

## Article

# Auto-Calibration Methods of Kinematic Parameters and Magnetometer Offset for the Localization of a Tracked Mobile Robot

Luciano Cantelli \*, Samuel Ligama, Giovanni Muscato and Davide Spina

Dipartimento di Ingegneria Elettrica Elettronica e Informatica(DIEEI), Università degli Studi di Catania, Catania 95125, Italy; ligamasamuel@yahoo.it (S.L.); gmuscato@dieei.unict.it (G.M.); davide.spina@dieei.unict.it (D.S.)

\* Correspondence: luciano.cantelli@dieei.unict.it; Tel.: +39-095-738-2349

Academic Editor: Huosheng Hu

Received: 5 August 2016; Accepted: 27 October 2016; Published: 1 November 2016

**Abstract:** This paper describes an automatic calibration procedure adopted to improve the localization of an outdoor mobile robot. The proposed algorithm estimates, by using an extended Kalman filter, the main kinematic parameters of the vehicles, such as the wheel radii and the wheelbase as well as the magnetometer offset. Several trials have been performed to validate the proposed strategy on a tracked electrical mobile robot. The mobile robot is aimed to be adopted as a tool to help humanitarian demining operations.

**Keywords:** localization; field robot; kinematic parameters calibration

## 1. Introduction

This paper presents a procedure and some experimental results concerning the auto-calibration of the kinematic parameters of a tracked vehicle, developed for humanitarian demining operations. The vehicle is aimed to bring metal detectors, ground penetrating radars and intelligent probes to discover the presence of buried mines, along with other tools to disarm or detonate them. It is estimated that around 100 million landmines are still present as the result of previous conflicts in more than 84 countries. Classical manual demining operations are very slow and dangerous and need the help of technology. The European Commission (EC) funded project TIRAMISU is developing several tools to speed-up demining operations [1,2]. Robots can represent a useful technology for humanitarian demining, allowing human workers to stay at a safe distance during the operations [3,4].

Localization is of paramount importance in this and many other robotic applications, in particular when autonomous operations are required. Outdoor mobile robots locate their positions by using GNSS (global navigation satellite systems) such as GPS, which nowadays can reach centimeter-level of accuracy. However in the presence of obstacles, such as big rocks, buildings, or vegetation, the satellite signal can be obstructed and this precision could quickly degrade. In these cases the adoption of relative positioning systems such as odometry, using encoders or Inertial Measurement Units (IMU), can greatly help to localize the robot.

Odometry is mainly based on the assumption that measures of wheel movements can be converted into measures of the robot motion using the kinematic model of the vehicle. This assumption is only of restricted value, because there are a number of factors bringing inaccuracies in the odometry. The main error sources are: wheels slipping, uncertainty in the kinematic parameters (wheel radii and wheelbase), misalignment of the wheels and finite encoder resolution and sampling frequency.

The errors caused by kinematic parameters uncertainty could be reduced if a good calibration is performed. However the errors caused by the wheels slipping cannot be estimated by the encoders and hence cannot be compensated. In particular, for a skid steering robot, during a rotation the estimation

of the orientation performed solely by odometry can result in very significant errors. In this case, the adoption of an inertial measurement unit together with a magnetometer can give the measurement of the absolute orientation of the robot. However, the magnetometer should also be calibrated, especially as regard its offset, which for an electric vehicle full of metal parts and magnets, can assume significant variations.

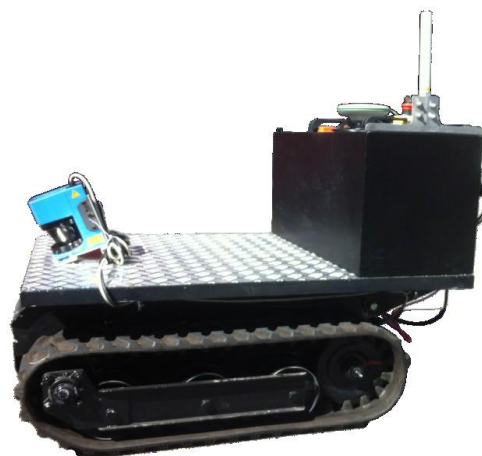
Calibration is essentially the problem of estimating the parameters of the robot model from sensor measures. Several papers in the past have addressed the problem of self-calibration of the kinematic parameters using a wide variety of algorithms and sensors. For example in [5] an Extended Kalman Filter is adopted to perform the calibration of the parameters of a six-wheel skid steering robot adopted for volcanic explorations. The calibration of the odometric parameters, together with the extrinsic pose estimation of an exteroceptive sensor, is formulated as a maximum-likelihood problem in [6]. A different approach is presented in [7], where the calibration is obtained as the solution of an optimization problem with a statistical method. In [8] an EKF is adopted for odometry calibration of a differential drive robot using the data collected from a floor color sensor. The redundancy of the sensor outputs is adopted to perform the automatic calibration on a multisensor system in [9], while in [10] the observable filter is introduced and used together with an augmented Kalman filter for the estimation of the non-systematic odometry error for a mobile robot. In [11,12] an EKF is used to estimate wheel slip and “virtual” velocity of a skid-steered mobile robot. An approach to estimate the odometry calibration parameters while performing SLAM is presented in [13] by using an expectation maximization algorithm, while [14] extends the graph-based formulation of the SLAM problem to handle the calibration parameters.

However in all these papers the estimation of the magnetometer offset is not considered.

This paper, after a short introduction on the considered platform and sensors, will describe the localization and calibration procedure for the kinematic and for the magnetometer offset and will show some examples of experimental results.

## 2. The Considered Tracked Platform

The mobile robot used in this work is shown in Figure 1. This is a tracked robot, in differential drive configuration. Through the tracks, the motion is transmitted from front actuated wheels to the back passive ones. A DC brushed motor actuates each front wheel, which, using the differential configuration, can run with different velocities, in order to control the robot direction.



**Figure 1.** The considered tracked platform.

From the structural point of view, the robot has a central compartment, with dimensions  $99.1 \text{ cm} \times 48.5 \text{ cm} \times 28.5 \text{ cm}$ . Each track has a width of 18 cm and the distance between the two tracks measures  $b = 70 \text{ cm}$ . The diameter of the actuated wheels is equal to  $D = 0.16 \text{ cm}$ .

Inside the central compartment, as it is shown in Figure 2, there are:

- The embedded robot controller based on the *Single Board RIO 9626 (National Instruments)*.
- LiFePO4 battery: eight cells 110 Ah in series.
- Battery control unit.
- Two DC brushed motors capable to supply a maximum power of 800 W each and to reach speeds of 2800 RPM without load. Each motor is equipped with a gear reducer having a speed reduction ratio of 1/32.
- Two digital drives of the series *DRUM* and produced by *Elmo Motion Control*, for the motor drive.
- Two encoders by *Kubler* of 1024ppr, placed on each rotor axis of the motors to perform the speed closed loop control.
- Four voltage regulators to give the correct power to all the on board devices, including the government units and the sensors.

In regard to other sensors, there is a “Leica gs10” Real Time Kinematics (RTK) Differential GPS (DGPS) system, a SICK LMS200 laser scanner and an Xsense MTI Attitude and Heading Reference System (AHRS), which includes an inertial platform and a triaxial magnetometer. This sensor gives directly the estimation of the attitude and heading of the robot. For the implementation of remotely radio-controlled algorithms, a digital joystick was used for sending linear and angular velocity references, thanks the use of a WI-FI connection, using a wireless Access Point by ACKSYS.

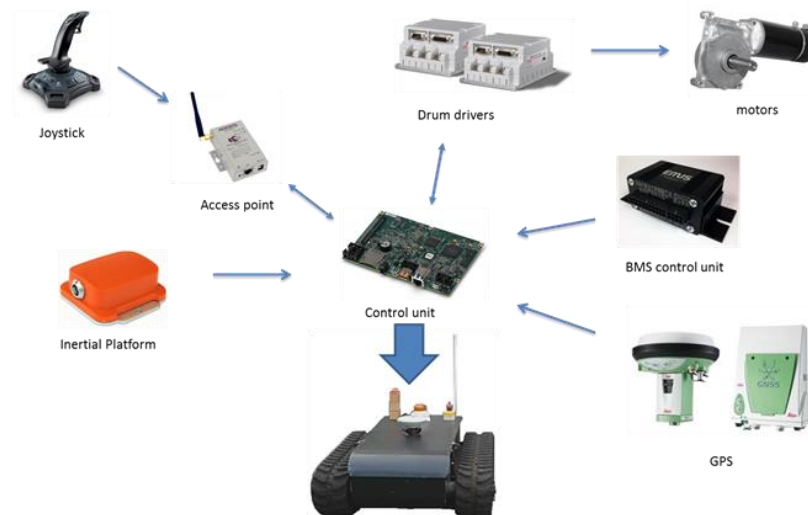


Figure 2. The hardware modules connected to the control board.

### 3. Auto-Calibration Methodology

An Extended Kalman Filter (EKF) algorithm is used to fuse the DGPS, encoder and AHRS data, in order to calculate the robot position and direction and to estimate the robot model parameters.

As it was mentioned in the introduction, the calibration of the robot parameters is periodically required. This can be achieved by fusing model data and internal measurements from the encoders, with external sensorial data obtained from the GPS and the AHRS platform.

#### 3.1. Robot Kinematic Model

The robot is characterized by a position  $(x, y)$  and an orientation  $\theta$  on a horizontal plane:

$$\begin{cases} \dot{x}(t) = vs(t)\cos\theta(t) \\ \dot{y}(t) = vs(t)\sin\theta(t) \\ \dot{\theta}(t) = \omega(t) \end{cases} \quad (1)$$

where  $vs(t)$  and  $\omega(t)$  are respectively the linear and angular speed of the robot. The discretized model is simply calculated as it follows:

$$\begin{cases} x(k+1) = x(k) + T \cdot vs(k) \cos \theta(k) \\ y(k+1) = y(k) + T \cdot vs(k) \sin \theta(k) \\ \theta(k+1) = \theta(k) + T \cdot \omega(k) \end{cases} \quad (2)$$

where  $T$  represents the sampling time.

Two optical encoders are used for evaluating the angular displacement of the robot wheels. Then  $vs(k)$  and  $\omega(k)$  are calculated by the following equations:

$$\begin{cases} vs(k) = \frac{R_L \cdot \Delta \theta_L(k) + R_R \cdot \Delta \theta_R(k)}{2T} \\ \omega(k) = \frac{R_L \cdot \Delta \theta_L(k) - R_R \cdot \Delta \theta_R(k)}{L \cdot T} \end{cases} \quad (3)$$

where:

- $\Delta \theta_L(k)$  and  $\Delta \theta_R(k)$ , are respectively the displacement of the right and left wheels, during the interval  $[T(k-1), T(k)]$ ;
- $R_L$  and  $R_R$  represent the radii of the right and left wheels, including track width;
- $L$  is the wheelbase, i.e., the distance between left and right tracks.

Combining the last equations, the mathematical model of the robot becomes:

$$\begin{cases} x(k+1) = x(k) + T \cdot \frac{R_L \cdot \Delta \theta_L(k) + R_R \cdot \Delta \theta_R(k)}{2T} \cos \theta(k) \\ y(k+1) = y(k) + T \cdot \frac{R_L \cdot \Delta \theta_L(k) + R_R \cdot \Delta \theta_R(k)}{2T} \sin \theta(k) \\ \theta(k+1) = \theta(k) + T \cdot \frac{R_L \cdot \Delta \theta_L(k) - R_R \cdot \Delta \theta_R(k)}{L \cdot T} \end{cases} \quad (4)$$

The EKF localization algorithm, using odometry measures, computes at high frequency the *expected position* and *orientation* of the robot, and the *update phase* corrects these estimations at a lower frequency using the GPS and AHRS platform measures [15]. The odometry parameters are considered as state variables in the EKF. In particular once defined the *state vector*:

$$X(k) = [x(k), y(k), \theta(k), R_L(k), R_R(k), L(k)]^T \quad (5)$$

The mathematical model becomes:

$$\begin{cases} x(k+1) = x(k) + T \cdot \frac{R_L(k) \cdot \Delta \theta_L(k) + R_R(k) \cdot \Delta \theta_R(k)}{2T} \cos \theta(k) \\ y(k+1) = y(k) + T \cdot \frac{R_L(k) \cdot \Delta \theta_L(k) + R_R(k) \cdot \Delta \theta_R(k)}{2T} \sin \theta(k) \\ \theta(k+1) = \theta(k) + T \cdot \frac{R_L(k) \cdot \Delta \theta_L(k) - R_R(k) \cdot \Delta \theta_R(k)}{L(k) \cdot T} \\ R_L(k+1) = R_L(k) \\ R_R(k+1) = R_R(k) \\ L(k+1) = L(k) \end{cases} \quad (6)$$

The inputs of the kinematic model are the encoder measures:

$$U(k) = [\Delta \theta_L(k), \Delta \theta_R(k)]^T \quad (7)$$

while the output vector is represented by the robot position:

$$Z(k) = H \cdot X(k) = [x(k), y(k)]^T \quad (8)$$



where  $H$  is the following matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

The kinematic model, so, can be simply written as:

$$\begin{cases} X(k+1) = f(X(k), U(k)) + w(k) \\ Z(k) = H \cdot X(k) + v(k) \end{cases} \quad (10)$$

where  $w(k)$  is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance  $C_w$  while  $v(k)$  is the observation noise which is assumed to be zero mean Gaussian white noise with covariance  $C_v$ .

### 3.2. Self-Calibrating Extended Kalman Filter (SC EKF)

#### 3.2.1. Predictive Phase

The predictive phase of the EKF algorithm, calculates the *predicted state*

$$\hat{X}(k+1|k) = f(\hat{X}(k|k), U(k)) \quad (11)$$

and the predicted covariance matrix

$$P(k+1|k) = J_x(k) \cdot P(k|k) \cdot J_x^T(k) + J_u(k) \cdot C_u(k) \cdot J_u^T(k) + C_w \quad (12)$$

where:

- $P$  is the covariance matrix and represents the uncertainty of the estimated state;
- $C_u$  is the covariance matrix of the input noise;
- $J_x(k)$  and  $J_u(k)$  are the Jacobians of the system.

#### 3.2.2. Update Phase

The GPS and AHRS platform measurements, which represent the external sensorial data, are read asynchronously with respect to the odometry ones: the encoders' measures are available at higher frequencies.

When a new external sensorial measurement is not available, the update phase is very simple: the state and the covariance matrix are obtained directly from the predicted state vector and the covariance matrix, while, if a new sensorial measurement is available, the *Kalman Gain Matrix*  $K(k+1)$  is calculated in order to update the results. Since we have two kinds of external sensorial measures, it is better to realize two update phases in series.

#### AHRS Update Phase

$$K_{\text{AHRS}}(k+1) = P(k+1|k) \cdot H^T \cdot \left[ H \cdot P(k+1|k) \cdot H^T + C_{v_{\text{AHRS}}}(k+1) \right]^{-1} \quad (13)$$

where  $C_{v_{\text{AHRS}}}(k+1)$  is the measure covariance matrix, obtained directly from the AHRS platform data:

$$C_{v_{\text{AHRS}}}(k+1) = \sigma_{\text{AHRS}}^2 \quad (14)$$

With  $\sigma_{\text{AHRS}}$ , representing the yaw uncertainty of the MTI AHRS sensor.

The measurement vector is also obtained directly from the AHRS platform measures:

$$Z_{\text{AHRS}}(k+1) = \text{Yaw}(k+1) \quad (15)$$

The predicted measurement vector is:

$$\begin{aligned} H_{\text{AHRS}} &= [0 \ 0 \ 1 \ 0 \ 0 \ 0] \\ \hat{Z}_{\text{AHRS}}(k+1|k) &= H_{\text{AHRS}} \cdot \hat{X}(k+1|k) \end{aligned} \quad (16)$$

The measurement vector is compared to the predicted measurement vector in order to obtain the estimated state  $\hat{X}(k+1)$ , using the Kalman Gain Matrix:

$$\hat{X}_{\text{update } 1}(k+1) = \hat{X}(k+1|k) + K_{\text{AHRS}}(k+1) \cdot [Z_{\text{AHRS}}(k+1) - \hat{Z}_{\text{AHRS}}(k+1|k)] \quad (17)$$

and the state covariance matrix is evaluated as follows:

$$P_{\text{update } 1}(k+1) = [I - K_{\text{AHRS}}(k+1) \cdot H_{\text{AHRS}}] \cdot P(k+1|k) \quad (18)$$

### GPS Update Phase

A similar procedure is performed for this phase, taking into account that in the previous case, only a measurement for each cycle was done. Now with the GPS data, two measurements for cycle ( $x_{\text{GPS}}$ ,  $y_{\text{GPS}}$ ) are available, so great attention should be given to the matrices dimensions.

Starting from the state vector and the covariance matrix, evaluated in the previous update phase:

$$\begin{aligned} K_{\text{GPS}}(k+1) &= P_{\text{update } 1}(k+1) \cdot H_{\text{GPS}}^T \\ &\cdot [H_{\text{GPS}} \cdot P_{\text{update } 1}(k+1) \cdot H_{\text{GPS}}^T + C_{v_{\text{GPS}}}(k+1)]^{-1} \end{aligned} \quad (19)$$

where  $C_{v_{\text{GPS}}}(k+1)$  is the measurement covariance matrix, obtained directly from the GPS data:

$$C_{v_{\text{GPS}}}(k+1) = \begin{bmatrix} \sigma_{\text{GPS}}^2(k) & 0 \\ 0 & \sigma_{\text{GPS}}^2(k) \end{bmatrix} \quad (20)$$

Observe that  $\sigma_{\text{GPS}}$  value changes, according to the quality of the GPS signal.

The measurement vector is obtained directly from the GPS measures:

$$Z_{\text{GPS}}(k+1) = \begin{pmatrix} x_{\text{GPS}}(k+1) \\ y_{\text{GPS}}(k+1) \end{pmatrix} \quad (21)$$

The predicted measurement vector is:

$$\begin{aligned} H_{\text{GPS}} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \hat{Z}_{\text{GPS}}(k+1|k) &= H_{\text{GPS}} \cdot \hat{X}_{\text{update } 1}(k+1) \end{aligned} \quad (22)$$

The measurement vector is compared to the predicted measurement vector in order to obtain the estimated state  $X(k+1)$ , using the Kalman Gain Matrix:

$$\begin{aligned} \hat{X}(k+1|k+1) &= \hat{X}_{\text{update } 1}(k+1) + K_{\text{GPS}}(k+1) \\ &\cdot [Z_{\text{GPS}}(k+1) - \hat{Z}_{\text{GPS}}(k+1|k)] \end{aligned} \quad (23)$$

and the state covariance matrix is evaluated as follows:

$$P(k+1|k+1) = [I - K_{\text{GPS}}(k+1) \cdot H_{\text{GPS}}] \cdot P_{\text{update } 1}(k+1) \quad (24)$$

Obviously, if the AHRS measurements are not available, the AHRS update phase will not be executed. Then GPS update will work with state vector and covariance matrix evaluated during the prediction phase with:

$$\hat{X}_{update1}(k+1) = \hat{X}(k+1|k); \quad (25)$$

$$P_{update1}(k+1) = P(k+1|k) \quad (26)$$

Similarly, if GPS measurements are not available, only the AHRS update phase will be executed. In this last case:

$$\hat{X}(k+1|k+1) = \hat{X}_{update1}(k+1) \quad (27)$$

$$P(k+1|k+1) = P_{update1}(k+1) \quad (28)$$

### 3.3. EKF with Offset Estimation

In this work, EKF was also used for another important aspect. Indeed, during experimental robot motions, it was possible to see a significant magnetic disturbance effect of building structures and robot structures on the magnetometer in the AHRS. This negative effect does not allow reaching very good results in terms of localization, because the acquired orientation data of the robot are affected by large errors. This means that the path reconstruction of the Kalman filter is not sufficient to predict the exact position of the robot and to localize its correct position, in case of GPS signal loss or low precision of GPS position coordinates.

This is why it was necessary to compensate the value of this disturbance, which can be seen as an offset, in order to improve EKF performances. An initial calibration was not sufficient, since it was experimentally observed that this offset value is not constant during long trajectories.

Another state variable, called “*Ofs*”, was added into the model:

$$X(k) = [x(k), y(k), \theta(k), R_L(k), R_R(k), L(k), Ofs(k)]^T \quad (29)$$

In this way, the mathematical model becomes:

$$\left\{ \begin{array}{l} x(k+1) = x(k) + T \cdot \frac{R_L(k) \cdot \Delta\theta_L(k) + R_R(k) \cdot \Delta\theta_R(k)}{2T} \cos\theta(k) \\ y(k+1) = y(k) + T \cdot \frac{R_L(k) \cdot \Delta\theta_L(k) + R_R(k) \cdot \Delta\theta_R(k)}{2T} \sin\theta(k) \\ \theta(k+1) = \theta(k) + T \cdot \frac{R_L(k) \cdot \Delta\theta_L(k) - R_R(k) \cdot \Delta\theta_R(k)}{L(k) \cdot T} \\ R_L(k+1) = R_L(k) \\ R_R(k+1) = R_R(k) \\ L(k+1) = L(k) \\ Ofs(k+1) = Ofs(k) \end{array} \right. \quad (30)$$

Now the model consists in seven equations, so also the Jacobian matrices and all the dimensions of the matrices of the filter also change.

The measurement vector is obtained again directly from the AHRS platform measures according to Equation (15), however the predicted measurement vector is now modified into:

$$H_{AHRS} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1] \quad (31)$$

Since it now takes into account the presence of the offset and in this way it affects the angle estimation.

## 4. Experimental Results

The previous equations have been implemented in the adopted control board *SbRIO 9626* by National Instruments. The board was programmed using the LabView™ programming language.

Various tests in different conditions have been performed. In the first two experiments two different trajectories, a circular path and a rectangular path, have been considered for each condition. Finally for the last trial a general-shape trajectory was also considered, reporting also more results in term of parameters estimation.

#### 4.1. Experiment 1: Self-Calibrating EKF with GPS Signal Loss

Taking into account the real position and yaw angle of the mobile robot, at the beginning we started the path prediction and correction with the self-calibrating EKF. At a certain time, we simulated a loss of the GPS signal by interrupting the GPS updating phase of the filter, in order to reconstruct robot path only with encoders and AHRS platform data, taking into account also the estimation of robot parameters, which are also the last three elements of the state vector:  $R_L(k)$ ,  $R_R(k)$ ,  $L(k)$ . The Figures 3–6 show the path reconstruction without GPS updating phase of EKF (red path), highlighting the advantages of the offset estimation. All the measurements are reported in meters.

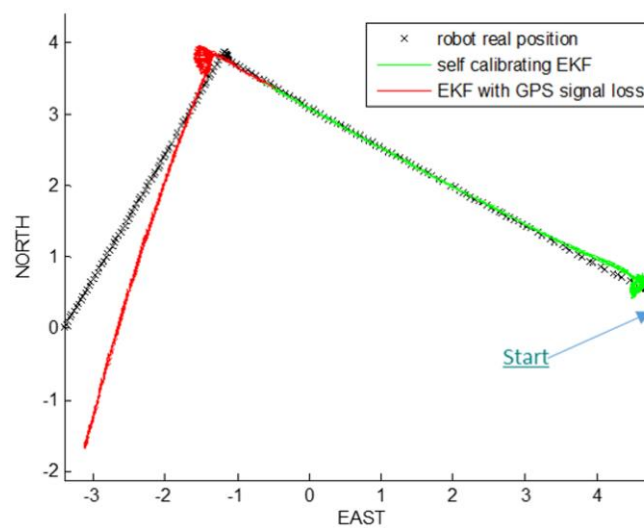


Figure 3. Experiment 1/rectangular path: Self-Calibrating Extended Kalman Filter (SC EKF).

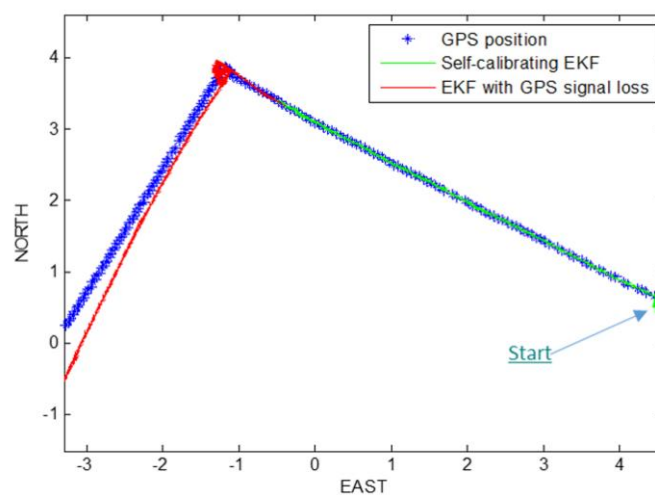
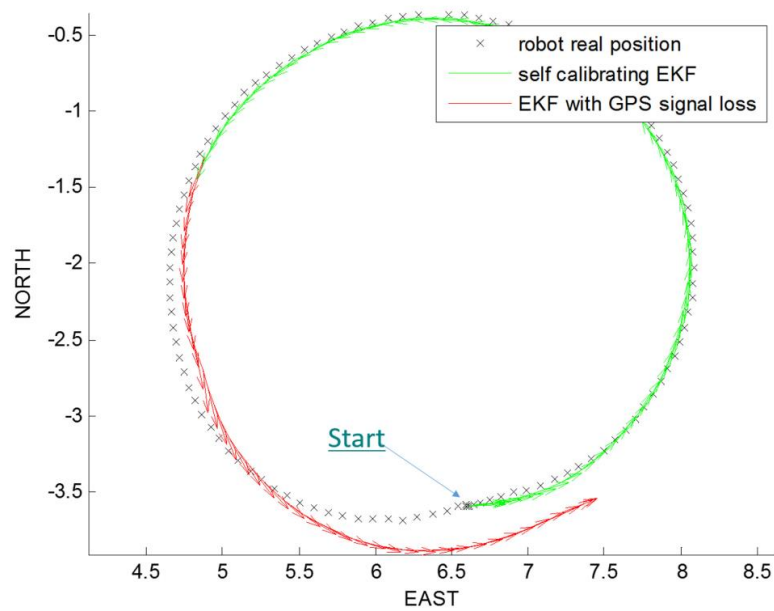
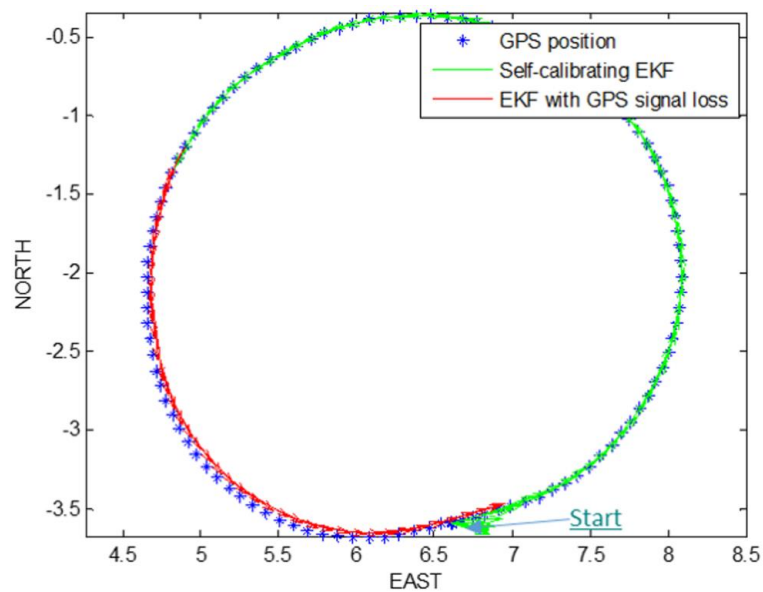


Figure 4. Experiment 1/rectangular path: SC EKF with offset estimation.



**Figure 5.** Experiment 1/circular path: SC EKF.



**Figure 6.** Experiment 1/circular path: SC EKF with offset estimation.

#### 4.2. Experiment 2: Decrease of GPS Signal Quality Simulations ( $\sigma_{noise} = 0.25\text{ m}$ )

Taking into account the real position and yaw angle of the mobile robot, at beginning we start the path prediction and correction with both the self-calibrating EKF. At a certain time, we add a white Gaussian noise, with  $\sigma_{noise} = 0.25\text{ m}$  (GPS fix quality = 2), and so affect the GPS position coordinates, in order to simulate a decrease in the GPS signal quality. In this way, EKF has to reconstruct robot path and to estimate robot parameters, taking into account the added noise. The Figures 7–10 show the noise addition in the GPS data, and the results of both algorithms.

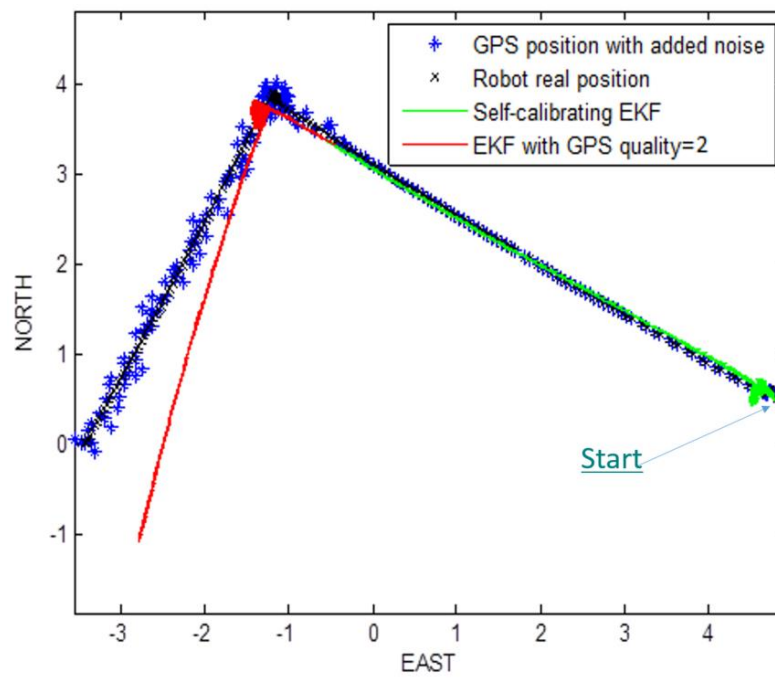


Figure 7. Experiment 2/rectangular path: SC EKF.

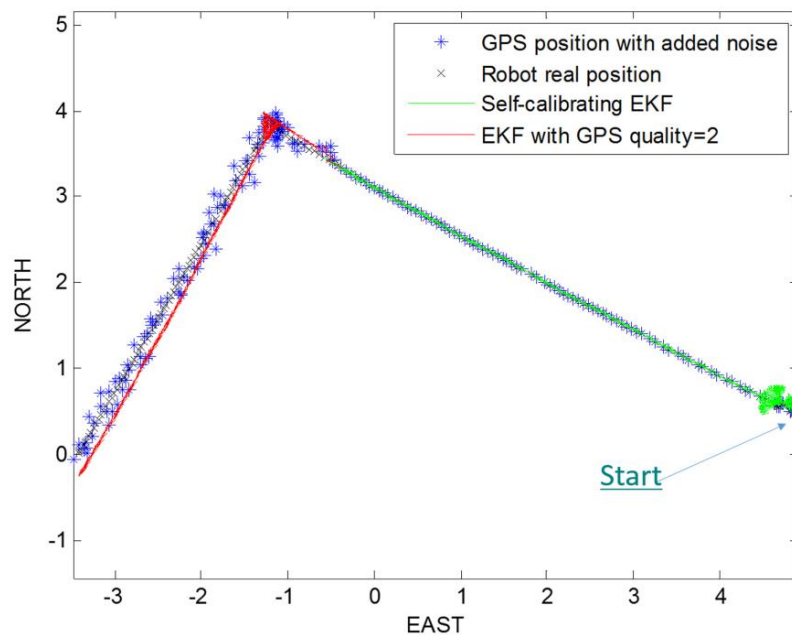
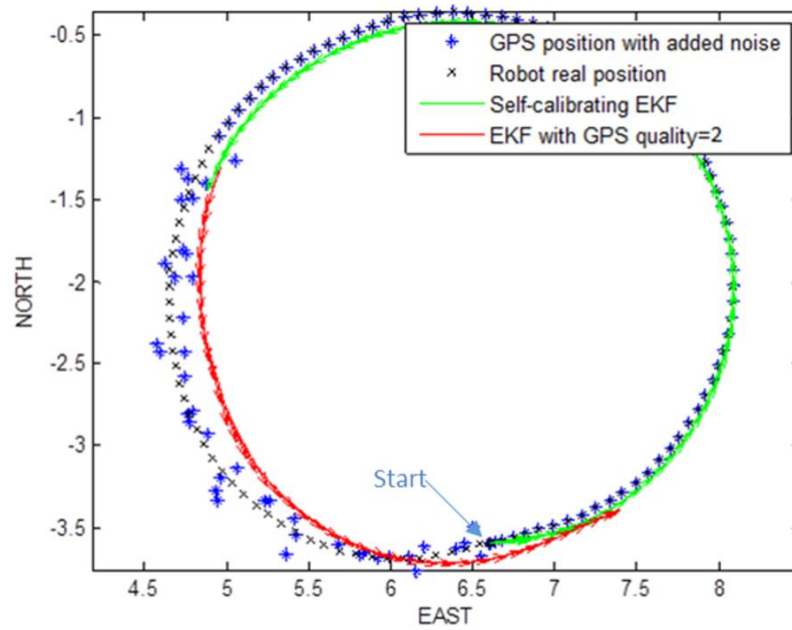
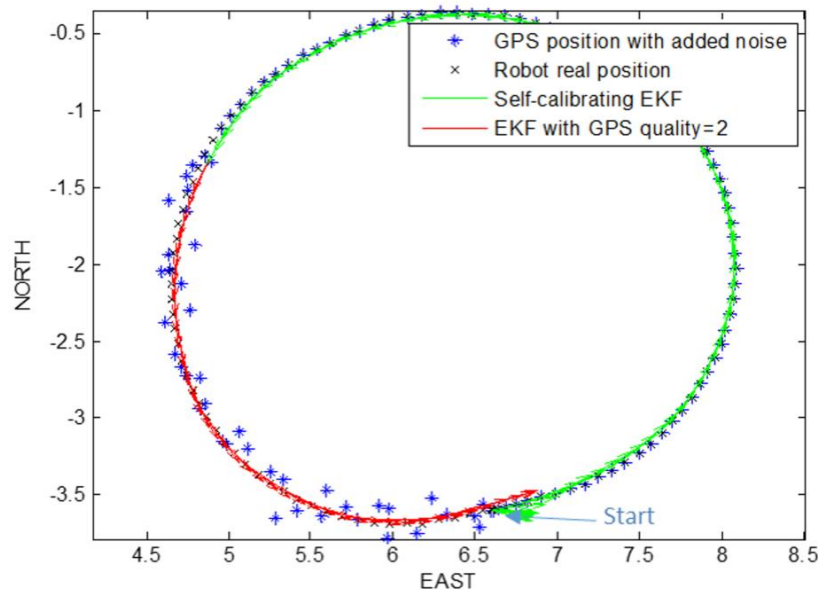


Figure 8. Experiment 2/rectangular path: SC EKF with offset estimation.





**Figure 9.** Experiment 2/circular path: SC EKF.



**Figure 10.** Experiment 2/circular path: SC EKF with offset estimation.

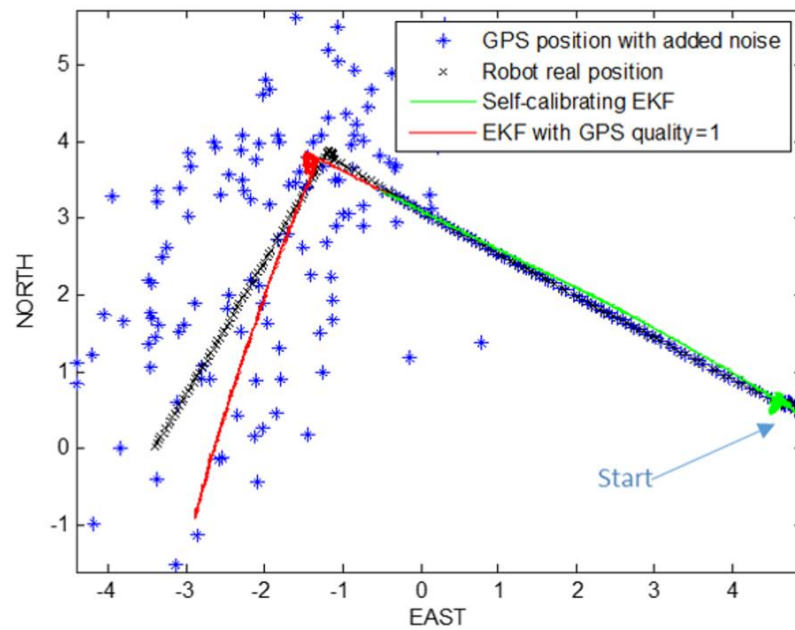
#### 4.3. Experiment 3: Self-Calibrating EKF with Offset Estimation and a Decrease of GPS Signal Quality Simulations ( $\sigma_{noise} = 5\text{ m}$ )

In this experiment a large white Gaussian noise, with ( $\sigma_{noise} = 5\text{ m}$ ), was added to GPS measurements (GPS fix quality = 1).

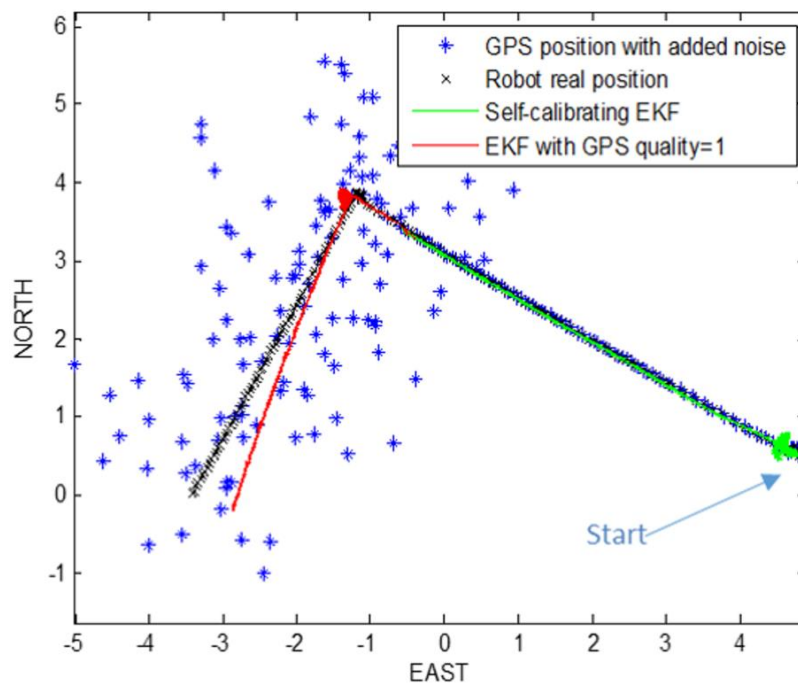
Also in this case the algorithm does not only estimate only the robot parameters  $R_L(k)$ ,  $R_R(k)$ ,  $L(k)$ , but also the angle offset,  $Ofs(k)$ , that we want to compensate. In this way, the trajectory reconstruction is made by considering this term for each step, with an improvement of the performance.

The Figures 11–14 show the estimated trajectories for rectangular and circular path. Finally the methods have been tested in a general-shape path where there are several changes of direction (see the Figure 15). The Figures 16 and 17 show the performance of the methods, without and with

offset estimation respectively, while Figures 18–21 show the estimated parameters in the last case. Observe how, after a short transient, the parameters rapidly converge.



**Figure 11.** Experiment 3/rectangular path: SC EKF.



**Figure 12.** Experiment 3/rectangular path: SC EKF with offset estimation.

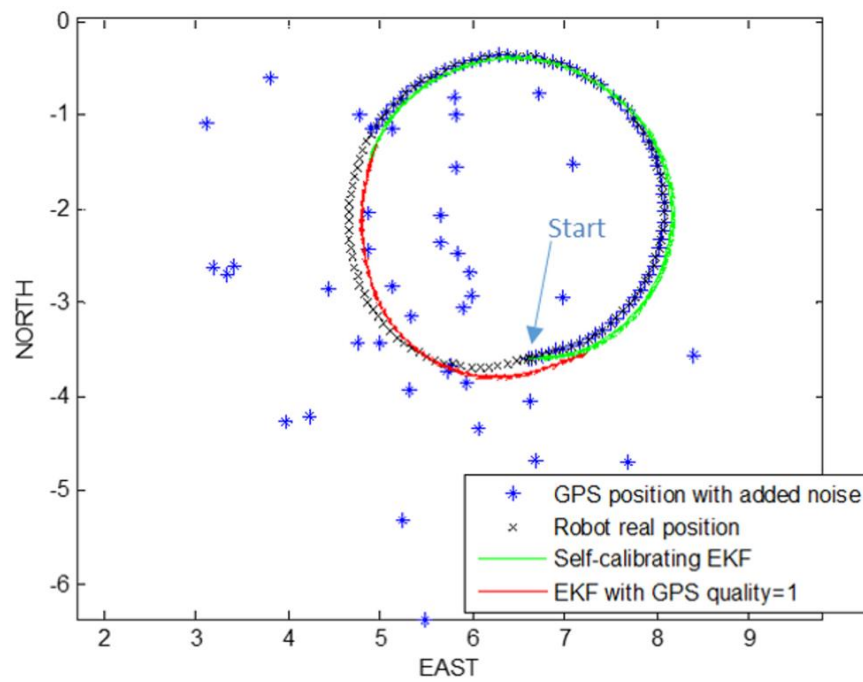


Figure 13. Experiment 3/circular path: SC EKF.

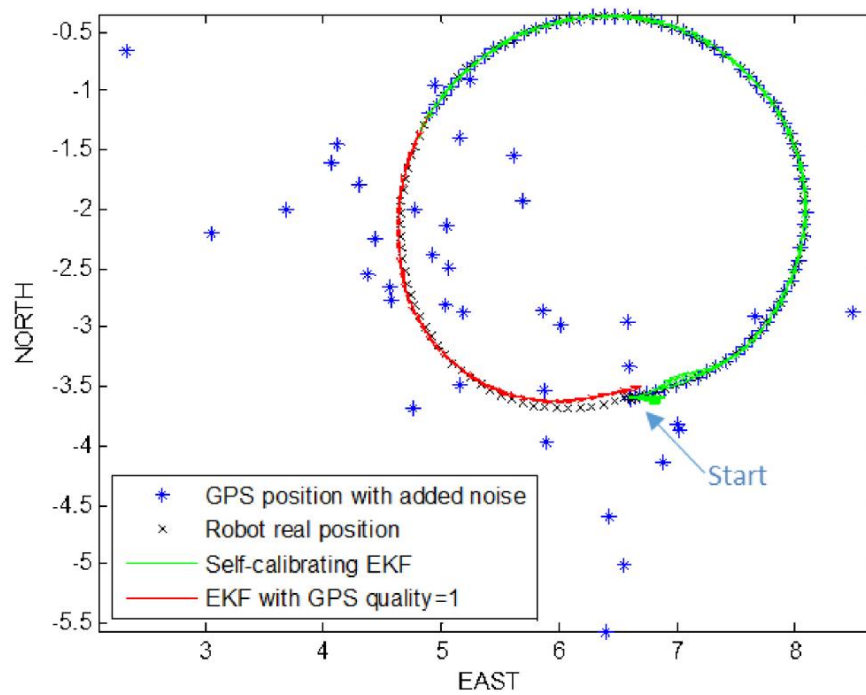
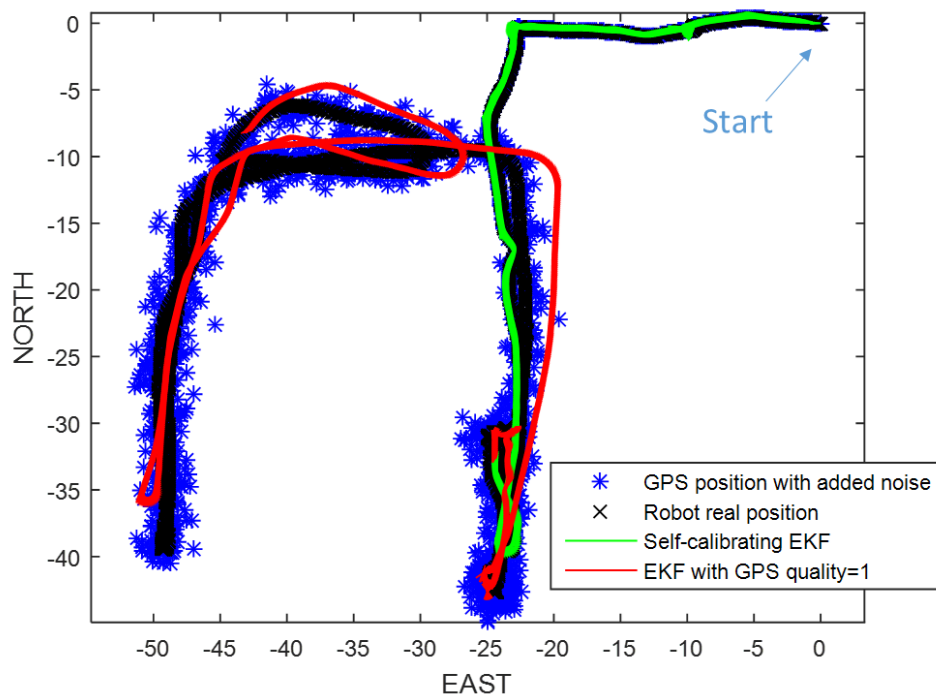


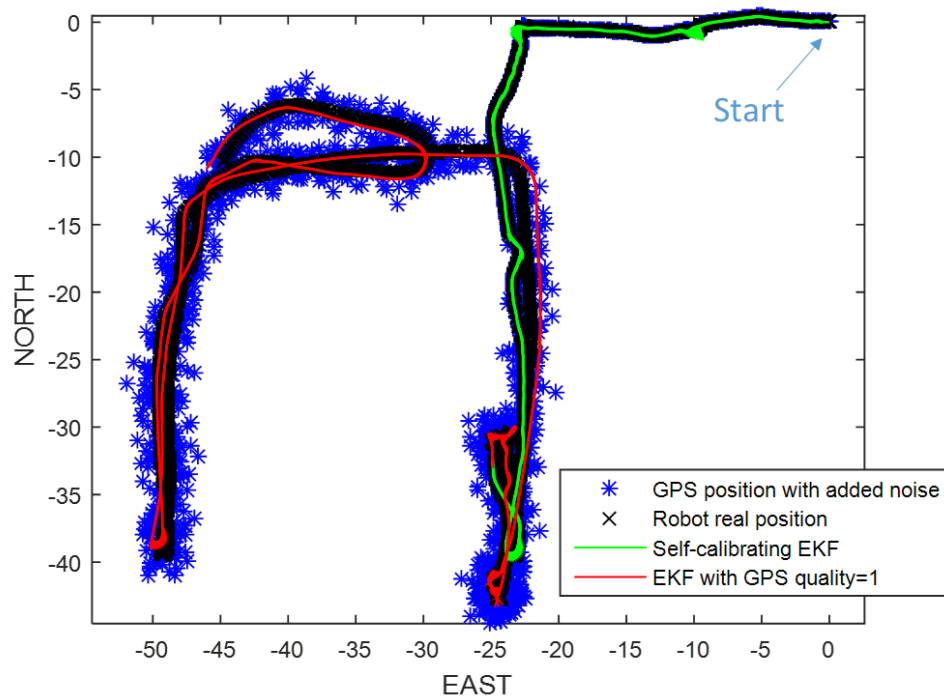
Figure 14. Experiment 3/circular path: SC EKF with offset estimation.



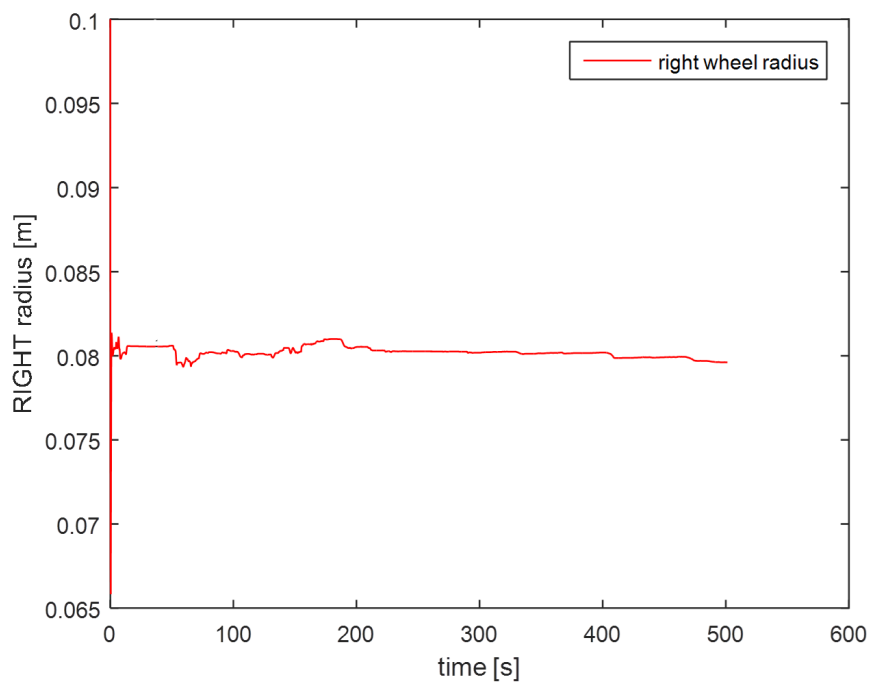
**Figure 15.** Experiment 3/general-shape path: multiple waypoints trajectory on the external area of DIEEI laboratories of the University of Catania.



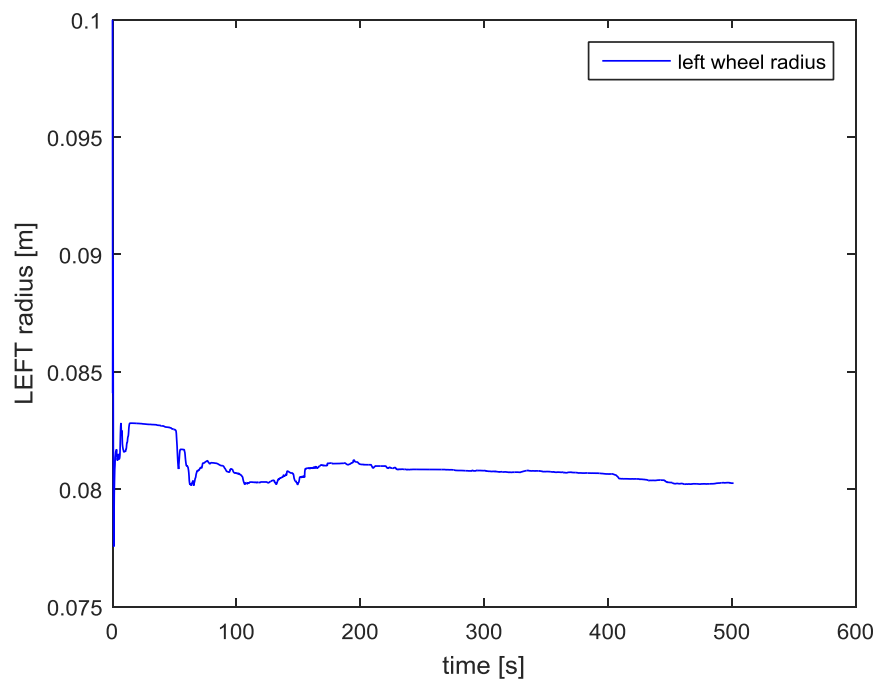
**Figure 16.** Experiment 3/general-shape path: SC EKF.



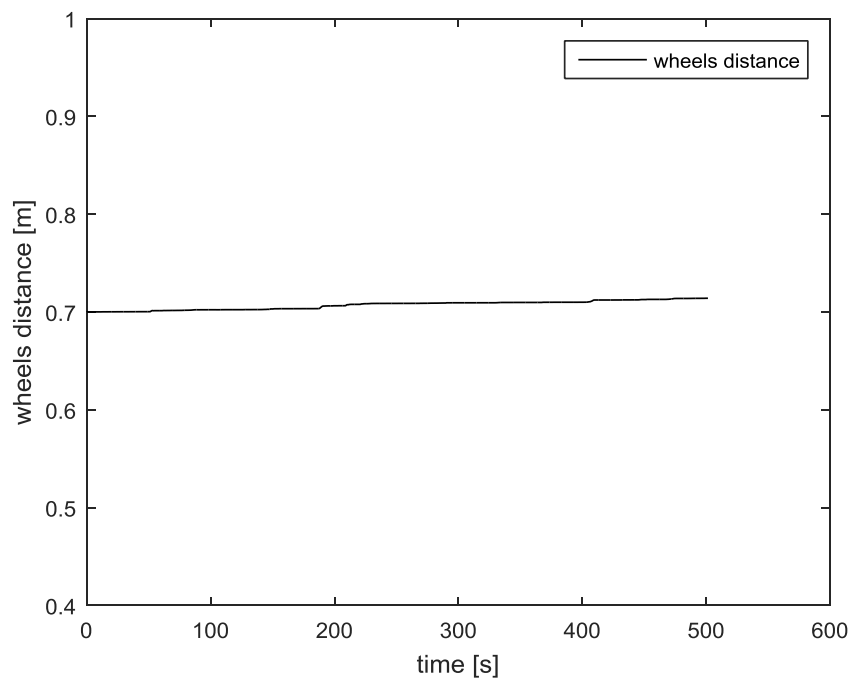
**Figure 17.** Experiment 3/general-shape path: SC EKF with offset estimation.



**Figure 18.** Experiment 3/general-shape path: right wheel radius estimation.



**Figure 19.** Experiment 3/general-shape path: left wheel radius estimation.



**Figure 20.** Experiment 3/general-shape path: wheel distance estimation.



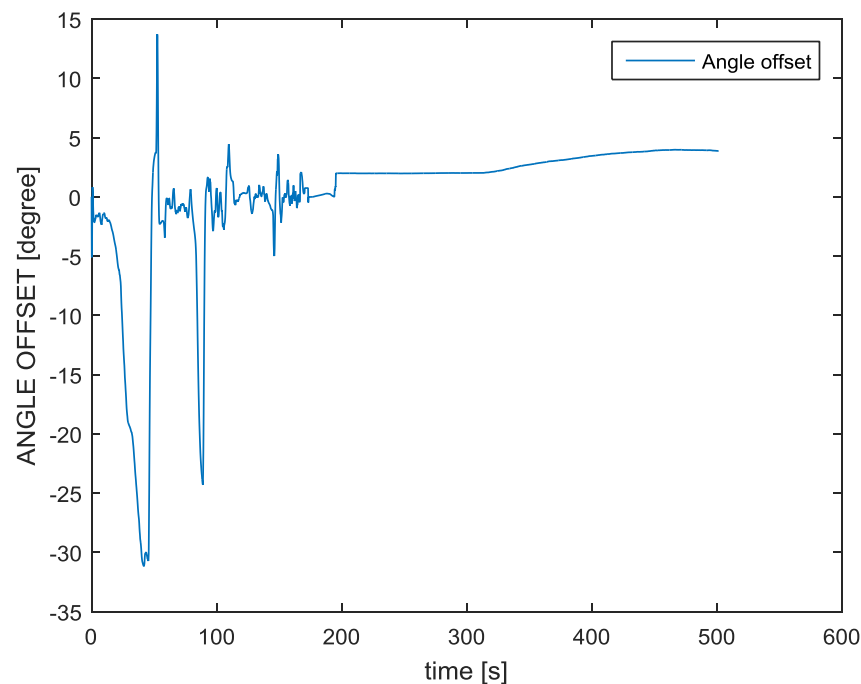


Figure 21. Experiment 3/general-shape path: Offset estimation.

## 5. Conclusions

From a careful analysis of the results, it is possible to observe, the great improvement of the self-calibrating EKF algorithm, in terms of path reconstruction, by adding the estimation and compensation of the robot orientation angle offset. It is possible to observe, also, the good estimation ability of robot parameters, which values are, at the end, very close to their measured dimensions:

$$R_r = 0.08 \text{ m}; R_l = 0.08 \text{ m}; L = 0.7 \text{ m}$$

Taking into account the influence of the magnetic structures near or inside the robot to the magnetometer, the basic self-calibrating EKF showed several drawbacks. Indeed, looking at the kinematic model of the robot (used in the EKF algorithm), it is possible to see that the angle term—the orientation of the robot—is present in  $X$  and  $Y$  position equations. This means that if we have errors about yaw information, its position will also have a wrong result, because simulating a GPS signal loss, or a GPS signal quality decrease, the algorithm should reconstruct robot trajectory only taking into account encoders data. However, the odometry is not sufficient, because it is low precise, considering errors depending by sliding behavior of tracks, digital sampling, and so on. In addition, the uncertainty of odometric measurements grows with the distance. Therefore, taking into account a failure or low-quality of the GPS signal in the algorithm, it is necessary to estimate, step by step, this disturbance value, in order to compensate it and to obtain robot position reconstruction with more precision.

The proposed algorithm allows reducing such errors and the experimental results presented confirm the suitability of the proposed approach. The presented results, by introducing the compensation of the magnetometer offset extend and improve the method of [5,9].

**Acknowledgments:** This work was carried-out in the framework of TIRAMISU project. This project is funded by the European Community's Seventh Framework Programme (FP7/2007–2013) under grant No. 284747.

**Author Contributions:** G.M. and L.C. conceived and designed the experiments; S.L. and D.S. implemented and performed the trials; All authors analyzed the data; L.C., S.L. and G.M. wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Baudoin, Y. TIRAMISU: FP7-Project for an Integrated Toolbox in Humanitarian Demining, Focus on UGV, UAV, Technical Survey and Close-in Detection. In Proceedings of the International Conference on Climbing and Walking Robots, CLAWAR 2013, Sydney, Australia, 14–17 July 2013.
2. TIRAMISU. Available online: <http://www.fp7-tiramisu.eu/> (accessed on 29 October 2016).
3. Marques, L.; de Almeida, A.T.; Armada, M.; Fernández, R.; Montes, H.; González, P.; Baudoin, Y. State of the Art Review on Mobile Robots and Manipulators for Humanitarian Demining. In Proceedings of the 10th International IARP Workshop HUDEM'2012, Šibenik, Croatia, 24–26 April 2012; p. 1.
4. Portugal, D.; Marques, L.; Armada, M. Deploying Field Robots for Humanitarian Demining: Challenges, Requirements and Research Trends. In Proceedings of the 17th International Conference on Climbing and Walking Robots (CLAWAR 2014), Poznan, Poland, 21–23 July 2014; pp. 21–23.
5. Caltabiano, D.; Muscato, G.; Russo, F. Localization and Self Calibration of a Robot for Volcano Exploration. In Proceedings of the 2004 IEEE International Conference on Robotics & Automation, New Orleans, LA, USA, 26 April–1 May 2004; pp. 586–591.
6. Censi, A.; Franchi, A.; Marchionni, L.; Oriolo, G. Simultaneous calibration of odometry and sensor parameters for mobile robots. *IEEE Trans. Robot.* **2013**, *29*, 475–492. [[CrossRef](#)]
7. Roy, N.; Thrun, S. Online Self-Calibration for Mobile Robots. In Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, MI, USA, 10–15 May 1999; pp. 2292–2297.
8. Pei, Y.; Lindsay, K. Online robot odometry calibration over multiple regions classified by floor colour. In Proceedings of the 2015 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 2–5 August 2015.
9. Von der Hardt, H.J.; Husson, R.; Wolf, D. An Automatic Calibration Method for a Multisensor System: Application to a Mobile Robot Localization System. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, 20 May 1998; pp. 3141–3146.
10. Martinelli, A.; Tomatis, N.; Tapus, A.; Siegwart, R. Simultaneous Localization and Odometry Calibration for Mobile Robot. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 27–31 October 2003; pp. 1499–1504.
11. Yi, J.; Zhang, J.; Song, D.; Jayasuriya, S. IMU-based localization and slip estimation for skid-steered mobile robots. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 2845–2850.
12. Yi, J.; Wang, H.; Zhang, J.; Song, D.; Jayasuriya, S.; Liu, J. Kinematic Modeling and Analysis of Skid-Steered Mobile Robots with Applications to Low-Cost Inertial-Measurement-Unit-Based Motion Estimation. *IEEE Trans. Robot.* **2009**, *25*, 1087–1097.
13. Tang, H.; Liu, Y.; Li, L. Simultaneous calibration and mapping for mobile robot with non-holonomic constraint. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 1543–1549.
14. Kümmerle, R.; Grisetti, G.; Burgard, W. Simultaneous parameter calibration, localization, and mapping. *Adv. Robot.* **2012**, *26*, 2021–2041. [[CrossRef](#)]
15. Grewal, M.S.; Andrews, A.P. *Kalman Filtering Theory and Practice*, 4th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2014.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).