM Army Research Laboratory. From 2014 to 2018, he conducted research on UAV localization and autonomy at NASA's Jet Propulsion Laboratory. His research interests include modular multi-sensor fusion and autonomy for robotic vehicles and swarms.

Alessandro Fornasier received the Ph.D. degree in robotics from the University of Klagenfurt, at the Control of Networked Systems group, Austria, in 2024, where he focused on equivariant aided inertial state estimation and multi-sensor fusion. He is currently a senior Robotics Engineer with Hexagon Robotics. His research interests include aided inertial localization and mapping, sensor fusion, and spatial AI for autonomous systems.

Jan Steinbrener received the Ph.D. degree in physics from Stony Brook University, NY, USA, in 2010, where he worked on inverse methods in X-ray diffraction microscopy. He since had various positions in industry and academia and is currently Vice Rector of Research and International Affairs and an Associate Professor with the department of Smart Systems Technologies at the University of Klagenfurt, Austria. His research focuses on AI-based methods for control and navigation of robotic systems.

Stephan Weiss received the Ph.D. degree in robotics from ETH Zurich, Switzerland. He is currently a Full Professor at the University of Klagenfurt, Austria, where he leads the Control of Networked Systems Group. He was a Research Technologist with the mobility and robotic systems section at the NASA Jet Propulsion Laboratory where he took part in initiating the Ingenuity Mars helicopter project. His research interests include vision-based navigation, visual-inertial state estimation, and autonomous aerial vehicles.