Using the Extended Information Filter for Localization of Humanoid Robots on a Soccer Field



Tobias Garritsen University of Amsterdam Faculty of Science BSc Artificial Intelligence

Using the Extended Information Filter for Localization of Humanoid Robots on a Soccer Field

Tobias Garritsen 10779582

Bachelor thesis Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam Faculty of Science Science Park 904 1098 XH Amsterdam

> Supervisor Prof. dr. A. Visser

Informatics Institute Faculty of Science University of Amsterdam Science Park 904 1098 XH Amsterdam June 29th, 2018

Contents

1	Introduction	4
2	Theory2.1Localization2.2Gaussian filters2.3Kalman filter2.4Information Filter2.5Extended Kalman/Information Filter	5 6 6 7 8
3	Related Work	9
4	Proposed Method 4.1 Accuracy and Computational Complexity 4.2 Soccer pitch 4.3 Dataset 4.4 Implementation 4.4.1 Landmark Recognition 4.4.2 Extended Information and Kalman Filter 4.4.3 Optimization	10 10 11 11 11 11 14 14
5	Results5.1Accuracy Results5.2Computational Complexity Results	15 15 15
6	Discussion and Conclusion	21
7	Future Research	21
8	Acknowledgements	21
A	ppendices	24
Α	Hardware/Software A.1 Hardware	24 24 24
в	Comparison of localization algorithms	24

Abstract

On of the contests of the RoboCup is Nao robot soccer. For Nao robots to play soccer, they need to be able to localize themselves on the field. The extended information filter (EIF) is an algorithm that is not yet researched on this problem, as opposed its dual, the extended Kalman filter (EKF). The EIF is computational less complex than the EKF in the second part, but vice versa in first part of the algorithm. Both filters require a robust landmark detector to work efficiently. There have been many studies regarding robot localization and the use of variations of the EKF in Nao robot soccer, but not yet of the EIF. In this thesis it is found that the EKF requires less computation time that the EIF, but the EIF has a better accuracy. The EKF and the EIF used the same data, were tested on the same computer, and have a similar form of implementation, thus the only factors that played a role in their comparison, where the filters themselves.



Figure 1: A Nao robot that is used for the SPL

1 Introduction

Robot soccer is a prominent part of the RoboCup¹, which is an annual international robotics competition. The RoboCup Standard Platform League $(SPL)^2$ is the soccer competition in which Nao robots³ (figure 1) are used. The highest goal of the RoboCup is to create humanoid robots that are able to defeat real professional soccer players. For this goal to be achieved, multiple problems must be solved by the soccer robots. One of the most significant problems is robot localization [Dellaert et al., 1999], which is the problem of determining the position of a robot within a certain known environment. The robot has a priori knowledge of the surroundings and by retrieving additional knowledge about them, it should determine its location. The known environment should be well recognizable and thus a number of landmarks should be used, in addition to them being stationary and having recognizable features. The soccer robots have to operate on a (small) soccer pitch, where the landmarks can be the goals, the lines and artificially added landmarks.

In this research, there will be artificially added landmarks, namely colorful and recognizable cones. For the robot to determine its location based on these landmarks, multiple algorithms have been researched. However the information filter, has not yet been studied and thus it will be tested on the localization problem and discussed in this paper.

In main focus of this thesis is researching if the use of the extended information filter for robot localization in the SPL is an interesting alternative to the extended Kalman filter. There will firstly be an overview of the basic idea of the Kalman and information filter, and some of their extensions and afterwards there will be an overview of related work. In the second part, a method will be proposed for solving the localization problem with the information filter after which the implementation will be discussed. Following the implementation, the

¹http://www.robocup.org/

²http://www.spl.robocup.org/

³https://www.softbankrobotics.com/emea/en/robots/nao

testing and test results will be discussed and a comparison between the Kalman filter and the information filter will be compared with each other. Finally a conclusion will be given and there will be a discussion of possible future work.

2 Theory

The basic theory about localization and the two Gaussian filters and their extended versions is explained in the book *Probabilistic Robotics* by Thrun [Thrun et al., 2005] of which a summary has been given below.

2.1 Localization

Localization is determining a robots pose on the field, whereas the pose consists of its actual position in Cartesian coordinates (x and y) as well as the orientation θ of the robot. The x and y coordinates of the pose will further on be called the location. Other possible variables considering the pose of the robot are not used in this thesis research.

There are multiple dimensions in localization, with two dimensions being important within this thesis. The first dimensions regards the local view of the robot, in which the robot knows its initial pose and it should be able to keep track of it. To determine the pose of the robot after it moves, the motion noise should be taken into account. This is a local problem, as the uncertainty about its pose is local and thus the estimated pose can only be close to the robots original position. The robot should also operate in a static environment, in which the robot is the only entity that is changing its position, with all other objects remaining in the same pose. This creates an easier environment for the robot to test the information filter with little noise from other sources, outside of the robot itself.

For the robot to localize itself on the soccer pitch, it should be capable of two things, besides walking around. It should be able control its actions in such a way, that it should know exactly (although noise is possible) what distance and at what angle the robot moved. The moved distance is written as v and the angle will be written as w, with both values together being the control vector u.

$$u = \begin{bmatrix} v \\ w \end{bmatrix}$$

Another important aspect of robot localization is for the robot to have the ability to detect its surroundings. With this detection, certain landmarks can be found, to create the measurement vector z. This will consist of the estimated range of the robot towards the landmark, the bearing of the landmark from robot and the unique sign of the landmark.

$$z = \begin{bmatrix} r \\ \phi \\ s \end{bmatrix}$$

The method to this landmark detection will be discussed in the 4.4.1 later on.

2.2 Gaussian filters

This thesis will focus on robot localization through the extended information filter, which is the dual of the extended Kalman filter. These are both Gaussian filters, which are recursive state estimators. They represent their estimated state by the multivariate normal distribution.

$$p(x) = det(2\pi\Sigma)^{-\frac{1}{2}}exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

2.3 Kalman filter

The Kalman filter (KF) is used for the localization problem, as it can estimates its current state with uncertain measurements. These estimations are created at time t, in which it uses two values to represent its state. The first value is the mean μ_t , which is the pose of the robot and the second value is the covariance Σ_t .

$$\mu = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$
$$\Sigma = \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} & \Sigma_{1,3} \\ \Sigma_{2,1} & \Sigma_{2,2} & \Sigma_{2,3} \\ \Sigma_{3,1} & \Sigma_{3,2} & \Sigma_{3,3} \end{bmatrix}$$

It is predicted with the values of the previous estimation at time t-1, which is represented by the previous mean μ_{t-1} and covariance Σ_{t-1} . The previous estimations is updated with the help of the control u_t and measurement z_t , to find the new predicted estimate $\bar{\mu}_t$ and $\bar{\Sigma}_t$. In algorithm 1 the KF is shown. In the lines 2 and 3, the estimation step is performed in which the new μ and Σ are estimated, based on the u. In the lines 4, 5 and 6, the correction step is performed. In this step, z_t is used to correct the previous estimate. The KF has a computational complexity of $O(k^{2.4})$, where k is the dimension of the measurement vector z_t , for every iteration. The complexity of the algorithm occurs as a result of line 4 of the algorithm, in which large matrix computations are performed.

Algorithm 1 Kalman Filter [Thrun et al., 2005]

1: function KALMAN FILTER($(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$) 2: $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ 3: $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ 4: $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ 5: $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ 6: $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$ 7: return μ_t, Σ_t 8: end function

2.4 Information Filter

The Information filter (IF) is a variation of the KF, mostly seen as the dual of it, because the KF represents Gaussians in their moments parameterization (μ, Σ) , as opposed to the IF which represents Gaussians in their canonical parameterization. The canonical parameterization consists of the information matrix Ω and the information vector ξ . Both Ω and ξ can be obtained from μ and Σ .

 $\Omega = \Sigma^{-1}$ $\xi = \Sigma^{-1} \mu$

The IF takes, besides Ω and ξ , two other input values, which are the same as the KF, being z_t and u_t . The output estimation is given in the information form, thus the information matrix and the information vector. In Algorithm 2, the IF is shown, where it can be seen to be very similar to the traditional KF. In the same way as the KF, the IF uses two steps, firstly the estimation step (line 2-3) where u_t is taken into account and secondly the correction step (line 4-5), where z_t is used.

The change of the moments parameterization to the canonical parameterization results in different complexities for the estimation step en correction step. The estimation step is costlier in the IF than in the KF, but the correction step is cheaper in the IF than in the KF. The complexities of both steps for both filters are shown in table 1.

Algorithm 2 Information Filter [Thrun et al., 2005]
1: function Information filter $((\xi_{t-1}, \Omega_{t-1}, u_t, z_t))$
2: $\bar{\Omega}_t = (A_t \Omega_{t-1}^{-1} A_t^T + R_t)^{-1}$
3: $\bar{\xi}_t = \bar{\Omega}_t (A_t \Omega_{t-1}^{-1} \xi_{t-1} + B_t u_t)$
4: $\Omega_t = C_t^T Q_t^{-1} C_t + \bar{\Omega}_t$
5: $\xi_t = C_t^{\bar{T}} Q_t^{-1} z_t + \bar{\xi}_t$
6: return ξ_t, Ω_t
7: end function

Table 1: Complexity of Kalman filter and information filter

	Estimation step	Correction step
Kalman filter	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{2.4})$
Information filter	$\mathcal{O}(n^{2.4})$	$\mathcal{O}(n^2)$

Extended Kalman/Information Filter 2.5

One of the notable conditions of both the KF and the IF is that they can only be used for estimating in linear Gaussian systems. A system is linear Gaussian if both the state transition probability $(p(x_t|u_t, x_t-1))$ is a linear function in its arguments with added Gaussian noise as well as the measurement probability $p(z_t|x_t)$. A third property that must hold is for the initial state (μ_0 and Σ_0 or ξ and Ω) to be normally distributed. However in robotics the state transitions measurements are rarely linear.

The extended Kalman filter (EKF) and the extended information filter (EIF) are variations of the KF and IF respectively, that are not bound to linear functions. The EKF uses a Gaussian approximation of the estimation for which the state transition probability and the measurement probabilities are calculated with the nonlinear functions q and h. The non-linear functions q and h are approximated with the corresponding tangent line, to g and h respectively, at the mean. The algorithm 3 represents the EKF with the prediction step in the lines 2-3 and the correction step in the lines 4-6. The EIF is shown in algorithm 4 and the prediction step is given by the lines 2-4 and the correction step by lines 5-7. The main difference is the replacement of the linear parameters A_t , B_t and C_t from algorithm 1 and 2 with the functions g and h and their Jacobians G_t and H_t . The EIF requires an extra step as opposed to the EKF, as it requires the μ_{t-1} for the function g and h. The information matrix Ω_t and the information vector ξ_t are used to convert back to the μ_t and use it for the algorithm.

4	lgorithm	3	Extended	. Ka	lman	Filte	r T	hrun	et	al.,	2005	5
---	----------	---	----------	------	------	-------	------	------	---------------------	------	------	---

1: function EXTENDED KALMAN FILTER $((\mu_{t-1}, \Sigma_{t-1}, u_t, z_t))$

 $\bar{\mu}_t = g(u_t, \mu_{t-1}) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ 2:

3:

 $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ 4:

 $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$ 5: $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ 6:

return μ_t, Σ_t 7:

8: end function

Algorithm 4 Extended Information Filter [Thrun et al., 2005]

1: function INFORMATION FILTER($(\xi_{t-1}, \Omega_{t-1}, u_t, z_t)$) 2: $\mu_{t-1} = \Omega_{t-1}^{-1} \xi_{t-1}$ 3: $\bar{\Omega}_t = (G_t \Omega_{t-1}^{-1} G_t^T + R_t)^{-1}$ 4: $\bar{\xi}_t = \bar{\Omega}_t g(u_t, \mu_{t-1})$ 5: $\bar{\mu}_t = g(u_t, \mu_{t-1})$ 6: $\Omega_t = \bar{\Omega}_t + H_t^T Q_t^{-1} H_t$ 7: $\xi_t = \bar{\xi}_t + H_t^T Q_t^{-1} [z_t - h(\bar{\mu}_t) + H_t \bar{\mu}_t]$ 8: return ξ_t, Ω_t 9: end function

3 Related Work

There are many different algorithms for localization as well as variations of these algorithms. The most important researches for this project have been about either the EKF, the EIF and comparisons of multiple localization algorithms.

The EKF is able to estimate precise local pose, but it fails to find the global pose [Kose et al., 2006], which is the reason of its inability to recover from failures. However for robot soccer, this is less important, as the robot is unlikely to be transported to another location, and with the addition of the validation gate [Kristensen and Jensfelt, 2003], in can cope even better with finding its local position. The validation gate checks if the results of the estimation step and the correction step are in line with each other. If they are vastly different, the z is expected to be corrupted and the correction step is skipped.

In the paper of Thrun [Thrun et al., 2004], the EIF is used to solve the problem of simultaneous localization and mapping (SLAM). It consists of simultaneously creating a map of the environment and localizing the robot within this map. However for this problem, the computational complexity of the EIF is not low enough. As quoted in the paper, "the standard EIF approach requires time quadratic in the number of features in the map, for each incremental update". According to Thrun, by creating a sparse version of the EIF, the computational complexity can be reduced, making the EIF faster.

In the paper of Walter [Walter et al., 2007], an extension of the SEIF is discussed, as proposed by Thrun [Thrun et al., 2004]. The ability of the information filters to use sparse data for less complexity can be extended to the the exactly sparse extended information filter (ESEIF), which is more consistent locally than the SEIF. By marginalizing the robots poses and relocalizing the robot periodically in the map, the number of features can be reduced. This creates a filter that performs nearly identical to the EKF with less computational complexity. This allows for many ways that the EIF can be improved for more accurate results as well as a more efficient processing time. The paper of Frese [Frese, 2005] provides mathematical proof of the following: "in the SLAM information matrix P' off-diagonal entries corresponding to two landmarks decay exponentially with the distance travelled between observation of first and second landmark". This proves the effectiveness of the SEIF, in which space is saved and less computation time is needed.

One of the alternatives to the Gaussian filters, is based on particle filters, which is Monte Carlo Localization (MCL) [Dellaert et al., 1999]. This method is robust, good at recovering from failures and it can be adjusted to use less computational power. The algorithm roughly works in three steps. The first step is processing the odometry, which is the data of the performed motion control of the robot. The next steps are processing the visual sensor input and re-sampling the probabilities of all possible location.

Multiple localization algorithms have been compared to each other in the papers of Gutmann et al. [Gutmann and Fox, 2002] [Gutmann et al., 1998] and Visser et al. [Visser et al., 2011]. The algorithms from these articles have been compared to each other in terms of their accuracy, robustness, time to recover from failures and computation time (table 5 in appendix B). The table shows that the Gaussian filter (EKF) performs best in terms of computation speed, whereas the particle filters (Mix-MCL, A-MCL) perform best in accuracy and robustness.

4 Proposed Method

4.1 Accuracy and Computational Complexity

There are two aspects that are vital for the information filter to be an effective Gaussian-based localization algorithm, thus both should be researched and compared with the Kalman filter. The first one is the accuracy, which shows how accurate the estimation of the robots location is, compared to the actual state. The estimated location will be compared to the actual location and the Euclidean distance between them will be calculated. This will be the first part of the accuracy test, purely based on location. Besides the accuracy, the robot should also be able to use the algorithm within reasonable time, thus both filters will be compared in their computation time.

4.2 Soccer pitch

In the report by Gudi et al. [Gudi et al., 2013], numerous localization problems for the SPL are discussed, based on the visual vulnerabilities of the robot. The main problem is the recognition of the features, like the lines and the goals, as these can be partially covered, resulting in a more difficult feature extraction, which would reduce the robustness of the localization algorithm. A better use of recognition would be the recognition of artificial added landmarks. The soccer pitch on which the robot must localize itself is green with white lines and white goals at both sides, and at the corners and sidelines six landmarks are added, as shown in figure 2. The six landmarks are uniquely recognizable due to their distinctive color combinations (figure 3). They are necessary for the robot to determine its location, as the measurement vector will be created according to whichever landmark has been detected (potentially multiple). For the detection to be as robust as possible, a white screen is placed behind the landmarks, so they will be easier to detect. The landmarks will be the same beacons as proposed in a similar thesis project from 2011 [van der Molen and Visser, 2011] and from the Gutmann dataset [Gutmann and Fox, 2002].

4.3 Dataset

During this project, a dataset⁴ is created by walking nearly a circle shape with a Nao robot. From this dataset, there are 16 data points created. The interval of these steps are predefined, having the robot walk 0.20 meter and changing its orientation with 20 degrees simultaneously in between all data points. A ground truth of the aforementioned path will be created with the OptiTrack. This system detects the robot with six motion capture cameras surrounding the pitch system. A reflecting object is placed on the robot, to track it with the OptiTrack system. The control vector u consists of the predifined steps, being 0.20 meters and 20 degrees each time. Lastly, as part of the dataset, the robot saves a picture with resolution of 1280x960 at every data point. As the robot is stationary when it is taking a picture, the OptiTrack will detect it to be static in one location during a longer period of time, thus it can be extracted that this is the precise moment in which the robot is detecting a landmark, and these values are later on used for the algorithm, instead of the entire dataset. Thus the ground truth, images and movements are used to test the EIF and the EKF in both accuracy and complexity and compare them with each other.

4.4 Implementation

4.4.1 Landmark Recognition

At each measuring point of the Nao, a picture is taken to detect the perceived landmarks. As the landmarks appear as square images when detected, they can be found by finding all squares in the image with the help of OpenCV. All found squares will then be compared to the possible landmarks and it will be determined if they are indeed a certain landmark, or just a random found square. Following the detection of the landmarks, the right measurement values must be extracted, which are the bearing and the range between the Nao and the landmark. The range or distance (D) can be found by using the formula shown below.

$$D = \frac{W \cdot F}{P}$$

⁴https://github.com/tobiasxg/Thesis



Figure 2: The soccer pitch used during the research. The landmarks are shown in the corresponding colors to the real landmarks.

W stands for the actual width of the object in cm, P is the pixel width of the object in the image and the F is the focal length, which is a fixed value that can be found by the following formula.

$$F = \frac{P \cdot D}{W}$$

This formula has the exact same values as the previous one, and must be used to find the focal length of the object for the first time. By using the actual width, using the OptiTrack to determine the distance between the object and the Nao when taking the picture and measuring the pixel width of the object in the image, the focal length can be found and used for calculating the distance in different situations. For finding the bearing, a simpler calculation can be performed. By dividing the pixel width of the image by the vision angle of the Nao (60.97 degrees)⁵ the angle per pixel is found. By multiplying the centre pixel of the object in the image by the angle per pixel, the bearing is found. The measurement vector is then constructed from the range, the bearing and the unique landmark sign for all detected landmarks and concatenated to the following matrix with the number of landmarks as n.

r_1	r_2	 r_n
ϕ_1	ϕ_2	 ϕ_n
s_1	s_2	 s_n

In figure 4, the pseudocode is shown for the landmark detection and creating the measurement vectors of the corresponding landmarks.

⁵http://doc.aldebaran.com/2-1/family/robots/videorobot.html



Figure 3: The beacons used as landmarks

1: for square in detected squares do
2: if square is landmark then
3: Calculate Range and Bearing
4: Create measurement vector
5: end if
6: Concatenate all found landmark measurement vectors
7: end for

Figure 4: Pseudocode Landmark detector and measurement

4.4.2 Extended Information and Kalman Filter

For the EIF as well as the EKF, the landmark detector is used for creating the measurement vector(s). As previously stated, the control vector is derived from the fixed path the robot has walked during the research, being 0.20 metres with an angle change of 20 degrees. The overall algorithms have been shown previously in 2.5, being algorithm 3 and 4.

The validation gate is implemented for the EKF as well as implementing a similar approach in the EIF. The validation gate compares the estimate derived from the estimation step and from the correction step and compares these estimates. The difference between the two estimates is compared with the threshold of the validation gate, and if the estimates difference is higher than the threshold, the correction step is ignored. This validation gate can reduce the measurement noise that could occur due to mistakes made during measuring the surrounding landmarks.

4.4.3 Optimization

There are numerous parameters that can be changed in order to optimize the algorithm. These parameters are the noise covariance matrix M, which consists of four values α that can can be optimized, and the covariance Q of additional measurement noise, consisting of three uncertainty values σ corresponding to the range, bearing and sign of the measurement vector. These parameters have been optimized by converging to the optimal values, which leads to the best results. The convergence graphs are shown in figure 5 for α and in figure 6 for σ . After the optimization process, the optimal parameters are set as follows for the Kalman filter:

$$M = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 w_t^2 & 0\\ 0 & \alpha_3 v_t^2 + \alpha_4 w_t^2 \end{bmatrix} = \begin{bmatrix} -0.9 \cdot v_t^2 + 0.5 \cdot w_t^2 & 0\\ 0 & -1.8 \cdot v_t^2 + 0.7 \cdot w_t^2 \end{bmatrix}$$
$$Q = \begin{bmatrix} \sigma_r^2 & 0 & 0\\ 0 & \sigma_{\phi}^2 & 0\\ 0 & 0 & \sigma_s^2 \end{bmatrix} = \begin{bmatrix} 14.2 & 0 & 0\\ 0 & 1.0 & 0\\ 0 & 0 & 1.0 \end{bmatrix}$$

The optimal parameters for the information filter are different and they are shown below:

$$M = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 w_t^2 & 0\\ 0 & \alpha_3 v_t^2 + \alpha_4 w_t^2 \end{bmatrix} = \begin{bmatrix} 9.5 \cdot v_t^2 + -2.0 \cdot w_t^2 & 0\\ 0 & -1.2 \cdot v_t^2 + 0.3 \cdot w_t^2 \end{bmatrix}$$
$$Q = \begin{bmatrix} \sigma_r^2 & 0 & 0\\ 0 & \sigma_{\phi}^2 & 0\\ 0 & 0 & \sigma_s^2 \end{bmatrix} = \begin{bmatrix} 0.4 & 0 & 0\\ 0 & 0.3 & 0\\ 0 & 0 & 1.0 \end{bmatrix}$$

It is crucial that these parameters are optimized correctly, as they can severely change the outcome of the algorithm, as seen in figures 5 and 6. The parameters are set at their optimum, however it is possible that only a local optimum has been found, and that there would still be a different global optimum.

The final parameter that can be optimized is the value for the validation gate. However the optimalization converges it to a relatively high threshold, resulting in no effect during the optimal performances.

5 Results

The results of the algorithm lay in two parts, the accuracy and the computation time. Below a comparison will be given based on both these results.

5.1 Accuracy Results

The EIF performs slightly better when optimized than the EKF. The estimated locations of both algorithms are shown in figure 7. The estimations (red) of the EKF are worse at the beginning, while the EIF estimates worse a few steps later. The total Euclidean distance as well as the average Euclidean distance of both algorithms are shown in table 2. The EIF estimates the average location 0.04511 meter closer than the EKF.

As can be seen in figure 7, there are many outliers, especially in the beginning. By removing the outliers from the total Euclidean distance, the average Euclidean distance goes down. In figure 8, a graph is shown with two lines are shown that show the average Euclidean distance with on the x-axis the number of best distances taken. It shows that the EIF is constantly performing better, regardless of outliers, but eventually being surpassed by the EKF.

5.2 Computational Complexity Results

The computation time is the other aspect on which the algorithms are compared. The EIF is slower in the the estimation step, but faster in the correction step, when compared to the EKF. In table 3, the average computation time is shown



Figure 5: Convergence graphs of optimizing for the values of α_i . The value for α_i is shown on the x-axis with *i* being a value from 1 to 4. On the y-axis the Euclidean distance of the ground truth and the estimation is shown.



Figure 6: Convergence graphs of optimizing for the values of σ_r , σ_{ϕ} and σ_s . The value for σ_i is shown on the x-axis. On the y-axis the euclidean distance of the ground truth and the estimation is shown. The convergence for the EKF is shown in the graphs on the left and for the EIF on the right.

Table 2: Euclidean distance and average Euclidean distance of the EKF and the EIF

		Euclidean distance	Average Euclidean distance
E	KF	2.316408	0.144776
E	IF	1.594656	0.099666



Figure 7: Robots initial position (green), the filter estimations (red) and the ground truth (blue) of EKF and EIF. In (a) the estimations of the EKF are shown and in (b) of the EIF. The orientation is shown by the dotted lines.



Figure 8: The EKF and EIF show the average Euclidean distance when only taking the best estimates. On the x-axis the number of best estimates is shown, with 16 being all the estimates.

Table 3: Computation time (seconds) of algorithm and the steps within the algorithm.

	EKF	EIF	Ratio EKF:EIF
Estimation step	0.000902	0.001367	0.66
Correction step	0.000822	0.000728	1.13
Complete Algorithm	0.001416	0.001822	0.78

for the two steps, the algorithm all together and the ratio of how the EIF and EKF compare to each other. The correction step of the EKF takes 1.13 times the time needed for the same step in the EIF, thus the EIF being slightly faster. However in the other step, the estimation step, the EKF only takes 0.66 times the time when compared to the EIF. The overall algorithm is thus slower with the EIF. If the robot would perceive more landmarks, the correction step would be more often performed (once per landmark), thus it would result in a change of overall computation time. In figure 9, a graph is shown with two hypothetical lines, corresponding to multiple perceived landmarks. It seems that if more than 5 landmarks are detected, the EIF requires less computation time than the EKF. In table 4 the corresponding values are shown.



Figure 9: In this graph, two hypothetical lines are drawn, when more landmarks would be detected, with on the y-axis the computation time. The black dotted line shows where EKF and the EIF have an (almost) equal computation time

	EKF	EIF	Ratio EKF:EIF
1 detected landmark	0.001724	0.002095	0.82
2 detected landmarks	0.002546	0.002823	0.90
3 detected landmarks	0.003368	0.003552	0.95
4 detected landmarks	0.004190	0.004280	0.98
5 detected landmarks	0.005012	0.005008	1.00
6 detected landmarks	0.005834	0.005736	1.02
7 detected landmarks	0.006656	0.006464	1.03

Table 4: Computation time of steps of algorithm and hypothetical time for more landmarks

6 Discussion and Conclusion

The EKF and EIF are both effective in their task of localizing the Nao robot on a soccer pitch, but the EIF is slightly better, by estimating on an average 0.04511 closer to the actual position than the EKF.

Removing the outliers shows that the EKF has a more outliers, but also more location predicted nearly correct. EIF seems to estimate the locations overall better and thus being more accurate with the outliers.

The computation time for the EKF is slightly faster than the EIF, thus the EKF has less computational complexity. The use of the EIF could be an alternative, but it would cost slightly more computation time. However, as the correction step is performed for every observed landmark, in a situation with multiple detected landmarks, the EIF could be more efficient. Thus when the number of landmarks in the field would be higher, and more than 5 landmarks would be observed during the localization, the EIF would be the better in both accuracy and computation time.

7 Future Research

For future research, numerous parts can be examined further, as well as different methods that can be used.

As the EIF performs relatively faster than the EKF when more landmarks are perceived, thus by using more landmarks or different detectable features, the EIF would likely be much faster than the EKF. It would be interesting to compare EIF to other localization algorithms when a high number of detectable features are used as measurement data

The accuracy results could be examined in a different way, as it is currently done with the Euclidean distance, but using statistical analysis might give different results. The dataset can be increased in size as well, as it is very sensitive to noise and coincidences.

Lastly, a similar version of the validation gate has been implemented for the EIF, however it has not been researched how it effected the EIF optimally. An extension of the EIF with an effective validation gate could result in a better EIF.

8 Acknowledgements

My sincere thanks to Arnoud Visser for guiding me through the process as well as providing me with insights and feedback. Also thanks to the Dutch Nao Team for helping me with the hardware needed for this thesis.

References

- [Dellaert et al., 1999] Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and Automation*, 1999. Proceedings. 1999 IEEE International Conference on, volume 2, pages 1322–1328. IEEE.
- [Frese, 2005] Frese, U. (2005). A proof for the approximate sparsity of slam information matrices. In *Robotics and Automation*, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, pages 329–335. IEEE.
- [Gudi et al., 2013] Gudi, A., de Kok, P., Methenitis, G. K., and Steenbergen, N. (2013). Feature detection and localization for the robocup soccer spl. *Project report, Universiteit van Amsterdam (February 2013).*
- [Gutmann et al., 1998] Gutmann, J.-S., Burgard, W., Fox, D., and Konolige, K. (1998). An experimental comparison of localization methods. In *Intelligent Robots and Systems, 1998. Proceedings.*, 1998 IEEE/RSJ International Conference on, volume 2, pages 736–743. IEEE.
- [Gutmann and Fox, 2002] Gutmann, J.-S. and Fox, D. (2002). An experimental comparison of localization methods continued. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 454– 459. IEEE.
- [Kose et al., 2006] Kose, H., Celik, B., and Akin, H. L. (2006). Comparison of localization methods for a robot soccer team. *International journal of* advanced robotic systems, 3(4):41.
- [Kristensen and Jensfelt, 2003] Kristensen, S. and Jensfelt, P. (2003). An experimental comparison of localisation methods, the mhl sessions. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, volume 1, pages 992–997. IEEE.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). Probabilistic robotics. MIT press.
- [Thrun et al., 2004] Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8):693–716.
- [van der Molen and Visser, 2011] van der Molen, H. and Visser, A. (2011). Selflocalization in the robocup soccer standerd platform league with the use of a dynamic tree. *Bachelor Thesis, University Of Amsterdam.*
- [Visser et al., 2011] Visser, A., de Bos, D., and van der Molen, H. (2011). An experimental comparison of mapping methods, the gutmann dataset. In *Proc.* of the RoboCup IranOpen 2011 Symposium (RIOS11).

[Walter et al., 2007] Walter, M. R., Eustice, R. M., and Leonard, J. J. (2007). Exactly sparse extended information filters for feature-based slam. *The International Journal of Robotics Research*, 26(4):335–359.

Appendices

A Hardware/Software

A.1 Hardware

- Aldebaran Robot Nao 5.0
- OptiTrack
 - Flex 13

A.2 Software

- Python 2.7.14
- Python libraries:
 - NumPy 1.14.0
 - OpenCV 3.4.1
 - PIL 5.0.0
 - Matplotlib 2.1.2
- Choregraphe 2.1.4
- Motive 1.7.5.0

B Comparison of localization algorithms

In the table below, numerous localization algorithms have been compared to each other in their strong and weak points.

very Time Computation Spee	++	-/+	I	-/+	-/+	-/+	-/+
Reco	1	+	+	++	+	1	+
Robustness	-/+	+	+	++++	+	+	1
Accuracy	-/+	-/+	+	+	+	+	+
Algorithm	Extended Kalman filter (EKF)	Markov Localization (ML)	Mixture MCL (Mix-MCL)	Adaptive MCL (A-MCL)	Markov Localization EKF (ML-EKF)	Sensor Resetting Localization 1 (SRL1)	Sensor Resetting Localization 2 (SRL2)

Table 5: A comparison of localization algorithms with their strong and weak parts. The information about ML is from Gutmann et al. [Gutmann et al., 1998] and the rest is from Gutmann et al. [Gutmann and Fox, 2002]