Traffic Management Strategies for Multi-Robotic Rigid Payload Transport Systems

by

Yahnit Sirineni, Pulkit Verma, Kamalakar Karlapalem

in

IEEE International Symposium on Multi-Robot and Multi-Agent Systems
(*IEEE MRS 2019*)

Report No: IIIT/TR/2019/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2019

Traffic Management Strategies for Multi-Robotic Rigid Payload Transport Systems

EXTENDED ABSTRACT

Yahnit Sirineni, Pulkit Verma, Kamalakar Karlapalem

Abstract— **In this work, we address traffic management of multiple payload transport systems (PTS) comprising of non-holonomic robots. Each PTS is a loosely coupled rigid robot formation carrying a payload. We ensure each PTS traverses its desired trajectory while avoiding collisions with other PTS and static obstacles in various kinds of complex environments. Each PTS has one leader and multiple followers and the followers maintain a desired distance and angle w.r.t the leader using a decentralized leader-follower control architecture. We showcase through simulations that our strategies help manage traffic for a large number of PTS moving around.**

I. INTRODUCTION

Many methods have been developed for Payload transportation[1][2], control and navigation of formations. [3], [4] propose navigation methods for a single formation without much emphasis on obstacles. [5], [6], [7], [8] shows navigation of single formation in environments with static obstacles. Navigation of a single formation using the leader-follower approach is shown in [7] and [9]. A computationally expensive centralized method for environments with dynamic obstacles is illustrated in [10]. Multi-agent pathfinding for multiple non-holonomic robots is shown in [11].

To the best of our knowledge, little or no work has been done on collision avoidance of multiple PTS/formations moving around in complex environments with obstacles. However, work has been carried out on collision avoidance of multiple robots/agents [12][13][14]. We could simply consider other PTS as dynamic obstacles and use the approach presented in [10][15] but we are not exploiting the idea that each PTS has a specific goal and *each PTS can take equal responsibility to avoid a collision* as given in [12].

II. BACKGROUND

A. Leader-Follower Formation Control

Each PTS comprises of a leader and multiple followers. A decentralized leader-follower control algorithm [16] is used in which the followers use the command velocities of the leader to derive their velocities such that they maintain a desired distance and angle w.r.t the leader.

Yahnit Sirineni and Pulkit Verma are students of Agents and Applied Robotics Group (AARG), Kohli Center on Intelligent Systems (KCIS), International Institute of Information Technology, Hyderabad (IIIT-H), India sirineni.yahnit@research.iiit.ac.in

Kamalakar Karlapalem is a faculty of Computer Science with International Institute of Information Technology, Hyderabad, India kamal@iiit.ac.in

B. Optimal Reciprocal Collision Avoidance for non-holonomic robots (nh-ORCA)

nh-ORCA [17] is a robust decentralized collision avoidance algorithm for non-holonomic robots which is an extension to ORCA[12] which deals only with holonomic robots. Each robot independently computes its velocity such that it is collision free for at least τ time assuming that the other robots also compute their velocity using the same method. We extend this approach to our problem for avoiding collisions among other formations and obstacles.

III. TRAFFIC MANAGEMENT

A decentralized leader follower-ORCA-RRT* framework is discussed which proposes efficient traffic management strategies for path planning and collision avoidance of PTS. Schematic of the system architecture is shown in Fig.1.

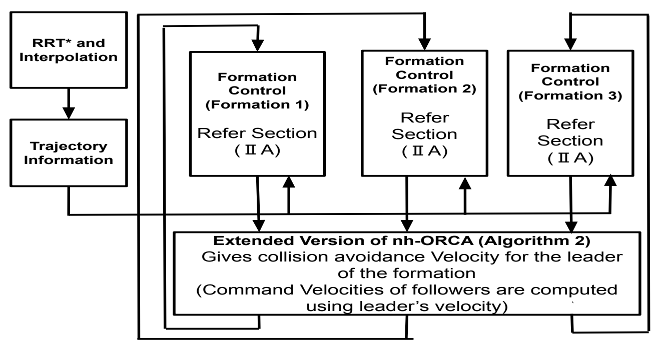

Fig. 1: System Architecture

Algorithm 1: Leader Follower-ORCA-RRT*

Input : Environment description, source and destination of formations, Configuration of formations ;
$path_i$ = RRT*($source_i$, $destination_i$) ;
$path_i$ = Interpolation($path_i$) ;
$nextDest_i$ = first way point of interpolated $path_i$;
v_i^{pref} = ($nextDest_i$ - p_i)
while f_i *has not reached destination* **do**
  timestep ++ ;
  v_i, ω_i = Algorithm2(i);
  LeaderFollower(v_i, ω_i) as given in [16] ;
  Apply corresponding controls to the leader and followers of the formation;
end

Algorithm 2: Extended nh-ORCA

Input : $path_i$, p_i, v_i, r_i, v_i^{pref}, v_i^{max}, $neighbours_i$;
Compute collision-free non-holonomic velocity for the Leader of the formation f_i as given in [17];
if *distance*(p_i, $nextDest_i$) $\leq \delta$ **then**
  $nextDest_i$ = next point of interpolated path;
  v_i^{pref} = ($nextDest_i$ - p_i);
return non-holonomic command velocities of Leader.

Multiple PTS are considered where each PTS is a set of robots with a leader and its followers working together to perform a particular task. Follower j of each formation (f_i) maintains a certain angle and distance relative to its leader. Each formation f_i has a current position p_i, radius r_i i.e the desired distance of followers w.r.t leader, current velocity v_i, number of followers $followers_i$, source src_i, destination $dest_i$, next immediate destination $nextDest_i$, preferred direction v_i^{pref} i.e. the direction through which it would traverse had there been no obstacles in its path, maximum velocity v_i^{max}, neighbours $neighbours_i$ which indicates the positions and velocities of neighboring formations, $neighbourDist_i$ which refers to the radius of visibility.

Each formation f_i uses the RRT*[18] algorithm to find the preferred path from src_i to $dest_i$ considering static obstacles and interpolates this path to get multiple way points. The $nextDest_i$ for each formation f_i is set to the first point of the interpolated path and the preferred direction v_i^{pref} for each formation is set to ($nextDest_i$ - p_i).

At each timestep, leader of each formation computes non-holonomic collision-free velocities as given in [17]. If the formation f_i at a distance less than δ from $nextDest_i$ where δ is an environment based parameter, then $nextDest_i$ is updated to the next point of the interpolated path and the preferred direction of formation v_i^{pref} to ($nextDest_i$ - p_i). This extended version of nh-ORCA is given in Algorithm 2.

Once the velocities of leaders are computed using Algorithm 2, command velocities of all the followers are computed using a decentralized leader-follower architecture[16] which are then applied to all the robots. The Leader Follower-ORCA-RRT* framework is given in Algorithm 1.

IV. Results

We consider an environment with five static obstacles and four PTS carrying payloads. Fig. 2 shows snapshots of the simulation carried. We observe that all the PTS successfully complete their task while avoiding collisions.

Fig. 3 shows that a PTS always maintains a minimum threshold distance with other PTS by plotting the distance of the closest neighboring PTS for each PTS at every instant. We also present the velocity variations of each PTS as it comes close to other PTS in Fig. 4. We observe that a PTS changes its velocities in such a way that it avoids collisions. We mark points A, B, C, D in Fig. 3 and Fig. 4 to relate variations in velocities of formations while avoiding collisions. For example, formation 2 and formation 4 are close to each other at Point B, the velocities of these formations are chosen such that they avoid a collision. Similar is the case with formation 1 and formation 3 at point B.

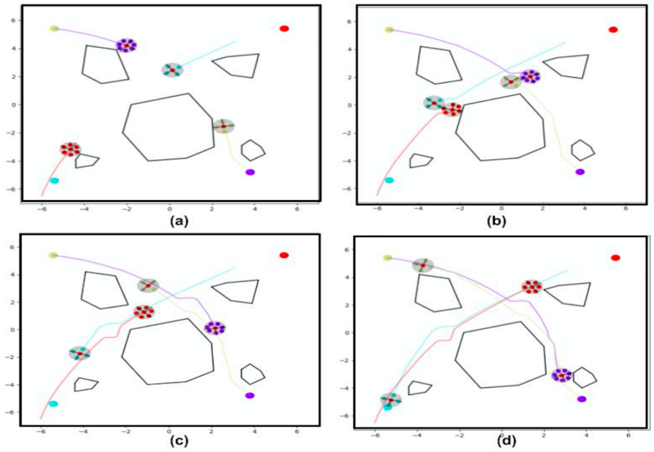

Fig. 2: Snapshots of the simulation

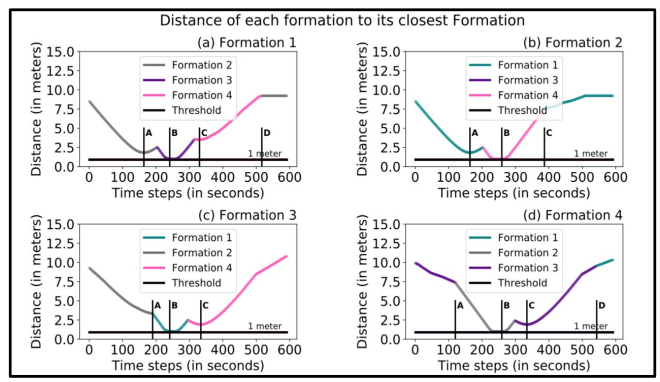

Fig. 3: Distance of each PTS with closest neighboring PTS.

In this simulation, we have set v^{max} of each robot as 0.03 m/s and radius of the formation as 0.35 m. However, these parameters can be set based on our requirements on desired speed, size and weight of payload.

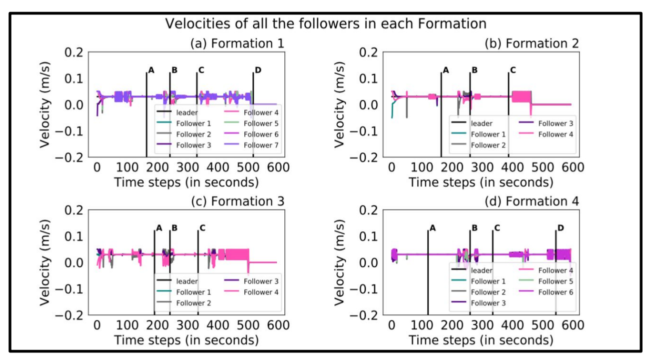

Fig. 4: Velocities of robots of all PTS

Being a decentralized framework, our system scales well with increase in number of robots/formations making it efficient, robust and a reliable method to work with large number of PTS in industrial/warehouse environment. More simulation results are given in [19].

REFERENCES

[1] P. Verma, R. Tallamraju, A. Rawat, S. Chand, and K. Karlapalem, "Loosely coupled payload transport system with robot replacement," in *Autonomous Robots and Multi-Robot Systems (ARMS) with AAMAS*, 2019.

[2] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[3] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME transactions on mechatronics*, vol. 16, no. 6, pp. 1021–1031, 2011.

[4] E. Pereyra, G. Araguás, and M. Kulich, "Path planning for a formation of mobile robots with split and merge," in *International Conference on Modelling and Simulation for Autonomous Systems*, pp. 59–71, Springer, 2017.

[5] Y. Shapira and N. Agmon, "Path planning for optimizing survivability of multi-robot formation in adversarial environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4544–4549, IEEE, 2015.

[6] L. Polkowski and P. Osmialowski, "Navigation for mobile autonomous robots and their formations: An application of spatial reasoning induced from rough mereological geometry," in *Mobile robots navigation*, InTech, 2010.

[7] S. Garrido, L. Moreno, J. V. Gomez, and P. U. Lima, "General path planning methodology for leader-follower robot formations," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 64, 2013.

[8] R. Gautam, R. Kala, *et al.*, "Motion planning for a chain of mobile robots using a* and potential field," *Robotics*, vol. 7, no. 2, p. 20, 2018.

[9] A. N. Asl, M. B. Menhaj, and A. Sajedin, "Control of leader–follower formation and path planning of mobile robots using asexual reproduction optimization (aro)," *Applied Soft Computing*, vol. 14, pp. 563–576, 2014.

[10] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[11] W. Hönig, T. S. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian, and S. Koenig, "Multi-agent path finding with kinematic constraints," in *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016.

[12] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[13] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," *arXiv preprint arXiv:1702.01182*, 2017.

[14] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE, 2017.

[15] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, pp. 1–22.

[16] Z. Peng, G. Wen, A. Rahmani, and Y. Yu, "Leader–follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach," *Robotics and autonomous systems*, vol. 61, no. 9, pp. 988–996, 2013.

[17] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*, pp. 203–216, Springer, 2013.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[19] Y. Sirineni, P. Verma, and K. Karlapalem, "Traffic management strategies for multi-robotic rigid payload transport systems," *arXiv preprint arXiv:1906.11452*, 2019.