

## Switching Manipulator Control for Motion on Constrained Surfaces

Xanthi Papageorgiou · Herbert G. Tanner ·  
Savvas G. Loizou · Kostas J. Kyriakopoulos

the date of receipt and acceptance should be inserted later

**Abstract** We develop non-smooth motion controllers that enable a manipulator to perform surface tasks that do not involve force/torque interactions, while maintaining bounded joint rates. The robot’s end-effector is first stabilized to a point close to the surface, and then it tracks a trajectory on the surface, while avoiding certain prescribed regions. We show that the resulting closed loop system is uniformly asymptotically stable, and we verify our analytical development with experiments on a Mitsubishi PA10-7C robotic manipulator.

**Keywords** Robot navigation · tracking · non-smooth analysis

---

This paper is part of the 03ED 77 research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from E.U.-European Social Fund). Also, this work is partially supported by the European Commission through contract “FP6 - IST - 001917 - NEUROBOTICS: The fusion of Neuroscience and Robotics”, and by Eugenides Foundation Scholarship.

---

X. Papageorgiou and K.J. Kyriakopoulos  
National Technical University of Athens, 9 Heroon Polytechniou St, Zografou Athens 15780,  
Greece  
Tel.: +30-210-772-4012  
E-mail: xpapag@mail.ntua.gr, kkyria@mail.ntua.gr

H.G. Tanner  
110 Spencer Lab, University of Delaware, Newark, DE 19716, USA  
Tel.: +1 (302) 831-6888  
E-mail: btanner@udel.edu

S.G. Loizou  
Frederick University, 7 Y. Frederickou St, Pallouriotisa, Nicosia 1036, Cyprus  
Tel.: + 357-22-345159  
E-mail: s.loizou@frederick.ac.cy

## 1 Introduction

### 1.1 Motivation

Hybrid Bionic Systems (HBS) link via neural interfaces the human nervous system with electronic and/or robotic artifacts. This research thrust aims not only at developing systems for restoring motor and sensory functions of injured and disabled people (Fig. 1), but also on exploring the possibility of augmenting sensorymotor capabilities of humans whenever appropriate.



**Fig. 1** Recovering human performance example. An intelligent exoskeleton is worn by a motor disabled person, in order to re-gain the use of the upper limb.

Conceptually, a robotic device may be interfaced with a human neural system. Such a robot processes electromyographic activity and uses it in the control loop that moves the mechanical links. In realizing such a neuro-robotic system, several challenges need to be met. One such challenge is using noisy neural signals as reference inputs in the feedback loop; another is the requirement to mimic human behavior during obstacle avoidance and interaction with non-planar surfaces.

It is difficult to use only human neural signals to control an artifact. A human arm cannot use the sensory feedback from the robot in order to directly control it, and the artifact does not have the exact same morphology as the human arm to be able to interpret the neural signals perfectly. This mismatch can jeopardize the person and inflict damage on the environment. Despite the significant amount of work on modeling and decoding the signals from the human brain or from a part of the human body [1–4], for safety reasons it is necessary to establish a low level control loop around the artifact, for the exoskeleton's collision avoidance and motion control. Further motivation for the design of such a control loop comes from robotic tele-operation applications, where the robot's environment may be dramatically different compared to that in which the human operates. Then, analysis of the human's signals alone may be insufficient to safely complete the desired task.

Another issue that needs to be considered in the design of neuro-robotic systems, (consisting of an exoskeleton cooperating with the human upper limb), is the fact that

---

motors must apply appropriate torque in order for the exoskeleton to aid, rather than impede, arm movement. The design of most exoskeletons is based on neuro-scientific models, which dictate an agonistic - antagonistic behavior of the mechanism joints. The robotic device motion is therefore constrained.

Another important types of applications that motivate this work come from the case of autonomous robotic surface painting, cleaning, inspection, etc (e.g. robotic manipulation for automotive applications). The main difficulties of the those tasks apart from the problems of joint limitations, the constrained motion etc, arise when the considered surface is not planar, and might includes bad regions that must be avoided.

In this paper we address the challenge of designing such controllers for the artifact. Our controllers drive the robot with safety for surface manipulation and compliant motion without collisions with objects in the environment, while considering input constraints which reflect motion limitations.

## 1.2 Related Work

There are several applications where a robotic manipulator performs a task along a surface, such as painting, cleaning, and inspection. All these applications pose similar control design problems, however, existing approaches typically rely on restrictive assumptions. Related work on robotic manipulation for automotive applications focuses, for example, on painting surfaces that are convex and have no holes [5–8]. The generation of the trajectory in order to cover the area in [6] is a three steps problem: first the problem of start curve selection, then the problem of speed profiles along each pass selection, and third the problem of spacing between the passes selection. Building on these ideas, [5] accounts for the effect of surface curvature, enabling the development of an automated trajectory generator. In [9], [10], the authors map the motion planning problem for articulated, rigid robots operating in 3D, dynamic, and complex environments into simulating a composed dynamical system where the geometric constraints induce some virtual forces that affect the motion of each rigid robot. This approach has applications to assembly line planning, automated car painting, and virtual prototyping, however, it is a computational approach and only simulation results are presented. Finally, the authors in [11], describe a methodology for automatically generating robot programs for spray painting of unknown parts.

The methodologies mentioned do not consider obstacles, holes or regions on the surfaces that need to be avoided, and there is very little emphasis on robot motion planning. In this body of work, motion planning, is considered a subproblem of a more general problem, which for the case of automated painting, for example, could be controlling the ratio of paint deposited on the surface.

Collision avoidance is generally treated at a local level, assuming kinematic redundancy and cartesian subtasks prioritization; there is no requirement for robotic motion on a rigid surface. The real-time collision avoidance problem in case of position-controlled robotic arm with seven degrees of freedom is presented in [12]. The arm’s intrusion into the safety zones of each obstacle introduces some virtual forces and the problem therefore is solved as a position-based force control problem. The collision avoidance and a self-collision avoidance problems for redundant manipulators are presented in [13]. In this work the robot and the environment are modeled as simple geometric schemes (cylinders and spheres).

The authors in [14] present a neural network approach for collision-free motion control of redundant manipulators. The problem formulation represents collision avoidance constraints as dynamic inequalities and incorporates joint physical limits into an optimization problem. The solution of the obstacle avoidance kinematics problem is computed by a dual neural network.

The authors in [15] study the problem of controlling the motion of a non-redundant manipulator towards a surface, with integrated collision avoidance of a obstacle set around the surface, while in [16], [17] a hierarchy of controllers built around belt zones, regions of specific depth defined around the surface is utilized to control the robot, with one controller being used to steer the robot inside the belt zone, a second controller stabilizes the robot inside the belt zone. This allows the motion tasks to be solved by considering the robot to be on the surface, but this methodology does not take into account various kinematic constraints. Building upon these works, in [18] and [19] the methodology is augmented by adding force control. Finally, in a similar context, [20] and [21] study the problem of motion control towards a surface while obeying a set of input constraints, using a non-smooth and a smooth kinematic controller respectively.

Input constraints have also been dealt extensively with in the context of linear systems. In [22], the design of PI controllers for systems with input constraints, (such as actuator saturation) is discussed. Input constraints on nonlinear control systems have been recently treated in the framework of multi-robot motion planning using navigation functions [23], [24]. In [25] such input constraints are being enforced by design in the form of a hybrid system. In [26], another switching controller is introduced that implements specific favorable velocity profiles on multiple micro-robots.

In [27], input constraints (defined as limits on jerk, velocity and acceleration of the joints) are taken into account during trajectory tracking. An algorithm based on the inverse kinematics of the robot manipulators is presented, that given an end-effector path ensures tracking of the path from the robot. For similar applications [28] presents adaptive schemes, and [29], develops a PI sliding mode controller.

### 1.3 Organization and Overview

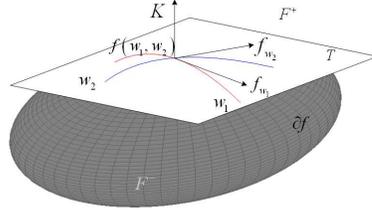
In this work, the problem of designing a controller that steers a manipulator while achieving limited joint velocities within specified bounds is studied. The proposed method lies in constructing a navigation function, and by showing, utilizing techniques in non-smooth analysis, that the closed loop system obeys the desired limits. The design of the system utilizes the concept of belt zones, that allow a clear and easy to define switching scheme. The novel contribution of this paper lies in presenting globally uniformly asymptotically stable dynamic controllers for robotic manipulators with joints, subject to joint rate constraints aiming to

- reference trajectory tracking with obstacle avoidance, and
- stabilization with obstacle avoidance,

on 2-D surfaces which are embedded manifolds in 3-D robot workspaces.

The paper is organized with the following way: A formal definition of the problem is presented in section 2, where all the necessary notation, definitions as well some technical lemmas are introduced. In section 3 a kinematic controller is described, while the stability analysis of the closed loop system is presented in section 4. In Section 5, we backstep the asymptotically stable kinematic controller to complete the design

of the desired dynamic controller. Section 6 presents the experimental results from the application of the proposed control algorithm on the robotic manipulator. The paper concludes with Section 7 that summarizes our main points and highlights future research directions.



**Fig. 2** Representation of the tangent plane  $T = \text{span}\{f_{w_1}, f_{w_2}\}$  and the normal vector  $K$  at point  $f(w_1, w_2)$  of a 3-D surface.

## 2 Problem Statement

Consider a robotic manipulator with  $m$  degrees of freedom, subject to kinematic input constraints, and operating in a workspace some regions of which are considered of limits to the robotic manipulator (modeling obstacles located in the workspace). The goal is for the robot to move proximal to a surface and to then track a specified trajectory on this surface. We make the assumption of a stationary workspace, and that the input to the robot is the direct control of the joint rates of the manipulator. The differential equations describing the robot's dynamics have the form

$$B(q) \cdot \ddot{q} + C(q, \dot{q}) + G_r(q) = \tau,$$

where  $B(q)$  is the inertia matrix,  $C(q, \dot{q})$  is the Coriolis and centrifugal term,  $G_r(q)$  is the gravity term,  $q = [q_1 \dots q_m]^T \in \mathbb{R}^m$  is the vector of arm joint displacements, and  $\tau$  is the vector of joint torque inputs [30]. A computed torque approach  $\tau = B(q) \cdot y + C(q, \dot{q}) + G_r(q)$ , transforms the system into a chain of integrators

$$\ddot{q} = y. \quad (1)$$

In order to successfully implement a computed torque approach, it is necessary to have a detailed model and an accurate identification of the parameters of the robotic system. We use, for this process, a friction model that includes non-linear elements, that is calibrated using experimental identification of the various parameters [31], [32]. Using an experimental procedure in loaded conditions, the stiffness effect of the joints is also identified. Identification experiments are also used for calibrating the the dynamic model parameters, grouped in suitable form.

We denote as  $\mathcal{F} \subset \mathbb{R}^m$  the admissible and feasible subset of the configuration space (workspace) for the manipulator. The obstacle free subset of the workspace is denoted  $\mathcal{F}_{free} \subseteq \mathcal{F}$ . Let  $\mathcal{O} \in \mathcal{F} \setminus \mathcal{F}_{free}$  be the set of all obstacles in the 3-D workspace, and define a vector valued  $C^2$  function

$$f(w_1, w_2) : \mathbb{R}^2 \rightarrow \mathcal{R}(f), \quad (2)$$

which represents a closed surface. The range of  $f$ ,  $\mathcal{R}(f) \subset \mathbb{R}^3$ , expresses mathematically the boundary of the surface, on which (at a small  $\epsilon > 0$  distance) the robot task is to be performed. Now we can decompose the space around the surface (see Fig. 2):

1. The internal region of the surface,  $F^-$ ;
2. The boundary of the surface,  $\partial f$ ;
3. The external region of the surface,  $F^+$ .

We can now define the tangent vectors on the surface, parameterized by  $w_1$  and  $w_2$ :

$$f_{w_1}(w_1, w_2) = \frac{\partial f(w_1, w_2)}{\partial w_1} = \left[ \frac{\partial f_x}{\partial w_1}, \frac{\partial f_y}{\partial w_1}, \frac{\partial f_z}{\partial w_1} \right]^T,$$

$$f_{w_2}(w_1, w_2) = \frac{\partial f(w_1, w_2)}{\partial w_2} = \left[ \frac{\partial f_x}{\partial w_2}, \frac{\partial f_y}{\partial w_2}, \frac{\partial f_z}{\partial w_2} \right]^T,$$

where the  $f_x, f_y, f_z$  denote the coordinate functions of  $f$  along the direction denoted by the respective subscript. Since  $f(w_1, w_2)$  is assumed twice differentiable,  $(f_{w_1} \times f_{w_2}) \neq 0, \forall w_1, w_2 \in \mathbb{R}$  [33], and the vectors  $f_{w_1}, f_{w_2}$  are linearly independent everywhere. Therefore, every vector tangent to the surface is a linear combination of the vectors  $f_{w_1}$  and  $f_{w_2}$  (Fig. 2). We can now introduce the normalized vector  $K$  which is perpendicular to the surface

$$K = \frac{f_{w_1} \times f_{w_2}}{\|f_{w_1} \times f_{w_2}\|}, \quad (3)$$

and we formally state our problem as follows:

We seek a feedback dynamic control law that steers the end-effector of a revolute joint robot manipulator moving with kinematic input constraints in a static and bounded known environment containing a known 2-D surface towards

1. navigating to any feasible surface point, and
2. tracking an a-priori known trajectory on the surface.

### 3 Kinematic Controller Design

We first close the control loop around the kinematic component of (1), namely

$$\dot{q} = u. \quad (4)$$

The process of designing a control law  $u = h(q)$  to render (4) asymptotically stable at the origin is completed in two stages, in which the system is in two distinct modes of operation. When the first mode, mode  $\mathcal{M}_1$ , is active the controller drives the end-effector towards the surface, while when the second mode is active, mode  $\mathcal{M}_2$ , the controller either steers the robot towards a point on the surface, or alternatively, when the goal point is not stationary, the controller steers the robot to track a reference trajectory on the surface. The current position of the robot inside the workspace determines which controller mode is used. Thus, the workspace is decomposed into distinct parts associated with the different controllers. The following section describes this decomposition.

### 3.1 Workspace Decomposition

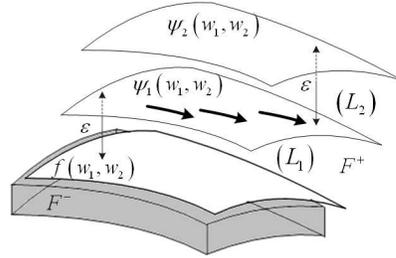
We utilize the concept of belt zones [15] (Fig. 3) as the basis for the decomposition of the workspace. We use a belt zone, which is a region close to the boundary of the surface to  $\partial f$ , further divided into an internal belt,  $\mathcal{L}_1$ , and an external belt,  $\mathcal{L}_2$ , while we fix the width of both the internal and external belt regions.

We define the bijective vector valued functions [15], which describes the belt zones as

$$\psi_1(w_1, w_2) = f(w_1, w_2) + \epsilon \cdot K, \quad (5)$$

$$\psi_2(w_1, w_2) = \psi_1(w_1, 1_2) + \epsilon \cdot K, \quad (6)$$

with  $0 < 2 \cdot \epsilon < r_m$  (a bounded constant in  $(0, r_m)$  [17]), with  $f$  and  $K$  given by (2) and (3), respectively. Most tasks dealing with surface processing usually require stabilization of the end-effector on the surface  $\psi_1(w_1, w_2)$ , defined by (5).



**Fig. 3** We decompose the workspace around the surface into a region (belt zone) where the transition (switch) from one mode to the other occurs. The external belt  $\mathcal{L}_2$  is defined from the vector functions  $\psi_2(w_1, w_2)$  and  $\psi_1(w_1, w_2)$ , and the internal belt  $\mathcal{L}_1$  from the vector functions  $\psi_1(w_1, w_2)$  and  $f(w_1, w_2)$ .

The internal and external belts are defined as (Fig. 3)

$$\mathcal{L}_1 = \{q : k(q) = (1 - \lambda) \cdot \psi_1 + \lambda \cdot f, \lambda \in [0, 1]\},$$

$$\mathcal{L}_2 = \{q : k(q) = (1 - \lambda) \cdot \psi_1 + \lambda \cdot \psi_2, \lambda \in (0, 1]\}.$$

Since functions  $f, \psi_1, \psi_2$  are bijective [17], [16], for every  $k(q) \in \mathcal{L}_1 \cup \mathcal{L}_2$  there is a unique couple  $(w_1, w_2)$ .

### 3.2 The Navigation Function

The controller that we design is based on an artificial potential field, created by a navigation function [34]. We construct two different navigation functions, one for mode  $\mathcal{M}_1$  and one for mode  $\mathcal{M}_2$ . The first navigation function, the one used in mode  $\mathcal{M}_1$ , is constructed to steer the end-effector into the interior the belt zone. This triggers a switch of the controller to mode  $\mathcal{M}_2$ , where the second part of the controller, steering the end-effector across to the surface. Across surface movement involves either stabilization towards a goal point on the surface, or tracking a reference trajectory on the surface.

The 3-dimensional manipulator inside the workspace is represented using a single point moving in a suitable transformed workspace, utilizing a series of successive transformations [24], and the navigation function incorporates the representation of the environmental obstacles. The variable  $\xi = q - q_d$  is used to construct these navigation functions. This variable represents the error between the joint configuration of the manipulator,  $q \in \mathbb{R}^m$ , and the desired joint configuration,  $q_d \in \mathbb{R}^m$ .

The navigation function used in mode  $\mathcal{M}_1$  is defined as [15]

$$\varphi_{\mathcal{M}_1}(\xi) = \frac{\gamma_d(\xi)}{(\gamma_d^\kappa(\xi) + \beta_{ws}(\xi) \cdot \beta_{\mathcal{O}}(\xi) \cdot \beta_s(\xi))^{\frac{1}{\kappa}}}, \quad (7)$$

where  $\gamma_d(\xi)$  is the “distance to goal” function  $\gamma_d(\xi) = \gamma_{\mathcal{M}_1}(\xi) = \|\xi\|^2$  and  $\beta_{ws}(\xi)$  provides the workspace potential, which depends on the mode in which the system operates. For mode  $\mathcal{M}_1$ ,  $\beta_{ws}(\xi) = \beta_{\mathcal{M}_1}(\xi) = -\|\xi - q_0\|^2 + r_0^2$ , with  $q_0 \in \mathbb{R}^m$  is the joint configuration at the center of the workspace (e.g., the center of the smallest ball containing  $\mathcal{F}$ ), and  $r_0 \in \mathbb{R}$  is the workspace’s radius (e.g., the radius of the smallest ball containing  $\mathcal{F}$ ). If we denote  $\mathcal{F}^*$  the workspace in which the robot and obstacles are represented by points through a series of transformations [15], we can account for the volume occupied by the manipulator, by using the function  $\beta_{\mathcal{O}}(\xi) \triangleq \prod_{j \in \mathcal{J}} \prod_{i \in \mathcal{I}} \|h_{R_j}^* - h_{\mathcal{O}_i}^*\|^2$  [15], where  $h_{R_j}^*$  is the position of the transformed robot part  $j$  in  $\mathcal{F}^*$ , and  $h_{\mathcal{O}_i}^*$  the position of the transformed obstacle  $i$  in  $\mathcal{F}^*$ .

Function  $\beta_s(\xi) \triangleq \prod_{j \in \mathcal{J}} \prod_{k \in \mathcal{K}} \|h_{R_j}^* - h_{s_k}^*\|^2$  expresses virtual obstacles representing kinematic singularities, and which are also transformed into points according to [24]. Finally,  $\kappa$  is a positive real tuning parameter.

Although in theory a system that flows along a vector field on the tangent space of the 2-D surface remains on this tangent space, sensor noise, model uncertainties and numerical diffusion cause it to deviate from the surface. We thus modify the navigation function that we define for mode  $\mathcal{M}_1$ , so that our potential field includes a component perpendicular to, and pointing toward, the 2 –  $D$  surface. We make this 2 –  $D$  surface attractive by adjusting function  $\gamma_d$  and introducing an additional obstacle function that prohibits the system from exiting the belt zone.

Let  $\pi(q)$  be the distance from surface  $f(w_1, w_2)$  when the end-effector is inside the belt zones. When  $\pi_0 = 0$  the end-effector is on the surface defined by  $f$  (the boundary of internal region); when  $\pi_{ext} = 2 \cdot \epsilon$  the end-effector is on the surface defined by  $\psi_2$  (the boundary of external region). Let us set the desired distance from surface  $f(w_1, w_2)$  to  $\pi_d = \epsilon$ ; there, the end-effector is on surface  $\psi_2(w_1, w_2)$ . Now  $\gamma_d$  is adjusted to

$$\gamma_d(\xi) = \gamma_{\mathcal{M}_2}(\xi) = \left\| \begin{bmatrix} q \\ \pi \end{bmatrix} - \begin{bmatrix} q_d \\ \pi_d \end{bmatrix} \right\|^2,$$

and the second component of the vector inside the norm serves to attract the end-effector to surface  $\psi_1$ . An additional obstacle function is defined as

$$\beta_{ws}(\xi) = \beta_{\mathcal{M}_2}(\xi) = \frac{(\pi_{ext} - \pi_d)^2 - (\pi(q) - \pi_d)^2}{(\pi_{ext} - \pi_d)^2},$$

and its role is to guarantee that the robot’s end-effector does not leave the belt zone. The navigation function generated in this process is  $\varphi_{\mathcal{M}_2}$ , and it can be time-varying

depending on whether the surface task is point stabilization or trajectory tracking. Function  $\varphi_{\mathcal{M}_2}$  is analytically expressed as

$$\varphi_{\mathcal{M}_2}(\xi, t) = \begin{cases} \frac{\gamma_{\mathcal{M}_2}(\xi)}{\left[\gamma_{\mathcal{M}_2}^{\kappa}(\xi) + \beta_{\mathcal{M}_2}(\xi) \cdot \beta_{\mathcal{O}}(\xi) \cdot \beta_s(\xi)\right]^{1/\kappa}}, & \text{for point stabilization} \\ \frac{\gamma_{\mathcal{M}_2}^{\kappa+1}(\xi, q_d(t))}{\left[\gamma_{\mathcal{M}_2}^{\kappa}(\xi, q_d(t)) + \beta_{\mathcal{M}_2}(\xi, q_d(t)) \cdot \beta_{\mathcal{O}}(\xi, q_d(t)) \cdot \beta_s(\xi, q_d(t))\right]^{1/\kappa}}, & \text{for trajectory tracking} \end{cases} \quad (8)$$

### 3.3 Kinematic Controller Synthesis

The system is defined to be in mode  $\mathcal{M}_1$  when  $p \in \mathcal{F}_{ext}$ , where  $\mathcal{F}_{ext} = \{\mathcal{F}_{free} \cap F^+\} \setminus \{\mathcal{L}_1 \cup \mathcal{L}_2\}$ , with  $p = k(q)$  computed using the manipulator's direct kinematics (that gives the robot end-effector's position and orientation as a function of the robot's configuration), [30]. If on the other hand  $p \in \mathcal{L}_1 \cup \mathcal{L}_2$ , then the system is said to be in mode  $\mathcal{M}_2$ . The mode of operation for the system is thus defined based on the set its operational space coordinates are in, and the transition from  $\mathcal{M}_1$  to  $\mathcal{M}_2$  is state-dependent.

Let  $q(0) \in \mathbb{R}^m$  be the robot's initial joint configuration, with  $p(0) = k(q(0)) \in F^+$ . The following vector fields represent desired motion directions for the system in modes  $\mathcal{M}_1$  and  $\mathcal{M}_2$ :

$$h_{\mathcal{M}_1}(\xi) \triangleq -\eta_1 \cdot \nabla \varphi_{\mathcal{M}_1}(\xi),$$

$$f_{\mathcal{M}_2}(\xi) \triangleq \begin{cases} -\eta_2 \cdot \nabla \varphi_{\mathcal{M}_2}(\xi), & \text{point stabilization} \\ -\eta_3 \cdot \nabla \varphi_{\mathcal{M}_2}(\xi, t) - \frac{1}{m} \cdot \begin{bmatrix} \left|\frac{\partial \varphi_{\mathcal{M}_2}}{\partial \xi_1}\right|^{-1} \\ \left|\frac{\partial \varphi_{\mathcal{M}_2}}{\partial \xi_2}\right|^{-1} \\ \dots \\ \left|\frac{\partial \varphi_{\mathcal{M}_2}}{\partial \xi_m}\right|^{-1} \end{bmatrix} \cdot \frac{\partial \varphi_{\mathcal{M}_2}}{\partial t}, & \text{trajectory tracking} \end{cases}, \quad (9)$$

where  $\eta_1 > 0$ ,  $\eta_2 > 0$ , and  $\eta_3 > 0$  are constant parameters.

We construct a controller to enable the robot's joint rates to converge to the reference vector field defined in (9). To enforce the input constraints imposed on the system the controller output is shaped through a non-smooth saturation function

$$h^{new} = \begin{cases} \text{sat}_{u_{\max}}(h_i) & , i \in \{\mathcal{M}_1, \mathcal{M}_2\} \\ \text{sat}_{\mu}(h_{\mathcal{M}_2}) & , \text{during trajectory tracking} \end{cases}, \quad (10)$$

with

$$\text{sat}_{\delta}(x) \triangleq \begin{cases} x & \text{if } |x| \leq \delta \\ -\delta & \text{if } x < -\delta, \\ \delta & \text{if } x > \delta \end{cases}$$

where  $\delta$  is a constant,  $u_{\max}$  is the vector of maximum joint velocity values, and  $\mu = u_{\max} - \dot{q}_d(t)$ , is chosen so that velocity input constraints are respected.

*Remark 1* In the case where  $|h^{new}| = |\mu| = u_{\max} - |\dot{q}_d(t)|$ , we have that  $|\dot{\xi}| = |\mu| \Rightarrow |\dot{q}| - |\dot{q}_d| \leq u_{\max} - |\dot{q}_d| \Rightarrow |\dot{q}| \leq u_{\max}$ , which ensure that we can guarantee the input constraints.

## 4 Stability Analysis

### 4.1 Outside the Belt Zone (Mode $\mathcal{M}_1$ )

We can guarantee that when the kinematic system (4) is in mode  $\mathcal{M}_1$  and steered along (10), it is asymptotically stable:

**Proposition 1** *Consider the system  $\dot{\xi} = v$ , where  $\xi = q - q_d$  and  $v = u - \dot{q}_d$ , with  $u$  the control law of (4). This system under the control law  $v = h^{\text{new}}(\xi) = \text{sat}_{u_{\max}}(h_{\mathcal{M}_1})$ , with  $h_{\mathcal{M}_1}$  as is defined in (9), is globally asymptotically stable, almost everywhere (a.e.)<sup>1</sup>.*

*Proof* We use the navigation function  $V(\xi) \triangleq \varphi_{\mathcal{M}_1}(\xi)$  as a Lyapunov function candidate. Function  $V$  is a regular function [35], since it is smooth as evident from (7).

Its time derivative is evaluated as

$$\begin{aligned} \dot{V} &= \nabla V^T \cdot h^{\text{new}} = \sum_{|h_{\mathcal{M}_1}(\xi)_i| \leq u_{\max}} \left( -\eta_1 \cdot \left| \frac{\partial V}{\partial \xi_i} \right|^2 \right) + \\ &+ \sum_{|h_{\mathcal{M}_1}(\xi)_j| > u_{\max}} \left[ -\text{sign} \left( \frac{\partial V}{\partial \xi_j} \right) \cdot \frac{\partial V}{\partial \xi_j} \cdot u_{\max} \right] \\ &= - \sum_{|h_{\mathcal{M}_1}(\xi)_i| \leq u_{\max}} \left( \eta_1 \cdot \left| \frac{\partial V}{\partial \xi_i} \right|^2 \right) - \sum_{|h_{\mathcal{M}_1}(\xi)_j| > u_{\max}} \left( \left| \frac{\partial V}{\partial \xi_j} \right| \cdot u_{\max} \right), \end{aligned}$$

where  $\eta_1 > 0$ . Thus,  $\dot{V}$  is strictly negative unless  $\nabla V = \nabla \varphi_{\mathcal{M}_1} = 0$ . Since  $\varphi_{\mathcal{M}_1}$  is a navigation function [34], the condition  $\nabla \varphi_{\mathcal{M}_1} = 0$  is true only at the destination configuration and a set of isolated saddle points. By construction [34], the region of attraction of the saddle points is a set of measure zero. Thus, the system converges to the destination configuration from almost everywhere.

### 4.2 Inside the Belt Zone (Mode $\mathcal{M}_2$ )

With  $p_0 \in F^+$  being the robot's end-effector initial configuration, and since  $p_d \in F^-$  by construction, the solutions of  $\dot{q} = u$  (which are absolutely continuous), can be shown using standard topological arguments [36] to intersect the surface  $\psi_2(w_1, w_2)$ . Therefore, there exists a finite time  $T$  at which the system enters the belt zones. When inside the belt zone, mode  $\mathcal{M}_2$  is activated. The robot's end-effector remains in the belt zone because its boundary is made repulsive by  $-\nabla \varphi_{\mathcal{M}_1}$ .

The transition from mode  $\mathcal{M}_1$  to mode  $\mathcal{M}_2$  happens just once, and then the system remains in mode  $\mathcal{M}_2$ , since the belt zone's workspace is positively invariant (the robot end-effector is trapped into the belt zones, since their boundaries are repulsive).

#### 4.2.1 Stabilization on the Surface

We now show that when the system is in mode  $\mathcal{M}_2$  and steered along (10), it is also asymptotically stable:

<sup>1</sup> i.e., everywhere except for a set of initial conditions of measure zero.

**Proposition 2** *The system  $\dot{\xi} = v$  under the control law  $v = h^{\text{new}}(\xi) = \text{sat}_{u_{\max}}(h_{\mathcal{M}_2})$ , with  $h_{\mathcal{M}_2}$  as is defined in (9) for point stabilization, is globally asymptotically stable, a.e.*

*Proof* It is shown in a way similar to the proof of Proposition 1, using  $V(\xi) \triangleq \varphi_{\mathcal{M}_2}(\xi)$  as a Lyapunov function candidate.

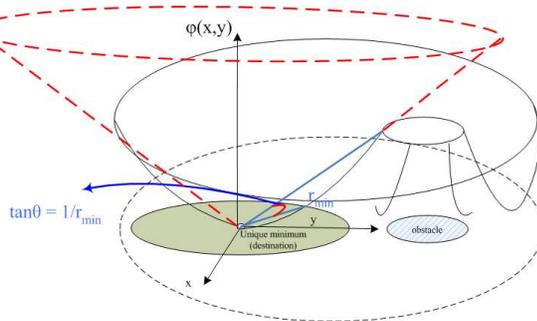
#### 4.2.2 Tracking on the Surface

We can show similar stability result for the task of trajectory tracking.

**Proposition 3** *The solutions of system  $\dot{\xi} = v$  under the control law  $v = h^{\text{new}}(\xi, t) = \text{sat}_{\mu}(h_{\mathcal{M}_2})$ , with  $h_{\mathcal{M}_2}$  as is defined in (9) for the case of trajectory tracking, are globally uniformly ultimately bounded, with a bound that can be made arbitrarily small with the suitable choice of  $\eta_3$ .*

*Proof* Using the same notation as in the previous proofs, we define a time-varying, continuously differentiable Lyapunov function candidate  $V(\xi, t) \triangleq \varphi_{\mathcal{M}_2}(\xi, t)$  (see (8)).

Inheriting the properties of navigation functions, function  $V(\xi, t)$  is positive definite for all  $t > 0$ , and there is therefore a class  $\mathcal{K}$  function  $V_1(\|\xi\|)$  which satisfy  $0 < V_1(\|\xi\|) \leq V(\xi, t)$ . In addition, there exists a sphere centered at the destination point  $q = q_d(t)$  (i.e.,  $\xi = 0$ ), with radius  $r(t)$ , which is tangent to, but not intersecting with, an obstacle in the workspace. For a constant  $r_{\min} \triangleq \inf_{t>0} r(t)$ , we can define a cone  $V_2(\|\xi\|) = c_1 \cdot \|\xi\|$  with  $c_1 = \frac{1}{r_{\min}}$ , such that  $V(\xi, t) \leq V_2(\|\xi\|)$  (Figure 4). Thus,  $V_1(\|\xi\|) \leq V(\xi, t) \leq V_2(\|\xi\|)$ .



**Fig. 4** A navigation function over a free configuration space. It is easy to find the minimum radius  $r_{\min}$  (radius of a tangent to an obstacle sphere, centered at the destination point), which guarantee that the destination point is collision free.

Let  $\nu_1$  be the number of components of  $h_{\mathcal{M}_2}$  that are not saturated, and  $\nu_2 = m - \nu_1$  the ones that are. Then,

$$\begin{aligned}\dot{V} &= \frac{\partial V}{\partial t} + \nabla V^T \cdot h^{\text{new}} = \\ &= \frac{\partial V}{\partial t} + \sum_{l_1=1}^{\nu_1} \left( -\eta_3 \cdot \left| \frac{\partial V}{\partial \xi_{l_1}} \right|^2 - \frac{1}{m} \cdot \frac{\partial V}{\partial t} \right) + \sum_{l_2=1}^{\nu_2} \left[ -\text{sign} \left( \frac{\partial V}{\partial \xi_{l_2}} \right) \cdot \frac{\partial V}{\partial \xi_{l_2}} \cdot \mu \right] \\ &= \frac{\partial V}{\partial t} - \sum_{l_1=1}^{\nu_1} \left( \eta_3 \cdot \left| \frac{\partial V}{\partial z_{l_1}} \right|^2 + \frac{1}{m} \cdot \frac{\partial V}{\partial t} \right) - \sum_{l_2=1}^{\nu_2} \left( \left| \frac{\partial V}{\partial \xi_{l_2}} \right| \cdot \mu \right), \quad (11)\end{aligned}$$

since it holds that  $\text{sign} \left( h_{\mathcal{M}_2}^{\xi_{l_2}} \right) = -\text{sign} \left( \frac{\partial V}{\partial \xi_{l_2}} \right)$ , with an appropriate choice of  $\eta_3$  (see Remark 2).

The isolated term in the above equation can be written in the following form  $\frac{\partial V}{\partial t} = -\frac{\gamma_d^{\kappa+1}}{\kappa \cdot (\gamma_d^\kappa + B)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial B}{\partial t} + \frac{(\kappa+1) \cdot \gamma_d^\kappa}{(\gamma_d^\kappa + B)^{\frac{1}{\kappa}}} \cdot \frac{\partial \gamma_d}{\partial t} - \frac{\gamma_d^{2 \cdot \kappa}}{(\gamma_d^\kappa + B)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial \gamma_d}{\partial t}$ , where  $B = (\beta_{\mathcal{M}_2} \cdot \beta_{\mathcal{O}} \cdot \beta_s)$ . Thus, it can be bounded as follows:

$$\begin{aligned}\left| \frac{\partial V}{\partial t} \right| &< \frac{1}{\kappa} \cdot \gamma_d^{\kappa+1} \cdot \frac{1}{(\gamma_d^\kappa)^{\frac{\kappa+1}{\kappa}}} \cdot \left| \frac{\partial B}{\partial t} \right| + \frac{(\kappa+1) \cdot \gamma_d^\kappa}{(\gamma_d^\kappa)^{\frac{1}{\kappa}}} \cdot \left| \frac{\partial \gamma_d}{\partial t} \right| + \frac{\gamma_d^{2 \cdot \kappa}}{(\gamma_d^\kappa)^{\frac{1+\kappa}{\kappa}}} \cdot \left| \frac{\partial \gamma_d}{\partial t} \right| = \\ &= \frac{1}{\kappa} \left| \frac{\partial B}{\partial t} \right| + (\kappa + 2) \cdot \gamma_d^{\kappa-1} \cdot \left| \frac{\partial \gamma_d}{\partial t} \right|.\end{aligned}$$

The properties of the navigation function ensure boundedness of its gradient within the workspace. This implies that there exists a positive bound  $\Delta = \left( \max_{\mathcal{F}/\mathcal{M}} \left\| \frac{\partial V}{\partial \xi_{l_2}} \right\|_{\infty} \right)^{-1}$ , where  $\mathcal{M}$  is a set of measure zero, including the unstable saddle points of the navigation function as well as the destination configuration. We can then write  $\Delta \cdot \left| \frac{\partial V}{\partial \xi_{l_2}} \right| < \mu$ , for all  $l_2$ , and bound the last term in (11) as

$$- \sum_{l_2=1}^{\nu_2} \left( \left| \frac{\partial V}{\partial \xi_{l_2}} \right| \cdot \mu \right) < - \sum_{l_2=1}^{\nu_2} \left( \left| \frac{\partial V}{\partial \xi_{l_2}} \right|^2 \cdot \Delta \right).$$

Equation (11) yields

$$\dot{V} \leq \left( 1 - \frac{\nu_1}{m} \right) \cdot \frac{\partial V}{\partial t} - \Delta \cdot \|\nabla V\|^2 - \sum_{l_1=1}^{\nu_1} \left( \eta_4 \cdot \left| \frac{\partial V}{\partial \xi_{l_1}} \right|^2 \right),$$

where the control gain  $\eta_3$  can be then decomposed in the form  $\eta_3 = \Delta + \eta_4$ , with  $\eta_4 > 0$ . Then,

$$\dot{V} \leq \left( 1 - \frac{\nu_1}{m} \right) \cdot D - \Delta \cdot \|\nabla V\|^2,$$

where  $D = \frac{1}{\kappa} \sup_{\mathcal{F}} \left| \frac{\partial B}{\partial t} \right| + (\kappa + 2) \cdot \max \left( \gamma_d^{\kappa-1} \right) \cdot \sup_{\mathcal{F}} \left| \frac{\partial \gamma_d}{\partial t} \right|$ , in which  $\sup_{\mathcal{F}} \left| \frac{\partial B}{\partial t} \right|$ , and  $\sup_{\mathcal{F}} \left| \frac{\partial \gamma_d}{\partial t} \right|$  depends on  $\sup_{\mathcal{F}} \|\dot{q}_d\|$ . In the region where

$$\|\nabla V\| > \sqrt{\frac{\left( 1 - \frac{\nu_1}{m} \right) \cdot D}{\Delta}}, \quad (12)$$

the Lyapunov function decreases and therefore  $\xi$  converges to 0. This limit point corresponds to the destination configuration  $q_d(t)$ . Thus,  $\xi$  is uniformly ultimately bounded in the region where (12) holds.

In the neighborhood of  $\xi = 0$ ,  $\nabla V$  does not vanish except for  $\xi = 0$ , since  $V$  is a navigation function of  $\xi$ . Thus,  $\|\nabla V\|$  is a positive definite scalar function. This implies that there exist  $W_4(\|\xi\|)$ ,  $W_5(\|\xi\|)$  class  $\mathcal{K}$  functions for which  $W_4(\|\xi\|) \leq \|\nabla V\| \leq W_5(\|\xi\|)$ . Using the lower bounding function  $W_4$ , if

$$\|\xi\| \geq W_4(\|\xi\|)^{-1} \cdot \sqrt{\frac{(1 - \frac{\nu_1}{m}) \cdot D}{\Delta}} \in \mathcal{K}, \quad (13)$$

then, for the gradient of  $V$  we can write

$$\|\nabla V\| \geq W_4(\|\xi\|) \geq \sqrt{\frac{(1 - \frac{\nu_1}{m}) \cdot D}{\Delta}},$$

implying that  $\dot{V}$  is strictly negative in the region defined by (13).

Application of Theorem 3 [36] (see Appendix), completes the proof by ensuring that  $\xi$  is globally uniformly ultimately bounded.

*Remark 2* It is needed to prove that under a suitable choice of  $\eta_3$ , it holds

$$\text{sign} \left( h_{\mathcal{M}_2}^{\xi_{l_2}} \right) = -\text{sign} \left( \frac{\partial V}{\partial \xi_{l_2}} \right). \quad (14)$$

If it is true that  $\eta_3 \cdot \left| \frac{\partial V}{\partial \xi_{l_2}} \right| > \frac{1}{m} \cdot \left| \frac{\partial V}{\partial \xi_{l_2}} \right|^{-1} \cdot \left| \frac{\partial V}{\partial t} \right|$ , then (14) is true. Since,  $\eta_3 = \Delta + \eta_4$ , it is possible to find now a parameter  $\eta_4 > \frac{1}{m} \cdot \max \left( \left| \frac{\partial V}{\partial t} \right| \cdot \left| \frac{\partial V}{\partial \eta_2} \right|^{-2} \right) > 0$ . The case where  $\left| \frac{\partial V}{\partial \eta_2} \right| = 0$ , is treated as a semi-singularity point which has to be avoided, and it is incorporated into the function  $\beta_s$  as a component of the total obstacle function  $B$  of (7).

*Remark 3* There are two ways in which we can turn uniform ultimate boundedness to uniform asymptotic stability in Proposition 3. When  $\dot{q}_d \rightarrow 0$ , the region defined by (13) reduces to a ball of radius zero, because as  $\dot{q}_d \rightarrow 0$ , we have  $\frac{\partial B}{\partial t} \rightarrow 0$ , and  $\frac{\partial \gamma_d}{\partial t} \rightarrow 0$ . Another way is to ensure that no component of  $h_{\mathcal{M}_2}$  is saturated; then  $\nu_2 = 0 \Rightarrow \nu_1 = m$ , and the right hand side of (13) vanishes.

## 5 Dynamic Controller Design

We now design a control law that renders (1), asymptotically stable for the case of trajectory tracking. Convergence to belt zone, and point stabilization inside the belt zone, can be treated as special cases where the controller (and therefore the closed loop system) is time invariant.

In the sequel, we adopt the notation  $\dot{(\cdot)}$  of Theorem 1 [37] (see Appendix) as applied to vector-valued functions. In this case,  $\dot{x}$ , where  $x \in \mathbb{R}^n$ , is the stack vector of the generalized time derivatives of the components of  $x$ .

**Proposition 4** Consider the system  $\ddot{\xi}(t) = U(t)$ , where  $\xi(t) = q - q_d(t)$ , and  $U = y - \ddot{q}_d(t)$ , with  $y$  the control law of (1). This system becomes semi-globally uniformly asymptotically stable to zero almost everywhere, under the control law

$$U = \theta_1 + \left[ \theta_2 + \text{diag} \left\{ \frac{\frac{\partial V}{\partial \xi} \cdot (\dot{\xi} - h^{\text{new}})}{\|\dot{\xi} - h^{\text{new}}\|^2} \right\} \right] \cdot (h^{\text{new}} - \dot{\xi}) \quad (15)$$

where  $h^{\text{new}}$  as is defined in (10),  $V : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}$  is a regular locally Lipschitz Lyapunov function,  $\theta_1 = \arg \min_{\zeta \in \dot{h}^{\text{new}}} \zeta^T (\dot{\xi} - h^{\text{new}})$  is the vector in  $\dot{h}^{\text{new}}$  yielding the smallest inner product with  $\dot{\xi} - h^{\text{new}}$ , and  $\theta_2$  is a positive definite constant matrix.

*Proof* The control law construction and the proof structure are inspired by the backstepping controller design proposed by [38].

Let  $\xi(\cdot, t)$  be a Filippov solution of  $\dot{\xi} = h(\xi, t)$ . We form the Lyapunov function candidate:

$$V_1(\xi, t) \triangleq V(\xi, t) + \frac{1}{2} \cdot (\dot{\xi} - h^{\text{new}}(\xi, t))^2, \quad (16)$$

where  $V(\xi) = \varphi_i(\xi)$  is regular (see (7) – (8) with  $i \in \{\mathcal{M}_1, \mathcal{M}_2\}$ ). Taking the time derivative of (16)

$$\dot{V}_1 = \left( \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \xi} \cdot \dot{\xi} \right) + (U - \dot{h}^{\text{new}})^T \cdot (\dot{\xi} - h^{\text{new}}),$$

and substituting  $U$  as defined in (15),

$$\dot{V}_1 = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \xi} \cdot h^{\text{new}} + (\theta_1 - \dot{h}^{\text{new}})^T \cdot (\dot{\xi} - h^{\text{new}}) - (\dot{\xi} - h^{\text{new}})^T \cdot \theta_2 \cdot (\dot{\xi} - h^{\text{new}}) \quad (17)$$

As we saw in the proof of Proposition 3,

$$\begin{aligned} \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \xi} \cdot h^{\text{new}} &= \frac{\partial V}{\partial t} - \sum_{l_1=1}^{\nu_1} \left( \eta_3 \cdot \left| \frac{\partial V}{\partial \xi_{l_1}} \right|^2 + \frac{1}{m} \cdot \frac{\partial V}{\partial t} \right) - \sum_{l_2=1}^{\nu_2} \left( \left| \frac{\partial V}{\partial \xi_{l_2}} \right| \cdot \mu \right) \leq \\ &\leq \left( 1 - \frac{\nu_1}{m} \right) \cdot D - \Delta \cdot \|\nabla V\|^2. \end{aligned}$$

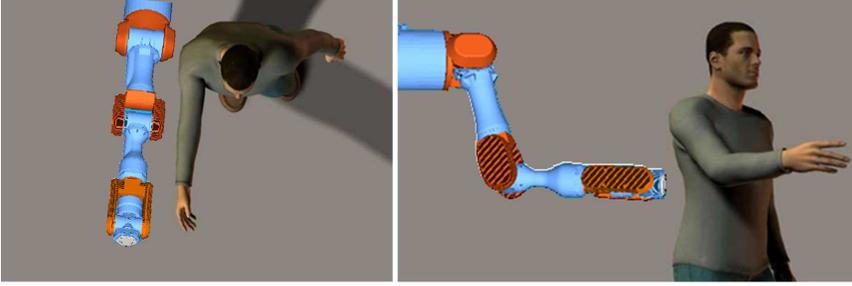
Thus, the first plus the second term in (17) is negative definite. Taking into account how  $\theta_1$  has been defined, we conclude that  $\dot{V}_1$  is negative definite.

*Remark 4* For the dynamic model, it holds that away from the surface of discontinuity the system dynamics reduce to a stable linear system, and the associated vector fields, point to the same side of the discontinuity surface. Thus, the switching will occur, and of course the solutions are unique.

The backstepping theorem guarantees that stability is maintained on switching, and it is not dependent on how many times the switching occurs. Even if  $V$  jumps on switches (take the values  $V_{\mathcal{M}_1}$  or  $V_{\mathcal{M}_2}$ ), the backstepping controller does not allow  $V_1$  to increase. Thus, it holds that  $V_1^{\mathcal{M}_2} - V_1^{\mathcal{M}_1} \leq -W$  (switch from mode  $\mathcal{M}_1$  to  $\mathcal{M}_2$ ) or  $V_1^{\mathcal{M}_1} - V_1^{\mathcal{M}_2} \leq -W$  (switch from mode  $\mathcal{M}_2$  to  $\mathcal{M}_1$ ). According to Theorem 2 (see Appendix), the switched system is globally asymptotically stable a.e., since the sufficient condition is satisfied.

## 6 Experimental Results

With an exoskeleton not being available, we use a Mitsubishi PA10-7C robotic manipulator with  $m = 7$  degrees of freedom (shown in Fig. 5) in order to test our controller. This type of manipulator has the same number of degrees of freedom with a human arm and the motion of each link is similar to that of a human's arm. In our approach, however, we test the free endpoint case, and thus only six degrees of freedom are being utilized. The identification of the robot's arm dynamic parameters is given in [32].



**Fig. 5** The controller design is tested on Mitsubishi PA10-7C robotic manipulator, with 7 d.o.f., to assess the performance it would have, if it were to be implemented on an exoskeleton. This robotic manipulator has the same number of d.o.f. with a human arm, and its motion is in analogy with the human's arm movement.

We assume a surface of interest,  $f(w_1, w_2)$ , is assumed to be an ellipsoid centered at  $(0, 0, 0)$  with semi-axes lengths  $(0.75, 0.25, 0.35)$  (Fig. 7).

The experimental scenario involves two 3-D (ellipsoid) obstacles centered at  $\mathcal{O}_1 : (-0.3, -0.4, 0.1)$  and  $\mathcal{O}_2 : (0.35, -0.3, -0.5)$  (Fig. 6) both having semi-axes lengths of  $(0.05, 0.10, 0.20)$  (referred to as  $x$ ,  $y$ , and  $z$  dimensions, respectively).

The obstacle regions on the surface  $f(w_1, w_2)$  are centered at

$$\mathcal{O}_{g1} : (-0.33, -0.08, 0.18), \mathcal{O}_{g2} : (0.33, -0.08, -0.18) \mathcal{O}_{g3} : (-0.33, -0.08, -0.18)$$

(Fig. 7). The manipulator's initial end-effector's configuration is given by a vector that contains three components of  $x$ ,  $y$ , and  $z$  coordinates, and the three components of the euler angles describing the end-effector orientation.

$$p_0 = [-0.61, -0.39, -0.13, 0.0, 0.0, 0.0]$$

The target configuration in the operational space is set at

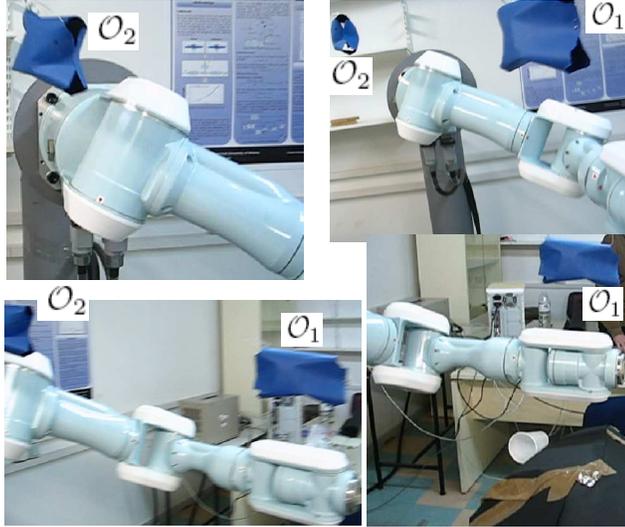
$$p_d = [0.49, -0.16, 0.13, 1.33, 0.87, -1.33]$$

The vector of joint velocity limitation in rad/sec, is

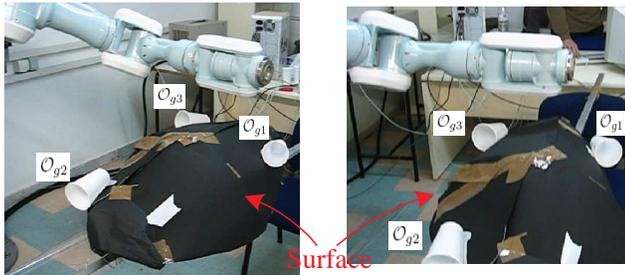
$$u_{\max} = [0.6 \ 0.6 \ 1.2 \ 1.2 \ 1.2 \ 1.2\pi \ 1.2\pi]^T$$

During the experiment, the first task for the robot is to move its end-effector to the desired configuration close to the surface, while each part of the robotic arm avoids collisions with obstacles. The next step for the robot is to move its end-effector in order

to track a predefined trajectory. To this end we set a sinusoidal known trajectory on the top of the surface, and now the goal is to make the end-effector track this trajectory, avoiding collisions with obstacles. During operation, we monitor joint rates to ensure that velocity constraints are respected.



**Fig. 6** Experimental Set-up. The blue objects at the right and left of the robotic manipulator, represent obstacles. The main goal of this part of experiment is to drive the manipulator's end-effector close to a surface, avoiding the obstacles.



**Fig. 7** Experimental Set-up. The robot manipulator in its final configuration. The black ellipsoid is the surface of interest. The plastic cups on the surface are obstacle regions, which have to be avoided. The goal of this part of experiment is to make the manipulator's end-effector to follow a predefined trajectory, and the same time to avoid all the surface's bad regions.

Figures 8 and 9 depict the experimental results, showing how the robot manipulator reaches the desired configuration and tracks the predefined trajectory, with simultaneous obstacle avoidance. Figures 10 and 11 present the cartesian and joint position. Fig. 12 depicts the joint velocity, and from the flat regions in these profiles it is evident

that the robot's kinematic constraints are respected. The time axis records time steps. Each time step is 2.5 milliseconds. Fig. 13 gives the error between the real cartesian position and the desired position during tracking, respectively, measured in meters. At the beginning, when the end-effector is away from the desired configuration, the error is 0.12 m, and the robot accelerates to get in position to track the reference trajectory. This error is then bounded below 1 cm.

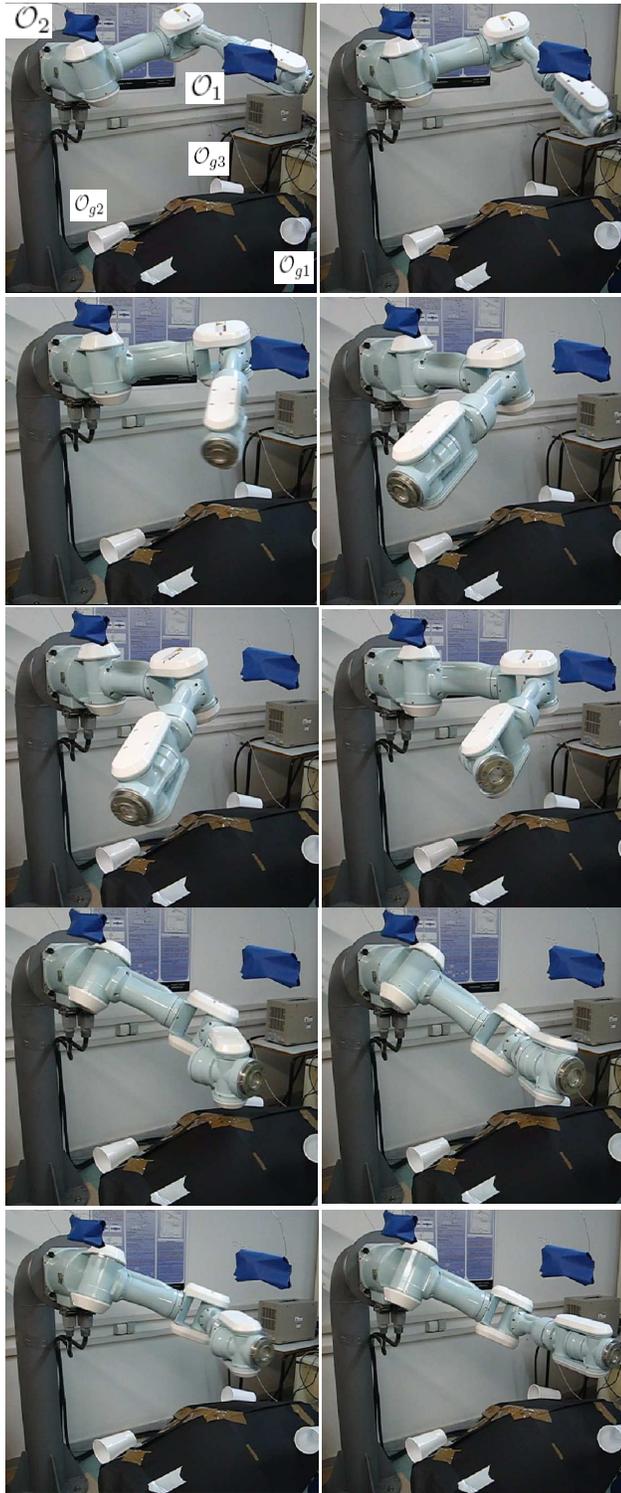
## 7 Conclusion and Future Work

We present a methodology for performing navigation and tracking tasks over a 2-dimensional manifold which is embedded in a 3-dimensional workspace. The method is applicable to articulated robotic manipulators with kinematic input constraints. After the manipulator is steered between obstacles to the 2-D surface of interest, its end-effector follows task specific vector fields that enable it to navigate or track a predefined trajectory along the surface. We theoretically guarantee global convergence of the system and the same time we have proved collision avoidance with the environmental obstacles.

Further research includes considering surface properties in the construction of the belt zone vector fields and implementing the methodology to real neuro-robotic systems taking into account their dynamics and kinematic constraints.

## References

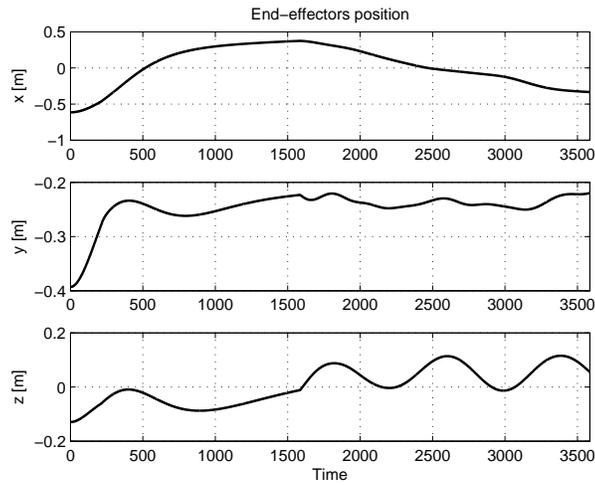
1. W. Wu, M. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. Donoghue, "Modeling and decoding motor cortical activity using a switching kalman filter," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 933–942, 2004.
2. W. Wu, Y. Gao, E. Bienenstock, J. Donoghue, and M. Black, "Bayesian population coding of motor cortical activity using a kalman filter," *Neural Computation*, vol. 18, pp. 80–118, 2005.
3. S. Micera, A. Sabatini, and P. Dario, "On automatic identification of upper-limb movements using small-sized training sets of emg signals," *Medical Engineering & Physics*, pp. 527–533, 2000.
4. X. Navarro, T. Krueger, N. Lago, S. Micera, P. Dario, and T. Stieglitz, "A critical review of interfaces with the peripheral nervous system for the control of neuroprostheses and hybrid bionic systems," *Journal of the Peripheral Nervous System*, vol. 10, pp. 229–58, 2005.
5. D. Conner, A. Greenfield, P. Atkar, A. Rizzi, and H. Choset, "Paint deposition modeling for trajectory planning on automotive surfaces," *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 4, 2005.
6. P. Atkar, D. Conner, A. Greenfield, H. Choset, and A. Rizzi, "Uniform coverage of simple surfaces embedded in  $\mathbb{R}^3$  for auto-body painting." Carnegie Mellon University, 2004.
7. P. Atkar, H. Choset, and A. Rizzi, "Towards optimal coverage of curve," *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2003.
8. D. Conner, P. Atkar, A. Rizzi, and H. Choset, "Deposition modeling for paint application on surfaces embedded in  $\mathbb{R}^3$ ," Carnegie Mellon University, Tech. Report, 2002.
9. M. Garber and M. C. Lin, "Constraint-based motion planning for virtual prototyping," *Proceedings of the 5th International Workshop on Algorithmic Foundations of Robotics (WAFR)*.
10. —, "Constraint-based motion planning using voronoi diagrams," *Springer Tracts in Advanced Robotics, Algorithmic Foundations of Robotics V*, vol. 7, pp. 541–558, 2003.
11. M. Vincze, A. Pichler, G. Biegelbauer, K. Hausler, H. Andersen, O. Madsen, and M. Kristiansen, "Automatic robotic spray painting of low volume high variant parts," *Proceedings of the 33rd ISR (International Symposium on Robotics)*, 2002.



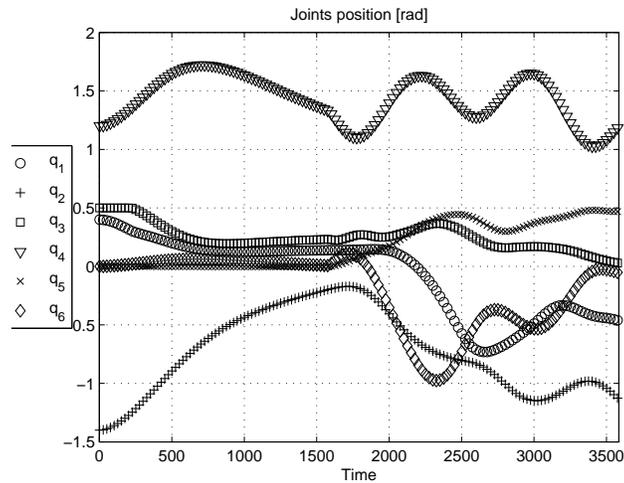
**Fig. 8** Experimental Set-up. Sequence of pictures that shows the motion of robot manipulator during the experiment. The robot manipulator is moving towards the (black) surface, avoiding blue obstacles. The end-effector reaches its destination configuration (second right picture). Then the trajectory tracking on the surface takes place, during which the white cups on the surface are being avoided.



**Fig. 9** Experimental Set-up. Top view of experimental. The robot manipulator is moving towards the (black) surface, avoiding blue obstacles. The end-effector reaches its destination configuration (second right picture). The white cups on the surface are being avoided during the tracking phase, that follows the stabilization to the destination configuration.

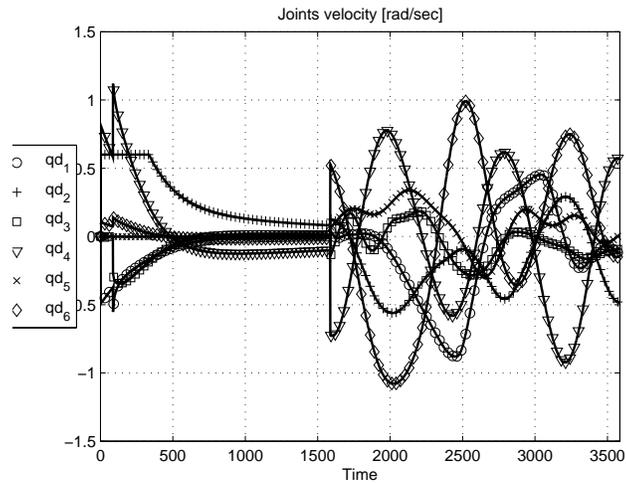


**Fig. 10** Experimental Results. Cartesian position of the robotic manipulator's end-effector, during the experiment. The designed controller drive the manipulator's end-effector close to the surface in order to reach the destination point (until time step 1500). Immediately after, the end effector traks a sinusoidal path, shown in the z plot. Each time step is 2.5 msec.

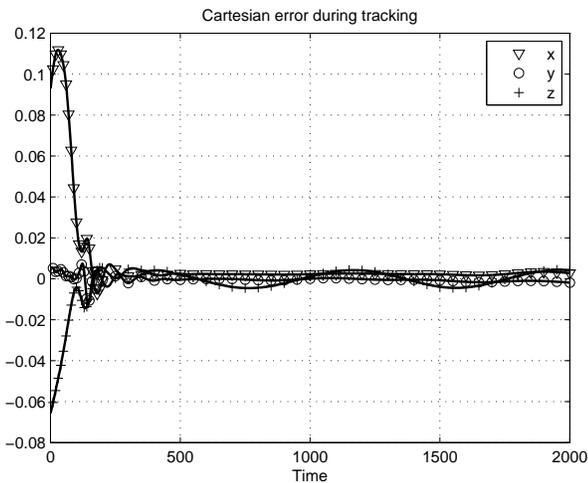


**Fig. 11** Experimental Results. Robotic manipulator's angles of joints, during the experiment, similar procedure with Figure 10. Circle -  $q_1$ , plus -  $q_2$ , square -  $q_3$ , triangle -  $q_4$ , x-mark-  $q_5$ , diamond -  $q_6$ . The time axis represents time steps. Each time step is 2.5 msec.

12. H. Seraji and B. Bon, "Real-time collision avoidance for position-controlled manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, 1999.
13. R. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: Theory and experiments," *Journal of Robotic Systems*, vol. 22, no. 12, pp. 737-757, 2005.
14. Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE Transactions on Systems, Man, & Cybernetics - Part B: Cybernetics*, vol. 34, no. 1, 2004.



**Fig. 12** Experimental Results. Robotic manipulator's velocities of joints. The velocity values of all joints are within the limits, as they defined for this robotic manipulator, at the beginning of this section. Circle -  $q_1$ , plus -  $q_2$ , square -  $q_3$ , triangle -  $q_4$ , x-mark-  $q_5$ , diamond -  $q_6$ . The time axis represents time steps. Each time step is 2.5 msec.



**Fig. 13** Experimental Results. Cartesian position error between the real position of the robot end-effector, and the predefined trajectory during trajectory tracking. Triangle -  $x$ , circle -  $y$ , plus -  $z$ . It is shown that this error is bounded, confirming our theoretical prediction regarding uniform ultimately boundness. The time axis represents time steps. Each time step is 2.5 msec. The error is in m. At the beginning when the end-effector is away from the desired configuration the error is 0.12 m, and therefore the robot is tracking the reference trajectory with bounded error 1 cm.

15. X. Papageorgiou, S. Loizou, and K. Kyriakopoulos, "Motion tasks for robot manipulators on embedded 2-D manifolds," *Joint 2006 IEEE Conference on Control Applications (CCA), 2006 IEEE Computer Aided Control Systems Design Symposium (CACSD) & 2006 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 3047–3052, 2006.
16. S. Loizou, H. Tanner, V. Kumar, and K. Kyriakopoulos, "Closed loop motion planning and control for mobile robots in uncertain environments," *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.
17. X. Papageorgiou, S. Loizou, and K. Kyriakopoulos, "Motion planning and trajectory tracking on 2-D manifolds embedded in 3-D workspaces," *2005 IEEE International Conference on Robotics and Automation*, pp. 501–506, 2005.
18. —, "Motion tasks and force control for robot manipulators on embedded 2-D manifolds," *2007 IEEE International Conference on Robotics and Automation*, pp. 4202–4207, 2007.
19. X. Papageorgiou and K. Kyriakopoulos, "Compliant motion tasks for constrained robot manipulators on embedded 2-D manifolds under input constraints," *47th IEEE Conference on Decision and Control*, pp. 4570–4575, 2008.
20. X. Papageorgiou, H. Tanner, and K. Kyriakopoulos, "Motion tasks for robot manipulators on embedded 2-D manifolds under input constraints," *2007 European Control Conference*, pp. 3783–3789, 2007.
21. X. Papageorgiou and K. Kyriakopoulos, "Motion tasks for robot manipulators subject to joint velocity constraints," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2139–2144, 2008.
22. A. Glattfelder and W. Schaufelberger, *Control Systems with Input and Output Constraints*. Springer-Verlag, 2003.
23. D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances Appl. Math.*, vol. 11, pp. 412–442, 1990.
24. H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.
25. D. Dimarogonas, M. Zavlanos, S. Loizou, and K. Kyriakopoulos, "Decentralized motion control of multiple holonomic agents under input constraints," *42nd IEEE Conference on Decision and Control*, 2003, submitted for Journal Publication.
26. S. Loizou and K. Kyriakopoulos, "Navigation of multiple input constraint micro-robotic agents," *2005 International Conference on Intelligent Robots and Systems*, 2005.
27. G. Antonelli, S. Chiaverini, and G. Fusco, "Real-time end-effector path following for robot manipulators subject to velocity, acceleration, and jerk joint limits," *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 452–457, 2001.
28. K. Wedeward and R. Colbaugh, "Adaptive controllers that employ saturated error update laws for robot trajectory tracking," *Proceedings of the 34th Conference on Decision & Control*, pp. 45–50, 1995.
29. M. Ahmad and J. Oman, "Application of proportional-integral sliding mode tracking controller to robot manipulators," *Proceedings of 2003 IEEE Conference on Control Application*, vol. 1, pp. 87–92, 2003.
30. L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.
31. C. Kennedy and J. Desai, "Modeling and control of the mitsubishi pa-10 robot arm harmonic drive system," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 3, 2005.
32. N. Bompos, P. Artemiadis, A. Oikonomopoulos, and K. Kyriakopoulos, "Modeling, full identification and control of the mitsubishi pa-10 robot arm," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2007.
33. W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 1986.
34. E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
35. F. Clarke, *Optimization and Nonsmooth Analysis*. Addison - Wesley, 1983.
36. H. Khalil, *Nonlinear Systems*. Prentice-Hall, 1996.
37. D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1910–1914, 1994.
38. H. Tanner and K. Kyriakopoulos, "Backstepping for nonsmooth systems," *Automatica*, vol. 39, pp. 1259–1265, 2003.
39. A. Filippov, *Differential equations with discontinuous right-hand sides*. Kluwer Academic Publishers, 1988.
40. D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.

## Appendix

**Definition 1** [39] A vector function  $x$  is called a solution of  $\dot{x} = f(x)$  if  $x$  is absolutely continuous and  $\dot{x} \in K[f](x)$  where

$$K[f](x) \triangleq \overline{\text{co}} \{ \lim f(\tilde{x}) \mid \tilde{x} \rightarrow x, \tilde{x} \notin \mathcal{N} \}$$

where  $\mathcal{N}$  is a set of measure zero.

**Theorem 1** [37] Let  $x(\cdot, t)$  be a Filippov solution of  $\dot{x} = f(x, t)$ , and  $V : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}$  be a Lipschitz and in addition, regular function, [35]. Then  $V(x, t)$  is absolutely continuous,  $\frac{d}{dt}V(x, t)$  exists almost everywhere, and

$$\frac{d}{dt}V(z, t) \in^{a.e.} \dot{V}(z, t)$$

where

$$\dot{V}(z, t) \triangleq \bigcap_{\xi \in \partial V(z, t)} \xi^T \cdot \begin{pmatrix} K[f](z, t) \\ 1 \end{pmatrix}$$

and  $\partial V$  is the Clarke's generalized gradient, [35].

**Theorem 2** [40] Let  $\dot{x} = f_v(x)$ , be a finite family of globally asymptotically stable systems, and let  $V_v, v \in \mathcal{P}$  be a family of corresponding radially unbounded Lyapunov functions, where  $\mathcal{P}$  is some index set. Suppose that there exists a family of positive definite continuous functions  $W_v, v \in \mathcal{P}$ , with the property that for every pair of switching times  $(t_i, t_j), i < j$  such that  $\sigma(t_i) = \sigma(t_j) = v \in \mathcal{P}$ , and  $\sigma(t_k) \neq v$ , for  $t_i < t_k < t_j$ , where  $\sigma$  is the switching signal, we have that

$$V_v(x(t_j)) - V_v(x(t_i)) \leq -W_v(x(t_i))$$

Then the switched system is globally asymptotically stable.

**Theorem 3** [36] Let  $\mathcal{D} \subset \mathbb{R}^n$  be a domain that contains the origin and  $V : [0, \infty) \times \mathcal{D} \rightarrow \mathbb{R}$  be a continuously differentiable function such that

$$\begin{aligned} W_1(x) &\leq V(t, x) \leq W_2(x) \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) &\leq -W_3(x), \quad \forall \|x\| \geq \lambda > 0 \end{aligned}$$

$\forall t \geq 0, \forall x \in \mathcal{D}$  where  $W_1(x)$  and  $W_2(x)$  are class  $\mathcal{K}$  functions and  $W_3(x)$  is a continuous positive definite functions. Take  $r > 0$  such that  $\mathcal{B}_r \subset \mathcal{D}$  and suppose that

$$\lambda < W_2^{-1}(W_1(r))$$

Then, there exist a class  $\mathcal{KL}$  function  $b$  for every initial state  $x(t_0)$ , satisfying  $\|x(t_0)\| \leq W_2^{-1}(W_1(r))$ , there is  $T \geq 0$  (dependent on  $x(t_0)$  and  $\lambda$ ) such that  $\forall x(t_0) \in \{x \in \mathcal{B}_r \mid W_2(x) \leq \rho\}$ , the solution of  $\dot{x} = f(t, x)$  satisfies

$$\begin{aligned} \|x(t)\| &\leq b(\|x(t_0)\|, t - t_0), \quad \forall t_0 \leq t \leq t_0 + T \\ \|x(t)\| &\leq W_1^{-1}(W_2(\lambda)), \quad \forall t \geq t_0 + T \end{aligned}$$

Moreover, if  $\mathcal{D} = \mathbb{R}^n$  and  $W_1$  belongs to class  $\mathcal{K}_\infty$ , then the last two conditions hold for any initial state  $x(t_0)$ , with no restriction on how large  $\lambda$  is.