ME 597 Lab 2 Report

Iain Peet Andrei Danaila Ming Ma Abdel Hamid Ahmed Salam
20252201

December 2, 2011

1 Introduction

In this lab, an Extended Kalman Filter state estimator is to be derived and implemented for a robot with Ackermann steering. In addition to state estimation, the robot's LIDAR will be used to construct an occupancy grid map of its surroundings.

2 Extended Kalman Filter Design

2.1 Motion Model

The robot motion model has the following state:

$$x_t = \begin{bmatrix} v_f \\ \psi \\ x \\ y \end{bmatrix}_t \tag{1}$$

Where v_f is forward velocity, ψ is heading, and x and y are position in the horizontal plane.

The inputs to the system are:

$$u_t = \begin{bmatrix} V_m \\ \delta \end{bmatrix}_t \tag{2}$$

Where V_m is motor power, and δ is steering angle.

The motion model is:

$$\dot{x_t} = \begin{bmatrix} -\alpha v_f + K\alpha V_m \\ \frac{v_f}{L} sin(\delta) \\ v_f cos(\psi) \\ sin(\psi) \end{bmatrix}_t + w_t \qquad w_t \sim N(0, Q) \tag{3}$$

Where K and α are constants which may be found empirically and L is the robot wheelbase length.

2.2 Measurement Models

A simple measurement model where all states are directly measured is used. Encoder readings are differentiated to yield direct forward velocity measurements. The local position system yields direct measurements of heading and planar position. GPS can be considered to have the same model as the LPS, with heading directly estimated from successive position measurements, so long as the distance travelled between measurements is large relative to position measurement error.

The measurement model is trivially defined as follows:

$$y_t = \begin{bmatrix} v_f \\ \psi \\ x \\ y \end{bmatrix}_t + v_t \qquad v_t \sim N(0, R) \tag{4}$$

2.3 Prediction

The a priori state expectation is computed from the motion model by

$$\bar{x}_k = f(\hat{x}_{k-1}, u_{k-1}) \tag{5}$$

Where $f(\cdot)$ may be trivially derived from (3).

The a priori covariance is computed by linearizing about the current state:

$$\bar{P}_k = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{6}$$

Where A_k is the Jacobian, $\frac{\delta f}{\delta x}|_{\hat{x}_{k-1}, u_{k-1}}$:

$$\begin{bmatrix} -\alpha T & 0 & 0 & 0 \\ \frac{T}{L} sin(\delta_{k-1}) & 1 & 0 & 0 \\ T cos(\psi_{k-1}) & -v_{f,k-1} T sin(\psi_{k-1}) & 1 & 0 \\ T sin(\psi_{k-1}) & v_{f,k-1} T cos(\psi_{k-1}) & 0 & 1 \end{bmatrix} \tag{7}$$

And W_k is the Jacobian of x_k with respect to process noise, which is simply I in our motion model.

2.4 Correction

The a priori state estimate is corrected by sensor measurements. First, the Kalman gain is computed:

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + V_k R_k V_k^T)^{-1} \tag{8}$$

Where H_k is the Jacobian of y_k with respect to x_k, and V_k is the Jacobian of y_k with respect to measurement noise. For our measurement model, H_k and V_k are both I.

The a posteriori state expectation and covariance can then be computed:

$$\hat{x}_k = \bar{x}_k + K_k (y_k - \bar{x}_k) \tag{9}$$

and

$$P_k = (I - K_k H_k) \bar{P}_k \tag{10}$$

2.5 Simulation Results

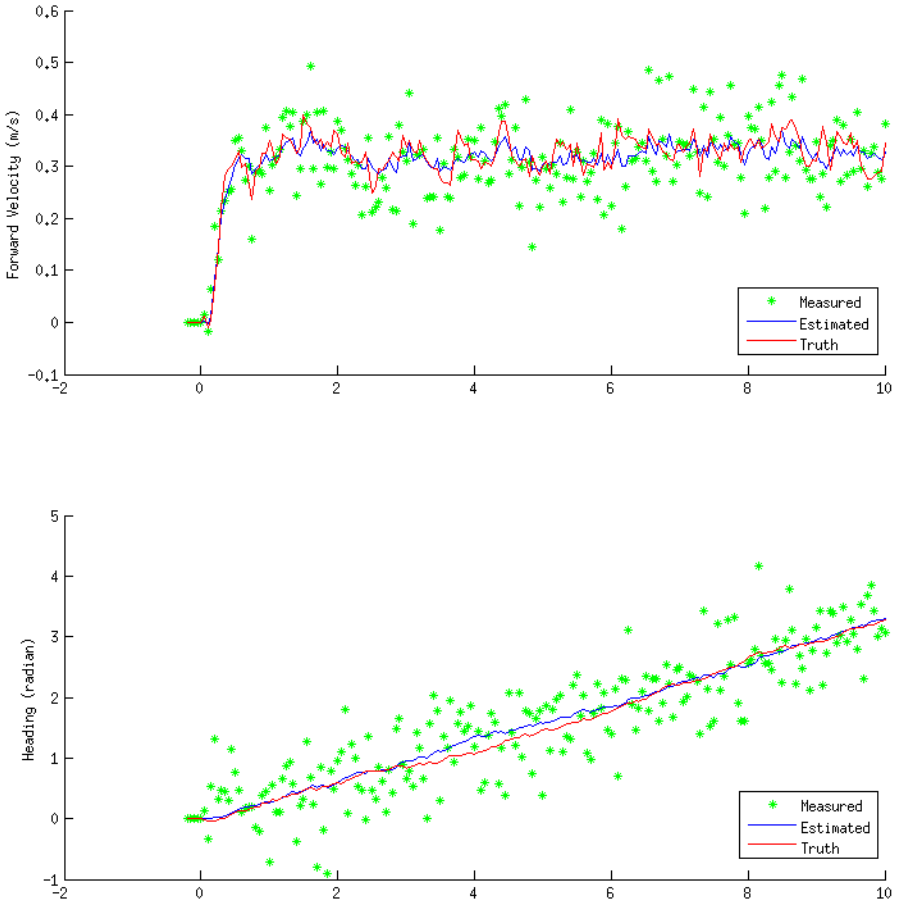

Figure 1: Simulated EKF estimation of forward velocity and heading

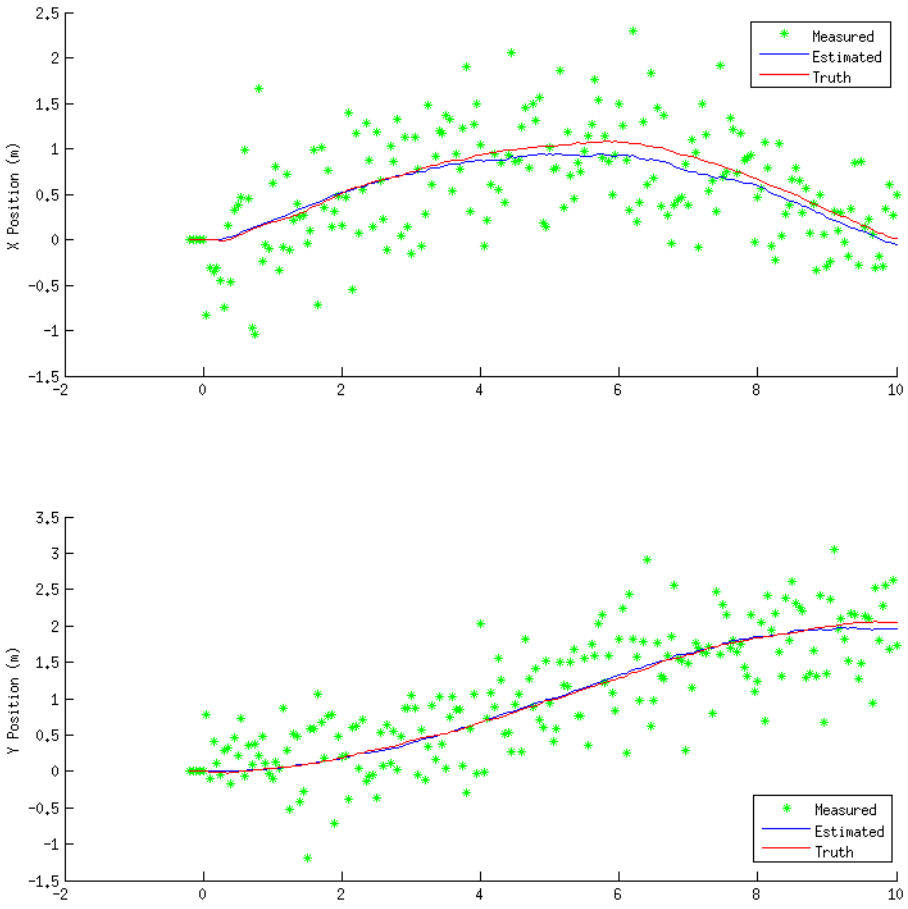

Figure 2: Simulated EKF estimation of planar position

2.6 Empirical Results

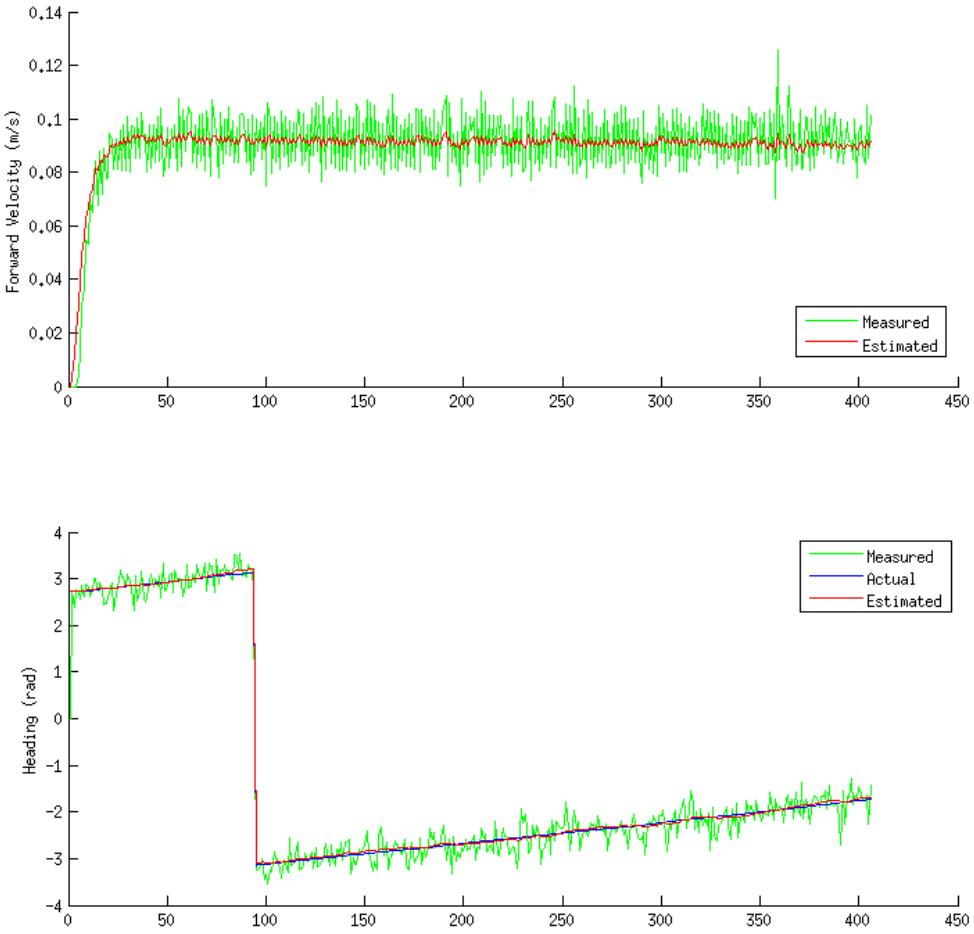

Figure 3: EKF estimation of forward velocity and heading

The local positioning system has been found to be extremely accurate. Practically speaking, the EKF presents little useful gain over the accuracy of the LPS measurements. We can, however, test the performance of the EKF with less accurate position data, by taking LPS data as ground truth, adding noise to the LPS data, and providing the resulting noisy data to the EKF.

Figures 3 and 4 show the estimates produced by the EKF, applied to actual system inputs and noise-corrupted LPS measurements. It can be seen that the EKF does an excellend job of recovering the original LPS measurement data. Although we do not have good ground truth measurements for velocity, the velocity estimates do seem reasonable.

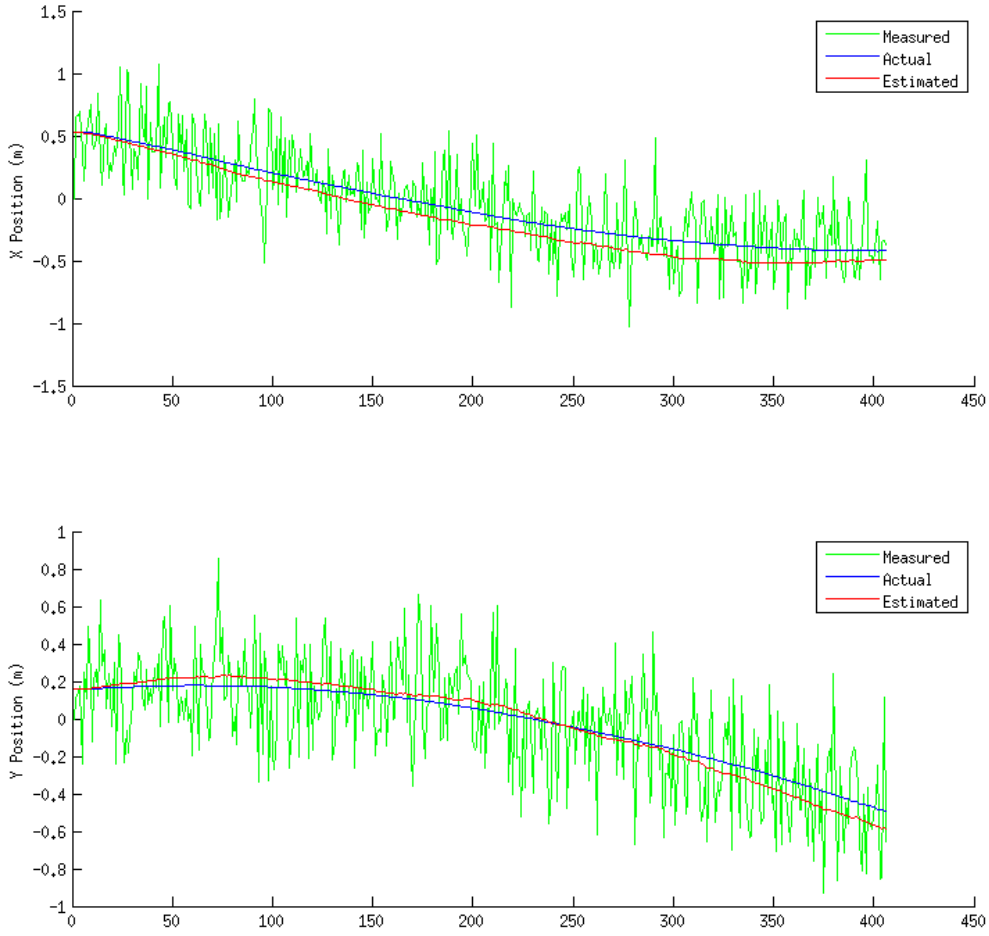

Figure 4: EKF estimation of planar position

3 Mapping Design

As it moves, the robot is required use LIDAR data to produce an occupancy grid map of its surroundings. In addition to EKF state estimates, the mapping algorithm will require an inverse measurement model for the LIDAR.

4 LIDAR Measurement Model

The LIDAR provides a sequence of range measurements r_i, each of which is made in a different direction, ϕ_i:

$$r = \{r_1 \dots r_k\} \quad r_i \in [r_{min}, r_{max}] \tag{11}$$

$$\phi = \{\phi_1 \dots \phi_k\} \quad \phi_i \in [\phi_{min}, \phi_{max}] \tag{12}$$

5 Inverse LIDAR Measurement Model

In order to compute an occupancy grid, we must use LIDAR to determine whether or not a particular grid cell is occupied. For any given cell, there are three possibilities:

1. At least one scanline passes through the cell, but no range measurements fall within the cell.
2. At least one range measurement falls within the cell.
3. No scanlines pass through the cell.

In the first case, it can be inferred with confidence that the cell is unoccupied. In the second case, it is clear that the cell contains an object. The third case arises either when a cell is obscured by a nearby object, or when the cell is outside of the area swept by the LIDAR. In the third case, no information is about the status of the cell.

In order to evaluate the state of a cell, we find the scanline which passes closest to the centre of the cell:

$$i_{nearest} = \underset{i}{\text{argmin}} \quad |\phi_i - tan^{-1}(\frac{y_c - y_r}{x_c - x_r})| \tag{13}$$

Where (x_c, y_c) are the co-ordinates of the centre of the cell, and (x_r, y_r) are the current co-ordinates of the robot.

Once $i_{nearest}$ is known, the range measurement $r_{i,nearest}$ is compared against the distance to the cell. If the measured range is within some threshold α of the distance to the cell, it is considered likely that the cell is occupied. If the range measurement is substantially greater than the distance

to the cell, it is considered likely that the cell is unoccupied. Finally, if the range measurement is less than the distance to the cell, no information is obtained about the state of the cell.

This is an extremely simplistic inverse measurement model. Uncertainty in the position and heading of the robot are ignored. Uncertainty in scanline direction is ignored, and the arc covered by a particular range measurement is assumed to be equal to the angular separation of consecutive measurements. The use of the scanline which passes nearest the centre of the cell ignores the possibility that multiple scanlines might pass through a single cell, or, alternatively, that even the nearest scanline does not pass through the cell at all.

5.1 Empirical Results

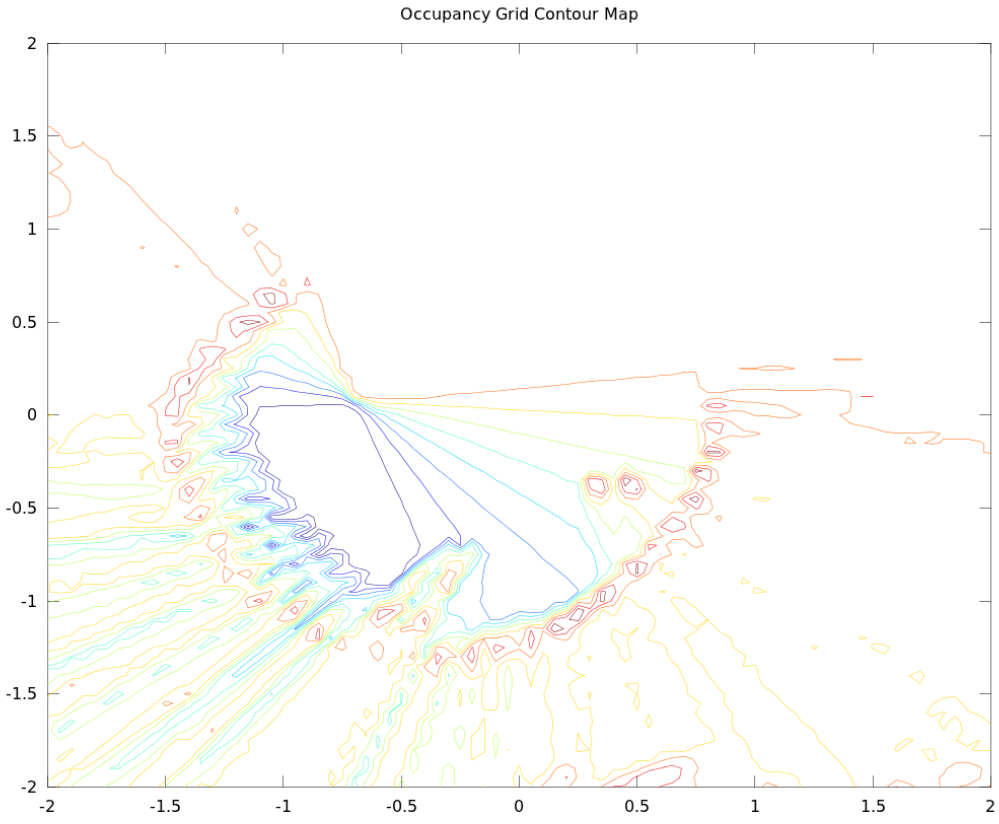

Figure 5: A contour map of the occupancy grid logit(probability) values.

Figures 5 and 6 show the completed occupancy grid map, resulting from driving the robot within a smiley face formed of traffic cones. Although the robot's turning radius was not sufficiently small to be able to scan the entire region without colliding with an obstacle, the occupancy grid seems to be a good representation of the actual environment.

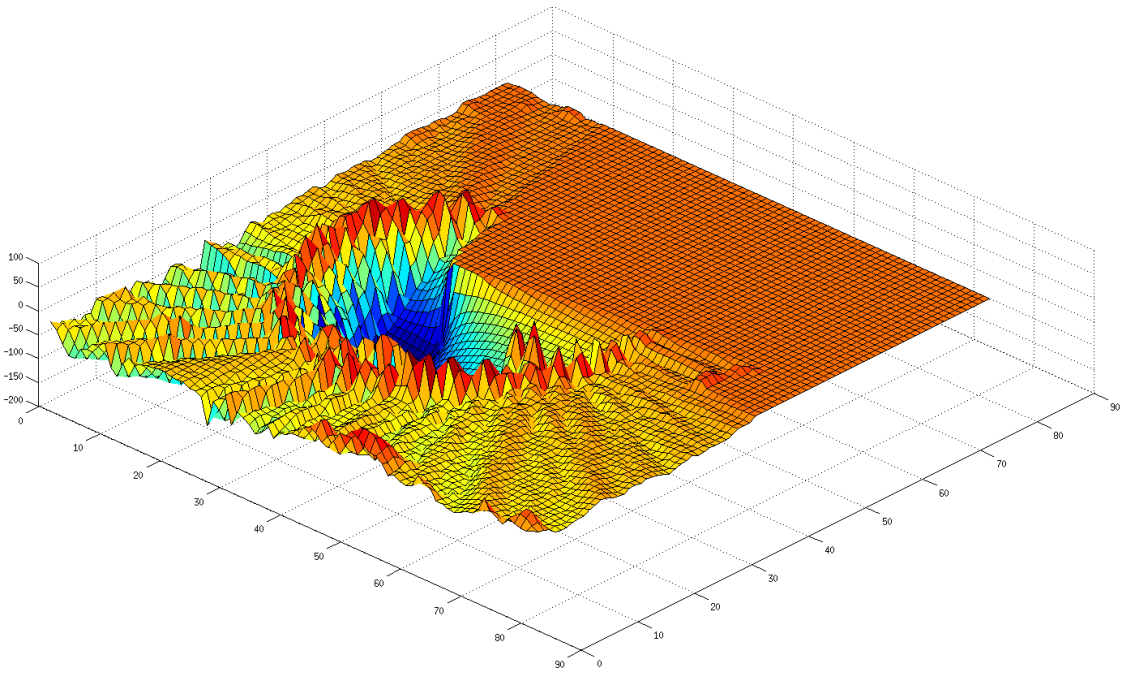

Figure 6: A surface map of the occupancy grid logit(probability) values.

6 Conclusions

In the first portion of this lab experiment, an EKF state estimator was implemented on the robotic platform to accurately estimate the robot's position. This estimator was implemented ontop of previously developed controllers for steering and velocity. Although the accuracy of the available measurement system is sufficient to obviate the need for an EKF estimator, it was found that the derived EKF estimator would provide excellent results if measurements were less accurate.

In the second portion of the experiment, a LIDAR system was used to dynamically generate an environment map in the form of an occupancy grid. The produced occupancy grid was found to be a reasonable representation of the robot's true environment.