Systems Article

Under the Sand: Navigation and Localization of a Micro Aerial Vehicle for Landmine Detection with Ground-Penetrating Synthetic Aperture Radar

Rik Bähnemann[1]ⓘ**, Nicholas Lawrance**[1]ⓘ**, Lucas Streichenberg**[1]ⓘ**, Jen Jen Chung**[1]ⓘ**, Michael Pantic**[1]ⓘ**, Alexander Grathwohl**[2]ⓘ**, Christian Waldschmidt**[2]ⓘ **and Roland Siegwart**[1]ⓘ

[1]Autonomous Systems Lab, ETH Zurich, Switzerland
[2]Institute of Microwave Engineering, Ulm University, Germany

Abstract: Ground-penetrating radar mounted on a micro aerial vehicle (MAV) is a promising tool to assist humanitarian landmine clearance. However, the quality of synthetic aperture radar images depends on accurate and precise motion estimation of the radar antennas as well as generating informative viewpoints with the MAV. This paper presents a complete and automatic airborne ground-penetrating synthetic aperture radar (GPSAR) system. The system consists of a spatially calibrated and temporally synchronized industrial grade sensor suite that enables navigation above ground level, radar imaging, and optical imaging. A custom mission planning framework allows generation and automatic execution of stripmap and circular GPSAR trajectories controlled above ground level as well as aerial imaging survey flights. A factor graph based state estimator fuses measurements from dual receiver real-time kinematic (RTK) global navigation satellite system (GNSS) and an inertial measurement unit (IMU) to obtain precise, high-rate platform positions and orientations. Ground truth experiments showed sensor timing as accurate as $0.8\,\mu s$ and as precise as $0.1\,\mu s$ with localization rates of $1\,kHz$. The dual position factor formulation improves online localization accuracy up to $40\,\%$ and batch localization accuracy up to $59\,\%$ compared to a single position factor with uncertain heading initialization. Our field trials validated a localization accuracy and precision that enables coherent radar measurement addition and detection of radar targets buried in sand. This validates the potential as an aerial landmine detection system.

Keywords: aerial robotics, navigation, position estimation, sensors

---

Figure 1. The DJI M600 Pro (a) equipped with a custom GPSAR sensor payload (b) for landmine detection.

1. Introduction

Antipersonnel landmines are a massive obstacle in the pursuit of well-being in more than 55 affected states and regions. Not only do they cause more than 4000 civilian casualties per year but also leave thousands of hectares of land uninhabitable. The financial support for clearance has declined in the last two years, even though ongoing conflicts have continued to cause additional contamination (International Campaign to Ban Landmines, 2020). Recently, a novel detection method is being investigated to survey suspected hazardous areas (SHAs) and eventually assist and accelerate demining. This technology combines the mobility and ubiquity of MAVs, such as the MAV in Figure 1, with the ground-penetrating capability of radar imaging (García-Fernández et al., 2020b). In comparison to ground-based surveying and demining methods, such as metal detectors, dogs, and mine plows, MAVs can access any terrain from a safe distance and nondestructively search the area. Our research shows that an MAV with a side-looking radar operating 2–4 m above ground with an along-track velocity of $1 \, \mathrm{m \, s^{-1}}$ could survey without further modification at least $500 \, \mathrm{m^2 \, h^{-1}}$. This high area throughput would allow demining operations to quickly narrow down larger SHAs so that limited clearance resources can be deployed more effectively (United Nations Mine Action Service (UNMAS), 2019).

While previous publications develop the radar imaging process on MAVs, this paper focuses on the autonomous navigation and localization of such a system. Both capabilities are prerequisite to generating high-quality GPSAR images. This research describes a completely integrated system where the user is able to specify a georeferenced GPSAR mission that the MAV then autonomously executes. After landing, the MAV automatically computes precise radar antenna positions from synchronized GNSS and IMU measurements. These positions are input to the radar imaging process. The accompanying video[1] summarizes the system capabilities. In the short term, this system allows executing repeatable measurement campaigns to further develop GPSAR imaging. In the long term, automation will open the system to a broad range of users since complex system features are combined into simple operating elements. Besides the overall functionality, the system includes several extensions to the state of the art in sensor interfacing, state estimation, and trajectory generation that are described in this paper.

---

[1] Watch the system summary video: https://youtu.be/EfDhFKg5GEk

- A low cost, lightweight, temporally synchronized and spatially calibrated sensor unit. The timing synchronization has an accuracy of 0.8 μs and precision of 0.1 μs with respect to GNSS time and allows IMU rates of 1 kHz.
- An extension to the Georgia Tech Smoothing and Mapping Library (GTSAM) framework to fuse IMU with multiple GNSS receivers in a fixed-baseline configuration. The estimator includes explicit GNSS antenna phase center calibration. The modeling advances improve estimation accuracy by 59 % over single GNSS receiver fusion.
- Dynamically feasible linear and circular GPSAR trajectory generation to ensure informative, uniform radar data sampling.
- Providing all software components open-source listed in Table 2 at the end of the document.

We organize the remainder of the paper as follows. Section 2 sets our contributions in the context of related work. Section 3 defines the notation used throughout the paper to describe the kinematic relations of the system's sensors. Section 4 gives an overview of the GPSAR image formation process, and the implications this sensor modality has on the system design. Section 5 describes the system operation overview as well as the hardware setup. Section 6 summarizes the autonomous mission generation, which includes the GPSAR trajectory generation framework. The novel sensor time synchronization framework is described in Section 7. In Section 8, we present our localization algorithm that fuses IMU and GNSS measurements. Section 9 presents the system results which demonstrate the effectiveness of the navigation and localization pipeline and validate the detection of buried objects. Finally, we conclude the system development and discuss future research directions.

2. Related Work

Ground-penetrating radar (GPR) is a complementary sensor to electromagnetic induction (EMI) for landmine detection. In particular its ability to discriminate low metal content makes it an attractive asset (Daniels, 2009). Synthetic aperture radar processing further improves cross-range resolution of GPR enabling 3D imaging of centimeter-sized objects. GPSAR in side-looking or forward-looking geometry has been integrated onboard Unimogs and other military ground vehicles (Wang et al., 2008; Peichl et al., 2014). Aerial vehicles offer considerable advantages over ground vehicles, particularly for demining where there is a need to cover large, inaccessible areas and avoid interaction with the ground. The Minesweeper demonstrator project proposed the use of GPSAR mounted on a crewed inflatable airship (Cristoforato et al., 2000). The project even conducted field tests in Kosovo but publications, in particular, regarding radar results, have been sparse since then (Cramer, 2001). A first uncrewed aerial vehicle (UAV) was later proposed by (Moussally et al., 2004). The authors integrate a GPSAR payload on a Schiebel CAMCOPTER. Their GPSAR system provides GNSS-inertial state estimation and stripmap and circular data collection capabilities to enable buried object detection on U.S. Army test sites. This system was commercialized and is available today as part of (Mirage Systems Inc., 1984). However, no further developments have since been shared with the research community, and we have no information that this system is actively used in humanitarian mine action. We suspect that the platform they propose is still too large and expensive to be attractive for the humanitarian demining market. As of today their system operates at 150 m flight altitude, and a CAMCOPTER has more than 3 m rotor diameter and weighs 200 kg (Schiebel Corporation, 1997). This is not only a logistical hurdle but also makes it unsafe to inspect cluttered unstructured terrain at low altitude. However, such a system may still be applicable in a civilian radar imaging context, where this flight limitation is not as critical (Frey et al., 2021).

In the humanitarian demining context, today's ambition is to integrate GPSAR on even smaller, affordable, programmable, globally available, electrically powered MAVs. These vehicles can operate in inaccessible, cluttered environments which increases their application range significantly over larger vehicles (Fang et al., 2017). Furthermore, they are small and in total weigh approximately 10 kg which simplifies transport and offers safer deployment. Also the hardware cost of less than ten thousand USD accelerates research and makes application in humanitarian mine action more

realistic. Finally, open access research results offer the possibility of wider adoption and rapid, branched development of such systems around the world. However, the miniaturization also raises new questions. Smaller sensors have to be developed and integrated. Localization and imaging processes have to be adapted to the new sensors and low-altitude flight characteristics. And new navigation methods have to be developed to make full use of the precise, autonomous flight capabilities of MAVs for radar imaging. Multiple research groups tackle these challenges simultaneously (Fasano et al., 2017; Colorado et al., 2017; García-Fernández et al., 2018; Schartel et al., 2018; Schreiber et al., 2019; Esposito et al., 2020; Šipoš and Gleich, 2020; Bekar et al., 2021; Svedin et al., 2021).

(García-Fernández et al., 2018) have been the first to present a complete GPSAR system that can map buried objects with an MAV. Their system has a downward-looking radar and uses RTK GNSS to localize in the horizontal plane as well as light detection and ranging (LiDAR) for ground distance measurement. The second iteration of their system improves data collection, localization and radar imaging (García-Fernández et al., 2019). In particular, they update their GNSS and radar hardware, implement autonomous waypoint navigation, and introduce radar subsampling to cope with nonuniform and inaccurate in-flight data collection. The third iteration further reduces ground surface reflection by applying singular value decomposition filtering and deploys a dual-channel radar to reduce clutter (García-Fernández et al., 2020b). To the best of our knowledge, the system presented in our work is the only other complete MAV-GPSAR system that has shown repeated buried object detection (Heinzel et al., 2019; Schartel et al., 2020a; Grathwohl et al., 2021).

Our system has a different operation principle than the system presented by (García-Fernández et al., 2020b). The GPR is mounted in a side-looking geometry which reduces ground reflections but also has implications on the navigation. First, the ground distance, necessary for terrain correction in the GPSAR process, cannot be measured with a downward-facing LiDAR. Instead, we implement a two-step approach, where we first perform an optical survey flight to create a digital surface model (DSM) and then fly the georeferenced GPSAR mission. Alternatively, the terrain model can be determined in the same flight with interferometric synthetic aperture radar (InSAR) (Burr et al., 2021). Second, the side-looking view point geometry changes the data collection. A downward-looking system has a similar field of view as nadir camera or LiDAR setups. It can thus utilize existing implementations of autonomous waypoint control and coverage path planning to collect radar measurements. On the contrary, the flight path for a side-looking geometry is horizontally offset from the surface patch that is being investigated. Thus special flight paths are necessary. Our system implements circular and stripmap trajectories that can be combined in an arbitrary fashion to illuminate the desired surface, increase the synthetic aperture length, and provide different view points. Despite these two key differences in the operation principle, we also approach localization and navigation differently.

One main requirement for GPSAR imaging is high-precision localization of the radar antennas in flight. In previous publications, we demonstrated repeated detections of buried landmines when we fused total station theodolite position measurements with high-rate IMU measurements (Heinzel et al., 2019; Schartel et al., 2020a). In this paper, we replace the total station with RTK GNSS as it has better timing characteristics, operates without direct line-of-sight, delivers georeferenced positioning and is more cost-efficient. Most importantly, it has proven to deliver similar quality radar imaging results over multiple measurement campaigns in open fields (Schartel, 2021). The general understanding is that GNSS is prone to errors due to multi-path effects, atmospheric interference and satellite occlusion in complex environments such as cities, forests, and mountains. With the more recent introduction of alternative GNSSs such as GLONASS, BeiDou, and Galileo, accuracy and coverage have increased notably (Li et al., 2015). Furthermore, various developments on antennas, receivers and navigation processors have improved multi-path rejection (Strode and Groves, 2016). Commercial solutions are available that implement tight GNSS IMU fusion such as the APX-15 UAV by Applanix (Mian et al., 2015). While these systems are well integrated and would probably deliver comparable positioning quality to our proposed system, the proprietary structure does not offer the same flexibility and control as is available on our own hardware system. In our system we have full control over the spatial and temporal calibration, the raw measurements and estimation

algorithms. This leaves room for future integration of additional sensor modalities such as LiDAR or vision, and allows for direct extensions to our work from the wider robotics community.

Our previous estimation algorithm was based on a Kalman filter, followed by a full factor graph smoothing step (Schartel et al., 2020a). Now, we combine the state estimation into a single GTSAM framework, where an incremental smoothing algorithm runs online during the flight and full inference is computed after landing. It has been shown that incremental smoothing, in particular for systems with highly nonlinear dynamics, such as rotary wing vehicles, performs significantly better than filtering (Indelman et al., 2013). Furthermore, the factor graph formulation provides a favorable formalism to include additional localization sensors in the future. The motion estimator proposed in this paper is loosely coupled, i.e., the GNSS raw double difference measurements are prefiltered into a single position measurement using commercial software before being fused with IMU linear acceleration and angular velocity measurements. This design decision simplifies the development process significantly, because the software handles difficulties such as multi-path, shadowing, carrier phase ambiguity and switching between the RTK and satellite-based augmentation system (SBAS) solution. Tightly coupled systems on the other hand would deliver positional constraints even in underconstrained cases, for example when observing only two satellites (Schneider et al., 2016; Cao et al., 2021). Furthermore, tight integration would allow more accurate uncertainty propagation of the raw distance measurements and modeling of the frequency-dependent antenna phase centers. To partially account for this error, loosely coupled frameworks often model the offset between the IMU and GNSS antenna (Kaplan and Hegarty, 2005). We integrate this offset into the GTSAM optimization framework which then permits the system to spatially calibrate the GNSS antenna position relative to the IMU. Additionally, we extend GTSAM to include GNSS moving baseline measurements, i.e., the RTK baseline vector between two GNSS antennas mounted on the vehicle. Finally, we analyse the influence of the two measurement factors.

One remaining issue when fusing multiple sensor modalities is time synchronization, as drifting or noisy time stamps usually break the modeling assumptions. (Ding et al., 2008) perform an error analysis of time delay on GNSS-inertial navigation systems. They come to the conclusion that time synchronization is particularly important in dynamically fast, self-calibrating systems with high-precision sensors, i.e., exactly the robotic system described in this work. Timing errors may occur due to varying clock speeds, protocol transfer delays, sensor exposure or computing capacities. Robotic systems often time stamp sensors on different clocks and then translate the stamp from one clock to the other by estimating the clock offset and skew. The time offset estimation can either be based on comparing the same time stamps on two clocks (Eidson et al., 2002; Sommer et al., 2017; Osadcuks et al., 2020) or based on the correlation of sensor motions, such as a camera and IMU movement (Furgale et al., 2012; Li and Mourikis, 2014; Kelly and Sukhatme, 2014; Qin and Shen, 2018). More specifically, (Skog and Handel, 2011) propose a GNSS-inertial fusion algorithm that self-calibrates the timing offset. Time translation in software only, however, does not solve the whole sensor data acquisition problem. (Kelly et al., 2021) showed recently, that co-estimating time offset in causal filters introduces inconsistencies and potentially worsens the estimation result. Furthermore, even though the sensor data may be time stamped accurately, the sensor triggering could still happen out of sync and indeterminately. This issue then needs to be handled by interpolating sensor data as well as implementing stochastic cloning and complex buffering schemes to insert delayed sensor data into the optimization framework (Roumeliotis and Burdick, 2002; Lynen et al., 2013; Cioffi and Scaramuzza, 2020). Instead of handling sensor timing compensation after acquisition, a dedicated hardware solution can trigger and time stamp sensor data from a single clock. Different designs based on field-programmable gate arrays (FPGAs) or microcontroller units (MCUs) offer more or less freedom to be extended to a custom sensor setup and GNSS synchronization (Kais et al., 2006; Ding et al., 2008; Huck et al., 2011; Nikolic et al., 2014; Albrektsen and Johansen, 2018; Tschopp et al., 2020; Faizullin et al., 2021b; Faizullin et al., 2021a). We base our work on the open-source *VersaVIS* (Tschopp et al., 2020) sensor synchronization board. Here a SAMD21 MCU triggers an IMU-camera setup and stores synchronized time stamps. This hardware setup offers a wide variety of serial interfaces such as serial peripheral interface (SPI), universal asynchronous

receiver-transmitter (UART) and inter-integrated circuit (I2C). We extend the hardware by adding a high-resolution oscillator which we synchronize with respect to GNSS. Furthermore, our system implements hardware-generated pulse width modulations (PWMs) that allow triggering multiple sensor sources in parallel at desired GNSS time stamps. And we utilize the SAMD21's hardware capture capability to improve timing precision of sensor strobe signals.

Finally, GPSAR imaging is heavily dependent on the data collection process. While in principle GPSAR can integrate measurements from anywhere along the trajectory, in practice over- and undersampling, e.g., due to alternating velocities, causes artifacts in the radar image. This effect has been studied by (García-Fernández et al., 2020a) who propose a post processing step to subsample informative radar measurements and normalize the image. A complementary way to improve sampling is to enforce constant velocity along the GPSAR measurement trajectory. Available trajectory planners such as QGroundControl primarily focus on waypoint missions with a rudimentary option to constrain the flight speed (Dronecode, 2021). These planners do not consider platform acceleration limitations. This results in nonuniform sampling when changing directions because the position tracking controller does not generate a smooth transition. Also geometric primitives, such as circular segments, are not necessarily implemented. Our approach to mission planning is based on polynomial trajectory generation. Rotary wing MAVs with a perpendicular thrust vector are differentially flat. Their pose is fully defined by the position, heading and their derivatives. Consequently, polynomial trajectories, defined in position, yaw, and time, are a suitable representation to generate fully-defined, smooth trajectories (Richter et al., 2016). Because the MAV state is defined at every step in time, dynamic feasibility can be enforced (Mueller et al., 2015). Our approach uses these principles to generate uniform sampling measurement trajectories. The trajectories have a predefined start and end time to mask the radar measurements. They enforce constant velocity along circular or linear trajectories. And they are dynamically feasible with respect to maximum thrust and rotation rates to enable precise tracking. In combination with robust altitude tracking and waypoint control this leads to a navigation system that allows precise measurement collection, even beyond visual line of sight.

3. Conventions and Notations

In this paper, coordinate systems are denoted by calligraphic capital letters, including $\mathcal{I}$ to represent the world-fixed, inertial frame and $\mathcal{B}$ to represent the body-fixed frame rigidly attached to the vehicle. A 3D vector is represented by a lower case letter with a leading subscript describing the coordinate system the vector is represented in. Vectors have one or two right-hand subscripts for further description. For instance, $_{\mathcal{I}}r_{\mathcal{IB}} \in \mathbb{R}^3$ is the vector from the origin of coordinate system $\mathcal{I}$ to the origin of coordinate system $\mathcal{B}$, represented in the coordinate system $\mathcal{I}$. Similarly, $_{\mathcal{B}}\omega_{\mathcal{IB}} \in \mathbb{R}^3$ describes the angular velocity of $\mathcal{B}$ with respect to $\mathcal{I}$, represented in coordinate system $\mathcal{B}$. Single right-hand subscripts refer to vectors that are fully described by their direction and magnitude. $_{\mathcal{I}}v_{\mathcal{B}} \in \mathbb{R}^3$ describes the translational velocity of the origin of coordinate system B, represented in coordinate system $\mathcal{I}$. $_{\mathcal{I}}r_g \in \mathbb{R}^3$ describes the direction of the gravitational field, represented in coordinate system $\mathcal{I}$. 3D rotations and rigid transformations are represented by the capital letters R and T respectively. $R_{\mathcal{IB}} \in SO(3)$ indicates the orientation, $T_{\mathcal{IB}} \in SE(3)$ the orientation and translation of frame $\mathcal{B}$ with respect to reference frame $\mathcal{I}$. We use passive rotations and transformations to map 3D vectors from one frame to another. For example, $_{\mathcal{I}}v_{\mathcal{B}} = R_{\mathcal{IB}}\ _{\mathcal{B}}v_{\mathcal{B}}$ maps the velocity represented in frame $\mathcal{B}$ to frame $\mathcal{I}$.

4. Synthetic Aperture Radar Imaging

Synthetic aperture radar imaging is the process of forming a high-resolution 3D image from a collection of radar range measurements. Figure 2a exemplifies the idealized range response of a single target. Typically, a single radar chirp contains not only one target but the sum of noisy responses of multiple targets with different scattering characteristics. In order to filter out the 3D location of

(a) Radar response of a single point target at 2.67 m.

(b) Back projection of radar response into grid cell.
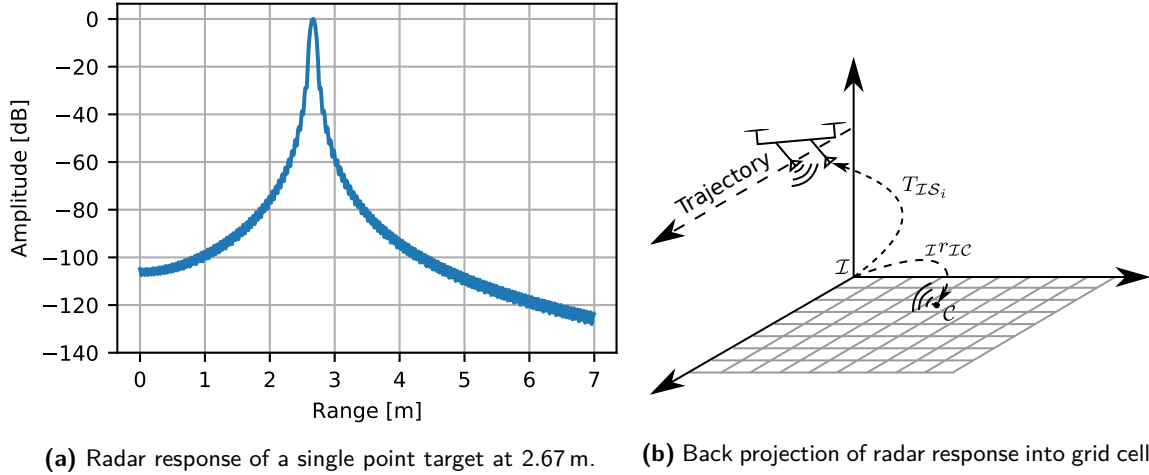
Figure 2. A back projection algorithm forms the radar image. Multiple radar range measurements are accumulated into a 3D voxel grid to resolve the bearing ambiguity and mitigate noise. This process requires a well defined measurement trajectory, accurate radar antenna localization and an accurate DSM.

prominent targets, the platform averages multiple measurements from different viewpoints gathered along a measurement trajectory (see Figure 2b). Observing a radar target from different view points resolves the bearing ambiguity and reveals the location of strong scatterers, which can include both plastic and metal landmines. The particular back projection algorithm used in this work enforces specific design decisions on the motion of the platform, the radar antenna localization, and the generation of a DSM (Zaugg and Long, 2015; Schartel, 2021). In this section, we elaborate the radar imaging process and identify the implications on the system design.

The system in this work has a side-looking single-input single-output radar. With each measurement, the radar emits a frequency-modulated continuous wave (FMCW) that reflects at the interface between materials with different permittivity. A frequency mixer creates the difference signal between the still transmitting FMCW and its returning time-delayed copies. For a single scatterer and assuming no antenna movement for the duration of the chirp, the difference signal has a constant frequency depending on the distance to the object and the propagation speed of the electromagnetic wave. An analog-to-digital converter discretizes the difference signal and a discrete Fourier transform isolates the frequency components. The resulting frequency spectrum correlates with the distance to all scatterers in the field of view.

In the next step, every single measurement is back projected into every grid cell of a 3D voxel map to evaluate the complex pixel value $A(_{\mathcal{I}}r_{\mathcal{IC}}) \in \mathbb{C}$ of each cell located at $_{\mathcal{I}}r_{\mathcal{IC}} \in \mathbb{R}^3$, where $\mathcal{I}$ denotes the inertial coordinate frame and $\mathcal{C}$ is the image cell center. Given the position of the radar antennas with respect to the cell location and the electromagnetic wave propagation speed in air and soil as well as possible refraction at the ground surface, the amplitude and phase of the measured, range-compressed radar signal S_η is determined, see Figure 2a. The resulting signal at the image cell location is multiplied with the negative expected response if a perfect point target was located inside the cell and summed up over all measurements η.

$$A(_{\mathcal{I}}r_{\mathcal{IC}}) = \sum_{\eta=1}^{N} w_\eta \; S_\eta \; e^{-j\phi_\eta}, \tag{1}$$

where N is the total number of measurements in one trajectory, w_η is a normalization weight to compensate for nonuniform sampling, and ϕ_η is the expected phase of the point target. This phase depends on the radar antenna position, the location of the imaging cell, and the DSM defining the interface between air and soil and thus the expected wave propagation time.

The algorithm has the following assumptions to perform range estimation and obtain focused images:

- an exact pose estimation $T_{\mathcal{IS}_i}$ of the transmitting and receiving radar antennas,
- an exact DSM,
- an exact model of soil permittivity and ground refraction.

These assumptions often lead to the conclusion that the range of the antenna to the target needs to be known within $\frac{1}{16}$ or $\frac{1}{8}$ of the wavelength, which in our case would be less than 10 mm (Schartel, 2021; García-Fernández et al., 2018). Otherwise, the coherent addition in Eq. (1) will not correlate due to the radar signals being out of phase for the same target observed from different viewpoints. In practice, however, the algorithm can image targets with less accurate range estimation. In the case of less accurate antenna position estimates, biases in the range estimate will split the energy of a single target across multiple image cells and cause defocus or *smearing*. The goal of this work is to minimize the errors that occur due to radar antenna positioning. In the next section, we provide an overview of our complete airborne GPSAR system. In Section 6, we introduce vehicle navigation principles that enforce uniform sampling of target cells from different view points as well as accurate trajectory tracking control. In Sections 7 and 8, we address time synchronization and vehicle localization to improve the position estimation of the radar antennas.

5. System Overview

Our system can generate georeferenced aerial imaging products as well as execute low-altitude GPSAR missions. This requires a specific hardware setup as well as mission workflow to efficiently combine both functionalities into a single system.

5.1. MAV Platform Setup

Our MAV setup consists of a commercially available DJI M600 Pro platform equipped with a custom sensor pod payload. The sensor pod holds all components that are necessary to implement the airborne GPSAR. We divide the setup into an upper compartment, that includes all processing, communication and interoceptive sensing (Figure 3a) and a lower compartment holding all exteroceptive sensors (Figure 3b). At the center of the upper compartment is an UP Squared single board

(a) Top view of sensor payload.　　　　**(b)** Bottom view of sensor payload.
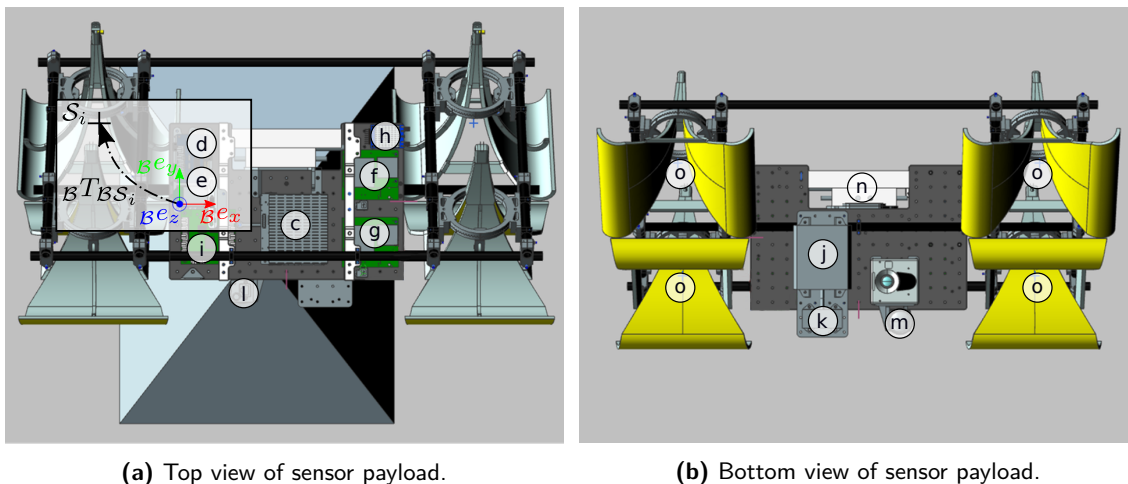
Figure 3. The sensor pod holds all sensors necessary for optical and radar imaging as well as platform localization and control. The carbon tube construction fixes the radar antennas in place such that the computer-aided design (CAD) model fully defines the position and orientation of the antenna phase centers with respect to the IMU.

Table 1. Component list of the landmine detecting MAV. The ID refers to the labels in Figures 1 and 3.
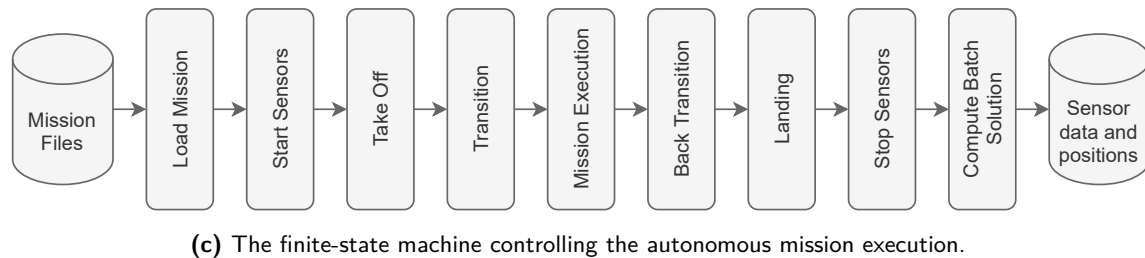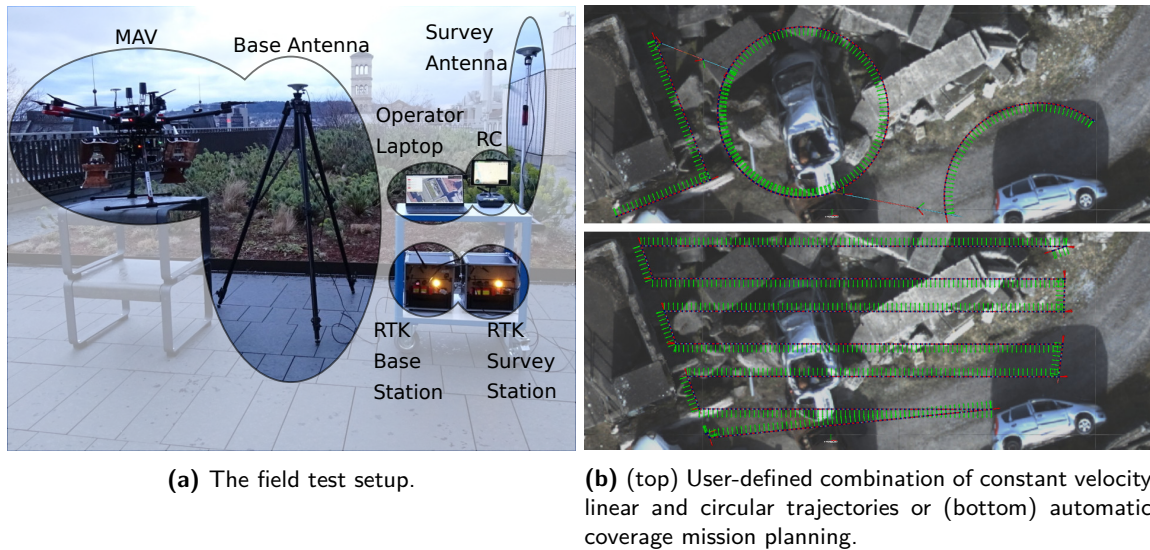
ID	Item	Description	Rate
a	DJI M600 Pro	Hexacopter with ROS onboad SDK	-
b	Sensor payload	Sensor pod (prepreg carbon, milled alloy, nylon 3D print)	-
c	UPS-APLP4-A20-0864	Onboard processing unit	-
d	VersaVIS	Sensor interface based on SAMD21 MCU	-
e	ADIS16448BMLZ	Industrial grade MEMS IMU	1 kHz
f	Piksi Multi reference	RTK GNSS position receiver	10 Hz
g	Piksi Multi attitude	RTK GNSS moving baseline receiver	5 Hz
h	RFD868x	Long range RTK GNSS correction modem	-
i	Power distribution board	16 V to 27 V DC input, 5 V, 12 V, and 24 V DC output	-
j	US-D1	24 GHz radar altimeter	100 Hz
k	LIDAR-Lite v3HP	LiDAR altimeter	10 Hz
l	BFLY-PGE-31S4C-C	Global shutter 3.2 Mpx color camera	2 Hz
m	Edmund Optics #35-139	6 mm fixed focal length lens	-
n	Custom radar	1 GHz to 4 GHz FMCW ground-penetrating radar	200 Hz
o	Horn antennas	Transversal electromagnetic horn antenna for GPSAR	-
-	33-HC882-28	Helical triple band GNSS antennas	-

computer (c), sufficiently powerful to log the sensor data, compute the online navigation solution, and control the MAV through its autopilot and robot operating system (ROS). Next to it is the *VersaVIS* (d), an MCU with peripherials to interface the sensors. The IMU (e) and two GNSS receivers (f) and (g), together with two helical triple band antennas, and the corrections modem (h) form the basis of the high-precision navigation solution necessary for exact platform localization. All components are powered by a single power board (i) capable of converting the MAV's 18 V power supply from the onboard batteries to 5, 12, and 24 V. On the bottom of the sensor pod the radar altimeter (j) and LiDAR altimeter (k) measure the platform height above ground level (AGL). The camera (l) and lens (m) create aerial imagery to build a photogrammetric map for path planning and radar imaging. Next to them, the custom 1 GHz to 4 GHz FMCW radar (n), and its four horn antennas (o) with different polarization direction are used to create GPSAR images (Burr et al., 2018). The four antenna phase center locations with respect to the IMU $_{\mathcal{B}}T_{\mathcal{BS}_i}$ for $i \in \{1, \ldots, 4\}$ are well defined through the CAD. The camera pose with respect to the IMU $_{\mathcal{B}}T_{\mathcal{BS}_5}$ is determined with a calibration software (Furgale et al., 2013).

5.2. Mission Overview

The complete system setup, shown in Figure 4a, includes the copter with payload, an RTK base station that supplies WiFi, 4G communication and RTK corrections, a survey station to measure geodetic ground control points (GCPs) and an operator laptop running ROS to monitor the platform status and select, configure and start the autonomous mission. For safety, the operator can always interrupt a mission with a remote control (RC). The mission itself can be planned in the field or in the office on the operator laptop with respect to a georeferenced map. Figure 4b shows two example missions consisting of concatenated motion primitives, e.g., circles or lines or an automatically generated boustrophedon coverage pattern (Bähnemann et al., 2021). After surveying the RTK GNSS base station position, the mission is executed autonomously with all processing and data logging happening on board the platform controlled by the finite-state machine depicted in Figure 4c (Pradalier, 2017). The finite-state machine starts sensor recording and takes off the MAV automatically. The platform ascends to a collision-free altitude, e.g., 30 m, and transitions to a location above the start of the mission. It then descends to the mission altitude and executes the preplanned trajectory. After finishing, the platform transitions back to high altitude, returns to the take off position and automatically lands. After landing the system post processes the sensor position batch solution. The sensor and positioning data are downloaded from the onboard computer and passed to the optical or radar imaging processor.

(a) The field test setup.

(b) (top) User-defined combination of constant velocity linear and circular trajectories or (bottom) automatic coverage mission planning.

**(c)** The finite-state machine controlling the autonomous mission execution.

Figure 4. An overview of the system setup, mission planning and execution. The operator plans the missions offline and the MAV executes the flight and sensor logging autonomously.

6. Navigation

According to the back projection Eq. (1), the radar response of a single radar image cell is the weighted sum of all radar measurements illuminating that cell. In order to observe all targets in a scene and simplify the weighting, it is important that the MAV illuminates all cells uniformly and from different view points. For this purpose we generate smooth, constant velocity radar measurement trajectories as well as a trajectory tracking controller that allows terrain following control.

6.1. Polynomial Trajectory Generation

Typical radar imaging modes are *stripmap* and *circular* GPSAR. *Stripmap* refers to flying over the target area along a straight line, while in *circular* GPSAR the platform orbits the target area with the radar pointing towards the center. Our trajectory generation framework allows combining these two imaging modes in an arbitrary fashion as shown in Figure 4b. The trajectories are formed based on high-level user input and are automatically transformed into continuous, smooth, and feasible polynomial trajectories based on a differentially flat multirotor model (Richter et al., 2016; Burri et al., 2015). For example, in Figure 5, a user defines a circular arc by its center C, radius r, start angle ψ_s and subtended angle θ, as well as altitude, platform velocity and heading while flying along the circle. In an intermediate step, this user input is converted into waypoint constraints that serve as polynomial support vertices. In the case of the circle, these are $M + 1$ vertices equally spaced

(a) User interface.

(b) Dynamically feasible circular trajectory.
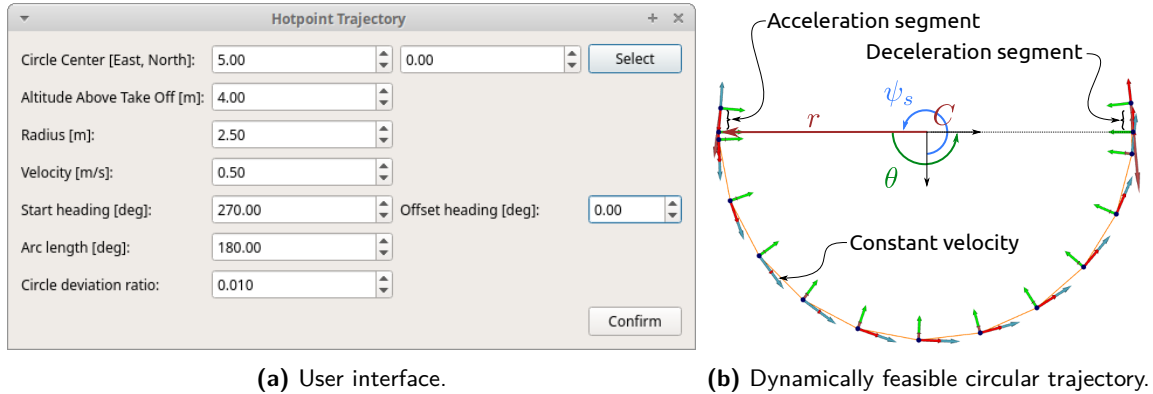
Figure 5. The user plans continuous, smooth and feasible polynomial measurement trajectories (orange) via high-level inputs. A trajectory fully defines the MAV position and orientation (coordinate frames), velocity (blue arrows) and acceleration (red arrows) at every time step during the mission.

along the perimeter. M is computed automatically as

$$M = \left\lceil \frac{\theta}{\arccos\left(2\left(1-d\right)^2 - 1\right)} \right\rceil, \tag{2}$$

where $d \in (0,1)$ is a parameter that describes the relative deviation of the vertices approximating the circle from an actual circle. In the case of a straight line the support vertices are simply the start and goal point of the line. Once the support vertices' positions have been defined, their velocity, i.e., the first derivative of the polynomial, is set to the user input velocity. In the *circular* case, this is angled tangentially to the trajectory, while in the *stripmap* the velocity vector points towards the next waypoint. All other derivatives, depending on the maximum polynomial degree, are set to zero, as the platform should move with constant velocity along the measurement trajectory. Similarly, the heading polynomial is fully constrained by pointing into the direction of the velocity vector. Furthermore, the segment transition times between two vertices are fully defined by the path length and constant velocity. The constrained vertices and segment times completely describe the polynomial coefficients and thus the full desired state of the MAV at every time.

Every measurement trajectory is started from rest. In order to bring the platform up to speed at the first vertex and slow it down after the final vertex we introduce acceleration and deceleration segments. At both the start and stop positions, an additional vertex is connected to the measurement trajectory. The time allocation of this segment is unknown and subject to physical feasibility constraints of the platform. A binary search minimizes the segment time such that the acceleration and deceleration motion is as short as possible but still feasible with respect to minimum and maximum thrust, velocity, and rotation rates (Mueller et al., 2015). Similarly, consecutive measurement trajectories are connected by rest-to-rest trajectories using the minimum segment time search.

6.2. Tracking Controller

Figure 6 displays the control loop that runs on board the MAV. The finite-state machine has two options to control the flight of the MAV. If the goal is to go as fast as possible from one waypoint to another, e.g., in the transition maneuver, it can directly send the desired position p_{ref} and heading ψ_{ref} to the tracking controller. If smooth, well-defined trajectories are desired, it can send polynomial missions as described above to the controller. The controller continuously evaluates the desired position, velocity v_{ref}, acceleration a_{ref}, heading, and yaw rate $\dot{\psi}_{\text{ref}}$, and compares it to the current position, velocity and acceleration estimated by the DJI autopilot (p, v, a) to compute the
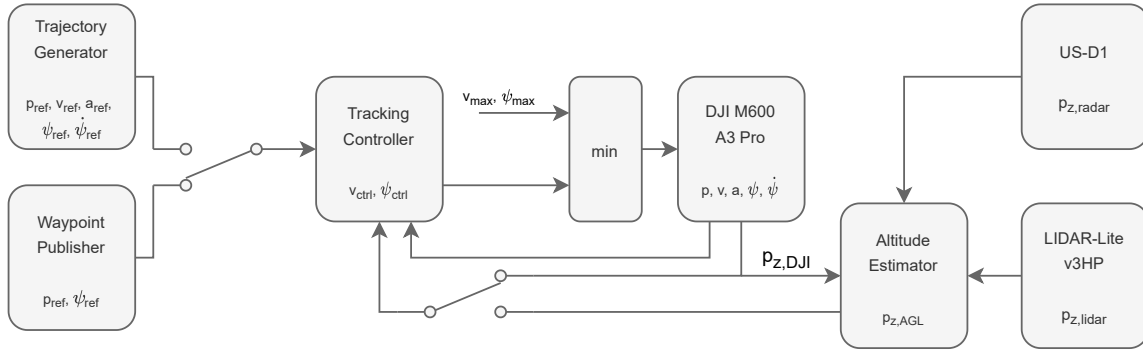
Figure 6. The platform control loop. The finite-state machine switches between trajectory tracking or waypoint tracking and between control at GNSS altitude or AGL.

controlled translational velocity v_{ctrl} and yaw rate $\dot{\psi}_{\text{ctrl}}$.

$$v_{\text{ctrl}} = K_p \left(p_{\text{ref}} - p\right) + K_v \left(v_{\text{ref}} - v\right) + K_a \left(a_{\text{ref}} - a\right), \tag{3}$$

$$\dot{\psi}_{\text{ctrl}} = K_\psi \left(\psi_{\text{ref}} - \psi\right) + K_{\dot{\psi}} \left(\dot{\psi}_{\text{ref}} - \dot{\psi}\right), \tag{4}$$

where K_i are user-defined gains. The reference trajectory and DJI state are represented in a local Cartesian east-north-up (ENU) coordinate system that is set by the DJI autopilot at the start of the mission. A formal introduction of the DJI ENU frame and DJI body frame with calligraphic letters is omitted for brevity. Note that since we are sending velocity references to the autopilot, the position error acts like an integration term and the acceleration error acts like a damping term on the control loop. The control signal is passed through a limiter before being sent to the autopilot to avoid infeasible input commands.

6.3. Altitude Estimation

Besides switching between trajectory tracking and waypoint tracking, the finite-state machine can also switch between flying with DJI's proprietary altitude estimate and flying relative to the ground surface. The latter mode is particularly useful when flying radar missions at 2–8 m because the absolute surface height is not necessarily known and we wish to take measurements from well-defined altitudes. Furthermore, the regular DJI altitude is subject to large fluctuations which makes terrain tracking control the safer option when flying below 8 m. As Figure 6 shows, the altitude estimator fuses the DJI altitude with two different altimeter measurements to compute the height above ground level $p_{z,\text{AGL}}$. Multiple range sensors ensure robustness against outliers and sensor outages.

The implementation of the altitude estimator resembles a Kalman filter, with the change in DJI altitude governing the process model. Both downward facing radar altimeter range $p_{z,\text{radar}}$ and LiDAR altimeter range $p_{z,\text{lidar}}$ serve as independent measurement updates. Since large roll and pitch angles invalidate the AGL measurements, different measures are taken to correct those. First, the range measurements are corrected by the current roll and pitch angle to represent the vertical distance to the ground assuming a plane environment. Second, their standard deviation is scaled based on the attitude. Third, a cutoff attitude rejects measurements when either of the platform's roll or pitch angles exceed a threshold. Furthermore, the filter implements a Mahalanobis threshold to reject outlier measurements. And the filter scales the sensor uncertainty based on the magnitude of the measured range, because with increasing height objects in the field of view may invalidate the measurement.

Figure 7 shows two example segments of the resulting altitude estimate. In the high-altitude segment in Figure 7a, the filter shows robustness to outlier range measurements. The LiDAR does not have the necessary maximum range and is prone to large roll and pitch angles during acceleration

**(a)** High-altitude estimation.

(b) Low-altitude estimation.
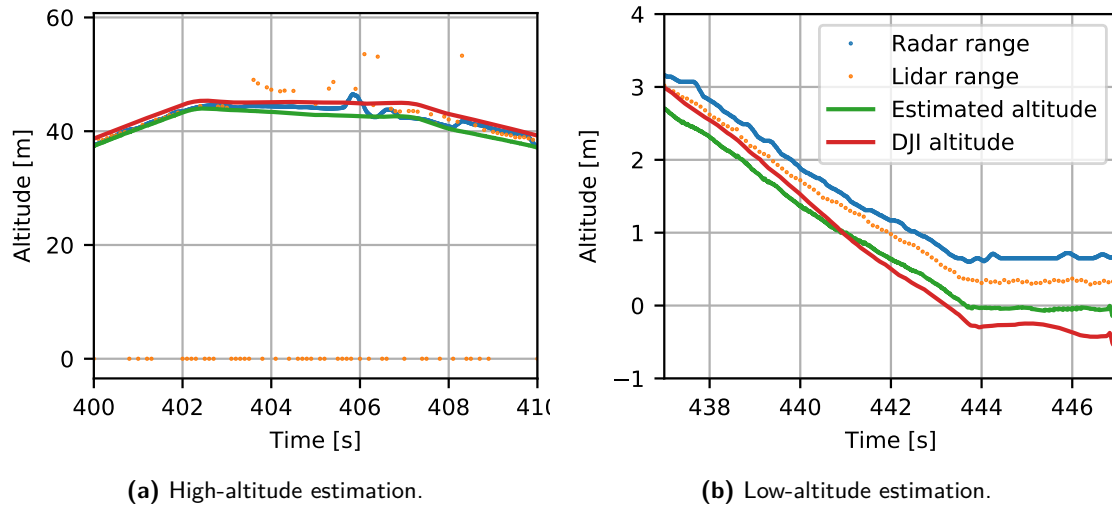
Figure 7. The altitude estimator during a typical mission over flat terrain. At high altitude, it rejects outlier range measurements. At low altitudes, it compensates for DJI altitude drift, here allowing safe landing at 0 m.

movements. The radar altimeter has a relatively large measurement cone and at $t = 406$ s measures the wrong AGL. The estimator completely ignores the invalid LiDAR measurments and only slowly adapts to the radar altimeter measurements while trusting the DJI altitude most. In the low-altitude segment in Figure 7b, when the distance to the surface decreases, the quality of the sensor measurements improves. The filter trusts the range measurements more than the DJI altitude which in this case allows safe landing at 0 m altitude. Note, that during this flight the DJI altitude drifted about 50 cm from take off to landing. Relying only on the DJI altitude would thus potentially lead to crashes at low altitude.
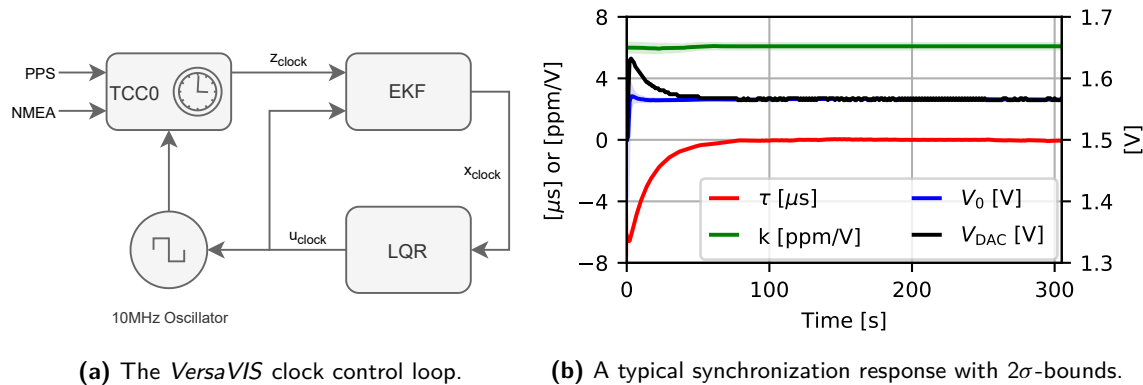
7. Time Synchronization

An important aspect in sensor fusion is time synchronization, that is, triggering and time stamping sensors accurately through the same clock source. For high-quality GPSAR reconstruction, it is particularly important to accurately time stamp the radar measurement with respect to the GNSS and the IMU to determine the exact radar antenna poses with every chirp. In our system, all navigation sensors and radar messages are time-stamped with a globally consistent time (GNSS time) to allow accurate motion estimation. Figure 8 shows an overview of the timing modalities.

The GNSS provides the time reference of the system and has a typical precision of 50 ns (Kaplan and Hegarty, 2005). Naturally, all GNSS measurements have a GNSS time stamp. All other sensors are time stamped with respect to the onboard processing unit or *VersaVIS* clock. To synchronize these two systems, one of the GNSS receivers emits a PPS signal. On the onboard processing unit a Linux kernel interrupt captures the PPS signal and the *chrony* protocol synchronizes its clock (Curnow and Lichvar, 1997). On the *VersaVIS* MCU the PPS continuously triggers a control loop that synchronizes a clock derived from a 10 MHz external oscillator. All navigation sensors and the RGB camera are triggered and time stamped through counters derived from the same oscillator. This allows sensor timing precisions as fine as 0.1 μs with respect to GNSS as shown in our evaluations below. The ground-penetrating radar system is not interfaced by the *VersaVis* board, as the radar driver board has its own internal clock that is synchronized to the primary onboard clock. It receives the onboard processing unit's time through USB at the beginning of a measurement stream and then open-loop propagates its internal clock. Due to the clock drift, this accumulates a clock offset of approximately 10 ms during a 15 min mission which corresponds to 1 ppm.
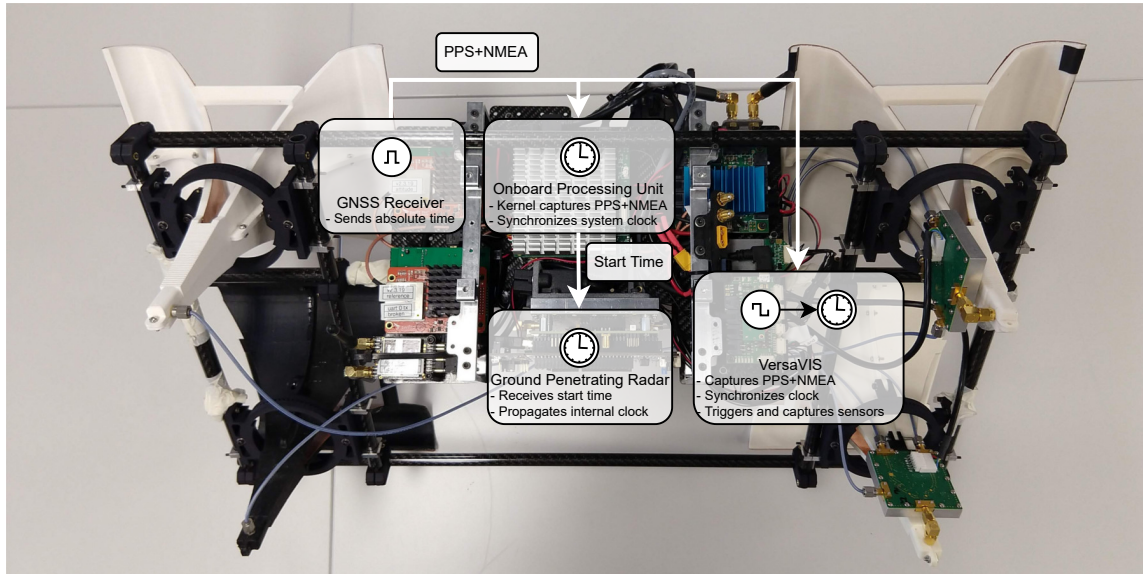
Figure 8. Top view of the sensor pod overlayed by the sensor timing overview. All sensors are time stamped with respect to GNSS time. The radar receives the current GNSS time from the onboard processing unit. The *VersaVIS* directly synchronizes with respect to the pulse per second (PPS) and time stamps and triggers all other sensors.

(a) The *VersaVIS* clock control loop.

(b) A typical synchronization response with 2σ-bounds.

Figure 9. A PPS driven control loop continuously steers the *VersaVIS* clock offset from GNSS time to zero. The controller modifies the frequency of an external oscillator that drives the clock.

7.1. VersaVIS to GNSS Synchronization

The *VersaVIS* has an internal record of time by counting the pulses coming from the 10 MHz external oscillator on a 24 bit timer/counter for control (TCC). The first PPS pulse in conjunction with its National Marine Electronics Association (NMEA) time and date sentence initializes this process. The TCC then wraps around every 10^7 ticks to increment one second. Synchronization delays on the MCU introduce an initial offset between PPS and internally propagated time. Furthermore, temperature changes and oscillator resolution cause a time varying clock drift. Hence, a 1 Hz control loop steers the external oscillator frequency to drive the time offset to zero. Figure 9 shows the closed-loop system that synchronizes the *VersaVIS* to GNSS time and a typical control response.

The key idea behind the synchronization mechanism shown in Figure 9a is that applying an input voltage to the external oscillator changes its frequency and allows driving the difference between internally tracked time and GNSS time to zero. With every captured PPS pulse an extended Kalman

filter (EKF) estimates the current time offset τ, the constant nominal control voltage V_0 to set the oscillator frequency to exactly $10\,\mathrm{MHz}$ and the constant factor k that converts oscillator input voltages to expected clock drift. The measurement input is the time difference between when a PPS pulse was captured by the TCC and its actual time according to the NMEA signal. The following nonlinear continuous-time state-space equations with state variable x_{clock}, control input u_{clock} and measurement input z_{clock} describe the system:

$$x_{\mathrm{clock}} = \begin{bmatrix} \tau & V_0 & k \end{bmatrix}^T, \qquad\qquad\qquad u_{\mathrm{clock}} = V_{\mathrm{DAC}}, \qquad\qquad (5)$$

$$\dot{x}_{\mathrm{clock}} = \begin{bmatrix} (V_{\mathrm{DAC}} - V_0)\,k \\ 0 \\ 0 \end{bmatrix} + w, \qquad\qquad w \sim \mathcal{N}(0, Q), \qquad\qquad (6)$$

$$z_{\mathrm{clock}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_{\mathrm{clock}} + v, \qquad\qquad v \sim \mathcal{N}(0, R), \qquad\qquad (7)$$

where the digital-to-analog converter (DAC) voltage V_{DAC} is the control input and w and v are normally distributed, additive white noise with variances Q and R, respectively.

Given the full state estimate x_{clock}, a linear-quadratic regulator (LQR) computes and sets the control voltage V_{DAC} to accelerate or decelerate the external oscillator to drive the estimated time offset τ to zero. The optimal gain K_c is computed offline based on clock dynamics in Eq. (6) discretized and linearized about the nominal state. The output voltage is clamped to remain within oscillator control voltage limits.

$$V_{\mathrm{DAC}} = -K_c\,\tau + V_0. \qquad\qquad (8)$$

Typically, the control loop converges within $100\,\mathrm{s}$ seconds as Figure 9b shows. Here, the tracked time on the *VersaVIS* has an initial delay of $6\,\mathrm{\mu s}$. Thus the controller increases the input voltage to accelerate the external oscillator. While the offset τ is settling, the control input V_{DAC} settles as well towards the nominal control voltage V_0. During the course of the run, the controller keeps alternating the control voltage stepwise to remain within zero time offset given the resolution of the DAC.

7.2. Sensor Measurement Time Stamps

The synchronized TCC0 is the basis for the synchronization of all sensors that connect to the *VersaVIS* peripherals. All TCCs, timer/counters (TCs) and the real time counter (RTC) count the $10\,\mathrm{MHz}$ external clock ticks synchronously with the TCC0 to generate PWMs to trigger and to provide capture channels to time stamp sensor measurements. Typically, the sensors supply general-purpose input/outputs to start a measurement or indicate its exposure. Figure 10 provides the timing schematics for all sensors that rely on the *VersaVIS* clock.

Figure 10a shows the image capture mechanism of the color camera. A PWM triggers the camera exposure. The camera returns a strobe and the TCC captures start and stop to time stamp the mid-exposure time. The time stamp and the incremented image counter are stored in a circular buffer and sent to the onboard computer whenever possible. Based on the image number, the time stamp is associated to an image that is sent to the computer via Ethernet (Tschopp et al., 2020).

Figure 10b shows the IMU synchronization. The TC generates a PWM that serves as an external sampling clock to the IMU. Once the IMU has sampled its micro-electro-mechanical systems (MEMS), it sends out a data ready signal that is captured on the *VersaVIS* external interrupt controller (EIC). The data ready flag indicates that the measurement can safely be read via SPI. The *VersaVIS* stores the measurement together with the trigger time in a circular buffer and sends it out to the onboard computer.

The LiDAR synchronization shown in Figure 10c works similarly. The TC generates a PWM to engage a measurement with known time stamp. After triggering, the MCU polls the I2C to receive the associated range measurement. Both time stamp and measurement are stored in a circular buffer and sent to the onboard computer.
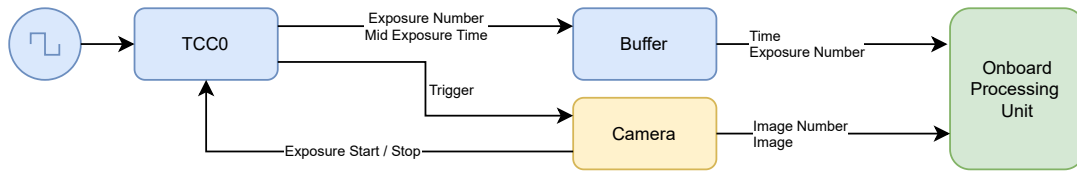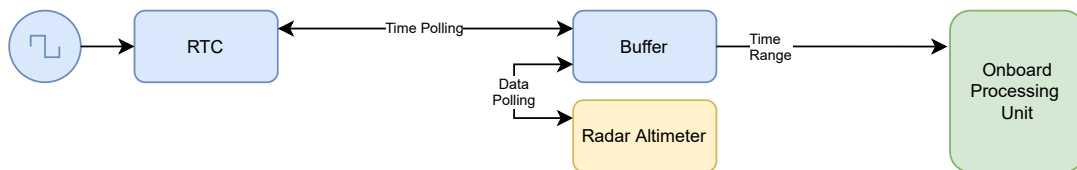
(a) The TCC0 triggers the color camera and captures its mid-exposure time. The onboard processing unit matches exposure time stamps with images.
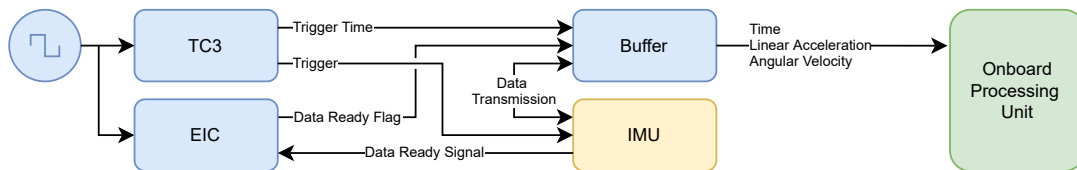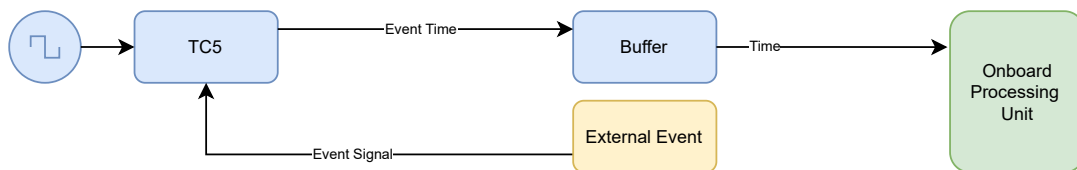

(b) The TC3 provides an external clock input to the IMU. The data ready signal allows safe data acquisition via SPI.
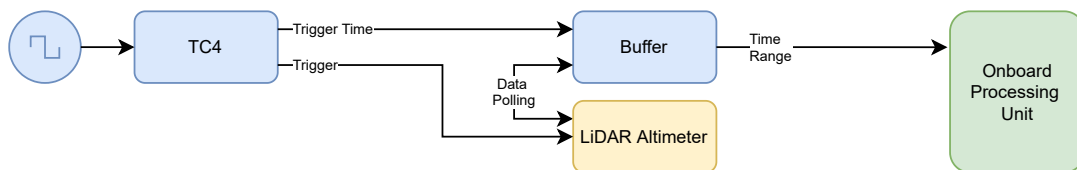
(c) The TC4 triggers and stamps LiDAR measurements. The measurements are polled via I2C.


(d) The radar altimeter does not have an external trigger input. It is continuously polled via UART. The measurement time is retrieved from a RTC with ms precision.

(e) The TC5 captures miscellaneous external events on a full resolution counter.

Figure 10. The time stamping and data retrieval mechanisms for the different sensor interfaces. Every sensor has a dedicated counter that time stamps the data synchronously to the known external oscillator ticks.

The radar altimeter does not have a particular synchronization interface. Thus we poll the UART when resources are available and time stamp the measurement via RTC as shown in Figure 10d. Furthermore, our firmware provides an external event channel. Figure 10e shows how a TC captures the external input and forwards the event time to the system. We use this mechanism to evaluate the absolute timing accuracy of the proposed sensor synchronization scheme.

(a) External signal and ground truth comparison.

(b) Time stamping error over time.
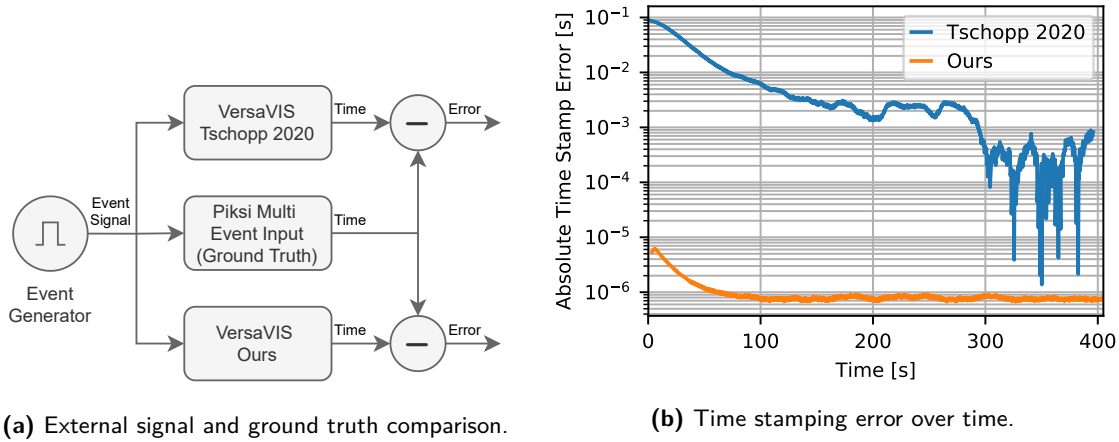
Figure 11. Time stamp accuracy evaluation. Our hardware-capture-based synchronization method reaches sub µs accuracy. In comparison to the previous software-based method it shows better accuracy, precision, and convergence time.

7.3. Timing Evaluation

We compared our MCU timing architecture to the original software trigger presented by (Tschopp et al., 2020) both in terms of accuracy and precision. A 5 Hz external signal generator was used to simulate a generic sensor strobe signal as depicted in Figure 11a. The rising edge of the signal was time stamped on both firmware versions and on a ground truth Piksi Multi receiver which has a timing accuracy of ±60 ns (Swift Navigation, 2019). The resulting time stamps were compared to the ground truth time stamps. The error over time in Figure 11b shows that our system is capable of time stamping sensor data with a mean accuracy of 0.8 µs and a standard deviation of 0.05 µs. The original firmware, which synchronizes only with respect to the onboard computer via universal serial bus (USB) and not directly to PPS, reached a mean accuracy of 180 µs and standard deviation of 312 µs after 300 s convergence time as opposed to 100 s. Our 0.05 µs precision shows that we can consistently time stamp sensor data with a full clock resolution of 0.1 µs provided by the 10 MHz oscillator. The remaining 0.8 µs time stamp offset probably results from a combination of signal edge rise times, GNSS PPS signal accuracy and steady state offset of our synchronization controller.

Note that, in practice every sensor will introduce some small uncompensated delay in addition to the 0.8 µs system accuracy. For example, the IMU requires 100 µs to sample and average its MEMS (Looney, 2018). In the case of GNSS-inertial navigation this additional delay is negligible. First, because other error sources, e.g., the gravity model and frame vibrations induced by rotor rotations, outweigh such small timing inaccuracies (see Section 9.2). Second, because the timing accuracy is still sufficient given the vehicle dynamics. According to (Ding et al., 2008), IMU time delay Δt does not significantly alter the localization result if it fulfills the following inequality:

$$|\Delta t| \ll \frac{|h_{\mathrm{pos}}(x, c) - z_{\mathrm{pos}}|}{|\iint j \ dt^2|}, \tag{9}$$

where the numerator represents the measurement innovation during a GNSS update and the denominator represents the double integrated change in acceleration (jerk) in between two GNSS measurements. Small timing errors are required either by high-accuracy sensors which lead to a small numerator or highly dynamic systems which lead to a large denominator. If we assume perfect IMU integration and GNSS lever arm calibration, our system's lower bound on the innovation is determined by the RTK GNSS measurement error which is expected to be about 5 mm. The maximum jerk we determined in regular operation was about $100 \ \mathrm{ms}^{-3}$. Double integration of the jerk over a GNSS measurement period of 0.1 s generates a velocity of $0.5 \ \mathrm{m \, s}^{-1}$. Thus the time delay has to be significantly smaller than 10 ms, i.e., at most 1 ms. When considering half of the
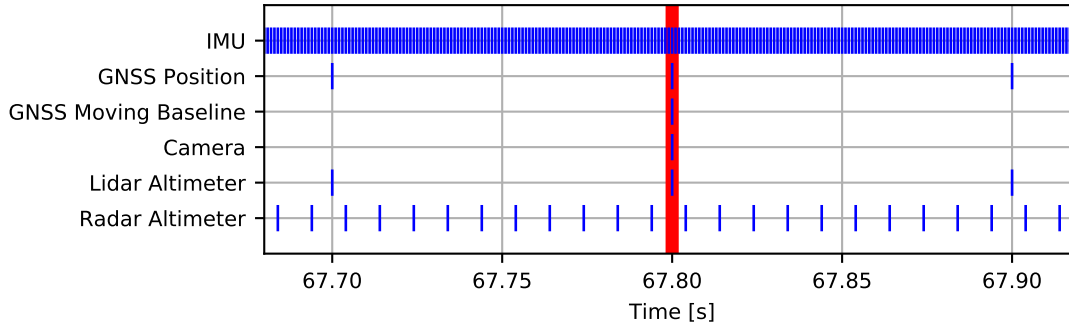
Figure 12. Timeline of 200 ms synchronized sensor data time stamps. The vertical red line indicates simultaneous capture of all sensors but the polled radar altimeter at GNSS time stamps. The IMU is sampled uniformly every 1 ms.

100 µs IMU sampling period as uncompensated delay, 0.8 µs synchronization offset and 2σ-jitter of 0.1 µs, our system has an uncompensated delay of 50.9 µs between IMU and GNSS measurements, approximately 1/20th of the required 1 ms. This margin prepares it for future use cases when motion sensing will become more precise (smaller numerator) and greater flight dynamics will be reached (greater denominator). Following the same calculation, the original firmware from (Tschopp et al., 2020) with 180 µs synchronization offset and 624 µs 2σ-jitter has an expected maximum uncompensated delay of 854 µs which would still be borderline sufficient.
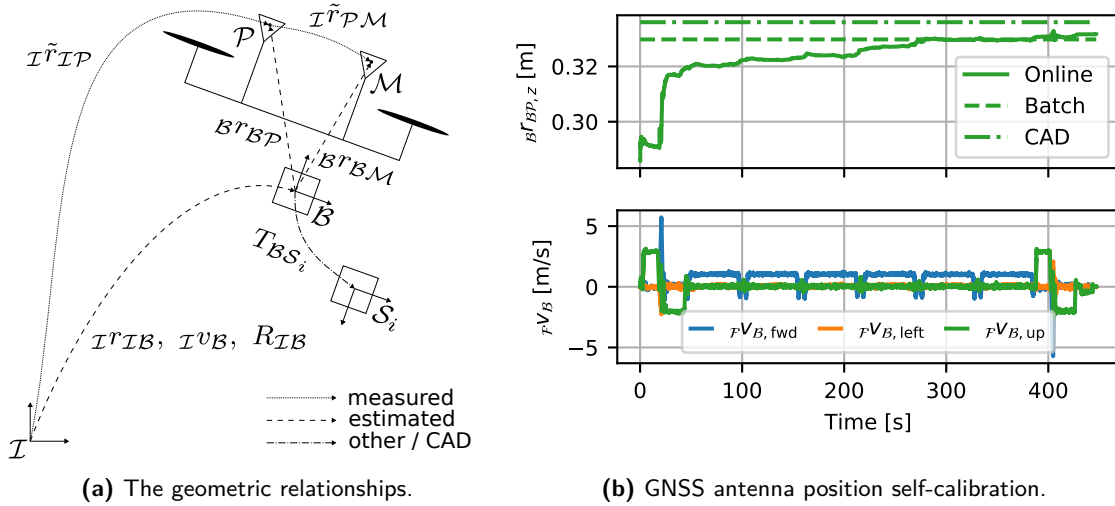
Finally, Figure 12 shows that all sensors except the polled US-D1 radar altimeter can be simultaneously evaluated at GNSS times due to the parallel running hardware triggering and capture mechanism. This also includes an image exposure compensation scheme as presented in (Nikolic et al., 2014) that ensures mid-exposure stamps of the camera at desired GNSS times. The parallel processing leads to jitter-free sensor sampling with well-defined time stamps. For example, the IMU is evaluated uniformly every 1 ms. As we will see in Section 8.1, having deterministic sensor time stamps simplifies the fusion algorithm design.

8. Localization

For localization our robot uses a 1 kHz IMU, a 10 Hz RTK GNSS position receiver, and a 5 Hz RTK GNSS moving baseline receiver. The IMU measures the linear acceleration $_{\mathcal{B}}\tilde{a}_{\mathcal{B}} \in \mathbb{R}^3$ and angular velocity $_{\mathcal{B}}\tilde{\omega}_{\mathcal{IB}} \in \mathbb{R}^3$ of the platform in body coordinates $\mathcal{B}$, where $\mathcal{B}$ aligns with the measurement axes of the IMU. The position receiver measures the position $_{\mathcal{I}}\tilde{r}_{\mathcal{IP}} \in \mathbb{R}^3$ of GNSS antenna $\mathcal{P}$ with respect to inertial frame $\mathcal{I}$. Without loss of generality we define frame $\mathcal{I}$ to correspond to a cartesian frame with orientation aligned with the local east-north-up frame and origin at the RTK GNSS base station position. The moving baseline receiver measures the baseline vector $_{\mathcal{I}}\tilde{r}_{\mathcal{PM}} \in \mathbb{R}^3$ between GNSS antennas $\mathcal{P}$ and $\mathcal{M}$ rigidly attached to the platform, see Figure 13a.

These four measurements let us estimate the sensor pod's position $_{\mathcal{I}}r_{\mathcal{IB}} \in \mathbb{R}^3$, velocity $_{\mathcal{I}}v_{\mathcal{B}} \in \mathbb{R}^3$, and orientation $R_{\mathcal{IB}} \in SO(3)$. The navigation state vector x concatenates these quantities. We choose an IMU-centered navigation framework, where the state represents the state of the IMU frame $\mathcal{B}$. All three state quantities are expressed with respect to the inertial frame $\mathcal{I}$. Apart from the sensor pod state, we estimate sensor-model-specific calibration parameters c. These are the slowly changing IMU accelerometer biases $b_a \in \mathbb{R}^3$, gyroscope biases $b_g \in \mathbb{R}^3$, the constant position receiver antenna phase center $_{\mathcal{B}}r_{\mathcal{BP}} \in \mathbb{R}^3$ and the moving baseline receiver antenna phase center $_{\mathcal{B}}r_{\mathcal{BM}} \in \mathbb{R}^3$. Note that, for simplicity, we assume a single phase center for each GNSS antenna, even though the receivers utilize L_1, L_2, and L_5 bands. Figure 13b shows an example self-calibration, where the GNSS antenna position is refined during the flight to improve global accuracy of the estimate.

$$x = \begin{bmatrix} _{\mathcal{I}}r_{\mathcal{IB}}, \ _{\mathcal{I}}v_{\mathcal{B}}, \ R_{\mathcal{IB}} \end{bmatrix}^T, \qquad\qquad c = \begin{bmatrix} b_a, \ b_g, \ _{\mathcal{B}}r_{\mathcal{BP}}, \ _{\mathcal{B}}r_{\mathcal{BM}} \end{bmatrix}^T. \qquad (10)$$

(a) The geometric relationships.

(b) GNSS antenna position self-calibration.

Figure 13. The localization algorithm estimates the 3D pose of the MAV and its sensors. It can self-calibrate GNSS antenna positions and IMU biases.

The following continuous time model determines the propagation of the state vector x and parameters c. The state is modeled based on kinematic relations, the bias parameters of the IMU are modeled as Brownian motion with standard deviations σ_{ba} and σ_{bg}, and the antenna phase centers are assumed to be constant.

$$\dot{x} = \begin{bmatrix} _\mathcal{I}v_\mathcal{B} \\ _\mathcal{I}a_\mathcal{B} \\ R_{\mathcal{I}\mathcal{B}} \ [_\mathcal{B}\omega_{\mathcal{I}\mathcal{B}}]_\times \end{bmatrix}, \qquad \dot{c} = \begin{bmatrix} \eta_{ba} \\ \eta_{bg} \\ 0 \\ 0 \end{bmatrix}, \qquad \eta_{ba} \sim \mathcal{N}(0, \sigma_{ba}^2 \ I), \qquad \eta_{bg} \sim \mathcal{N}(0, \sigma_{bg}^2 \ I). \tag{11}$$

The $[\cdot]_\times$ operator maps a vector to its skew symmetric matrix, $_\mathcal{I}a_\mathcal{B}$ is the linear acceleration of the base frame and $_\mathcal{B}\omega_{\mathcal{I}\mathcal{B}}$ its angular velocity.

Let $\mathcal{X}$, $\mathcal{C}$ and $\mathcal{Z}$ be the sets of all discrete states, parameters, and measurements available up to the current time T.

$$\mathcal{X} = \{x_i\}_{i=0}^T, \qquad\qquad \mathcal{C} = \{c_i\}_{i=0}^T, \qquad\qquad \mathcal{Z} = \{z_i\}_{i=0}^T. \tag{12}$$

Our estimator constantly reevaluates the maximum a posteriori (MAP) state estimates $\mathcal{X}^*$ and parameter estimates $\mathcal{C}^*$ with every new GNSS measurement available.

$$(\mathcal{X}^*, \mathcal{C}^*) = \arg\max_{(\mathcal{X}, \mathcal{C})} \ p\left(\mathcal{X}, \mathcal{C} | \mathcal{Z}\right). \tag{13}$$

To solve this inference problem efficiently, we use the GTSAM framework (Dellaert et al., 2017). The framework uses factor graphs to model the joint probability density function (PDF) $p\left(\mathcal{X}, \mathcal{C} | \mathcal{Z}\right)$. Essentially, this decomposes the joint PDF into a product of simpler PDFs, where each PDF is dependent only on a single sensor measurement $z_i \in \mathcal{Z}$. For Gaussian noise distributions the MAP estimation problem of Eq. (13) can then be rewritten as the minimization of the sum of residuals

$$\arg\min_{(\mathcal{X}, \mathcal{C})} \sum_i ||h_i(\mathcal{X}, \mathcal{C}) - z_i||_{\Sigma_i}^2, \tag{14}$$

where $||e||_\Sigma^2 = e^T \Sigma^{-1} e$ is the squared Mahalanobis distance and Σ the covariance matrix corresponding to the measurement. The measurement function h_i defines how the states and parameters are related to the respective measurement and needs to be defined for each sensor modality. Furthermore, to minimize Eq. (14) the partial derivatives of h_i with respect to the state x and parameters c need to be known.
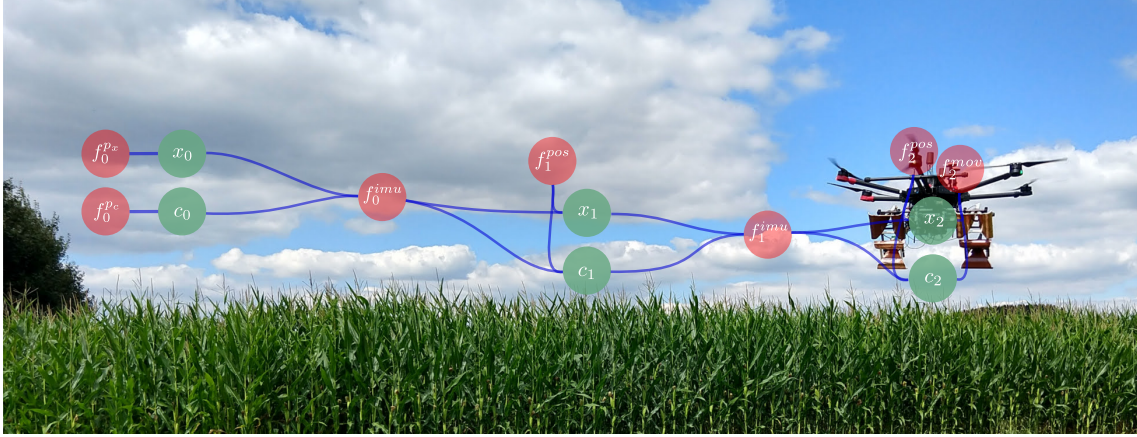
Figure 14. Illustration of the factor graph.

Figure 14 sketches the graph structure of our estimation problem. The estimation process starts with a prior factor $f_0^{p_x}$ constraining the first state and $f_0^{p_c}$ constraining the belief about the initial calibration parameters. Discrete optimization variables x_i and c_i are introduced with every GNSS measurement. These are modeled through unary position factors f_i^{pos} and moving baseline factors f_i^{mov}. The odometry between the discrete times is determined by preintegrated IMU factors f_i^{imu}. In the following, we describe the formulation of each of these factors.

Prior Factor

The states and calibration parameters in the factor graph need to be initialized to start the iterative smoothing. The TRIAD method determines the initial orientation of the platform (Black, 1964). This method computes the rotation matrix between two coordinate frames, i.e., the orientation $R_{\mathcal{IB}}$ of the base frame $\mathcal{B}$ with respect to the inertial frame $\mathcal{I}$, based on two linearly independent vectors expressed in both frames. The first vector pair is the direction of gravity. $_{\mathcal{I}}r_g$, the gravity vector in inertial coordinates, points in the downward-direction. The corresponding gravity vector in base coordinates, $_{\mathcal{B}}r_g$, is the linear acceleration $_{\mathcal{B}}\tilde{a}_{\mathcal{B}}$ measured by the IMU at standstill.

$$_{\mathcal{I}}r_g = \begin{bmatrix} 0, 0, -1 \end{bmatrix}^T, \qquad\qquad\qquad _{\mathcal{B}}r_g = -_{\mathcal{B}}\tilde{a}_{\mathcal{B}}(0). \tag{15}$$

The second vector pair is the baseline vector between the two GNSS antennas. $_{\mathcal{I}}r_b$, the baseline in inertial coordinates, is measured by the GNSS moving baseline receiver $_{\mathcal{I}}\tilde{r}_{PM}$. The baseline vector in base coordinates is the difference between the attitude antenna phase center and position antenna phase center, which is known approximately a priori from CAD data.

$$_{\mathcal{I}}r_b = _{\mathcal{I}}\tilde{r}_{PM}(0), \qquad\qquad\qquad _{\mathcal{B}}r_b = _{\mathcal{B}}r_{\mathcal{BM}}(0) - _{\mathcal{B}}r_{\mathcal{BP}}(0). \tag{16}$$

Following TRIAD, the initial orientation is

$$R_{\mathcal{IB}}(0) = \left[\frac{_{\mathcal{I}}r_g}{||_{\mathcal{I}}r_g||}, \frac{_{\mathcal{I}}r_g \times _{\mathcal{I}}r_b}{||_{\mathcal{I}}r_g \times _{\mathcal{I}}r_b||}, \frac{_{\mathcal{I}}r_g}{||_{\mathcal{I}}r_g||} \times \frac{_{\mathcal{I}}r_g \times _{\mathcal{I}}r_b}{||_{\mathcal{I}}r_g \times _{\mathcal{I}}r_b||} \right] \cdot \left[\frac{_{\mathcal{B}}r_g}{||_{\mathcal{B}}r_g||}, \frac{_{\mathcal{B}}r_g \times _{\mathcal{B}}r_b}{||_{\mathcal{B}}r_g \times _{\mathcal{B}}r_b||}, \frac{_{\mathcal{B}}r_g}{||_{\mathcal{B}}r_g||} \times \frac{_{\mathcal{B}}r_g \times _{\mathcal{B}}r_b}{||_{\mathcal{B}}r_g \times _{\mathcal{B}}r_b||} \right]^T. \tag{17}$$

The initial position is determined from the first GNSS position measurement and the initial position receiver phase center position.

$$_{\mathcal{I}}r_{IB}(0) = _{\mathcal{I}}\tilde{r}_{IP} - R_{\mathcal{IB}}(0) _{\mathcal{B}}r_{\mathcal{BP}}(0). \tag{18}$$

All other parameters, i.e., velocity, gyroscope bias, and accelerometer bias are initialized to zero. Again we assume the vehicle is stationary at startup. This also allows us to calibrate the IMU gyroscope biases to zero at startup by averaging the first gyroscope measurements. The same process cannot be assumed for the accelerometer because we cannot ensure a perfectly leveled platform. The initial uncertainty is left as a tuning parameter.

IMU Factor

The IMU measures the linear acceleration $_{\mathcal{B}}\tilde{a}_{\mathcal{B}}$ and angular velocity $_{\mathcal{B}}\tilde{\omega}_{\mathcal{IB}}$ of the platform. Ignoring effects due to earth's rotation, the measurements are modelled as,

$$_{\mathcal{B}}\tilde{a}_{\mathcal{B}} = R_{\mathcal{IB}}^T \ (_{\mathcal{I}}a_{\mathcal{B}} - {}_{\mathcal{I}}g) + b_a + \eta_a, \qquad\qquad \eta_a \sim \mathcal{N}(0, \sigma_a^2 \ I), \qquad (19)$$

$$_{\mathcal{B}}\tilde{\omega}_{\mathcal{IB}} = {}_{\mathcal{B}}\omega_{\mathcal{IB}} + b_g + \eta_g, \qquad\qquad\qquad \eta_g \sim \mathcal{N}(0, \sigma_g^2 \ I), \qquad (20)$$

where $_{\mathcal{I}}g$ is the gravity vector in inertial coordinates, b_a and b_g are the IMU biases modeled as random walk in Eq. (11) and η_a and η_g are additive white noise with standard deviation σ_a or σ_g respectively. The set of IMU measurements between two GNSS measurements are summarized into a single preintegrated factor as described in (Forster et al., 2015).

Position Factor

The GNSS position receiver measures the distance between the inertial frame and position antenna phase center $_{\mathcal{I}}\tilde{r}_{\mathcal{IP}}$. We can express the position of the position antenna as a function of the current position and orientation estimate.

$$h_{\text{pos}}(x, c) = {}_{\mathcal{I}}r_{\mathcal{IB}} + R_{\mathcal{IB}} \ {}_{\mathcal{B}}r_{\mathcal{BP}}. \qquad (21)$$

The nonzero partial derivatives (Dellaert, 2020, p. 6) are,

$$\frac{\partial h_{\text{pos}}(x, c)}{\partial {}_{\mathcal{I}}r_{\mathcal{IB}}} = I, \qquad\qquad \frac{\partial h_{\text{pos}}(x, c)}{\partial R_{\mathcal{IB}}} = -R_{\mathcal{IB}} \ [{}_{\mathcal{B}}r_{\mathcal{BP}}]_\times, \qquad\qquad \frac{\partial h_{\text{pos}}(x, c)}{\partial {}_{\mathcal{B}}r_{\mathcal{BP}}} = R_{\mathcal{IB}}. \qquad (22)$$

The partial derivatives show that the position measurement not only gives direct feedback on the position estimate $_{\mathcal{I}}r_{\mathcal{IB}}$ but, due to the influence of the lever arm $_{\mathcal{B}}r_{\mathcal{BP}}$, also indirectly measures the orientation $R_{\mathcal{IB}}$ under certain motions. Note that this factor holds for both the RTK GNSS solution, which measures the baseline between the RTK base station antenna and rover antenna, and the SBAS GNSS solution, which measures the position in the world-fixed WGS84 frame, under the assumption that the RTK base station position is determined bias-free with respect to the WGS84 frame. Thus the estimator is robust to switching between the RTK fixed and the SBAS solution.

Moving Baseline Factor

Since the platform orientation from a single GNSS receiver is only observed under certain motions, a second GNSS receiver is added that measures the baseline vector $_{\mathcal{I}}\tilde{r}_{\mathcal{PM}}$ between two GNSS antennas rigidly attached to the platform. This measurement gives direct information about the orientation. The measurement is actually given at the local WGS84 tangent plane centered at the position antenna $\mathcal{P}$ but we assume that for small distances from the inertial frame, this tangent plane is parallel to the inertial frame's east-north-plane. The resulting moving baseline measurement function maps the current estimated difference of the GNSS antenna positions into in the inertial frame.

$$h_{\text{mov}}(x, c) = R_{\mathcal{IB}} \ (_{\mathcal{B}}r_{\mathcal{BM}} - {}_{\mathcal{B}}r_{\mathcal{BP}}). \qquad (23)$$

The nonzero partial derivatives (Dellaert, 2020, p. 6) are

$$\frac{\partial h_{\text{mov}}(x, c)}{\partial R_{\mathcal{IB}}} = R_{\mathcal{IB}} \ [{}_{\mathcal{B}}r_{\mathcal{BP}} - {}_{\mathcal{B}}r_{\mathcal{BM}}]_\times, \qquad \frac{\partial h_{\text{mov}}(x, c)}{\partial {}_{\mathcal{B}}r_{\mathcal{BP}}} = -R_{\mathcal{IB}}, \qquad \frac{\partial h_{\text{mov}}(x, c)}{\partial {}_{\mathcal{B}}r_{\mathcal{BM}}} = R_{\mathcal{IB}}. \qquad (24)$$

The first partial derivative states that the orientation perpendicular to the moving baseline vector, in our case roll and heading, is directly measureable through the moving baseline factor. The other two derivatives, similar to the last derivative in Eq. (22), indicate, that the GNSS antenna positions are observable under orientation changes. This observability is validated in Figure 13b. Here the antenna position estimate is plotted alongside the velocity of the base frame represented in forward-left-up coordinates $\mathcal{F}$. The antenna position is particularly well observed during fast forward flight at $t = 21\,\text{s}$ and $t = 405\,\text{s}$, i.e., where the platform tilts heavily to generate accelerations.

8.1. Algorithm

The localization algorithm runs in three parallel processing threads. The highest priority is given to the navigation thread (Indelman et al., 2013). This thread integrates incoming IMU measurements given the latest available MAP state estimate and delivers a high-rate odometry estimate for control purposes.

The second thread iteratively solves the inference problem stated in Eq. (13). Given the synchronization between IMU and GNSS, and the fixed GNSS solution rate, the time stamps of new factors are known a priori. When a new IMU measurement arrives, it is preintegrated into a new IMU factor up to the next expected GNSS measurement time stamp. Once the IMU factor is completed, it is inserted into the factor graph. The GNSS measurements typically arrive delayed and are inserted at the respective positions into the factor graph. The solver is polled whenever a new factor is added to the graph and the thread is available. We use ISAM2 with a configurable sliding window to reduce inference time and bound graph size and thus relinearization efforts (Kaess et al., 2012).

The third thread can be called on demand, e.g., after completing the mission, and computes the full batch solution for GPSAR mapping purposes. While running the sliding window estimator, the program stores a copy of the full graph, the estimator solution, and all IMU measurements. When the batch solver is triggered, it uses a Levenberg-Marquardt optimizer to find the MAP estimate given the full graph and prior online solution. Additionally, it predicts intermediate states propagating the stored 1 kHz IMU measurements. The sensor poses $T_{\mathcal{IS}_i}$, i.e., the ground-penetrating radar antenna phase center and RGB camera poses, can then be evaluated given the known sensor locations with respect to the IMU $T_{\mathcal{BS}_i}$,

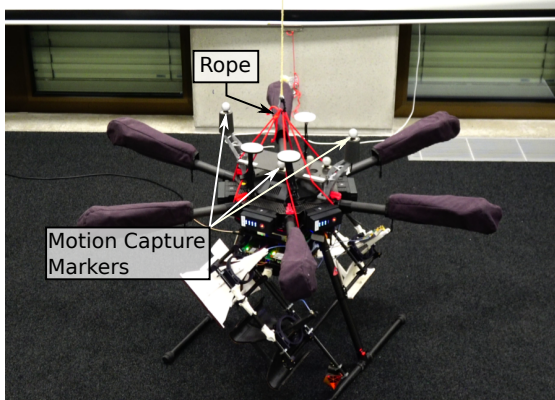$$T_{\mathcal{IS}_i} = T_{\mathcal{IB}} \, T_{\mathcal{BS}_i}. \tag{25}$$

9. Results

In this section we evaluate the state estimation accuracy and overall functionality of the proposed system. As Section 4 lays out, the final application of airborne buried object detection is a function of the flight path, antenna localization, knowledge of the terrain, and millimeter wave signal processing. Thus, evaluating individual components with respect to the downstream task is impossible. Instead, we design individual experiments that allow testing the components presented earlier in isolation. The goal is to practically verify the correct implementation of each subsystem, set a system performance benchmark, and last but not least determine system limitations and performance margins. This methodology allows the reader to draw conclusions for future system designs.
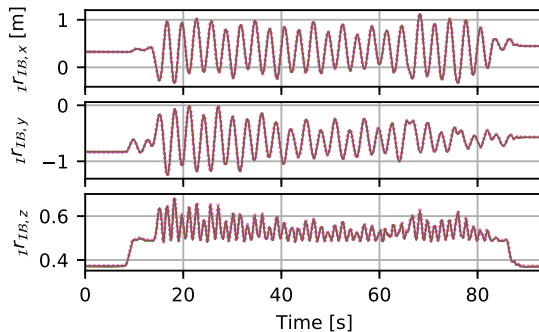
In the first experiment, we validate the algorithmic soundness of the derived estimator in a ground truth motion tracking setting. We demonstrate that the introduction of the second GNSS receiver improves positioning accuracy, in particular for initialization. Furthermore, the experiment shows that the full batch inference computed after landing improves overall estimation smoothness and accuracy over online sliding window estimation. Following this, we conducted two field trials with the GPSAR. First we present mapping of known radar calibration targets placed on the ground surface. The experiments show an overall system sensitivity that allows identifying the scattering characteristics of individual targets. The results also highlight the importance of accurate GNSS antenna calibrations. Furthermore, the experiments suggest that the presented localization method exceeds the resolving power of the underlying imaging solution. As a result of this we present and evaluate a simple alternative solution to localize the radar antennas that still generates focused radar images. Finally, we show the detection of buried metal can lids as a proof-of-concept for landmine detection from an aerial vehicle.

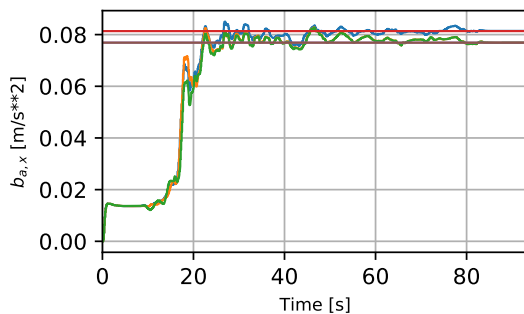### 9.1. Localization Ground Truth Evaluation

The dual GNSS receivers are one of the defining features of the factor graph-based localization algorithm. In this experiment we evaluate their influence to the overall localization accuracy. In
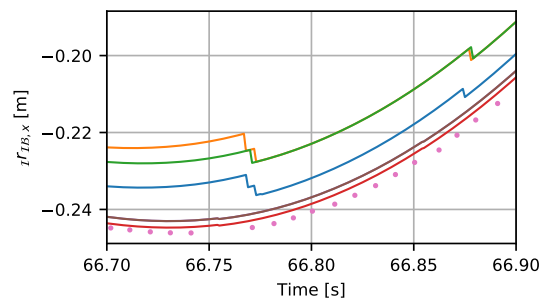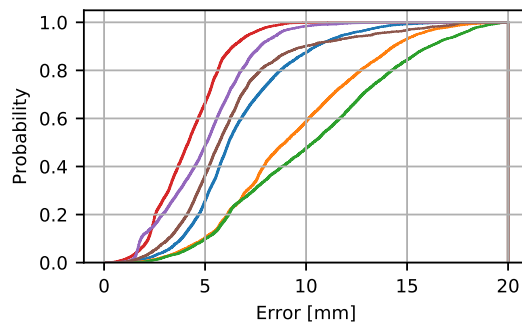
(a) Motion tracking setup.

Configuration	Legend	RMSE	Improvement
Ground truth	▪	0 mm	100 %
Batch (position + moving baseline)	—	**4.24 mm**	**59 %**
Batch (position only)	—	5.02 mm	52 %
Batch (position only + heading offset)	—	6.29 mm	40 %
Online (position + moving baseline)	—	6.74 mm	**36 %**
Online (position only)	—	9.41 mm	10 %
Online (position only + heading offset)	—	10.46 mm	0 %

(b) RMSE of the different configurations.

(c) The experiment trajectory.

(d) Detailed position estimate.

(e) Accelerometer bias estimate.

(f) Cumulative absolute error.

Figure 15. The localization ground truth evaluation using a Vicon motion tracking system. Batch estimation and dual GNSS receivers improve the accuracy and precision of the localization algorithm.

particular, we show the benefits of GNSS moving baseline measurements for initialization and during the online sliding window estimate. Since it is difficult to obtain outdoor ground truth position and orientation measurements that are more accurate than the RTK GNSS used, we use a Vicon motion tracking system to evaluate the state estimation algorithm (Vicon Motion Systems, 2014). Figure 15a shows the experiment setup.

Motion capture markers are attached to the platform to measure the ground truth position and orientation. The measurements simulate noise-free GNSS position and moving baseline

measurements at 10 and 5 Hz, respectively. Due to limited flight space, platform motion is emulated by swinging randomly on a rope for approximately 1.5 min. Before the experiment, the IMU, GNSS, and prior noise models were tuned iteratively, with the goal of obtaining steady bias and GNSS phase center estimates. In post-processing, we run the estimator multiple times with different configurations: with two emulated GNSS receiver factors (position + moving baseline), with only one emulated GNSS receiver factor (position only), and with one emulated GNSS receiver where the initial state is offset by 5° in heading (position only + heading offset). The last configuration resembles a setting where the platform was equipped with only one GNSS receiver and a magnetometer was used for initialization. Besides the three different configurations we also compare the full batch inference with the online estimator with a 3 s sliding window.

Figure 15c plots the complete trajectory over our experiment. At this scale, the different configurations (online, batch, only position measurements, position and moving baseline measurements or with initial heading offset) cannot be distinguished. All estimates agree with the ground truth position. Only the detailed view of the position estimates in Figure 15d reveals that the batch estimation is always smoother than the online estimation. This is expected, since the batch optimizer incorporates all measurements, future and past, while the online estimator only has knowledge of past measurements. The jumps in the online estimation occur because of modeling mismatch between the propagated IMU measurements and the emulated GNSS measurements, e.g., erroneous GNSS phase center calibration, biases, noise models, and time delay. Note that these jumps are in the order of millimeters.

Evaluating the error (Figure 15f) shows that the batch estimation incorporating both position and moving baseline measurements achieves the best RMSE. Table 15b is sorted by ascending RMSE and shows that batch estimation significantly improves estimation accuracy. One reason is that the online estimator is subject to the availability of computational resources and thus has to rely on IMU integration for longer periods before the factor graph is re-evaluated. Another reason is, again, the additional information available to the batch optimizer. For example, the batch optimization has a steady estimate of the IMU acceleration bias over the whole trajectory length exemplified in Figure 15e. A converged bias estimate has a positive influence on the estimation accuracy. An incorrect acceleration bias estimate leads to an attitude estimation error, since the gravity vector is observed incorrectly. This also stresses the importance of (1) finding well-bounded initial estimates for the GNSS's extrinsic calibration and (2) fixing the gyroscope biases before startup to obtain precise results. Equations (22) and (24) reveal that the dynamics of the antenna positions, accelerometer and gyroscope bias are strongly coupled. Estimating all of them simultaneously without good initial guesses would require complex platform movements and extremely accurate modeling assumptions.

The RMSE analysis of Figure 15f and Table 15b also shows that incorporating moving baseline measurements principally improves the accuracy, even when using it only for initialization (position only). However, it is remarkable that the batch estimate with only position measurements is almost as good as with additional moving baseline measurements. On the contrary, the online estimation with moving baseline measurements is significantly more accurate than the online estimate with only position measurements. We suppose that this is due to the influence of the position measurement on the orientation estimate stated in Eq. (22). The orientation is observable under certain motions due to the lever arm between IMU and GNSS antenna. The batch estimation with only position measurements finds a good orientation estimate over all the estimates, while the online estimation first needs to converge. In the case where the orientation is directly measured through the moving baseline measurement, the online estimation can correct the estimate immediately.

Note that this motion tracker experiment does not allow us to draw conclusions on the accuracy and precision of the localization estimate with actual GNSS measurements. Both simulated GNSS measurements and the ground truth have been derived from the same Vicon measurements. This generally biases the estimation towards the ground truth and correlates the position and moving baseline measurements. Additionally, no noise has been added to the simulated measurements, which is not the case for true GNSS measurements. On the other hand, it can be expected that the actual

(a) Target placement.

(b) Left: Corner reflector, right: metal can lid.
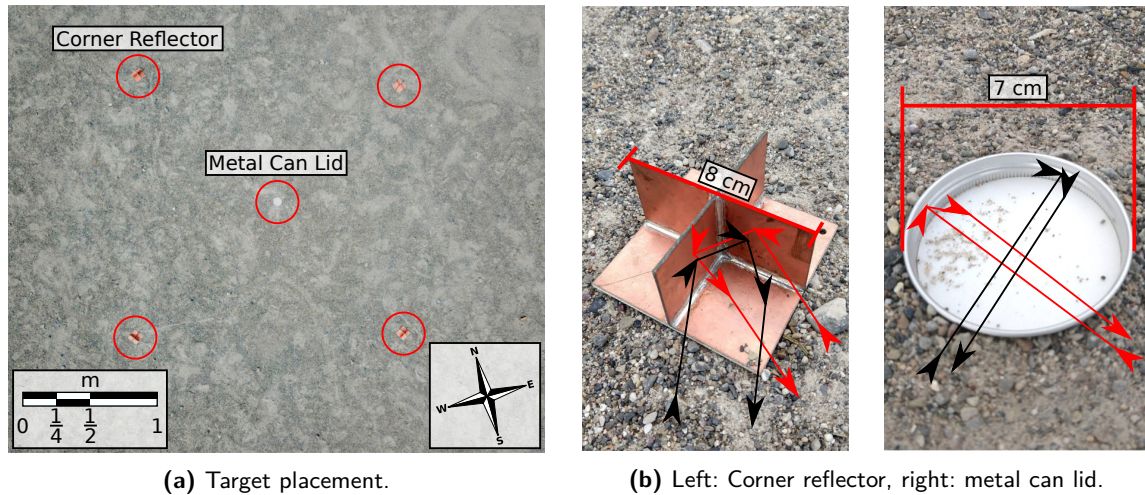
Figure 16. We placed radar calibration targets in a distinctive pattern on the ground. This allows radar back projection independent of the ground surface reflection and soil properties and lets us draw conclusions about the radar antenna positioning performance. (b) The corner reflectors and metal can lid produce distinctive patterns in the radar image.

GNSS timing is better than the Vicon timing. In addition, the GNSS phase center position can probably be determined more accurately than the Vicon marker positions with respect to the IMU. Nevertheless, this experiment still lets us draw the following conclusions:

- batch processing always improves the pose estimation and should be used for mapping purposes;
- online pose estimation benefits significantly from GNSS moving baseline measurements;
- initializing the estimator as accurately as possible improves overall accuracy. This includes orientation initialization from GNSS moving baseline measurements but also finding well-constrained initial parameters for the gyroscope biases or GNSS antenna locations.

9.2. Radar Calibration Target Detection

To evaluate the quality of the radar imaging and thus the quality of the overall navigation and radar antenna localization, we performed a well-controlled radar calibration target imaging experiment. This experiment allows excluding environmental effects, i.e., the necessity for an exact DSM, knowledge of the ground surface refraction and soil properties. Figure 16a shows a photo of the setup. Four corner reflectors were placed on the ground in a square with approximately 2 m distance. In the center of the square was a metal can lid. The targets are expected to reflect the radar waves with certain characteristics. The corner reflectors, as depicted in Figure 16b, should show four distinctive features along the approximate compass directions in the resulting radar image. The metal can lid should show a point symmetrical reflection.

We used the RTK survey station to measure the targets' ground truth position and the imaging plane. The target positions are visualized in the georeferenced map in Figure 17a. The platform flew two missions with six circles each with 15 m diameter centered around the targets. The platform velocity was set to $0.5\,\mathrm{m\,s^{-1}}$ in the first flight and $1\,\mathrm{m\,s^{-1}}$ in the second flight. The flight altitude increased from 2 m to 4 m AGL. To generate constant velocity circular trajectories, the platform thrust was constrained to $9.81 \pm 1.00\,\mathrm{m\,s^{-2}}$, the maximum roll and pitch rate to $\pi/12\,\mathrm{rad\,s^{-1}}$, and the maximum yaw acceleration to $\pi/2\,\mathrm{rad\,s^{-2}}$. Consequently, the automatically generated acceleration and deceleration segments of the polynomial trajectory (see Figure 5b) were sufficient to reach cruising speed at the circle entry and come to a stop after finishing the circle. Figure 17b shows the velocity profile of one of the six circles at $1\,\mathrm{m\,s^{-1}}$.
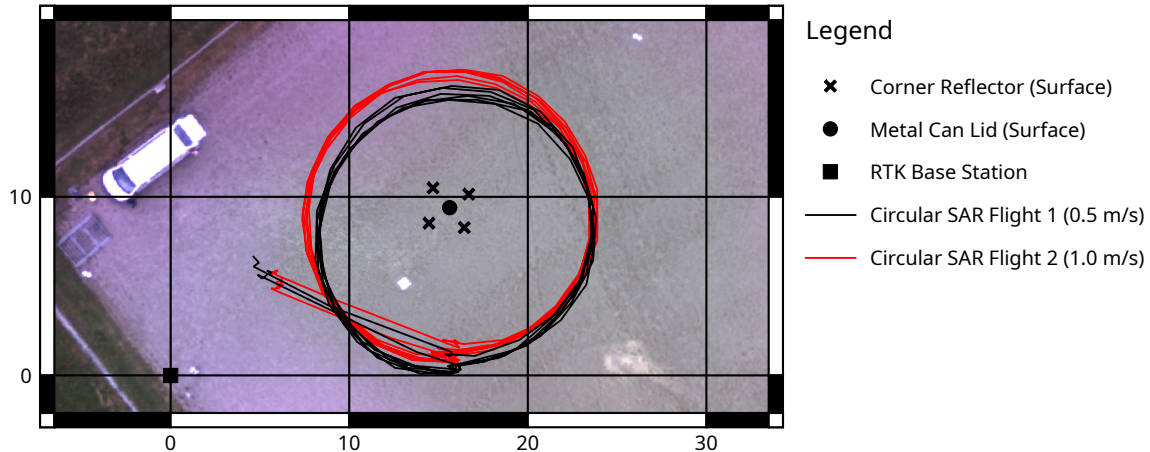
(a) The platform transitions to the circle start at 45 m AGL and performs 6 circles from 2 m to 4 m AGL.
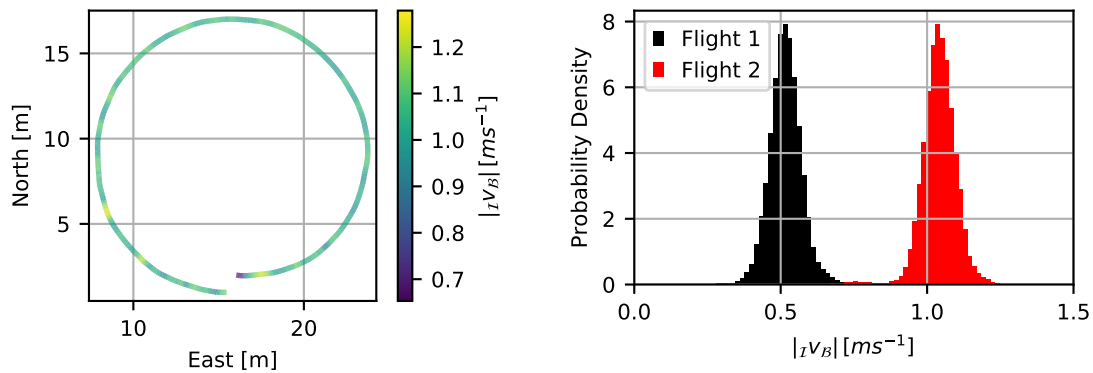

(b) Masked circular trajectory at 2 m AGL and $1\,\mathrm{m\,s^{-1}}$.

**(c)** Velocity distribution of all circular trajectories.

Figure 17. Radar experiment with known surface targets. The dedicated polynomial circular trajectories enforce constant velocity and thus uniform radar sampling. This holds for slow and fast flights.

To generate the radar image for each mission, the radar measurements were masked along the constant velocity circles based on the segment time stamps. As described in Section 6.1, we know at the planning stage how long it takes the MAV to complete any part of the trajectory. This leads to a constant velocity profile and thus a uniform sampling rate across all six circles. Figure 17c shows that this holds both for slow and fast trajectories. The only additional radar measurement filtering step was to exclude saturated radar measurements and the RC frequency band. For each circle the coherent radar image was formed and for each individual flight all six images were added coherently to form the final products.

Figure 18 shows the two resulting radar images at ground surface level with 1 cm cell resolution. Despite the two flight paths in Figure 17a being offset horizontally by approximately 1 m due to the DJI GNSS position control (see Section 6.2) and different reference velocities, both images in Figure 18a and 18b are similar. This highlights the capability of our system to perform repeatable experiments. The main difference is a greater amplitude in the slow flight due to almost twice as many radar measurements back projected into the image. Furthermore, the corner reflector signature, seen in the bottom right of both figures, shifts north-west by approximately 1.5 px (1.5 cm). This systematic shift may be caused by radar imaging artefacts which emerge from the shifted flight paths and the geometry of the corner reflectors (Moreira et al., 2013). Still, one can clearly identify the four radar reflector locations and the location of the metal can lid in both images. A detailed
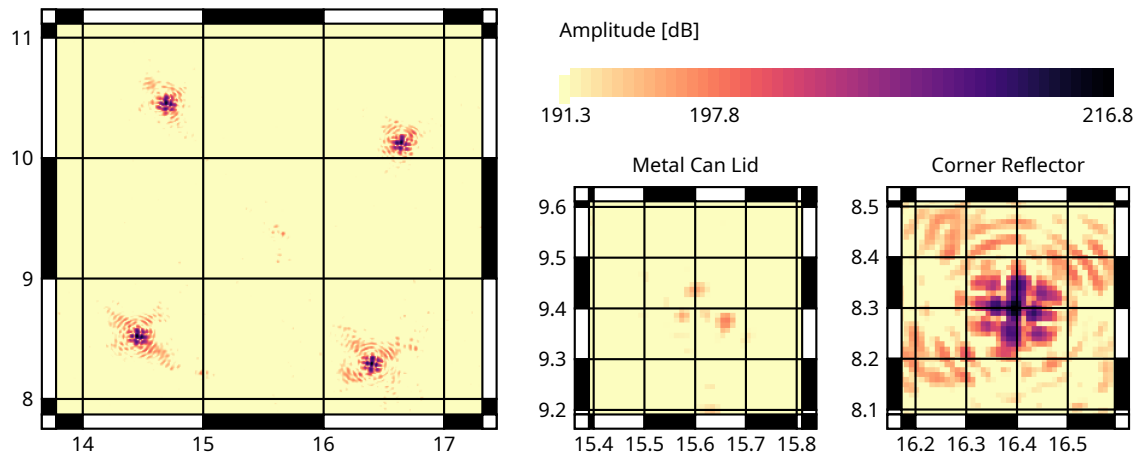
(a) Radar image of flight 1 $(0.5\,\mathrm{m\,s^{-1}})$ evaluated at ground surface level with our localization solution and horizontally polarized antennas. The amplitude is cutoff at the top 2 %.
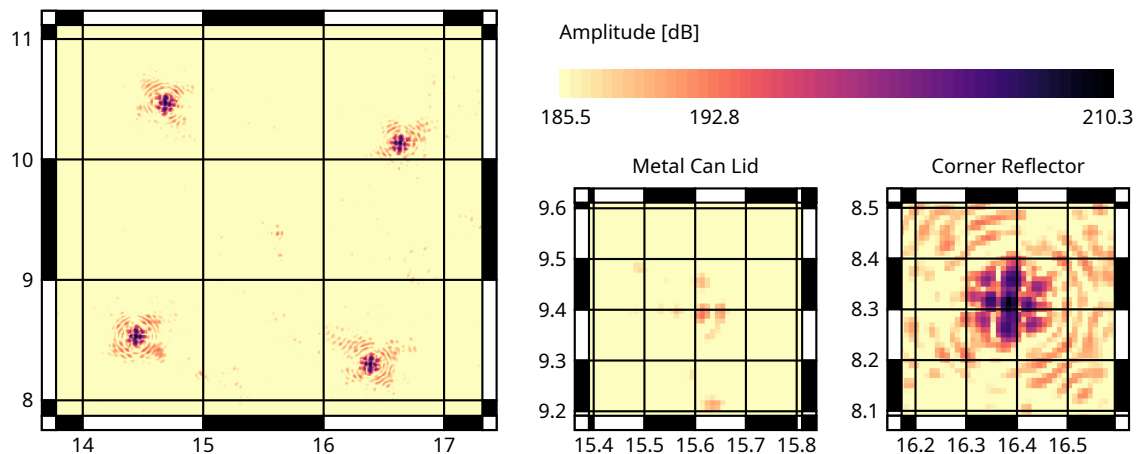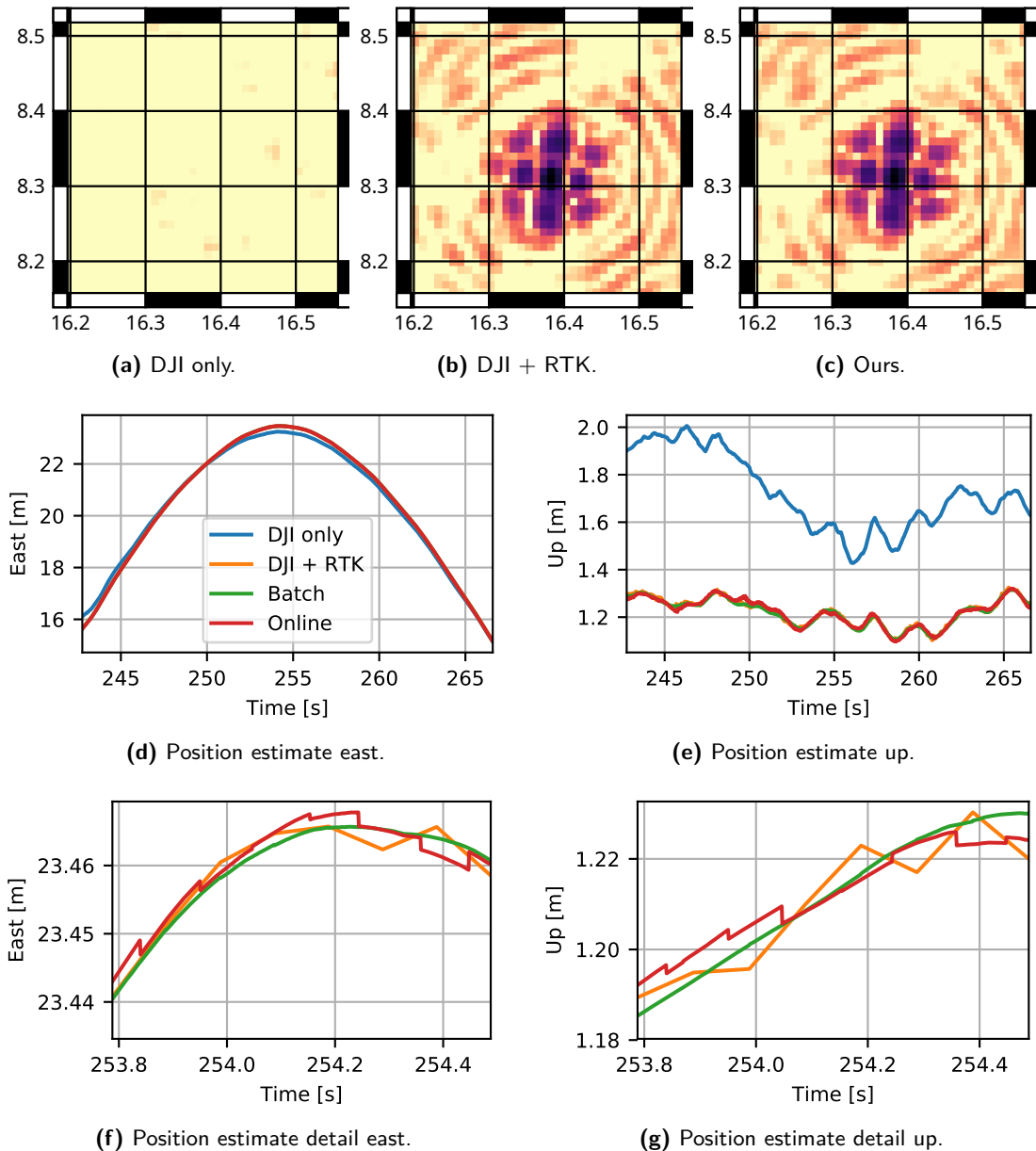

(b) Radar image of flight 2 $(1.0\,\mathrm{m\,s^{-1}})$ evaluated at ground surface level with our localization solution and horizontally polarized antennas. The amplitude is cutoff at the top 2 %.

Figure 18.   Both flights deliver comparable results with a greater amplitude in the slow flight due to approximately twice the number of radar measurements. The similarity of the images shows system repeatability. The metal can lid and the four corner reflectors are detectable. The details show a circular response for the can. The corner reflectors create a well-defined signature with visible corners approximately along the compass directions and surrounding antenna side lobes.

view on the individual targets reveals a point symmetrical response of the metal can lid and the geometric pattern of the corner reflectors. In particular, the four corners stand out pointing towards north, east, south and west. Furthermore, the center of the target has a strong response and the four corner edges of the copper base plate stick out. This demonstrates the sensitivity of the proposed system as well as the repeated localization precision to generate focused radar images. Additionally, periodic circular bands show up around the target, these indicate the side lobes. In the remainder of this section we will investigate only the second flight in detail.

In the next step, we performed an ablation study to investigate the specific impact of our batch antenna localization algorithm. For comparison, we processed the radar returns using only the localization solution provided out-of-the-box by DJI (DJI only) and the solution interpolated from the raw DJI attitude outputs and the raw RTK GNSS positioning (DJI+RTK). We highlight the differences in the resulting radar images by zooming in on a single corner reflector in Figure 19.

(a) DJI only.　　　**(b)** DJI + RTK.　　　**(c)** Ours.

(d) Position estimate east.

(e) Position estimate up.

(f) Position estimate detail east.

(g) Position estimate detail up.

Figure 19. Ablation study analyzing the radar imaging performance of different antenna localization solutions in flight 2. Our GTSAM batch solution delivers the smoothest and possibly most accurate position estimation result. However, a simple DJI+RTK solution is also sufficient for the radar imaging task if the transformations between radar and GNSS antenna are known. The standalone DJI solution is not sufficient.

The differences between the three images are most prominent between the DJI only solution and the other two which use RTK GNSS measurements. The radar image in Figure 19a does not show a focused image in the expected target position. The pure DJI position estimation accuracy is not sufficient to run the back projection algorithm. Figure 19d shows that the DJI only solution is offset by about 20 cm from the RTK solutions in the horizontal direction. The altitude estimation in Figure 19e shows an even more severe drifting offset of up to 70 cm.

On the other hand, both the DJI+RTK and our localization solution provide comparable radar image quality. This is surprising because Figure 19f and 19g show that our batch estimation solution

(a) DJI + RTK.

(b) Ours.
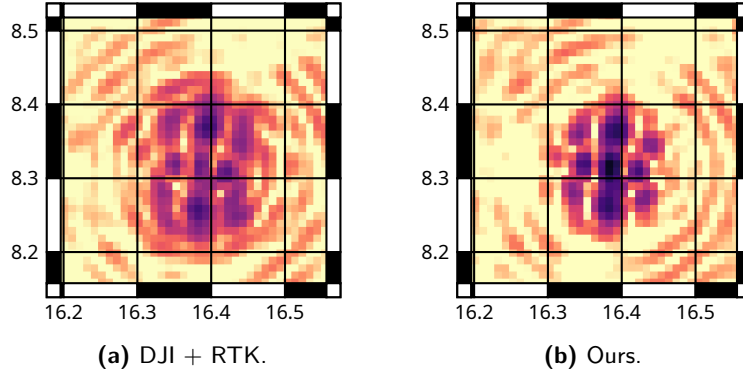
Figure 20. Analysis of system sensitivity to RTK GNSS receiver position offsets in flight 2. Both radar images are generated given a 3 cm offset between the expected and true GNSS receiver position. Our solution is able to perform self-calibration to account for this error, recovering a sharp radar image of similar quality to that shown in Figure 19c.

is smoother and possibly more accurate than the DJI+RTK solution. This indicates, that the downstream task of GPSAR imaging is not sensitive enough to resolve a few millimeters difference in localization. The experiment also demonstrates that the DJI+RTK solution is an attractive alternative to integrate GPSAR on an MAV in situations where our level of accuracy as well as future sensor integration and self-calibration is not required. To obtain the pose of the individual antennas with the DJI+RTK solution $\hat{T}_{\mathcal{IS}_i}$ we solve the following kinematic chain:

$$\hat{T}_{\mathcal{IS}_i} = \tilde{T}_{\mathcal{ID}} \, T_{\mathcal{DS}_i} = \tilde{T}_{\mathcal{ID}} \, T_{\mathcal{DB}} \, T_{\mathcal{BS}_i}, \tag{26}$$

where $\tilde{T}_{\mathcal{ID}}$ is composed of the measured 10 Hz RTK position and interpolated DJI orientation and $T_{\mathcal{DS}_i}$ describes the rigid transform from the RTK position antenna to the respective radar antenna. Note that in our case the latter is obtained from a series of calibration efforts. $T_{\mathcal{DB}}$ is composed of the known rotation from the body frame to DJI frame $R_{\mathcal{BD}}$ and our self-calibrated RTK antenna position $_{\mathcal{B}}r_{\mathcal{BP}}$. $T_{\mathcal{BS}_i}$ is known from CAD. Obtaining this transformation accurately is critical (Yanghuan et al., 2011).

To stress the importance of an accurate GNSS to radar antenna calibration, we investigated the effect of an inaccurate GNSS receiver to base frame transform. In this case, we applied a 3 cm offset in forward, left and up direction to the receiver antenna centers $_{\mathcal{B}}r_{\mathcal{BP}}$ and $_{\mathcal{B}}r_{\mathcal{BM}}$ and re-generated the resulting radar images for the interpolated DJI+RTK localization solution and our localization solution. Figure 20 shows a sharp contrast between the image quality from the two solutions. With only the interpolated position and attitude measurements, the DJI+RTK solution has no way to identify and correct for the inaccurate receiver position, resulting in a blurry radar image. Our solution offers self-calibration in the GTSAM fusion, which is shown for this particular flight in Figure 13b, and is able to recover a sharp radar image despite the initially poor GNSS receiver position estimate.

Finally, we discuss the online and batch solution of our localization algorithm in flight. Figure 19f and 19g show that the batch solution smoothes the online solution significantly, improving localization precision. This is consistent with the Vicon motion tracking experiments in Section 9.1. However, compared to the rope swinging experiment in Figure 15d, the jumps in the online estimate between IMU integration and GNSS update are greater in flight. This indicates that additional effects such as frame vibrations and RTK GNSS noise influence the estimate.

Summarized, this experiment lead to the following findings.

- Polynomial trajectory generation is an adequate technique to generate dedicated GPSAR trajectories that enforce uniform sampling.
- Our self-contained positioning solution enables repeatable, high-resolution radar imaging.
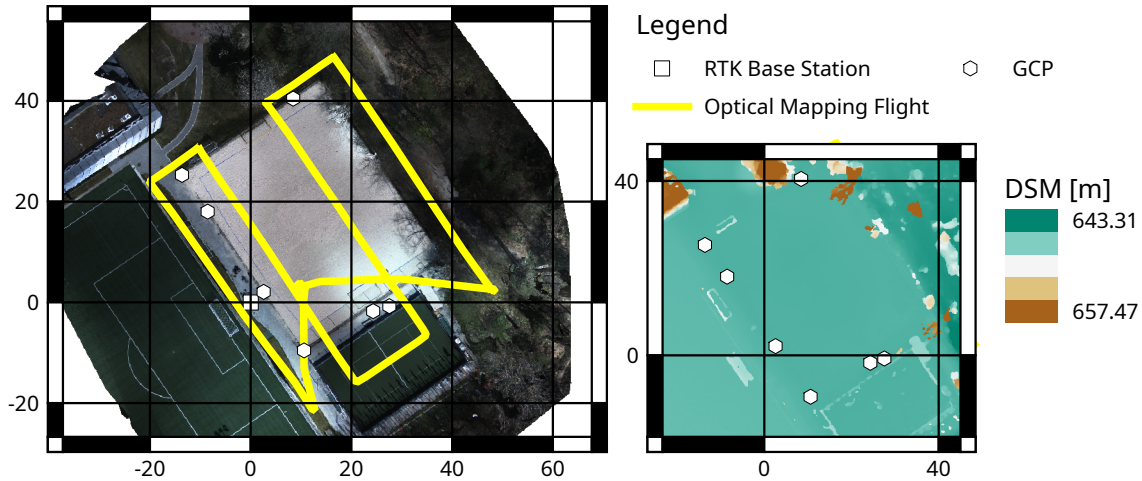
Figure 21. A georeferenced orthomosaic and DSM with 2.5 cm GSD created by our platform. The maps are used to plan the circular GPSAR trajectory and compute the ground surface transition in the radar imaging algorithm.

- Interpolating off-the-shelf RTK GNSS and DJI autopilot attitude is also a viable positioning solution for low-frequency coherent radar imaging.
- Calibrating the radar antenna positions accurately with respect to the GNSS antennas is critical. Our self-calibrating algorithm helps to determine the GNSS antenna positions $_{\mathcal{B}}r_{\mathcal{B}\mathcal{P}}$, $_{\mathcal{B}}r_{\mathcal{B}\mathcal{M}}$ to generate focused images.
- Batch estimation to improve localization precision has been confirmed in flight. Sensor noise such as frame vibrations and GNSS measurement uncertainties may alter the estimation quality.

9.3. Buried Object Detection

Finally, we validated the whole system for buried object detection. The experiment environment is a clean beach soccer field where we buried the same metal can lids as in the previous experiment. Four outer targets forming a square were buried at a depth of around 2–5 cm and a center target was buried at lower depth of around 19 cm. In a first flight, we collected georeferenced images using our system at 45 m altitude. We created the orthomosaic and DSM in Figure 21 with a ground sampling distance (GSD) of 2.5 cm of the environment using off-the-shelf photogrammetry software (Pix4D, 2021). The maps are georeferenced using a set of GCPs placed outside of the target area and surveyed with our equipment.

In a second flight we performed the actual GPSAR survey from safe distance shown in Figure 22. The platform flew six circles with 15 m diameter centered around the target area. The platform velocity was set to $1\,\text{m}\,\text{s}^{-1}$ and the flight altitude increased from 2 m to 4 m AGL over approximately 6 min. The dynamic constraints were identical to the previous experiment, enforcing constant velocity along the circles. The collected radar responses, trajectory time stamp mask, and batch processed radar antenna poses are forwarded to a compute cluster to perform the coherent radar back projection algorithm. The permittivity of the soil was not modified and left at its default value of $\epsilon_r = 8$. The resulting radar image extends $7\,\text{m} \times 7\,\text{m}$ and spans a volume from -0.3–$0.1\,\text{m}$ above surface level with a resolution of 1 cm. Figure 23 shows the resulting GPSAR image at 3 cm and 13 cm depth as well as the root mean square (RMS) radar image cell magnitudes around the ground truth position of the targets. The metal can lids buried at shallow depth can be detected in the radar image. This validates that the system presented in this paper can potentially be used to detect landmines.

The deep object shows the strongest response at 13 cm depth. The offset between the manually measured depth and the peak radar response may be due to several confounding factors that make

(a) Time lapse of an automated 6 min circular GPSAR trajectory with increasing altitudes.
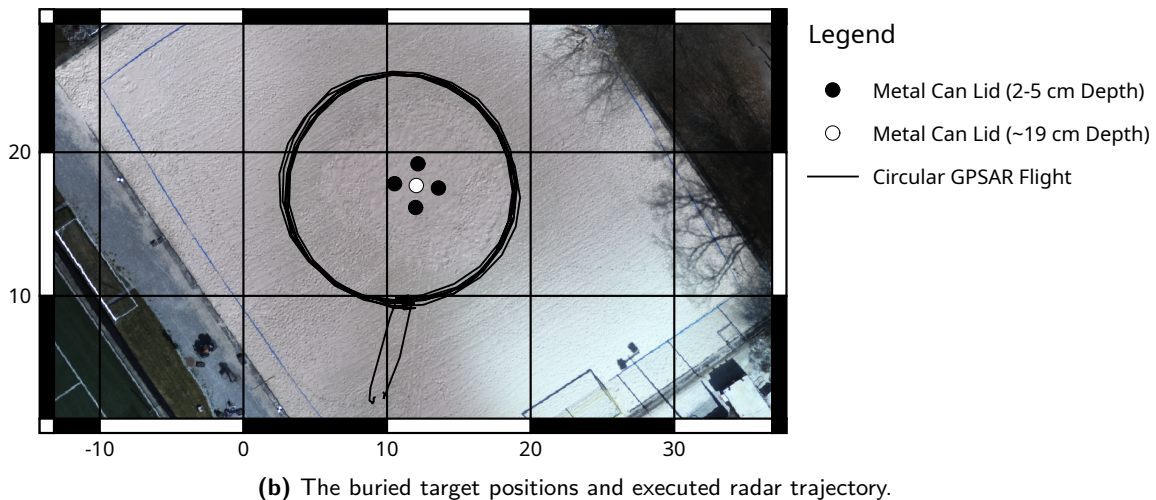

**(b)** The buried target positions and executed radar trajectory.

Figure 22. Buried object detection experiment. The platform scanned a 7 m × 7 m patch without the operators entering the field.

subsurface sensing especially challenging. These include unmodeled complex effects of soil moisture and heterogeneity on the ground permittivity, digital surface map accuracy as well as errors induced through the manual ground truth surveying process. As described in the experiment setup, we do not calibrate ϵ_r to correct the depth.

To confirm that DJI+RTK presented in the previous section is a viable solution for buried object detection as well, we repeat the imaging process. The result in Figure 24 resembles the result obtained with our localization solution in Figure 23. This indicates that localization for GPSAR is not the bottle neck anymore and other error sources outweigh. Possible significant error sources in GPSAR imaging are clutter responses, modelling the millimeter wave propagation in heterogeneous soil, the frequency dependent radar antenna phase center positions, and errors in the surface model. Note that a dedicated localization fusion algorithm such as the one presented in this paper may still give advantages over the DJI+RTK solution. It is extendable to additional sensor modalities in GNSS denied environments, it can include self-calibration routines, and it has greater precision as the previous section suggests. The latter may be relevant to other mapping modalities where greater precision is required, e.g., high-frequency radar, or once the major modelling errors in GPSAR are resolved.
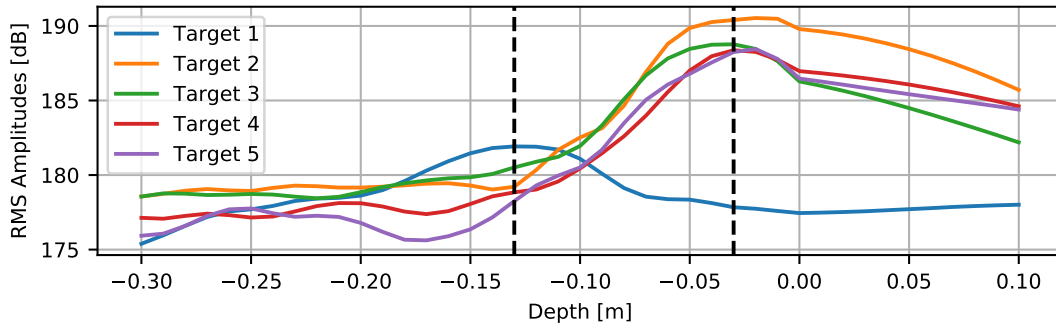
(a) Target RMS amplitudes in a 20 cm × 20 cm image patch around the ground truth position with our antenna positioning solution.
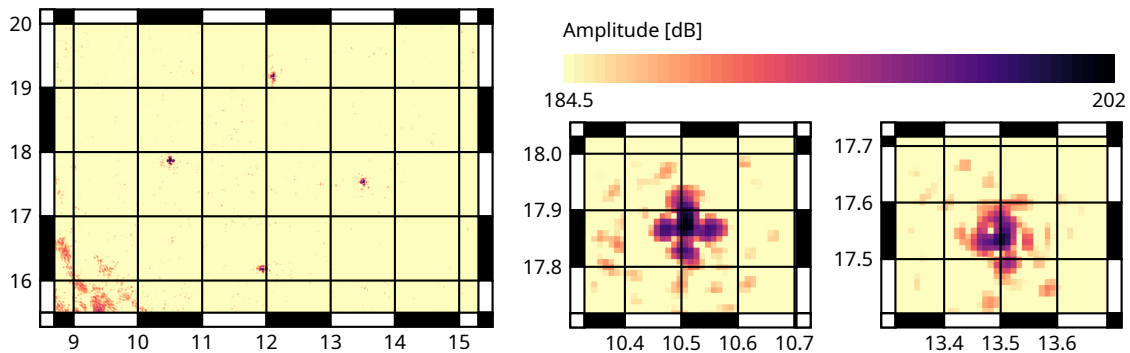
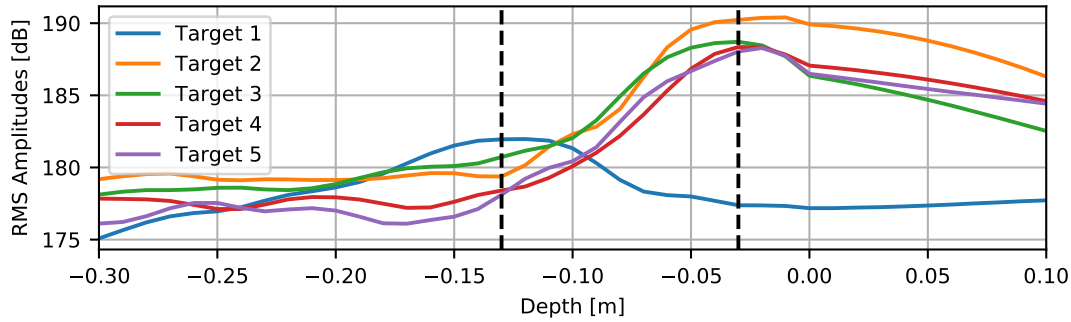**(b)** 3 cm underground with vertically polarized antennas with our positioning solution.
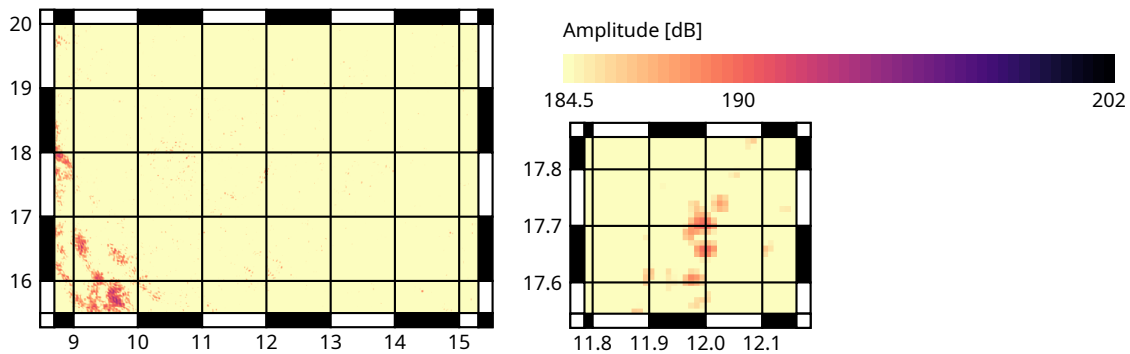
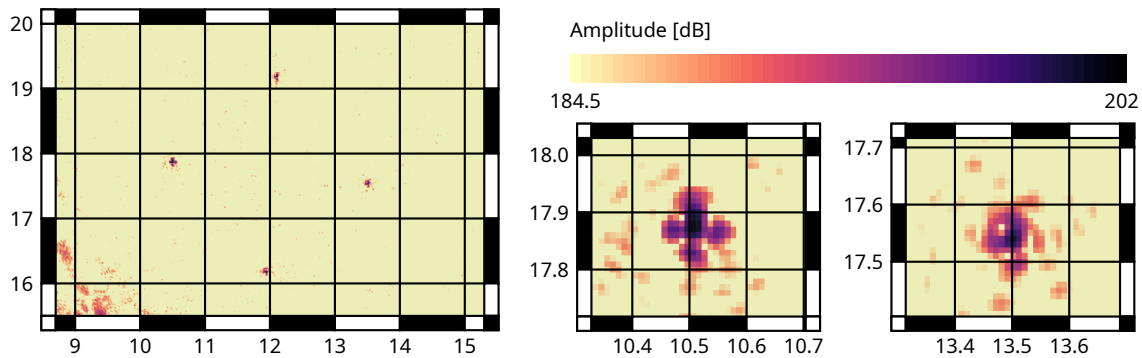**(c)** 13 cm underground with vertically polarized antennas with our positioning solution.

Figure 23. Radar images evaluated at different depths with our localization solution. The shallow targets are clearly visible verifying the capability to detect landmines. The deep object is also detectable, however the challenges of deeper subsurface sensing are also highlighted by the attenuation of the radar signal in (c). Additional clutter shows up in the bottom left-hand corner of the image.

10. Conclusion

In this paper we presented an airborne GPSAR solution capable of detecting buried objects such as landmines. The complete system consists of a navigation stack to fly user-defined autonomous survey missions, a time-synchronized sensor suite and high-accuracy online and offline localization algorithms. Our experiments evaluated the individual contributions of the system components. We demonstrated the robustness of the terrain tracking controller, the 0.8 μs timing accuracy and

(a) Target RMS amplitudes in a 20 cm × 20 cm image patch around the ground truth position with the DJI+RTK antenna positioning.

(b) 3 cm underground with vertically polarized antennas with the DJI+RTK positioning.
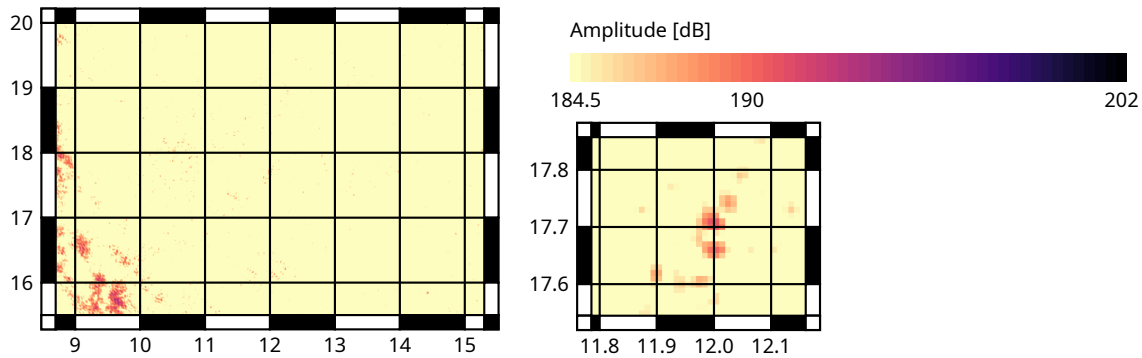
(c) 13 cm underground with vertically polarized antennas with the DJI+RTK positioning.

Figure 24. Radar images evaluated at different depths with the DJI+RTK antenna positioning solution. The similarity to Figure 23 confirms that DJI+RTK is a viable solution for GPSAR imaging. Since the images do not show significant differences, localization does not seem to be the bottleneck of GPSAR anymore. Other radar processing errors, e.g., millimeter wave soil propagation modelling, seem to outweigh localization errors.

0.1 µs precision of the sensor synchronization, the advantage of dedicated radar trajectory generation, and the effectiveness of the localization algorithm due to self-calibration, batch optimization and a dual GNSS receiver setup. Most notably, we demonstrated that this combination of features enables side-looking GPSAR imaging on a rotary wing MAV, an imaging technique which heavily relies on an accurate estimation of the sensor trajectory for reconstruction.

Table 2. List of open-source contributions.

Package	Link	Description (Section)
mav_findmine	github.com/ethz-asl/mav_findmine	Main project repository
		Platform user interface (5.2)
		finite-state machine (5.2)
		Radar trajectory generation (6.1)
		DJI tracking controller (6.2)
		Altitude estimator (6.3)
		Orthomosaic generation utilities (9.3)
polygon_coverage_planning	github.com/ethz-asl/polygon_coverage_planning	Optical survey mission planning (5.2, 9.3)
ethz_piksi_ros	github.com/ethz-asl/ethz_piksi_ros	ROS driver for RTK GNSS
		PPS sync Kernel module (7)
		RTK base and survey station setup (5.2)
mav_gtsam_estimator	github.com/ethz-asl/mav_gtsam_estimator	GTSAM GNSS and IMU fusion (8)
versavis	github.com/rikba/versavis/tree/feature/gnss_sync	*VersaVIS* firmware with PPS sync (7)

The system presented in this paper is a valid alternative to GPSAR on large aerial vehicles (Moussally et al., 2004) as well as GPSAR on MAVs with downward-facing radar (García-Fernández et al., 2020b). The system particularly stands out due to its full, self-contained autonomy. Camera data collected on an initial flight can be used to generate georeferenced DSMs, which allow users to safely plan radar flight missions close to the surface, even in regions beyond visual line of sight. The system enables autonomous radar flights with localization accuracies that allow the generation of crisp radar images in which buried objects are clearly distinguishable. This is an important step towards an aerial device that is capable of detecting landmines in the context of humanitarian demining. In addition, we have open-sourced each of the software components of our system to encourage reuse and further development by the community, the details are provided in Table 2. In fact our research partners have been able to utilize the platform to continue improving radar imaging on MAVs without further assistance. For example, the system enabled InSAR to create DSMs (Burr et al., 2021), as well as the detection of avalanche victims (Grathwohl et al., 2021) or tripwires (Schartel et al., 2020b).

11. Outlook

We believe that airborne radar imaging on MAVs is a unique opportunity to reveal hidden objects in inaccessible places and should be pursued further. Despite the recent progress in the field and the promising results from the presented system, several open research questions remain.

Navigation

We think that the proportional gain control structure utilized in this work is sufficient to operate in a multitude of terrains. A simple way to immediately improve flight accuracy and precision is to replace the DJI pose feedback with our online, sliding window state estimation. A more challenging, but promising task would be the development of radar survey trajectories in undulating and cluttered terrain. The path planning would have to avoid collisions while simultaneously sampling the terrain uniformly. In a first step, the survey surface could be approximated as piecewise linear such that the presented geometric motion primitives are still applicable. In the long run, the detailed terrain and

admissible flight space has to be considered to generate informative surveys (Brüggenwirth and Rial, 2019; Zhang et al., 2018). Additionally, it is important to continue developing radar subsampling strategies to filter informative measurements given the flight path and terrain (García-Fernández et al., 2020a).

Localization

Both sensor timing and localization quality showed sufficient performance margins for GPSAR imaging. However, we reached the point where the downstream imaging task did not discriminate between different localization qualities. In order to continue high-precision state estimation research a precise outdoor ground truth such as a 6DOF total station would be required (Hexagon, 2021). This would allow investigating the GNSS error noise model, and IMU vibration damping (Braun, 2016). Furthermore, quantifiable integration of LiDAR, vision or radar to enable GPSAR imaging in GNSS denied environments would be possible (Zhang and Singh, 2014; Bähnemann et al., 2019; Kramer et al., 2020). Additionally, an automatic calibration routine to determine the frequency-dependent radar antenna phase centers with respect to the IMU could be developed (Doer and Trommer, 2020; Wise et al., 2021). At the moment, this is still determined from CAD and probably one of the largest error sources in the localization pipeline.

Airborne GPSAR in Humanitarian Demining

This and the related works show great potential of using MAVs for demining. From a development viewpoint, it would be important to collaborate with research groups from affected regions in order to transition the technique into a realistic use case. Knowledge of the target environment, such as terrain, typical flight obstacles, vegetation, and landmine types, would help focus the development of the individual components. This includes the navigation and localization, but also the development of the radar imaging. The radar back projection algorithm still assumes a simplified soil model and the surface model is limited by the quality of the DSM. Techniques that deal with heterogeneous environments, clutter, and vegetation still need to be developed. More advanced radar imaging capabilities will automatically demand even better navigation and localization capabilities.

Nevertheless, the results obtained in this work are already applicable across multiple domains, especially in cases where automatic data acquisition and/or accurate outdoor localization is required. Examples include direct photogrammetry, LiDAR mapping, and time-series data acquisition.

Acknowledgments

ORCID

Rik Bähnemann https://orcid.org/0000-0002-2548-7746
Nicholas Lawrance https://orcid.org/0000-0003-2167-7427
Lucas Streichenberg https://orcid.org/0000-0002-7736-0069
Jen Jen Chung https://orcid.org/0000-0001-7828-0741
Michael Pantic https://orcid.org/0000-0002-2410-5152
Alexander Grathwohl https://orcid.org/0000-0002-5892-4017

Christian Waldschmidt ⓘ https://orcid.org/0000-0003-2090-6136
Roland Siegwart ⓘ https://orcid.org/0000-0002-2760-7983

References

Albrektsen, S. M. and Johansen, T. A. (2018). User-Configurable Timing and Navigation for UAVs. *Sensors*, 18(8):2468.

Bähnemann, R., Lawrance, N., Chung, J. J., Pantic, M., Siegwart, R., and Nieto, J. (2021). Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem. In *Field and Service Robotics*, pages 277–290. Springer.

Bähnemann, R., Pantic, M., Popović, M., Schindler, D., Tranzatto, M., Kamel, M., Grimm, M., Widauer, J., Siegwart, R., and Nieto, J. (2019). The ETH-MAV Team in the MBZ International Robotics Challenge. *Journal of Field Robotics*, 36(1):78–103.

Bekar, A., Antoniou, M., and Baker, C. J. (2021). Low-Cost, High-Resolution, Drone-Borne SAR Imaging. *IEEE Transactions on Geoscience and Remote Sensing*.

Black, H. D. (1964). A Passive System for Determining the Attitude of a Satellite. *AIAA journal*, 2(7):1350–1351.

Braun, B. (2016). *High Performance Kalman Filter Tuning for Integrated Navigation Systems*. PhD thesis, Technische Universität München.

Brüggenwirth, S. and Rial, F. (2019). Robotic Control for Cognitive UWB Radar. In *Robotic Intelligence*, pages 145–151. World Scientific.

Burr, R., Schartel, M., Grathwohl, A., Mayer, W., Walter, T., and Waldschmidt, C. (2021). UAV-Borne FMCW InSAR for Focusing Buried Objects. *IEEE Geoscience and Remote Sensing Letters*.

Burr, R., Schartel, M., Schmidt, P., Mayer, W., Walter, T., and Waldschmidt, C. (2018). Design and Implementation of a FMCW GPR for UAV-based Mine Detection. In *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4. IEEE.

Burri, M., Oleynikova, H., Achtelik, M. W., and Siegwart, R. (2015). Real-Time Visual-Inertial Mapping, Relocalization and Planning Onboard MAVs in Unknown Environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1872–1878. IEEE.

Cao, S., Lu, X., and Shen, S. (2021). GVINS: Tightly Coupled GNSS-Visual-Inertial for Smooth and Consistent State Estimation. *arXiv preprint arXiv:2103.07899*.

Cioffi, G. and Scaramuzza, D. (2020). Tightly-coupled Fusion of Global Positional Measurements in Optimization-based Visual-Inertial Odometry. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5089–5095. IEEE.

Colorado, J., Perez, M., Mondragon, I., Mendez, D., Parra, C., Devia, C., Martinez-Moritz, J., and Neira, L. (2017). An integrated aerial system for landmine detection: SDR-based Ground Penetrating Radar onboard an autonomous drone. *Advanced Robotics*, 31(15):791–808.

Cramer, E. A. (2001). The Mineseeker Airship: supporting the UN. In *The Journal of Mine Action*, volume 5, pages 108–113.

Cristoforato, S., Bishop, P., and Thornhill, C. (2000). The feasibility of operating an Ultra Wideband Synthetic Aperture Radar (UWB SAR) from an airship for the detection of mined areas in a humanitarian role. *3rd International Airship Convention and Exhibition*.

Curnow, R. and Lichvar, M. (1997). Chrony - a versatile implementation of the Network Time Protocol (NTP). https://chrony.tuxfamily.org/. Last checked on June 07, 2021.

Daniels, D. J. (2009). Ground Penetrating Radar for Buried Landmine and IED Detection. In *Unexploded Ordnance Detection and Mitigation*, pages 89–111. Springer.

Dellaert, F. (2020). Derivatives and Differentials. https://github.com/borglab/gtsam/blob/4.0.3/doc/math.pdf. Last checked on November 04, 2021.

Dellaert, F., Kaess, M., et al. (2017). Factor Graphs for Robot Perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139.

Ding, W., Wang, J., Li, Y., Mumford, P., and Rizos, C. (2008). Time Synchronization Error and Calibration in Integrated GPS/INS Systems. *ETRI journal*, 30(1):59–67.

Doer, C. and Trommer, G. F. (2020). Radar Inertial Odometry With Online Calibration. In *2020 European Navigation Conference (ENC)*, pages 1–10. IEEE.

Dronecode (2021). Plan - QGroundControl User Guide. https://docs.qgroundcontrol.com/master/en/PlanView/PlanView.html#flight-speed. Last checked on Nov 28, 2021.

Eidson, J. C., Fischer, M., and White, J. (2002). IEEE-1588™ Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, pages 243–254.

Endress, U. (2021). FindMine and Urs Endress Foundation Against Mines and Bombs. https://www.ue-stiftung.org/findmine. Last checked on June 16, 2021.

Esposito, G., Noviello, C., Soldovieri, F., Catapano, I., Fasano, G., Gagliarde, G., Luisi, G., and Saccoccio, F. (2020). The UAV radar imaging prototype developed in the frame of the VESTA project. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–5. IEEE.

Faizullin, M., Kornilova, A., Akhmetyanov, A., Pakulev, K., Sadkov, A., and Ferrer, G. (2021a). Synchronized Smartphone Video Recording System of Depth and RGB Image Frames with Sub-millisecond Precision. *arXiv preprint arXiv:2111.03552*.

Faizullin, M., Kornilova, A., and Ferrer, G. (2021b). Open-Source LiDAR Time Synchronization System by Mimicking GPS-clock. *arXiv preprint arXiv:2107.02625*.

Fang, Z., Yang, S., Jain, S., Dubey, G., Roth, S., Maeta, S., Nuske, S., Zhang, Y., and Scherer, S. (2017). Robust Autonomous Flight in Constrained and Visually Degraded Shipboard Environments. *Journal of Field Robotics*, 34(1):25–52.

Fasano, G., Renga, A., Vetrella, A. R., Ludeno, G., Catapano, I., and Soldovieri, F. (2017). Proof of concept of micro-UAV-based radar imaging. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1316–1323. IEEE.

Forster, C., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Georgia Institute of Technology.

Frey, O., Werner, C. L., Manconi, A., and Coscione, R. (2021). Measurement of surface displacements with a UAV-borne/car-borne L-band DInSAR system: system performance and use cases. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 628–631. IEEE.

Furgale, P., Barfoot, T. D., and Sibley, G. (2012). Continuous-Time Batch Estimation using Temporal Basis Functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095. IEEE.

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified Temporal and Spatial Calibration for Multi-sensor Systems. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

García-Fernández, M., López, Y. Á., and Andrés, F. L.-H. (2019). Autonomous Airborne 3D SAR Imaging System for Subsurface Sensing: UWB-GPR on Board a UAV for Landmine and IED Detection. *Remote Sensing*, 11(20):2357.

García-Fernández, M., López, Y. Á., and Andrés, F. L.-H. (2020a). 3D-SAR Processing of UAV-mounted GPR Measurements: Dealing with Non-Uniform Sampling. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE.

García-Fernández, M., López, Y. Á., and Andrés, F. L.-H. (2020b). Airborne Multi-Channel Ground Penetrating Radar for Improvised Explosive Devices and Landmine Detection. *IEEE Access*, 8:165927–165943.

García-Fernández, M., López, Y. Á., Arboleya, A. A., Valdés, B. G., Vaqueiro, Y. R., Andrés, F. L.-H., and García, A. P. (2018). Synthetic Aperture Radar Imaging System for Landmine Detection Using a Ground Penetrating Radar on Board a Unmanned Aerial Vehicle. *IEEE Access*, 6:45100–45112.

Grathwohl, A., Hinz, P., Burr, R., Steiner, M., and Waldschmidt, C. (2021). Experimental Study on the Detection of Avalanche Victims using an Airborne Ground Penetrating Synthetic Aperture Radar. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–6. IEEE.

Heinzel, A., Schartel, M., Burr, R., Bähnemann, R., Schreiber, E., Peichl, M., and Waldschmidt, C. (2019). A comparison of ground-based and airborne SAR systems for the detection of landmines, UXO, and IEDs. In *Radar Sensor Technology XXIII*, volume 11003, page 1100304. International Society for Optics and Photonics.

Hexagon (2021). Leica Absolute Tracker AT960. https://www.creativeinfocom.com/pdfs/leica-absolute-tracker-at930-brochure.pdf. Last checked on October 22, 2021.

Huck, T., Westenberger, A., Fritzsche, M., Schwarz, T., and Dietmayer, K. (2011). Precise Timestamping and Temporal Synchronization in Multi-Sensor Fusion. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 242–247. IEEE.

Indelman, V., Williams, S., Kaess, M., and Dellaert, F. (2013). Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, 61(8):721–738.

International Campaign to Ban Landmines (2020). Landmine Monitor 2020. Technical report, Cluster Munition Coalition (ICBL-CMC).

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235.

Kais, M., Millescamps, D., Bétaille, D., Lusetti, B., and Chapelon, A. (2006). A Multi-Sensor Acquisition Architecture and Real-Time Reference for Sensor and Fusion Methods Benchmarking. In *2006 IEEE Intelligent Vehicles Symposium*, pages 418–423. IEEE.

Kaplan, E. and Hegarty, C. (2005). *UNDERSTANDING GPS: PRINCIPLES AND APPLICATIONS*. Artech house.

Kelly, J., Grebe, C., and Giamou, M. (2021). A Question of Time: Revisiting the Use of Recursive Filtering for Temporal Calibration of Multisensor Systems. *arXiv preprint arXiv:2106.00391*.

Kelly, J. and Sukhatme, G. S. (2014). A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors. In *Experimental Robotics*, pages 195–209. Springer.

Kramer, A., Stahoviak, C., Santamaria-Navarro, A., Agha-mohammadi, A.-a., and Heckman, C. (2020). Radar-Inertial Ego-Velocity Estimation for Visually Degraded Environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5739–5746. IEEE.

Li, M. and Mourikis, A. I. (2014). Online temporal calibration for camera–IMU systems: Theory and algorithms. *The International Journal of Robotics Research*, 33(7):947–964.

Li, X., Zhang, X., Ren, X., Fritsche, M., Wickert, J., and Schuh, H. (2015). Precise positioning with current multi-constellation Global Navigation Satellite Systems: GPS, GLONASS, Galileo and BeiDou. *Scientific reports*, 5:8328.

Looney, M. (2018). ADIS16448 Data Sampling. https://ez.analog.com/mems/w/documents/4122/adis16448-data-sampling. Last checked on June 07, 2021.

Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., and Siegwart, R. (2013). A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 3923–3929. IEEE.

Mian, O., Lutes, J., Lipa, G., Hutton, J., Gavelle, E., and Borghini, S. (2015). DIRECT GEOREFERENCING ON SMALL UNMANNED AERIAL PLATFORMS FOR IMPROVED RELIABILITY AND ACCURACY OF MAPPING WITHOUT THE NEED FOR GROUND CONTROL POINTS. *The international archives of photogrammetry, remote sensing and spatial information sciences*, 40(1):397.

Mirage Systems Inc. (1984). Mirage Systems - a subsurface imaging, technology development and systems company. https://miragesystems.com/. Last checked on Nov 24, 2021.

Moreira, A., Prats-Iraola, P., Younis, M., Krieger, G., Hajnsek, I., and Papathanassiou, K. P. (2013). A Tutorial on Synthetic Aperture Radar. *IEEE Geoscience and remote sensing magazine*, 1(1):6–43.

Moussally, G. J., Fries, R. W., and Bortins, R. (2004). Ground-penetrating synthetic-aperture radar for wide-area airborne minefield detection. In *Detection and Remediation Technologies for Mines and Minelike Targets IX*, volume 5415, pages 1042–1052. International Society for Optics and Photonics.

Mueller, M. W., Hehn, M., and D'Andrea, R. (2015). A Computationally Efficient Motion Primitive for Quadrocopter Trajectory Generation. *IEEE Transactions on Robotics*, 31(6):1294–1310.

Nikolic, J., Rehder, J., Burri, M., Gohl, P., Leutenegger, S., Furgale, P. T., and Siegwart, R. (2014). A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 431–437. IEEE.

Osadcuks, V., Pudzs, M., Zujevs, A., Pecka, A., and Ardavs, A. (2020). Clock-based time synchronization for an event-based camera dataset acquisition platform. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4695–4701. IEEE.

Peichl, M., Schreiber, E., Heinzel, A., and Kempf, T. (2014). TIRAMI-SAR-a synthetic aperture radar approach for efficient detection of landmines and UXO. In *EUSAR 2014; 10th European Conference on Synthetic Aperture Radar*, pages 1–4. VDE.

Pix4D (2021). PIX4Dmapper photogrammetry software. https://www.pix4d.com/product/pix4dmapper-photogrammetry-software. Last checked on June 07, 2021.

Pradalier, C. (2017). A task scheduler for ROS. Technical Report UMI 2958, GeorgiaTech-CNRS. Last checked on May 25, 2021.

Qin, T. and Shen, S. (2018). Online Temporal Calibration for Monocular Visual-Inertial Systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE.

Richter, C., Bry, A., and Roy, N. (2016). Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. In *Robotics research*, pages 649–666. Springer.

Roumeliotis, S. I. and Burdick, J. W. (2002). Stochastic Cloning: A generalized framework for processing relative state measurements. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1788–1795. IEEE.

Schartel, M. (2021). *Signalverarbeitungskonzepte zur Minendetektion mittels drohnengestütztem Ground Penetrating Synthetic Aperture Radar*. PhD thesis, Universität Ulm.

Schartel, M., Burr, R., Bähnemann, R., Mayer, W., and Waldschmidt, C. (2020a). An Experimental Study on Airborne Landmine Detection Using a Circular Synthetic Aperture Radar. *arXiv preprint arXiv:2005.02600*.

Schartel, M., Burr, R., Mayer, W., Docci, N., and Waldschmidt, C. (2018). UAV-Based Ground Penetrating Synthetic Aperture Radar. In *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4. IEEE.

Schartel, M., Grathwohl, A., Schmid, C., Burr, R., and Waldschmidt, C. (2020b). TRIPWIRE DETECTION IN SAR IMAGES USING A MODIFIED RADON TRANSFORM. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 746–749. IEEE.

Schiebel Corporation (1997). CAMCOPTER S-100 Technical Specifications. https://schiebel.net/products/camcopter-s-100-system-2/. Last checked on Nov 24, 2021.

Schneider, J., Eling, C., Klingbeil, L., Kuhlmann, H., Förstner, W., and Stachniss, C. (2016). Fast and Effective Online Pose Estimation and Mapping for UAVs. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4784–4791. IEEE.

Schreiber, E., Heinzel, A., Peichl, M., Engel, M., and Wiesbeck, W. (2019). Advanced Buried Object Detection by Multichannel, UAV/Drone Carried Synthetic Aperture Radar. In *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE.

Šipoš, D. and Gleich, D. (2020). A lightweight and low-power UAV-borne ground-penetrating radar design for landmine detection. *Sensors*, 20(8):2234.

Skog, I. and Handel, P. (2011). Time Synchronization Errors in Loosely Coupled GPS-Aided Inertial Navigation Systems. *IEEE transactions on intelligent transportation systems*, 12(4):1014–1023.

Sommer, H., Khanna, R., Gilitschenski, I., Taylor, Z., Siegwart, R., and Nieto, J. (2017). A low-cost system for high-rate, high-accuracy temporal calibration for lidars and cameras. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2219–2226. IEEE.

Strode, P. R. and Groves, P. D. (2016). GNSS multipath detection using three-frequency signal-to-noise measurements. *GPS solutions*, 20(3):399–412.

Svedin, J., Bernland, A., and Gustafsson, A. (2021). Small UAV-based high resolution SAR using low-cost radar, GNSS/RTK and IMU sensors. In *2020 17th European Radar Conference (EuRAD)*, pages 186–189. IEEE.

Swift Navigation (2019). Piksi Multi - GNSS Module Hardware Specification. https://www.swiftnav.com/resource-files/Piksi%20Multi/v2.2/Specification/Piksi%20Multi%20HW%20Specification%20v2.2%20%5B000-534-02-02%5D.pdf. Last checked on June 07, 2021.

Tschopp, F., Riner, M., Fehr, M., Bernreiter, L., Furrer, F., Novkovic, T., Pfrunder, A., Cadena, C., Siegwart, R., and Nieto, J. (2020). VersaVIS—An Open Versatile Multi-Camera Visual-Inertial Sensor Suite. *Sensors*, 20(5):1439.

United Nations Mine Action Service (UNMAS) (2019). IMAS 08.10 - Non-technical survey. https://www.mineactionstandards.org/fileadmin/MAS/documents/standards/IMAS-08.10-Ed1-Am4-Non-Technical_Survey.pdf. Last checked on September 27, 2021.

Vicon Motion Systems (2014). Bonita Optical Motion Tracking. https://est-kl.com/images/PDF/Vicon/Bonita_Optical.pdf. Last checked on June 07, 2021.

Wang, J., Li, Y., Zhou, Z., Jin, T., Yang, Y., and Wang, Y. (2008). Image Formation Techniques for Vehicle-Mounted Forward-Looking Ground Penetrating SAR. In *2008 International Conference on Information and Automation*, pages 667–671. IEEE.

Wise, E., Peršić, J., Grebe, C., Petrović, I., and Kelly, J. (2021). A Continuous-Time Approach for 3D Radar-to-Camera Extrinsic Calibration. *arXiv preprint arXiv:2103.07505*.

Yanghuan, L., Fulai, L., Qian, S., and Zhimin, Z. (2011). Lever Arm Rotation Compensation Method for UAV Mounted SAR. In *2011 3rd International Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, pages 1–3. IEEE.

Zaugg, E. C. and Long, D. G. (2015). Generalized Frequency Scaling and Backprojection for LFM-CW SAR Processing. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7):3600–3614.

Zhang, J., Chadha, R. G., Velivela, V., and Singh, S. (2018). P-CAP: Pre-computed Alternative Paths to Enable Aggressive Aerial Maneuvers in Cluttered Environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8456–8463. IEEE.

Zhang, J. and Singh, S. (2014). LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science & Systems (RSS)*. IEEE.