TU Delft
Faculty of Electrical Engineering, Mathematics, and Computer Science
Circuits and Systems Group

ET 4386 Estimation and Detection

ASSIGNMENT

State Estimation of Formation Control Systems

1 Context

Multi-agent formation control is essential for many applications such as drone swarms and satellite interferometry in space systems. The primary task of such a system is steering the agents from random positions to the target shape and maintaining the stable geometry onward. In a typical distributed formation control system, relative positions among agents are actively measured and controlled locally and are therefore referred to as states from a control perspective. These states are usually acquired from the devices such as GPS, radar, etc. and noises are inevitably introduced. This uncertainty could slow the convergence, jeopardize the optimality, and even destabilize the system. Therefore, accurate estimation of the states with statistical modeling is advantageous in such a setup.

In this project, you are given the recipe to build a simple distributed formation control framework corrupted with noise. You will implement estimators from ET4386 to improve the performance. In a group of 2 students, make a short report (4-5 pages; pdf file) containing the required MATLAB scripts, plots, and answers.

System model

A formation is usually described in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (e.g. Fig. 1) with the vertices $\mathcal{V}$ being the agents and edges $\mathcal{E}$ being communication links. The set of neighbors of node i is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Consider N mobile agents that are deployed in $\mathbb{R}^D$ and their positions are stored in a vector $\mathbf{z} = [\mathbf{z}_1^T, ..., \mathbf{z}_N^T]^T \in \mathbb{R}^{DN}$ in which $\mathbf{z}_i \in \mathbb{R}^D$ is the individual position of agent i. Under single-integrator dynamics $\dot{\mathbf{z}}_i = \mathbf{u}_i$, distributed control law

$$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_i - \mathbf{z}_j), \tag{1}$$

where $\mathbf{u}_i$ is the control input and l_{ij} is edge weights, can be locally used by agents to reach the target locations. In presence of noise, the relative positions $\mathbf{z}_i - \mathbf{z}_j$ (or states) will be corrupted by additive noise $\mathbf{v}_{ij} \in \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{R})$ and the data model will therefore become

$$\mathbf{x}_{ij} = \mathbf{z}_i - \mathbf{z}_j + \mathbf{v}_{ij}. \tag{2}$$

We assume that onboard sensors are able to acquire T independent measurement for each control period meaning that the noises are uncorrelated across the T measurements.

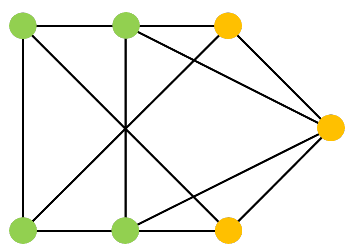

Figure 1: A graph representation of formation.

To assess the performance of the system, a common error function is used

$$\delta = \sum_{i \in \mathcal{V}} \|\mathbf{z}_i - \mathbf{z}_i^*\|_2, \tag{3}$$

where it sums the differences between actual locations and target locations.

2 Assignment

For this project, you will implement and reach the formation in $\mathbb{R}^2$ as shown in Fig. 1 where there are $N = 7$ agents and $M = 12$ edges. Relevant parameters can be found in *data.mat* file including the edge weights, target positions, noises, etc. For practical formation control systems, some agents (e.g. the orange ones in Fig. 1) need to adopt a different control law from (1) such that all agents can be steered to the exact target locations, but for the simplicity of this project we can assume that they stay in the target locations from the beginning.

1. **(2pts)** To begin with, you will implement the formation control system without noise modeling and show the trajectories of convergence. The way to plot a graph can be found in *plot_formation.m*. You may refer to the pseudocode given in Algorithm 1 or in [1]. You will also need to show the convergence plot in terms of the error (3).

2. **(4pts)** Now you will generate the noise based on the given covariance matrix $\mathbf{R}$ and plug in the noisy observations in the control systems. Show that the performance degrades with noisy observations. Next, you will derive and implement no less than 2 estimators to estimate the states from T noisy measurements. Note that you can aggregate T equations into one equation for the derivation using the Kronecker product $\otimes$.

3. **(2pts)** Show that your estimators improve the performance of the control system by plotting the trajectories and the error. Discuss how the number of measurements T impacts your estimation performance. Also, discuss the reasons for the choices of your estimators.

4. **(2pts)** Report writing and research.

Algorithm 1 *Formation control in noiseless case*

Initialization
  Get edge weights $l_{ij} = [\mathbf{L}]_{ij}$
  Get initial positions $\mathbf{z}_i(0)$
  $k = 0$
while $k \leq K$ **do**
  for $i \in \mathcal{V}$ **do**
    $\mathbf{u}_i(k) = -\sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_i(k) - \mathbf{z}_j(k))$ ▷ Control law (1)
    $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$ ▷ Single-integrator update
  end for
  $k = k + 1$
end while

References

[1] M. van der Marel, "Edge state kalman filtering for distributed formation control systems," MS Thesis, 2021, TU Delft.