Autonomous navigation with convergence guarantees in complex dynamic environments

Citation for the original published paper (version of record):
Dahlin, A., Karayiannidis, Y. (2025). Autonomous navigation with convergence guarantees in complex dynamic environments. Automatica, 173.
http://dx.doi.org/10.1016/j.automatica.2024.112026

Automatica 173 (2025) 112026

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Autonomous navigation with convergence guarantees in complex dynamic environments[☆]

Albin Dahlin [a,*], Yiannis Karayiannidis [b,1]

[a] Department of Electrical Engineering, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden
[b] Department of Automatic Control, LTH, Lund University, SE-221 00, Lund, Sweden

ARTICLE INFO

Article history:
Received 28 June 2023
Received in revised form 4 August 2024
Accepted 25 October 2024
Available online xxxx
Communicated by T. Parisini

Keywords:
Guidance navigation and control
Autonomous systems
Modeling for control optimization
Control of constrained systems

ABSTRACT

This article addresses the obstacle avoidance problem for setpoint stabilization tasks in complex dynamic 2-D environments that go beyond conventional scenes with isolated convex obstacles. A combined motion planner and controller is proposed that integrates the favorable convergence characteristics of closed-form motion planning techniques with the intuitive representation of system constraints through Model Predictive Control (MPC). The method is analytically proven to accomplish collision avoidance and convergence under soft conditions. Simulation scenarios using a non-holonomic unicycle robot is provided to showcase the efficacy of the control scheme.

1. Introduction

As autonomous agents, or robots, are increasingly employed in dynamically changing environments, the need for sensor-based motion controllers able to react to unforeseen circumstances is prominent. To achieve successful online navigation in such environments, a key aspect is to adaptively modify the constraints imposed by the robot's surroundings, possibly involving the presence of moving obstacles. A vast part of the literature in online obstacle avoidance are based either on closed-loop or optimization based control solutions, where specific requirements on the obstacle shapes are imposed and cases of intersecting obstacles are ignored. However, closely positioned obstacles are frequently perceived as intersecting, e.g., when inflation is used to account for robot radius or safety margins, or in case of perception uncertainties. Breaking the conditions of disjoint obstacles yields local minima, jeopardizing convergence to the desired goal or path. In this work we combine the convergence properties of closed-form Dynamical Systems (DS) and the intuitive encoding of system constraints for Model Predictive Control (MPC) to propose a holistic control solution with guaranteed convergence also in scenarios of nontrivial obstacle constellations.

Expressing the desired robot motion in terms of DS is a computationally efficient strategy to generate the subsequent motion in response to current observations. Specifically, artificial potential fields (APF) (Khatib, 1985), repelling the robot from the obstacles, are widely used (Ginesi, Meli, Calanca, Dall'Alba, Sansonetto, & Fiorini, 2019; Stavridis, Papageorgiou, & Doulgeri, 2017). A well-known drawback of APF is the possible occurrences of local minimum other than the goal point. To address this problem, navigation functions (Kumar, Paternain, & Ribeiro, 2022; Loizou, 2011; Rimon & Koditschek, 1992) and harmonic potential fields (Connolly, Burns, & Weiss, 1990; Feder & Slotine, 1997) have emerged. Collision avoidance with (almost) global convergence can also be ensured by modulating some nominal linear dynamics using a modulation matrix (Huber, Billard, & Slotine, 2019; Huber, Slotine, & Billard, 2022). A repeated assumption in the aforementioned works enabling the proof of (almost) global convergence is the premise of the environment being a *disjoint star world* (DSW), i.e. all obstacles are starshaped and mutually disjoint.[2] However, intersecting obstacles are frequently occurring, e.g., when modeling complex obstacles as a combination of several simpler shapes, or when the obstacle regions are padded to take robot radius or safety margins into account. To handle intersecting circular obstacles, Daily and Bevly (2008) proposed weighted average of harmonic functions, but unwanted local

[☆] This work was supported in part by the Chalmers AI Research Centre (CHAIR) and AB Volvo through the project AiMCoR and in part by the ELLIIT Strategic Research Area. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor C.C. Cheah under the direction of Editor Thomas Parisini.

[*] Corresponding author.
E-mail addresses: albinjdahlin@gmail.com (A. Dahlin), yiannis@control.lth.se (Y. Karayiannidis).

[1] Y. Karayiannidis is a member of the ELLIIT Strategic Research Area at Lund University.

[2] See Section 2.2 for complete definition.

https://doi.org/10.1016/j.automatica.2024.112026

minima still occurred and the authors recommended to keep the number of combined obstacles low. In Dahlin and Karayiannidis (2023a) we presented a method, here referred to as ModEnv*, which modifies the robot environment into a DSW. This effectively extends the applicable scenarios where the aforementioned DS methods achieve convergence properties. The approach was limited to the case with a robot operating in the full Euclidean space and no conditions for successful generation of a DSW were provided. In Dahlin and Karayiannidis (2023b) we presented a motion control scheme for setpoint stabilization with collision avoidance consisting of three main components: environment modification into a DSW, DS-based generation of a receding horizon reference path (RHRP), and an MPC to compute admissible control inputs to drive the robot along the RHRP. Whereas collision avoidance is ensured, no guarantees for convergence were provided. Convergence may be inhibited by two situations; (1) the modified environment is not a DSW such that convergence guarantees for the DS method are lost, (2) the MPC solution does not provide a movement of the robot along the RHRP due to limited control horizon and robot constraints.

In this work, we expand upon the control scheme introduced in Dahlin and Karayiannidis (2023b) to enable the derivation of convergence properties and to facilitate its implementation within confined workspaces. As convergence rely on environment modification into a DSW, we first derive sufficient conditions to obtain a DSW for ModEnv* (Algorithm 2 in Dahlin and Karayiannidis (2023a)). Additionally, the method is enhanced to treat also the case of confined workspaces. In contrast to a pure closed form approach, embedding the DS in an MPC scheme allows for simple adaptation of the robot constraints to find an admissible and smooth control input. Compared to other MPC approaches, collision avoidance is here achieved by relying on a reference path generator, simplifying the formulation of the optimal control problem to be independent of workspace complexity. Moreover, the global convergence is achieved despite the receding horizon nature of MPC schemes. In all, the main contributions are:

- Extension of ModEnv* to allow for confined workspaces and derivation of sufficient conditions to successfully obtain a DSW.
- A control scheme with derivation of sufficient conditions ensured for collision avoidance and convergence to a goal point.

2. Preliminaries

2.1. Notation

Let $A = \{A^1, A^2, \ldots\}, A^i \in \mathbb{R}^d$ be a collection of sets. The union and intersection of A are denoted by $A_\cup = \bigcup_{A^i \in A} A^i$ and $A_\cap = \bigcap_{A^i \in A} A^i$, respectively. If all sets $A^i \in A$ are starshaped, the kernel intersection is denoted by $\ker_\cap(A) = \bigcap_{A^i \in A} \ker(A^i)$. For convenience, an improper use of the Minkowski sum, $\oplus$, will be applied as follows: $A \oplus B = \{A^i \oplus B\}_{\forall A^i \in A}$, given $B \in \mathbb{R}^d$. The closest distance between two sets, A^1 and A^2, is denoted by $\mathrm{dist}(A^1, A^2)$. $\mathbb{B}(a, b)$ and $\mathbb{B}[a, b]$ are the open and closed balls of radius b centered at a, respectively. The line segment from point x to point y is denoted by $l[x, y]$. A robot workspace $\mathcal{W} \subset \mathbb{R}^2$ and a collection of obstacles $\mathcal{O} = \{\mathcal{O}^1, \mathcal{O}^2, \ldots\}$ in $\mathbb{R}^2$ are jointly called the robot environment, denoted by $E = \{\mathcal{W}, \mathcal{O}\}$. The corresponding free set is denoted by $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}_\cup$.

2.2. Starshaped sets and star worlds

A set A is *starshaped with respect to* (w.r.t.) x if for every point $y \in A$ the line segment $l[x, y]$ is contained by A. The set A is said to be *starshaped* if it is starshaped w.r.t. some point, i.e. $\exists x$ s.t. $l[x, y] \subset A, \forall y \in A$. The set of all such points is called the *kernel of* A and is denoted by $\ker(A)$, i.e. $\ker(A) = \{x \in A : l[x, y] \subset A, \forall y \in A\}$. For any convex set A we have $\ker(A) = A$. The set A is *strictly starshaped w.r.t. x* if it is starshaped w.r.t. x and any ray emanating from x crosses the boundary only once. We say that A is strictly starshaped if it is strictly starshaped w.r.t. some point. The starshaped hull of A with specified kernel K, $SH_{\ker K}(A)$, is the smallest starshaped set such that $A \subset SH_{\ker K}(A)$ and $K \subset \ker(SH_{\ker K}(A))$.

The robot environment $E = \{\mathcal{W}, \mathcal{O}\}$ is said to be a *star world* if all obstacles are strictly starshaped, and the workspace is strictly starshaped or the full Euclidean space. A *disjoint star world* (DSW) refers to a star world where all obstacles are mutually disjoint and where any obstacle which is not fully contained in the workspace has a kernel point in the exterior of the workspace, as exemplified in Fig. 1(b). For more information on starshaped sets and star worlds, see Hansen, Herburt, Martini, and Moszyńska (2020) and Dahlin and Karayiannidis (2023a).

2.3. Obstacle avoidance for dynamical systems in star worlds

Given a star world, E, collision avoidance can be achieved using a DS approach (Huber et al., 2022) with dynamics:

$$\dot{r} = \eta(r, r^g, E) = M(r, E)(r^g - r), \tag{1}$$

where r is the current robot position and $r^g \in \mathcal{F}$ is the goal position. $M(\cdot)$ is a modulation matrix used to adjust the attracting dynamics to r^g based on the obstacles tangent spaces. Convergence to r^g is guaranteed for a trajectory following (1) from any initial position, $r^0 \in \mathcal{F}$, if E is a DSW and no obstacle center point is contained by the line segment $l[r^0, r^g]$. For more information, see Huber et al. (2019, 2022).

3. Problem formulation

Consider an autonomous agent with dynamics

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ p(t) &= h(x(t)), \end{aligned} \tag{2}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ is the robot state, $p \in \mathbb{R}^2$ is the robot position and $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control signal. It is assumed that there exists a control input such that the robot does not move, i.e. $\exists u' \in \mathcal{U}$ s.t. $f(x, u') = 0, \ \forall x \in \mathcal{X}$. The robot is operating in a dynamic environment, $E(t) = \{\mathcal{W}(t), \mathcal{O}(t)\}$, where each dynamic obstacle is convex and each static obstacle is either convex or a simple polygon. The workspace is either strictly starshaped or the full Euclidean plane.

Remark 1. Although $\mathcal{O}$ formally contains only polygons and convex shapes, the formulation allows for more general complex obstacles as intersections are allowed. In particular, any shape can be described as a combination of several polygon and/or convex regions.

The robot body is modeled as a point, assuming that any robot radius is taken into account by the environment model. The following assumptions are stated for the environment.

Assumption 2. The workspace does not change such that the current robot position becomes an exterior point of the workspace, i.e., $\exists \delta t > 0, \ p(t) \in \mathcal{W}(t) \Rightarrow p(t) \in \mathcal{W}(t'), \ \forall t' \in [t, t + \delta t], \forall t$.

Assumption 3. The complete shape of each obstacle is known by the robot as well as the current position and velocity. The velocity is constant over a sampling interval Δt.

Assumption 4. The obstacles do not actively move into a region currently occupied by the robot, such that the implication $p(t) \in \mathcal{F}(t) \Rightarrow p(t) \in \mathcal{F}(t + \Delta t)$ holds.

Allowing a time-varying workspace enables considering also cases with non-starshaped confined workspaces as exemplified in Section 6. However, Assumption 2 restricts the workspace changes to enable sound motion planning. Assumption 3 suggests complete obstacle recognition and that a constant velocity model is used. Assumption 4 implies that the obstacles are aware of the robot's position and are not hostile, thereby preventing any collisions while the robot remains stationary.

The objective is to find a control policy, $u(t) \in \mathcal{U}$, that drives a robot with dynamics (2) to a specified goal position, $p^g \in \mathbb{R}^2$, while avoiding collision in the environment, $E(t)$, i.e., $p(t) \in \mathcal{F}(t), \forall t$.

In the following sections, we will omit the time notation for convenience unless some ambiguity exists.

4. Guaranteed DSW generation

The obstacle avoidance approach laid out in Section 5 is dependent on ModEnv*, developed in Dahlin and Karayiannidis (2023a), to generate a DSW. No guarantee for successfully obtaining a DSW was however derived, preventing convergence guarantees to be established for the proposed control scheme. Moreover, the workspace was assumed to be the full Euclidean space, ignoring situations where the workspace is bounded. Here, we extend ModEnv* to address both these issues.

At a high level, ModEnv* divides all obstacles into clusters of intersecting obstacles, followed by constructing a single starshaped obstacle for each cluster, *cl*, that fully encloses the cluster. This is done using starshaped hull with specified kernel. An important step is the selection of kernel points, such that neither robot nor goal position are included in the created obstacle regions. This is ensured by following Algorithm 3 in Dahlin and Karayiannidis (2023a) for kernel point selection. The selection is based on computing a restricted set, S, from where to select the specified kernel. Here, in Algorithm 1, we extend this restriction in two ways: (1) in scenarios with confined workspace, the kernel selection is restricted to the workspace exterior for any cluster which is not fully contained in the workspace, if possible (line 2–3), (2) the kernel selection is restricted to the intersecting kernel region of the clustered obstacles, if possible (lines 4–5). For more details on the full procedure, the reader is referred to Dahlin and Karayiannidis (2023a).

Algorithm 1 Kernel point selection for ModEnv*

```
1: Assign $S$ according to line 1 in Algorithm 3 of Dahlin and Karayiannidis (2023a)
2: if not $cl_\cup \subset \mathcal{W}$ and $S \setminus \mathcal{W} \neq \emptyset$ then
3:     $S \leftarrow S \setminus \mathcal{W}$
4: if $\forall \mathcal{O}^i \in cl$ starshaped and $S \cap \ker_\cap(cl) \neq \emptyset$ then
5:     $S \leftarrow S \cap \ker_\cap(cl)$
6: else if $S \cap cl_\cup \neq \emptyset$ then
7:     $S \leftarrow S \cap cl_\cup$
8: Lines 4-13 in Algorithm 3 of Dahlin and Karayiannidis (2023a)
```

To declare a sufficient condition for DSW generation, the following definition of a *DSW equivalent* set is established.

Definition 5 (*DSW Equivalent*)**.** A star world is DSW equivalent if the set, Cl, formed by partitioning $\mathcal{O}$ into mutually disjoint clusters of obstacles, satisfies

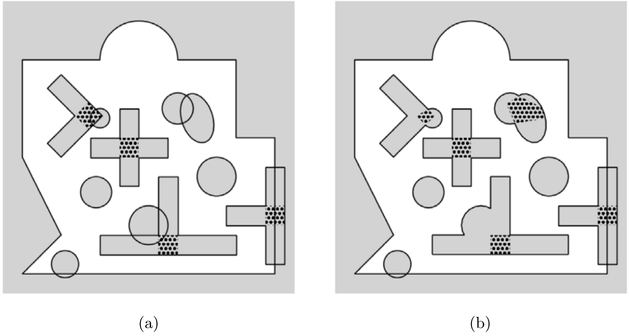

(a)
(b)

Fig. 1. A DSW equivalent environment (a) and the corresponding DSW environment (b). The kernels for all non-convex obstacles are shown as dotted regions.

(i) the obstacles in each cluster have intersecting kernels,

$$\ker_\cap(cl) \neq \emptyset, \ \ \forall cl \in Cl, \tag{3}$$

(ii) any cluster which intersects with the workspace exterior has an intersecting kernel region that to some extent lies in the workspace exterior,

$$cl_\cup \cap \operatorname{ext}\mathcal{W} \neq \emptyset \Rightarrow \ker_\cap(cl) \cap \operatorname{ext}\mathcal{W} \neq \emptyset, \quad \forall cl \in Cl. \tag{4}$$

An example of a DSW equivalent scene is shown in Fig. 1(a). Note that if all obstacles are convex, it holds that $\ker_\cap(cl) = cl_\cap$. With the adjusted kernel point selection, a sufficient condition to establish a DSW can be presented as stated in the following theorem.

Theorem 6 (*Guaranteed DSW Generation*)**.** *Consider a DSW equivalent environment with free space $\mathcal{F}$, a robot position, $p \in \mathcal{F}$, and a goal position, $p^g \in \mathcal{F}$. The environment, $\{\mathcal{W}, \mathcal{O}^\star\}$, resulting from ModEnv* with kernel point selection as in Algorithm 1 is a DSW with $\mathcal{O}^\star_\cup = \mathcal{O}_\cup$.*

Proof. See Appendix A.

5. Control scheme

In this section, a control scheme for setpoint stabilization with obstacle avoidance is proposed. The scheme is divided into four main components, as depicted in Fig. 2, where the important steps are illustrated in Fig. 3. The environment is modified to form a DSW, $E^\star$, where any free point has an appropriately selected minimum clearance, ρ, to the obstacles and workspace boundary (Section 5.1). This enables generation of a *receding horizon reference path* (RHRP), $\mathcal{P}$, based on (1) to ensure collision clearance and convergence to the goal (Section 5.2). A control sequence, $\bar{u}^*$, is computed using an MPC which yields a robot movement along the RHRP within the specified clearance to ensure collision avoidance (Section 5.3). To provide guaranteed forward motion, initial movement of the reference is enforced in the MPC formulation. As a consequence, there may be occasions where the MPC problem is infeasible for non-holonomic robots. To handle this, a backup control law is formulated (Section 5.4) and a switching control law is defined (Section 5.5) yielding the control policy, μ, applied by the controller over the following sampling period. The complete control scheme is outlined in Section 5.6 where collision avoidance and convergence properties are also analyzed.

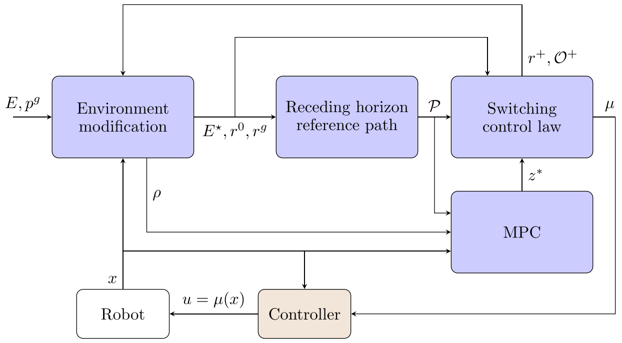
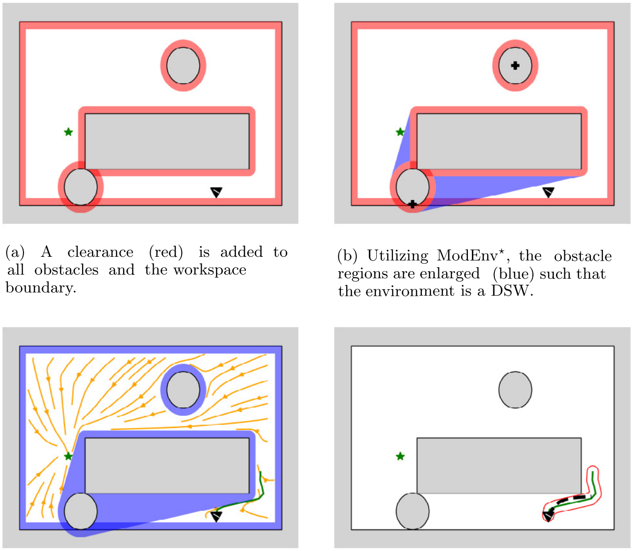


Fig. 2. Proposed motion control scheme for setpoint stabilization.



(a) A clearance (red) is added to all obstacles and the workspace boundary.

(b) Utilizing ModEnv*, the obstacle regions are enlarged (blue) such that the environment is a DSW.

(c) The RHRP (green line) is obtained by following the vector field of (7) (yellow arrows).

(d) A control sequence yielding a robot trajectory (dashed black) that stays within a ρ-neighborhood of the RHRP is computed using an MPC.

Fig. 3. Illustration of the main steps: (a–b) Environment modification, (c) RHRP, (d) MPC.

5.1. Environment modification

First, the obstacle set is adjusted to account for the dynamic obstacles. Given Assumption 3, $\mathcal{O}(t + \Delta t)$ can be predicted by translation corresponding to the obstacle velocities. Hence, the inflated obstacles $\hat{\mathcal{O}}(t) = \{CH(\{\mathcal{O}^i(t), \mathcal{O}^i(t+\Delta t)\}) : \mathcal{O}^i \in \mathcal{O}\}$ represent the regions occupied by the obstacles during the time period $[t, t+\Delta t]$. The proposed method relies on generating the RHRP with a (time-varying) minimum clearance, ρ, to all obstacles using the DS approach (1). To this end, the clearance environment $E^\rho = \{\mathcal{W}^\rho, \mathcal{O}^\rho\}$ is defined, where $\mathcal{W}^\rho = \mathcal{W} \ominus \mathbb{B}[0,\rho]$ and $\mathcal{O}^\rho = \hat{\mathcal{O}} \oplus \mathbb{B}[0,\rho]$, with corresponding clearance set $\mathcal{F}^\rho = \mathcal{W}^\rho \setminus \mathcal{O}^\rho_\cup$. As stated in Section 2.3, any star world is positively invariant for the dynamics (1) and convergence to a goal position is guaranteed for a DSW. Since E^ρ may include both intersecting and non-starshaped obstacles, it provides none of the aforementioned guarantees. The objective of the environment modification is therefore to find a DSW $E^\star = \{\mathcal{W}^\rho, \mathcal{O}^\star\}$ with corresponding free set $\mathcal{F}^\star \subset \mathcal{F}^\rho$, as well as initial and goal positions, $r^0 \in \mathcal{F}^\star$ and $r^g \in \mathcal{F}^\star$, for the RHRP. A procedure to specify ρ and to compute $E^\star$, r^0 and r^g is given in Algorithm 2 and the steps are elaborated below.

Initial and goal reference position selection (lines 6–8):

The initial reference position, r^0, is chosen as the point closest to an input candidate, r^+, within the initial reference set $\mathcal{P}^0 = \mathcal{F}^\rho \cap \mathbb{B}[p,\rho]$. In this way, the distance from r^0 to any obstacle and workspace boundary is greater than ρ, while the distance to the robot is less than or equal to ρ. As specified in Section 5.5, r^+ is appropriately selected along the previously computed RHRP. In particular, r^+ is chosen to stimulate a forward shift of the RHRP towards the goal, compared to the previous sampling instance. The reference goal, r^g, is chosen as the point in $\mathcal{F}^\rho$ closest to p^g.

Clearance selection (lines 1–5):

To have a valid initial reference position, ρ is set to a strict positive value such that $\mathcal{P}^0$ is nonempty. This is done by utilizing the equivalence

$$p \in \mathcal{C}^\rho \Leftrightarrow \mathcal{P}^0 \neq \emptyset, \quad \mathcal{C}^\rho = \mathcal{F}^\rho \oplus \mathbb{B}[0,\rho]. \tag{5}$$

For robot positions $p \notin \mathcal{C}^{\bar{\rho}}$, i.e. when the default selection $\rho = \bar{\rho}$ yields $\mathcal{P}^0 = \emptyset$, the clearance is reduced to $\rho = \gamma \operatorname{dist}(\partial\mathcal{F}, p)$ to ensure $p \in \mathcal{F}^\rho$ and thus $\mathcal{P}^0 \neq \emptyset$ according to (5). This is a conservative reduction of ρ since larger values could in many cases be used while still obtaining $\mathcal{P}^0 \neq \emptyset$. The procedure is illustrated Fig. 4.

Establishment of a DSW (lines 9–12):

To obtain a DSW such that $\mathcal{F}^\star \subset \mathcal{F}^\rho$, $r^0 \in \mathcal{F}^\star$ and $r^g \in \mathcal{F}^\star$, a first attempt is to use an input candidate set of starshaped obstacles, $\mathcal{O}^+$. If this does not satisfy the conditions for $\mathcal{F}^\star$, the obstacles are computed using ModEnv*. As specified in Section 5.5, $\mathcal{O}^+$ is the previously computed $\mathcal{O}^\star$ under the condition that the corresponding environment is a DSW.

Convexification (lines 13–15):

To avoid unnecessary “detours” in concave obstacle regions, see Dahlin and Karayiannidis (2023b), the generated obstacles, $\mathcal{O}^\star$, are made convex provided that the following conditions are not violated: (1) r^0 and r^g remain exterior points of the obstacle, and (2) the resulting obstacle region does not intersect with any other obstacle. Due to these conditions, any DSW $E^\star$ remains a DSW also after convexification.

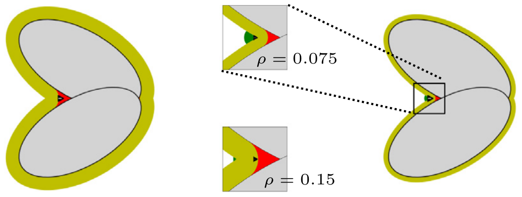
In addition to the revised kernel point selection in Algorithm 1, the environment modification is adjusted compared to Dahlin and Karayiannidis (2023b) in four ways: (1) an initial inflation of the obstacles corresponding to the obstacle velocity is included to handle constant velocity models during each sampling period, rather than constant position, (2) the clearance, ρ, is determined in a one step check ensuring $\mathcal{P}^0 \neq \emptyset$ as opposed to using an iterative approach, (3) the selection of the initial reference point, r^0, is based on an input candidate, r^+, rather than the robot position p, and (4) an input set of starshaped obstacles, $\mathcal{O}^+$, is used if feasible instead of recalculating $\mathcal{O}^\star$. The first adjustment extends the applicability to scenarios assuming constant velocity obstacles during the sampling period instead of constant position, while the second is a pure simplification of the algorithm. The two last adjustments are instrumental to obtain the convergence properties derived in Section 5.6. Moreover, attempting to set $\mathcal{O}^\star = \mathcal{O}^+$, i.e. remaining $\mathcal{O}^\star$ constant over sampling instances, lead to a higher degree of consistency over time for the vector field associated with (1), which is utilized for generating the RHRP. By additionally aiming to initialize the RHRP along the previously computed RHRP facilitates smoother shifts of the path between control sampling instances.

5.2. DS-based receding horizon reference path

The RHRP is given as a parameterized regular curve

$$\mathcal{P} = \left\{ r \in \mathbb{R}^2 : s \in [0, L] \to r(s) \right\} \tag{6}$$

with $L = T w^{\max}$. Here, T is the MPC horizon described in Section 5.3 and $w^{\max} = \max_{u \in \mathcal{U}, x \in \mathcal{X}} \| \frac{\partial h}{\partial x}(x) f(x,u) \|_2$ is the maximum



(a) The default clearance $\rho = \bar{\rho} = 0.2$ yields $\mathcal{P}^0 = \emptyset$ since $p \notin \mathcal{C}^{\bar{\rho}}$.

(b) The clearance is reduced to $\rho = \gamma\mathrm{dist}(\partial\mathcal{F}, p)$ to ensure $\mathcal{P}^0 \neq \emptyset$. The less conservative reduction $\rho = 0.15$ would also be a valid selection since this still yields $\mathcal{P}^0 \neq \emptyset$.

Fig. 4. Two obstacles $\hat{\mathcal{O}}$ (gray) with inflated regions $\mathcal{O}^\rho$ (yellow), the initial reference set $\mathcal{P}^0$ (green), the region $\mathcal{F}\backslash\mathcal{C}^\rho$ (red) corresponding to robot positions where $\mathcal{P}^0 = \emptyset$, and the robot position p (black triangle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2 Environment modification

```
Parameters: $\gamma \in (0, 1)$, $\bar{\rho} \in \mathbb{R}^+$
Input: $\mathcal{W}$, $\mathcal{O}$, $p^g$, $p$, $r^+$, $\mathcal{O}^+$
Output: $\mathcal{W}^\rho$, $\mathcal{O}^\star$, $r^0$, $r^g$, $\rho$
if $p \in \mathcal{C}^{\bar{\rho}}$ then
    $\rho \leftarrow \bar{\rho}$
else
    $\rho \leftarrow \gamma\,\mathrm{dist}(\partial\mathcal{F}, p)$
$\{\mathcal{O}^\rho, \mathcal{W}^\rho\} \leftarrow \{\mathcal{O} \oplus \mathbb{B}[0, \rho], \mathcal{W} \ominus \mathbb{B}[0, \rho]\}$
$\mathcal{P}^0 \leftarrow \mathcal{F}^\rho \cap \mathbb{B}[p, \rho]$
$r^0 \leftarrow \mathrm{argmin}_{r^0 \in \mathcal{P}^0} \|r^0 - r^+\|_2$
$r^g \leftarrow \mathrm{argmin}_{r^g \in \mathcal{F}^\rho} \|r^g - p^g\|_2$
if $r^0 \in \mathcal{F}^+$ and $r^g \in \mathcal{F}^+$ and $\mathcal{F}^+ \subset \mathcal{F}^\rho$ then
    $\mathcal{O}^\star \leftarrow \mathcal{O}^+$
else
    Compute $\mathcal{O}^\star$ using ModEnv$^\star$ with $\mathcal{W}^\rho$, $\mathcal{O}^\rho$, $r^0$ and $r^g$ as input
for $\mathcal{O}^j \in \mathcal{O}^\star$ do
    if $CH(\mathcal{O}^j) \cap \left\{r^0 \cup r^g \cup \left(\mathcal{O}^\star \setminus \mathcal{O}^j\right)_\cup\right\} = \emptyset$ then
        $\mathcal{O}^j \leftarrow CH(\mathcal{O}^j)$
```

linear speed which can be achieved by the robot. The mapping r is given by the solution to the ODE

$$\frac{dr(s)}{ds} = \bar{\eta}(r(s), r^g, E^\star), \quad r(0) = r^0, \tag{7}$$

where $\bar{\eta}(\cdot) = \frac{\eta(\cdot)}{\|\eta(\cdot)\|_2}$ are the normalized dynamics in (1). As the path is initialized in the star world $\mathcal{F}^\star$ and the dynamics are positively invariant in any star world, we have $\mathcal{P} \subset \mathcal{F}^\star \subset \mathcal{F}^\rho$. Thus, the tunnel-region $\mathcal{P}^\rho = \mathcal{P} \oplus \mathbb{B}[0, \rho]$ is in the free set, $\mathcal{F}$.

5.3. Model predictive controller

In Dahlin and Karayiannidis (2023b), an MPC is used to compute a control input driving the robot along the RHRP. Whereas collision avoidance is proven, local attractors away from the goal may arise in the workspace depending on control horizon and robot constraints. To improve attracting behavior towards the goal and derive convergence conditions, we here introduce an *enforced initial forward motion* of the reference position resulting in the following MPC.

Adhering the path-following MPC framework (Faulwasser & Findeisen, 2016), the system state and input are augmented with path coordinate, $s \in [0, L]$, and path speed, $w \in [0, w^{\max}]$, respectively. This embeds the reference trajectory $r(s)$ as part of the optimization problem. The bounds on path variables ensure valid mapping $r(s)$ for all admissible s and that the reference moves in forward direction along $\mathcal{P}$ with a reference speed, $\|\dot{r}\|_2 = \|\frac{dr(s)}{ds}w\|_2$, less than or equal to the maximum linear speed of the robot, $w^{\max}$. Similar to the tunnel-following MPC (van Duijkeren, 2019), a constraint is imposed on the tracking error, $\varepsilon(\tau) = \|r(s(\tau)) - h(\bar{x}(\tau))\|_2$ such that the robot position is in a ρ-neighborhood of the reference position.[3] As standard in the MPC framework, the control variables are piecewise continuous over a sampling interval Δt and are computed over a horizon $T = N\Delta t$, with $N \in \mathbb{N}^+$. The optimization problem for the MPC to find the control sequence, $\{\bar{u}_i,\ w_i : i \in [0, 1, \ldots, N-1]\}$, is proposed as

$$\min_{\bar{u}, w} \int_0^T -c_w w(\tau) + c_e\varepsilon(\tau)d\tau + J(\bar{u}) \tag{8a}$$

$$\text{subject to} \tag{8b}$$

$$\tau \in [0, T]: \quad \dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)),\ \ \bar{x}(0) = x(t) \tag{8c}$$

$$\dot{s}(\tau) = w,\ \ s(0) = 0, \tag{8d}$$

$$\bar{x}(\tau) \in \mathcal{X},\ \ \bar{u}(\tau) \in \mathcal{U}, \tag{8e}$$

$$s(\tau) \in [0, L],\ \ w(\tau) \in [0, w^{\max}], \tag{8f}$$

$$\bar{u}(\tau) = \bar{u}_i,\ \ w(\tau) = w_i,\ \ i = \left\lfloor \frac{\tau}{\Delta t} \right\rfloor, \tag{8g}$$

$$\varepsilon(\tau) \leq \rho, \tag{8h}$$

$$w_0 \geq \frac{\lambda\rho}{\Delta t}. \tag{8i}$$

Here, the notation $\bar{x}$ and $\bar{u}$ is used to denote the internal variables of the controller and distinguish them from the real system variables. The scalars $c_w > 0$, $c_e > 0$ and $\lambda \in (0, 1)$ are tuning parameters, and $J(\bar{u})$ is a regularization term for the control input which can be tailored for the robot at hand, if desired. To ensure that the upper and lower bound on w_0 do not conflict, the relationship $\lambda\bar{\rho} \leq w^{\max}\Delta t$ must be satisfied. The inclusion of enforced initial forward motion of the reference position (8i) is key when deriving the convergence properties in Section 5.6.

5.4. Stabilizing backup controller

It can be shown that the MPC problem (8) without constraint (8i) is feasible at all times by following the proof of Theorem 1 in Dahlin and Karayiannidis (2023b). With the constraint (8i), existence of solution is however no longer guaranteed. Consider for instance the example with a non-holonomic robot in Fig. 5 where the robot position is outside the region $\mathbb{B}[r(\lambda\rho), \rho]$. Depending on robot constraints, forcing an initial displacement such that $s(\Delta t) = \lambda\rho$ may lead to $\varepsilon(\Delta t) > \rho$, violating constraint (8h). To handle these cases, a fallback strategy is here presented.

Let $\mathcal{K} \subset \mathcal{X} \times \mathbb{R}^2 \to \mathcal{U}$ be a family of control laws such that any $\kappa \in \mathcal{K}$:

- renders the closed-loop error dynamics $\dot{e}$ asymptotically stable in the origin, for the error $e(t) = p(t) - r^0$ given r^0,
- does not allow the error to exceed its initial value, i.e., $\|e(t)\|_2 \leq \|e(t_0)\|_2,\ \forall t \geq t_0$.

We will refer to any $\kappa \in \mathcal{K}$ as a *stabilizing backup controller* (SBC).

Unicycle example: Obviously, the SBC needs to be designed for the robot at hand, but an example is here presented for a unicycle robot kinematic model

$$f(x, u) = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \omega \end{bmatrix}, \quad h(x) = \begin{bmatrix} p^x \\ p^y \end{bmatrix}, \tag{9}$$

[3] In contrast to van Duijkeren (2019) we apply strict, and not soft, constraints on the tracking error. This can be done and still ensure existence of solution from the design of the reference path. In particular, since $r(0) \in \mathcal{P}^0 \subset \mathbb{B}[p, \rho]$.

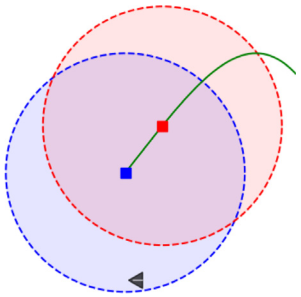


Fig. 5. The initial part of $\mathcal{P}$ (green line) is depicted alongside r^0 (blue square) and $r(\lambda\rho)$ (red square) with corresponding regions $\mathbb{B}[r^0, \rho]$ (blue) and $\mathbb{B}[r(\lambda\rho), \rho]$ (red). If the robot constraints prohibit movement to achieve $p(\Delta t) \in \mathbb{B}[r(\lambda\rho), \rho]$, no solution to (8) exists. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $x = [p^x, p^y, \psi]^T$ are the Cartesian position and orientation of the robot, and $u = [v\ \omega]^T$ are the linear and angular velocities. The controller

$$\kappa(x, r^0) = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -k_1\left(e^x \cos\psi + e^y \sin\psi\right) \\ k_2 e_\psi \end{bmatrix}, \tag{10}$$

with $e_\psi = (\text{atan2}(e^y, e^x) - \psi + \pi)$, $k_1 > 0$ and $k_2 > 0$ then satisfies the conditions for being an SBC given $\mathcal{U} = \mathbb{R}^2$. This can be derived using Barbalat's lemma with the positive semi-definite function $V = e^T e$ which has a negative semi-definite derivative, $\dot{V} = -k_1\left(e^x \cos\psi + e^y \sin\psi\right)^2$, under control law (10) (Siciliano, Sciavicco, Villani, & Oriolo, 2008). To handle saturated controllers as well, upper bounds for k_1 and k_2 can be found to ensure $\kappa(x, r^0) \in \mathcal{U}$ by utilizing the fact $r^0 \in \mathbb{B}[p, \rho] \subset \mathbb{B}[p, \bar{\rho}]$ and thus $\|e\|_2 \leq \bar{\rho}$. Consider $\mathcal{U} = [v^{\min}, v^{\max}] \times [-\omega^{\max}, \omega^{\max}]$ with $v^{\min} < 0 < v^{\max}$ and $\omega^{\max} > 0$. We have $|v| \leq k_1\left(|e^x| + |e^y|\right) \leq 2k_1\|e\|_2 \leq 2k_1\bar{\rho}$, and the upper bound $k_1 \leq \frac{\min(-v^{\min}, v^{\max})}{2\bar{\rho}}$ yields $v \in [v^{\min}, v^{\max}]$. Assuming e_ψ is evolving such that $e_\psi \in (-\pi, \pi]$, we have $|\omega| \leq k_2\pi$ and the upper bound $k_2 \leq \frac{\omega^{\max}}{\pi}$ yields $\omega \in [-\omega^{\max}, \omega^{\max}]$.

5.5. Switching control law

The control law μ is updated at a sampling interval Δt, such that it is constant over each period $t \in [t_k, t_{k+1})$ with $t_k = k\Delta t$, $k \in \mathbb{N}$. The control law switches between two modes depending on feasibility of (8) and if the RHRP has converged to the goal or not. The two modes are

$$\begin{aligned} &\textbf{MPC MODE} : r^0 \neq p^g \text{ and (8) is feasible} \\ &\textbf{SBC MODE} : \text{otherwise,} \end{aligned} \tag{11}$$

and the switching control law is given as

$$\mu(\cdot) = \begin{cases} \bar{u}_0^*, & \textbf{MPC MODE}, \\ \kappa(\cdot, r^0), & \textbf{SBC MODE}, \end{cases} \tag{12}$$

where $\bar{u}_0^*$ is derived from the optimal solution of (8). The control input is hence constant over a sampling interval when **MPC MODE** is active while the feedback controller κ is applied with r^0 as setpoint when **SBC MODE** is active. The update laws for the internal control variables r^+ and $\mathcal{O}^+$ are given as

$$r^+ = \begin{cases} r(w_0^*\Delta t), & \textbf{MPC MODE}, \\ r^0, & \textbf{SBC MODE}, \end{cases} \tag{13}$$

$$\mathcal{O}^+ = \begin{cases} \mathcal{O}^\star, & E^\star \text{ is a DSW} \\ \emptyset, & \text{otherwise}, \end{cases} \tag{14}$$

where w_0^* is derived from the optimal solution of (8). The initial reference candidate is specified along the RHRP, $\mathcal{P}$. When in **MPC MODE**, a solution to the MPC problem exists and r^+ is chosen as the 1-step predicted reference position of the MPC solution. This encourages forward shift of the RHRP at the next sampling instance, while ensuring r^+ to stay in a ρ-neighborhood of the robot due to (8h). When in **SBC MODE**, the control target is to realign the robot configuration to enable MPC feasibility at future sampling instances, and the next initial reference candidate is chosen as the current initial reference, suggesting no forward shift of the RHRP.

5.6. Motion control scheme

The complete control scheme for setpoint stabilization is outlined in Algorithm 3. Although no information about the environment is used in the MPC formulation nor for the SBC, collision avoidance is achieved as stated below by Theorem 7. This is obtained by ensuring a close tracking (with error less than ρ) of the path which is at least at a distance ρ from any obstacle and workspace boundary.

Algorithm 3 Setpoint control scheme

```
Parameters: $\bar{\rho} \in \mathbb{R}^+$, $\gamma \in (0, 1)$, $\Delta t \in \mathbb{R}^+$, $N \in \mathbb{N}^+$, $\lambda \in (0, 1)$, $c_w \in \mathbb{R}^+$, $c_e \in \mathbb{R}^+$, $\kappa \in \mathcal{K}$
Input: $E$, $p^g$, $x$
Output: $\mu(\cdot)$
Init: $r^+ \leftarrow p$, $\mathcal{O}^+ \leftarrow \emptyset$
Compute $E^\star$, $r^0$, $r^g$, $\rho$ using Algorithm 2
Compute $\mathcal{P}$ according to (6)-(7)
Run solver for (8)
Update $r^+$ and $\mathcal{O}^+$ according to (13)-(14)
Determine $\mu$ as in (12)
```

Theorem 7 (*Collision Avoidance*). *The trajectory for a robot with dynamics (2) and initial position $p(t_0) \in \mathcal{F}(t_0)$ following the motion control scheme in Algorithm 3 is collision-free, i.e. $p(t) \in \mathcal{F}(t), \forall t \geq t_0$, if Assumptions 2–4 hold.*

Proof. See Appendix B.

While convergence properties cannot be stated for generic scenarios with dynamic obstacles (consider the case with iteratively opening and closing of two separated gaps in a room), it can be stated under the following assumptions.

Assumption 8. There exists a time instance after which the environment is static, i.e. $\exists k^s \in \mathbb{N} \setminus \infty$ s.t. $E(t) = E_s = \{\mathcal{W}_s, \mathcal{O}_s\}, \forall t \geq t_{k^s}$.

Assumption 9. The workspace boundary and all obstacles after time t_{k^s} are at least at a distance $\bar{\rho}$ from the goal, i.e. $\text{dist}(p^g, \{\partial\mathcal{W}_s, \mathcal{O}_s\}) \geq \bar{\rho}$.

Without loss of generality, we will in the following assume $k^s = 0$. Note that Assumptions 2–4 and Assumption 8 trivially hold for a static scene and Assumption 9 can easily be obtained by adjustment of $\bar{\rho}$ if $p^g \in \mathcal{F}_s$. The proposed control scheme provides convergence from the set $\mathcal{C}_s^{\bar{\rho}} = \mathcal{C}^{\bar{\rho}}(t_0)$ given by (5) as stated by the following proposition.

Proposition 10 (*Convergence to Goal by Successful DSW Generation*). *The trajectory for a robot with dynamics (2) following the motion control scheme in Algorithm 3 converges to p^g from any position $p(t_0) \in \mathcal{C}_s^{\bar{\rho}}$ if $E^\star(t_0)$ is a DSW and Assumptions 2–4 and 8–9 hold.*

Proof. See Appendix C.

Table 1
Control parameters.

$\bar\rho$	γ	N	λ	c_w	c_e	R	R_Δ
0.2	0.9	6	0.5	1	1	$\begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0 \\ 0 & 0 \end{pmatrix}$

The convergence in Proposition 10 is dependent on a successful environment modification at time t_0. However, Theorem 6 can be used to declare an a priori sufficient condition for convergence as stated below.

Theorem 11 (*Convergence to Goal*). *The trajectory for a robot with dynamics* (2) *following the motion control scheme in Algorithm* 3 *converges to* p^g *from any position* $p(t_0) \in \mathcal{C}_s^{\bar\rho}$ *if* $E_s^{\bar\rho}$ *is DSW equivalent and Assumptions* 2–4 *and* 8–9 *hold.*

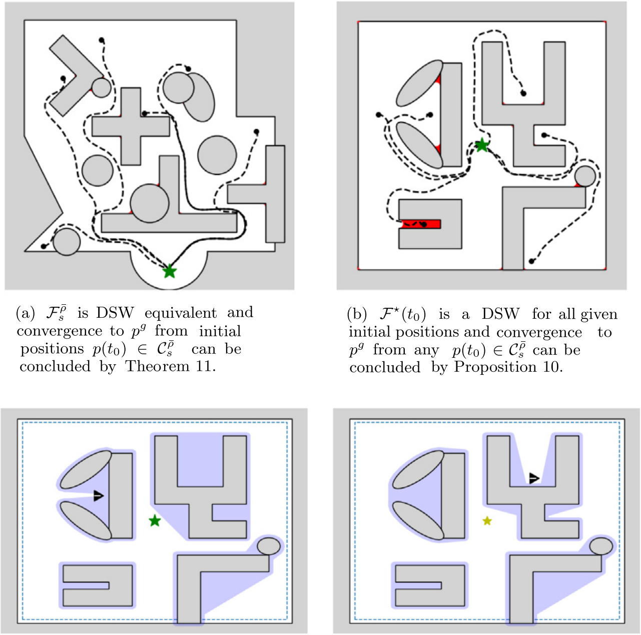
Proof. Since $p(t_0) \in \mathcal{C}^{\bar\rho}(t_0)$, Algorithm 2 yields $\rho = \bar\rho$. Then $E^\rho(t_0) = E_s^{\bar\rho}$ is DSW equivalent. By design we have $r^0(t_0) \in \mathcal{F}^\rho(t_0)$ and $r^g(t_0) \in \mathcal{F}^\rho(t_0)$. From Theorem 6 it can then be concluded that $E^\star(t_0)$ is a DSW and convergence to p^g follows from Proposition 10.

The positions from where convergence is not guaranteed, $\mathcal{F} \setminus \mathcal{C}_s^{\bar\rho}$, appear in the neighborhood of obstacle intersections and in the neighborhood of concave obstacle vertices as seen in Figs. 4 and 6.

6. Results

Three simulation scenarios are carried out to illustrate the performance of the proposed control scheme. A unicycle robot described by (9) and input constraints $\mathcal{U} = [-0.1, 1] \times [-1, 1]$ is considered in all cases. The Runge–Kutta method (RK4) is applied for integration of the system evolution which is updated at a frequency of 100 Hz. Function approximation of the RHRP using a sixth-degree polynomial and RHRP buffering are applied as described in Dahlin and Karayiannidis (2023b). The regularization term is defined to smoothen the trajectory as $J(\bar{u}) = \sum_{i=0}^{N-1} (\bar{u}_i - u^d)^T R(\bar{u}_i - u^d) + (\bar{u}_i - \bar{u}_{i-1})_i^T R_\Delta(\bar{u}_i - \bar{u}_{i-1})$, with $u^d = [w^{\max}\ 0]^T$ being the desired control input and $\bar{u}_{-1}$ being the previously applied control input. The control sampling period is $\Delta t = 0.2$ and the state integration in the MPC is performed using RK4. The SBC is defined as in (10) with $k_1 = 0.15, k_2 = 0.3$ ensuring $\kappa(x, r) \in \mathcal{U},\ \forall x \in \mathcal{X}, \forall r \in \mathbb{R}^2$. All numerical values for the control parameters are stated in Table 1.[4]

To illustrate the convergence properties derived in Section 5.6, two static scenes as shown in Fig. 6 are considered.[5] The robot is initialized at different positions $p(t_0) \in \mathcal{F}$ with horizontal orientation, $\psi(t_0) = 0$, for all cases. In Fig. 6(a), the environment form a DSW equivalent $\mathcal{F}_s^{\bar\rho}$ and convergence can be concluded a priori from any position $p(t_0) \in \mathcal{C}_s^{\bar\rho}$ by Theorem 11 which is confirmed by the simulation results. The environment in Fig. 6(b) is not DSW equivalent. However, $E^\star(t_0)$ is a DSW for all given initial positions and convergence to p^g from any $p(t_0) \in \mathcal{C}_s^{\bar\rho}$ follows from Proposition 10 which is also confirmed by the simulations. The robot is also initialized at one position in the set $\mathcal{F} \setminus \mathcal{C}_s^{\bar\rho}$ from where Proposition 10 provides no convergence guarantee (lower left). Nonetheless, the robot converges to p^g, indicating stronger convergence than is theoretically proved. Note that the shape of the obstacles $\mathcal{O}^\star$ depends on the robot position, and $E^\star$ is thus different for each case as illustrated in Fig. 6(c)–6(d).


(a) $\mathcal{F}_s^{\bar\rho}$ is DSW equivalent and convergence to p^g from initial positions $p(t_0) \in \mathcal{C}_s^{\bar\rho}$ can be concluded by Theorem 11.

(b) $\mathcal{F}^\star(t_0)$ is a DSW for all given initial positions and convergence to p^g from any $p(t_0) \in \mathcal{C}_s^{\bar\rho}$ can be concluded by Proposition 10.

(c)

(d)

Fig. 6. (a–b) A static set of obstacles $\mathcal{O}$ (gray), and the set $\mathcal{F} \setminus \mathcal{C}_s^{\bar\rho}$ (red) from where convergence cannot be stated by Proposition 10. Traveled path (dashed black lines) to a goal position p^g (green star) is shown from different initial positions $p(t_0)$ (black dots), all with horizontal initial orientation $\psi(t_0) = 0$. (c–d) The starshaped obstacles $\mathcal{O}^\star$ (blue) are formed to obtain a starshaped enclosing of the obstacles while remaining the robot position (black triangle) an exterior point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

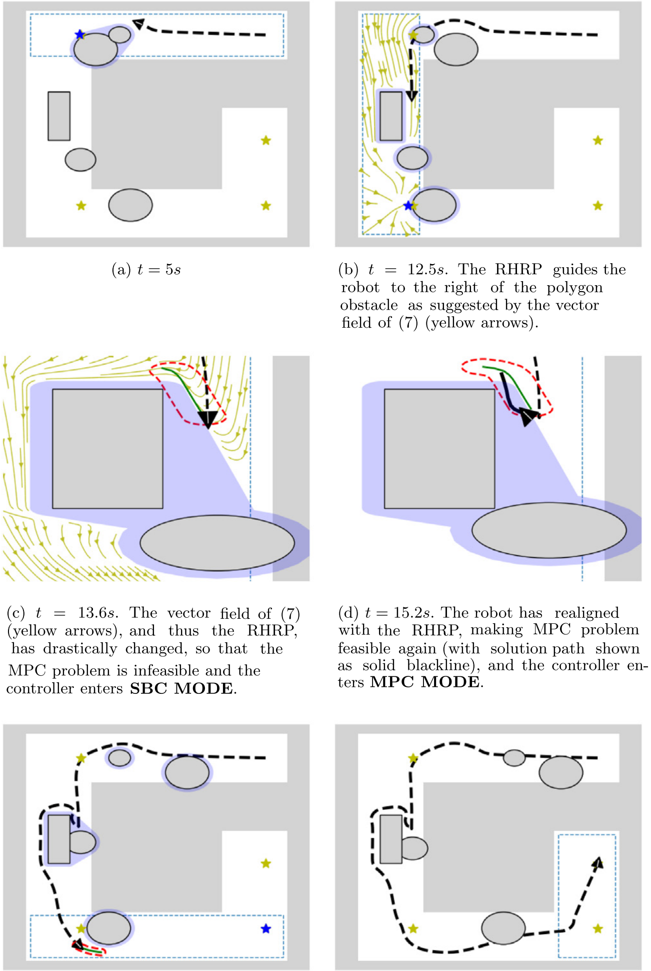
As suggested in Huber et al. (2022), also non-starshaped workspaces can be treated by dividing the workspace into several ordered subregions with corresponding local dynamics generated by a high-level planner. In Fig. 7, a concave workspace is divided into four intersecting rectangles, each assigned with a goal point that guides the robot towards the next subregion (final goal position for the last subregion). The subregions are activated as current workspace in a consecutive manner when the robot enters the region interior. The robot is here additionally simulated to be subject to additive input disturbance, randomly sampled from a normal distribution $\mathcal{N}(0, 0.1^2)$ in both dimensions. The original environment contains four moving circular obstacles and one static polygon. At time 12.5s, the robot enters on the right side of the polygon obstacle. As the closest circular obstacle moves towards the polygon, the gap between the polygon and the workspace boundary is reduced and finally closing at time 13.6s. At this point, the RHRP drastically changes to circumvent the polygon on the left side. Due to the limited rotational velocity and reverse speed of the robot, the enforced initial forward reference motion (8i) is conflicting with the tracking error constraint (8h) and the MPC problem is infeasible. During the time period $t \in [13.6, 15.2]$, the SBC is applied and the robot realigns with the RHRP, enabling feasibility of (8) at $t = 15.2$s and afterwards.

7. Conclusion

This article proposed a motion control scheme for robots operating in a workspace containing a collection of dynamic, possibly intersecting, obstacles. The method combines environment modification into a scene of disjoint obstacles with a dynamical systems formulation to generate a receding horizon path. A novel

[4] The code for the implementation can be found at https://github.com/albindgit/star_navigation. The code is aimed to showcase the applicability of the proposed method, without focusing on computational performance.

[5] Further examples can be found in Dahlin (2023), including also an extension of the method for path-following control.



(a) $t = 5s$

(b) $t = 12.5s$. The RHRP guides the robot to the right of the polygon obstacle as suggested by the vector field of (7) (yellow arrows).

(c) $t = 13.6s$. The vector field of (7) (yellow arrows), and thus the RHRP, has drastically changed, so that the MPC problem is infeasible and the controller enters **SBC MODE**.

(d) $t = 15.2s$. The robot has realigned with the RHRP, making MPC problem feasible again (with solution path shown as solid blackline), and the controller enters **MPC MODE**.

(e) $t = 27s$

(f) $t = 40s$

Fig. 7. The robot navigates in a non-starshaped workspace divided into several subregions with corresponding goal points (yellow stars) to the final goal. The active clearance workspace $\mathcal{W}^{\rho}(t)$ (blue dashed line), starshaped obstacles $\mathcal{O}^{\star}(t)$ (blue), and active goal point (blue star) depends on the robot position. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

MPC formulation was proposed to enforce forward motion along the path within an obstacle-clearance zone, which in combination with a stabilizing backup controller allowed for formal derivations of collision avoidance and convergence properties.

The proposed method relies on momentary observations of the environment. In scenarios where predictions of obstacle movements are available, it would be valuable to adapt the predictive controller to leverage these. In case of uncertainties, the theoretical guarantees provided by the theorems do not hold even though preliminary simulation studies have demonstrated a certain degree of robustness. One development direction involves integrating uncertainties and disturbances into the robot model to enable a robust controller. This can be accomplished using methods such as tube-based MPC (Mayne, Kerrigan, van Wyk, & Falugi, 2011), corridor MPC (Roque, Cortez, Lindemann, & Dimarogonas, 2022), or by utilizing the FaSTrack framework (Chen et al., 2021).

Appendix A. Proof of Theorem 6

Given the cluster set, Cl, at the start of an iteration of Algorithm 2 of Dahlin and Karayiannidis (2023a), it follows from Property 4b and 4d of Dahlin and Karayiannidis (2023a) that for the generated starshaped obstacles set at that iteration, $\mathcal{O}^{\star}$, it holds for any $i \in [1, \dots, |Cl|]$ that $K^i \subset \ker_{\cap}(cl^i) \Rightarrow \mathcal{O}^{\star,i} = SH_{\ker K^i}(cl^i) = cl^i_{\cup}$. The kernel selection Algorithm 1 computes $K^i \subset S^i$ with $S^i = \ker_{\cap}(cl^i)$ if $cl_{\cup} \subset \mathcal{W}$ and $S^i = \ker_{\cap}(cl^i) \setminus \mathcal{W}$ if $cl_{\cup} \not\subset \mathcal{W}$, given that S^i is nonempty for these selections. At first iteration in Algorithm 2 of Dahlin and Karayiannidis (2023a) we have $Cl = \mathcal{O}$ and $\ker_{\cap}(cl^i) = \ker(\mathcal{O}^i)$. Since the environment is DSW equivalent, the set S^i specified above is nonempty. Specifically, $\ker(\mathcal{O}^i)$ is nonempty since $\mathcal{O}^i$ is starshaped and $\ker(\mathcal{O}^i) \setminus \mathcal{W}$ is nonempty for any obstacle $\mathcal{O}^i \not\subset \mathcal{W}$ due to condition (4). Thus, $K^i \subset \ker_{\cap}(cl^i)$ and $\mathcal{O}^{\star,i} = \mathcal{O}^i,\ i \in [1, \dots, |Cl|]$. As a consequence, $Cl = Cl^{\star}$ after the assignment in line 13. By construction, $Cl^{\star}$ consists of mutually disjoint connected subsets of $\mathcal{O}^{\star}$ and Cl thus consists of mutually disjoint connected subsets of $\mathcal{O}$. If the environment is a DSW, it follows that $Cl = \mathcal{O}$ and the algorithm returns the DSW. Otherwise, Cl satisfies (3) and (4) since the environment is DSW equivalent and the division of a set into mutually disjoint connected subsets is unique. Hence, S^i specified above is nonempty. Thus, $K^i \subset \ker_{\cap}(cl^i)$ and $\mathcal{O}^{\star,i} = cl^i_{\cup},\ i \in [1, \dots, |Cl|]$. Hence, $\mathcal{O}^{\star}$ is a mutually disjoint subset of $\mathcal{O}$ with $\mathcal{O}^{\star}_{\cup} = \mathcal{O}_{\cup}$, and $|\mathcal{O}^{\star}| = |Cl|$ such that the algorithm terminates. Since all regions in $\mathcal{O}^{\star}$ are starshaped by construction, and where any clustered obstacle satisfies (4) the environment $\{\mathcal{W}, \mathcal{O}^{\star}\}$ is a DSW.

Appendix B. Proof of Theorem 7

From Assumptions 2–4 it follows that $p(t) \in \hat{\mathcal{F}}(t_k),\ \forall t \in (t_k, t_{k+1}] \Rightarrow p(t) \in \mathcal{F}(t),\ \forall t \in (t_k, t_{k+1}]$, where $\hat{\mathcal{F}} = \mathcal{W} \setminus \hat{\mathcal{O}}_{\cup}$. Since $p(t_0) \in \mathcal{F}(t_0)$, it suffices to show that $p(t) \in \hat{\mathcal{F}}(t_k),\ \forall t \in (t_k, t_{k+1}]$ given $p(t_k) \in \hat{\mathcal{F}}(t_k)$ at any sampling instance, t_k. Consider first the case where the controller is in **MPC MODE** at time t_k and thus $u(t) = \bar{u}_0^*(t_k),\ \forall t \in [t_k, t_{k+1})$. Then $\|p(t+\tau) - r_k(w^*_{k,0}\tau)\|_2 \leq \rho(t_k), \forall \tau \in [0, \Delta t]$ from (2) and (8c)–(8h). Here $r_k(\cdot) \in \mathcal{P}(t_k)$ is the RHRP-mapping at time instance t_k and $w^*_{k,0}$ is the initial path speed of the optimal solution to (8). Thus, $p(t) \in \mathcal{P}^{\rho}(t_k) \subset \hat{\mathcal{F}}(t_k),\ \forall t \in (t_k, t_{k+1}]$. If the controller instead is in **SBC MODE**, the SBC is applied, $u(t) = \kappa(x(t), r^0(t_k)),\ \forall t \in [t_k, t_{k+1})$. Since $\|r^0(t_k) - p(t_k)\|_2 \leq \rho(t_k)$ by definition of r^0, it follows from the definition of SBC that $\|r^0(t_k) - p(t)\| \leq \|r^0(t_k) - p(t_k)\|_2 \leq \rho(t_k), \forall t \in (t_k, t_{k+1}]$. That is $p(t) \in \mathbb{B}[r^0(t_k), \rho(t_k)] \subset \mathcal{P}^{\rho}(t_k) \subset \hat{\mathcal{F}}(t_k),\ \forall t \in (t_k, t_{k+1}]$.

Appendix C. Proof of Proposition 10

For ease of notation, let $\rho_k = \rho(t_k)$. The proof is given in five steps. In the first two steps, we show that the environment modification is static after $k = 0$ in the sense that $\rho_k = \bar{\rho}, \forall k \geq 0$ and $\mathcal{F}^{\star}_k = \mathcal{F}^{\star}_0, \forall k \geq 0$. In step 3 we show that the initial reference point r^0_k is following the parameterized regular curve $\hat{\Gamma} = \{\hat{\gamma} \in \mathbb{R}^2 : \hat{\theta} \in [0, \infty) \to \hat{\gamma}(\hat{\theta})\}$ given by

$$\frac{d\hat{\gamma}(\hat{\theta})}{d\hat{\theta}} = \bar{v}(\hat{\gamma}(\hat{\theta}), p^g, \mathcal{O}^{\star}_0), \quad \hat{\gamma}(0) = r^+_0, \tag{C.1}$$

Specifically, we show $r^0_k = \hat{\gamma}(\hat{\theta}_k), \forall k \geq 0$ given the virtual path coordinate $\hat{\theta}$ with $\hat{\theta}_0 = 0$ and dynamics

$$\hat{\theta}_{k+1} = \begin{cases} \hat{\theta}_k + w^*_{k,0}\Delta t, & \textbf{MPC MODE} \\ \hat{\theta}_k, & \textbf{SBC MODE}. \end{cases} \tag{C.2}$$

In step 4 we show that r^0_k converges to p^g in finite time, and in step 5 it is shown that this implies convergence of p to p^g.

Step 1 ($\rho_k = \bar{\rho},\ \forall k \geq 0$): Assume $\rho_k = \bar{\rho}$. From the proof of Theorem 7 we have $p_{k+1} \in \mathcal{P}^{\rho}_k \subset \mathcal{F}^{\bar{\rho}}_k \oplus \mathbb{B}[0, \bar{\rho}]$. Since $\mathcal{O}_{k+1} = \mathcal{O}_k$, it follows that $p_{k+1} \in \mathcal{F}^{\bar{\rho}}_{k+1} \oplus \mathbb{B}[0, \bar{\rho}] = \mathcal{C}^{\bar{\rho}}_{k+1}$. Hence, Algorithm 2 yields $\rho_{k+1} = \bar{\rho}$. Since $p_0 \in \mathcal{C}^{\bar{\rho}}_0$ and therefore $\rho_0 = \bar{\rho}$, we can conclude that $\rho_k = \bar{\rho},\ \forall k \geq 0$.

Step 2 ($\mathcal{F}_k^\star = \mathcal{F}_0^\star,\ \forall k \geq 0$)*:* Since $\rho_k = \bar{\rho},\ \forall k \geq 0$ it follows that $\mathcal{F}_k^\star \subset \mathcal{F}_{k+1}^{\rho},\ \forall k \geq 0$. Given $\mathcal{F}_k^\star$ is a DSW, it follows from (14) and Algorithm 2 that $\mathcal{F}_{k+1}^\star = \mathcal{F}_k^\star$ if $r_{k+1}^0 \in \mathcal{F}_k^\star$ and $r_{k+1}^g \in \mathcal{F}_k^\star$. From the proof of Theorem 7 and from (13) we have $p_{k+1} \in \mathbb{B}[r_k(w_{k,0}^*\Delta t), \bar{\rho}] = \mathbb{B}[r_k^+, \bar{\rho}]$ when in **MPC MODE** and $p_{k+1} \in \mathbb{B}[r_k^0, \bar{\rho}] = \mathbb{B}[r_k^+, \bar{\rho}]$ when in **SBC MODE**. Since $r_k^+ \in \mathcal{P}_k \subset \mathcal{F}_{k+1}^{\rho}$ by definition, it follows that $r_k^+ \in \mathcal{P}_{k+1}^0$ and hence $r_{k+1}^0 = r_k^+$. Then, $r_{k+1}^0 \in \mathcal{P}_k \subset \mathcal{F}_k^\star$. The reference goal is given by $r_{k+1}^g = \arg\min_{r^g \in \mathcal{F}_{k+1}^{\rho}} \|r^g - p^g\|_2 = \arg\min_{r^g \in \mathcal{F}_k^{\rho}} \|r^g - p^g\|_2 = r_k^g \in \mathcal{F}_k^\star$. Thus, $\mathcal{F}_k^\star$ being a DSW implies $\mathcal{F}_{k+1}^\star = \mathcal{F}_k^\star,\ \forall k \geq 0$. Since $\mathcal{F}_0^\star$ is a DSW, it follows that $\mathcal{F}_k^\star = \mathcal{F}_0^\star,\ \forall k \geq 0$.

Step 3 $\left(r_k^0 = \hat{\gamma}\left(\hat{\theta}_k\right), \forall k \geq 0\right)$*:* Assume $r_k^0 = \hat{\gamma}(\hat{\theta}_k)$ and hence $r_k(s) = \hat{\gamma}(\hat{\theta}_k + s)$. From Step 2 we have $r_{k+1}^0 = r_k^+$. When in **SBC MODE**, $r_{k+1}^0 = r_k^0 = \hat{\gamma}(\hat{\theta}_k) = \hat{\gamma}(\hat{\theta}_{k+1})$. When in **MPC MODE**, $r_{k+1}^0 = r_k(w_{k,0}^*\Delta t) = \hat{\gamma}(\hat{\theta}_k + w_{k,0}^*\Delta t) = \hat{\gamma}(\hat{\theta}_{k+1})$. Thus, $r_k^0 = \hat{\gamma}(\hat{\theta}_k) \Rightarrow r_{k+1}^0 = \hat{\gamma}(\hat{\theta}_{k+1})$. Now, $r_0^0 = r_0^+ = \hat{\gamma}(0) = \hat{\gamma}(\hat{\theta}_0)$ and we can conclude $r_k^0 = \hat{\gamma}(\hat{\theta}_k), \forall k \geq 0$.

Step 4 ($\exists j < \infty\ s.t.\ r_k^0 = p^g, \forall k \geq j$)*:* Let $\hat{\theta}^g$ be the arc length of $\hat{\Gamma}$ such that $\hat{\gamma}(\hat{\theta}) = p^g\ \forall \hat{\theta} \geq \hat{\theta}^g$. Such a $\hat{\theta}^g$ exists due to the converging properties of (7) in a DSW. Let $K = \left\lceil \frac{\hat{\theta}^g}{\lambda\bar{\rho}} \right\rceil$. If at time step k the controller has been in **MPC MODE** at K previous time iterations, it follows from (8i) and (C.2) that $\hat{\theta}_k \geq K\lambda\bar{\rho} \geq \hat{\theta}^g$. Now assume $\hat{\theta}_k < \hat{\theta}^g\ \forall k$. A solution to (8) can then be found at most $K-1$ times, i.e. there exists a $k' < \infty$ where (8) is infeasible for any $k \geq k'$. Then $r_k^0 = r_{k'}^0, \mathcal{P}_k = \mathcal{P}_{k'}, \forall k \geq k'$ and the SBC is applied from this time instance, $u(t) = \kappa(x(t), r_{k'}^0), \forall t \geq t_{k'}$. Define $\alpha = \max_{s\in[0,\lambda\bar{\rho}]} \|r_{k'}^0 - r_{k'}(s)\|_2$. Due to the normalized dynamics (6) we have $\alpha \leq \lambda\bar{\rho} < \bar{\rho}$. Since $\lim_{t\to\infty} p(t) = r_{k'}^0$ according to the definition of the SBC, there exists a finite $k'' < \infty$, s.t. $p_{k''} \in \mathbb{B}[r_{k'}^0, \bar{\rho} - \alpha]$ due to the continuity of the solution. Now consider the solution for (8) $w_0 = \frac{\lambda\bar{\rho}}{\Delta t}$, $w_i = 0, \forall i > 0$ and $\bar{u}_i = u', \forall i$ with $f(x, u') = 0,\ \forall x$. This is a feasible solution at time instance k'' since $\varepsilon(\tau) \leq \max_{s\in[0,\lambda\bar{\rho}]} \|p_{k''} - r_{k'}(s)\|_2 \leq \bar{\rho},\ \forall \tau \in [0, T]$. This is a contradiction to the conclusion that (8) is infeasible for $k \geq k'$ and no such k' exists. Thus, $\exists k < \infty$ s.t. $\hat{\theta}_k \geq \hat{\theta}^g$. Since $w_{k,0}^* > 0$, we have $\hat{\theta}_{k+1} \geq \hat{\theta}_k, \forall k$ and thus $\exists j < \infty$ s.t. $\hat{\theta}_k \geq \hat{\theta}^g, \forall k \geq j$. Then, from Step 3 and the definition of $\hat{\theta}_g$, it follows that $\exists j < \infty$ s.t. $r_k^0 = p^g, \forall k \geq j$.

Step 5 ($\lim_{t\to\infty} p(t) = p^g$)*:* From step 4 we have that there exists some time instance j where $r_k^0 = p^g,\ \forall k \geq j$. At this point, the controller enters **SBC MODE**. Since the SBC renders asymptotically stable closed-loop error dynamics, it can be concluded that $\lim_{t\to\infty} p(t) = p^g$.

References

Chen, M., Herbert, S., Hu, H., Pu, Y., Fisac, J., Bansal, S., et al. (2021). FaSTrack: A modular framework for real-time motion planning and guaranteed safe tracking. *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, *66*(12), 5861–5876.

Connolly, C., Burns, J., & Weiss, R. (1990). Path planning using Laplace's equation. In *Proc., IEEE int. conf. on robotics and automation: vol.3*, (pp. 2102–2106).

Dahlin, A. (2023). *Reactive motion planning and control under constraints* (Ph.D. thesis), Chalmers University of Technology, [Online] Available: https://research.chalmers.se/publication/538686.

Dahlin, A., & Karayiannidis, Y. (2023a). Creating star worlds: Reshaping the robot workspace for online motion planning. *IEEE Transactions on Robotics*, 1–16.

Dahlin, A., & Karayiannidis, Y. (2023b). Obstacle avoidance in dynamic environments via tunnel-following MPC with adaptive guiding vector fields. In *2023 62nd IEEE conference on decision and control* (pp. 5784–5789).

Daily, R., & Bevly, D. M. (2008). Harmonic potential field path planning for high speed vehicles. In *Proc. American control conference* (pp. 4609–4614).

Faulwasser, T., & Findeisen, R. (2016). Nonlinear model predictive control for constrained output path following. *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, *61*(4), 1026–1039.

Feder, H., & Slotine, J.-J. (1997). Real-time path planning using harmonic potentials in dynamic environments. In *Proc. of int. conf. on robotics and automation: vol. 1*, (pp. 874–881).

Ginesi, M., Meli, D., Calanca, A., Dall'Alba, D., Sansonetto, N., & Fiorini, P. (2019). Dynamic movement primitives: Volumetric obstacle avoidance. In *Int. conf. on advanced robotics* (pp. 234–239).

Hansen, G., Herburt, I., Martini, H., & Moszyńska, M. (2020). Starshaped sets. *Aequationes Mathematicae*, *94*.

Huber, L., Billard, A., & Slotine, J.-J. (2019). Avoidance of convex and concave obstacles with convergence ensured through contraction. *IEEE Robotics and Automation Letters*, *4*(2), 1462–1469.

Huber, L., Slotine, J.-J., & Billard, A. (2022). Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds. *IEEE Transactions on Robotics and Automation*, *38*(5), 3113–3132.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proc. IEEE int. conf. on robotics and automation* (pp. 500–505).

Kumar, H., Paternain, S., & Ribeiro, A. (2022). Navigation of a quadratic potential with ellipsoidal obstacles. *Automatica*, *146*.

Loizou, S. G. (2011). Closed form navigation functions based on harmonic potentials. In *IEEE conf. on decision and control and European control conf.* (pp. 6361–6366).

Mayne, D. Q., Kerrigan, E. C., van Wyk, E. J., & Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, *21*(11), 1341–1353.

Rimon, E., & Koditschek, D. (1992). Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, *8*(5), 501–518.

Roque, P., Cortez, W. S., Lindemann, L., & Dimarogonas, D. V. (2022). Corridor MPC: Towards optimal and safe trajectory tracking. In *2022 American control conference* (pp. 2025–2032).

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2008). Robotics: Modelling, planning and control. In *Advanced textbooks in control and signal processing*, London: Springer.

Stavridis, S., Papageorgiou, D., & Doulgeri, Z. (2017). Dynamical system based robotic motion generation with obstacle avoidance. *IEEE Robotics and Automation Letters*, *2*(2), 712–718.

van Duijkeren, N. (2019). *Online motion control in virtual corridors: for fast robotic systems* (Ph.D. Thesis), KU Leuven, [Online] Available: https://lirias.kuleuven.be/retrieve/527169.

Albin Dahlin received a Diploma in control and mechatronics Eng. (2018) and a Ph.D. degree in Electrical Eng. (2024) from Chalmers University of Technology, Sweden.

He is currently working as a software engineer at Cognibotics AB, Sweden, with focus on robot motion and precision.

His research interests include robot control, calibration and navigation in human-centered environments, but also adaptive and nonlinear control systems.

Yiannis Karayiannidis received a Diploma in Electrical and Computer Eng. (2004) and a Ph.D. degree in Electrical Eng. from Aristotle University of Thessaloniki, Greece (2009).

He is currently an Associate Professor with the Dept. of Automatic Control, Faculty of Engineering at Lund University, Sweden. He is ELLIIT-recruited and WASP-affiliated faculty.

He was previously affiliated with KTH, Royal Institute of Technology, Stockholm, Sweden (2011–2020) and Chalmers University of Technology, Gothenburg, Sweden (2015–2022). His research interests include robot control, manipulation and navigation in human-centered environments, dual-arm manipulation, mobile manipulation, force control, haptic perception, robotic assembly, cooperative multi-agent robotic systems, physical human–robot interaction but also adaptive and nonlinear control systems.