

Mobile Robot Localization Using On-board and Off-board Laser Range Finders and Reflective Markers

Henrique Carvalho

henrique.melo.de.carvalho@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2017

Abstract

The application of mobile robots is a reliable solution for autonomous transportation in industry, as well as in nuclear facilities to transport radioactive loads. However, the common solutions used in industry based on on-board sensors can be compromised by the exposure to activated loads. A redundant solution based on off-board sensors can improve the reliability of the robots localisation means. The sensors can be used independently as two separate systems, one on-board and other off-board, or they can be used together as single system. The addressed sensors are the Laser Range Finders and the landmarks are reflective markers, which can both be installed on the environment and on the vehicle. The focus of this work are the localisation algorithms and their performance. The addressed algorithms are: the Extended Kalman Filter, the Unscented Kalman Filter and the Monte Carlo Localisation, since they are the most used and available solutions in the literature. However, the novelty of this work is to combine these algorithms with on-board and off-board sensors. The motion model used is specific for a vehicle with two driving and steerable wheels, the developed method does not lose generality for other vehicles as long as they move in a two dimensional environment. The developed methods assume marker matching without error and the detection of markers with any incident angle of the laser. Simulation results were obtained for scenarios with a layout similar to the basement level of the International Thermonuclear Experimental Reactor. Nevertheless, the proposed solution can be also applied to industrial warehouse scenarios.

Keywords: Laser Range Finder, Reflective Markers, Extended Kalman Filter, Unscented Kalman Filter, Monte Carlo Localisation

1. Introduction

Common AGVs have on-board sensors for self localisation and navigation. This solution is the most used nowadays since it requires little modifications to the surrounding environment. Moreover, the sensor-controller connection can be easily wired, which makes it completely autonomous in navigation matters.

The “traditional” solution issue is the exposure of the sensor to high energy radiation. The electronics in the most commonly used sensors relies strongly on semi-conductor materials, making them vulnerable to the loads radiation. An extended exposure might render the sensors unreliable or even inoperable, leaving the robot devoid of its crucial perception of the world.

The vulnerability of the electronics to radiation leads to another solution, distributing sensors over the facility, off-board of the vehicle. If the sensors are placed in the scenario, they are less exposed to the load which can extend their life and improve reliability. A big disadvantage is that in order to

control based on this localisation method, there is the need of a wireless data connection. In case of connection failure, the AGV is left with no means of localisation besides increasingly inaccurate predictions.

The sweeping Laser Range Finders are expected to resist more in this high energy radiation hostile environment. The electronics of the LRF can be shielded with lead which can stop most of the dangerous radiation. To make measurements, a mirror system can be used to guide the laser pulses outside the box and capture its echo. By combining on- and off-board LRFs, the localisation system should be more robust and fail-safe.

Although the proposed localisation methods were developed in the ITER scope, they have application in the industry. Such methods can be used in industrial warehouses as long as it is possible to ensure a plain ground and the disposition of reflective markers. It can be used as a primary or secondary localisation method (for redundancy).

2. Vehicle and Sensors

This section introduces the Rhombic vehicle, the Laser Range Finder sensors and their models.

2.1. Rhombic Vehicle

The vehicle has a Rhombic configuration of wheels, i.e. two driven and steerable wheels. Figure 1 shows the configuration of a Rhombic vehicle. This configuration was designed to allow high manoeuvrability in relatively small spaces.

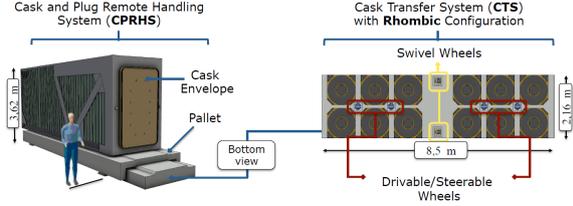


Figure 1: Cask and Plug Remote Handling System. Source: [1].

The state space of this vehicle is defined as

$$\mathbf{x} = [x \ y \ \theta]^T, \quad (1)$$

where (x, y) are the abscissa and ordinate of the robot's geometric centre in the global (world) frame and θ is the heading or orientation. As seen in the schematic of Figure 2, L_R and L_F are the distance from the geometric centre to the rear and front wheels respectively, L is the sum of both distances, θ_R and θ_F are the rear and front wheels angles, v_F and v_R are their linear speeds.

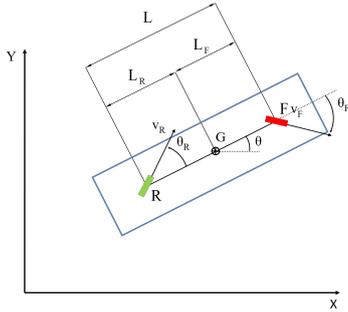


Figure 2: Dimensions and actuation of the Rhombic vehicle. Source: [2].

The discrete motion model expressed in (2) is based on the kinematic model proposed by [2]. This model is used by the three developed filters.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + (A_F v_F + A_R v_R) T_s, \quad (2)$$

$$A_F = \begin{pmatrix} \frac{L_R}{L} \cos(\theta + \theta_F) \\ \frac{L_R}{L} \sin(\theta + \theta_F) \\ \frac{\sin(\theta_F)}{L} \end{pmatrix} \quad (3)$$

$$A_R = \begin{pmatrix} \frac{L_F}{L} \cos(\theta + \theta_R) \\ \frac{L_F}{L} \sin(\theta + \theta_R) \\ -\frac{\sin(\theta_R)}{L} \end{pmatrix} \quad (4)$$

However the simulation engine used in this work has an extra dynamic which is not considered by the filters. The linear speeds and angles of the wheels can not vary instantaneously, so the model used by the simulator considers an acceleration as described by (5).

$$\begin{bmatrix} v_{F,k} \\ v_{R,k} \\ \theta_{F,k} \\ \theta_{R,k} \end{bmatrix} = \begin{bmatrix} (1 - \tau_v) v_{F,k-1} \\ (1 - \tau_v) v_{R,k-1} \\ (1 - \tau_\theta) \theta_{F,k-1} \\ (1 - \tau_\theta) \theta_{R,k-1} \end{bmatrix} + \begin{bmatrix} \tau_v v_{F,k-1}^* \\ \tau_v v_{R,k-1}^* \\ \tau_\theta \theta_{F,k-1}^* \\ \tau_\theta \theta_{R,k-1}^* \end{bmatrix} \quad (5)$$

The model of the motion model's adds noise ε the control inputs

$$\mathbf{u}_k(\varepsilon_k) = \begin{bmatrix} v_{F,k} + \varepsilon_{v_F,k} \\ v_{R,k} + \varepsilon_{v_R,k} \\ \theta_{F,k} + \varepsilon_{\theta_F,k} \\ \theta_{R,k} + \varepsilon_{\theta_R,k} \end{bmatrix}, \quad (6)$$

where $\varepsilon_{v_F,k}$, $\varepsilon_{v_R,k}$, $\varepsilon_{\theta_F,k}$ and $\varepsilon_{\theta_R,k}$ are independent random variables with a Gaussian distribution. Thus, $\varepsilon = [\varepsilon_{v_F,k} \ \varepsilon_{v_R,k} \ \varepsilon_{\theta_F,k} \ \varepsilon_{\theta_R,k}]^T$ has a diagonal covariance.

The noise simulation is done according to kinematic constraints also described by [2] and translated in (7).

$$v_R \cos(\theta_R) = v_F \cos(\theta_F). \quad (7)$$

The noise is added to the controls \mathbf{u} and calculated as in (8), where one of the control inputs must be function of the other three.

$$v_R = (v_F + \varepsilon_{v_F}) \frac{\cos(\theta_F + \varepsilon_{\theta_F})}{\cos(\theta_R + \varepsilon_{\theta_R})} \quad (8)$$

To avoid singularities in (8) is used a saturation at 80 deg in the noise corrupted inputs $\theta_F + \varepsilon_{\theta_F}$ and $\theta_R + \varepsilon_{\theta_R}$. The linear speed of the wheels is also saturated at 0.2 m/s by design [3].

2.2. Laser Range Finder

The sensors considered in the scope of this work take multiple measurements in a known azimuthal aperture. The beam is emitted and reflected by a rotating mirror, orienting it in different azimuthal angles, all equally spaced. As in Figure 3, let d_i denote the distance to the target and φ_i the angle in which the measurement i was taken. The index i refers to which beam took the distance measurement in a given sweep. $\Delta\varphi$ is the angular spacing between consecutive measurements.

The considered reflectors are rectangular strips of high reflectivity material. The devices are assumed

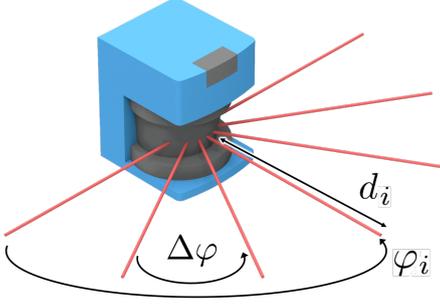


Figure 3: LRF measurement schematic. A LRF takes equally angularly spaced distance measurements in an azimuth. d_i is the distance and $\Delta\varphi$ the angular resolution.

to have a length of 10 cm by design. Then, the markers can be approximated by their geometric centre. However, if the marker is long, the error of associating each ray to the centre grows.

In the model of the sensor, the distance and angle are corrupted with additive Gaussian white noise as in (9). In (10), R is the covariance matrix of the noise. For simplification, the distance and angle are independent and therefore R is diagonal.

$$\begin{bmatrix} d_i \\ \varphi_i \end{bmatrix} = \begin{bmatrix} D_i \\ \Phi_i \end{bmatrix} + \begin{bmatrix} \delta_{d,i} \\ \delta_{\varphi,i} \end{bmatrix}, \begin{bmatrix} \delta_{d,i} \\ \delta_{\varphi,i} \end{bmatrix} \sim \mathcal{N}([0 \ 0]^T, R) \quad (9)$$

$$R = \begin{bmatrix} \text{Var}(d) & 0 \\ 0 & \text{Var}(\varphi) \end{bmatrix} \quad (10)$$

3. Localisation Algorithms

This section details the observation model and how the three filters work.

3.1. Observation Model

The sensor model produces a distance and angle measurement which is assumed to be the centre of a marker. This marker is also correctly matched. To obtain the observation function of an on-board LRF, the already known deterministic position of the marker is converted to the robot's frame from the world frame, and then to the LRF's frame as depicted in Figure 4. The frame transformation can be achieved by using regular two-dimensional rigid body transformations (11) or its inverse. Here, \mathbf{f}_{bT} is a function which transforms coordinates $(^a x, ^a y)$ in the frame a to the coordinates $(^b x, ^b y)$ in the frame b . $(^b x_a, ^b y_a)$ are the coordinates of the a frame's origin in the frame b and θ is the angle between the two frames.

$$\mathbf{f}_{bT} = \begin{bmatrix} ^b x \\ ^b y \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & ^b x_a \\ \sin(\theta) & \cos(\theta) & ^b y_a \end{bmatrix} \begin{bmatrix} ^a x \\ ^a y \\ 1 \end{bmatrix}, \quad (11)$$

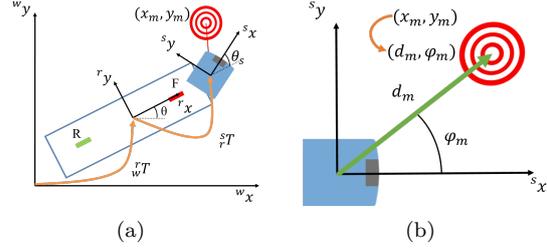


Figure 4: On-board LRF observation model. In 4(a) are represented the three coordinate frames: the world frame, the robot frame and the sensor frame. The observation model transforms the coordinates of the marker from the world frame to the robot's frame and from there to the sensor frame. The transformations are represented with orange arrows. Finally, in 4(b) the Cartesian coordinates are converted to polar.

Once the coordinates of the marker are obtained in the sensor's frame, they are converted to polar coordinates with the function $\mathbf{f}_{C \rightarrow P}$ in (12).

$$\mathbf{f}_{C \rightarrow P}(^s x_m, ^s y_m) = \begin{bmatrix} \sqrt{^s x_m^2 + ^s y_m^2} \\ \text{atan}(\frac{^s y_m}{^s x_m}) \end{bmatrix} \quad (12)$$

The on-board LRF observation function is the composition of the two rigid body transformations and the conversion to polar coordinates in Figure 4. The function is described by (13).

$$\mathbf{h}_{\text{on-board}}(\mathbf{x}) = \begin{bmatrix} ^s d_m \\ ^s \varphi_m \end{bmatrix} = \mathbf{f}_{C \rightarrow P} \circ \mathbf{f}_{rT}^s \circ \mathbf{f}_{wT}^r(\mathbf{x}) \quad (13)$$

The process to obtain the off-board LRF observation function is similar to the an on-board LRF. The rigid body transformations are presented in Figure

The off-board LRF observation function is given by (14).

$$\mathbf{h}_{\text{off-board}}(\mathbf{x}) = \begin{bmatrix} ^s d_m \\ ^s \varphi_m \end{bmatrix} = \mathbf{f}_{C \rightarrow P} \circ \mathbf{f}_{wT}^s \circ \mathbf{f}_r^w(\mathbf{x}) \quad (14)$$

3.2. Extended Kalman Filter

The Extended Kalman Filter is an extension of the Kalman Filter for non-linear systems. It linearises the dynamics of the system and the observation in order to propagate the first and second moments of the a state which follows a Gaussian distribution.

The non-linear state space considered by the EKF is given by (15) and (16).

$$\mathbf{x}_{k+1} = g(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\varepsilon}_k) \quad (15)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\delta}_k \quad (16)$$

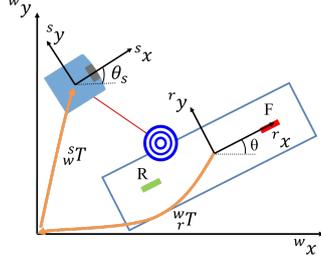


Figure 5: Off-board LRF observation model. There are three coordinate frames represented: the world frame, the robot frame and the sensor frame. The observation model transforms the coordinates of the marker from the robot's frame to the world's frame and from there to the sensor frame. The transformations are represented with orange arrows. The transformation to polar coordinates is also needed, as previously shown in Figure 4(b).

The EKF is initialised at \mathbf{x}_0 with an infinite covariance P_0 . As demonstrated by [4], the equations (17), (18) and (19) rule the EKF's prediction step.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{g}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, \hat{\boldsymbol{\varepsilon}}_k) \quad (17)$$

$$G_k^{(\mathbf{x})} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Big|_{\mathbf{x} = \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u} = \mathbf{u}_k, \boldsymbol{\varepsilon}_k = 0} \quad (18)$$

$$P_{k|k-1} = G_k^{(\mathbf{x})} P_{k-1|k-1} (G_k^{(\mathbf{x})})^T + Q_k \quad (19)$$

$$Q_k = G_k^{\boldsymbol{\varepsilon}} \text{Cov}_{\boldsymbol{\varepsilon}_k} (G_k^{\boldsymbol{\varepsilon}})^T \quad (20)$$

$$G_k^{\boldsymbol{\varepsilon}} = \frac{\partial \mathbf{g}}{\partial \boldsymbol{\varepsilon}} \Big|_{\mathbf{x} = \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u} = \mathbf{u}_k, \boldsymbol{\varepsilon}_k = 0} \quad (21)$$

In (20) and (21), $G_k^{(\mathbf{x})}$ and $G_k^{(\boldsymbol{\varepsilon})}$ are de partial Jacobians of $g(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\varepsilon}_k)$ in order to the state variables \mathbf{x} and noise $\boldsymbol{\varepsilon}$, respectively. They are given by

$$G_k^{(\mathbf{x})} = \begin{bmatrix} \frac{\partial \mathbf{g}_{(1)}}{\partial x_{(1)}} & \cdots & \frac{\partial \mathbf{g}_{(1)}}{\partial x_{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{(n)}}{\partial x_{(1)}} & \cdots & \frac{\partial \mathbf{g}_{(n)}}{\partial x_{(n)}} \end{bmatrix} \quad (22)$$

and

$$G_k^{(\boldsymbol{\varepsilon})} = \begin{bmatrix} \frac{\partial \mathbf{g}_{(1)}}{\partial \varepsilon_{(1)}} & \cdots & \frac{\partial \mathbf{g}_{(1)}}{\partial \varepsilon_{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{(n)}}{\partial \varepsilon_{(1)}} & \cdots & \frac{\partial \mathbf{g}_{(n)}}{\partial \varepsilon_{(n)}} \end{bmatrix}. \quad (23)$$

For the correction step, [4] derived the equations (24), (25), (26) and (27).

$$H(k) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \Big|_{\mathbf{x} = \hat{\mathbf{x}}_{k|k-1}, \mathbf{u} = \mathbf{u}_k} \quad (24)$$

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1} H_k P_{k|k-1} \quad (25)$$

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (26)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})) \quad (27)$$

The Jacobian $H(k)$ is calculated using the differentiation chain rule (28).

$$\frac{d}{d\mathbf{a}} \mathbf{f}_b \circ \mathbf{f}_a(\mathbf{a}) = \frac{d}{d\mathbf{b}} \mathbf{f}_b \left(\mathbf{f}_a(\mathbf{a}) \right) \frac{d}{d\mathbf{a}} \mathbf{f}_a(\mathbf{a}) \quad (28)$$

Applying it to the on-board LRF observation function one has (30).

$$\begin{aligned} \frac{d}{d\mathbf{x}} \mathbf{f}_{C \rightarrow P} \circ \mathbf{f}_{s^T} \circ \mathbf{f}_{r^T}(\mathbf{x}) &= \quad (29) \\ \frac{d}{d [s_x^T \ s_y^T]^T} \mathbf{f}_{C \rightarrow P} \left(\mathbf{f}_{r^T} \circ \mathbf{f}_{w^T}(\mathbf{x}) \right) \cdots \\ \frac{d}{d r_{\mathbf{x}_s}} \mathbf{f}_{r^T} \cdots \\ \frac{d}{d\mathbf{x}} \mathbf{f}_{w^T}(\mathbf{x}) \end{aligned}$$

The same for the off-board LRF, resulting (??).

$$\begin{aligned} \frac{d}{d\mathbf{x}} \mathbf{f}_{C \rightarrow P} \circ \mathbf{f}_{w^T} \circ \mathbf{f}_{r^T}(\mathbf{x}) &= \quad (30) \\ \frac{d}{d [s_x^T \ s_y^T]^T} \mathbf{f}_{C \rightarrow P} \left(\mathbf{f}_{w^T} \circ \mathbf{f}_{r^T}(\mathbf{x}) \right) \cdots \\ \frac{d}{d^w \mathbf{x}_s} \mathbf{f}_{w^T} \cdots \\ \frac{d}{d\mathbf{x}} \mathbf{f}_{r^T}(\mathbf{x}) \end{aligned}$$

3.3. Unscented Kalman Filter

The Unscented Kalman Filter is also an extension of the Kalman Filter for non-linear systems. The UKF uses the Unscented Transform instead of linearising the system. This transform calculates sigma points based on the latest state estimate, propagates them through the non-linear system and observation models and then approximates the resulting points with a normal distribution. The UKF uses the concept of an augmented state space with N dimensions

$$\mathbf{x}^{(a)} = [\mathbf{x} \ \boldsymbol{\varepsilon}]^T, \quad (31)$$

in which are included the noise variables. The dimension of the state is given by

$$N_k = N^{(\mathbf{x})} + N^{(\boldsymbol{\varepsilon}_k)} + N^{(\boldsymbol{\delta}_k)}, \quad (32)$$

the sum of the dimensions of the state, system noise and observation noise. Consequently, the covariance matrix is also expanded, but the noises of the considered system do not have statistical dependence on each other. Therefore, the augmented covariance is block diagonal

$$P_k^a = \begin{bmatrix} P_k & 0 & 0 \\ 0 & \text{Cov}_{\boldsymbol{\varepsilon}_k} & 0 \\ 0 & 0 & \text{Cov}_{\boldsymbol{\delta}_k} \end{bmatrix}. \quad (33)$$

The sigma points are generated based on the dimension of the augmented state. If P_k^a is block diagonal like (33), then the sigma points are separable in three sets. Let \mathcal{X} be the set of $1 + 2N$ points to sample the state space and its noise. The set of sigma points is

$$\mathcal{X}_{k-1} = \left\{ \mathbf{x}_{k-1}, \mathbf{x}_{k-1} \pm \left(\sqrt{(N + \lambda) P_k^a} \right)_i \right\}, \quad (34)$$

where $\left(\sqrt{(N + \lambda) P_k^a} \right)_i$ is the i -th column or row of the matricial square root and $i = 1, \dots, N$.

In [5] the prediction is proposed as in Equations (35) through (41). As this filter is an heuristic of the KF, the notation used in literature is rather informal. In (35), every sigma point is individually propagated through the system model $\mathbf{g}(\mathbf{x}_{k-1}, \mathbf{u}_k, \boldsymbol{\varepsilon}_k)$, as done with the mean for the EKF in (27). The dimension of the $2N_k + 1$ sigma points in $\mathcal{X}_{k|k-1}^{(\mathbf{x})}$ is $N^{(\mathbf{x})}$. The predicted pose is calculated in (36) with a weighted mean of these propagated points. The weights of iteration k are calculated for each sigma point l . The weights of the mean are given by $w_{k,l}^{(m)}$, while the weights of the covariance are given by $w_{k,l}^{(c)}$. The weights for the sigma point zero specified in (38) and (39) are different from the other points' weights. The terms α , β and γ are tuning knobs of the filter. The primary point spreading term should stay in the interval $\alpha \in [0, 1]$. The term β is a real positive should incorporate the higher moments information in the covariance of the estimation. The term γ in (41) is also a positive real number and is used as a secondary spreading factor. The prediction covariance is calculated in (37), incorporating the error of the previous estimate and the noise of the control inputs.

$$\mathcal{X}_{k|k-1}^{(\mathbf{x})} = \mathbf{g}(\mathcal{X}_{k-1}^{(\mathbf{x})}, \mathbf{u}_k, \mathcal{X}_k^{(\boldsymbol{\varepsilon})}) \quad (35)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{l=0}^{2N} w_{k,l}^{(m)} \mathcal{X}_{k|k-1,l}^{(\mathbf{x})} \quad (36)$$

$$P_{k|k-1} = \sum_{l=0}^{2N} w_l^{(c)} (\mathcal{X}_{k|k-1,l}^{(\mathbf{x})} - \hat{\mathbf{x}}_{k|k-1}) (\mathcal{X}_{k|k-1,l}^{(\mathbf{x})} - \hat{\mathbf{x}}_{k|k-1})^T \quad (37)$$

$$w_{k,0}^{(m)} = \frac{\lambda_k}{N_k + \lambda_k} \quad (38)$$

$$w_{k,0}^{(c)} = \frac{\lambda_k}{N_k + \lambda_k} + 1 - \alpha^2 + \beta \quad (39)$$

$$w_{k,l}^{(m)} = w_{k,l}^{(c)} = \frac{1}{2(N_k + \lambda_k)}, \quad l = 1, \dots, 2N \quad (40)$$

$$\lambda_k = \alpha^2(N_k + \gamma) - N_k \quad (41)$$

The correction equations of [5] are given by Equations (42) through (47).

$$\mathcal{Y}_k = \mathbf{h}(\mathcal{X}_{k|k-1}^{(\mathbf{x})}, \mathbf{u}_k, \mathcal{X}_k^{(\boldsymbol{\delta})}) \quad (42)$$

$$\hat{\mathbf{y}}_k = \sum_{l=0}^{2N} w_{k,l}^{(m)} \mathcal{Y}_{k,l} \quad (43)$$

$$P_k^{(\mathbf{y}\mathbf{y})} = \sum_{l=0}^{2N} w_{k,l}^{(c)} (\mathcal{Y}_{k,l} - \hat{\mathbf{y}}_k) (\mathcal{Y}_{k,l} - \hat{\mathbf{y}}_k)^T \quad (44)$$

$$P_k^{(\mathbf{x}\mathbf{y})} = \sum_{l=0}^{2N} w_{k,l}^{(c)} (\mathcal{X}_{k|k-1,l}^{(\mathbf{x})} - \hat{\mathbf{x}}_{k|k-1}) (\mathcal{Y}_{k,l} - \hat{\mathbf{y}}_k)^T \quad (45)$$

$$K_k = P_k^{(\mathbf{x}\mathbf{y})} (P_k^{(\mathbf{y}\mathbf{y})})^{-1} \quad (46)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} - K_k (\mathbf{y}_k - \hat{\mathbf{y}}_k) \quad (47)$$

$$P_{k|k} = P_{k|k-1} - K_k P_k^{(\mathbf{y}\mathbf{y})} K_k^T \quad (48)$$

In (42), the propagation of sigma points is analogous to (35) with the observation noise as specified for the EKF in (16). The points from (42) are then used to calculate the observation error covariance in (44) and (45). The Kalman gain is calculated in (46), which is used to calculate the new estimate (47) and its error covariance (48).

3.4. Monte Carlo Localisation

The Monte Carlo Localisation method is composed by three steps. The first is the sampling of the motion model, when the new pose for each particle m in the set $\bar{\mathcal{X}}_k$ is calculated. In this process, it is added a different random noise to the vehicle's inputs of each particle. The second step is based on the observation model and the particle's pose. By combining the information of the pose and observation it is given to each particle a weight proportional to its likelihood,

$$w^{[m]} \propto P(\mathbf{y}|\mathbf{x}). \quad (49)$$

The third step is the re-sampling stage, when all the M particles are drawn from the particle set with

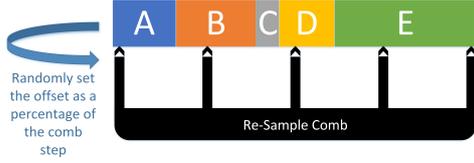


Figure 6: Stochastic Universal Sampling. The letters A,B,C,D and E are the particles designation or naming. The rectangle widths are proportional to the weights. The teeth are equally spaced and each tooth tip indicates the selected particle.

a probability proportional to its weight. The particles with poses that produce estimated observations closer to the real observation are more likely to survive the re-sampling stage.

The motion model sampling concept begins with considering the vehicle’s pose stochastic. It has an associated probability density function (pdf), which might be too complex to derive analytically. The model is based on the vehicle’s discrete kinematic model. To sample it, will be added Gaussian noise to the inputs as in (6).

The particle weight should be proportional to its likelihood. If the observations are independent, then the probability of all observations is the product of every single observation as in (50).

$$w^{(i)} = P(\mathbf{y}|\mathbf{x}) = \prod_{n=0}^{N-1} P(\mathbf{y}^{(n)}|\mathbf{x}^i) \quad (50)$$

In the case of the considered observation model, the weight is the product of the pdf values for the Gaussian distributions of each marker observation. For all N ray-marker pairs, the weight of a particle is given by

$$w^{(i)} = \prod_{n=0}^{N-1} \mathcal{N}(err_{(i,n)}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, R_n), \quad (51)$$

where $err_{(i,n)}$ is the error calculated in the sensor s frame, of the pair n and for the particle i given by

$$err_n = \begin{bmatrix} s d_{m(n)} \\ s \varphi_{m(n)} \end{bmatrix} - \begin{bmatrix} s \hat{d}_i \\ s \hat{\varphi}_i \end{bmatrix}. \quad (52)$$

The re-sample stage uses the Stochastic Universal Sampling (SUS) algorithm, illustrated in Figure 6. It works by disposing the weights in a strip like in Figure 6. The step to iterate is given by

$$step = \frac{\sum_{j=1}^J weight^{(j)}}{J}. \quad (53)$$

The particles will be selected at fixed intervals, like shown in Figure 6. The particles will be selected at fixed intervals, like shown in Figure 6.

Algorithm $SUS(\bar{\mathcal{X}}_k, w_k)$

```

totalWeight =  $\sum_{m=0}^{M-1} w_k^{(m)}$ 
step = totalWeight/M
pointer = step * randNumBetween(0,1)
currentWeight = 0
j = 0
for  $w_k^{[i]}$  in  $w_k$  do
    currentWeight +=  $w_k^{[i]}$ 
    while currentWeight  $\geq$  pointer do
         $\mathcal{X}_k^{[j]} \leftarrow \bar{\mathcal{X}}_k^{[i]}$ 
        pointer += step
        j += 1
    end while
end for
return  $\mathcal{X}_k$ 

```

4. Results

The results of the simulation are presented in this chapter. The standard deviation values for the inputs ε_{vF} and ε_{vR} are proportional to the linear speeds v_F and v_R , while $\varepsilon_{\theta F}$ and $\varepsilon_{\theta R}$ are constant. The covariance matrix used for the EKF and UKF are given in (54), where the standard deviation of the linear speeds are 20% and the orientation of the wheels have a standard deviation of 10 deg. The MCL uses a covariance matrix $3 \text{Cov}(\varepsilon_k)$ as it was unable to follow to vehicle with the covariance used by EKF and UKF.

$$\text{Cov}(\varepsilon_k) = \begin{bmatrix} (0.2v_{F,k})^2 & 0 & 0 & 0 \\ 0 & (0.2v_{R,k})^2 & 0 & 0 \\ 0 & 0 & (10 \frac{\pi}{180})^2 & 0 \\ 0 & 0 & 0 & (10 \frac{\pi}{180})^2 \end{bmatrix} \quad (54)$$

The sensors have a scan frequency of 10 Hz. The distance measurements have an error with 2 cm standard deviation. The variance of the angular error is considered a normal distribution with variance equal to a normal distribution with 1 degree length. The UKF covariance was obtained empirically by observing the filter estimations. In Figure 7 are labelled the elements present in a simulation window. The trajectories followed in the “L shaped scenario” and the “Level B1 scenario” are shown in Figure 8.

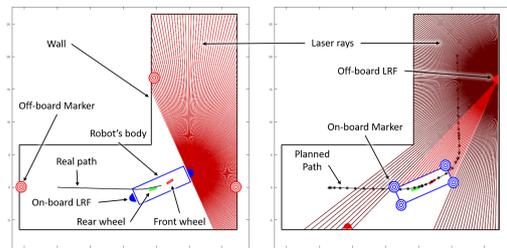


Figure 7: On the left a set-up of on-board LRFs, on the right a set-up with off-board LRFs.

The control algorithm used to follow the trajectory is proposed by [6] and further developed by [7].

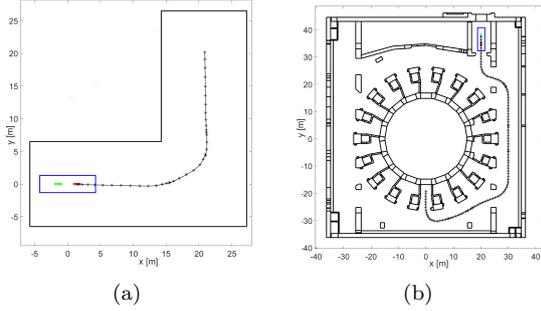


Figure 8: 8(a) is the trajectory for the L scenario and 8(b) for the Level B1 at ITER.

For the experiments will be shown graphics with the pose error (x_{err} , y_{err} , θ_{err}), the area of the ellipses and variance of θ , the total position error with the ellipses area and the orientation error with its variance.

4.1. “L” Shaped Scenario

The followed path as well as the iterations from the filter are shown in Figure 9. The position error for all filter is given in Figure 10 and the orientation error in Figure 11.

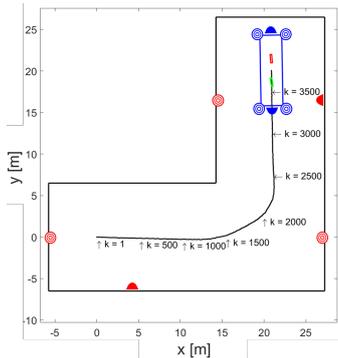
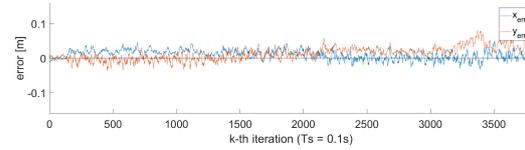


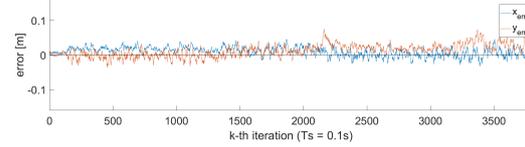
Figure 9: Path followed in the L scenario. The iterations of the filters are indicated with arrows.

From the iteration $k = 3000$ to $k = 3802$, the robot is in a particular blind spot, where the marker on the left of the robot is not seen by the on-board sensors and the marker on the right-behind is far and has few sightings. The off-board sensor rays are approximately parallel to the on-board markers providing a few sightings due to the small section of the markers as depicted in Figure 14. In the end of the graphic the robot is approximating the end of the trajectory where it stops. The linear speed of the wheels decreases and the non-modelated dynamic affects the filters. The robot is still on the blind spot when it comes to a halt and the filters rely only on predictions. The filters assume that the vehicle stopped while it keeps moving forward, increasing its y coordinate. Hence, y_{err} grows to

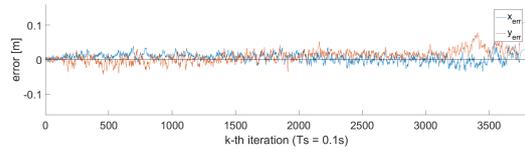
positive values and stabilises.



(a) EKF



(b) UKF



(c) MCL

Figure 10: Position error for the three filters in the L scenario with on- and off-board LRFs.

The performance of the three filters can be compared using Table 4.1. The metrics shown are the mean of absolute value for the each error and total position error $\|(x_{err}, y_{err})\|$. The performance of the EKF and UKF are on par, but the MCL is better than both Kalmans. The absolute position error graphs are shown in Figure 12.

Table 1: Mean error in the L scenario.

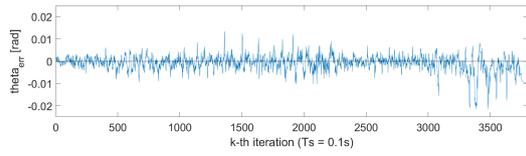
[cm]	$ x_{err} $	$ y_{err} $	$ \theta_{err} [^\circ]$	$\ (x_{err}, y_{err})\ $
EKF	1.42	1.88	0.146	2.62
UKF	1.38	1.78	0.143	2.50
MCL	1.09	1.62	0.137	2.13

4.2. Real Scenario: Level B1 of the Reactor

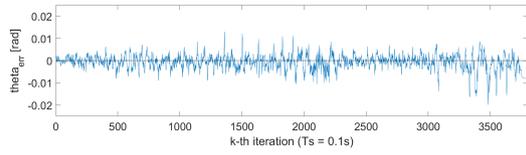
At the ITER B1 level is used the off-board laser configuration which ensured a coverage of 97% of the area according to [8]. The off-board sensors coverage can be seen in Figure 15(a). The distribution of sensors and markers is shown in Figure 15(b).

The performances of the three filters are compared using sensors on-board, off-board and both. In this scenario the values presented are the average of the three runs for each configuration of sensors. The error metrics of the on-board and off-board LRF 3 experiments is in Table 2, for the on-board LRF are in Table 3 and for off-board in Table 4.

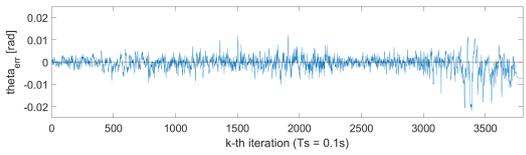
The errors, as expected, are lower with all sensors active. When the off-board sensors are disabled, the



(a) EKF

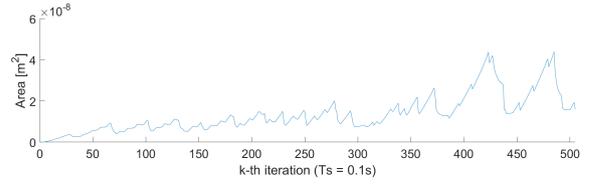


(b) UKF

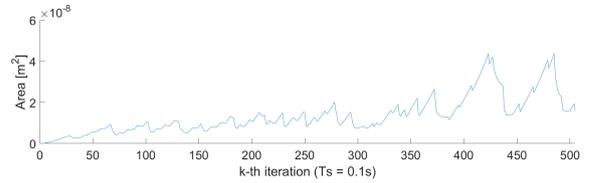


(c) MCL

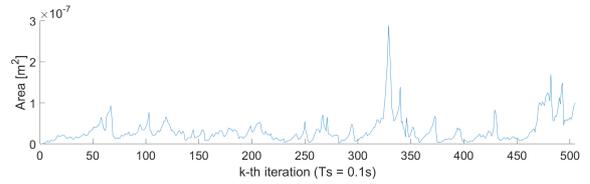
Figure 11: Orientation error for the three filters in the L scenario with on- and off-board LRFs.



(a) EKF

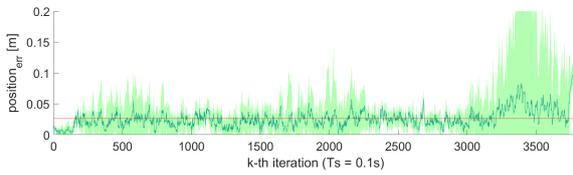


(b) UKF

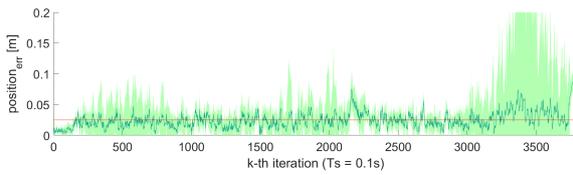


(c) MCL

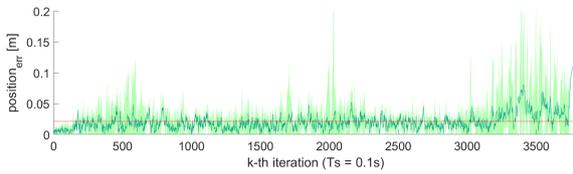
Figure 13: Area of the uncertainty ellipses with off-board LRF in the L scenario.



(a) EKF



(b) UKF



(c) MCL

Figure 12: In blue the position error, red the mean value and the green shade is proportional to the covariance of the estimation for the three filters in the L scenario with on- and off-board LRFs.

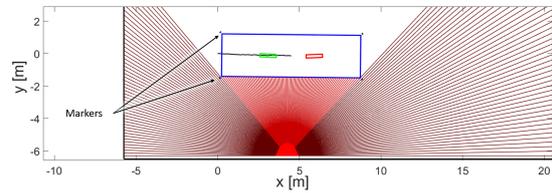


Figure 14: In this situation the effective area of the markers is minimal and the sensors provides few sightings.

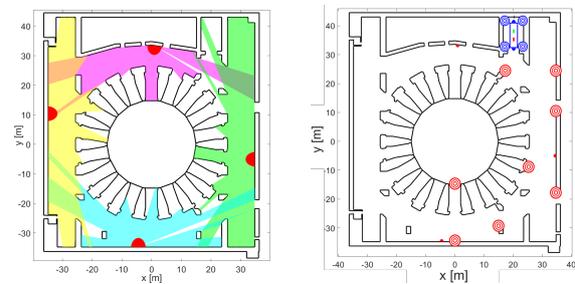


Figure 15: (a) Coverage of the off-board markers. (b) LRFs and reflective markers.

error does not grow significantly. On one hand, this suggests that the distribution of off-board markers provide enough information. On the other hand, the off-board sensors do not contribute significantly for the localisation.

The values of position error in Table 4 are two orders of magnitude above the other filters. The MCL could not track the vehicle close enough while exiting the elevator in the beginning of the trajectory. When the off-board sensors start to see the the vehicle, the Kalmans converge to the real position. However, the MCL can not recover without the implementation of heuristics for global positioning.

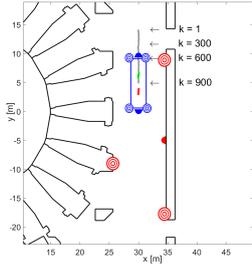


Figure 16: Path followed in the experiment of sensors failure. The iterations are indicated.

Table 2: Mean error B1 using on & off-board LRF.

[cm]	$ x_{err} $	$ y_{err} $	$ \theta_{err} [^\circ]$	$\ (x_{err}, y_{err})\ $
EKF	1.34	1.77	0.126	2.46
UKF	1.36	1.78	0.131	2.46
MCL	1.19	1.62	0.117	2.21

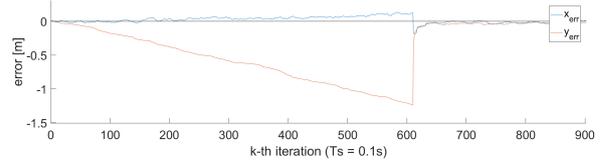
Table 3: Mean error in B1 using on-board LRF.

[cm]	$ x_{err} $	$ y_{err} $	$ \theta_{err} [^\circ]$	$\ (x_{err}, y_{err})\ $
EKF	1.45	2.26	0.111	2.90
UKF	1.60	2.28	0.120	3.03
MCL	1.36	1.84	0.099	2.51

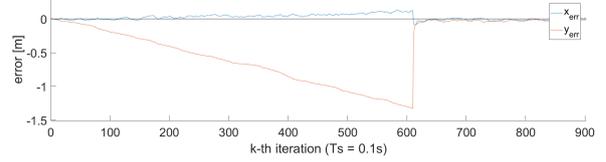
The next experiment illustrates the recovery of the three filters in case of temporary sensors failure. The path followed and iterations of the filters are shown in Figure 16. The plots of position error are shown in Figure 17 and the orientation error in Figure 18.

5. Conclusions

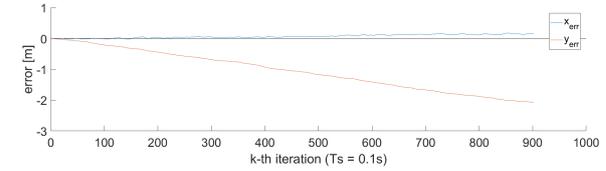
The three compared algorithms were previously applied in similar applications, but the combination of on-board and off-board LRFs with reflective markers are novel in the context of AGVs. Although the focus of the work was set on ITER, this solution can be transported to other environments of industrial warehouses.



(a) EKF

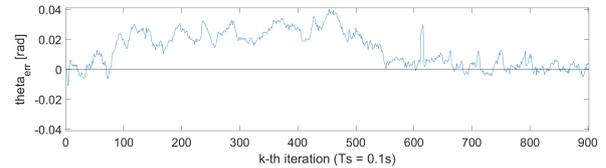


(b) UKF

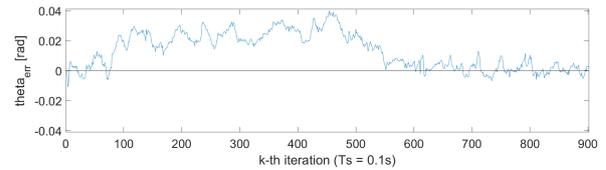


(c) MCL

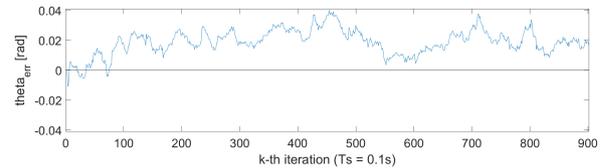
Figure 17: Position error for the three filters in sensor failure scenario in B1 level. Both on- and off-board sensors are used. The sensors are turned off approximately until $k = 600$.



(a) EKF



(b) UKF



(c) MCL

Figure 18: θ error for the three filters in sensor failure scenario in B1 level. Both on- and off-board sensors are used. The sensors are turned off approximately until $k = 600$.

Table 4: Mean error in B1 using off-board LRF.

[cm]	$ x_{err} $	$ y_{err} $	$ \theta_{err} [^\circ]$	$\ (x_{err}, y_{err})\ $
EKF	4.38	5.44	0.587	7.8
UKF	3.82	4.77	0.495	6.92
MCL	226.74	777.28	3.004	824.07

The model of the LRF and observation provide good corrections as the filters are able to track the robot with the precision of 2 – 3 centimetres. It was presented a qualitative and quantitative comparison of the three algorithms. The MCL provides the estimates with the least average error. However, in conditions of worse LRF coverage, namely when only the off-board LRFs are used, the MCL is unable to track the position of the vehicle. The model is not close enough to the simulated vehicle and the predictions without sensor updates are insufficient. This problem can be solved with heuristics to spread more the particles, like increasing the noise covariance in the motion model sampling, or increasing the speed and steer of the wheels on half of the particles. The UKF does not present a great improvement when compared to the EKF, although there is an improvement in the experiment using only off-board sensors. Overall, both the EKF and the UKF proved reliable. The UKF is easier to implement as it is not necessary to calculate Jacobians. The Unscented Transform does not require much more computational power when comparing with the EKF. The estimation of the UKF has the potential to be better than the EKF, but it relies strongly on the tuning parameters. The dependence of the MCL on heuristics might be a drawback regarding safety and reliability. However, the particle filter has better precision with good sensor coverage.

In the localisation scope, the EKF is the first option. Even though there is the need to calculate Jacobians, this filter worked out-of-the box without major problems. If the precision of the EKF does not meet the requirements, then the UKF can be developed in second place. The covariance matrices from the EKF can be used. The tuning of the UKF can then be done empirically by comparison with the EKF or using other optimisation methods. The MCL can be used for greater precision, or if the marker matching reveals to be inefficient with the Kalman filters. The most important issue yet to solve is the data association problem. In this work, the matching of markers seen by the sensors to the map is assumed ideal. If the marker association is done wrongly, it can have a major impact on the localisation algorithm. The Kalman filters do not have tools to deal with wrong matches. Both the EKF and the UKF do not consider an event

of incorrect matching. On the contrary, the MCL deals with this problem naturally. If the marker matching is done for each particle, the MCL considers multiple hypothesis which are discarded or not depending on their likelihood. Besides the data association problem, the detection of a marker in the intensity signal provided by the LRF is also an important matter. The marker detection could rely on the localisation estimate besides the intensity value.

For the B1 scenario, the off-board markers layout was planned so that there was always a marker to be sighted by the on-board LRF. The off-board LRFs were distributed by optimising the coverage of the corridors. This metric does not consider the influence of their measurements in the localisation. The distribution of sensors and markers can be done regarding their contribution to the localisation algorithms. Instead of coverage, the optimisation can use observability metrics for non-linear systems.

References

- [1] Alberto Vale and Isabel Ribeiro. Mobile robot navigation for remote handling operations in iter. Available at http://www.ipfn.tecnico.ulisboa.pt/rh/publications/AVale_Robot2011.pdf (last accessed on 10/11/2016), 2011.
- [2] Daniel Fonte. Motion Planning for a Rhombic-Like Vehicle Operating in ITER Scenarios. Master’s thesis, Instituto Superior Tecnico (IST), June 2011.
- [3] Filipe Valente, Alberto Vale, Daniel Fonte, and Isabel Ribeiro. Optimized trajectories of the transfer cask system in iter. *Fusion Engineering and Design*, 86(9):1967 – 1970, 2011. Proceedings of the 26th Symposium of Fusion Technology (SOFT-26).
- [4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. MIT Press, 2005.
- [5] Rambabu Kandepu, Bjarne Foss, and Lars Imsland. Applying the unscented kalman filter for nonlinear state estimation. *Journal of Process Control*, 18(7):753 – 768, 2008.
- [6] Alonzo Kelly. A vector algebra formulation of kinematics of wheeled mobile robots. Technical Report CMU-RI-TR-10-33, The Robotics Institute, Carnegie Mellon University, September 2010. Technical Report.
- [7] Nuno Silva, Alberto Vale, and Luca Baglivo. Path following control of rhombic like vehicles - performance assessment with dynamic vehicle model. In *Proceedings of the 10th International*

Conference on Informatics in Control, Automation and Robotics, pages 48–57, 2013.

- [8] Joo Filipe Teles Ferreira. Sensor network and localization methods for a mobile robot. Master's thesis, Instituto Superior Técnico (IST), October 2010. Honorable mention from the Portuguese Robotics Society award in 2010.