IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 13, 2020, accepted June 4, 2020, date of publication June 22, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004205

Collision-Free Trajectory Planning With Deadlock Prevention: An Adaptive Virtual Target Approach

RISHI MOHAN[1], EMILIA SILVAS[1,2], HENRY STOUTJESDIJK[3], HERMAN BRUYNINCKX[1], AND BRAM DE JAGER[1]

[1]Department of Mechanical Engineering, Control Systems Technology Section, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands
[2]Netherlands Organization for Applied Scientific Research (TNO), 5708 JZ Helmond, The Netherlands
[3]Mechatronics Development Cluster, Philips Medical Systems, 5684 PC Best, The Netherlands

Corresponding author: Rishi Mohan (r.mohan@tue.nl)

ABSTRACT Most human-centred robotic applications require robots to follow a certain pre-defined path. This makes the robot's autonomous movements acceptable and predictable for humans. Planning a trajectory for the robot thus involves guiding it along this desired path. The classical approach of segmenting a path into multiple waypoints and tracking them only works well in environments which are obstacle-free or contain fixed stationary obstacles. Movable or dynamic obstacles that can potentially lie directly on waypoints result in deadlock situations causing the robot to oscillate around the desired waypoint without moving forward. This chapter presents a novel approach for trajectory planning in which an Adaptive Virtual Target (AVT) is formulated that follows the desired path irrespective of surrounding obstacles. The AVT essentially plays the role of a moving reference for the trajectory planner to track. Additionally, the AVT velocity can be adapted such that the robot can catch up in case of deviations from the path due to obstacle avoidance manoeuvres. A model predictive control (MPC) based trajectory planner tracks the AVT and accounts for obstacle avoidance. The proposed approach allows the robot to keep moving towards the goal by preventing deadlocks while simultaneously minimizing deviation from the desired path. Simulations based on a medical X-ray robot are provided to validate the approach.

INDEX TERMS Trajectory planning, model predictive control, autonomous robots, deadlock prevention.

I. INTRODUCTION

Autonomous mobile robots are gaining increased acceptance in human-centred workspaces such as hospitals, museums, production shop floors and warehouses [1]. While navigating in populated human environments mobile robots should move not only in a safe manner but also in a socially acceptable way [2]. To this end, in many applications (e.g. hospitals or offices) the robot is expected to follow a certain prescribed path while navigating from start to finish. This process makes the autonomous movements of the robot more *natural* and *predictable* for humans [3]. Thus, while generating collision-free trajectories to reach the goal position is of prime importance, the planned trajectories should also be able to guide the robot along this desired path.

A common guidance strategy for autonomous robots is segmenting the desired path into a series of stationary waypoints [4]. Trajectories are then generated such that each waypoint is tracked by the robot while moving along this path [5].

The waypoint tracking approach works well in environments which are obstacle-free or contain fixed stationary objects. However, workspaces in which the location of objects can change as a result of being moved (e.g. a medical cart) and/or contain moving entities (e.g. a doctor or nurse), waypoint tracking can become contradictory in nature to the obstacle avoidance problem. While the robot tries to achieve the objective of approaching a waypoint, obstacles lying directly on (or very close to) the waypoint lead to constraints that prevent the robot from reaching that waypoint. This results in a deadlock where the trajectory planner continues to move the robot around a neighbourhood of the desired waypoint, consequently leading to oscillatory behaviour [6] and rendering the trajectory planner incomplete [7].

This paper presents a waypoint independent approach for collision-free trajectory planning. For this, an

The associate editor coordinating the review of this manuscript and approving it for publication was Farhana Jabeen Jabeen.

Adaptive Virtual Target (AVT) is formulated that moves along the desired path irrespective of obstacles. It plays the role of a moving waypoint and acts as a reference for the robot to track. Assuming there exists no obstacle that intentionally blocks the AVT, forward motion of the AVT ensures existence of an obstacle-free waypoint to be tracked, thus preventing deadlocks.

A. BACKGROUND AND RELATED WORK

1) TRAJECTORY PLANNING

A key component which facilitates robot autonomy is collision-free trajectory planning which is the process of using accumulated sensor data and *apriori* information to find the best trajectory for the mobile robot between initial and goal positions [8], [9]. It is an exhaustively studied concept in robotics and several methodologies such as sampling based methods, graph search algorithms, artificial potential fields (APF's) and optimal control formulations are most widely implemented. A detailed review of each method along with their advantages and disadvantages can be found in [10]–[13]. More recently, Model Predictive Control (MPC) has received attention from researchers for the purpose of collision-free trajectory planning for autonomous robots [14] (for basic notions in MPC see [15], [16]). This is due to MPC's ability to systematically account for the robot's dynamics along with formulating time-varying obstacle avoidance constraints for a dynamically changing environment [17]–[19]. The main advantage of the MPC approach is that collision avoidance is guaranteed, under the conditions that the optimization problem has a feasible solution and environment information is available to a sufficient level of certainty [20], [21]. To leverage these advantages, collision-free trajectory planning in this paper is formulated using an MPC framework.

2) WAYPOINT TRACKING

When a robot is required to follow a path, a guidance approach usually determines the course, heading and speed of the vehicle. A widely recognized guidance strategy is waypoint tracking control in which the path is defined by a set of distinct waypoints and the vehicle is driven to the final goal via these waypoints [22]. Trajectory generation for waypoint tracking can be achieved by using methodologies such as sliding mode control, backstepping and MPC [23]. Although the conventional method of path following via waypoint tracking works well in most cases without obstacles [24]–[27], there exists a major drawback to this approach. When the surrounding environment contains movable or dynamic obstacles, the generated waypoints become non-reachable if a waypoint is covered by (or close to) an obstacle [28]. As a consequence, when the robot approaches a non-reachable waypoint it enters into a deadlock situation where the simultaneous objectives of reaching the waypoint and avoiding the obstacle contradict each other [29]. The existing waypoint tracking literature is restricted to obstacle-free environments and attempts to overcome deadlocks that arise from non-reachable waypoints have not been addressed.

B. CONTRIBUTIONS AND PROPOSED APPROACH

This paper presents two main contributions in the context of collision-free trajectory generation for autonomous robots:

- An Adaptive Virtual Target (AVT) approach that prevents deadlocks which usually hamper the completeness of trajectory planners in waypoint-based approaches.
- An adaptive velocity function that adapts the AVT speed in order to minimize deviation of the robot from the desired path during obstacle avoidance manoeuvres.

The AVT is essentially an exosystem that tracks the desired path irrespective of the obstacles in the surrounding environment. This concept is inspired by the approach developed for unmanned aerial vehicles (UAV's) in [30], [31] where the AVT yields an additional control input for a path-following controller. The work in the present paper adapts the AVT strategy such that a trajectory planner receives the position and orientation of the AVT as a reference signal. Due to its constant forward motion, the AVT serves as a moving reference point to be tracked by the actual robot. This approach ensures the existence of a reference point which is not blocked by an obstacle for infinite time (deadlock prevention) and guides the robot along a desired path towards its goal. Based on the path tracked by the preceding AVT, a trajectory is generated for the following robot which converges to the given path of its predecessor. To this end, a Model Predictive Control (MPC)-based trajectory planner minimizes the cost associated with deviation from the AVT path while also accounting for obstacles through positional constraints for the robot.

During obstacle avoidance manoeuvres the robot might be forced to deviate considerably from the desired path. Once the obstacle is avoided, the robot has to then track the AVT which would have moved significantly ahead during the manoeuvre. This leads to 'corner-cutting' behaviour of the robot and consequently poor convergence to the path is observed with increasing distance from the preceding AVT [32]. In order to track the AVT accurately and improve convergence to the desired path, the AVT is provided with an adaptive velocity function which is based on the relative distance of the actual robot and the AVT [33]. This function slows down the AVT in order for the robot to catch up in case of large deviations from the desired path.

Simulation results based on collision-free navigation of a medical X-ray robot [34] are provided to illustrate the feasibility of the proposed AVT approach.

The paper is organized as follows. Section II provides the problem formulation. The proposed AVT approach is discussed in Section III with Section IV presenting the MPC-based trajectory planner. Section V provides a numerical validation of the proposed approach. Section VI is dedicated to concluding remarks.

II. PROBLEM FORMULATION

Prior to formalizing the virtual target approach and the collision-free trajectory planner, the symbols and definitions used in this paper are introduced. Next, an application scenario is provided which serves as a running example to illustrate the proposed approach and a base for modelling the robot motion, robot geometry and obstacles.

A. PRELIMINARIES

The following notations and definitions are used throughout this paper: Let $\mathbb{R}$, $\mathbb{R}^n$ and $\mathbb{Z}_+$ denote the set of real numbers, n-dimensional coordinate space and set of non-negative integers, respectively. For any set $\mathcal{X} \subseteq \mathbb{R}^n$, int($\mathcal{X}$) denotes the interior of the set. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ denote a $m \times n$ matrix, $\mathbf{0}_n \in \mathbb{R}^{n\times n}$ denote an $n \times n$ zero matrix and $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ denote an $m \times m$ identity matrix. For any vector $\mathbf{x} \in \mathbb{R}^n$ and symmetric matrix $\mathbf{M}$, $||\mathbf{x}||^2_{\mathbf{M}} = \mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}$ denotes the weighted squared 2-norm. A scalar value is represented as $x \in \mathbb{R}$ or $X \in \mathbb{R}$ and an optimal value is denoted by $(\cdot)^*$.

Definition 1 (Convex Polyhedron): A *convex polyhedron*, $\mathcal{H}$, is a set of points obtained by the intersection of a finite number of closed half-spaces, i.e., the set of solutions to some finite system of inequalities [35].

$$\mathcal{H} := \left\{\mathbf{w} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{w} \leq \mathbf{b}\right\} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$ for an n-dimensional polyhedron with m faces and $\mathbf{w}$ is a point in $\mathcal{H}$ that satisfies (1).

Definition 2 (Non-Convex Polyhedron): A *non-convex polyhedron* is the union of a finite number of convex polyhedra such that the union cannot be represented as (1), [35].

Definition 3 (Time-Varying Polyhedron): A *time-varying polyhedron*, $\mathcal{H}_{\text{TV}}$, represents a polyhedron which is translating and/or rotating in time [36]. $\mathcal{H}_{\text{TV}}(t)$ at time instant t is

$$\mathcal{H}_{\text{TV}}(t) := \mathbf{S}(\delta(t))\mathcal{H}_{\text{TV}}(t-1) + \mathbf{q}(t) \tag{2}$$

where $\mathbf{S}(\delta(t))$ is the rotation matrix, $\delta(t)$ is the angle of rotation and $\mathbf{q}(t)$ is the translation vector of a specific reference point in $\mathcal{H}_{\text{TV}}$.

Definition 4 (Deadlock): At any time instant, a robot is in a *deadlock* if it has not achieved its goal configuration; it does not make progress towards this goal, i.e., it is not moving; and will stay in its current position as time $\rightarrow \infty$ [37].

B. APPLICATION SCENARIO

To illustrate the proposed trajectory planning approach, the collision-free navigation of an interventional X-ray system in a surgical room is considered (Fig. 1). Prior to a medical X-ray imaging procedure of the patient, it is required to navigate the X-ray system from its prescribed stand-by position to the patient bed. Autonomous navigation of the system is not a trivial task as there exists a risk of collision between the X-ray system and the surrounding objects in the room (e.g. medical equipment, moving clinical staff). The scenario is analogous to standard motion planning problems in mobile robotic applications and can be regarded as a representative example for the general robotics community.

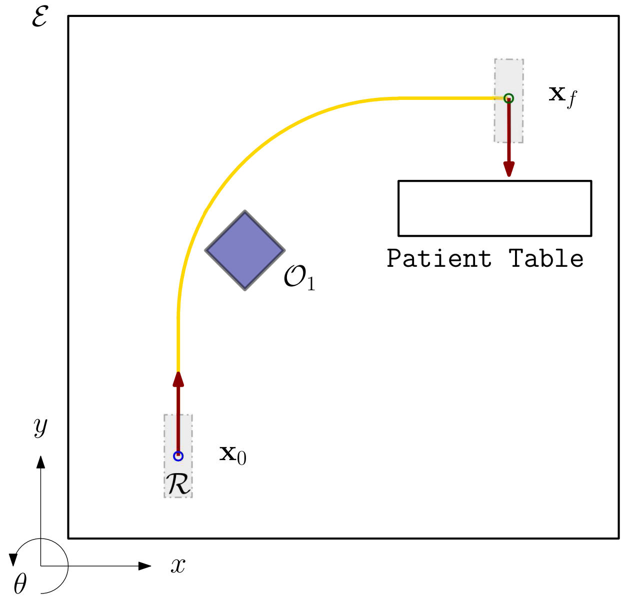


FIGURE 1. Schematic (top-view) representation of the positioning procedure of the X-ray system ($\mathcal{R}$) in the interventional room $\mathcal{E}$.

The interventional X-ray system (henceforth referred to as *robot*), $\mathcal{R}$, is required to autonomously navigate from its dedicated stand-by configuration, $\mathbf{x}_0$ to a final configuration, $\mathbf{x}_\mathrm{f}$, at the patient table while avoiding collisions with a surrounding static object, $\mathcal{O}_1$ (e.g. a medical cart). Additionally, $\mathcal{R}$ is required to follow a desired path which is pre-defined by the medical staff, such that the robot's autonomous behaviour is intuitive and socially acceptable to the staff. The problem is formulated based on the following set of assumptions:

- A1 $\mathcal{R}$ is equipped with sensor systems which measure its position in the environment as well as the relative positions and velocities of surrounding objects.
- A2 $\mathcal{R}$ is equipped with low-level control systems capable of following the planned trajectory.
- A3 For the trajectory planning problem to be well-posed, the initial and goal configurations should be admissible. Thus, it is assumed that the initial and goal positions are always obstacle-free.
- A4 There exists a feasible path at all times such that the entire path is not blocked by obstacles.

C. ENVIRONMENT MODELLING

Let the environment in which the robot and obstacles co-exist be defined by $\mathcal{E} \subset \mathbb{R}^2$. All objects in $\mathcal{E}$ are represented by polyhedra in $\mathbb{R}^2$.

1) ROBOT GEOMETRY

The robot geometry is modelled as a time-varying convex polyhedron $\mathcal{R}$, as defined in (2). Throughout this paper and without loss of generality, $\mathcal{R}$ is assumed to be a rectangle. The motion of a polyhedron is specified (for each time instant) by the location of a particular reference point, which in this case is the geometric centre of $\mathcal{R}$.

2) ROBOT MOTION

The robot's motion is defined with respect to the location of its geometric centre. To model this, the robot is represented as a point mass holonomic system, with state ($\mathbf{x}$), control ($\mathbf{u}$) and output ($\mathbf{y}$) vectors chosen to be

$$\begin{aligned}\mathbf{x}(t) &= \begin{bmatrix} x(t) & y(t) & \theta(t) & v_x(t) & v_y(t) & \omega(t)\end{bmatrix}^\intercal \\ \mathbf{u}(t) &= \begin{bmatrix} a_x(t) & a_y(t) & \alpha(t)\end{bmatrix}^\intercal \\ \mathbf{y}(t) &= \begin{bmatrix} x(t) & y(t) & \theta(t)\end{bmatrix}^\intercal \end{aligned} \tag{3}$$

where $x(t)$ and $y(t)$ are the lateral and longitudinal positions, $\theta(t)$ is the orientation, $v_x(t)$ and $v_y(t)$ denote lateral and longitudinal velocity, $\omega(t)$ is angular velocity, and $a_x(t)$, $a_y(t)$, $\alpha(t)$ are the lateral, longitudinal and angular accelerations, respectively of the robot's geometric centre. The robot dynamics are expressed by the linear time-invariant (LTI), continuous time system as

$$\begin{aligned}\frac{d(\mathbf{x}(t))}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned} \tag{4}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_3 & \mathrm{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_3 \\ \mathrm{I}_3 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} \mathrm{I}_3 & \mathbf{0}_3 \end{bmatrix} \tag{5}$$

The system model in (5) will eventually be utilized in the formulation of the Model Predictive Control (MPC) based trajectory planner. Since practical implementation and evaluation requires a discrete-time approach [38], the continuous time system (4) is discretized using zero-order hold with a discretization period T_s.

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}_\mathrm{d}\mathbf{x}_k + \mathbf{B}_\mathrm{d}\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}_\mathrm{d}\mathbf{x}_k\end{aligned} \tag{6}$$

where $\mathbf{A}_\mathrm{d}$, $\mathbf{B}_\mathrm{d}$ and $\mathbf{C}_\mathrm{d}$ are the discretized versions of the matrices in (5) and $k \in \mathbb{Z}_+$ is the discrete time instant related to continuous time by $t = kT_s$.

3) OBSTACLE MODELLING

Consider $n \geq 1$ obstacles in the environment which are modelled as static convex polyhedra of the form (1). Obstacles are denoted by $\mathcal{O}_i,\ i \in \{1, \ldots, n\}$. Throughout this paper and without loss of generality, $\mathcal{O}_i \subset \mathbb{R}^2$ is a square with geometric centre as the reference point.

4) OBSTACLE ENLARGEMENT

In order to ensure collision avoidance, it is required that the robot and obstacle polyhedra do not overlap, i.e.,

$$\mathcal{R} \cap \mathcal{O}_i = \varnothing, \quad \forall i \in \{1, \ldots, n\}. \tag{7}$$

However, directly implementing the constraint (7) in the MPC optimization problem is not trivial and computationally inefficient [39], [40]. To overcome this, the actual obstacles are enlarged in size with the length of $\mathcal{R}$ and are denoted by $\mathcal{O}_\mathrm{i}^\mathrm{safe}$, $i \in \{1, \ldots, n\}$. It allows treating the robot as a point mass (3) and explicitly formulating collision avoidance as state constraints in the MPC optimization problem for numerical feasibility [9].

Note that obstacle enlargement is highly conservative and limits the solution space even though it is standard practice in robot motion planning [8], [41]. This conservativeness, proposed as a part of future work by the authors, can be addressed using alternatives to enlargement (e.g. [42], [43]).

5) DESIRED PATH

The pre-defined path between the initial and goal configurations that the robot is required to follow is given by a planar, geometric curve $\mathcal{P}(s) \in \mathbb{R}^2$ parametrized by a scalar parameter s and curvature $\kappa(s)$. Note that for a fixed $s \geq 0$, $[x(s)\ y(s)] \in \mathcal{P}(s)$ is a point on the path in environment $\mathcal{E}$. The set of all such points is the path $\mathcal{P}(s)$. It is assumed that $\mathcal{P}(s)$ is a $\mathcal{C}^2$ function to ensure a smooth and continuous curve.

D. PROBLEM STATEMENT

Considering the above described environment, robot, obstacles and path to be tracked, the general collision-free trajectory planning problem can be formulated as follows (Fig. 1):

Given a robot, $\mathcal{R}$, as described in (6) and a desired path, $\mathcal{P}$, derive an optimal reference trajectory $\mathbf{x}^*$, $\mathbf{u}^*$ starting at initial configuration $\mathbf{x}_0$ which (i) stabilizes $\mathcal{R}$ at final configuration $\mathbf{x}_f$ while (ii) preventing deadlocks; (iii) avoiding obstacles $\mathcal{O}_i,\ i \in \{1, 2, \ldots, n\}$; and (iv) ensuring the motion of $\mathcal{R}$ along $\mathcal{P}$ with minimum deviation.

III. DEADLOCK PREVENTION - ADAPTIVE VIRTUAL TARGET APPROACH

In order to prevent deadlocks during trajectory planning, an Adaptive Virtual Target (AVT) approach is formulated that serves as the reference to guide the trajectory planner. The AVT is constrained to move along the desired path irrespective of surrounding obstacles and is equipped with an adaptive velocity function to improve path tracking.

A. VIRTUAL TARGET FORMULATION

Assuming a path is defined by waypoints, a deadlock occurs when the robot is faced with the contradictory objectives of reaching a waypoint while trying to avoid a collision with an obstacle which is blocking this waypoint. The crux of this paper is a waypoint independent approach which eliminates the occurrence of this contradiction. To this end, a virtual target is formulated which is constrained to travel along the desired path irrespective of the surrounding obstacles. This virtual target can be thought of as a dynamic waypoint which serves as a moving reference point to be tracked by the actual robot. Under assumption A4, this approach ensures the existence of a reference point which is not blocked by an obstacle for infinite time and guides the robot along a desired path towards its goal.

The virtual target is modelled as a wheeled mobile robot [44] which moves along $\mathcal{P}(s)$ according to

$$\begin{aligned}\dot{x}_{\mathrm{v}}(s) &= V\cos\theta_{\mathrm{v}}(s)\\ \dot{y}_{\mathrm{v}}(s) &= V\sin\theta_{\mathrm{v}}(s)\\ \dot{\theta}_{\mathrm{v}}(s) &= V\kappa(s)\end{aligned} \tag{8}$$

where the dot on a variable represents the first derivative, V is virtual target's linear velocity and $\mathbf{y}_{\mathrm{v}} = [x_{\mathrm{v}}\ y_{\mathrm{v}}\ \theta_{\mathrm{v}}]$ is the configuration (position and heading) of the virtual target which effectively yields the reference for the actual robot.

Remark 1: The virtual target does not account for the kinematics and/or dynamics of the actual robot and is not restricted by any constraints to avoid collisions.

B. ADAPTIVE VELOCITY FUNCTION

The choice of V is a design parameter to be chosen by the user. Although a constant value of V would serve the purpose of a moving virtual target which guides the actual robot towards the final goal, it does not account for any deviations between the position of the target and robot. Since the robot is constrained to follow the preceding AVT, in case of a curved path or a large deviation (due to obstacle avoidance) the phenomenon of 'corner cutting' is observed. This behaviour escalates with increasing deviation from the AVT [32]. To this end, instead of choosing a constant V, an adaptive velocity function based on [33] is formulated as

$$V = V_{\mathrm{d}}\left(1 - \eta\tanh\left(\gamma\right)\right) \tag{9}$$

where $V_{\mathrm{d}} > 0$ is the fixed AVT velocity, $0 < \eta < 1$ is a constant and γ is the euclidean distance between the AVT and the actual robot. The function provides AVT with the capability to slow down when the robot is far away from $P(s)$ and the strength of the function to reduce AVT velocity depends on parameter η.

Remark 2: The condition $0 < \eta < 1$ ensures that the virtual target does not

(i) move backwards along the path ($V < 0$) ensuring a forward moving reference progressing towards the goal and not away from it;

(ii) come to a complete halt (V = 0) resulting in a deadlock. Although, $\eta \approx 1$ does cause the AVT to move extremely slowly when γ is large.

IV. COLLISION-FREE TRAJECTORY PLANNING

Trajectory generation aims to satisfy objectives (i)-(iv) as mentioned in the problem statement (Sect. II-D). In general, a trajectory is planned such that the robot is guided along the desired path by the AVT. However, the point stabilization objective imposes strict requirements on $\mathcal{R}$ to be at rest and achieve a desired position and heading at $\mathbf{x}_{\mathrm{f}}$. Thus, point stabilization is prioritized at a certain point and the specification to track the AVT is dropped. This necessitates the need for a switch (or a jump) during trajectory planning.To address this, trajectory planning is formulated as a hierarchical two-level framework with switching described by a hybrid automaton and collision-free trajectories generated using MPC.

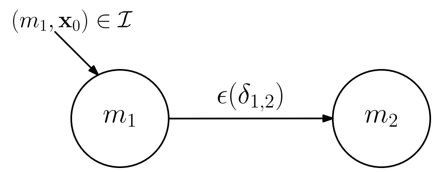


FIGURE 2. **Hybrid automaton Σ for decision making.**

A. HYBRID AUTOMATON FOR SWITCHING

Switching between AVT tracking and point stabilization is encoded as a hybrid automaton. Based on the definition in [45], consider the hybrid automaton $\Sigma = \{\mathcal{M}, \mathcal{X}, \mathcal{I}, \mathcal{U}, \mathcal{D}, \epsilon, f\}$ with

1) $\mathcal{M} = \{m_1, m_2\}$ is a set of discrete modes.
2) $\mathcal{X} \subseteq \mathbb{R}^6$ and $\mathcal{U} \subseteq \mathbb{R}^3$ are the continuous state space and input space of the system.
3) $\mathcal{I} \subseteq \mathcal{M} \times \mathcal{X}$ is a set of initial states.
4) $\mathcal{D} \subseteq \mathcal{M} \times \mathcal{M}$ is the set of discrete transitions.
5) $\epsilon(\delta_{i,j})$ is a decision event which enables the transition $\delta_{i,j} = (m_i, m_j) \in \mathcal{D}$.
6) $f : \mathcal{M} \times \mathcal{X}$ is a flow function which governs the evolution of the continuous state in an individual mode.

The two modes, which correspond to MPC-based trajectory planner with different objectives (detailed in the next subsection), are explained as follows:

1) AVT TRACKING (m_1)

This mode corresponds to objective (iv) in which the robot should follow the desired path by means of tracking the AVT. Additionally, it is expected that in this mode, an avoidance manoeuvre for obstacles should be preformed, which corresponds to objective (iii).

2) POINT STABILIZATION (m_2)

This mode addresses *point stabilization* which refers to the problem of steering a system to a final target point, with a desired velocity and orientation [46]. The specification to track the AVT is not necessary after a certain point, which is when the trajectory planner switches to this mode.

The decision event, $\epsilon(\delta_{1,2})$, which enables the transition between mode 1 and 2 is defined as:

$$\epsilon(\delta_{1,2}) := \{\mathbf{y}_k | \Delta(\mathbf{y}_k, \mathbf{x}_{\mathrm{f}}) < \xi\} \tag{10}$$

where, $\Delta(\cdot,\cdot)$ is the euclidean distance between the two entities in brackets, $\mathbf{y}_k$ is the current position and orientation of the robot and ξ is a constant, application-specific value, which is taken as the distance-to-brake (DTB) for a system.

The initial state of the automaton is always $(m_1, \mathbf{x}_0) \in \mathcal{I}$. Since mode m_2 (point stabilization) is the target mode for the automaton, no transition exists from there. Also, from an application perspective, it is assumed that once m_2 is activated the robot (C-arm) is close enough to its goal configuration and does not need to track the AVT. Thus, the transition $\delta_{2,1}$ does not exist. The flow function, f, is the discrete-time model (6) which evolves based on an input given by the MPC-based trajectory planner.

Remark 3: Although this work considers only two modes and one event, the structure of the automaton allows more modes and decision events to be added easily. This can be, for example, an obstacle prioritization mode where the robot should 'give way' to moving obstacles in the environment.

B. MPC-BASED TRAJECTORY PLANNING

Trajectory generation in each mode is executed using the MPC-based planner which aims at addressing the respective objectives and system constraints. The general MPC constrained optimization formulation is as follows:

$$\begin{aligned} \min_{\mathbf{U}_k} \; & J(\mathbf{x}_k, \mathbf{U}_k) \\ \text{s.t. } & \text{system dynamics as in (6)} \\ & \text{system constraints} \\ & \text{obstacle avoidance constraints} \end{aligned} \quad (11)$$

where $\mathbf{U}_k = [\mathbf{u}_0, \; \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}]$ is the vector of stacked inputs obtained over a prediction horizon N and $J(\mathbf{x}_k, \mathbf{U}_k)$ is a objective function. The optimization problem is solved in a receding horizon manner with the current state $\mathbf{x}_k$ and horizon N. At each time instant, an optimal input sequence $\mathbf{U}_k^* = [\mathbf{u}_0^*, \ldots, \mathbf{u}_{N-1}^*]$ is obtained and the first element $\mathbf{u}_0^*$ is used as a control input to the robot model (6). The MPC trajectory planner for AVT tracking (m_1) and Point Stabilization (m_2) is formulated as (12) and (13), respectively.

$$\begin{aligned} \min_{\mathbf{U}_k} \; & \sum_{\ell=0}^{N-1} ||\mathbf{y}_\ell - \mathbf{y}_\text{v}||_\mathbf{Q}^2 + ||\mathbf{u}_\ell||_\mathbf{R}^2 \\ \text{s.t. } & \mathbf{x}_{\ell+1} = A_d \mathbf{x}_\ell + B_d \mathbf{u}_\ell \\ & \mathbf{y}_\ell = C_d \mathbf{x}_\ell \\ & \mathbf{x}_\ell \in \mathcal{X}, \quad \mathbf{y}_\ell \in \mathcal{Y}, \; \mathbf{u}_\ell \in \mathcal{U} \\ & \mathbf{y}_\ell \notin \mathcal{O}_i^\text{safe}, \quad \forall i \in \{1, \ldots, n\} \end{aligned} \quad (12)$$

$$\begin{aligned} \min_{\mathbf{U}_k} \; & \sum_{\ell=0}^{N-1} ||\mathbf{x}_\ell - \mathbf{x}_\text{f}||_\mathbf{P}^2 + ||\mathbf{u}_\ell||_\mathbf{R}^2 \\ \text{s.t. } & \mathbf{x}_{\ell+1} = A_d \mathbf{x}_\ell + B_d \mathbf{u}_\ell \\ & \mathbf{y}_\ell = C_d \mathbf{x}_\ell \\ & \mathbf{x}_\ell \in \mathcal{X}, \quad \mathbf{y}_\ell \in \mathcal{Y}, \; \mathbf{u}_\ell \in \mathcal{U} \end{aligned} \quad (13)$$

Trajectory planning is executed as follows:

1) Given the initial configuration $\mathbf{x}_0$, trajectory planning is initialized in m_1. The MPC planner minimizes the objective function as in (12). The first term penalizes deviations from the current AVT state $\mathbf{y}_\text{v}$ using penalty matrix $\mathbf{Q}$. The second term discourages high system inputs (using matrix $\mathbf{R}$) to ensure smoother trajectories. Obstacle avoidance is formulated as $\mathbf{y}_\ell \notin \mathcal{O}_i^\text{safe}$, which restricts the planned trajectory from entering the obstacle's safe zone. At each time instant k, $\mathbf{U}_k^*$ is obtained which steers the system according to the flow function (6) and the mode remains unchanged for the next time instant, $k+1$, until $\epsilon(\delta_{1,2})$ is not satisfied.
2) At a certain distance, ξ, from the goal, transition from m_1 to m_2 is triggered by $\epsilon(\delta_{1,2})$. Consequently, the MPC planner (13) aims to steer the robot to the final state $\mathbf{x}_\text{f}$. This is ensured by the first term of the objective function with the second term being responsible for smoothness. The obstacle avoidance constraint is dropped assuming no obstacles are present when the robot is close to the final position. Trajectory generation is completed once the robot is stabilized at $\mathbf{x}_\text{f}$.

The optimal trajectory is then provided to a tracking controller as reference input. Fig. 3 presents a schematic of the trajectory planning architecture.

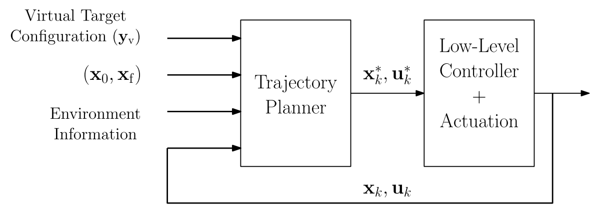


FIGURE 3. Schematic representation of the trajectory planning framework.

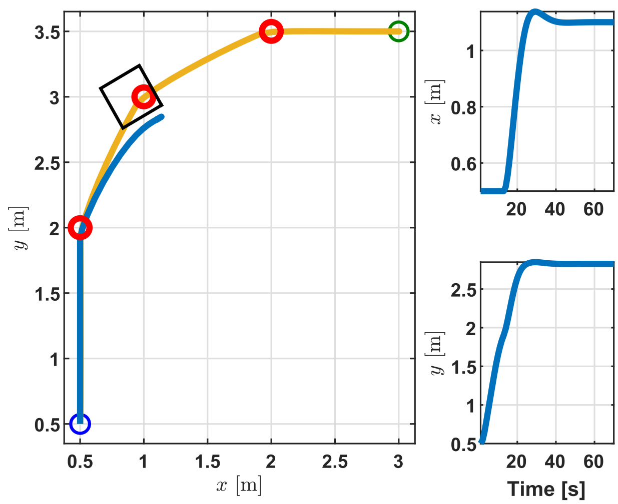
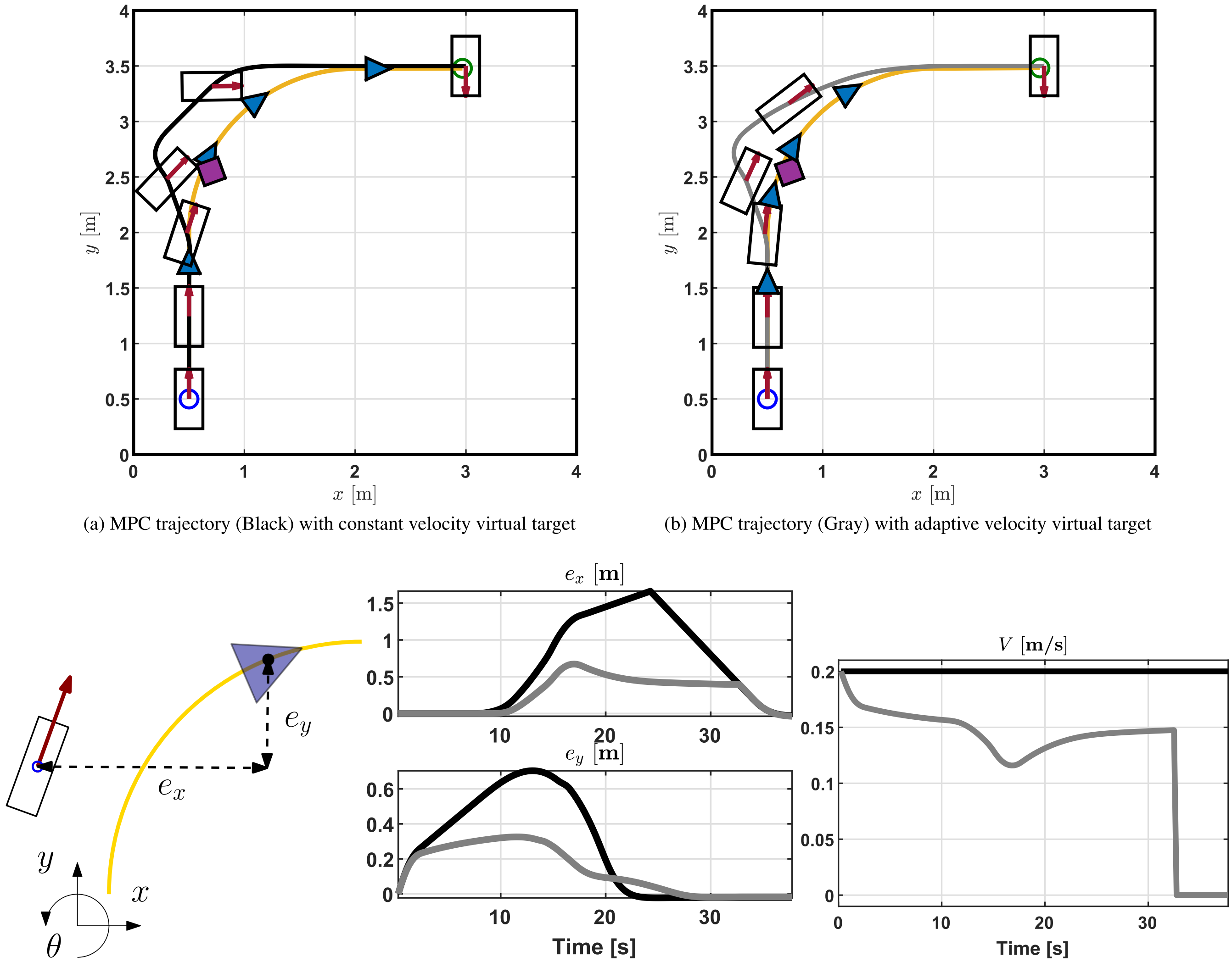


FIGURE 4. Deadlock scenario: Starting configuration ($\mathbf{x}_0$) - blue circle, final configuration ($\mathbf{x}_\mathbf{f}$) - green circle, desired path ($\mathcal{P}$) - yellow, waypoints - red circles, obstacle ($\mathcal{O}_1$) - white square, robot trajectory - blue.

V. RESULTS

In this section, the proposed AVT approach is validated using the application scenario described in Section II-B. First, an example of a deadlock situation due to non-reachable waypoints is provided. Next, results presenting the successful deadlock prevention and obstacle avoidance using the AVT approach are given. Finally, benefits of the adaptive nature of the AVT for tracking a curved path without 'corner cutting' are illustrated.



(a) MPC trajectory (Black) with constant velocity virtual target (b) MPC trajectory (Gray) with adaptive velocity virtual target

(c) Tracking error coordinates (d) Error in tracking virtual target (e) Virtual target velocity

FIGURE 5. **Collision-free trajectory planning with deadlock prevention: Robot ($\mathcal{R}$) - white rectangle, heading direction $\mathcal{R}$ - red arrow, static obstacle ($\mathcal{O}_1$) - purple square, obstacle enlargement ($\mathcal{O}_1^{safe}$) - black dash-dot, vitrual target - blue triangle, starting configuration ($\mathbf{x}_0$) - blue circle, final configuration ($\mathbf{x}_f$) - green circle, desired path ($\mathcal{P}$) - yellow.** ***Black and gray curves represent results with constant and adaptive velocity for the virtual target, respectively.***

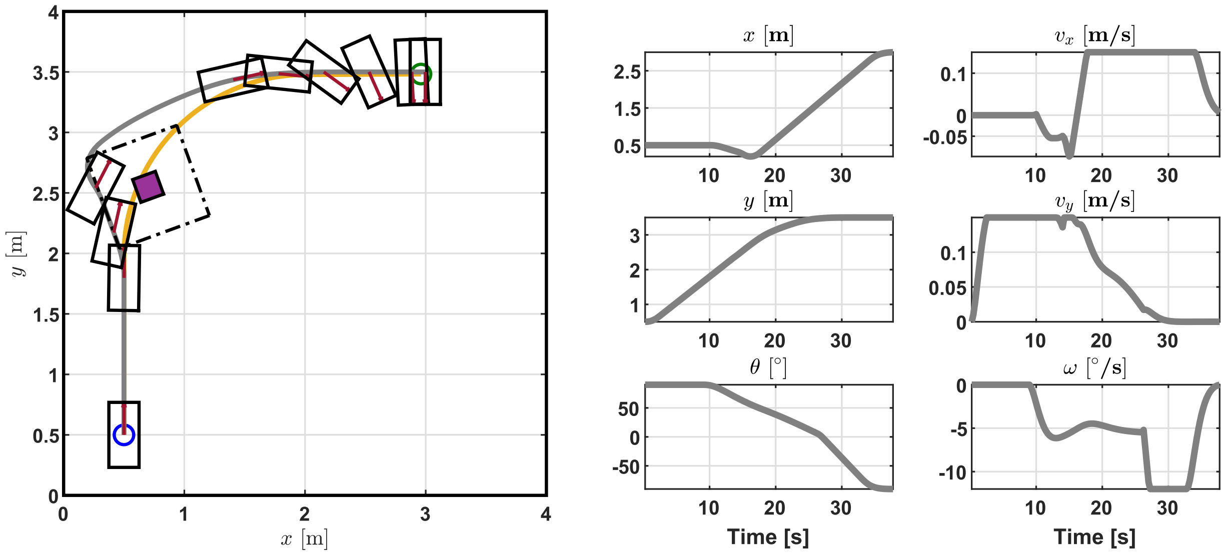
It is assumed that the position of the obstacle is known apriori at all time instants. The desired path is specified as a concatenation of two straight lines (zero curvature) of 1.5 [m] and a circular arc (constant curvature) of radius 1.5 [m] in between. The trajectory planning algorithm is implemented as a constrained MPC problem in MATLAB using the Multi-Parametric Toolbox (MPT3.0) [47] and applied to the scenario with design parameters provided in Table 1.

A. DEADLOCK EXAMPLE

To provide a baseline for comparison, consider the example in Fig. 4 where the desired path (yellow) is segmented into three waypoints (red circles) which the robot must track to follow the desired path (robot geometry is omitted for clarity). The obstacle $\mathcal{O}_1$ (white square) is placed (after waypoint generation) such that it completely envelopes the second waypoint. Using the MPC-based planner in (12), a trajectory is generated from start (blue circle) towards the goal (green circle). The generated trajectory (blue) tracks the first waypoint, however it is constrained to avoid the obstacle while trying to reach the second waypoint. This creates a deadlock situation rendering the robot stationary around the waypoint without ever reaching it. Fig. 4 also depicts the longitudinal (x) and lateral (y) coordinates which remain unchanged after $t = 40$ [s], thus resulting in a stationary robot for infinite time.

B. DEADLOCK PREVENTION - AVT APPROACH

Trajectory planning based on the proposed ATV approach is compared to the baseline example described above. Consider the scenario in Fig. 5, where $\mathcal{R}$ needs to move from its starting configuration, $\mathbf{x}_0$ (blue circle), to its final



(a) Timestamps for robot geometry

(b) Time evolution of robot states

FIGURE 6. Obstacle avoidance and point stabilization: Robot ($\mathcal{R}$) - white rectangle, heading direction - red arrow, static obstacle ($\mathcal{O}_1$) - purple square, obstacle enlargement ($\mathcal{O}_1^{\text{safe}}$) - black dash-dot, starting configuration ($\mathbf{x}_0$) - blue circle, final configuration ($\mathbf{x}_f$) - green circle, desired path ($\mathcal{P}$) - yellow, MPC trajectory - gray.

TABLE 1. Design parameters - simulation results.

Interventional Environment			
$\mathcal{E}$	4×4 [m]	$\mathcal{R}$	1.075×0.5 [m^2]
$\mathcal{O}_1$	0.20×0.20 [m^2]	$\mathcal{O}_1^{\text{safe}}$	0.80×0.80 [m^2]
MPC-based Trajectory Planner			
T_s	0.25 [s]	$x_{\min} = 0$ [m]	$x_{\max} = 4$ [m]
D_p	5 [s]	$y_{\min} = 0$ [m]	$y_{\max} = 4$ [m]
N	20	$\theta_{\min} = -\pi/2$ [rad]	$\theta_{\max} = \pi/2$ [rad]
$\mathbf{P}$	I_6	$v_{x_{\min}} = -0.15$ [m/s]	$v_{x_{\max}} = 0.15$ [m/s]
$\mathbf{Q}$	I_3	$v_{y_{\min}} = -0.15$ [m/s]	$v_{y_{\max}} = 0.15$ [m/s]
$\mathbf{R}$	I_3	$\omega_{\min} = -0.2$ [rad/s]	$\omega_{\max} = 0.2$ [rad/s]
$\mathbf{x}_0$	$[0.5\ 0.5\ \frac{\pi}{2}\ 0\ 0\ 0]^\mathsf{T}$	$a_{x_{\min}} = -0.1$ [m/s^2]	$a_{x_{\max}} = 0.1$ [m/s^2]
$\mathbf{x}_f$	$[3\ 3.5\ -\frac{\pi}{2}\ 0\ 0\ 0]^\mathsf{T}$	$a_{y_{\min}} = -0.1$ [m/s^2]	$a_{y_{\max}} = 0.1$ [m/s^2]
ξ	1 [m]	$\alpha_{\min} = -0.14$ [rad/s^2]	$\alpha_{\max} = 0.14$ [rad/s^2]
Adaptive Virtual Target			
V_d	0.20 [m/s]	β	0.70 [-]

configuration, $\mathbf{x}_f$ (green circle) along the desired path (yellow). The static obstacle $\mathcal{O}_1$ (purple) intersects the path of $\mathcal{R}$. Figure 5a presents the MPC-based collision-free trajectory (black curve) generated using a constant velocity virtual target, i.e., $\eta = 0$ in (9). In Figure 5b, the resultant trajectory (grey curve) is based on a virtual target with adaptive velocity ($\eta > 0$). For both cases a collision-free trajectory is successfully planned from initial to final configuration, illustrating the prevention of a deadlock as compared to the baseline example. Time stamps of the virtual target (blue triangle)[1] at $t \in \{6.25, 11.25, 14.5, 20.25\}$ [s] in both cases show that the AVT moves along $\mathcal{P}(s)$ irrespective of $\mathcal{O}_1$, thereby ensuring an obstacle-free reference point for the MPC-planner to track.

1) AVT TRACKING

To further analyse the behaviour of $\mathcal{R}$ in tracking the AVT, consider the lateral (e_x) and longitudinal (e_y) error coordinates depicted in Figure 5c. The trajectory based on adaptive velocity is able to track the virtual target better (lower tracking error in Figure 5d) and deviates less from the desired path. This is a result of the AVT slowing down ($t = 11$[s] in Figure 5e) when $\mathcal{R}$ moves away from the desired path to avoid $\mathcal{O}_1$. This is also clear from the timestamps of the virtual target which are closer for adaptive V as compared to constant V for the same time instant.

Another observation from Figure 5e is the initial decrease in AVT velocity ($t = 0$ to 3 [s]). While the model (8) allows instantaneous acceleration of the AVT to reach desired velocity V_d, the MPC trajectory is generated considering the robot model and physical constraints. Since $\mathcal{R}$ cannot accelerate instantaneously, it lags behind the AVT as observed in the initial longitudinal error (Figure 5d). Consequently, the AVT has to slow down as γ increases. At first glance, reducing V_d seems like the logical solution, however similar behaviour is observed for values of $V_d \in \{0.15, 0.17\}$ [m/s].[2] The root cause is the inherent limitation of $\mathcal{R}$ to accelerate instantaneously, which is also the case in practical applications. Although this does not impact accurate path tracking in the present example, a path which curves immediately after the initial position could result in path tracking inaccuracy.

[1]The virtual target is a point. The triangular shape is for illustration only.

[2]A value of $V_d < 0.15$ [m/s] is inefficient since it limits the ability of $\mathcal{R}$ to operate at its maximum velocity (0.15 [m/s]).

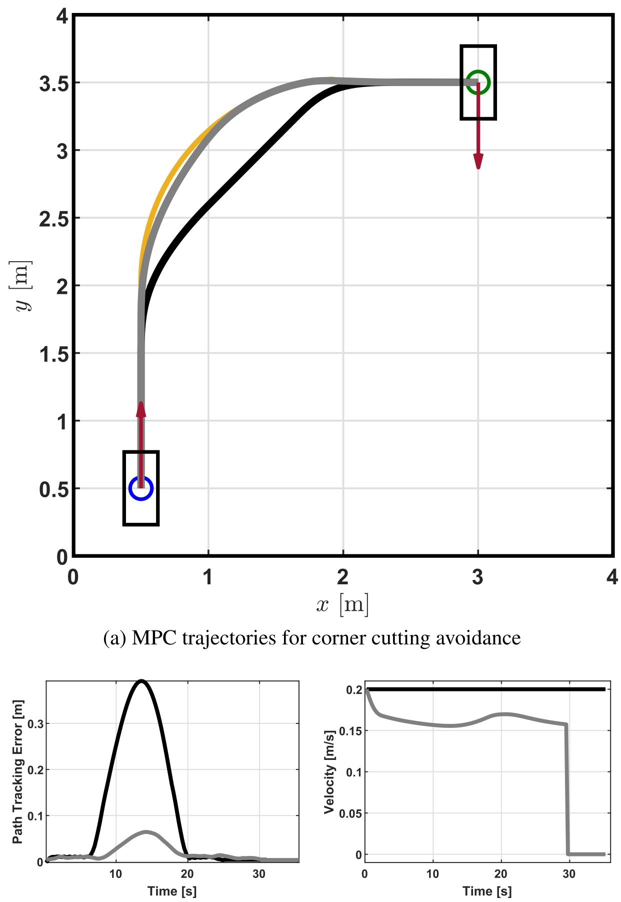
Developing an alternative solution deserves further investigation of the AVT parameters.

2) OBSTACLE AVOIDANCE & POINT STABILIZATION

The robot geometry, i.e., polyhedron $\mathcal{R}$ is provided in Figure 6a for different time instants to illustrate the obstacle avoidance and point stabilization behaviour. The robot trajectory (gray) is planned along the edge of the obstacle safe zone (black dashed) and it is seen that due to obstacle enlargement the robot geometry does not intersect with $\mathcal{O}_1$ (purple). Further along the planned trajectory, after $\mathcal{R}$ has crossed the threshold distance ξ, it successfully converges to the goal configuration. The state evolution of the robot (Figure 6b) indicates that $\mathcal{R}$ is stationary at the final position (v_x, v_y, $\omega = 0$) and is at the desired orientation ($\theta = -\pi/2$).

C. CORNER CUTTING AVOIDANCE

The scenario in Fig. 7 presents the additional advantage of the ATV approach to avoid corner cutting behaviour when tracking a curved path. The elements from the previous scenario are carried over except the existence of $\mathcal{O}_1$, i.e., obstacle-free motion is considered.



(a) MPC trajectories for corner cutting avoidance

(b) Path tracking error

(c) Virtual target velocity

FIGURE 7. Trajectory planning with corner cutting avoidance (*Black and gray curves represent results with constant and adaptive velocity for the virtual target, respectively*).

The virtual target is initially simulated to evolve along the desired path (yellow) at a constant velocity. The resulting trajectory planned by the MPC-based planner in depicted in Figure 7a (black curve). Although the trajectory is planned successfully from start to goal configuration, corner cutting behaviour is observed. This is a consequence of the virtual target being too far ahead of the robot, resulting in a robot trajectory that deviates from the desired path to track the virtual target. This is also reflected in the path tracking error observed in Figure 7b (black). On the contrary, when the AVT is assigned an adaptive velocity according to (9), the MPC-based trajectory (gray) tracks the desired path with less deviation (lower tracking error in Figure 7b). Figure 7c shows that the virtual target slows down for $\mathcal{R}$ to catch up when the distance between them increases. The adaptive velocity approach allows improved convergence to the desired path.

VI. CONCLUSION

This paper presented an approach to prevent deadlock situations during collision-free trajectory planning for autonomous robots. While tracking a desired path defined by waypoints, the existence of an obstacle on a waypoint creates the contradictory objectives of reaching that waypoint while also avoiding the obstacle. This results in a deadlock and is detrimental to the completeness of the trajectory planner. To solve this, a waypoint independent approach in the form of an Adaptive Virtual Target (AVT) is proposed. The AVT moves towards the goal along the desired path irrespective of surrounding obstacles and essentially yields an obstacle-free reference for the robot to track. Additionally, a velocity function adapts the speed of the VTV relative to its distance from the actual robot. Generating collision-free trajectories using the AVT as reference is done using a MPC framework.

Simulation results using a healthcare robot as an example demonstrate the effectiveness of the proposed approach. Results show successful guidance of the robot from start to goal while preventing a deadlock with an obstacle when compared to a baseline example based on waypoints. Since the AVT moves forward irrespective of the obstacle, it eliminates the contradictory nature of obstacle-occupied waypoints. Collision avoidance itself is catered for by positional constraints on the robot in the MPC formulation. A key aspect observed during obstacle avoidance is the adaptive nature of the AVT. During the obstacle avoidance manoeuvre, it slows down for the robot to catch up thus minimizing the robot's deviation from the path. This is significant in case of environments with large obstacles where the robot has to deviate far away from its path during obstacle avoidance. Results are also presented to illustrate an additional benefit of the adaptiveness which prevents the 'corner cutting' behaviour of the robot while tracking a curved path. This is advantageous for leader-follower applications like cooperative driving or drone formation where deviation from the desired path is not tolerated.

Future work is aimed at optimizing the parameters of the adaptive velocity function and considering more complex scenarios like moving between two obstacles or a dynamic obstacle that always tries to block the AVT path.

REFERENCES

[1] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *Int. J. Social Robot.*, vol. 7, no. 2, pp. 137–153, Apr. 2015.
[2] M. Svenstrup, T. Bak, and H. J. Andersen, "Trajectory planning for robots in dynamic human environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4293–4298.
[3] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.
[4] W. H. Chun and N. Papanikolopoulos, "Integrals and series: Special functions," in *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2016, pp. 1605–1626.
[5] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.*, vol. 29, no. 2, pp. 315–378, Mar. 2012.
[6] A. P. Aguiar and A. M. Pascoal, "Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents," *Int. J. Control*, vol. 80, no. 7, pp. 1092–1108, Jul. 2007.
[7] J. J. M. Lunenburg, S. A. M. Coenen, G. J. L. Naus, M. J. G. van de Molengraft, and M. Steinbuch, "Motion planning for mobile robots: A method for the selection of a combination of motion-planning algorithms," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 107–117, Dec. 2016.
[8] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer, 1991.
[9] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[10] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.
[11] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
[12] K. Berntorp, A. Weiss, C. Danielson, I. V. Kolmanovsky, and S. Di Cairano, "Automated driving: Safe motion planning using positively invariant sets," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
[13] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[14] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty—A control perspective," *Eur. J. Control*, vol. 24, pp. 14–32, Jul. 2015.
[15] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory Design*. San Francisco, CA, USA: Nob Hill, 2009.
[16] J. M. Maciejowski, *Predictive Control: With Constraints*. London, U.K.: Pearson, 2002.
[17] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annu. Rev. Control*, vol. 45, pp. 76–86, Jan. 2018.
[18] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.
[19] U. Rosolia, X. Zhang, and F. Borrelli, "Data-driven predictive control for autonomous systems," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 259–286, May 2018.
[20] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Eng. Pract.*, vol. 41, pp. 124–133, Aug. 2015.
[21] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Trans. Intell. Transport. Syst.*, vol. 19, no. 9, pp. 2994–3008, Sep. 2018.
[22] G. Antonelli, T. I. Fossen, and D. R. Yoerger, "Modeling and control of underwater robots," in *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2016, pp. 1285–1306.
[23] D. W. Kim, "Tracking of REMUS autonomous underwater vehicles with actuator saturations," *Automatica*, vol. 58, pp. 15–21, Aug. 2015.
[24] L. Singh and J. Fuller, "Trajectory generation for a UAV in urban terrain, using nonlinear MPC," in *Proc. Amer. Control Conf.*, 2001, pp. 2301–2308.
[25] I. Prodan, R. Bencatel, S. Olaru, J. Borges de Sousa, C. Stoica, and S.-I. Niculescu, "Predictive control for autonomous aerial vehicles trajectory tracking," *IFAC Proc. Volumes*, vol. 45, no. 17, pp. 508–513, 2012.
[26] C. M. Kang, S.-H. Lee, and C. C. Chung, "On-road path generation and control for waypoints tracking," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 36–45, Jul. 2017.
[27] T. Templeton, D. H. Shim, C. Geyer, and S. S. Sastry, "Autonomous vision-based landing and terrain mapping using an MPC-controlled unmanned rotorcraft," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 1349–1356.
[28] Y. Choi, H. Jimenez, and D. N. Mavris, "Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories," *Robot. Auto. Syst.*, vol. 98, pp. 158–173, Dec. 2017.
[29] T. T. Mac, C. Copot, A. Hernandez, and R. De Keyser, "Improved potential field method for unknown obstacle avoidance using UAV in indoor environment," in *Proc. IEEE 14th Int. Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2016, pp. 345–350.
[30] G. Flores, I. Lugo-Cardenas, and R. Lozano, "A nonlinear path-following strategy for a fixed-wing MAV," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2013, pp. 1014–1021.
[31] A. Rucco, A. P. Aguiar, and J. Hauser, "Trajectory optimization for constrained UAVs: A virtual target vehicle approach," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 236–245.
[32] E. Lefeber, J. Ploeg, and H. Nijmeijer, "A spatial approach to control of platooning vehicles: Separating path-following from tracking," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15000–15005, Jul. 2017.
[33] C. Paliotta, E. Lefeber, K. Y. Pettersen, J. Pinto, M. Costa, and J. T. de Figueiredo Borges de Sousa, "Trajectory tracking and path following for underactuated marine vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1423–1437, Jul. 2019.
[34] R. Fahrig, J. Starman, E. Girard, A. Al-Ahmad, H. Gao, N. Kothary, and A. Ganguly, "C-arm CT in the interventional suite: Current status and future directions," in *Cone Beam Computed Tomography*. Boca Raton, FL, USA: CRC Press, 2014, p. 199.
[35] G. M. Ziegler, *Lectures on Polytopes*. New York, NY, USA: Springer, 1995.
[36] C. L. Shih, T.-T. Lee, and W. A. Gruver, "A unified approach for robot motion planning with moving polyhedral obstacles," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 4, pp. 903–915, 1990.
[37] J. Alonso-Mora, J. A. DeCastro, V. Raman, D. Rus, and H. Kress-Gazit, "Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles," *Auto. Robots*, vol. 42, no. 4, pp. 801–824, Apr. 2018.
[38] D. Nešić and L. Grüne, "A receding horizon control approach to sampled-data implementation of continuous-time controllers," *Syst. Control Lett.*, vol. 55, no. 8, pp. 660–672, Aug. 2006.
[39] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, Aug. 2014.
[40] T. Schouwenaars, A. Stubbs, J. Paduano, and E. Feron, "Multivehicle path planning for nonline-of-sight communication," *J. Field Robot.*, vol. 23, nos. 3–4, pp. 269–290, 2006.
[41] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2008.
[42] R. Van Parys and G. Pipeleers, "Distributed MPC for multi-vehicle systems moving in formation," *Robot. Auto. Syst.*, vol. 97, pp. 144–152, Nov. 2017.
[43] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," 2017, *arXiv:1711.03449*. [Online]. Available: http://arxiv.org/abs/1711.03449
[44] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, Feb. 1996.
[45] J. Lygeros, "Lecture notes on hybrid systems," Autom. Control Lab., ETH Zürich, Zürich, Switzerland, Tech. Rep. AUT06-08, Dec. 2006.

[46] F. Kuhne, W. F. Lages, and J. G. Da Silva, "Point stabilization of mobile robots with nonlinear model predictive control," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, vol. 3, Jul. 2005, pp. 1163–1168.

[47] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 502–510.

RISHI MOHAN received the B.E. degree in mechanical engineering from the Vishwakarma Institute of Technology, Pune, India, and the M.Sc. degree (Hons.) in mechanical engineering from the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, where he is currently pursuing the Ph.D. degree with the Control Systems Technology Section. His research interests include sensing technologies, object detection, and collision-free motion planning for autonomous robots.

EMILIA SILVAS received the B.Sc. degree in automatic control and computer science from the Politehnica University of Bucharest, Bucharest, Romania, in 2009, and the M.Sc. degree in systems and control and the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2011 and 2015, respectively. Since 2016, she has been a Research Scientist with the Netherlands Organization for Applied Scientific Research (TNO), Helmond, The Netherlands, working in the areas of cooperative automated vehicle systems and mobile robots. In 2017, she was an Assistant Professor with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology. Her research interests include advanced control, system identification and modeling, artificial intelligence techniques, and optimal system design.

HENRY STOUTJESDIJK received the M.Sc. degree in advanced mechatronics from the Delft University of Technology, Delft, The Netherlands. He was with the Philips Centre for Production Technology, Eindhoven, The Netherlands. He is currently with the Mechatronics Development Cluster, Philips Medical Systems, PC Best, The Netherlands. His research interests include mechatronics, robust control of (nonlinear) mechanical systems, and interaction of robotic systems with the environment, especially in the area of medical applications.

HERMAN BRUYNINCKX received the master's degrees in mathematics (Licentiate), computer science (Burgerlijk Ingenieur), and mechatronics, and the Ph.D. degree in engineering from KU Leuven, Belgium, in 1984, 1987, 1988, and 1995, respectively. His thesis entitled Kinematic Models for Robot Compliant Motion with Identification of Uncertainties. He held a Visiting Research positions with the Grasp Laboratory, University of Pennsylvania, Philadelphia, in 1996, the Robotics Laboratory, Stanford University, in 1999, and the Kungl Tekniska Hogskolan, Stockholm, in 2002. In 2001, he started the Free Software (open source) Project Orocos to support his research interests and to facilitate their industrial exploitation. Since 2014, he has been a Part-Time Affiliation with the Eindhoven University of Technology. He is currently a Full-Time Professor with KU Leuven. He has participated in about a 12 European research projects on robotics, with a focus in the recent years on the software engineering aspects. His current research interests include on-line Bayesian estimation of model uncertainties in sensor-based robot tasks, kinematics and dynamics of robots and humans, and the software engineering of large-scale robot control systems. In October 2014, he received the Honorary Doctorate from the University of Southern Denmark, for his leading role in software development and research in robotics.

BRAM DE JAGER received the M.Sc. degree in mechanical engineering from the Delft University of Technology, Delft, The Netherlands, and the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands. He was with the Delft University of Technology and Stork Boilers BV, Hengelo, The Netherlands. He is currently with the Eindhoven University of Technology. His research interests include robust control of (nonlinear) mechanical systems, integrated control and structural design, control of fluidic systems, control structure design, and applications of (nonlinear) optimal control.