UNIVERSITY POLITEHNICA OF BUCHAREST

Doctoral School of Automatic Control and Computers

Department of Computer Science

Ph.D. THESIS SUMMARY

Ing. George-Cristian Pătru

Soluții de Localizare pentru Robotică în Medii fără GPS

Solutions for Robotics in GPS-Denied Environments

THESIS COMMITTEE

Prof. Dr. Ing. Florin Pop Politehnica University of Bucharest	President
Prof. Dr. Ing. Răzvan-Victor Rughiniş Politehnica University of Bucharest	PhD Supervisor
Conf. Dr. Ing. Daniel Rosner Politehnica University of Bucharest	Referee
Prof. Dr. Răzvan Bologa Bucharest University of Economic Studies	Referee
Conf. Dr. Ing. Mircea Hulea „Gheorghe Asachi” Technical University of Iasi	Referee

BUCHAREST 2023

Abstract

Autonomous robots are becoming ubiquitous in modern society - from industrial and commercial applications - to entertainment and educational applications. With the reduction in cost size and the increase in application area - they are becoming more and more present not just in outdoor settings - but in indoor and other scenarios with a limited GPS signal as well. This increase in demand and applications will continue, driven by a combination of rising demand for automated small-scale robots, advancements in embedded computation power, improvements in sensor precision, and emerging radio chips.

The current thesis explores solutions for autonomous robotic positioning in GPS-denied environments - with a focus on solutions for small-scale, low-power (embedded-driven) robots. The thesis explores approaches and solutions and proposes novel paths for efficient and flexible positioning solutions, testing out practical applications on real-life environments.

The thesis is structured into one introductory chapter, one State of the Art chapter, three contributions chapters, and one conclusion chapter. The first contribution chapter is focused on visual servoing approaches for drone tracking - with applications in scenarios that required fast and accurate relative positioning between autonomous drones and other mobile targets of devices - and presents solutions focused on fiducial markers and a complex hardware-software solution for real-time autonomous tracking of a moving target. The second contribution chapter is focused on UWB-based localization - with both software and hardware approaches for implementing a high-precision, high-speed indoor positioning system for indoor robotic applications that required both high precision and fast localization. A complete solution - FlexTDOA - is proposed and tested in numerous complex scenarios - including comparisons for Line of Sight (LOS) and Non-Line of Sight (NLOS) scenarios. The third contribution chapter is built around a practical application for an autonomous robot designed for an industrial application - automated tire measurements - that employs a navigation algorithm based on sensor fusion between LIDAR and optical flow.

All the chapters are built around hardware and software solutions used to validate the proposed mathematical models and software algorithms - the thesis setting up the basis for future exploration of next-gen robotics applications in indoor and GPS-denied environments.

Table of contents

Chapter 1

Introduction

1.1 Presentation of the Field of the Doctoral Thesis

Autonomous robots, both ground-based and aerial-based, are becoming omnipresent, evolving into an integral part of both our daily lives and of many industrial and commercial applications. As we rely more and more on them for a multitude of tasks, we also need to empower them with better localization, navigation, and capabilities for the understanding of their surrounding. Applications for autonomous mobile robots range from industrial ones (moving heavy payloads, fast moving of postal packets, mobile cleaning, inspection, or aiding in critical situations), and commercial applications (in retail, in office spaces, and anywhere a payload delivery or a repetitive task is necessary) to services of specialized fields (medical, military, scientific research, journalism, and entertainment).

This is not just momentary hype. Robots are here to stay and to expand their use, driven by a mixture of advancements in sensors, better algorithms, advancements in radio communication and the ubiquity of wireless communication technologies, lower prices for controllers and off-the-shelf electronic components - as well as an increase in computing power available in low-cost and low-power packages.

We are, however, far away from a Terminator-like scenario, in which mobile robots would be interpreting the world around us with a human-like approach and precision. While autonomous cars might be getting closer to this, they generally rely on tremendous processing power, and high-cost sensors and they have several power consumption limitations (in comparison with the power used by the drive-train). For “day-to-day” robots, especially the ones used indoors, which are constrained in terms of physical size, electrical power availability, and price tag, current solutions must be based on smarter solutions that use smaller computing platforms and lighter and cheaper sensors (no Radar, no 4 x HD cameras and 4 x LIDAR) to navigate their environment.

Nonetheless, the above-mentioned progress in computation power for embedded systems and better low-power, low-cost sensors is indeed helpful, opening even wider the existing research field dedicated to how to position and help navigate autonomous robots in indoor / GPS-denied scenarios. The current thesis explores various state-of-the-art solutions for autonomous robotic positioning and proposes novel approaches, more efficient and flexible, with both outdoor and indoor applications, testing out practical applications of the proposed solutions in real-life applications.

1.2 Scope of the Doctoral Thesis

The thesis aims to explore, test out and validate solutions that make the best out of new embedded compute and wireless solutions for small dimensions autonomous robots (both for ground-based robots and for aerial drones) positioning and navigation.

The thesis aims to answer the following research questions:

(Q1) How can localization methods be improved in GPS-denied environments to achieve the best results for small, power-constrained autonomous robot applications?

(Q2) How does the usage of current novel wireless chips improve positioning accuracy on small-scale, embedded system based robots?

(Q3) How to improve results in scenarios involving real-time relative positioning between two 3D systems (e.g.: an aerial drone versus a mobile target)?

1.3 Content of the Doctoral Thesis

The thesis is structured into 6 chapters: one introductory chapter, one State of the Art chapter exploring work related to the subject of the current Ph.D. thesis, three contributions chapters, and one conclusion chapter that summarises the thesis contributions and sets the ground for future exploratory research work. The first chapter (the current one) introduces the thesis and the motivation behind it.

The second chapter explores the current state of the art in the field of positioning and navigation solutions for robotic applications for GPS-denied environments - ranging from indoor environments to outdoor environments with geographical, structural, or ambient limitations that force researchers and developers to use alternative technical solutions for precision positioning, orientation, and navigation. This chapter first explores general characteristics, and limitations, and discusses examples of applications where alternative solutions are needed. Then, a series of techniques that are used by state-of-the-art approaches are presented, through discussion examples of recent literature publications: classic WiFi-based solutions used in indoor environments with pre-existing WiFi infrastructure that is exploited as positioning beacons; UWB (Ultra Wide Bandwidth) based solutions that promise better accuracy using new generation chips; vision based solutions that use advancements in image-focused computational power which enable registering visual cues as a reference, advanced object tracking, path estimation, and positioning estimation; visual servoing implementations that are used with a focus on final task / final trajectory corrections for precision landing, docking, and object delivery; as well as the classic dead-reckoning approach that is making a comeback thanks to advancements in sensors, embedded processing, and tracking algorithms.

The first contribution chapter explores solutions for drone tracking, with a focus on visual servoing approaches. This is a priority chapter, as the last years have brought about a multitude of use cases for precision landing and delivery, in-flight docking, and in other scenarios that required fast and accurate relative positioning between a moving robotic platform and a static or moving secondary platform / “target”. The first sub-chapter explores fiducial markers variants with a focus on determining the best alternatives to be used for UAVs (Unmanned Aerial Vehicle) usage, while the second sub-chapter explores a complex implementation for real-time tracking of a moving target from a moving 3D platform (aerial) - using sensor fusion and fast real-time processing in order to achieve a fast and reliable target acquisition.

The second contribution chapter explores the broader subject of positioning in any environment - and it is focused on UWB-based localization. This chapter explores both software and hardware approaches for implementing a high-precision indoor positioning system that can be used for indoor robotic applications that required both high precision and fast localization. It presents how the system is built up from basic distance measurement techniques (Single-Sided Two-Way Ranging and Time Difference of Arrival), then explores location computing algorithms (EKF - Extended Kalman Filter based and LSE - Least Square Error minimization based), and then a flexible time-division multiple-access scheme in which time is divided into slots, which manage when every anchor and can start communication. Based on this, a complete solution - FlexTDOA - is proposed and tested in numerous complex scenarios - including comparisons for Line of Sight (LOS) and Non-Line of Sight (NLOS) scenarios.

The third contribution chapter explores applied scenarios and solutions for precision navigation without GPS for robotic applications. This chapter is built around a practical application for an autonomous robot designed for an industrial application - automated tire measurements. The proposed solution contains multiple hardware and software implementations, including a navigation algorithm based on sensor fusion between LIDAR and optical flow, a complete industrial robust hardware implementation, as well as specific algorithms for image processing and high-accuracy measurements.

The conclusions chapter summarises the thesis drawing the main conclusions and presents results both in terms of contributions (21 original contributions) and in terms of publications: one Q1 journal as first author, one Q1 journal as second author, 3 publications as first author, and other 7 co-authored papers.

Chapter 2

Related Work

Navigation, mapping, surveying, and robotics require exact placement. In urban canyons, dense woodlands, and interior spaces, accuracy and precision might be difficult. Multipath, signal attenuation, and interference can reduce positioning system performance in certain settings. Obstacles and dynamic surroundings can also cause position, velocity, and orientation problems. Hence, academics have offered numerous methods to solve these issues and increase positioning system accuracy and precision. The state-of-the-art chapter will cover some of these techniques using tech literature publications that have made substantial contributions to this subject. We'll start with several papers on precise location in different contexts, focusing on GPS-denied localization methods. After that, we'll discuss six localization technologies that meet accuracy requirements. The first portion covers WiFi-based localization, the second UWB, the third and fourth computer vision and visual servoing, the fourth IMU-only localization, and the sixth fusion-based localization. Lastly, we will conclude with some open research topics and real-world issue solutions.

2.1 GPS-denied Environments

Numerous articles in the tech literature address the problem of precision positioning in various environments. Rahman et al. [1] propose an indoor positioning system based on AM radio waves analyzed through the fingerprinting method, which combines three variations of the nearest neighbor algorithms. Sahin et al. [2] created a positioning system for indoor spaces by making use of active tags that communicate with the nearest reader through distinct levels of power. Fang et al. [3] developed an indoor solution that is based on LTE networks and performs real-time accurate localizations. Li et al. [4] explored the fingerprinting localization technique by developing a system that uses integrated channel state information and magnetic field strength information. Bencak et al. [5]

propose a Bluetooth Low Energy indoor positioning solution with the purpose of aiding the management of warehouses. Shi et al. [6] developed a system for Automatic Guided Vehicles to move inside a warehouse with an accuracy of centimeters. These approaches present different advantages and disadvantages, such as infrastructure requirements, error rates, and update rates.

Various methods have been proposed to address GPS-denied settings for indoor and outdoor navigation. Matos Carvalho et al. [7] use terrestrial radio data and UAV inertial sensors combined with Kalman filters for smooth navigation. You et al. [8] suggest using Ultra-Wide Band and Inertial Measurement Unit data to navigate quadrotor UAVs inside buildings. Huang and Wu [9] developed a distributed WiFi RSS-based Direction of Arrival module for search and rescue in GPS-denied conditions. Famili et al. [10] created a robust acoustic indoor localization approach using ultrasonic acoustic signals with hybrid FH-CDMA. Stockel et al. [11] present new techniques for accurately calculating a UAV's indoor mobility using phase measurements from a spinning radar and Inertial Measurement Unit data. Oelsch et al. [12] use Simultaneous Localization and Mapping and 3D LiDAR data to improve GPS-denied situations. Norton et al. [13] examine navigation methods for US military small unmanned aerial systems in subterranean and interior locations without GPS signal. Ismail et al. [14] offer a low-cost LTE-based outdoor GPS localization alternative, while Haddadi et al. [15] use two modules to estimate quadrotor UAV posture. Finally, Miraglia et al. [16] study data fusion methods for a 3D positioning system based on UWB signals and a 3 axis, 9-DOF Inertial Measurement Unit. These methods offer a range of approaches for addressing GPS-denied settings, each with its own advantages and limitations.

Drones have multiple applications, ranging from search and rescue operations to creative exhibitions. Farooq et al. [17] developed a perception-aware UAV platform using computer vision for target recognition and collision avoidance, while Kolawole et al. [18] designed a simulation platform for emergency drone mapping and indoor drone positioning. Shu et al. [19] proposed a method to localize Micro Aerial Vehicles using only IMU and four ultrasonic sensors, making it useful for amusement, monitoring, and rescue purposes. These studies highlight the need for drone positioning solutions besides GPS-based techniques to improve real-world scenarios.

2.2 Technological Approaches to Positioning

WiFi-based localization methods have been developed to provide indoor positioning due to the ubiquity of WiFi in modern buildings [20]. These methods use received signal strength indicator (RSSI) [21] to estimate the target node's position, weight range localizer (WRL), relative span exponential weight range localizer (RS-WRL) [22], and Angle of Arrival (AOA) [23] to improve range-based indoor localization, and wireless indoor positioning and navigation approaches for Autonomous Ground Vehicles (AGVs). [24] with measurement uncertainties. These methods have been shown to improve positioning accuracy and determine safe moving pathways with high reliability. They have also been utilized in indoor micro UAV localization and unmanned aerial vehicles (UAVs) for search and rescue (SAR) [25] in GPS-denied interior situations. Although these methods have their limitations, they provide a low-cost and existing infrastructure for indoor positioning.

Ultra-wideband (UWB) technology has several advantages over other localization systems for high-precision indoor and outdoor localization [26] despite its high cost and computational complexity. The technology has been used in various applications [27], such as emergency response, vehicle interior navigation, and intelligent warehouse management systems [28]. The literature suggests different UWB implementations, including neural networks [29], Kalman filters [30], machine learning algorithms [31], ad-hoc networking [32], and probabilistic model-based approaches [33]. These approaches have improved UWB localization accuracy [34], reduced non-line-of-sight inaccuracy, and minimized robot navigation drift error [35]. While some of the proposed systems have been tested under simulated data [36], others have undergone successful real-world implementation.

Recent research has also focused on employing image processing techniques to develop accurate and efficient visual-based indoor localization systems for robotic applications. Li et al. [37] proposed a robust stereo visual SLAM system for AGVs, while Dong et al. [38] introduced ViNav, a smartphone-based indoor navigation system. Zhang et al. [39] suggested using cellphones for continuous indoor localization, and Zhao et al. [40] introduced Vivid, a visual indoor navigation system. Zhang [41] proposed using ORB features for interior spaces such as malls. Guan et al. [42] created a ROS-based indoor robot visible light positioning (VLP) localization system, while Al-Hameed et al. [43] proposed LiDAL, a visible light communication-based indoor light-based detection and localization system. Additionally, Bavle et al. [44] described the VPS-SLAM method, which enables airborne robotic systems to perceive and use semantic information from their surroundings. Furthermore, Putra and Saputra [45] proposed using a virtual 3D map to determine the position of an indoor drone based on IMU data, Naufal et al.[46] developed a vision-based autonomous landing system for quadcopter drones using OpenMV, while Liang and Liu[47] proposed a robust VLC-inertial localization method using visible light communication and inertial measurements, and Ali et al. [48] proposed an indoor vision-based localization and orientation determination method for a quadrotor, by determining accurate absolute position, orientation, and altitude through image processing and Markov localization methods. All these researchers leverage different image processing procedures to enhance localization accuracy and reduce errors, providing efficient real-time placement solutions for robotics applications.

Another related approach is present in various studies on visual servoing for UAVs, including using machine vision for drone placement, tracking linear structured infrastructures [49], passivity-based visual servoing for crop line tracking [50], vertical and horizontal target tracking through image moments [51], and image-based visual servoing of unmanned aerial vehicles for variable angle target tracking [52]. The studies also explore techniques for landing a quadrotor UAV [53], robot-centric model-based visual servoing [54], and ORB feature detection and descriptor computation for SLAM-based navigation systems for indoor situations [41]. Additionally, a new multi-copter airborne recovery control technique is developed [55]. Simulation and field test results prove the effectiveness of the suggested techniques.

Some studies discuss the use of dead reckoning for location and trajectory reconstruction in areas with poor or lost GPS signals. Aparna et al. [56] use a Nonlinear Autoregressive Exogenous (NARX) algorithm to estimate the present position in GPS-denied areas using GPS coordinates and IMU data. Jeong and Ko [57] suggest using Lie group theory to dead reckon a mobile robot, improving its localization and attitude estimates. Zhang et al. [58] propose a Dead-Reckoning-Based Local Positioning System (LPS) for intelligent cars that estimates posture using speed and direction sensor outputs and past state information. Xue and Jiang [59] offer a UWB-based navigation system that uses dead reckoning and Time-of-Flight (TOF) distance measurements to increase indoor location accuracy and stability. Zhou et al. [60] describe dead reckoning and Kalman filter architectures for UAV trajectory tracking in complicated urban situations, while another research group, Zhou et al. [61], suggest using the Invariant Extended Kalman Filter (IEKF) technique to dynamically change process and observation noise covariance matrixes using Attention mechanism and Recurrent Neural Network (RNN) to enhance UAV localization.

One final approach present in the reviewed literature includes fusion-based techniques that combine different technologies to compensate for shortcomings or enhance each other to achieve precise indoor localization [62]. The methods include the use of accelerometer, gyroscope, and magnetometer data [63], Bluetooth Low Energy, ultrasound time difference of arrival [64], UWB positioning technology [65, 66], RFID data [67], GPS, and image processing such as rotational vision [68], optical flowE [69], and SLAM [35, 70]. These techniques are used to increase the accuracy of indoor localization in areas with poor coverage, reduce robot navigation drift, achieve centimeter-level self-localization, improve GPS localization accuracy, and provide precise localization estimates, for example, in underground coal mining robots [71]. However, the efficacy of a fingerprinting and image-processing technique was sometimes disappointing [72].

The research aims to develop cost-effective and robust indoor positioning services to enhance indoor localization performance.

Finally, the present scientific literature study examines different localization applications using advanced technologies such as ultrawideband sensors [73], machine learning-based computer vision algorithms [74], multi-sensor fusion self-localization systems, indoor positioning systems [75], high-accuracy calibration approaches [76], and sensor-assisted range systems for drones. These technologies offer solutions for GPS-denied or GPS-challenged scenarios [77] in various indoor and underwater environments [78], underground environments [79], or aerial missions [80] making autonomous and semi-autonomous missions safer and more reliable [81]. The studies show that these technologies outperform traditional positioning methods like GPS and IMUs in indoor and subterranean locations, making them cheaper and more accessible. Additionally, these studies introduce mathematical models, calibration and optimization approaches [82], and innovative algorithms for indoor and various natural environments localization applications.

Chapter 3

Fast and Reliable Real-Time Tracking of Moving Targets

3.1 Introduction

There is an undeniable flurry of increasingly bold applications for drones such as remote inspections, package delivery, or even remote maintenance operations, stressing the need for developing drone autonomy solutions. Two possible applications may require the drone to:

- inspect railway tracks reducing the effort of manual inspections, by aerially collecting imaging data and dispatching quickly to events; a novel solution for flying along tracks is proposed due to the limitations of GPS navigation.
- assess forest inventory and health through the use of a small quadrotor drone in conjunction with a larger fixed-wing drone for tree inventory assessments; the focus is on a novel solution for the task of recovering the quadrotor drone using the two drones in the air.

Fiducial markers are 2D symbols that can be printed on flat surfaces and used for various applications, such as localization, tracking, robotics, camera calibration, position estimation [83], orientation, landing and automatic control for drones [84], and augmented reality [85], and existing marker systems are typically monochrome, but alternatives using circular [86] or chromatic [87] markers also exist.

3.2 Related Work

Drone Tracking

Onboard processing of drone-captured images has drawn the attention of researchers for its potential use in various applications, including visual servoing, tracking, and autonomous flight. Previous works have explored fixed-wing aircraft [88] and quadrotors [89, 90] for target tracking and landing, utilizing various computer vision systems and control algorithms [91–93]. Some examples include using a gimbaled video camera transmitted via a wireless link and processed on a stationary computer [94], a color-based target segmentation and tracking system [95], and neural networks running on an off-board computer [96]. A recent work [97] installed a gimbaled camera and a powerful embedded computer on a commercial quadrotor to perform visual tracking, providing detailed information about the computer vision and control algorithms used. All this work justifies the creation of a low-cost platform that is easy to build and operate, allowing for further exploration and experimentation in autonomous drone operations using onboard cameras.

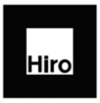
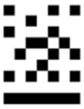
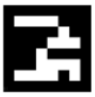
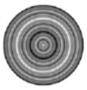
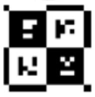
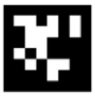
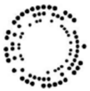
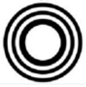
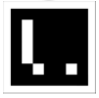
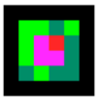
Overview of marker systems

Multiple fiducial marker systems have been proposed in the literature. Table 3.1 provides an overview of the analyzed marker systems by presenting their name, the year of their first publication, an example of the fiducial markers, and the area of application. The marker systems in the table are listed in ascending order by the year of their appearance.

Several fiducial marker systems have been proposed, having different particular properties that provide robustness. ARToolKit [98] uses a white square with a black border and can be customized to include a distinctive symbol as a marker but with decreased reliability [107]. ARTag [100] uses an array of black and white squares instead of a symbol, making it more robust [108]. AprilTag [103], while similar to the ARTag, brings several improvements in terms of detection speed. ArUco [106, 109] is a similar type of square-shaped marker with information coded in black and white that can be customized in terms of size. Other notable systems that use square-shaped fiducial markers are Cybercode [99] and CALTag [102].

A visually appealing solution, ChromaTag [87] uses a chromatic marker to reduce false positives. Circular fiducial markers [110], such as the Rune-Tag [104] and CCTag [105], offer resistance to occlusion and are used in location and position estimation and camera calibration. FourierTag [101] allows for a gradual degradation of the number of data bits

Table 3.1 Fiducial marker systems.

Marker	Year	Marker example	Used in
ARToolkit [98]	1999		Augmented reality
Cybercode [99]	2000		Augmented reality
ARTag [100]	2005		Augmented reality
FourierTag [101]	2007		Robotics, Virtual reality
CALTag [102]	2010		Camera calibration
AprilTag [103]	2011		Augmented reality, Camera calibration, Robotics
Rune-Tag [104]	2011		Localisation, position estimation
CCTag [105]	2012		Camera calibration
ArUco [106]	2014		Augmented reality, Camera calibration, Robotics
ChromaTag [87]	2017		Robotics

that can be extracted, improving detection at distance. Depending on the application requirements, these markers have different performances.

Comparison of marker systems performance

Marker systems have been compared in various studies based on metrics such as false positive rate, inter-marker confusion rate, minimal marker size, immunity to lighting conditions, and distance detection. ARTag [100, 108] was compared to other systems like Data Matrix, Maxicode, QR Code, ARSTudio, and ARToolkit, and found to have better performance in detecting markers at long distances, with less distortion, and allowing 3D pose estimation. Rune-Tag [104] and CCTag [111] are circular fiducial marker systems that offer resistance to occlusion, with Rune-Tag having higher accuracy for position estimation and CCTag having the highest detection rate of all in terms of distance, occlusion, and motion blur.

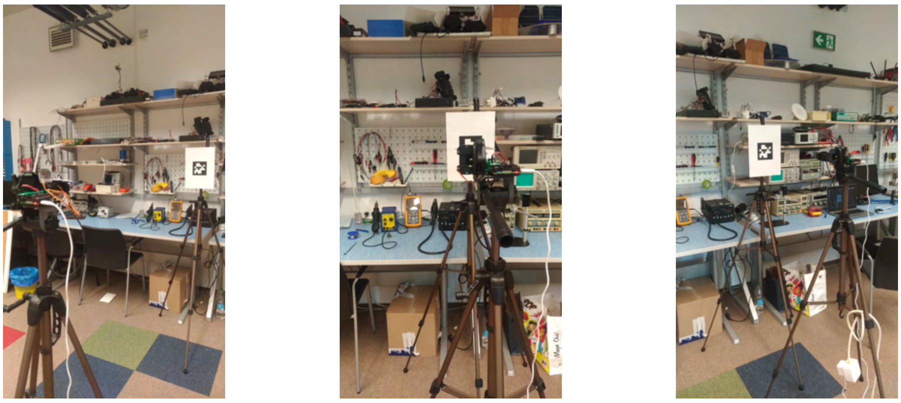
In addition, comparisons between marker systems such as ArUco, AprilTag, ARTag, and ChromaTag [110, 87] have shown differences in performance based on metrics such as light variation, floor pattern, image blur level, and detection distance. AprilTag [83] had the best detection rate for distance variations, while ARTag had the lowest. ArUco and AprilTag detected the marker in 90% of cases for orientation variations, while ARTag only detected it in 45%. ChromaTag was found to be faster in marker detection compared to other systems, but should be used when the task involves detecting a marker from a short distance and positioned perpendicular to the camera, while AprilTag should be used for detection from a greater distance or a sharp angle. A new approach for the ArUco marker called ArUco3 [112] was proposed, resulting in faster marker detection by detecting the marker in an image smaller than the original.

3.3 Fiducial Markers Performances

The performance of different marker parameters is an important subject when choosing what type of marker is appropriate in critical scenarios. Three open-source fiducial marker systems [110], ArUco, AprilTag, and CCTag, have not been compared despite previous studies [87] showing varying performance results between them [112]. We tested against each other, in different conditions, with markers having a value of 10 as ID.

3.3.1 Comparison Methods

The experimental setup displayed in Figure 3.1 consists of testing the performance of different markers while varying several parameters.



(a) Right view

(b) Center view

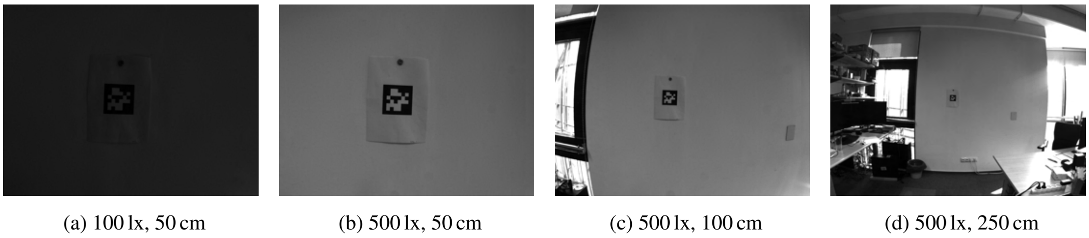
(c) Left view

Fig. 3.1 Setup

These parameters of interest include the type of marker, the distance to the camera, the brightness of the light, and the exposure time. Three 10cm x 10cm size markers were printed on separate A4 sheets of paper and placed on a tripod one by one. Another tripod was used for varying the distance to the camera. In the end, by varying the distance, the light, the exposure time, and the angular velocity of the camera, we managed to create a vast data set, collecting a total of 13200 pictures.

The main testing scenarios include a fixed camera scenario and a moving camera scenario:

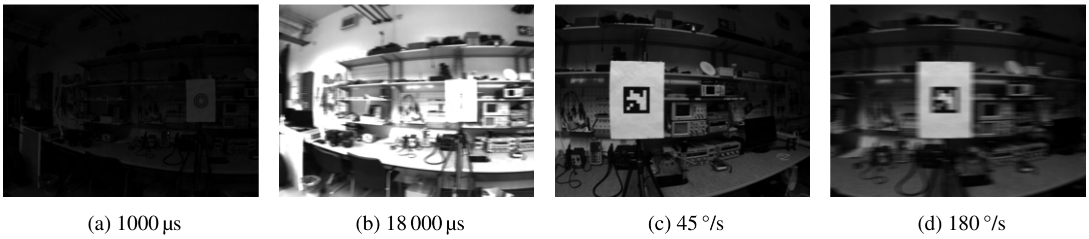
- the fixed scenario displayed in Figure 3.2. tested all three markers, at a constant exposure of 3000 µs, in two lighting conditions of 100 and 500 lx, and from a varying distance of 50, 100, 150, 200, and 250 cm.



(a) 100 lx, 50 cm (b) 500 lx, 50 cm (c) 500 lx, 100 cm (d) 500 lx, 250 cm

Fig. 3.2 Apriltag: distance variation and light intensity variation

- the moving scenario displayed in Figure 3.3. tested all three markers, at varying exposure times between 1000 µs and 18 000 µs, in varying lighting conditions between 100 and 500 lx, from a varying distance of 50, 100, and 150 cm, and varying another parameter, the angular velocity, from 45 °/s to 180 °/s.



(a) 1000 µs (b) 18 000 µs (c) 45 °/s (d) 180 °/s

Fig. 3.3 CCTag: exposure time variation and angular velocity variation

3.3.2 Results

The algorithms for each marker system were run on a Linux 18.04 system installed on an HP ProBook 440 G4 Notebook PC with the following specifications: Intel Core i7-7500U Intel HD Graphics 620, 2.7 GHz, up to 3.5 GHz with Intel Turbo Boost technology, 4 MB L3 cache.

3.3.2.1 Distance variation

The data set contains pictures where the distance between the marker and the camera is from 50 to 250 cm, and the size of the markers is 10 cm x 10 cm. For moving pictures, we observed that CCTag had the highest detection rate for distances of 50 and 100 cm, with a percentage of 74.64% and 56.14% of correct positives, respectively. In addition, we noticed that on the distance of 250 cm, we did not manage to obtain a 100% performance as with ArUco and AprilTag. We can therefore conclude that the maximum detection distance of the CCTag markers is 200 cm at most between the camera and the marker, at a marker size of 10 cm x 10 cm.

3.3.2.2 Intensity of light variation

In order to determine the performance of the three systems considering variations in light intensity, we analyzed two situations, the first with artificial light from the laboratory turned on, called light (325 lx), and the second without artificial light, called dark (175 lx). Considering the distance-based results, we decided that there was no need to run and display the results for the fixed images anymore.

We observed there was a small difference between dark and light conditions on all 3 systems, with a surprising finding that CCTag had a higher detection rate in dark conditions; and a lower detection rate in light conditions. This is due to the fact that we started from a small exposure of 1000 microseconds and went up to 18 000 µs. More precisely, long exposures of 15 000 – 18 000 µs bring a disadvantage to light pictures and an advantage to dark ones in the CCTag system.

3.3.2.3 Angular velocity variation

In this subsection, we focus only on the pictures taken while the camera was moving using the servo motor at five different angular velocities: 45, 60, 90, 135, 180 °/s; with 840 pictures per each angular velocity.

What we noticed from the results was that as the angular velocity increases, the detection rate of all markers decreases, which is expected because the higher the angular velocity, the more blurry the pictures become. The second thing we noticed is that throughout all the runs, the markers have kept their place in the hierarchy, thus AprilTag had the lowest detection rate of all three, followed by ArUco and CCTag. The first place with the best detection rate was CCTag. If at the angular velocity of 45 °/s the scores concerning the number of detected markers were relatively close (CCTag = 670, ArUco = 645, AprilTag = 616), there were higher disparities at the angular velocity of 180 °/s(CCTag = 323, ArUco = 246, AprilTag = 185). Regarding the latter scenario, it can be observed that CCTag detected almost twice as much as AprilTag.

3.3.2.4 Resistance to occlusion

To test the resistance to occlusion, we took a picture where the marker was alternately covered on its left, right, top, and bottom half with a blue square. We observed that AprilTag and ArUco were not immune to occlusion: when an object was placed in front of the marker, the algorithm stopped detecting the marker. On the contrary, CCTag was able to successfully detect 3 out of 4 pictures.

This made us conclude that CCTag is resistant to occlusion. Moreover, we also identified this as the reason why it had the highest detection rate in previous tests on moving pictures. When the pictures are moved, the entire marker is no longer visible, but only a part of it, as if it was hidden behind an object.

3.3.2.5 Performance of algorithms

To analyze the performance of the algorithms, we started a clock just before calling the detection function and stopped it immediately thereafter. We ran the algorithms of the three systems on the fixed data set separately from the moving set and averaged how long it took for the algorithms to detect whether or not there is a marker in a picture.

The marker system with an outstanding 2 ms performance was ArUco, which we then used as a benchmark for the other 2 systems. Dividing the running time of AprilTag by the running time of ArUco, we concluded that AprilTag was about 5 times slower than ArUco on still images and about 4 times slower on moving ones. CCTag had a rather low performance, respectively 58 and 55 times slower than ArUco.

3.4 Drone Traking using Fiducial Markers

In this section, we propose and test a novel, low-cost quadrotor drone platform capable of visual servoing using an onboard camera placed on an active gimbal and onboard computation.

3.4.1 Proposed Solution Design

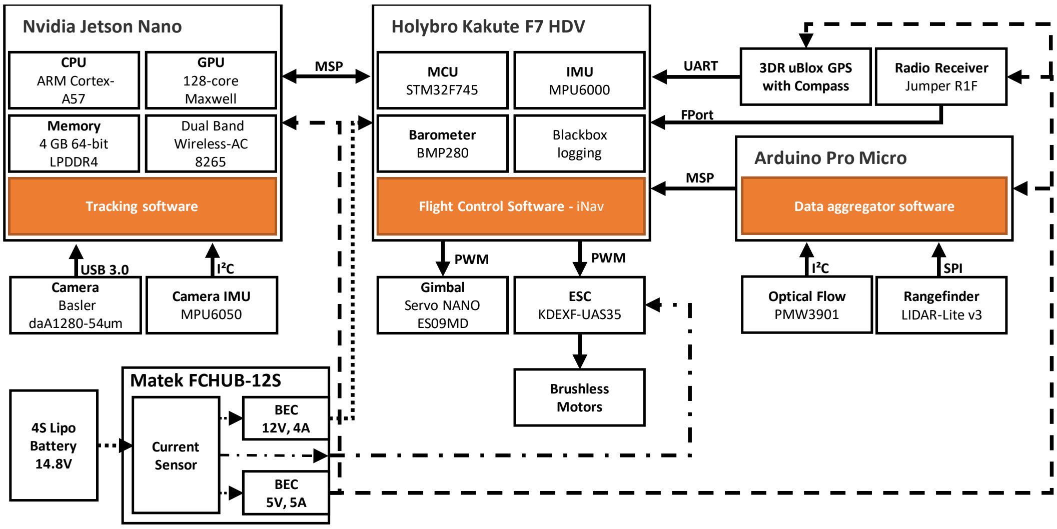


Fig. 3.4 The hardware architecture. In this figure, the clear rectangles represent hardware components or PCBs (printed circuit board) containing several components, the orange-filled rectangles represent the software modules, the dotted lines represent power connections and the solid lines represent digital communication lines.

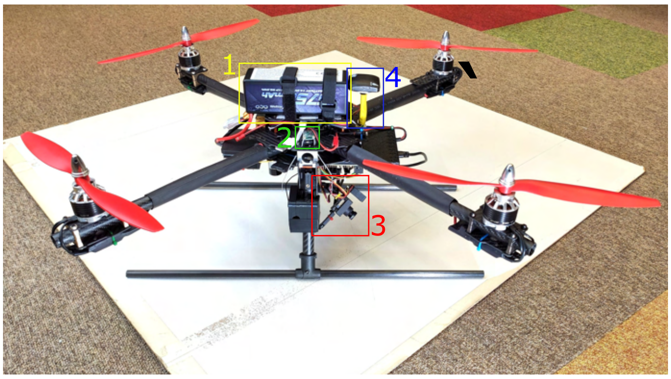


Fig. 3.5 The UAV side view and bottom view. Legend: (1) Battery, (2) RC Radio Receiver and Telemetry, (3) Gimbal and Camera, (4) GPS Receiver, Arduino, Optical Flow Sensor, Height Lidar, NVIDIA Jetson Board

The UAV designed for this work is a custom-made quadcopter, built on a carbon fiber frame spanning 70cm, which is shown in Fig. 3.5. It is equipped with 4 brushless motors, KDE Direct 2814XF, that drive 12-inch plastic propellers, a gimbaled camera, an on-board computer, a flight control system, and other sensors like GPS, IMU, rangefinder,

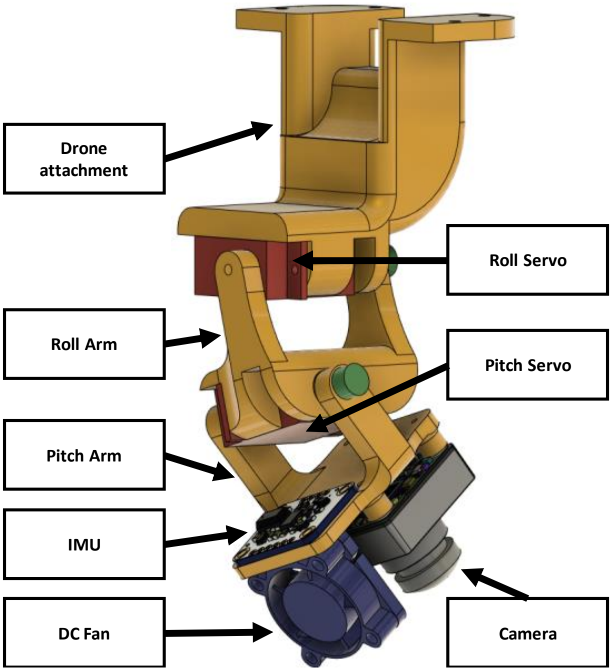


Fig. 3.6 The gimbal assembly. It is composed of two mini RC Servos, the camera, the IMU, and a cooling fan for the camera. All the components are linked through three 3D-printed parts.

and optical flow. To power all of these components we used a Lipo battery from Gens, which is a 4S battery with 6750 mAh autonomy and 70C discharging rate.

The Flight Control Software that we chose was INAV, an open-software solution to control multirotor drones which is a fork of Cleanflight software. Compared to the other programs, INAV is built with a strong focus on GPS and waypoint navigation. INAV is fully compatible with the flight controller that we chose and with the INAV Configurator. It also provides us with access to modify any setting without the need to reprogram the controller.

We display the hardware architecture in Fig. 3.4. We have built the drone around three main components: Nvidia Jeson Nano for visual detection and tracking, Holybro Kakute F7 HDV for flight control, and Arduino Pro Micro for aggregating data from sensors.

In Fig. 3.6 there is the gimbal assembly. The mechanical part is built from 3D-printed components designed by us. To drive de arms of the gimbal we used two servos. On the head of this gimbal, we mounted a monochrome camera, model Basler daA1280-54um. This camera was chosen for its global shutter, which helps us to reduce de number of blurred images.

In the end, we created a test platform for our algorithms with the total cost of 1400 euros.

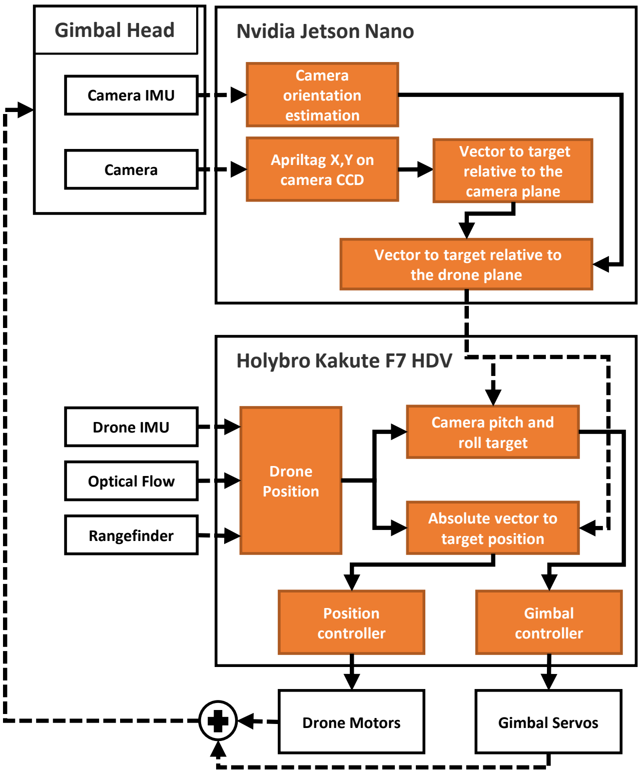

Gimbal Head
Camera IMU
Camera
Nvidia Jetson Nano
Camera orientation estimation
Apriltag X,Y on camera CCD
Vector to target relative to the camera plane
Vector to target relative to the drone plane
Holybro Kakute F7 HDV
Drone IMU
Optical Flow
Rangefinder
Drone Position
Camera pitch and roll target
Absolute vector to target position
Position controller
Gimbal controller
Drone Motors
Gimbal Servos

Fig. 3.7 Data flow

3.4.2 Visual Processing

As can be seen in Fig. 3.7 we have two large blocks in which we process the information coming from the camera and sensors. The first block is the Nvidia Jetson development board. First of all, we take images from the Basler camera at a resolution of 1280x960 pixels. For each frame, we apply the Apriltag 2 marker detection algorithm[113]. If the marker was found, we generate a position vector of the marker relative to the camera plane. At the same time, we read the values from the IMU of the camera, over which we apply a complementary filter to obtain the orientation of the camera in space. Thus with these two pieces of information, we know with high accuracy what was the position of the camera when the respective frame was made, and we can build the vector to the target relative to the drone plane.

The second block represented by the Kakute F7 HDV board that runs INAV 2.5 flight control software [114], generates the commands for the drone's motors to move towards the target, as well as the commands for the gimbal's servos so that the camera keeps the target in the foreground. At this point, we know both the target position relative to the drone and the estimated position of the drone, the latter being calculated by the INAV software flight control, which takes data from drone sensors such as the IMU, the optical flow sensor, and the rangefinder. With these two pieces of information, we calculate the commands for the position controller of the drone and the gimbal.

3.4.3 Experiments

To test our platform we prepared a task in which the drone has to hover above a target using only onboard visual feedback and computation. On the ground, we printed an AprilTag fiducial marker [113, 115] which the drone uses as a target. For image processing, we used the AprilTag library running on the Jetson Nano. The experimental setup is presented in Figure 3.8. For experiment initiation and for safety, the drone has a manual override using the remote control.

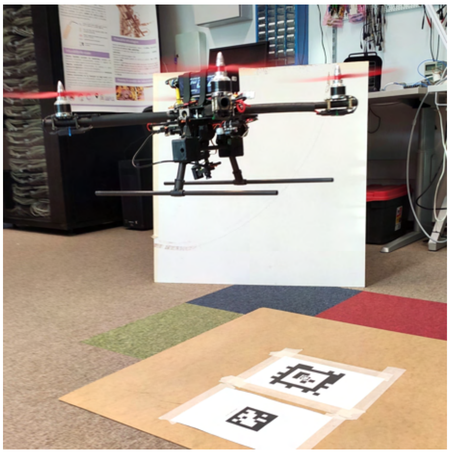


Fig. 3.8 The drone flying in our office. The image was taken after the targe was acquired and the camera was automatically oriented towards the target.

The drone takes off under manual control and is manually driven above the target. Once the visual system detects the fiducial marking the drone starts autonomous flying, controlling the drone height (1.4m), drone position, and camera gimbal orientation. We evaluated the reliability of the entire system over approximately 80 minutes of experiments in which the drone remained stably above the target. Small oscillations of less than 10cm are observed due to the inherent noise in the system. Various disturbances are applied by manually pushing the drone or sending movement commands through the radio remote. Additionally, we manually moved the target on the ground to test the drone's ability to maintain lock.

To test the step response of the system, we placed two markers approximately 80cm apart and we manually covered them alternatively, forcing the drone to move from one position to the other. The data collected in one such motion is presented in Figure 3.9. The gimbal quickly orients itself towards the new target and maintains target lock regardless of the drone's attitude (the drone needs to pitch towards the target in order to advance, thus orienting the camera away from the target).

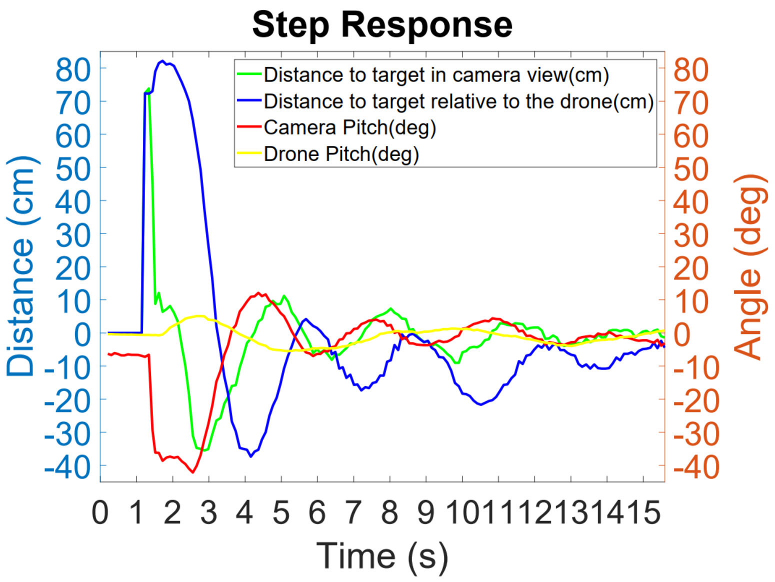


Fig. 3.9 Step response.

3.5 Conclusions and Further Work

We design, implement, and evaluate a novel, low-cost (1400 EUR) replicable drone platform capable of autonomous flight, with a gimbaled camera and onboard vision. We demonstrated its performance in flight in a target-tracking experiment. Compared to commercial offerings, the user has total control over the hardware and software.

After comparing the performance of the three fiducial marker systems using multiple criteria, we can conclude that CCTag can be used with confidence in real-time detections and in situations that involve occlusion. However, ArUco is a better alternative when only detection speed is considered. The latter obtained the best running time of only 2 ms. AprilTag also performed well on various metrics, but not as well as CCTag and ArUco. It should also be noted that neither AprilTag nor ArUco is occlusion resistant: if an object covers part of the fiducial marker, the algorithms no longer detect the marker.

CCTag is a fiducial marker system that has special properties, such as an outstanding detection rate on blurred pictures or resistance to occlusion. However, the performance of the algorithm in terms of run time is rather low. Therefore, further work will focus on a deeper understanding of the algorithm behind the CCTag fiducial marker system with the aim of improving its runtime performance.

Chapter 4

Ubiquitous Positioning

4.1 Introduction

Ultra-wideband (UWB) technology has experienced a revival during the past few years, mainly for its high-accuracy ranging and localization capabilities. It is estimated that more than 1 billion UWB devices will be shipped by 2025, and that over the next 5–10 years all smartphones will have UWB capabilities [116].

UWB-based localization systems usually consist of a mobile node that needs to be localized, called *tag*, and several fixed nodes with known locations, called *anchors*, which communicate with the tag and aid the localization process. Range-based localization is arguably the most popular localization technique since it provides the highest accuracy and, at least in systems using two-way ranging, clock synchronization is not required neither between the anchors and the tag nor between the anchors themselves [117, 118].

Time-difference of arrival (TDOA) is an alternative localization method that uses the difference between the arrival times of two packets (usually, exchanged by the tag and two anchors) [119]. By computing the TDOA for more anchor pairs, the tag's location can be found at the intersection of multiple hyperbolas [120].

In this chapter, we propose, implement and evaluate a novel, efficient TDMA scheduling scheme for TDOA localization called FlexTDOA. In FlexTDOA, there is no single reference anchor; instead, all the anchors in the system can be configured to take turns in transmitting the synchronization beacon. Similarly, the order of the anchors that respond to the beacon changes in a round-robin manner. Therefore, depending on the needs of the system, fewer anchors than the maximum available can respond to a beacon, which reduces clock drift errors caused by the delay between the first and the last response, while allowing all anchors to participate in the localization process. FlexTDOA therefore exploits the full channel diversity of the environment, is not subject to single-link failures, and can maintain small errors even in large networks.

We implement FlexTDOA in a localization system based on the Qorvo DW3000 UWB chipset [121]. We compare the proposed system against the classic TDOA approach and the standard range-based multilateration algorithm in a deployment of ten anchors and one tag in an office environment, in both line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. We also evaluate the impact of several parameters on the ranging and localization accuracy, such as: the number of responses for each synchronization beacon for different system update rates, the number of anchors in the system, and the impact of changing the initiator and/or the order of responses.

4.2 Related work

In the following, we will review the most important previous works on TDOA localization, with a focus on *DL* TDOA schemes that offer the best multi-user scalability.

4.2.1 Scalable UWB Localization

In [122], a DL TDOA localization system that implements a clock synchronization protocol with a reference anchor is proposed. The authors mention that the system does not scale to large anchor networks. In a setup of eight anchors, the system obtained a 2D localization root-mean-squared error (RMSE) of 14 cm and a 3D RMSE of 28 cm. In a comparable setup of seven anchors in LOS (but over a slightly smaller tracking area than the one in [122]), FlexTDOA obtained a 2D RMSE of 7 cm and a 3D RMSE of 13.26 cm, so twice as small as the ones in [122].

In [123], the authors propose a DL TDOA scheme in which the anchors respond only to the previously-transmitted message instead of responding to a single synchronization beacon, as in our case. Although named concurrent ranging, the works in [124, 125] essentially implement the classic DL TDOA scheme. In [126], a TDOA localization system implemented using UWB devices called `ATLAS` is introduced. A localization system named `VULoc` that follows the principles of DL TDOA has recently been proposed in [127]. Perhaps the most significant difference between `VULoc` and FlexTDOA is that we also propose a flexible, highly-configurable TDMA scheme for anchor transmissions, whereas in [127] it is mentioned that `VULoc` does not need a scheduling protocol because tags are passive.

Although it does not implement a DL TDOA scheme, the work in [128] proposes a scalable UL TDOA localization scheme called `TALLA`. The high-precision synchronization necessary for TDOA localization is maintained by a server.

Another important contribution of our work is that we evaluated the performance of a DL TDOA localization system in NLOS conditions *experimentally*, since most previous works either consider only LOS scenarios or base their observations on simulated data. In [127], the proposed TDOA system is also evaluated in NLOS conditions and an anchor selection method based on an empirically-chosen confidence parameter is proposed.

In [129], the authors propose a sensor placement strategy for cluttered environments that is validated through experimental data. A UL TDOA localization system that takes into account NLOS conditions has been proposed and evaluated experimentally [130]. In [131], the authors propose an algorithm to select anchor pairs in a UL TDOA by taking into account errors caused by NLOS propagation.

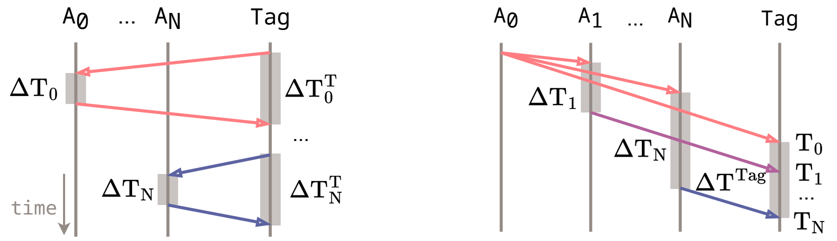
4.2.2 Clock Offset Correction

In our TDOA scheme, we avoid tracking the clock parameters using Kalman filters like in previous works [122, 132]. Instead, we correct the relative clock offset between two devices directly at the receiver using the CFO estimation feature of the DW3000 chipset. The method has been described in [133] and the systematic error has been derived for single-sided two-way ranging (SS-TWR), A-TDOA, and SS-TWR with A-TDOA extension. The method has been evaluated experimentally but only for TWR schemes. A similar CFO correction is evaluated for a TDOA scheme in [134]. However, the proposed TDOA scheme is based on the alternative double-sided TWR (AltDS-TWR) method, in which the tag is active, which is different from the DL TDOA schemes evaluated in our work.

4.3 Background

In this section, we introduce the basic principles of the localization methods proposed in our system. For more in-depth details on UWB ranging and localization, we refer the reader to the papers [135, 136]. In Section 4.3.1 and 4.3.2 we explain how distance measurements and, respectively, TDOA measurements are obtained using UWB devices. In Section 4.3.3, we describe the two approaches we use to solve the system of equations in order to estimate the user's location: either least-squares minimization between the measured and the calculated ranges or an extended Kalman filter (EKF). In Section 4.3.4

we describe the scheduling scheme implemented in the proposed FlexTDOA localization system.



(a) Range-based localization

(b) Downlink (DL) TDOA

Fig. 4.1 Localization based on TWR or on DL TDOA (the time periods are not to scale).

4.3.1 Two Way Ranging

Range-based localization uses distances between a mobile target, called tag, and anchors with known locations to compute the location of the tag at the intersection of circles (in a 2D space) or spheres (in a 3D space) with a radius equal to the anchor–tag distances and centered at the anchors.

To avoid synchronizing the anchors and the tag, the distances are usually obtained using TWR by exchanging at least two messages between the tag and each anchor [135]. We implemented the SS-TWR variant which uses two message exchanges between each anchor and the tag, illustrated in Fig. 4.1a. The measured distance between a tag and an anchor A_i is obtained as:

$$\tilde{d}_{TA_i} = c \cdot \frac{\Delta T_i^T - \Delta T_i}{2}, \tag{4.1}$$

where c is the speed of light. $\Delta \mathrm{T}_i$ is the time between the arrival of the tag's request at anchor $\mathtt{A}_\mathrm{i}$ and the anchor's transmission of the response message, as measured by the anchor. Similarly, ΔT_i^T is the time between the tag's transmission of the request and the arrival of the anchor's response, as measured by the tag.

SS-TWR based localization (which we will alternatively call TWR localization) is attractive because it enables centimeter-level localization and does not need any synchronization between the devices. However, it does not scale well when increasing the number of anchors and tags, since it needs pair-wise message exchanges between each anchor and each tag in the system.

4.3.2 Time Difference of Arrival

An alternative localization technique uses the time *difference* between the arrival of two messages either at one device or at multiple clock-synchronized devices[118].

To avoid synchronizing the anchors, we use a DL TDOA variant with reference and responding anchors previously used in [122, 132, 124, 125] The scheme is illustrated in Fig. 4.1b. Anchor $\mathtt{A}_0$ is the initiator and transmits a broadcast message, received by the tag at time T_0. Anchors $\mathtt{A}_1$ to $\mathtt{A}_\mathrm{N}$ receive the message and then wait a period ΔT_i which includes the processing time and a delay necessary to avoid overlapping transmissions from successive anchors. The tag receives the responses at times T_1 to T_N.

For the general case in which $\mathtt{A}_\mathrm{i}$ is the initiator and $\mathtt{A}_\mathrm{j}$ is the responder, let us denote by $\Delta T_{ij} \triangleq T_j - T_i$ the difference between the time at which the tag receives the response (T_j) and the time at which the tag receives the request (T_i). In order to obtain the *estimated* TDOA between the tag and the anchors $\mathtt{A}_\mathrm{i}$ and $\mathtt{A}_\mathrm{j}$ denoted by $\tilde{t}_{TA_iA_j}$, we need to subtract the processing time ΔT_j and the TOF between anchors $\mathtt{A}_\mathrm{i}$ and $\mathtt{A}_\mathrm{j}$ (denoted by t_{ij}) from the timestamp difference ΔT_{ij}:

$$\tilde{t}_{TA_iA_j} = T_j - T_i - \Delta T_j - t_{ij}, \tag{4.2}$$

The TOF t_{ij} is usually known because the anchors are placed at fixed, known locations.

4.3.3 Localization Algorithms

So far, we have discussed the basic principles to obtain the ranges or the range differences between the anchors and the tag. We aim to estimate the user's location, thus, we implemented two localization algorithms, each capable of operating with either TWR or TDOA data, each suiting different needs.

The first algorithm, AlgMin, solves the localization problem for a series of consecutive measurements using squared error minimization. This algorithm does not track the user's location nor does it smooth the location estimates, and it is therefore suitable to evaluate the impact of several parameters (e.g., the number of responses or anchors) on the localization accuracy.

The second algorithm, AlgEKF, solves the localization problem using an EKF, by incrementally updating the location with each additionally available measurement. This approach is advantageous because we do not need to wait for the minimum number of measurements (four in the case of TWR localization and five for TDOA localization) in order to update the tag's location. However, it smooths the location estimates and hides the impact of noisy measurements, which is why we use it only when we compare several setups that generate a different number of equations per time slot in Section 4.5.3.

4.3.4 Scheduling

In the “classic” TDOA approach, there is a single designated reference anchor which broadcasts the synchronization message, to which the rest of the anchors respond in a predefined order. Instead, we propose, implement, and evaluate a *flexible* TDOA scheduling scheme in which all the anchors in the system can play the role of the initiator and the order of responses can also change.

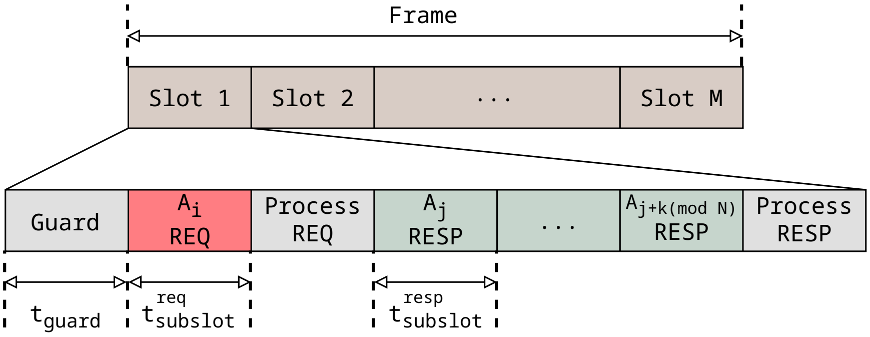


Fig. 4.2 TDMA schedule used for both TDOA and TWR localization. Each slot in a frame belongs to a node that is the initiator in that slot and decides which K nodes to interrogate (depending on the current scheme).

We propose and implement a time-division multiple access (TDMA) scheme shown in Fig. 4.2, which can be configured for either TWR or TDOA localization. At this point, we do not differentiate between anchors and tags and instead consider all of them equally participating nodes. The distinction will be made according to the implemented localization method.

The TDMA scheme is organized in *time slots*, which are comprised of a broadcast message sent by an initiating node, which we will call a *request*, and K responses from other nodes, where $\mathrm{K} < \mathrm{N}$ and N is the number of nodes in the system. Each response will provide a TWR measurement between the initiator node and the responding node and a TDOA measurement for each of the other nodes listening to the exchange. Inside a time slot, each transmission by a node occurs in a *subslot* with duration t_{subslot}. At the beginning of a time slot, there is a guard time, followed by the request of the initiating node. The request includes the number of nodes that will respond, their ID, and the order of their response. All the listening nodes in the system process the request. If the initiator requested a response from the listening node in the subslot with index $k \in \{1, ..., \mathrm{K}\}$, the node will wait a period of $(k-1) \times t_{\text{subslot}}$ and then answer. During the last part of a time slot, the initiator processes the responses.

The time slots are organized in frames (Fig. 4.2). Each frame contains M time slots, each of them assigned to one of the nodes (anchors or tags). To configure the TDMA scheme to perform TWR localization, the tag will be set as initiator in all slots and the anchors will be the responders. In one time slot with K responses, the tag obtains K raw distance measurements which are input to the multilateration system to estimate the tag's location.

To perform DL TDOA localization, only the anchors will be initiators, interrogating other anchors, while the tag will be a passive listener. Depending on how we choose the initiators and the responders, we can derive four variants of TDOA localization:

- *Fixed initiator, fixed responders* (FI-FR), or the "classic" TDOA, which has a single initial anchor and all other anchors respond in a fixed order according to their index.
- *Fixed initiator, changing responders* (FI-CR), which also has a single initial anchor, but other anchors respond in a cyclical order.
- *Changing initiator, fixed responders* (CI-FR), which has multiple initial anchors, and all other anchors respond in a fixed order.
- *Changing initiator, changing responders* (CI-CR), which has multiple initial anchors, but other anchors respond in a cyclical order. This method is opposed, developed, and validated by the author of the thesis.

We note that the TDMA scheme also allows the anchors to localize *themselves* using TWR.

4.4 Evaluation Methodology

In this section, we present the localization system used to evaluate the TWR and TDOA algorithms. In Section 4.4.1, we describe the hardware used; in Section 4.4.2, we present the settings used for the UWB radio, scheduling algorithm, and EKF. In Section 4.4.3, we describe the environment in which we acquired the measurements and the placement of the anchors.

4.4.1 Hardware System Design

For the experimental evaluation, we designed and fabricated our own UWB node, shown in Fig. 4.3, using the Qorvo DWM3000 wireless transceiver [121], which implements the IEEE 802.15.4 standard [137].

At the core of the UWB node is an Arm Cortex-M4 based STM32F429ZIT6 microcontroller [138]. The UWB node is powered by a single Li-ion rechargeable battery providing over 15h of autonomy.

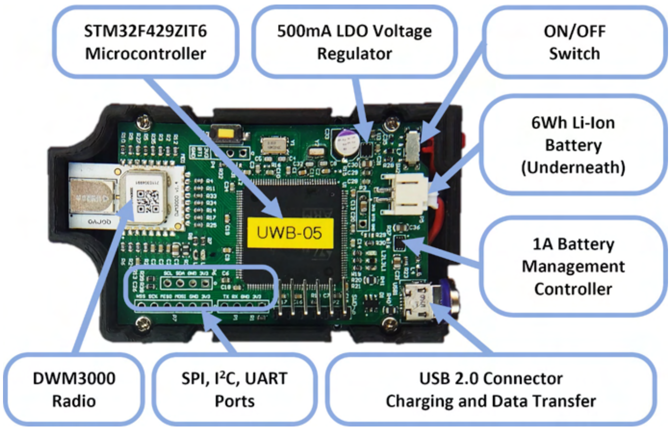


Fig. 4.3 Custom build UWB Node: a completed hardware and software UWB node which is battery powered and capable to expose ranging information over multiple serial communication interfaces.

4.4.2 System Settings

We configured the UWB transceiver to operate on channel 5 (6.5 GHz) with a preamble length of 128 symbols, a 6.8 Mb/s data rate, and a pulse repetition frequency of 64 MHz. The duration of one time slot in the TDMA scheme is shown in Fig. 4.2.

$$t_{\mathrm{TS}} = t_{\mathrm{guard}} + t_{\mathrm{subslot}}^{\mathrm{req}} + t_{\mathrm{process}}^{\mathrm{req}} + \mathrm{K} * t_{\mathrm{subslot}}^{\mathrm{resp}} + t_{\mathrm{process}}^{\mathrm{resp}} \tag{4.3}$$

Table 4.1 shows the duration of one time slot for each number of responders.

Table 4.1 Duration of one time slot (t_{TS}) for each number of responders K.

K	1	2	3	4	5	6	7	8	9
t_{TS}(ms)	3.35	4.20	5.05	5.90	6.75	7.60	8.45	9.30	10.15

For both EKF filters (based on TWR and TDOA measurements) we chose a standard deviation of the model uncertainty of $\sigma_Q^2 = 100\,\mathrm{cm}^2$, which accounts for the motion of the tag between measurements and assumes a maximum speed of the tag of 10 cm/s. We chose a standard deviation of $\sigma_R^2 = 10\,\mathrm{cm}^2$ for the measurement noise, which was based on the measurement noise we obtained during experiments.

For the EKF filter used for the self-localization of the anchors, we used $\sigma_Q^2 = 1\,\mathrm{cm}^2$ because the anchors are static. The location of the anchors is determined once, at the beginning of the experiments, and kept fixed thereafter.

4.4.3 Environment and Anchor Placement

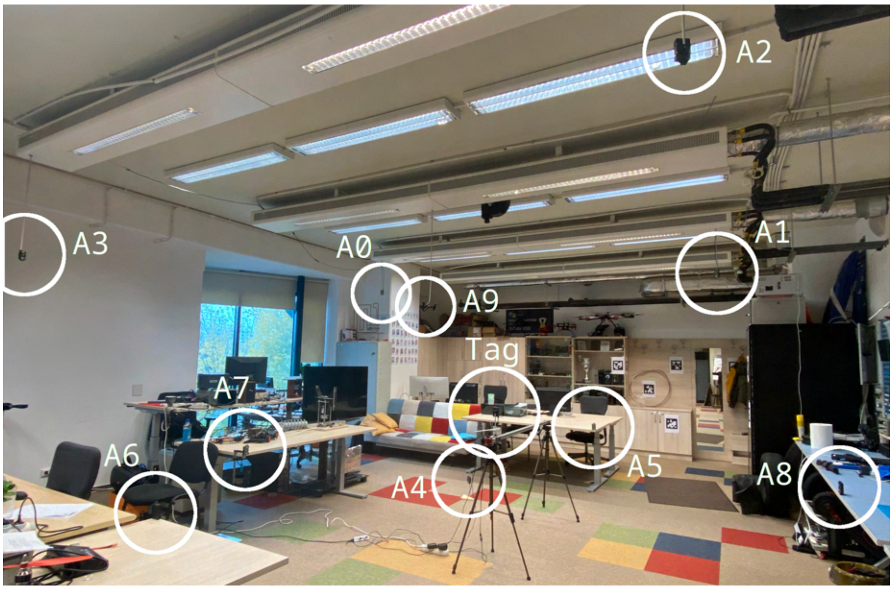
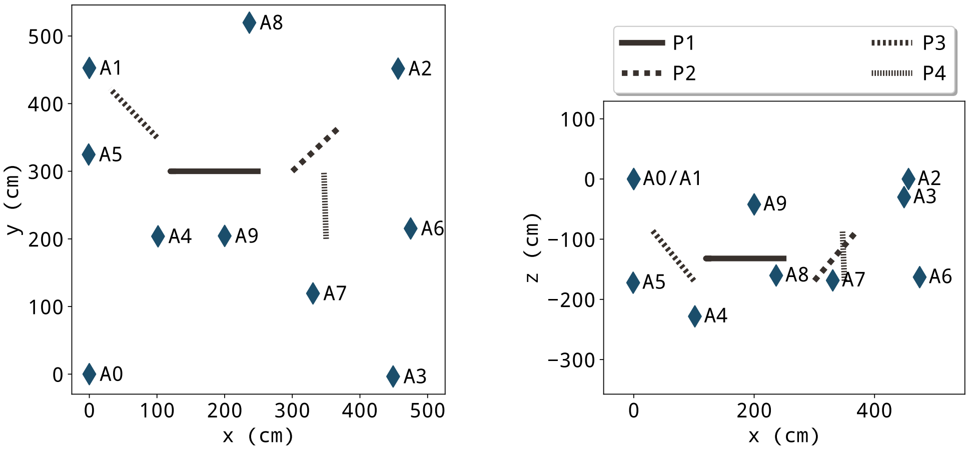


Fig. 4.4 Office setup.



(a)

(b)

Fig. 4.5 Setup of the anchors and the tag in the (a) *xy* and (b) *xz* planes. The anchors are denoted by $\mathtt{A}_0$ to $\mathtt{A}_9$. We evaluate the localization accuracy at four positions of the linear actuator, along which the tag moves, denoted by P1 to P4.

We evaluate the localization systems in the office shown in Fig. 4.4. The 3D anchor placement is shown more clearly in Fig. 4.5. Five of the anchors ($\mathtt{A}_0$ to $\mathtt{A}_3$ and $\mathtt{A}_9$) are fixed on the ceiling using metallic bars, while the rest of the anchors are either placed on the ground (A_4) or on tables (A_5 to A_8). The location of the anchors is determined using the self-localization algorithm described in Section 4.3.3 and validated using a laser level and a laser rangefinder, both with mm-level precision.

To accurately measure the ground truth (GT) of the tag, we have built a custom electronic linear actuator shown in Fig. 4.6.

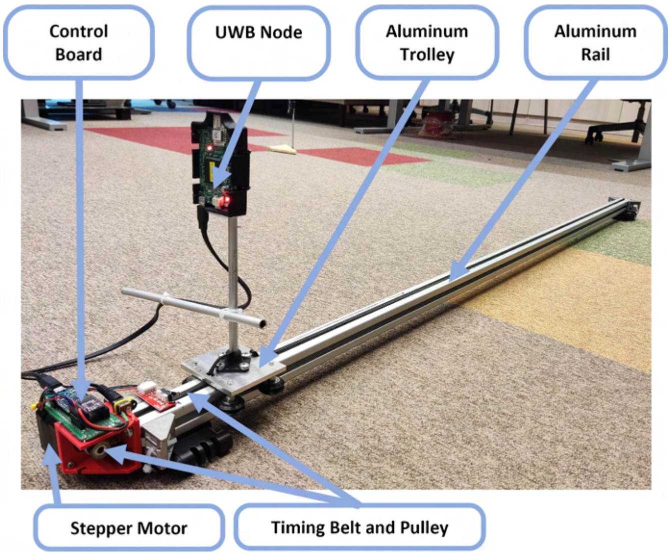


Fig. 4.6 Ground truth linear actuator: An aluminum trolley that carries the tag and returns over USB the position of the tag relative to the zero point of the actuator.

4.5 Evaluation of System Parameters

In this section, we evaluate the impact of several factors on localization accuracy: the order of response of an anchor, the number of responses in a time slot for the maximum update rate of the localization system and for lower update rates, and the number of anchors available. We evaluate these parameters for localization algorithms that use distance measurements (obtained using TWR), which we call "TWR localization," and for the proposed FlexTDOA system, called simply "TDOA localization." The goal of the comparison between TWR and TDOA localization is to evaluate the impact of system parameters of both distance and TDOA measurements.

Unless explicitly mentioned, we use the AlgMin algorithm described in Section 4.3.3 to estimate the user's location.

We will use boxplots to illustrate the error distributions. In a boxplot, such as the ones in Fig. 4.7, the box is drawn from the first to the third quartiles (or, respectively, the 25th and the 75th percentiles), which is also known as the interquartile range (IQR). Boxplots drawn for samples that can take negative and positive values (e.g., the distance and

TDOA errors) have whiskers that extend from the 5th to the 95th percentiles. For strictly positive errors (for instance, the localization errors which are computed as the Euclidean distance between the estimated and the ground truth locations), the whiskers extend from the 0th to the 95th percentiles. The reasoning is that, when we plot the distribution of *absolute* errors, we are interested in the minimum value of the error. We omit to plot the outliers for simplicity. We will frequently report the 95th percentile, which we will alternatively call the 95% error (or P_{95}) for short, which represents the value below which 95% of the errors are found.

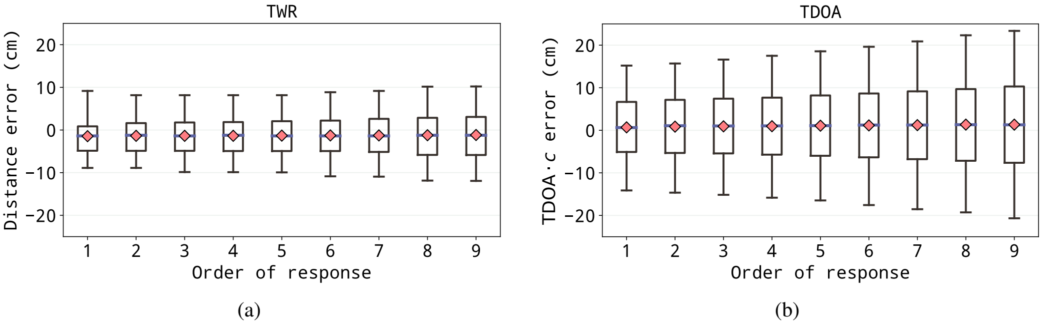


Fig. 4.7 The distribution of (a) TWR and (b) TDOA errors (expressed in cm using the speed of light) against the order of response aggregated over all anchors.

4.5.1 Order of Response

First, we investigate how the TWR or TDOA *measurement* error changes depending on the order of the response in a time slot.

To evaluate the magnitude of the errors, we perform an experiment in which the tag is kept unmoved to avoid any accuracy loss due to the movement of the tag. We configure the system to compute either the distance (using TWR) between each anchor and the tag or the TDOA between the tag and each pair of anchors. We use the maximum number of anchors ($N = 10$) and of responses ($K = 9$ for TDOA and $K = 10$ for TWR).

Fig. 4.7 shows the TWR and TDOA error distributions for each order of response.

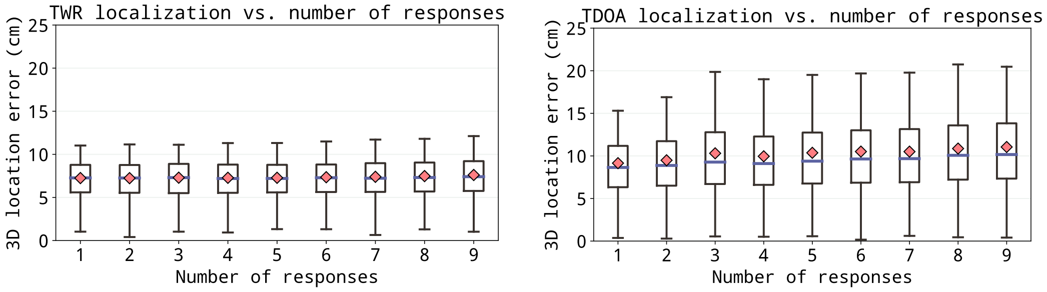
4.5.2 Number of Responses

Since the TWR and TDOA *measurement* error increases with the order of response in the time slot, we investigate to what extent the *localization* accuracy changes with the number of responses in a time slot.

For this evaluation, we use $N = 10$ anchors, but we vary the number of responses $K \in \{1,...,9\}$ and let the tag move on the trolley.

The results seem counter-intuitive because the system generates about 300 measurements per second which is enough to approach the maximum theoretical performance given the relatively slow speed of the tag of 10 cm/s.

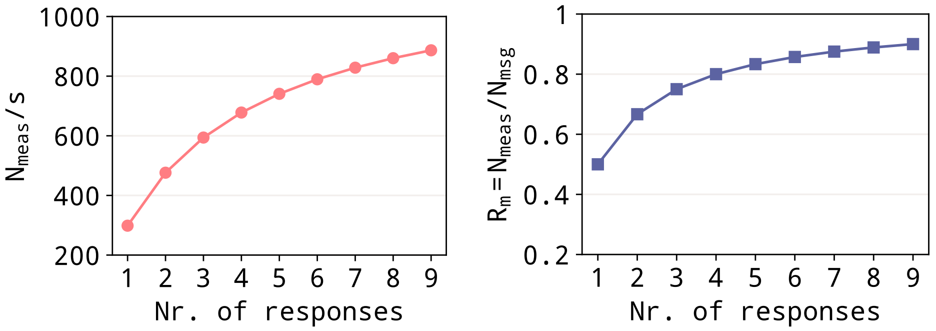
Fig. 4.8 shows the localization error of TWR and TDOA localization with a varying number of responses.



(a)

(b)

Fig. 4.8 Localization error of TWR and TDOA localization depending on the number of responses in a slot. The error increases for more responses due to the longer period between the first and the last response, which increases the effect of clock drift estimation error.



(a)

(b)

Fig. 4.9 (a) The number of TWR/TDOA measurements (N_{meas}) obtained per second vs. the number of responses (K) in a slot and (b) the ratio between N_{meas} and the number of messages (N_{msg}) obtained per second vs. K.

4.5.3 Number of Responses for Different Tag Speeds

Although using only one response per time slot yields the smallest error spread, this configuration has at least two disadvantages. First, over a fixed time period, the number of TDOA measurements decreases with the number of responses per time slot, because of the overhead added by the initiator's request. This trend is illustrated in Fig. 4.9a, which shows the number of distance or TDOA measurements (N_{meas}) obtained per second for $K \in 1, \ldots, 9$ responses.

Second, because of the same reason, the energy consumed by the tag to receive a certain number of TDOAs increases as the number of responses decreases. We can compute the ratio between the number of measurements and the number of exchanged messages (N_{msg}) over the same time period, which is an indicator of the efficiency of the tag. This ratio (denoted by R_m) is illustrated in Fig. 4.9b.

we evaluate how the 3D localization error changes with K when we keep constant over the same time period either the *total transmission time* (T_{TX}) or the *number of exchanged messages* (N_{msg}).

Additionally, to measure the effect on errors of a tag that is moving faster than our ground truth trolley, we deliberately slow down our system by introducing some idle time. To achieve all this, we group one or more time slots plus some of the necessary idle time in a frame that has a fixed length which we called the repetition period (Fig. 4.10), called $T_{rep} \in \{0.02, 0.5, 1\}$ s.

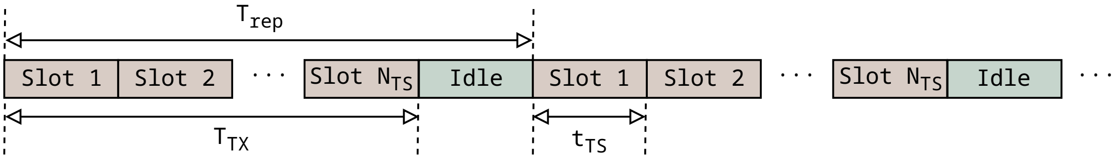


Fig. 4.10 To evaluate the optimal K under various simulated situations (time constraints, energy constraints, and tag speed), we create localization frames of fixed duration T_{rep}, containing N_{TS} time slots and the required idle time. Each time slot contains one request and K responses. We do two experiments: (1) we vary N_{TS} and K while keeping the total air time ($T_{TX} = N_{TS} * t_{TS}$) constant, thus simulating time constraints, and (2) we vary N_{TS} and K but we keep the total number of messages exchanged ($N_{msg} = N_{TS} * (K+1)$) constant, thus simulating energy constraints. In both cases, we vary the idle time to simulate a tag moving at various speeds.

In both experiments, we used $N = 10$ anchors and varied the number of responses $K \in \{1, 4, 9\}$. We, therefore, aimed to find the *repetition time* (or tag speed) for which more TDOA measurements compensate for the clock drift error incurred by a higher number of responses either when we have a fixed time budget (T_{TX}) or a fixed energy budget (N_{msg}). We consider that the number of received messages is proportional to the energy consumed by the tag. For the evaluation, we perform the localization using the AlgEKF algorithm.

4.5.3.1 Same Transmission Time

We first evaluate how the 3D localization error changes when the total transmission time (T_{TX}) is constant and the number of responses varies. In all three experiments, during each repetition period, we have approximately 10 ms of transmission time, the rest being idle time, as seen in Table 4.2.

Fig. 4.11 shows the error distributions for all combinations of number of responses and number of TDOA measurements per repetition period ((K, N_{meas})) and for all repetition

Table 4.2 Setup for experiments with approximately the same transmission time (T_{TX}).

K	N_{TS}	N_{meas}	T_{TX} (ms)
9	1	9	10.15
4	2	8	11.80
1	3	3	10.05

periods (T_{rep}). Therefore, at a high tag speed or a low update rate, it is preferable to use a high number of responses.

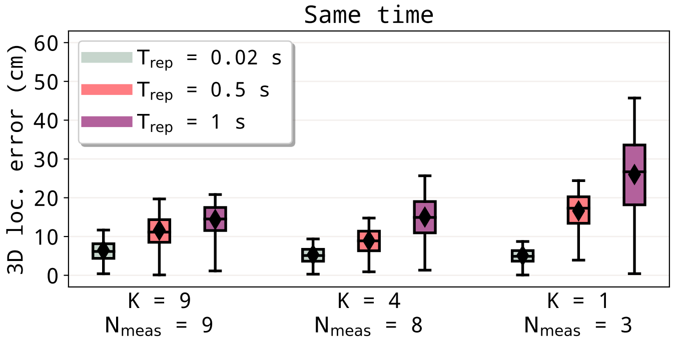


Fig. 4.11 Error distributions for the same TX time (T_{TX}). By increasing T_{rep}, we simulate a higher speed of the tag.

4.5.3.2 Same Number of Messages

We consider that the energy consumed by the tag is proportional to the number of received messages denoted by N_{msg}.

As seen in Table4.3, we keep the number of messages (N_{msg}) fixed during each repetition period and we vary the repetition time as in the previous experiment. Fig. 4.12 shows the error distributions for a fixed number of exchanged messages. Similar to the previous case, for a high update rate, $K = 1$ is the optimal number of responses. However, as T_{rep} increases, it is more beneficial to have more TDOAs than to minimize the clock drift error.

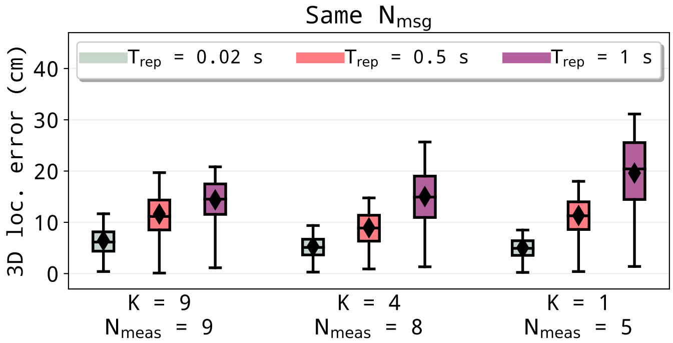


Fig. 4.12 Error distributions for the same number of transmitted messages (N_{msg}).

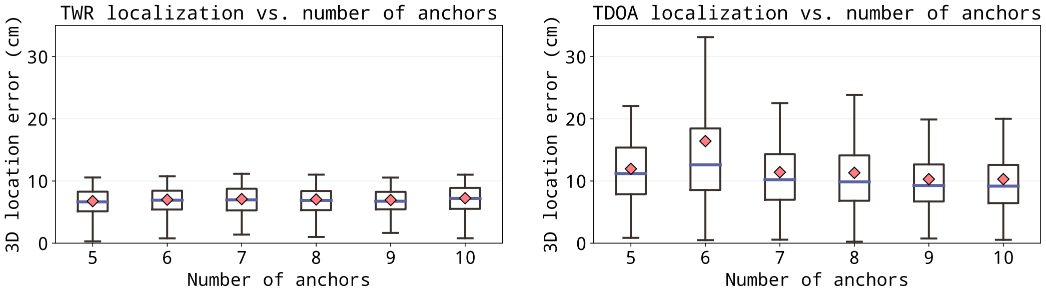
Table 4.3 Setup for experiments with the same number of messages (N_{msg}).

K	N_{TS}	N_{meas}	N_{msg}
9	1	9	10
4	2	8	10
1	5	5	10

4.5.4 Number of anchors

We evaluate the 3D localization error while varying the number of anchors participating in the localization between 5 and 10.

Fig. 4.13 shows the 3D localization error for TWR and TDOA localization when varying the number of anchors. The general trend is that the localization error decreases for more anchors.



(a)

(b)

Fig. 4.13 The 3D localization error of range-based localization depending on the number of devices.

4.6 Comparison of Localization Methods

In this section, we compare the localization accuracy of the considered localization methods. In Section 4.6.1, we first compare the four variants of TDOA localization presented in Section 4.3.4: FI-FR (or the classic TDOA), FI-CR, CI-FR, and CI-CR (or FlexTDOA). The goal is to evaluate the improvement brought by changing only the initiator, only the list of responders, or both. In Section 4.6.2, we compare only the classic TDOA, FlexTDOA, and TWR localization in a NLOS scenario. Throughout this section, we use the AlgMin algorithm to estimate the user's location.

4.6.1 Fixed vs. Changing Initiator and/or Response Order

In this part, we evaluate to what extent the channel diversity improves the localization accuracy in LOS conditions. We evaluate the localization errors for $(N, K) \in$

$\{(5,4),(7,6),(10,9)\}$[1] at three positions of the rail on which the tag moves, denoted by P1, P2, and P3 in Fig. 4.5. Position P1 is in the center of the room, parallel to the XY plane, where we should have the highest accuracy. Position P2 and P3 are inclined relative to the XY plane so that we can evaluate the errors at multiple tag heights.

Fig. 4.14 shows the distributions of the localization errors for the classic TDOA (FI-FR) and FlexTDOA (CI-CR) aggregated over all considered locations.

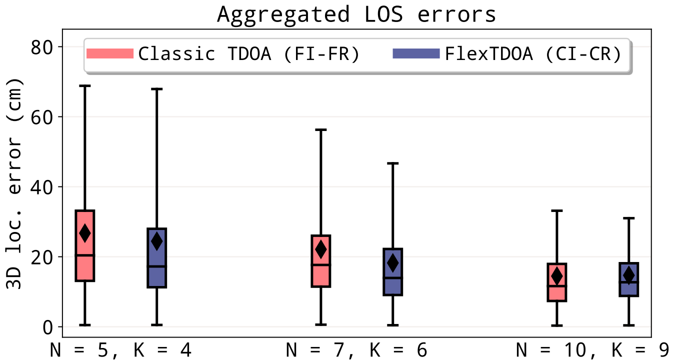


Fig. 4.14 Distributions of the localization errors of classic TDOA (FI-FR) and FlexTDOA (CI-CR), in LOS, aggregated over all the evaluated positions (P1, P2, P3).

4.6.2 NLOS Propagation

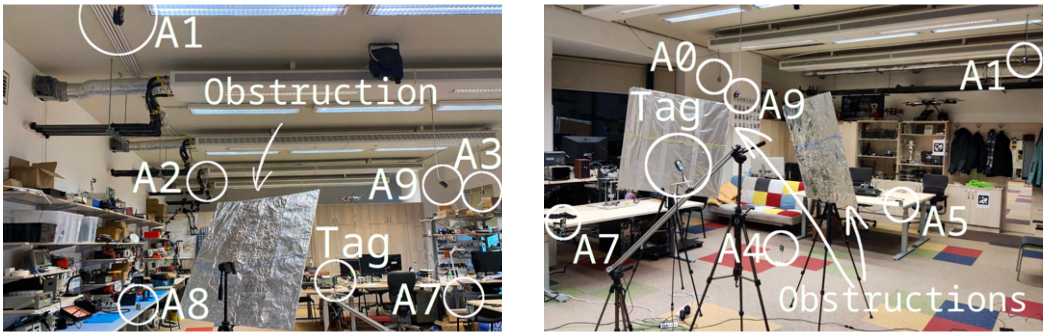
Obstacles between the nodes of a localization system are common in real-life scenarios. Therefore, in this part, we evaluate the performance of the three localization approaches considered so far: based on TWR, the classic TDOA, and FlexTDOA.

We performed measurements at two positions of the rail on which the tag moves. The positions are denoted by P1 and P4 in Fig. 4.5. At P1, we placed a panel covered in aluminum foil between the anchor $\mathtt{A}_1$ and the tag, shown in Fig. 4.15a. At P4, we placed two such panels, shown in Fig. 4.15b.

We perform the experiments for $\mathrm{N} \in \{5,7,10\}$ anchors. For TDOA localization, we use $\mathrm{K} = \mathrm{N} - 1$ responses, while for TWR we use $\mathrm{K} = \mathrm{N}$ responses. Fig. 4.16 shows the distribution of 3D localization errors aggregated over both locations. FlexTDOA achieves lower errors than the classic TDOA in all NLOS scenarios.

In TDOA localization, if the obstacle is between the initiating anchor and the tag, it incurs an error in *all* the TDOAs from that time slot. This is where FlexTDOA is more advantageous than the classic TDOA: by changing the initiating anchor, we ensure enough channel diversity to improve the robustness of the location estimate if the initiating anchor is obstructed.

[1] In each case, $\mathrm{K} = \mathrm{N} - 1$ so that, even for a fixed order of responses, all anchors get to participate in the localization process.



(a)

(b)

Fig. 4.15 Photos of the setups used to obtain NLOS measurements. The setup in Fig. 4.15a, which corresponds to the actuator position P1 from Fig. 4.5, includes one aluminum panel placed as an obstruction between anchor $\texttt{A}_1$ and the tag. The setup in Fig. 4.15a, which corresponds to the actuator position P4 from Fig. 4.5, includes two aluminum panels placed as obstructions. The rightmost aluminum panel blocks the LOS between the tag and $\texttt{A}_1$ and partially $\texttt{A}_5$ while the leftmost aluminum panel blocks $\texttt{A}_0$ and $\texttt{A}_9$. There is significant interference due to multipath propagation for $\texttt{A}_4$, $\texttt{A}_7$, and $\texttt{A}_8$.

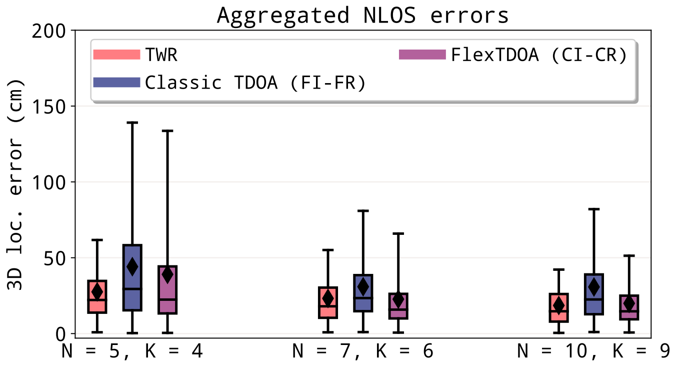


Fig. 4.16 Distributions of localization errors in NLOS using TWR, classic TDOA (FI-FR), and FlexTDOA (CI-CR), aggregated over both NLOS scenarios (at location P1, with one obstruction, and at location P4, with two obstructions).

4.7 Conclusions

In this paper, we propose, implement, and evaluate a novel, flexible TDMA scheduling scheme for TDOA localization that fully exploits the channel diversity in the environment. We compared FlexTDOA, the proposed method, against the classic TDOA implementation with a fixed reference anchor and responder list and against range-based localization in a deployment of up to ten anchors and one tag in an office environment.

FlexTDOA achieves lower localization errors than the classic TDOA in most scenarios, with and without obstructions. In LOS, the improvement in the median accuracy brought by FlexTDOA compared to the classic TDOA is modest (2–3 cm) because the initiator in the classic TDOA already has a good link to the tag. However, the robustness brought by the increased diversity is evident in NLOS, where FlexTDOA reduces 95^{th} percentile of the localization error with up to 38% compared to the classic TDOA. Overall, FlexTDOA achieves a median localization error of 13–17 cm in LOS and 15–22 cm when one or more anchors are in NLOS with the tag (the error depends on how many anchors are used).

While TWR localization yields the highest accuracy among all methods, it has poor scalability with a growing number of anchors and responders. In contrast, FlexTDOA can scale to an unlimited number of tags.

In the future, we will scale up the proposed system to a multi-room or building environment which will require several issues to be addressed: pairs of anchors that are not in communication range, system calibration (self-localization) for the sparsely connected network, and an efficient TDMA scheme that reuses slots for out-of-range nodes.

Chapter 5

Navigation without GPS

5.1 Introduction

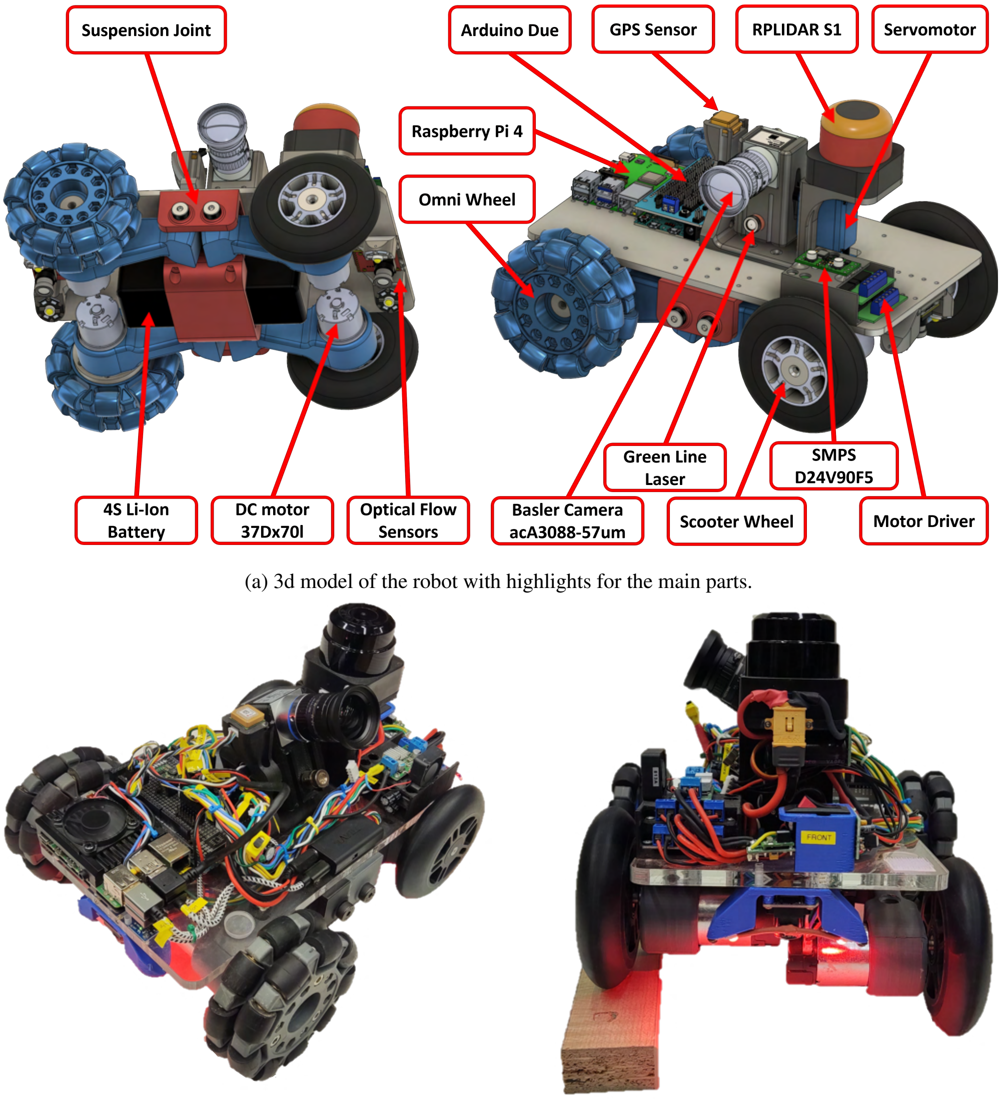
The transport industry faces a challenging business environment with low-profit margins, leading companies to seek ways to optimize every aspect of their operations [139]. One major expense for transport companies is the lack of regular tire checks and non-compliant tire usage, which can result in serious accidents, increased fuel consumption, and environmental pollution. Tire conditions are regulated in most countries and the EU [140], with legal requirements for tire grooves. However, the checks are subject to limitations, including difficulties in digitization, tracking tire conditions, and purchasing new tires, leading to poor fleet management.

Based on these requirements in collaboration with the TSG Romania [141] we work on a research project to improve the quality of transportation and to reduce the losses of the companies caused by the defective condition of the tires. The scope of the project was to build a custom solution to measure the degree of tire wear to help transportation companies optimize some aspects of the transport process.

The solution that we implemented was a terrestrial robot capable of autonomous navigation under cars and trucks in a park station and automatically detecting the degree of tire wear. To make this robot navigate autonomously we implemented two algorithms: one for navigation and the other one for positioning the robot under the car. These algorithms are based on LIDAR techniques and optical flow-based dead reckoning, respectively. To detect the degree of tire wear we use the triangulation method.

5.2 Hardware System Design

Our wheel wear measurement system consists of a small terrestrial robot shown in Figure 5.1 along with its main components. The robot is named TireBot, and is 28 cm long, 25 cm wide and 20 cm tall.



(a) 3d model of the robot with highlights for the main parts.

(b) Fully assembled robot with suspension joints capable of maintaining traction on bumpy roads.

Fig. 5.1 TireBot: an autonomous mobile robot capable of moving under cars and trucks and automatically detecting the degree of wear of their tires.

As seen in Fig. 5.2, we mounted the laser at 10° to the camera and 45 mm distance, which resulted in th distance between the camera sensor and the point of intersection of the normal at the center of the camera sensor and the laser beam being 255 mm. The system was mounted to scan the tire from the sides of TireBot. With this setup, TireBot can measure the tire parameter from the maximum distance of about 30 cm and from both sides of the robot, as the gimbal has a 180° steering angle.

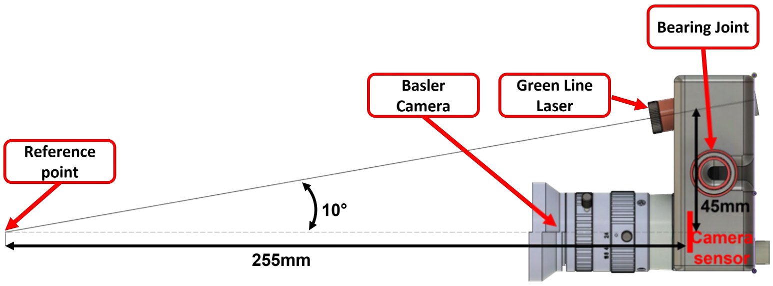


Fig. 5.2 Laser beam camera alignment

We are processing information and controlling TireBot using two computing boards: the Raspberry Pi 4 [142], which is the main processing board (called Main Boardin future references), and the Arduino Due based on the Atmel SAM3X8E ARM Cortex-M3 CPU [143], which is the low-level control board (called Control Boardin future references).

5.3 LIDAR-based Navigation

The navigation stack is comprised of three main modules: the navigation and planning state machine (which can be seen in Fig. 5.3), the Control Boardwhich handles the low-level control, and the link layer between the two.

5.3.1 The Navigation State Machine

The state machine developed for navigating the robot has the task to position it in the scanning area of each wheel of a car.

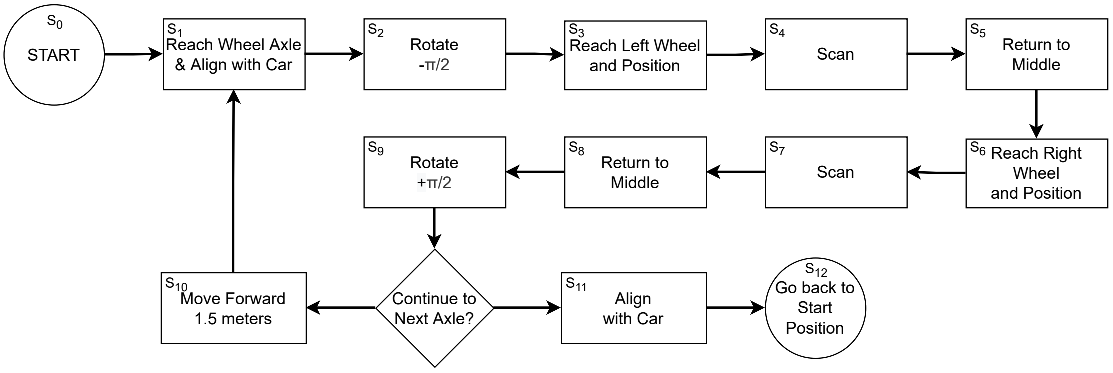

S0 START
S1 Reach Wheel Axle & Align with Car
S2 Rotate -π/2
S3 Reach Left Wheel and Position
S4 Scan
S5 Return to Middle
S6 Reach Right Wheel and Position
S7 Scan
S8 Return to Middle
S9 Rotate +π/2
S10 Move Forward 1.5 meters
Continue to Next Axle?
S11 Align with Car
S12 Go back to Start Position

Fig. 5.3 The Navigation State Machines

The input data for the state machine is the output of the LIDAR sensor, which consists of 720 distances (equally distributed in a circle, meaning that every measured point has an angle resolution of 0.5°) representing points where a surface/object is present.

During a state, we issue move commands to the link layer, which are defined as a relative position (x,y) in mm and rotation $\angle yaw$ in rad, where we wish the robot to move to. All the movements are relative to the robot's point of reference as shown in Fig. 5.4.

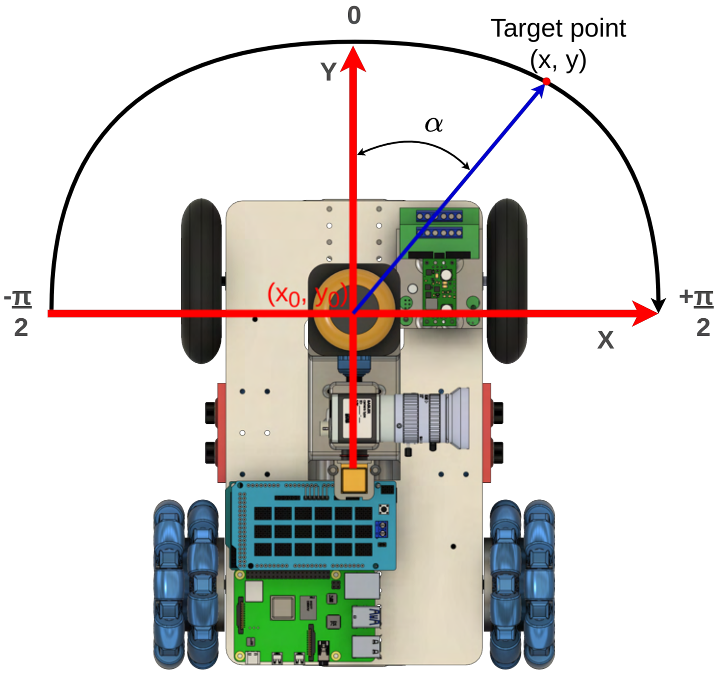


Fig. 5.4 Tirebot coordinate system.

Next, the flow through the state machine will ve explained, where the red dots are what the robot perceives as a wheel, the green dots are ignored because they are outside the quadrants where the robot searches for the wheels, and the blue dots represent the robot (four dots represent the wheels of the robot and the additional dot in the center is the rotation center of the robot).

- In the first stage of the algorithm, S_1, we identify the left and right wheels as being the points closest in the $(-\frac{\pi}{2},0)$ and $(0,\frac{\pi}{2})$ sectors respectively, as it can be seen in Fig. 5.5a and Fig. 5.5b. After reaching the target, we align with the car's axis using again the closest points in the $(-\pi,0)$ and $(0,\pi)$ sectors, rotating in order to position our heading as the bisector of the angle created by the 2 points and our origin.
- In state S_2 the robot rotates with $-\frac{\pi}{2}$ towards the wheel as in Fig. 5.5c.
- State S_3 is used to move the robot in the scanning area of the left tire (relative to the robot) as in Fig. 5.5d.
- Because of the errors in the input and in the robot's movement, after reaching the general scanning area, a finer step is performed.

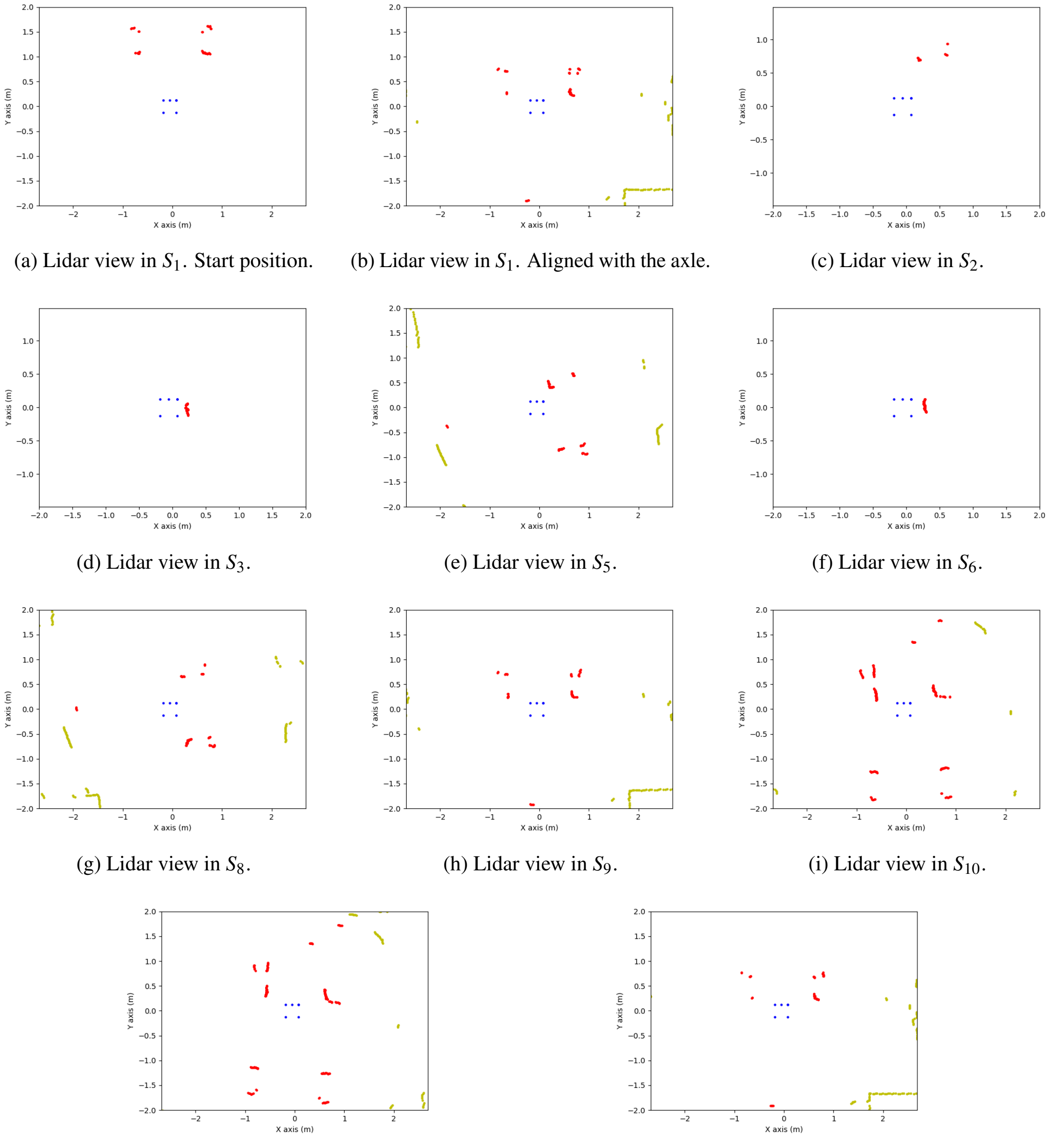
- State S_4 is a state in which we scan the depths of the grooves of the left tire. It will be described in section 5.5.
- In state S_5 we return to the middle of the wheel's axle as in Fig. 5.5e, a movement similar to S_1.
- State S_6 is used to move the robot in the scanning area of the other tire as in Fig. 5.5f. Similar to state S_3, finer movements are performed.
- State S_7 is a state in which we scan the depths of the grooves of the tire.
- In state S_8 we return to the middle of the wheel's axle as in Fig. 5.5g.
- In state S_9 the robot rotates with $\frac{\pi}{2}$ as in Fig. 5.5h.
- State S_{10} is the next state if there are more wheel axles remaining to be scanned. This is done to pass the previous axle and be able to move back to state S_1 and continue scanning the remaining wheels. The new position can be seen in Fig. 5.5i.
- State S_{11} is the next state if there are no other wheel axles to be scanned. In this state, the robot aligns with the car's center axis as in Fig. 5.5j.
- In state S_{12} the robot will try to return to the front of the car by moving backwards as in Fig. 5.5k.

5.3.2 Link Layer between State Machine and Control Board

This link layer has as an input a move command formed by the (x,y) point, which is the desired position relative to the current position of the robot, and $\angle yaw$ angle, which is the desired orientation compared to the actual orientation. The purpose of this part in the navigation stack is to transform the target robot state into simple $rotate$ and $forward/backward\ move$ commands.

5.3.3 Low-Level Control

To control the movement of the robot's servo motor, turn the laser on/off, or read the optical flow sensors into the Main Board, we use the Control Boardas an intermediary. The Main Boardissues commands through the serial interface such as: move a certain distance, rotate a certain degree, rotate the servo a certain degree, toggle the laser, and get displacement distance from the optical flow sensors.



(a) Lidar view in S_1. Start position.

(b) Lidar view in S_1. Aligned with the axle.

(c) Lidar view in S_2.

(d) Lidar view in S_3.

(e) Lidar view in S_5.

(f) Lidar view in S_6.

(g) Lidar view in S_8.

(h) Lidar view in S_9.

(i) Lidar view in S_{10}.

(j) Lidar view in S_{11}.

(k) Lidar view in S_{12}.

Fig. 5.5 The state machine for navigation

To execute the command received from “Link Layer between State Machine and Control Board”, on the Control Boardthere are implemented two PID controllers one for each wheel of the robot.

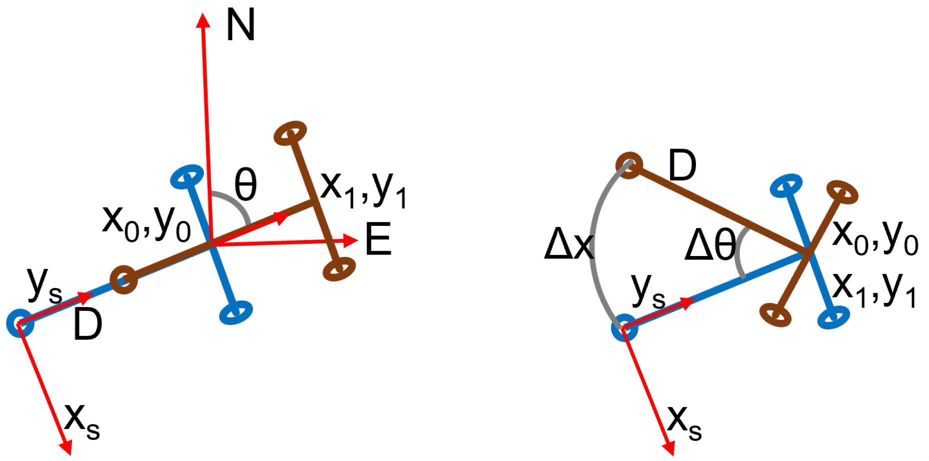
5.4 Localization through Dead Reckoning

Starting from the purpose of this robot to run under cars or trucks, we had to implement a localization solution for it so that it knows where it is located at any moment and how to return to the docking station. Because it will be under the car most of the time we cannot use a GPS-based solution for localization, because the signal will be blocked by the car and generally the GPS has an error over 1 m.

Our solution to localize the robot under the car is based on the dead reckoning method. We chose to use an optical flow sensor that can measure displacement through image processing. For our application, we read the data from these sensors, and through integration, we obtain the relative position to the starting point.

To implement the localization algorithm with this setup we analyzed three different cases of motion of the robot which can appear:

- straight forward, it can seen in Fig. 5.6a
- rotation in place, it can seen in Fig. 5.6b
- rotation on an arc of a circle, it can seen in Fig. 5.7



(a) Description of straight forward motion of the robot.

(b) Description of rotation in place of the robot.

Fig. 5.6 Straight forward and rotation in place movements

For all cases the position of the robot (x, y) can be computed by knowing the following values, where:

- (x_0, y_0) is the initial point
- D is the distance between the center of the rotation axis of the robot and the optical flow sensor.

- Δx_s and Δy_s are movement variations recorded by sensors
- θ is the angle yaw of the robot

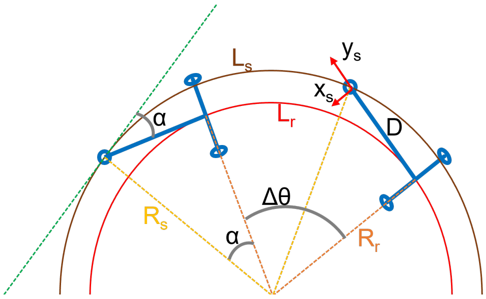


Fig. 5.7 Description of robot rotation on an arc of a circle.

5.4.1 Results

This solution helps us to map the movement of the robot under the car and know at any moment where it is. Fig. 5.8 illustrates the trajectories of the robot when it performs a scan. Also, this solution is integrated with the navigation part to compensate for wheel slip. When the robot moves, we compare the distance and rotation that the robot has to do, with the displacement recorded by this solution. If they are not the same, the robot adapts the command to the motor controller until the robot reaches the desired position.

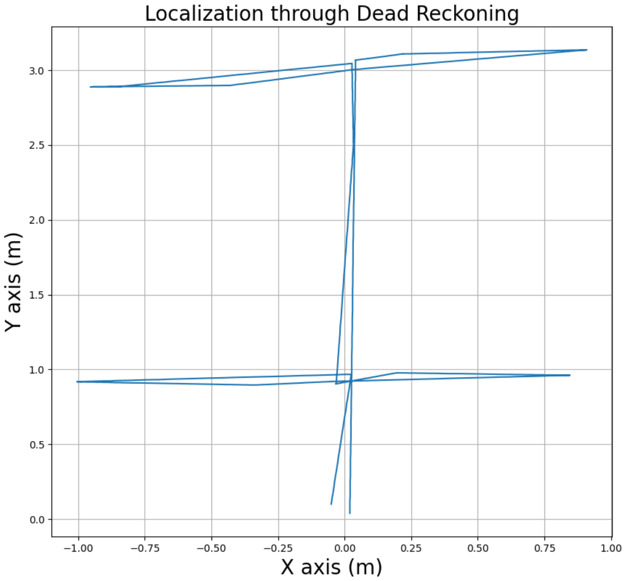


Fig. 5.8 Real-time trajectories of the robot when it performs a scan.

5.5 Positioning Through Computer Vision

In this chapter, we proposed a triangulation method that uses an optical solution to determine the distance between the camera and the tread of the tire. The optical solution is built from a video camera and a laser. A schematic depiction of the key steps of the suggested solution for computing the distance is shown in Fig. 5.9.

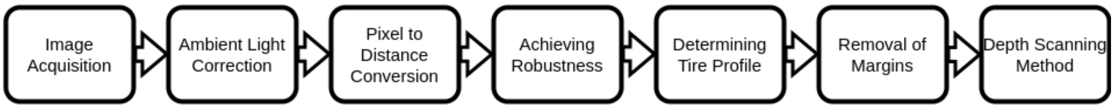
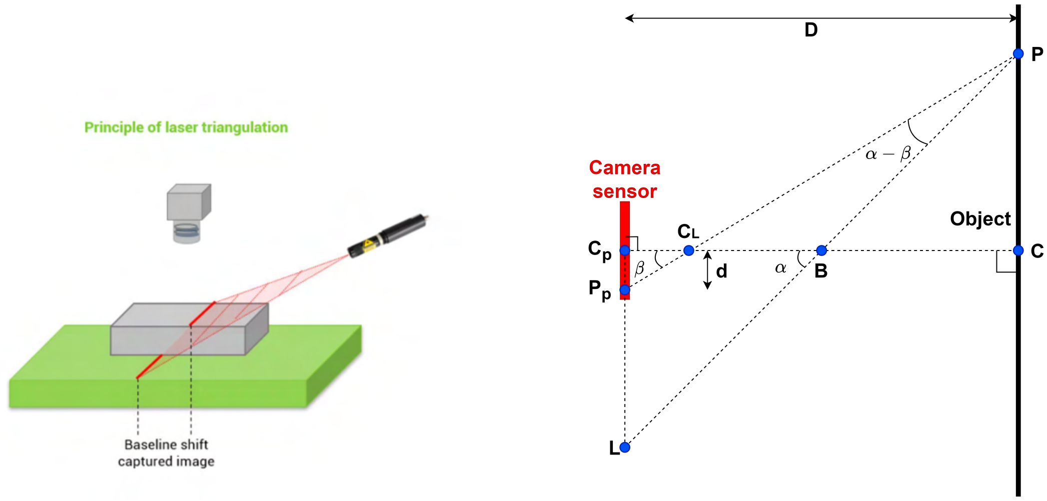


Fig. 5.9 Workflow of Proposed Solution

5.5.1 Positioning by Triangulation

To calculate distances and depth maps to various objects or to sketch the contour of the things under study, we used a laser and a camera, as demonstrated in Fig. 5.10a. We schematically draw this technique in Fig. 5.10b:



(a) Method based on laser and camera.[144]

(b) Geometric scheme.[144]

Fig. 5.10 Triangulation schematic.

In order to be able to compute the depth of the tread, it is essential to know: the angle between the camera and the laser itself α (in Fig. 5.10b), and the distance between the laser's and the camera's LC_p (in Fig. 5.10b). Both the angle and the distance mentioned was defined in Section 5.2, also the focal length of the camera is known from the technical information of the lenses used for the camera (C_pC_L in Fig. 5.10b).

Based on the specification of the camera sensor, we know the size of each pixel and we can calculate the millimeter-long distance of C_pP_p in Fig. 5.10b. Knowing all data we can calculate:

$$C_LP = \frac{BC_L \cdot \sin(180^\circ - \alpha)}{\sin(\alpha - \beta)} \tag{5.1}$$

According to the triangulation principles, the only way to determine the distance of any point from the camera is to know which pixel of the image it is represented in. The formulas described above regarding triangulation are valid, however, they do not account for lens imperfections (distortion) or the addition of a filter between the lens and the camera. For our implementation, we used an empirical method to calibrate and find the corresponding pixel for each distance where the laser beam can be located. This method naturally maintains the theory of the triangulation process.

As shown in Fig. 5.11 the tirebot, including the camera and laser, was mounted in front of a white wall parallel to it. Then, we moved the robot from 10cm with a step of 1 cm up to 35 cm away from the wall, took an image at every step, and saved it. Based on the captured image and the distance where it was taken, we applied interpolation to obtain a function that takes a given pixel index as an input and returns a specific distance.

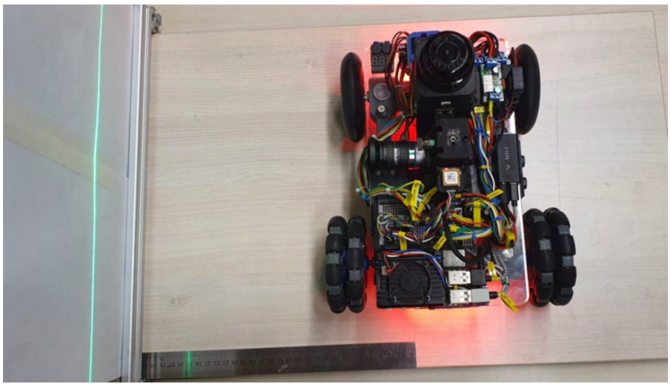


Fig. 5.11 Calibration setup to determine the relationship between pixel position and distance to the laser beam.

5.5.2 Image Acquisition

For an accurate measurement of the groove depth on the tire, the angle at which the laser wave strikes the tire is important. The ideal situation is when the sensor of the camera is parallel to the surface of the wheel. If this condition is not met, the measured distances will be greater than the real ones.

Due to the camera and laser frame being mounted on a gimbal, the optical assembly can be turned until the camera becomes parallel to the front of the tire, which makes the optimal position possible to find.

In Fig. 5.12 we show how the optical system is turned to the position for scanning the tire.

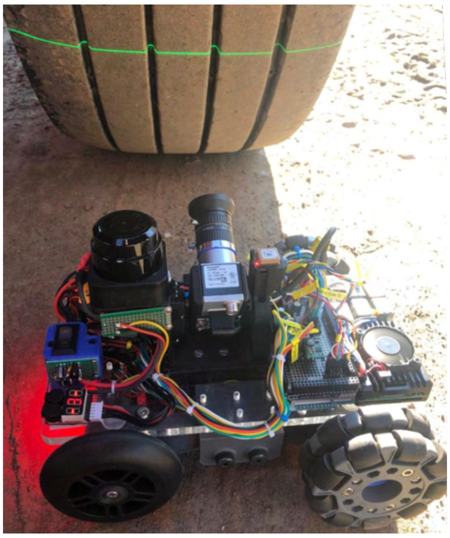
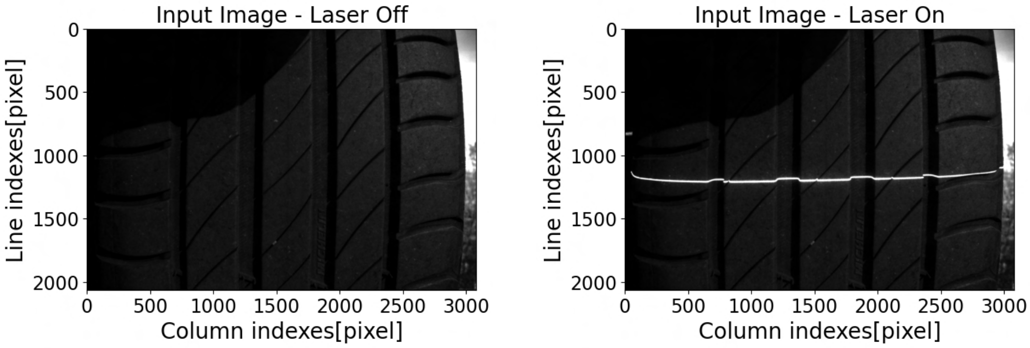


Fig. 5.12 Example of the optimal position for the camera and laser beam.

Finally, the first step of the positioning algorithm returns two images with the tire viewed at 90 degrees, Fig. 5.13.



(a) Laser beam OFF

(b) Laser beam ON

Fig. 5.13 The input images in the two cases

5.5.3 Ambient Light Correction

Detecting the laser beam in the image and measuring the real value of the depth of the tire might be difficult because of the ambient light and its reflections. To reduce the effect of ambient light we chose to take two images of the tire, one when the laser is off, and the other one when the laser is on, and then subtract one from the other. The resulting image can be seen in Fig. 5.14.

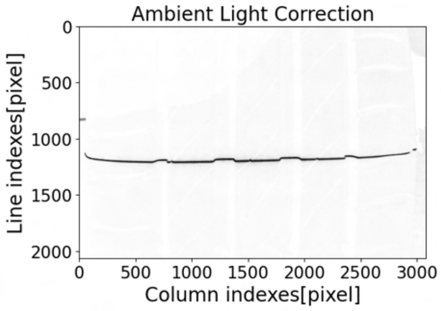


Fig. 5.14 The input image resulted from the pixel difference between the images in the Fig. 5.13.

5.5.4 Pixel to Distance Conversion

At this step, we use as input the image from Fig. 5.14. According to the explanations on triangulation in Section 5.5.1, in the image, the laser beam on the surface of the tire is lower than the laser beam on the tread. From the image, we extract an array that contains the brightest pixel on every column and its index. This array represents the laser beam on the tire. The obtained array is converted into a distance using the principles from Section 5.5.1. In Fig. 5.15 the measured distances are displayed.

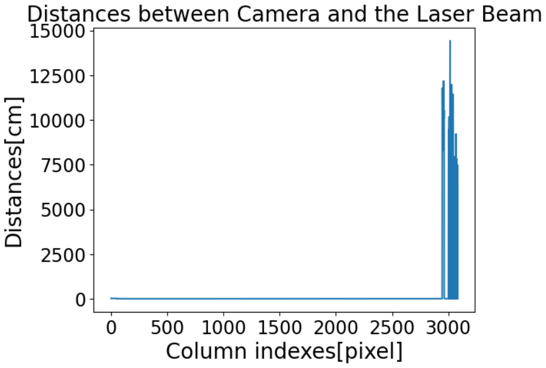


Fig. 5.15 Distances between the camera and the laser beam in cm.

5.5.5 Achieving Robustness

Based on our navigation, the tire can only be positioned between 5cm and half a meter away from the camera. We remove all the distances that are above or under the imposed limit, and as observed in Fig. 5.16, all the distances measured at this step are between the imposed limits.

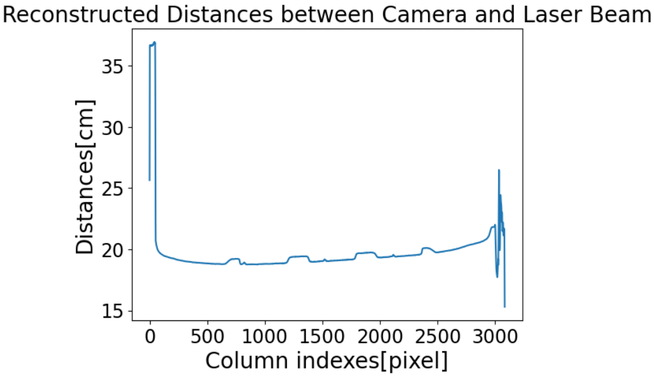
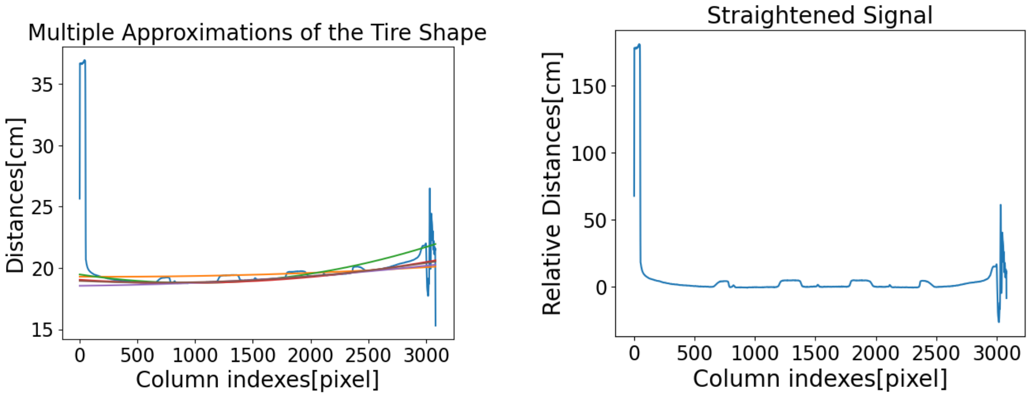


Fig. 5.16 Distances between the camera and the laser beam after removing the outliers.

5.5.6 Determining the Tire Profile

Because the shape of the signal does not correspond to the real shape of the tire, we implemented a solution to find an approximation of the signal using a parabola of a second-degree function, which will then be subtracted from the actual signal. In Fig. 5.17a, multiple generated parabolas approximating the signal were drawn, and in Fig. 5.17b, the signal after subtraction.



(a) The shape of the parabolas fitted to the shape of the tire.

(b) Distances between the camera and the laser beam after compensating for the shape of the tire.

Fig. 5.17 Tire Profile approximation

5.5.7 Removal of Margins

To eliminate the possibility of still having outliers caused by ambient light and/or to eliminate the edges of the tire, we made another refinement step. We started from the assumption that the surface of the tire is the closest object to the camera, so the furthest distance cannot exceed the cutLimit. This step returns all values under cutLimit, as seen in Fig. 5.18.

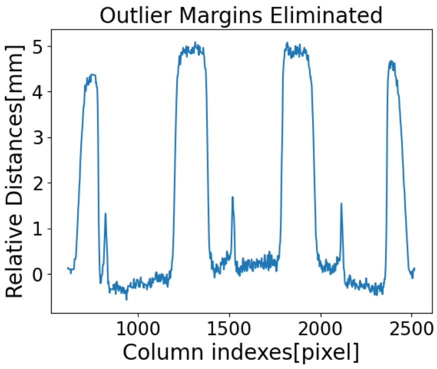
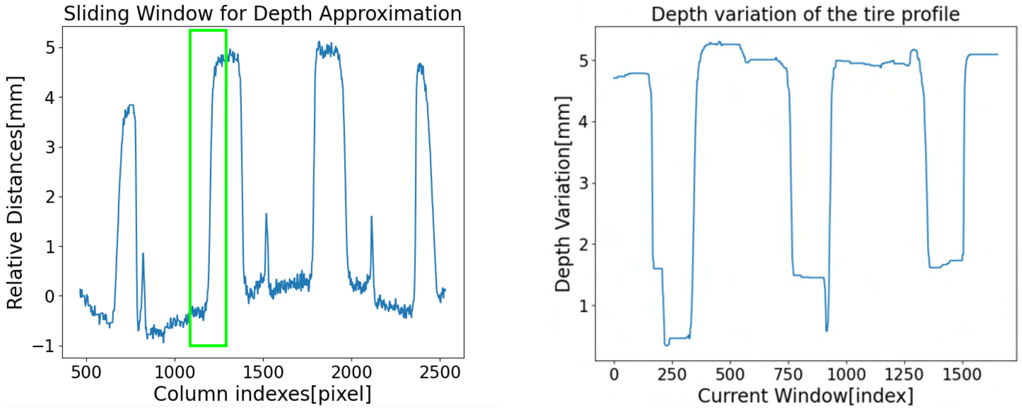


Fig. 5.18 Final signal shape of distances between the camera and the laser beam after a new outliers removal.

5.5.8 Depth Scanning Method

To finally return the depth of all steps and detect how many there are, we implemented an algorithm that uses a sliding window technique to determine the difference between the largest and smallest distance. It slides through the signal from one end to the other and returns the difference between the maximum value and the minimum value included in the window. The lower values represent the tire surface area and the higher values represent the tread depth.



(a) Sliding window moving on the final signal shape.

(b) Depth variation of the tire profile.

Fig. 5.19 Input(a) and output(b) images from the depth scanning algorithm

We illustrated in Fig. 5.19a the total number of depth fluctuations found during the input signal and the value of the depths are shown in Fig. 5.19b. As a result, the maximum depth that was detected for the tire under study in this chapter was 5.32mm, which is not far from the manually scanned depth of 5.22mm.

5.6 Outcomes Assessment

To demonstrate the efficiency of the algorithms used to calculate tire wear using positioning principles, we created a database of 360 photos, which is equivalent to 180 different tires. There are two images of each tire, given the method used for ambient light adjustment: one image with the laser on and one with the laser off. For every automatic scan of the tire, we manually measured on-site the ground truth wear of the tire with a digital caliper made specifically to gauge groove depth, as depicted in Fig.5.20

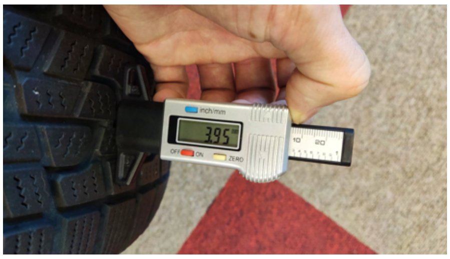


Fig. 5.20 Manual measurement of tire tread with a digital caliper.

The dataset for testing comprises different variations such as: tire type, tire wear degree, tread shape, environmental temperature, environmental illumination, car shapes (resulting in varying measurement distances and angles).

We ran the algorithm for the 180 tires and we obtained the error distribution illustrated in Fig.5.21 and an average inaccuracy of 0.28mm. As can be seen, the error distribution naturally takes the shape of a normal distribution with its center at 0mm. This means that the majority of automatic scans produced results that were comparable to those performed manually, which represent the standard in the current experiment, thus demonstrating the accuracy of the automatic approach to detecting tire wear.

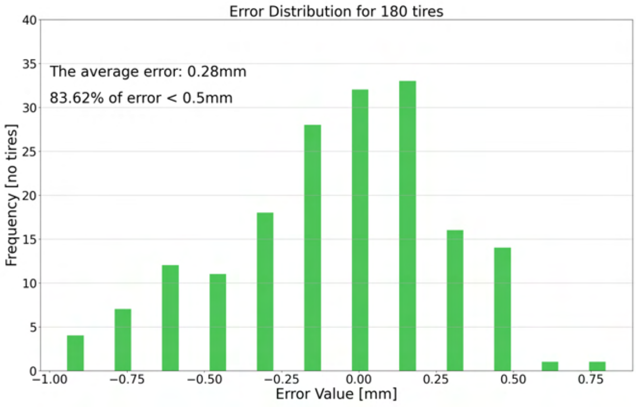


Fig. 5.21 Error distribution for 180 tires.

The majority of automated measurements have produced acceptable errors, such that 83.62% of scans had an inaccuracy smaller than 0.5mm. This proportion demonstrates the method's efficacy and expanded possibilities. Also at the same time, we measured $\Delta time$ needed for finding the position of the laser beam projected on the tire. We may observe from Fig.5.22 the distribution of running time that was obtained by running the algorithm 150 times.

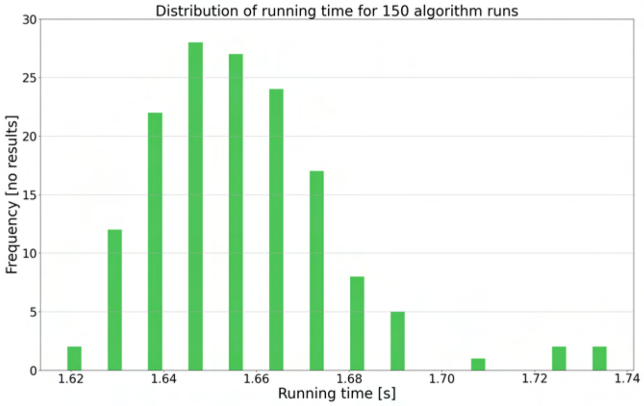


Fig. 5.22 Distribution of running time for 150 algorithm runs.

Due to tire outline approximation, the algorithm is not deterministic. The proposed method produces different values for the maximum depth for each of the repeated runs on the data set, on the same tire, and from the same perspective. After 150 runs, the average depth of the tire is 5.32mm, resulting in an error of 0.1mm compared to the real value of 5.22mm. The distribution of the depth values acquired for one of the scanned tires is shown in Fig. 5.23.

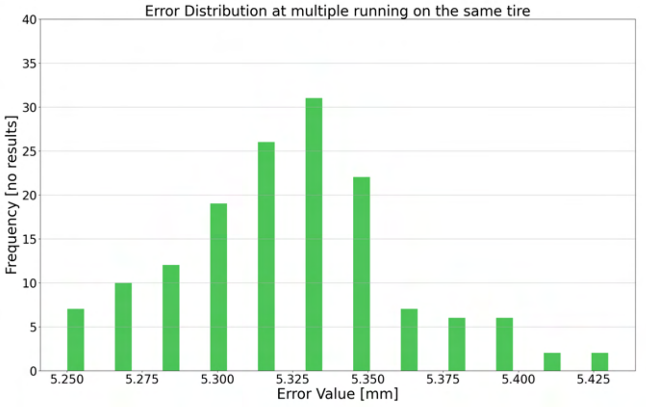


Fig. 5.23 Error distribution at multiple runs on the same tire.

5.7 Conclusions

We design, implemented, and evaluated a novel solution for tire wear measurements. The solution is represented by an autonomous mobile robot that can move under parked cars or trucks and scans their tires using an optical system consisting of a camera and a linear laser. The robot can measure the degree of tire wear automatically, precisely, objectively, and periodically, leading to increased profit for transport companies, less lost time for traffic participants, better safety on the roads, as well as to a cleaner environment. Wear detection was tested on 180 tires obtaining an average error of 0.28 mm with a run time of 1.65 s. Also, 83.62% of scans had an inaccuracy smaller than 0.5 mm.

Chapter 6

Conclusions

6.1 Obtained Results

The importance of optimizing the navigation of small autonomous robots in GPS-denied environments cannot be overstated, as these robots are increasingly being utilized in a wide variety of industrial, commercial, and various specialized applications. While advancements in sensors, algorithms, and wireless communication technologies have made autonomous robots more ubiquitous, they still face many limitations in terms of physical size, electrical power availability, and price tag. Therefore, it is crucial to develop smart, efficient, and flexible solutions that utilize smaller computing platforms and lighter, more affordable sensors to empower these robots to navigate their surroundings. The present doctoral thesis provides exploratory insights, designs, implementations, and validated solutions for positioning and navigation of autonomous robots in GPS-denied scenarios, both indoors and outdoors. By testing out these solutions on real-life applications, the thesis is contributing to the development of practical and effective methods for optimizing the navigation of autonomous robots in a wide range of contexts.

The research has focused on identifying and further developing the best methods for localization in GPS-denied environments for small, power-constrained autonomous robots. Additionally, the thesis has investigated and proposed solutions for improvements in positioning accuracy, using new wireless chips in small-scale, embedded system based robots. Finally, the thesis has proposed a novel approach for improved real-time relative positioning between two 3D systems, such as an aerial drone and a mobile target.

The first main contribution of the thesis consists of a comprehensive study on drone tracking with an active gimbal system using a sensor fusion approach. Firstly, a hardware architecture was designed to validate the proposed approach for asset tracking. The architecture integrates data from multiple sensors in a sensor-fusion type system, and a complete hardware set-up was implemented based on commercial grade drone parts and integrating multiple sensors, including Inav integration, GPS data, OpticalFlow data, RangeFinder data, optical camera data, a powerful Jetson Nano processing platform, and additional electronics for control and power management. Secondly, a new detection and control algorithm was proposed, implemented, and validated for controlling the drone through a real-time adjustment system relative to a moving target. The algorithm was deployed on the proposed hardware set-up and found to afford a significantly enhanced target acquisition and target lock solution. Thirdly, a novel classification for current fiducial marker systems was proposed. To test out the system, a data set comprised of 13200 pictures was developed, and a testing methodology was proposed and tested on three fiducial marker systems on the proposed data set, together with a hardware set-up for assessing extra parameters such as performance at certain angular velocities.

For the second main contribution of the thesis, concerning positioning, the development and validation of a hardware architecture for an improved indoor high precision localization system using the novel proposed approach FlexTDOA were proposed. A complete test bed was designed, built and tested, comprising an array of custom-made sensor nodes for testing, adjusting and improving the proposed algorithm. Secondly, two methods were developed to measure distances between nodes: the first is based on Single-Sided Two-Way Ranging (SS-TWR), which uses two message exchanges between two nodes, and the second is based on the Time Difference of Arrival (TDOA), which uses the time difference between the arrival of two messages. Two algorithms were also implemented that use TWR or TDOA measurements to compute the location of the node that is tracked. The first algorithm computes the location of the moving tag iteratively using an Extended Kalman Filter (EKF), and the second algorithm computes the location using least square error minimization between the measured and the calculated distances. Thirdly, a custom flexible time-division multiple-access (TDMA) scheme was developed, based on an approach where anchors do not need to have their clocks synchronized. Fourthly, based on the above a complex software system named FlexTDOA was proposed and tested out, as a custom localization method with both fixed and mobile anchors, that is simultaneously fast, precise, and scalable. Fifthly, a ground-truth system was designed, implemented, and validated in order to have a reference point for the experimental results involving indoor positioning. Lastly, the localization accuracy of FlexTDOA was evaluated in different scenarios, varying the number of responses, the order of responses, and the number of anchors, both in Line of Sight (LOS) and in Non-Line of Sight (NLOS) scenarios. The effect of the physical speed of the tag on the choice of optimum system parameters was also simulated and evaluated. FlexTDOA was compared against the classic TDOA approaches and range-based localization - and found to have several distinct advantages.

The third main contribution of the thesis concerns navigation without GPS. For this section, a new technique for automated tire wear measurements was proposed, emphasizing practicality and mobility in real-world scenarios. The research contributions include the design of a hardware architecture that was tested in both laboratory and industrial settings, along with the implementation of a robust mathematical model for tire depth extraction from automatically obtained images. Additionally, an autonomous driving system was developed that fuses data from LIDAR and optical flow sensors, to enable the robot to self-navigate around a given truck, identify tire position and self-position for optimal image acquisition. Multiple hardware subsystems were designed and tested to ensure a robust, efficient hardware implementation that addresses various real-world problems. A communication API was implemented between the image processing and navigation planning board and the speed control and acquisition board. Overall, the proposed approach shows promising results for accurate tire wear measurements and provides a practical and efficient solution for real-world applications.

The following subsections present a detailed outline of the original contributions of the thesis in relation to the state of the art, and the scientific publications resulting from the doctoral research.

6.2 Original Contributions

- **Fast and Reliable Real-Time Tracking of Moving Targets**
  - I designed a hardware architecture to validate the proposed approach for using an active gimbal system for asset tracking that integrates data from multiple sensors in a sensor-fusion type of approach.
  - I implemented a complete hardware set-up, based on commercial-grade drone parts and integrating multiple sensors, including - Inav integration, GPS data, OpticalFlow data and RangeFinder data, camera data together with a powerful Jetson Nano processing platform and additional electronics for control and for power management, in order to test out the proposed approach.
  - I proposed, implemented, and validated a new detection and control algorithm for controlling the drone through a real-time adjustment system relative to a moving target, which allows for a much better target acquisition and target lock in real-world environments.
  - I proposed a classification for current fiducial marker systems.
  - I developed a data set of 13200 pictures for testing out the system.
  - I proposed a testing methodology and tested three fiducial marker systems with the help of the obtained data set, together with a hardware set-up for assessing extra parameters (e.g.: performance at certain angular velocities).
- **Ubiquitous Positioning**
  - I designed a hardware architecture to validate the proposed approach (FlexTDOA) for an improved indoor high-precision localization system.
  - I designed, built, and tested a complete test bed - comprising an array of custom-made sensor nodes for testing, adjusting, and improving the proposed algorithm.
  - I developed two methods to measure distances between nodes. The first is based on Single-Sided Two-Way Ranging (SS-TWR) which uses two message exchanges between two nodes, and the second is based on the Time Difference of Arrival (TDOA), which uses the time difference between the arrival of two messages.
  - I implemented two algorithms that use TWR or TDOA measurements to compute the location of the node that we track. The first algorithm computes the location of the moving tag iteratively, using an Extended Kalman Filter (EKF) and the other algorithm computes the location using a least square error minimization between the measured and the calculated distances.

- I developed a custom flexible time-division multiple-access (TDMA) scheme in which time is divided into slots, which manages the moments when every anchor can start communication. In each time slot, one anchor interrogates one or more anchors, which respond in the same slot. The anchors do not need to have their clocks synchronized.
- I proposed and tested out a complex software system named FlexTDOA, which is a custom localization method that uses time-difference of arrival (TDOA) measurement so that the user device remains passive and can compute its location simply by listening to the communication between the fixed anchors, ensuring the scalability of the system.
- I designed, implemented, and validated a ground-truth system in order to have a validated reference point for the experimental results involving indoor positioning.
- I evaluated the localization accuracy of FlexTDOA in different scenarios such as varying the number of responses, the order of responses, and the number of anchors, in Line of Sight (LOS) and Non Line of Sight(NLOS) scenarios. Also, I simulated and evaluated the effect of the physical speed of the tag on the choice of optimum system parameters.
- I compared FlexTDOA against the classic TDOA approach and range-based localization in a deployment of ten anchors and one tag, both with and without obstructions.

- **Navigation without GPS**
  - I designed a hardware architecture to provide a new technique for automated tire wear measurements - with a focus on a practical and mobile solution that can be deployed swiftly in real-world scenarios.
  - I built a complete hardware and software implementation - in order to test out the proposed approach both in a laboratory scenario and in an industrial or practically relevant scenario (such as an industrial parking lot with real trucks).
  - I implemented a robust mathematical model for a multi-stage processing Algorithm, in order to extract the tire depth from images in a broad case of scenarios, under less-than-ideal conditions. The algorithm first receives as input a black and white image of the wheel, on which a laser line beam was projected. It then has 5 processing steps: the first detects the laser line in the image and converts it into a distance on a graph; the second removes outliers caused by sunlight; the third removes the distortions caused by the camera lens; the fourth removes the edges of the signal (because they are

representative of the edges of the envelope), and the last step determines the depth variation using a sliding window.

- I implemented and tested an autonomous driving system that fuses data from optical flow sensors and LIDAR to allow the robot to self-navigate around the truck and find the truck tires.
- I designed and tested out multiple hardware subsystems, in order to obtain a robust hardware implementation, accounting for multiple real-world problems (e.g.: an advanced power management system).
- I implemented a communication API between the image processing and navigation planning board, and the speed control and acquisition board.

6.3 List of Original Publications

Journals

- **George-Cristian Pătru**, Laura Flueratoru, Iuliu Vasilescu, Dragoș Niculescu, and Daniel Rosner. FlexTDOA: Robust and Scalable Time-Difference of Arrival Localization Using Ultra-Wideband Devices. *IEEE Access*, 11:28610–28627, 2023. [145]

- Vladimir Tanasiev, **George Cristian Pătru**, Daniel Rosner, Gabriela Sava, Horia Necula, and Adrian Badea. Enhancing environmental and energy monitoring of residential buildings through iot. *Automation in Construction*, 126:103662, 2021. [146]

Journals: accepted to publication

- **George-Cristian PĂTRU**, Alina-Irina PÎRVAN, Daniel ROSNER and Răzvan-Victor RUGHINIȘ. Fiducial marker systems overview and empirical analysis of ArUco, AprilTag and CCTag. *University POLITEHNICA of Bucharest Scientific Bulletin. Series C: Electrical Engineering and Computer Science*. 2023

Conferences

- **George-Cristian Pătru**, Iuliu Vasilescu, Daniel Rosner, and Dan Tudose. Aerial drone platform for asset tracking using an active gimbal. In *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pages 138–142. IEEE, 2021. [147]

- **George-Cristian Pătru**, Dumitru-Cristian Trancă, Ciprian-Marian Costea, Daniel Rosner, and Răzvan-Victor Rughiniș. Lora based, low power remote monitoring and control solution for industry 4.0 factories and facilities. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019. [148]

- Denis Ilie-Ablachim, **George Cristian Pătru**, Iulia-Maria Florea, and Daniel Rosner. Monitoring device for culture substrate growth parameters for precision agriculture: Acronym: Monisen. In *2016 15th RoEduNet Conference: Networking in Education and Research*, pages 1–7. IEEE, 2016. [149]

- Alina Irina Pîrvan, **George Cristian Pătru**, Dumitru Cristian Trancă, Cristian Contașel, and Daniel Rosner. Infrastructure independent rail quality diagnosis and monitoring system. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–5. IEEE, 2019. [150]

- Andrei-Bogdan Stanescu, Vlad-Ioan Pantea, and **George Cristian Patru**. Application of mobile technology in sport leisure time activities. In *The International Scientific Conference eLearning and Software for Education*, volume 3, page 215. " Carol I" National Defence University, 2017. [151]

- Dumitru-Cristian TRANCĂ, Eugen BUZILĂ, Daniel Rosner, **George Cristian PĂTRU**, and Răzvan Victor RUGHINIȘ, . Intact industrial internet of things communication solution. *University POLITEHNICA of Bucharest Scientific Bulletin. Series C: Electrical Engineering and Computer Science*, 2018. [152]

- Vladimir Tanasiev, Horia Necula, Adrian Alistar, **George Cristian Pătru**, and Adrian Badea. Energy-efficient solution for smart lighting through iot. In *2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM)*, pages 1–4. IEEE, 2021. [153]

- Andra-Laura Antonache, Silvia Cristina Stegaru, Mihail-Bogdan Carutașiu, and **Cristian Patru**. Modeling a thermal area for energy consumption estimation using artificial neural networks. In *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–4. IEEE, 2020. [154]

- Daniel Marian Nicolescu, Răzvand Tătăroiu, Dumitru Cristian Trancă, and **George Cristian Pătru**. Logger and analyser for modbus-based industrial networks. In *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–4. IEEE, 2020. [155]

References

[1] Md Mahbubur Rahman, Vahideh Moghtadaiee, and Andrew G. Dempster. Design of fingerprinting technique for indoor localization using am radio signals. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, 2017.

[2] Suhap Sahin, Hikmetcan Ozcan, and Kerem Kucuk. Smarttag: An indoor positioning system based on smart transmit power scheme using active tags. *IEEE Access*, 6:23500–23510, 2018.

[3] Wei Fang, Changjun Xie, and Bin Ran. An accurate and real-time commercial indoor localization system in lte networks. *IEEE Access*, 9:21167–21179, 2021.

[4] Peihao Li, Xu Yang, Yuqing Yin, Shouwan Gao, and Qiang Niu. Smartphone-based indoor localization with integrated fingerprint signal. *IEEE Access*, 8:33178–33187, 2020.

[5] Primož Bencak, Darko Hercog, and Tone Lerher. Indoor positioning system based on bluetooth low energy technology and a nature-inspired optimization algorithm. *Electronics*, 11(3), 2022.

[6] Dongqing Shi, Haiyan Mi, Emmanuel G. Collins, and Jun Wu. An indoor low-cost and high-accuracy localization approach for agvs. *IEEE Access*, 8:50085–50090, 2020.

[7] J. P. Matos-Carvalho, Ricardo Santos, Slavisa Tomic, and Marko Beko. Gtrs-based algorithm for uav navigation in indoor environments employing range measurements and odometry. *IEEE Access*, 9:89120–89132, 2021.

[8] Weide You, Fanbiao Li, Liqing Liao, and Meili Huang. Data fusion of uwb and imu based on unscented kalman filter for indoor localization of quadrotor uav. *IEEE Access*, 8:64971–64981, 2020.

[9] Shih-Yi Huang and Ruey-Beei Wu. Positioning for search and rescue in gps-denied area by distributed wifi rss-based doa modules. *IEEE Access*, 10:76105–76113, 2022.

[10] Alireza Famili, Angelos Stavrou, Haining Wang, and Jung-Min Jerry Park. Rail: Robust acoustic indoor localization for drones. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–6, 2022.

[11] Philipp Stockel, Patrick Wallrath, Nils Pohl, and Reinhold Herschel. High accuracy position calculation of a hovering uav using a rotating radar. In *2022 19th European Radar Conference (EuRAD)*, pages 129–132, 2022.

[12] Martin Oelsch, Mojtaba Karimi, and Eckehard Steinbach. Init-loam: Lidar-based localization and mapping with a static self-generated initial map. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 865–872, 2021.

[13] Adam Norton, Peter Gavriel, Brendan Donoghue, and Holly Yanco. Test methods to evaluate mapping capabilities of small unmanned aerial systems in constrained indoor and subterranean environments. In *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–8, 2021.

[14] Khairuldanial Ismail, Ran Liu, Jie Zheng, Chau Yuen, Yong Liang Guan, and U-Xuan Tan. Mobile robot localization based on low-cost lte and odometry in gps-denied outdoor environment. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2338–2343, 2019.

[15] Seyed Jamal Haddadi and Eugenio B. Castelan. Visual-inertial fusion for indoor autonomous navigation of a quadrotor using orb-slam. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 106–111, 2018.

[16] Giovanni Miraglia, K. Niki Maleki, and Loyd R. Hook. Comparison of two sensor data fusion methods in a tightly coupled uwb/imu 3-d localization system. In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 611–618, 2017.

[17] Adil Farooq, Antreas Anastasiou, Nicolas Souli, Christos Laoudias, Panayiotis S. Kolios, and Theocharis Theocharides. Uav autonomous indoor exploration and mapping for sar missions: Reflections from the icuas 2022 competition. In *2022 19th International Conference on Ubiquitous Robots (UR)*, pages 621–626, 2022.

[18] Oluwatayo Kolawole and Mythri Hunukumbure. A drone-based 3d localization solution for emergency services. In *ICC 2022 - IEEE International Conference on Communications*, pages 1–6, 2022.

[19] Xiangqian Shu, Lingyu Yang, Xiaoke Feng, and Jing Zhang. An imu/sonar-based extended kalman filter for mini-uav localization in indoor environment. In *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, pages 1–6, 2018.

[20] Christian Gentner, Markus Ulmschneider, Isabel Kuehner, and Armin Dammann. Wifi-rtt indoor positioning. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 1029–1035, 2020.

[21] Biljana Risteska Stojkoska, Jordan Palikrushev, Kire Trivodaliev, and Slobodan Kalajdziski. Indoor localization of unmanned aerial vehicles based on rssi. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pages 120–125, 2017.

[22] Jirapat Sangthong, Jutamas Thongkam, and Sathapom Promwong. Indoor wireless sensor network localization using rssi based weighting algorithm method. In *2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pages 1–4, 2020.

[23] Weizhong Ding, Shengming Chang, and Jun Li. A novel weighted localization method in wireless sensor networks based on hybrid rss/aoa measurements. *IEEE Access*, 9:150677–150685, 2021.

[24] Po Ting Lin, Che-An Liao, and Shu-Hao Liang. Probabilistic indoor positioning and navigation (pipn) of autonomous ground vehicle (agv) based on wireless measurements. *IEEE Access*, 9:25200–25207, 2021.

[25] Md Moin Uddin Chowdhury, Fatih Erden, and Ismail Guvenc. Rss-based q-learning for indoor uav navigation. In *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pages 121–126, 2019.

[26] Srinivas Karuparthi and Taşkın Padır. System design for quadrant-based indoor localization of emergency responders. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6, 2019.

[27] Tong Wu, Min Lu, Zhipeng Xi, and Dengkun Xiao. Uwb-based multi-source fusion positioning for cooperative uavs in complex scene. In *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2022.

[28] Kang Zhao, Minghua Zhu, Bo Xiao, Xuguang Yang, Changlei Gong, and Junyi Wu. Joint rfid and uwb technologies in intelligent warehousing management system. *IEEE Internet of Things Journal*, 7(12):11640–11655, 2020.

[29] Peter Krapež and Marko Munih. Uwb-radio distance measurements error mitigation due to tag orientation in localization systems. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 836–841, 2021.

[30] Jian Wang, Minmin Wang, Deng Yang, Fei Liu, and Zheng Wen. Uwb positioning algorithm and accuracy evaluation for different indoor scenes. *International Journal of Image and Data Fusion*, 12(3):203–225, 2021.

[31] Leyla Nosrati, Mohammad Sadegh Fazel, and Mohammad Ghavami. Improving indoor localization using mobile uwb sensor and deep neural networks. *IEEE Access*, 10:20420–20431, 2022.

[32] Yu Xianjia, Li Qingqing, Jorge Peña Queralta, Jukka Heikkonen, and Tomi Westerlund. Applications of uwb networks and positioning to autonomous robots and industrial systems. In *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–6, 2021.

[33] Felix Vollmer, Jan Graßhoff, and Philipp Rostalski. Probabilistic ultra-wideband tdoa localization with bias correction. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1512–1516, 2022.

[34] Roman S. Kulikov. Integrated uwb/imu system for high rate indoor navigation with cm-level accuracy. In *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, pages 1–4, 2018.

[35] Huei-Yung Lin and Ming-Chi Yeh. Drift-free visual slam for mobile robot localization by integrating uwb technology. *IEEE Access*, 10:93636–93645, 2022.

[36] Youssef Ibnatta, Mohammed Khaldoun, and Mohammed Sadik. Indoor localization system based on mobile access point model mapm using rss with uwb-ofdm. *IEEE Access*, 10:46043–46056, 2022.

[37] Gang Li, Xiang Liao, Huilan Huang, Shaojian Song, Bin Liu, and Yawen Zeng. Robust stereo visual slam for dynamic environments with moving object. *IEEE Access*, 9:32310–32320, 2021.

[38] Jiang Dong, Marius Noreikis, Yu Xiao, and Antti Ylä-Jääski. Vinav: A vision-based indoor navigation system for smartphones. *IEEE Transactions on Mobile Computing*, 18(6):1461–1475, 2019.

[39] Xing Zhang, Jing Lin, Qingquan Li, Tao Liu, and Zhixiang Fang. Continuous indoor visual localization using a spatial model and constraint. *IEEE Access*, 8:69800–69815, 2020.

[40] Wei Zhao, Liangjie Xu, Bozhao Qi, Jia Hu, Teng Wang, and Troy Runge. Vivid: Augmenting vision-based indoor navigation system with edge computing. *IEEE Access*, 8:42909–42923, 2020.

[41] Chengyi Zhang. Sparse visual localization in gps-denied indoor environments. In *2019 International Conference on Information Technology and Computer Application (ITCA)*, pages 87–91, 2019.

[42] Weipeng Guan, Shihuan Chen, Shangsheng Wen, Zequn Tan, Hongzhan Song, and Wenyuan Hou. High-accuracy robot indoor localization scheme based on robot operating system using visible light positioning. *IEEE Photonics Journal*, 12(2):1–16, 2020.

[43] Aubida A. Al-Hameed, Safwan Hafeedh Younus, Ahmed Taha Hussein, Mohammed Thamer Alresheed, and Jaafar M. H. Elmirghani. Lidal: Light detection and localization. *IEEE Access*, 7:85645–85687, 2019.

[44] Hriday Bavle, Paloma De La Puente, Jonathan P. How, and Pascual Campoy. Vps-slam: Visual planar semantic slam for aerial robotic systems. *IEEE Access*, 8:60704–60718, 2020.

[45] Jonathan Putra and Dany Eka Saputra. Autonomous drone indoor navigation based on virtual 3d map reference. In *2022 International Conference on ICT for Smart Society (ICISS)*, pages 01–05, 2022.

[46] Rizqy Ilmi Naufal, Nyoman Karna, and Soo Young Shin. Vision-based autonomous landing system for quadcopter drone using openmv. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1233–1237, 2022.

[47] Qing Liang and Ming Liu. A tightly coupled vlc-inertial localization system by ekf. *IEEE Robotics and Automation Letters*, 5(2):3129–3136, 2020.

[48] Fares Tarek Ali, Omar AbdulAziz Fahmy, and Ayman A. El-Badawy. An indoor vision-based markov localization technique of a quadrotor. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6, 2019.

[49] O. Araar and N. Aouf. Visual servoing of a quadrotor uav for the tracking of linear structured infrastructures. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3310–3315, 2013.

[50] Jorge A. Sarapura, Flavio Roberti, Ricardo Carelli, and José M. Sebastián. Passivity based visual servoing of a uav for tracking crop lines. In *2017 XVII Workshop on Information Processing and Control (RPIC)*, pages 1–6, 2017.

[51] Xuefei Wang, Yizhen Yin, Hongjun Ma, and Hua Bai. Image-based visual servoing of unmanned aerial vehicles for variable angle target. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 6065–6070, 2021.

[52] Xuetao Zhang, Yongchun Fang, Xiao Liang, and Xuebo Zhang. Geometric adaptive dynamic visual servoing of a quadrotor uav. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 312–317, 2016.

[53] Yongwei Zhang, Yangguang Yu, Shengde Jia, and Xiangke Wang. Autonomous landing on ground target of uav by using image-based visual servo control. In *2017 36th Chinese Control Conference (CCC)*, pages 11204–11209, 2017.

[54] Yihang Li, Guozheng Lu, Dongjiao He, and Fu Zhang. Robocentric model-based visual servoing for quadrotor flights. *IEEE/ASME Transactions on Mechatronics*, pages 1–12, 2023.

[55] Jiaxing Che, Kun Yang, Zhiming Zhou, Yi Ding, Haotian Zhang, and Quan Quan. Hi-speed visual servo docking for multicopter uav based on velocity control mode. In *2022 China Automation Congress (CAC)*, pages 6909–6914, 2022.

[56] Ravula Aparna, H. T. Sai Ruchitha, Nanawath Pranavi, and Avinash G. keskar. Imu based tracking of a person using nonlinear autoregressive exogenous(narx) algorithm in gps-denied areas. In *2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA)*, pages 1–4, 2020.

[57] Da Bin Jeong and Nak Yong Ko. Dead reckoning of a mobile robot in 2-dimensional special euclidean group. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 1069–1071, 2022.

[58] Meng Zhang, Jian Yang, Jifu Zhao, and Yanjie Dai. A dead-reckoning based local positioning system for intelligent vehicles. In *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 513–517, 2019.

[59] Wei-Wei Xue and Ping Jiang. The research on navigation technology of dead reckoning based on uwb localization. In *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 339–343, 2018.

[60] Qing-Li Zhou, Youmin Zhang, Yao-Hong Qu, and Camille-Alain Rabbath. Dead reckoning and kalman filter design for trajectory tracking of a quadrotor uav. In *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pages 119–124, 2010.

[61] Hang Zhou, Yibo Zhao, Xiaogang Xiong, Yunjiang Lou, and Shyam Kamal. Imu dead-reckoning localization with rnn-iekf algorithm. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11382–11387, 2022.

[62] Jong Tai Jang and Wonkeun Youn. Autonomous indoor proximity flight of a quadcopter drone using a directional 3d lidar and vio sensor. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pages 1981–1983, 2022.

[63] Yali Yuan, Christian Melching, Yachao Yuan, and Dieter Hogrefe. Multi-device fusion for enhanced contextual awareness of localization in indoor environments. *IEEE Access*, 6:7422–7431, 2018.

[64] Georg Fischer, Joan Bordoy, Dominik Jan Schott, Wenxin Xiong, Andrea Gabbrielli, Fabian Höflinger, Kai Fischer, Christian Schindelhauer, and Stefan Johann Rupitsch. Multimodal indoor localization: Fusion possibilities of ultrasonic and bluetooth low-energy data. *IEEE Sensors Journal*, 22(6):5857–5868, 2022.

[65] Yu-Chih Chen, Alexsander I-Chi Lai, and Ruey-Beei Wu. Uwb-assisted high-precision positioning in a utm prototype. In *2020 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, pages 42–45, 2020.

[66] Kate Blake, Robert Huey, Devaughn Menezes, James Root, Lycia Tran, and Haige Chen. Robot navigation using ultra-wideband indoor localization and dead reckoning algorithms. In *2022 Opportunity Research Scholars Symposium (ORSS)*, pages 12–15, 2022.

[67] Christoph Kammel, Tobias Kögel, Matthias Gareis, and Martin Vossiek. A cost-efficient hybrid uhf rfid and odometry-based mobile robot self-localization technique with centimeter precision. *IEEE Journal of Radio Frequency Identification*, 6:467–480, 2022.

[68] Bingjie Chen, Zhiwen Zhu, Yuanze Zhang, Shuai Yue, Bo Yu, and Haibo Du. Indoor uav positioning system based on rotational vision and cascade controller. In *2022 41st Chinese Control Conference (CCC)*, pages 3621–3626, 2022.

[69] Shengyang Ge, Feng Pan, Dadong Wang, and Pu Ning. Research on an autonomous tunnel inspection uav based on visual feature extraction and multi-sensor fusion indoor navigation system. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 6082–6089, 2021.

[70] Pawarut Karaked, Watcharapol Saengphet, and Suradet Tantrairatn. Multi-sensor fusion with extended kalman filter for indoor localization system of multirotor uav. In *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5, 2022.

[71] Meng-Gang Li, Hua Zhu, Shao-Ze You, and Chao-Quan Tang. Uwb-based localization system aided with inertial sensor for underground coal mine applications. *IEEE Sensors Journal*, 20(12):6652–6669, 2020.

[72] Zhongshuai Wang, Pheng Sokliep, Chengpei Xu, Jiayu Huang, Linfa Lu, and Zhuo Shi. Indoor position algorithm based on the fusion of wifi and image. In *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, pages 212–216, 2019.

[73] Samet Güler, Mohamed Abdelkader, and Jeff S. Shamma. Peer-to-peer relative localization of aerial robots with ultrawideband sensors. *IEEE Transactions on Control Systems Technology*, 29(5):1981–1996, 2021.

[74] Yeon Ji Choi, Tariq Rahim, I Nyoman Apraz Ramatryana, and Soo Young Shin. Improved cnn-based path planning for stairs climbing in autonomous uav with lidar sensor. In *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–7, 2021.

[75] Gennaro Ariante, Salvatore Ponte, and Giuseppe Del Core. Bluetooth low energy based technology for small uas indoor positioning. In *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 113–118, 2022.

[76] Markus Hehn, Erik Sippel, Christian Carlowitz, and Martin Vossiek. High-accuracy localization and calibration for 5-dof indoor magnetic positioning systems. *IEEE Transactions on Instrumentation and Measurement*, 68(10):4135–4145, 2019.

[77] Pei-Yuan Hong, Chi-Yu Li, Hong-Rong Chang, YuanHao Hsueh, and Kuochen Wang. Wbf-ps: Wigig beam fingerprinting for uav positioning system in gps-denied environments. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1778–1787, 2020.

[78] Huiming Xing, Yu Liu, Shuxiang Guo, Liwei Shi, Xihuan Hou, Wenzhi Liu, and Yan Zhao. A multi-sensor fusion self-localization system of a miniature underwater robot in structured and gps-denied environments. *IEEE Sensors Journal*, 21(23):27136–27146, 2021.

[79] Alireza Famili, Angelos Stavrou, Haining Wang, and Jung-Min Jerry Park. Spin: Sensor placement for indoor navigation of drones. In *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, 2022.

[80] Alireza Famili and Jung-Min Jerry Park. Rolatin: Robust localization and tracking for indoor navigation of drones. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2020.

[81] Antoine Courtay, Mickaël Le Gentil, Olivier Berder, Pascal Scalart, Sébastien Fontaine, and Arnaud Carer. Anchor selection algorithm for mobile indoor positioning using wsn with uwb radio. In *2019 IEEE Sensors Applications Symposium (SAS)*, pages 1–5, 2019.

[82] Vlad Niculescu, Daniele Palossi, Michele Magno, and Luca Benini. Energy-efficient, precise uwb-based 3-d localization of sensor nodes with a nano-uav. *IEEE Internet of Things Journal*, 10(7):5760–5777, 2023.

[83] Michail Kalaitzakis, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Experimental comparison of fiducial markers for pose estimation. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 781–789. IEEE, 2020.

[84] Mohammad Fattahi Sani and Ghader Karimian. Automatic navigation and landing of an indoor ar. drone quadrotor using aruco marker and inertial sensors. In *2017 International Conference on Computer and Drone Applications (IConDA)*, pages 102–107, 2017.

[85] Silvio RR Sanches, Daniel M Tokunaga, Valdinei F Silva, Antonio C Sementille, and Romero Tori. Mutual occlusion between real and virtual elements in augmented reality based on fiducial markers. In *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, pages 49–54. IEEE, 2012.

[86] Youngkwan Cho Jongweon Lee Ulrich Neumann. A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality. In *Proc. Int'l Workshop Augmented Reality*, pages 147–165, 1999.

[87] Joseph DeGol, Timothy Bretl, and Derek Hoiem. Chromatag: A colored marker and fast detection algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1472–1481, 2017.

[88] Zhiyuan Li, Naira Hovakimyan, Vladimir Dobrokhodov, and Isaac Kaminer. Vision-based target tracking and motion estimation using a small uav. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2505–2510. IEEE, 2010.

[89] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos. Quadrotor landing on an inclined platform of a moving ground vehicle. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2202–2207, 2015.

[90] K Boudjit and Chevif Larbes. Detection and implementation autonomous target tracking with a quadrotor ar. drone. In *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 2, pages 223–230. IEEE, 2015.

[91] Yi Feng, Cong Zhang, Stanley Baek, Samir Rawashdeh, and Alireza Mohammadi. Autonomous landing of a uav on a moving platform using model predictive control. *Drones*, 2(4):34, 2018.

[92] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.

[93] Aytaç Altan and Rıfat Hacıoğlu. Model predictive control of three-axis gimbal system mounted on uav for real-time target tracking under external disturbances. *Mechanical Systems and Signal Processing*, 138:106548, 2020.

[94] V. N. Dobrokhodov, I. I. Kaminer, K. D. Jones, and R. Ghabcheloo. Vision-based tracking and motion estimation for moving targets using small uavs. In *2006 American Control Conference*, pages 6 pp.–, 2006.

[95] Yingcai Bi and Haibin Duan. Implementation of autonomous visual tracking and landing for a low-cost quadrotor. *Optik-International Journal for Light and Electron Optics*, 124(18):3296–3300, 2013.

[96] Ali Rohan, Mohammed Rabah, and Sung-Ho Kim. Convolutional neural network-based real-time object detection and tracking for parrot ar drone 2. *IEEE Access*, 7:69575–69584, 2019.

[97] Hui Cheng, Lishan Lin, Zhuoqi Zheng, Yuwei Guan, and Zhongchang Liu. An autonomous vision-based target tracking system for rotorcraft unmanned aerial vehicles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1732–1738. IEEE, 2017.

[98] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94, 1999.

[99] Jun Rekimoto and Yuji Ayatsuka. Cybercode: designing augmented reality environments with visual tags. In *Proceedings of DARE 2000 on Designing augmented reality environments*, pages 1–10, 2000.

[100] M. Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596 vol. 2, 2005.

[101] Junaed Sattar, Eric Bourque, Philippe Giguere, and Gregory Dudek. Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction. In *Fourth Canadian Conference on Computer and Robot Vision (CRV'07)*, pages 165–174. IEEE, 2007.

[102] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. Caltag: High precision fiducial markers for camera calibration. In *VMV*, volume 10, pages 41–48, 2010.

[103] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011.

[104] Filippo Bergamasco, Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *CVPR 2011*, pages 113–120, 2011.

[105] L. Calvet, P. Gurdjos, and V. Charvillat. Camera tracking using concentric circle markers: Paradigms and algorithms. In *2012 19th IEEE International Conference on Image Processing*, pages 1361–1364, 2012.

[106] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[107] Michail Kalaitzakis, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems*, 101(4):1–26, 2021.

[108] Mark Fiala. Comparing artag and artoolkit plus fiducial marker systems. In *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pages 6–pp. IEEE, 2005.

[109] Sergio Garrido-Jurado, Rafael Munoz-Salinas, Francisco José Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 2016.

[110] Gaetano C La Delfa, Salvatore Monteleone, Vincenzo Catania, Juan F De Paz, and Javier Bajo. Performance analysis of visualmarkers for indoor navigation systems. *Frontiers of Information Technology & Electronic Engineering*, 17(8):730–740, 2016.

[111] Lilian Calvet, Pierre Gurdjos, Carsten Griwodz, and Simone Gasparini. Detection and accurate localization of circular fiducials under highly challenging conditions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 562–570, 2016.

[112] Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and vision Computing*, 76:38–47, 2018.

[113] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[114] iNavFlight. inavflight/inav. http://www.inavflight.com/, Accessed on: Feb. 10, 2023, 2023.

[115] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *IROS*, pages 1898–1903, 2019.

[116] FiRa™. Unleashing the potential of UWB: Regulatory considerations. https://www.firaconsortium.org/sites/default/files/2022-08/Unleashing-the-Potential-of-UWB-Regulatory-Considerations.pdf, August 2022.

[117] Matteo Ridolfi, Abdil Kaya, Rafael Berkvens, Maarten Weyn, Wout Joseph, and Eli De Poorter. Self-calibration and collaborative localization for UWB positioning systems: a survey and future research directions. *ACM Computing Surveys (CSUR)*, 54(4):1–27, 2021.

[118] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, 2009.

[119] Fredrik Gustafsson and Fredrik Gunnarsson. Positioning using time-difference of arrival measurements. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 6, pages VI–553. IEEE, 2003.

[120] Andreu Urruela, Josep Sala, and Jaume Riba. Average performance analysis of circular and hyperbolic geolocation. *IEEE Transactions on Vehicular Technology*, 55(1):52–66, 2006.

[121] Qorvo. Dwm3000 6.5 & 8.0 ghz ultra-wideband (uwb) module. https://www.qorvo.com/products/p/DWM3000, Accessed on: Nov. 15, 2022, 2023.

[122] Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. A robot self-localization system using one-way ultra-wideband communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137. IEEE, 2015.

[123] Mathias Pelka and Horst Hellbrück. S-TDoA—sequential time difference of arrival—A scalable and synchronization free approach for positioning. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE, 2016.

[124] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. SnapLoc: An ultra-fast UWB-based indoor localization system for an unlimited number of tags. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 61–72. IEEE, 2019.

[125] Pablo Corbalán, Gian Pietro Picco, and Sameera Palipana. Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 133–144. IEEE, 2019.

[126] Janis Tiemann, Fabian Eckermann, and Christian Wietfeld. Atlas-an open-source tdoa-based ultra-wideband localization system. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2016.

[127] Jing Yang, BaiShun Dong, and Jiliang Wang. VULoc: Accurate UWB localization for countless targets without synchronization. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–25, 2022.

[128] Davide Vecchia, Pablo Corbalán, Timofei Istomin, and Gian Pietro Picco. TALLA: Large-scale TDoA localization with ultra-wideband radios. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2019.

[129] Wenda Zhao, Abhishek Goudar, and Angela P Schoellig. Finding the right place: Sensor placement for UWB time difference of arrival localization in cluttered indoor environments. *IEEE Robotics and Automation Letters*, 7(3):6075–6082, 2022.

[130] Amanda Prorok, Phillip Tomé, and Alcherio Martinoli. Accommodation of NLOS for ultra-wideband TDOA localization in single-and multi-robot systems. In *2011 international conference on indoor positioning and indoor navigation*, pages 1–9. IEEE, 2011.

[131] Ben Van Herbruggen, Jaron Fontaine, and Eli De Poorter. Anchor pair selection for error correction in time difference of arrival (TDoA) ultra wideband (UWB) positioning systems. In *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2021.

[132] Michael Hamer and Raffaello D'Andrea. Self-calibrating ultra-wideband network supporting multi-robot localization. *IEEE Access*, 6:22292–22304, 2018.

[133] Igor Dotlic, Andrew Connell, and Michael McLaughlin. Ranging methods utilizing carrier frequency offset estimation. In *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, pages 1–6. IEEE, 2018.

[134] Juri Sidorenko, Volker Schatz, Norbert Scherer-Negenborn, Michael Arens, and Urs Hugentobler. Error corrections for ultrawideband ranging. *IEEE Transactions on Instrumentation and Measurement*, 69(11):9037–9047, 2020.

[135] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84, 2005.

[136] Moe Z Win and Robert A Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Transactions on communications*, 48(4):679–689, 2000.

[137] IEEE. IEEE standard for low-rate wireless networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pages 1–800, 2020.

[138] STMicroelectronics. Stm32f429zi - high-performance advanced line, arm cortex-m4 core with dsp and fpu, 2 mbytes of flash memory, 180 mhz cpu, art accelerator, chrom-artaccelerator, fmc with sdram, tft. https://www.st.com/en/microcontrollers-microprocessors/stm32f429zi.html, Accessed on: Nov. 15, 2022, 2023.

[139] Rahul Dasgupta. The brutal truth about the transportation industry's cutthroat profit margins. https://blog.route4me.com/transportation-profit-margins/, Accessed on: Ian. 16, 2023, 2021.

[140] THE COUNCIL OF THE EUROPEAN COMMUNITIES. Council directive of 18 july 1989 on the approximation of the laws of the member states relating to the tread depth of tyres of certain categories of motor vehicles and their trailers (89/459/eec). https://www.legislation.gov.uk/eudr/1989/459?view=plain, Accessed on: Ian. 16, 2023, 1989.

[141] TSG Romania. Technical Services Group - TSG Romania. https://www.tsg-solutions.com/ro/despre-noi/, Accessed on: Ian. 18, 2023, 2023.

[142] Raspberry Pi. Raspberry Pi 4. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, Accessed on: Ian. 10, 2023, 2023.

[143] Arduino. Arduino Due. https://store.arduino.cc/products/arduino-due, Accessed on: Ian. 10, 2023, 2023.

[144] Germany phil-vision GmbH. High-performance 3d laser triangulation systems. https://www.phil-vision.com/en/products/3d-laser-triangulation, Accessed on: Ian. 18, 2023, 2023.

[145] George-Cristian Pătru, Laura Flueratoru, Iuliu Vasilescu, Dragoş Niculescu, and Daniel Rosner. FlexTDOA: Robust and Scalable Time-Difference of Arrival Localization Using Ultra-Wideband Devices. *IEEE Access*, 11:28610–28627, 2023.

[146] Vladimir Tanasiev, George Cristian Pătru, Daniel Rosner, Gabriela Sava, Horia Necula, and Adrian Badea. Enhancing environmental and energy monitoring of residential buildings through iot. *Automation in Construction*, 126:103662, 2021.

[147] George-Cristian Pătru, Iuliu Vasilescu, Daniel Rosner, and Dan Tudose. Aerial drone platform for asset tracking using an active gimbal. In *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pages 138–142. IEEE, 2021.

[148] George-Cristian Pătru, Dumitru-Cristian Trancă, Ciprian-Marian Costea, Daniel Rosner, and Răzvan-Victor Rughiniş. Lora based, low power remote monitoring and control solution for industry 4.0 factories and facilities. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–6. IEEE, 2019.

[149] Denis Ilie-Ablachim, George Cristian Pătru, Iulia-Maria Florea, and Daniel Rosner. Monitoring device for culture substrate growth parameters for precision agriculture: Acronym: Monisen. In *2016 15th RoEduNet Conference: Networking in Education and Research*, pages 1–7. IEEE, 2016.

[150] Alina Irina Pîrvan, George Cristian Pătru, Dumitru Cristian Trancă, Cristian Contaşel, and Daniel Rosner. Infrastructure independent rail quality diagnosis and monitoring system. In *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–5. IEEE, 2019.

[151] Andrei-Bogdan Stanescu, Vlad-Ioan Pantea, and George Cristian Patru. Application of mobile technology in sport leisure time activities. In *The International Scientific Conference eLearning and Software for Education*, volume 3, page 215. " Carol I" National Defence University, 2017.

[152] Dumitru-Cristian TRANCĂ, Eugen BUZILĂ, Daniel Rosner, George Cristian PĂTRU, and Răzvan Victor RUGHINIȘ. Intact industrial internet of things communication solution. *University POLITEHNICA of Bucharest Scientific Bulletin. Series C: Electrical Engineering and Computer Science*, 2018.

[153] Vladimir Tanasiev, Horia Necula, Adrian Alistar, George Cristian Pătru, and Adrian Badea. Energy-efficient solution for smart lighting through iot. In *2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM)*, pages 1–4. IEEE, 2021.

[154] Andra-Laura Antonache, Silvia Cristina Stegaru, Mihail-Bogdan Carutaşiu, and Cristian Patru. Modeling a thermal area for energy consumption estimation using artificial neural networks. In *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–4. IEEE, 2020.

[155] Daniel Marian Nicolescu, Răzvand Tătăroiu, Dumitru Cristian Trancă, and George Cristian Pătru. Logger and analyser for modbus-based industrial networks. In *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, pages 1–4. IEEE, 2020.