Outdoor Multi-Sensor Navigation of an Unmanned Ground Vehicle

David Dias
david.miguel.dias@tecnico.ulisboa.pt
Instituto Superior Técnico

***Abstract*—Since the development of the Global Positioning System (GPS), the outdoor localization problem was significantly overcome due to the availability of outdoor position estimates. Yet, localization is still a difficult problem under certain constraints. Currently, in several areas such as cities, the urban canyon often limits the number of satellites in view used to obtain a position estimate. In these cases, the GPS receivers cannot obtain enough satellites signals to compute an estimate. In these conditions, it is often necessary to rely on dead reckoning methods to update the position estimate. Still, corrections to these estimates need to be performed as its uncertainty increases. In this thesis, we present localization methods for an Unmanned Ground Vehicle (UGV). The developed methods intend to decrease the resilience of GPS measurements by relying on different configurations of wheel odometry and two Inertial Measurement Units (IMUs) to update the pose estimate. A set of indoor and outdoor experiments were performed to assess the accuracy of the proposed methods when comparing these to an estimate obtained using GPS measurements. Results showed that our methods do provide a continuously reliable pose estimate over short distances.**

***Index Terms*—Localization, Odometry calibration, Sensor modeling, Sensor fusion, Outdoor unmanned ground vehicle**

I. Introduction

Unstructured environments and their unpredictability pose a serious challenge for robots and humans, not only not knowing what does the environment look like, but also take into account the unpredictable behaviors that bodies might take within it. For the safe and efficient deployment of UGVs, efficient methods are required to gather as much useful information from the environment as possible.

The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) is a robotics competition which started in 2017. It focuses on enabling technologies for applications in several areas, these include, disaster response, domestic tasks, transport, and construction, with the aim of promoting "robots working autonomously in dynamic, unstructured environments, while collaborating and interacting with other robots" [1].

Often when performing tasks outdoor GPS localization estimates are used, yet these are not available everywhere. Although the proposed challenges will be held mostly outdoors, where GPS signal should be available, its use is discouraged. The Institute for Systems and Robotics - Lisbon from Instituto Superior Técnico, along with the Robotics, Vision and Control Group from the University of Seville applied to jointly participate in MBZIRC 2020, and were accepted as a participating team.

Localization in outdoor environments is still a challenging task under some constraints [2]. We modify a UGV for outdoor use and explore localization techniques, avoiding the use of GPS assisted methods. Such methods are evaluated by performing real-world experiments in an outdoor environment. The sensor suite already implemented on the UGV was studied and upgraded. The selected sensors are investigated and its noise modeled.

The implemented methods do not rely on GPS measurements. However, an estimate using GPS measurements is computed in order to obtain a pose reference to which our methods are compared to.

II. Related work

Odometry estimates are one of the simplest methods applied to localization. Odometry methods use an approximate model of the vehicle and information provided by the wheel encoders. Using these and the model of the vehicle, the measurements are integrated providing a continuous position estimate.

Two types of errors affect odometry estimation, systematic and nonsystematic errors [3]. Systematic errors are the ones which occur constantly, nonsystematic errors occur randomly. Unequal wheel diameters, limited encoder resolution, and sampling rate are some examples of systematic errors in this application. Slippery floors and skidding are some examples of non-systematic errors. Due to the non-aleatory nature of systematic errors, these can be removed from the estimate by tuning the vehicle model.

One of the most widely known methods for odometry calibration is UMBmark [3]. This method was proposed as a simple experiment from which the parameters of a differential drive robot could be calibrated. The method involves several steps and the execution of a predetermined path, from which, the starting and ending position are measured. With those, the parameters which calibrate the model are computed.

In [4] a different approach is taken by directly measuring the wheels velocities of a differential drive robot while its wheels were rotating without any load. Using a tachometer the real wheels velocities are measured the relation between these and the estimate provided by the wheel encoders is used to calibrate both wheels velocities. With measurements from the robot's attitude while skid steering, the coefficient between the effective axle length is estimated.

The method presented in [5] relies on least-squares over a non-defined trajectory to compute a odometry calibration matrix. Using an estimation of the path, provided by the UGV's odometry, and a ground truth the calibration matrix is computed from the estimate and ground truth 2D homogeneous coordinates difference in the robot frame. This matrix minimizes the sum of the squares of the residuals of the 2D homogeneous coordinates difference in the robot frame, approximating this way the estimated path to the ground truth. This method was preferred due the availability of a ground truth provided by a Motion Capture (MOCAP) system and the possibility of a wide range of motions being used in the calibration.

There are two types of localization, relative and absolute [6]. Dead reckoning is a relative localization method, which relies on a given initial pose of the agent and on information about how it moved to update its belief. These provide a good estimate on a per step basis, but when integrating the velocities over a larger time interval and distance the estimation error grows without bound [7]. In order to correct drift in dead reckoning, measurements of the actual pose of the agent are required. These can be in relation to a map, in case of map matching, or in relation to a global frame such as the ones obtained when using GPS.

Multiple methods combine measurements from different sources to obtain a better estimate. Example of these are particle filter, Kalman filter, Extended Kalman Filter (EKF) and unscented Kalman filter [8].

Localization can be thought as a state estimation problem where the state to be estimated contains the localization of the agent. EKFs have been extensively used in localization [9], they are efficient state estimators for nonlinear problems.

In [9], four different estimation architectures which rely on an EKF are proposed and evaluated for six Degrees Of Freedom (DOF) pose estimation of a skid steer mobile robot. The author performs a series of experiments in different scenarios and determines that the second approach yields better results. We used the first architecture, this is simpler than the estimate which yields better results, yet it performed similarly in terms of return position and attitude error.

III. Background

A. Odometry calibration method

Considering the 2D ground truth pose $\mathbf{u}_t$ and the odometry estimated pose $\mathbf{z}_t$, both at time *t*, we have:

$$\begin{aligned}\mathbf{u}_t &= [x_t, y_t, \theta_t]^T \\ \mathbf{z}_t &= [\hat{x}_t, \hat{y}_t, \hat{\theta}_t]^T\end{aligned} \tag{1}$$

Using these, we compute the 2D homogeneous transformation vectors between $t-1$ and t for both of our sequences of measurements, $\mathbf{u}_{0:N-1}$ and $\mathbf{z}_{0:N-1}$. This is done by computing the 2D homogeneous transformation matrix for t, A_t, then computing the homogeneous transform between $t-1$ and t, B_t, and finally obtaining the 2D homogeneous coordinates, $\mathbf{s}_t$, which correspond to the 2D homogeneous coordinates difference in the robot frame. These series of transformations are depicted for $\mathbf{u}_t$ in the sequence of operations in (2).

$$A_t = \begin{bmatrix} cos(\theta_t) & -sin(\theta_t) & x_t \\ sin(\theta_t) & cos(\theta_t) & y_t \\ 0 & 0 & 1 \end{bmatrix} \tag{2a}$$

$$B_t = A_t A_{t-1}^{-1} \tag{2b}$$

$$\mathbf{s}_t = [B_{0,2_t}, B_{1,2_t}, arctan2(B_{1,0_t}, B_{0,0_t})]^T \tag{2c}$$

With these, two matrices are obtained, U and Z, which are matrices of dimension $3 \times N$, being each one the concatenation of the robot frame velocities for $\mathbf{u}_t$ and $\mathbf{z}_t$, respectively. This is shown for $\mathbf{u}$ in (3).

$$U = \begin{bmatrix} s_{0_0} & \dots & s_{0_t} & \dots & s_{0_{N-1}} \\ s_{1_0} & \dots & s_{1_t} & \dots & s_{1_{N-1}} \\ s_{2_0} & \dots & s_{2_t} & \dots & s_{2_{N-1}} \end{bmatrix} \tag{3}$$

The problem was then formulated as a linear system of equations as in (4), where X is the 3×3 calibration matrix.

$$U = XZ \tag{4}$$

The calibration matrix X was then computed by applying least-squares minimization as in (5), where N is the number of 2D poses used in the minimization problem.

$$\min_X \sum_{t=0}^{N-1} (XZ_{:,t} - U_{:,t}) \tag{5}$$

We then calibrate the 2D homogeneous coordinates difference in the robot frame (6).

$$\tilde{Z} = XZ \tag{6}$$

Taking $\tilde{Z}$, we compute the calibrated robot frame velocities $\tilde{v}$ and $\tilde{\omega}$ by differentiating (7).

$$\tilde{L} = \frac{\tilde{Z}}{\delta} \tag{7}$$

Where δ represents the measurement's acquisition period and the first row of $\tilde{L}$ is the calibrated linear velocities, $\tilde{v}$, and the third row contains the calibrated angular velocities, $\tilde{\omega}$. The calibrated trajectory is computed recursively from $\tilde{Z}$ as in (8).

$$C_t = \begin{bmatrix} cos(\tilde{Z}_{1,t}) & -sin(\tilde{Z}_{1,t}) & \tilde{Z}_{0,t} \\ sin(\tilde{Z}_{1,t}) & cos(\tilde{Z}_{1,t}) & \tilde{Z}_{1,t} \\ 0 & 0 & 1 \end{bmatrix} \tag{8a}$$

$$P_t = P_{t-1} C_t \tag{8b}$$

$$\tilde{\mathbf{z}}_t = [P_{0,2_t}, P_{1,2_t}, arctan2(P_{1,0_t}, P_{0,0_t})]^T \tag{8c}$$

Where the vector $\tilde{\mathbf{z}}_t$ is the calibrated 2D pose at the time instant t.

B. Noise models

1) Additive white Gaussian noise model: One of the most commonly used models is the additive white Gaussian noise model, which models the noise as an additive element which is represented as Gaussian white noise, as shown in (9).

$$\begin{aligned} r(t) &= s(t) + w(t) \\ w(t) &\sim N(0, \sigma^2) \end{aligned} \tag{9}$$

Where $r(t)$ represents our measurement, $s(t)$ is the true value and $w(t)$ the white Gaussian noise with 0 mean and variance σ^2.

2) Additive white Gaussian noise model with proportional variance: In some cases, the white Gaussian noise variance tends to increase with the absolute value of the measurement's true value. For these cases, a model where the variance is proportional to the estimated measurement's absolute true value is used. The model is depicted in (10).

$$\begin{aligned} r(t) &= s(t) + w(t) \\ w(t) &\sim N(0, \sigma^2(t)) \\ \sigma^2(t) &= m \cdot |\hat{r}(t)| + b \end{aligned} \tag{10}$$

Where $r\hat{(t)}$ represents the true value's estimate, m is the slope of the linear regression and b the variance when $|\hat{r}(t)| = 0$.

IV. Methodology

A. Odometry velocities calibration

We performed indoor experiments in which the UGV was driven randomly at varying speeds in order to collect data encompassing the full range of possible movements. Both the UGV's odometry and the MOCAP system's pose measurements were recorded.

The marker array was attached to the UGV by aligning it with the UGV's frame and placed directly above it.

B. Sensor models

1) Calibrated odometry velocities noise model: Some noise sources add to the noise an element which depends on the velocities' absolute estimate value, this relation can be modeled linearly in the velocities' absolute true value. We observed that these elements are large enough to dominate the overall error behavior. Considering this, we model both of our calibrated odometry's velocities, v, and ω, with the additive white Gaussian noise model with proportional variance.

2) Crossbow IMU noise model: The Crossbow DMU-6X-003 IMU measures angular velocity and linear acceleration. Both quantities were modeled using the additive white Gaussian noise model. Due to the lack of a datasheet for this specific model, the variances were computed assuming that they are the same at rest and while moving. Crossbow IMU's data was recorded for two minutes while at rest and the measurements' variance estimated.

In order to better model the real value, the mean of the white Gaussian noise was not considered 0, but equal to the estimated bias.

3) MPU IMU noise model: The MPU-6050 IMU measures angular velocity and linear acceleration. With these measurements, it estimates the IMU's attitude relative to an inertial frame. These were modeled after the additive white Gaussian noise model.

The orientation's variance about the IMU's Z axis was estimated using data collected during indoor experiments. Considering the MOCAP system measurements, the variance of the IMU's attitude estimation is computed from the error between the two.

For the angular velocity and linear acceleration, we compute the variance using parameters provided in the IMU's datasheet.

4) NovAtel GPS receiver noise model: GPS measurements, in general, contain several sources of noise. These include ephemeris errors, tropospheric and ionospheric delays, multipath signals, etc. The three measurements' noise was modeled using the additive white Gaussian noise model but with a varying variance.

In the message sent by the GPS receiver, along with the measurements, an estimate of the standard deviation for the latitude, longitude, and height is provided. The variance for the three different quantities is computed using this value.

C. Extended Kalman Filter

We use an already available EKF implementation [10]. The estimated state, $\mathbf{x}_t$, is defined with dimension 15×1. This encompasses the following fields, 3D pose (position and orientation), 3D velocities (linear and angular) and linear accelerations. The 3D pose is defined in the world frame while the velocities and accelerations are defined in the robot frame.

$$\begin{aligned} \mathbf{x}_t = [&\underbrace{x_w, y_w, z_w, \gamma_w, \beta_w, \alpha_w}_{\text{World frame}}, \\ &\underbrace{v_{x_r}, v_{y_r}, v_{z_r}, \omega_{\gamma_r}, \omega_{\beta_r}, \omega_{\alpha_r}, \dot{v}_{x_r}, \dot{v}_{y_r}, \dot{v}_{z_r}}_{\text{Robot frame}}]^T \end{aligned} \tag{11}$$

The next state prediction $(\overline{\mu}_t, \overline{\Sigma}_t)$ is not a function of the input u_t and the previous state, $(\mu_{t-1}, \Sigma_{t-1})$, but only of the previous state. The state estimate covariance matrix Σ_t is computed using the Joseph covariance equation to promote filter stability [11]. Since the time between each filter cycle is not constant, the matrix R is scaled by this factor, here represented by δ_t, this results in higher prediction uncertainty for cycles in which the previous happened longer ago.

Algorithm 1 presents the considered EKF implementation for this problem.

Algorithm 1 Extended Kalman Filter as implemented in robot_localization package.

```
Algorithm Extended Kalman Filter($\mu_{t-1}$, $\Sigma_{t-1}$, $z_t$)
$\overline{\mu}_t = g(\mu_{t-1})$
$\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R\delta_t$
$K_t = \overline{\Sigma}_t H_t^T (H_t \overline{\Sigma}_t H_t^T + Q_t)^{-1}$
$\mu_t = \overline{\mu}_t + K_t(z_t - H_t \overline{\mu}_t)$
$\Sigma_t = (I - K_t H_t) \overline{\Sigma}_t (I - K_t H_t)^T + K_t Q_t K_t^T$
return $\mu_t, \Sigma_t$
```

The function $g(.)$ used in the prediction step is a "standard 3D kinematic model derived from Newtonian mechanics" [10], the model is described in (12).

$$\begin{bmatrix} x_{w_t} \\ y_{w_t} \\ z_{w_t} \end{bmatrix} = \begin{bmatrix} x_{w_{t-1}} \\ y_{w_{t-1}} \\ z_{w_{t-1}} \end{bmatrix} + R_{ZYX_{t-1}} \begin{bmatrix} v_{x_{r_{t-1}}} \\ v_{y_{r_{t-1}}} \\ v_{z_{r_{t-1}}} \end{bmatrix} \delta + \frac{1}{2} R_{ZYX_{t-1}} \begin{bmatrix} \dot{v}_{x_{r_{t-1}}} \\ \dot{v}_{y_{r_{t-1}}} \\ \dot{v}_{z_{r_{t-1}}} \end{bmatrix} \delta^2 \tag{12a}$$

$$\begin{bmatrix} \gamma_{w_t} \\ \beta_{w_t} \\ \alpha_{w_t} \end{bmatrix} = \begin{bmatrix} \gamma_{w_{t-1}} \\ \beta_{w_{t-1}} \\ \alpha_{w_{t-1}} \end{bmatrix} + R_{ZYX_{t-1}} \begin{bmatrix} \omega_{\gamma_{r_{t-1}}} \\ \omega_{\beta_{r_{t-1}}} \\ \omega_{\alpha_{r_{t-1}}} \end{bmatrix} \delta \tag{12b}$$

$$\begin{bmatrix} v_{x_{r_t}} \\ v_{y_{r_t}} \\ v_{z_{r_t}} \end{bmatrix} = \begin{bmatrix} v_{x_{r_{t-1}}} \\ v_{y_{r_{t-1}}} \\ v_{z_{r_{t-1}}} \end{bmatrix} + \begin{bmatrix} \dot{v}_{x_{r_{t-1}}} \\ \dot{v}_{y_{r_{t-1}}} \\ \dot{v}_{z_{r_{t-1}}} \end{bmatrix} \delta \tag{12c}$$

$$\begin{bmatrix} \omega_{\gamma_{r_t}} \\ \omega_{\beta_{r_t}} \\ \omega_{\alpha_{r_t}} \end{bmatrix} = \begin{bmatrix} \omega_{\gamma_{r_{t-1}}} \\ \omega_{\beta_{r_{t-1}}} \\ \omega_{\alpha_{r_{t-1}}} \end{bmatrix} \tag{12d}$$

$$\begin{bmatrix} \dot{v}_{x_{r_t}} \\ \dot{v}_{y_{r_t}} \\ \dot{v}_{z_{r_t}} \end{bmatrix} = \begin{bmatrix} \dot{v}_{x_{r_{t-1}}} \\ \dot{v}_{y_{r_{t-1}}} \\ \dot{v}_{z_{r_{t-1}}} \end{bmatrix} \tag{12e}$$

Where $R_{ZYX_{t-1}}$ is a matrix of dimension 3×3 defined as in (13), and R_Z, R_Y, and R_X are the rotation matrices of dimension 3×3 about Z, Y and X axis.

$$R_{ZYX_{t-1}} = R_Z(\alpha_{w_{t-1}}) R_Y(\beta_{w_{t-1}}) R_X(\gamma_{w_{t-1}}) \tag{13}$$

This model represents a body which can move in 3D without any constraints. This is not ideal for our application since it does not reflect the unicycle model's constraints. We addressed this problem by feeding the filter an estimate for Y_r velocity of 0 with a low variance.

The measurements' vector size has dimention $k \times 1$, this value may differ from update to update cycle since sensors have different acquisition frequencies. As a result, matrix Q_t with dimention $k \times k$ and matrix K_t with dimention $n \times k$ also differ.

The measurements model function, $h(.)$, contrasting with function $g(.)$, is not used in the algorithm since the measurements arrive already transformed, yet matrix H_t still needs to be defined. The matrix H_t has dimension $k \times 15$, all its entries are zeros with the exception to one column for each measurement, which corresponds to the state variable the measurement will update, meaning that if the measurement i, updates the state variable j, $H_{i,j} = 1$.

D. Localization methods

For the estimates obtained using the EKF, the estimation was performed for 2D measurements only, x_w, y_w, α_w, v_{x_r}, v_{y_r}, ω_{α_r}, $\dot{v}_{x_r}$ and $\dot{v}_{y_r}$. The rest of the state variables (z_w, γ_w, β_w, v_{z_r}, ω_{γ_r}, ω_{β_r} and $\dot{v}_{z_r}$) were kept at 0 by forcing this value and a low variance (1×10^{-6}) at every iteration. For these estimates, the EKF computes an estimate at a frequency of $30Hz$. Note that the odometry and the calibrated odometry are also only 2D estimates.

Method A is the UGV's odometry. The estimate is computed by integrating the wheel encoders values using an approximate model of the UGV. This model uses a wheel diameter and axes track approximation.

Method B computed by calibrating the UGV's odometry.

Method C is the first in which the EKF was used, we configured it to use:

- the calibrated odometry velocities in X_r, Y_r and about Z_r;
- and the MPU IMU attitude about Z_r, angular velocity about Z_r and linear accelerations in X_r and Y_r.

Method D is built on top of the configuration from the previous one (method C). In this method we added:

- the Crossbow IMU measurements of angular velocity about Z_r and linear accelerations in X_r and Y_r.

Method E is also built on top of the configuration from the previous one (method D). In this estimate we added:

- the NovAtel GPS receiver's measurements along X_w and Y_w.

These measurements are converted into metric units by converting the latitude, longitude measurements to values in the respective Universal Transverse Mercator (UTM) coordinate system region.

V. Hardware

A. Unmanned Ground Vehicle

The UGV used is an ATRV-Jr mobile robot developed by iRobot, it was previously used in other projects, such as RESCUE - Cooperative Navigation for Rescue Robots, to conduct research in navigation and, search and rescue[1]. The UGV is steered using differential drive, being modeled after the unicycle model. Figure 1 illustrates the current condition of the UGV.

B. Sensors

When localizing a mobile robot with wheel odometry, one of the biggest difficulties is to keep an accurate heading. Odometry's heading drifts very quickly, in order to improve the localization estimation, a reliable source of heading is required.

IMUs usually measure linear acceleration, angular velocity and, less frequently, magnetic field. All this information can be fused to obtain a heading estimate.

The Crossbow IMU measures linear acceleration and angular velocity, multiple filters take advantage of this information

[1] http://rescue.isr.ist.utl.pt/publications.php

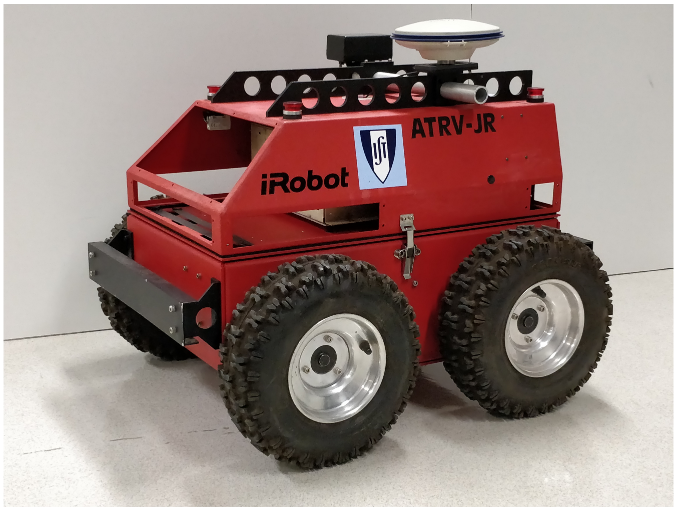


Fig. 1. ATRV-Jr's current configuration.

and estimate the IMU's attitude in relation to an inertial reference frame. To verify if we could rely on this sensor to obtain an attitude estimate, we estimate the UGV's attitude using the Madgwick filter [12] and a Complementary filter [13]. Using data collected during an indoor experiment, we compute the attitude estimate.

In our experiments, the Madgwick filter's attitude estimation error grows without bound at an approximately constant rate. The Complementary filter's attitude estimation error varies between approximately -0.5 and 0.61 radians.

In order to obtain a reliable attitude estimate, an MPU-6050 IMU was added. The estimate provided by the MPU IMU has shown to be more reliable than the ones provided by the Crossbow IMU.

While not ideal, the MPU IMU attitude estimation's error is lower than any of the previous estimates' error by more than one order of magnitude. In the same experiment, the MPU IMU estimate deviates from the real heading at a rate of approximately $1.22 \times 10^{-4} [rad/s]$. Ideally, an IMU with a magnetometer should be used. Only with a magnetic north reference, along the gravity vector is possible to uniquely define the inertial frame's three axes and thus eliminate the existent heading rate error.

At this point, the only state variables for which we have no measurements are x_w, y_w, and z_w, the position in the world frame. To address this problem, we added a NovAtel OEM4-G2 GPS receiver and compute Cartesian coordinates using the UTM coordinate system.

Figure 2 illustrates the present data flow diagram of the UGV.

VI. Software

Our hardware platform is comprised of several sensors and systems, in order to interact with all of them in a standard way we used ROS. To perform our experiments, several ROS packages were developed, adapted or used as they are.

The following ROS packages were developed solely for this application:

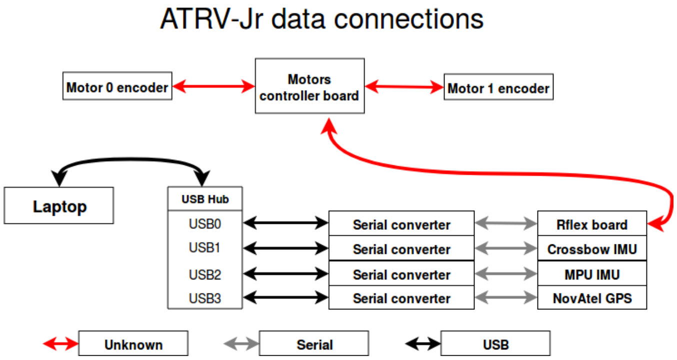


Fig. 2. Present ATRV-Jr's data connections' diagram.

- *atrvjr_description* - contains the description of all the frames present in the UGV in relation to the UGV's frame defined, as usual for robots modeled after the unicycle model, in the geometric center between the four wheels. The frames of both IMUs and the NovAtel GPS receiver are defined in this package.
- *atrvjr* - contains configuration and *roslaunch* files, a method used to start our system with ease.
- *host_console* - provides an interface to basic UGV functionalities, such as turn off the computer and display text in the rFLEX screen, useful when performing experiments.
- *odom_calibration* - contains the implementation of the calibration method and to apply the calibration to the UGV's odometry.
- *xbow6x* - has the tools required to connect to the Crossbow IMU via serial and parse the incoming measurements.

VII. Odometry calibration matrix

Using the data from the UGV's odometry and the MOCAP system poses, the calibration matrix was computed using data collected during indoor experiment. This matrix is then used by the *odom_calibration* package to compute the calibrated odometry. The estimated calibration matrix is shown in (14).

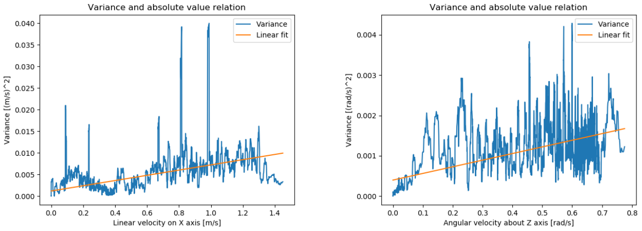
$$X = \begin{bmatrix} 9.47 \times 10^{-1} & -8.08 \times 10^{-3} & 1.84 \times 10^{-4} \\ -2.35 & 1.49 \times 10^{-1} & -4.78 \times 10^{-1} \\ 1.26 \times 10^{-1} & 2.35 \times 10^{-2} & 9.72 \times 10^{-1} \end{bmatrix} \quad (14)$$

VIII. Sensor models parameters

A. Calibrated odometry velocities model parameters

Using data collected during indoor experiments, the ground truth linear and angular velocities are computed. After computing the odometry velocities, its variance is determined by sorting them after taking their absolute value and computing the variance regarding the ground truth velocities in bins of 15 samples.

In Figure 3, we observe the relation between linear and angular calibrated odometry velocities absolute value and the measurements' variance. A linear regression was estimated between the two.



(a) Calibrated linear velocity absolute value's variance.

(b) Calibrated angular velocity absolute value's variance.

Fig. 3. Calibrated odometry velocities' linear fit.

TABLE I
CROSSBOW IMU AXES' BIAS.

	X	Y	Z	
Lin. acc.	-1.96×10^{-1}	-5.74×10^{-1}	5.63×10^{-2}	m/s^2
Ang. vel.	-3.47×10^{-2}	8.64×10^{-2}	6.19×10^{-2}	rad/s

The noise model parameters are defined in (15) for the linear velocity and in (16) for angular velocity.

$$\begin{aligned} m_v &= 4.92 \times 10^{-3}\ [m/s] \\ b_v &= 1.80 \times 10^{-3}\ [(m/s)^2] \end{aligned} \tag{15}$$

$$\begin{aligned} m_\omega &= 1.64 \times 10^{-3}\ [rad/s] \\ b_\omega &= 3.99 \times 10^{-4}\ [(rad/s)^2] \end{aligned} \tag{16}$$

The regressions' determination coefficient, R^2, for the linear velocity variance is 0.23, while for the angular velocity is 0.31. As evidenced in Figure 3, and confirmed by the R^2 values the fitness of the model is very poor.

B. Crossbow IMU model parameters

The Crossbow IMU's variance is computed using values from a static experiment, the bias and variance values for the three axes are shown in Tables I and II, respectively.

The bias values are subtracted to the measurements in the Crossbow IMU's driver, making the mean value of the measurements closer to the true value mean.

C. MPU IMU model parameters

The orientation estimate's variance is computed from the error present between the ground truth and the IMU estimate. Since, in the indoor experiments, we are moving in 2D, only the variance of α is estimated, and assumed the same for γ and β. The estimated variance corresponds to the value of $5.81 \times 10^{-2} [rad^2]$.

TABLE II
CROSSBOW IMU AXES' VARIANCE.

	X	Y	Z	
Lin. acc.	7.71×10^{-3}	8.94×10^{-3}	6.53×10^{-3}	$(m/s^2)^2$
Ang. vel.	5.83×10^{-7}	1.22×10^{-6}	1.15×10^{-6}	$(rad/s)^2$

The variances for the angular velocities and linear accelerations are computed from MPU-6050 datasheet's values. The gyroscope noise performance is 0.05 $[°/s]$, the angular velocity variance is defined in (17).

$$\begin{aligned} \sigma_\omega &= 0.05\ [°/s] \\ &= 8.73 \times 10^{-4}\ [rad/s] \\ \sigma_\omega^2 &= 7.62 \times 10^{-7}\ [(rad/s)^2] \end{aligned} \tag{17}$$

The accelerometer power spectral density is $400\,[\mu\, g \quad /\sqrt{Hz}]$, where g denotes the gravitational acceleration. The accelerometer uses a first-order RC low-pass filter, for which the bandwidth is computed by $B = 1.57 \cdot f_{-3dB}[Hz]$. Knowing the filter's cut off frequency, $42\,Hz$ and assuming a flat power spectral density, we compute the root mean squared noise (RMS) as in (18).

$$\begin{aligned} RMS &= \sqrt{42 \cdot 1.57} \cdot 400\,[\mu\, g\ /\sqrt{Hz}] \\ &= 3.19 \times 10^{-2}\ [m/s^2] \end{aligned} \tag{18}$$

Assuming the noise is approximately Gaussian, the RMS is approximately the standard deviation. The linear acceleration variance is defined in (19).

$$\begin{aligned} \sigma_v^2 &\approx RMS^2 \\ &\approx 1.02 \times 10^{-3}\ [(m/s^2)^2] \end{aligned} \tag{19}$$

The variances are the same for the three IMU axes.

D. NovAtel GPS receiver model parameters

The NovAtel GPS receiver estimates the measurements' standard deviations internally, the square of these values is taken to obtain its variance.

IX. EXTENDED KALMAN FILTER

The initial state, μ_0 and Σ_0, is assumed to be known. The initial x_w and y_w are obtained from the first GPS measurement, and the initial α is obtained from the first MPU IMU measurement, the rest of the state variables are 0. Its covariance matrix is defined as in (20), where I is the identity matrix with dimension 15×15.

$$\Sigma_0 = I \cdot 1 \times 10^{-9} \tag{20}$$

The process noise covariance matrix R reflects how much we trust our model to predict the next state. It is required to estimate this matrix, but in general, there is no systematic way to calculate the process noise covariance matrix [14]. The estimation was done using the knowledge we have about the process and is the same for every experiment. Meaning that it was not tuned to obtain a better result in a specific one, but to best represent the process noise covariance. The matrix R is determined as in (21).

$$\begin{aligned} \mathbf{r} = [&0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.5, \\ &0.5, 0.5, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3]^T \\ R_{i,i} = \mathbf{r}&_i, \forall\ i \in [0, 14] \end{aligned} \tag{21}$$

X. Results and discussion

A. Outdoor experiments

We conducted several outdoor experiments inside the IST Alameda Campus. In Figure 4, the experiments' approximate path is represented by GPS measurements. These measurements are undersampled and represented by blue dots.

The first and second experiments were conducted far from any buildings or large trees. For that reason, the measurements are relatively close to each other and do not show any large discontinuities.

The first and second experiments were performed approximately under the same conditions. The terrain, Portuguese pavement, is not smooth, contains loose stones, and an approximately constant slope.

In the third experiment, the terrain type, and slope changes. This experiment begins in Portuguese pavement and ends in asphalt, it also encompasses diverse slopes along the way.

During the fourth experiment, we drove the UGV on asphalt for the full duration of the test, the slope is approximately constant and leveled. This terrain is smoother than the Portuguese pavement. As demonstrated further on, this results in better overall localization estimation for this experiment.

During the outdoor experiments, we do not have access to a reliable pose ground truth, the collected GPS measurements act as a position reference but note that the confidence of this position estimate is low. Also, the GPS measurements do not provide a reliable attitude estimation.

Table III contains the traveled distance during each experiment. In experiments with a higher traveled distance, we expect higher pose errors for the estimates A to D, since these only predict the position but do not update it.

TABLE III
Experiments' traveled distance.

Experiment	Traveled distance $[m]$
First	141.16
Second	275.04
Third	309.32
Fourth	137.15

The third experiment's average position errors are the highest among all experiments, since it is the one with the largest traveled distance, and its path encompasses the largest differences in terrain type and slope, which lead to increased measurements' noise.

B. Localization methods

The estimates are first compared using the average position error along the experiment, taking the GPS's measurement as a reference. Note that in the figures of estimates C and D the final position's 2D standard deviation ellipse is given. This ellipse represents the 2D position uncertainty with 99% confidence.

1) Localization method A and B: As usual with UGV's, our localization estimation starts with the study of the vehicle's odometry (localization method A).

TABLE IV
Estimate A's average position error.

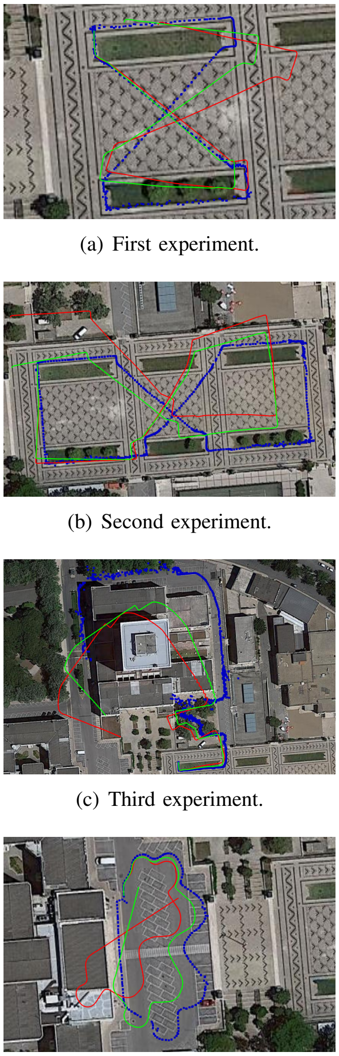
Experiment	Average position error $[m]$
First	5.98
Second	10.74
Third	25.75
Fourth	12.44

The average position error considering the GPS's measurements is depicted in Table IV.

As expected the error is higher for the experiments with higher traveled distance, although not true for every experiment. The first experiment has a lower error but a marginally higher traveled distance than the fourth.

The estimate B is obtained by calibration of the UGV's odometry.

Figure 4 depicts the comparison between GPS samples, the UGV's odometry (shown in red), and the calibrated odometry here (shown in green).



(a) First experiment.

(b) Second experiment.

(c) Third experiment.

(d) Fourth experiment.

Fig. 4. Estimate B, calibrated odometry. Undersampled GPS' samples in blue, estimate A in red and estimate B in green.

The first, second and fourth experiments finish with almost the same pose as they started. For those, we observe that the calibrated odometry's last estimate is much closer to the starting position than the non-calibrated odometry.

The third experiment does not finish with the same pose as it started. Nonetheless, we observe that the calibrated odometry is closer to the approximated path given by the GPS samples.

Analyzing the path shape for the experiments, still with heavy distortion, we see that the overall calibrated path shape is closer to the GPS samples than the odometry's path (estimate A). The average position error taking the GPS's measurement is depicted in Table V.

TABLE V
ESTIMATE B'S AVERAGE POSITION ERROR.

Experiment	Average position error $[m]$
First	3.42
Second	8.22
Third	16.37
Fourth	4.96

The error follows the same behavior as in the estimate A, regarding the traveled distance. The error lowered across all experiments, but for the fourth experiment the decrease is especially evident being of approximately 250% when compared to the one from the previous method. For the rest of the experiments is of approximately 150%.

2) Localization method C: Estimate C is obtained by filtering the calibrated odometry velocities and the MPU IMU measurements. In Figure 5, the filter's estimate is illustrated (here shown in magenta), along with the final 2D position's uncertainty.

The average position error considering the GPS's measurements is depicted in Table VI.

TABLE VI
ESTIMATE C'S AVERAGE POSITION ERROR.

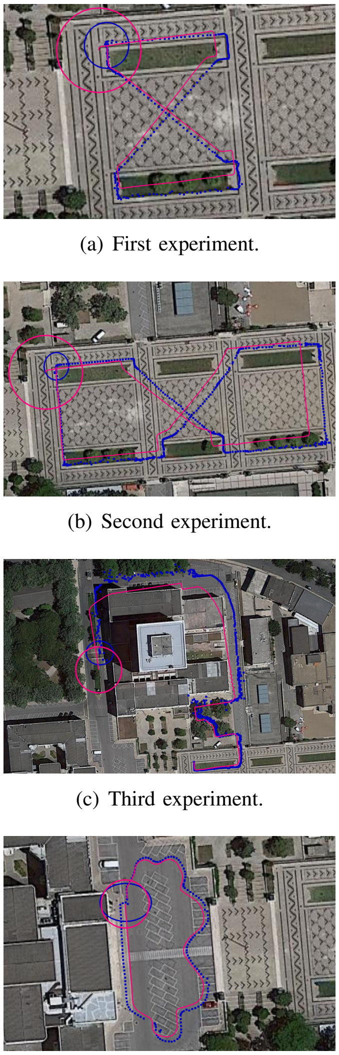
Experiment	Average position error $[m]$
First	2.00
Second	4.05
Third	5.67
Fourth	1.40

The use of a reliable heading estimate, along with the angular velocities and linear accelerations, results in a decrease in the average position error of more than 350% in the case of the fourth experiment. For the remaining experiments, the decrease is above 170%.

It is to note the difference between the first and fourth experiments' error. As mentioned earlier, the first experiment's traveled distance is only marginally higher than the fourth one, yet the error is 140% higher than the fourth experiment's error. This is due to the exceptionally favorable conditions under which the fourth experiment was conducted.

3) Localization method D: The estimate D is obtained by filtering the calibrated odometry velocities, the MPU IMU and the Crossbow IMU measurements. Its results are shown in Figure 6.

The overall path is similar to the previous estimate, which is to be expected since the addition of the Crossbow IMU brings measurements to state variables which were already being updated. The average position error considering the GPS's measurements is depicted in Table VII.



(a) First experiment.

(b) Second experiment.

(c) Third experiment.

(d) Fourth experiment.

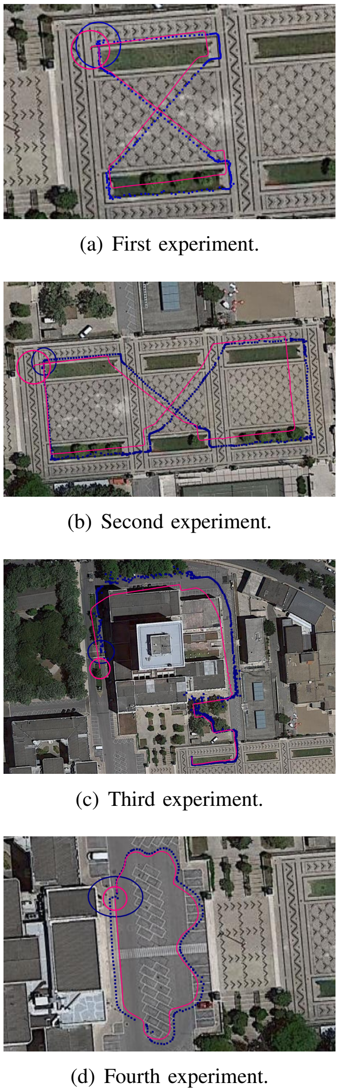
Fig. 5. Estimate C, EKF filtering calibrated odometry and MPU IMU's measurements. Undersampled GPS' samples in blue and estimate C in magenta.

TABLE VII
ESTIMATE D'S AVERAGE POSITION ERROR.

Experiment	Average position error $[m]$
First	2.01
Second	4.03
Third	5.45
Fourth	1.42

This estimate's overall path is similar to estimate C's, this is also apparent in the experiments' average position error. These are similar to the previous estimate's experiments average position errors for all the experiments, having marginally decreased for the second and third experiments, while marginally increasing for the first and fourth experiments.

4) Localization method E: The estimate E is obtained by filtering the calibrated odometry velocities, the MPU IMU,



(a) First experiment.

(b) Second experiment.

(c) Third experiment.

(d) Fourth experiment.

Fig. 6. Estimate D, EKF filtering calibrated odometry, MPU and Crossbow IMU's measurements. Undersampled GPS' samples in blue and estimate D in magenta.

TABLE VIII
ESTIMATE E'S AVERAGE POSITION ERROR.

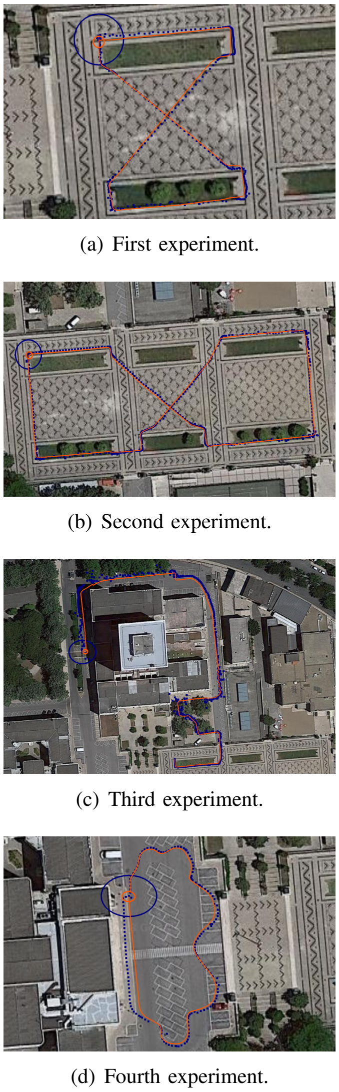
Experiment	Average position error $[m]$
First	0.52
Second	0.44
Third	1.34
Fourth	0.97

the Crossbow IMU and the NovAtel GPS receiver's measurements. Its results are shown in Figure 7.

The average position error considering the GPS's measurement is depicted in Table VIII.

The addition of global position's measurements results in lower errors across all experiments, the effects of the GPS's measurements use are evident in the estimated path.

We note the high error for the third experiment. This occurs due to the GPS's measurements higher mean dispersion in the location where the experiment finished. Estimate E's final 2D position's uncertainty is also much lower than in any of the previous estimates.



(a) First experiment.

(b) Second experiment.

(c) Third experiment.

(d) Fourth experiment.

Fig. 7. Estimate E, EKF filtering calibrated odometry, MPU IMU, Crossbow IMU and NovAtel GPS receiver's measurements. Undersampled GPS' samples in blue and estimate E in orange.

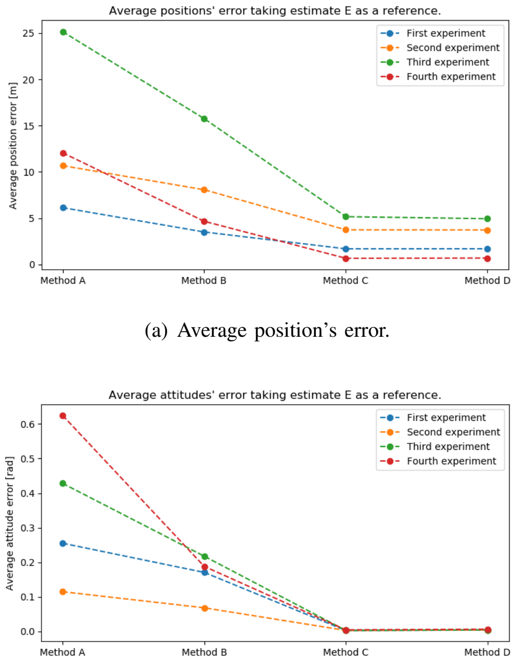
Considering all the estimates, this is the one which we expect to have the lowest pose error.

C. Discussion

Two different metrics are now used to compare all the viable estimates to estimate E. For position, we use the average Cartesian distance between Estimate A to D and Estimate E, for orientation we use the average absolute error between the two estimates α. These are illustrated in Figures 8(a) and 8(b).

Regarding the position, we find that the experiments' average error does not converge to the same value. This indicates that the error is still dependent on the experiment's characteristics, such as terrain and traveled distance. Regarding method D, the average position error is $1.70[m]$ for the first experiment, $3.74[m]$ for the second, $4.95[m]$ for the third and $0.7[m]$ for the fourth.

Regarding the attitude, the estimates' average error behaves differently than in the position's error.



(a) Average position's error.

(b) Average attitude's error.

Fig. 8. Position and attitude average error taking estimate E as a reference.

In estimate A, the UGV's odometry, the experiment with the highest error is the fourth experiment. This contrasts with the previous results, where the third experiment is, by a large margin, the one with the highest errors.

In estimate B, calibrated odometry, all the errors decrease, with special significance for the fourth experiment. While these experience's error decreased significantly, this experiment still has the third highest error. The experiment with the highest is the third experiment.

In estimate C, the MPU IMU attitude estimate is introduced, lowering the error significantly in both metrics. This is expected since we identified that heading error significantly affected the localization estimation.

In the fourth estimate, the Crossbow IMU is introduced. As observed in the experiments' figures, this did not change the path shape significantly. This is reflected in the error, remaining approximately constant for estimate C and D. The relevance of this last estimate is in the method robustness. If for some reason the MPU IMU stops providing measurements, the Crossbow IMU still continues to provide the same measurements as the MPU IMU with exception to the attitude.

It is to note the difference between the position and attitude's average error. In the position's error estimation, the localization estimates are compared against one which uses a global localization system. This system's measurements were not used in the previous estimates, thus estimate A to D's position state variables were not updated but only predicted.

In the attitude error estimation, the localization estimates are compared against estimate E which updates its state using the same references as D, the IMUs. This results in the different behavior observed between the position and attitude error for estimate D.

The fact that the position error highly differs between experiments, indicates that our approach is not robust enough for outdoor environments in which we are deprived of global position measurements.

XI. Conclusions

We updated the UGV and studied the capabilities of its previous sensor suite, the sensors which were deprecated were removed. Working on the existent sensors, we identified the missing features by evaluating the performance of each one. Using this information, updated sensors were added and its capabilities studied.

We conducted real-world experiments in outdoor environments in which all the data from the sensors is recorded. With it and using different methods, we obtained several estimates which performed differently. We assessed their performance using an estimate which was obtained but comprises GPS's measurements, making it an invalid solution for our problem but a reliable pose estimate.

For accomplishing such solution, we developed several software packages, this allowed us to assess and test multiple sensors which data was not available before. The ROS packages *atrvjr_description*, *atrvjr*, *host_console*, *odom_calibration*, and *xbow6x* were developed during the course of this thesis, and several other studied and modified.

The obtained results do not serve a wide range of applications due to its low accuracy, but given the simplicity of this approach and flexibility, they are a strong foundation for incorporation of different measurements.

Besides the ones used, there are other sensors which are commonly used in localization such, scanning laser rangefinders, depth and, RGB cameras. For future work we propose to assess the performance of a scanning laser rangefinder, methods such as scan matching can be used to provide a second source of velocities in 2D. Depth cameras usually use infrared laser technology these are not reliable outdoor due to the presence of infrared radiation emitted by the sun, yet depth information can be obtained from other sensors such as an array of RGB cameras.

We prepared our method to estimate 3D pose, but due to the noisy measurements along the Z_w axis, specifically from the IMUs, the estimation diverged along this. With the addition of other sensors, such as visual odometry, this problem can be attenuated and a 3D pose estimate obtained. One of the pose error sources is bad attitude estimation, this has been addressed with the addition of the MPU-6050 IMU, but an IMU with a magnetometer must be used to reliably estimate the attitude over longer periods of time.

The ROS package which has the implementation of the used EKF also contains another filter which uses most of the same configuration, but it is based on unscented Kalman filtering. Due to time constraints, it was not possible to obtain an estimation using such filter, but improvements in the state estimation have been reported by others when using such filter.

References

[1] MBZIRC organization, "MBZIRC 2020 challenge description," MBZIRC, Tech. Rep. v2, May 2018.

[2] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, “Recent advances in indoor localization: A survey on theoretical approaches and applications,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1327–1346, Secondquarter 2017.

[3] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, Dec 1996.

[4] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, “Robust localization using relative and absolute position estimates,” in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, vol. 2, Oct 1999, pp. 1134–1140 vol.2.

[5] K. A. M. B. W. B. Giorgio Grisetti, Cyrill Stachniss. (2010) Odometry calibration by least squares. [Online]. Available: http://ais.informatik.uni-freiburg.de/teaching/ws10/robotics2/pdfs/rob2-05-odometry-calib-practical.pdf

[6] L. Jetto, S. Longhi, and G. Venturini, “Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 219–229, 1999.

[7] I. J. Cox, “Blanche-an experiment in guidance and navigation of an autonomous robot vehicle,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 193–204, April 1991.

[8] S. D. Gupta, J. Y. Yu, M. Mallick, M. Coates, and M. Morelande, “Comparison of angle-only filtering algorithms in 3d using ekf, ukf, pf, pff, and ensemble kf,” in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 1649–1656.

[9] J. Simanek, M. Reinstein, and V. Kubelka, “Evaluation of the ekf-based estimation architectures for data fusion in mobile robots,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 985–990, April 2015.

[10] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[11] G. J. Bierman and C. L. Thornton, “Numerical comparison of kalman filter algorithms: Orbit determination case study,” *Automatica*, vol. 13, no. 1, pp. 23 – 35, 1977.

[12] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of imu and marg orientation using a gradient descent algorithm,” in *2011 IEEE International Conference on Rehabilitation Robotics*, June 2011, pp. 1–7.

[13] R. G. Valenti, I. Dryanovski, and J. Xiao, “Keeping a good attitude: A quaternion-based orientation filter for imus and margs,” *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[14] H. . Nguyen and F. Guillemin, “On process noise covariance estimation,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*, July 2017, pp. 1345–1348.