Dynamic trust consensus for cooperative tracker using potential field approach

Atul Shrotriya
Department of Mechanical and Aerospace Engineering
The University of Texas at Arlington
Arlington, United States
atul.shrotriya@mavs.uta.edu

***Abstract*—Swarm motion has long been of interest in robotic applications due to its wide range of applications. It is quickly becoming mainstream with the development of sensing and computing hardware along with better materials for robot development. However, working in extreme conditions can make these machines susceptible to malfunction, damage or being compromised. This paper attempts to evaluate the feasibility of using trust consensus to exclude robots that have malfunctions when they are led through a cooperative tracker without human intervention using the potential field approach. It is found that dynamic trust consensus is a reliable alternative to achieve such collaborative motion using distributed decision and control techniques.**

I. INTRODUCTION

Swarm motion is of high importance within a variety of scenarios especially in cases where human intervention is not feasible or quick response is required. Researchers have studied the application of collaborative motion for a variety of purposes as it is also useful higher efficiency, reliability and distributed detection [1,2] among other uses. Collaborative motion can be readily seen in natural and biological systems such as the movement of birds and fishes. It is also evident in human behavior and can be observed through voting patterns, internet connectivity etc. Studying these systems further boosts the foundations of distributed decision and control theory while gaining more knowledge through deeper studies. [3]

Different types of methodologies and formations are used for guiding a swarm of robots to its target. Formations can include varying topologies, directed or undirected connection graphs and have limitless possibilities for the connecting methods used [4]. Different approaches for swarm control can include methods such as creating multiple targets for swarm motion, consideration of only the nearest obstacle to avoid force balance points which are away from the target or jump algorithms to initiate a random movement in case the robot gets stuck away from the target [5,6]. However, robots can malfunction or be compromised when operating in harsh environments. Increasing the robustness of each individual unit will result in higher costs which increase rapidly as the number of involved devices increases. This paper aims to propose an approach for dynamically determining a node(robot)'s failure and cutting it from the graph structure. This will allow other nodes to reach the target quickly even if one of the nodes gets stuck.

II. TRUST CONSENSUS WITH RANDOM HEADINGS

Trust consensus can be applied using (1), (2) and (3) on a connected graph structure. The heading angles are chosen randomly and trust consensus is applied with random initial trust vectors.

A. Trust consensus when all nodes function well and there is no mistrust among them

When there is no mistrust, the trust gain can be completely removed and Reynolds' Rules [7] for heading alignment are used:

$$\dot{\theta}_i = \sum_{j \in N_i} a_{ij}(\theta_j - \theta_i) \quad (1)$$

Here a_{ij} is the connection gain from node j to node i. θ and $\dot{\theta}_i$ simply represent the heading and heading change for the node i at a particular time step. Doing this for all nodes can be achieved in a simpler way with the help of matrices. After derivation, the following formula is obtained:

$$\dot{x} = -(D - A)x = -Lx \quad (2)$$

Where D is a diagonal matrix indicating the in-degree for a graph structure with each diagonal element showing the number of in nodes and the A matrix gives information on the direction of connectivity for every connection. The L matrix is highly useful for figuring out the stability of the formation structure using Gershgorin criteria. [8]

The consensus control was implemented using (1) and (2) in MATLAB for the graph structure shown in Figure (1). Same results for consensus were obtained for each case with a similar time required for convergence of the agents. The convergence results are visible in Figure (2) and Figure (3).

The University of Texas at Arlington

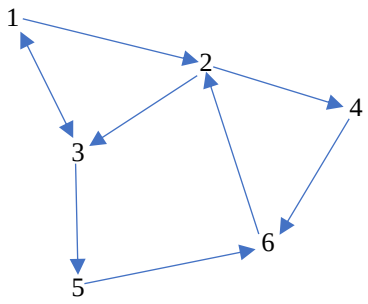

Figure 1. Connection graph structure

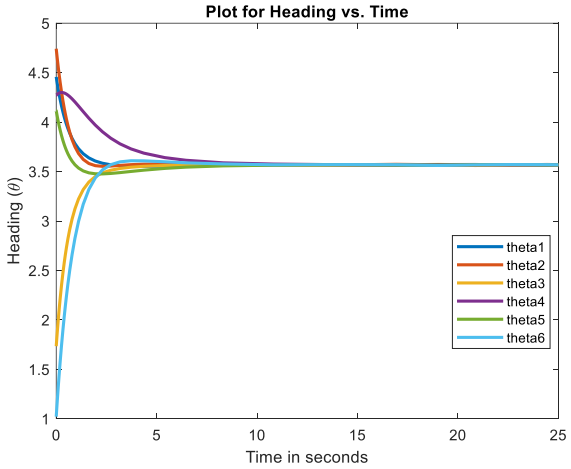

Figure 2. Plot of heading angle vs. time

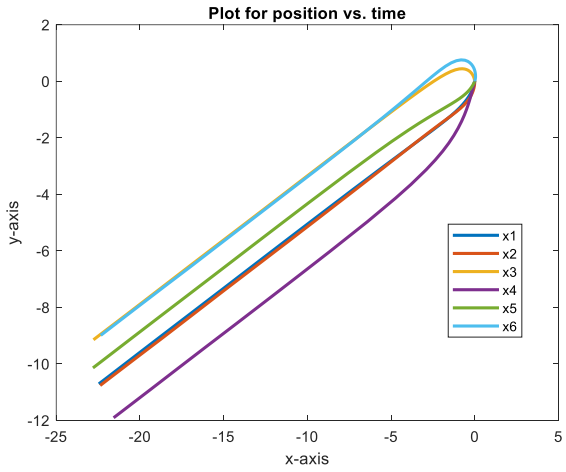

Figure 3. Plot of position in the x-y plane

B. Application of trust consensus to the same graph

The following steps were taken for trust consensus:

1. Random initial trust vectors were defined with a mistrust for node 5 by nodes 2 and 4
2. Trust consensus was achieved using (1) and (3) and negative trust was established for node 5. [9]

$$\dot{\xi}_i = \sum_{j \in N_i} a_{ij}(\xi_j - \xi_i)$$

$$\dot{\theta}_i = \sum_{j \in N_i} \xi_i a_{ij}(\theta_j - \theta_i) \qquad (3)$$

Here, ξ_i is a vector matrix representing the difference of opinion of node i with other nodes in the formation.

3. An algorithm was developed and implemented to figure out which node has a negative trust consensus.
4. This node was cut off from the graph by replacing its gain value with 0 in the D and A matrices.
5. While the graph would not able to reach consensus using either (1) or (2), after the application of this algorithm, it was able to achieve consensus as shown in Figures (4), (5) and (6).

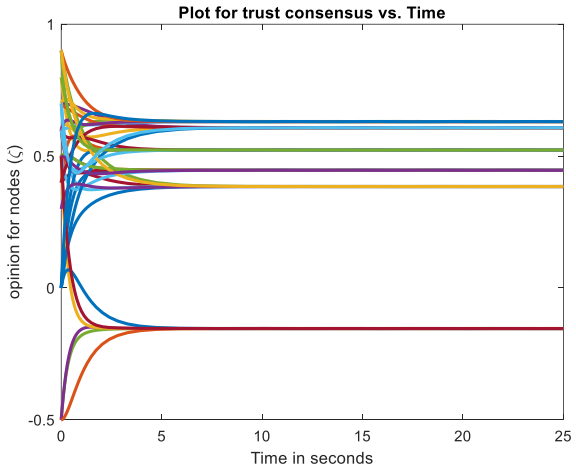

Figure 4. Convergence of opinion for negative trust towards node 5

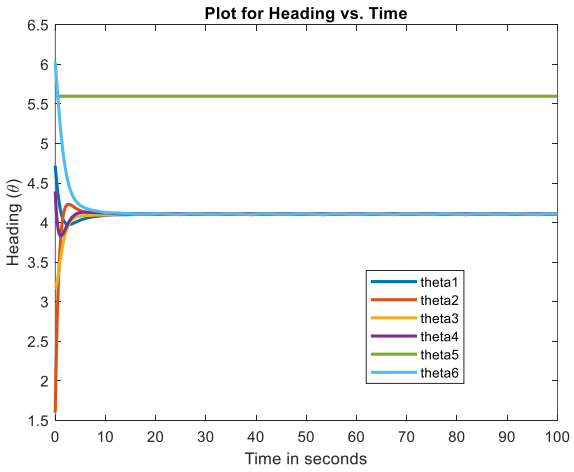

Figure 5. Heading convergence with node 5 cut-off

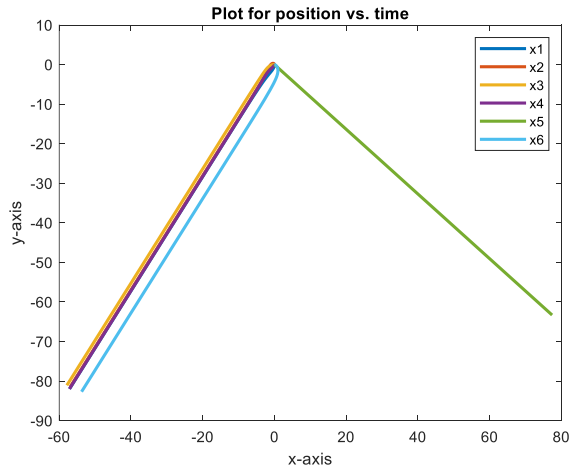

Figure 6. Plot of positions in x-y plane with node 5 removed from the graph structure

III. Trust Consensus with a Leading Node

The convergence of 7 common graph structures along with the graph in Figure 1 was studied. The common graph structures all had 6 nodes and are Directed Tree graph, Undirected Tree Graph, Star $K_{1,5}$ graph, Star with extra edges, Directed 6-cycle graph (6-periodic graph), 6-cycle C^6 (2-regular graph) and Path P^5 graph. After comparing the convergence rates and the impact of cutting a faulty node, the Star with extra edges (Figure 8) was chosen. [10]

A. *Heading convergence in the presence of Leading Node*

Following steps were taken to achieve heading convergence for star with extra edges graph structure.

1. A leader node was defined at a constant state of 5.
2. The initial condition for the nodes in the graph structure were defined randomly between 1 and 10
3. Central node was connected to leading node and cooperative tracker algorithm was applied using (4).

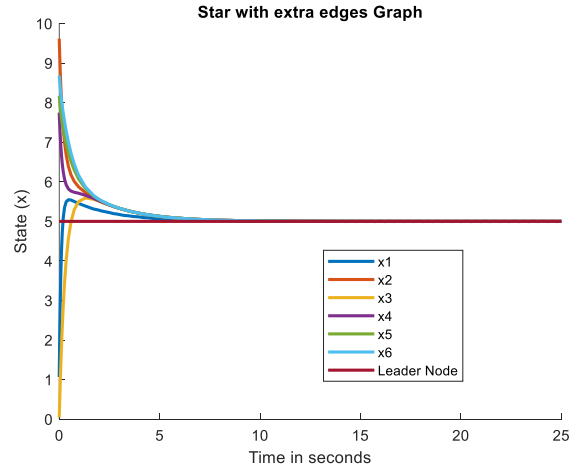

Figure 7. Convergence of all the 6 nodes to the leader node

$$\dot{x} = -(L+G)x + (L+G)\underline{x_0} \quad (4)\ [11]$$

Where G is the diagonal pinning gain matrix connecting the leader node x_0 to the graph structure.

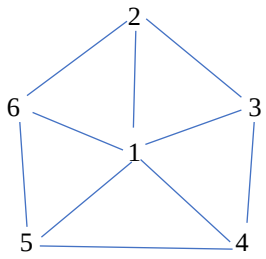

Figure 8. Star with extra edges graph structure

It can be seen from Figure 7 that all the nodes in the graph structure quickly converge to the position of the leader node i.e., 5 within less than 10 seconds.

B. *Trust Consensus in the presence of Leading Node*

The trust consensus algorithm was applied to the star with extra edges graph structure. Initial mistrust conditions dictated that node 2 and 4 have a mistrust for node 5.

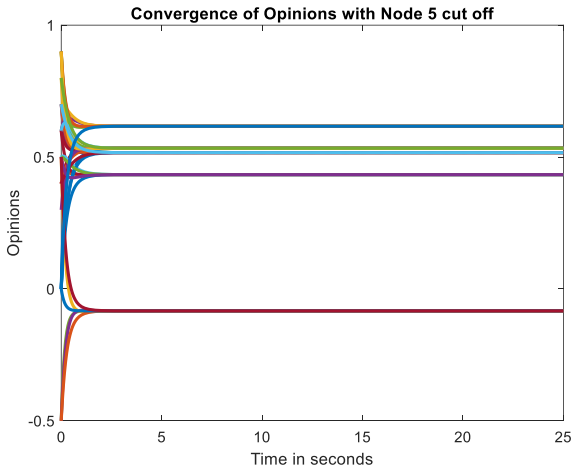

Figure 9. Convergence of Opinions in star formation

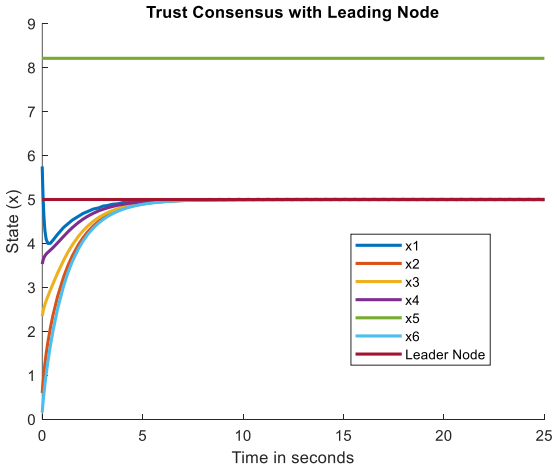

Figure 10. Convergence of nodes to leader node

IV. Potential Field Approach for Star with Extra Edges Graph Structure

Star graph formation was implemented in a potential field with the target set to (10,10) and two obstacles set at (3,4) and (8,5) in the x-y plane respectively. The velocity was kept constant at 1m/s. The following steps were taken to model the system and achieve collaborative motion using star with extra edges graph structure.

A. Preparation of the potential field and modeling of robot dynamics for motion

1. Potential field was modeled using (5) and (6). [12]

$$F_x = -\frac{\partial V}{\partial x} = k_T \frac{(x_T - x)}{r_T}$$
$$F_y = -\frac{\partial V}{\partial y} = k_T \frac{(y_T - y)}{r_T} \qquad (5)$$
$$F_x = -\frac{\partial V}{\partial x} = -\frac{k_o}{r_o^2}\frac{(x_o - x)}{r_o}$$
$$F_y = -\frac{\partial V}{\partial y} = -\frac{k_o}{r_o^2}\frac{(y_o - y)}{r_o} \qquad (6)$$

In (5) the attractive forces by the target are defined whereas in (6) the repulsive forces by obstacles are defined in the x and y directions. Here, k_T and k_o are the gains associated with the target and obstacle whereas r_T and r_o define the distance of the body from the target and obstacle respectively. Similarly, (x_T, y_T), (x_o, y_o) and (x, y) define the positions of target, obstacles and the body in the x-y plane respectively.

2. Potential forces are combined using (7)

$$\vec{F} = \vec{F}_{target} + \sum_{i=1}^{N} \vec{F}_{obstacle\ i} \qquad (7)$$

3. On plotting these potential fields, Figures 11 and 12 are generated in MATLAB.

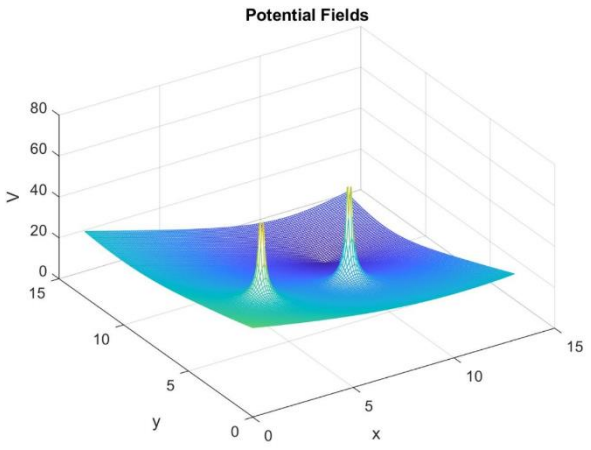

Figure 11. 3D Plot of potential field with obstacles and target

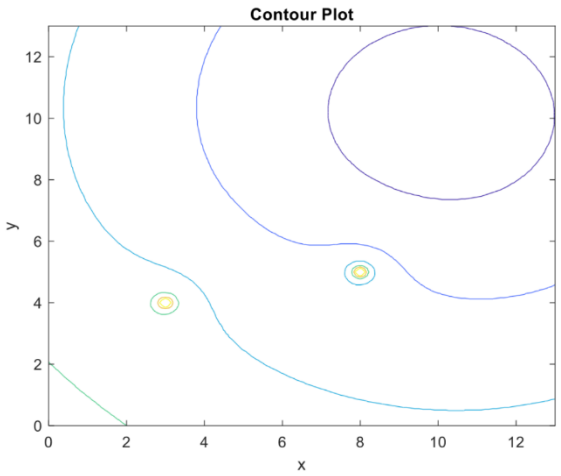

Figure 12. Contour plot for potential field and target

4. The dynamics of each robot was modeled as (8) using a front steered robot as shown in Figure 13.

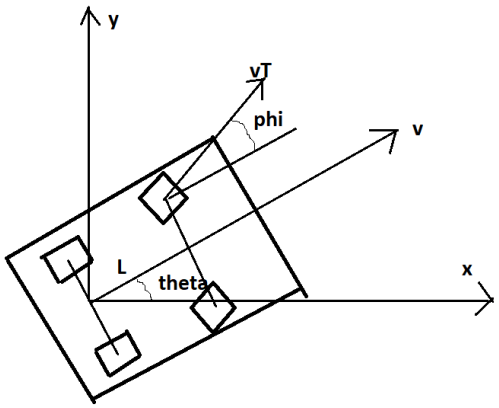

Figure 13. Front steered robot model

$$\dot{x} = vT * cos(phi) * \cos(theta)$$
$$\dot{y} = vT * cos(phi) * \sin(theta)$$
$$\dot{\theta} = \frac{vT}{L} * sin(phi) \qquad (8)$$

5. Here the control input is the steering angle phi.

B. Formation control for star with extra edges graph using theta consensus protocol

Following measures were taken to achieve formation control for star with extra edges.

1. The length shown as L in Figure 13 was kept as 2.
2. The gains for target and obstacles were set as 3 and (4,5) respectively.

3. Small repulsive potentials were given to each robot to avoid collision while heading towards the target. These repulsive forces were turned to 0 as the nodes started to reach the target.
4. Control protocol shown in Figure 14 was used to control each of the nodes.

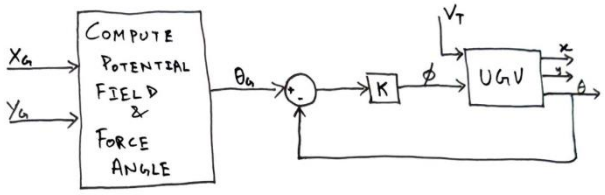

Figure 14. Control Protocol for motion of each node

5. In Figure 14, K is the total steering gain for motion of the unmanned (UGV) nodes.
6. The robots moved independently towards the target without interaction with each other when there was no formation and some get stuck as shown in Figure 15.

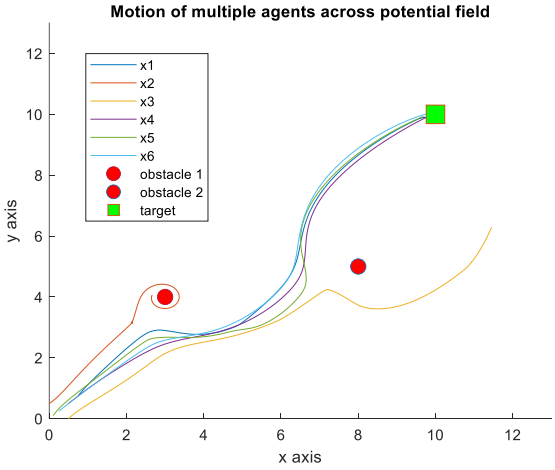

Figure 15. Motion of multiple agents independently

7. The control protocol shown in figure 14 was modified to add a gain from consensus for theta values as shown in Figure 15.

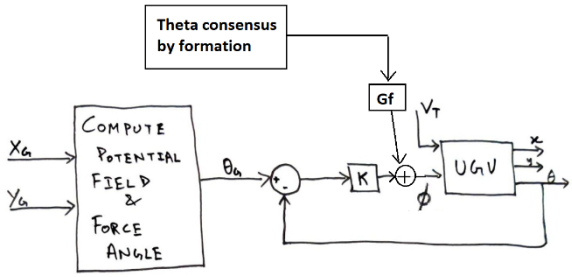

Figure 16. Potential field motion with theta consensus

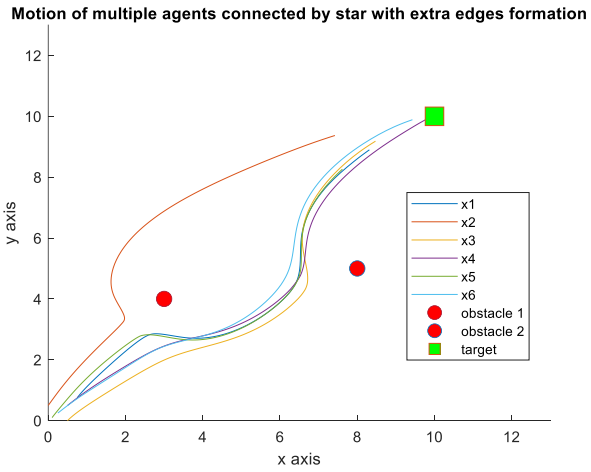

Figure 17. Motion of multiple agents across potential field when connected by star with extra edges formation

8. Theta consensus works well and it is seen that the nodes that were stuck previously are now able to move forward towards the target using the theta consensus protocol.
9. However, this motion is restricted and only works in the case when the initial condition of nodes is really close so that the change of heading angle helps with their motion.
10. It should be noted that this formation is still robust and can work well until the central node is placed too close to an obstacle. In this case, the motion of the remaining nodes overpower it and drive it towards the obstacle as seen in Figure 18.

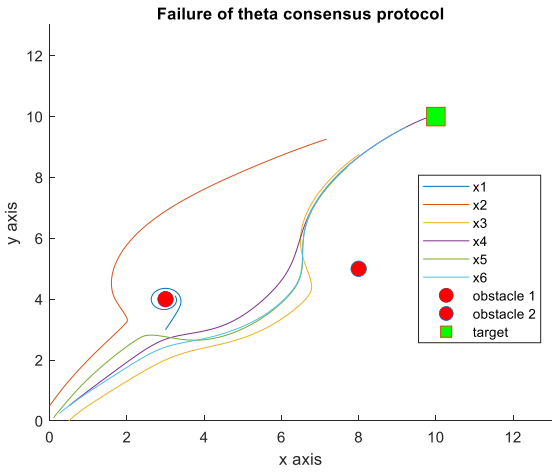

Figure 18. Failure of theta consensus protocol when central node is placed too close to the obstacle

11. The robustness of this control algorithm further decreases as the leader node is introduced.
12. Although the nodes perform well when placed within close distances, it malfunctions when the leader node

is placed ahead of the other nodes in the formation structure as shown in Figure 19-a and Figure 19-b.

13. The leader node only pins to the central node as it has the highest number of connections within this graph structure.

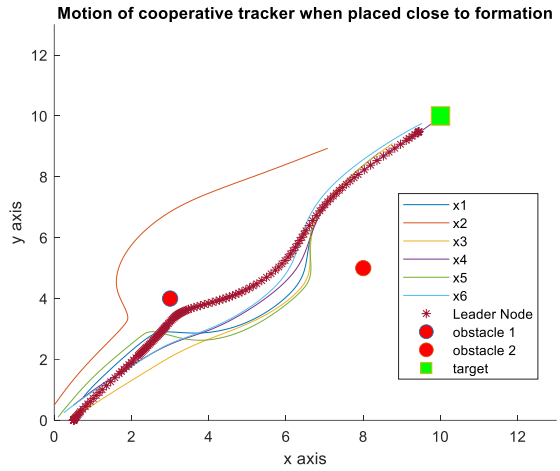

Figure 19-a. Reliable motion of co-operative tracker formation when leader node is placed close within the formation

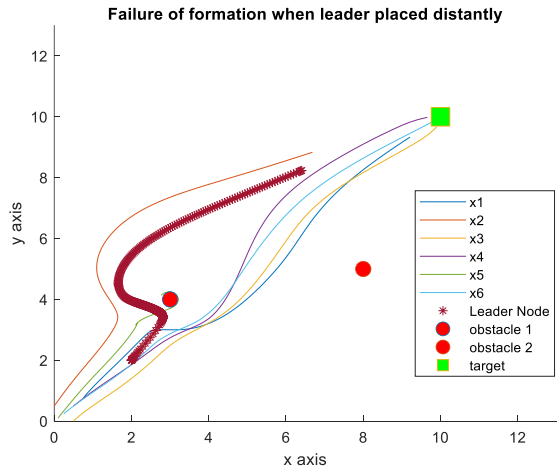

Figure 19-b. Failure of formation as leader is placed at a distance from the connected nodes

14. It should be noted that the leader does not interact with the other nodes through a repulsive gain. While the same principle can be applied, it is intended for the leader to be ahead of the formation and thus not pose a chance of collision with other nodes. However, if required due to complex structures or motion, this can be changed by simply giving the leader node a repulsive field and enabling it to interact with repulsive fields from other nodes.

15. Another observation is that this formation can still function well if the initial heading angles are chosen carefully. However, this can be time consuming and might not be a feasible solution for all scenarios especially when working over longer distances.

C. Formation control for star with extra edges graph using position consensus protocol

Since the theta consensus protocol is not robust enough, a new protocol based on position is developed. This protocol adds a gain to the derivative of x and y positions for each node and is demonstrated in Figure 20.

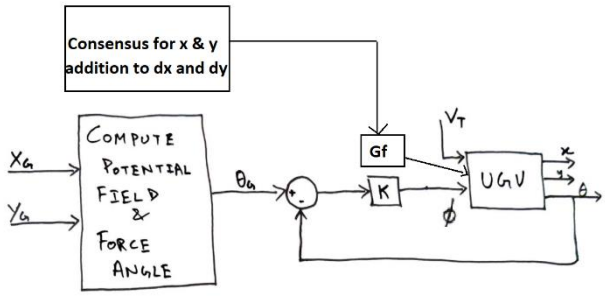

Figure 20. Consensus protocol by adding the formation change terms to position change of unmanned nodes

1. On testing this algorithm, a much robust control formation is achieved.
2. It is tested using various positions and the performance is quite well in all cases irrespective of initial positions.
3. It is observed that the system is robust even when the node 1 has same initial condition of being near to an obstacle as in previous case.
4. The corresponding results plotted are shown in Figure 21-a and 21-b.

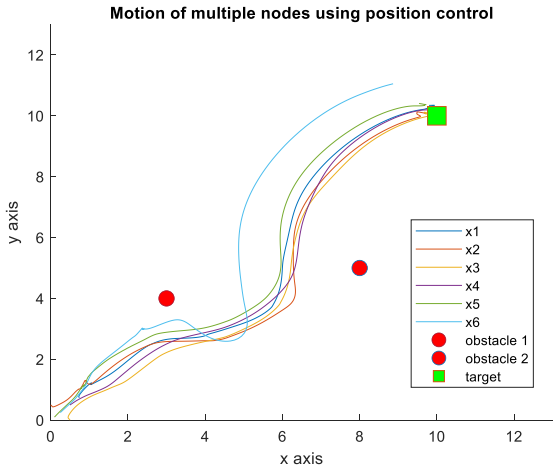

Figure 21-a. Motion of multiple nodes when using the position control algorithm

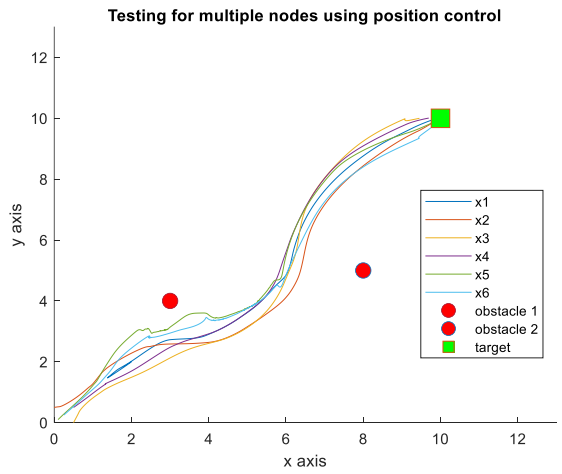

Figure 21-b. Testing the behavior of nodes by placing the central node initially at (2,2)

D. Leading the formation using a leader node

1. As the position control method performs satisfactorily, a leader node is introduced to guide the formation.
2. The leader node can start anywhere in the graph but it performs best when placed near to the path for determining the motion independently.

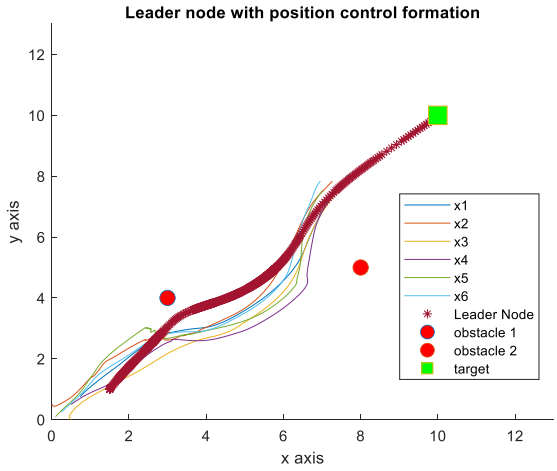

Figure 22. Leader node leading the formation using position control algorithm

3. It is visible from Figure 22 that introducing the leader node significantly reduces the unnecessary motion within the nodes that are connected through the star graph structure.
4. It performs reliably even when the leader node is initiated at a location where it has to take a completely different route than the other nodes.
5. While the motion is a bit choppy, the performance is still quite acceptable as seen in Figure 23.

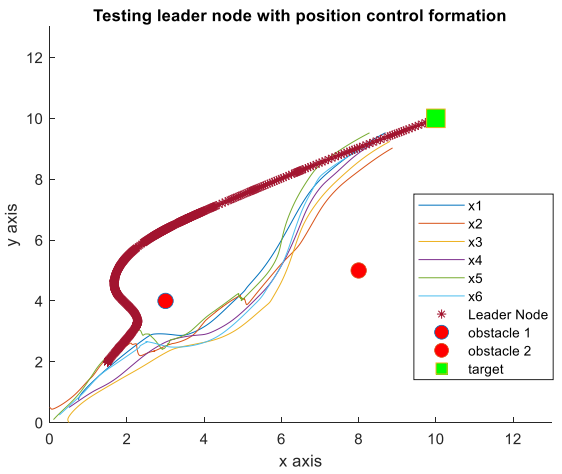

Figure 23. Testing the leader node as it takes a different path as compared to the other nodes

6. The gains for target, obstacles, individual repulsive forces, formation graph structure and pinning matrix can be adjusted suitably to obtain the best possible motion depending on the scenario.

V. TRUST CONSENSUS IN FORMATION STRUCTURES

The trust consensus protocol is applied with and without the leader to check for the performance using following steps:

1. The initial matrices are set to zero.
2. Depending on the distance between two nodes, their trust is determined for each other.
3. The trust factor is divided into steps to ensure that a small deviation doesn't automatically push the trust into negative values.
4. An ode function is then initialized within the original ode function to determine the trust vector and figure out the erroneous node.
5. The erroneous node is then cut off from the graph structure.
6. As seen in Figure 24 and Figure 25, the trust consensus factor is very volatile.
7. The increments for the trust consensus factors must be defined carefully for expected motion of the formation structure. Otherwise, it is possible for the nodes to behave in rapid erratic movements.
8. In some cases, the nodes may even diverge from the path altogether after being cut off from the graph structure.
9. This may be due to a programming error and needs to be thoroughly checked.

10. However, when the trust or opinion increments are defined correctly, the nodes behave as expected and are able to reach the target.
11. If a node is cut off in a position where the repulsive forces and attractive forces balance, it may get lost.
12. A jump algorithm can be implemented to allow such nodes to reach the target by taking a jump towards it.

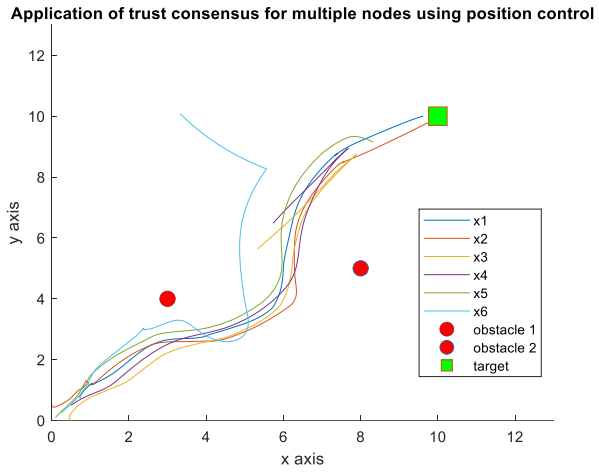

Figure 24. Erratic behavior of formation when trust steps are not defined well

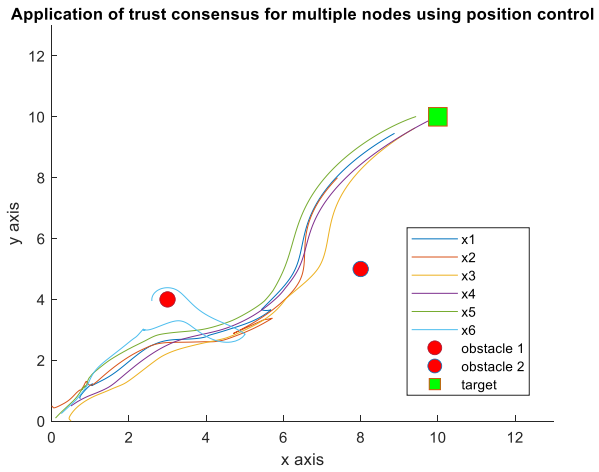

Figure 25. Improved motion of the formation structure as the trust steps are tuned

13. The same algorithm is then implemented with a leader node to check for results.
14. As seen in Figure 26, the results improve significantly with all nodes reliably moving towards the target.
15. Upon testing, it is observed that the formation behaves well and all nodes are still able to reach the target even when the leading node takes a different path. This can be seen in Figure 27.
16. It should be noted that the gain for formation and leading node are kept equal and changing these gains can cause significant changes to the movement of the nodes.

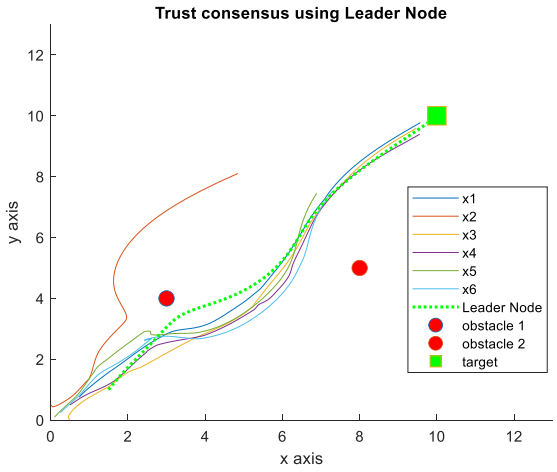

Figure 26. Application of trust consensus algorithm using the leader node

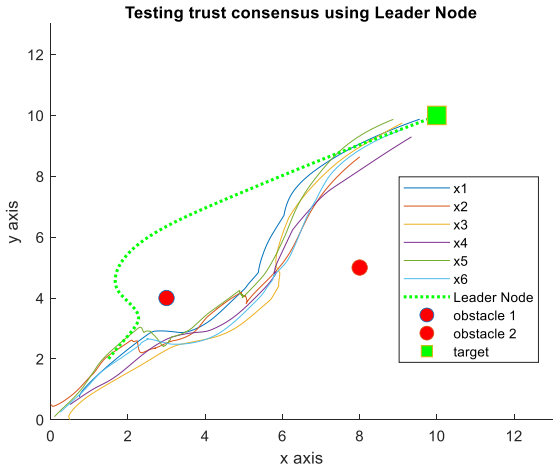

Figure 27. Testing the trust consensus algorithm

17. Another observation is that many times all the nodes are left behind and are thus cut off from the formation. By the end, all nodes are behaving independently.

VI. Conclusions and Discussion

Based on different algorithms, it was observed that application of a leader node along with trust consensus significantly increases the reliability of a formation structure to reach its goal within a potential field.

To avoid the cut off of many nodes towards the end, a simple reset algorithm can be implemented for the L matrix so that nodes can rejoin the structure if they are moving towards the same point.

This methodology can further be improved to include dynamically changing graphs or for formations where a

specific type of formation is desired position wise. The trust consensus can be further modified by reducing the step sizes and even normalizing the deviation from the average of the position of other nodes.

ACKNOWLEDGMENT

I would like to thank the University of Texas at Arlington for providing the necessary resources to perform this research. I would like to thank my research advisor Dr. David A. Hullender for introducing me to the concept of Dynamic Systems Modeling and Simulation which proved to be crucial for this paper. I'm immensely grateful to my professors, Dr. Frank Lewis and Dr. Michael Niestroy for explaining the concepts to me in a practical manner which allowed me to implement them in a versatile manner on this project. I would further like to thank Dr. Frank Lewis and Dr. Michael Niestroy for brainstorming with me to find the best available solutions for formation control within potential fields.

REFERENCES

[1] Y. Tang, H. Gao, J. Kurths and J. Fang, "Evolutionary pinning control and its application in UAV coordination", IEEE Trans. Ind. Informat., vol. 8, no. 4, pp. 828-838, Nov. 2012.

[2] L. Liu, D. Wang, Z. Peng and H. H. Liu, "Saturated coordinated control of multiple underactuated unmanned surface vehicles over a closed curve", Sci. China Inf. Sci., vol. 60, no. 7, 2017.

[3] F. L. Lewis, "Cooperative Control of Multi - Agent Systems on Communication Graphs," Distributed Decision and Control EE 5329, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Jan., 19, 2021

[4] D. Yu, C. L. P. Chen, C. -E. Ren and S. Sui, "Swarm Control for Self-Organized System With Fixed and Switching Topology," in *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4481-4494, Oct. 2020, doi: 10.1109/TCYB.2019.2952913.

[5] D. D. Tsankova and N. Isapov, "Potential field based formation control in trajectory tracking and obstacle avoidance tasks," *2012 6th IEEE International Conference Intelligent Systems*, 2012, pp. 76-81, doi: 10.1109/IS.2012.6335117.

[6] M. Zhang, Y. Shen, Q. Wang and Y. Wang, "Dynamic artificial potential field based multi-robot formation control," *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings*, 2010, pp. 1530-1534, doi: 10.1109/IMTC.2010.5488238.

[7] C. W. Reynolds, "Flocks Herds and Schools: A Distributed Behavioral Model", *Computer Graphics*, pp. 21, 1987.

[8] F. L. Lewis, "Graphs, Dynamic Graphs, Gershgorin, Eigenstructure," Distributed Decision and Control EE 5329, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Feb., 23, 2021

[9] X. Liu and J. S. Baras, "Using trust in distributed consensus with adversaries in sensor and other networks," *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1-7.

[10] F. L. Lewis and L. Bosen, "Homework 3 - Consensus for Different Graph Eigenvalues," Distributed Decision and Control EE 5329, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Mar., 2, 2021

[11] F.L. Lewis, H. Zhang, K. Hengster-Movric, A. Das, *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches*, SpringerVerlag, Berlin, 2014.

[12] F.L. Lewis and E. Stingu, *Potential Fields in Cooperative Motion Control and Formations*, Intelligent Control Systems EE 5322, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Feb., 24, 2018

[13] D. A. Hullender, "Dynamic Systems Modeling and Simulation," Dynamic Systems Modeling ME 5305, Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Nov. 14, 2019

[14] M. A. Niestroy – "Optimal Control using LQR Tracker," Optimal Control EE 5321, Department of Electrical Engineering, The University of Texas at Arlington, Apr., 26, 2021