2

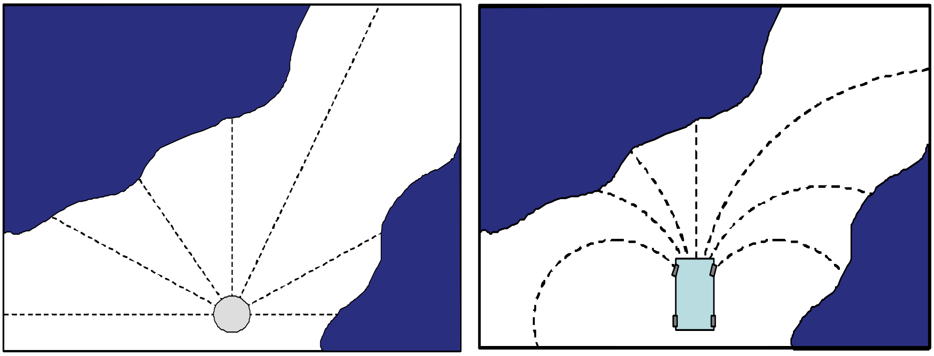
Foundations of Parameterized Trajectories-based Space Transformations for Obstacle Avoidance

J.L. Blanco, J. González and J.A. Fernández-Madrigal
University of Málaga
Spain

1. Introduction

It is essential for any approach to motion planning to account for some spatial representation of obstacles where collision-free paths could be found efficiently. This problem has been extensively studied by the robotics community and has traditionally led to two different research areas. On the one hand we have motion planning approaches, where an optimal path is computed for a known scenario and a target location. The Configuration Space (C-Space) (Lozano-Perez, 1983) has been successfully employed as representation in this scope. In C-Space the robot can be represented as a single point in the high-dimensionality space of its degrees of freedom. On the other hand, some navigation approaches deal with unknown or dynamic scenarios, where motion commands must be periodically computed in real-time during navigation (that is, there is no planning). Under these approaches, called *reactive* or obstacle avoidance, the navigator procedure can be conveniently seen as a mapping between sensor readings and motor action (Arkin, 1998). Although reactive methods are quite efficient and have simple implementations, many of them do not work properly in practical applications since they often rely on too restrictive assumptions, like a point or circular representation of robots or considering movements in any direction, that is, ignoring kinematic restrictions. C-Space is not an appropriate space representation for reactive methods due to its complexity, which prohibits real-time execution. Hence simplifications of C-Space have been proposed specifically for reactive methods. Finally, combinations of the two above approaches have also been proposed (Khatib et al., 1997; Lamiraux et al., 2004; Quinlan and Khatib, 1993), which usually start computing a planned path based on a known static map, and then deform it dynamically to avoid collision with unexpected obstacles. These hybrid approaches successfully solve the navigation problem in many situations, but purely reactive methods are still required for partially known or highly dynamic scenarios, where an a priori planned path may need excessive deformation to be successfully constructed by a hybrid method.
In this work we address purely reactive methods exclusively, concretely, the problem of reactively driving a kinematically-constrained, any-shape mobile robots in a planar scenario. This problem requires finding movements that approach the target location while avoiding obstacles and fulfilling the robot kinematic restrictions. Our main contribution is related to the process for detecting free-space around the robot, which is the basis for a reactive

navigator to decide the best instantaneous motor action. For this task, existing methods consider certain families of simple paths for measuring obstacle distances (which is equivalent to sampling the free-space). These families of paths, that we will call *path models*, must be considered not as planned paths but as artifacts for taking nearby obstacles into account. All existent reactive methods use path models that are an extension of the robot short-term action, as illustrated in Fig. 1: for holonomic robots that can freely move at any direction, straight lines are used, while for non-holonomic robots virtually all methods employ circular arcs.



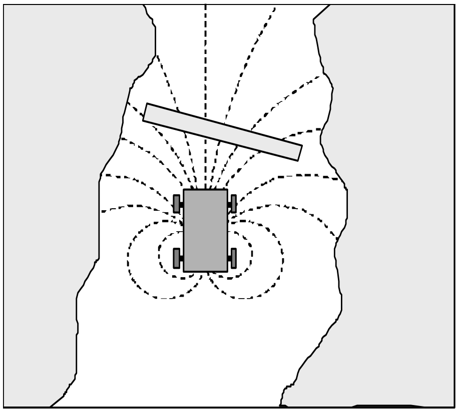
(a) (b)

Figure 1. (a) Holonomic robots can move following straight lines without restrictions, while (b) realistic non-holonomic robot can only move following sequences of circular arcs

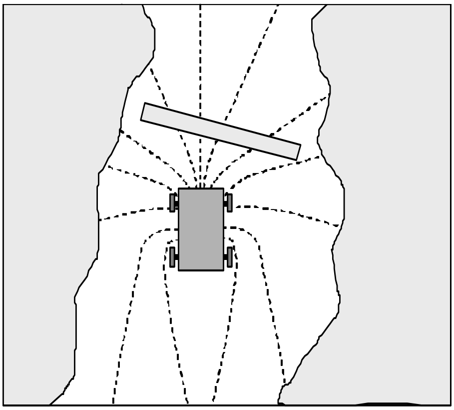
We claim that straight and circular paths, used in previous reactive methods, are just two out from the infinity of path models that can be followed by a robot in a memory-less system, that is, reactively. It is clear that considering other path models is more appropriate to sample the free-space than using the classic straight or circular models only. We shed light into this issue through the example in Fig. 2, where a robot (reactively) looks for possible movements. If we employ a single circular path model for sampling obstacles as in Fig. 2(a), it is very likely that the obstacle avoidance method overlooks many good potential movements – notice that any reactive method must decide according solely to the information that path models provide about obstacles. In contrast, using a diversity of path models, as the example shown in Fig. 2(b), makes much easier to find better collision free movements. This is one of the distinctive features of our approach: the capability to handle a variety of path models simultaneously.

A fundamental point in the process of using path models to sample obstacles is that *not any arbitrary* path model is suitable for this purpose, since it must assure that the robot kinematic constrains are fulfilled while still being able of following the paths in a memory-less fashion, i.e. by a reactive method. It is worth discussing the properties of trajectories that fulfill this condition, called *consistent reactive trajectories* in Section 2.2, since it is an important reflection that cannot be found in previous works.

To motivate the discussion, consider the robot in Fig. 3(a), which must decide its next movement from a family of circular arcs, each one giving a prediction for the distance-to-obstacles. Since reactive navigation is a discrete time process, the decision will be taken iteratively, in a timely fashion, though at each time step the family of paths will be considered starting at the *current pose* of the robot. The central issue here is that, implicitly, it is assumed that if the robot chooses one path at some instant of time, at the next time step it will have the possibility of *continuing along the same trajectory*. Otherwise, the prediction of distance-to-obstacles would be useless since foreseen trajectories can not be actually followed. In the case of circular arcs, this property indeed holds, as illustrated in the example in Fig. 3(b). The main contribution of the present work is a detailed formalization of this and other properties that need to hold for a path model being applicable to obstacle avoidance.



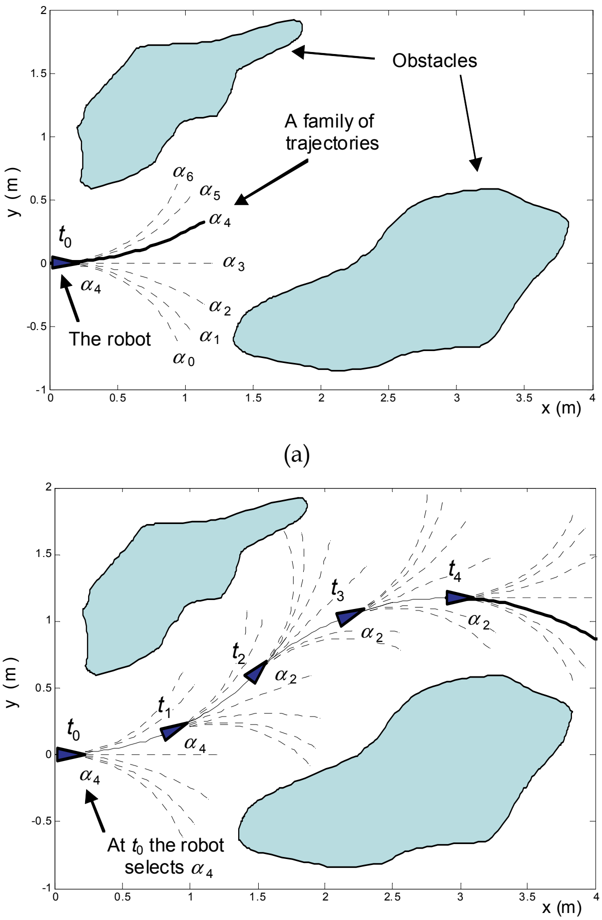
(a)



(b)

Figure 2. Reactive methods take obstacles into account through a family of paths, typically circular arcs (a). However, we claim that other possibilities may be useful for finding good collision-free movements, as the path family shown in (b)

As detailed in previous works (Blanco et al., 2006; Blanco et al., 2008), we decouple the problems of kinematic restrictions and obstacle avoidance by using path models to transform kinematic-compliant paths and real-world obstacles into a lower complexity space, a Trajectory Parameter Space (TP-Space for short). The transformation is defined in such a way that the robot can be considered as a free-flying-point in the TP-Space since its shape and kinematic restrictions are already embedded into the transformation. We can then entrust the obstacle avoidance task to any standard holonomic method operating in the transformed space.



(b)

Figure 3. A schematic representation of the process involved in reactive navigation. At each time step, the robot employs a family of paths to sample the obstacles in the environment, and then chooses the most convenient action according to that information. It must be highlighted the important implicit assumption in the process, that the robot will be able to continue trajectories chosen at previous time steps. Since this does not hold in general for all path models, we develop in this work a template for path models that are proven to fulfill this requirement

This idea was firstly introduced by Minguez and Montano in (Minguez et al., 2002), and has subsequently evolved in a series of works (Minguez et al., 2006; Minguez and Montano, 2008). Our framework can be seen as an extension of (Minguez et al., 2002) since multiple space transformations can be defined instead of just the one corresponding to circular arcs. We allow any number of space transformations by generalizing path models through Parameterized Trajectory Generators (PTGs), which are described in detail in subsequent sections. For further details on how our framework can be integrated into a real navigation system, and for extensive experimental results from both simulations and real robots, the reader is referred to our previous works (Blanco et al., 2006; Blanco et al., 2008).

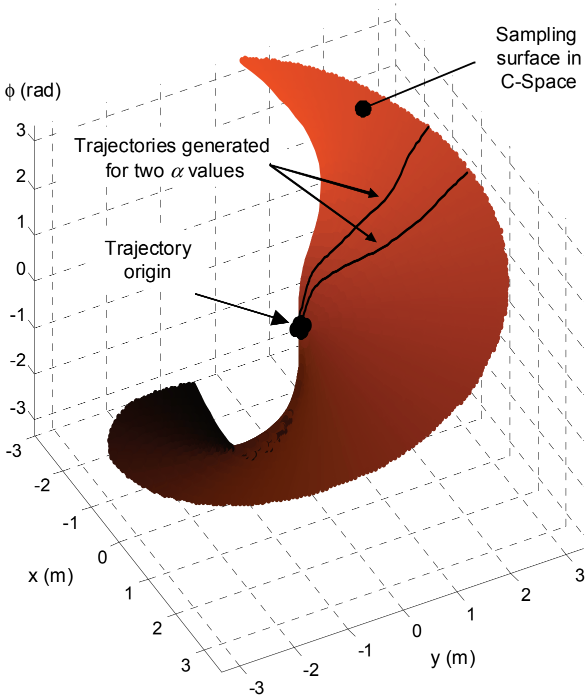


Figure 4. The simultaneous representation of all the trajectories of a path family in C-Space generates a 3D surface which comprises all the potential poses the robot can reach using the path family

2. Space Transformations for Obstacle Avoidance

2.1 Overview

Although not always put explicitly, any reactive navigation algorithm relies on the calculation of distance-to-obstacles to provide the robot with information for choosing the next movement. To the best of our knowledge, all previous (reactive) works make an implicit assumption that has never been questioned: distance-to-obstacles (i.e. collision distances) are computed by means of a single fixed path model: either straight or circular, commonly depending on the robot being holonomic or not. Distances are then taken along those 2D paths, though robot paths are actually defined as continuous sequences of locations and orientations, that is, as three-dimensional curves in C-Space – refer to Fig. 4. We propose instead to define distance-to-obstacles directly in C-Space, as described next.

If all the paths from a given path model are represented in C-Space simultaneously we obtain a 3D surface, as the example in Fig. 4. We will refer to these surfaces as *sampling surfaces*, since distance-to-obstacle can be computed by measuring the distance from the origin to the intersection of those surfaces with C-Obstacles. Next we can *straighten out* the surface into a lower dimensionality space where obstacle avoidance becomes easier, that is, a TP-Space. In this process the topology of the surface is not modified. Since we are proposing a diversity of path models to be used simultaneously, we will have different associated sampling surfaces in C-Space to compute distance-to-obstacles. The whole process is illustrated in Fig. 5.

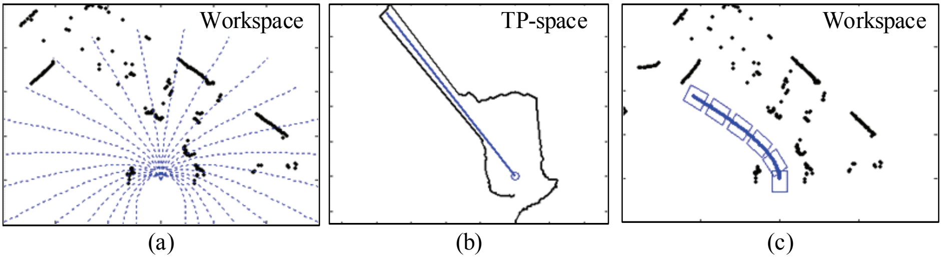


Figure 5. The process to apply simple obstacle avoidance methods to any-shape, non-holonomic robots comprises these steps: (a) A family of path is used to sample distance-to-obstacles, which gives us the obstacles in the transformed space (TP-Space), where (b) the obstacle avoidance method chooses a preferred direction. This straight line in TP-Space actually corresponds to a feasible path, as shown in (c)

We define a TP-Space as any two-dimensional space where each point corresponds to a robot pose in a sampling surface. It is convenient to consider points in a TP-Space by their polar coordinates: an angular component α and a distance *d*. In this way the angular coordinate has a closed range of possible values. The mapping between a TP-Space and a sampling surface is carried out by selecting an individual trajectory out from the family using the coordinate, while d establishes the distance of the pose along that selected trajectory.

This idea of applying obstacle avoidance in a transformed space was introduced in (Minguez et al., 2002), where the authors employed the Euclidean distance in the 2D plane, disregarding the robot orientation, as the distance measure for *d*. Alternatively, we measure distances through a non-Euclidean metric, directly along C-Space sampling surfaces. This has the advantage of taking into account robot turns, thus providing a more realistic estimate of how much a robot needs to move to follow a given trajectory. The region of interest in TP-space is a circle centered at the origin and of radius R_m (a constant that settles the collision avoidance maximum foresee range). We will refer to the TP-space domain as the 2D space $\mathbb{S}\times D$, with $\mathbb{S}=\left]-\pi,\pi\right]$ and $D=\left[0,R_m\right]$. Note as well that the transformation is applied at each iteration of the navigation process, thus for all our derivations the robot is always at the origin.

2.2 Definitions

We define a 2D robot trajectory for a given parameter value as:

$$\mathbf{q}(\alpha,t)=\begin{bmatrix} x(\alpha,t) \\ y(\alpha,t) \\ \phi(\alpha,t) \end{bmatrix} \qquad , t \geq 0 \tag{1}$$

Since we address PTGs for realistic robots subject to non-holonomic constrains, trajectories are defined as the integration of their time derivative $\dot{\mathbf{q}}(\alpha,t)$, that is,

$$\mathbf{q}(\alpha,t)=\int_0^t \dot{\mathbf{q}}(\alpha,\tau)\,d\tau \tag{2}$$

where it applies the initial condition q(, 0) = 0 for any . Note from Eq. (4) that we define the transformed space in terms of distance d rather than time t, in which the kinematic equations are naturally defined. The reason for this change of variable is that we are interested in the geometry of paths, which remains unmodified if the velocity vector u(·) is multiplied by any positive scalar, an operation equivalent to modifying the speed of the robot dynamically. For example, it is common in navigation algorithms to adapt the robot velocities to the clearness of its surroundings.
Therefore, we define a PTG as:

$$PTG(\alpha,d) \triangleq \mathbf{q}\left(\alpha,\mu_\alpha^{-1}(d)\right) \tag{3}$$

where the function $\mu_\alpha^{-1}(d)$, mapping distances *d* to times *t*, is not relevant at this point and will be introduced later on. Thus, a PTG is a mapping of TP-Space points to a subset of C-Space:

$$\begin{aligned} PTG: \quad \mathbb{S}\times D \quad &\mapsto \quad \mathbb{R}^2\times\mathbb{S} \\ (\alpha,d) \quad &\mapsto \quad \mathbf{q} \end{aligned} \tag{4}$$

In the common case of car-like or differentially-driven robots, the derivatives in Eq. (2) are given by the same set of kinematic equations:

$$\begin{aligned} \dot{\mathbf{q}}(\alpha,t) &= \mathbf{J}\left(\mathbf{q}(\alpha,t)\right)\mathbf{u}(\alpha,t) \\ &= \begin{bmatrix} \cos\phi(\alpha,t) & 0 \\ \sin\phi(\alpha,t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(\alpha,t) \\ \omega(\alpha,t) \end{bmatrix} \end{aligned} \tag{5}$$

Here u is the vector comprising the linear (v) and angular (ω) velocities of the robot at each instant of time t and for each value of the PTG parameter. The freedom for designing different PTGs is therefore bound up with the availability of different implementations of the actuation vector $\mathbf{u}(\alpha,t)$.

In despite of the fact *any* function $\mathbf{u}(\cdot)$ represents a kinematically valid path for a robot, which follows from Eq. (5) by definition, the present work is built upon the realization that not any arbitrary function leads to *valid* space transformations for obstacle avoidance methods. We specify next when such a transformation is valid for our purposes.

Definition. A space transformation between C-space and TP-space is *valid* when it fulfils the following conditions:

- **C1**. It generates *consistent reactive trajectories*. All path models are not applicable to reactive navigation because of the memoryless nature of the movement decision process, as discussed in section 1.
- **C2**. It is *WS-bijective*. For each WS location (*x*,*y*), at most one trajectory can exist taking the robot to it, regardless the orientation. Otherwise, the target position would be seen at two different directions (straight lines) in TP-Space – recall that a PTG maps straight lines of the TP-Space into trajectories of the C-Space.
- **C3**. It is *continuous*. Together with the last restriction, this condition assures that transformations do not modify the topology of the real workspace around the robot.

These three conditions hold for the case of paths that are circular arcs. The main contribution of the present work is the following theorem, which proves that a broader variety of *valid* PTGs indeed exist and is suitable to reactive navigation.

Theorem 1. *A sufficient, but not necessary, condition for a PTG to be valid is that its velocity vector is of the form:*

$$\mathbf{u}(\alpha,t) = \begin{bmatrix} v_m \cdot f_v\left(a\alpha + b\phi(\alpha,t)\right) \\ \omega_m \cdot \left(a\alpha + b\phi(\alpha,t)\right) \end{bmatrix} \tag{6}$$

where v_m *and* ω_m *settle the desired maximum linear and angular velocities in absolute value, respectively,* $f_v(\alpha,t)$ *is any Lipschitz continuous function which evaluates to non-zero over the whole domain, and* a , b *are arbitrary constants with the restrictions* $0 < |a/b| \leq 1$ *and* $b < 0$ *. Furthermore, the velocity vector becomes fully defined by just specifying its value for* $t = 0$ *.*

The following section is devoted to a detailed analysis of PTGs in this form and to prove our claim of them always are valid in the sense that they fulfill all the conditions listed above.

3. Proofs

We start by defining the function $\mu_\alpha(t)$ as the distance traveled by the robot along a given trajectory α in C-space from the origin and until the instant *t*, that is:

$$\mu_\alpha(t) = \int_0^t \left\| \frac{\partial \mathbf{q}(\alpha,t)}{\partial t} \right\| d\tau \tag{7}$$

where the norm could be the Euclidean distance, though we will employ here a custom metric introduced in (Blanco et al., 2008), which accounts for robot turns through a constant ρ that roughly represents the robot radius, leading to:

$$\mu_\alpha(t) = \int_0^t \left(v(\alpha,t)^2 + \rho^2 \omega(\alpha,t)^2 \right)^{\frac{1}{2}} d\tau \tag{8}$$

Then, we can state the following lemma about the existence of $\mu_\alpha^{-1}(d)$, required in Eq. (3) for the definition of PTGs.

Lemma 1. *The function* $\mu_\alpha : t \mapsto d$ *is continuous and its inverse* $\mu_\alpha^{-1} : d \mapsto t$ *is well-defined for any* $t \geq 0$.

Proof. The first part, proving the continuity of $\mu_\alpha(t)$ is trivial since the function is defined as an integral, therefore it is differentiable. Next, it can be seen that the function is strictly increasing due to its derivative being the norm of $\dot{\mathbf{q}}$, which in general is non-negative, but given the hypothesis from theorem 1 of f_v evaluating to non-zero over all the domain, the case $\dot{\mathbf{q}} = 0$ can be ruled out. Being continuous and strictly-increasing $\mu_\alpha(t)$ becomes bijective for any $t \geq 0$ thus its inverse is well-defined.

An important feature of any valid PTG is that different values of α must generate unique trajectories (see condition C2), which is assured by the following lemma.

Lemma 2. *Provided* $b < 0$ *and* $0 < |a/b| \leq 1$, *then each value* $\alpha \in \mathbb{S}$ *determines a unique trajectory passing through the origin with its heading tending to* $-\alpha a/b$ *as* $t \to \infty$.

Proof. Since $\dot{\mathbf{q}}(\alpha,t)$ is Lipschitz continuous, and given the initial conditions $\mathbf{q}(\alpha,t) = \mathbf{0}$ for any value of α, there exists only one trajectory for each α value (Evans and Gariepy, 1992), which is determined by the value of $\dot{\mathbf{q}}(\alpha,0) = \left[v(\alpha,0) \quad \omega(\alpha,0) \right]^T$. From the hypotheses of theorem 1 it easily follows $\omega(\alpha_1,0) \neq \omega(\alpha_2,0)$ for any $\alpha_1 \neq \alpha_2$ as long as a $a \neq 0$ (refer to Eq. (6)), thus the uniqueness of each trajectory is assured.

Regarding the limit of the robot heading $\phi(\alpha,t)$, we can solve the differential equation of the kinematic model in Eq. (5) for this term, that is:

$$\begin{aligned} \dot{\phi}(\alpha,t) &= \omega(\alpha,t) \\ &= \omega_m \cdot \left(a\alpha + b\phi(\alpha,t) \right) \end{aligned} \tag{9}$$

which can be straightforwardly solved giving us:

$$\phi(\alpha,t) = -\alpha \frac{a}{b} \left(1 - e^{\omega_m b t} \right) \tag{10}$$

The parameter b determines the evolution of the heading over time. Since the robot heading must be bounded to the domain of $\mathbb{S}$, we discard the values $b > 0$. The case $b = 0$ must be avoided as well since in that case Eq. (10) is not defined. Therefore, for the valid values $b < 0$, the heading converges to:

$$\lim_{t\to\infty} \phi(\alpha,t) = -\alpha \frac{a}{b} \tag{11}$$

Notice the condition $0 < |a/b| \le 1$ assures $\phi(\alpha,t)$ will always remain within its valid domain $\mathbb{S}$.

We address next the fundamental property of generated paths being consistent reactive trajectories – as stated by condition C1. The geometrical meaning of this property was discussed in section 1, recall Fig. 3 , and is now stated formally as follows.

Lemma 3. *For any* $\alpha \in \mathbb{S}$ *and* $t_0 \ge 0$, *there exists a function* $A(\alpha,t_0) : \mathbb{S}\times\mathbb{R}^{0+} \mapsto \mathbb{S}$ *such as:*

$$\mathbf{q}(\alpha, t_0 + t) = \mathbf{q}(\alpha, t_0) \oplus \mathbf{q}(\alpha', t) \qquad , \forall t \ge 0 \tag{12}$$

with $\alpha' = A(\alpha, t_0)$ *and where the* $\oplus$ *operator stands for pose composition (Smith et al., 1988).*

Proof. It can be trivially shown that this statement holds for $t = 0$, when Eq. (12) becomes:

$$\begin{aligned} \mathbf{q}(\alpha,t_0) &= \mathbf{q}(\alpha,t_0) \oplus \mathbf{q}(\alpha',0) \\ &= \mathbf{q}(\alpha,t_0) \oplus \mathbf{0} \\ &= \mathbf{q}(\alpha,t_0) \end{aligned} \tag{13}$$

Now, since both trajectories $\mathbf{q}(\alpha, t_0 + t)$ and $\mathbf{q}(\alpha,t_0) \oplus \mathbf{q}(\alpha',t)$ pass through a common point in C-space at $t = 0$, it is enough to prove that their derivatives $\dot{\mathbf{q}}$ are identical at that instant for Lemma 2 to imply that both trajectories coincide for any $t > 0$.

Taking into account the change of coordinates introduced by the pose composition operator, the condition of both time derivatives $\dot{\mathbf{q}}(\cdot)$ must coincide amounts to their velocity vectors $\mathbf{u}(\cdot)$ being identical at $t = 0$, that is, we must prove:

$$\mathbf{u}(\alpha',0) = \mathbf{u}(\alpha,t_0) \tag{14}$$

By noticing from Eq. (6) that $\mathbf{u}$ is a function of the term $a\alpha + b\phi(\alpha,t)$, the above condition can be rewritten as:

$$a \underbrace{\alpha'}_{A(\alpha,t_0)} + b\underbrace{\phi(\alpha',0)}_{0} = a\alpha + b\phi(\alpha,t_0)$$

$$aA(\alpha,t_0) = a\alpha + b\phi(\alpha,t_0) \tag{15}$$

$$A(\alpha,t_0) = \alpha + \frac{b}{a}\phi(\alpha,t_0)$$

where $\phi(\alpha,t_0)$ is given by the closed form expression in Eq. (11).

It is interesting to highlight that the resulting expression for $\alpha' = A(\alpha,t)$ indicates that α' is well-behaved, in the sense that it never exceed the limits $]-\pi,\pi]$. It also reveals that all trajectories eventually become a straight path, as can be seen by taking the limit:

$$\lim_{t\to\infty} A(\alpha,t_0) = \alpha - \frac{a}{b}\alpha\frac{b}{a} = 0 \tag{16}$$

where the fact that $\alpha = 0$ generates a straight trajectory follows from the PTG design equations in theorem 1. Note how the final part of all the trajectories being identical to one of them aligns perfectly with our goal of *consistent reactive trajectories* (condition C1).

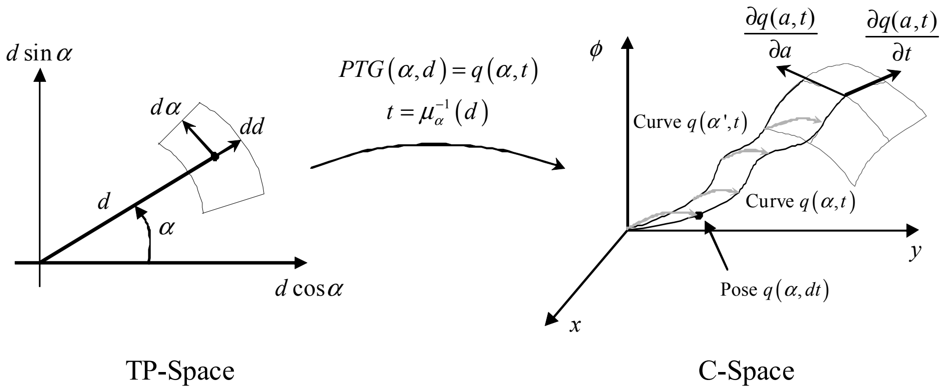


Figure 6. A schematic representation of the mapping a PTG performs between TP-Space points and C-Space robot poses. We represent the infinitesimal elements used in the proof of lemma 4. Basically, the idea represented here is that the curve for the trajectory $\mathbf{q}(\alpha',t)$ matches precisely trajectory $\mathbf{q}(\alpha,t)$ if the coordinate origin of the former is changed to $\mathbf{q}(\alpha,dt)$ for some α' infinitesimally close to α

Finally, the last requisite of a valid PTG (condition C3) is to generate *continuous* sampling surfaces in C-space, that is, the function $PTG(\alpha,d)$ must be continuous.

Lemma 4. *Given the hypotheses of theorem 1,* $PTG(\alpha,d)$ *is a 2-manifold of C-space with boundaries, and is continuous and derivable over the whole domain* $(\alpha,d)\in \mathbb{S}\times D$.

Proof. Firstly, due to lemma 1 it is enough to prove the continuity and differentiability of $\mathbf{q}(\alpha,t)$, since the mapping between distances d and times t conserves those properties of $\mathbf{q}$.

We show next that $\mathbf{q}(\alpha,t)$ has well-defined derivatives, which in turns implies it is continuous. For the case of $\frac{\partial \mathbf{q}(\alpha,t)}{\partial t}$ the proof is trivial since by definition this derivative is given by Eq. (5).

The derivation of $\frac{\partial \mathbf{q}(\alpha,t)}{\partial \alpha}$ is more involved. It is illustrative to keep Fig. 6 as a reference through the following derivations to clarify the geometrical meaning of each term. Let dt be an infinitesimal increment in time, and (α,t) some fixed point in the domain of TP-space. Then, using lemma 3.3 we can rewrite $\mathbf{q}(\alpha,t+dt)$ as:

$$\mathbf{q}(\alpha,t+dt)=\mathbf{q}(\alpha,dt)\oplus\mathbf{q}(\alpha',t) \tag{17}$$

where α' is given by:

$$\begin{aligned} A(\alpha,t) &= \alpha+\frac{b}{a}\phi(\alpha,dt) \\ &= \alpha+\underbrace{\frac{b}{a}\omega(\alpha,0)\,dt}_{d\alpha} \\ &= a+d\alpha \end{aligned} \tag{18}$$

Making use of the definition of pose composition operators (Smith et al., 1988) we can rearrange Eq. (17) as follows:

$$\mathbf{q}(\alpha',t)=\mathbf{q}(\alpha,t+dt)\ominus\mathbf{q}(\alpha,dt) \tag{19}$$

The geometrical meaning of this operation is that, as illustrated in Fig. 6, the curve $\mathbf{q}(\alpha',t)$ matches the curve $\mathbf{q}(\alpha,t)$ if translated and rotated to the pose $\mathbf{q}(\alpha,dt)$. As a result, this means that infinitesimal changes $d\alpha$ in a pair (α,t) leads to infinitesimal changes in $\mathbf{q}(\alpha,t)$ that can be written down as:

$$\mathbf{q}(\alpha + d\alpha, t) = \mathbf{q}(\alpha, t) \oplus \mathbf{J}(\mathbf{q}(\alpha, t))\mathbf{u}(\alpha, t)\,dt \ominus \begin{bmatrix} v(\alpha, 0)\,dt \\ 0 \\ \omega(\alpha, 0)\,dt \end{bmatrix} \quad (20)$$

which follows from Eq. (19) and the definition of $\mathbf{q}$ as an integral of the velocity vector $\mathbf{u}$. Since the pose composition $\oplus$ and inverse composition $\ominus$ operators are both continuous and differentiable, it follows that the derivative $\frac{\partial \mathbf{q}}{\partial \alpha}$ is well-defined for any point (α, t). Finally, given $\mathbf{q}(\alpha, t)$ is differentiable and so is $PTG(\alpha, d)$ at the whole domain of (α, d), the surface generated by a PTG can be seen as a 2-manifold with boundaries (Spanier, 1981).

4. Related work

In this section we review the most well-known space representations that have been employed in mobile robot motion planning and collision avoidance, and put them in contrast with our approach.

The C-Space has been extensively used in many fields, including robotic manipulators (Lozano-Perez, 1987), maneuver planning (Latombe, 1991), and mobile robot motion planning (Murphy, 2000). The complexity derived from its high dimensionality makes C-Space not applicable to real-time reactive navigation.

A first simplification for dealing with C-Space more efficiently is to assume a circular robot. Thus, C-Obstacles are no longer dependent on the robot orientation and the C-Space reduces to a planar space, the Workspace (WS). This space is employed by the well known potential field methods, like the VFF (Borenstein and Koren, 1989), VFH (Borenstein and Koren, 1991), and others (Haddad et al., 1998, Balch, 1993). Other reported methodologies are based on neural nets (Pal and Kar, 1995) and, more recently, the Nearness-Diagram (ND) approach (Minguez and Montano, 2004), which relies on a divide-and-conquer strategy that defines a set of different states according to the arrangement of nearby obstacles. All these methods deal with circular shaped robots, a too restrictive assumption for many real-life situations. For example, if a robotic wheelchair were assumed to be circular, it would never pass through a narrow doorway.

Most of the approaches that deal with any-shape robots and take into account their kinematic restrictions propose working with another less limiting simplification of C-Space: the velocity space (Arras et al., 2002; Feiten et al., 1994; Ramírez and Zeghloul, 2001; Schlegel, 1998; Simmons, 1996), or V-Space for short. For mobile robots of our interest, V-Space represents the space of the potential linear and angular robot velocities, hence the next movement can be chosen as a point in V-Space that results in constant curvature paths (i.e. circular paths). A common feature in many V-Space methods is the inclusion of a dynamic window (Fox et al., 1997), which restricts the range of reachable velocities to that compatible with the robot maximum acceleration. An important limitation of these methods is that, although many obstacles may be sensed, not all of them are exploited: only those ones falling into the robot dynamic window for the next step are considered for choosing the instantaneous motion command. It is clear that better paths would be found if more comprehensive obstacle information were taken into account, which indeed implies looking ahead for more than one step, as our approach does. In addition to the utilization of a dynamic window, most V-Space approaches use only the family of circular paths to sample the free-space, which entails the risk of not detecting many free-space areas. There are some exceptions (Ramírez and Zeghloul, 2001; Xu and Yang, 2002) that make use of straight paths, but this model is not appropriate for most actual mobile robots. Only these two path models have been reported in the reactive collision avoidance literature. While a generic path can only be described in the three-dimensional C-Space (2D position plus heading), poses along a circular path can be defined through two parameters: the path curvature and the distance along the arc. Upon this parameterization, a TP-Space was proposed in (Minguez et al., 2006) as an elegant and mathematically sound alternative to V-Space: if the navigation is carried out in the 2D TP-Space, the robot can be treated as a free-flying-point. That work demonstrates that navigation in a parameterized space allows us to decouple the problems of kinematic restrictions and obstacle avoidance. However, this approach has never been extended neither to cope with other path models apart from the circular one, nor to a number of different transformations, which are the contributions of the present work.

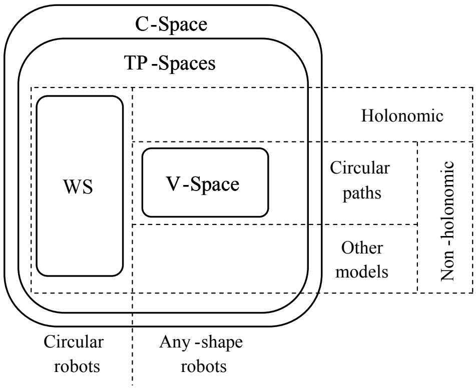


Figure 7. A summary with the classification of space representations used in obstacle avoidance

To further clarify the relationship between the different existing representation spaces, please refer to Fig. 7, where TP-Spaces appear as a generalization of spaces such as WS and V-Space. However, it must be remarked that C-Space is the most general space representation, but at the price of an elevated computational cost due to its high dimensionality.

5. Conclusions

In this work we have reviewed existing methods for obstacle avoidance and reactive navigation, and discussed how space transformations can be employed to extend their applicability to kinematically-constrained and any-shape mobile robots, making use of a clear and useful separation of the problems of robot shape and kinematic restrictions, and collision avoidance. We have developed a generalized kinematics abstraction mechanism which allows us using a variety of path models (PTGs) to obtain a better sampling of the whole C-Space from which more and better collision-free paths towards the target location can be found. We have settled the conditions that a PTG must hold to lead to a valid space transformation, and then we have introduced a PTG template which has been proven to always fulfill all the requirements. However, it must be remarked that theorem 1 determines a sufficient, but not necessary condition, thus indeed more valid PTGs may exist out of the given template (a prominent example are circular arcs). Finally, we should highlight that the applicability of PTGs is not limited to purely reactive navigation frameworks, hence their integration with hybrid planned-reactive approaches reveals as a promising research topic for the future.

6. References

Arkin, R. (1998). *Behavior-Based Robotics*. MIT Press.

Arras, K., Persson, J., Tomatis, N., and Siegwart, R. (2002). *Real-time obstacle avoidance for polygonal robots with a reduceddynamic window*. Vol. 3.

Balch, T. (1993). Avoiding the past: a simple but effective strategy for reactivenavigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 678–685.

Blanco, J., Gonzalez, J., and Fernández-Madrigal, J. (2006). The Trajectory Parameter Space (TP-Space): A New Space Representation for Non-Holonomic Mobile Robot Reactive Navigation. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1195–1200.

Blanco, J., Gonzalez, J., and Fernández-Madrigal, J. (2008). *Extending Obstacle Avoidance Methods through Multiple Parameter-Space Transformations. Autonomous Robots*, 24(1):29–48.

Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fact mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187.

Borenstein, J. and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobilerobots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288.

Evans, L. and Gariepy, R. (1992). *Measure Theory and Fine Properties of Functions*. CRC Press.

Feiten, W., Bauer, R., and Lawitzky, G. (1994). *Robust obstacle avoidance in unknown and cramped environments*. pages 2412–2417.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Haddad, H., Khatib, M., Lacroix, S., and Chatila, R. (1998). Reactive navigation in outdoor environments using potential fields. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2.

Khatib, M., Jaouni, H., Chatila, R., Laumond, J., and LAAS-CNRS, T. (1997). Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4.

Lamiraux, F., Bonnafous, D., and Lefebvre, O. (2004). Reactive path deformation for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 20(6):967–977.

Latombe, J. (1991). *Robot Motion Planning*. Kluwer Academic Publishers.

Lozano-Perez, T. (1983). Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, 32(2):108–120.

Lozano-Perez, T. (1987). A simple motion-planning algorithm for general robot manipulators. *IEEE Journal of Robotics and Automation*, 3(3):224–238.

Minguez, J. and Montano, L. (2004). Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59.

Minguez, J. and Montano, L. (2008). Extending Reactive Collision Avoidance Methods to Consider any Vehicle Shape and the Kinematics and Dynamic Constraints. *IEEE Transactions on Robotics*.

Minguez, J., Montano, L., and Santos-Victor, J. (2002). Reactive navigation for non-holonomic robots using the ego-kinematic space. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3.

Minguez, J., Montano, L., and Santos-Victor, J. (2006). Abstracting Vehicle Shape and Kinematic Constraints from Obstacle Avoidance Methods. *Autonomous Robots*, 20(1):43–59.

Murphy, R. (2000). *Introduction to Ai Robotics*. MIT Press.

Pal, P. and Kar, A. (1995). Mobile robot navigation using a neural net. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2.

Quinlan, S. and Khatib, O. (1993). Elastic bands: connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 802–807.

Ramírez, G. and Zeghloul, S. (2001). Collision-free path planning for nonholonomic mobile robots using a new obstacle representation in the velocity space. *Robotica*, 19(05):543–555.

Schlegel, C. (1998). *Fast local obstacle avoidance under kinematic and dynamicconstraints for a mobile robot*. volume 1.

Simmons, R. (1996). The curvature-velocity method for local obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4.

Smith, R., Self, M., and Cheeseman, P. (1988). A stochastic map for uncertain spatial relationships. *The fourth international symposium on Robotics Research*, pages 467–474.

Spanier, E. (1981). *Algebraic Topology*. Springer.

Xu, H. and Yang, S. (2002). Real-time collision-free motion planning of nonholonomic robots using a neural dynamics based approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3.