EKF-BASED LOCALIZATION WITH LRF

Gonçalo Castilho (75305), Tiago Gomes (76079), Pedro Sousa (79205), and Markus Kühlen (85746)

***Abstract*—This project report deals with the implementation of an extended Kalman Filter (EKF) on a mobile robot, which is equipped with a laser rangefinder (LRF). The goal of the project is to estimate the two dimensional pose of the mobile robot in real time. The available odometry information of the robot is used for the state prediction. For the observation model a map of the environment is pre-acquired and used. The laser range finder gives the observation information by scanning the robots environment in real time. Point clouds are then used for comparing the predicted and the real observations in the matching step.**

***Index Terms*—Extended Kalman Filter, Localization, Robotics, Laser Rangefinder, ROS.**

I. Introduction

MOBILE autonomous systems are fundamentally dependent on localization. Their motion and task planing require knowledge about the current robot state. For mobile robots the current posture, which includes position and orientation, is an important part of their state.

For this reason the report for the group project of the experimental part within the ”Autonomous Systems” class at Instituto Superior Técnico in 2016/2017 deals with EKF-based localization with a LRF. The objective for the students is to prove their theoretical knowledge on mobile robotics localization in a practical scenario and to gain first experiences with implementing in a robot operating system (ROS) environment.

The used hardware in this project consists of a Pioneer 3DX mobile robot, a Hokuyo URG-04LX-UG01 laser rangefinder and at two laptops. The Pioneer 3DX comes with implemented motion sensors that provide odometry information. The Pioneer’s integrated sonar sensors are not used.

The available odometry information can be used straight away for localization. However, relying only on odometry is inaccurate, since the errors arising from the uncertainties of the odometry model and the measurement noise of the odometric sensor are accumulating over time. To improve it’s localization the robot can use available information from other sensors, such as a sonar, a camera or a laser rangefinder, each having different advantages and disadvantages. The task for this project is to use a laser rangefinder, which is more accurate in comparison to a sonar, but is not able to measure transparent objects. The additionally gained information has to be merged with the odometry information. Therefore, different kind of algorithms, so called filters, can be used.

The original Kalman Filter is the optimal estimate algorithm for linear system models with additive independent white noise in the prediction and measurement systems. To be able to apply the Kalman Filter based filtering method to non-linear systems the extended Kalman Filter uses linearization around a working point. As long as the system model is well known and accurate, the EKF is the most widely used estimation algorithm. Otherwise Monte Carlo methods, especially particle filters, will lead to better results, despite being computationally more expensive. [1]

As the considered robot system is real world non-linear system the used filter for localization has to be robust to the influence of noise and non-linearities. The movement of a wheeled mobile robot in a two dimensional environment can be described by an accurate system model. Therefore, the EKF can be used for computationally efficient localization.

This paper is organized as follows. In section II the used methods and algorithms are introduced. Afterwards the implementation of the EKF is described in section III. The results of the algorithm running on the real Pioneer 3DX robot in a test environment are then discussed in section IV. The paper then is concluded in section V.

II. Methods

A. Robot Operating System

The Robot Operating System is a set of software libraries and tools that helps building robot applications. It is a modularized, portable and standard system which allows it to be developed by different system designers, and has a wide range of uses, from quadcopters to industrial-type robotic manipulators.

ROS is mainly composed by 4 elements:

- Roscore – main program, which acts primarily as a name server
- Node – process that uses ROS framework and preforms a specific task
- Topic – mechanism to send messages from a node to one or more nodes. Follows publisher-subscriber design pattern
- Service – mechanism for a node to send a request to another node and receive a response in return.

In order to develop the project, several ROS packages were downloaded. A Package is a self-contained directory containing sources, makefiles, builds and others. Below there is a list of the main packages used in this project.

- rosaria – ROS interface for the pioneer 3DX robot. It allows issuing commands to the robot wheel motors as well as retrieving information on the odometric sensors.
- teleop_twist_keyboard – Provides teleoperation using a keyboard. Two computers are communicating through a wireless internet connection; the second computers keyboard is used to remotely control the robot, connected to the first computer.
- slam_gmapping – Provides laser-based SLAM (Simultaneous Localization and Mapping), elaborating a 2-D occupancy grid map from laser and pose data collected by the LRF and the Pioneer robot.

- map_server – Provides a map_server ROS Node, which offers map data as a ROS Service. It also provides the map_saver command-line utility, which allows dynamically generated maps to be saved to file.
- rviz – visualizing tool for displaying sensor data and state information from ROS.
- tf – Used to keep track of the robots frame in relation to the static world reference
- laser_assembler – Provides nodes to assemble point clouds from the LaserScan messages.

B. Mapping

The LRF was fitted on top of the Pioneer P3-DX unit, both connected to one computer using the rosaria, slam_gmapping, map_server, rviz and tf packages. Through the slam_gmapping the map was generated, being stored with the map_server package. The rviz was used to visualize the mapping and the current robot position, based on the odometry, and the tf was used to keep the localization of the robot in the map reference. A second computer was used with the teleop_twist_package to control the robot from a distance. The final map was obtained using certain landmarks, which made it possible to go around the odometry imprecision.

C. Simulation

Gazebo software was used during the development of the code to test the behavior of the robot in a controlled environment without having to use the real robot. This expedited the code testing.

III. Implementation

This section describes how the EKF with it's three basic steps has been implemented for the faced robot localization problem with a LRF. In subsection III-A the motion model and the observation model are discussed. The subsections III-B and III-C deal with the matching respectively the update step. The resulting scheme of the implementation is shown in figure 1.

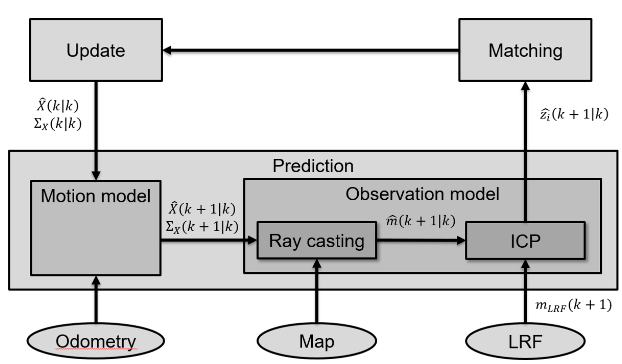

Fig. 1: Scheme of the implemented EKF

The robot's state X is defined as it's current two dimensional pose. The observation Z used for the Extended Kalman filter, like explained in section III-B, also includes variables to describe a pose.

$$X = \begin{pmatrix} x_s & y_s & \theta_s \end{pmatrix}^T \tag{1}$$

$$Z = \begin{pmatrix} x_{obs} & y_{obs} & \theta_{obs} \end{pmatrix}^T \tag{2}$$

$$V = Z(k+1) - \hat{Z}(k+1) = \begin{pmatrix} \Delta x_v & \Delta y_v & \Delta\theta_v \end{pmatrix}^T \tag{3}$$

With the state and the observation being a 3 by 1 vector, the dimensions of all the variables used in the EKF are defined. Table I lists the used variables with their symbols, dimensions and a short description..

Symbol	Dimension	Description
X	3 x 1	State vector - Robot pose
v	3 x 1	????
Z	3 x 1	Observation
Σ	3 x 3	Robot state covariance
f	3 x 1	Robot motion model function
F	3 x 3	Robot motion Jacobian
Q	3 x 3	Motion noise
U	2 x 1	Motion input
m^i	2 x 1	Laser beam measurement
h	3 x 1	Observation model function
H	3 x 3	Observation Jacobian
R	3 x 3	Observation noise
K	3 x 3	Kalman gain

TABLE I: Variables and functions used for the EKF

A. Prediction

The aim of the EKF prediction step is to receive the updated state and covariance and then from there on determine the predicted observation for the next time step $\hat{Z}_i(k+1|k)$. To do so, the motion model predicts the future state in the next time step $\hat{X}(k+1|1)$ and the corresponding covariance matrix $\Sigma_X(k+1|k)$. From there on it is possible for the observation model to obtain the predicted observation $\hat{Z}_i(k+1|k)$. The motion and observation model need to be defined by the user depending on the problem and solution approach.

1) State Prediction: One of the main ideas of the EKF lies in linearising the motion and measurement model. For the state prediction, the motion model is decomposed a the noise-free part in (4) and a random noise component with zero mean in equation (5).

$$\hat{X}(k+1|k) = f(\hat{X}(k), U(k)) \tag{4}$$

$$\Sigma_X(k+1|k) = F\Sigma_X(k|k)F^T + Q(k) \tag{5}$$

As the considered robot is a three-wheeled vehicle that drives in a planar two dimensional environment, a motion model can be developed for specifically for this case. The Pioneer 3DX is steered by giving different speed commands to the left and right wheel. Therefore, the motion input vector is defined as in (6).

$$U = \begin{pmatrix} \omega_{Right} & \omega_{Left} \end{pmatrix}^T \tag{6}$$

$$\beta = \frac{\omega_{Right} - \omega_{Left}}{d} \quad (7) \qquad R = \frac{\omega_{Left}}{\beta} \quad (8)$$

With the help of d, the distance between the robot's driven wheels, and the defined parameters R and β in (7) respectively

(8), it is possible to formulate the motion model function in equation (9).

$$f = \begin{pmatrix} \hat{x}_s(k) + (R + \frac{d}{2})(sin(\hat{\theta}_s + \beta) - sin(\hat{\theta}_s)) \\ \hat{y}_s(k) + (R + \frac{d}{2})(-cos(\hat{\theta}_s + \beta) + cos(\hat{\theta}_s)) \\ \hat{\theta}_s(k) + \beta \end{pmatrix} \tag{9}$$

For calculation the predicted state covariance $\Sigma_X(k+1|k)$ with equation (5) the Jacobian F of the motion model function is needed. It can be calculated by derivating f with respect to the state X.

$$F = \frac{\partial f}{\partial X} = \begin{pmatrix} 1 & 0 & (R + \frac{d}{2})(cos(\hat{\theta}_s + \beta) - cos(\hat{\theta}_s)) \\ 0 & 1 & (R + \frac{d}{2})(sin(\hat{\theta}_s + \beta) - sin(\hat{\theta}_s)) \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

The Pioneer 3DX robot is able to determine its odometry off the shelf. By using backward differences the next state can also be predicted. With this solution approach the equations for the motion model function f and its Jacobian F can be simplified.

$$f = \begin{pmatrix} \hat{x}_s(k) + \Delta x_{odom}(k) \\ \hat{y}_s(k) + \Delta y_{odom}(k) \\ \hat{\theta}_s(k) + \Delta \theta_{odom}(k) \end{pmatrix} \tag{11}$$

$$F = \begin{pmatrix} 1 & 0 & -\Delta y_{odom}(k) \\ 0 & 1 & \Delta x_{odom}(k) \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} q_{x,stat} + q_{x,dyn}|\Delta x_{odom}(k)| \\ q_{y,stat} + q_{y,dyn}|\Delta y_{odom}(k)| \\ q_{\theta,stat} + q_{\theta,dyn}|\Delta \theta_{odom}(k)| \end{pmatrix} \tag{13}$$

2) Observation Prediction: Subsubsection text here.

B. Matching

Maybe add the anti-kidnapping stuff to the scheme picture.

C. Update

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{14}$$

$$V(k+1) = Z(k+1) - \hat{Z}(k+1) = \begin{pmatrix} \Delta x_v & \Delta y_v & \Delta \theta_v \end{pmatrix}^T \tag{15}$$

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)V(k+1) \tag{16}$$

$$\Sigma_X(k+1|k+1) = (I - K(k+1) \cdot H) \cdot \Sigma_X(k+1|k) \tag{17}$$

$$K(k+1) = \frac{\Sigma_X(k+1|k)}{\Sigma_X(k+1|k) + R(k+1)} \tag{18}$$

$$R = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix} \tag{19}$$

IV. Results

The used results will be presented here. This sentence is only to fill one more line. "Make the text beautiful again".

IST benefits from an IBM supercomputer built in 2007, which is one of the most powerful in Portugal (1.6 TFLOPS as of 2007).[3]

V. Conclusion

Instituto Superior Tcnico (IST) was created in 1911 from the division of the Industrial and Commercial Institute of Lisbon. Alfredo Bensade, an engineer, was IST's first dean (19111922) and promoted a wide-rang reform in the Portuguese higher technical education, including the first engineering courses at IST: mining, civil, mechanical, electrical, and chemical-industrial. IST's second dean was Duarte Pacheco (19271932), also an engineer, who was responsible for the construction of the university campus at Alameda. The architect Porfrio Pardal Monteiro designed it. Meanwhile, IST became part of the recently created Technical University of Lisbon. Throughout the following decade, the image of engineers from IST was projected into major engineering works, promoted by Duarte Pacheco, who was by the time Minister of Public Works.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] S.J. Julier and J.K. Uhlmann, *Unscented filtering and nonlinear estimation*, Proceedings of the IEEE: 401422., 2004.