Decentralized Collision Avoidance and Motion Planning for Multi-Robot Deformable Payload Transport Systems

by

Yahnit Sirineni, Rahul Tallamraju, Abhay Rawat, Kamalakar Karlapalem

in

IEEE International Symposium on Safety, Security, and Rescue Robotics (IEEE SSRR 2020)
: 1
-8

Report No: IIIT/TR/2020/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2020

Decentralized Collision Avoidance and Motion Planning for Multi-Robot Deformable Payload Transport Systems

Yahnit Sirineni, Rahul Tallamraju, Abhay Rawat, Kamalakar Karlapalem

Abstract— **We present a decentralized motion planning and collision avoidance algorithm for multi-robot payload transport systems (PTS). A PTS is a formation of loosely coupled non-holonomic robots that cooperatively transport a deformable payload. Each PTS is constrained to navigate safely in a dynamic environment by inter-formation, environmental, and intra-formation collision avoidance. Real-time collision avoidance for such systems is challenging due to the deformability of formations and high dimensional multi-robot non-convex workspace. We resolve the above challenges by embedding workspaces defined by a multi-robot collision avoidance algorithm and multi-scale repulsive potential fields as constraints within a decentralized convex optimization problem. Specifically, we present two main steps to plan the motion of each formation. First, we compute collision-free multi-scale convex workspaces over a planning horizon using a combination of ORCA and repulsive potential fields. Subsequently, we compute the motion plans of formation over a horizon by proposing a novel formulation for collision avoidance, and we leverage a model predictive controller (MPC) to solve the problem. The results validate that our solution facilitates real-time navigation of formations and computationally scales well with an increase in the number of robots and formations used. The algorithm is validated through extensive preliminary simulations, experiments in the gazebo simulator, and a proof of concept using real robots.**

I. INTRODUCTION

In recent years, motion planning and navigation for payload transportation through dynamic environments have received increased attention [1][2][3][4]. In this paper, we present a decentralized solution for collision avoidance of deformable formations through dynamic environments. This has applications in industries and warehouses for transporting deformable payloads such as cloth, rope, pipes etc. Each formation navigates through an environment having other dynamic decision-making formations and static obstacles. Computing trajectories for such systems in real-time is challenging owing to the high dimensional configuration space of multiple formations, non-convexities in workspaces, and heterogeneity in the type of formation (number of robots, size, deformable or non-deformable, etc.). Moreover, existing solutions for navigation of formations do not consider heterogenous multi-formation collision avoidance. To overcome the above problems, we propose the following key novelties.

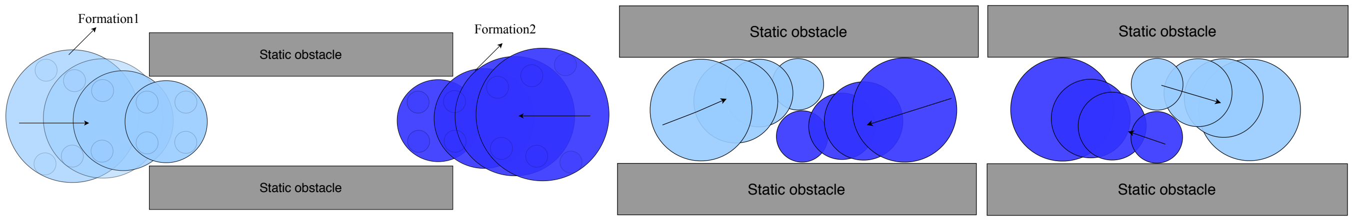
1) Each formation uses the shared trajectories from other formations to compute its deformations over a fixed planning horizon, using virtual repulsive potential fields [3]. We derive upper and lower bounds on the radii of influence of potential fields that change dynamically to allow temporal deformations of the formation over the planning horizon. Moreover, these deformations are also constrained by the payload deformation limits.
2) Constraints over the robot control velocities at each step of the planning horizon are computed using ORCA [5] considering deformations of all the formations at that corresponding step in the horizon. The computed constraints are subsequently embedded into a model-predictive controller formulation. The collision-free formation control inputs are numerically computed in real-time using convex optimization methods.
3) We additionally characterize failure scenarios for our MPC and propose methods to resolve detected failure scenarios.

In summary, we use existing approaches of (a) virtual potential fields for predictive formation collision avoidance [3], and (b) reactive collision avoidance guarantees from ORCA [5], to develop predictive collision avoidance in the velocity space of formations. We novelly embed these constraints into a model-predictive control problem to provide a comprehensive solution for multi-formation collision avoidance.

The leader agent of each formation computes its collision-free trajectories by considering motions of neighboring formations. Each leader agent receives the planned trajectories for a horizon of nearby formations. The formation is deformed over the planning horizon using repulsive potential fields calculated from the received trajectories. Subsequently, ORCA is leveraged to compute a multi-scale convex workspace over the horizon for each formation. Then, a novel formulation of decentralized model-predictive controller (MPC) constrained by the computed workspaces is developed, to solve for locally optimal collision-free trajectories in real-time. This ensures inter-formation and static obstacle collision avoidance. Finally, a decentralized leader-follower algorithm is incorporated to compute the locally optimal control inputs for the followers of the formation which also ensures intra-formation collision avoidance. Our approach scales well with the number of formations as it is decentralized. We validate our algorithm through extensive simulations and experiments in the Gazebo simulator.

We discuss the related work in Section II. The notations and preliminaries are briefed in Section III. The presented approach for motion planning and collision avoidance of multi-robot payload transport systems is discussed in Section IV. The simulation results are presented in Section V.

Authors are with the Agents and Applied Robotics Group (AARG), Kohli Center on Intelligent Systems, International Institute of Information Technology, Hyderabad (IIIT-H), India sirineni.yahnit@research.iiit.ac.in, kamal@.iiit.ac.in



(a) formations of robots deform to enter a narrow corridor

(b) formations deform even more to cross each other

(c) formations start expanding after they crossed each other

Fig. 1: Configuration of formations over the PF-FORCA-MPC planning horizon at (a) t = T_1, (b) t = T_2, (c) t = T_3

Finally, we conclude in Section VI.

II. Related Work

Sequential convex programming [6] was utilized to identify collision-free formation configurations for a single formation navigating through a dynamic environment. A distributed extension of this work was introduced in [7]. In contrast, we propose a convex model-predictive controller for formation navigation through an environment having other dynamic decision-making agents and formations. In [3], repulsive potential fields and a convex model-predictive controller were leveraged to ensure obstacle-avoidance for cooperative spatial manipulation of a rigid payload. In our current work, we do not make assumptions on the type of payload transported and solve a formation-level navigation problem for robots carrying both deformable and rigid payloads. Moreover, the paper i) does not address multi-formation collision avoidance, ii) deals only with rigid payloads, and iii) does not use ORCA, and only uses potential fields for collision avoidance. Thereby their approach does not guarantee system safety.

The paper [8] utilized virtual force based communication and velocity obstacles to plan the collision-free motion for a single deformable payload and multiple robots. However, the approach does not use MPC or plan the motion of the formation through other deformable formations. In our previous work [9], we proposed a multi-formation algorithm for inter-formation and static obstacle avoidance. However, the scope of the work is limited to rigid payloads and moreover, it does not use predictive planning for formation navigation.

Model-predictive control for ORCA was introduced in [10] to perform predictive collision avoidance. However, the paper considers only single agents and does not deal with payload transportation or formation navigation constraints. Moreover, the approach does not (a) identify cases where ORCA fails to provide a regional constraint, (b) use a global path planner to handle oscillations in ORCA planning, (c) consider non-holonomic agents/robots.

III. Background

A. Payload Transport Systems

A Payload transport system (PTS) is a formation of loosely coupled non-holonomic robots that cooperatively transport a deformable payload from one place to another. Each PTS has a leader and a set of followers that maintain a desired distance relative to the leader. Each PTS can carry payloads of various sizes and can have different number of robots in the formation. The terms *PTS* and *formation* are used interchangeably in the paper.

Each formation F_i is defined by the number of followers, maximum speed $v_i^{max} \in \mathbb{R}$ which denotes the maximum speed of the leader, maximum radius r_i^{max} which indicates the radius of the payload in undeformed state and minimum radius r_i^{min} which is defined by the limits of deformation of payload. r_i^{max} and r_i^{min} are equal for rigid payloads. We conservatively approximate all the formation shapes to be circular. The sampling time is Δt and at each timestep t, each formation F_i has a position $\mathbf{p}_i(t) \in \mathbb{R}^2$ which denotes the position of the leader, radius $r_i(t) \in \mathbb{R}^+$ which denotes the radius of the formation, velocity $\mathbf{v}_i(t) \in \mathbb{R}^2$ which denotes the velocity, $\mathbf{p}_i^{pref}(t) \in \mathbb{R}^2$ which denotes the next immediate destination, $\mathbf{v}_i^{pref}(t) \in \mathbb{R}^2$ which denotes the preferred velocity i.e the velocity with which it would move if there are no obstacles in the surroundings. Ideally, it is a vector pointing towards the $\mathbf{p}_i^{pref}(t)$ from $\mathbf{p}_i(t)$ with a magnitude v_i^{max} and is given by (1).

$$\mathbf{v}_i^{pref}(t) = v_i^{max} \frac{\mathbf{p}_i^{pref}(t) - \mathbf{p}_i(t)}{\|\mathbf{p}_i^{pref}(t) - \mathbf{p}_i(t)\|} \tag{1}$$

B. Leader-Follower Formation Control

A decentralized bioinspired neurodynamic based leader-follower algorithm presented in [11] is incorporated in this work for formation control where the followers compute their control inputs using the command velocities and odometery of the leader such that they maintain a desired distance and angle w.r.t the leader.

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

nh-ORCA [12] is a robust decentralized collision avoidance algorithm for non-holonomic robots built from the concepts of ORCA [5], which deals with holonomic robots. Each robot i constructs $ORCA_{i|j}^{\tau}$ $\forall$ $j \neq i$ and is given by (2),

$$ORCA_{i|j}^{\tau} = \left\{ \mathbf{v}_{H_i} \mid \left(\mathbf{v}_{H_i} - \left(\mathbf{v}_i^{opt} + 0.5 * \mathbf{u} \right) \right) . \mathbf{n} \geq 0 \right\} \tag{2}$$

where $\mathbf{v}_i^{opt}$ is the current velocity, $\mathbf{u}$ is the smallest change required to the relative velocity of robots i and j to avoid collision within τ time, and n denotes the outward normal of the boundary of $VO_{i|j}$ at ($\mathbf{v}_i^{opt}$ - $\mathbf{v}_j^{opt}$) + $\mathbf{u}$ as shown in [5]. The holonomic velocity of the robot should lie in the convex region P_{AHV_i},

$$\mathbf{v}_{H_i} \in P_{AHV_i} \tag{3}$$

which is approximated from a non-convex region S_{AHV_i} which is the set of allowed holonomic velocities for which there exists a control input within the set of non-holonomic controls that guarantees at all times a tracking error lower than the pre-defined maximum tracking error.

Now, the set of collision free velocities $ORCA_i^\tau$ for robot i is given by (4).

$$ORCA_i^\tau = P_{AHV_i} \cap (\bigcap_{j \neq i} ORCA_{i|j}^\tau) \tag{4}$$

The optimal collision-free holonomic velocity $\mathbf{v}_{H_i}$ of the robot is given by (5).

$$\mathbf{v}_{H_i}^* = \underset{\mathbf{v}_{H_i} \in ORCA_i^\tau}{\arg\min} \parallel \mathbf{v}_{H_i} - \mathbf{v}_i^{pref} \parallel \tag{5}$$

In dense cases, the linear program might be infeasible in which case a 3D Linear program is solved to compute the velocity and is given by (6),

$$\mathbf{v}_{H_i}^* = \underset{\mathbf{v} \in D(0, v_{H_i}^{max})}{\arg\min} \ max(d_{i|j}(\mathbf{v})) \tag{6}$$

where $d_{i|j}(\mathbf{v})$ is the signed distance of velocity $\mathbf{v}$ to the edge of the half plane $ORCA_{i|j}^\tau$. Finally, the holonomic velocity $\mathbf{v}_{H_i}^*$ is mapped to non-holonomic control inputs (u_i, ω_i) as shown in [12].

IV. Our Approach

Let there be N Payload Transport Systems (PTS) and Q static obstacles in the environment. The objective of each PTS is to i) ensure that it reaches its desired destination with minimum possible deviation from the desired trajectory, ii) avoid collisions with other PTS, and static obstacles, iii) navigate through narrow spaces while respecting the deformation constraints of the payload being transported and the formation kinematic constraints.

To address these objectives, we present a novel PotentialField-Formation-ORCA-MPC (PF-FORCA-MPC) algorithm. Each formation assumes that the other formations use the same algorithm to plan their motions, and the main steps for each formation $n \in \{1 \dots N\}$ are as follows.

1) Global path planning to reach the desired destination (Sec. IV-A).
2) Compute the net repulsive potential field vector over the horizon based on positions and trajectories of static obstacles and other formations (Sec. IV-B).
3) Deform the formation over the horizon in proportion to the repulsive potential field magnitude (Sec. IV-C).
4) Compute ORCA constraints over the horizon using the deformed formation sizes to avoid collisions with the obstacles and other formations (Sec. IV-D).
5) Solve the proposed MPC problem to compute the locally optimal control input of the leader of the formation (Sec. IV-E).
6) Compute the command velocities of the followers using a Leader-Follower control algorithm (Sec. IV-F).

Fig. (**??**) illustrates how two formations navigate through a narrow corridor using our proposed algorithm.

A. Global Path Planning

We compute a collision-free path for the formation F_i from its respective source to destination using a global path planning algorithm (RRT* [13]) and interpolate it to get multiple waypoints. Our approach is agnostic to the type of global path planner used. The $\mathbf{p}_i^{pref}(0)$ is the first point of the interpolated path, $v_i^{pref}(0)$ is computed as shown in (1). The $\mathbf{p}_i^{pref}(t)$ is updated to the next point of the interpolated path when the distance between the formation position $\mathbf{p}_i(t)$ and $\mathbf{p}_i^{pref}(t)$ is less than a constant threshold δ.

B. Artificial Repulsive Potential Fields

We modified the approach presented in [3] i) to handle both deformable and rigid payloads unlike the paper which only considers rigid payloads, ii) to use ORCA in combination with potential fields for collision avoidance, iii) and to propose novel upper and lower bounds on radii of influence of potential fields which dynamically change over the horizon.

We compute artificial repulsive potential field vectors over the time horizon. The magnitude of the vector increases hyperbolically with a decrease in distance to obstacles and other formations in the vicinity, as shown in [3] . We use this vector to deform the formation over the horizon and navigate it through narrow spaces. The repulsive potential P_i^j on Formation F_i w.r.t obstacle j is defined as follows.

$$P_i^j(d_i^j) = \begin{cases} P_i^{max} & \text{if } \ d_i^j < l_i \\ \frac{\pi}{2}\left(\frac{y + cot(y) - \pi/2}{u_i - l_i}\right) & \text{if } \ l_i \leq d_i^j < u_i \\ 0, & \text{if } \ d_i^j \geq u_i \end{cases} . \tag{7}$$

Here, $y = \frac{\pi}{2}\left(\frac{d_i^j - l_i}{u_i - l_i}\right)$, d_i^j is the distance between Formation F_i and obstacle j, l_i and u_i define the minimum and maximum radii of influence of the potential field. These radii are computed as follows.

$$l_i(k) = r_i^{min} + r_j^{min} \tag{8}$$

$$u_i(k) = r_i^{max} + r_j^{max} + \alpha \frac{\|\mathbf{v}_i(k) - \mathbf{v}_i^{pref}(k)\|}{\|\mathbf{v}_i^{pref}(k)\|} \tag{9}$$

Here, r_i^{min} indicates the radius of formation at its maximum possible deformation, and r_i^{max} indicates the radius of formation in the undeformed state, $k \in [1, H]$ represents a time-step in the planning horizon. The last term in (9) defines normalized velocity deviation from preferred velocity.

When the normalized velocity deviation is high, it implies that the formations are nearly static and have deviated from their preferred velocities due to the presence of other formations and obstacles. This leads to an increase in potential field region of influence, which leads to formation deformation. The deformation over the horizon squeezes the robot workspace and enables the planner to look ahead for feasible trajectories. However, deviation in velocity from preferred velocity does not always mean it has to deform in order to move ahead. Hence, the constant α is specific to the density of formations and obstacles in the environment. We set the value of α as r_i^{max} in the simulations, which implies that the radius of influence of the potential field is increased by r_i^{max} when the formation is static. This is one of the key novelties of our work which enables PF-FORCA-MPC to deform formations predictively in real time.

The repulsive potential field $\mathbf{f}_i^d(k)$ due to the influence of other formations F_j is given by,

$$\mathbf{f}_i^d(k) = \sum_j^N P_i^{F_j}(d_i^j(k))\, \beta_i^{F_j}(k) \ , \tag{10}$$

where $d_i^j(k) = \|\mathbf{p}_i(k) - \mathbf{p}_j(k)\|_2$ and $\beta_i^{F_j}(k) = \frac{\mathbf{p}_i(k)-\mathbf{p}_j(k)}{\|\mathbf{p}_i(k)-\mathbf{p}_j(k)\|_2} \forall k \in [1,H]$ is a unit vector pointing away from the formation. The repulsive potential field due to the Q static obstacles is given by,

$$\mathbf{f}_i^s(k) = \sum_j^Q P_i^{S_j}(d_i^j(k))\, \beta_i^{S_j}(k) \ , \tag{11}$$

where $d_i^j(k) = \|\mathbf{p}_i(k) - \mathbf{p}_j(k)\|_2$ and $\beta_i^{S_j}(k) = \frac{\mathbf{p}_i(k)-\mathbf{p}_j(k)}{\|\mathbf{p}_i(k)-\mathbf{p}_j(k)\|_2} \forall k \in [1,H]$. Note that static obstacles are considered to be circular in shape. However, any arbitrary polygon can be represented by a collection of circles. The total repulsive potential field $\mathbf{f}_i^{total}$ is given by,

$$\mathbf{f}_i^{total}(k) = \mathbf{f}_i^d(k) + \mathbf{f}_i^s(k) \tag{12}$$

The potential field is clamped to a positive value P_i^{max} which denotes the maximum possible deformation in the radius of a formation in a single timestep. This is specific to the payload being carried. P_i^{max} for formations carrying rigid payloads is zero.

$$\mathbf{f}_i^r(k) = \begin{cases} \mathbf{f}_i^{total}(k) & \text{if } \ \|\mathbf{f}_i^{total}(k)\| < P_i^{max} \\ P_i^{max} \frac{\mathbf{f}_i^{total}(k)}{\|\mathbf{f}_i^{total}(k)\|} & \text{if } \ \|\mathbf{f}_i^{total}(k)\| \geq P_i^{max} \end{cases} \tag{13}$$

Note that the potential field $\mathbf{f}_i^r$ for formations carrying rigid payloads is always zero. As a result, the radius of the formation always remains constant and the formation does not deform whatsoever. Hence, we can conclude that our algorithm works for rigid payloads as well.

C. Computing Radius over the horizon

The formation is deformed by reducing the radius r_i over the horizon to reduce the region of influence of the potential field, which in turn enables the formation to navigate through narrow and tight spaces. The deformation in radius is applied over the horizon. The deformation rate is directly proportional to the repulsive potential field $\|\mathbf{f}_i^r(k)\|$. Since it is essential for the payload to get back to its original size, we include an expansion potential field $f_i^e(k)$, which increases the radius. The radius of the formation F_i at any step k over the horizon H is given as,

$$r_i(k+1) = r_i(k) - \|\mathbf{f}_i^r(k)\| + \gamma f_i^e(k) \tag{14}$$

where γ is a positive constant. $\gamma < 1$ as deformation is of higher priority than expansion, and we want the formation to expand only when the repulsive potential field is negligible. l_i and u_i for computing $f_i^e(k)$ are r_i^{min} and r_i^{max} respectively. Note that $\mathbf{f}_i^r$ and f_i^e for formations carrying rigid payloads are zero.

D. Computing ORCA constraints over the horizon

Each formation F_i computes $ORCA_{i|j}^{\tau}(k)$ as shown in Section III-C $\forall j$, $\forall k \in [1,H]$ using the received predicted states and control inputs for the horizon from other formations.

$ORCA_{i|j}^{\tau}(k)$ refers to the collision avoidance constraints on the control input $v_i(k)$ for the k^{th} time step of the planning horizon. Since we use repulsive potential fields over the planning horizon, the formation geometries potentially have different radii $r_i(k)$ over this horizon. Note that $ORCA_{i|j}^{\tau}(k)$ is a function of radii $r_i(k)$ and $r_j(k)$ which enables real-time formation deformations for collision avoidance.

Hence, each formation F_i requires the radius of all the other formations to compute $ORCA_{i|j}^{\tau}(k)$ at each horizon step. The two possible solutions to this are,

1) The formations explicitly communicate the radius over the horizon to other formations.
2) The formation computes the radius over the horizon of all the other formations using the communicated trajectories.

Based on the communication and computation constraints, we can choose either of the solutions. For simulations presented in the paper, we chose the first solution.

E. Model Predictive Controller (MPC)

For the formation to navigate towards the destination at preferred velocity, while avoiding collisions with other formations, we present a novel MPC formulation to compute the leader's locally optimal control input. The MPC is constrained by (a) linear translation dynamics, (b) inter-formation collision avoidance, (c) environmental collision avoidance, (d) position bounds and, (e) bounds on control inputs. We discuss the cost function and formulation of the MPC in the following subsections.

1) Cost Function: At time t, the desired position is given by $\mathbf{p}_i^{pref}(t)$ and the desired velocity is given by $\mathbf{v}_i^{pref}(t)$. The cost function for any step $k \in [1,H]$ of the horizon at time t is as given below.

$$\begin{aligned} C_i(k) = ((&\mathbf{v}_i(k) - \mathbf{v}_i^{pref}(k))\Omega_v(\mathbf{v}_i(k) - \mathbf{v}_i^{pref}(k))^\top + \\ &(\mathbf{p}_i(k+1) - \mathbf{p}_i^{pref}(k+1))\Omega_p(\mathbf{p}_i(k+1) - \mathbf{p}_i^{pref}(k+1))^\top) \end{aligned} \tag{15}$$

In the MPC, we minimise the deviation of velocity and position w.r.t the preferred velocity $\mathbf{v}_i^{pref}(t)$ and preferred position $\mathbf{p}_i^{pref}(t)$.

2) MPC Formulation:

$$\mathbf{p}_i^*(2)\ldots\mathbf{p}_i^*(H+1), \mathbf{v}_i^*(1)\ldots\mathbf{v}_i^*(H) = \underset{\mathbf{v}_i(1)\ldots\mathbf{v}_i(H)}{\mathbf{arg\,min}} \sum_{k=1}^{H} C_i(k) \tag{16}$$

such that $\forall\ k$,

$$\mathbf{p}_i(k+1)^\top = \mathbf{A}\mathbf{p}_i(k)^\top + \mathbf{B}\mathbf{v}_i(k)^\top, \tag{17}$$

$$\mathbf{v}_i(k) \in \bigcap_{j\neq i} ORCA^{\tau}_{i|j}(k) \tag{18}$$

$$\mathbf{v}_i(k) \in \mathbf{P}_{AHV_i}, \tag{19}$$

$$\mathbf{p}_i^{\min} \leq\ \mathbf{p}_i(k+1) \leq\ \mathbf{p}_i^{\max}. \tag{20}$$

The motion planning adheres to the following constraints.

1) Linear translational dynamics of the leader of formation F_i given by (17),
2) collision avoidance constraints given by (18).
3) velocity bounds on $\mathbf{v}_i$ as shown in (3).
4) position bounds on $\mathbf{p}_i$.

The dynamics matrix is given by $A = I_{2\times2}$, and the control transfer matrix is given by $B = \Delta t I_{2\times2}$ where $I_{2\times2}$ is an identity matrix, and Δt is the sampling time. The output of the MPC is the trajectory over the horizon $[\mathbf{p}_i^*(2)\cdots\mathbf{p}_i^*(H+1)]$ and the locally optimal control inputs $[\mathbf{v}_i^*(1)\cdots\mathbf{v}_i^*(H)]$ over the horizon. Note that regions in (18) and (19) are convex making it a convex optimization problem. The formations subsequently communicate with each other to transmit their trajectory over the horizon. The planning horizon H is chosen such that $H.\Delta t < \tau$ as given in [10]. ORCA implicitly assumes that the robots move with uniform velocity in the next τ time. We additionally optimize the velocity over the ORCA time horizon τ in with MPC.

F. Computing Robot Control Inputs

Now that we have computed the locally optimal holonomic control input $\mathbf{v}_i^*(1)$ for the leader of the Formation F_i, we map this to the non-holonomic control inputs (u_i, ω_i) as illustrated in [12]. We then compute and apply the control inputs to the followers of the formation using a decentralized Leader-Follower algorithm presented in [11] where the desired distance of each follower w.r.t leader is the radius of formation $r_i(1)$ computed from (14). Finally, $\mathbf{p}_i^{pref}$ and $\mathbf{v}_i^{pref}$ are updated based on the resulting state.

G. Failure Detection and Resolution

One of the key insights of our work is the detection and resolution of failures of the model predictive controller.

1) MPC fails: In cluttered and complex environments with a high number of formations and static obstacles in the formation's vicinity, the MPC fails to find a feasible solution due to small formation workspaces enforced by ORCA constraints over the horizon. There are two possible cases which can lead to this.

Case 1 : *The half-planes corresponding to the ORCA lines of the first step of the horizon, i.e., k = 1, do not have a common region.*

In this case, we solve a 3-D linear program (LP) to compute the control input only for the present time step. We find the safest possible velocity, which deviates the least from the ORCA constraints. We choose a velocity that minimizes the maximum distance to any of the half-planes induced by other formations, as given in [5]. The LP formulation is as follows,

$$\mathbf{v}_i^*(t), d_i^* = \underset{\mathbf{v}_i(t)\in\mathbf{P}_{AHV_i}}{\mathbf{arg\,min}}\ \mathbf{max}\ d_{i|j}(\mathbf{v}_i(t))\ \forall j \tag{21}$$

where $d_{i|j}(\mathbf{v}_i(t))$ is the signed distance of velocity $\mathbf{v}_i(t)$ to the edge of the half plane $ORCA^{\tau}_{i|j}(1)$. Note that the ORCA constraints induced by static obstacles are not relaxed.

Case 2 : *The half-planes corresponding to the ORCA lines of the first step of the horizon, i.e., k = 1, have a common region, but MPC failed due to tight ORCA constraints at some step $k > 1$.*

In this case, we solve a Quadratic Program (QP) to compute the control input only for the current timestep while minimizing a cost function. The LP formulation is as follows.

$$\mathbf{v}_i^*(t), \mathbf{p}_i^*(t+1) = \underset{\mathbf{v}_i(t)}{\mathbf{arg\,min}}\ ((\mathbf{v}_i(t)-\mathbf{v}_i^{pref}(t))\Omega_v(\mathbf{v}_i(t)-\mathbf{v}_i^{pref}(t))^\top + (\mathbf{p}_i(t+1)-\mathbf{p}_i^{pref}(t+1))\Omega_p(\mathbf{p}_i(t+1)-\mathbf{p}_i^{pref}(t+1))^\top) \tag{22}$$

subject to,

$$(\mathbf{p}_i(t+1))^\top = (\mathbf{A}\mathbf{p}_i(t))^\top + (\mathbf{B}\mathbf{v}_i(t))^\top, \tag{23}$$

$$\mathbf{v}_i(t) \in \bigcap_{j\neq i} ORCA^{\tau}_{i|j}(1) \tag{24}$$

$$\mathbf{v}_i(t) \in \mathbf{P}_{AHV_i}, \tag{25}$$

$$\mathbf{p}_i^{\min} \leq\ \mathbf{p}_i(t+1) \leq\ \mathbf{p}_i^{\max}. \tag{26}$$

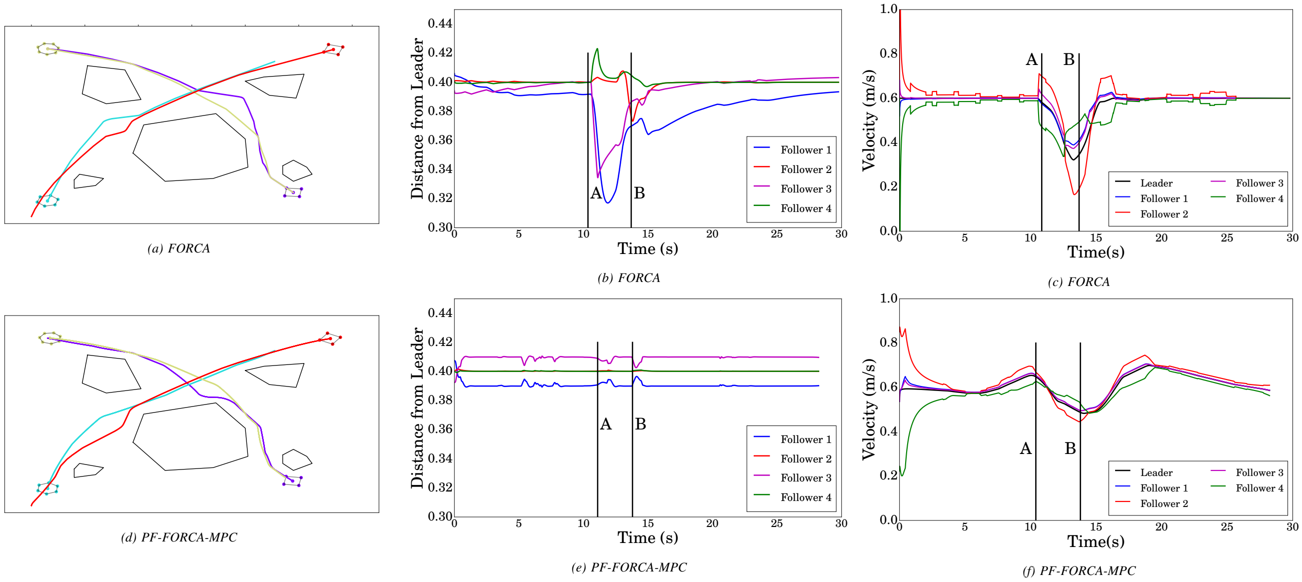
Now, once we compute the control input for the current timestep, we compute the predicted states over the horizon by assuming that the formation traverses with uniform velocity throughout the horizon so that we could use these predicted states to compute ORCA constraints for future timesteps.

V. Results

In this section, we discuss the results of the presented PF-FORCA-MPC algorithm. The MPC is solved using the operator splitting quadratic program (OSQP) solver, which runs the ADMM algorithm. The proposed algorithm is implemented in Python and runs on the Intel Core i5-5250U processor. The algorithm is validated using several simulation environments varying in i) the no. of formations, ii) no. of static obstacles, iii) size and configuration of formations, iv) density and complexity. We first compare the PF-FORCA-MPC algorithm presented in this paper with the Leader-Follower-ORCA-RRT* (or FORCA) algorithm presented in [9], then discuss a narrow corridor environment and antipodal simulation. The values of the parameters α, γ, Δ and P^{max} used in the algorithm are r_i^{max}, 0.1, 0.1(s) and 0.01 respectively.

A. Comparison of FORCA with PF-FORCA-MPC

In this section, we compare the performances of the FORCA algorithm proposed in [9] with the PF-FORCA-MPC algorithm proposed in this paper. We consider an environment used in [9] with four formations and five static obstacles where the formations navigate to reach their respective destinations while avoiding collisions with obstacles and other formations.



(a) FORCA

(b) FORCA

(c) FORCA

(d) PF-FORCA-MPC

(e) PF-FORCA-MPC

(f) PF-FORCA-MPC

Fig. 2: Four formations with 7, 4, 6, 4 followers respectively. (a) Motion Trajectories with FORCA, (b) leader-follower distance with FORCA for Red formation, (c) Velocities of robots with FORCA for Red formation, (d) Motion Trajectories with PF-FORCA-MPC, (e) leader-follower distance with PF-FORCA-MPC for Red formation, (f) Velocities of robots with PF-FORCA-MPC for Red formation

Note that our algorithm works with rigid payloads as well. In this simulation, we consider all the formations to be rigid to compare the results with the FORCA algorithm, which only deals with rigid payloads. r_i^{max} and r_i^{min} for the formations are set to 0.4m. The planning horizon was set as $H = 10$. The motion trajectories of the formations with both the algorithms are shown in Fig. (2a) and Fig. (2d) respectively.

The average radius of curvature of the paths of Red and Blue formations while crossing each other with the PF-FORCA-MPC algorithm is 5% less than that with the FORCA algorithm. This can be seen in Fig. (2a) and (2d) that the trajectory curves of Red and Blue formations while avoiding collisions with each other is smoother with the PF-FORCA-MPC algorithm. Similar is the case with Purple and Yellow formations while they avoid collisions with each other, where the average radius of curvature with PF-FORCA-MPC is 6% less than with FORCA. Lower curvature is more favorable for stable payload transport, and hence we can conclude that the PF-FORCA-MPC algorithm ensures smoother payload transport. Our solution shows the ability of formations to look ahead into the future and take their present action hence avoiding sudden jumps in their velocities and trajectories.

The leader-follower distance graph is plotted for the red formation using both the algorithms in Fig. (2b) and (2e). Points A and B are marked in these figures corresponding to the timestamps when i) the red formation deviates from its desired path to avoid collision with the blue formation, ii) the red formation returns to its desired path after crossing the blue formation. We can infer from the plot that the leader-follower distance curve in Fig. (2e) at Points A and B is much smoother and stable without sudden jumps and less deviation from its preferred distance, which is favorable for carrying payloads. In contrast, the leader-follower distance curve in Fig. (2b) has sudden jumps with a relatively more deviation from the desired distance.

The velocity graph for red formation is plotted in Fig. (2c) and (2f). It can be seen that the velocity curve between the points A and B with the PF-FORCA-MPC algorithm is smoother without sudden drops and closer to the preferred velocity which is in contrast to the velocity curve with the FORCA algorithm where the velocities drop suddenly, indicating that the followers slow down suddenly to avoid a collision.

B. Narrow corridor simulation

In this section, we illustrate a narrow corridor scenario simulated in the Gazebo simulator. We consider an environment with two deformable PTS moving towards the opposite sides of a narrow corridor of width 5m. Each PTS has four followers in its formation. r_i^{max} and r_i^{min} are 2.4m and 1.2m, respectively. The snapshots of the simulation at six different timesteps are shown in Fig. (3). It is seen that the formations successfully navigate to their destinations through the narrow corridor while deforming themselves to avoid collision with other formation and the wall.

A plot of the radius of formation $F1$ during the simulation is shown in Fig. (4a) and the leader-follower distance for the same formation is plotted in Fig. (4b). Points A, B, C are marked corresponding to i) t = 11s when the formation deviates from its desired path to avoid a collision, ii) t = 29s when the formation is closest to $F2$ and iii) t = 35s when the formation start expanding to its original size. At point A, the radius of the formation starts to decrease, as seen in Fig. (4a) in order to navigate through the corridor while avoiding collision with formation $F2$ and the wall. Consequently, the leader-follower distance of all the followers of the formation starts decreasing, as seen in Fig. (4b). The followers choose

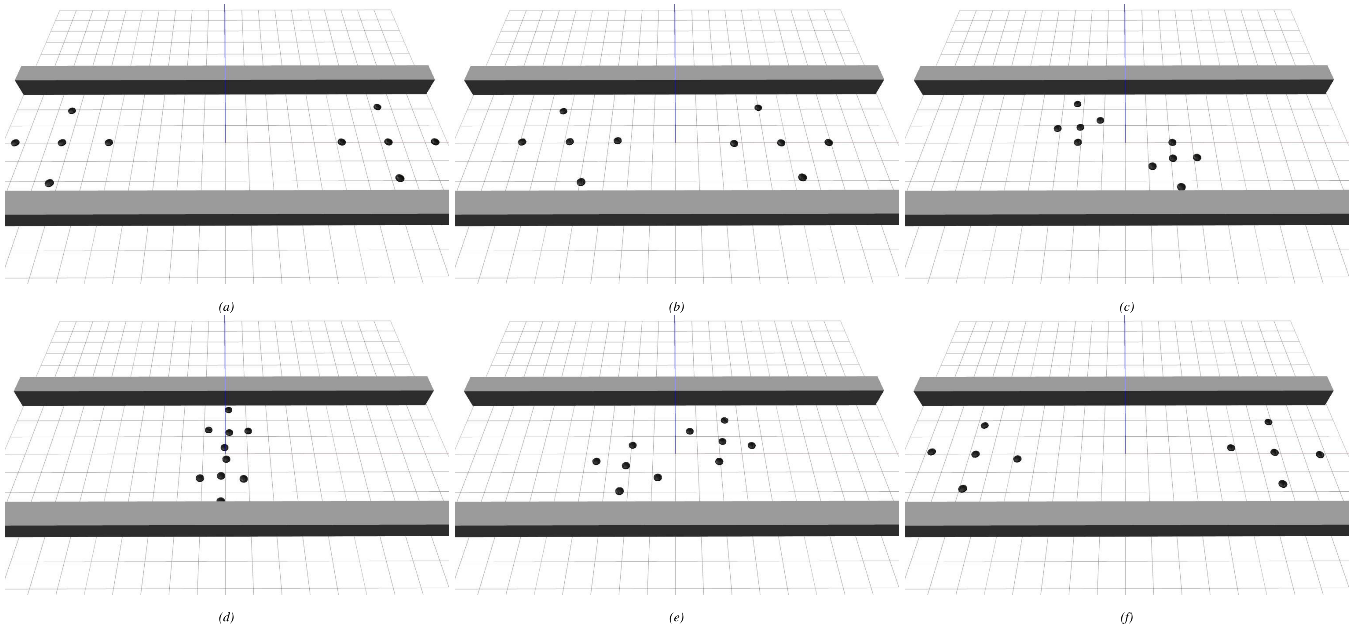


Fig. 3: *Snapshots of narrow corridor simulation in the Gazebo simulator with PF-FORCA-MPC at (a) t = 1s, (b) t = 11s, (c) t = 20s, (d) t = 29s, (e) t = 35s, (f) t = 50s. The formation coming from the left side of the corridor is F1 and the formation coming from the right side is F2.*

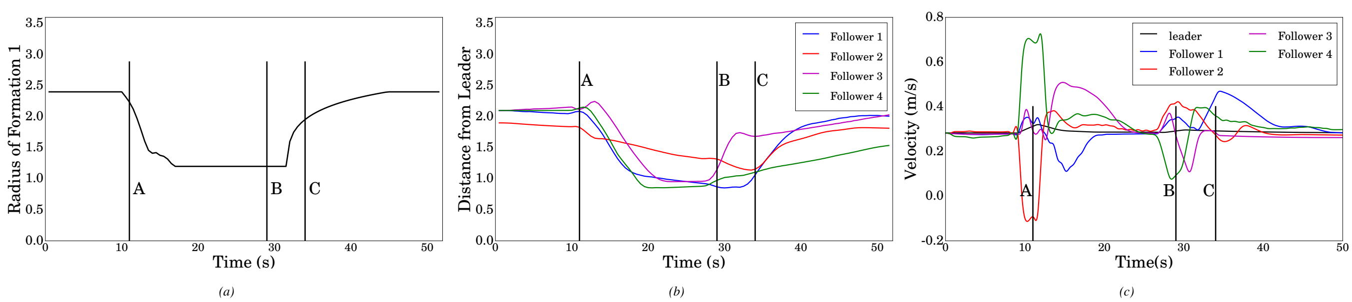


Fig. 4: *(a) Radius of formation F1, (b) Leader-Follower distance in F1, (c) Velocities of the robots in formation F1*

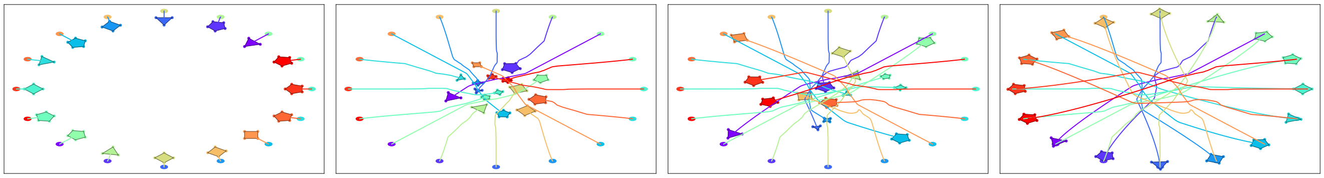
appropriate velocities such that they maintain the desired distance w.r.t the leader ensuring that payload stays intact and can be seen at point A in Fig. (4c) . At point B, when the formations are closest to each other, it can be seen from Fig. (4a), (4b) that the radius of the formation and the leader-follower distance is minimum so that both the formations can cross each other without colliding. At point C, the radius and leader-follower distance start increasing, which implies that the formation is expanding after they have crossed each other. The corresponding velocities chosen by the followers are shown in Fig. (4c).

C. Antipodal Simulation

In this simulation, we consider 16 PTS carrying payloads, each with unique deformability constraints where the r_i^{max} of each formation F_i is 0.4m and $r_i^{min} \in [0.2, 0.4]$. Each formation has a varying number of followers between 3 to 5. The formations are located uniformly on a circle, and the formations at antipodal positions are expected to swap positions with each other. The planning horizon was set as $H = 10$. The snapshots of the simulation are shown in Fig. (5), the destinations of the formations are marked with the same color for our reference. We can infer from the figure that the formations successfully reached their destinations while avoiding collisions with neighboring formations and deforming themselves in dense scenarios to reach their destinations faster.

We ran around 100 simulations and compared the time taken by PTS to reach their respective destinations when i) deformability of the payloads can be exploited, ii) deformability of the payloads is not exploited and all the formations are assumed to be rigid even though they could be deformed. It was seen that the percentage decrease in time taken for the formations to reach their destinations when deformability could be exploited using our algorithm is roughly around 10%. Hence, we can state that in environments with deformable PTS, our algorithm helps all the PTS (Rigid and Deformable) reach their destinations faster.

We have tabulated the statistics of execution time per iteration in high-density cases where there are 15 formations (roughly 60 robots) in the radius of visibility in Table I. We attain a low net average execution time t_{avg} of 0.025 per iteration. The computation time is seen to increase linearly with an increase in the planning horizon H.



(a) (b) (c) (d)

Fig. 5: Snapshots of antipodal simulation with 16 formations at (a) t = 1s, (b) t = 9s, (c) t = 13s, (d) t = 21s

(a) (b) (c) (d)

Fig. 6: Snapshots of the Real-Robot experiment at (a) t = 1s, (b) t = 10s, (c) t = 20s, (d) t = 30s

TABLE I: Statistics of Execution Time per iteration

Optimization	$t_{avg}(s)$	$t_{max}(s)$	$t_{min}(s)$	std
Potential Fields	0.002	0.004	0.0014	0.0004
ORCA	0.02	0.05	0.008	0.01
MPC and Formation	0.0025	0.008	0.0005	0.002

D. Real Robot Proof of Concept Experiment

To demonstrate the proof of concept of our algorithm in the real world, we use three custom-built non-holonomic differential drive robots to cooperatively move in a predefined geometric formation and avoid static obstacles in a narrow corridor. Each robot has an onboard Raspberry Pi and an Arduino Uno board. The formation transports a virtual deformable payload whose radius in the undeformed state is 1.4m and can deform up to a radius of 0.6m. The formation uses the proposed PF-FORCA-MPC algorithm to navigate in the environment with tight and narrow spaces. Specifically, artificial repulsive potential fields act on the formation when it approaches the narrow corridor due to static obstacles. As a result, the formation deforms itself to navigate through the narrow corridor. Snapshots of the experiment are shown in Fig. (6). This proof-of-concept showcases the feasibility of using our algorithm to perform real-time formation collision-avoidance. This behavior is also visualized schematically in Fig. (3). The video attached along with this paper showcases the complete experiment.

VI. Conclusion

A novel decentralized PotentialField-Formation-ORCA-MPC (PF-FORCA-MPC) algorithm is presented in this paper to address the motion planning and collision avoidance of multi-robot payload transport systems carrying deformable payloads. The presented approach safely navigates formations (PTS) to their desired destinations in optimal time and distance while ensuring inter-formation, intra-formation, and environmental collision avoidance in addition to formation deformation constraints. The proposed approach applies to holonomic robots as well, provided a holonomic version of formation controller is used [3][6]. The algorithm does not make any assumptions regarding the size and number of robots in the formation. Hence, it can be incorporated by any general payload transport system of arbitrary size and configuration as shown in the simulations. Being a decentralized method, the algorithm scales well with an increase in the number of robots. The efficacy of the algorithm is demonstrated through extensive python and gazebo simulations, and a proof of concept using real robots.

References

[1] J. Alonso-Mora *et al.*, "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.

[2] P. Verma *et al.*, "Loosely coupled payload transport system with robot replacement," in *ARMS with AAMAS*, 2019.

[3] R. Tallamraju *et al.*, "Motion planning for multi-mobile-manipulator payload transport systems," in *2019 CASE*, pp. 1469–1474, IEEE, 2019.

[4] S. Chand *et al.*, "Transportation of deformable payload through static and dynamic obstacles using loosely coupled nonholonomic robots," in *2019 MRS*, pp. 228–230, IEEE, 2019.

[5] J. Van Den Berg *et al.*, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[6] J. Alonso-Mora *et al.*, "Multi-robot navigation in formation via sequential convex programming," in *2015 IROS*, pp. 4634–4641, IEEE, 2015.

[7] J. Alonso-Mora *et al.*, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *2016 ICRA*, pp. 5356–5363, IEEE, 2016.

[8] J. Alonso-Mora *et al.*, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *2015 ICRA*, pp. 5495–5502, IEEE, 2015.

[9] Y. Sirineni *et al.*, "Traffic management strategies for multi-robotic rigid payload transport systems," in *2019 MRS*, pp. 225–227, IEEE, 2019.

[10] H. Cheng *et al.*, "Decentralized navigation of multiple agents based on orca and model predictive control," in *2017 IROS*, pp. 3446–3451, IEEE, 2017.

[11] Z. Peng *et al.*, "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.

[12] J. Alonso-Mora *et al.*, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed autonomous robotic systems*, pp. 203–216, Springer, 2013.

[13] F. Islam *et al.*, "Rrt-smart: Rapid convergence implementation of rrt towards optimal solution," in *2012 IEEE ICMA*, pp. 1651–1656, IEEE, 2012.