Distributed Autonomous Systems

Course Project #1

The project consists in two main tasks. The first concerns a data analytics application while the second deals with control for cyber-physical systems.

Task 1: Distributed Classification via Neural Networks

Suppose to have N agents that want to cooperatively determine a classifier for a set of images of hand-written digits. Each agent i is equipped with a set of $m_i \in \mathbb{N}$ images $\mathcal{D}^i \in \mathbb{R}^d$ with associated digit y^i coming from the MNIST dataset, consisting of $70,000$ images of hand-written digits. The dataset is split in a training set $\{\mathcal{D}^j, y^j\}_{j=1}^{60,000}$ and a test set $\{\mathcal{D}^k, y^k\}_{k=1}^{10,000}$. Each sample consists of a pair with a grey-scale image, represented as a matrix in $[0,255]^{28\times 28}$, and an associated label in $\{0,1,\dots,9\}$.
Task 1 is:

1. select a digit (say 4)
   (a) assign to all images representing the selected digit (say 4) the label 1
   (b) assign to all images not representing the selected digit (say $0,1,2,3,5,6,7,8,9$) the label -1
2. split (randomly) the entire training set in N subsets, one for each agent i
3. run the Distributed Gradient Tracking to train the neural network and generate a set of simulations showing the convergence of the algorithm. Use also different dataset sizes (start with a small number of samples).
4. evaluate the quality of the obtained solution by computing its accuracy (say computed by agent 1) on the test set. That is, compute the percentage of success of the following test, for $k = 1,\dots,10,000$

$$\hat{y}^k = \begin{cases} 1 & \text{if } (\mathbf{u}^\star)^\top \mathcal{D}^k \geq 0 \\ -1 & \text{if } (\mathbf{u}^\star)^\top \mathcal{D}^k < 0. \end{cases}$$

   The classifier succeeds if $\hat{y}^k = y^k$.

Hints:

1. reshape and normalize the samples so that $\mathcal{D}^k \in [0,1]^{784}$
2. the dataset can be imported from the Keras Python Library (`from keras.datasets import mnist`)

Task 2: Formation Control

Consider a network of N robotic agents modeled by double-integrator dynamics, whose state includes position $p_i(t) \in \mathbb{R}^d$ and velocity $v_i(t) \in \mathbb{R}^d$

$$x_i(t) = \begin{bmatrix} p_i(t) \\ v_i(t) \end{bmatrix}.$$

The agents have to control the translation and scale of a desired formation while maintaining the desired formation pattern. Robots are partitioned in two groups, namely leaders and followers. The leaders move with constant velocity profiles (possibly zero), while the followers implement the following control law

$$u_i(t) \; = - \sum_{j \in \mathcal{N}_i} P^{\star}_{g_{ij}} \left[k_p(p_i(t) - p_j(t)) + k_v(v_i(t) - v_j(t)) \right]$$

where k_p, k_v are positive constant gains, $P^{\star}_{g_{ij}} := I_d - g^{\star}_{ij} g^{\star\top}_{ij}$ is an orthogonal projection matrix associated to the desired bearing unit vector of agent j relative to agent i. The desired bearing unit vector $g^{\star}_{ij}$ is related to the position of the agents in the desired formation

$$g^{\star}_{ij} := \frac{p^{\star}_j - p^{\star}_i}{\|p^{\star}_j - p^{\star}_i\|}$$

Further details can be found in [1].
Task 2 is as follows.

1. Write a discrete-time version of the continuous-time model described in [1, Section II]
2. Starting from the "Consensus ROS 2 package", create a ROS 2 package written in Python that deals with the Formation Control problem described in [1, Section III.A].
3. Using the developed software, generate a set of simulations in the two-dimensional space, with different number of robots, and show the results.
4. Develop a set of formation patterns so that robots draw, in the two-dimensional space and one at a time, letters of a chosen word.
5. (optional) Set up a simulation that deals with time-varying leaders velocities (see [1, Section III.B]).
6. (optional) Handle simulations with Gazebo[1] or Rviz[2].

Notes

1. Any other information and material necessary for the project development will be given during project "meetings".
2. The project report must be written in Latex and follow the main structure of the attached template.
3. Any email for project support must have the subject:

   "[DAS2022]-Group X: rest of the subject".

References

[1] S. Zhao and D. Zelazo, "Translational and scaling formation maneuver control via a bearing-based approach," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 429–438, 2015.

[1] `https://classic.gazebosim.org/tutorials?tut=ros2_installing&cat=connect_ros`
[2] `https://github.com/ros2/rviz`