CPE470/670

**Project 3. Consensus Filters for a Multi-robot system (100 points)**

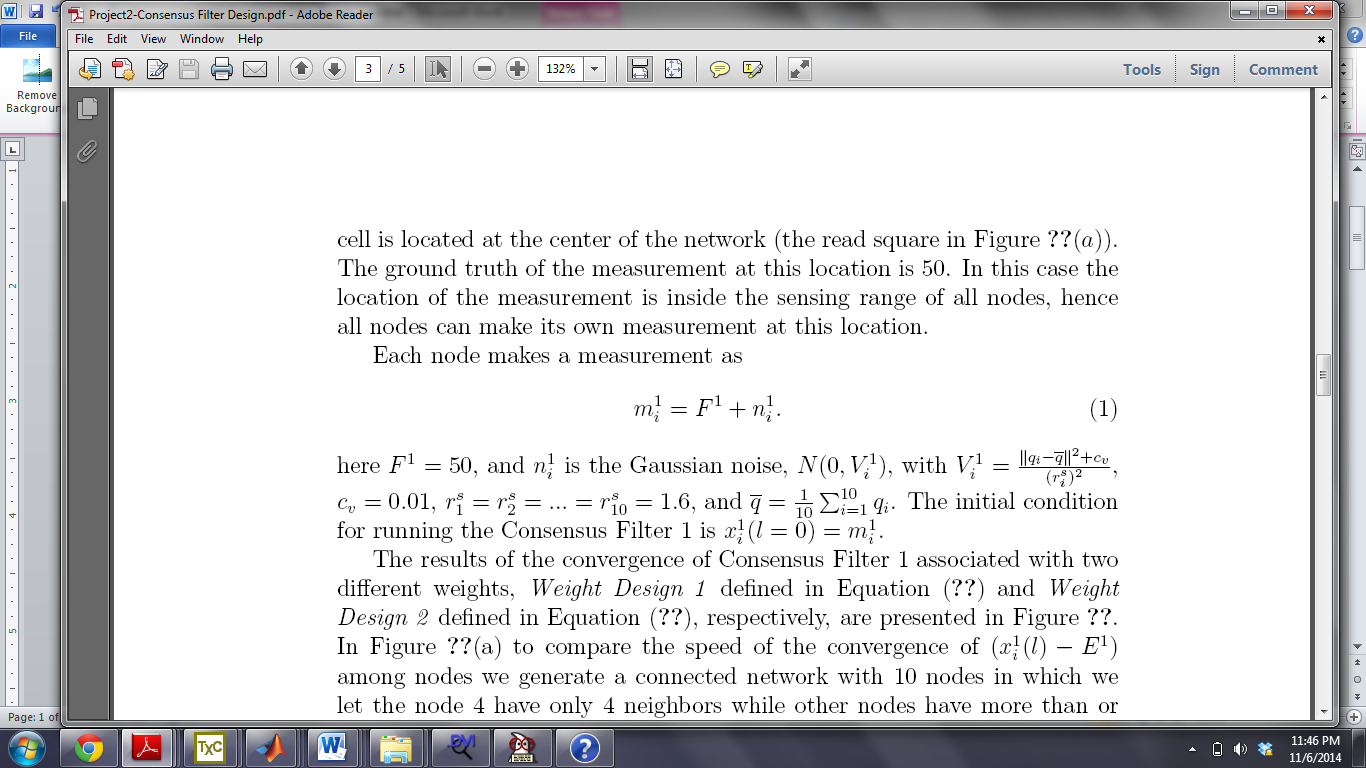
**Project Deadline: May 3rd, 2022.**

**Submit your project to CANVAS with source code.**

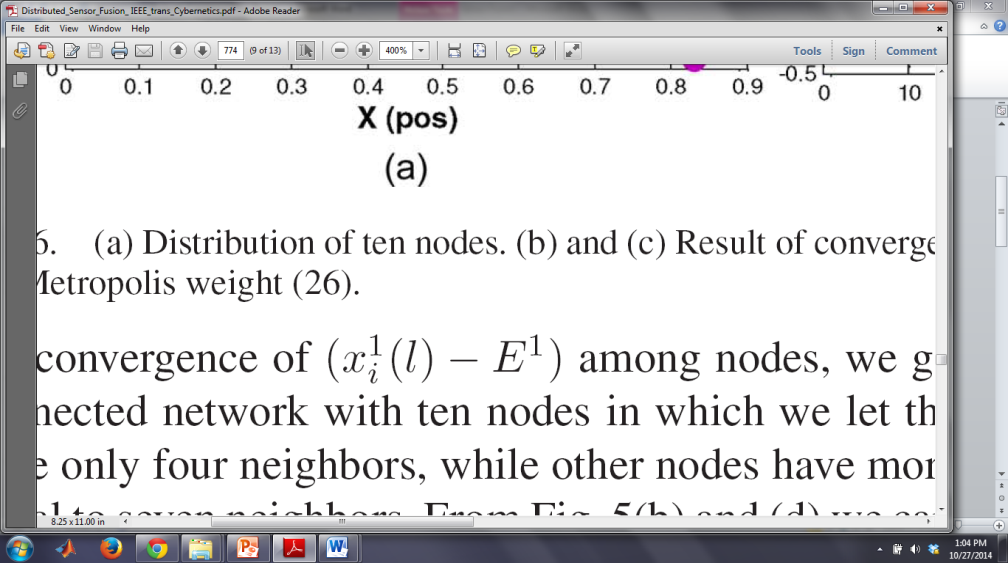
**Case 1. Estimate single cell (single scalar value): 50 points**

Graphical user interface, text, application, Word

Description automatically generatedWe randomly generate a connected network of 10 nodes in the area of 4x4. The cell is located at the mean of the sensor positions or qc =  The ground truth of the measurement at this location is 50.



1. Show the results of the convergence of consensus filter 1 (Weighted Average Consensus) associated with two different weights, i.e., *weight design 1* and *weight design 2*. Explain the obtained results.(UG: 20points; G:15points)
2. Show the results of the convergence of consensus filter 2 (Average Consensus) with both Max-degree and Metropolis weights for a network of 10 nodes (area of 4x4), and a network of 50 nodes (area of 20x20). Explain the obtained results. (UG: 20points; G:15points)
3. Show the convergence of the node which has the smallest number of neighbors and the node which has the largest number of neighbors for both networks of 10 and 50 nodes, respectively. Observe the obtained results and give explanation. (UG: 10points; G:10points)



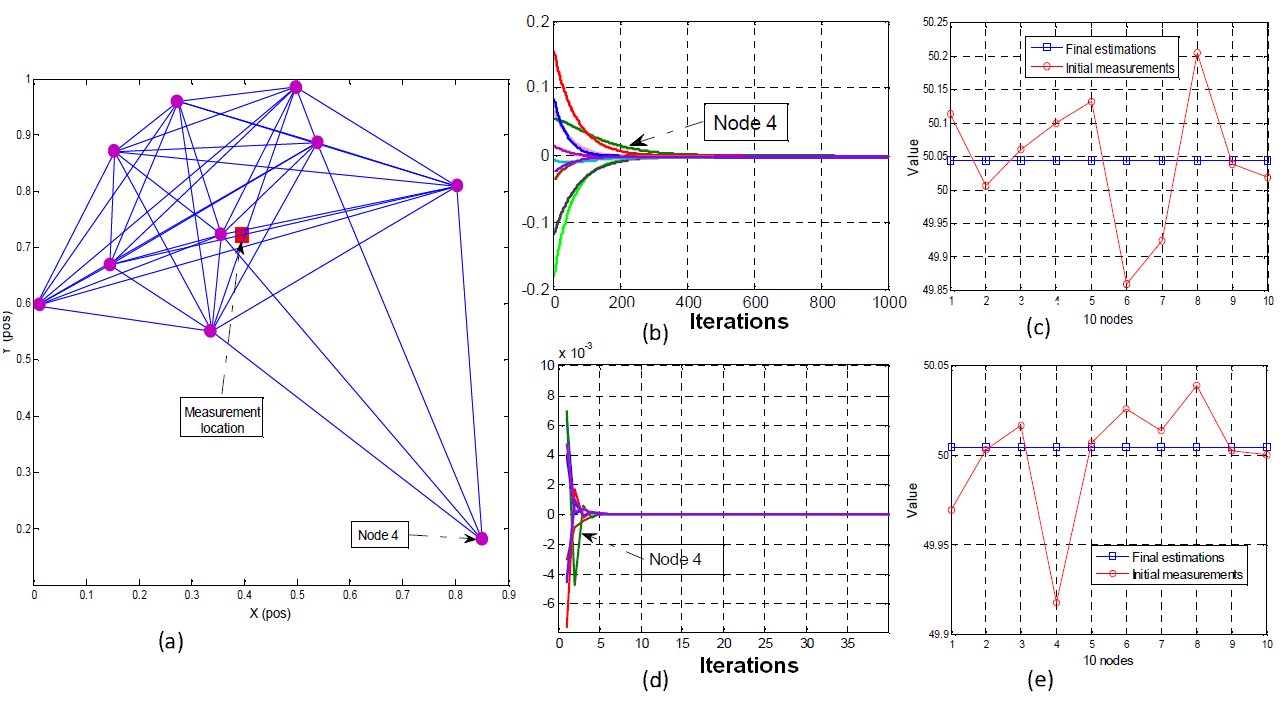
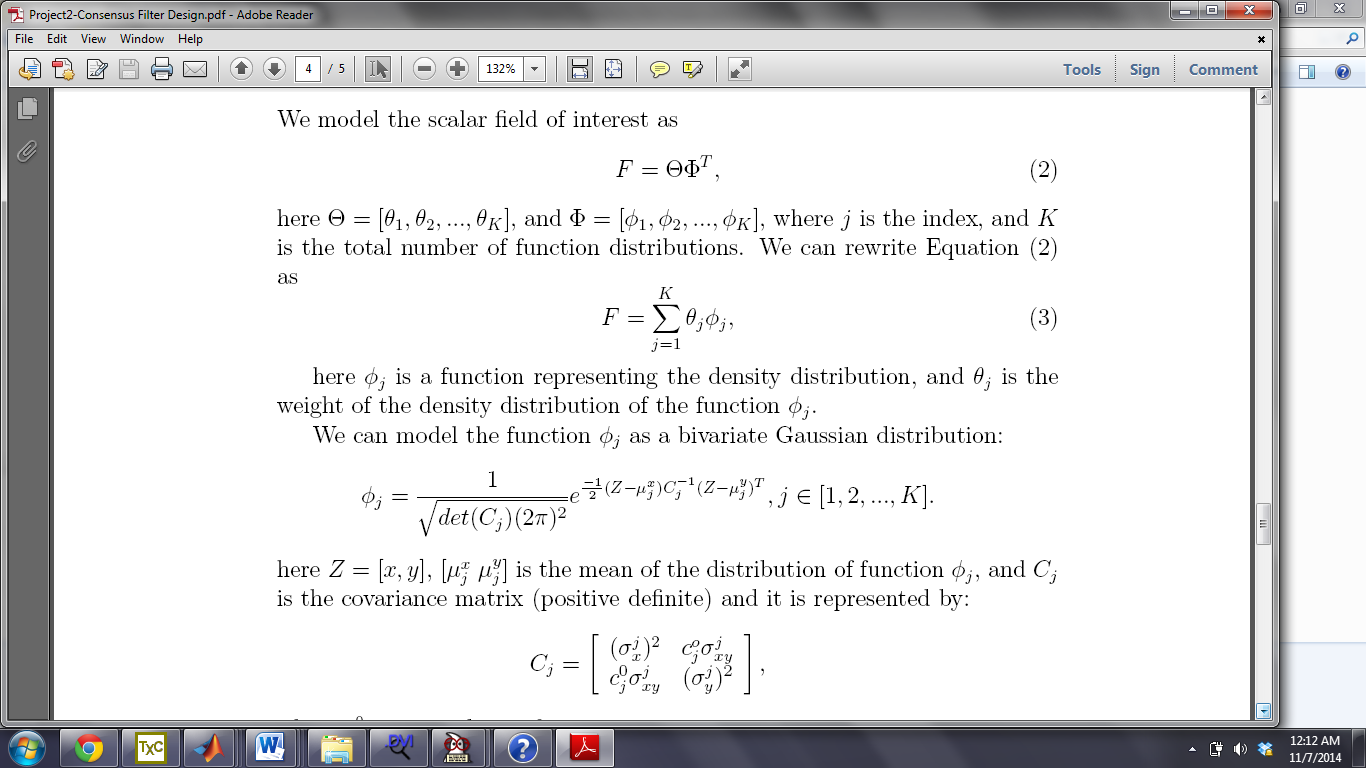


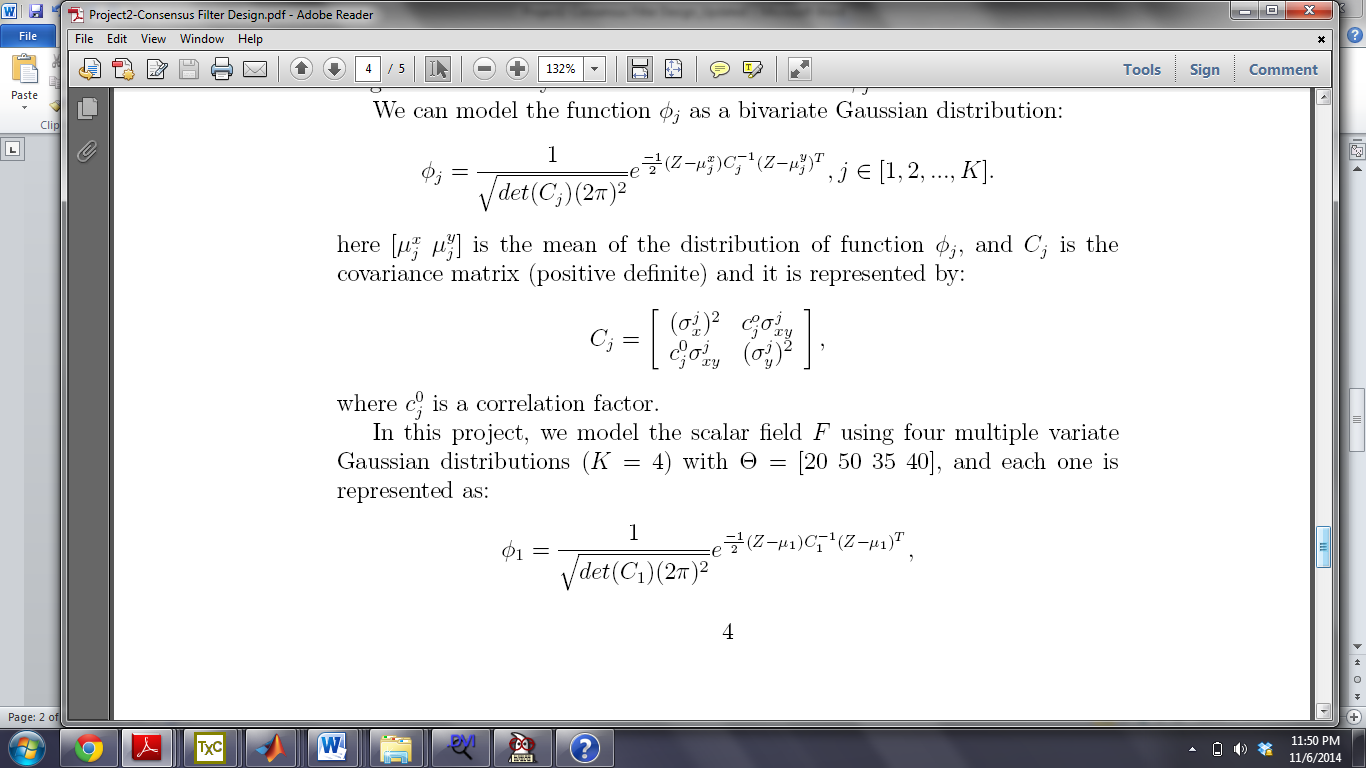
Figure 1: Examples of convergence results.

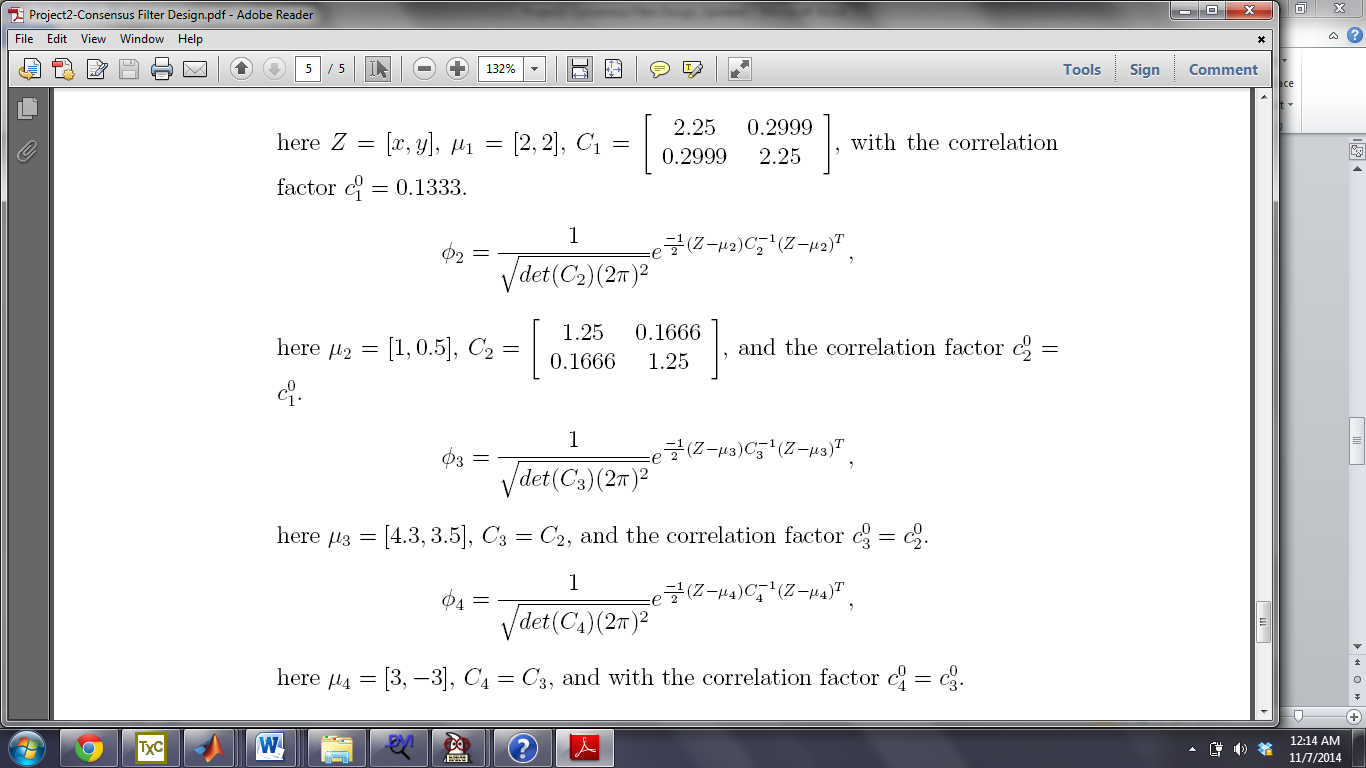
1. Show the convergence of the node which has the smallest number of neighbors and the node which has the largest number of neighbors in the dynamic network case where the node’s neighbors are changing over time. Observe the obtained results and give explanation. (**Grad Students Only**): 10points

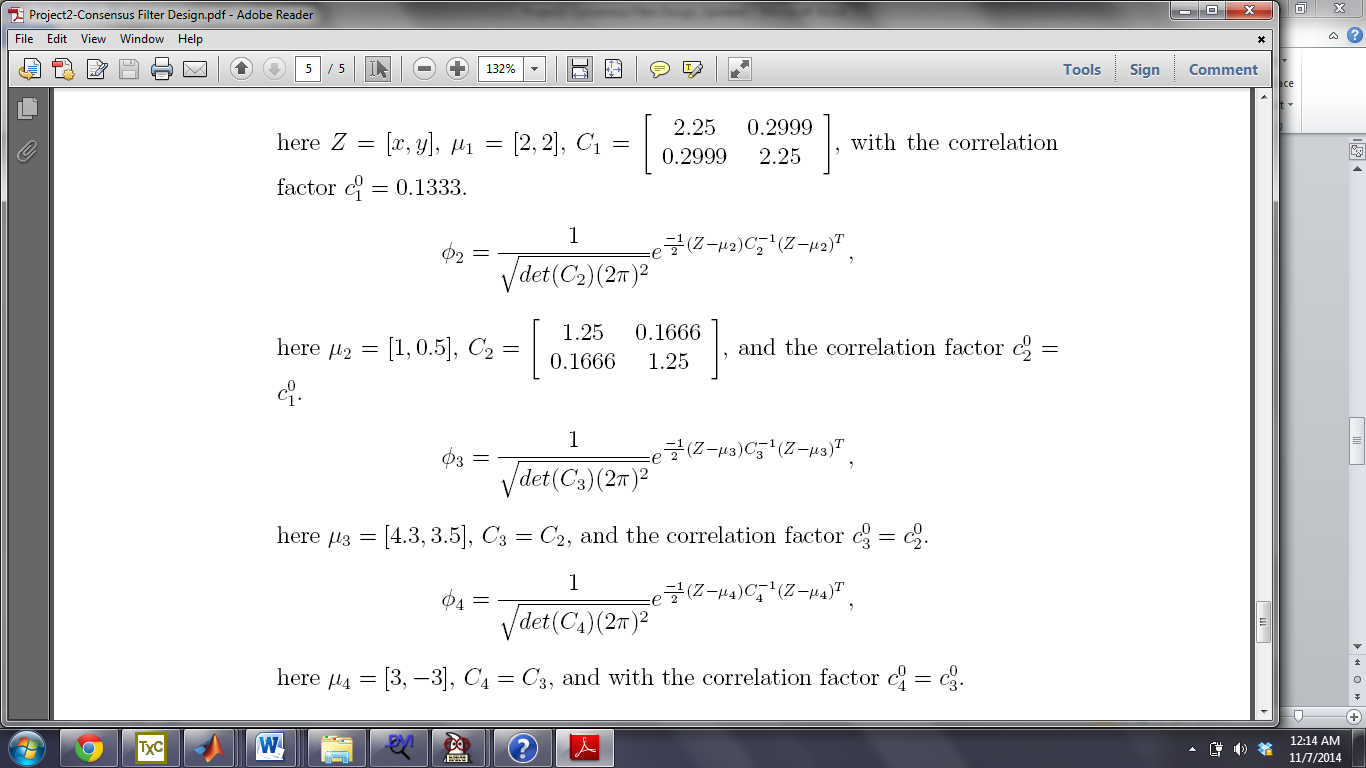
**Some notes to be considered:**

1. **Since we implement this consensus for a static sensor network (*t* is constant) the weight may not get updated in case both node i and its neighbors do not sense the cell. Therefore, we should assume that at least one of node i’s neighbors can sense cell k. (You can try to enlarge the sensing range.)**
2. **When node *i* can not sense the cell, you may try to set up the weight is empty instead of zero to avoid wrong update. If you consider the observability O here, then you do not need to pay attention to this note.**

**Case 2. Estimate multiple cells (scalar field): 50 points**









Scalar field F in 3D



Scalar field F in 2D

The field *F* has a size *x×y* = 25*×*25, and it is partitioned into 25*×*25 = 625 cells. You can set variables *x* and *y* run as: *0 to 25* with scale of *1* as presented in the above scalar field figures.

1. Generate a connected network of 30 nodes to **cover the entire area**. You can select the node’s sensing range (may be rsi = 5). (UG: 10points; G:10points)
2. Running the Consensus 1 (Weighted Average Consensus) to obtain the estimate at each cell of the field F. Then, build the map of this scalar field. (UG: 25points; G:20points)
3. Plot the error between the built map and the original one in both 2D and 3D (ignore if difficult), respectively, and give explanation for the obtained results. (UG: 15points; G:10points)
4. Running the Consensus 2 (Average Consensus) to find out the confidence (weight) of the estimate at each cell, then plot the confidence (weight) in both 2D and 3D. **(Graduate Students Only): 10 points**

**Note: If you feel difficult to model the scalar field F, you can download the .txt data file or Matlab data file of the field F. The data file will contain a valued matrix of *25x25*, and all you need to do is to assign each scalar value with its own *(x, y)* coordinate.**