Finite-time consensus for unicycle robots

Anna Fumagalli, Giuliano Giampietro, Vincenzo Maria Vitale

June 2024

1 Introduction

This project aims to implement a finite-time consensus protocol for non-holonomic ground robots. The latter should achieve the consensus and form a predefined regular shape within a finite time (*rendezvous* and alignment[1]). At convergence time, the agents should be at the desired pose (position and orientation) with zero velocity, avoiding the singularities. The simulation consists of a certain number of robots required to meet at an unknown rendezvous point while forming the shape of a regular polygon, as in the case of the paper by Lorìa *et al.*[2]. However, there are some differences with the paper mentioned: firstly, unlike the paper, which envisages exactly six unicycles, our simulation supports a variable number of them, and consequently, so does the regular polygon they generate: as many agents as there are sides from which the polygon is composed. Secondly, they are required to come into formation in a finite time. In order for them to be able to place themselves in formation in finite time, we respected what was obtained in the paper by Wang *et al.*[3].

Notations: Let N be the number of agents, $\mathcal{I}_n = 1, 2, ..., N$, $\mathcal{N}_i$ the index set of neighbors of the agent i, $\mathcal{E}$ the edge set.

2 Agents Modelling

The agents, in this case unicycles, are modelled as points. They are identified in a system that depends on their Cartesian coordinates in the plane (x, y), and a θ angle which describes their orientation. Considering in their displacement a linear velocity v and a linear ω, the kinematic system results:

$$\begin{cases} \dot{x}_i = cos(\theta_i)v_i \\ \dot{y}_i = sin(theta_i)v_i \\ \dot{\theta}_i = \omega_i \end{cases}, \tag{1}$$

[1]With the term *rendezvous* we refer to achieve the uniformity in position with respect to a common point, with alignment the uniformity of the orientation angles.

with $i \in \mathcal{I}_n$, since they are subject to the non-holonomic pure rolling constraint $\dot{x}sin(\theta) - \dot{y}cos(\theta) = 0$. Instead, their dynamics are of the form:

$$\begin{cases} \dot{v} = u_v \\ \dot{\omega} = u_\omega \end{cases}, \tag{2}$$

which represent also the control inputs.It follows that linear motion depends on linear velocity and acceleration and can be described more compactly by the system:

$$\begin{cases} \dot{z} = \phi(\theta)v \\ \dot{v} = u_v \end{cases}, \tag{3}$$

where $z = \begin{bmatrix} x \\ y \end{bmatrix}$ and $\phi = \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix}$, while the angular motion:

$$\begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = u_\omega \end{cases}. \tag{4}$$

Agents must start from a random position[2] with zero initial velocities and arrive in the formation of a regular polygon around the consensus point, described as:

$$\delta_i = \begin{bmatrix} \delta_{x_i} \\ \delta_{y_i} \end{bmatrix} = \begin{bmatrix} \mathcal{A}cos(i2\pi/N) \\ \mathcal{A}sin(i2\pi/N) \end{bmatrix} \tag{5}$$

The system in (5) describes the position of the acting $i-th$ concerning the consensus point and defines the distance that the robot should reach relative to the centre in any regular polygon, with $\mathcal{A}$ representing its apothem.
The agents interaction is represented by a two-way, time-invariant, unweighted graph for reasons dictated by the finite time constraint analysed in the later section 3.2.

3 Consensus Protocol

Let us now go into the details of the consensus protocol. In this section, we will evaluate its application without finite time limits.

Let us consider the N agents, their respective z and θ states, and the internal dynamics, modelled in this case as a double integrator. Let us consider also the interaction graph $\mathcal{G}(\mathbf{A})$, where $\mathbf{A} \in R^{n \times n}$ is the adjacency matrix, representing the connections and having the agents at the vertices. The vertex v_i represents the agent i, the existence of the edge $(v_i, v_j) \in \mathcal{E}(\mathcal{G}(\mathbf{A}))$ implies an available information channel from i to j, and the weight a_{ij} represents the reliability of the corresponding channel. The consensus is reached when the position and

[2]In the case implemented, the random positions are in a symbolic range between zero and thirty for both the x- and y-coordinates

orientation (pose) $p_i \to p_c$ and the velocities (both linear and angular) $\dot{p}_i \to 0$ with $i \in \mathcal{I}_n$ for a non-a-priori-given set point. In other words[3],

$$\begin{cases} \lim\limits_{t\to\infty}[p_i(t) - p_j(t)] = 0 \\ \lim\limits_{t\to\infty}[\dot{\theta}_i(t)] = 0 \end{cases} \qquad \forall i, j \in I_n \tag{6}$$

The problem is to design a proportional-derivative consensus problem with control inputs of the form:

$$\begin{cases} u_{\omega_i} = -k_{\omega_{d_i}}\dot{\theta}_i - k_{\omega_{p_i}} e_{\theta_i} \\ u_{v_i} = -k_{v_{d_i}} v_i - k_{v_{p_i}} \phi_i(\theta_i)^T e_{z_i} \end{cases} \tag{7}$$

where $k_{\omega_{d_i}}$, $k_{\omega_{p_i}}$,$k_{v_{d_i}}$, and $k_{v_{p_i}} > 0$ are the derivative and proportional gains respectively for the angular and linear accelerations, while $e_{\theta_i} = \sum_{j\in\mathcal{N}_i} a_{ij}(\theta_i - \theta_j)$, and $e_{z_i} = \sum_{j\in\mathcal{N}_i} a_{ij}(\overline{z}_i - \overline{z_j})$ In turn, $\overline{z}_i = z_i - \delta_i$, and represents the Cartesian position of the $i - th$ robot in the shape.

3.1 Persistency of excitation

Since the angular position might get stuck in unwanted equilibria, we endowed the angular motion controller with a persistently exciting term that excites *all modes in the system to overcome the stabilisation obstacles imposed by the non-holonomic constraints* [2]. In other words, to prevent the robot from orienting itself at a desired angle but from arriving at the desired position due to the non-holonomic constraint, a perturbative term α is considered so that it does not get stuck in local minima.

That term α has to be added to the angular acceleration $\dot{\omega}$, which also represents a PD-like consensus controller input, i.e.

$$\dot{\omega}_i = u_{\omega_i} = k_{\omega_{d_i}}\dot{\theta}_i - k_{\omega_{p_i}} e_{\theta_i} + \boldsymbol{\alpha}_i, \tag{8}$$

namely the same as in the first equation of the system in (7) with the addition of the perturbing term. We then define the form of the term α_i as[4]:

$$\alpha_i = \psi(t)\phi_i(\theta_i)^{\perp T} e_{z_i}, \tag{9}$$

where $\phi_i(\theta_i)^{\perp T}\phi_i(\theta_i) = 0$, and as $\psi(t)$ we employed the periodic and thus persistently exciting function taken from [2] in which: $\psi_i(t) := 2.5 + (4/\pi)sin(0.5t)$ for all agents. The resulting closed-loop of the system dynamics is in Fig. 1.

A brief explanation of what is in the figure: at the consensus protocol, added the perturbating term - which as mentioned depends on the cartesian

[3]The formulas in (6) are only valid when no constraints on temporal finiteness are considered.

[4]The dependence on the linear position error in the equation (9) is justified because when the robots are in the target Cartesian position, the linear position error is zero. Therefore, the oscillation is no longer necessary and must stop.

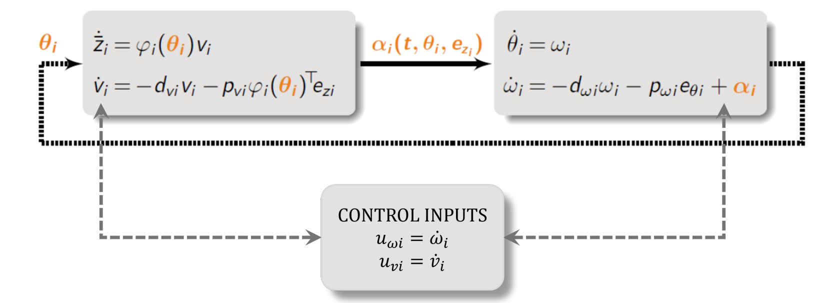


Figure 1: closed-loop of the system dynamics and control inputs. Picture modified from [1]

positions-, we find the angular accelerations. By double-integrating them, we find the angular positions θ, which, once updated, we use to calculate the angular acceleration input, which in turn will be double-integrated to identify the linear positions.

3.2 Finite Time

Agent convergence must occur within a finite time in many circumstances, especially when particular control accuracy is required. Finite-time convergence protocols speed up convergence and provide more significant noise reduction and robustness to uncertainties. In contrast to not placing finiteness constraints on time - as described in eq.6 - , a protocol solves finite time consensus if it solves a consensus problem with any initial state within a time t^* such that[5]:

$$x_i(t) = x_i^* \quad \forall t \geq t^* \text{and} \quad i \in \mathcal{I}_n, \tag{10}$$

where x_i^* is the desired state. The class of protocols that the paper [3] investigated to be suitable for solving finite-time consensus problems is of the type:

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} sign(x_j - x_i)|x_j - x_i|^{\beta_{ij}}, \tag{11}$$

where $0 < \beta_{ij} < 1$ and reflect in this protocol, as a_{ij} do, the reliability of the information, $|\cdot|$ is the absolute value, $sign|\cdot|$ is defined as:

$$sign(r) = \begin{cases} 1, & r > 0 \\ 0, & r = 0 \\ -1, & r < 0 \end{cases} \tag{12}$$

Precisely, the research mentioned states a *theorem.* Assuming the interaction graph $\mathcal{G}(\mathbf{A}(t))$ is undirected, the sum of the intervals in which it is connected

[5] In this case, we refer to x as the overall state of the agent and not just the Cartesian position.

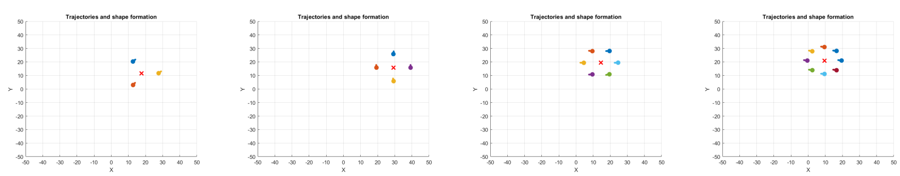
is sufficiently large, $\beta_{ij}(t) = \beta_{ji}(t)$ and $0 < \beta_{ij}(t) < 1 \ \forall i, j, t$, then the protocol solves the *finite time average consensus problem.* A protocol is said to solve the *average-consensus problem* when the final states are a function of the form:

$$\chi(\mathbf{x}(0)) = \sum_{i=1}^{n} x_i(0)/n \tag{13}$$

The reasons for the interaction graph structure chosen are in the theorem mentioned above. The graph had to be undirected. Moreover, being strongly connected, it maximises the algebraic connectivity [6], increasing the convergence rate. But since the complete connectivity between all agents is not entirely realistic - and would not even perhaps justify the use of a distributed control for study and simulation purposes - we did evaluate the graph both fully connected, both partially, to investigate the different results.
In the case of this project, as already shown in Sec. 3, the proportional derivative control inputs are angular and linear accelerations. Therefore, we applied precisely this class of protocols guaranteeing finite-time consensus to the control inputs u_{ω_i}, and u_{v_i}.

4 Results, plots and animations

As mentioned in the introduction, the simulation admits a variable number of agents for training, examples of which are in Fig. 2. The apothem of the regular polygon concerning to which the agents will stand at the vertices has been set to $\mathcal{A} = 25$.



(a) Three agents (b) Four Agents (c) Six Agents (d) Eight Agents

Figure 2: Variable number of agents forming the shape of the corresponding regular polygon

The proposed controller contains both proportional correction terms for errors in the consensus both the persistent excitation term, and this has one main drawback: the difficulty in tuning control gains to avoid excessive oscillatory behaviour [2]. Another parameter that strongly modifies the simulation results, especially regarding the timing of velocities and positions' convergence is β_{ij}. If we set that parameter to 1, for each $i, j \in \mathcal{I}_n$, the protocol becomes the typical linear one, which solves the asymptotic consensus under the hypothesis that the

[6] The algebraic connectivity corresponds to the second smallest eigenvalue of the graph's Laplacian matrix

interaction graph contains a spanning tree. If we set $\beta_{ij} = 0$, it would become discontinuos [3]. In addition, we noticed that the differences between applying a protocol that solved the consensus asymptotically or in finite time were much more appreciable when considering the non-fully connected graph.

We thus now investigate how possible variations of the β parameter can modify the simulation results, comparing what happens with protocols that solve the consensus asymptotically (thus setting β to one) or what changes occur when reducing it. These simulations were carried out with different topologies of graphs, more or less disconnected. In order that the differences generated by different topologies and with different β could be appreciated and compared, we decided to keep all gains at one, finding that this value, while probably not optimal, always led to reasonable simulations.

4.1 Impact of Graph Topologies and β on Consensus Time

We compare here the impact of various graph topologies and different values of parameter β on the consensus time. To assess how graph topology affects consensus, we conducted simulations with different topologies, restricting them to subsets of connected graphs. Within our framework, an arbitrary number of connections can be dropped while maintaining the symmetric property of the graph. In the following images, plot for the position and velocities are shown. As previously mentioned, unitary gains were used to develop these simulations, and a constant α gain $k_\alpha = 0.5$.

Table 1 shows the average time to reach consensus for six agents connected by graphs where k pairs of connections were removed, as the β parameter varies in the protocol.

Table 1: Average Consensus Time (s) for Various Graph Topologies

k \ β	0.3	0.5	0.9
0	24	18	18
3	26	25	20
7	27	24	23
10	30^+	28	29

The Fig.3 shows a typical formation of the six agents: to avoid redundancies, we only show one as an example, well underlining that for each simulation, the consensus point around which the robots are placed in formation and their orients vary.

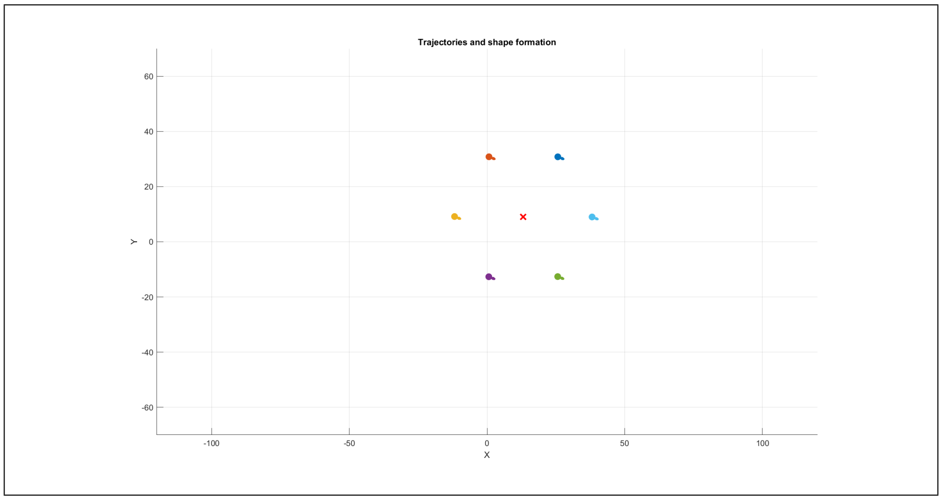
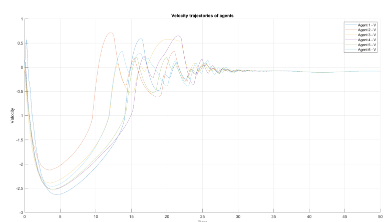
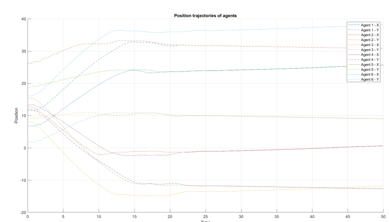


Figure 3: Standard formation around the consensus point, with six agents

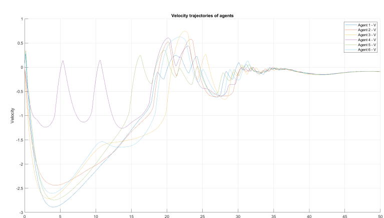
The two velocity profiles in Fig. 4 further illustrate the effects of graph connectivity on consensus time. The figures 4(a-b) depicts the velocity and position profiles for a fully connected graph, i.e. thirty connections, while figures 4(c-d) show a plots with only sixteen connections remaining, which barely maintains the connectivity. These simulations are all obtained setting $\beta = 0.3$.



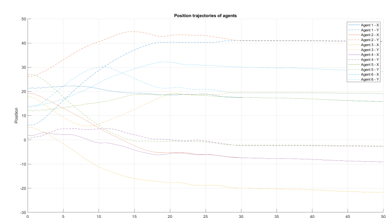
(a) Velocities with 30 connections



(b) Positions with 30 connections



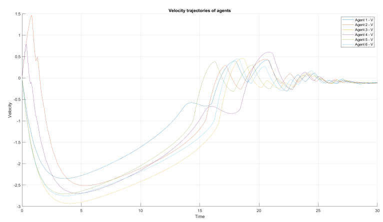
(c) Velocities with 16 connections



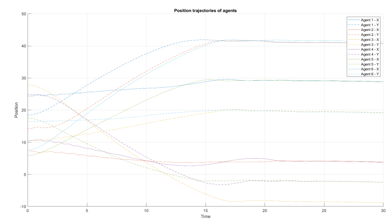
(d) Positions with 16 connections

Figure 4: In (a) and (b) the velocities and positions of six agents with an interaction graph of 30 connections, in (c) and (d) with only 16 connections.

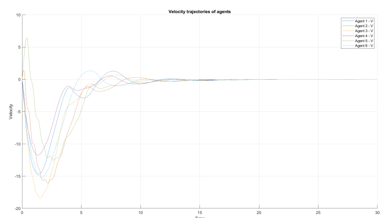
On the other hand, in Fig. 5 the velocity profiles and the positions of agents are compared as β varies. In figures 5(a-b) are the velocities and positions plots with $\beta = 0.3$, while in figures 5(c-d) the velocities and positions with $\beta = 0.9$, all in a fully connected graph.



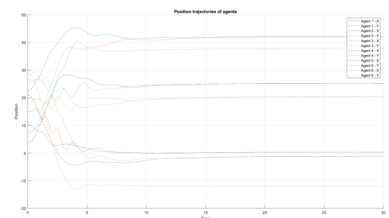
(a) $\beta = 0.3$ Resulting Velocities



(b) $\beta = 0.3$ Resulting Positions



(c) $\beta = 0.9$ Resulting Velocities



(d) $\beta = 0.9$ Resulting Positions

Figure 5: Comparison between $\beta = 0.3$ (a-b) and $\beta = 0.9$ (c-d) in a fully connected graph.

5 Conclusion

Generally speaking, reducing a large number of connections has a significant impact on the time required to reach consensus. This result is intuitive, as fewer connections imply less communication between agents. As would have been expected, as the number of connections approaches the critical number for which the graph is no longer connected, agents exhibit unstable behavior. The experiments were done using empirically found gain parameters; we do not rule out the possibility that there may be more balanced values that help achieve consensus in even less time and with greater stability. As for β, the results show that its impact is also relevant on the consensus time, acting analogously to a gain factor by directly influencing the input. From our results, we find that as we approach the limiting threshold of $\beta = 1$, at the same graph connection level, the time required to reach consensus decreases, stabilizing the velocities of the agents. In general, the work and results produced, appear to be in line with the results found by the reference papers, subject to gain-dependent constants and other minor constant dependant from computation.

References

[1] Lorenzo Govoni and Andrea Cristofaro. *Distributed Consensus Control of second-order systems with non-holonomic constraints*. University Lecture. 2022/2023.

[2] Antonio Loría, Emmanuel Nuño, and Elena Panteley. "Observerless Output-Feedback Consensus-Based Formation Control of Second-Order Nonholonomic Systems". In: *IEEE Transactions on Automatic Control* 67.12 (2022), pp. 6934–6939. DOI: 10.1109/TAC.2021.3136140.

[3] Long Wang and Feng Xiao. "Finite-Time Consensus Problems for Networks of Dynamic Agents". In: *IEEE Transactions on Automatic Control* 55.4 (2010), pp. 950–955. DOI: 10.1109/TAC.2010.2041610.