Stefan B. Williams
March, 2016

MTRX5700: Experimental Robotics

Assignment 3

Note: This assignment contributes 10% towards your final mark. This assignment is due on Wednesday, May 4th during **Week 9** before 5pm or via email prior to that time. The demonstrations will be conducted during the supplementary lab session in Week 9. Submit your report via the eLearning TurnItIn submission on the course website. Late assignments will not be marked unless accompanied by a valid special consideration form. Plagiarism will be dealt with in accordance with the University of Sydney plagiarism policy. The objective of this assignment is to explore data fusion and mapping techniques suitable for use with mobile robotic systems. The choice of tools will depend on your preference and level of proficiency in Matlab, Python and/or C/C++. **This assignment should take an average student 15-20π hours to complete to a passing grade**.

Total Marks: 60

The front page of your report should include:
- Your name and SID
- The names of your group members (you are encouraged to work in small groups of 2 or 3)

1 Data Fusion

One important aspect of robotic navigation is the ability to fuse multiple sources of data. In the case of a mobile robot, we might have a number of sensors telling us our current position. Each of these sensors is subject to noise and errors of various kinds. Some sources of position data, such as triangulated range and bearing observations to known targets, are often quite noisy and subject to short term errors. Other position information, such as dead reckoning, is subject to an accumulation of errors that result from inaccuracies in our model and noise on the control lines. The fusion of these two sources of data can, however, give us much better results since one is good over the long term while the other is fairly reliable for predicting our position over a short distance.

We have given you a number of data files collected during deployment of a simulated robotic vehicle. **Write a robot position estimator that will fuse the data received from the vehicle to generate a reliable estimate of the robot's position.** The information you are interested in is the velocity and turn rate information derived from the vehicle's wheel encoders and observations from a compass, global positioning system and to the laser strips in the environment. In this case the predictions will act as a low pass filter for the noisy pose estimates. There are typically three stages involved in an estimator. The first stage is the prediction stage in which we use the control data to predict the position of the vehicle given a vehicle model. When an observation such as a range and bearing to a beacon is made the estimate of the vehicle position can be updated.

1.1 Prediction Stage

In the prediction stage we will use a simple kinematic based vehicle model. Vehicle control signals consisting of velocity and steer angle can be retrieved from the robot. The vehicle model and update equations are included here for reference purposes.

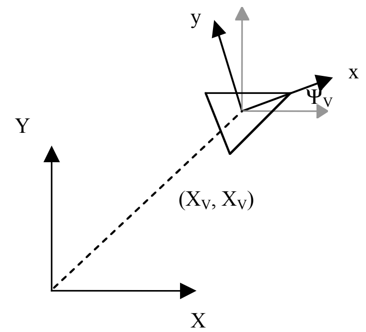


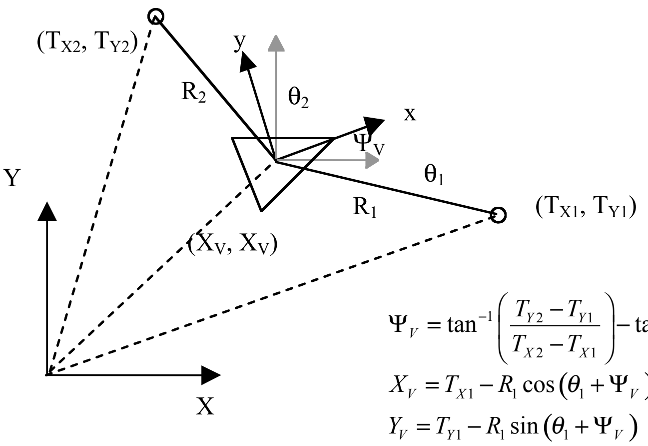
Figure 1 Vehicle model

$$\begin{bmatrix} \hat{X}_V(k|k-1) \\ \hat{Y}_V(k|k-1) \\ \hat{\Psi}_V(k|k-1) \end{bmatrix} = \begin{bmatrix} \hat{X}_V(k-1|k-1) + \Delta t V(k)\cos(\hat{\Psi}_V(k-1|k-1)) \\ \hat{Y}_V(k-1|k-1) + \Delta t V(k)\sin(\hat{\Psi}_V(k-1|k-1)) \\ \hat{\Psi}_V(k-1|k-1) + \Delta t \dot{\Psi}(k) \end{bmatrix} \quad (1)$$

where V(k) is the measured vehicle velocity and $\dot{\Psi}(k)$ is the turnrate at time k. The hat denotes the fact that this is our estimate of the value while the (k|k-1) indicates that this is the estimate of the state at time k given the information available at time k-1.

1.2 The Observation Stage

The observation stage consists of a series of observations that arrive from other sensors. In this case, the vehicle is equipped with a laser range finder and a noisy GPS style system. Start by using the noisy position and compass observations we have provided. Observations of retro-reflective beacons in the environment can also be used to provide a noisy position estimate if the position of the beacons is known. This signal is likely to be quite accurate over the long term (i.e. at low frequency) assuming that you can associate the observations to the correct beacon but will be subject to high frequency noise induced by errors in the sensor readings. The update stage will be used to filter out this noise.



$$\Psi_V = \tan^{-1}\left(\frac{T_{Y2}-T_{Y1}}{T_{X2}-T_{X1}}\right) - \tan^{-1}\left(\frac{R_2\sin\theta_2 - R_1\sin\theta_1}{R_2\cos\theta_2 - R_1\cos\theta_1}\right)$$
$$X_V = T_{X1} - R_1\cos(\theta_1 + \Psi_V)$$
$$Y_V = T_{Y1} - R_1\sin(\theta_1 + \Psi_V)$$

Figure 2 Computing the position of the vehicle P in a global reference frame using Range and Bearing information from the known targetsT_1 and T_2.

1.3 The Update Stage

The update stage is responsible for fusing the predictions and the observations to provide a more accurate estimate of the true position of the vehicle. The update equations for a simple low pass filter are

$$\begin{bmatrix} \hat{X}_V(k\,|\,k) \\ \hat{Y}_V(k\,|\,k) \end{bmatrix} = \left(1-\alpha_p\right)\begin{bmatrix} \hat{X}_V(k\,|\,k-1) \\ \hat{Y}_V(k\,|\,k-1) \end{bmatrix} + \alpha_p \begin{bmatrix} X_{Obs}(k) \\ Y_{Obs}(k) \end{bmatrix} \tag{2}$$

$$\left[\hat{\Psi}_V(k\,|\,k)\right] = \left(1-\alpha_\Psi\right)\left[\hat{\Psi}_V(k\,|\,k-1)\right] + \alpha_\Psi\left[\Psi_{Obs}(k)\right] \tag{3}$$

where the constants α_i are weighting factors that represent our confidence in the observed data. If we trust our predictions more than our observations α would be small.

We'd like you to change the value of α to observe what happens to the estimate. Use the values of α in the range of 0 to 1. Describe what happens to the estimated path and the obstacle map in each case. The relationship between the prediction, observations and update is shown in Figure 3.

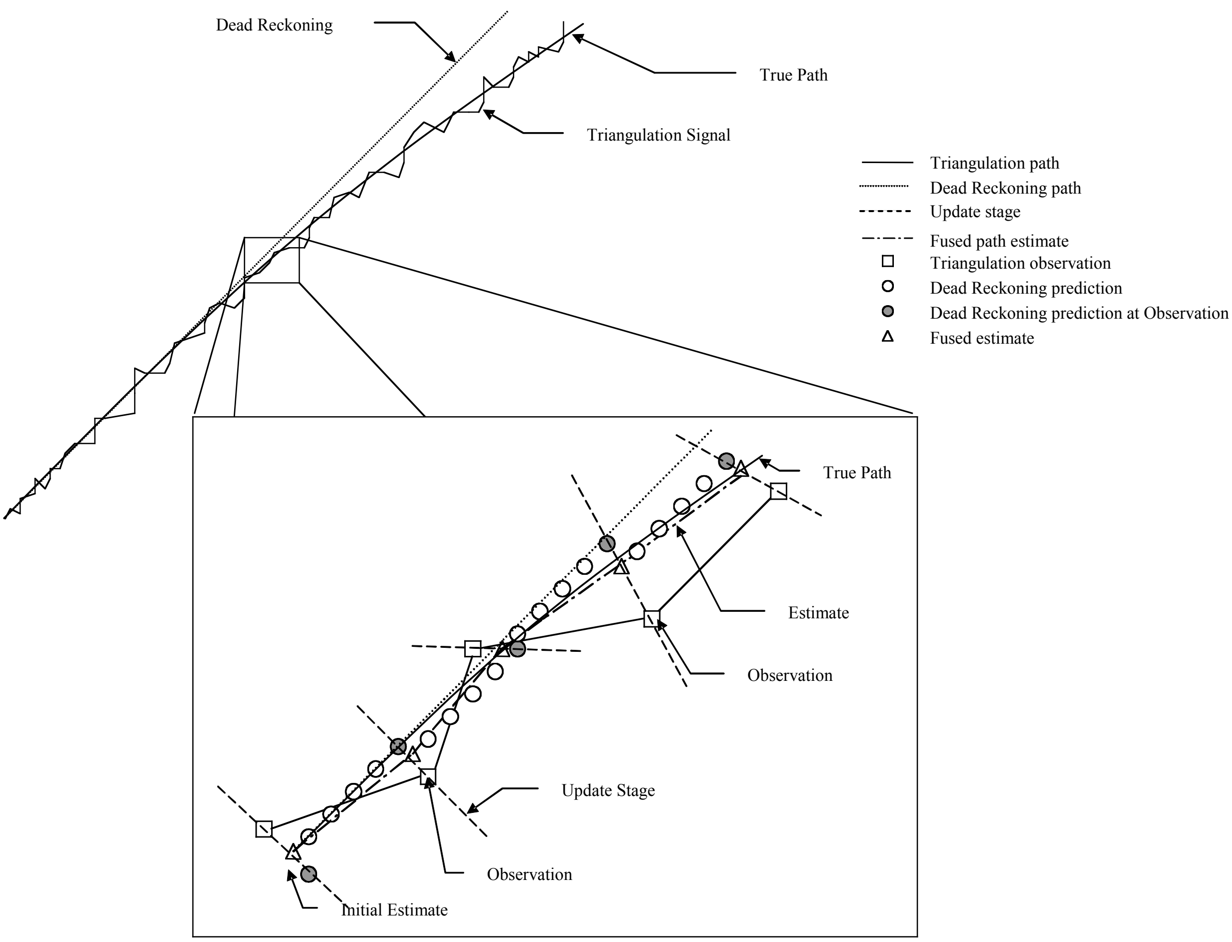


Figure 3 The Prediction-Observation-Update stages of the estimator. Notice that the estimate always lies between the Prediction and the Observation.

2 Occupancy Grid Mapping

The occupancy grid is a fairly simple representation of an obstacle map. It consists of dividing an area into equal squares and checking for the existence of obstacles in each grid location. When an obstacle is detected in a particular location in space, the count in that grid space is incremented. When the count exceeds a certain threshold (set to ignore spurious data readings) the grid square is considered to be occupied by an obstacle (see Figure 4).

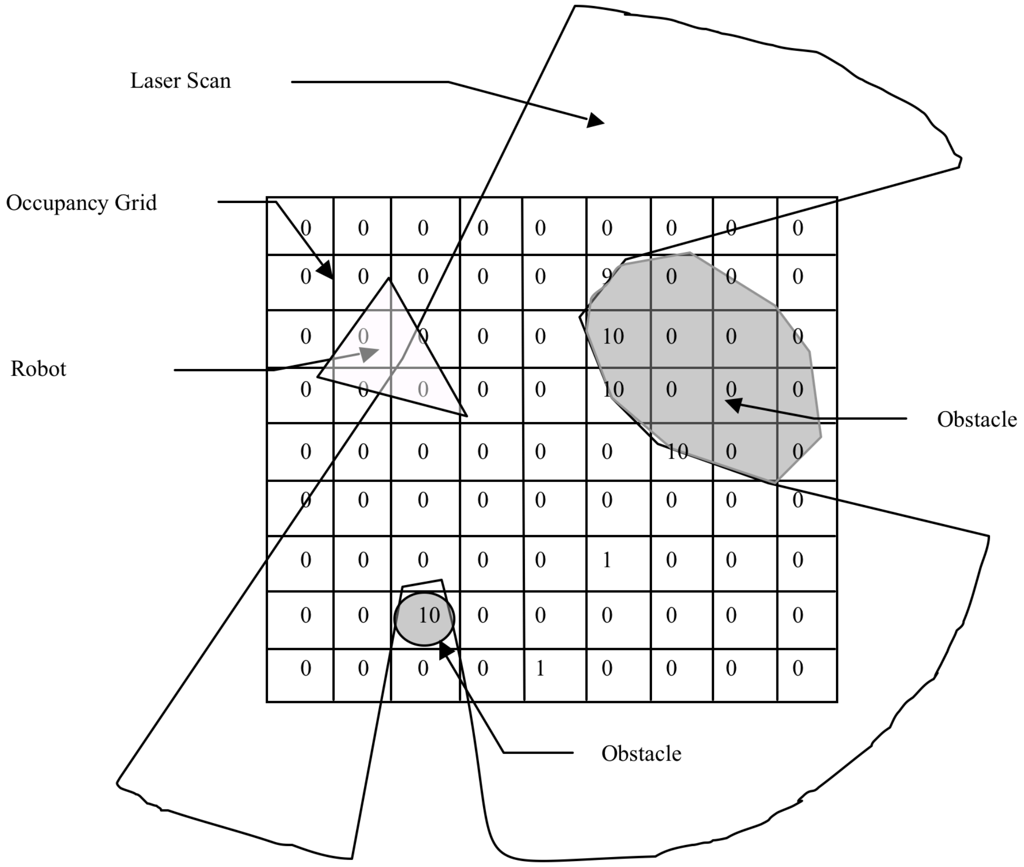


Figure 4 Occupancy grid after 10 scans

Our robot is equipped with a laser range scanner that gives us range and bearing information about obstacles in the environment. The robot is moving around an area with the laser positioned on the centre of the robot. Whenever a laser reading arrives, the data should be analysed to check for obstacles and the occupancy grid should be updated using the current position estimate.

What happens to the quality of the map when you use different filter values to generate different path estimates? Note that when there is no obstacle at a particular bearing, the laser returns its maximum range of 8.0m. This should not be considered a valid return and should not be used to update obstacle positions.

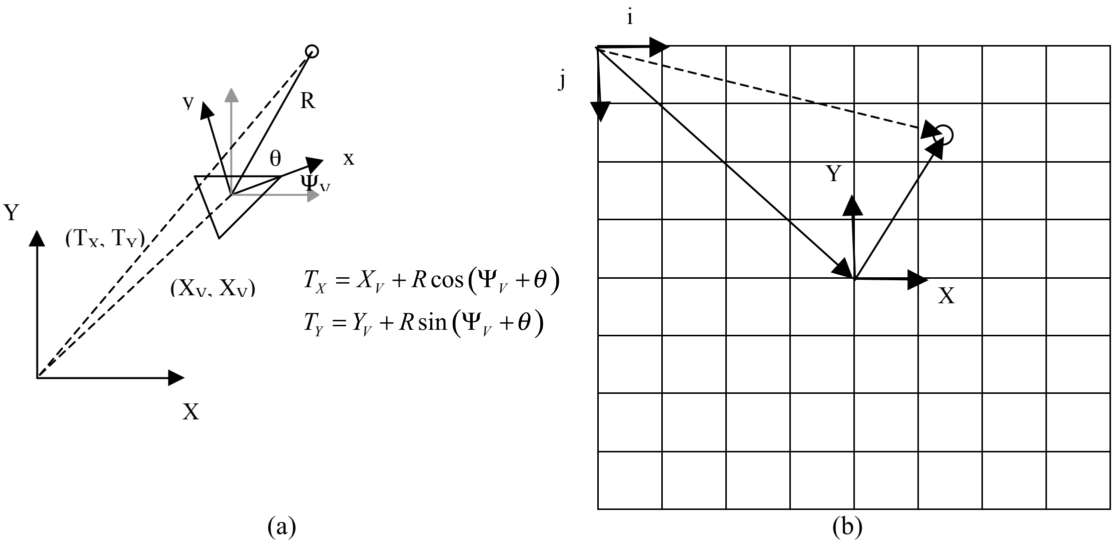


Figure 5 (a) Computing the position of a target T in a global reference frame using Range and Bearing information from the moving reference frame P on a robot (b) There is also a direct relationship between the position in world coordinates and the location of the grid cell. This relationship is a function of the size of the occupancy grid, its resolution and the relative position of the origin within the map.

Your second task in this assignment will be to generate an occupancy grid map of the environment in which the robot is operating. We will use the position estimates of the vehicle pose you generated earlier as your reference. You should create an array of integers that is large enough to accommodate the environment in which the robot is working and that has enough resolution to discriminate targets. You should provide a mechanism for changing the resolution of the grid and experiment with different map resolutions. Consider how the storage requirement changes in terms of the size and resolution of the map. Each time a laser reading is received, it should be translated from world coordinates (i.e. (T_x, T_y) in Figure 5) to grid coordinates (i, j) that will act as an index into your grid. The value of that grid cell can then be incremented. Once you have finished creating the map, select an appropriate threshold and transform the array into a binary array of ones and zeros representing occupied and unoccupied space. Find some means of displaying this data in a meaningful format (i.e. Matlab, Excel or as ASCII text). Alternatively, you may wish to investigate methods for creating bitmaps from an arbitrary array from within your program or, if you have experience in this area, you may wish to create a graphical user interface (in Matlab, C++ or Java perhaps) to display the data. This could be done in 'real-time' as the data is being received from the robot or offline once the occupancy grid is complete.

3 Bonus! Real Data

We have provided you with a second data set collected using a robot equipped with wheel encoders, a localisation system, and a Hokuyo laser rangefinder. The format of the data will match the synthetic data already provided. **Your task is to rerun the path estimation and occupancy grid generation components of the assignment using this "real" data set.**

The positioning observations will be noisy. There are likely to be observations that are wildly incorrect. **How can we detect these outliers? What effect do they have on the quality of the path estimate and map? How can we eliminate them? Show how the quality of your path estimate changes after mitigating the effect of outliers.**

4 Report & Marking

You are to submit a brief report detailing the work you have undertaken as part of this lab. We are particularly interested in seeing a discussion of the principles covered by this lab. Your report should discuss your implementation of the data fusion and occupancy grid methods discussed above. We expect that you will understand and be able to explain the fundamental principles behind estimation and be able to relate the results you obtain to those principles. Plots of the Prediction-Observation-Update cycle similar to the one shown in this handout but using real data will help you to demonstrate your understanding of how the filter is working. We'd also like some discussion relating to the software design decisions you made in the implementation of the filter and the occupancy grid.

A demonstration of your working system will be expected during the supplementary lab session in Week 9. You and your group members will be asked to answer questions about your demonstration. You must be present at the demonstration to receive demonstration marks and are expected to be able to answer questions relevant to all aspects of the system you and your team members are demonstrating.

The grading will be based 50% on the demonstration of the system operation, including a discussion of the system design, and 50% on the report. The grading will fall roughly into the following divisions:

- Pass: implement occupancy grip using dead reckoned position estimates. Some elementary form of plotting of the grid.
- Credit: same as above plus a more elaborate display of the output. Also independent results of the position data fusion process should be shown
- Distinction: use of the fused position information to improve the resulting map. Answering of the questions posed throughout the lab handout
- High Distinction: a useful front end added to the program to allow the user to see the results being generated on the fly while the program is connected via Player to the robot and/or simulator.
- Bonus: Complete Section 3 (Real Data) of the assignment. Partial marks will be awarded for path estimation and mapping without considering outliers in the acoustic data. Full marks will be awarded for improving your results by considering these outliers and answering the related questions posed.

Marks will be assigned according to the following breakdown:

1.	Demonstration	30
2.	Description of the problem	5
3.	Data Fusion Results	10
4.	Mapping Results	10
5.	Presentation	5
6.	TOTAL	60
7.	BONUS	5