

# COMP0130 Coursework 02: Graph-based Optimisation and SLAM

---

Assignment Project Exam Help

<https://tutorcs.com>

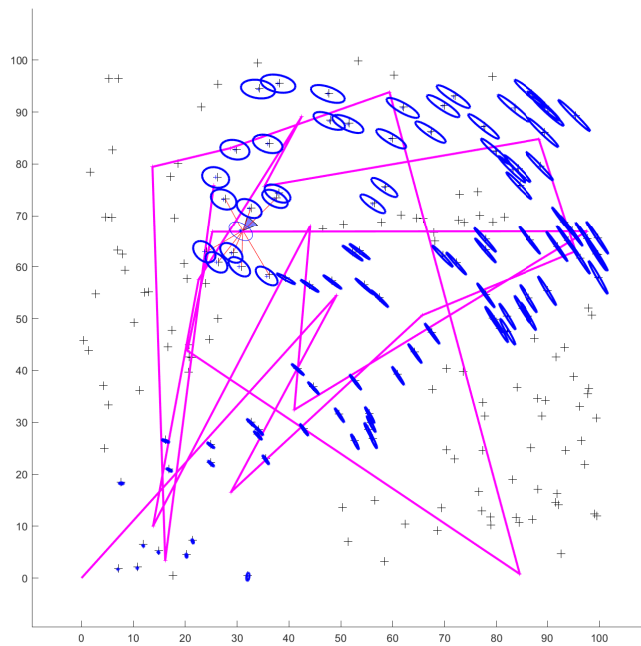
Current Revision: 20220305-001320

## 1 Overview **WeChat: cstutorcs**

- Assignment Release Date: Saturday 5th March, 2022
- Assignment Submission Date: 16:00 Friday 25th March, 2022
- Weighting: 33% of module total
- Final Submission Format: a PDF file containing a report, and a zip file containing source code.

## 2 Coursework Description

The goal of this coursework is to experiment with how a graphical-model based SLAM system is implemented and assess its properties. The system will be implemented using the MiniSLAM



## Assignment Project Exam Help

Figure 2.1: The scenario. The vehicle (small blue triangle, starting from (0,0) at bottom left) drives a path (purple line) through an environment populated by landmarks (blue crosses). Various landmark estimates, with their associated covariance ellipses, are shown.

event-based framework you met in Topic 02 Lab 01 and the MATLAB port of the g<sup>2</sup>o port you met in Topic 02 Lab 02. We assume you have familiarity with the questions and worked answers to those workshops.

The scenario is shown in Figure 2.1. A wheeled robot (DriveBot) drives through a two-dimensional environment which is populated by a set of landmarks. The number of landmarks and their locations are not known, and must be estimated by the SLAM system.

The vehicle is equipped with three types of sensors:

1. A vehicle odometry sensor, which records speed and heading.
2. A “GPS” sensor which measures vehicle position.
3. A 2D range bearing sensor which measures the range and bearing to the landmarks in 2D.

In addition, an external is provided to specify the initial position and orientation of the robot.

The mathematical models for these data sources and sensors are described in Appendix A.

A simulation system will create a stream of events which the SLAM system will need to process. Data association has already been addressed. Therefore, each landmark measurement you receive will have a unique (and correct) landmark ID.

The coursework consists of three questions:

Q1: Implement the prediction and update steps using the vehicle process model and the GPS sensor. You will also investigate some general properties about how the graph optimizers work. (28%)

Q2: Extend your implementation to undertake SLAM. (34%)

Q3: Discuss and implement a couple of ways improve the performance of the SLAM system (38%).

Please note that different subparts of different questions have their own launcher scripts of the form `qx_y.m`. Although these scripts look very similar to one another, there are subtle differences between them to support the different subquestions (for example, some datasets are considerably longer than others). Furthermore, to mark your coursework, we will execute each script in turn and so the changes required to answer each question should not interfere with us being able to run the launcher code for a previous question.

### 3 Questions

1. In this question you will implement, within the `DriveBotSLAMSystem` class, the prediction step (using the process model in Subsection A.2) and the GPS sensor (using the observation model in Subsection A.3). Note that the GPS data arrives less frequently than the vehicle odometry data, and *you must ensure that timestamps are handled correctly*.

- a. Describe the structure of the graph which will be used to support predicting the robot through time and updating its position periodically with the GPS sensor. Your description should list the different types of vertices and edges. Provide the equations for the different types of edges and vertices.

*Hint:* Think about how angular discontinuities affect your solution and how these are related to the  $\oplus$  operator.

[6 marks]

b. Implement the prediction step in the `DriveBotSLAMSystem`.

- i. In your report describe which methods you edited to complete the functionality. In the code, please provide comments which explain what your implementation does. These should be prefixed by “Q1b”.

[10 marks]

- ii. Using the script `q1_b.m`, plot the graphs of the state error and the covariances computed by the estimator. Describe the behaviours you see and explain why you think they arise.

[7 marks]

c. Implement the GPS update step in the `DriveBotSLAMSystem` class.

- i. In your report describe which methods you edited to complete the functionality. In the code, please provide comments which explain what your implementation does. These should be prefixed by “Q1c”.

[8 marks]

- ii. Using the script `q1_c.m` generate plots of the state error and the covariance. Present these graphs. Discuss the behaviours you see and why they arise.

[8 marks]

d. Modify the script `q1_d.m` so that the optimizer runs once every timestep instead of at the end of the run. Plot the curves of the optimization time and the `chi2` over the run. Describe the behaviours you observe and propose an explanation for why they arise.

[10 marks]

[Total 49 marks]

2. In this question you will extend the `DriveBotSLAMSystem` to support SLAM. The observation model for these landmarks is given in Subsection A.4. The GPS sensor is not used.

- a. Describe the structure of the graph for the SLAM system. Your description should include a discussion on the different types of edges and vertices and you should include the error models and Jacobians needed to implement them.

*Hint:* Think about how angular discontinuities affect your solution and how these are related to the  $\oplus$  operator.

[11 marks]

- b. Implement SLAM in `DriveBotSLAMSystem`.

- i. In your report describe which methods you edited to complete the functionality.

In the code, please provide comments which explain what your implementation does. These should be prefixed by “Q2b”.

[9 marks]

- ii. Run the script `q2_b.m` which runs a SLAM example on a single closed loop.

Plot the vehicle covariances, `chi2` values and the amount of time to complete an optimization step. Discuss the behaviours you see and why you think they arise.

[15 marks]

c. We will now look at some more details about the properties of the constructed graph.

- i. By modifying the script `q2_c.m`, investigate the properties of the structure of the graph. You should obtain information on: the number of vehicle poses stored, the number of landmarks initialized, the average number of observations made by a robot at each timestep, and the average number of observations received by each landmark.

In your report, describe how you calculated these quantities. In the code, please prefix your changes by “Q2c”.

*Hint:* You can obtain all the necessary information from querying the constructed graph, Matlab’s `isa` function, and using methods in the `BaseVertex` and `BaseEdge` classes.

[7 marks]

- ii. Present the results you obtained from analysing the graph and discuss what these quantities imply.

[8 marks]

- <https://tutorcs.com>
- d. One of most important advantages of SLAM is the ability for it to perform loop closure.

**WeChat: cstutorcs**

Using the script `q2_d.m`, generate figures which show the map, vehicle and landmark states just before and just after the loop closure event. Explain the form of the covariance matrices you see on the landmark and vehicle estimates immediately before and after the update. By looking at the determinant of the covariance of each landmark, compute the amount by which the uncertainty has changed for each landmark.

[10 marks]

[Total 60 marks]

- e. This is a purely optional activity for those who are interested. No marks are awarded for it.

Run the script `q2_large_map.m`. This tests the code on a much more substantial example. The landmark covariances matrices look like they are oriented as tangents on a set of concentric circles. Where do you think the origin might be, and what do you think might cause it?

**Assignment Project Exam Help**

**<https://tutorcs.com>**

**WeChat: cstutorcs**

3. This question explores several ways to improve the performance of a SLAM system by deleting parts of the factor graph. Again, the GPS is not used.

a. One way to improve the performance of a SLAM system is to maintain all the sensor observation data, but to remove all the vehicle prediction edges.

i. By referring to the structure of the factor graph, why do you think this might be a sensible strategy to take? When do you think this is likely to be a successful strategy? When is it likely to be unsuccessful?

[8 marks]

ii. Modify the `DriveBotSLAMSystem` system so that, when requested, it will delete the vehicle edges before optimization. Your code should support two cases: when all of the prediction edges are deleted, and when all but the first prediction edge should be deleted.

In your report, describe which methods you edited to provide this functionality. In the code, please provide comments which explain what your implementation does. These should be prefixed by “Q3a”.

<https://tutorcs.com>

[4 marks]

iii. Using the script `q3_a.m`, present the three cases: all the prediction edges maintained, all prediction edges removed, and all but the first prediction edge is removed. By referring to the structure of the graph, explain the differences in performance and behaviour.

*Hint:* you should find that the algorithm will fail if all the prediction edges are removed.

[8 marks]

iv. Plot the optimization time, covariance and chi2 values for the working case where the edges have been removed. How do the results differ from the full SLAM system? Given the difference in performance, discuss if you this strategy on its own successful.

[8 marks]



b. Two approaches for simplifying graphs have been proposed: sparse keyframe based SLAM, and graph pruning.

i. Describe how keyframe-based SLAM and graph pruning methods work. Discuss their strengths and weaknesses.

[8 marks]

ii. Chose one strategy — either sparse key frames or graph pruning — to implement. Explain which one you chose and why.

[3 marks]

iii. Implement your solution. You can either modify the existing code (but ensure that it will run the earlier questions correctly) or you can create new classes to implement it (either by subclassing or simply copying the existing code to create a new class). To accompany it, you should write your own script `q3_b.m`, following the pattern of the other scripts, to run your revised SLAM system with the necessary settings.

In your report, describe how you implemented the functionality. In the code, please provide comments which explain what your implementation does. These should be prefixed by “Q3b”.

[15 marks]

iv. Evaluate the results of your algorithm. How you chose this will depend upon the algorithm implemented. However, key characteristics to include are the graph properties (such as number of vertices), chi2 performance, and error performance. Your evaluation should show whether your changes have improved performance, and provide an explanation of why you think you achieved the results presented.

[12 marks]

[Total 66 marks]

[Coursework Total 175 marks]

## A System Models

### A.1 State Description

The vehicle state is described by its position and orientation in 2D:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix}. \quad (\text{A.1})$$

A standard right handed coordinate system is used:  $x$  points to the right,  $y$  points up and a positive rotation is in an anticlockwise direction. Angles are in radians.

Landmarks are in 2D. The state of the  $i$ th landmark is given by

$$\mathbf{m}^i = \begin{bmatrix} x^i \\ y^i \end{bmatrix}. \quad (\text{A.2})$$

### A.2 Process Model

The vehicle process model is the similar to the one you saw in Workshop 04. The model has the form,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta T_k \mathbf{M}_k (\mathbf{u}_k + \mathbf{v}_k). \quad (\text{A.3})$$

The matrix

$$\mathbf{M}_k = \begin{bmatrix} \cos \psi_k & -\sin \psi_k & 0 \\ \sin \psi_k & \cos \psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the rotation from the vehicle-fixed frame to the world-fixed frame and  $\Delta T_k$  is the length of the prediction interval. Note that the prediction interval is governed by when various sensors are available, and can vary through the simulation.

The control input consists of the speed of the vehicle (which is oriented along a body-fixed  $x$  axis) together with an angular velocity term,

$$\mathbf{u}_k = \begin{bmatrix} s_k \\ 0 \\ \dot{\psi}_k \end{bmatrix}.$$

The process noise is zero mean, Gaussian and additive on all three dimensions,

$$\mathbf{v}_k = \begin{bmatrix} v_k \\ v_y \\ v_\psi \end{bmatrix}.$$

The noise in the body-fixed  $y$  direction allows for slip and the fact, as discussed in the lectures, that the velocity is related to the front wheel and not the body orientation. The process noise covariance  $\mathbf{Q}_k$  is diagonal.

### A.3 GPS Observation Model

The GPS sensor is a highly idealized one which provides direct measurements of the position of the robot. The observation model is

$$\mathbf{z}_k^G = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \mathbf{w}_k^G$$

The covariance of  $\mathbf{w}_k^G$ ,  $\mathbf{R}_k^G$  is diagonal and constant.

<https://tutorcs.com>

### A.4 Landmark Observation Model

The landmark observation model measures the range, azimuth and elevation of a landmark relative to the platform frame,

$$\mathbf{z}_k^L = \begin{bmatrix} r_k^i \\ \beta_k^i \end{bmatrix} + \mathbf{w}_k^L,$$

where

$$r_k^i = \sqrt{(x^i - x_k)^2 + (y^i - y_k)^2}$$

$$\beta_k^i = \tan^{-1} \left( \frac{y^i - y_k}{x^i - x_k} \right) - \phi_k$$

The covariance of  $\mathbf{w}_k^L$ ,  $\mathbf{R}_k^L$  is diagonal and is assumed to be time invariant and the same for all landmarks.