Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Compliant Motion Tasks for Robot Manipulators Subject to Joint Velocity Constraints

Xanthi Papageorgiou and Kostas J. Kyriakopoulos

Abstract— **We develop non-smooth motion controllers that enable redundant manipulators to perform surface tasks, that involve force/torque interactions. The robot joint rates are constrained within given limits. The end-effector is stabilized to a point, on the surface, and then tracks a trajectory on the surface, while respecting the input constraints and avoiding obstacles, and the same time applying a specific force on it. We show that the resulting closed loop system is uniformly asymptotically stable, and we verify our analytical development with simulations on a redundant robotic manipulator model.**

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.

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. At the other hand literature is 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 presence of holes on the surface, i.e. regions of the surface that the robot tip must avoid.

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 [8] 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 [9] 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 [10] 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 [11], 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. Also, in [12] the force control is added in those surface tasks. For the problem of kinematic input constraints, a non-smooth kinematic controller is proposed in [13].

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, respecting joint's limitations. We do so by building navigation functions, and analyzing the closed loop system that has saturated inputs by means of non-smooth stability analysis. The system switches between different controllers when it finds itself within certain regions of the workspace (called belt zones, [14]). The contribution of this paper is the development of globally uniformly asymptotically stable compliant controllers for redundant articulated robot manipulators, subject to input constraints, to achieve reference trajectory tracking or point stabilization with obstacle avoidance and force control on 2-D manifolds embedded in 3-D 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 to make the robot 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 and known environment and that we have direct control on the manipulator joint

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

rates. Thus, the robot dynamics are: $B(q)\ddot{q} + C(q,\dot{q}) + Gr(q) + J^T(q)F = \tau$, 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 (we can measure this vector by using a force/torque sensor at the robot's end-effector), $q = [q_1 \ \dots \ q_m]^T \in \mathbb{R}^m$ is the vector of arm joint variables and $\tau \in \mathbb{R}^m$ the joint torque inputs, [15]. Using inverse dynamics control (since it is based on computation of manipulator inverse dynamics) $\tau = B(q)\cdot y + C(q,\dot{q}) + Gr(q) + J^T(q)F$, 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 [16], [17]. 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.

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 $(g_{s_1} \times g_{s_2}) \neq 0$, $\forall s_1, s_2 \in \mathbb{R}$, [18] (the vectors g_{s_1}, g_{s_2} are linearly independent everywhere). A normalized vector, perpendicular to the surface is then expressed as $N = (g_{s_1} \times g_{s_2}) / \|g_{s_1} \times g_{s_2}\|$.

The problem is stated as follows: *Given a redundant revolute joint robot manipulator, with kinematic input constraints (joint velocity constraints), operating in a known static and bounded environment, find a feedback dynamic control law that allows the end-effector of the manipulator to execute surface tasks as 1) navigation to any feasible surface point, and 2) tracking a predefined trajectory across the surface, and at the same time to be compliant with the surface.*

III. Controller Design

A. Workspace Decomposition

First of all we would like to drive the robot's end-effector towards the surface, and then to control it in order to execute the desired surface task. To this end we can say that the task is completed in two stages. In the first mode $\mathcal{A}$, the end-effector is driven close to the surface. In the second mode $\mathcal{B}$, the robot is steered to a specified point on the surface, or is controlled to track a reference trajectory, and the same time is applying a specific force on the surface.

Fig. 1. The problem motivation (main tasks of neuro-robotics).

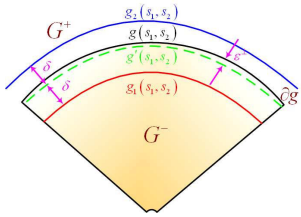

Fig. 2. Representation of Belt Zones, in a part of a surface.

Thus, we need to decompose the workspace. This workspace decomposition requires the definition of a region in which the transition from the one mode to the other occurs. To do so, we use the concept of belt zones [11], Fig. 2. The "belt zone" is the region close to the surface's boundary, consisting of an "internal belt" and an "external belt". We assume that the widths of the internal and external belt regions are 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. 2). Let $F_d \in \mathbb{R}$ is the desired force that we would like the robot apply perpendicular to the surface, that denote a displacement from the boundary of the surface $\epsilon_d^\perp = K_e^{-1} F_d$.

Let us define the vector valued bijective functions that describe the belt zones $g'(s_1,s_2) = g(s_1,s_2) - \epsilon_d^\perp N$, $g_1(s_1,s_2) = g(s_1,s_2) - (\delta + \epsilon_d^\perp)N$, and $g_2(s_1,s_2) = g(s_1,s_2) + (\delta - \epsilon_d^\perp)N$, with $0 < 2\cdot\delta < \rho_m$, where ρ_m as described in [14]. Surface processing tasks require stabilization of the end-effector on the surface $g'(s_1,s_2)$, defined above.

The internal and external belts are defined as (Fig. 2), $\mathcal{I} = \{q : k(q) = (1-\lambda)g' + \lambda g_1, \lambda \in [0,1]\}$, and $\mathcal{E} = \{q : k(q) = (1-\lambda)g' + \lambda g_2, \lambda \in (0,1]\}$. Since functions g, g', g_1, g_2 are bijective ([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, [19]. Modes $\mathcal{A}$ and $\mathcal{B}$ use a different navigation function, because of the motion's nature. 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 (time invariant destination), or tracking a reference trajectory (time varying destination) on the surface, and the same time applying a specific force on it by the robot end-effector.

The volume of the manipulator is represented by a point, using a series of transformations. The obstacles presented in the environment are modeled by the navigation function. In order to construct such function, we need to introduce the following parameter $z = q - q_d$, with $q, q_d \in \mathbb{R}^m$, which is the error between the manipulator's joint angles, and the desired joint angles. Redundancy is resolved by calculating joint rates that implement the desired trajectory in operational space (see Appendix).

1) Mode $\mathcal{A}$: The navigation function $\varphi_{\mathcal{A}} : \mathcal{W}_{ws} \rightarrow [0,1]$ used in mode $\mathcal{A}$, is defined as follows: $\varphi_{\mathcal{A}}(z) = \frac{\gamma_{\mathcal{A}}^{\kappa+1}(z)}{\left[\gamma_{\mathcal{A}}^{\kappa}(z)+\beta_{\mathcal{A}}(z)\beta_{\mathcal{O}}(z)\beta_s(z)\beta_b(z)\right]^{\frac{1}{\kappa}}}$, 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 angles 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 [11], that represents a measure of proximity of the robot to the obstacles. The functions $\beta_s(z)$ and β_b represent the virtual obstacles, in order to achieve singularities avoidance, according to [11], 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: $\varphi_{\mathcal{B}}(z,t) = \frac{\gamma_{\mathcal{B}}^{\kappa+1}(z,t)}{\left[\gamma_{\mathcal{B}}^{\kappa}(z,t)+\beta_{\mathcal{B}}(z,t)\beta_{\mathcal{O}}(z,t)\beta_s(z,t)\beta_b(z,t)\right]^{1/\kappa}}$, 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 $g_1(s_1, s_2)$ on the belt zones. Additionally, we have $h = 0$ on the surface defined by g_1 (boundary of internal region), and $h_{ext} = 2\delta$ on the surface defined by g_2 (boundary of external region). Also, the desired distance from the surface $g(s_1, s_2)$ is at $h_d = \delta$, on the surface $g'(s_1, s_2)$, at $\epsilon_d^{\perp}$ distance inside of g. Thus, the second term on this vectors is used to attract the end-effector to the surface g_1.

Also, the function $\beta_{\mathcal{B}}$ is called the "perpendicular" workspace function which is given from the equation $\beta_{\mathcal{B}}(z) = \frac{(h_{ext}-h_d)^2-(h(q)-h_d)^2}{(h_{ext}-h_d)^2}$. It holds that $\beta_{\mathcal{B}}(z) = 0$ when the robot's end-effector is in the outer boundary of the belt zone's external region (where $h(q) = 2\delta$), or in the inner boundary of the belt zone's internal region (where $h(q) = 0$). Also, $\beta_{\mathcal{B}}(z) = 1$ in the middle surface (where $h(q) = \delta$). Thus, it makes the boundaries of the belt zone repulsive and the middle surface attractive. This construction of $\beta_{\mathcal{B}}$ guarantees that the robot's end-effector cannot leave the belt zone, (Fig. 2).

C. Vector Fields - Controller Synthesis

Assume that the robot's initial position is away from the surface, where the system operates in mode $\mathcal{A}$. The desired position is defined to be in the interior of the belt zone. The controller in this mode drives the robot towards the surface, until the robot's end-effector intersect the external boundary of the belt zone, the surface $g_2(s_1, s_2)$.

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 (simple time invariant case, since we set constant target position), and the trajectory tracking task (time varying case, since we set time varying target position) while it has to apply a specific force on the surface.

We define the following vector field which is applicable for any mode: $f(z,t) \triangleq -k_1 \nabla\varphi_i(z,t) - D$, where $D = \frac{1}{m}\left[\; |\frac{\partial\varphi_i}{\partial z_1}|^{-1} \quad \cdots \quad |\frac{\partial\varphi_i}{\partial z_m}|^{-1} \; \right]^T \frac{\partial\varphi_i}{\partial t}$, for all operational modes $i \in \{\mathcal{A}, \mathcal{B}\}$, and $k_1 > 0$ is constant parameter (e.g., when the robot is operating in mode $\mathcal{A}$ then the desired configuration is time invariant and $D = 0$, which is the simplest case, i.e., $f(z) = -k_1 \nabla\varphi_{\mathcal{A}}(z)$).

In order to compensate the manipulator's kinematics input constraints, we have to construct an appropriate controller for both of the above modes. The controller output is shaped through a non-smooth saturation function $f^{new} \triangleq \mathrm{sat}_{\mu}(f)$, 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, $\mu = u_{\max} - \dot{q}_d(t)$, chosen so that velocity input constraints are satisfied, $u_{\max}$ is the vector of maximum joint velocity values.

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 [20]

$$\dot{q} = u. \tag{2}$$

The above subsystem (kinematic) needs to be asymptotically stable, [20]. Thus, the kinematic controller which compose an intermediate step to the dynamic controller synthesis, is analyzed.

A. Kinematic Controller Design

***Proposition** 1:* Consider the system $\dot{z} = v$, where $z = q - q_d$, and $v = u - \dot{q}_d$, with u the input of (2). The solutions of this system under the control law $v = f^{new}(z,t)$, are globally uniformly ultimately bounded with the bound that can be made arbitrarily small with the suitable choice of k_1.

B. Dynamic Controller Design

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 simple cases, where the controller (and therefore the closed loop system) is time invariant.

***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 law of (1). This system becomes semi-globally uniformly asymptotically stable to zero almost everywhere, under the control input

$$U = \pi_1 + \left[\pi_2 + \text{diag}\left\{\frac{\frac{\partial V}{\partial z}\xi}{\| \xi \|^2}\right\}\right](-\xi) \tag{3}$$

where $\xi = \dot{z} - f^{new}$, $V : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is a regular locally Lipschitz Lyapunov function, $\pi_1 = \arg\min_{\zeta \in \dot{\tilde{f}}^{new}} \zeta^T \xi$ is the vector in $\dot{\tilde{f}}^{new}$ giving the smallest inner product with ξ, while $\dot{\tilde{f}}^{new}$ is the stack vector of the generalized time derivatives of the components of f^{new}, [21], and π_2 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 use 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's velocity and angle limitation in (rad/sec), and (rad) respectively, are $u_{max} = \begin{bmatrix} 0.6 & 0.6 & 1.2 & 1.2 & 1.2 & 1.2\pi & 1.2\pi \end{bmatrix}^T$, and $q_{lim} = \begin{bmatrix} \pm 3.1 & \pm 1.6 & \pm 3.0 & \pm 2.4 & \pm 4.5 & \pm 2.9 & \pm 4.5 \end{bmatrix}^T$. The scenario of the simulation contains two 3D (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 coordinates, 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 $(0.75, 0.25, 0.35)$ and uniform stiffness $K_e = 10^2 N/m$. In order to be able for the robot's end-effector to apply a constant force in the perpendicular to the surface direction, $F^d = 2N$, we have adjusted the displacement from the surface's boundary, $\epsilon_d^{\perp} = 0.02m$.

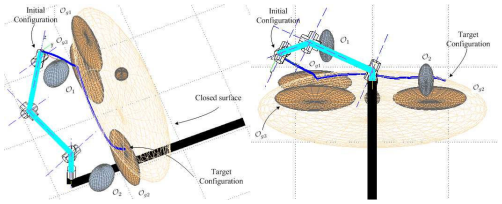

Fig. 3. Simulation results during Point Stabilization.

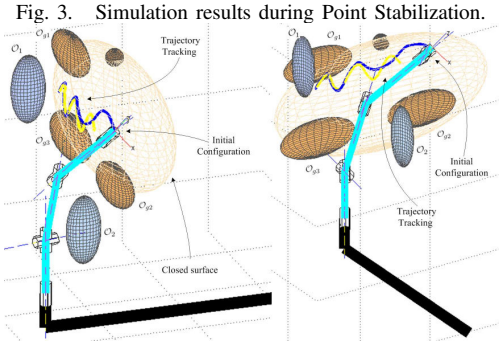

Fig. 4. Simulation results during Trajectory Tracking.

The obstacle regions on the surface 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 end-effector 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 components of those vectors are the x, y, and z coordinates (the first three numbers), and the euler's angles of the end-effector's orientation (the last three numbers).

In the first part of the simulation the robot is starting to move in order for the end-effector to reach its desired configuration (Fig. 3), while each part of the robotic arm successfully avoids any environmental obstacle. The second part of the simulation is to make the end-effector to track a predefined trajectory, while again the collision avoidance with the obstacles takes place (Fig. 4). Another significant issue from this simulation is to avoid the violation of velocity constraints, and angle limitations of each joint of the robot, and to make the robot end-effector complaint to the surface.

Fig. 5, 7, and 8, present the cartesian and joint position as well as the joint's velocity, respectively (the time axis is refered to both point stabilization and trajectory tracking tasks). The time axis represents time steps. Each time step is 2.5 msec. In the representation of the velocities it is clear that the robot's joint velocity constraints are not violated (flat regions). Fig. 6 depicts the error in (m), between the real cartesian position and the desired position during tracking (the time axis is refered only on this task). At the beginning when the end-effector is away from the desired configuration the error is about 0.19 m, and therefore the robot is accelerating to track the reference trajectory and therefore to reduce the error between its position and the desired position. Thus, this error during the simulation is

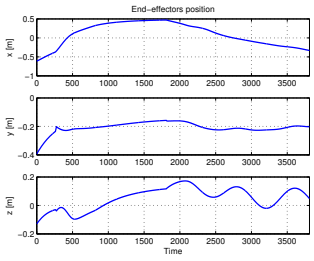

Fig. 5. Cartesian position.

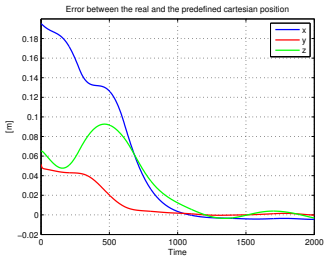

Fig. 6. Error in trajectory tracking.

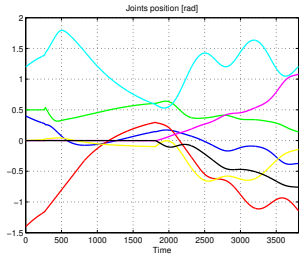

Fig. 7. Joint's angles, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6, black - q_7.

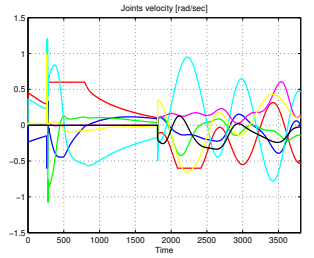

Fig. 8. Joint's velocities, blue - q_1, red - q_2, green - q_3, cyan - q_4, magenta - q_5, yellow - q_6, black - q_7.

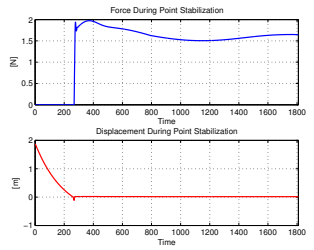

Fig. 9. End-effector's distance from the surface (bottom), and force that the robot produce in its contact with the surface (top), during point stabilization.

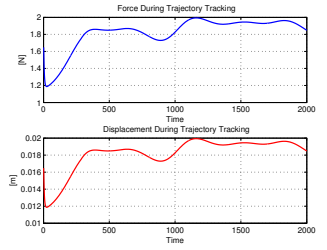

Fig. 10. End-effector's displacement inside the surface, $\epsilon^{\perp}$ (bottom), and the produced force by the end-effector (top), during the trajectory tracking task.

bounded by 1 cm. Our algorithm allows the end effector to successfully converge to the goal configuration, and track the predefined trajectory avoiding obstacles, applying specific force on the surface, and satisfying the input constraints.

Fig. 9 depicts the distance of the end-effector's position from the surface during the point stabilization task (bottom), and the force that the robot end-effector produce when it is in contact with the surface (top). The contact is occurred at the time instant $n = 250$. Fig. 10 present the displacement inside the surface, $\epsilon^{\perp}$ (bottom), and the produced force (top), during the trajectory tracking task (robot motion in contact with the surface). It is obvious that the produced force is kindly dependent on the surface's stiffness. As stiff is the surface, as difficult is to produce a constant force on it.

VI. Conclusion and Future Work

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

The dynamic control approach is using the backstepping methodology, by using the established closed loop control for the kinematic subsystem (2).

Appendix

Theorem 1: [22] Let $\mathcal{D} \subset \mathbb{R}^n$ be a domain that contains the origin and $V : [0, \infty) \times \mathcal{D} \to \mathbb{R}$ be a continuously differentiable function such that $W_1(x) \leq V(t,x) \leq W_2(x)$, $\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t,x) \leq -W_3(x)$, $\forall \|x\| \geq \lambda > 0$, $\forall t \geq 0$, $\forall x \in \mathcal{D}$ where $W_1(x)$ and $W_2(x)$ are class $\mathcal{K}$ functions and $W_3(x)$ is a continuous positive definite functions. Take $r > 0$ such that $\mathcal{B}_r \subset \mathcal{D}$ and suppose that $\lambda < W_2^{-1}(W_1(r))$. Then, there exist a class $\mathcal{KL}$ function b for every initial state $x(t_0)$, satisfying $\|x(t_0)\| \leq W_2^{-1}(W_1(r))$, there is $T \geq 0$ (dependent on $x(t_0)$ and λ) such that $\forall x(t_0) \in \{x \in \mathcal{B}_r | W_2(x) \leq \rho\}$, the solution of $\dot{x} = f(t,x)$ satisfies $\|x(t)\| \leq b(\|x(t_0)\|, t - t_0)$, $\forall t_0 \leq t \leq t_0 + T$, $\|x(t)\| \leq W_1^{-1}(W_2(\lambda))$, $\forall t \geq t_0 + T$. Moreover, if $\mathcal{D} = \mathbb{R}^n$ and W_1 belongs to class $\mathcal{K}_\infty$, then the last two conditions hold for any initial state $x(t_0)$, with no restriction on how large λ is.

A. Proofs of Propositions

Proof: (***Proposition 1***) We use the navigation function $V(z,t) \triangleq \varphi_i(z,t)$, with $i \in \{\mathcal{A}, \mathcal{B}\}$, as a Lyapunov function candidate. Function V is a regular function [23], since it is smooth, and it holds that $V_1(\|z\|) \leq V(z,t) \leq V_2(\|z\|)$, [13], where $V_1(\|z\|)$ and $V_2(\|z\|)$ are class $\mathcal{K}$ functions.

Assume ν_1 the number of f components that are not saturated, and ν_2 the saturated components, thus $\nu_1 + \nu_2 = m$. Then, $\dot{V} = \frac{\partial V}{\partial t} + \nabla V^T f^{new} = \frac{\partial V}{\partial t} + \sum_{l_1=1}^{\nu_1} \left(-k_1 \left| \frac{\partial V}{\partial z_{l_1}} \right|^2 - \frac{1}{m} \frac{\partial V}{\partial t} \right) + \sum_{l_2=1}^{\nu_2} \left[-\text{sign}\left(\frac{\partial V}{\partial z_{l_2}} \right) \frac{\partial V}{\partial z_{l_2}} \mu \right] = \frac{\partial V}{\partial t} - \sum_{l_1=1}^{\nu_1} \left(k_1 \left| \frac{\partial V}{\partial z_{l_1}} \right|^2 + \frac{1}{m} \frac{\partial V}{\partial t} \right) - \sum_{l_2=1}^{\nu_2} \left(\left| \frac{\partial V}{\partial z_{l_2}} \right| \mu \right)$.

We can be write that $\frac{\partial V}{\partial t} = -\frac{\gamma_i^{\kappa+1}}{\kappa \cdot \left(\gamma_i^\kappa + P\right)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial P}{\partial t} + \frac{(\kappa+1) \cdot \gamma_i^\kappa}{\left(\gamma_d^\kappa + P\right)^{\frac{1}{\kappa}}} \cdot \frac{\partial \gamma_i}{\partial t} - \frac{\gamma_d^{2 \cdot \kappa}}{\left(\gamma_i^\kappa + P\right)^{\frac{\kappa+1}{\kappa}}} \cdot \frac{\partial \gamma_i}{\partial t}$, where $P = (\beta_{\mathcal{B}} \beta_{\mathcal{O}} \beta_s \beta_b)$. It can therefore be bounded as follows: $\left| \frac{\partial V}{\partial t} \right| < \frac{1}{\kappa} \cdot \gamma_i^{\kappa+1} \cdot \frac{1}{\left(\gamma_i^\kappa\right)^{\frac{\kappa+1}{\kappa}}} \cdot \left| \frac{\partial P}{\partial t} \right| + \frac{(\kappa+1) \cdot \gamma_i^\kappa}{\left(\gamma_i^\kappa\right)^{\frac{1}{\kappa}}} \cdot \left| \frac{\partial \gamma_i}{\partial t} \right| + \frac{\gamma_i^{2 \cdot \kappa}}{\left(\gamma_i^\kappa\right)^{\frac{1+\kappa}{\kappa}}} \cdot \left| \frac{\partial \gamma_i}{\partial t} \right| = \frac{1}{\kappa} \left| \frac{\partial P}{\partial t} \right| + (\kappa + 2) \cdot \gamma_i^{\kappa-1} \cdot \left| \frac{\partial \gamma_i}{\partial t} \right|$.

The properties of the navigation function ensure boundedness of its gradient within the workspace, and therefore ensure the existence of a positive bound $\theta = \left(\max_{\mathcal{W}/\mathcal{M}} \left\| \frac{\partial V}{\partial z_{l_2}} \right\|_\infty \right)^{-1}$, where $\mathcal{M}$ is a set of measure zero, including the unstable saddle points of the navigation function as well as the destination configuration. We can then write $\theta \left| \frac{\partial V}{\partial z_{l_2}} \right| < \mu_{l_2}$, $\forall l_2$, and bound the last term of

the Lyapunov's function derivative as $-\sum_{l_2=1}^{\nu_2}\left(\left|\frac{\partial V}{\partial z_{l_2}}\right|\mu\right) < -\sum_{l_2=1}^{\nu_2}\left(\left|\frac{\partial V}{\partial z_{l_2}}\right|^2\theta\right)$ Thus, it holds that $\dot{V} \leq \left(1-\frac{\nu_1}{m}\right)\frac{\partial V}{\partial t} - \theta\|\nabla V\|^2 - \sum_{l_1=1}^{\nu_1}\left(k_2\left|\frac{\partial V}{\partial z_{l_1}}\right|^2\right)$ where the control gain k_1 can be then decomposed in the form $k_1 = \theta + k_2$, with $k_2 > 0$. Then, $\dot{V} \leq (1-\frac{\nu_1}{m})C - \theta\|\nabla V\|^2$, where $C = \frac{1}{\kappa}\sup_{\bar{\mathcal{W}}}\left|\frac{\partial P}{\partial t}\right| + (\kappa+2)\cdot\max\left(\gamma_i^{\kappa-1}\right)\cdot\sup_{\bar{\mathcal{W}}}\left|\frac{\partial \gamma_i}{\partial t}\right|$, in which $\sup_{\bar{\mathcal{W}}}\left|\frac{\partial P}{\partial t}\right|$, and $\sup_{\bar{\mathcal{W}}}\left|\frac{\partial \gamma_i}{\partial t}\right|$ depends on $\sup_{\bar{\mathcal{W}}}\|\dot{q}_d\|$. In the region where $\|\nabla V\| > \sqrt{\frac{(1-\frac{\nu_1}{m})\cdot C}{\theta}}$, the Lyapunov function decreases and therefore z converges to 0. This limit point 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 $\|z\| \geq V_3(\|z\|)^{-1}\sqrt{\frac{(1-\frac{\nu_1}{m})C}{\theta}} \in \mathcal{K}$, then, for the gradient of V we can write $\|\nabla V\| \geq V_3(\|z\|) \geq \sqrt{\frac{(1-\frac{\nu_1}{m})C}{\theta}}$, implying that $\dot{V}$ is strictly negative in the region defined above. Application of Theorem 1, ensures that z is globally uniformly ultimately bounded. ■

Remark 1: We can turn uniform ultimate boundedness to uniform asymptotic stability in Proposition 1. When $\dot{q}_d \to 0$, the region defined above reduces to a ball of radius zero, because as $\dot{q}_d \to 0$, we have $\frac{\partial P}{\partial t} \to 0$. Another way is to ensure that no component of f is saturated; then $\nu_2 = 0 \Rightarrow \nu_1 = m$, and the right hand side of the appropriate condition vanishes.

Proof: (***Proposition 2***) The control law construction and the proof structure are inspired by the backstepping controller design proposed by [24].

Let $z(\cdot,t)$ be a Filippov solution of $\dot{z} = f(z,t)$, [25]. We form the Lyapunov function candidate: $V_n(z,t) \triangleq V(z,t) + \frac{1}{2}\xi^2$, where $V(z,t) = \varphi_i(z,t)$ is regular (with $i \in \{\mathcal{A},\mathcal{B}\}$). Taking the time derivative of V_n, it holds that $\dot{V}_n = \left(\frac{\partial V}{\partial t} + \frac{\partial V}{\partial z}\dot{z}\right) + \left(U - \dot{\hat{f}}^{new}\right)^T\xi$, and substituting U as defined in (3), $\dot{V}_n = \frac{\partial V}{\partial t} + (\pi_1 - \dot{\hat{f}}^{new})^T\xi - \xi^T\pi_2\xi + \frac{\partial V}{\partial z}f^{new}$ As we saw in the proof of Proposition 1, $\frac{\partial V}{\partial t} + \frac{\partial V}{\partial z}f^{new} = \frac{\partial V}{\partial t} - \sum_{l_1=1}^{\nu_1}\left(k_2\left|\frac{\partial V}{\partial z_{l_1}}\right|^2 + \frac{1}{m}\frac{\partial V}{\partial t}\right) - \sum_{l_2=1}^{\nu_2}\left(\left|\frac{\partial V}{\partial z_{l_2}}\right|\mu\right) \leq \left(1-\frac{\nu_1}{m}\right)C - \theta\|\nabla V\|^2$.

Thus, the addition of the first and the last term in the derivative of V_n is negative definite. Taking into account how π_1 has been defined, we conclude that $\dot{V}_n$ 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] 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] H. Seraji and B. Bon, "Real-time collision avoidance for position-controlled manipulators," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, 1999.

[9] 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.

[10] 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.

[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)*, pp. 3047–3052, 2006.

[12] ——, "Motion tasks and force control for robot manipulators on embedded 2-D manifolds," *2007 IEEE International Conference on Robotics and Automation*, pp. 4202–4207, 2007.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

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

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

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

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

[24] H. Tanner and K. Kyriakopoulos, "Backstepping for nonsmooth systems," *Automatica*, vol. 39, pp. 1259–1265, 2003.

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