Motion Tasks and Force Control for Robot Manipulators on Embedded 2-D Manifolds

Xanthi Papageorgiou, Savvas G. Loizou and Kostas J. Kyriakopoulos

Abstract— In this paper we present a methodology to drive the end effector of a robotic manipulator across the surface of an object in the workspace, and at the same time the manipulator can apply a force to the object, through its end-effector. Three typical tasks are considered, namely stabilization of the end effector over the object's surface and applying a specific force on it, motion planning and eventually trajectory tracking of the end effector across the object's surface. The proposed controllers utilize navigation functions and are based on the belt zone vector fields concept. The derived dynamic controllers are realized using an integrator backstepping methodology. The derived feedback based controllers guarantee global convergence and collision avoidance. The closed form solution provides fast feedback rendering the methodology particularly suitable for implementation on real time systems. The properties of the proposed methodology are verified through non-trivial computer simulations.

I. INTRODUCTION

Performing motion tasks across object surfaces constitutes a challenging problem of the robotics field with many applications including robotic surface painting, surface cleaning, surface inspection, etc. Our main motivation comes from the field of neuro-robotics. One of the main tasks of neuro-robotics is to make a robot execute a task by interfacing with the neural system (Fig. 1) e.g. by processing electromyographic activity, etc. In most of the cases, these signals are noisy and rather "incomprehensible" to directly control a robot particularly in clattered environments. In those cases we need a strategy to make the robot compliant with its environment and at the same time avoiding obstacles. The main difficulties of the above tasks arise when the considered surface is not planar. Moreover the non-planar surface might include "bad regions" that must be avoided.

Most of previous relevant research has focused on the problem of automotive painting of surfaces that are convex and have no holes, [1], [2], [3]. Also 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. At the other hand literature is

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

S.G. Loizou is a Post Doctoral Researcher in the Grasp Laboratory, University of Pennsylvania, Philadelphia, PA 19104-6228, USA, `sloizou@seas.upenn.edu`

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

rich in the field of robot force control. The main approaches in this area are impedance control, [4], hybrid position/force control, [5], and parallel control, [6], [7]. All these schemes are not applicable in the case of cluttered environment or in appearance of "bad regions" during the constraint motion.
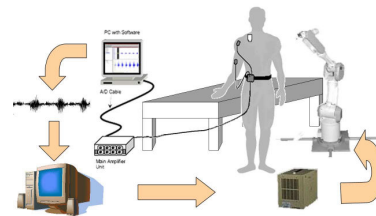

Fig. 1. The problem motivation.

In the present work, we have used navigation functions, [8], to drive the manipulator safely in its environment and at the same time to execute some specific tasks. The goal is reached when several subtasks, in which we have divided this work, are executed. The first phase is to make the manipulator to approach a predefined surface which is called the surface of interest. Once the end-effector is in the close proximity of the surface, we have constructed a second controller in order to succeed in the second phase of this work. This phase is consisted of the stabilization of the manipulator's end-effector across the surface in order to apply a specific force on it, while at the same time, depending on the application, performs a motion planning or trajectory tracking task across the surface, avoiding the "bad regions". Our basic idea is to use a navigation controller to drive the manipulator's end effector safely to the proximity area of the surface of interest applying a specific force on it, and then according to the required task (motion planning or trajectory tracking), switch to a task specific controller. Each of those two task specific controllers can successfully carry out the task while making the manipulator's end effector to produce a predefined force to the surface of interest. This is achieved through the use of appropriately constructed belt zone and task specific vector fields, introduced in [9], [10], [11]. To handle the volume and the articulated nature of the robot manipulator we have used the methodology that was first introduced in [12]. The main contributions of this paper can be summarized as follows:

- A novel theoretically guaranteed compliant dynamic trajectory tracking controller, achieving obstacle avoidance and concurrent stabilization during tracking over 2-D manifolds embedded in 3-D workspaces, for articulated robot manipulators.

- A provably correct compliant dynamic controller performing motion planning and concurrent stabilization tasks over 2-D manifolds embedded in 3-D workspaces, for articulated robot manipulators.

The rest of the paper is organized as follows: Section II formally states the considered problem, introducing preliminary definitions, notation and some technical Lemmas, required for further discussion. Section III describes the navigation of an articulated non-point robot while section IV presents the construction of the vector field that is used for robot navigation. Section V introduces the proposed control law. Section VI presents the simulation results and the paper concludes with section VII.

II. PROBLEM STATEMENT

Our analysis is demonstrated by considering the motion planning problem of a robotic manipulator in a workspace with obstacles under additional task constraints. Consider a manipulator of m-dof, which is described dynamically as:

$$B(q) \cdot \ddot{q} + C(q, \dot{q}) + gr(q) + J^T(q) \cdot F = \tau \quad (1)$$

where $B(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis term, $gr(q)$ is the Gravity term, $J(q)$ is the geometric Jacobian matrix, $F \in \mathbb{R}^6$ denotes the wrench (vector of forces and torques) exerted by the end-effector of the robot manipulator on the environment, the $q = [q_1 \ldots q_m]^T \in \mathbb{R}^m$ is the vector of arm joint variables and $\tau \in \mathbb{R}^m$ the joint torque inputs, [13]. Let the admissible and feasible configuration space (workspace) for the manipulator be $\mathcal{W} \subset \mathbb{R}^m$. The obstacle free subset of the workspace is denoted $\mathcal{W}_{free} \subseteq \mathcal{W}$, and $\varphi : \mathcal{W} \to \mathbb{R}$ is the potential (navigation) function. Let $\mathcal{O} \in \mathcal{W} \backslash \mathcal{W}_{free}$ be the set of all obstacles in 3-D workspace. Let now define the closed surface of interest, which is represented by the vector valued C^2 function:

$$g(s_1, s_2) : \mathbb{R} \times \mathbb{R} \to \mathcal{R}(g) \quad (2)$$

The range $\mathcal{R}(g) \subset \mathcal{W}_{free}$ of the function will serve as the boundary of the surface across which the surface processing task will take place. Let us define the following topology (Fig. 2):

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

We can now define the tangent vectors on the surface w.r.t parameters s_1 and s_2 as $g_{s_1}(s_1, s_2) = \frac{\partial g(s_1, s_2)}{\partial s_1}$, and $g_{s_2}(s_1, s_2) = \frac{\partial g(s_1, s_2)}{\partial s_2}$.

Due to the C^2 continuity of $g(s_1, s_2)$, we have [14] that $(g_{s_1} \times g_{s_2}) \neq 0$, $\forall s_1, s_2 \in \mathbb{R}$, 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).

We now define the vector valued function

$$a(s_1, s_2) = g(s_1, s_2) + \rho \cdot N(s_1, s_2) \quad (3)$$

where $N = \frac{g_{s_1} \times g_{s_2}}{\|g_{s_1} \times g_{s_2}\|}$ is the normalized perpendicular vector to the surface.

Following the same line of thought as in our previous work, [9] we have proved, we need the function $a(s_1, s_2)$, from (3) to be bijective.
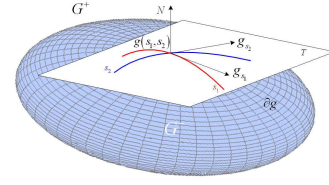

Fig. 2. Representation of tangent's and perpendicular's vectors, of a surface, variations w.r.t parameter's modification.

This implies that locally, in order to satisfy the needed condition of the function $a(s_1, s_2)$, we must choose $0 < \rho < \rho_m$ as in [9]. This justifies the selection of the maximum curvature, so a non-negative ρ can always be defined.

The problem can now be stated as follows:
Given a robot manipulator driven by (1), and a closed surface $g(s_1, s_2)$, (2), find a feedback dynamic control law $\tau = \tau(q)$, that stabilizes the end-effector of the manipulator over the given surface and making it compliant to the surface, while steering it across the surface to either

- *navigate to any feasible surface point, or*
- *track a predefined trajectory across the surface*

The environment is considered known, static and bounded.

III. NAVIGATION OF AN ARTICULATED NON-POINT ROBOT

The main goal of this section is to construct a potential function in order to use it at an articulated non-point robot. By using this potential function it would be possible to navigate a robotic manipulator into a cluttered by obstacles environment, in order to execute a task, while a collision avoidance with obstacles is occurred (in both end-effector and robot links point of view). Based on this scope this potential function is including the volumes of the links and the singularity points of the manipulator.

A. A Potential Field for Non-Point Robots

For constructing a potential field it is necessary to represent mathematically the system and its environment. Most of previous works are based on the assumption that we can represent the system as a point in the workspace, [9]. To apply the methodology to an actual articulated robotic manipulator, though, we need to also consider the volume occupied by the manipulator. To this extend we have used the method proposed in [12] which extends the navigation function approach of [15] to articulated non-point robots using a series of diffeomorphic transformations.

According to [12], the shape of the robotic system R and the obstacles $\mathcal{O}$ in a 3-D workspace $\mathcal{W} \subset \mathbb{R}^m$ are considered as a unions of generalized n-ellipsoids: $R = \cup_{j \in J} R_j$ and $\mathcal{O} = \cup_{i \in I} \mathcal{O}_i$, where $J = \{1, \ldots, n_R\}$ and $I = \{1, \ldots, n_{\mathcal{O}}\}$ the number of ellipsoids covering the volume of robot and obstacles, respectively.

In $\mathcal{W}$, a sequence of smooth transformations is used to create spaces where the robot and all obstacles are represented by points. Therefore, we have a workspace $\mathcal{W}^*$ where the robot and obstacles are represented by points. The following represents a measure of proximity of the robot to the obstacles in the transformed workspace $\beta_{\mathcal{O}} \triangleq \prod_{j\in J}\prod_{i\in I}\|h^*_{R_j} - h^*_{\mathcal{O}_i}\|^2$, where $h^*_{R_j}$ 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 $\mathcal{W}^*$. For details on the construction, the reader is referred to [12].

B. Singularities Avoidance

An introduction of a number of virtual obstacles can be used in order to avoid the manipulator's singularities. Singularity regions $S = \cup_{k\in\mathcal{K}}S_k$ are sets of measure zero within the configuration space. Singularities can be considered as solutions of the equation: $det(J^T J) = 0$, with J the Jacobian of the robot. Thus, it is feasible to enclose the singularity regions inside ellipsoids $\mathcal{O}_{s_k}$, representing virtual obstacles affecting the motion of the robot end-effector. Following the approach of [12], each ellipsoid $\mathcal{O}_{s_k}$ is reduced into a point $h^*_{s_k}$. Therefore, the singularity avoidance is achieved by introducing virtual obstacles $\beta_s \triangleq \prod_{j\in J}\prod_{k\in\mathcal{K}}\|h^*_{R_j} - h^*_{s_k}\|^2$.

C. Navigation Vector Field

Let us now construct the navigation function that achieves the predefined requirements. Assume that $\varphi : \mathcal{W}_{ws} \to \mathbb{R}$ according to [8], is the navigation function with the following form:

$$\varphi(q) = \frac{\gamma_d(q)}{(\gamma_d^\kappa(q) + \beta_{ws}(q)\cdot\beta_{\mathcal{O}}(q)\cdot\beta_s(q))^{\frac{1}{\kappa}}} \quad (4)$$

where $\gamma_d(q) = \|k(q) - p_{d_1}\|^2$ is the distance to the goal function, $\|\cdot\|$ is the Euclidean norm, $p = k(q)$ is the end effector position, with $k(q)$ the manipulator kinematics. Also, $p_{d_1} = [\mathbf{x}_{d_1} \quad \theta_{d_1}]^T$ is the target configuration for the navigation function, where $\mathbf{x}_{d_1} \in \mathbb{R}^3$, and $\theta_{d_1} \in \mathbb{R}^3$ are the desired position and orientation of the robot's end-effector. Furthermore, $\mathcal{W}_{ws} \subset \mathcal{W}_{free}$ is the set where the task takes place, and the function $\beta_{ws}(q) := \beta_0(q) = -\|k(q) - p_0\|^2 + r_0^2$ provides the workspace potential for the navigation in G^+ (i.e. $k(q) \in G^+$), $\beta_{\mathcal{O}}(q)$ and $\beta_s(q)$ as are described above, and $\kappa > 0$ is a parameter. Thus, for the navigation task in G^+, the vector field is given by $\nabla\varphi_\tau = \nabla\varphi|_{\beta_{ws}:=\beta_0}$.

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.

This navigation function $\nabla\varphi_\tau = \nabla\varphi|_{\beta_{ws}:=\beta_0}$ of (4), is implemented to drive the end-effector towards $g(s_1, s_2)$. This is achieved by placing the destination configuration, p_{d_1} in G^- (inside of the surface of interest). Since our goal is to perform compliant motion tasks across the boundary of $g(s_1, s_2)$, when the robot's end-effector reaches a certain distance from the boundary of $g(s_1, s_2)$, a switch to an appropriate task (navigation or trajectory tracking) specific controller occurs in order to allow the interaction between the

robot and its environment. To construct such a vector field we are going to use the concept of belt zones, [9]- [11].

IV. TASK SPECIFIC VECTOR FIELDS

A. Belt Zones

The "belt zone" is the region close to the ∂g, and is thought to be composed of an "internal belt" ($\mathcal{I}$), and an "external belt" ($\mathcal{E}$) region, which is represented in Fig. 3. For the motion tasks considered in this paper the widths of the internal and external belt regions are considered to be fixed.

Assume that the surface of interest $g(s_1, s_2)$ is modeled as a spring with K_e its (homogenous) stiffness matrix. When the robot's end-effector interact with this surface, in order to apply a constant force on it, it is assumed that the robot penetrate the surface by the meaning of $\epsilon^\perp$ displacement from the surface's boundary (Fig. 3). Let $F^d \in \mathbb{R}^6$ is the desired wrench that we would like the robot apply to the surface, that denote a displacement from the boundary of the surface $\epsilon_d^\perp = K_e^{-1}F^d$.

Thus, we can now define the above regions of the belt zones, by using the vector functions, $g'(s_1, s_2) = g(s_1, s_2) - \epsilon_d^\perp \cdot N$, $\beta(s_1, s_2) = g(s_1, s_2) - (\delta + \epsilon_d^\perp)\cdot N$, and $\gamma(s_1, s_2) = g(s_1, s_2) + (\delta - \epsilon_d^\perp)\cdot N$, where $0 \le \epsilon_d^\perp < \delta$, $0 < 2\cdot\delta < \rho_m$, and N is the normalized perpendicular vector to the surface. Both surface processing tasks require stabilization of the end-effector on distance $\epsilon_d^\perp$ from the surface $g(s_1, s_2)$.
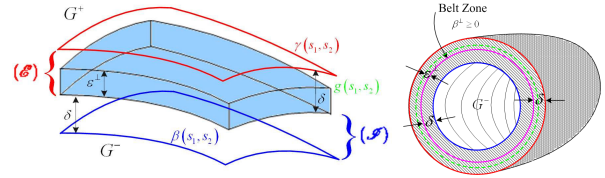

Fig. 3. Representation of Belt Zones, Fig. 4. Representation of the in a part of a surface. workspace obstacle function.

The sets of "internal belt", $\mathcal{I}$, and "external belt", $\mathcal{E}$ are denoted (Fig. 3), $\mathcal{I} = \{q : k(q) = (1-\lambda)\cdot g' + \lambda\cdot\beta, \lambda \in [0,1]\}$, and $\mathcal{E} = \{q : k(q) = (1-\lambda)\cdot g' + \lambda\cdot\gamma, \lambda \in (0,1]\}$. Since functions g, β, γ are bijective [9], [11], for every $k(q) \in \{\mathcal{E}\bigcup\mathcal{I}\}$ there is a unique couple (s_1, s_2).

Using the above belt zones, we can now define the appropriate vector field, that will be used so that the robot end-effector navigates across the surface of interest to reach a specified destination point $p_{d_2} = [\mathbf{x}_{d_2} \quad \theta_{d_2}]^T$, while at the same time the force requirement is held.

B. Vector Field on a Surface

To this extend we need to define a navigation function across the 2-D surface, in order to execute the force control task, 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, and which makes it compliant to the surface. Such an attractive vector field is provided by (4), through an appropriate construction of the γ_d function and by introduction of an additional "perpendicular" workspace function that prohibits exiting the belt zone and allows applying force on the 2-D surface.

Assume that $h(x,y,z)$ is the distance from the surface $\beta(s_1,s_2)$ on the belt zones. For $h_0 = 0$ we have that the end-effector is on the surface defined by β (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 $g'(s_1,s_2)$, at $\epsilon_d^\perp$ distance inside of g. Thus, we can define the distance to the goal function as:

$$\gamma_d(q) = \left\| \begin{bmatrix} k(q) \\ h\left(k(q)\right) \end{bmatrix} - \begin{bmatrix} p_{d_2} \\ h_d \end{bmatrix} \right\|^2 \tag{5}$$

where the second term of the vectors is used to attract the end-effector to the surface g'. Also the "perpendicular" workspace function is given from $\beta^\perp(q) = \frac{(h_{ext}-h_d)^2 - (h(k(q))-h_d)^2}{(h_{ext}-h_d)^2}$.

The function $\beta^\perp(q)$ in this case guarantees that the robot's end-effector cannot leave the belt zone, (Fig. 4). 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 β_O function) and in the "perpendicular" direction to prohibit exiting the belt zone. The vector field in this case is provided by $\nabla\varphi_\tau = \nabla\varphi|_{\beta_{ws}:=\beta^\perp}$ for the surface navigation task.

Now we are in position to define the control law in order to drive the robot's end-effector towards the predefined tasks.

V. CONTROL STRATEGY

We assume that we have a stationary environment and the robot manipulator can be described trivially by a fully actuated, second order dynamic model. 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. The goal is for the robot to be able to navigate using the navigation function constructed on a 3-D space, to move its end-effector across the surface $g'(s_1,s_2)$ and perform a task on the surface, avoiding entering "bad" regions or colliding with obstacles.

A. Reaching a point on a surface

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 and the same time apply force on it.

We will consider the system as operating in two possible modes: ***Mode*** Φ where $p \in \mathcal{W}_{ext}$, where $\mathcal{W}_{ext} = \{\mathcal{W}_{free} \bigcap G^+\} \setminus \{(\mathcal{E} - \delta\mathcal{E}) \bigcup \mathcal{I}\}$, with $p = k(q)$ and $\delta\mathcal{E} =$

$\{q : k(q) = (1-\lambda) \cdot g' + \lambda \cdot \gamma, \lambda \in (1 - \epsilon_{\mathcal{E}}, 1]\}$, $\epsilon_{\mathcal{E}} > 0$, and ***Mode*** $\mathcal{B}$, where $p \in \{\mathcal{E} \bigcup \mathcal{I}\}$. We define the following vector fields for each mode:

$$\begin{aligned} f_\Phi(q) &= -(\nabla\varphi|_{\beta_{ws}:=\beta_0} - F_T) = -(\nabla\varphi_\Phi - F_T) \\ f_\mathcal{B}(q) &= -(\nabla\varphi|_{\beta_{ws}:=\beta^\perp} - F_T) = -(\nabla\varphi_\mathcal{B} - F_T) \end{aligned} \tag{6}$$

where $F_T = \begin{cases} \varnothing & , \quad p \in \{\mathcal{W}_{ext} \bigcup \mathcal{E}'\} \\ -\frac{\dot{q}}{\|\dot{q}\|} \cdot \tanh(\|\dot{q}\|) , & p \in \{\mathcal{I}'\} \end{cases}$, denotes the dissipative forces of the system, $\mathcal{E}' = \{q : k(q) = (1-\lambda)\cdot g+\lambda\cdot\gamma, \lambda \in [0,1]\}$, $\mathcal{I}' = \{q : k(q) = (1-\lambda)\cdot g+\lambda\cdot\beta, \lambda \in [0,1]\}$, $\dot{q}$ is the vector of joint velocities and $\tanh(\cdot)$ is the hyperbolic tangent function.

The dynamic representation of the system is given from (1). Using the inverse dynamics control law with force measurements $\tau = B(q) \cdot u + C(q,\dot{q}) + gr(q) + J^T(q) \cdot F$ we have the linear representation of the system:

$$\ddot{q} = u \tag{7}$$

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. We have the following:

Proposition 1: Consider the system (7) and the control law:

$$u = -c \cdot (\dot{q} - f_i) + \frac{\partial f_i}{\partial q} \cdot \dot{q} + f_i \tag{8}$$

where c a positive tuning constant. For initial conditions in $\{W_{free} \bigcap G^+\} \setminus \{\mathcal{E} \bigcup \mathcal{I}\}$ and with the vector field as defined for $i = \Phi$, the system converges to the set $\{\mathcal{E} \bigcup \mathcal{I}\}$, a.e.[1]. When the system is in $\{\mathcal{E} \bigcup \mathcal{I}\}$, the vector field as defined for $i = \mathcal{B}$ is activated and the system converges globally asymptotically to the destination configuration, a.e.

Proof: The control law construction is inspired by the backstepping controller design proposed by [16].

Assume that the robot's initial conditions are in $\{W_{free} \bigcap G^+\} \setminus \{\mathcal{E} \bigcup \mathcal{I}\}$. Then the vector field is f_Φ from (6).

We form the Lyapunov function $V(q,\dot{q}) = \varphi_\Phi(q) + \frac{1}{2} \cdot (\dot{q} - f_\Phi(q))^2$ where $\varphi_\Phi = \varphi|_{\beta_{ws}:=\beta_0}$ is given from (4). Taking the time derivative of the Lyapunov function $\dot{V} = \nabla\varphi_\Phi \cdot \dot{q} + (\dot{q} - f_\Phi) \cdot \left(u - \frac{\partial f_\Phi}{\partial q} \cdot \dot{q}\right)$. Substituting the control law u as it is defined in (8), we have that: $\dot{V} = -\|f_\Phi\|^2 - c \cdot (\dot{q} - f_\Phi)^2 + \dot{q} \cdot F_T \leq -\|\nabla\varphi_\Phi - F_T\|^2$ since, $\dot{q} \cdot F_T = \begin{cases} 0 & , \quad p \in \{\mathcal{W}_{ext} \bigcup \mathcal{E}\} \\ -\|\dot{q}\| \cdot \tanh(\|\dot{q}\|) \leq 0 , & p \in \{\mathcal{I}'\} \end{cases}$, $\forall \dot{q} \in \mathbb{R}^m$.

Hence, LaSalle's Invariance Principle, [17] guarantees convergence of q to the largest invariant set contained in the set $\{q \mid \|\nabla\varphi_\Phi(q)\| = 0\}$, since $\|\nabla\varphi_\Phi - F_T\| = 0 \Rightarrow \dot{q} = 0 \Rightarrow F_T = \nabla\varphi_\Phi = 0$. The critical points of the navigation function are isolated, [8]. Thus the set of initial conditions that lead to saddle points are set of measure zero. Thus, the largest invariant set contained in the $\|\nabla\varphi_\Phi\| = 0$ consists of the target configuration p_{d_1} and the saddle points.

---

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

Since the robot's end-effector initial condition is $p_0 \in G^+$ and by construction it holds that $p_{d_1} \in G^-$, the solutions of (7), which are absolutely continuous, intersect the surface $g(s_1, s_2)$, using standard topological arguments. 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 $\mathcal{B}$ and the control law (8) changes to $u_{f_i = f_{\mathcal{B}}}$. Once the robot end-effector enters the belt zone, it remain there as the boundaries of the belt zone are repulsive due to the construction of the workspace. Following the same procedure as for the mode Φ, by choosing a Lyapunov function $V(q, \dot{q}) = \varphi_{\mathcal{B}}(q) + \frac{1}{2} \cdot (\dot{q} - f_{\mathcal{B}}(q))^2$ where $\varphi_{\mathcal{B}} = \varphi|_{\beta_{ws} := \beta^\perp}$, and with the vector field $f_{\mathcal{B}}$ from (6), it holds that the system converge to the target configuration p_{d_2} a.e. ∎

B. Tracking

Now, let us consider a tracking task across the surface $\beta(s_1, s_2)$. The task is described by a known trajectory $q_d(t)$ on it (e.g. for neuro-robotic set-up, a manipulator keep a pen and we would like it to write something on a paper). Let us now define the appropriate control law in order to track the predefined trajectory.

We introduce a navigation function of the form:

$$\varphi_{tr}(q, t) = \frac{\gamma_d(q, t)}{\left(\gamma_d^\kappa(q, t) + \beta^\perp(q) \cdot \beta_\mathcal{O}(q) \cdot \beta_s(q)\right)^{1/\kappa}} \quad (9)$$

where γ_d is similar to (5), and represents the distance to the trajectory.

We consider convergence of the system to a small ball of radius $\varepsilon > 0$ containing the target.

To define the tracking controller we will use the P_1 region introduced in [18]. This set is used to identify sets of points that contain measure zero sets whose positive limit sets are saddle points: $P_1 \triangleq \{p : (\lambda_{\min} < 0) \wedge (\lambda_{\max} > 0) \wedge (|\hat{v}_{\lambda_{\min}} \cdot \nabla \varphi| \leq \varepsilon_1)\}$ with $\varepsilon_1 < \min\limits_{C = \{p : \|p - p_d\| = \varepsilon\}} (\|\nabla \varphi(C)\|)$, where $\lambda_{\min}$, λ_{max} are the minimum and the maximum eigenvalues of the Hessian $\nabla^2 \varphi$, and $\hat{v}_{\lambda_{\min}}$ the unit eigenvector corresponding to the minimum eigenvalue of the Hessian. If $|\hat{v}_{\lambda_{\min}} \cdot \nabla V| = 0$ then the set P_1 consists of the measure zero set of initial conditions that lead to saddle point, [18], [9], [11]. In view of this, ε_1 can be chosen to be arbitrarily small so the sets defined by P_1 eventually consist of thin sets containing sets of initial conditions that lead to saddle points.

Now consider that the system is operating in tracking mode. Then the task specific vector field is $f^{tr} = -(\nabla \varphi_{tr} - F_T) - \frac{\partial \varphi_{tr}}{\partial t} \cdot \frac{(\nabla \varphi_{tr} - F_T)}{sw(\|\nabla \varphi_{tr} - F_T\|^2, \varepsilon^2) - \varepsilon^2 \cdot \sigma\left(\frac{\partial \varphi_{tr}}{\partial t}\right) \cdot \sigma(\|\nabla \varphi_{tr} - F_T\|^2)}$ with F_T denotes the dissipative forces of the system, in the same notation as in the previous subsection, $\sigma(x) = \frac{x}{1 + |x|}$, $sw(x, e) = \begin{cases} x, & x \geq e \\ e, & x < e \end{cases}$, and φ_{tr} is given from (9).

***Proposition* 2:** Consider the system (7) with initial conditions in $\{\mathcal{E} \bigcup \mathcal{I}\} \backslash P_1$. Then the control law

$$u = -c \cdot (\dot{q} - f^{tr}) + \left(\frac{\partial f^{tr}}{\partial q} \cdot \dot{q} + \frac{\partial f^{tr}}{\partial t}\right) - (\nabla \varphi_{tr} - F_T) \quad (10)$$

converges to the set $T = \{q : \|k(q) - p_d\| < \varepsilon\}$, a.e.

Proof: We form the Lyapunov function $V(q, \dot{q}) = \varphi_{tr}(q) + \frac{1}{2} \cdot (\dot{q} - f^{tr}(q))^2$ and take its time derivative $\dot{V} = \left(\frac{\partial \varphi_{tr}}{\partial t} + \nabla \varphi_{tr} \cdot \dot{q}\right) + (\dot{q} - f^{tr}) \cdot \left(u - \frac{\partial f^{tr}}{\partial q} \cdot \dot{q} - \frac{\partial f^{tr}}{\partial t}\right)$. Substituting the control law u as it is defined in (10), we have that: $\dot{V} = \left[\frac{\partial \varphi_{tr}}{\partial t} + (\nabla \varphi_{tr} - F_T) \cdot f^{tr}\right] - c \cdot (\dot{q} - f^{tr})^2 + \dot{q} \cdot F_T$
$$\Rightarrow \dot{V} \leq \frac{\partial \varphi_{tr}}{\partial t} + (\nabla \varphi_{tr} - F_T) \cdot f^{tr} = \dot{V}_{tr}$$
since, $\dot{q} \cdot F_T \leq 0$, $\forall \dot{q} \in \mathbb{R}^m$, as has proved in the previous subsection.

Substituting f^{tr} to $\dot{V}_{tr}$, and since we pursue convergence in the set T, we get: $\dot{V}_{tr} = -\|\nabla \varphi_{tr} - F_T\|^2 + \frac{\partial \varphi_{tr}}{\partial t} \cdot \left(1 - \frac{\|\nabla \varphi_{tr} - F_T\|^2}{\|\nabla \varphi_{tr} - F_T\|^2 - \varepsilon^2 \cdot \sigma\left(\frac{\partial \varphi_{tr}}{\partial t}\right) \cdot \sigma(\|\nabla \varphi_{tr} - F_T\|^2)}\right)$. We can now discriminate the following cases:

- $\frac{\partial \varphi_{tr}}{\partial t} = 0 \Rightarrow \dot{V}_{tr} = -\|\nabla \varphi_{tr} - F_T\|^2 \leq 0$
- $\frac{\partial \varphi_{tr}}{\partial t} > 0 \Rightarrow 0 < \sigma(\frac{\partial \varphi_{tr}}{\partial t}) < 1 \Rightarrow \|\nabla \varphi_{tr} - F_T\|^2 - \varepsilon^2 < \|\nabla \varphi_{tr} - F_T\|^2 - \varepsilon^2 \cdot \sigma\left(\frac{\partial \varphi_{tr}}{\partial t}\right) \cdot \sigma\left(\|\nabla \varphi_{tr} - F_T\|^2\right) < \|\nabla \varphi_{tr} - F_T\|^2 \Rightarrow \dot{V}_{tr} \leq 0$
- $\frac{\partial \varphi_{tr}}{\partial t} < 0 \Rightarrow -1 < \sigma(\frac{\partial \varphi_{tr}}{\partial t}) < 0 \Rightarrow \|\nabla \varphi_{tr} - F_T\|^2 < \|\nabla \varphi_{tr} - F_T\|^2 - \varepsilon^2 \cdot \sigma\left(\frac{\partial \varphi_{tr}}{\partial t}\right) \cdot \sigma\left(\|\nabla \varphi_{tr} - F_T\|^2\right) < \|\nabla \varphi_{tr} - F_T\|^2 + \varepsilon^2 \Rightarrow \dot{V}_{tr} \leq 0$

Hence, LaSalle's Invariance Principle, guarantees convergence of q to the largest invariant set contained in the set $\mathcal{P} = \{q \,|\, \|\nabla \varphi_{tr} - F_T\| \leq \varepsilon_1\}$. We have assumed that the system's initial conditions are in the set $\{\mathcal{E} \bigcup \mathcal{I}\} \backslash P_1$. Since the set P_1 is repulsive, it holds that $\dot{V} \overset{a.e.}{<} 0$. ∎

Remark 1: In practice we can choose an ε_1, such that $\varepsilon_1 < \min\{\varepsilon_0, \|\nabla \varphi_{tr}(q_0, t_0)\|\}$, so we can be sure that the system's initial conditions are not in $\mathcal{P}$.

VI. 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 PUMA-560 Unimate, with $m = 6$ d.o.f. The scenario of the simulation contains two 3-D (ellipsoid) obstacles centered at $\mathcal{O}_1 : (-0.2, -0.5, 0.0)$ and $\mathcal{O}_2 : (0.2, -0.4, -0.5)$. The surface of interest $g(s_1, s_2)$ is assumed to be an ellipsoid, centered at $(0, 0, 0,)$ with semi-axes lengths $(0.5, 0.3, 0.2)$, and uniform stiffness $K_e = 10^3 N/m$. In order to be able for the robot's end-effector to apply a constant force in the perpendicular to the surface direction, $F^d = 5N$, we have adjusted the displacement from the surface's boundary, $\epsilon_d^\perp = 5 \cdot 10^{-3} m$. The "bad" region's obstacles are centered at $\mathcal{O}_{g1} : (-0.25, 0.15, 0.14)$, $\mathcal{O}_{g2} : (0.25, 0.15, 0.14)$ and $\mathcal{O}_{g3} : (-0.25, -0.15, -0.14)$. The robot manipulator's initial configuration was $p(0) = (-0.33, -0.41, -0.08, 0.0, 0.0, 1.5)$, the first target was set at $p_{d_1} = (0, 0, 0, 0, 0, 0)$ and the second target was set at $p_{d_2} = (0.2475, -0.1475, -0.1375, 1.03, 0.50, -1.03)$. After the end-effector of the robot manipulator reaches its destination point (p_{d_2}), it starts a tracking task, to track a predefined trajectory which is the yellow colored line in Fig. 5-6, and the same time to apply the specific force profile on the surface.
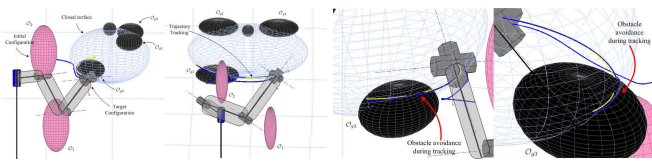
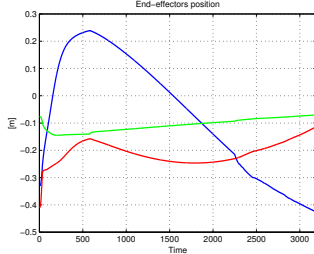Fig. 5. *Simulation Results:* reaching a point on the surface and tracking.

Fig. 6. *Simulation Results:* collision avoidance during tracking.

Fig. 7. *Simulation Results:* End-Effector's (cartesian) position, blue - x, red - y, green -z.
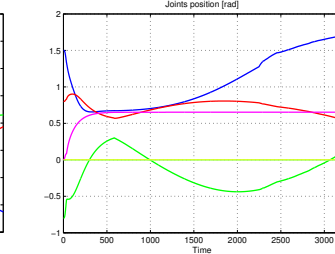
Fig. 8. Joint's angles, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6.
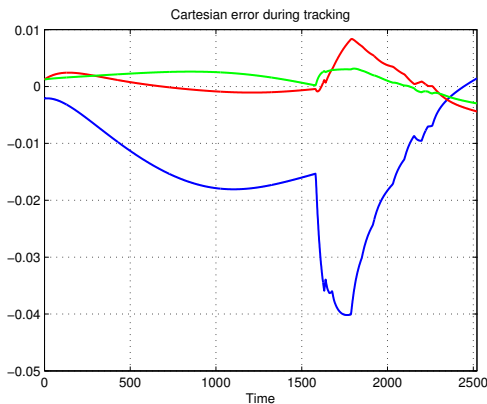

Fig. 9. Cartesian position error between the real position of the robot and the predefined trajectory during trajectory tracking, which is bounded by $\epsilon = 0.01$, blue - x, red - y, green - z .

The scenario of the simulation include a path, in a part of which, it penetrate a surface obstacle, Fig. 5-9. Figures 5-6, present the results from a different point of view, since there are 3-D view simulations.

Figures 7-8, present the cartesian and joint position. Our algorithm successfully converges to the goal configuration and track the predefined trajectory avoiding obstacles. More specifically, in Fig. 6,9 is depicted clearly, that when the trajectory is passing through an obstacle, the robot can avoid it by leaving the trajectory until this is out of the obstacle again.

VII. CONCLUSIONS AND FUTURE WORK

We presented a methodology for performing navigation and tracking tasks over a 2-dimensional manifold embedded in a 3-dimensional workspace, with concurrent force control across the surface, applicable to articulated robotic manipulators. 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, and the same time these fields make the robot complaint to the 2-D manifold. The methodology has theoretically guaranteed global convergence and collision avoidance properties. Due to the closed form of the dynamic feedback controller, the methodology is particularly suitable for implementation on real time systems with limited computation capability.

Further research includes the implementation of the methodology to real neuro-robotic systems taking into account their dynamics and kinematic constraints. Furthermore, another future idea is to restate the whole problem in a spatially "distributed" fashion as the manipulator will be interacting in multiple points/surfaces with the environment.

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] N. Hogan, "Impedance control: An approach to manipulation; part i: Theory; part ii: Implementation; part iii: Applications," *ASME Journal of Dynamic Systems, Mearurement and Control*, vol. 107, no. 1, pp. 1–24, 1985.

[5] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Mearurement and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[6] B. Siciliano and L. Villani, *Robot Force Control*. Kluwer Academic Publishers, 1999.

[7] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.

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

[9] 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.

[10] 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.

[11] 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)*, 2006.

[12] 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.

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

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

[15] 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.

[16] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. Wiley-Interscience, 1995.

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

[18] S. Loizou, D. Dimarogonas, and K. Kyriakopoulos, "Decentralized feedback stabilization of multiple nonholonomic agents," *2004 IEEE International Conference on Robotics and Automation*, 2004.