

# The VFO-Driven Motion Planning and Feedback Control in Polygonal Worlds for a Unicycle with Bounded Curvature of Motion

Tomasz Gawron  · Maciej Marcin Michałek

Received: 24 June 2016 / Accepted: 5 April 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** Integrated motion planning and control for the purposes of maneuvering mobile robots under state- and input constraints is a problem of vital practical importance in applications of mobile robots such as autonomous transportation. Those constraints arise naturally in practice due to specifics of robot mechanical construction and the presence of obstacles in motion environment. In contrast to approaches focusing on feedback control design under the assumption of given reference motion or motion planning with neglectation of subsequent feedback motion execution, we adopt a *controller-driven* motion planning paradigm, which has recently gained attention of many researchers. It postulates design of motion planning algorithms dedicated to specific feedback control policies, which compute a sequence of feedback control subtasks instead of classically planned open-loop controls or parametric paths. In this spirit, we propose

a motion planning algorithm driven by the VFO (Vector Field Orientation) control law for the waypoint-following task. Presented analysis of the VFO control law reveals its beneficial properties, which are subsequently utilized to solve a generally nonlinear and non-convex optimal motion planning problem by formulating it as a mixed-integer linear program (MILP). The solution proposed in this paper yields a waypoint sequence, which is designed for execution by application of the VFO control law to drive a robot to a prescribed final configuration under an input constraint imposed by bounded curvature of robot motion and state constraints resulting from a convex decomposition of task space. Satisfaction of these constraints is guaranteed analytically and exactly, i.e., without utilization of numerical approximations. Moreover, for a given discrete set of possible waypoint orientations, the proposed algorithm computes plans optimal w.r.t. given cost functional, which can be any convex linear combination of quantities such as robot path length, curvature of robot motion, distance to imposed state constraints, etc. Furthermore, the planning algorithm exploits the possibility of both forward or backward movement of the robot to allow maneuvering in demanding environments. Generated waypoint sequences are a compact representation of a motion plan, which can be immediately executed with the VFO controller without any additional post-processing. Validity of the proposed approach has

---

This work was financially supported by the National Science Centre (NCN), Poland, as the research grant No. 2016/21/B/ST7/02259.

---

T. Gawron (✉) · M. M. Michałek  
Chair of Control and Systems Engineering, Poznan  
University of Technology, Piotrowo 3A, 60-065,  
Poznań, Poland  
e-mail: tomasz.gawron@doctorate.put.poznan.pl

M. M. Michałek  
e-mail: maciej.michalek@put.poznan.pl

been confirmed by simulation studies and experimental motion execution with a laboratory-scale mobile robot.

**Keywords** Controller-driven motion planning · Waypoint-following · State and curvature constraints · Mobile robots · Unicycle · Feedback control

## 1 Introduction

The ability to perform autonomous maneuvers in cluttered environments is desired in many practical applications of wheeled mobile robots such as search-and-rescue missions, warehouse management automation, service robotics, parking assistance systems, and lately autonomous road vehicles. In this work we propose a *controller-driven* motion planning algorithm and feedback control strategy for a generic robot-body model in the form of a kinematic unicycle. It is simple and captures nonholonomic constraints present in various autonomous vehicles of interest, such as N-trailer trucks, cars and differentially-driven robots. Thus, it can serve as a good starting point for extensions of our method to more complex kinematics. Mechanical construction of vehicles often imposes additional limitations on robot motion resulting from, e.g., bounded steering angle in case of car-like vehicles and bounded articulation angles in case of N-trailer vehicles. Those limitations are reflected in the kinematic robot model by an input constraint imposing bounded curvature of motion. Considered robot model is complemented by state constraints, which can directly represent a model of the cluttered environment or designated safe motion areas.

Traditionally, the motion planning problem is solved by implementing a multi-stage sequential planning procedure as mentioned, e.g. in [29] and implemented, e.g., in [27]. Such planners often generate a purely geometric, and in some cases initially infeasible, collision-free path in configuration space, which is subsequently parametrized with time (see [50] for an energy-optimal approach) and optionally smoothed (as demonstrated in [9]) to form a trajectory. During the motion execution stage, the computed path or trajectory can be tracked using one of the available feedback controllers. Although such an approach yields acceptable results, there are various limitations

resulting from the fact that behavior of the closed-loop motion control system is not explicitly considered during motion planning stage. The practically important effects of non-nominal motion conditions (e.g., initial robot configuration placed away from the planned one), exogenous disturbances, and uncertain measurements may render a feasible motion plan unfit for safe execution with a given feedback control law in practical conditions.

We adopt a *controller-driven motion planning* design paradigm envisioned in [29] under the name of *algorithmic control theory* and applied in [5, 44] as *policy based motion planning*. The controller-driven methodology postulates design of specialized motion planners dedicated to specific feedback motion controllers. Such a specialization makes it possible to generate a motion plan tailored specifically to the needs of the assumed feedback control law. At the same time, the availability of information about closed-loop system behavior enables the planner to exploit specific properties of the assumed controller leading to a potentially simpler and more efficient planning process. The same information can be also utilized to avoid motion conditions, which are adverse for the utilized feedback control law by appropriate motion planning.

In this paper, we extend the results presented in [12] and propose a controller-driven motion planner for a kinematic unicycle with bounded curvature of motion and state constraints. Constraints imposed on the robot state are described by a sequence of traversable convex polygons (convex decomposition), which is a convenient representation for cluttered motion environments. Presented motion planning algorithm is dedicated to the VFO (Vector Field Orientation) controller in the version for the waypoint-following task proposed in [34] as an extension of the methodology presented in [8]. The design of our planner relies upon specific beneficial properties of the VFO control law, some of which were found recently and were presented in the preliminary work [12]. Thanks to those properties the generally nonlinear and non-convex planning problem was casted to a mixed-integer linear program (MILP) in our specific case. As argued later in Section 5, the complexity of formulated problem depends on topology of the environment as opposed to the methods employing direct time-discretization of robot control signals. What is more, satisfaction of state and curvature of motion constraints is guaranteed

analytically, while planning of waypoint positions is conducted in a continuous domain. Another closely related benefit of the controller-driven design methodology is the fact that our algorithm generates motion plans represented compactly as short sequences of waypoint configurations, which are executed directly (i.e. without any postprocessing) by the VFO control law. Although motion plans contain only a small number of waypoints, robot path during plan execution is perceived as natural thanks to application of the VFO controller.

In the sequel, we systematically describe the design and verification of the proposed motion planning algorithm. In Section 2 we shortly review motion planning and control algorithms for kinematic models similar to the assumed constrained unicycle model. In Section 3 we formally state the problem, introduce notational conventions and sketch a general strategy of our solution. Section 4 contains introductory information regarding the VFO control law necessary for further considerations and the description of its specific properties useful in motion planning. Section 5 provides a more detailed description of the motion planning algorithm. For simplicity of exposition the method is presented first without consideration of the state constraints, and later extended to the convex polygon sequences. In Section 6 we touch upon the issues of practical motion execution. Section 7 contains results of simulations and experimental verification of the proposed solution for selected scenarios. Results of motion execution in non-nominal motion conditions arising from practical issues demonstrate viability of our approach. The work is summarized with final remarks gathered in Section 8.

## 2 Related Work

Let us briefly review selected motion planning algorithms for mobile robots with bounded curvature of motion in the presence of state constraints representing cluttered environment. Emphasis is put on controller-driven methods. A wider review of motion planning methods for mobile robots can be found in [28] and [26].

In [32] a particular polynomial spline analyzed in [41] was utilized in conjunction with a multicriterial optimization algorithm to plan autonomous parking maneuvers. This solution, although attractive due to

its ability to generate  $\mathbb{G}^3$ -continuous paths, assumes a fixed number of splines and predefined motion sense (forward or backward robot motion) dependent on specific maneuver type and motion environment topology. According to the authors of [32], those assumptions can be lifted at the expense of utilizing stochastic global optimization algorithms, which are characterized by high computational cost and potential convergence problems. Even under mentioned assumptions, the authors of [32] design a nonlinear optimization problem, for which in general only locally optimal paths are attained. Moreover, to satisfy state and curvature constraints, discretization in the domains of robot configuration and path parameters must be performed. Those issues remain relevant for various other methods based on nonlinear optimization or direct nonlinear optimal control such as [16]. Nevertheless, a promising local planning and control system utilizing a similar methods was proposed in [51].

One can also find several approaches relying on formulation of a MILP (mixed-integer linear programming) problem for model-predictive control or planning, utilizing discrete time problem formulations with explicit discretization of controls such as [3, 19, 46, 53]. Another interesting approach shown in [49] depends upon consideration of obstacle avoidance problem using optimal control framework.

Another approach shown in, e.g., [31, 37, 42, 48, 55] relies on utilization of graph search algorithms to find a path in a lazily evaluated graph representing discrete points in state space connected by reference motions taken from a finite set of precomputed motion primitives. In the same spirit, an interesting combination of purely geometric space exploration followed by motion primitive based planning similar to [45] was proposed in [4]. Such an approach is very flexible, since it utilizes a well known versatile grid-based environment representation and makes it conceptually easy to consider nearly arbitrary nonlinear system dynamics. On the other hand, such lattice based planners may have high memory requirements and can be applied only for discretized dynamical systems.

Moreover, care must be taken when designing a finite set of motion primitives utilized in various sampling-based or lattice-based planners. Many motion primitive sets constructed by sampling of various control signals may produce path segments hard to describe or approximate analytically, posing issues for feedback control during motion execution. This

problem can be alleviated by sampling in the space induced by a family of control policies. Such a concept is in fact very similar to the work presented in [5]. Worth noting that the assumption of a discrete set of motion primitives excludes asymptotic convergence of a computed motion plan to a solution, which is globally optimal w.r.t. assumed cost functional as shown, e.g. in [38].

Recently, we observe fast development of sampling-based motion planners building upon the concept of RRT (Rapidly Random exploring Trees) [11, 15, 17, 20, 23, 30] and other closely related probabilistic approaches such as [13]. Their controller-driven variants are usually obtained by integration of a specific extend procedure into the planner. Utilized extend procedures range from general optimal-control approaches such as [40], through spline path segments (e.g., [18, 54]), to more controller-driven approaches such as simulation of mobile robot with closed-loop control system proposed in [38] and control-Lyapunov function approach from [39]. Sampling-based algorithms are attractive due to their generality and ability to cope with virtually any robot and motion environment model. On the other hand, in general, they guarantee only probabilistic completeness as opposed to resolution completeness and can suffer from slow convergence to optimal plans for nonholonomic systems. What is more, in some cases their computational efficiency can deteriorate in environments with very narrow passages due to their otherwise beneficial Voronoi bias property. However, the intensity of this effect seems to be rarely high enough to severely impact planning on modern hardware. It is worth noting, that their efficient implementation can be a complex task due to auxiliary algorithmic problems such as, e.g., nearest-neighbor search.

Reviewed planning algorithms are often complemented by methods used for postprocessing sub-optimal plans using local optimization algorithms such as [24, 56]. In the same manner, one can utilize path generation methods [1, 10] providing continuous-curvature approximations of Dubins [6] and Reeds/Shepp [43] paths, which are time-optimal and highly applicable to cluttered environments (see [2]).

The presence of various issues in the above reviewed motion planning algorithms justifies further investigation of the problem and development of

alternative solutions. One such approach is the method based on global optimization, which is proposed in this work. Namely, in contrast to the method proposed in [32], we formulate a MILP optimization problem to find a sequence of waypoints, and, as a consequence, ensure convergence to global optimum w.r.t. assumed criteria and constraints for a given waypoint orientation discretization resolution. The number of motion segments, which is related to the number of splines in [32], does not need to be fixed but it can be fixed if the motion task at hand specifically requires so. Note that the proposed method considers optimization in the whole motion environment at once as opposed to [49]. While such a global approach can potentially lead to better solutions, it is also characterized by larger computational cost. The local method from [49] avoids the high computational cost and potential infeasibility thanks to consideration of one obstacle at a time. This allows for fast online replanning, which is usually not attained by the global methods. It is also worth noting that our approach requires only discretization in the space of waypoint orientations. Waypoint positions are planned in a continuous domain, while state constraints and curvature constraints are checked exactly, i.e., without any approximations and using analytically derived conditions. This differentiates the proposed method from the ones available in the literature and is congruent with the ongoing trend of seeking more efficient constraint checking procedures (visible in works such as [22]), and with efforts to guarantee constraint satisfaction in a continuous domain, see e.g., [25]. The benefits and necessity of seeking feedback control policies as opposed to open-loop controls or parametric paths has been also argued extensively in [5, 29, 38, 39, 44].

In our approach we utilize MILP optimization, which is also used in other MILP-based methods such as [3, 19, 46, 53]. However, there are significant differences between those methods and our approach. Namely, the method proposed in [53] is applicable only to linear dynamical systems (specifically, authors assumed point-mass dynamics), while in the present paper we consider unicycle kinematics with bounded curvature of motion, which is a strongly nonlinear system with a nonholonomic constraint and input constraints imposed. Since other MILP-based methods generate optimization constraints directly based on system dynamics (as opposed to our method leveraging the VFO control law), they are inherently

limited to only linear systems, since other differential constraints cannot be expressed as linear constraints in a MILP. Thus, other MILP methods cannot be directly applied to a kinematic unicycle. Note that due to the presence of a nonholonomic constraint, unicycle kinematics cannot be linearized without serious consequences. Namely, if one applies a local Taylor linearization at a constant operating point (such as a waypoint configuration) to the unicycle kinematics, the resultant linear system is not fully controllable as shown in [36]. The linearization of unicycle kinematics with a time-varying operating point is also not possible in the case of our problem, since the location of this point is not known at planning stage. On the other hand, application of the input-output feedback linearization loop to unicycle kinematics results in a loss of direct control over a robot orientation and ability to only control a position of an auxiliary point displaced w.r.t. the guidance point of a robot platform, which is also not acceptable for our problem. Even if one neglects the above arguments, makes an assumption of a linear point-mass dynamics model from [53], and attempts to incorporate the curvature constraint by constraining curvature of a path drawn by a position output of dynamics given in [53], one arrives at a nonlinear constraint resulting from curvature constraint involving velocity and acceleration components, which cannot be represented as a linear function of the variables introduced in [53]. Similar arguments can be formulated for other MILP-based methods.

Moreover, MILP-based methods considered in the literature guarantee constraint satisfaction only for a discrete set of time instants, while our approach gives such guarantees in continuous time and configuration domain (even though waypoint orientations are chosen from a discrete set). It is also worth emphasizing that the algorithms from [3, 19, 46, 53] plan piecewise-constant open-loop control signals, while our method generates a sequence of feedback control policies. The policies are ready to execute and they generate a continuously varying feedback control signal (everywhere except a finite sequence of switching points between the waypoints). Precise maneuvers can be planned with a relatively low number of waypoints due to consideration of a continuous planning domain as opposed to a finite set of motion primitives as proposed in methods utilizing grid search. On the other hand, design of a feedback controller for

the plan provided by a classical MILP-based planner generating open-loop controls is a nontrivial problem. The same observations can be made for other similar approaches such as [3] and [46]. Note that the controller-driven sampling-based methods such as [38, 39, 44] guarantee satisfaction of state and curvature constraints using discretization.

It is also important to note, that the proposed controller-driven planning method has certain limitations. Namely, it is characterized by a relatively large computational cost due to the fact, that it involves solving an optimization problem formulated as MILP, which is known to be NP-hard, as opposed to several other formulations. Furthermore, as stated in Section 3, our approach requires exact geometric knowledge of a robot motion environment. The presence of uncertainties is not explicitly considered. While uncertainties can be possibly handled by conservatively formulating state constraints and by online replanning, such an approach might be time consuming due to the computational cost of MILP optimization. However, in contrast to local planning methods, our planning method is designed as a global one.

### 3 Prerequisites and Problem Statement

#### 3.1 System Model and Notation

We model a mobile robot as a kinematic unicycle

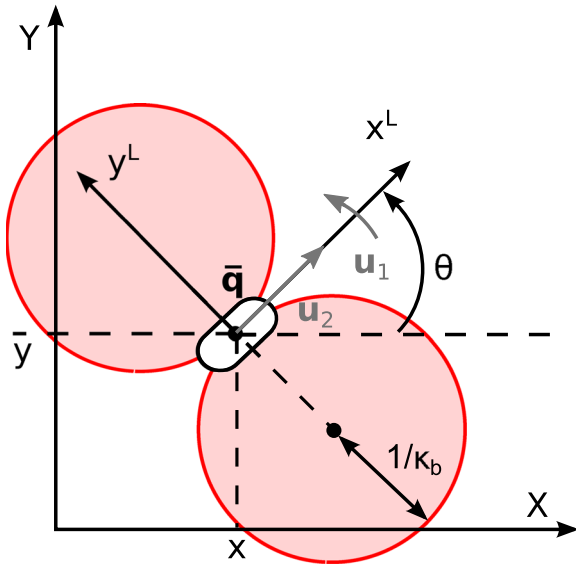
$$\dot{\mathbf{q}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} u_2 = [\mathbf{g}_1 \ \mathbf{g}_2(\theta)] \mathbf{u}, \quad (1)$$

$$\mathbf{q} = [\theta \ x \ y]^\top = [\theta \ \bar{\mathbf{q}}^\top]^\top \in \mathcal{Q}_{free} = \mathcal{P}_{free} \times \mathbb{R} \subseteq \mathcal{Q}, \quad (2)$$

with bounded curvature of motion

$$\forall t \geq 0 \quad |\kappa(t)| \triangleq \frac{|u_1(t)|}{|u_2(t)|} \leq \kappa_b, \quad \kappa_b < \infty, \quad (3)$$

where  $\mathcal{Q}$  denotes robot configuration space with an admissible subset  $\mathcal{Q}_{free}$  defined by state constraints expressed as an admissible positions set  $\mathcal{P}_{free}$  (defined in assumption A4), while  $\mathbf{q}$  is a configuration vector and  $\mathbf{u} = [u_1 \ u_2]^\top \in \mathcal{U} \subset \mathbb{R}^2$  denotes control input which consists of robot body angular velocity  $u_1$  and longitudinal velocity  $u_2$  of guidance point  $\bar{\mathbf{q}}$  shown in Fig. 1. Curvature bound  $\kappa_b$  is a prescribed constant resulting from robot motion task specification. It determines input constraint (3) defining the set



**Fig. 1** Considered curvature-constrained kinematic unicycle. Red circles denote paths of maximum admissible curvature  $\kappa_b$

of admissible control inputs  $\mathcal{U} = \{u \in \mathbb{R}^2 : |u_1| \leq \kappa_b |u_2|\}$ . Model (1) contains essential nonholonomic and curvature constraints imposed on robot motion. Thus, as argued in the Introduction, it can be utilized as a basic building block for modeling more complex kinematics and motion task specifications. For example, constraint (3) can be defined by steering-angle and articulation-angle bounds resulting from mechanical construction of car-like and tractor-trailer robots, respectively, or prescribed motion smoothness requirements resulting from, e.g., fragile payload carried by the robot.

In this paper we assume that the robot is driven through a sequence  $W$  of  $N + 1$  waypoint configurations, which is also called a plan. It is defined as follows:

$$W \triangleq \{q_{di}\}_{i=0}^N, \tag{4}$$

$$q_{di} = [\theta_{di} \ x_{di} \ y_{di}]^T = [\theta_{di} \ \bar{q}_{di}^T]^T \in \mathcal{Q}_{free},$$

where  $q_{di}$  is the  $i$ -th waypoint. During the subsequent analysis we often express variables in coordinate frame fixed at the  $(i + 1)$ -st waypoint. Symbols expressed in the local coordinate frame  $\{i + 1\}$  are denoted with upper index " $i + 1$ ". Variables denoted without an upper index are expressed in the global coordinate frame. For example,  $q_{di}^{i+1} = [\theta_{di}^{i+1} \ x_{di}^{i+1} \ y_{di}^{i+1}]^T = [\theta_{di}^{i+1} \ (\bar{q}_{di}^{i+1})^T]^T$  denotes the

$i$ -th waypoint expressed in coordinate frame fixed at the  $(i + 1)$ -st waypoint as shown in Fig. 2. Plan  $W$  splits robot motion execution process into  $N$  segments with index  $i$  denoting a motion segment number. We say that the robot is in the  $i$ -th motion segment, when it is driven towards  $(i + 1)$ -st waypoint. The current  $i$ -th segment is switched to the  $(i + 1)$ -st one at time  $t_{i+1}$ , as defined in assumption A2. We denote by  $\tau_i \triangleq t - t_i$  the relative motion-segment time.

### 3.2 Problem Statement and Assumptions

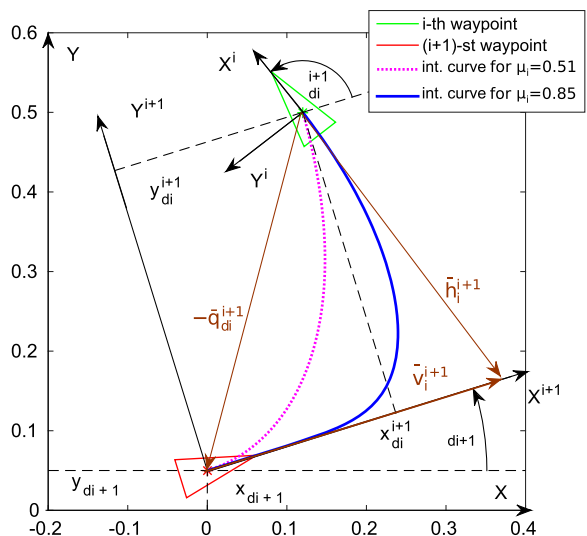
Let us investigate the controller-driven motion planning task dedicated to the VFO control law and defined as follows.

**Problem 1** Given an admissible subset of configuration space  $\mathcal{Q}_{free} \subseteq \mathcal{Q}$ , an initial configuration  $q(t = 0) \in \mathcal{Q}_{free}$ , and goal configuration  $q_{dN} \in \mathcal{Q}_{free}$ , the objective is to plan a waypoint configuration sequence  $W$  (cf. (4)) such that its sequential execution with the VFO control law applied to system (1) with constraint (3) guarantees that

$$\forall t \geq 0 \quad q(t) \in \mathcal{Q}_{free},$$

and

$$\bar{q}(t) \xrightarrow{t \rightarrow T} \bar{q}_{dN}, \quad \theta(t) \xrightarrow{t \rightarrow T} (\theta_{dN} \bmod 2\pi),$$



**Fig. 2** Paths of a unicycle driven by the VFO controller under nominal conditions during  $i$ -th motion segment execution for two values of  $\mu_i$  coefficient. (note:  $\bar{h}_i^{i+1}$  and  $\bar{v}_i^{i+1}$  have been illustrated for  $k_p = 1$ )

with  $T \in (0, \infty)$  being a control time-horizon.

Problem 1 shall be solved under the following assumptions:

- A0.  $\mathbf{q}_{d0} := \mathbf{q}(t = 0)$
- A1. Robot body is represented by a point in  $\mathcal{Q}_{free}$  and  $\mathcal{Q}_{free}$  is appropriately shrunk to account for actual robot body size.
- A2. The robot switches motion execution from the  $i$ -th motion segment to the  $(i + 1)$ -st one at time  $t_{i+1}$ , when  $\|\bar{\mathbf{q}}^{i+1}(t_{i+1})\| \leq \epsilon$ , where  $\|\cdot\|$  denotes the Euclidean norm, and  $\epsilon \geq 0$  is a design parameter.
- A3. Every robot orientation  $\theta$  is admissible.
- A4. The set of admissible robot positions  $\mathcal{P}_{free}$  is known a priori and given by sequence of  $M + 1$  convex polygons

$$\mathcal{P}_{free} \triangleq \{\mathcal{P}_j\}_{j=0}^M, \quad M > 0$$

with

$$\bar{\mathbf{q}}_{d0} = [x_{d0} \ y_{d0}]^T \notin \partial\mathcal{P}_{free} \quad \wedge$$

$$\bar{\mathbf{q}}_{dN} = [x_{dN} \ y_{dN}]^T \notin \partial\mathcal{P}_{free},$$

where  $\partial\mathcal{P}_{free}$  denotes the boundary of set  $\mathcal{P}_{free}$ .

- A5. Subsequent polygons share a transition edge  $\mathcal{E}_{j+1}$ , that is,
 
$$\mathcal{E}_{j+1} \triangleq \mathcal{P}_j \cap \mathcal{P}_{j+1}, \quad \forall j \in \{0, 1, \dots, M - 1\}.$$
- A6. No polygon in  $\mathcal{P}_{free}$  contains an edge co-linear with vertical line  $x = c$ ,  $c = const$  in the global frame.
- A7. Polygons shall be visited sequentially, i.e.,

$$\forall j \in \{0, 1, \dots, M - 1\} \quad \forall t \in [t_j, t_{j+1}] \quad \bar{\mathbf{q}}(t) \in \mathcal{P}_j, \text{ and}$$

$$\forall j \in \{0, 1, \dots, M - 1\} \quad \forall t \notin [t_j, t_{j+1}] \quad \bar{\mathbf{q}}(t) \notin \mathcal{P}_j,$$

where  $t_j$  denotes the time instant during which the robot passes from polygon  $\mathcal{P}_j$  to polygon  $\mathcal{P}_{j+1}$ , i.e.,  $\bar{\mathbf{q}}(t_j) \in \mathcal{E}_j$ .

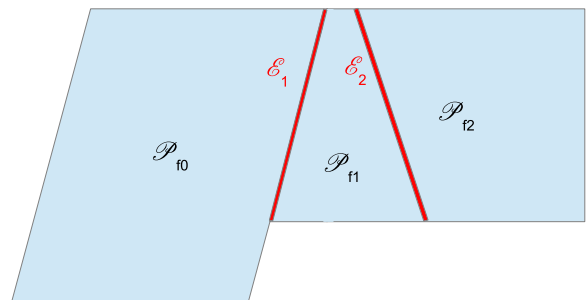
Assumptions A0 through A7 were made for two reasons. First, they simplify the description of state constraints used during planning. Second, they help in sound and efficient formulation of the optimization problem solved by the planner. Their conservativeness is often not an issue due to preference for planning safe maneuvers, which should keep the robot sufficiently far from obstacles in many practical scenarios.

Polygon sequence  $\mathcal{P}_{free}$  satisfying assumptions A4-A6, such as the one in Fig. 3, can be obtained from a motion environment map in two stages. First, environment map is decomposed into convex polygons utilizing algorithms such as [47] (for arbitrary polygons) and [14] (for grid-based maps). Second, the generated decomposition can be treated as a graph of polygons, which can be searched using, e.g., Dijkstra’s algorithm to find the sequence  $\mathcal{P}_{free}$ . In the next Subsection we will sketch our solution to Problem 1 exploiting a structure of the described environment.

### 3.3 General Description of the Proposed Solution

The planning algorithm proposed in the sequel is designed to generate waypoint sequence  $W$  using the controller-driven methodology. Thus, we begin with studying useful properties of the closed-loop system resulting from application of the VFO control law to system (1). Sufficient conditions for motion segment execution without violation of constraint (3) under specific nominal conditions are derived in Section 4.2. After that, Problem 1 is separated into two subproblems: waypoint orientations planning and waypoint positions planning. A method of determining a set of possible waypoint orientation choices and encoding those choices as binary decision variables is proposed in Section 5.2. Subsequently, we show that the remaining problem of waypoint positions planning is linear. Those two problems constitute a single MILP (mixed-integer linear program) with the integer part corresponding to waypoint orientations planning and the continuous linear part corresponding to waypoint positions planning.

In Section 5.4, the simplest case  $\mathcal{Q}_{free} = \mathcal{Q}$  is assumed and insights about VFO properties are used



**Fig. 3** Exemplary environment  $\mathcal{P}_{free}$  defined by a convex decomposition resulting in 3 polygons

to formulate waypoint positions planning for a given sequence of waypoint orientations as a MILP. This formulation constitutes the core of our approach.

In Section 5.5, conditions for satisfaction of state constraints given by a single convex polygon are derived and used to extend the MILP to solve Problem 1 for the case  $\mathcal{P}_{free} = \mathcal{P}_{f0}$ , which corresponds to planning in a single convex polygon.

In Section 5.6, the general case of Problem 1 is treated as a concatenation of planning problems for consecutive polygons in  $\mathcal{P}_{free}$ , which are coupled by additional constraints placing last waypoints in consecutive polygons on the corresponding transition edges  $\mathcal{E}_j$ .

In the next Section, the VFO control law will be recalled and analyzed to an extent necessary for understanding of the proposed planning algorithm.

### 4 Principles of the VFO Control Strategy

#### 4.1 VFO Controller for the Waypoint-Following Task

For simplicity of subsequent planning algorithm design, let us formulate the VFO control law for the waypoint-following task taken from [34] by expressing all of its components in the  $(i + 1)$ -st waypoint coordinate frame as follows:

$$\mathbf{u} \triangleq \begin{bmatrix} h_{ai}^{i+1} \\ \rho_i \sigma_i \cos(e_{ai}^{i+1}) \end{bmatrix}, \tag{5}$$

where:

$$\begin{aligned} h_{ai}^{i+1} &\triangleq k_a e_{ai}^{i+1} + \dot{\theta}_{ai}^{i+1}, \\ e_{ai}^{i+1} &\triangleq \theta_{ai}^{i+1} - \theta^{i+1}, \end{aligned} \tag{6}$$

$$\theta_{ai}^{i+1} \triangleq \text{Atan2c}(\sigma_i h_{yi}^{i+1}, \sigma_i h_{xi}^{i+1}) \in \mathbb{R}, \tag{7}$$

$$\bar{\mathbf{h}}_i^{i+1} = \begin{bmatrix} h_{xi}^{i+1} \\ h_{yi}^{i+1} \end{bmatrix} \triangleq -k_p \bar{\mathbf{q}}^{i+1} + \bar{\mathbf{v}}_i^{i+1}, \tag{8}$$

$$\bar{\mathbf{v}}_i^{i+1} \triangleq -k_p \mu_i \sigma_i \|\bar{\mathbf{q}}^{i+1}\| \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mu_i \triangleq \frac{\eta_i}{k_p},$$

$$\rho_i \triangleq \begin{cases} U_2 & \text{for } i < N - 1, \\ U_2 \frac{\|\bar{\mathbf{h}}_i^{i+1}\|}{\|\bar{\mathbf{h}}_i^{i+1}(t_i)\|} & \text{for } i = N - 1, \end{cases} \tag{9}$$

where design parameters  $k_a, k_p, U_2 > 0$  and  $\eta_i \in (0, k_p)$  are constant, while operator  $\text{Atan2c}(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  introduced in [8] corresponds to the continuous version of  $\text{Atan2}(\cdot, \cdot)$  function. The orienting control  $u_1$  aligns robot configuration with the convergence vector field  $\mathbf{h}_i^{i+1} \triangleq \left[ h_{ai}^{i+1} (\bar{\mathbf{h}}_i^{i+1})^\top \right]^\top$ , the integral curves of which tend to the  $(i+1)$ -st waypoint. This alignment results from tracking of auxiliary orientation  $\theta_{ai}^{i+1}$ , which corresponds to minimizing of the auxiliary orientation error  $e_{ai}^{i+1}$ . Convergence vector field  $\mathbf{h}_i^{i+1}$  can be shaped using relative directing coefficient  $\mu_i \in (0, 1)$ , which is a normalized version of directing coefficient  $\eta_i \in (0, k_p)$  originally defined in [8]. As shown in Fig. 2, relative directing coefficient  $\mu_i$  influences a convergence character of orientation  $\theta$  to waypoint orientation  $\theta_{di+1}$ . Motion sense in the  $i$ -th motion segment is determined by a decision variable  $\sigma_i \in \{-1, 1\}$ , where  $\sigma_i = -1$  forces backward motion while  $\sigma_i = 1$  forces forward motion. Robot longitudinal velocity is determined by velocity profile  $\rho_i$ , which can be shaped flexibly as stated in [34]. Proposed definition (9) is designed to ensure that robot will travel through waypoints with prescribed constant velocity  $U_2$  and gradually stop at final waypoint configuration  $\mathbf{q}_{dN}$ . For brevity, we omit the design of more complex velocity profiles and refer the reader, e.g., to [52] for algorithms computing time-optimal profiles with prescribed intermediate velocities. Design parameter  $k_a$  can be chosen heuristically as  $k_a = 2 U_2$  to make curvature of robot motion independent of chosen  $U_2$  value when  $e_{ai}^{i+1} \neq 0$ .

Relevant stability guarantees and convergence properties of the closed-loop system with the VFO control law in the absence of curvature and state constraints are summarized by the following lemma proven in [34].

**Lemma 1** *Assuming  $\kappa_b = \infty$ ,  $\mathcal{Q}_{free} = \mathcal{Q}$  and waypoint switching strategy given in assumption A2, the application of control law (5) to system (1) yields the guarantees:*

- G1. *Robot orientation  $\theta$  converges exponentially fast to the auxiliary orientation  $\theta_{ai}$ , which converges to waypoint orientation  $\theta_{di+1}$  in the sense  $\theta_{ai}(t) \rightarrow (\theta_{di+1} \bmod 2\pi)$  as  $\bar{\mathbf{q}}(t) \rightarrow \bar{\mathbf{q}}_{di}$ .*



G2. Robot position  $\bar{\mathbf{q}}$  converges to the assumed neighborhood of waypoint position  $\bar{\mathbf{q}}_{di}$  in finite time, that is,

$$\forall 0 \leq i < N \quad \exists t_i < \infty : \quad \|\bar{\mathbf{q}}_{di} - \bar{\mathbf{q}}(t_i)\| \leq \epsilon.$$

G3. Robot configuration  $\mathbf{q}$  converges asymptotically to the final waypoint configuration  $\mathbf{q}_{dN}$  in the sense

$$\bar{\mathbf{q}}(t) \xrightarrow{t \rightarrow T} \bar{\mathbf{q}}_{dN}, \quad \theta(t) \xrightarrow{t \rightarrow T} (\theta_{dN} \bmod 2\pi),$$

where  $T \in (0, \infty)$  is a control time-horizon.

Let us now introduce the notions of *nominal* and *non-nominal* conditions utilized in further considerations. If one assumes switching distance  $\epsilon = 0$  in waypoint switching condition from assumption A2 and perfect tracking of auxiliary orientation  $\theta_{ai}^{i+1}$ , i.e.,  $e_{ai}^{i+1}(t_i) = 0 \Rightarrow e_{ai}^{i+1}(t) \equiv 0$ , one says that the motion execution takes place under *nominal conditions*. In those conditions, consecutive waypoint positions are reached accurately by the robot. Obviously, nominal conditions are impossible to achieve during the practical motion execution due to sensing noise, imperfect actuation, etc., but they can be very closely approximated. Thus, we will utilize the nominal conditions in the subsequent formal considerations, since they are formally possible. Violation of the nominal conditions will be addressed in Section 6.

#### 4.2 Characteristic Properties of the VFO Control System

Recent analysis of closed-loop dynamics with the VFO control law under nominal conditions exposed a number of features useful for motion planning. They will be presented in the form of three main properties.

*Property 1* For  $y_{di}^{i+1} \neq 0$  application of control law (5) to system (1) under nominal conditions implies that robot position  $\bar{\mathbf{q}}$  converges to the  $(i + 1)$ -st waypoint position  $\bar{\mathbf{q}}_{di+1}$  along the integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  given by

$$x^{i+1} = \frac{\sigma_i |y^{i+1}|}{2} \left[ \left( \frac{y^{i+1}}{p_i} \right)^{\mu_i} - \left( \frac{y^{i+1}}{p_i} \right)^{-\mu_i} \right], \quad (10)$$

with

$$p_i \triangleq y_{di}^{i+1} \exp \left( \frac{|\operatorname{arsinh}(x_{di}^{i+1}/y_{di}^{i+1})|}{\mu_i} \right), \quad (11)$$

where  $p_i$  is treated a parameter. In case when  $y_{di}^{i+1} = 0$ , the integral curve is a line  $y^{i+1} = 0$ .

*Remark 1* The parameter  $p_i$  in (10) corresponds to such coordinate  $y^{i+1} > 0$  where  $f_i^{i+1}(y^{i+1}, p_i) = 0$ , that is, it describes a point at which curve (10) crosses the y-axis of the  $(i + 1)$ -st waypoint coordinate frame.

*Proof* Let us assume temporarily that  $\rho_i = \|\bar{\mathbf{h}}_i^{i+1}\|$ . Such an assumption can be made without loss of generality, because an integral curve of the closed-loop system is determined by the value of auxiliary orientation  $\theta_{ai}^{i+1}$  given by (7), which (under nominal conditions) does not depend on the velocity profile  $\rho_i$ . Under nominal conditions, substitution of  $\rho_i = \|\bar{\mathbf{h}}_i^{i+1}\|$  into (5) yields  $\dot{\bar{\mathbf{q}}}^{i+1} = \bar{\mathbf{h}}_i^{i+1}$ , which implies

$$\dot{y}^{i+1} \stackrel{(8)}{=} -k_p y^{i+1} \Rightarrow y^{i+1}(\tau_i) = y_{di}^{i+1} \exp(-k_p \tau_i), \quad (12)$$

$$\dot{x}^{i+1} \stackrel{(8)}{=} -k_p \left( x^{i+1} + \sigma_i \mu_i \|\bar{\mathbf{q}}^{i+1}\| \right). \quad (13)$$

One can solve (13) by substitution  $x^{i+1} = z_i y^{i+1}$  that is possible only for  $y^{i+1} \neq 0$ , which in turn imposes  $y_{di}^{i+1} \neq 0$ . Thus, we will analyze the two cases corresponding to  $y_{di}^{i+1} = 0$  and  $y_{di}^{i+1} \neq 0$ , respectively.

In the case of  $y_{di}^{i+1} = 0$ , one has  $y^{i+1}(\tau_i) = 0$  immediately from (12), which means that the integral curve must be  $y^{i+1} = 0$ .

In the case when  $y_{di}^{i+1} \neq 0$ , one substitutes  $x^{i+1} = z_i y^{i+1}$  in (13) and after some algebra obtains:

$$\dot{z}_i y^{i+1} + \dot{y}^{i+1} z_i = -k_p y^{i+1} \left( z_i + \sigma_i \mu_i \operatorname{sign}(y^{i+1}) \sqrt{(z_i)^2 + 1} \right),$$

which can be further simplified using (12) to:

$$(\dot{z}_i - k_p z_i) y^{i+1} = -k_p y^{i+1} \left( z_i + \sigma_i \mu_i \operatorname{sign}(y^{i+1}) \sqrt{(z_i)^2 + 1} \right).$$

Now one can arrange the variables as follows:

$$\dot{z}_i = -k_p \sigma_i \mu_i \operatorname{sign}(y^{i+1}) \sqrt{(z_i)^2 + 1},$$

separate them,

$$\frac{dz_i}{\sqrt{(z_i)^2 + 1}} = -k_p \sigma_i \mu_i \text{sign}(y^{i+1}) d\tau_i,$$

and integrate by sides to obtain

$$\text{arsinh}(z_i) = -k_p \sigma_i \mu_i \text{sign}(y^{i+1}) \tau_i + C_i,$$

where  $C_i = \text{const}$  is an integration constant. After computation of  $C_i$  from boundary conditions and returning to original variables one arrives at the solution:

$$x^{i+1} = \sinh \left[ -k_p \sigma_i \text{sign}(y^{i+1}) \mu_i \tau_i + \text{arsinh}(x_{di}^{i+1}/y_{di}^{i+1}) \right] y^{i+1}. \tag{14}$$

It can be seen that there are infinitely many pairs  $(x_{di}^{i+1}, y_{di}^{i+1})$  corresponding to the same curve (14). Now we will compute  $p_i$ , such that a pair  $(p_i, \sigma_i)$  corresponds to a single integral curve, which is a function of  $y^{i+1}$ , as opposed to time. Let us start by considering the interpretation of  $p_i$  given in Remark 1. Since  $p_i$  corresponds to such  $y^{i+1} > 0$  along the integral curve (14) that  $f_i^{i+1}(y^{i+1}, p_i) = 0$ , we look for a  $\tau_i$  corresponding to  $f_i^{i+1}(y^{i+1}, p_i) = 0$ :

$$f_i^{i+1}(y^{i+1}, p_i) = 0 \stackrel{(14)}{\Rightarrow} \tau_i = \frac{\text{arsinh}(x_{di}^{i+1}/y_{di}^{i+1})}{k_p \mu_i \sigma_i}. \tag{15}$$

Substitution of (15) into (12) yields (11) corresponding to  $y^{i+1}$  satisfying the interpretation of  $p_i$ . If one eliminates  $\tau_i$  from (14) utilizing (12), and once again uses the interpretation of  $p_i$  from Remark 1 to substitute  $y_{di}^{i+1} = p_i$  and  $x_{di}^{i+1} = 0$  in the resulting expression, then after some algebra the parametrized integral curve (10) is obtained. Observe that the purpose of substitution  $y_{di}^{i+1} = p_i, x_{di}^{i+1} = 0$  is to change parameters of curve (14) from  $x_{di}^{i+1}, y_{di}^{i+1}$  to  $p_i$ .  $\square$

**Corollary 1** For every  $\alpha > 0$  the scaling relation  $f_i^{i+1}(\alpha y^{i+1}, \alpha p_i) = \alpha f_i^{i+1}(y^{i+1}, p_i)$  is satisfied by the integral curve (10).

*Proof* It is known from Remark 1 and proof of Property 1 that  $p_i$  is, by construction, a specific value of

$y^{i+1}$ . It means that multiplication of  $y^{i+1}$  in (10) by factor  $\alpha$  requires also multiplication of  $p_i$ , by factor  $\alpha$ . Given this observation one can write:

$$\begin{aligned} f_i^{i+1}(\alpha y^{i+1}, \alpha p_i) &= \frac{\sigma_i |\alpha y^{i+1}|}{2} \left[ \left( \frac{\alpha y^{i+1}}{\alpha p_i} \right)^{\mu_i} - \left( \frac{\alpha y^{i+1}}{\alpha p_i} \right)^{-\mu_i} \right] \\ &= \frac{\sigma_i |\alpha y^{i+1}|}{2} \left[ \left( \frac{y^{i+1}}{p_i} \right)^{\mu_i} - \left( \frac{y^{i+1}}{p_i} \right)^{-\mu_i} \right] \\ &= \alpha f_i^{i+1}(y^{i+1}, p_i), \end{aligned}$$

which proves the postulated relation.  $\square$

The easily verified scaling relation described by Corollary 1 is key in our planning approach. Still, to effectively plan robot motion under curvature constraint (3) one must analyze curvature of integral curve (10). Results of this analysis are summarized in the following property.

*Property 2* Under nominal conditions and for  $\mu_i \in (0.5, 1)$ , curvature  $\check{\kappa}_i(y^{i+1}, p_i)$  of the integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  (cf. (10)) is bounded, that is,

$$|\check{\kappa}_i(y^{i+1}, p_i)| \leq K_i = \frac{|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|}{|p_i|} \leq \infty, \tag{16}$$

where curvature  $\check{\kappa}_i(y^{i+1}, p_i)$  is given by

$$\check{\kappa}_i(y^{i+1}, p_i) = \frac{y^{i+1} \left( \mu_i \sigma_i + \mu_i^2 \frac{f_i^{i+1}(y^{i+1}, p_i)}{r_i} \right)}{\sigma_i(r_i)^2 \left( 1 + \mu_i^2 + 2\sigma_i \mu_i \frac{f_i^{i+1}(y^{i+1}, p_i)}{r_i} \right)^{3/2}}, \tag{17}$$

$$r_i \triangleq \sqrt{\left( f_i^{i+1}(y^{i+1}, p_i) \right)^2 + (y^{i+1})^2}, \tag{18}$$

and point  $\tilde{y}^{i+1}$  at which curvature  $\check{\kappa}_i(y^{i+1}, p_i)$  of curve  $f_i^{i+1}(y^{i+1}, p_i)$  attains its extremum is given by

$$\tilde{y}^{i+1} = \left( p_1 - \frac{-6\mu_i^3 + \mu_i^2 + 2\mu_i + 3}{(6\mu_i + 3)(\mu_i + 1)^2} + p_2 \right)^{1/2\mu_i}, \tag{19}$$

with

$$p_1 = \left( \frac{4}{3} p_6 - p_3 \right)^{1/3},$$

$$\begin{aligned}
 p_2 &= \frac{4\mu_i^2 (18\mu_i^4 + 3\mu_i^3 - 17\mu_i^2 - 11\mu_i + 7)}{9p_1 (\mu_i + 1)^4 (2\mu_i + 1)^2}, \\
 p_3 &= \frac{p_4^3}{27(2\mu_i + 1)^3(\mu_i + 1)^9} + p_7 - p_5, \\
 p_4 &= -6\mu_i^4 - 5\mu_i^3 + 3\mu_i^2 + 5\mu_i + 3, \\
 p_5 &= \frac{p_4(-6\mu_i^4 + 5\mu_i^3 + 3\mu_i^2 - 5\mu_i + 3)}{6(2\mu_i + 1)^2(\mu_i + 1)^6}, \\
 p_6 &= \sqrt{\frac{\mu_i^6(\mu_i - 1)^3(-36\mu_i^4 + 33\mu_i^2 - 29)}{(2\mu_i + 1)^4(\mu_i + 1)^9}}, \\
 p_7 &= \frac{(2\mu_i - 1)(\mu_i - 1)^3}{(4\mu_i + 2)(\mu_i + 1)^3}.
 \end{aligned}$$

*Proof* One can obtain relation (17), corresponding to the curvature of integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  by considering (3) under nominal conditions as follows:

$$\check{\kappa}_i \stackrel{(3)}{=} \frac{u_1}{u_2} \stackrel{(5)}{=} \frac{\dot{\theta}_{ai}^{i+1}}{\sigma_i \rho_i}.$$

After computing  $\dot{\theta}_{ai}^{i+1}$  by time-differentiation of  $\theta_{ai}^{i+1}$  given by (7), substituting  $\theta^{i+1} = \theta_{ai}^{i+1}$  (upon the assumption of nominal conditions) and  $x^{i+1} = f_i^{i+1}(y^{i+1}, p_i)$  (by virtue of Property 1) into the resulting expression, one applies some algebra to obtain (17).

It can be seen from (17) that, similarly to the integral curve (10), for every  $\alpha > 0$  relation

$$\check{\kappa}_i(\alpha y^{i+1}, \alpha p_i) = \frac{\check{\kappa}(y^{i+1}, p_i)}{\alpha} \tag{20}$$

holds, because using Corollary 1 one can write

$$\check{\kappa}_i(\alpha y^{i+1}, \alpha p_i) \stackrel{(17)}{=} \frac{\alpha y^{i+1} \left( \mu_i + \mu_i^2 \sigma_i \frac{\alpha f_i^{i+1}(y^{i+1}, p_i)}{\alpha r_i} \right)}{\alpha^2 (r_i)^2 \left( 1 + \mu_i^2 + 2\sigma_i \mu_i \frac{\alpha f_i^{i+1}(y^{i+1}, p_i)}{\alpha r_i} \right)^{3/2}},$$

which can be reduced to  $\check{\kappa}_i(x^{i+1}, y^{i+1})/\alpha$ . It is known from Corollary 1 that scaling  $y^{i+1}$  in (10) by a positive factor  $\alpha$  corresponds to scaling of  $p_i$  and  $f_i^{i+1}(y^{i+1}, p_i)$  by the same factor, hence after considering relation (20) one concludes that, for any  $\alpha > 0$ , curvature of robot motion along curve (10) with parameter  $p_i$  multiplied by  $\alpha$  corresponds to curvature of robot motion along the original curve (10) divided by  $\alpha$ . It means that one can compute the bound  $K_i$  as given by (16) for an integral curve (10) with any  $p_i$

if the supremum of curvature of motion  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$  attained at point  $\tilde{y}^{i+1}$  for the specific integral curve (10) with  $p_i = 1$  is known. Thus, we will now assume  $p_i = 1$  and focus on finding the supremum  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$ .

To find the supremum  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$ , one must first analyze the behavior of  $\check{\kappa}_i(y^{i+1}, 1)$  as  $y^{i+1} \rightarrow 0$ . For simplicity, let us consider the limits of the following two auxiliary expressions for the particular case  $p_i = 1$ :

$$\begin{aligned}
 &\lim_{y^{i+1} \rightarrow 0} \frac{f_i^{i+1}(y^{i+1}, 1)}{r_i} = \\
 \stackrel{(10)}{=} &\lim_{y^{i+1} \rightarrow 0} \frac{0.5\sigma_i ((y^{i+1})^{2\mu_i} - 1)}{\sqrt{0.25 ((y^{i+1})^{4\mu_i} + 1 - 2(y^{i+1})^{2\mu_i}) + (y^{i+1})^{2\mu_i}}} \\
 &= -\sigma_i,
 \end{aligned}$$

and

$$\begin{aligned}
 &\lim_{y^{i+1} \rightarrow 0} \frac{y^{i+1}}{(r_i)^2} \stackrel{(17)}{=} \lim_{y^{i+1} \rightarrow 0} \frac{4}{(y^{i+1})^{1-2\mu_i} + (y^{i+1})^{1+2\mu_i} + 2y^{i+1}} \\
 &= \kappa_{li},
 \end{aligned}$$

where

$$\begin{aligned}
 \mu_i < 0.5 &\Rightarrow |\kappa_{li}| = \infty, \\
 \mu_i = 0.5 &\Rightarrow |\kappa_{li}| = 4, \\
 \mu_i > 0.5 &\Rightarrow |\kappa_{li}| = 0,
 \end{aligned}$$

with the value of  $\kappa_{li}$  for  $\mu_i = 0.5$  computed under the assumption of  $0^0 = 1$  as suggested in [7]. Using computed auxiliary limits one can write

$$\lim_{y^{i+1} \rightarrow 0} \check{\kappa}_i(y^{i+1}, 1) \stackrel{(17)}{=} \frac{\kappa_{li}(\mu_i \sigma_i - \sigma_i \mu_i^2)}{\sigma_i(1 + \mu_i^2 - 2\mu_i)^{3/2}},$$

which in light of scaling relation (20) means that  $\forall p_i \lim_{y^{i+1} \rightarrow 0} \check{\kappa}_i(y^{i+1}, p_i) = 0$  for  $\mu_i > 0.5$ .

Thus, assuming  $\mu_i \in (0.5, 1)$  one concludes that the supremum  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$  occurs at the point  $\tilde{y}^{i+1}$  where  $\partial |\check{\kappa}_i(y^{i+1}, 1)| / \partial y^{i+1} = 0$ . It remains to compute  $\tilde{y}^{i+1}$  by differentiation of curvature relation (17) w.r.t.  $y^{i+1}$ , which yields:

$$\frac{\partial |\check{\kappa}_i(f_i^{i+1}(y^{i+1}, p_i), y^{i+1})|}{\partial y^{i+1}} = \frac{\check{N}_i}{\check{M}_i} = 0,$$

where  $\check{M}_i \neq 0$  and  $\check{N}_i$  are polynomials of  $y^{i+1}$ . Since one can show that  $\check{M}_i \neq 0$ , it suffices to find a solution of  $\check{N}_i = 0$ , which can be obtained by substitution  $v_i = (y^{i+1})^{2\mu_i}$  and explicit solution of resultant cubic polynomial, yielding two conjugate complex roots and

one real root  $\tilde{y}^{i+1}$  given by (19). Existence of a single real root means that  $\tilde{y}^{i+1}$  is exactly the point at which curvature along considered specific integral curve (10) with  $p_i = 1$  attains its extremum  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$ . Since  $|\check{\kappa}_i(\tilde{y}^{i+1}, 1)|$  corresponds to curvature bound  $K_i$  for the case of  $p_i = 1$ , one can compute the curvature bound  $K_i$  for an arbitrary value of  $p_i$  by applying the scaling relation (20), which yields (16).  $\square$

**Corollary 2** Under nominal motion conditions, constraint (3) is satisfied in the  $i$ -th motion segment iff  $\tilde{q}_{di}^{i+1}$  is such that  $|p_i| \geq p_{bi}$ , where  $p_{bi} = |\check{\kappa}_i(\tilde{y}^{i+1}, 1)|/\kappa_b$  (cf. (16)).

Corollary 2 results from substitution of  $K_i = \kappa_b$  in (16) to find a boundary value of  $p_i$  denoted by  $p_{bi}$ . It is graphically illustrated in Fig. 4. The sets of admissible integral curves shown in blue and magenta, respectively, have the borders, which are integral curves (10) evaluated for  $p_{bi}$ . While Properties 1 and 2 along with Corollary 2 are strong motion planning tools, there is another useful characteristic of the VFO control law anticipated by Corollary 1 and captured by the following property.

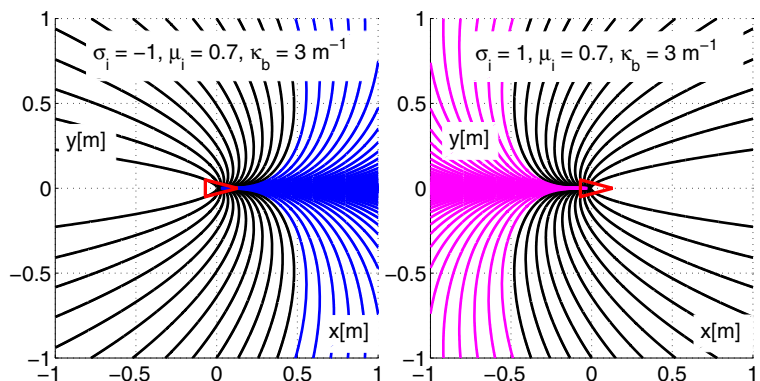
*Property 3* Under the nominal conditions, let us assume  $\sigma_i = -\text{sign}(x_{di}^{i+1})$ , and  $|\theta_{di}^{i+1}| < \pi/2$ . Then for a given waypoint orientation  $\theta_{di}^{i+1}$  the position  $\tilde{q}_{di}^{i+1}$  of the  $i$ -th waypoint lies on a line

$$y_{di}^{i+1} = a_i^{i+1} x_{di}^{i+1},$$

where

$$a_i^{i+1} = \frac{\phi_i \mu_i \sqrt{\phi_i^2 - \phi_i^2 \mu_i^2 + 1 - \phi_i}}{\phi_i^2 \mu_i^2 - 1}, \quad \phi_i = \tan(\theta_{di}^{i+1}). \quad (21)$$

**Fig. 4** Integral curves (10) for  $q_{di} = \mathbf{0}$  with backward- (left-hand side) and forward- (right-hand side) motion sense. Blue and magenta curves have absolute values of curvature lower than  $\kappa_b$ . Therefore, under nominal conditions, if initial position  $\tilde{q}_{di}^{i+1}$  lies in blue or magenta region, then constraint (3) is satisfied within the  $i$ -th motion segment



*Proof* From the definition of nominal conditions we know that  $e_{ai}^{i+1} \equiv 0$ , therefore

$$\begin{aligned} \theta_{di}^{i+1} &= \theta_{di}^{i+1}(t_i) \\ &\stackrel{(7)}{=} \text{Atan2c}(\sigma_i h_{yi}^{i+1}(t_i), \sigma_i h_{xi}^{i+1}(t_i)), \end{aligned}$$

which under assumption  $|\theta_{di}^{i+1}| < \pi/2$  yields

$$\theta_{di}^{i+1} = \arctan\left(\frac{h_{yi}^{i+1}}{h_{xi}^{i+1}}\right) \stackrel{(8)}{=} \arctan\left(\frac{y_{di}^{i+1}}{x_{di}^{i+1} + \sigma_i \mu_i \|\tilde{q}_{di}^{i+1}\|}\right).$$

Along a straight line  $y_{di}^{i+1} = a_i^{i+1} x_{di}^{i+1}$  one obtains

$$\theta_{di}^{i+1} = \arctan\left(\frac{a_i^{i+1}}{1 - \mu_i \sqrt{1 + (a_i^{i+1})^2}}\right),$$

which after taking  $\phi_i = \tan(\theta_{di}^{i+1})$ , leads to

$$\phi_i = \frac{a_i^{i+1}}{1 - \mu_i \sqrt{1 + (a_i^{i+1})^2}},$$

which can be solved w.r.t.  $a_i^{i+1}$  yielding one positive and one negative solution. The choice of the appropriate solution is possible thanks to the assumptions of  $\sigma_i = -\text{sign}(x_{di}^{i+1})$ , and  $|\theta_{di}^{i+1}| < \pi/2$ , which ensure that the integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  is contained entirely in a single quadrant of coordinate frame  $\{i + 1\}$  (cf. Figs. 4 and 7 for graphical interpretation). It means that given  $\text{sign}(\theta_{di}^{i+1})$  and  $\sigma_i$  one can clearly determine the appropriate quadrant of coordinate frame  $\{i + 1\}$  and the value of  $\text{sign}(a_i^{i+1})$  for segments contained in this quadrant. Under the assumption of  $|\theta_{di}^{i+1}| < \pi/2$ , one concludes that

sign  $(\phi_i) = \text{sign}(a_i^{i+1})$ . Thus, from the two obtained solutions one chooses the one satisfying  $\text{sign}(\phi_i) = \text{sign}(a_i^{i+1})$ . That solution is (21).  $\square$

In the sequel we will use properties of the VFO control law proven above to design the VFO-driven motion planning algorithm solving Problem 1.

### 5 The VFO-Driven Motion Planning Algorithm

#### 5.1 Formulation of Problem 1 as a General Optimization Problem

In what follows we assume  $\sigma_i = -\text{sign}(x_{di}^{i+1})$  and  $|\theta_{di}^{i+1}| < \pi/2$  to utilize Property 3. Using results from Section 4.2 one can formulate the task of finding a waypoint sequence corresponding to a solution of Problem 1 as the following dynamical nonlinear optimization problem:

$$\min_{\{q_{di}^{i+1}\}_{i=0}^{N-1}} \sum_{i=0}^{N-1} z_i(q_{di}^{i+1}), \tag{22}$$

subject to constraints:

$$\dot{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} u_2, \tag{23}$$

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} h_{di}^{i+1} \\ \rho_i \sigma_i \cos(\theta_{di}^{i+1}) \end{bmatrix}, \tag{24}$$

$$\forall t \in [0, T] \quad q(t) \in Q_{free}, \tag{25}$$

$$|\theta_{di}^{i+1}| \leq \frac{\pi}{2} \quad \forall i \in \{0, \dots, N-1\}, \tag{26}$$

$$|p_i| \geq p_{bi} \quad \forall i \in \{0, \dots, N-1\}, \tag{27}$$

$$\theta_{d0} = \theta_{dN} + \sum_{i=0}^{N-1} \theta_{di}^{i+1}, \tag{28}$$

$$\bar{q}_{d0} = \bar{q}_{dN} + \sum_{i=0}^{N-1} R(\theta_{di+1}) \bar{q}_{di}^{i+1}, \tag{29}$$

$$R(\theta) \triangleq \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix},$$

where  $T \in (0, \infty)$  denotes a control time horizon and  $R$  is a 2D rotation matrix. Cost function  $z_i$  in (22)

represents an arbitrary additive cost-to-come measure such as, e.g., predicted robot path length. Differential constraint (23) is imposed by assumed kinematic model of a unicycle (cf. (1)). Since Problem 1 postulates application of a feedback control law, the control input  $u$  of system (23) is given by (24) corresponding to the VFO control law (cf. (5)). Constraint (27) is formulated to guarantee satisfaction of motion curvature constraint (3) under nominal conditions by virtue of Corollary 2. An effective formulation of state constraints given by (25) is derived later in Section 5.5. For now, in anticipation of that formulation we impose an additional constraint (26) to guarantee that Property 3 can be used. Boundary conditions are enforced by equality constraints (28) and (29), which reflect the requirement that planned waypoint sequence must start and end at prescribed configurations  $q_{d0}$  and  $q_{dN}$ , respectively. Observe that constraints (28) and (29) result from utilization of homogeneous 2D transformations to represent prescribed initial configuration  $q_{d0}$  in terms of prescribed goal configuration  $q_{dN}$  and decision variables  $q_{di}^{i+1}$ , which correspond to waypoint configurations expressed in local coordinate frames of subsequent waypoints.

*Remark 2* Note that differential constraint (23) imposed by unicycle kinematics is represented in continuous time and robot configuration domains, as opposed to discretized state transition equations assumed in various methods available in the literature. Constraints (23) and (25) necessitate constraint satisfaction checking in continuous time and robot configuration domains, which is possible thanks to the application of the VFO feedback control law and utilization of its properties developed in Section 4. In our approach, we opted for MILP formulation to perform the optimization in (at least partially) a continuous domain and exploit linearities visible in our optimization problem. It is worth noting that while the discontinuous portion of our MILP model corresponds to discretization of robot orientation values at waypoints, the constraints (23)-(29) are satisfied in a continuous domain (i.e., no discretization is performed) and the planned VFO feedback control policies are also continuous. To the best of our knowledge, straightforward application of global optimization algorithms such as sampling-based or grid-search-based methods to optimization problem given by (22)-(29) would require discretization in time, control or

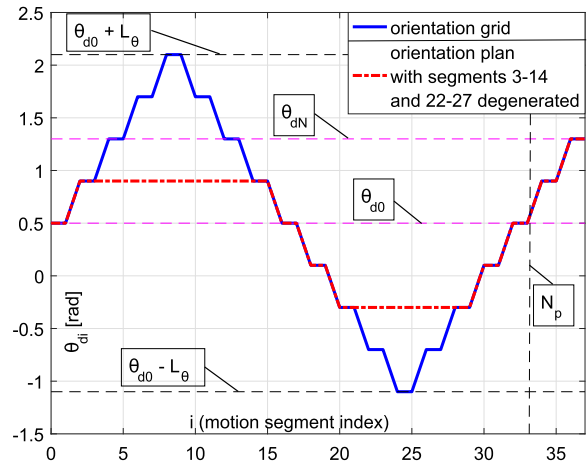
configuration domain, which is incompatible with our problem statement.

For a prescribed upper bound on number  $N$  of motion segments, one can attempt to solve the presented nonlinear optimization problem using properties from Section 4 and continuous nonlinear optimization algorithms. To the best of our knowledge this problem is in general non-convex. As a result if a nonlinear optimization solver is directly applied, its solutions in general can be only locally optimal, while the optimization process can be costly and highly dependent on initial guess, which for some optimization algorithms must be a feasible. To show non-convexity of the proposed optimization problem formulation it suffices to notice that even if one assumes convexity of  $z_i(q_{di}^{i+1})$  and neglects state constraints (25), there remain two constraints violating assumptions of convex optimization problems. First, equality constraint (29) is non-linearly dependent on waypoint angles  $\theta_{di}^{i+1}$ . Second, after representing (27) in terms of  $q_{di}^{i+1}$  by using (10) and investigating second order partial derivatives of constraint (27) w.r.t.  $\theta_{di}^{i+1}$ ,  $x_{di}^{i+1}$  and  $y_{di}^{i+1}$ , one determines that its feasible set is non-convex.

To overcome the described issue of non-convexity, we convert the above nonlinear optimization problem to a MILP formulation. The conversion is performed by applying a waypoint orientation discretization and selection algorithm described in Section 5.2. Upon application of this algorithm the number of motion segments  $N$  is determined, while constraints (28) and (26) are represented by linear constraints with auxiliary binary decision variables. Subsequently, boundary position constraints (29) and curvature constraints (27) are represented as linear constraints in Section 5.4. After that, we give an efficient and exact linear formulation of state constraints (25) represented by a single convex polygon in Section 5.5, which is later generalized to a sequence of convex polygons in Section 5.6.

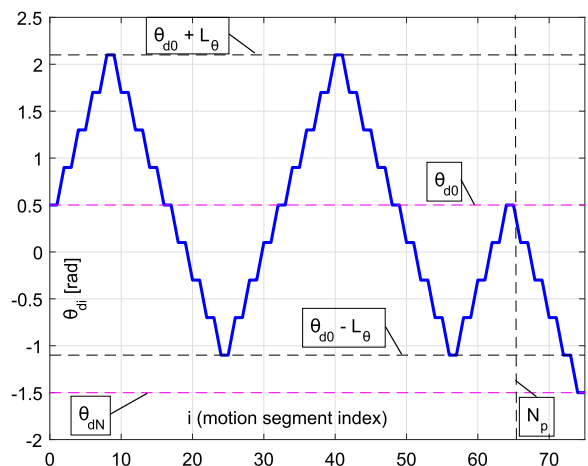
### 5.2 Discretization and Preparation of a Grid for Waypoint Orientations

The goal of the proposed waypoint discretization algorithm is to generate a *waypoint orientation grid*, determining a prescribed finite set of waypoint



**Fig. 5** Exemplary orientation grid and orientation plan for  $\theta_{d0} = 0.5$  rad,  $\theta_{dN} = 1.3$  rad,  $N_L = 1$ ,  $L_\theta = 1.6$  rad, and  $\Delta\theta = 0.4$  rad. Orientation plan is generated from the orientation grid by degeneration of automatically selected motion segments

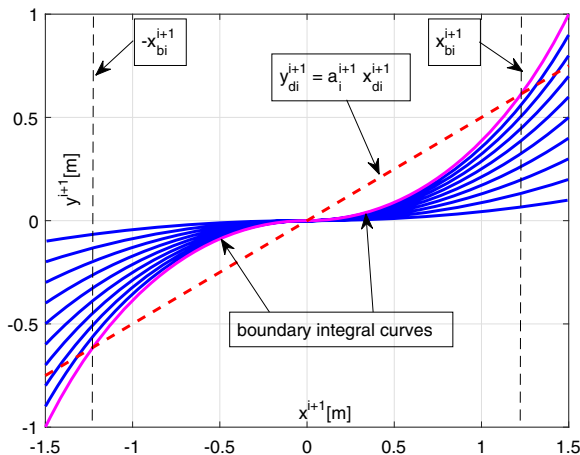
orientation sequences called *waypoint orientation plans*, from which the planner will be choosing an optimal one. Given a waypoint orientation grid, one can encode the choice of optimal waypoint orientation plan using a finite number of binary decision variables in the MILP as shown in Section 5.3. Exemplary orientation grids are presented in Figs. 5 and 6. Waypoint orientation  $\theta_{di}$  is illustrated as a function of motion segment index. Notice that, while the number  $N$  of motion segments is fixed, some motion segments can be *degenerated* as shown in Fig. 5. When the  $i$ -th motion segment is degenerated its length is forced to



**Fig. 6** Exemplary orientation grid for  $\theta_{d0} = 0.5$  rad,  $\theta_{dN} = -1.5$  rad,  $N_L = 2$ ,  $L_\theta = 1.6$  rad, and  $\Delta\theta = 0.4$  rad

0, which means that during plan execution  $i$ -th and  $(i + 1)$ -st waypoints will have equal positions and by virtue of assumption A2 the controller will switch instantaneously to the  $(i + 2)$ -nd waypoint as if the  $(i + 1)$ -st waypoint did not exist. Such an instantaneous switch implies that current waypoint orientation will never change to the  $i$ -th waypoint orientation given by original waypoint grid as illustrated by difference between the plots of an orientation *grid* and an orientation *plan* shown in Fig. 5. An optimal waypoint orientation plan is automatically chosen by *degeneration* of contiguous subsequences of motion segments contained in the waypoint orientation grid as shown in Figs. 5, 6 and 7.

Assuming an ability to degenerate sequences of motion segments (the technicalities of which are described in Section 5.4) we propose a heuristic approach to generation of a waypoint orientation grid. It shall be generated on the basis of prescribed  $N_L \geq 0$  periods of a triangle wave with amplitude  $L_\theta \in (0, \pi)$ , and offset  $\theta_{d0}$  (cf. Figs. 5 and 6). Moreover, it shall be quantized with some prescribed resolution  $\Delta\theta \in (0, \pi/2)$ . It is assumed that resolution  $\Delta\theta$  is chosen to satisfy  $(\theta_{dN} - \theta_{d0}) \bmod \Delta\theta = 0$  and  $L_\theta \bmod \Delta\theta = 0$  so that the zeroth motion segment starts precisely at orientation  $\theta_{d0}$  and the last motion segment ends precisely at orientation  $\theta_{dN}$ . A waypoint orientation grid resulting from such a triangle wave is shown in Fig. 5.



**Fig. 7** The relation between boundary integral curves (i.e., the ones evaluated for  $p_{bi}$ ) and a line of waypoint positions corresponding to a constant waypoint orientation. If the  $i$ -th waypoint is placed on this line and  $|x_{di}^{i+1}| \geq x_{bi}$ , curvature constraint (3) is satisfied under nominal conditions

It can be seen that in the actual waypoint orientation grid from Fig. 5 every second waypoint orientation is equal to the previous one. Such a redundant choice of waypoint orientations in the grid allows addition of straight-line motion segments (corresponding to the same orientation angles at the ends of the segment) to the generated motion plan, which is of vital practical importance as explained in Remark 4. As a result of the above considerations, we propose the following waypoint orientation grid generation algorithm:

- SO1. Choose the design parameters: resolution  $\Delta\theta \in (0, \frac{\pi}{2})$ , the number of periods  $N_L \geq 0$  and amplitude  $L_\theta \in (0, \pi)$ .
- SO2. Compute the number  $N$  of motion segments as follows:

$$N \triangleq N_p + 2 \frac{|\theta_{dN} - \theta_{d0}|}{\Delta\theta} + 1, \quad N_p \triangleq 8n N_L, \quad n \triangleq \frac{L_\theta}{\Delta\theta}. \tag{30}$$

- SO3. Compute the waypoint orientations in the grid:

$$\theta_{di+1} \triangleq \theta_{di} + \text{sign} \left( 1 - 2 \left( \frac{i}{2} - \lfloor \frac{i}{2} \rfloor \right) \right) \delta_i \Delta\theta, \tag{31}$$

with

$$\delta_i \triangleq \begin{cases} +1 & \text{for } (\lfloor \frac{i}{2} \rfloor \bmod 4n) \leq n & \wedge & i \leq N_p, \\ +1 & \text{for } (\lfloor \frac{i}{2} \rfloor \bmod 4n) > 3n & \wedge & i \leq N_p, \\ -1 & \text{for } (\lfloor \frac{i}{2} \rfloor \bmod 4n) \in (n, 3n) & \wedge & i \leq N_p, \\ +1 & \text{for } \text{sign}(\theta_{dN} - \theta_{d0}) \geq 0 & \wedge & i > N_p, \\ -1 & \text{for } \text{sign}(\theta_{dN} - \theta_{d0}) < 0 & \wedge & i > N_p, \end{cases}$$

where the orientations  $\theta_{d0}, \theta_{dN}$  are prescribed by the user, while  $\lfloor \cdot \rfloor$  denotes the integral part of a value (i.e. the value is rounded down to the nearest integer).

The choice of parameters  $\Delta\theta, N_L$  and  $L_\theta$  is a trade off between the number of decision variables in the resulting optimization problem and the resolution of waypoint orientation grid explored by the algorithm. This choice can also be determined by specific applications. For example, one can choose large  $N_L$  (at the expense of a large number of variables in MILP) to allow for potential generation of plans with a large number of robot turns understood as changes in sign of motion curvature. On the other hand, domain-specific knowledge arising from a particular motion scenario (e.g., local planning) could suggest decreasing the values of  $N_L$  and  $L_\theta$  or even shaping the waypoint

orientation grid differently (see Remark 3). In general, thanks to the properties of the VFO control law, acceptable motion plans are often computed with a very small number of waypoints corresponding to high values of  $\Delta\theta$ . That is why we assume in the sequel a high initial value of  $\Delta\theta$  and iteratively decrease its value, which corresponds to solving gradually harder optimization problems.

*Remark 3* The strategy of iterative computation of waypoint orientation sequences is just a proposition. Based on particular application, one could devise an entirely different strategy. For example, the waypoint orientation sequence could be randomized or chosen according to another heuristic. Any strategy ensuring that  $\Delta\theta$  decreases over planning time horizon seems to be viable in practice.

*Remark 4* Addition of potential straight-line motion segments to the motion plan is a result of practical considerations. Simulation results have shown that in large-scale environments or in the presence of tight state and curvature constraints, one must wait till the resolution  $\Delta\theta$  is close to 0 to obtain a feasible solution due to the lack of straight-line motion planning ability for larger resolution  $\Delta\theta$ . Addition of straight-line segments alleviates this issue by giving the optimizer additional degrees of freedom at the cost of slightly increased problem size (i.e., a number of decision variables).

*Remark 5* For simplicity of exposition, the proposed waypoint orientation grid definition assumes fixed constant value of  $\mu_i$ . One can consider a finite set of  $\mu_i$  by adjusting the waypoint orientation grid definition to contain multiple motion segments differing only by their respective  $\mu_i$  values.

### 5.3 General Formulation of the VFO-Driven Motion Planning Strategy

In reference to the considerations from the previous subsection, we propose the iterative motion planning procedure described by Algorithm 1.

The iterative approach from Algorithm 1 allows for relatively fast computation of feasible motion plans optimal for resolution  $\Delta\theta$  with relatively low computational cost, due to the fact that first iterations consider motion plans with a small number of waypoints.

---

#### Algorithm 1 VFO-driven motion planning procedure

---

```

forall  $\mu_i \leftarrow \tilde{\mu}$ , where  $\tilde{\mu} \in (0.5, 1)$ 
 $\tilde{n} \leftarrow 4, \tilde{c} \leftarrow \infty$ 
while planning time not exceeded do
     $\Delta\theta \leftarrow \pi/\tilde{n}$ 
     $L_\theta \leftarrow \pi/2$ 
    round  $\Delta\theta$  to satisfy  $(\theta_{dN} - \theta_{d0}) \bmod \Delta\theta = 0$ 
    round  $L_\theta$  to satisfy  $L_\theta \bmod \Delta\theta = 0$ 
     $N_L \leftarrow 2(M + 1)$  ( $M$  is known from assumption A3)
    compute  $N$  and generate the orientation grid (steps SO1-SO3)
    [preprocess the orientation grid using Algorithm 2] (optionally)
    generate and solve the VFO-driven MILP
    if computed solution cost  $\geq \tilde{c}$  then
        return VFO-driven MILP solution corresponding to  $\tilde{c}$ 
     $\tilde{c} \leftarrow$  computed solution cost
     $\tilde{n} \leftarrow \tilde{n} + 2$ 

```

---

Algorithm 2 is optionally utilized to reduce the size of orientation grid and improve computational efficiency of the planning algorithm by generation of smaller MILP problems. The algorithm prunes elements of the orientation grid, which are guaranteed to be unreachable. Pruning is carried out in a breadth-first manner starting from the end of the orientation

---

#### Algorithm 2 Orientation grid preprocessing procedure

---

```

mark all elements of the orientation grid as pruned
 $\Theta \leftarrow \{\theta_{dN}\}$ 
while  $\Theta \neq \emptyset$  do
     $\lambda \leftarrow$  pop the first element from  $\Theta$ 
     $\mathcal{P}_j \leftarrow$  polygon assigned to  $\lambda$ 
    for all  $\theta_{di}$  such that  $|\lambda - \theta_{di}| \leq \Delta\theta$  do
        if polygon assigned to  $\theta_{di}$  is  $\mathcal{P}_j$  then
             $\theta_{di-1}^i \leftarrow \lambda - \theta_{di}$ 
            compute  $\bar{q}_{di-1}^i$  such that  $|p_{i-1}| = |p_{bi-1}|$ 
            if curve (10) between  $q_{di-1}$  and  $q_{di}$  contained in  $\mathcal{P}_j$  then
                mark  $\theta_{di}$  as not pruned and add  $\theta_{di}$  to  $\Theta$ 
remove all elements marked as pruned from the orientation grid

```

---



grid. Pruning procedure utilizes VFO properties given in the Section 4 and state-constraints checking method described in Section 5.5. The information on how one determines which polygon  $\mathcal{P}_j$  is assigned to a given orientation grid element is provided in Section 5.6. Since a set of possible waypoint orientations is now determined by a waypoint orientation grid, we proceed to waypoint position planning.

We propose to solve the waypoint position planning problem by solving the appropriately formulated MILP. For the sake of clarity, first the general structure of the considered MILP will be given, while further details along with definitions of particular matrices and vectors used in our formulation will be explained gradually in the sequel. The vector of decision variables  $\xi$  optimized in the MILP is defined as follows:

$$\xi \triangleq [x^T \ y^T \ \zeta^T \ l^T]^T \in \mathbb{R}^N \times \mathbb{Z}^{2N} \times \mathbb{R}^N,$$

with

$$x \triangleq \begin{bmatrix} x_{dN-1}^N \\ x_{dN-2}^N \\ \vdots \\ x_{d0}^1 \end{bmatrix}, y \triangleq \begin{bmatrix} \gamma_{N-1} \\ \gamma_{N-2} \\ \vdots \\ \gamma_0 \end{bmatrix}, \zeta \triangleq \begin{bmatrix} \zeta_{N-1} \\ \zeta_{N-2} \\ \vdots \\ \zeta_0 \end{bmatrix}, l \triangleq \begin{bmatrix} l_{N-1} \\ l_{N-2} \\ \vdots \\ l_0 \end{bmatrix},$$

where variables  $x_{di}^{i+1}$ ,  $i = \{0, \dots, N - 1\}$  correspond to  $x$ -coordinates of subsequent waypoint positions expressed in local (waypoint) frames, variables  $\gamma_i \in \{0, 1\}$  and  $\zeta_i \in \{0, 1\}$  encode choices between forward motion sense ( $\gamma_i = 1$ ,  $\zeta_i = 0$ ), backward motion sense ( $\gamma_i = 0$ ,  $\zeta_i = 1$ ) and segment degeneration ( $\gamma_i = 0$ ,  $\zeta_i = 0$ ) in consecutive motion segments, while variables  $l_i$  (when minimized) correspond to  $|x_{di}^{i+1}|$  and have been introduced to model an absolute value function in the considered MILP. Using the introduced variables, the MILP solving Problem 1 is defined as follows:

$$\min_{\xi} (1 + w_N N) \tilde{l}^T \xi, \quad \tilde{l} \triangleq [0_{3N}^T \ \tilde{l}_{N-1} \ \tilde{l}_{N-2} \ \dots \ \tilde{l}_0]^T, \tag{32}$$

subject to constraints:

$$\begin{bmatrix} A_g & 0_{2 \times N} & 0_{2 \times N} & 0_{2 \times N} \\ T_{M-1} & 0_{1 \times N} & 0_{1 \times N} & 0_{1 \times N} \\ T_{M-2} & 0_{1 \times N} & 0_{1 \times N} & 0_{1 \times N} \\ \vdots & \vdots & \vdots & \vdots \\ T_0 & 0_{1 \times N} & 0_{1 \times N} & 0_{1 \times N} \end{bmatrix} \xi = \begin{bmatrix} b_g \\ b_{M-1} \\ b_{M-2} \\ \vdots \\ b_0 \end{bmatrix}, \tag{33}$$

$$\begin{bmatrix} 0_{N \times N} & A_{\theta 1} & A_{\theta 1} & 0_{N \times N} \\ 0_{N \times N} & A_{\theta 2} & A_{\theta 2} & 0_{N \times N} \\ F_{N-1} & 0_{2E_{N-1} \times N} & 0_{2E_{N-1} \times N} & 0_{2E_{N-1} \times N} \\ F_{N-2} & 0_{2E_{N-2} \times N} & 0_{2E_{N-2} \times N} & 0_{2E_{N-2} \times N} \\ \vdots & \vdots & \vdots & \vdots \\ F_0 & 0_{2E_0 \times N} & 0_{2E_0 \times N} & 0_{2E_0 \times N} \\ K_1 & K_2 & K_3 & 0_{N \times N} \\ K_4 & K_5 & K_6 & 0_{N \times N} \\ 0_{N \times N} & K_7 & K_8 & 0_{N \times N} \\ A_{l1} & 0_{N \times N} & 0_{N \times N} & A_{l2} \\ A_{l3} & 0_{N \times N} & 0_{N \times N} & A_{l4} \end{bmatrix} \xi \leq \begin{bmatrix} b_{\theta 1} \\ b_{\theta 2} \\ b_{fN-1} \\ b_{fN-2} \\ \vdots \\ b_{f0} \\ b_{k1} \\ b_{k2} \\ b_{k3} \\ b_{l1} \\ b_{l2} \end{bmatrix}, \tag{34}$$

where  $\tilde{l}$  comprises the positive weights of the cost vector. The motion cost (32) is a sum of partial motion costs for all  $N$  segments scaled by a factor  $(1 + w_N N)$  penalizing high number of motion segments with  $w_N \geq 0$  being a design parameter. The partial motion cost for the  $i$ -th segment corresponds to the length of integral curve  $f_i^{i+1}(y^{i+1}, p_i)$ , which is calculated by taking the precomputed length  $\tilde{l}_i$  of integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  for  $|x_{di}^{i+1}| = 1$  with  $\theta_{di}^{i+1}$  known from the orientation grid and scaling it by a factor of  $l_i$ . The matrices  $F_i$  and vectors  $b_{fi}$  contain state constraints corresponding to  $q \in \mathcal{Q}_{free}$ , matrices  $K_1$  through  $K_8$  and vectors  $b_{k1}$  through  $b_{k3}$  impose curvature bounds, while  $A_g$  along with  $b_g$  handle position boundary conditions, and  $A_{\theta 1}$ ,  $A_{\theta 2}$  along with  $b_{\theta 1}$ ,  $b_{\theta 2}$  handle orientation boundary conditions. Matrices  $T_j$  paired with vectors  $b_j$  for  $j \in \{0, \dots, M - 1\}$  ensure that appropriate waypoints are placed on transition edges of consecutive polygons in  $\mathcal{P}_{free}$ . Matrices  $A_{l1}$  through  $A_{l4}$  with vectors  $b_{l1}$   $b_{l2}$  ensure that  $\min l_i = |x_{di}^{i+1}|$ . In the context of matrices and vectors,  $\leq$  and  $\geq$  operators denote element-wise operations. We will now systematically derive the constraints used in our problem formulation and represent them in terms of the above matrices and vectors for the cases of unbounded configuration space, state constraints represented by a single convex polygon and state constraints represented by a sequence of convex polygons.

To complete the general description of considered MILP, we will now discuss assumed cost functional (32) in details. As mentioned earlier, it corresponds to the total length of a composite curve resulting from concatenation of  $N$  integral curves  $f_i^{i+1}(y^{i+1}, p_i)$  scaled by term  $(1 + w_N N)$ , which essentially penalizes motion plans generated by solution of MILPs with lower values of resolution  $\Delta\theta$ . Since  $\theta_{di}^{i+1}$  is

determined by waypoint orientation grid, one pre-computes the base integral curve length  $\tilde{l}_i$  for all  $N$  motion segments numerically in one shot using (10) and assuming  $\left|x_{di}^{i+1}\right| = 1$ . It is easy to verify that the term  $\tilde{l}_i \left|x_{di}^{i+1}\right|$  represents the true length of integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  length, because of linear scaling of integral curve length w.r.t.  $\left|x_{di}^{i+1}\right|$ , which can be established by virtue of Property 1 and Corollary 1.

One cannot straightforwardly represent an absolute value of decision variable such as  $x_{di}^{i+1}$  in a MILP cost functional such as (32), because MILP requires that the cost functional is a linear combination of optimized decision variables. For the purpose of expressing  $\left|x_{di}^{i+1}\right|$  in the considered MILP, we apply the so-called minimax reformulation and introduce additional variables  $l_i$  and formulate constraints ensuring that  $\min l_i = \left|x_{di}^{i+1}\right|$ . Upon the solution of considered MILP, motion cost (32) will be minimized, that is variables  $l_i$  will attain their minimum values allowed by constraints and one will have  $l_i = \left|x_{di}^{i+1}\right|$ . To formulate constraints necessary to ensure that  $\min l_i = \left|x_{di}^{i+1}\right|$ , one must consider the fact that  $l_i = \left|x_{di}^{i+1}\right| = \max \left\{-x_{di}^{i+1}, x_{di}^{i+1}\right\}$ , which implies that constraints

$$l_i \geq x_{di}^{i+1} \quad \wedge \quad l_i \geq -x_{di}^{i+1} \tag{35}$$

must be satisfied by  $l_i$ . Since (35) ensure that  $l_i$  is lower-bounded by  $\max \left\{-x_{di}^{i+1}, x_{di}^{i+1}\right\}$ , it will converge to the value of  $\max \left\{-x_{di}^{i+1}, x_{di}^{i+1}\right\}$  upon minimization of motion cost (32) (i.e., upon solution of MILP), thus  $l_i = \left|x_{di}^{i+1}\right|$  will be satisfied. Constraints (35) can be transformed to the following matrices and vectors utilized in (34)

$$\begin{aligned} A_{I1} &\triangleq I_N, & A_{I2} &\triangleq -I_N, & A_{I3} &\triangleq -I_N, & A_{I4} &\triangleq -I_N, \\ b_{I1} &\triangleq \mathbf{0}_N, & b_{I2} &\triangleq \mathbf{0}_N. \end{aligned}$$

The cost functional (32) is designed to ensure that setting  $w_N = 0$  will lead to lower total sums of integral curves lengths, that is, shorter motion plans at the expense of higher number of waypoints. High number of waypoints should be avoided if possible due to

growth of plan representation and resultant high number of switches between the waypoints during plan execution. Note that the value of  $(1 + w_N N)$  comes into play only when the solution of the current MILP instance is compared with the best solution encountered so far in of the proposed iterative motion planning algorithm. For a single MILP instance  $(1 + w_N N)$  is a precomputed constant, therefore it does satisfy linearity requirements imposed on (32) by definition of a MILP.

### 5.4 Planning in Unbounded Configuration Space

Let us consider the simplest planning scenario, where  $Q_{free} = Q$ . In that case one considers state constraint matrices  $F_j$  and transition constraint matrices  $T_j$  to be zero matrices. The definitions of remaining constraints constitute the core of our MILP formulation in this case, and correspond to constraints (27), (28) and (29) from the general nonlinear problem discussed earlier.

We now direct our attention to the position boundary conditions from Problem 1, which are represented by constraint matrix  $A_g$ . We assume that once a motion segment is *degenerated* its length is forced to 0, thus  $x_{di}^{i+1} = 0$  is forced in this case by constraints discussed later. Therefore, when boundary conditions and state constraints are considered, *non-degenerated* and *degenerated* motion segments can be treated in the same manner. Observe that since  $\theta_{di}^{i+1}$  is determined by the waypoint orientation grid specified by (31),  $\mu_i$  is determined by choice in Algorithm 1, and since Property 3 holds, one can describe waypoint position  $\bar{q}_{di}^{i+1}$  only in terms of  $x_{di}^{i+1}$  as  $y_{di}^{i+1} = a_i^{i+1} x_{di}^{i+1}$  with known  $a_i^{i+1}$ . Thus, position constraint ensuring that the zeroth motion segment begins at prescribed position  $\bar{q}_{d0}$ , while the last motion segment ends at  $\bar{q}_{dN}$  (corresponding to original nonlinear constraint (29)) arises naturally and is imposed by matrix-vector pair used in (33):

$$\begin{aligned} A_g &\triangleq \begin{bmatrix} \mathbf{a}_{gx}^\top \\ \mathbf{a}_{gy}^\top \end{bmatrix}, & \mathbf{b}_g &\triangleq \begin{bmatrix} x_{d0} - x_{dN} \\ y_{d0} - y_{dN} \end{bmatrix}, \\ \mathbf{a}_{gx} &\triangleq \begin{bmatrix} \cos \theta_{dN} - a_{N-1}^N \sin \theta_{dN} \\ \cos \theta_{dN-1} - a_{N-2}^{N-1} \sin \theta_{dN-1} \\ \vdots \\ \cos \theta_{d1} - a_0^1 \sin \theta_{d1} \end{bmatrix}, \end{aligned}$$

$$\mathbf{a}_{gy} \triangleq \begin{bmatrix} \sin \theta_{dN} + a_{N-1}^N \cos \theta_{dN} \\ \sin \theta_{dN-1} + a_{N-2}^{N-1} \cos \theta_{dN-1} \\ \vdots \\ \sin \theta_{d1} + a_0^1 \cos \theta_{d1} \end{bmatrix},$$

which results from  $N$ -number of applications of the waypoint coordinate frame transformations as expressed in (29).

Let us move on to curvature constraints formulation corresponding to (27). Again, since  $\theta_{di}^{i+1}$  is determined by the waypoint orientation grid, one has  $y_{di}^{i+1} = a_i^{i+1} x_{di}^{i+1}$  with  $a_i^{i+1}$  known from Property 3. It means that (for fixed  $\theta_{di}^{i+1}$ )  $p_i$  scales linearly w.r.t.  $x_{di}^{i+1}$ , because it is known (see Corollary 1 and the Remark 1) that  $p_i$  scales linearly w.r.t.  $y_{di}^{i+1}$ , which in turn scales linearly w.r.t.  $x_{di}^{i+1}$  as determined by Property 3. Therefore, one can write

$$\begin{aligned} & p_i \left( x_{di}^{i+1}, y_{di}^{i+1} = a_i^{i+1} x_{di}^{i+1} \right) \\ &= x_{di}^{i+1} p_i \left( x_{di}^{i+1} = 1, y_{di}^{i+1} = a_i^{i+1} \right), \end{aligned} \tag{36}$$

where  $p_i(x_{di}^{i+1} = 1, y_{di}^{i+1} = a_i^{i+1})$  denotes a specific value of (11) for particular values of  $x_{di}^{i+1}, y_{di}^{i+1}$ . Substitution of (36) in the constraint (27), which ensures satisfaction of (3) under nominal conditions, yields

$$\left| x_{di}^{i+1} \right| \geq x_{bi}^{i+1} = \frac{p_{bi}}{\left| p_i(x_{di}^{i+1} = 1, y_{di}^{i+1} = a_i^{i+1}) \right|}, \tag{37}$$

as illustrated in Fig. 7. At the same time one must notice that, the constraint imposing  $x_{di}^{i+1} = 0$  for *degenerated* segments is contradictory to  $\left| x_{di}^{i+1} \right| \geq x_{bi}^{i+1}$ , i.e, constraint (37) cannot be satisfied simultaneously with constraint  $\left| x_{di}^{i+1} \right| = 0$ . That is why we consider curvature constraints corresponding to (3) and segment degeneration mechanism in common constraint inequalities as follows:

$$\begin{aligned} x_{di}^{i+1} &\geq \gamma_i x_{bi}^{i+1} - \zeta_i \overline{\left| x_{di}^{i+1} \right|} \wedge \\ x_{di}^{i+1} &\leq -\zeta_i x_{bi}^{i+1} + \gamma_i \overline{\left| x_{di}^{i+1} \right|}, \end{aligned} \tag{38}$$

assuming additional constraint

$$\gamma_i + \zeta_i \leq 1, \tag{39}$$

where  $\overline{\left| x_{di}^{i+1} \right|}$  denotes an admissible upper bound of  $\left| x_{di}^{i+1} \right|$ , taken as an arbitrarily large finite value in this

case. Inequality (39) ensures that one of the following three admissible possibilities can be chosen for the  $i$ -th motion segment:

- $\gamma_i = 0 \wedge \zeta_i = 0$  - segment is *degenerated*
- $\gamma_i = 1 \wedge \zeta_i = 0$  - backward robot motion sense  $\sigma_i = -1$  is forced implying  $x_{di}^{i+1} > 0$ ,
- $\gamma_i = 0 \wedge \zeta_i = 1$  - forward robot motion sense  $\sigma_i = 1$  is forced implying  $x_{di}^{i+1} < 0$ .

Observe that when motion segment is *degenerated*, inequalities (38) correspond to an equality  $x_{di}^{i+1} = 0$ , which forces the length of a degenerated segment to 0. When backward sense of robot motion is chosen, then (38) take the form of  $x_{bi}^{i+1} \leq x_{di}^{i+1} \leq \overline{\left| x_{di}^{i+1} \right|}$ , while the choice of forward motion sense yields  $-x_{bi}^{i+1} \geq x_{di}^{i+1} \geq -\overline{\left| x_{di}^{i+1} \right|}$ . Those two conditions correspond to (37). Introduced constraints can be transformed to a matrix representation used in (34) by taking:

$$\begin{aligned} \mathbf{K}_1 &\triangleq -\mathbf{I}_N, \quad \mathbf{K}_2 \triangleq \text{diag} \left( x_{bN-1}^N, x_{bN-2}^{N-1}, \dots, x_{b0}^1 \right), \\ \mathbf{K}_3 &\triangleq \text{diag} \left( -\overline{\left| x_{dN-1}^N \right|}, -\overline{\left| x_{dN-2}^{N-1} \right|}, \dots, -\overline{\left| x_{d0}^1 \right|} \right), \\ \mathbf{K}_4 &\triangleq \mathbf{I}_N, \quad \mathbf{K}_5 \triangleq \mathbf{K}_3, \quad \mathbf{K}_6 \triangleq \mathbf{K}_2, \\ \mathbf{K}_7 &\triangleq \mathbf{I}_N, \quad \mathbf{K}_8 \triangleq \mathbf{K}_7, \quad \mathbf{b}_{k1} \triangleq \mathbf{0}_N, \quad \mathbf{b}_{k2} \triangleq \mathbf{0}_N, \quad \mathbf{b}_{k3} = \mathbf{1}_N, \end{aligned}$$

where  $\mathbf{1}_N$  denotes a vector of  $N$ -elements, all of which are equal to 1.

Let us turn to the description of orientation boundary conditions. In Section 5.2, it was mentioned that a possibility of degenerating certain motion segments by forcing their length to 0 must be ensured. To utilize Property 3, upon which our planning approach hinges, one must ensure that the waypoint orientation  $\theta_{di}$  of the next non-degenerated waypoint after an  $i$ -th waypoint satisfies

$$\theta_{di} - \theta_{di} = \theta_{di}^{i+1}, \tag{40}$$

as postulated by waypoint orientation grid (31). Relation (40) holds only for the waypoint orientation plans corresponding to particular configurations of the decision variable vectors  $\boldsymbol{\gamma}$  and  $\boldsymbol{\zeta}$ . Thus, feasible waypoint orientation plans satisfying (40) must satisfy the following constraints for all motion segments:

$$\begin{aligned} (\theta_{dN} - \theta_{di}) + (1 - \gamma_i - \zeta_i) L &\geq \sum_{k=i}^{N-1} \gamma_k \theta_k^{k+1} + \sum_{k=i}^{N-1} \zeta_k \theta_k^{k+1}, \\ (\theta_{dN} - \theta_{di}) - (1 - \gamma_i - \zeta_i) L &\leq \sum_{k=i}^{N-1} \gamma_k \theta_k^{k+1} + \sum_{k=i}^{N-1} \zeta_k \theta_k^{k+1}, \end{aligned}$$

which result from consideration of (40) and the fact that the  $i$ -th motion segment is *non-degenerated* iff  $\gamma_i = 1$  or  $\zeta_i = 1$ . Note that  $2L_\theta < L < \infty$  corresponds to an arbitrarily chosen very large value, which influences the above constraints in such a way that they are always satisfied upon degeneration of the segment regardless of other conditions. Introduced constraints guarantee simultaneous satisfaction of (40) and constraint (28) from the general nonlinear optimization problem. One can now straightforwardly define the matrices and vectors used in (34) corresponding to those constraints:

$$\begin{aligned}
 \mathbf{A}_{\theta 1} &\triangleq \begin{bmatrix} \theta_{dN-1}^N + L & 0 & \dots & 0 \\ \theta_{dN-1}^N & \theta_{dN-2}^{N-1} + L & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{dN-1}^N & \theta_{dN-2}^{N-1} & \dots & \theta_{d0}^1 + L \end{bmatrix}, \\
 \mathbf{b}_{\theta 1} &\triangleq \begin{bmatrix} \theta_{dN-1} - \theta_{dN} + L \\ \theta_{dN-2} - \theta_{dN} + L \\ \vdots \\ \theta_{d0} - \theta_{dN} + L \end{bmatrix}, \\
 \mathbf{A}_{\theta 2} &\triangleq \begin{bmatrix} -\theta_{dN-1}^N - L & 0 & \dots & 0 \\ -\theta_{dN-1}^N & -\theta_{dN-2}^{N-1} - L & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\theta_{dN-1}^N & -\theta_{dN-2}^{N-1} & \dots & -\theta_{d0}^1 - L \end{bmatrix}, \\
 \mathbf{b}_{\theta 2} &\triangleq \begin{bmatrix} -\theta_{dN-1} + \theta_{dN} + L \\ -\theta_{dN-2} + \theta_{dN} + L \\ \vdots \\ -\theta_{d0} + \theta_{dN} + L \end{bmatrix}.
 \end{aligned}$$

### 5.5 Planning in a Single Convex Polygon

In this subsection the introduced optimization framework is applied to case when  $\mathcal{Q} \neq \mathcal{Q}_{free}$ . Let us assume that  $\mathcal{Q}_{free} = \mathcal{P}_{free} \times \mathbb{R}$  is described according to assumption A4 with  $M = 0$  and formulate constraints guaranteeing that robot position will remain in a single convex polygon  $\mathcal{P}_{f0}$ . For convenience, we exploit assumption A6 to represent lines, which contain all  $E_0$  edges of polygon  $\mathcal{P}_{f0}$  as follows:

$$Y = c_p X + d_p \quad \forall p \in \{0, 1, \dots, E_0 - 1\}.$$

Using such a representation one can write robot position constraints:

$$\beta_{pi} y \leq \beta_{pi} (c_p x + d_p) \quad \forall p \in \{0, 1, \dots, E_0 - 1\}, \tag{41}$$

where  $\beta_{pi} \in \{-1, 1\}$  denotes the sign which can be found by checking which value of sign  $\beta_{pi}$  satisfies inequalities (41) for any chosen point lying inside the polygon  $\mathcal{P}_{f0}$ . To clarify, the sign  $\beta_{pi}$  determines whether all the points lying inside the polygon  $\mathcal{P}_{f0}$  must lie above or below the line  $Y = c_p X + d_p$ . Note that constraints (41) correspond to the constraint (25) and must be satisfied  $\forall t \in [t_i, t_{i+1}]$ .

Checking (41)  $\forall t \in [t_i, t_{i+1}]$  is computationally expensive and intractable to do exactly, therefore we utilize the VFO properties and notice that (under nominal conditions) it suffices to check only 3 points on an integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  to determine the satisfaction of constraints (41) in the  $i$ -th motion segment, namely  $\bar{q}_{di+1}$  (i.e., the end of the  $i$ -th motion segment),  $\bar{q}_{di}$  (i.e., the beginning of the  $i$ -th motion segment), and point  $\bar{s}_{pi}^{i+1} \triangleq [s_{xpi}^{i+1} \ s_{ypi}^{i+1}]^T$ . Point  $\bar{s}_{pi}^{i+1}$  lies on the integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  and is such that the tangent to this curve at point  $s_{ypi}$  is parallel to the considered  $p$ -th polygon edge. It is known that for a given curve  $f_i^{i+1}(y^{i+1}, p_i)$  and  $p$ -th polygon edge, there exists at most one such point, because the auxiliary orientation  $\theta_{ai}^{i+1}$ , which is tangent to integral curve  $f_i^{i+1}(y^{i+1}, p_i)$ , varies monotonically w.r.t.  $y^{i+1}$ . It means that, if the curve  $f_i^{i+1}(y^{i+1}, p_i)$  satisfies constraint (41) for the  $p$ -th polygon edge, then the point  $\bar{s}_{pi}^{i+1}$  is the one lying the closest or the farthest from the line  $Y = c_p X + d_p$ , because for all other points on integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  the vector tangent to this curve has a non-zero projection on a vector orthogonal to the  $p$ -th polygon edge. Therefore, if the mentioned 3 points  $\bar{q}_{di+1}$ ,  $\bar{q}_{di}$  and  $\bar{s}_{pi}^{i+1}$  satisfy constraint (41) for a  $p$ -th polygon edge, then the whole curve  $f_i^{i+1}(y^{i+1}, p_i)$  satisfies this constraint. Point  $\bar{s}_{pi}^{i+1}$  is computed by substitution of the considered  $p$ -th polygon edge slope angle into (21), which yields a directional coefficient  $a_{spi}^{i+1}$  of a particular line  $s_{ypi}^{i+1} = a_{spi}^{i+1} s_{xpi}^{i+1}$ , the intersection of which with curve  $f_i^{i+1}(y^{i+1}, p_i)$  given by (10) corresponds to  $\bar{s}_{pi}$ . Note that in the case of our method, we are only concerned with satisfaction of constraints (41) for a segment of integral curve  $f_i^{i+1}(y^{i+1}, p_i)$  contained

between the end of the  $i$ -th motion segment  $\bar{q}_{di+1}$  and the beginning of the  $i$ -th motion segment  $\bar{q}_{di}$ . Thus, if  $|s_{y_{pi}}^{i+1}| \geq |y_{di}^{i+1}|$ , then we fix  $\bar{s}_{pi}^{i+1}$  to the value of  $\bar{q}_{di}^{i+1}$ . This leads to the following relation describing coordinates of the point  $\bar{s}_{pi}^{i+1}$ :

$$|s_{y_{pi}}^{i+1}| = \min \left\{ |y_{di}^{i+1}|, \left| p_i \left( \frac{\sigma_i \frac{p_i}{|p_i|}}{a_{s_{pi}}^{i+1}} + \sqrt{\frac{1}{(a_{s_{pi}}^{i+1})^2} + 1} \right)^{\frac{1}{\mu_i}} \right| \right\}, \quad (42)$$

with  $\text{sign}(s_{y_{pi}}^{i+1}) = \text{sign}(y_{di}^{i+1})$  and  $s_{x_{pi}}^{i+1} = f_i^{i+1}(s_{y_{pi}}^{i+1}, p_i)$ , where  $f_i^{i+1}$  is given by (10). The geometric interpretation of the point  $\bar{s}_{pi}^{i+1}$  is shown in Fig. 8.

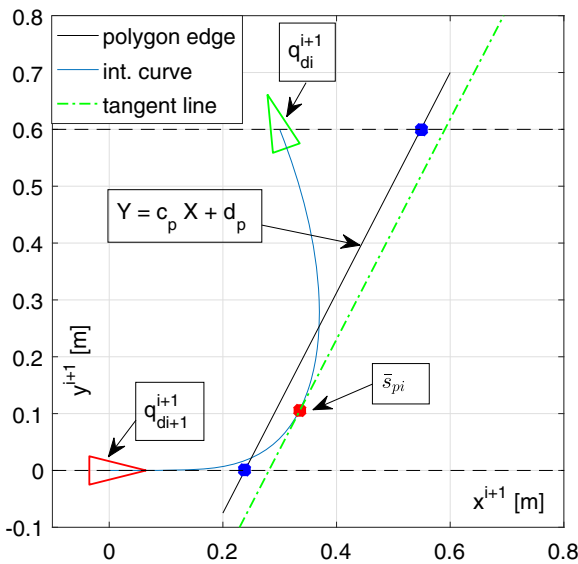
One concludes from the above considerations, that to satisfy constraints (41) for integral curves  $f_i^{i+1}(y^{i+1}, p_i)$  of all  $N$  motion segments it is sufficient to ensure that waypoint positions  $\bar{q}_{di}, \forall i \in \{0, \dots, N\}$  and critical points  $\bar{s}_{pi}, \forall i \in \{0, \dots, N\}$  satisfy (41). Those requirements correspond to constraints

$$\beta_{pi} y_{di} \leq \beta_{pi} (c_p x_{di} + d_p) \quad \forall p \in \{0, 1, \dots, E_0 - 1\}, \quad (43)$$

and

$$\beta_{pi} s_{y_{pi}} \leq \beta_{pi} (c_p s_{x_{pi}} + d_p) \quad \forall p \in \{0, 1, \dots, E_0 - 1\}, \quad (44)$$

respectively. Upon the application of waypoint coordinate frame transformations in a manner similar to the derivation of matrix  $A_g$  and vector  $b_g$  representing position boundary conditions in (33), one can



**Fig. 8** The critical point  $\bar{s}_{pi}$  for the  $p$ -th polygon edge utilized to efficiently express state constraints during planning

represent constraints (43) and (44) as matrices  $F_i$  and vectors  $b_{fi}$  used in (34). They are given by the following relations:

$$F_i \triangleq \begin{bmatrix} w_{0i}^\top \\ w_{1i}^\top \\ \vdots \\ w_{E_0i}^\top \\ \bar{w}_{0i}^\top \\ \bar{w}_{1i}^\top \\ \vdots \\ \bar{w}_{E_0i}^\top \end{bmatrix}, \quad b_{fi} \triangleq \begin{bmatrix} \beta_{0i} (d_0 + c_0 x_{dN} - y_{dN}) \\ \beta_{1i} (d_1 + c_1 x_{dN} - y_{dN}) \\ \vdots \\ \beta_{E_i i} (d_{E_i} + c_{E_i} x_{dN} - y_{dN}) \\ \beta_{0i} (d_0 + c_0 x_{dN} - y_{dN}) \\ \beta_{1i} (d_1 + c_1 x_{dN} - y_{dN}) \\ \vdots \\ \beta_{E_i i} (d_{E_i} + c_{E_i} x_{dN} - y_{dN}) \end{bmatrix},$$

where

$$w_{pi} \triangleq \begin{bmatrix} \beta_{pi} (y_{fN-1} - c_p x_{fN-1}) \\ \vdots \\ \beta_{pi} (y_{fi} - c_p x_{fi}) \\ \mathbf{0}_i \end{bmatrix},$$

while

$$\bar{w}_{pi} \triangleq \begin{bmatrix} \beta_{pi} (y_{fN-1} - c_p x_{fN-1}) \\ \vdots \\ \beta_{pi} (y_{fi+1} - c_p x_{fi+1}) \\ \beta_{pi} (y_{pi} - c_p x_{pi}) \\ \mathbf{0}_i \end{bmatrix},$$

and

$$x_{fi} = \cos \theta_{di+1} - a_i^{i+1} \sin \theta_{di+1},$$

$$y_{fi} = \sin \theta_{di+1} + a_i^{i+1} \cos \theta_{di+1},$$

$$x_{pi} = \cos \theta_{di+1} - a_{pi}^{i+1} \sin \theta_{di+1},$$

$$y_{pi} = \sin \theta_{di+1} + a_{pi}^{i+1} \cos \theta_{di+1},$$

with  $a_{pi}^{i+1} = s_{y_{pi}}^{i+1} / s_{x_{pi}}^{i+1}$ , where  $s_{y_{pi}}^{i+1}$  and  $s_{x_{pi}}^{i+1}$  are computed from (42) using the value of  $p_i$  corresponding to  $x_{di}^{i+1} = 1$  and  $y_{di}^{i+1} = a_i^{i+1}$  (cf. Property 3).

### 5.6 Planning in Convex Decompositions

The framework developed in Sections 5.2-5.5 scales to the case of convex decomposition under assumption A7 (sequential polygon visting), because it guarantees an existence of a simple mapping between polygons and motion segments. Since in Section 5.2 we assumed  $N_L = 2(M + 1)$  (cf. the iterative motion planning algorithm), we propose distributing the  $N_L$  periods of the waypoint orientation grid evenly between the  $M + 1$  convex polygons comprising  $\mathcal{P}_{free}$ . To this aim, we map waypoints

$jN_p/(M+1), jN_p/(M+1)+1, \dots, (j+1)N_p/(M+1)-1$  to polygon  $\mathcal{P}_j$  (cf. definition of  $N$  and (31)). The remaining last  $N - N_p$  waypoints shall be mapped to the last polygon  $\mathcal{P}_{fM}$ . Given this mapping, constraint matrices  $F_i$  and vectors  $b_{fi}$  are computed exactly as in Section 5.5, with constraints (41) computed for a particular polygon from the set  $\mathcal{P}_{free}$  given by the introduced mapping.

It remains to ensure that the last waypoint mapped to polygon  $\mathcal{P}_j$  is lying on transition edge  $\mathcal{E}_{j+1}$  (cf. assumption A5). This is vital for satisfaction of state constraints, because the method introduced in Section 5.5 does not consider a case of a motion segment spanning two or more convex polygons. Let us denote transition edge index in a given  $j$ -th polygon by  $k > 0$  and the index of the last motion segment mapped to this polygon by  $m$ . Then, one can write appropriate equality constraint

$$y_{dm} = (c_k x_{dm} + d_k), \tag{45}$$

which ensures that the  $m$ -th waypoint position lies on an appropriate transition edge. Note that the equality constraint (45) is similar to inequality constraint (43). This similarity is reflected in the matrices  $T_j$  and vector  $b_j$  used in used in (33). Application of waypoint coordinate frame transformations to (45) yields:

$$T_j \triangleq \begin{bmatrix} y_{fN-1} - c_k x_{fN-1} \\ \vdots \\ y_{fm} - c_k x_{fm} \\ \mathbf{0}_m \end{bmatrix}^\top, \quad b_j \triangleq d_k + c_k x_{dN} - y_{dN}.$$

The above constraints ensure that the considered waypoint position satisfies the equation of a line containing a transition edge. Those constraints in conjunction with (41), which are also defined for every motion segment, ensure proper placement of the considered waypoint on the transition edge. In practice, one often wants to ensure that the last waypoint mapped to polygon  $\mathcal{P}_j$  is also located sufficiently far from endpoints of the transition edge  $\mathcal{E}_{j+1}$ . This is achieved by appropriate modification of coefficients  $d_p$  in (43) for  $i = m$  and  $p \in \{k - 1, k + 1\}$ . The modified coefficients  $d_p$  shall be chosen to make the resultant new virtual transition edge shorter than the original transition  $\mathcal{E}_{j+1}$  and contained entirely within it. The particular values of modified coefficients  $d_p$  can be obtained using simple algebra.

To conclude presentation of the planning algorithm, let us briefly discuss its completeness. Upon

consideration of assumption A4, one concludes that there exists a composite curve with absolute curvature bounded by  $\kappa_b$ , which connects  $q_{d0}$  and  $q_{dN}$ . On the other hand, recall that angular resolution  $\Delta\theta$  decreases monotonically over iterations of the planning algorithm by definition of Algorithm 1. Monotonic decreasing of  $\Delta\theta$  over planning iterations, implies monotonic decreasing of  $|\theta_{di}^{i+1}|$  and  $x_{di}^{i+1} \rightarrow 0$  due to growth of motion segments number  $N$ . Therefore, over iterations of the planning algorithm the generated plans converge to a sequence of straight-line segments, which can approximate any curve satisfying (3) including the one connecting  $q_{d0}$  with  $q_{dN}$ . We argue that since the set  $\mathcal{P}_{free}$  is compact and (by virtue of assumption A4) neither  $\bar{q}_{d0}$ , nor  $\bar{q}_{dN}$  lie on its boundary, an approximation of considered curve connecting  $q_{d0}$  with  $q_{dN}$  is sufficient. In light of that, after a certain number of iterations,  $\Delta\theta$  will attain a value such that the proposed MILP will have at least one feasible solution, computation of which is guaranteed by the MILP solver given an arbitrarily long solution time.

### 6 Execution of the Planned Motion in the VFO Control System

So far, we have considered only nominal motion conditions. From now on, we will consider the non-nominal motion conditions admitting  $e_{ai}^{i+1} \neq 0$  and  $\epsilon > 0$ . Consideration of  $e_{ai}^{i+1} \neq 0$  is motivated by the fact that, since the auxiliary orientation  $\theta_{ai}$  is defined for all positions  $\bar{q} \neq \bar{q}_{di}$ , any deviation of the initial robot configuration  $q(0)$  from the zeroth waypoint  $q_{d0}$ , (i.e.,  $q(0) \neq q_{d0}$ ) is perceived as  $e_{ai}^{i+1}(0) \neq 0$ . We consider  $\epsilon > 0$  (cf. assumption A2), because  $\epsilon = 0$  requires accurate execution of consecutive waypoint positions (i.e.,  $\bar{q}(t_i) = \bar{q}_{di}$ ), which is not possible in practical motion conditions due to, e.g., imperfect measurements.

Introduction of non-nominal conditions outlined above has various consequences. For example, even if one had  $e_{ai}^{i+1}(0) = 0$ , there would occur  $e_{ai}^{i+1}(t_1) \neq 0$ , because taking  $\epsilon > 0$  in the switching condition from assumption A2 implies  $q(t_i) \neq q_{di}$ . It means that the  $i$ -th motion segment will not be started in the planned robot configuration  $q_{di}$ , but in some other point in

$\epsilon$ -vicinity of  $q_{di}$ . To account for that effect we introduce 3 modifications, which robustify the proposed motion strategy:

1. As discussed before, polygon representations  $\mathcal{P}_j$  are appropriately shrunk to ensure that planned waypoint sequences result in integral curves  $f_i^{i+1}(y^{i+1}, p_i)$  no closer than  $\epsilon$  to  $\mathcal{P}_{free}$  boundaries.
2. Planning is performed with curvature bound  $\tilde{\kappa}_b \triangleq \psi \kappa_b$ ,  $\psi \in (0, 1)$  to introduce a margin of available motion curvature which can be utilized by the VFO control law to reject possible disturbances.
3. Relative directing coefficient  $\mu_i$  is replanned at the time of waypoint switch (i.e., at every time instant  $t_i$ ) to minimize  $e_{ai}^{i+1}(t_i)$ .

The replanned value of  $\mu_i$  mentioned in modification 3 is computed from definition of  $\theta_{di}^{i+1}$  (cf. (7)), by taking  $\mu_i := \hat{\mu}_i$ :

$$\hat{\mu}_i = |\lambda_i|, \quad \hat{\sigma}_i = \text{sign}(\lambda_i), \quad \lambda_i = \frac{y_{di}^{i+1} / \tan(\theta_{di}^{i+1}) - x_{di}^{i+1}}{\|\tilde{q}_{di}^{i+1}\|}.$$

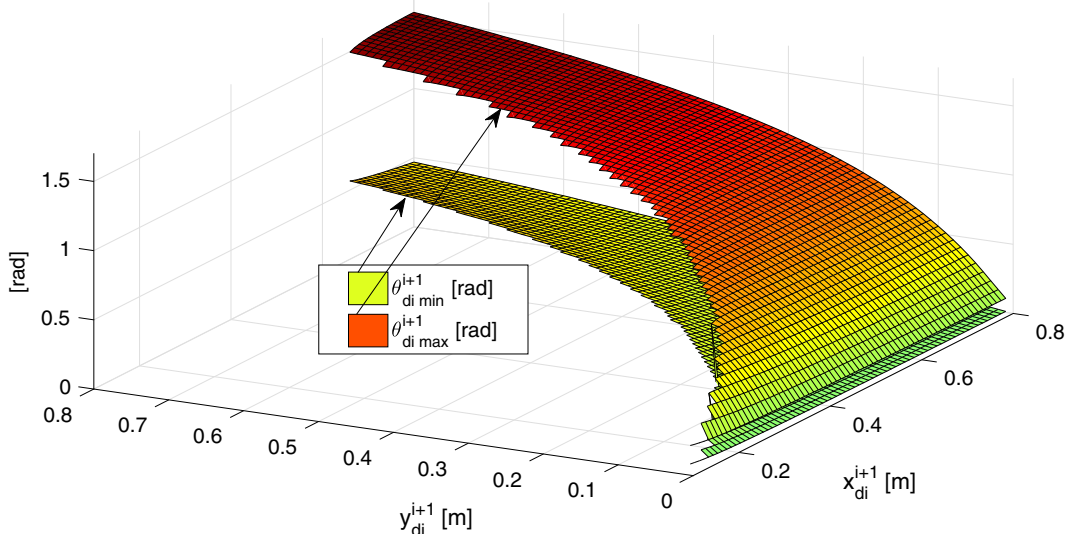
The above value of  $\hat{\mu}_i$  can be used only iff the resulting new  $i$ -th motion segment is contained in  $\mathcal{Q}_{free}$  (checked by verifying satisfaction of constraints represented by matrix  $F_i$  and vector  $b_{fi}$  from (34)) and satisfies curvature constraints (checked using

Property 2). Figure 9 illustrates the range of  $\theta_{di}^{i+1}$  values guaranteeing  $e_{ai}^{i+1}(t_i) = 0$  thanks to replanning of parameter  $\mu_i$ . This range is highly dependent on robot position and grows rapidly with  $|y_{di}^{i+1}|$ .

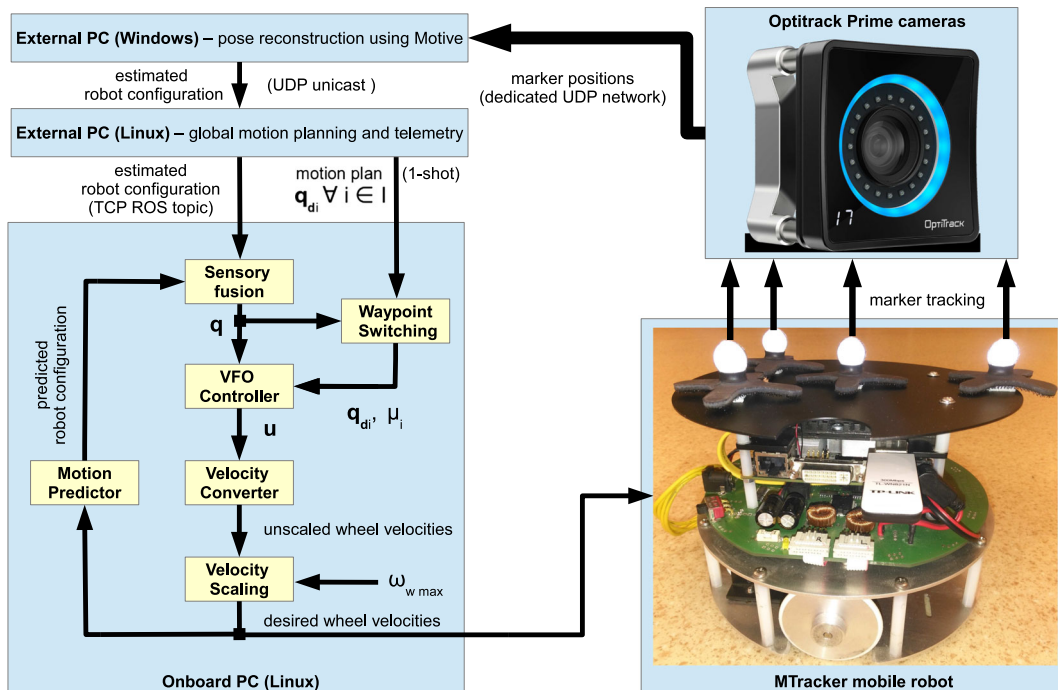
## 7 Numerical Examples and Experimental Results

### 7.1 Experimental Setup

To verify effectiveness of the proposed method, a series of experiments has been performed on the mobile robot MTracker integrated into the experimental system shown in Fig. 10. MTracker is a small-scale laboratory robot equipped with differential drive (i.e., (2,0)-class kinematics). It has base width of approximately 0.2 m. During experiments, the constraint (3) was implemented by software saturation imposed on control signal  $u_1$  dependent on instantaneous  $u_2$  value. Replanning of the parameter  $\mu_i$  during waypoint switching and execution of VFO control law took place on an onboard embedded PC with Intel Atom processor and 2 GB of RAM. Commands in the form of desired wheel velocities were transferred using UART interface to TMS320-F28335 DSP, which was responsible for motor control. Wheel motors were controlled in a velocity-based feedback loop with PI controller endowed with a simple



**Fig. 9** Intervals  $[\theta_{di}^{i+1, \min}, \theta_{di}^{i+1, \max}]$  representing admissible values of  $\theta_{di}^{i+1}$  guaranteeing  $e_{ai}^{i+1} = 0$  upon computation of  $\mu_i$  using modification 3



**Fig. 10** Schematic of experimental setup used to validate proposed motion planning and execution algorithms

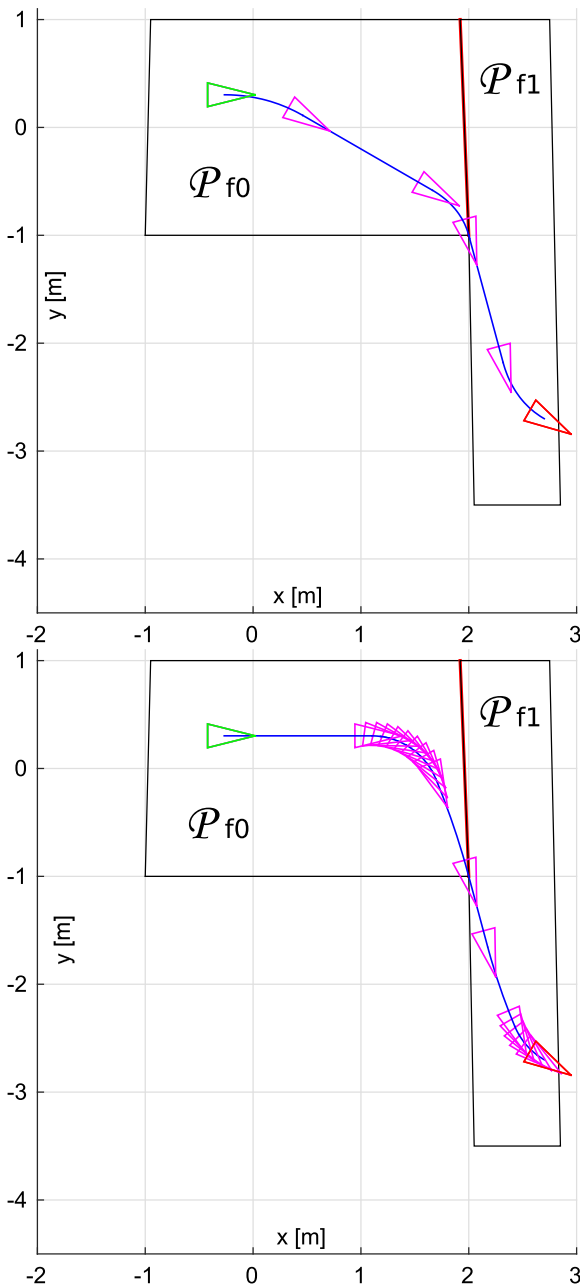
anti-windup block. Motion plans were computed on an external supervising PC, which aside from computing motion plans, was only used to visualize the data captured from MTracker and launch the experiments. The robot was localized using the NaturalPoint Optitrack system, which detected passive markers placed in a characteristic pattern on top of the robot. Robot pose acquired from Optitrack system is combined in a simple sensory fusion algorithm with incremental robot localization computed by motion prediction upon the kinematic model (1). The VFO control loop ran with frequency of 100 Hz, while Optitrack localization data was captured with frequencies ranging from 40 Hz to 120 Hz depending on WLAN quality. Wheel velocities computed by the VFO control law were subsequently scaled in the Velocity Scaling block visible in Fig. 10 to account for actuator limits (see [8] for details of the Velocity Scaling block).

### 7.2 Results

We present exemplary results of simulations and experimental results obtained by application of the proposed *motion strategy* comprising VFO-driven

motion planning and motion execution. Both simulations and experiments were performed with the same parameter values:  $k_a = 4U_2, k_p = 1, U_2 = 0.05 \text{ m/s}, \psi = 0.8, \kappa_b = 2 \text{ m}^{-1}, \epsilon = 0.01 \text{ m}, w_N = 0.5, \mu_i = 0.51$  unless explicitly stated otherwise. In case of scenarios S2 and S3, polygons were shrunk for planning to ensure that integral curves  $f_i^{i+1}(y^{i+1}, p_i)$  lie no closer than  $0.05 \text{ m}$  to boundaries of  $\mathcal{P}_{free}$ . In Figs. 11, 12, 13, 14 and 15, the real boundaries of  $\mathcal{P}_{free}$  are shown as black lines. Transition edges are shown as red lines. Green triangles denote  $q(0)$ , which in case of scenarios S2 and S3 is purposefully different from  $q_{d0}$  (first magenta waypoint) to exemplify robustness of the proposed motion strategy. Red triangles denote  $q_{dN}$ , while magenta triangles denote consecutive waypoints planned by the algorithm. Gray paths correspond to the composite curve computed by concatenation of integral curves (10) for each planned motion segment with consideration of the replanned  $\mu_i$  value, which was actually computed on-line during motion execution as described in modification 3. Blue paths correspond to the actual robot position during simulated or experimental motion execution stage.





**Fig. 11** Results of simulation scenario S1. Motion plan visible on *top* of the figure was generated with  $w_N = 0.5$ , while plan visible on the *bottom* was generated with  $w_N = 0$ . It can be seen that plan generated with  $w_N = 0$  accurately approximates the shortest Dubins-like path at the expense of high number of waypoints

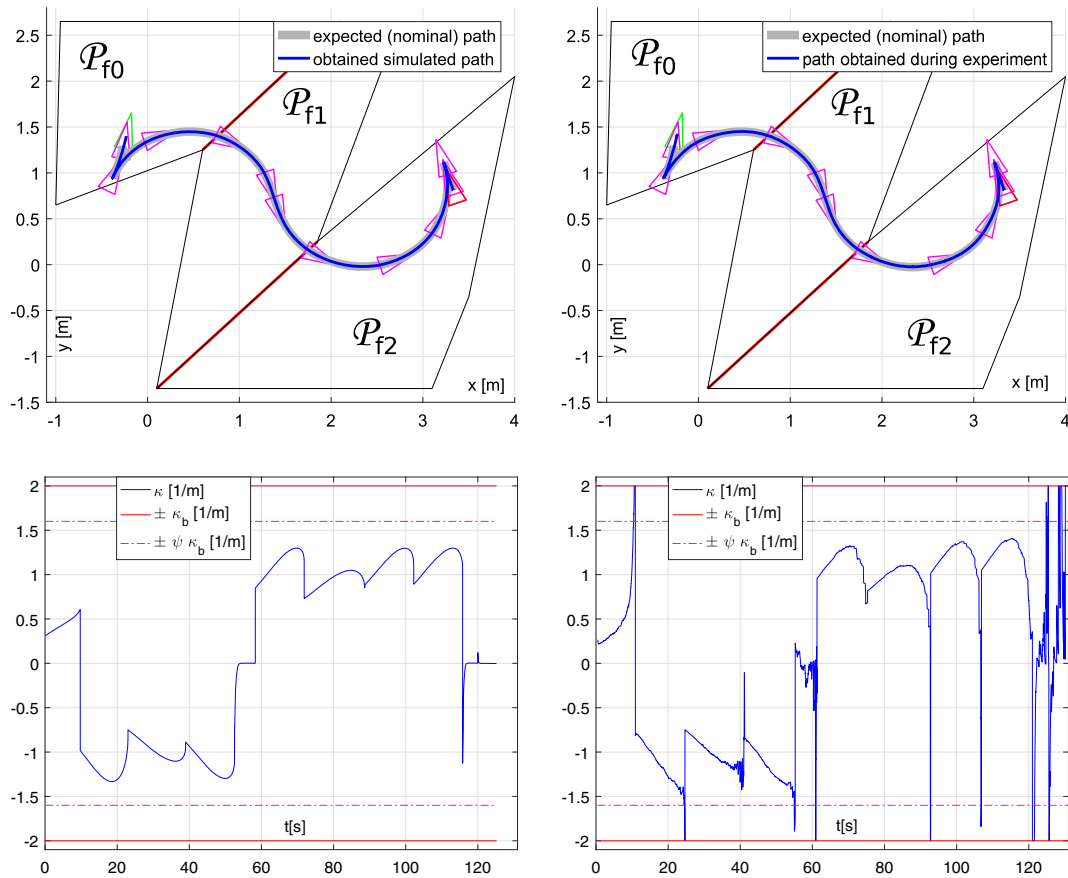
*Remark 6* To account for simple definition of velocity profile  $\rho_i$  (cf. (5)), in our implementation we additionally assume that  $\overline{|x_{dN-1}^N|} = \max \{0.1, x_{bi}^{i+1}\}$  to

keep final motion segments short and avoid long robot stopping time.

Let us begin by discussing simulation scenario S1 presented in Fig. 11. It was designed to demonstrate the influence of cost functional (32) on the generated motion plans. For scenario S1 motion sense was fixed to forward motion ( $\sigma_i = 1, \forall i \in \{0, 1, \dots, N - 1\}$ ) to focus on the effect of parameter  $w_N$  in the generated plans. Results of scenario S1 verify that in case when  $w_N = 0$  the shortest Dubins-like path between initial and final configuration is approximated by the waypoint sequence generated using the proposed algorithm. On the other hand, the plan generated with  $w_N = 0.5$  contains significantly less waypoints (i.e., the planning algorithm terminated with smaller  $N$ ) while maintaining a relatively short path composed from integral curves  $f_i^{i+1}(y^{i+1}, p_i)$ . Similar results were consistently produced across various motion scenarios.

Let us now discuss results of scenario S2 shown in Figs. 12–13 and scenario S3 shown in Figs. 14–15. Planning times for those scenarios are presented in Table 1. In the case of both S2 and S3, the prescribed motion tasks were executed successfully both in simulation and in experiments with a real robot. Execution was successful despite significant displacement between the actual initial configuration  $q(t = 0)$  and the prescribed initial configuration  $q_{d0}$  assumed during motion planning. Actual robot path was very close to the one predicted by using (10), even though motion execution was performed under the non-nominal conditions.

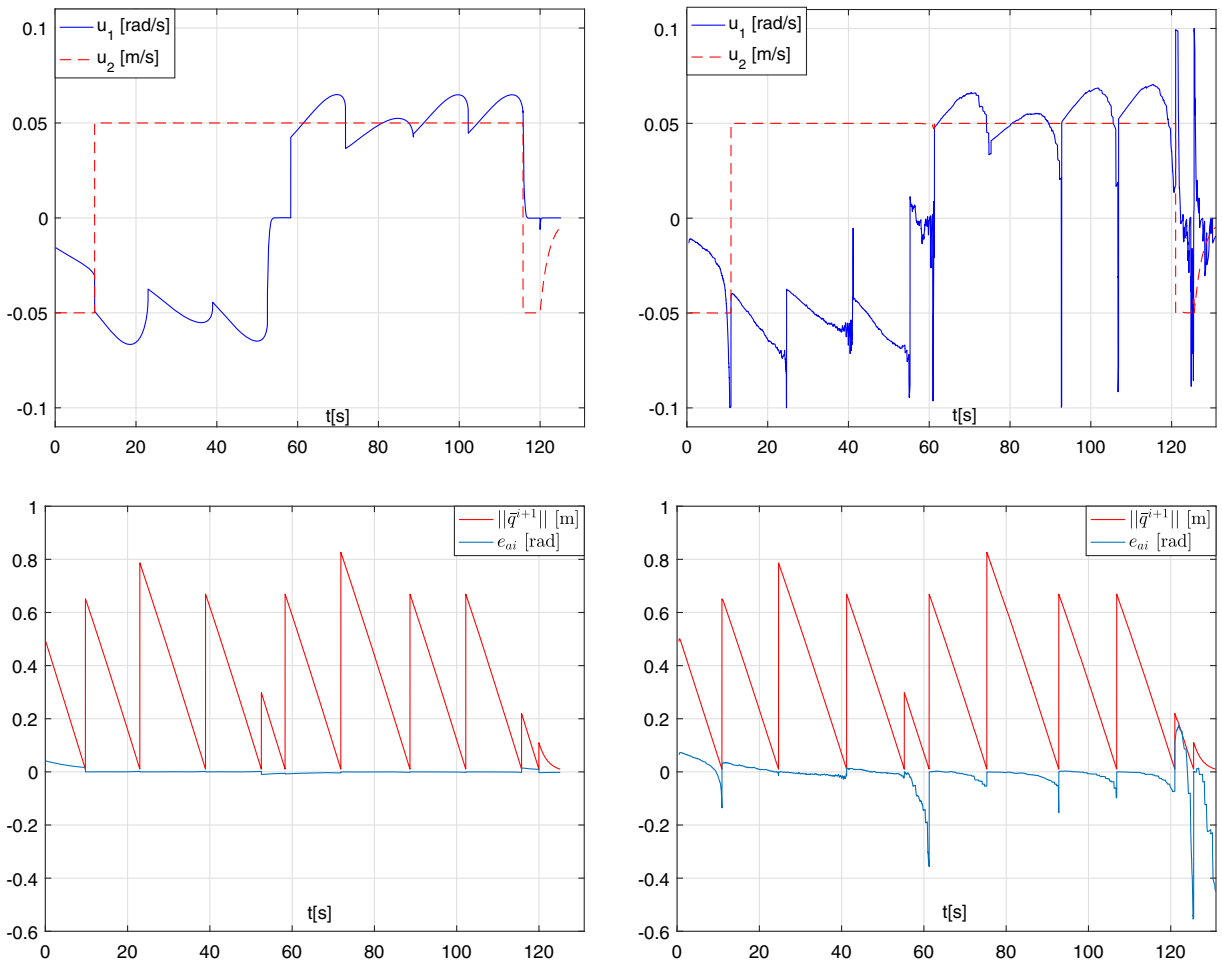
In case of scenario S3 one can see in Fig. 15 that curvature bound  $\psi\kappa_b$  used during the planning stage was significantly exceeded even during simulated execution. It was caused by waypoint switching in the non-nominal motion conditions, which took place at a position  $\bar{q}$  corresponding to the point in which the curve  $f_i^{i+1}(y^{i+1}, p_i)$  has high curvature. This phenomenon was anticipated by introduction of design parameter  $\psi$ , which significantly reduces its influence as can be seen from results of scenario S3. Thanks to curvature margin resulting from  $\psi < 1$ , the actual curvature bound  $\kappa_b$  was not reached during simulations. On the other hand, bound  $\kappa_b$  was reached for multiple short periods of time during experimental motion execution for both scenarios S2 and S3. It is mostly due to the discontinuity in control input



**Fig. 12** Robot path and curvature of motion during simulated (*left-hand side*) and experimental (*right-hand side*) motion execution of scenario S2

$u$  at time instants of waypoint switching, which is present even in nominal conditions, because one has  $\dot{\theta}_{ai}^{i+1} \rightarrow 0$  as  $\bar{q} \rightarrow \bar{q}_{di}$  and at the same time  $\dot{\theta}_{ai+1}^{i+2} \neq 0$ ,  $\forall \bar{q}^{i+2} \neq \bar{0}$  by definition (7). Such a discontinuity introduces a disturbance into the control process, which results in control signal peaks. While resolving this issue will be the topic of our future work, it can be seen from experimental results that it is not critical at least for unicycle-like mobile robot kinematics. The disturbance is visible on time plots of both auxiliary orientation error  $e_{ai}^{i+1}$  and orienting control  $u_1$ , but it has no significant influence on robot path in the positions space. State constraints are still satisfied in the presence of curvature bound  $\kappa_b$  and robot is successfully driven to final configuration  $q_{dN}$ . Moreover, auxiliary orientation error  $e_{ai}^{i+1}$  is in many cases driven back to the close vicinity of zero at the time instant of waypoint switching thanks to modification 3.

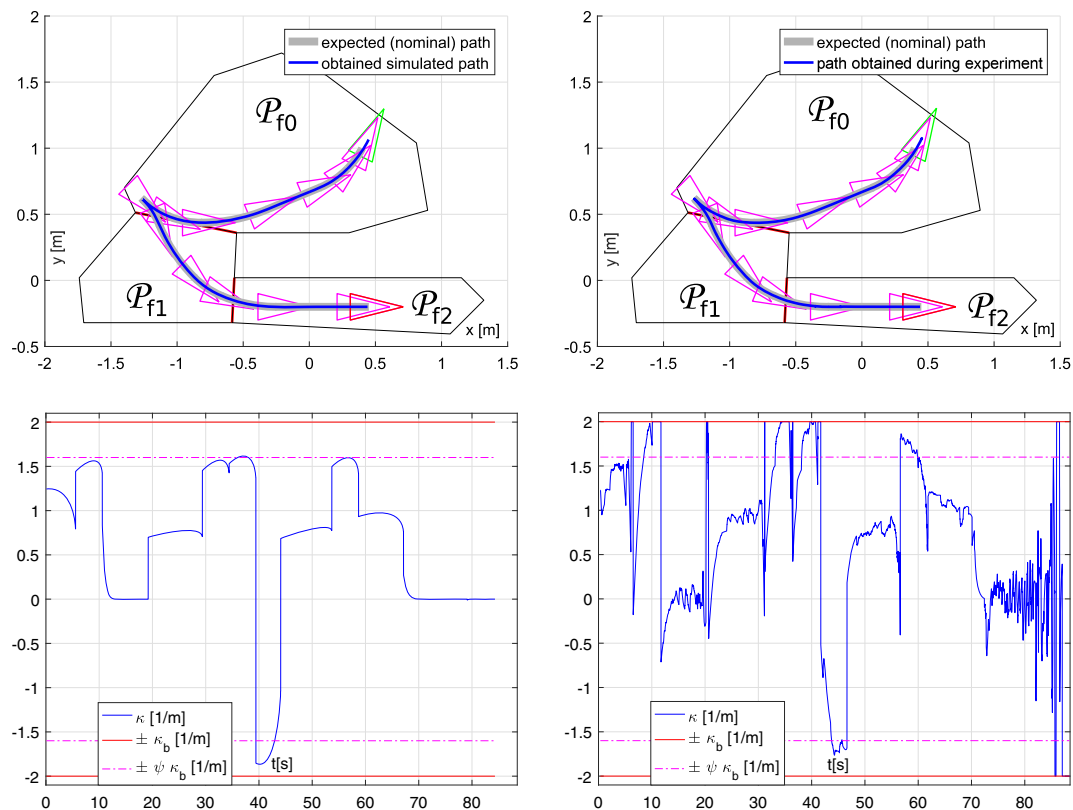
In Figs. 16 and 17 we present the results of waypoint planning for scenarios S4 and S5, respectively. Those scenarios were prepared to show the computational efficiency of the proposed waypoint planning algorithm in more challenging complex motion environments. The environment from scenario S4 contains randomly distributed convex (rectangular) obstacles, while the environment from scenario S5 contains narrow passages and several larger non-convex obstacles, which can pose difficulties for planning algorithms. The sequence  $\mathcal{P}_{free}$  of polygons traversed in scenarios S4 and S5 has been obtained as follows. First, a binary occupancy grid with resolution  $900 \times 250$  has been processed to detect a non-convex polygon with holes representing the free space. Second, this non-convex polygon was decomposed into triangles using constrained Delaunay triangulation. Finally, the triangulation structure was searched using Dijkstra algorithm to



**Fig. 13** Control signals and errors during simulated (*left-hand side*) and experimental (*right-hand side*) execution of scenario S2

find a polygon sequence  $\mathcal{P}_{free}$  connecting initial configuration  $\mathbf{q}_{d0}$  with final reference configuration  $\mathbf{q}_{dN}$ . Determination of the polygon sequence took less than 1 ms on the test PC. The planning times achieved for scenarios S4 and S5 have been gathered in Table 1. The planning times were measured on a Linux PC with a Core i7-2600K processor and GUROBI optimization solver. To assess a computational cost of our method, planning times obtained on average for 10 planning runs of RRT\* and SST (Stable Sparse RRT; an algorithm dedicated for kinodynamic planning, see [30]) algorithms have been also shown in Table 1. RRT\* and SST algorithms were used to plan piecewise-constant open-loop control signals for system (1) under constraint (3) while minimizing the resultant robot path

length. The results for RRT\* and SST algorithms were obtained using publicly available implementations provided by authors of [30]. Final solutions for RRT\* and SST algorithms (cf. Table 1) were chosen as first obtained solutions that corresponded to robot paths of length no bigger than  $1.05l^*$ , where  $l^*$  denotes robot path length resulting from execution of a waypoint sequence obtained at the end of Algorithm 1 under nominal conditions. While in our approach we do not always explicitly look for the shortest robot paths, such a criterion seems to be best suited for comparisons. It can be seen from Figs. 16, 17 and Table 1 that planning succeeds in a reasonable amount of time even for quite complex environments if the pruning procedure presented by Algorithm 2 is utilized. If the



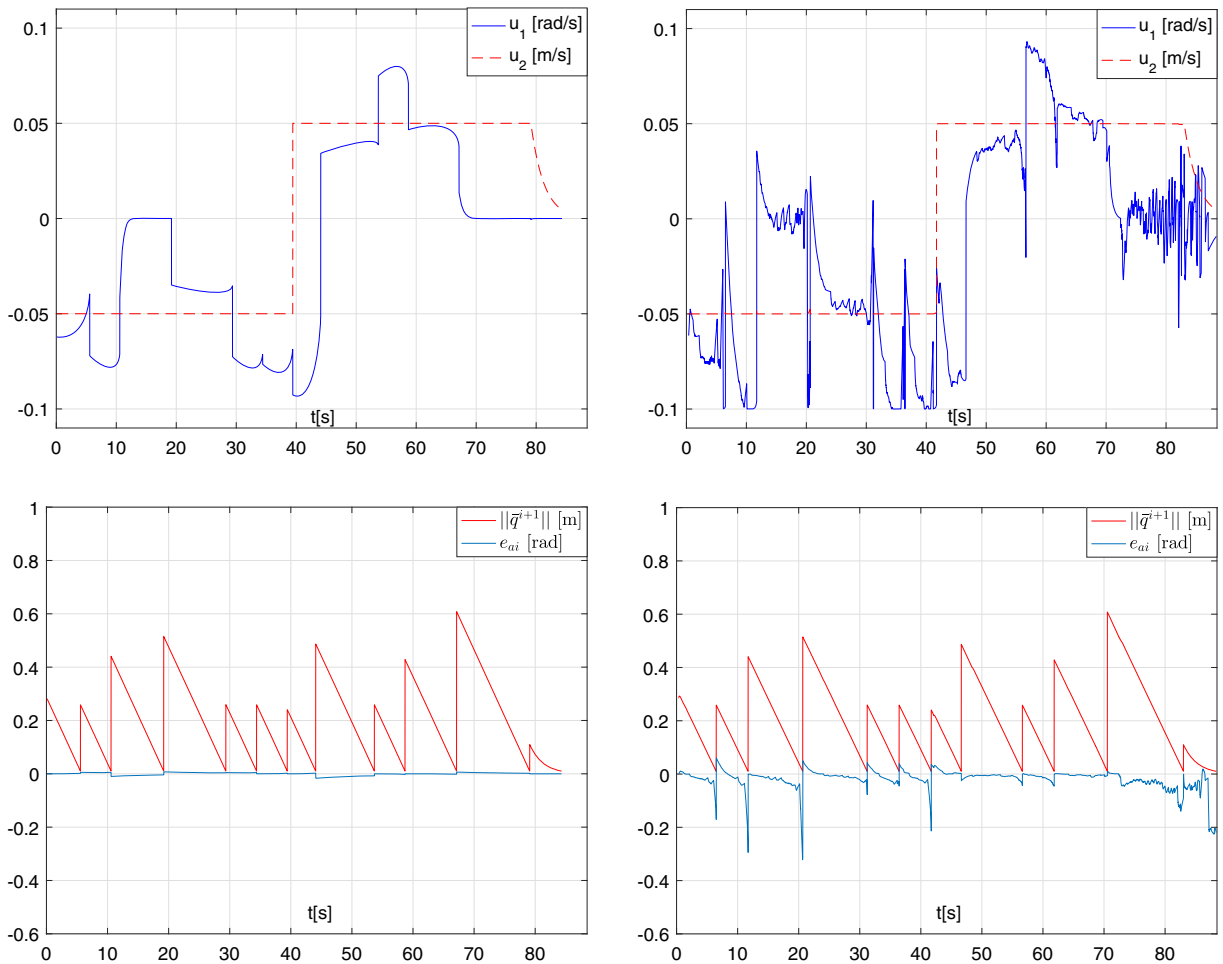
**Fig. 14** Robot path and curvature of motion during simulated (*left-hand side*) and experimental (*right-hand side*) motion execution of scenario S3

pruning procedure is omitted, the large amount of variables and constraints in the MILP problem can significantly increase the computational cost of the planning process. Note that our algorithm reaches initial solutions faster than considered sampling-based algorithms for the case studies represented by scenarios S4 and S5. It seems to be the result of analytic constraints satisfaction checking and exploitation of environment structure provided by a convex decomposition. A similar trend is visible for the optimized solutions. Although the presented results allow concluding that the computational cost of our approach can be acceptable, the differences between computational costs of the respective algorithms can vary depending on many factors such as, e.g., environment topology. One must also consider that sampling-based algorithms are usually more amenable to replanning (as shown in, e.g., [21]) than MILP-based methods such as ours.

## 8 Final Remarks

The VFO-driven motion strategy proposed and experimentally verified in this paper enables application of the VFO methodology in complex motion scenarios, which require imposition of constraints on robot state. The controller-driven motion planner was designed to exploit specific beneficial properties of the VFO control law described in Section 4.2. Such an approach fits into the current trend of complementing analytical methods known from control theory with purely algorithmic methods traditionally utilized in computer science. As predicted in [29], this trend leads to establishment of a new area of the so called *algorithmic control theory*.

Thanks to the controller-driven design methodology, motion plans computed by the proposed planning algorithm can be immediately executed with the VFO control law, and thus the complexity accompanying



**Fig. 15** Control signals and errors during simulated (*left-hand side*) and experimental (*right-hand side*) execution of scenario S3

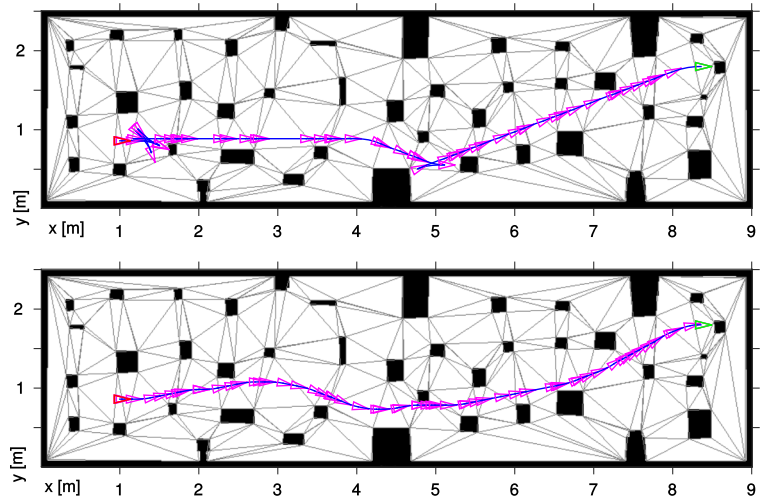
otherwise necessary post-processing procedures (e.g., proper trajectory parametrization and determination of distance to a reference path) was avoided. A plan represented by a waypoint sequence is simple, easy to store and can be easily modified by human users. Moreover, the waypoint positions are planned in a

continuous domain, so that various issues inherent for discrete planning are alleviated. It is also worth noting, that the controller-driven approach allowed for analytically proven, exact and relatively efficient checking of state and curvature constraints crucial in many practical applications.

**Table 1** Planning times for scenarios S2-S5. No planning time limit was assumed. The times obtained when pruning with Algorithm 2 was disabled have been shown in parentheses

Scenario	Time to 1-st sol. [s]	Time to end of Alg. 1 [s]	RRT* 1-st sol. [s]	RRT* final sol. [s]	SST 1-st sol. [s]	SST final sol. [s]
S2	0.05 (0.07)	0.11 (0.11)	n/a	n/a	n/a	n/a
S3	0.05 (0.12)	0.18 (0.26)	n/a	n/a	n/a	n/a
S4	0.23 (64)	1.4 (746)	0.28	9.31	1.61	1.82
S5	0.34 (72)	2.37 (552)	2.04	15.04	3.09	10.21

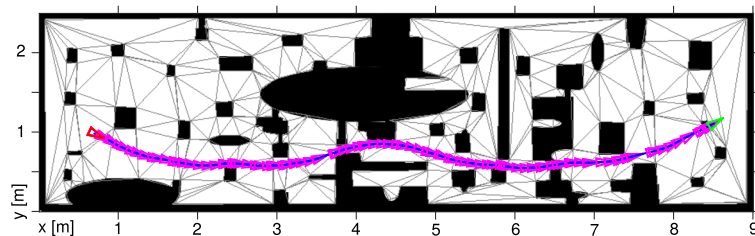
**Fig. 16** Results of simulation scenario S4. Motion plan visible on *top* of the figure was the first feasible solution computed by the planner, while plan visible on the *bottom* is an optimal solution for  $w_N = 0.5$ . Design parameters  $\mu_i = 0.65$  and  $\psi\kappa_b = 2\text{ m}^{-1}$  were chosen in this case. *Green triangle* denotes initial configuration, while *red triangle* denotes prescribed final configuration



It is also worth noting that our approach seems to be quite flexible. The motion sense (i.e., the decision between forward and backward robot motion) for consecutive motion segments is automatically determined by the planner. It can also be chosen by the user for one or more motion segments. The waypoint orientation grid generation algorithm from Section 5.2 is another flexible component of the proposed motion strategy. It can be substituted with a probabilistic algorithm, an application-specific heuristic approach (e.g. for zig-zag parallel parking maneuvers planning) or a higher-level motion planner. Similarly, the MILP-based formulation of the planning problem empowers the designer with an ability to extend our method with additional constraints in an organized manner.

Presented results have opened several research avenues currently investigated by the authors. One of them concerns the most severe limitation of the

VFO motion execution algorithm, which is related to the discontinuity in control input  $\mathbf{u}$  at the time instants of waypoint switching. It seems that this issue could be solved by an appropriate on-line interpolation between convergence vector field  $\tilde{\mathbf{h}}_i$  of subsequent motion segments. Proposed method could be also extended by further exploitation of convenient polygonal environment representation and linear dependency of various useful motion plan characteristics on assumed optimization variables. For example, one could design the motion cost utilized in optimization with consideration of how far away from obstacles should the robot be kept or how fast is curvature of robot motion changing over time. Furthermore, application of the presented approach to more complex robot kinematics such as car-like and N-trailer robots using results from [35] and [33], respectively, is possible.



**Fig. 17** Results of simulation scenario S5 with an optimal solution for  $w_N = 0.5$ . Design parameters  $\mu_i = 0.65$  and  $\psi\kappa_b = 2\text{ m}^{-1}$  were chosen in this case. *Green triangle* denotes

initial configuration, while *red triangle* denotes prescribed final configuration

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Bakolas, E., Tsiotras, P.: On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient. In: In: American Control Conference ACC '09, pp. 385–390, St. Louis, Missouri, USA (2009)
- Bicchi, A., Casalino, G., Santilli, C.: Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. *J. Intell. Robot. Syst.* **16**(4), 387–405 (1996)
- Blackmore, L.: Robust path planning and feedback design under stochastic uncertainty. In: In: Proceedings of the AIAA Guidance, Navigation and Control Conference, Honolulu, Hawaii (2008)
- Chen, C., Rickert, M., Knoll, A.: Combining space exploration and heuristic search in online motion planning for nonholonomic vehicles. In: In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 1307–1312, Gold Coast, Australia (2013)
- Conner, D.C., Choset, H., Rizzi, A.A.: Integrating planning and control for single-bodied wheeled mobile robots. *Auton. Robots* **30**(3), 243–264 (2011)
- Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **79**(3), 497–516 (1957)
- Su Francis, E. et al.: Zero to the zero power. <https://www.math.hmc.edu/funfacts/ffiles/10005.3-5.shtml>. Accessed: 2016-06-20
- Michalek, M., Kozłowski, K.: Vector-Field-Orientation feedback control method for a differentially driven vehicle. *IEEE Trans. Control Syst. Technol.* **18**(1), 45–65 (2010)
- Elbanhawi, M., Simic, M., Jazar, R.: Continuous path smoothing for car-like robots using B-spline curves. *J. Intell. Robot. Syst.*, 1–34 (2015)
- Fraichard, T., Scheuer, A.: From Reeds and Shepp's to continuous-curvature paths. *IEEE Trans. on Robotics* **20**(6), 1025–1035 (2004)
- Gammell, J.D., Srinivasa, S.S., Barfoot, T.D.: Batch informed trees (BIT\*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3067–3074, Seattle, Washington, USA (2015)
- Gawron, T., Michalek, M.M.: VFO stabilization of a unicycle robot with bounded curvature of motion. In: In: International Workshop on Robot Motion and Control, p. 2015, Poznań, Poland (2015)
- Haddad, M., Khalil, W., Lehtihet, H.E.: Trajectory planning of unicycle mobile robots with a trapezoidal-velocity constraint. *IEEE Trans. on Robotics* **26**(5), 954–962 (2010)
- Hale, D.H., Youngblood, G.M., Dixit, P.N.: Automatically-generated convex region decomposition for real-time spatial agent navigation in virtual worlds. In: In: AIIDE, Stanford, California, USA (2008)
- Jaillet, L., Hoffman, J., van den Berg, J., Abbeel, P., Porta, J.M., Goldberg, K.: EG-RRT Environment-guided random trees for kinodynamic motion planning with uncertainty and obstacles. In: In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2646–2652 (2011)
- Janiak, M., Tchoń, K.: Constrained motion planning of nonholonomic systems. *Sys. Control Lett.* **60**(8), 625–631 (2011)
- Janson, L., Schmerling, E., Clark, A., Pavone, M.: Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions. *Int. J. Rob. Res.* **34**(7), 883–921 (2015)
- Hwan Jeon, J., Cowlagi, R.V., Peters, S.C., Karaman, S., Frazzoli, E., Tsiotras, P., Iagnemma, K.: Optimal motion planning with the half-car dynamical model for autonomous high-speed driving. In: In: American Control Conference (ACC), 2013, pp. 188–193, Washington, DC, USA (2013)
- Afonso, R.J.M., Galvao, R.K.H., Kienitz, K.H.: Waypoint trajectory planning in the presence of obstacles with a tunnel-MILP approach, Zürich, Switzerland (2013)
- Karaman, S., Frazzoli, E.: Sampling-based optimal motion planning for non-holonomic dynamical systems, Karlsruhe, Germany (2013)
- Karaman, S., Walter, M.R., Perez, A., Frazzoli, E., Teller, S.: Anytime motion planning using the RRT\*, Shanghai, China (2011)
- Knepper, R.A., Mason, M.T.: Real-time informed path sampling for motion planning search. *Int. J. Rob. Res.* **31**(11), 1231–1250 (2012)
- Kuwata, Y., Fiore, G.A., Teo, J., Frazzoli, E., How, J.P.: Motion planning for urban driving using rrt, Nice, France (2008)
- Lamiroux, F., Bonnafous, D., Lefebvre, O.: Reactive path deformation for nonholonomic mobile robots. *IEEE Trans. on Robotics* **20**(6), 967–977 (2004)
- Landry, B., Deits, R., Florence, P.R., Tedrake, R.: Aggressive quadrotor flight through cluttered environments using mixed integer programming, Stockholm, Sweden (2016)
- Latombe, J.-C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)
- Lau, B., Sprunk, C., Burgard, W.: Kinodynamic motion planning for mobile robots using splines. In: Proceedings

- of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2427–2433, St. Louis, Missouri, USA (2009)
28. LaValle, S.M.: Planning algorithms. Cambridge University Press, Cambridge, U.K. (2006)
  29. LaValle, S.M.: Motion planning: The wild frontiers. *IEEE Robot. Autom. Mag.* **18**(2), 108–118 (2011)
  30. Li, Yanbo., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. *Int. J. Rob. Res.* **35**(5), 528–564 (2016)
  31. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A. (Tony), Thrun, S.: Anytime dynamic A\*: An anytime, replanning algorithm. In: In: Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS) (2005)
  32. Lini, G., Piazzi, A., Consolini, L.: Multi-optimization of  $\eta^3$ -splines for autonomous parking. In: In: Decision and Control and European Control Conference (CDC-ECC) 2011, pp. 6367–6372, Orlando, Florida, USA (2011)
  33. Michałek, M., Kielczewski, M., Jedwabny, T.: Cascaded VFO control for non-standard N-trailer robots. *J. Intell. Rob. Syst.* **77**(3), 415–432 (2015). doi:[10.1007/s10846-013-9881-z](https://doi.org/10.1007/s10846-013-9881-z)
  34. Michałek, M., Kozłowski, K.: Motion planning and feedback control for a unicycle in a way point following task: the VFO approach. *Int. J. Appl. Math. Comput. Sci.* **19**(4), 533–545 (2009)
  35. Michałek, M., Kozłowski, K.: Feedback control framework for car-like robots using the unicycle controllers. *Robotica* **30**, 517–535 (2012)
  36. Morin, P., Samson, C.: Motion Control of Wheeled Mobile Robots, pp. 799–826. Springer, Heidelberg, Berlin (2008)
  37. Narayanan, V., Phillips, M., Likhachev, M.: Anytime safe interval path planning for dynamic environments (2012)
  38. Palmieri, L., Arras, K.O.: A novel RRT extend function for efficient and smooth mobile robot motion planning. In: In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 205–211, Chicago, Illinois, USA (2014)
  39. Park, J.J., Kuipers, B.: Feedback motion planning via non-holonomic RRT for mobile robots. In: In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4035–4040, Hamburg, Germany (2015)
  40. Perez, A., Platt, R., Konidaris, G., et al.: LQR-RRT\*: Optimal sampling-based motion planning with automatically derived extension heuristics. In: In: 2012 IEEE International Conference Robotics and Automation, pp. 2537–2542, Saint Paul, MN (2012)
  41. Piazzi, A., Bianco, C.G.L., Romano, M.:  $\eta^3$  splines for the smooth path generation of wheeled mobile robots. *IEEE Trans. on Robotics* **23**(5), 1089–1095 (2007)
  42. Pivtoraiko, M., Knepper, R.A., Kelly, A.: Differentially constrained mobile robot motion planning in state lattices. *J. Field Rob.* **26**(3), 308–333 (2009)
  43. Reeds, J.A., Shepp, R.A.: Optimal paths for a car that goes both forward and backwards. *Pac. J Mathematics* **145**(2), 367–393 (1990)
  44. Reist, P., Preiswerk, P., Tedrake, R.: Feedback-motion-planning with simulation-based LQR-trees. *Int. J. Rob. Res.* **35**(11), 1393–1416 (2016)
  45. Rickert, M., Brock, O., Knoll, A.: Balancing exploration and exploitation in motion planning. In: In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2812–2817, Pasadena, California, USA (2008)
  46. Schouwenaars, T., How, J., Feron, E.: Receding horizon path planning with implicit safety guarantees. In: In: Proceedings of the American Control Conference, vol. 6, p. 2004, Boston, Massachusetts (2004)
  47. Seidel, R.: A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.* **1**(1), 51–64 (1991)
  48. Shiller, Z., Gwo, Y.R.: Dynamic motion planning of autonomous vehicles. *IEEE Trans. Robot. Autom.* **7**(2), 241–249 (1991)
  49. Shiller, Z., Sharma, S., Stern, I., Stern, A.: Online obstacle avoidance at high speeds. *Int. J. Rob. Res.* **32**(9-10), 1030–1047 (2013)
  50. Tokekar, P., Karnad, N., Isler, V.: Energy-optimal trajectory planning for car-like robots. *Auton. Robots.* **37**(3), 279–300 (2014)
  51. van Duijkeren, N., Nilsson, P., Laine, L., Keviczky, T.: Real-time NMPC for semi-automated highway driving of long heavy vehicle combinations. In: In: Proceedings 5th IFAC Conference on Nonlinear Model Predictive Control 2015, pp. 39–46, Seville, Spain (2015)
  52. Velenis, E., Tsiotras, P.: Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding-horizon implementation. *J. Optim. Theory Appl.* **138**(2), 275–296 (2008)
  53. Vitus, M.P., Pradeep, V., Hoffmann, G.M., Wasl, S.L., Tomlin, C.J.: Tunnel-MILP: Path planning with sequential convex polytopes. In: In: AIAA Guidance, Navigation, and Control Conference (2008)
  54. Yang, K., Moon, S., Yoo, S., Kang, J., Doh, N., Kim, H., Joo, S.: Spline-based RRT path planner for non-holonomic robots. *J. Intell. Robot. Syst.* **73**(1-4), 763–782 (2014)
  55. Zhang, H., Butzke, J., Likhachev, M.: Combining global and local planning with guarantees on completeness (2012)
  56. Zhu, Z., Schmerling, E., Pavone, M.: A convex optimization approach to smooth trajectories for motion planning with car-like robots. In: In: Proceedings of the IEEE Conference on Decision and Control, Osaka, Japan (2015)



**Tomasz Gawron** received M.Sc. degree in automation and robotics from the Poznan University of Technology (PUT), Poznań, Poland in the year 2013. He is currently pursuing his Ph.D. in automation and robotics under the supervision of dr. Maciej Marcin Michałek at the Chair of Control and Systems Engineering, Faculty of Computing, PUT. His current research is focusing on controller-driven feedback motion planning and control of wheeled mobile robots in the presence of state constraints and input constraints.

**Maciej Marcin Michałek** received the M.Sc., Ph.D., and Habilitation degrees in automation and robotics from the Poznan University of Technology (PUT), Poznań, Poland, in the years 2001, 2006, and 2015, respectively. He is currently an Assistant Professor with the Chair of Control and Systems Engineering, Faculty of Computing, PUT. His current research interests include control design problems for nonholonomic systems, in particular, for mobile robots and N-trailer vehicles, and for systems with nonminimum-phase dynamics. Dr. Michałek has been a member of the IEEE (senior member since 2016), the IEEE Robotics and Automation Society, and the IEEE Control Systems Society since 2009. He serves as an Editorial Board Member of the Journal of Intelligent and Robotic Systems.