Freeze and Unfreeze - Enhanced Connectivity maintenance

Karthik Soma (40240691)
Polytechnique Montreal

Rahul Thepperumal Venkatesh(40156065)
Concordia University

***Abstract*—Most of the current rendezvous problem setting in multi-robotic system assume static interaction graph and the ability to communicate with their neighbours at any distance. Whereas, in real life, robots can talk to each other only when they are under a disc of radius of 'R', therefore breaking both the assumption. It has been shown in [1] that even in a rendezvous setting agents can loose connectivity and don't reach consensus to a single common point. To solve this problem, based on the work of [2] and [3] we show that freezing certain agents based on a local policy, much like repeaters in networking terminology will allow the followers to converge into the hull formed by the frozen agents. The frozen agents will then unfreeze if all their neighbours are in the same sector. We also compare our method with the method proposed in the works of [1] and [4].**

***Index Terms*—Freeze, Unfreeze, Connectivity Maintenance, Consensus**

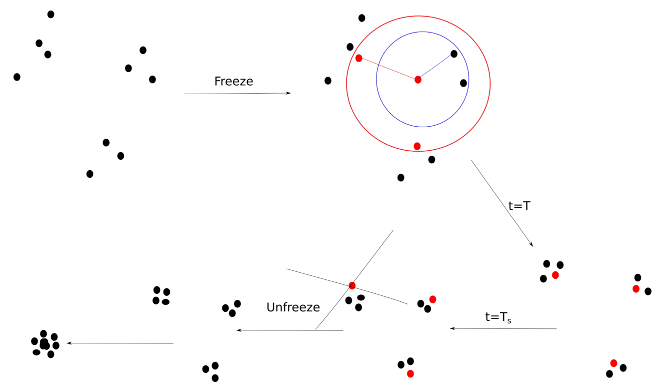

Fig. 1. The proposed method explained for a random graph. The freeze and unfreeze methods.

Introduction

Consensus is an important phenomenon in networked systems. Some applications using consensus are distributed estimation, formation control. For consensus to happen, the networked system should have an underlying topology which is connected or in other words, for the graph $G = (V, E)$ denoted by the vertices which are the agents and the Edges the communication links, should have an Laplacian matrix defined as $D - A$ of rank $N - 1$. N being the number of agents, D being the degree matrix and A being the adjacency matrix. In theoretical and simulation networks, it is always assumed that the underlying graph is connected. This is a big assumption coming into the real world applications as robots have limited connectivity(disc of radius 'R') and therefore it is very possible that agents loose connectivity even while doing simple consensus[1]. Taking this particular topology shown in figure 7 and rules defined in the paper we show that agents following a local policy can still converge. In this report we will explore this method and compare it to two methods from the literature. We will also attempt to prove the convergence of our method in the following sections.

Related work

In the work of [1] they solve this problem by modifying the consensus equation in a way that the system reduces the edge tension energy from its initial topology. This is a big change as most systems dynamics will change because of changing consensus equation. This is also introduces points of singularity where the system blows up. In the work of [5], the followers are herded from one place to another place by making a set of leaders follow a stop-go policy. The leaders are pre selected and remain as such. In the work of [4], they use control barrier functions to maintain connectivity of the system. This is a centralised solution requiring complete knowledge of the system. Also this method needs the knowledge of λ_2 and the vectors associated with it. Calculating this in a real world system can prove very challenging as it is a global variable.

Problem Statement

Given an initially connected system of N agents, following single integrator dynamics $\dot{x} = u$. Where u is control input given to the system. Can we select the critical agents and the edges associated with it so that the followers can get into the hull spanned by the frozen agents?. Can we unfreeze this agents at some time so that the system is not frozen for ever?. Is this the system also connected all throughout?

Methodology

The methodology is rather simple, In the starting the agents have a local policy where they classify their agents into far neighbours and near neighbours. Where the R_n is the near neighbour cutoff, with $R_n < R$. a being the near agents and b b being the far agents adjacency matrices.

$$a_{ij} = \begin{cases} 1 & ||x_{ij}|| <= R_n \\ 0 & ||x_{ij}|| > R \end{cases} \tag{1}$$

$$b_{ij} = \begin{cases} 1 & R_n < ||x_{ij}|| <= R \\ 0 & ||x_{ij}|| > R \\ 0 & ||x_{ij}|| <= R_n \end{cases} \tag{2}$$

The algorithm for freezing and unfreezing is given by the pseudo algorithm adopted from [3]. It is also shown in the pseudo algorithm 2. It should be noted that the freeze algorithm is done only once. So the critical agents identified in the starting will not move meaning that the agents will have $\dot{x_f} = 0$. All the other agents will do a consensus with all their neighbours(including the frozen agents) i.e,

$$\dot{x_{uf}} = \sum_{j=N_f(i)} (x_j - x_i) \tag{3}$$

Algorithm 1: Freeze

```
/* N_f, N_n Far & Near neighbours                */
Function Freeze(N_f,N_n):
    /* ∀i ∈ N                                     */
    state=critical
    if N_f is ∅ then
        state=Notcritical
    if ∀j ∈ N_f^i ∃K ∈ N_n^i s.t. j ∈ N_n^k then
        state=Notcritical
    return
```

Algorithm 2: UnFreeze

```
/* N_i neighbours of i                           */
Function UnFreeze(N_i):
    /* ∀i state=critical ∈ N                      */
    Split the j ∈ N_i in to n sectors
    if j ∈ N_i in one sector then
        state=Notcritical
    return
```

PROOFS

We will state three lemmas which will help to prove the fact that the agents will converge to a point following our freeze and unfreeze algorithm.

Lemma 1: The freeze algorithm will freeze the weak links of the system thereby making sure that the system will never become unconnected.

This is a rather straight forward proof as we know the initial graph is completely connected. Freezing the weak links, will not break the system as the system will be connected throughout.

Lemma 2: The unfrozen agents will converge into the convex hull spanned by the frozen agents.

For the proof of this kindly refer to the paper [2]. The idea is simple, when we freeze a graph the frozen agents become nodes with rooted-out branching trees. All the neighbours of those agents will then start converging to the location of the the frozen agents. Once they have converged to the frozen agent, they will see other frozen agents if there are any. This way they will bought into a convex hull spanned by the agents.

Lemma 3: The unfrozen neighbour agents once inside the convex hull spanned by the frozen agents will be in one sector, therefore not needing to be frozen anymore.

When all the neighbours of a agent are in one sector there is no way a link can break locally w.r.t to the agents near the frozen agents, This is because the agent will move towards the centroid of all the agents thereby staying in the sector. Therefore the agent can be unfrozen allowing the system to reach consensus.

EXPERIMENTS

The following experiments were carried out as a part of this report. We have taken the particular topology example explained in the papers [4] and [1]. The particular topology used for our experiments is shown in the figure 7. The figure is taken from the paper [4]. If used with the normal consensus rule with a cutoff distance implemented the graph will get disconnected as shown in figure 8. So first to analyse what happens in this, this example was implemented. The λ_2 and the velocity of a critical agent was plotted. The same experiments were then carried out with the methods implemented in the papers[1] and [4]. The table summarizes all the different methods used. To emphasis that our method works for both the linear and non-linear weighted adjacency matrix is shown by the idea of having A and B as the non-linear and linear weights associated with the equations below.

$$a_{ij} = \begin{Bmatrix} e^{\frac{(R^2 - x_{ij}^2)}{\sigma^2}} - 1 & ||x_{ij}|| <= R \\ 0 & ||x_{ij}|| > R \end{Bmatrix} \tag{4}$$

$$b_{ij} = \begin{Bmatrix} 1 & ||x_{ij}|| <= R \\ 0 & ||x_{ij}|| > R \end{Bmatrix} \tag{5}$$

To compare our method with the linear method, we used the [1] work. The consensus equations implemented by the group are

$$\dot{x} = \sum_{j=N_f(i)} \frac{(x_j - x_i)}{(||(x_j - x_i)|| - R)^2(||(x_j - x_i)||)} \tag{6}$$

As is given in the equation, there is a singularity point arising in at the region when $||(x_j - x_i)|| - R$. Therefore new edges have to added after a delay to avoid the agents shooting out of range. Similarly to compare with the non-linear weighted matrices, we re implement [4]. Where the implemented equations are using control barrier functions. Formally Control barrier functions for rendezvous example can be written down as

$$u(x) =_{u \in R^m} \frac{1}{2}||u - u_{des}||^2$$
$$\ni \dot{h(x)}(f(x) + g(x)) \geq -\alpha(h(x))$$

where, $\dot{x} = Bu$, thus making $f(x) = 0$ this enables us to use $u_{des} = -Lx$. This system gives maximum freedom for the system to do it's work but takes back the control when it is about to loose connectivity. The $h(x) = \lambda_2 - \epsilon$.

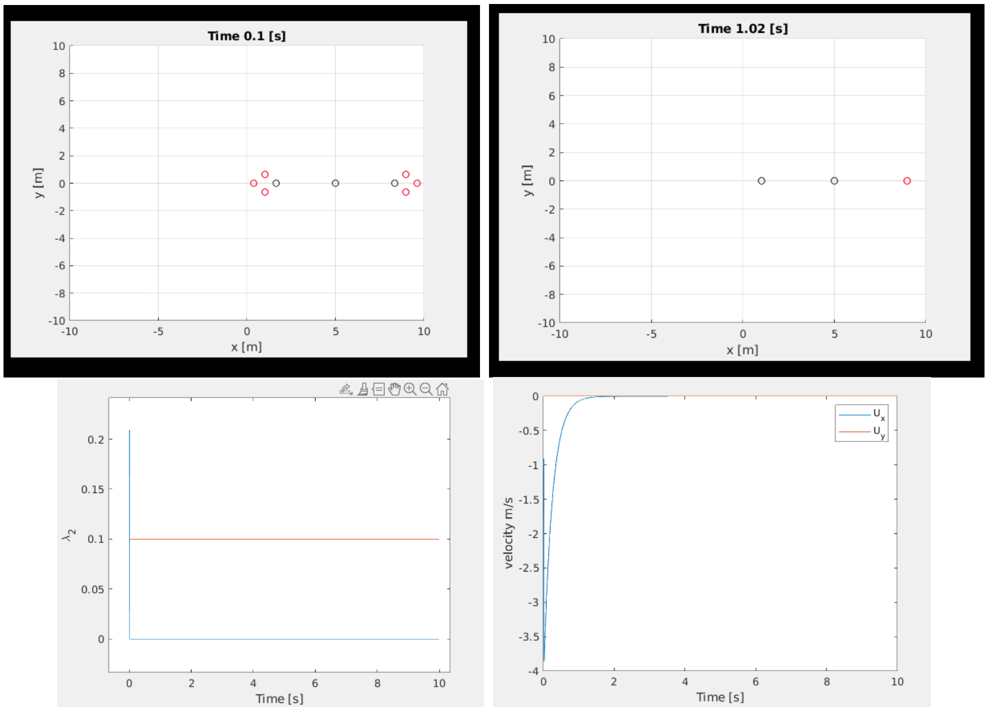

Fig. 2. Evolution of normal consensus. With cutoff distance as 3.01m and linear weighted adjacency matrix. In the top is the screenshots of the system in the start and as shown by the λ_2 and velocity of the agent(right of the middle agent) graphs in the bottom the system breaks soon and never recovers from it.

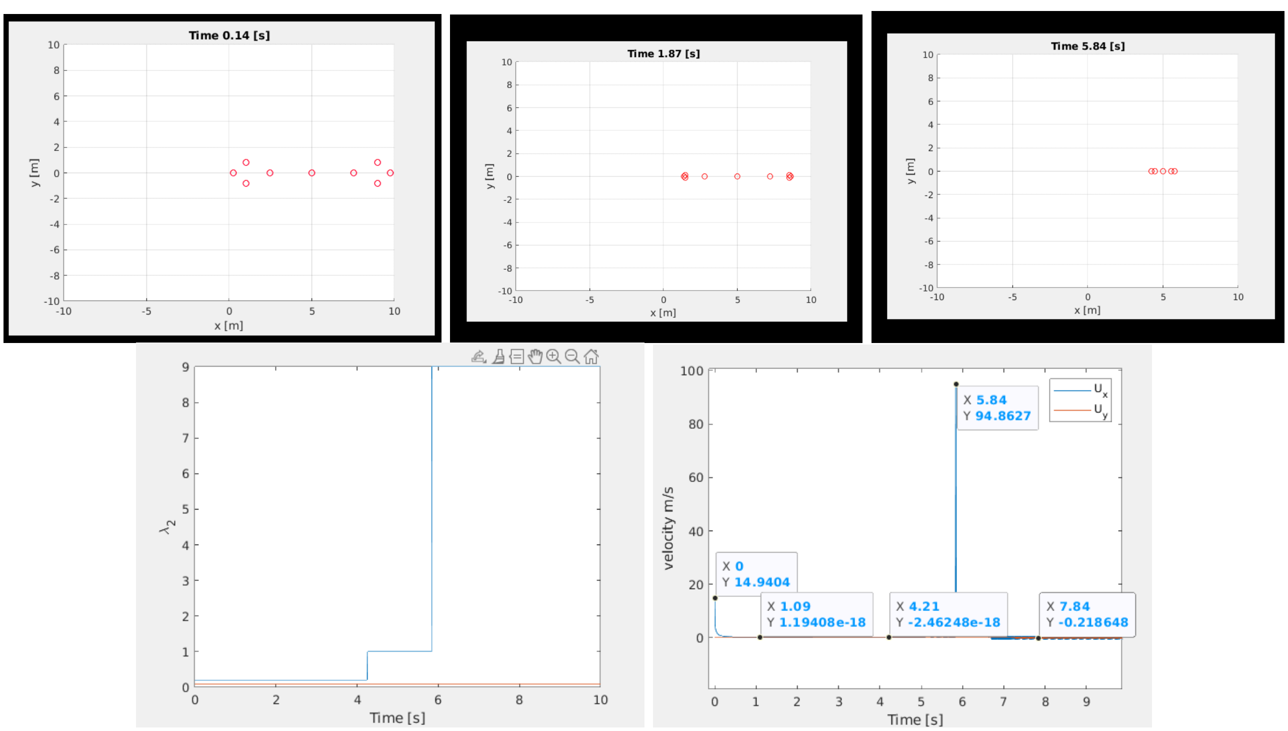

Fig. 3. Evolution of connectedness[1]. The top row shows the evolution of the system at three different time steps. In the bottom the λ_2 and the velocity of the critical agent is shown. It can be seen that the system introduces problems when new edges are just encountered.

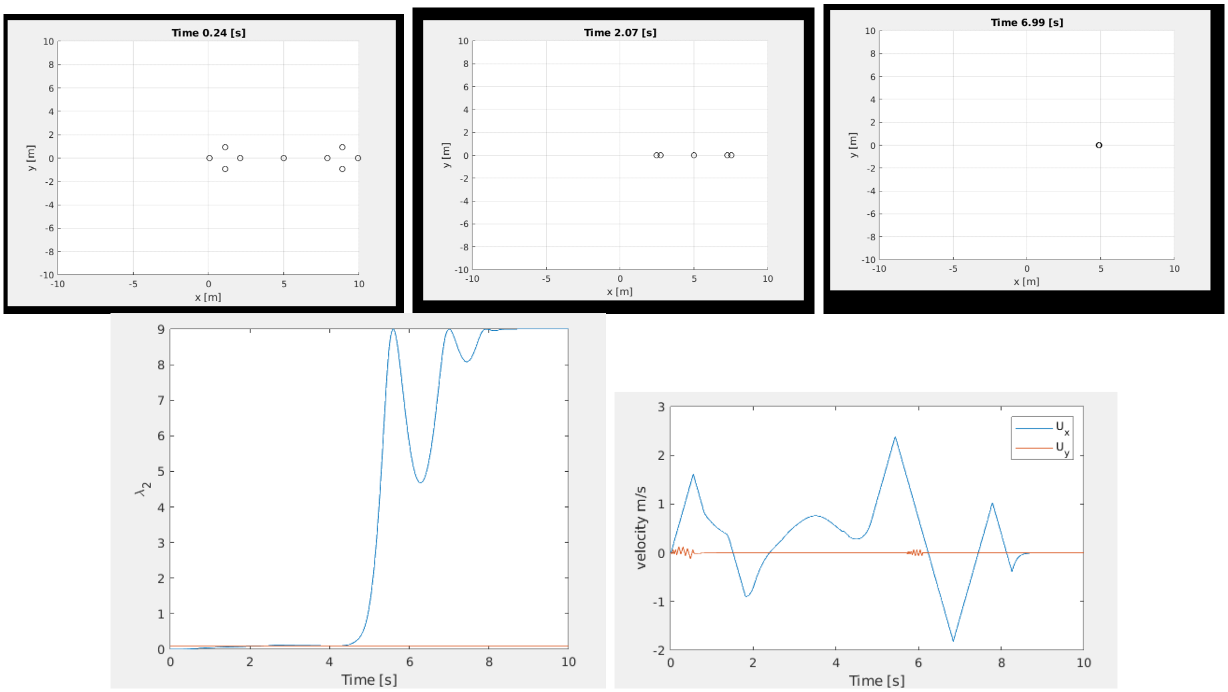

Fig. 4. Evolution of connectivitycbf[4]. The top row shows the evolution of the system at three different time steps. In the bottom the λ_2 and the velocity of the critical agent is shown. It can be seen that the system has a smooth convergence

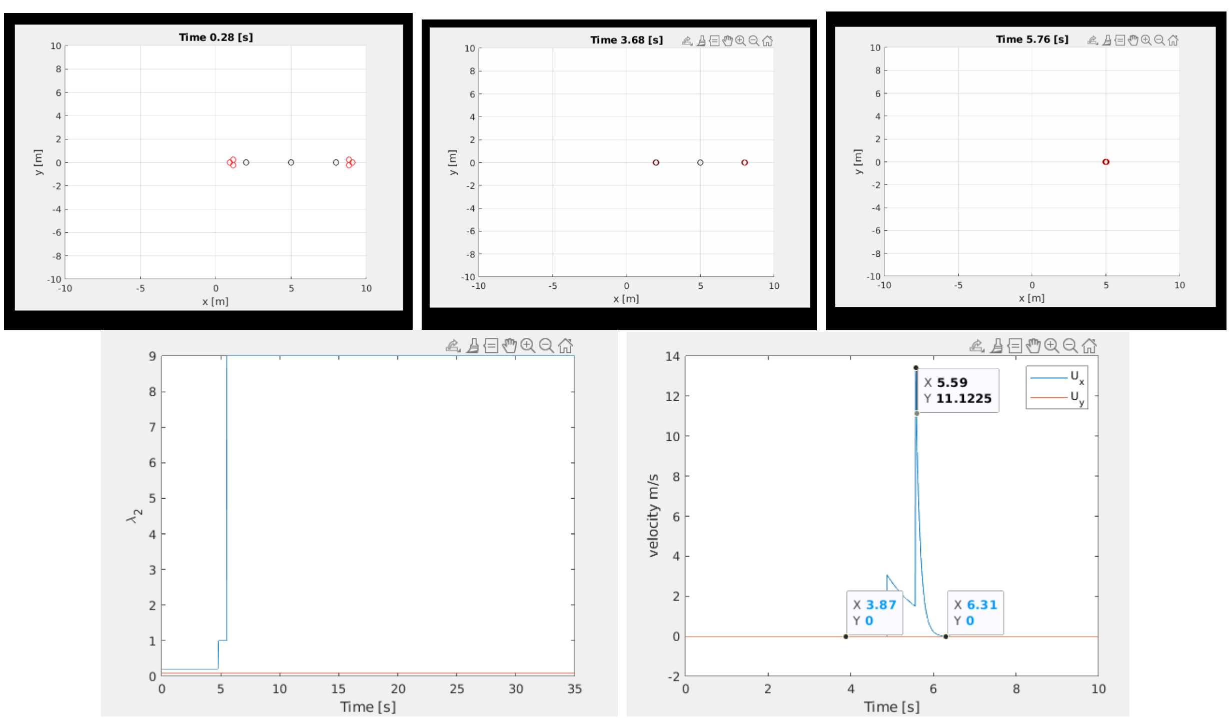

Fig. 5. Evolution of Freeze and unfreeze method for linear. The top row shows the evolution of the system at three different time steps. In the bottom the λ_2 and the velocity of the critical agent is shown. It can be seen that the system performs better than the linear weight method [1] and figure 3.

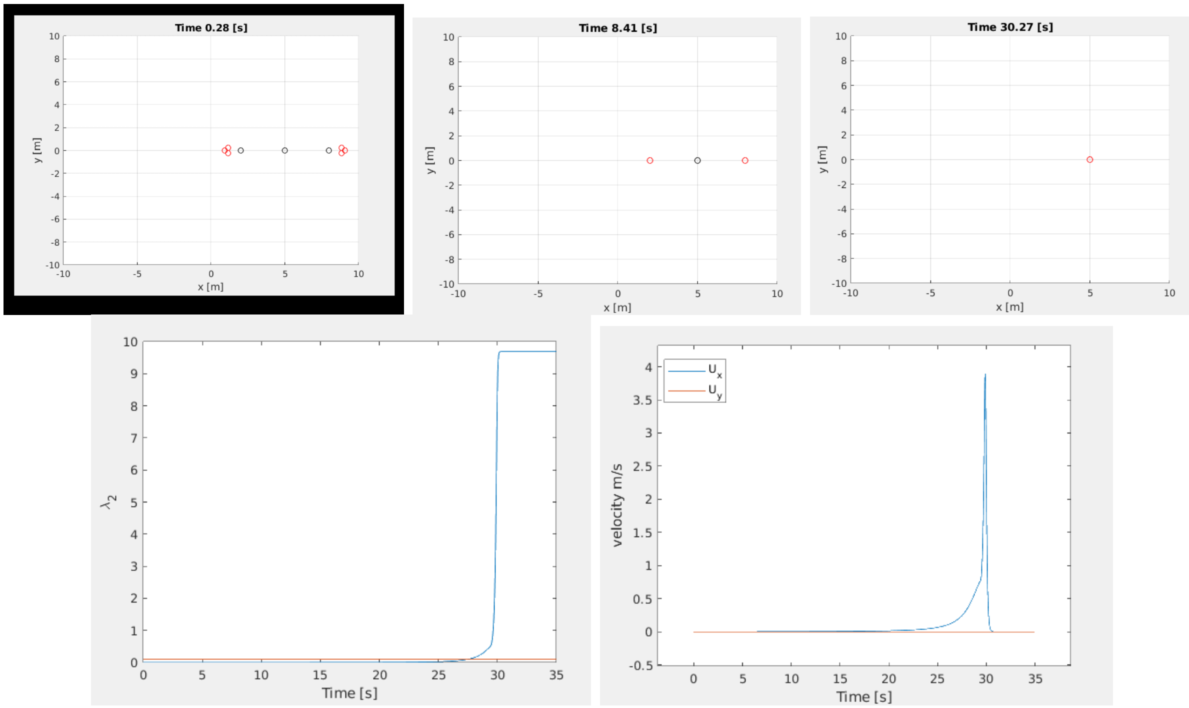

Fig. 6. Evolution of Freeze and unfreeze method for non-linear. The top row shows the evolution of the system at three different time steps. In the bottom the λ_2 and the velocity of the critical agent is shown. It can be seen that the system takes a lot of time compared to the 4 and [4]

S.NO	method	type of weight	R(cutoff distance)	Time
1	Connectedness[1]	Linear	3.25m	5.84s
2	Freeze & Unfreeze	Linear	3.01m	5.76s
3	Freeze & Unfreeze	Non-Linear	3.05m	30s
4	ConnectivityCbf[4]	non-linear	3.01m	5.6s

Fig. 7. Graph topology used for the report. Agents can only see connected by the black line.

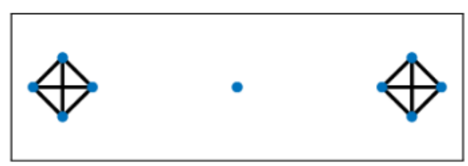

Fig. 8. Normal consensus with a cutoff distance implemented.

I. Results and Discussion

The following figures and graphs are reported as a result of the experiments explained above. A small note, the velocity of the critical agents are plotted to give an idea of how each method is different. This is done for only one of the agents as the system(agent left of the middle agent) is symmetric and the middle agent each doesn't move.

In figure 2, the plots for normal consensus is shown. In figure 3, the plots for connectedness is shown. In the figure 4, the plots for ConnectivityCbf is shown. In the figure 5 and 6 our method and the results are shown both linear and non-linear weighted.

From the plots of the figure 2 it is clear that the system will loose connectivity. It is interesting as looking at the sign of the critical agent, it goes in a direction where the graph looses connectivity. Comparing this to all the other methods it is evident that the choice of critical agents is perfect from our freeze algorithm. We can see from the table that our method does better in terms of performance time and the cutoff distance required in the linear weighted methods (Fig 3). It is although very interesting to see that our method works for non-linear weighted methods but takes a lot of time compared to Fig 4.

References

[1] Meng Ji and M. Egerstedt. "Connectedness preserving distributed coordination control over dynamic graphs". In: *Proceedings of the 2005, American Control Conference, 2005*. 2005, 93–98 vol. 1. DOI: 10.1109/ACC.2005.1469914.

[2] G. Notarstefano, M. Egerstedt, and M. Haque. "Rendezvous with multiple, intermittent leaders". In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese*

Control Conference. 2009, pp. 3733–3738. DOI: 10.1109/CDC.2009.5400560.

[3] Lorenzo Sabattini et al. "Distributed Control of Multirobot Systems With Global Connectivity Maintenance". In: *IEEE Transactions on Robotics* 29.5 (2013), pp. 1326–1332. DOI: 10.1109/TRO.2013.2267971.

[4] Beatrice Capelli and Lorenzo Sabattini. "Connectivity Maintenance: Global and Optimized approach through Control Barrier Functions". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 5590–5596. DOI: 10 . 1109 / ICRA40945 . 2020 . 9197109.

[5] G. Ferrari-Trecate et al. "Laplacian Sheep: A Hybrid, Stop-Go Policy for Leader-Based Containment Control". In: *Hybrid Systems: Computation and Control*. Ed. by João P. Hespanha and Ashish Tiwari. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 212–226. ISBN: 978-3-540-33171-1.