Decentralized MPC based Obstacle Avoidance for Multi-Robot Target Tracking Scenarios

Rahul Tallamraju[1,3], Sujit Rajappa[2], Michael Black[1], Kamalakar Karlapalem[3] and Aamir Ahmad[1]

Abstract— In this work, we consider the problem of decentralized multi-robot target tracking and obstacle avoidance in dynamic environments. Each robot executes a local motion planning algorithm which is based on model predictive control (MPC). The planner is designed as a quadratic program, subject to constraints on robot dynamics and obstacle avoidance. Repulsive potential field functions are employed to avoid obstacles. The novelty of our approach lies in embedding these non-linear potential field functions as constraints within a convex optimization framework. Our method convexifies non-convex constraints and dependencies, by replacing them as pre-computed external input forces in robot dynamics. The proposed algorithm additionally incorporates different methods to avoid field local minima problems associated with using potential field functions in planning. The motion planner does not enforce predefined trajectories or any formation geometry on the robots and is a comprehensive solution for cooperative obstacle avoidance in the context of multi-robot target tracking. Video of simulation studies: `https://youtu.be/umkdm82Tt0M`

I. INTRODUCTION

The topic of multi-robot cooperative target tracking has been researched extensively in recent years [1]–[5]. Target, here refers to a movable subject of interest in the environment for e.g., human, animal or other robot. Cooperative target tracking methods focus on improving the estimated pose of a tracked target while simultaneously enhancing the localization estimates of poorly localized robots, e.g., [1], by fusing the state estimate information acquired from team mate robots. The general modules involved in decentralized multi-robot target tracking are summarized in Fig. 1. Our work focuses on the modules related to obstacle avoidance (blue in Fig. 1). The other related modules (green in Fig. 1), such as target pose estimation, are assumed to be available (see [6]). The developed obstacle avoidance module fits into any general cooperative target tracking framework as seen in Fig. 1.

Robots involved in tracking a desired target must not collide with each other, and also with other entities (human, robot or environment). While addressing this problem, the state-of-art methodologies for obstacle avoidance in the context of cooperative target tracking have drawbacks. In [7], [8] obstacle avoidance is imposed as part of the weighted MPC based optimization objective, thereby providing no guaranteed avoidance. In [1] obstacle avoidance is a separate planning module, which modifies the generated optimization trajectory using potential fields. This leads to a sub-optimal trajectory and field local minima.

rahul.tallamraju, black, aamir.ahmad@tuebingen.mpg.de, sujit.rajappa@uni-tuebingen.de, kamal@iiit.ac.in

[1] Perceiving Systems Department, Max Planck Institute for Intelligent Systems, Tübingen, Germany.

[2] The Chair of Cognitive Systems, Department of Computer Science, University of Tübingen, Tübingen, Germany.

[3] Agents and Applied Robotics Group, International Institute for Information Technology, Hyderabad, India.
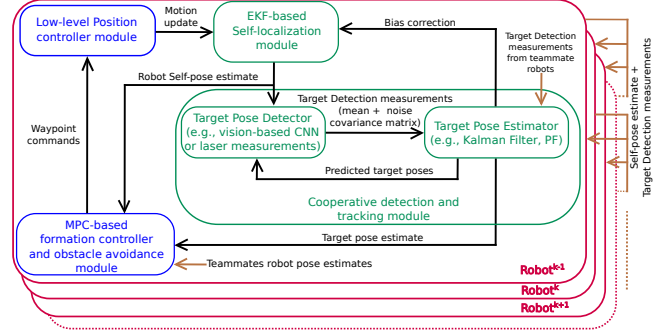
Fig. 1: General modules involved in multi-robot target tracking. Our work focuses on the modules highlighted in blue

The goal of this work is to provide a holistic solution to the problem of obstacle avoidance, in the context of multi-robot target tracking in an environment with static and dynamic obstacles. Our solution is an asynchronous and decentralized, model-predictive control based convex optimization framework. Instead of directly using repulsive potential field functions to avoid obstacle, we convexify the potential field forces by replacing them as pre-computed external input forces in robot dynamics. As long as a feasible solution exists for the optimization program, obstacle avoidance is guaranteed. In our proposed solution we present three methods to resolve the field local minima issue. This facilitates convergence to a desired surface around the target.

The main contributions of this work are,

- Fully convex optimization for local motion planning in the context of multi-robot target tracking
- Handling non-convex constraints as pre-computed input forces in robot dynamics, to enforce convexity
- Guaranteed static and dynamic obstacle avoidance
- Asynchronous, decentralized and scalable algorithm
- Methodologies for potential field local minima avoidance

Sec. II details the state-of-art methods related to obstacle avoidance, Sec. III discusses the Decentralized Quadratic Model Predictive Controller along with the proposed methodologies to solve the field local minima and control deadlock problem, Sec. IV elaborates on simulation results of different scenarios, and finally we discuss future work directions.

II. STATE-OF-THE-ART

The main goal in this work is to develop a decentralized multi-robot target tracker and collision free motion planner in obstacle (static and dynamic) prone environments. The multi-agent obstacle avoidance problem has gained a lot of attention in recent years. Single agent obstacle avoidance, motion planning and control is well studied [9], [10]. However,

the multi-agent obstacle avoidance problem is more complex due to motion planning dependencies between different agents, and the poor computational scalability associated with the non-linear nature of these dependencies. In general, collision free trajectory generation for multi-agents can be classified into, (i) reactive and, (ii) optimization based approaches. Many reactive approaches are based on the concept of velocity obstacle (VO) [11], whereas, optimization based approaches avoid obstacles by embedding collision constraints (like VO) within cost function or as hard constraints in optimization. Recently, a mixed integer quadratic program (MIQP) in the form of a centralized non-linear model predictive control (NMPC) [12] has been proposed for dynamic obstacle avoidance, where feedback linearization is coupled with a variant of the branch and bound algorithm. However, this approach suffers with agent scale-up, since increase in binary variables of MIQP has an associated exponential complexity. In general, centralized optimization approaches [13], [14] are not computationally scalable with increase in number of agents. In [15], a decentralized convex optimization for multi-robot collision-free formation planning through static obstacles is studied. This approach involves, triangular tessellation of the configuration space to convexify static obstacle avoidance constraints. Tessellated regions are used as nodes in a graph and the paths between the cells are determined to guarantee obstacle avoidance. A decentralized NMPC [16] has been proposed for pursuit evasion and static obstacle avoidance with multiple aerial vehicles. Here the optimization is constrained by non-linear robot dynamics, resulting in non-convexity and thereby affecting the real-time NMPC performance. Additionally, a potential field function is also used as part of a weighted objective function for obstacle avoidance. A similar decentralized NMPC has been proposed for the task of multiple UAVs formation flight control [17].

Sequential convex programming (SCP) has been applied to solve the problems of multi-robot collision-free trajectory generation [18], trajectory optimization and target assignment [19] and formation payload transport [20]. These methodologies principally approximate and convexify the non-convex obstacle avoidance constraints, and iteratively solve the resulting convex optimization problem until feasibility is attained. Due to this approximation, the obtained solutions are fast within a given time-horizon, albeit sub-optimal. SCPs have been very effective in generating real-time local motion plans with non-convex constraints. Recent work in multi-agent obstacle avoidance [21] builds on the concept of reciprocal velocity obstacle [22], where a local motion planner is proposed to characterize and optimally choose velocities that do not lead to a collision. The approach in [21] convexifies the velocity obstacle (VO) constraint to guarantee local obstacle avoidance.

In summary, due to non-linear dynamics constraints or obstacle avoidance dependencies, most multi-robot obstacle avoidance techniques are either, (i) centralized, (ii) non-convex, or (iii) locally optimal. Furthermore, some approaches only explore the solution space partially due to constraint linearization [21]. Additionally, current NMPC target tracking approaches using potential fields do not provide guarantees on obstacle avoidance and are linked with the field local-minima problem.

Recent reinforcement learning solutions [23], [24] require large number of training scenarios to determine a policy and also do not guarantee obstacle avoidance.

Our work generates collision-free motion plans for each agent using a convex model-predicitive quadratic program in a decentralized manner. This approach guarantees obstacle avoidance and facilitates global convergence to a target surface. To the best of our knowledge, the method of using tangential potential field functions [25] to generate different reactive swarming behaviors including obstacle avoidance, is most similar to our approach. However in [25], the field local minima is persistent in the swarming behaviors.

Unlike previous NMPC based target tracking approaches, which use potential fields in the objective, here we use potential field forces as constraints in optimization. Specifically, the non-linear potential field functions are not directly used as constraints in optimization. Instead, potential field forces are pre-computed for a horizon using the horizon motion trajectory of neighboring agents and obstacles in the vicinity. A feasible solution of the optimization program guarantees obstacle avoidance. The pre-computed values are applied as external control input forces in the optimization process thereby preserving the overall convexity.

III. Proposed Approach

A. Preliminaries

We describe the proposed framework for a multi-robot system tracking a desired target. For the concepts presented, we consider Micro Aerial Vehicles (MAVs) that hover at a pre-specified height h_{gnd}. Furthermore, we consider 2D target destination surface. However, the proposed approaches can be extended to any 3D surface. Let there be K MAVs $R_1, ..., R_K$ tracking a target x_t^P, typically a person P. Each MAV computes a desired destination position $\check{\mathbf{x}}_t^{R_k}$ in the vicinity of the target position. The pose of kth MAV in the world frame at time t is given by $\xi_t^{R_k} = [(\mathbf{x}_t^{R_k})^\top (\Theta_t^{R_k})^\top] \in \mathbb{R}^6$. Let there be M obstacles in the environment $O_1, ..., O_M$. The M obstacles include R_k's neighboring MAVs and other obstacles in the environment.

The key requirements in a multi-robot target tracking scenario are, (i) to not lose track of the moving target, and (ii) to ensure that the robots avoid other robot agents and all obstacles (static and dynamic) in their vicinity. In order to address both these objectives in an integrated approach, we formulate a formation control (FC) algorithm, as detailed in Algorithm 1. The main steps have the following functionality, (i) destination point computation depending on target movement, (ii) obstacle avoidance force generation, (iii) decentralized quadratic model predictive control (DQMPC) based planner for way point generation, and (iv) a low-level position controller.

To track the waypoints generated by the MPC based planner we use a geometric tracking controller. The controller is based on the control law proposed in [26], which has a proven global convergence, aggressive maneuvering controllability and excellent position tracking performance. Here, the rotational dynamics controller is developed directly on $SO(3)$ and thereby avoids any singularities that arise in local coordinates. Since the MAVs used in this work are under-actuated systems,

the desired attitude generated by the outer-loop translational dynamics is controlled by means of the inner-loop torques.

B. DQMPC based Formation Planning and Control

The goal of the formation control algorithm running on each MAV R_k is to

1) Hover at a pre-specified height h_{gnd}.
2) Maintain a distance d^{R_k} to the tracked target.
3) Orient at yaw, ψ^{R_k}, directly facing the tracked target.

Additionally, MAVs must adhere to the following constraints,

1) To maintain a minimum distance d_{min} from other MAVs as well as static and dynamic obstacles.
2) To ensure that MAVs respect the specified state limits.
3) To ensure that control inputs to MAVs are within the pre-specified saturation bounds.

Algorithm 1 MPC-based formation controller and obstacle avoidance by MAV R_k with inputs $\{\mathbf{x}_t^P,\ \mathbf{x}_t^{O_j}; j = 1 : M\}$

1: $\{\check{\mathbf{x}}_t^{R_k}\} \leftarrow$ Compute Destination Position $\{\psi_t^{R_k}, \mathbf{x}_t^{R_k}, d^{R_k}, h_{gnd}\}$
2: $\left[\mathbf{f}_t^{R_k}(0), \ldots, \mathbf{f}_t^{R_k}(N)\right] \leftarrow$ Obstacle Force $\{\mathbf{x}_t^{R_k}, \mathbf{x}_t^{O_j}(1 : N+1), \forall j\}$
3: $\{\mathbf{x}_t^{R_k*}, \dot{\mathbf{x}}_t^{R_k*}, \nabla J_{DQMPC}\} \leftarrow$ DQMPC$\{\check{\mathbf{x}}_t^{R_k}, \mathbf{x}_t^{R_k}, \mathbf{f}_t^{R_k}(0 : N), \mathbf{g}\}$
4: $\{\psi_{t+1}^{R_k}\} \leftarrow$ Compute Desired Yaw $\{\mathbf{x}_t^{R_k}, \|\nabla J_{DQMPC}\|\}$
5: **Transmit** $\mathbf{x}_t^{R_k*}(N+1), \dot{\mathbf{x}}_t^{R_k*}(N+1), \psi_{t+1}^{R_k}$ to Low-level Controller

---

Algorithm 1 outlines the strategy used by each MAV R_k at every discrete time instant t. In line 1, MAV R_k computes its desired position $\check{\mathbf{x}}_t^{R_k}$ on the desired surface using simple trigonometry. For example, if the desired surface is a circle, centered around the target location $\mathbf{x}_t^P$ with a radius $d^{R_k} = constant\ \forall R_k$, then the desired position for time instant t is given by $\check{\mathbf{x}}_t^{R_k} = \mathbf{x}_t^P + \left[d^{R_k}cos(\psi_t^{R_k})\ \ d^{R_k}sin(\psi_t^{R_k})\ \ h_{gnd}\right]^\top$. Here $\psi_t^{R_k}$ is the yaw of R_k w.r.t. the target. It is important to note that the distance d^{R_k} is an input to the DQMPC and is not necessarily the same for each MAV.

In line 2, an input potential field force vector $\left[\mathbf{f}_t^{R_k}(0), \ldots, \mathbf{f}_t^{R_k}(N)\right]^\top \in \mathbb{R}^{3 \times (N+1)}$ is computed for a planning horizon of $(N+1)$ discrete time steps by using the shared trajectories from other MAVs and positions of obstacles in the vicinity. If no trajectory information is available, the instantaneous position based potential field force value is used for the entire horizon. Section III-C details the numerical computation of these field force vectors.

In line 3, an MPC based planner solves a convex optimization problem (DQMPC) for a planning horizon of $(N+1)$ discrete time steps. We consider nominal accelerations $\left[\mathbf{u}_t^{R_k}(0) \cdots \mathbf{u}_t^{R_k}(N)\right]^\top \in \mathbb{R}^{3 \times (N+1)}$ as control inputs to DQMPC. The accelerations describe the 3D translational motion of R_k.

$$\mathbf{u}_t^{R_k}(n) = \ddot{\mathbf{x}}_t^{R_k}(n) \tag{1}$$

where n is the current horizon step. The state vector of the discrete-time DQMPC consists of R_k's position $\mathbf{x}_t^{R_k}(n) \in \mathbb{R}^3$ and velocity $\dot{\mathbf{x}}_t^{R_k}(n) \in \mathbb{R}^3$. The optimization objective is,

$$J_{DQMPC} = \Big(\sum_{n=0}^N (\mathbf{\Omega}_i(\mathbf{u}_t^{R_k}(n) + \mathbf{f}_t^{R_k}(n) + \mathbf{g})^2) +$$
$$\mathbf{\Omega}_t\big(\left[\mathbf{x}_t^{R_k}(N+1)^\top\ \dot{\mathbf{x}}_t^{R_k}(N+1)^\top\right] - \left[(\check{\mathbf{x}}_t^{R_k})^\top\ \mathbf{0}^\top\right]\big)^2\Big) \tag{2}$$

The optimization is defined by the following equations.

$$\mathbf{x}(1)_t^{R_k*} \ldots \mathbf{x}(N+1)_t^{R_k*}, \mathbf{u}_t^{R_k*}(0) \ldots \mathbf{u}_t^{R_k*}(N) = \underset{\mathbf{u}_t^{R_k}(0) \ldots \mathbf{u}_t^{R_k}(N)}{\textbf{arg min}}\ (J_{DQMPC}) \tag{3}$$

subject to,

$$\left[\mathbf{x}_t^{R_k}(n+1)^\top\ \dot{\mathbf{x}}_t^{R_k}(n+1)^\top\right]^\top =$$
$$\mathbf{A}\left[\mathbf{x}_t^{R_k}(n)^\top\ \dot{\mathbf{x}}_t^{R_k}(n)^\top\right]^\top + \mathbf{B}(\mathbf{u}_t^{R_k}(n) + \mathbf{f}_t^{R_k}(n) + \mathbf{g}), \tag{4}$$

$$\mathbf{u}_{min} \leq\ \mathbf{u}_t^{R_k}(n) \leq \mathbf{u}_{max}, \tag{5}$$

$$\mathbf{x}_{min} \leq\ \mathbf{x}_t^{R_k}(n) \leq\ \mathbf{x}_{max}, \tag{6}$$

$$\dot{\mathbf{x}}_{min} \leq\ \dot{\mathbf{x}}_t^{R_k}(n) \leq\ \dot{\mathbf{x}}_{max} \tag{7}$$

where, $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_t$ are positive definite weight matrices for input cost and terminal state (computed desired position $\check{\mathbf{x}}_t^{R_k}$ and desired velocity $\dot{\check{\mathbf{x}}}_t^{R_k} = 0$) respectively, $\mathbf{f}_t^{R_k}(n)$ is the pre-computed external obstacle force, $\mathbf{g}$ is the constant gravity vector. The discrete-time state-space evolution of the robot is given by (4). The dynamics ($\mathbf{A} \in \mathbb{R}^{3 \times 3}$) and control transfer ($\mathbf{B} \in \mathbb{R}^{3 \times 3}$) matrices are given by,

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & \Delta t\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I}_3 \\ \Delta t\mathbf{I}_3 \end{bmatrix}. \tag{8}$$

where, Δt is the sampling time. The quadratic program generates optimal control inputs $\left[\mathbf{u}_t^{R_k}(0) \cdots \mathbf{u}_t^{R_k}(N)\right]$ and the corresponding trajectory $\left[\mathbf{x}_t^{R_k}(1)\ \dot{\mathbf{x}}_t^{R_k}(1) \cdots \mathbf{x}_t^{R_k}(N+1)\ \dot{\mathbf{x}}_t^{R_k}(N+1)\right]$ towards the desired position. The final predicted position and velocity of the horizon $\mathbf{x}_t^{R_k*}(N+1), \dot{\mathbf{x}}_t^{R_k*}(N+1)$ is used as desired input to the low-level flight controller. The MPC based planner avoids obstacles (static and dynamic) through pre-computed horizon potential force $\mathbf{f}_t^{R_k}(n)$. This force is applied as an external control input component to the state-space evolution equation, thereby, preserving the optimization convexity. Previous methods in literature consider non-linear potential functions within the MPC formulation, thereby, making optimization non-convex and computationally expensive. In the next step (line 4 of Algorithm 1), the desired yaw is computed as $\psi_{t+1}^{R_k} = atan2\big(\frac{y_t^P - y_t^{R_k}}{x_t^P - x_t^{R_k}}\big)$. This describes the angle with respect to the target position $\mathbf{x}_t^P$ from the MAV's current position $\mathbf{x}_t^{R_k}$. The way-point commands consisting of the position and desired yaw angle are sent to the low-level flight position controller. Although no specific robot formation geometry is enforced, the DQMPC naturally results in a dynamic formation depending on desired destination surface.

C. Handling Non-Convex Collision Avoidance Constraints

In our approach, at any given point there are two forces acting on each robot, namely (i) the attractive force due to the optimization objective (eq.(3)), and (ii) the repulsive force due to the potential field around obstacles ($\mathbf{f}_t^{R_k}(n)$). In general, a repulsive force can be modeled as a force vector based on the distance w.r.t. obstacles. Here, we have considered the potential field force variation as a hyperbolic function ($F_{hyp}^{R_k, O_j}(d(n))$) of distance between MAV R_k and obstacle O_j. We use the formulation in [27] to model $F_{hyp}^{R_k, O_j}(d(n))$. Here,

(a) Field local minima problem   (b) Swivelling Robot Destination Method.   (c) Approach Angle Method   (d) Tangential Band Method
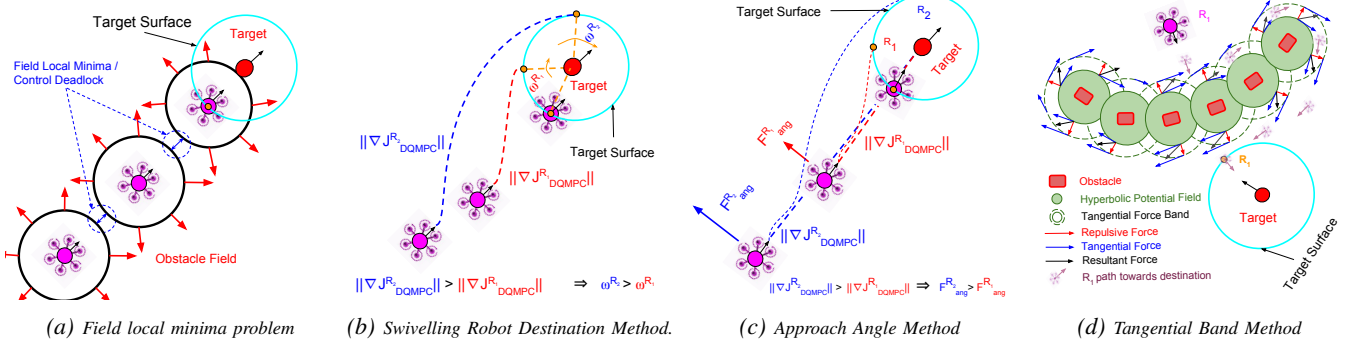
Fig. 2: Illustration of the field local minima problem in obstacle avoidance and different proposed methods.

$d(n) = \|\mathbf{x}_{t-1}^{R_k}(n) - \mathbf{x}_t^{O_j}(n)\|_2$, $\forall n \in [0,\dots,N]$ is the distance between the MAV's predicted horizon positions from the previous time step $(t-1)$ and the obstacles (which includes shared horizon predictions of other MAVs). The repulsive force vector is,

$$\boldsymbol{F}_{rep}^{R_k,O_j}(n) = \begin{cases} F_{hyp}^{R_k,O_j}(d(n))\,\alpha, & \text{if } d(n) < d_{safe} \\ 0, & \text{if } d(n) > d_{safe} \end{cases}, \quad (9)$$

where, d_{safe} is the distance from the obstacle where the potential field magnitude is non-zero. $\alpha = \dfrac{\mathbf{x}_t^{O_j}(n) - \mathbf{x}_{t-1}^{R_k}(n)}{\|\mathbf{x}_t^{O_j}(n) - \mathbf{x}_{t-1}^{R_k}(n)\|_2}$ is the unit vector in the direction away from the obstacle. Additionally we consider a distance $d_{min} << d_{safe}$ around the obstacle, where the potential field magnitude tends to infinity. The potential force acting on an agent per horizon step n is,

$$\mathbf{f}_t^{R_k}(n) = \sum_{\forall\, j} \mathbf{F}_{rep}^{R_k,O_j}(n), \quad (10)$$

which is added into the system dynamics in eq.(4).

D. Resolving the Field Local Minima Problem

The key challenge in potential field based approaches is the field local minima issue [28]. When the summation of attractive and repulsive forces acting on the robot is a zero vector, the robot encounters field local minima problem [1]. Equivalently, a control deadlock could also arise when the robot is constantly pushed in the exact opposite direction. Both local minima and control deadlock are undesirable scenarios. From equation (4) and Algorithm 1, it is clear that the optimization can characterize control inputs that will not lead to collisions, but, cannot characterize those control inputs that lead to these scenarios. In such cases the gradient of optimization would be non-zero indicating that the robot knows its direction of motion towards the target, but cannot reach the destination surface because the potenial field functions are not directly used in DQMPC constraints. Here, we propose three methodologies for field local minima avoidance.

1) Swivelling Robot Destination (SRD) method: This method is based on the idea that the MAV destination $\check{\mathbf{x}}_t^{R_k}$ is an external input to the optimization. Therefore, each MAV can change its $\check{\mathbf{x}}_t^{R_k}$ to push itself out of field local minima. For example, consider the scenario shown in Fig. 2(a), where three robots are axially aligned towards the target. Since the angles of

approach are equal, the desired destination positions are the same for R_1 and R_2, i.e., $\check{\mathbf{x}}_t^{R_1} = \check{\mathbf{x}}_t^{R_2}$. This results in temporary deadlock and will slow the convergence to desired surface. We construct the SRD method to solve this deadlock problem as follows: (i) the gradient of DQMPC objective of R_k is computed, (ii) a swivelling velocity ω^{R_k} is calculated based on the magnitude of gradient, and (iii) $\check{\mathbf{x}}_t^{R_k}$ swivels by a distance proportional to ω^{R_k} as shown in Fig.2(b). This ensures that the velocities at which each $\check{\mathbf{x}}_t^{R_k}$ swivels is different until the robot reaches the target surface, where the gradient tends to zero. The gradient of the optimization with respect to the last horizon step control and state vectors, is computed as follows.

$$\frac{\partial J_{DQMPC}}{\mathbf{x}_t^{R_k}(N+1)} = 2\boldsymbol{\Omega}_t\left(\left[\mathbf{x}_t^{R_k}(N+1)^\top \quad (\dot{\mathbf{x}}_t^{R_k}(N+1))^\top\right] - \left[(\check{\mathbf{x}}_t^{R_k})^\top \quad \mathbf{0}^\top\right]\right)^\top$$

$$\frac{\partial J_{DQMPC}}{\mathbf{u}_t^{R_k}(N)} = 2\boldsymbol{\Omega}_i(\mathbf{u}_t^{R_k}(n) + \mathbf{f}_t^{R_k}(n) + \boldsymbol{g}) + 2\boldsymbol{\Omega}_t B(\mathbf{x}_t^{R_k}(N+1) - \check{\mathbf{x}}_t^{R_k})$$

$$\nabla J_{DQMPC}^{R_k} = \frac{\partial J_{DQMPC}}{\mathbf{x}_t^{R_k}(N+1)} + \frac{\partial J_{DQMPC}}{\mathbf{u}_t^{R_k}(N)}. \quad (11)$$

For circular target surface, the destination point swivel rate is,

$$\check{\mathbf{x}}_t^{R_k} = \mathbf{x}_t^{P} + \begin{bmatrix} d^{R_k}cos(\psi_t^{R_k} \pm k_s\|\nabla J_{DQMPC}^{R_k}\|) \\ d^{R_k}sin(\psi_t^{R_k} \pm k_s\|\nabla J_{DQMPC}^{R_k}\|) \\ h_{gnd} \end{bmatrix}^\top \quad (12)$$

where, k_s is a user-defined gain controlling the impact of $\|\nabla J_{DQMPC}^{R_k}\|$. The swivel direction of each R_k is decided by its approach direction to target. Positive and negative $\psi_t^{R_k}$ leads to a clockwise and anti-clockwise swivel respectively.

2) Approach Angle Towards Target Method: In this method, the local minima and control deadlock is addressed by including an additional potential field function which depends on the approach angle of the robots towards the target. Here, we (i) compute the approach angle of robot R_k w.r.t. the target, (ii) compute the gradient of the objective, and (iii) compute a force $F_{ang}^{R_k,O_j}$ in the direction normal to the angle of approach, as shown in Fig. 2(c). The magnitude of $F_{ang}^{R_k,O_j}$ depends on the sum of gradients $\nabla J_{DQMPC}^{R_k}$ and the hyperbolic function (see Sec. III-C) between the approach angles of robot R_k and obstacles O_j w.r.t. the target. This potential field force is computed as,

$$\mathbf{F}_{ang}^{R_k,O_j}(n) = \nabla J_{DQMPC}^{R_k} F_{hyp}^{R_k,O_j}((\theta^{R_k}(n) - \theta^{O_j}(n))^2)\hat{\beta} \quad \forall j \quad (13)$$

---

[1]note that this is different from optimization objective's local minima.

$$\beta = \pm \frac{\mathbf{x}_t^{R_k}(n) - \check{\mathbf{x}}_t^{R_k}}{\|\mathbf{x}_t^{R_k}(n) - \check{\mathbf{x}}_t^{R_k}\|_2} ; \quad \hat{\beta}.\beta = 0. \tag{14}$$

Here $\theta^{R_k}(n)$ and $\theta^{O_j}(n)$ are the angles of R_k and obstacle O_j with respect to the target. The angles $\theta^{O_j} \; \forall \; j$ w.r.t target are computed by each R_k, as part of the force pre-computation using O_j's position. β and $\hat{\beta}$ are the unit vectors in the approach direction to the target and its orthogonal respectively, with $\pm$ dependent on θ^{R_k} w.r.t. the target. The $\mathbf{f}_t^{R_k}(n)$ for n^{th} horizon step is therefore

$$\mathbf{f}_t^{R_k}(n) = \sum_{\forall \; j} \mathbf{F}_{rep}^{R_k,O_j}(n) + \mathbf{F}_{ang}^{R_k,O_j}(n). \tag{15}$$

Notice that the non-linear constraint of the two approach angles not being equal is converted into an equivalent convex constraint using pre-computed force values. This method ensures collision avoidance in the presence of obstacles and fast convergence to the desired target, because the net potential force direction is always away from the obstacle.

3) Tangential Band Method: The previous methods at times, do not facilitate target surface convergence because of field local minima and control deadlock. For example, static obstacles forming a U-shaped boundary between the target surface and R_k's position, as shown in Fig.2(d). If the target surface is smaller than the projection of the static obstacle along the direction of approach to the desired surface, the planned trajectory is occluded. Therefore, the SRD method cannot find a feasible direction for motion. Furthermore, angle of approach field acts only when the θ^{R_k} and θ^{O_j} are equal w.r.t. the target.

In order to resolve this, we construct a band around each obstacle where an instantaneous (at $n = 0$) tangential force acts about the obstacle center. The width of this band is $> (\dot{\mathbf{x}}_{max}^\top \Delta t + u_{max} \frac{\Delta t^2}{2})$ and therefore, the robot cannot tunnel out of this band within one time step Δt. This makes sure that once the robot enters tangential band, it exits only after it has overcome the static obstacles. The direction depends on R_k's approach towards the target, resulting in clockwise or anti-clockwise force based on $-ve$ or $+ve$ value of $\psi_t^{R_k}$ respectively. The outer surface of the band has only the tangential force effect, while the inner surface has both tangential and repulsive force (repulsive hyperbolic field) effects on R_k.

Within the band the diagonal entries of the positive definite weight matrix $\mathbf{\Omega}_t$ are reduced to a very low value ($\prec\prec \Omega_{t,max}$). This ensures that the attraction field on the robot is reduced while it is being pushed away from the obstacle. Consequently, the effect of tangential force is higher in the presence of obstacles. Once the robot is out of the tangential field band (i.e., clears the U-shaped obstacles), the high weight of the $\mathbf{\Omega}_t$ is restored and the robot converges to its desired destination. Fig. 2(d) illustrates this method. The tangential force is,

$$\mathbf{F}_{tang}^{R_k,O_j}(0) = k_{tang} \nabla J_{DQMPC}^{R_k} \; \hat{\alpha}, \tag{16}$$

where k_{tang} is user-defined gain and $\hat{\alpha}$ is defined s.t. $\pm \hat{\alpha}.\alpha = 0$. The weight matrix and step horizon potential are therefore,

$$\mathbf{\Omega}_t = \mathbf{\Omega}_{t,min}, \quad \text{if } \mathbf{x}_t^{R_k} \le d(0) + d_{band} \tag{17}$$
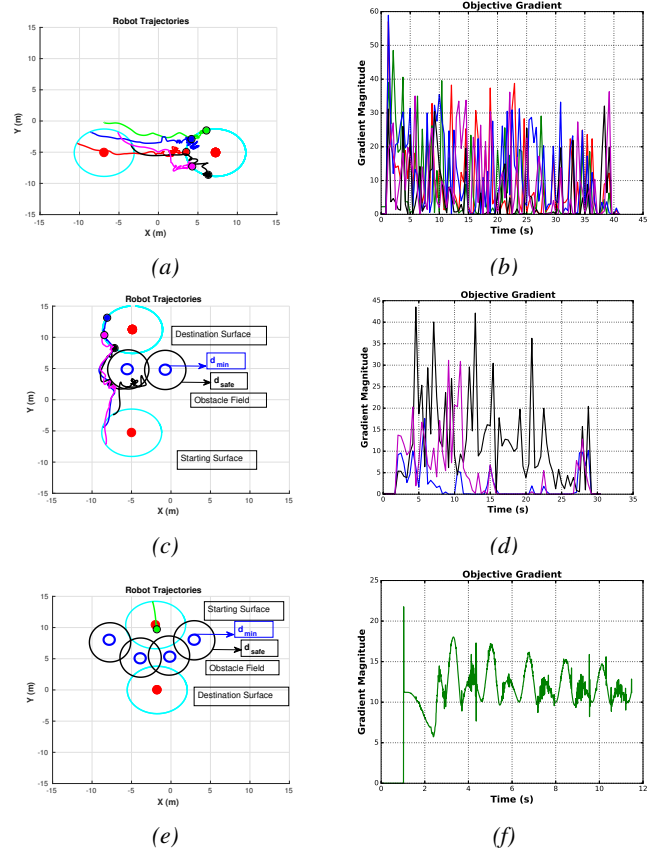

(a)   *(b)*

(c)   *(d)*

(e)   *(f)*

Fig. 3: MAV trajectories and optimization gradients of baseline DQMPC optimization. The colors (red, green, blue, black, magenta) represent R_ks and their respective gradients.

$$\mathbf{f}_t^{R_k}(n) = \sum_{\forall j} \mathbf{F}_{rep}^{R_k,O_j}(n) + \mathbf{F}_{tang}^{R_k,O_j}(0) , \tag{18}$$

where d_{band} is the tangential band width. The values of the weights can vary between $\mathbf{\Omega}_{t,min} \prec \mathbf{\Omega}_t \prec \mathbf{\Omega}_{t,max}$ and changes only when the robot is within the influence of tangential field of any obstacle. In summary, the tangential band method not only guarantees collision avoidance for any obstacles but also facilitates robot convergence to the target surface. In rare scenarios, e.g., when the static obstacle almost encircles the robots and if the desired target surface is beyond such an obstacle, the robots could get trapped in a loop within the tangential band. This is because a minimum attraction field towards the target always exists. Since in this work, the objective is local motion planning in dynamic environments with no global information and map, we do not plan for a feasible trajectory out of such situations.

IV. RESULTS AND DISCUSSIONS

In this section, we detail the experimental setup and the results of our DQMPC based approach along with the field local minima resolving methods proposed in Sec. III for obstacle avoidance and reaching the target surface.

A. Experimental Setup

The algorithms were simulated in a Gazebo+ROS integrated environment to emulate the real world physics and enable a decentralized implementation for validation of the proposed methods. The setup runs on a standalone Intel Xeon E5-2650 processor. The simulation environment consists of multiple
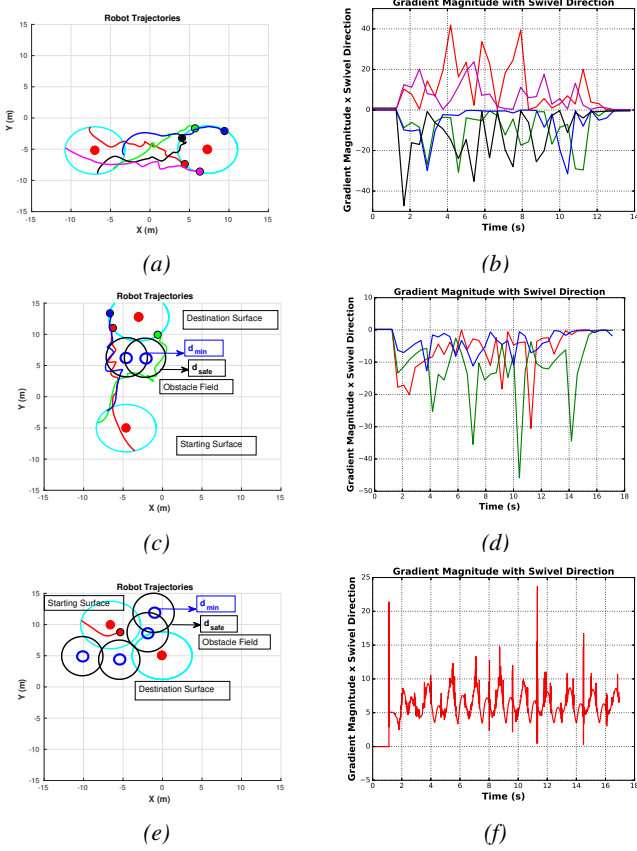
(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4: MAV trajectories and optimization gradients of swivelling destination method.

hexarotor MAVs confined in a 3D world of $20m \times 20m \times 20m$. All the experiments were conducted using multiple MAVs for 3 different task scenarios involving simultaneous target tracking and obstacle avoidance namely,

- Scenario I: 5 MAVs need to traverse from a starting surface to destination surface without collision. Each agent acts as a dynamic obstacle to every other MAV.
- Scenario II: 2 MAVs hover at certain height and act as static obstacles. The remaining 3 MAVs need to reach the desired surface avoiding static and dynamic obstacles.
- Scenario III: 4 MAVs hover and form a U-shaped static obstacle. The remaining MAV must reach destination while avoiding field local minima and control deadlock.

It may be noted that in all the above scenarios, the target's position is drastically changed from initial to final destination. This is done so as to create a more challenging target tracking scenario than simple target position transitions. Furthermore, the scalability and effectiveness of the proposed algorithms are verified by antipodal position swap of 8 MAVs within a surface. The MAVs perform 3D obstacle avoidance to reach their respective positions while ensuring that the surface center (target) is always in sight.

The convex optimization (3)-(7) is solved as a quadratic program using CVXGEN [29]. The DQMPC operates at a rate of 100 Hz. The state and velocity limits of each MAV R_k are $[-20, -20, 3]^\top \leq \mathbf{x}_t^{R_k}(n) \leq [20, 20, 10]^\top$ in m and $[-5, -5, -5]^\top \leq \dot{\mathbf{x}}_t^{R_k}(n) \leq [5, 5, 5]^\top$ in m/s respectively, while the control limits are $[-2, -2, -2]^\top \leq \mathbf{u}_t^{R_k}(n) \leq [2, 2, 2]$ in m/s^2. The desired hovering height of each MAV is $h_{gnd} = 5m$

and the yaw ψ^{R_k} of each MAV is oriented towards the target. The horizon N for the DQMPC and potential force computation is 15 time steps each. It is important to mention that if no trajectory information is available for an obstacle or adversary O_j, the $n = 0$ magnitude of potential field ($F_{rep}^{R_k, O_j}(0)$) is used for the entire horizon. $d_{min} = 0.5m$ and $d_{safe} = 3m$ for the potential field around obstacles. The destination surface is circular, with radius $d^{R_k} = 4m \; \forall k$, around the target for all experiments. However, as stated earlier our approach can attain any desired 3D destination surface.

B. DQMPC: Baseline Method

Figure 3 showcases the multi-robot target tracking results for the three different scenarios (see Sec. IV-A) while applying the baseline DQMPC method. As clearly seen in Fig. 3(a,c), the agents find obstacle free trajectories from starting to destination surface for the scenarios I and II. This is also indicated by the magnitude of gradient dropping close to 0 after 40s and 30s respectively as seen in Fig. 3(b,d). The rapidly varying gradient curve of Scenario I (Fig. 3(b)) shows that, each agent's potential field pushes other agents to reach the destination surface[2]. We observe that the MAVs converge to the destination (Fig. 3(c)) by avoiding the obstacle fields. In the U-shaped static obstacle case (scenario III), the agent is stuck because of control deadlock and fails to reach the destination surface as seen in Fig. 3(e). The periodic pattern of the gradient curve (Fig. 3(f)) and non-zero gradient magnitude makes the deadlock situation evident. Most of these scenarios can also be visualized in the attached video submission.

C. Swivelling Robot Destination Method

In the swivelling destination method, we use $k_s = 0.05$ as the gain for gradient magnitude impact. Fig. 4(a) shows, the MAVs spreading themselves along the destination surface depending on their distance from target and therefore the gradient. This method has a good convergence time of about 12s and 15s (about 3 times faster than the baseline DQMPC) for the scenarios I and II as can be observed in Fig. 4(b) and Fig. 4(d) respectively. Here, the direction of swivel depends on the agents orientation w.r.t. to target and each agent takes a clockwise or anti-clockwise swivel based on its orientation as clearly seen in Fig. 4(c). The positive and negative values of the gradient indicate the direction. It can be observed in Fig. 4(c), that the agent at times crosses over the d_{safe} because of the higher target attraction force but respects the d_{min} where the repulsion force is infinite. Despite the better convergence time, the MAV is stuck when encountered with a U-shaped static obstacle (Fig. 4(e)) because of a control deadlock. This can also be observed with the non-zero gradient in Fig. 4(f).

D. Approach Angle Towards Target Method

Figure 5 showcases the results of the approach angle method. Since the additional potential field force enforces that not more than one agent has the same approach angle, the MAVs spread themselves while approaching the destination surface. This is

---

[2]Note that the scale of the gradient plots are different. The magnitude of the gradient is just to suggest that the robot has reached the target surface.
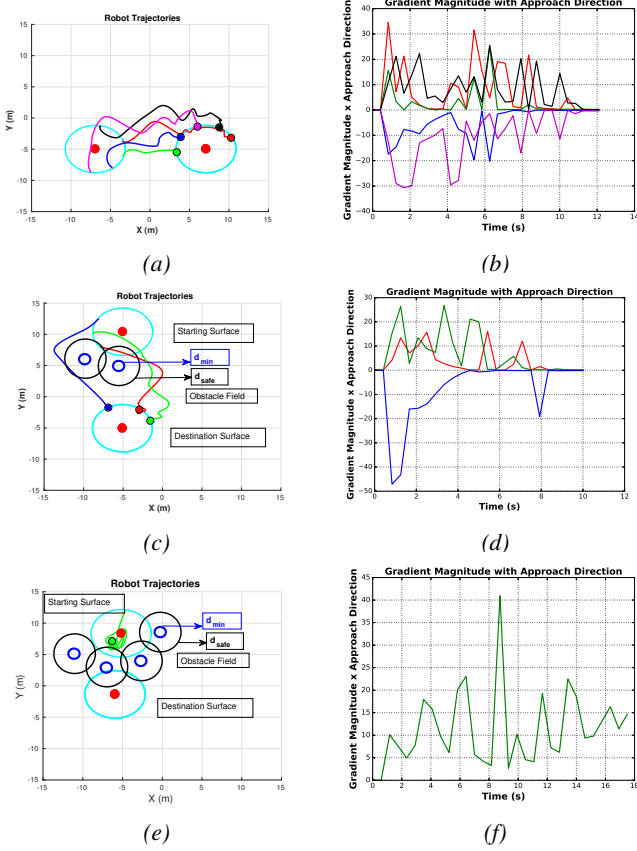
Fig. 5: MAV trajectories and optimization gradients of approach angle force.

Fig. 6: MAV trajectories and optimization gradients of tangential band method.

associated with fast convergence to target surface as observed in Fig. 5(a,b,c,d), for the scenarios I and II. The direction of the approach angle depends on the orientation of the MAVs w.r.t. to the target and is also indicated by the positive and negative values of the gradient magnitude. From the many experiments conducted in environments having only dynamic obstacles or sparsely spaced static obstacles, we observed that this method has the smoothest transition to the destination surface. However, similar to the swivelling destination method, it also fails to overcome control deadlock in case of a U-shaped obstacle as observed in Fig. 5(e,f).

E. Tangential Band Method

The MAVs using tangential band method reach the destination surface in all the scenarios of static and dynamic obstacles as seen in Fig. 6. This method facilitates convergence to the target, for complex static obstacles, because, by principle the MAV traverses within the band until it finds a feasible path towards the target surface. $k_{tang} = 2$ was used in simulations. As seen in Fig. 6(e), as soon as the MAV reaches obstacle surface, the tangential force acts, pushing it in the anti-clockwise direction. Then the UAV travels within this band until it is finally pulled towards the destination surface. The same principle applies for dynamic obstacles scenarios (see Fig. 6(a,c)) as well. Fig. 6(f) shows that, since the MAV overcomes field local minima and control deadlock, the gradient reaches 0 for the U-shaped obstacle with a convergence time of 12 seconds.
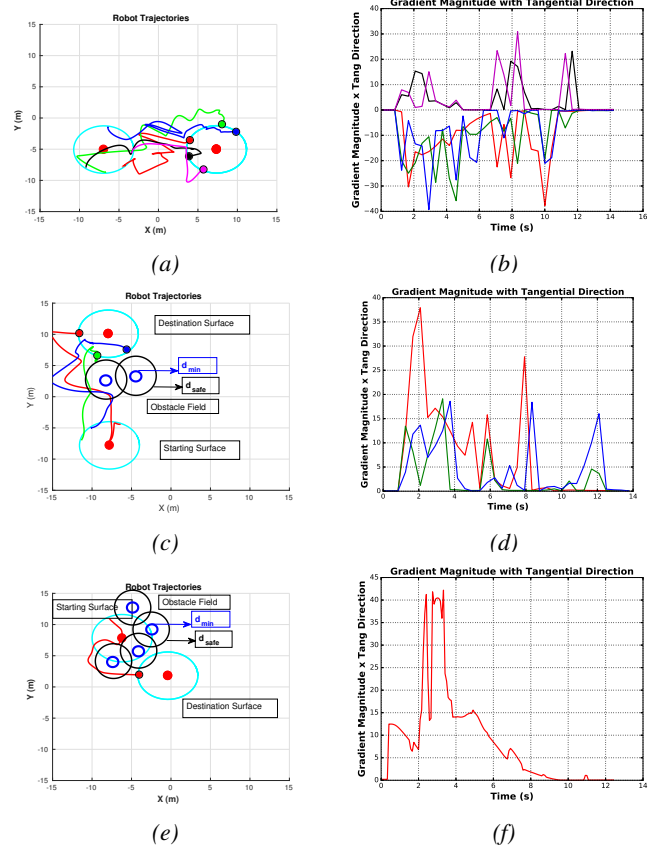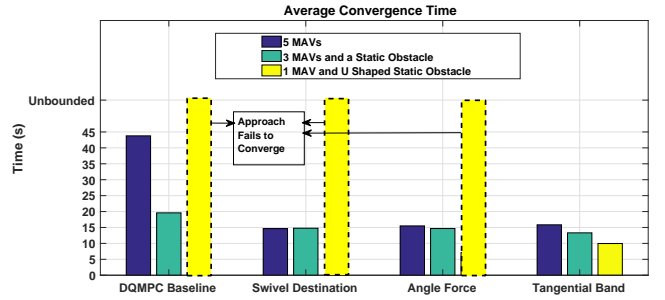

Fig. 7: Average convergence time comparison.

F. Convergence Time Comparison

Figure 7 compares the average convergence time T_{cvg} for each of the proposed methods for 3 trails, with approximately equal travel distances between starting and destination surface for the different scenarios mentioned in Sec. IV-A. The T_{cvg} for swivelling destination, approach angle and tangential band is approximately (15s), which is better compared to DQMPC in the scenarios of only dynamic (Scenario I) or simple static (Scenario II) obstacles. In the U-shaped obstacle (Scenario III), except tangential band the other methods get stuck in field local minima. It may be noted that all the method can be generalized for any shape of the destination surface. In summary, the tangential band method would be the most preferred choice when the type of obstacles in environment are unknown. All the mentioned results and additional experimental results can also be observed in the enclosed video file.

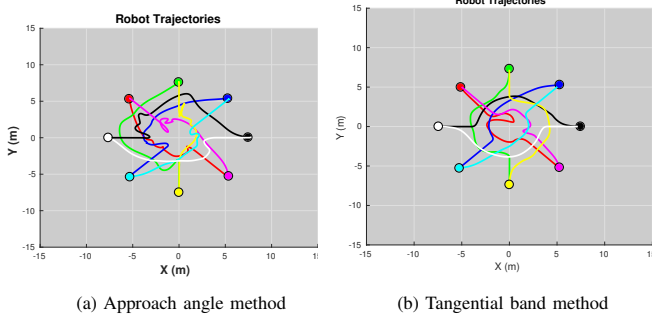(a) Approach angle method   (b) Tangential band method

TABLE I: Robot trajectories during antipodal position swapping.

G. Antipodal Movement

In order to further emphasize the efficacy of tangential band and approach angle methods, we demonstrate obstacle avoidance for a task of intra-surface antipodal position swapping. Here 8 MAVs start on a circular surface with a radius of $8m$, at equal angular distance from each other w.r.t. the center (world frame origin $[0,0,0]^\top$). The MAVs must reach a point 180° in the opposite direction while simultaneously maintaining their orientation towards the center. The plot in Tab. I (a) shows the trajectories taken by MAVs in this task using the approach angle method. All the MAVs convergence to their antipodal positions in around $T_{cvg} = 13s$. It may be noted that there is no trajectory specified and the MAVs compute their own optimal motion plans. To improve the convergence time in this scenario, a consistent clockwise direction is enforced, but this is not necessary and convergence is guaranteed either way. The video attachment visually showcases this antipodal movement.

Similarly the antipodal position swapping was carried out using the tangential band method with the same experimental criteria (see figure in Tab. I (b)). A convergence time of around $T_{cvg} = 11s$ was observed, further emphasizing the speed of obstacle avoidance and convergence of the proposed method.

V. Conclusions and Future Work

In this work, we successfully address the problem of obstacle avoidance in the context of decentralized multi-robot target tracking. Our algorithm uses convex optimization to find collision free motion plans for each robot. We convexify the obstacle avoidance constraints by pre-computing the potential field forces for a horizon and using them as external force inputs in optimization. We show that non-linear dependencies could be converted into such external forces. We validate three methods to avoid field local minima by embedding external forces into the convex optimization. We showcase the efficacy of our approach through gazebo+ROS simulations for various scenarios. Future work involves physically validating the proposed methodology using multiple real robots on different robot platforms (aerial and ground).

Acknowledgments

References

[1] P. Lima, A. Ahmad, and et al., "Formation control driven by cooperative object tracking," *Robotics and Autonomous Systems*, vol. 63, 2015.
[2] S. S. Dias and M. G. Bruno, "Cooperative target tracking using decentralized particle filtering and rss sensors," *IEEE Transactions on Signal Processing*, vol. 61, no. 14, pp. 3632–3646, 2013.
[3] A. Ahmad, G. Lawless, and P. Lima, "An online scalable approach to unified multirobot cooperative localization and object tracking," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1184–1199, 2017.
[4] K. Hausman and et al., "Cooperative control for target tracking with onboard sensing," in *Experimental Robotics*. Springer, 2016.
[5] M. Zhang and H. H. Liu, "Cooperative tracking a moving target using multiple fixed-wing uavs," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3-4, pp. 505–529, 2016.
[6] E. Price, G. Lawless, H. H. Bülthoff, M. Black, and A. Ahmad, "Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles," *arXiv preprint arXiv:1802.01346*, 2018.
[7] T. P. Nascimento and et al., "Nonlinear model predictive formation control: An iterative weighted tuning approach," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 3-4, pp. 441–454, 2015.
[8] T. P. Nascimento, A. G. Conceiçao, and A. P. Moreira, "Multi-robot nonlinear model predictive formation control: the obstacle avoidance problem," *Robotica*, vol. 34, no. 3, pp. 549–567, 2016.
[9] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.
[10] Y. Liu, S. Rajappa, and et al., "Robust nonlinear control approach to nontrivial maneuvers and obstacle avoidance for quadrotor uav under disturbances," *Robotics and Autonomous Systems*, vol. 98, 2017.
[11] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
[12] H. Fukushima, K. Kon, and F. Matsuno, "Model predictive formation control using branch-and-bound compatible with collision avoidance problems," *IEEE Transactions on Robotics*, vol. 29, 2013.
[13] A. U. Raghunathan and et al., "Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft," *Journal of guidance, control, and dynamics*, vol. 27, no. 4, pp. 586–594, 2004.
[14] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, 2013.
[15] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1252–1259, 2007.
[16] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *CDC*, vol. 4. IEEE, 2003.
[17] Z. Chao, L. Ming, Z. Shaolei, and Z. Wenguang, "Collision-free uav formation flight control based on nonlinear mpc," in *ICECC*. IEEE, 2011.
[18] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *IROS*. IEEE, 2012, pp. 1917–1922.
[19] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and model predictive control," in *AIAA*, 2015, p. 0599.
[20] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot navigation in formation via sequential convex programming," in *IROS*. IEEE, 2015.
[21] J. Alonso-Mora and et al., "Collision avoidance for aerial vehicles in multi-agent scenarios," *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, 2015.
[22] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.
[23] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 285–292.
[24] P. Long, W. Liu, and J. Pan, "Deep-learned collision avoidance policy for distributed multiagent navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 656–663, 2017.
[25] D. E. Chang and et al., "Collision avoidance for multiple agent systems," in *CDC*, vol. 1. IEEE, 2003, pp. 539–543.
[26] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *CDC*. IEEE, 2010, pp. 5420–5425.
[27] C. Secchi, A. Franchi, H. H. Bülthoff, and P. R. Giordano, "Bilateral control of the degree of connectivity in multiple mobile-robot teleoperation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3645–3652.
[28] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *ICRA*. IEEE, 1991.
[29] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, 2012.