Motion Tasks for Robot Manipulators Subject to Joint Velocity Constraints

Xanthi Papageorgiou 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 presence 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. One of the main tasks of neuro-robotics is to make a robot execute a task by interfacing with the neural system e.g., by processing electromyographic activity, etc.

Previous 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.

Also, there are several applications introducing only local methods for redundant arm collision avoidance based on cartesian subtasks priorization, but without any requirement for surface's tasks. The authors in [4] develop the real-time collision avoidance for position-controlled dexterous 7 DOF arms. The problem is formulated and solved as a position-based force control problem by using virtual forces that represent the intrusion of the arm into the obstacle safety zone. In [5] a collision avoidance and a self-collision avoidance scheme for redundant manipulators is discussed. The method is based on modeling the arm and its environment by simple geometric primitives (cylinders and spheres). The authors in [6] present a neural network approach for collision-free motion control of redundant manipulators. The problem formulation represents the collision free requirement as dynamic inequality constraint and incorporates joint physical limits into an optimization problem. The solution of the obstacle-avoidance kinematics problem, is computed by a dual neural network.

In our previous work [7], 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. For the problem of kinematic input constraints, a non-smooth kinematic controller is proposed in [8].

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, in a simple way without any need of non-smooth stability analysis, [8]. The system switches between different controllers when it finds itself within certain regions of the workspace (called belt zones, [9]). 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 or point stabilization with obstacle avoidance on 2D manifolds embedded in 3D workspace.

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

This research project is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%). 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 are with the Department of Mechanical Engineering Department, National Technical University of Athens, Athens, Greece, xpapag@mail.ntua.gr; kkyria@mail.ntua.gr

Proc. IEEE/RSJ Int'l Conf. Intelligent Robots & Systems (IROS-2008), Acropolis Convention Center, Nice, France, Sept, 22-26, 2008, p.p. 2139-2144.

move towards the surface, and track a predefined trajectory on it, while the end-effector is compliant with the surface. We assume that we have a stationary environment and that we have direct control on the manipulator joint rates. Thus the robot can be dynamically described by a set of integrators: $B(q) \cdot \ddot{q} + C(q, \dot{q}) + Gr(q) = \tau$, where $B(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis term, $Gr(q)$ is the Gravity term, $q = [q_1 \ \dots \ q_m]^T \in \mathbb{R}^m$ is the vector of arm joint variables, and τ the joint torque inputs [10]. Using inverse dynamics control (since it is based on computation of manipulator inverse dynamics) $\tau = B(q) \cdot y + C(q, \dot{q}) + Gr(q)$, the system becomes

$$\ddot{q} = y \tag{1}$$

where y represents a new input vector. The system (1) is linear and decoupled with respect to the new input y. In other words, the component y_j influences, with a double integrator relationship, only the joint variable q_j, independently of the motion of the other joints, where $j = 1, \dots, m$.

Detailed modeling and accurate parameter identification is necessary for a successful implementation of a computed torque approach. In our case, this process involves the use of a non-linear friction model for the robot joints, coupled with experimental parameter identification [11], [12]. The stiffness effect of the joints is also identified through experimental procedure in loaded conditions. The parameters of the dynamic model are grouped to an identifiable form, and identified through experiments.

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 $\mathcal{W}_{free} \subseteq \mathcal{W}$. Define a vector valued C^2 function $g(s_1, s_2) : \mathbb{R}^2 \rightarrow \mathcal{R}(g)$ 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 the robot task is to take place. Let us decompose the space around this surface as G^-, G^+, ∂g the internal, the external, and the boundary of the surface, respactively (Fig. 2).

We can define the tangent vectors on the surface $g_{s_1}(s_1, s_2) = \frac{\partial g(s_1, s_2)}{\partial s_1}$, and $g_{s_2}(s_1, s_2) = \frac{\partial g(s_1, s_2)}{\partial s_2}$, with respect to parameters s_1 and s_2. Due to the C^2 continuity of $g(s_1, s_2)$, we have that $(g_{s_1} \times g_{s_2}) \neq 0, \ \forall s_1, s_2 \in \mathbb{R}$, [13] (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). A normalized vector, perpendicular to the surface is then expressed as $N = \frac{g_{s_1} \times g_{s_2}}{\| g_{s_1} \times g_{s_2} \|}$.

The problem is stated as follows: *Given a redundant revolute joint robot manipulator, with kinematic input constraints, operating in a known static and bounded environment, find a feedback dynamic control law that allows the end-effector of the manipulator to*

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

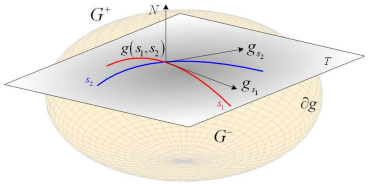

Fig. 2. Representation of tangent's and perpendicular's vectors, of a surface, variations w.r.t parameter's modification.

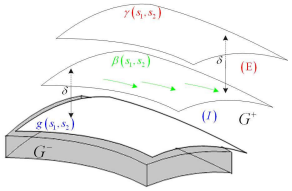

Fig. 3. Representation of Belt Zones, in a part of a surface.

III. Controller Design

A. Workspace Decomposition

The task is completed in two stages. In the first mode, mode $\mathcal{A}$, the end-effector is driven close to the surface. In the second mode, mode $\mathcal{B}$, the robot is steered to a specified point on the surface, or is controlled to track a reference trajectory.

This workspace decomposition requires the definition of a region in which the transition from the one mode to the other occurs. Towards this end, we use the concept of belt zones [7], Fig. 3. The "belt zone" is the region close to the ∂g, consisting of an "internal belt" and an "external belt" region. We assume that the widths of the internal and external belt regions are fixed.

Let us define the vector valued bijective functions that describe the belt zones

$$\beta(s_1, s_2) = g(s_1, s_2) + \delta \cdot N \tag{2}$$
$$\gamma(s_1, s_2) = \beta(s_1, s_2) + \delta \cdot N \tag{3}$$

with $0 < 2 \cdot \delta < \rho_m$, where ρ_m as described in [9]. Surface processing tasks require stabilization of the end-effector on the surface $\beta(s_1, s_2)$, defined by (2).

"Internal belt", $\mathcal{I}$, and the "external belt", $\mathcal{E}$ are defined as (Fig. 3)

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

Since functions g, β, γ are bijective [9], [14], for every $k(q) \in \mathcal{E} \bigcup \mathcal{I}$ there is a unique couple (s_1, s_2), where $k(q)$ is the vector of robot's position in operational space (direct kinematics).

B. Navigation Function

The controller's design is based on the navigation function, [15]. Modes $\mathcal{A}$ and $\mathcal{B}$ use a different navigation function. The navigation function active in mode $\mathcal{A}$ brings the end-effector inside the belt zone. Then the system switches to mode $\mathcal{B}$, and another controller is activated, enabling the

end-effector to navigate on the surface. Navigation on the surface may involve stabilization to a particular point, or tracking a reference trajectory on the surface. This last case is the general one, where the desired configuration is time varying.

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, $q_d \in \mathbb{R}^m$. Redundancy is resolved by calculating joint rates that implement the desired trajectory in operational space, [8].

1) Mode $\mathcal{A}$: The navigation function $\varphi_{\mathcal{A}} : \mathcal{W} \to \mathbb{R}$ used in mode $\mathcal{A}$, is defined as follows:

$$\varphi_{\mathcal{A}}(z) = \frac{\gamma_{\mathcal{A}}^{\kappa+1}(z)}{\left(\gamma_{\mathcal{A}}^{\kappa}(z) + \beta_{\mathcal{A}}(z) \cdot \beta_{\mathcal{O}}(z) \cdot \beta_s(z) \cdot \beta_b(z)\right)^{\frac{1}{\kappa}}} \quad (4)$$

where $\gamma_{\mathcal{A}}(z) = \|z\|^2$ is the distance to goal function, and $\beta_{\mathcal{A}}(z) = -\|q - q_0\|^2 + r_0^2$ provides the workspace potential, 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 $\beta_{\mathcal{O}}(z)$, as defined in [7], that represents a measure of proximity of the robot to the obstacles. The function $\beta_s(z)$ and $\beta_b(z)$ represent the virtual obstacles, in order to achieve singularities avoidance, according to [7], and in order to avoid the joint's limits, respectively. Finally, $\kappa > 0$ is a parameter.

2) Mode $\mathcal{B}$: We need to define a navigation function across the 2-D manifold, 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.

This navigation function is analytically expressed as:

$$\varphi_{\mathcal{B}}(z,t) = \frac{\gamma_{\mathcal{B}}^{\kappa+1}(z,t)}{\left[\gamma_{\mathcal{B}}^{\kappa}(z,t) + \beta_{\mathcal{B}}(z,t) \cdot \beta_{\mathcal{O}}(z,t) \cdot \beta_s(z,t) \cdot \beta_b(z,t)\right]^{1/\kappa}} \quad (5)$$

and it can be time-varying depending on whether the surface task is point stabilization or trajectory tracking. The functions $\beta_{\mathcal{O}}$, β_s and β_b are exactly the same functions as in case of $\varphi_{\mathcal{A}}$, and $\kappa > 0$ is a parameter. The function $\gamma_{\mathcal{B}}$ is defined as

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

where $h(q)$ is the distance from the surface $g(s_1, s_2)$ on the belt zones. Additionally, 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 on the surface $\beta(s_1, s_2)$. Thus, the second term on this vectors is used to attract the end-effector to the surface β.

Also, the function $\beta_{\mathcal{B}}$ is called the "perpendicular" workspace function which is given from the equation

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

This construction of $\beta_{\mathcal{B}}$ guarantees that the robot's end-effector cannot leave the belt zone.

C. Vector Fields - Controller Synthesis

The system is defined to operate in mode $\mathcal{A}$ when $p \in \mathcal{W}_{ext}$, where $\mathcal{W}_{ext} = \{\mathcal{W}_{free} \bigcap G^+\} \setminus \{\mathcal{E} \bigcup \mathcal{I}\}$, with $p = k(q)$ the manipulator's direct kinematics. Otherwise, if $p \in \mathcal{E} \bigcup \mathcal{I}$ the system is operating in mode $\mathcal{B}$.

Assume that the robot's initial configuration is $q(0) \in \mathbb{R}^m$, with $p(0) = k(q(0)) \in G^+$, and by construction it holds that $p_d \in G^-$, the solutions of (8), 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.

We define the following vector fields for each mode:

$$\begin{aligned} f_{\mathcal{A}}(z) &\triangleq -k_1 \cdot \nabla\varphi_{\mathcal{A}}(z) \\ f_{\mathcal{B}}(z,t) &\triangleq -k_2 \cdot \nabla\varphi_{\mathcal{B}}(z,t) - k_3 \cdot D \end{aligned} \quad (6)$$

with $D = \frac{\nabla\varphi_{\mathcal{B}}}{\nabla\varphi_{\mathcal{B}}^T \cdot (u_{\max} \circ \nabla\varphi_{\mathcal{B}})} \cdot \frac{\partial\varphi_{\mathcal{B}}}{\partial t}$, where $u_{\max}$ is the vector of maximum joint velocity values, the $\circ$ operator is the Hadamard product of the vectors (entry-wise product), and $k_1, k_2, k_3 > 0$ are constant parameters.

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, inspired by [17]:

$$f^{new} = \frac{u_{\max} \circ f_i}{\nu + \|f_i\|} \quad (7)$$

where $\nu > 0$ is a parameter, with $i = \{\mathcal{A}, \mathcal{B}\}$ for each mode of operation, respectively.

Corollary 1: The control law defined in (7) satisfies the input constraints $|u^j| \leq |u^j_{\max}|$, for every $j = 1, \ldots, m$.

Proof: By using (7), we have for the j component of u that: $|u^j| = |f_j^{new}| = u^j_{\max} \cdot \frac{|f_i^j|}{\nu + \|f_i\|} \leq u^j_{\max}$, since it holds that $|f_i^j| \leq \nu + \|f_i\|$, $\forall \nu > 0$. ■

IV. Stability Analysis

We design the controller that renders (1), asymptotically stable for the general case of time varying system. 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.

Our main goal is to develop motion controllers that enable redundant manipulators to perform surface tasks. The dynamic representation of the system is given by (1). For the dynamic control of this system we use the backstepping methodology, and to do so, we need to establish the closed loop control for the kinematic subsystem [18]

$$\dot{q} = u \tag{8}$$

where u is the kinematic control input. The control of this subsystem needs to satisfy the Assumption 1 (see Appendix) for the kinematic controller which compose an intermediate step to the dynamic controller synthesis. Thus, we need first to show the asymptotic stability for the kinematic subsystem.

A. Kinematic Controller Design

***Proposition** 1:* The solution of the system $\dot{z} = v$, where $z = q - q_d$, and $v = u - \dot{q}_d$, with u the input of (8), under the control law $v = f^{new}(z,t)$ as is defined in (7) in the general case of $f_{\mathcal{B}}(z,t)$ as is defined in (6), is uniformly asymptotically stable almost everywhere[1].

B. Dynamic Controller Design

***Proposition** 2:* Consider the system $\ddot{z}(t) = U(t)$, where $z(t) = q - q_d(t)$, and $U = y - \ddot{q}_d(t)$, with y the control input of (1). This system becomes uniformly asymptotically stable to zero a.e.[1], under the control input

$$U = -c \cdot (\dot{z} - f^{new}) + \dot{f}^{new} - \nabla\varphi_{\mathcal{B}} \tag{9}$$

where f^{new} as is defined in (7) in the general case of $f_{\mathcal{B}}(z,t)$ as is defined in (6), and c is a positive definite constant matrix.

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. 1, with $m = 7$ d.o.f.

The vector of joint velocity limitation in rad/sec, is $u_{\max} = \begin{bmatrix} 0.6 & 0.6 & 1.2 & 1.2 & 1.2 & 1.2\pi & 1.2\pi \end{bmatrix}^T$. The scenario of the simulation involves two 3-D (ellipsoid) obstacles centered at $\mathcal{O}_1 : (-0.3, -0.4, 0.1)$ and $\mathcal{O}_2 : (0.35, -0.3, -0.5)$ both having semi-axes lengths of $(0.05, 0.10, 0.20)$ (referred to x, y, and z dimensions, respectively). 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 obstacle regions on the surface ("bad regions") $g(s_1, s_2)$ 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 manipulator's initial end-effector's configuration is $p_0 = [-0.61, -0.39, -0.13, 0.0, 0.0, 0.0]$, and 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 first three components of these vectors are the x, y, and z coordinates, and the last three are the euler angles describing the end-effector orientation.

In the first part of the simulation scenario the robot moves to position its end-effector to the desired configuration close to the surface, while each part of the robotic arm avoids collisions with obstacles. In the second part of the experiment the goal is to make the end-effector track a predefined trajectory, avoiding collisions with obstacles. During operation, we monitor joint rates to ensure that velocity constraints are respected.

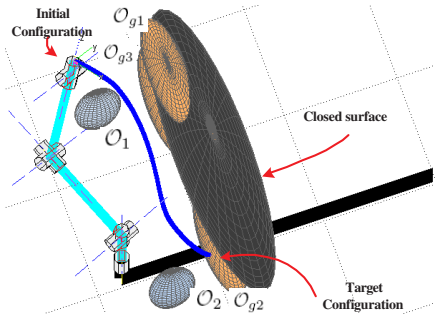

Fig. 4. Simulation results during point stabilization.

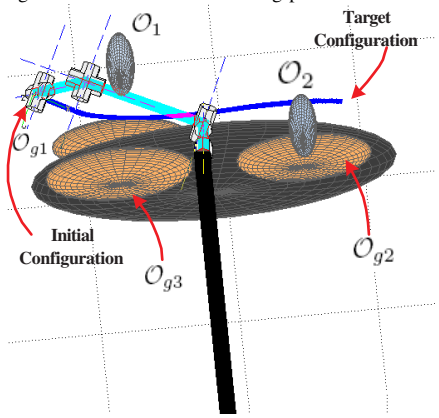

Fig. 5. Simulation results during point stabilization.

Fig. 4 - 7 depict the simulation results, showing how the robot manipulator reaches the desired configuration and tracks the predefined trajectory, with simultaneous obstacle avoidance. Fig. 8 presents the cartesian position. Fig. 9 depicts the joint velocity. It is evident that the robot's kinematic constraints are respected. The time axis records time steps. Each time step is 2.5 miliseconds. Fig. 10 gives the error between the real cartesian position and the desired position during tracking, for each axis, measured in meters. At the beginning, when the end-effector is away from the desired configuration, the maximum error is in x axis, and it is about 0.19 m, and the robot accelerates to get in position to track the reference trajectory. This error is then bounded below 1 cm, in all axis x, y, z.

VI. Conclusion and Future Work

We presented a methodology for performing navigation and tracking tasks over a 2-dimensional manifold embed-

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

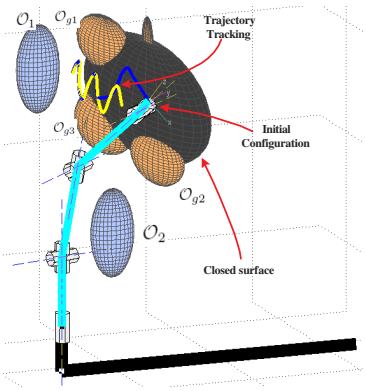

Fig. 6. Simulation results during trajectory tracking.

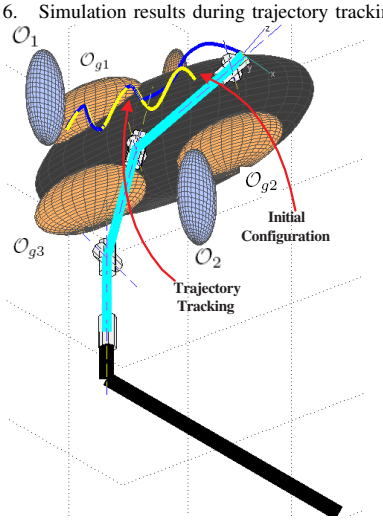

Fig. 7. Simulation results during trajectory tracking.

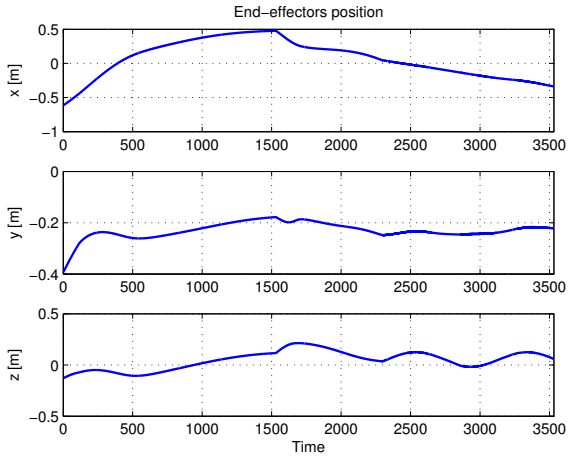

Fig. 8. Cartesian position.

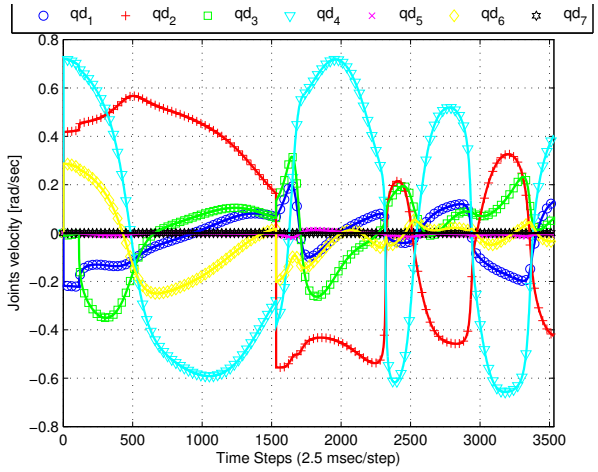

Fig. 9. Joint's velocities, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6, black - q_7.

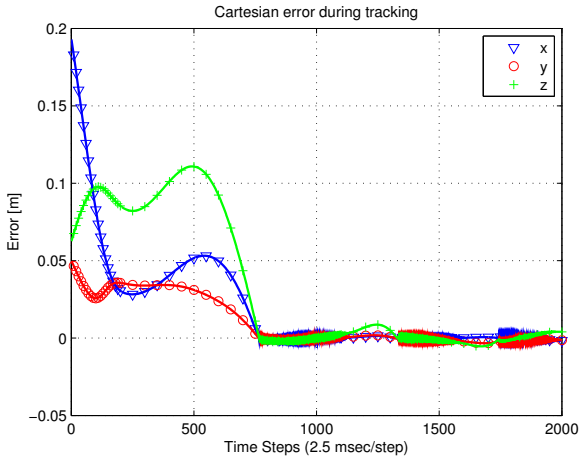

Fig. 10. Cartesian position error between the real position of the robot and the predefined trajectory during trajectory tracking, blue - x, red - y, green - z.

ded 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.

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

Assumption 1: [18] Consider the system (8) where $q \in \mathbb{R}^m$ is the state and $u \in \mathbb{R}$ is the control input. There exist a continuously differentiable feedback control law $u = f(q)$, $f(0) = 0$, and a smooth, positive definite, radially unbounded function $V : \mathbb{R}^m \to \mathbb{R}$ such that $\dot{V} \leq -W(q) < 0, \forall q \in \mathbb{R}^m$, where $W(q) : \mathbb{R}^m \to \mathbb{R}$ is positive definite. Then the solution of (8) is globally asymptotically stable.

A. Proofs of Propositions

Proof: (*Proposition 1*) We set a time-varying, continuously differentiable Lyapunov function candidate $V(z,t) \triangleq \varphi_{\mathcal{B}}(z,t)$ (see (5)). As it is proved in [8], it holds that $V_1(\|z\|) \leq V(z,t) \leq V_2(\|z\|)$.

We examine the time derivative of $V(z,t)$:
$\dot{V} = \frac{\partial V}{\partial t} + \nabla V^T \cdot f^{new} = \frac{\partial V}{\partial t} + \nabla V^T \cdot \frac{u_{\max} \circ f_{\mathcal{B}}}{\nu + \|f_{\mathcal{B}}\|} = -\frac{k_2}{\nu+\|f_{\mathcal{B}}\|} \cdot \nabla V^T \cdot (u_{\max} \circ \nabla V) + \frac{\partial V}{\partial t} \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right) = -E + \frac{\partial V}{\partial t} \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)$, where $E = \frac{k_2}{\nu+\|f_{\mathcal{B}}\|} \cdot \nabla V^T \cdot (u_{\max} \circ \nabla V) \geq 0$, $\forall q \in \mathbb{R}^m$.

The properties of the navigation function ensure boundedness of its gradient within the workspace, and therefore ensure the existence of a positive bound $k_4 = \max_{\mathcal{W}/\mathcal{M}} \|f_{\mathcal{B}}\|$, 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 choose $0 \leq k_3 \leq \nu$, $\forall \nu > 0$, and thus it holds that $\left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right) \geq 0$.

Also, it holds that $\frac{\partial V}{\partial t} = -\frac{\gamma_{\mathcal{B}}^{\kappa+1}}{\kappa \cdot \left(\gamma_{\mathcal{B}}^{\kappa}+B\right)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial B}{\partial t} + \frac{(\kappa+1)\cdot\gamma_{\mathcal{B}}^{\kappa}}{(\gamma_{\mathcal{B}}^{\kappa}+B)^{\frac{1}{\kappa}}} \cdot \frac{\partial \gamma_{\mathcal{B}}}{\partial t} - \frac{\gamma_{\mathcal{B}}^{2\cdot\kappa}}{(\gamma_{\mathcal{B}}^{\kappa}+B)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial \gamma_{\mathcal{B}}}{\partial t}$, where $B = (\beta_{\mathcal{B}} \cdot \beta_{\mathcal{O}} \cdot \beta_s \cdot \beta_b)$. It can therefore be bounded as follows: $\left|\frac{\partial V}{\partial t}\right| < \frac{1}{\kappa}\left|\frac{\partial B}{\partial t}\right| + (\kappa+2) \cdot \gamma_{\mathcal{B}}^{\kappa-1} \cdot \left|\frac{\partial \gamma_{\mathcal{B}}}{\partial t}\right|$. Thus, let $P = \frac{1}{\kappa} \sup_{\mathcal{W}/\mathcal{M}} \left|\frac{\partial B}{\partial t}\right| + (\kappa+2) \cdot \max\left(\gamma_{\mathcal{B}}^{\kappa-1}\right) \cdot \sup_{\mathcal{W}/\mathcal{M}} \left|\frac{\partial \gamma_{\mathcal{B}}}{\partial t}\right|$, in which $\sup_{\mathcal{W}/\mathcal{M}} \left|\frac{\partial B}{\partial t}\right|$, and $\sup_{\mathcal{W}/\mathcal{M}} \left|\frac{\partial \gamma_{\mathcal{B}}}{\partial t}\right|$ depends on $\sup_{\mathcal{W}/\mathcal{M}} \|\dot{q}_d\|$.

Thus, it holds: $\dot{V} \leq -E + P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)$. For appropriate choice of the parameter k_2, such that $k_2 \geq \frac{\nu+k_4}{\|u_{\max}\|_{-\infty}}$, it is true that $E \geq \|\nabla V\|^2$, and therefore $\dot{V} \leq -\|\nabla V\|^2 + P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)$. In the region where $\|\nabla V\| > \sqrt{P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)}$ the Lyapunov function is decreasing, and therefore, z converges to 0 which corresponds to the destination configuration ($q_d(t)$). Thus, z is uniformly ultimately bounded in the region where the above condition holds.

In the neighborhood of $z = 0$, ∇V does not vanish except for $z = 0$, since V is defined to be equal to a navigation function of z. Thus, $\|\nabla V\|$ is a positive definite scalar function, and thus there exist $V_3(\|z\|)$, $V_4(\|z\|)$ class $\mathcal{K}$ functions for which $V_3(\|z\|) \leq \|\nabla V\| \leq V_4(\|z\|)$. Using the lower bounding function V_3, if $\|z\| \geq V_3(\|z\|)^{-1} \cdot \sqrt{P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)} \in \mathcal{K}$, then, for the gradient of V we can write $\|\nabla V\| \geq V_3(\|z\|) \geq \sqrt{P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)}$, implying that $\dot{V}$ is strictly negative in the region defined above.

Application of a Theorem in [16], ensures that z is globally uniformly ultimately bounded. ■

Proof: (*Proposition 2*) The control law construction and the proof structure are inspired by the backstepping controller design proposed by [18].

We form the Lyapunov function candidate: $V_a(z,t) \triangleq V(z,t) + \frac{1}{2} \cdot (\dot{z} - f^{new})^2$, where $V(z,t) = \varphi_{\mathcal{B}}(z,t)$, with $\varphi_{\mathcal{B}}$ as defined in (5) (general case of time varying system). Taking the time derivative of V_a, $\dot{V}_a = \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial z}\right)^T \cdot \dot{z} + \left(U - \dot{f}^{new}\right)^T \cdot (\dot{z} - f^{new})$, and substituting U from (9), we have that $\dot{V}_a = -c \cdot (\dot{z} - f^{new})^T \cdot (\dot{z} - f^{new}) - E + \frac{\partial V}{\partial t} \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)$, in the same notation of the proof of Proposition 1, and it holds that $\dot{V}_a \leq -c \cdot (\dot{z} - f^{new})^T \cdot (\dot{z} - f^{new}) - E + P \cdot \left(1 - \frac{k_3}{\nu+\|f_{\mathcal{B}}\|}\right)$. Thus, we conclude that $\dot{V}_a$ is negative definite. ■

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] H. Seraji and B. Bon, "Real-time collision avoidance for position-controlled manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, 1999.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[11] 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.

[12] 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.

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

[14] 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.

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

[17] S. Loizou and K. Kyriakopoulos, "Motion planning of piezoelectrically driven micro-robots via navigation functions," *2005 ISIC-MED*, 2005.

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