

An Ultra-Wideband Localization Approach for Unknown Anchor Distributions

Martin Hesse¹, Dorit Borrmann², Andreas Nüchter¹ and Sergio Montenegro³

Abstract—Precise localization of mobile robots is key to many (semi)-autonomous operations, such as planetary exploration. In situations, where Global Navigation Satellite Systems (GNSS) is unavailable, Ultra-Wideband (UWB) technology is a common replacement. This typically relies on the positions of distributed anchors to be known beforehand. In this work we expand on a system, that remotely distributes the anchors, which means the position of them is unknown. The robot is equipped with three UWB tags, which perform Two-Way-Ranging (TWR) distance measurements with all the anchors. These distances are used to determine the relative position of the anchors to the robot. These positions are interpreted as landmarks in an Extended Kalman Filter (EKF)-Simultaneous Localization and Mapping (SLAM) algorithm, which combines them with the wheel odometry of the robot. Our experiments show promising results in a setup with four anchors, performing considerably better than the wheel odometry on its own. The system is also capable of operating through an outage of the UWB anchors. After such an outage the pose of the robot is corrected in multiple experiments, though not to the same standard as before the outage.

I. INTRODUCTION

Mobile robots play an important role in scientific research, as they have the ability to explore, inspect, map and transverse areas, that are too dangerous, difficult or remote for humans to access. For many of these tasks (semi)-autonomous operations are required, which rely on localization of the robot. On many places on Earth this is usually accomplished using Global Navigation Satellite Systems (GNSS), but there are areas where this is infeasible, including in planetary exploration outside of Earth. For this other forms of localization are necessary, which ideally do not have to be set up in advance, but can rather be put in place by the robot itself during its operation. Simultaneous Localization and Mapping (SLAM) and Ultra-Wideband (UWB) sensors are two such options.

We have designed a mobile system that distributes multiple UWB sensors using small CO₂ powered 3D-printed rockets, to create a network of these sensors, which are used for the localization of the robot [1]. This enables us to set up this localization system remotely wherever the robot goes, although the position of the anchors after the rocket launch is unknown. We tested the distribution of the sensors and a first

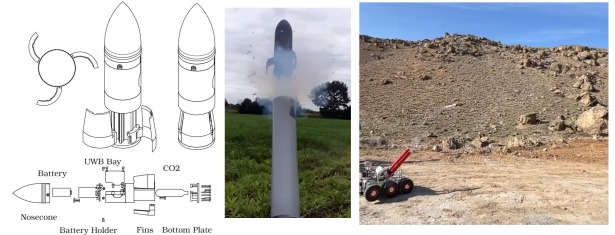


Fig. 1: Design of the rockets, including the CO₂-Cartridge propulsion system (left), rocket launch (middle), System tests during AMADEE-24 Mars analog mission (right). [1]

iteration of the positioning system at the ESA-ESRIC Space Resources Challenge in 2021 [2] and more recently in 2024 as part of the AMADEE-24 Mars analog mission of the ÖWF in Armenia [3]. A custom PCB with an UWB transceiver and a microprocessor is used to measure the distances in between the UWB nodes. This data is combined in a Kalman Filter, with wheel odometry data, which provides an estimation for the pose of the robot.

UWB positioning schemes have recently gained traction for precise short range position determination, although they require the position of the anchors to be known a priori and usually only determine the position and not the heading of the robot [4]–[6]. [7] demonstrates using passive RFID sensors the feasibility of using passive sensors as landmarks in an Extended Kalman Filter (EKF)-SLAM approach. A significant problem with the rocket based anchor distribution approach compared to a more standard position determination scheme with UWB sensors, that this paper aims to solve, is the unknown position of the anchors. This work presents a novel evaluation system, that relies on three tags placed at known position on the robot. With this additional range information we determine the position of the anchors using trilateration and using this positional information in a SLAM based approach we estimate not only the position, but also the heading of the robot. This enables a fully autonomous system for both the distribution and usage of the UWB sensors for localization, which is carried by the robot itself.

II. THEORETICAL BACKGROUND

A. Ultra-Wideband Technology

UWB technology is a general classification for wireless communication schemes, that utilize a larger than usual bandwidth. While other traditional communication schemes, like Wi-Fi, Bluetooth or cellular networks try to reduce the bandwidth as much as possible, to create space in

¹Informatics XVII - Robotics, Julius-Maximilians-Universität Würzburg, Germany. Contact: martin.hesse@uni-wuerzburg.de

²Technical University of Applied Sciences Würzburg-Schweinfurt, Germany

³Informatics VIII - Aerospace Computer Science, Julius-Maximilians-Universität Würzburg, Germany

We acknowledge support from the Elite Network of Bavaria for the academic program Satellite Technology – Advanced Space Systems.

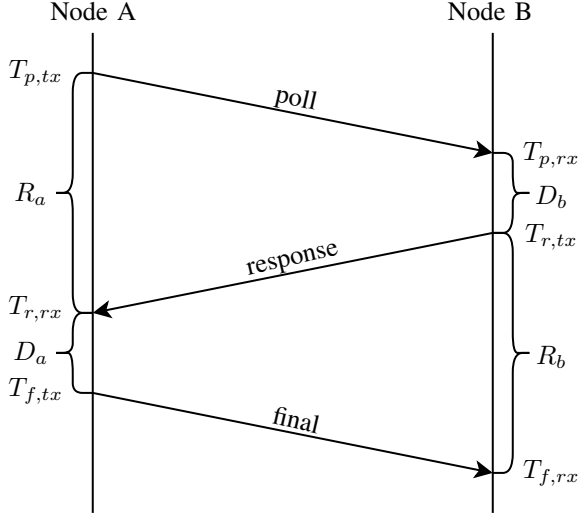


Fig. 2: Overview of the messages sent and their respective timestamps during one cycle of the TWR Scheme.

the frequency spectrum for as many unique channels as possible, the bandwidth of the transmitted signal with UWB communication schemes is generally at least 500 MHz. UWB signals are usually strictly limited to -41.3 dBm/MHz by the responsible government agencies [8]–[12]. This very low signal intensity means a UWB signal, that overlaps with another wireless communication signal, acts like regular background noise to it [9], [10]. This allows UWB technology to be used in many applications, although it is also responsible for the obvious disadvantage of the technology, the comparative short range of the signal, usually at most a couple of hundred meters [13]. But there are many advantages, that come with the unique design. Due to the low signal power, very little energy is needed for the transmission of UWB messages. This makes the system highly energy efficient and allows for flexible data rates, depending on the requirements of the exact use case [8]. [9] also highlights the simplified transceiver architecture, which also helps to keep the cost of UWB ICs low. Finally, [10], [14] note, that UWB signals have a greater ability to penetrate into obstacles avoid them entirely through multipath. While there are many use cases for UWB technology that use it for low energy short range data transmission, we are interested in its ranging capabilities. For this not the data transmitted, but rather the signal itself is used to determine the distance between two UWB nodes. For outdoor positioning GNSS has long been the obvious and widespread solution, but it faces significant problems in areas, where there is no GNSS signal, like indoors, caves or planetary exploration. UWB technology is ideally suited to fill this gap, by performing positioning with some fixed distributed anchors at known locations and a moving tag. [8], [10], [14]

B. Two-Way-Ranging (TWR)

TWR is one option to use the aforementioned UWB messages to determine the distance in between two UWB nodes. It uses the time of flight (TOF) of any wirelessly sent message, which is recorded using multiple timestamps. With

the speed of light c_0 the distance d follows as

$$d = T_f c_0. \quad (1)$$

To determine the TOF T_f using the UWB messages there are multiple options. To minimize error due to clock drift we use the Asymmetric Double Sided Two-Way-Ranging (ADS-TWR) method as presented in [15]. There are three UWB messages associated with one cycle of the ADS-TWR method, which alongside their respective timestamps are depicted in Fig. 2. We call the time it takes for a node to respond to a received message reply time $D_b = T_{r,tx} - T_{p,rx}$ and $D_a = T_{f,tx} - T_{r,rx}$, respectively. The time it takes to receive a response after a message is sent out is known as the round time $R_a = T_{r,rx} - T_{p,tx}$ and $R_b = T_{f,rx} - T_{r,tx}$. Using these we calculate the TOF T_f [15]:

$$T_f = \frac{R_a R_b - D_a D_b}{R_a + R_b + D_a + D_b} \quad (2)$$

The corresponding error due to clock drift ΔT_f is

$$\Delta T_f = \frac{T_f}{2} (e_a + e_b). \quad (3)$$

From (3) it is obvious, that the error due to clock drift is not an issue, because the error is at most a small fraction of the TOF, as the clock drift of the Nodes A e_a and B e_b are below ± 20 ppm [16]. Therefore, we have an evaluation method in (2), that is robust against clock drift, which forms the basis for our UWB localization system, by providing accurate distances between the nodes. As it is necessary to know all timestamps to calculate the distance, the timestamps of Node A are sent along the final message, from which Node B calculates the distance between the nodes, using (2) and (1).

C. Kalman Filter

For the SLAM algorithm, that we use in our UWB localization approach, an EKF with non-additive process noise and the potential for missing observations is needed. The Kalman Filter consists of a state $\mathbf{x}$ of dimension n and a corresponding covariance matrix $\mathbf{P}$ of dimension $n \times n$. The state at a discrete time step $\mathbf{x}_k$ and the observation $\mathbf{z}_k$, with dimension m , is dependent on the previous state $\mathbf{x}_{k-1}$, the system input $\mathbf{u}_k$ and the process noise $\mathbf{w}_k$ and observation noise $\mathbf{v}_k$, with their respective covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$. These relations are governed by the differentiable functions $\mathbf{f}$ and $\mathbf{h}$:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (4)$$

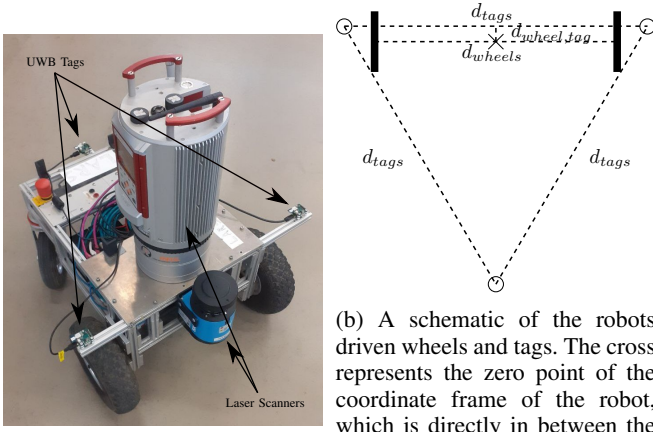
$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (5)$$

The Kalman Filter is split into a prediction step and an update step.

Prediction:

$$\text{predicted state: } \hat{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) \quad (6)$$

$$\text{predicted covariance: } \hat{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^\top + \mathbf{L}_k \mathbf{Q}_k \mathbf{L}_k^\top \quad (7)$$



(a) A picture of the robot, with the laser scanners and tags pointed out.

Fig. 3: The robot (a) used as a platform to test the UWB localization system, featuring three fixed UWB tags (b).

Update:

$$\text{Innovation: } \mathbf{y}_k = \mathbf{z}_k - \mathbf{G}_k \mathbf{h}(\hat{\mathbf{x}}_k) \quad (8)$$

$$\text{Innovation covariance: } \mathbf{S}_k = \mathbf{H}_k^* \hat{\mathbf{P}}_k \mathbf{H}_k^{*\top} + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^\top \quad (9)$$

$$\text{Kalman Gain: } \mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{H}_k^{*\top} \mathbf{S}_k^{-1} \quad (10)$$

$$\text{updated state: } \mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k \mathbf{y}_k \quad (11)$$

$$\text{updated covariance: } \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k^*) \hat{\mathbf{P}}_k \quad (12)$$

The state transition matrix $\mathbf{F}_k$ and observation matrix $\mathbf{H}_k$ are defined by the derivatives of their respective functions $\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}_{k-1}, \mathbf{u}_k}$ and $\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}_k}$. $\mathbf{L}_k = \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \Big|_{\mathbf{x}_{k-1}, \mathbf{u}_k}$ is needed as the process noise $\mathbf{w}_k$ is non-additive. $\mathbf{H}_k^* = \mathbf{G}_k \mathbf{H}_k$ is a shorthand, that uses the matrix $\mathbf{G}_k$, which represents the observations, that actually exist at time step k . We define m_k to be the actual number of new observation at time step k . The observations that are available at time step k , are at the indices $i_1, i_2, \dots, i_{m_k}$ of $\mathbf{z}_k$. Matrix $\mathbf{G}_k$ is of dimension $m_k \times m$, with ones at the entries $(1, i_1), (2, i_2), \dots, (m_k, i_{m_k})$ and zeros everywhere else. After the prediction step of the EKF, we determine m_k , set up $\mathbf{G}_k$ accordingly and then perform the update step. In the case that m_k is 0, the update step will not change the prediction at all, meaning it is skipped entirely. [17]–[19]

III. APPROACH

A. Mobile Robot

We use a four-wheeled mobile robot based on a Volksbot RT3 [20] as a platform to test the UWB localization. Two 150 W Maxon DC motors, with digital encoders, that provide wheel odometry, with a 74 : 1 planetary gearbox drive the two front wheels spaced $d_{wheels} = 44$ cm apart. Three custom UWB PCBs are mounted in an equilateral triangle with side length $d_{tags} = 60$ cm to the frame of the robot. Two of the tags are mounted in the front parallel to the wheel axis, with a distance of $d_{wheel,tag} = 1.6$ cm (cf. Fig. 3b). The



Fig. 4: The custom PCB with the UWB chip. Left: UWB PCB mounted in a stack with a USB communication board on the robot. Middle: The front of the PCB. Right: The back of the PCB.

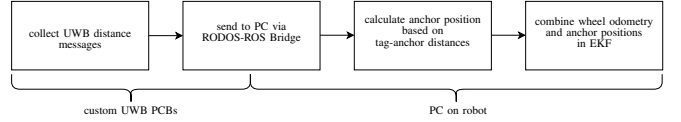


Fig. 5: Flowchart of the individual steps used in our UWB localization system.

custom PCB was originally designed as a stackable PCB for a small Drone, that uses among others UWB transceivers for positioning [21]. The boards use the DecaWave (now Qorvo) DWM1000 UWB transceiver [13] integrated with a STM32F407 microprocessor [22], which runs the embedded code to perform and evaluate the TWR events. The anchors, spread out in the surroundings, consist of only this board, which is powered using a 9 V block battery. The tags have a second board stacked on the UWB board, which has a USB connection to the PC on the robot through which the stack is powered (cf. Fig. 4). A SICK LMS100-10000 [23] laser scanner is mounted on the front of the robot. This 2D laser scanner provides us with quasi continuous coverage of the plane parallel to and ca. 30 cm above the ground. Mounted on top of the robot is a RIEGL VZ-400 Laser Scanner [24]. This laser scanner provides non-continuous high definition 3D laser scans. Both laser scanners are used to create a ground truth, to compare the UWB localization system to.

B. UWB Localization

Fig. 5 gives an overview of the localization process, which consists of four parts. First the distance measuring using the UWB nodes, the data of which is then transferred to the PC on the robot, and finally two evaluation steps on the robot.

1) *Embedded Software on the UWB nodes*: The UWB nodes repeatedly perform TWR measurements. To increase the update rate we implement concurrent TWR, in which the distance from one UWB node to all others is determined in a single TWR process using staggered responses from the passive nodes, as described in [25]. After one node has performed a TWR measurement it activates the next node, so that all nodes perform TWR measurements cyclically (cf. Algorithm 1), thus creating a full mesh of distance measurements between all nodes.

2) *RODOS-ROS-Bridge*: The distance data gathered by the individual UWB nodes is sent to a tag on the robot, from where it is transferred to the PC using a RODOS-ROS-Bridge for further evaluation.

Algorithm 1: The general principle of our implementation of the TWR scheme.

```

i ← 0
while true do
  node i sends poll msg
  forall j ∈ {0, 1, ..., n − 1} \ i do
    | node j sends response msg after j × 5 ms
  end
  node i waits n × 5 ms after it sent poll msg
  node i populates and sends final msg
  forall j ∈ {0, 1, ..., n − 1} \ i do
    | node j calculates distance
    if node j is tag then
      | node j publishes distance to the
      |   RODOS-ROS-Bridge
    else
      | node j sends distance via distance msg
      |   to node 0 after j × 5 ms
    end
  end
  node 0 publishes all distance values received to
  | the RODOS-ROS-Bridge
  node i waits n × 5 ms after it sent final msg
  node i sends activate msg to node
  | ((i + 1) mod n)
  i ← ((i + 1) mod n)
end

```

3) *Anchor Position Determination:* As the global position of the anchors are unknown we use the distance measurements $d_{t,a}$ from each of the tags on the robot, at positions $(x_t|y_t)$ relative to the robot, to each of the anchors to determine their respective position $(x_a|y_a)$ relative to the robot, by solving the following minimization problem:

$$\forall a \in \text{anchors} : \min_{x_a, y_a} \sum_{\forall t \in \text{tags}} ((x_a - x_t)^2 + (y_a - y_t)^2 - d_{t,a}^2)^2 \quad (13)$$

The minimization is solved using the minimize function of scipy. The resulting positions of the anchors are then interpreted as landmarks in the following step.

4) *EKF-SLAM:* The EKF we use is inspired by an EKF introduced in [7]. We consider a system with n_A anchors in use, in which the state $\mathbf{x}$ of the EKF has $3 + 2n$ entries:

$$\mathbf{x} = [x_R, y_R, \theta, x_{A,1}, y_{A,1}, \dots, x_{A,n_A}, y_{A,n_A}] \quad (14)$$

x_R and y_R represent the position of the robot, θ the orientation and $x_{A,i}$ and $y_{A,i}$ the position of the i th anchor. All of these coordinates are in the fixed global frame. Using the kinematics of the robot we define $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ as:

$$\mathbf{f} = [x_{R,k-1} + v_k \cos \theta_k, y_{R,k-1} + v_k \sin \theta_k, \theta_k + \omega_k, x_{A,1}, y_{A,1}, \dots, x_{A,n_A}, y_{A,n_A}] \quad (15)$$

with $v_k = \frac{v_{r,k} + v_{l,k}}{2}$ and $\omega_k = \frac{v_{r,k} - v_{l,k}}{d_{wheels}}$, where $v_{r,k}$ and $v_{l,k}$ are the distances the wheels moved forward during the last

iteration of the EKF, which takes 0.1 s. For the covariance $\mathbf{Q}_k$ we use $\begin{bmatrix} K_r |v_{r,k}| & 0 \\ 0 & K_l |v_{l,k}| \end{bmatrix}$ with $K_r = K_l = 0.01$. The observations are the previously calculated positions of the anchors in the robots coordinate frame transformed into the global frame and the $\binom{n_A}{2}$ distance measurements in between the anchors. $\mathbf{R}_k$ is $\begin{bmatrix} \sigma_{anPos}^2 \mathbf{I}_{2n_A \times 2n_A} & \mathbf{0}_{2n_A \times \binom{n_A}{2}} \\ \mathbf{0}_{\binom{n_A}{2} \times 2n_A} & \sigma_{dis}^2 \mathbf{I}_{\binom{n_A}{2} \times \binom{n_A}{2}} \end{bmatrix}$ with the variances σ_{anPos}^2 and σ_{dis}^2 , for which multiple values are tested in each experiment to tune the EKF.

For the initial setup of the EKF we set the position and orientation of the robot to 0, and we use an average over ten determined anchor position values as their initial position.

C. Ground Truth

For the ground truth we use two systems. The data from the 2D SICK LMS100 laser scanner is used in the Hector SLAM algorithm, that provides us with a continuous pose of the robot [26]. As all experiments are done indoors, this works well, as compared to a low feature outdoor scenario. Over the course of one experiment we take multiple high definition laser scans with the Riegl VZ-400. From the scans we determine the positions of the anchors and using an Iterative Closest Point (ICP) algorithm implemented in [27], that maps the scans onto each other, we determine the position of the robot at the scan locations.

IV. EXPERIMENTS AND EVALUATION

To validate the UWB localization system we perform six test runs. In each one we use four anchors randomly spread out in the robots surroundings, and drive an arbitrary path with the robot. The anchors are placed on a tripod 66cm above the ground. We perform one laser scan with the Riegl VZ-400 at the beginning and the end of each test run. In the 4., 5. and 6. test run we perform two intermediate scans, splitting the run in three sections. During the middle section of these test runs we disable all UWB nodes, to test the ability of the system to work without UWB measurements for a short period and how well it corrects itself after they are reactivated. Each test run is evaluated using multiple values for the parameters σ_{anPos}^2 and σ_{dis}^2 of the EKF. To quantify the performance we use the Root Mean Square Error (RMSE) of the trajectory compared to that of the Hector-SLAM, listed in Table I. The plot of the trajectories in Fig. 6 and Fig. 7 show key findings of certain test cases.

In all test runs the wheel odometry drifts away from the ground truth over time, due to cumulative error. This effect grows stronger, the longer the test run is. The EKF generally performs better for lower values for σ_{anPos}^2 , as they put a larger emphasis on the UWB measurements and therefore less on the wheel odometry. This is especially true after an outage of the UWB measurements, so in Section 3 of Test Runs 4 and 5. Fig. 6 shows, that the trajectories for lower σ_{anPos}^2 , though on average better, are less smooth, as the UWB measurements are less consistent. They also face a greater risk to large errors, if the initial anchor position

TABLE I: Numerical results of the full localization tests. The RMSE of the wheel Odometry and the EKF is calculated using the trajectory of the Hector-SLAM algorithm as the ground truth. The error of the Hector-SLAM algorithm is the distance to the point, where the Riegl scan is taken. Dashes (-) represent test cases, that were not analyzed, and red or green highlighted cells represent test cases, that performed particularly poor or good and are closer discussed in the text.

Test # - Section #	Runtime [s]	RMSE Wheel Odometry [m]	RMSE EKF [m], without inter-anchor measurements, varying σ_{anPos}^2				RMSE EKF [m], $\sigma_{anPos}^2 = 1$, varying σ_{dis}^2				Error Hector- SLAM [m]
			0.0001	0.01	1	100	0.001	0.005	0.05	0.5	
1	65.5	0.19	0.26	0.14	0.14	0.09	-	-	-	-	0.096
2	75.9	1.08	-	0.57	0.51	0.95	0.44	0.36	0.44	0.49	0.078
3	107.9	1.59	0.38	0.31	0.43	0.73	-	0.43	0.43	0.43	0.215
4 - 1	61.2	0.15	0.42	0.31	0.33	0.11	0.26	0.29	0.31	0.29	0.174
4 - 2	29.8	0.89	1.06	1.17	1.30	1.11	1.26	1.30	1.25	1.24	0.042
4 - 3	49.9	1.88	0.54	0.58	1.07	1.45	0.88	0.95	0.98	1.00	0.199
4 - total	140.9	1.20	0.65	0.67	0.90	1.01	0.80	0.85	0.84	0.85	-
5 - 1	44.1	0.88	0.25	0.23	0.08	0.43	0.25	0.22	0.23	0.11	0.057
5 - 2	48.2	3.13	0.72	1.15	0.86	1.22	1.58	1.41	1.39	1.01	0.158
5 - 3	41.4	3.78	1.18	1.14	1.28	1.32	1.39	1.43	1.41	1.33	0.079
5 - total	133.7	2.87	0.81	0.95	0.90	1.07	1.24	1.17	1.15	0.97	-
6 - 1	60.3	0.81	2.59	2.04	0.61	1.00	0.37	0.41	0.41	0.42	0.205
6 - 2	62.3	2.88	4.40	4.61	1.68	2.79	1.24	1.25	1.21	1.21	0.219
6 - 3	46.5	4.42	2.06	2.49	0.91	1.24	1.45	1.83	1.48	1.34	0.216
6 - total	169.1	2.94	3.27	3.32	1.18	1.91	1.08	1.25	1.09	1.04	-
average		1.65	1.07	0.99	0.68	0.96	0.89	0.81	0.79	0.76	0.143
average without Test 6		1.39	0.53	0.53	0.58	0.77	0.83	0.70	0.72	0.69	0.13

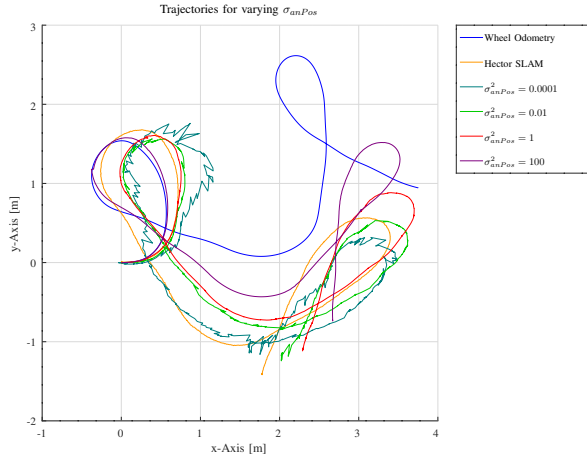


Fig. 6: Trajectories for varying σ_{anPos}^2 in Test Run 3, with no inter-anchor distance measurements.

determination process works particularly poorly. This is what happened during Test Run 6 in the initial setup, which is why we list the average without Test Run 6 in Table I separately. The reason for this particularly bad outlier in the initial anchor position determination process is still unclear to us. Fig. 7 shows the effect of a temporal outage of the UWB measurements. During Section 1 the EKF and wheel odometry still track the ground truth well, but in Section 2, during which there are no UWB measurements, the EKF drifts away along with the wheel odometry. The covariance of the EKF also increases during Section 2. At the beginning of Section 3 the EKF immediately corrects itself but not enough. The correction is larger for smaller values of σ_{anPos}^2 , which perform particularly well in Section 3 of Test Run

4 and 5. Over the course of the section the covariance decreases to a new equilibrium and the overall trajectory is closer to the ground truth, than the wheel odometry. Overall $\sigma_{anPos}^2 = 1$ is a good trade off, as low as possible, but still large enough, so that the trajectory remains smooth and is not massively affected by a poor anchor position determination. The inclusion of the inter-anchor distance measurements to the system does not have a large effect on the result, no matter what value for the parameter σ_{dis}^2 is used.

V. CONCLUSION

We have introduced and implemented a novel way to combine UWB ranging measurements with an EKF-SLAM algorithm to perform localization of a robot. Using three tags mounted on the robot we are able to determine the position of randomly distributed anchors, with which our localization approach outperforms wheel odometry considerably and consistently. Also, it is capable to continue functioning in the case of an UWB node outage and self corrects after such an outage. We have seen, that generally putting more emphasis on the UWB measurements improves the results further, though the resulting trajectories are less smooth and face greater risks to outliers. Needless to say, a lot of work remains to be done. Further improvements to the embedded UWB system are to be made, to improve the reliability and precision of the distance measurements. Also, instead of using the inter-anchor measurements as an observation of the EKF, where they do not have an impact, using them to determine the anchor positions together instead of independently of one another has to be studied. Furthermore, we have to examine the impact of locating the anchors inside plastic rockets on the ground rather than on a tripod. In future work, we will use the UWB nodes to perform localization in

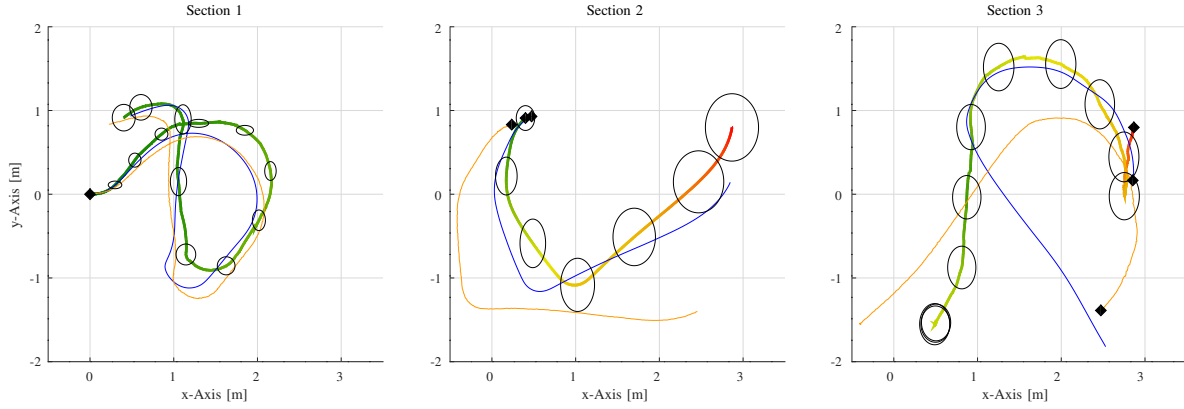


Fig. 7: The output trajectory of the EKF of Test Run 4 for $\sigma_{anPos}^2 = 1$ and $\sigma_{dis}^2 = 0.005$, where the color represents the distance error to the trajectory of the Hector-SLAM algorithm. The ellipses represent the 3σ range in the x- and y-axis, based on the covariance matrix of the EKF. The orange line represents the output of the Hector-SLAM and the blue line the output of the wheel odometry. The black diamonds (◆) represents the starting point of each section.

3D with 6DoF instead of 2D/3DoF. Also, a larger network of anchors and potentially multiple robots, will increase the overall system size, which will require a system design adaptation, but has the potential for higher accuracy.

REFERENCES

- [1] A. Nüchter, L. Werner, M. Hesse, D. Borrmann, T. Walter, S. Montenegro, and G. Grömer, "UWB Anchor Based Localization of a Planetary Rover," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS '24)*, Brisbane, Australia, November 2024. [Online]. Available: <https://robotik.informatik.uni-wuerzburg.de/telematics/download/isairas2024.pdf>
- [2] ESA, "ESA-ESRIC Space Resources Challenge," <https://www.spaceresourceschallenge.esa.int/home>, 2021, accessed: 24.10.2024.
- [3] ÖWF, "AMADEE-24 Mars simulation," <https://oewf.org/amadee-24/>, 2024, accessed: 24.10.2024.
- [4] S. Gezici and H. V. Poor, "Position estimation via Ultra-Wideband signals," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 386–403, 2009.
- [5] N. Zhou, L. Lau, R. Bai, and T. Moore, "Novel prior position determination approaches in particle filter for Ultra-Wideband-based indoor positioning," *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 277–292, 2021. [Online]. Available: <https://navi.ion.org/content/68/2/277>
- [6] H. Soganci, S. Gezici, and H. V. Poor, "Accurate positioning in Ultra-Wideband systems," *IEEE Wireless Communications*, vol. 18, no. 2, pp. 19–27, 2011.
- [7] F. Martinelli and F. Romanelli, "A slam algorithm based on range and bearing estimation of passive uhf-rfid tags," in *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, 2021, pp. 20–23.
- [8] M. Qian, K. Zhao, B. Li, and A. Seneviratne, "An overview of Ultra-Wideband technology and performance analysis of UWB-TWR in simulation and real environment," *IPIN-WiP*, 2022.
- [9] T. Tsang and M. El-Gamal, "Ultra-Wideband (UWB) communications systems: an overview," in *The 3rd International IEEE-NEWCAS Conference, 2005.*, 2005, pp. 381–386.
- [10] W. Hirt, "Ultra-Wideband radio technology: overview and future research," *Computer Communications*, vol. 26, no. 1, pp. 46–52, 2003.
- [11] Bundesnetzagentur, "Allgemeinzuteilung von Frequenzen für die Nutzung durch Ultrabreitbandgeräte (UWB) Vfg. 135 / 2019." 2019, accessed: 30.10.2024.
- [12] Federal Communications Commission, "47 CFR § 15.519 - Technical requirements for hand held UWB systems." <https://www.govinfo.gov/content/pkg/CFR-2010-title47-vol1/pdf/CFR-2010-title47-vol1-sec15-517.pdf>, 2002, accessed: 30.10.2024.
- [13] Qorvo, Inc, "DWM1000," <https://www.qorvo.com/products/p/DWM1000>, 2024, accessed: 05.11.2024.
- [14] M. Z. Win, D. Dardari, A. F. Molisch, W. Wiesbeck, and W. Jinyun Zhang, "History and applications of UWB." Institute of Electrical and Electronics Engineers, 2009.
- [15] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016, pp. 1–4.
- [16] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans): Amendment 1: Add alternate phys," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1–210, 2007.
- [17] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, no. 46, pp. 3736–3741, 2004.
- [18] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [19] T. Cipra and R. Romera, "Kalman filter with outliers and missing observations," *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, vol. 6, pp. 379–395, 02 1997.
- [20] A. Nüchter, J. Elseberg, and D. Borrmann, "Irma3D – An Intelligent Robot for Mapping Applications," in *Proceedings of the 3rd IFAC Symposium on Telematics Applications (TA '13)*, Seoul, Korea, November 2013, pp. 119–124. [Online]. Available: <https://robotik.informatik.uni-wuerzburg.de/telematics/download/ta2013.pdf>
- [21] M. Strohmeier, T. Walter, J. Rothe, and S. Montenegro, "Ultra-Wideband based pose estimation for small unmanned aerial vehicles," *IEEE Access*, vol. 6, pp. 57 526–57 535, 2018.
- [22] STMicroelectronics, "STM32F407/417," <https://www.st.com/en/microcontrollers-microprocessors/stm32f407-417.html>, 2024, accessed: 05.11.2024.
- [23] SICK AG, "SICK LMS100-10000," <https://www.sick.com/be/de/p/produkte/lidar-und-radarsensoren/lidar-sensoren/lms1xx/lms100-10000/p/p109841>, 2024, accessed: 04.11.2024.
- [24] RIEGL Laser Measurement Systems GmbH, "RIEGL VZ-400," 2017, accessed: 24.10.2024.
- [25] Decawave, *DW1000 User Manual*, 2017.
- [26] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 155–160.
- [27] Automation Group (Jacobs University Bremen) and Knowledge-Based Systems Group (University of Osnabrück), "3DTK – The 3D Toolkit," <http://slam6d.sourceforge.net/>, 2011, accessed: 24.10.2024.