Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Decentralized Control for Multi-Agent System formation based on Regular Polygons

Ernesto Franklin Marçal Ferreira *
João Viana da Fonseca Neto ** **Rastko R. Selmic** ***

* *IFMA - Instituto Federal do Maranhão, Brasil (e-mail: ernesto.ferreira@ifma.edu.br).*
** *UFMA - Universidade Federal do Maranhão, Brasil (e-mail: jviana@dee.ufma.br)*
*** *Concordia University - Montreal, Canada(e-mail: rastko.selmic@concordia.ca)*

Abstract: In multi-agent systems, the control is decentralized when decision-making is done by agents individually and not by a centralized unit that processes the states of all agents together. The main mathematical difference when working with centralized versus decentralized control, more precisely when presenting the dynamic system, is the absence of a switching or grouping matrix such as the Laplacian matrix, which presents the characteristics of interconnection between the agents of the system, with the absence of the same the system does not have a complete information on the number of agents and the way of connection between them.

Keywords: Multi-agent System, Cooperative Architectures, Graph Theory and Formation Control.

1. INTRODUCTION

In decentralized control, the agreement protocol must be inserted in the agent's own behavior, and there is no need for an agreement matrix. Internally, the agent must have all the information necessary for its operation, Shamma (2007).

When it comes to formation control, the idea of decentralized control is more difficult to conceive due to the need for a set of information (position and velocity) of agents to reach the control target (form). As an example, it is impossible only with the position of one agent to assemble a triangular formation of three agents.

To perform decentralized control in formation, a methodology to compute the distances between the agent and its neighbors (agents in the vicinity) is the is presented in this paper.

This control methodology doesn't need of positions and velocities of other neighborhood agents, but, only the distance between the controlled agent and the reference agent is necessary to assemble the formation.

Assuming on this work that formation should keep agents equidistant, a method to define the form would be the use of regular polygons as the default shape. The regular polygons where the amount of sides is defined by the number of agents N and θ is the angle between the agents for each formation, Durell and Robson (2003).

In the next section, the decentralized control system for the formation of multi-agents is presented, with all necessary equations for the elaboration of the decentralized control algorithm, as well as the restrictions that guarantee the rigidity of formation. Section 3 presents the computational results, for two training control simulations of three agents. Finally the conclude remarks are presented in Section 4.

2. FORMATION DECENTRALIZED CONTROL SYSTEM

In this approach the control objective is to reduce the error between desired distances and measured/calculated distances. The system of equations that represent the dynamics of the multi-agent system, Anokina Shalimoon (2018), is given by

$$\dot{x}_i = A_i x_i + B_i u_i, \tag{1}$$

where the matrices A_i and B_i are the ith agent's dynamic, x_i are the agent's states and the control output of system u_i is given by

$$u_i = -K_i e_{ij}, \tag{2}$$

where K_i is the ith agent's gain and the control error is computed by the state differences d_{ij} minus the desired differences h_{ij} that is given by

$$e_{ij} = d_{ij} - h_{ij}. \tag{3}$$

The vector of agents state x_i is given by

⋆ Decentralized Control for Multi-Agent System formation based on Regular Polygons

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

$$x_i = \begin{bmatrix} P_{x_i} \\ V_{x_i} \\ P_{y_i} \\ V_{y_i} \end{bmatrix},$$

where P_{x_i} it the position in $x - axis$, V_{x_i} is the velocity in $x - axis$, P_{y_i} is the position in $y - axis$ and V_{y_i} is the velocity in $y - axis$.

Replacing (2) and (3) in (1) we obtained the closed loop Equation given by

$$\dot{x}_i = A_i x_i - B_i K_i (d_{ij} - h_{ij}), \tag{4}$$

where d_{ij} is difference vector is given by

$$d_{ij} = [x_i - x_j] = \begin{bmatrix} \Delta P_{x_{ij}} \\ \Delta V_{x_{ij}} \\ \Delta P_{y_{ij}} \\ \Delta V_{y_{ij}} \end{bmatrix},$$

where $\Delta P_{x_{ij}}$ is the distance on the $x - axis$, $\Delta V_{x_{ij}}$ is the speed difference on the $x - axis$, $\Delta P_{y_{ij}}$ is the distance on the $y - axis$, and $\Delta V_{y_{ij}}$ is the speed difference on the $y - axis$ between agent i and agent j.

The desired formation vector is given by

$$h_{ij} = \begin{bmatrix} h_{P_{x_{ij}}} \\ h_{V_{x_{ij}}} \\ h_{P_{y_{ij}}} \\ h_{V_{y_{ij}}} \end{bmatrix},$$

where $h_{P_{x_{ij}}}$ is the desired distance in $x - axis$, $h_{V_{x_{ij}}}$ is the desired speed difference in $x - axis$, $h_{P_{y_{ij}}}$ is the desired distance in $y - axis$ and $h_{V_{y_{ij}}}$ is the desired speed difference in $y - axis$ between agent i and agent j,

In the formation control one of the objectives is that after the formation is reached this is maintained, for that the agents must have the same velocity in both axes (x, y) that is why in the vector h_{ij} the speed difference is zero.

The definition of the h_{ij} vector values depends on the chosen form, as mentioned earlier, the form is selected by the number of system agents N. For three agents system $N = 3$, the shape is an equilateral triangle as shown in Figure 1.

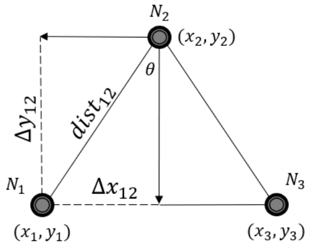

Fig. 1. Three nodes formation with distances between agents.

In Figure 1 the distance $dist_{ij}$ and the angle θ between agents are defined, this absolute distance (Euclidean) can be decompose using simple geometrical definitions into Cartesian $(xy - axis)$ distances, to assemble the desired distance vector h_{ij}.

2.1 Restrictions

To perform the formation control based on the h_{ij} vector using only the absolute distance (Euclidean), the following restrictions must be followed:

- The formation must be a regular polygon to ensure the angle value θ.
- The rigidity of form can only be obtained by decomposing Euclidean distance to Cartesian distance together with the node‘s position orientation (Forward/ Backward, Left/Right).

3. COMPUTATIONAL RESULTS

In this subsection is presented the computational result for the system of Equations (1-4). Its used the parforloop function of MATLAB® to simulate the individual processing of each agent, this function executes a set of statements in the loop body in parallel, the execution of instructions are carried out simultaneously, for each agent an individual loop will be used.

To represent the dynamic system of the agents, a cinematic model of robot with movement in 2-axis (x, y) is chosen. The values of the matrices A_i and B_i are given by,

$$A_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & a_{21} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix}, \tag{5}$$

and

$$B_i = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6}$$

The values for the gain K_i are calculated by pole allocation.

3.1 Triangular Formation

The results for a formation with three agents are presented in this section, the desired form is an equilateral triangle. The computational results of this subsection is obtained using the Decentralized Algorithm 1.

In this simulation the dynamics of all agents are the same so $A_1 = A_2 = A_3$, $B_1 = B_2 = B_3$ and $K_1 = K_2 = K_3$, however the values of the vectors h_{ij} are calculated using the equilateral triangle as desired shape.

Presented as late breaking results contribution
21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Decentralized Algorithm 1 - 3 Agents

Setup (Initial Conditions)
Dynamic System Matrices: A_i, B_i, h_{ij}, K_i.
Number of Agents: N; Simulation time: $t_0 \rightarrow t_f$.
Initial State: $x_{1_0}, x_{2_0}, x_{3_0}$.

Iterative Process
Parfor $t_f \leftarrow t_0$
  do
    Position Update $Agent_1$
    $\dot{x}_1 \leftarrow A_1 x_1 + B_1 K_1 (d_{12} - h_{12})$
    $x_1 \rightarrow [P_{x_1}, V_{x_1}, P_{y_1}, V_{y_1}]^T$
    Position Update $Agent_2$
    $\dot{x}_2 \leftarrow A_2 x_2 + B_2 K_2 (d_{23} - h_{23})$
    $x_2 \rightarrow [P_{x_2}, V_{x_2}, P_{y_2}, V_{y_2}]^T$
    Position Update $Agent_3$
    $\dot{x}_3 \leftarrow A_3 x_3 + B_3 K_3 (d_{31} - h_{31})$
    $x_3 \rightarrow [P_{x_3}, V_{x_3}, P_{y_3}, V_{y_3}]^T$
    Distances Update
    $d_{12} \leftarrow [x_1 - x_2]$
    $d_{23} \leftarrow [x_2 - x_3]$
    $d_{31} \leftarrow [x_3 - x_1]$
end

The iterative process is the execution of the control algorithm to assemble the desired formation, from start time t_0 to final time t_f.

For simulation, the processing parallelism of the dynamics of the agents is performed by the function parfor as previously said, this guarantees that the processing of the states is in parallel.

Figure 2 presents the iterative formation process in which the three agents begins at the initial position (represented by x) and start moving towards the desired equilateral triangle formation, after the agents reach the formation, the simulation is interrupted.

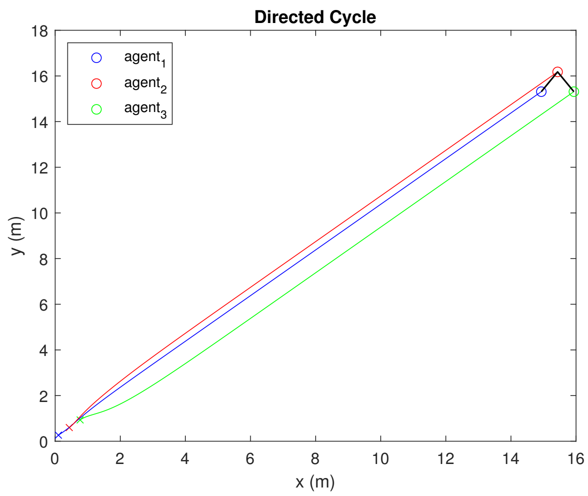

Fig. 2. Iterative process for triangular formation.

In this simulation, the desired Euclidean distance between agents is 1m, Figure 3 shows the position of three agents (agent_1, agent_2 and agent_3) in the last iteration.

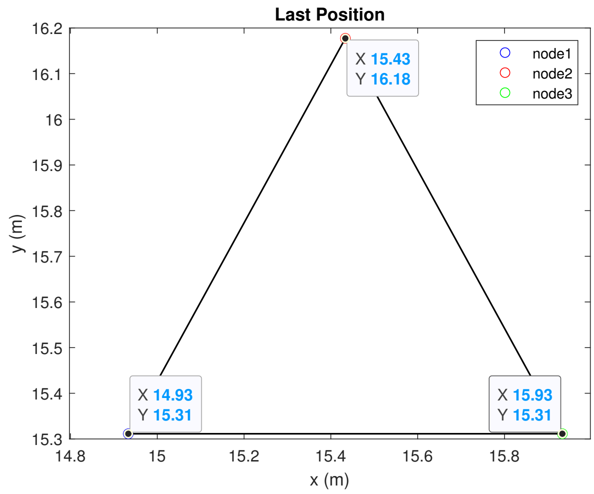

Fig. 3. Final position of agents in the iterative process for triangular formation.

The simulation error is the desired Euclidean distance minus computed distance, the value of error in iterative process for a decentralized formation is shown in Figure 4.

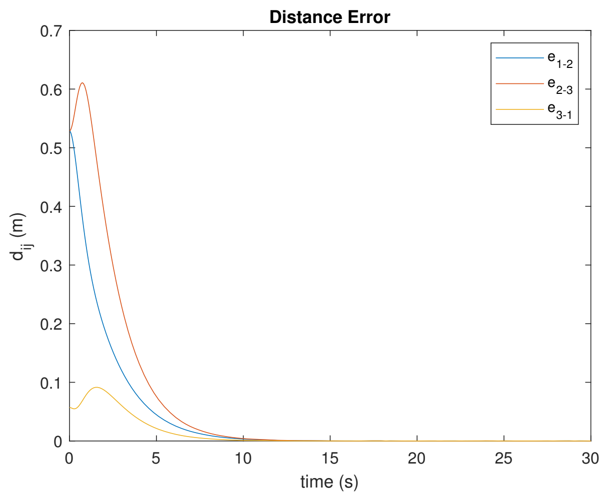

Fig. 4. Iterative process error of the triangular formation.

Figure 4 shows that the errors of the iterative process tend to zero, which can be verified through Figure 3 where, the absolute distance from agent_1 to agent_3 is equal to 1m, that is the desired distance between agents.

3.2 Position Change

In this subsection the computational results are presented to variations in the initial positions and sudden changes of position during the iterative process.

The experiments are repeated for 3 agents, but in this simulation the values of the initial states (P_{x_i}, P_{y_i}) are randomly defined with a value between $0 - 10$ meters. In order to test formation stability (property of agents to achieve and maintain a desired formation), at 10 seconds of simulation, the agent_1 is moved to a new position on the x-axis and y-axis, this position is also randomly set to a value between 0-10 meters again.

The iterative formation process with three agents is presented in Figure 5, the agents start moving from initial position (represented by x points) towards the desired equilateral triangle formation, after 10 seconds of simulation the agent$_1$ is taken to a new position $P_{x_1} = 4.5m$ and $P_{y_1} = 8.5m$, In sequence the agents start moving again to the desired formation, after more 10 seconds the simulation is finished.

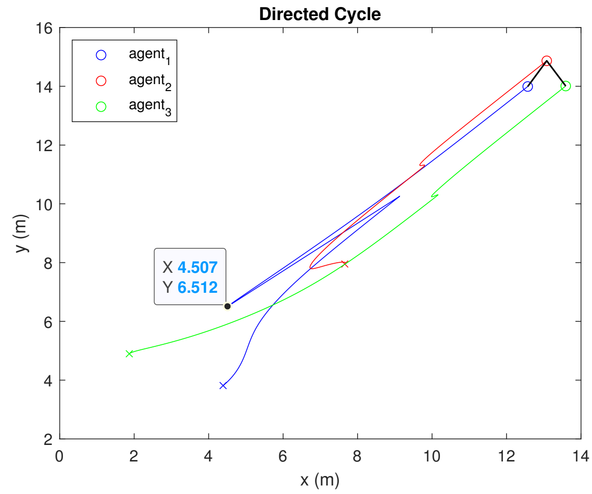

Fig. 5. Iterative process for formation with 3 agents with 'position' change of agent$_1$.

In this simulation, the desired Euclidean distance between agents is 1m, Figure 6 shows the position of three agents (agent$_1$, agent$_2$ and agent$_3$) in the last iteration.

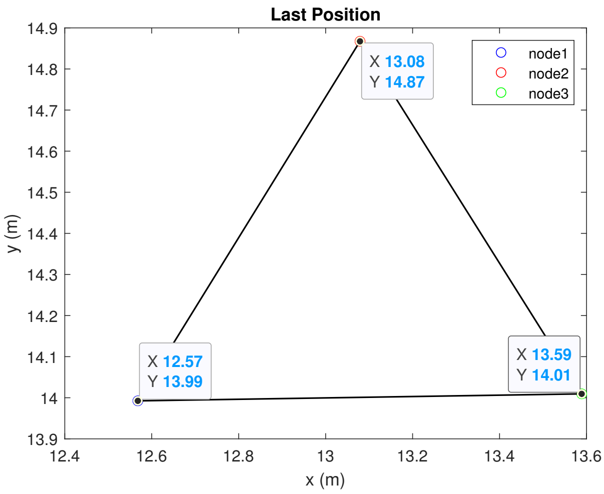

Fig. 6. Final position of agents in the iterative process with 'position' change of agent$_1$.

The value of error in iterative process for a decentralized formation is shown in Figure 7.

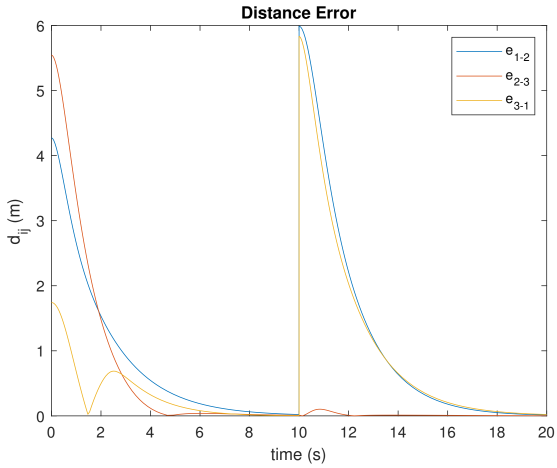

Fig. 7. Iterative process error with 'position' change of agent$_1$.

Figure 7 shows that the errors of the iterative process is close to zero, which can be verified through Figure 6 where the distance from agents agent$_1$ to agent$_3$ is 1.0132m, this error of 1.3% in relation of desired distance is explained because in this simulation have two process of formation each one with 10 seconds.

The simulation presented in this subsection proved that there was no loss of formation stability due to sudden variations in agent position, and it was also observed the cooperativism among the agents during the assembly of the formation.

4. CONCLUSION

Based on regular polygons, the latest results of a method for decentralized multi-agent formation control was presented in this paper. Motivated in real time drone formation applications, the control results were satisfactory, because the error between calculated distance and desired distance is close to zero and the and a good stability in recovering the desired formation even the agent being taken a new position suddenly had shown the ability of proposed algorithm (methodology).

ACKNOWLEDGEMENTS

The authors would like to thank to the Federal University of Maranhão (UFMA), the Concordia University, the Federal Institute of Maranhão (IFMA), the Coordination of Undergraduate Personnel Improvement (CAPES), FAPEMA and the National Council for Scientific and Technological Development (CNPq) for the development infrastructure and financial support to carry out this work.

REFERENCES

Anokina Shalimoon (2018). *Cooperative Control of Multi-Agent Systems: Consensus, Flocking and Formation Control.* Master's thesis, CALIFORNIA STATE UNIVERSITY, NORTHRIDGE.

Durell, C.V. and Robson, A. (2003). *Advanced Trigonometry.* Dover Publications.

Shamma, J. (2007). *Cooperative Control of Distributed Multi-Agent Systems.* Wiley-Blackwell.