Motion Tasks for Robot Manipulators on Embedded 2-D Manifolds under Input Constraints

Xanthi Papageorgiou, Herbert G. Tanner and Kostas J. Kyriakopoulos

***Abstract*— We present a methodology to steer the end effector of a robotic manipulator, which is constrained in terms of joint rates, on the surface within the workspace. We develop controllers for stabilizing the end effector to a point, and for tracking a trajectory on this surface, while respecting the input constraints. We show that the resulting closed loop system is uniformly asymptotically stable and we verify our analytical development with computer simulations.**

I. Introduction

Robotic applications where the manipulator is supposed to perform a task along a particular surface, such as robotic surface painting, surface cleaning, and surface inspection, pose challenging control design problems. Our motivation comes from the field of neuro-robotics, and specifically from an application where a robot executes a task through interfacing with the neural system (Fig. 1), thus by processing electromyographic activity. In most cases, neural signals are noisy and inappropriate for controlling a robot directly. The presense of obstacles in the environment, and consideration of non-planar surfaces complicates the problem further. We need a strategy to combine compliant behavior of the robot with respect to its environment, and obstacle avoidance.

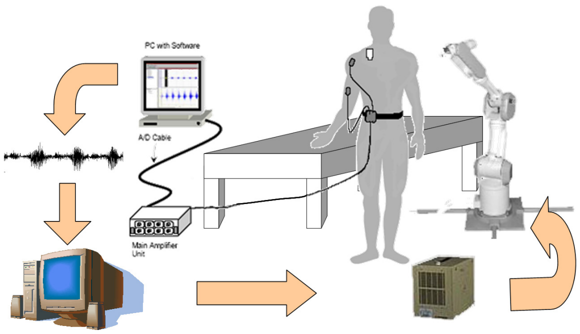

Fig. 1. The problem motivation.

Related previous relevant work has focused on the problem of automotive painting of surfaces that are convex and have no holes, [1], [2], [3]. In [1], the authors decompose the coverage trajectory generation problem into three subproblems: selection of the start curve, selection of the speed profiles along each pass, and selection of the spacing between the passes.

In previous work [4], we presented a methodology to drive the end-effector of a non-redundant manipulator to a surface while avoiding obstacles. Once the end-effector is in close proximity of the surface, a second controller takes over to stabilize the end-effector at a predefined distance to the surface. Motion planning and tracking tasks are then considered, without however taking into account kinematic input constraints. In a recent work, [5], the navigation function methodology established for decentralized multiple robot navigation with input constraints by using hybrid systems analysis in the navigation function methodology, for the problem of existence of input constraints. Also, in [6], the consideration of such input constraints was applied in microrobots where several input sets are more favorable that others. In this attempt a switching controller based on the navigation functions was implemented.

In this paper, we consider the control design problem for a kinematically redundant manipulator, the joint rate inputs of which must remain within pre-specified bounds. We do so by building navigation functions [7], [8] and analyzing the closed loop system that has saturated inputs by means of nonsmooth stability analysis. The system switches between different controllers when it finds itself within certain regions of the workspace (called belt zones [9], [10]). The contribution of this paper is the development of globally uniformly asymptotically stable controllers for redundant articulated robot manipulators, subject to input constraints, to achieve

- reference trajectory tracking with obstacle avoidance on 2-D manifolds embedded in 3-D workspaces, and
- stabilization with obstacle avoidance on 2-D manifolds embedded in 3-D workspaces.

II. Problem Statement

Considering the motion planning problem of a redundant robotic manipulator, with kinematic input constraints, in a workspace with obstacles. The objective is for the robot to move near the surface, and track a predefined trajectory on it. We assume that we have a stationary environment and that we have direct control on the manipulator joint rates. Thus the robot can be kinematically described by a set of integrators:

$$\dot{q} = u, \tag{1}$$

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 is a PhD Student in the Mechanical Engineering Department, National Technical University of Athens, Athens, Greece, xpapag@mail.ntua.gr

H.G. Tanner is with the Department of Mechanical Engineering, MSC01 1150, the University of New Mexico, Albuquerque, NM 87131, USA, tanner@unm.edu

K.J. Kyriakopoulos is with the Faculty of Mechanical Engineering, National Technical University of Athens, Athens, Greece, kkyria@mail.ntua.gr

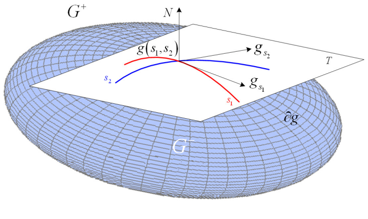

Fig. 2. Representation of tangent's and perpendicular's vectors, of a surface, variations w.r.t parameter's modification.

where $q = [q_1 \ \dots \ q_m]^T \in \mathbb{R}^m$ is the vector of arm joint variables, and u the joint velocity inputs. Let the admissible and feasible configuration space (workspace) for the manipulator be denoted $\mathcal{W} \subset \mathbb{R}^m$. The obstacle free subset of the workspace is denoted $\mathcal{W}_{free} \subseteq \mathcal{W}$. Let $\mathcal{O} \in \mathcal{W} \setminus \mathcal{W}_{free}$ be the set of all obstacles in 3-D workspace. Define a vector valued C^2 function

$$g(s_1, s_2) : \mathbb{R}^2 \rightarrow \mathcal{R}(g), \qquad (2)$$

which represents a closed surface. The range $\mathcal{R}(g) \subset \mathcal{W}_{free}$ of the function expresses mathematically the boundary of the surface across which (at a $\delta > 0$ distance) the robot task is to take place. Let us decompose the space around this surface as follows (Fig. 2):

1) The surface's internal, G^-.
2) The surface's boundary, ∂g.
3) The surface's external, G^+.

The tangent vectors on the surface with respect to parameters s_1 and s_2 are defined as

$$g_{s_1}(s_1, s_2) = \frac{\partial g(s_1,s_2)}{\partial s_1} = \left[\frac{\partial g_x}{\partial s_1}, \ \frac{\partial g_y}{\partial s_1}, \ \frac{\partial g_z}{\partial s_1}\right]^T$$

$$g_{s_2}(s_1, s_2) = \frac{\partial g(s_1,s_2)}{\partial s_2} = \left[\frac{\partial g_x}{\partial s_2}, \ \frac{\partial g_y}{\partial s_2}, \ \frac{\partial g_z}{\partial s_2}\right]^T$$

where the g_x, g_y, g_z denote the coordinate functions of g across the respective dimension.

Due to the C^2 continuity of $g(s_1, s_2)$, we have that $(g_{s_1} \times g_{s_2}) \neq 0, \ \forall s_1, s_2 \in \mathbb{R}$, [11] , and the vectors g_{s_1}, g_{s_2} are linearly independent everywhere. Therefore, every tangent vector to the surface is a linear combination of the vectors g_{s_1} and g_{s_2}, (Fig. 2).

Also, we can define a normalized perpendicular vector to the surface as: $N = \frac{g_{s_1} \times g_{s_2}}{\left\| g_{s_1} \times g_{s_2} \right\|}$.

The goal is for the robot to be able to navigate, to move its end-effector across the surface, and perform a task on the surface, avoiding entering "bad" regions or colliding with obstacles.

The problem can now be stated as follows:
Given a robot manipulator, with kinematic input constraints, and a closed surface, in a known static and bounded environment, find a feedback kinematic control law that stabilizes the end-effector of the manipulator over the given surface, while steering it across the surface

1) *to navigate to any feasible surface point*
2) *to track a predefined trajectory across the surface*

III. Controller Design

A. Redundancy Resolution

We have assumed a redundant manipulator with m d.o.f., like Mitsubishi PA10-7C, Fig. 3, with $m = 7$ d.o.f. The desired position for the manipulator's end-effector in operational space is expressed $p_d = \begin{bmatrix} R_d & x_d \\ O^T & 1 \end{bmatrix}$, and the desired trajectory is represented as and $p_d(t) = \begin{bmatrix} R_d(t) & x_d(t) \\ O^T & 1 \end{bmatrix}$, and the desired velocity is $\dot{p}_d(t) \in \mathbb{R}^6$, where $R_d \in \mathbb{SO}(3)$ is the rotation matrix, and $x_d \in \mathbb{R}^3$ is the matrix of the cartesian coordinates.

Fig. 3. Mitsubishi PA10-7C robotic manipulator configuration, with 7 d.o.f.

Since, we have a redundant manipulator, it is necessary to find the desired joint's configuration by using the data from the operational space. We can use an infinity number of solutions in the joint space, in order to reach the same position in the operational space. It holds that the velocity is equal to $\dot{p}_d = J(q) \cdot \dot{q}_d$, where $J(q)$ is the geometric Jacobian matrix. In case of a redundant manipulator and for inverse kinematics it holds that $\dot{q}_d = J^\dagger \cdot \dot{p}_d + \left(I - J^\dagger \cdot J\right) \dot{q}_a$, [12], where $J^\dagger = J^T \cdot \left(J \cdot J^T\right)^{-1}$ is the right pseudo-inverse of J, and $\dot{q}_a$ is the velocity vector of the redundant degrees of freedom (in the null space). The decision of this vector is based on the maximization of the manipulability measure $\omega(q) = \sqrt{det(J(q)J^T(q))}$.

B. Workspace Decomposition

We have decomposed the workspace in two modes, in order to execute the predefined tasks. The first task is to drive the manipulator towards the surface, which forms the *mode Φ*. The next tasks are, when the manipulator's end-effector is very close to the surface of interest, to drive the manipulator around this surface, in order to reach a point on it, or to make the end-effector to follow a predefined trajectory around the surface. Those tasks are introducing the *mode $\mathcal{B}$*.

The implementation of this workspace decomposition needs the definition of a region, in which the transition from the one mode to the other, occurs. We are going to use the concept of belt zones, in the same notation as in [4], Fig. 4. The "belt zone" is the region close to the ∂g and is thought to be composed of an "internal belt" and an "external belt" region. For the motion tasks considered in this paper the widths of the internal and external belt regions are considered to be fixed.

Let us define the vector valued bijective function, [4]

$$a(s_1, s_2) = g(s_1, s_2) + \rho \cdot N(s_1, s_2) \tag{3}$$

which will help us on the construction of some regions around the surface of interest, with g and N as described in the previous section, and where $0 < \rho < \rho_m$ as in [10].

Therefore, we can now define the vector functions which are used to describe the belt zones to which we have referred above.

$$\beta(s_1, s_2) = g(s_1, s_2) + \delta \cdot N \tag{4}$$

$$\gamma(s_1, s_2) = \beta(s_1, s_2) + \delta \cdot N \tag{5}$$

with $0 < 2 \cdot \delta < \rho_m$, from (3). Both surface processing tasks require stabilization of the end-effector on the surface $\beta(s_1, s_2)$, defined by (4).

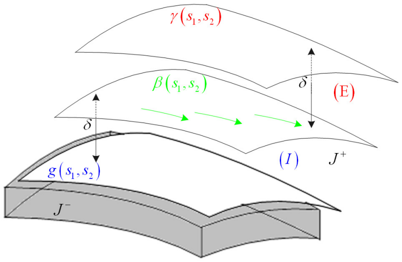

Fig. 4. Representation of Belt Zones, in a part of a surface.

Now we are in position to define the sets of "internal belt", $\mathcal{I}$, and the "external belt", $\mathcal{E}$ (Fig. 4).

$$\mathcal{I} = \{q : k(q) = (1 - \lambda) \cdot \beta + \lambda \cdot g, \lambda \in [0, 1]\}$$

$$\mathcal{E} = \{q : k(q) = (1 - \lambda) \cdot \beta + \lambda \cdot \gamma, \lambda \in (0, 1]\}$$

Since functions g, β, γ are bijective [10], [9], for every $k(q) \in \mathcal{E} \bigcup \mathcal{I}$ there is a unique couple (s_1, s_2).

C. Navigation Function

The controller's design is based on the navigation function, $\varphi : \mathcal{W} \rightarrow \mathbb{R}$, [13]. In the workspace, the volume of the manipulator is represented by a point, using a series of transformations. The obstacles present in the environment are modeled by the navigation function. In order to construct such function, we need to introduce the following parameter $z = q - q_d$, which is the error between the joint configuration of the manipulator, $q \in \mathbb{R}^m$, and the desire joint configuration, $q_d \in \mathbb{R}^m$. The form of the navigation function $\varphi : \mathcal{W}_{ws} \rightarrow \mathbb{R}$ according to [4], is as follows:

$$\varphi(z) = \frac{\gamma_d(z)}{(\gamma_d^{\kappa}(z) + \beta_{ws}(z) \cdot \beta_{\mathcal{O}}(z) \cdot \beta_s(z))^{\frac{1}{\kappa}}} \tag{6}$$

where $\gamma_d(z)$ is the distance to goal function,

$$\gamma_d(z) = \gamma_\Phi(z) = \|z\|^2$$

where the goal is to drive the error to zero, and $\beta_{ws}(z)$ provides the workspace potential, which are defined in dependence on which mode is the end-effector of the manipulator. For mode Φ

$$\beta_{ws}(z) = \beta_\Phi(z) = -\|q - q_0\|^2 + r_0^2$$

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 $\mathcal{W}$), and $r_0 \in \mathbb{R}$ is the workspace's radius. In order to consider the volume occupied by the manipulator, we have used the function $\beta_{\mathcal{O}}(q) \triangleq \prod_{j \in \mathcal{J}} \prod_{i \in \mathcal{I}} \|h^*_{\mathrm{R}_j} - h^*_{\mathcal{O}_i}\|^2$, where $h^*_{\mathrm{R}_j}$ is the position of the transformed robot part j in $\mathcal{W}^*$ and $h^*_{\mathcal{O}_i}$ the position of the transformed obstacle i in $\mathcal{W}^*$, with $\mathcal{W}^*$ is the workspace in which the robot and obstacles are represented by points, as this is defined in [4], and it represents a measure of proximity of the robot to the obstacles. The function $\beta_s(q) \triangleq \prod_{j \in \mathrm{J}} \prod_{k \in \mathcal{K}} \|h^*_{\mathrm{R}_j} - h^*_{s_k}\|^2$ represents the virtual obstacles, in order to achieve singularities avoidance, where $h^*_{s_k}$ are the position of the transformed obstacles, according to [8]. Finally, and $\kappa > 0$ is a parameter.

The convergence to the point on the surface is considered in a two step fashion: First a navigation controller brings the end effector in the belt zone and then a second controller takes over to navigate the system across the surface, and the end-effector is running in the mode Φ. In this mode the surface described by the function $g(s_1, s_2)$, defined in (2) is modeled in the navigation function as an obstacle for the robot's links.

Now it is necessary to define the appropriate form for those function, where the manipulator's end-effector is running in mode $\mathcal{B}$.

To this extend we need to define a navigation function across the 2-D surface, that will provide the navigation vector field. Although theoretically a system that flows according to the tangent space of the 2-D, surface-wrapped navigation field, remains in that 2-D surface, various sources of uncertainty, like sensor noise, model uncertainties and numerical diffusion cause the system to deviate from this surface. To compensate for this problem, we designed an additional vector field perpendicular to the 2-D surface wrapped vector field, which attracts the system on the surface of interest. Such an attractive vector field is provided through an appropriate construction of the γ_d function and by introduction of an additional "perpendicular" workspace function that prohibits exiting the belt zone.

Assume that $h(q)$ is the distance from the surface $g(s_1, s_2)$ on the belt zones. For $h_0 = 0$ we have that the end-effector is on the surface defined by g (boundary of internal region), and for $h_{ext} = 2 \cdot \delta$ we have that the end-effector is on the surface defined by γ (boundary of external region). Also, the desired distance from the surface $g(s_1, s_2)$ is at $h_d = \delta$, that is, when the end-effector is in the surface $\beta(s_1, s_2)$. Thus, we can define the distance to the goal function as:

$$\gamma_d(z) = \gamma_{\mathcal{B}}(z) = \left\| \begin{bmatrix} q \\ h(q) \end{bmatrix} - \begin{bmatrix} q_d \\ h_d \end{bmatrix} \right\|^2$$

where the second term of the vectors is used to attract the end-effector to the surface β. Also the "perpendicular" workspace function is given from the equation:

$$\beta_{ws}(z) = \beta_{\mathcal{B}}(z) = \frac{(h_{ext}-h_d)^2-(h(q)-h_d)^2}{(h_{ext}-h_d)^2}$$

The function $\beta_{\mathcal{B}}(z)$ in this case guarantees that the robot's end-effector cannot leave the belt zone, (Fig. 5). The workspace boundary for the navigation task is thus defined in both the 2-D workspace, where we usually place it to cover a "bad" region (incorporated in the $\beta_{\mathcal{O}}$ function) and in the "perpendicular" direction to prohibit exiting the belt zone.

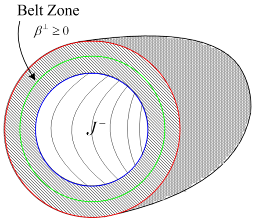

Fig. 5. Representation of the workspace obstacle function based on the Belt Zones construction.

When the manipulator's end-effector is running in the mode $\mathcal{B}$, it can perform a trajectory tracking task. In this case, we have to define an appropriate vector field in order to track the predefined trajectory $(\dot{q}_d,\ q_d)$.

Our system now is time varying, and we have to recast the navigation function as follows:

$$\varphi_{tr}(z,t) = \frac{\gamma_{\mathcal{B}}(z)}{\left[\gamma_{\mathcal{B}}^{\kappa}(z) + \beta_{\mathcal{B}}(z+q_d(t)) \cdot \beta_{\mathcal{O}}(z+q_d(t)) \cdot \beta_s(z+q_d(t))\right]^{1/\kappa}} \quad (7)$$

D. Controller Synthesis

Assume that the robot's initial configuration is $q(0) \in \mathbb{R}^m$, with $p(0) = k(q(0)) \in G^+$, and we would like the end-effector move towards the surface, in order to reach a specific point on it.

We will consider the system as operating in two possible modes: **mode Φ** where $p \in \mathcal{W}_{ext}$, where $\mathcal{W}_{ext} = \{\mathcal{W}_{free} \bigcap G^+\} \setminus \{\mathcal{E} \bigcup \mathcal{I}\}$, with $p = k(q)$ the manipulator's direct kinematics, and **mode $\mathcal{B}$** where $p \in \mathcal{E} \bigcup \mathcal{I}$. We define the following vector fields for each mode:

$$\begin{aligned} f_{\Phi}(z) &= -k_1 \cdot \nabla\varphi_{|\beta_{ws}:=\beta_{\Phi},\gamma_d:=\gamma_{\Phi}}(z) \triangleq -k_1 \cdot \nabla\varphi_{\Phi}(z) \\ f_{\mathcal{B}}(z) &= -k_2 \cdot \nabla\varphi_{|\beta_{ws}:=\beta_{\mathcal{B}},\gamma_d:=\gamma_{\mathcal{B}}}(z) \triangleq -k_2 \cdot \nabla\varphi_{\mathcal{B}}(z) \end{aligned} \quad (8)$$

where $k_1 > 0$, $k_2 > 0$ are constant parameters. During tracking phase, it holds that:

$$f_{tr}(z,t) = -k_3 \cdot \nabla\varphi_{tr}(z,t) - \frac{\nabla\varphi_{tr}}{\|\nabla\varphi_{tr}\|^2} \cdot \frac{\partial\varphi_{tr}}{\partial t} \quad (9)$$

where $k_3 > 0$ is constant parameter.

In order to compensate the manipulator's kinematics input constraints, we have to construct an appropriate controller for each of the above modes. We have introduced the following vector field's form:

$$f_i^{new} = sat_{u_{max}}(f_i) \quad (10)$$

with $sat_c(x) = \begin{cases} x & , \quad |x| \leq c \\ -c & , \quad x < -c \\ c & , \quad x > c \end{cases}$ where c is a constant, u_{max} is the vector of maximum joint velocity values, and $i = \{\Phi, \mathcal{B}\}$ for each mode of operation, respectively. During tracking phase, we have the following form of the vector field:

$$f_{tr}^{new} = sat_{\mu}(f_{tr}) \quad (11)$$

where $\mu = u_{max} - \dot{q}_d(t)$, in order to satisfy the velocity input constraints.

Since, the control scheme we are considering is discontinuous, the right hand of the (10) and (11) are discontinuous, hence we need to consider the Filippov sets created over the switching regions. We have used some results from non-smooth analysis, which are presented in the Appendix.

IV. Stability Analysis

A. Outside the Belt Zone (*Mode* Φ)

Proposition 1: Consider the system $\dot{z} = v$, where $z = q - q_d$, and $v = u - \dot{q}_d$, with u the control law of (1). This system under the control law $v = f_{\Phi}^{new}(z)$, with f_{Φ}^{new} as is defined in (8), is globally asymptotically stable, almost everywhere[1].

Proof: We use the navigation function $V(z) \triangleq \varphi_{\Phi}(z)$ as a Lyapunov function candidate. Function V is a regular function, [14], since it is smooth and it is given from (6). To examine its time derivative, we need to use the *Theorem* 1 (Appendix).

$$\dot{\tilde{V}} = \bigcap_{\xi \in \partial V} \xi^T \cdot K\left[f_{\Phi}^{new}\right](z)$$

Since V is smooth, it holds:

$$\dot{\tilde{V}} = \nabla V^T \cdot K[f_{\Phi}^{new}](z)$$

According to [15], it holds that,

$$K\left[f_{\Phi}^{new}\right](z) = K\left[sat_{u_{max}} \circ f_{\Phi}\right](z) = K\left[sat_{u_{max}}\right](f_{\Phi}(z))$$

since f_{Φ} is C^1, and the function $g(x) = sat_{u_{max}}(x)$ is locally bounded. Therefore, it holds that

$$\dot{\tilde{V}} = \begin{cases} -k_1 \cdot \|\nabla V\|_2^2 & , \ \|f_{\Phi}\| \leq \|u_{max}\| \\ -k_1 \cdot \|\nabla V\|_1 \cdot \|u_{max}\| & , \ \|f_{\Phi}\| > \|u_{max}\| \end{cases}$$

where $k_1 > 0$. Thus, it holds that $\dot{\tilde{V}}$ is strictly negative except for the origin $\nabla V = \nabla\varphi_{\Phi} = 0$. But since φ_{Φ} is a navigation function, the condition $\nabla\varphi_{\Phi} = 0$ holds only at the destination configuration and at a set of measure zero of saddle points. Thus, the system converges almost everywhere to the destination configuration. ■

[1] i.e. everywhere except a set of initial conditions of measure zero.

B. Inside the Belt Zone (*Mode* $\mathcal{B}$)

Since the robot's end-effector initial condition is $p_0 \in G^+$ and by construction it holds that $p_d \in G^-$, the solutions of (1), which are absolutely continuous, intersect the surface $\gamma(s_1, s_2)$, using standard topological arguments, [16]. Therefore there exists finite time T for which the system enters the belt zones. When in the belt zone a mode switch occurs that activates mode $\mathcal{B}$. Once the robot end-effector enters the belt zone, it remains there as the boundaries of the belt zone are repulsive due to the construction of the workspace. Therefore, it has to execute the stabilization over the surface task, and the trajectory tracking task.

1) Stabilization on the Surface:

***Proposition** 2:* The system $\dot{z} = v$ under the control law $v = f_{\mathcal{B}}^{new}(z)$, with $f_{\mathcal{B}}^{new}$ as is defined in (8), is globally asymptotically stable, a.e.

Proof: The proof is very similar to that of Proposition 1, using $V(z) \triangleq \varphi_{\mathcal{B}}(z)$ as a Lyapunov function candidate. ■

2) Tracking on the surface:

***Proposition** 3:* Consider the system $\dot{z} = v$, with admissible initial conditions, i.e. away from the critical points, which is guaranteed by satisfying the following condition:

$$\inf_{\bar{W}} \|\nabla V\|_1 > \frac{\bar{V}_t}{\|\mu\| \cdot k_3} \tag{12}$$

where $V : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$, ∇V is the function's gradient w.r.t. z, $\bar{W}$ is the admissible workspace, and $\bar{V}_t \triangleq \max|\frac{\partial V}{\partial t}|$ is the time derivative bound of V. This system under the control law $v = f_{tr}^{new}(z)$, with f_{tr}^{new} as is defined in (9), is semi-globally uniform asymptotically stable, a.e.

Proof: In the same notation as in previous proof, we set a Lyapunov function candidate $V(z,t) \triangleq \varphi_{tr}(z,t)$, which is a regular function, [14], since it is smooth, (7). We examine its time derivative, based on *Theorem* 1 (Appendix).

$$\dot{\tilde{V}}(z,t) \triangleq \bigcap\nolimits_{\xi \in \partial V(z,t)} \xi^T \cdot \begin{pmatrix} K\left[f_{tr}^{new}\right](z,t) \\ 1 \end{pmatrix} =$$
$$= \begin{pmatrix} \nabla V^T & \frac{\partial \varphi_{tr}}{\partial t} \end{pmatrix} \cdot \begin{pmatrix} K\left[sat_\mu\right](f_{tr}(z,t)) \\ 1 \end{pmatrix} =$$
$$= \begin{cases} -k_3 \cdot \|\nabla V\|_2^2 & , \|f_\Phi\| \le \|\mu\| \\ -k_3 \cdot \|\nabla V\|_1 \cdot \|\mu\| + \frac{\partial V}{\partial t} & , \|f_\Phi\| > \|\mu\| \end{cases}$$

Thus, $\dot{\tilde{V}}(z,t)$ is strictly negative except for the origin $\nabla V = \nabla \varphi_{tr} = 0$, since the condition $-k_3 \cdot \|\nabla V\|_1 \cdot \|\mu\| + \frac{\partial V}{\partial t} < 0 \Rightarrow \inf_{\bar{W}} \|\nabla V\|_1 > \frac{\bar{V}_t}{\|\mu\| \cdot k_3}$ is satisfied, (12). When we are close to the destination, it holds that $\frac{\partial V}{\partial t} = \frac{\partial V}{\partial \gamma_d} \cdot \frac{\partial \gamma_d}{\partial q_d} \cdot \frac{\partial q_d}{\partial t} \to 0$, since $\frac{\partial V}{\partial \gamma_d}$ and $\frac{\partial q_d}{\partial t}$ are bounded and $\frac{\partial \gamma_d}{\partial q_d} \to 0$. Therefore, we are in the case of $\|f_\Phi\| \le \|\mu\|$. Since φ_{tr} is a navigation function, the condition $\nabla \varphi_{tr} = 0$ holds only at the destination configuration, and at a set of measure zero of saddle points. Thus, the system converges almost everywhere to the destination configuration $q_d(t)$. If the condition $\frac{\partial V}{\partial t} \to 0$ is not true, then we can guarantee uniform boundedness.

Since $V(z,t) = \varphi_{tr}(z,t)$, as defined in (7), we can find easily that there is a class $\mathcal{K}$ function $V_1(\|z\|)$ which satisfy $0 < V_1(\|z\|) \le V(z,t)$. Also, we can find a sphere centered at the the destination point $q_d(t)$ with radius $R(t)$ which is tangent to an obstacle of the workspace. Then it is easy to find the minimum radius R_{min}, which guarantee that the destination point is collision free, since q_d is admissible, Fig. 6. Therefore, we can always find a class $\mathcal{K}$ function $V_2(\|z\|) = c_1 \cdot \|z\|^2$, such that $V(z,t) \le V_2(\|z\|)$, where $c_1 = \frac{1}{R_{min}}$.

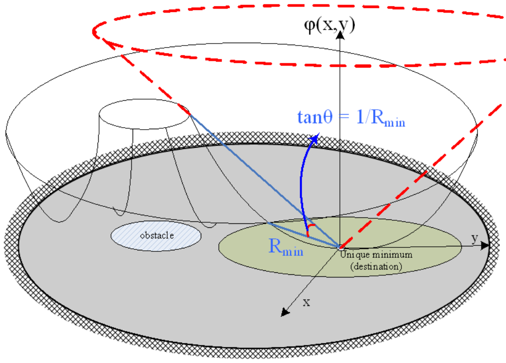

Fig. 6. A navigation function over a free configuration space.

Also, it holds that the $\dot{\tilde{V}}(z,t)$ is strictly negative. Thus, the destination configuration $q_d(t)$, by *Theorem* 2 (Appendix), is uniformly asymptotically stable. ■

C. Combined Hybrid System

The transition from mode Φ to mode $\mathcal{B}$ is happened just once, and then the system is staying at the second mode, since the belt zone's workspace is positively invariant. According to *Theorem* 3 (Appendix), it is hold that, the switched system is globally asymptotically stable a.e., since the condition of this theorem

$$V_{\mathcal{B}}(z,t) - V_\Phi(z,t_1) \le -W$$

is trivially satisfied, because we have just one switch from the mode Φ to the mode $\mathcal{B}$, with V_ϕ and $V_{\mathcal{B}}$ the Lyapunov functions for each mode, t_1 is the time in which the switch from the mode Φ to the mode $\mathcal{B}$ is occurred, and W is a positive definite continuous function.

V. Simulation Results

Computer simulations have been carried out to verify the feasibility and efficacy of the proposed methodology. The robot manipulator that we have used for the implementation of the simulations, is the model of Mitsubishi PA10-7C, in the configuration of Fig. 3, with $m = 7$ d.o.f. The vector of joint's velocity limitation in (rad/sec), is $u_{max} = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 2\pi & 2\pi \end{bmatrix}^T$. The scenario of the simulation contains two 3-D (ellipsoid) obstacles centered at $\mathcal{O}_1 : (-0.3, -0.4, 0.1)$ and $\mathcal{O}_2 : (0.35, -0.3, -0.5)$, and with semi-axes lengths $(0.05, 0.10, 0.20)$, both of them. The surface of interest $g(s_1, s_2)$ is assumed to be an ellipsoid, centered at $(0, 0, 0)$ with semi-axes lengths $(0.75, 0.25, 0.35)$.

The "bad" region's obstacles, which are the areas on the surface that the robot cannot approach, are centered at $\mathcal{O}_{g1} : (-0.33, -0.08, 0.18)$, $\mathcal{O}_{g2} : (0.33, -0.08, -0.18)$ and $\mathcal{O}_{g3} :$

$(-0.33, -0.08, -0.18)$. The robot manipulator's initial configuration was $p(0) = (-0.61, -0.39, -0.13, 0.0, 0.0, 0.0)$, and the target configuration in the operational space was set at $p_d = (0.49, -0.16, 0.13, 1.33, 0.87, -1.33)$. After the end-effector of the robot manipulator reaches its destination point (p_d) (the end-effectors path is red line), it starts a tracking task (blue line), to track a predefined trajectory which is the yellow colored line in Fig. 7-8.

Fig. 7, 8 present the simulation results from a different point of view. Fig. 9, 10 present the cartesian and joint position, Fig. 11, 12 depict the joint's velocity and the error between the real cartesian position and the desired position during tracking, respectively. Our algorithm successfully converges to the goal configuration and track the predefined trajectory avoiding obstacles.

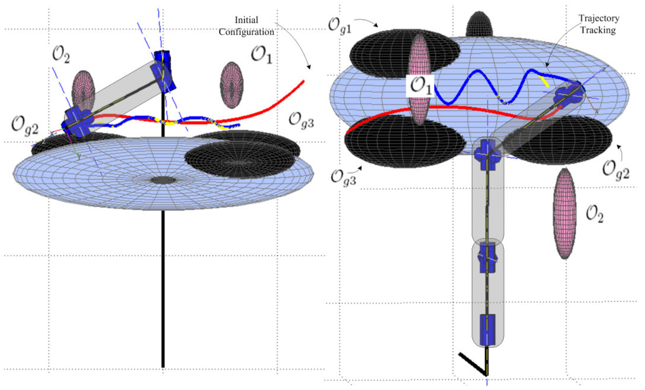

Fig. 7. *Simulation Results:* reaching a point on the surface and tracking.

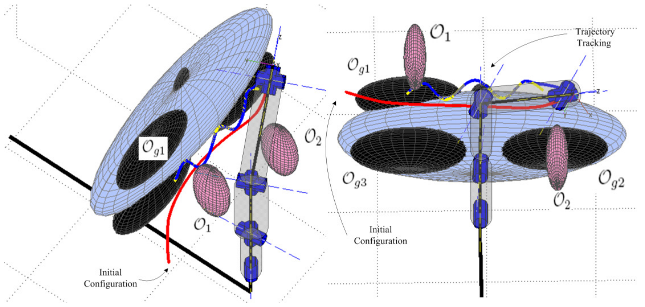

Fig. 8. *Scenario Simulation Results:* reaching a point on the surface and tracking.

VI. Conclusion and Future Work

We presented a methodology for performing navigation and tracking tasks over a 2-dimensional manifold embedded in a 3-dimensional workspace applicable to articulated robotic manipulators, with kinematic input constraints. After safely navigating the manipulator's end-effector to the 2-D manifold, task specific vector fields direct the end-effector towards accomplishing a navigation or a trajectory tracking task across the 2-D manifold. The methodology has theoretically guaranteed global convergence and collision avoidance properties.

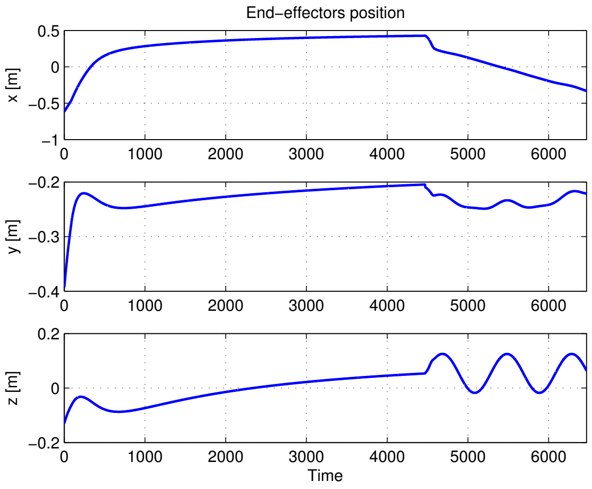

Fig. 9. *Scenario Simulation Results:* Cartesian position.

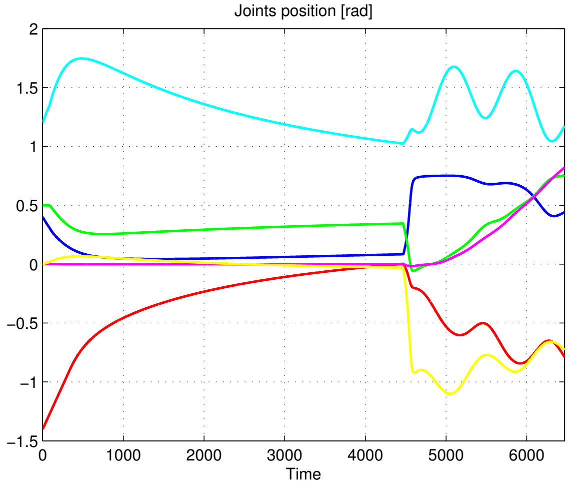

Fig. 10. *Scenario Simulation Results:* Joint's angles, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6, black - q_7.

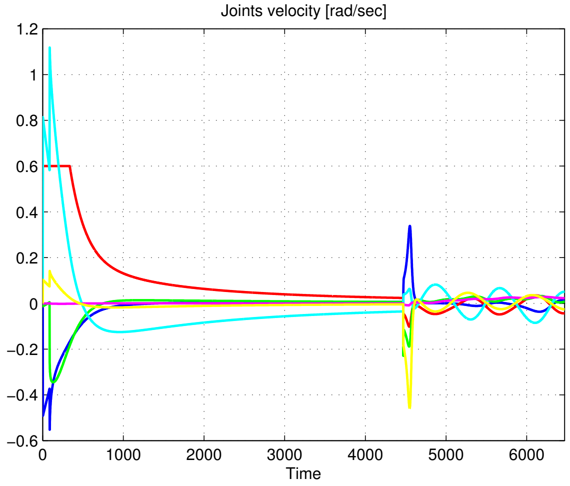

Fig. 11. *Scenario Simulation Results:* Joint's velocities, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6, black - q_7.

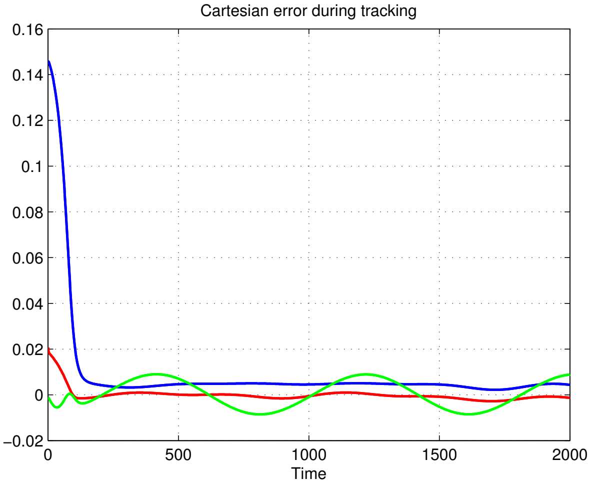

Fig. 12. *Scenario Simulation Results:* Cartesian position error between the real position of the robot and the predefined trajectory during trajectory tracking, blue - x, red - y, green - z.

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.

Appendix

Definition 1: [17] 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

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

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

Theorem 1: [18] Let $x(\cdot, t)$ be a Filippov solution of $\dot{x} = f(x,t)$, and $V : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ be a Lipschitz and in addition, regular function, [14]. Then $V(x,t)$ is absolutely continuous, $\frac{d}{dt}V(x,t)$ exists almost everywhere, and

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

where

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

and ∂V is the Clarke's generalized gradient, [14].

Theorem 2: [16] Let $\dot{x} = f(x,t)$ be essentially locally bounded and $0 \in K[f](0,t)$ in a region $\mathcal{Q} \supset \{x \in \mathbb{R}^n | \|x\| < r\} \times \{t | t_0 \le t < \infty\}$. Also, let $V : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ be a regular function satisfying $V(0,t) = 0$ and

$$0 < V_1(\|x\|) \le V(x,t) \le V_2(\|x\|), \quad for \ \ x \neq 0$$

in $\mathcal{Q}$ for some $V_1, V_2 \in$ class$\mathcal{K}$. Then if there exists a class $\mathcal{K}$ function $w(\cdot)$ in $\mathcal{Q}$ with the property $\dot{\tilde{V}} \le -w(x) < 0$ then the solution $x(t) \equiv 0$ is uniformly asymptotically stable.

Theorem 3: [19] Let $\dot{x} = f_\upsilon(x)$, be a finite family of globally asymptotically stable systems, and let V_υ, $\upsilon \in \mathcal{P}$ 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_υ, $\upsilon \in \mathcal{P}$, with the property that for every pair of switching times (t_i, t_j), $i < j$ such that $\sigma(t_i) = \sigma(t_j) = \upsilon \in \mathcal{P}$, and $\sigma(t_k) \neq \upsilon$, for $t_i < t_k < t_j$, where σ is the switching signal, we have that

$$V_\upsilon(x(t_j)) - V_\upsilon(x(t_i)) \le -W_\upsilon(x(t_i))$$

Then the switched system is globally asymptotically stable.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] S. Loizou and K. Kyriakopoulos, "Navigation of multiple input constraint micro-robotic agents," *2005 International Conference on on Intelligent Robots and Systems*, 2005.

[7] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances Appl. Math.*, vol. 11, pp. 412–442, 1990.

[8] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.

[9] S. Loizou, H. Tanner, V. Kumar, and K. Kyriakopoulos, "Closed loop motion planning and control for mobile robots in uncertain environmentss," *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

[10] 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.

[11] W. M. Boothby, *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 1986.

[12] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.

[13] 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.

[14] F. Clarke, *Optimization and Nonsmooth Analysis*. Addison - Wesley, 1983.

[15] B. Paden and S. Sastry, "A calculus for computing filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Transactions on Circuits and Systems*, vol. 34, no. 1, p. 7382, January 1987.

[16] H. Khalil, *Nonlinear Systems*. Prentice-Hall, 1996.

[17] A. Filippov, *Differential equations with discontinuous right-hand sides*. Kluwer Academic Publishers, 1988.

[18] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1910–1914, 1994.

[19] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.