MTRN 4068 Wireless Mechatronics

Test#1 (Assignment): Robot Mapping and Navigation 2022

Note: This assignment contributes 10% towards your final mark. This assignment is due on Sunday, August 21st 2022 before 11:59pm. Submit your report via the iLearn TurnItIn submission on the unit website. Late assignments will not be marked unless accompanied by a valid special consideration form. Plagiarism will be dealt with in accordance with the Macquarie University 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 Python, Matlab and/or C/C++. **This assignment should take an average student 10 hours to complete to a passing grade**. You should work individually on completing this assignment.

Total Marks: 100

The front page of your report should include:

- Your SID and name.

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 the current position of the robot. 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. In this assignment, you will **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 laser range and bearing observations to 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 or a gps observation is made the estimate of the vehicle position can be updated.

We would like you to write code that implements a filter that fuses the information provided by vehicle encoder data with observations of the position and orientation of the vehicle as well as observations of known beacons in the robot's environment.

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.

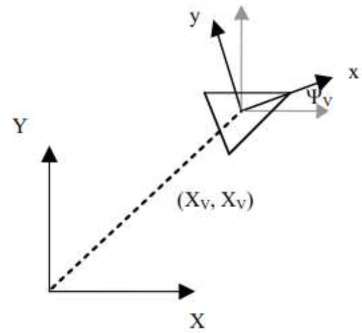


Figure 1 Vehicle model

The prediction stage will simply update the previous estimate according to Equation (1).

$$\mathbf{x}_{k+1}^{-} = f(\mathbf{x}_k^{+}, \mathbf{u}_k) \tag{1}$$

For our 2D robot where the state is represented by the (x, y) position and heading, ψ, of the vehicle, this can be written as:

$$\begin{bmatrix} x_{k+1}^{-} \\ y_{k+1}^{-} \\ \psi_{k+1}^{-} \end{bmatrix} = \begin{bmatrix} x_k^{+} + \Delta t v_k cos\psi_k^{+} \\ y_k^{+} + \Delta t v_k sin\psi_k^{+} \\ \psi_k^{+} + \Delta t\dot{\psi}_k \end{bmatrix} \tag{2}$$

where v_k is the measured vehicle velocity and $\dot{\psi}_k$ is the turnrate at time k. The subscripts k and k+1 represent the timestamp while the superscript $^{+/-}$ represent the posterior and prior estimate.

1.2 The Observation/Update Stage

The observation/update stage consists of fusing a series of observations that arrive from other sensors with the prior estimate generated by the prediction stage. In this case, the vehicle is equipped with a laser range finder, a noisy GPS style system and a compass. Start by using the noisy position and compass observations we have provided. As part of your report, describe the process you have taken toselect the filter parameters. Demonstrate how different filter parameters affect the estimated vehicle pose through time. The relationship between the prediction, observations and update is shown in Figure 2.

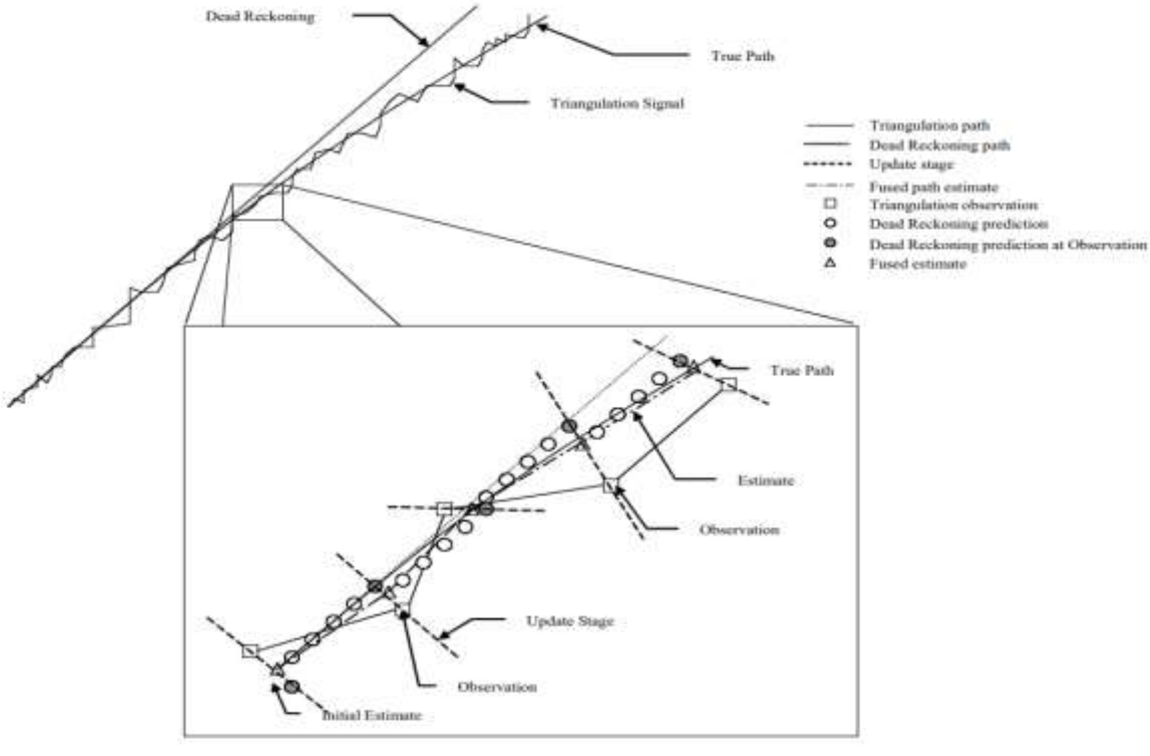


Figure 2 The Prediction-Observation-Update stages of the estimator. Notice that the estimate always lies between the Prediction and the Observation.

We will be fusing the prior estimate with the observation using a filter parameter. This is effectively a low pass filter parameter with bandpass determined by the filter parameter α_p.

$$\mathbf{x}_{k+1}^{+} = (1-\alpha_i)\mathbf{x}_{k+1}^{-} + \alpha_i \mathbf{z}_{\mathrm{k+1}} \tag{4}$$

Rearranging this equation, we can also think about this as updating the posterior estimate of the state by adding an update that is a fraction of the difference between the estimate and the observation.

$$\mathbf{x}_{k+1}^{+} = \mathbf{x}_{k+1}^{-} + \alpha_i(\mathbf{z}_{\mathrm{k+1}} - \mathbf{x}_{k+1}^{-}) \tag{5}$$

As you vary α_i from 0 to 1, you move from trusting the prediction entirely to trusting the observation.

1.2.1 GPS observation

The GPS observations consist of observations of the (x, y) location of the robot in the global frame of reference. If you plot these observations, you'll see that they are noisy but will give you an overview of the path the vehicle has followed. You can fuse these observations with the prior estimate resulting from the prediction stage to update the estimate of the vehicle's location. The noisy observations provided by the GPS sensor will be filtered by the smoother path provided by the prediction stage. Theparameter α_p can be used to tune the amount of faith that the filter puts in the new observation versus the estimate.

$$\begin{bmatrix} x_{k+1}^{+} \\ y_{k+1}^{+} \end{bmatrix} = \begin{bmatrix} x_{k+1}^{-} + \alpha_p(z_x - x_{k+1}^{-}) \\ y_{k+1}^{-} + \alpha_p(z_y - y_{k+1}^{-}) \end{bmatrix} \tag{6}$$

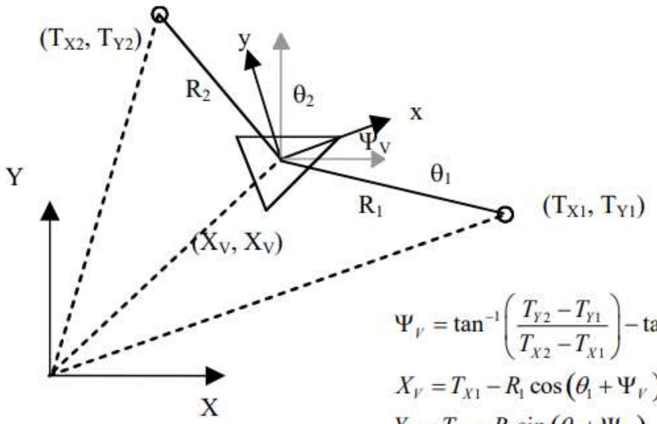
1.2.2 Compass observation

The compass observations consist of observations of the heading of the robot. If you plot these observations, you'll see that they are noisy but give an idea of the heading of the vehicle through the deployment. You can fuse these observations with the prior estimate of heading resulting from the prediction stage to update the estimate of the vehicle's heading. The noisy observations provided by the compass will be filtered by the smoother heading prior provided by the prediction stage. Note thatyou must ensure to account for angular wrap-around when computing the difference between the observation and estimated observation (i.e. the difference should be restricted to $\pm\pi$).

$$[\psi_{k+1}^{+}] = [\psi_{k+1}^{-} + \alpha_{\psi}(z_{\psi} - \psi_{k+1}^{-})] \tag{7}$$

1.2.3 Laser Beacon Observation

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 3 Computing the position of the vehicle P in a global reference frame using Range and Bearing information from the known targets T_1 and T_2.

We can use the estimated position available by using a pair of observed beaons to update our estimate of the vehicle position.

$$\begin{bmatrix} x_{k+1}^{+} \\ y_{k+1}^{+} \end{bmatrix} = \begin{bmatrix} x_{k+1}^{-} + \alpha_p(z_x - x_{k+1}^{-}) \\ y_{k+1}^{-} + \alpha_p(z_y - y_{k+1}^{-}) \end{bmatrix} \tag{8}$$

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).

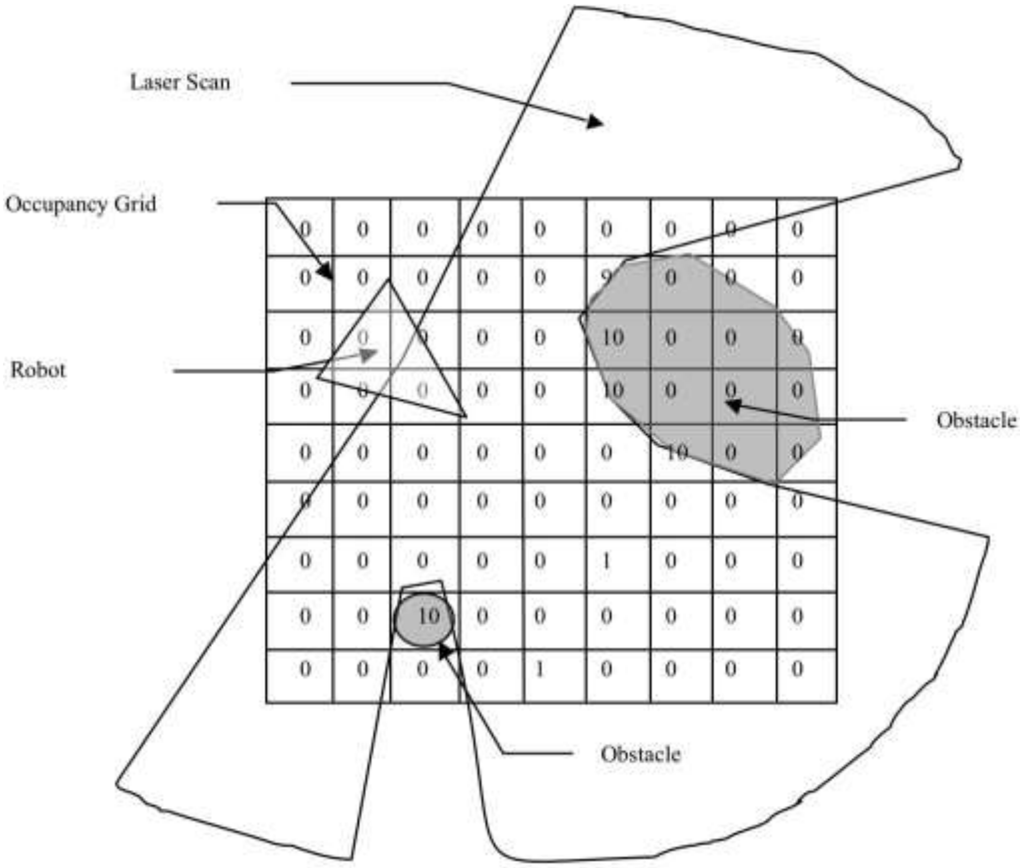


Figure 4 Occupancy grid after 10 scans

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.

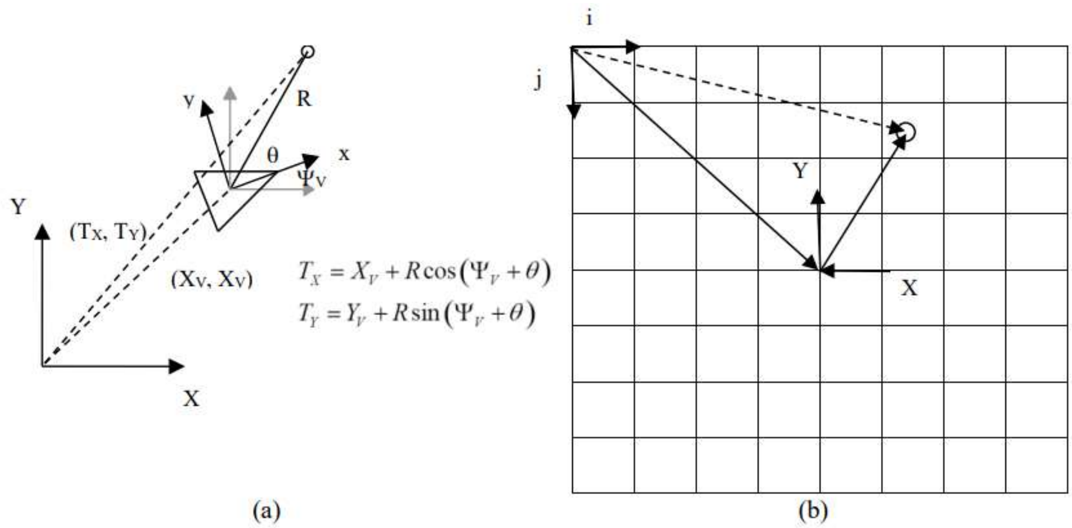


Figure 5 (a) Computing the position of a target T in a global reference frame using Range and Bearing information from the moving reference frame P on a robot (b) There is also a direct relationship between the position in world coordinates and the location of the grid cell. This relationship is a function 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 Python, 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 Report & Marking

You are to submit a brief report detailing the work along with the source code of the programme you have undertaken as part of this assignment. We are particularly interested in seeing a discussion of the principles covered by this assignment. 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. The understanding will be beneficial towards the implementation of the project activities. Plots of the Prediction-Observation-Update cycle similar to the one shown in this handout but using real data from your project will help you to demonstrate yourunderstanding 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.

The grading will fall roughly into the following divisions:

- Pass: implement occupancy grid 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.

Marks will be assigned according to the following breakdown:

1. Sensors and their details (in terms of specification, interfacing, data format, storage and analysis)	**30**
2. Data Fusion Results	**30**
3. Mapping Results	**30**
4. Report Style	**10**
TOTAL	100