

# Dealing With Joint Limits in Task-Space Control for Mobile Manipulators

Jonathan Obregon Gustavo Arechavaleta  
América Morales-Díaz

*Robótica y Manufactura Avanzada, CINVESTAV, Unidad Saltillo,  
Av. Industria Metalúrgica 1062, Ramos Arizpe, Coah., C.P. 25900,  
e-mail: {jonathan.obregon,garechav,america.morales}@cinvestav.mx.*

---

Abstract: Joint limits represent unavoidable constraints to be satisfied in robot motion. For the particular case of redundant robots, the Task-Space Control (TSC) is widely employed to accomplish robot motion tasks while avoiding joint limits. The TSC framework is able to simultaneously regulate and track a set of objectives in the task space according to a strict hierarchy between them. The resulting control law drives the robot towards the objectives. However, when one or more joint limits are reached the robot's end-effector could oscillate during the motion execution. This undesired behavior is due to the instantaneous change appeared in the corresponding joint velocity profiles. In this paper we provide a solution to this problem by means of smooth activation and deactivation mechanism based on time-varying transition functions within the TSC. We demonstrate throughout simulations and real experiments the effectiveness of the proposed control scheme for a kinematically redundant mobile manipulator KUKA youBot with eight degrees of freedom.

*Keywords:* Task-space control, mobile manipulators, kinematic redundancy.

---

## 1. INTRODUCTION

There exist a long history of efforts in the context of robot motion and control that have been nicely assembled in Siciliano and Khatib (2016). One of the fundamental aspects to be considered is the robot kinematics, which has been widely studied for several types of mechanisms such as parallel, soft and redundant robotic systems. An attractive feature of kinematically redundant robots is related to their inherent mobility for satisfying task objectives while exploiting their redundancy to deal with joint limits. The seminal work of Liégeois (1977) applied the Gradient Projection Method (GPM) to take into account joint limits as a secondary objective to be satisfied by projecting the resulting gradient of these constraints within the null-space of the primary task Jacobian. Later, Hanafusa et al. (1981) carefully analyzed different tasks expressed as convex functions where their gradients are also incorporated in the same scheme. The operational space control introduced in (Khatib, 1987) extended these ideas by projecting the equations of motion of redundant robotic arms into the robot's end-effector. This control framework also uses the redundancy to deal with obstacles and joint limits formulated as repulsive potential fields by means of the GPM (Khatib, 1986). Recently, these methods have been revisited due to the interest for providing human-like mobility to humanoid robots (Saab et al., 2013; Estopier-Castillo et al., 2014). In particular, all these strategies can be formulated in terms of Quadratic Programs (QP) that allow to consider both equality and inequality constraints at any hierarchical level. From this

perspective, special attention has been given to inequality constraints because their activation causes discontinuities in the joint velocity profiles and as a by-product the robot could oscillate during the motion execution (Lee et al., 2012; Han and Park, 2013). To overcome such undesired behavior Lee et al. (2012) proposed the intermediate desired value strategy, which is based on time-varying functions that smoothly activate and deactivate constraints within a transition time interval. This strategy scales easily when several hierarchical tasks have to be accomplished by the redundant robot. The same strategy has been applied within the operational space control in (Han and Park, 2013).

In this paper we adopt the Task-Space Control (TSC) framework together with the intermediate desired value strategy for controlling an omnidirectional mobile manipulator in velocity and torque modes. The control law is able to achieve task space objectives while handling smooth activation and deactivation of joint limits.

The paper is organized as follows. Section 2 describes the omnidirectional mobile manipulator model. In particular the torque distribution between the wheels and the generalized torques. In Section 3 the TSC framework is briefly recalled to solve both the hierarchical inverse kinematics and dynamics. Then, in Section 4 we describe the smooth transition mechanism to activate and deactivate inequality constraints, and the particular problem of joint limits is described in Section 5. Finally, some simulations and experimental results are described in Section 6, and the concluding remarks are in Section 7.

## 2. MOBILE MANIPULATOR MODEL

The equations of motion of a mobile manipulator in generalized coordinates have the following form:

$$H(q)\ddot{q} + h(q, \dot{q}) = \tau, \quad (1)$$

where the joint configuration, velocity and acceleration are  $q = (q_b, q_m) \in \mathcal{CS} = \mathcal{CS}_b \times \mathcal{CS}_m$ ,  $\dot{q} \in \mathbb{R}^n$  and  $\ddot{q} \in \mathbb{R}^n$ , respectively. The robot configuration is composed by the configuration of the wheeled platform  $q_b \in \mathcal{CS}_b = SE(2)$  and the configuration of the manipulator,  $q_m \in \mathcal{CS}_m$ , mounted on such mobile platform. Since  $n = \dim(\mathcal{CS})$  and  $3 = \dim(\mathcal{CS}_b)$ , the dimension of  $\mathcal{CS}_m$  is  $n - 3$ . The inertia matrix  $H(q) \in \mathbb{R}^{n \times n}$  is symmetric and positive definite, and  $h(q, \dot{q}) \in \mathbb{R}^n$  denotes the Coriolis, centrifugal and gravity forces. The generalized torques are:

$$\begin{bmatrix} \tau_b \\ \tau_m \end{bmatrix} = \begin{bmatrix} J_b & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} f_b \\ \tau_m \end{bmatrix} \in \mathbb{R}^n \quad (2)$$

where

$$J_b = \begin{bmatrix} \frac{hr_z - 1}{r} & -\frac{hr_z - 1}{r} & \frac{hr_z - 1}{r} & -\frac{hr_z - 1}{r} \\ -\frac{\ell^r}{2r} & -\frac{\ell^r}{2r} & \frac{\ell^r}{2r} & \frac{\ell^r}{2r} \end{bmatrix} \quad (3)$$

depends on the vehicle parameters, and it maps the wheel's torques  $f_b \in \mathbb{R}^4$  to the vehicle generalized torques  $\tau_b = [\tau_x \ \tau_y \ \tau_\psi]^T \in \mathbb{R}^3$  as it can be observed in Figure 1.(a). The generalized torques of the manipulator attached to the vehicle are denoted by  $\tau_m \in \mathbb{R}^{3-n}$  as it is illustrated in Figure 1.(b).

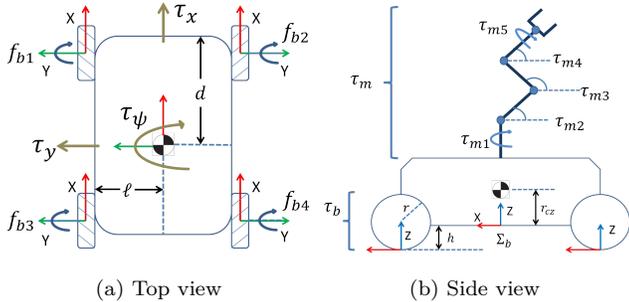


Figure 1. **The mobile manipulator.** Left: torque distribution of the mobile platform. Right: kinematic structure of the mobile manipulator.

## 3. TASK-SPACE CONTROL

One of the purposes of the task-space control is to exploit the kinematic redundancy of the robot. This kind of redundancy appears when the dimension of  $\mathcal{CS}$  is greater than the dimension of the task space  $\mathcal{TS}$ . In other words,  $m < n$  where  $m = \dim(\mathcal{TS})$ .

A task can be either defined at kinematic or dynamic level by means of an error function in terms of the robot's configuration together with the corresponding differential mapping between the task and control spaces of the robotic system (Samson et al., 1991):

$$e = x(q) - x_d \in \mathcal{TS} \quad (4)$$

where  $x(q)$  is evaluated by computing the forward kinematics,  $x_d$  is the desired value of the task. The behavior of the task error in terms of the control is (Saab et al., 2013):

$$\dot{e} = Qu - \mu \quad (5)$$

where  $Q$  is the differential mapping between task and control spaces, and  $\mu$  is the drift of the task. For first order systems,  $Q = J$  is the task Jacobian  $J = \frac{\partial e}{\partial q}$ ,  $\mu = 0$  and  $u = \dot{q}$  such that:

$$\dot{e} = J\dot{q} \quad (6)$$

For second order systems,  $e$  is assumed to be twice differentiable with respect to time:

$$\ddot{e} = J\ddot{q} + \dot{J}\dot{q} \quad (7)$$

Solving for  $\ddot{q}$  in (1) and plugging it in (7), yields:

$$\ddot{e} = JH^{-1}\tau - JH^{-1}h + \dot{J}\dot{q}. \quad (8)$$

In this case  $Q = JH^{-1}$ ,  $u = \tau$  and  $\mu = JH^{-1}h - \dot{J}\dot{q}$

The control law becomes:

$$u = Q^{\#w}(\ddot{e} + \mu) \quad (9)$$

where

$$Q^{\#w} = WQ^T(QWQ^T)^{-1}. \quad (10)$$

is the right pseudo-inverse of  $Q$  weighted by a symmetric positive-definite matrix  $W$ . For inverse kinematics  $W = I$  and for inverse dynamics  $W = H$ . The next ingredient is to impose an exponential convergence of the error for first and second order systems as follows:

$$\dot{e} = -\Lambda_p e \quad (11)$$

$$\ddot{e} = -\Lambda_p e - \Lambda_v \dot{e} \quad (12)$$

where  $\Lambda_p$  and  $\Lambda_d$  are diagonal matrices with positive constant gains of appropriate dimensions.

### 3.1 Simultaneous execution of tasks

The control law (9) considers a single task. However, a stack of  $r$  tasks similar to (8) can be simultaneously solved if the redundancy of the robot is enough, i.e.  $m_1 + \dots + m_r \leq n$ . In this case, the control law for  $r$  tasks becomes:

$$u = \begin{bmatrix} Q_1 \\ \vdots \\ Q_r \end{bmatrix}^{\#w} \begin{bmatrix} \ddot{e}_1 + \mu_1 \\ \vdots \\ \ddot{e}_r + \mu_r \end{bmatrix}. \quad (13)$$

### 3.2 Simultaneous execution of hierarchical tasks

The definition of hierarchical tasks allows to handle conflicts between them by projecting the solution of tasks with less hierarchy into the shared null-space of previous tasks with greater hierarchy. Let us define the null-space projector as:

$$N = I - Q^{\#w}Q, \quad (14)$$

By applying this operator, the recursive projection for  $p$  hierarchical tasks is (Siciliano and Slotine, 1991; Baerlocher and Boulic, 2004):

$$u = \sum_{k=1}^p u_k, \quad (15)$$

where

$$u_k = \overline{Q}_k^{\sharp W} (\ddot{e}_k + \mu_k - Q_k u_{k-1}),$$

$$N_k = N_{k-1} - \overline{Q}_k^{\sharp W} \overline{Q}_k$$

such that  $u_0 = 0$ ,  $\overline{Q}_k = N_{k-1} Q_k$ ,  $W = H$  and  $N_0 = I_n$ . The control law (15) considers  $p$  hierarchical tasks.

#### 4. HANDLING SMOOTH TASK TRANSITIONS

The stack of tasks used in (13), and the hierarchy defined in (15), are commonly assumed to be fixed along the robot motion. However, if at some instant of time a task is added to or removed from the stack, discontinuous input signals occur which lead to instability of the system. This is directly related to the instantaneous change of the rank of  $Q$  in (10) (see Keith et al. (2011)). Such undesired behavior can be avoided by defining transition intervals where some  $\xi$  function continuously evolve between 0 and 1. Thus, a task is in transition when it is activated or deactivated.

We adopted the strategy suggested in Lee et al. (2012) to define intermediate desired values  $\ddot{e}^i$  for considering the contribution of the tasks different from  $j$  to perform the  $j$  task. According to this strategy, the control law (13) is modified as follows:

$$u = \begin{bmatrix} Q_1 \\ \vdots \\ Q_r \end{bmatrix}^{\sharp W} \begin{bmatrix} \ddot{e}_1^i + \mu_1 \\ \vdots \\ \ddot{e}_r^i + \mu_r \end{bmatrix}, \quad (16)$$

Each intermediate value  $\ddot{e}_j^i$  is defined as

$$\ddot{e}_j^i = \xi_j \ddot{e}_j + (1 - \xi_j) Q_j u_{[\setminus j]}, \quad (17)$$

where the transition function  $\xi_j$  varies from 0 to 1,  $j = \{1, \dots, r\}$ , and  $u_{[\setminus j]}$  denotes the control law for all tasks different from  $j$  task, and it is expressed as:

$$u_{[\setminus j]} = \begin{bmatrix} Q_1 \\ \vdots \\ Q_{j-1} \\ Q_{j+1} \\ \vdots \\ Q_r \end{bmatrix}^{\sharp W} \begin{bmatrix} \ddot{e}_1 + \mu_1 \\ \vdots \\ \ddot{e}_{(j-1)} + \mu_{(j-1)} \\ \ddot{e}_{(j+1)} + \mu_{(j+1)} \\ \vdots \\ \ddot{e}_r + \mu_r \end{bmatrix} \quad (18)$$

#### 5. JOINT LIMITS AVOIDANCE

The feasibility of the task is inherently constrained by the joint limits of the robot arm. Such constraints are naturally formulated as inequalities. If the robot arm reaches a joint limit, the corresponding inequality is activated. In other words, the inequality switches to an equality for restricting the robot motions before the joint configuration contacts the forbidden region. In these terms, the abrupt change generates discontinuous signals and, consequently, some vibrations of the mobile manipulator could appear.

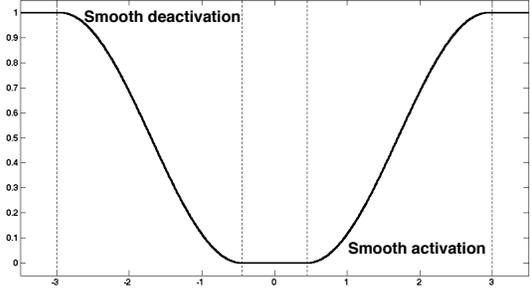


Figure 2. Buffers for smooth joint limit transitions.

To deal with this kind of constraints we define a stack of tasks denoted by  $\ddot{e}_i$  at the first hierarchical level with the following form:

$$\ddot{e}_{l_j} = \lambda_{p_j} (\tilde{q}_j - q_j) - \lambda_{v_j} \dot{q}_j, \quad (19)$$

where  $\lambda_{p_j}$  and  $\lambda_{v_j}$  are positive constant gains. The current joint position and velocity values are  $q_j$  and  $\dot{q}_j$ , respectively. The bound is computed as:

$$\tilde{q}_j = \bar{q}_j - \beta \quad (20)$$

where  $\bar{q}_j$  is the upper limit and  $\beta$  represents an activation buffer. The task definition for the lower limit is similar to (20), but in this case the bound is computed as:

$$q_j = \underline{q}_j + \beta. \quad (21)$$

where  $\underline{q}_j$  is the lower limit of joint  $j$ . The differential mapping for the stack of joint limits task  $Q_l$  contains the joint limits Jacobian  $J_l$ , which is a row-dimension-variable matrix because each of its rows corresponds to an active, or in transition, joint limit. The  $j$  row of  $J_l$  has the following form:

$$J_{l_j} = [0 \ \dots \ \alpha_j \ \dots \ 0] \in \mathbb{R}^{1 \times n}, \quad (22)$$

where  $\alpha_j = 1$  if  $q_j$  is reaching the joint limit, i.e. the constraint is active in the stack of joint limits task, otherwise  $\alpha_j = 0$ . It is at this stage when we apply the transition functions to smoothly handle the addition and removal of joint limit constraints. The main purpose of these transitions is to avoid an abrupt stop of the robot motion, or an instantaneous change of motion direction that could damage the robot actuators.

For the particular case of joint limits, the transition function  $\xi_j$  in (17) depends on the time-varying joint coordinate  $q_j$ , and it is defined as follows (see Figure 2):

$$\xi_j = \begin{cases} 1, & \text{if } q_j \geq \bar{q}_j \\ f(q_j), & \text{if } \tilde{q}_j < q_j < \bar{q}_j \\ 0, & \text{if } \underline{q}_j \leq q_j \leq \tilde{q}_j \\ g(q_j), & \text{if } \underline{q}_j < q_j < \tilde{q}_j \\ 1, & \text{if } q_j \leq \underline{q}_j \end{cases} \quad (23)$$

where

$$f(q_j) = \frac{1}{2} + \frac{1}{2} \sin \left( \frac{\pi}{\beta} (q_j - \tilde{q}_j) - \frac{\pi}{2} \right) \quad (24)$$

$$g(q_j) = \frac{1}{2} + \frac{1}{2} \sin \left( \frac{\pi}{\beta} (q_j - \underline{q}_j) + \frac{\pi}{2} \right) \quad (25)$$

The control law associated to the joint limits task

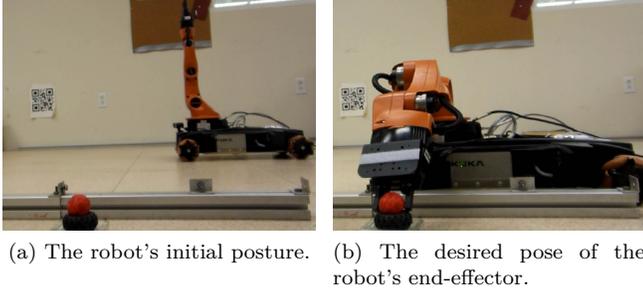


Figure 3. **The experimental scenario** for reaching a small ball without colliding with a static obstacle.

together with the intermediate desired value strategy becomes:

$$u_l = \begin{bmatrix} Q_{l_1} \\ \vdots \\ Q_{l_j} \\ \vdots \\ Q_{l_r} \end{bmatrix} \overset{\#w}{=} \begin{bmatrix} \ddot{e}_{l_1}^i + \mu_{l_1} \\ \vdots \\ \ddot{e}_{l_j}^i + \mu_{l_j} \\ \vdots \\ \ddot{e}_{l_r}^i + \mu_{l_r} \end{bmatrix}, \quad (26)$$

with

$$\ddot{e}_{l_j}^i = \xi_j \ddot{e}_{l_j} + (1 - \xi_j) Q_{l_j} u_{l_{[j]}}, \quad (27)$$

where  $u_{l_{[j]}}$  is computed as in (18). The null-space projector of the joint limits task is:

$$N_l = I - Q_l \overset{\#w}{=} Q_l, \quad (28)$$

which is used to project secondary tasks as follows:

$$u = u_l + N_l u_s, \quad (29)$$

where  $u_s$  represents the control input of the lower hierarchical levels as a result of the recursive computation in (15). For instance, we could define three hierarchical levels to reach a desired position and orientation with the mobile manipulator's end-effector. The first one handles the active joint limits task as previously described. At the second hierarchical level the control law regulates the end-effector's orientation, and at the third one the control law regulates the end-effector's position.

## 6. SIMULATION AND EXPERIMENTAL RESULTS

The experiments were conducted with an eight degrees of freedom omnidirectional mobile manipulator KUKA youBot. It is important to mention that the TSC schemes are model-based controllers, which means that they rely on the accuracy of the kinematic and dynamic parameters provided by KUKA manufacturer. The computation of the equations of motion in (1) were performed with the spatial and recursive formulation introduced by Featherstone (2010) and implemented in Felis (2017). In particular, the spatial Newton-Euler algorithm efficiently evaluates the non-linear terms encoded in  $h(q, \dot{q})$  that appears in (1). The Composite-Rigid-Body algorithm allows to efficiently compute the inertia matrix  $H(q)$  that is also widely used in (9). Since the time derivative of the task Jacobian is needed for each task, we adopted the method introduced in (Estopier-Castillo et al., 2014) to evaluate it recursively.

The simulation considered second-order tasks to command the robot in torque-mode, and the implementation was coded in MATLAB R2015b. For the experiments, the computation was carried out at kinematic level on the robot's onboard computer. The whole control scheme was implemented in ANSI C++, and the routines related to numerical linear algebra used Eigen 3.1.1.

The experimental scenario is depicted in Figure 3. The controller has to regulate the mobile manipulator from its initial configuration (see Figure 3.(a)) to a feasible placement of the robot's end-effector near the small ball (Figure 3.(b)).

### 6.1 Simulation with the torque-mode controller

The TSC law is computed with (15) where the task with highest hierarchy was designed with (29) to handle smooth joint limit transitions. At the second hierarchical level, the robot's end-effector orientation was regulated towards a desired orientation parametrized as a quaternion for avoiding singularities. At the third hierarchical level, a reaching task regulated the robot's end-effector position to a desired value represented by ball's position.

It can be seen in Figure 4 that two joint limits are activated. The top row of Figure 4 shows the profiles of  $q_1$  and  $q_5$ , respectively. Both joints reached the buffer of the corresponding joint limit. At the bottom left the joint velocities of those joints are depicted. As it is observed, the joint limit activation mechanism did not produce discontinuities in  $\dot{q}_1$  and  $\dot{q}_5$ . Moreover, the end-effector position converged smoothly to the desired position as it is depicted in the bottom right. It is important to mention that we did not impose an exponential convergence of the error (12). Instead of that, we applied a predefined time convergence for a second-order system as it is explained in Becerra et al. (2017). Thus, we were able to impose 5 seconds to accomplish the reaching task while avoiding the discontinuity at  $t = 0$  induced by the exponential convergence.

### 6.2 Experiments with the first-order kinematic controller

We performed two experiments. In the first one the joint limits task was not considered. Thus, the first hierarchical task regulated the end-effector's orientation to a desired value, and at the second hierarchical level the end-effector reaching task was performed in position. The control law was computed at kinematic level where the robot's end-effector's approached an object. However, a static obstacle was placed on the floor to interfere the motion of the mobile platform. Hence, the robot must stop rolling to avoid a collision with the obstacle. Because of that the robot activated a repulsive potential field that caused an abrupt change of the corresponding translational joint velocity, and as a by-product the end-effector mounted on the mobile platform oscillated. Also, such oscillation was amplified by the non-smooth activation of  $q_5$ . This undesired behavior is compared with the second experiment where smooth joint limit transitions were considered in Figure 5. The associated video of these experiments: <https://sites.google.com/site/gustavoarechavaleta/tscmm>

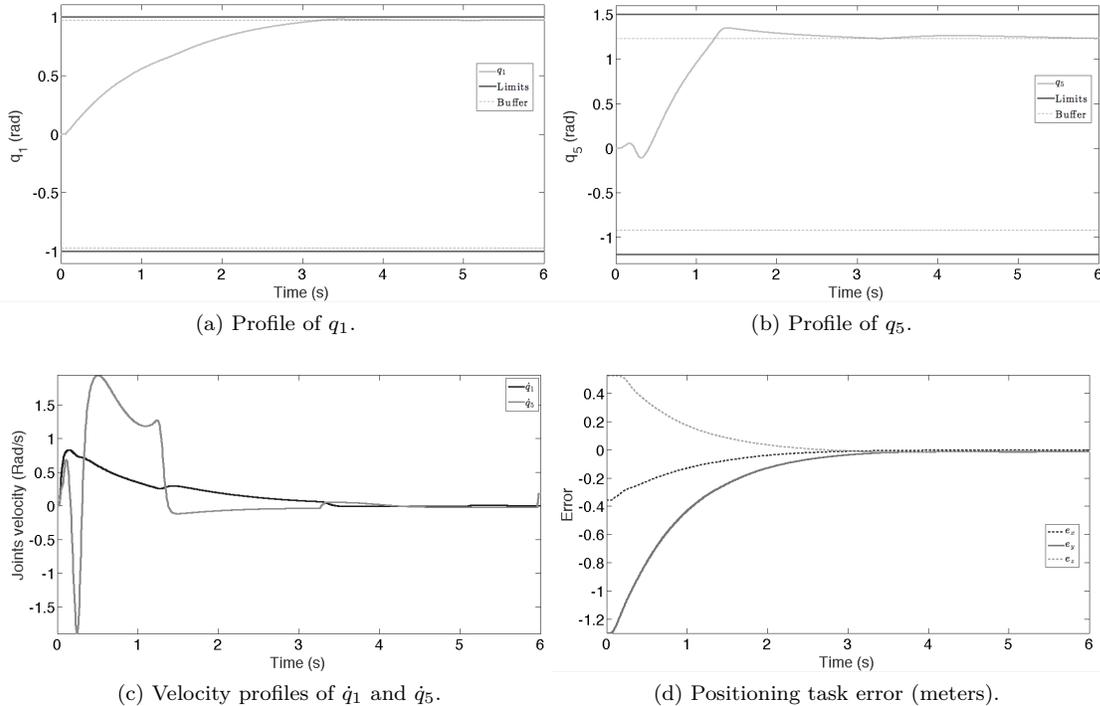


Figure 4. **The torque-controlled mobile manipulator performs three hierarchical tasks with smooth joint limit transitions.** **Top-left:** the translational joint  $q_1$  is activated due to a static obstacle on the floor. **Top-right:** the rotational joint  $q_5$  of the manipulator is activated when reaching the desired end-effector position. **Bottom-left:** the joint velocity profiles are not discontinuous. **Bottom-right:** the error convergence of the end-effector position task.

## 7. CONCLUSIONS

In this work we have demonstrated with some simulations and real experiments the use of task transition functions for smoothly activating inequality constraints associated to the joint limits of an omnidirectional mobile manipulator composed by eight degrees of freedom. The task space control framework is a general model-based scheme suitable to control redundant robots in both velocity and torque modes. However, we have shown that undesired robot motion behavior appeared when joint limits are not carefully handled. Moreover, the kinematically redundancy of the robot has been exploited to simultaneously achieve a set of hierarchical tasks by means of the dynamically consistent projectors and the generalized inverse. Currently, we are dealing with the sensitivity of the inverse dynamics (Chang and Jeong, 2012), and we are working on the incorporation of contact forces.

## REFERENCES

- Baerlocher, P. and Boulic, R. (2004). An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer: International Journal of Computer Graphics*, 20(6), 402–417.
- Becerra, H.M., Vázquez, C.R., Arechavaleta, G., and Delfin, J. (2017). Predefined-time convergence control for high-order integrator systems using time base generators. *IEEE Transactions on Control Systems Technology*.
- Chang, P.H. and Jeong, J.W. (2012). Enhanced operational space formulation for multiple tasks by using time-delay estimation. *IEEE Transactions on Robotics*, 28(4), 773–786.
- Estopier-Castillo, V., Arechavaleta, G., and Olguín-Díaz, E. (2014). Generación de movimientos humanoides con dinámica inversa jerárquica. In *XVI Congreso Latinoamericano de Control Automático*, 266–217. Cancún, Quintana Roo, México.
- Featherstone, R. (2010). A beginner’s guide to 6-d vectors (part 1). *IEEE robotics & automation magazine*, 17(3), 83–94.
- Felis, M.L. (2017). Rbd1: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, 41(2), 495–511.
- Han, H. and Park, J. (2013). Robot control near singularity and joint limit using a continuous task transition algorithm. *International Journal of Advanced Robotic Systems*, 10(10), 346.
- Hanafusa, H., Yoshikawa, T., and Nakamura, Y. (1981). Analysis and control of articulated robot arms with redundancy. In *IFAC 8th. Triennial World Congress*, 1927–1932. Kyoto, Japan.
- Keith, F., Wieber, P.B., Mansard, N., and Kheddar, A. (2011). Analysis of the discontinuities in prioritized task-space control under discrete task scheduling operations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3887–3892. San Francisco, CA, USA.

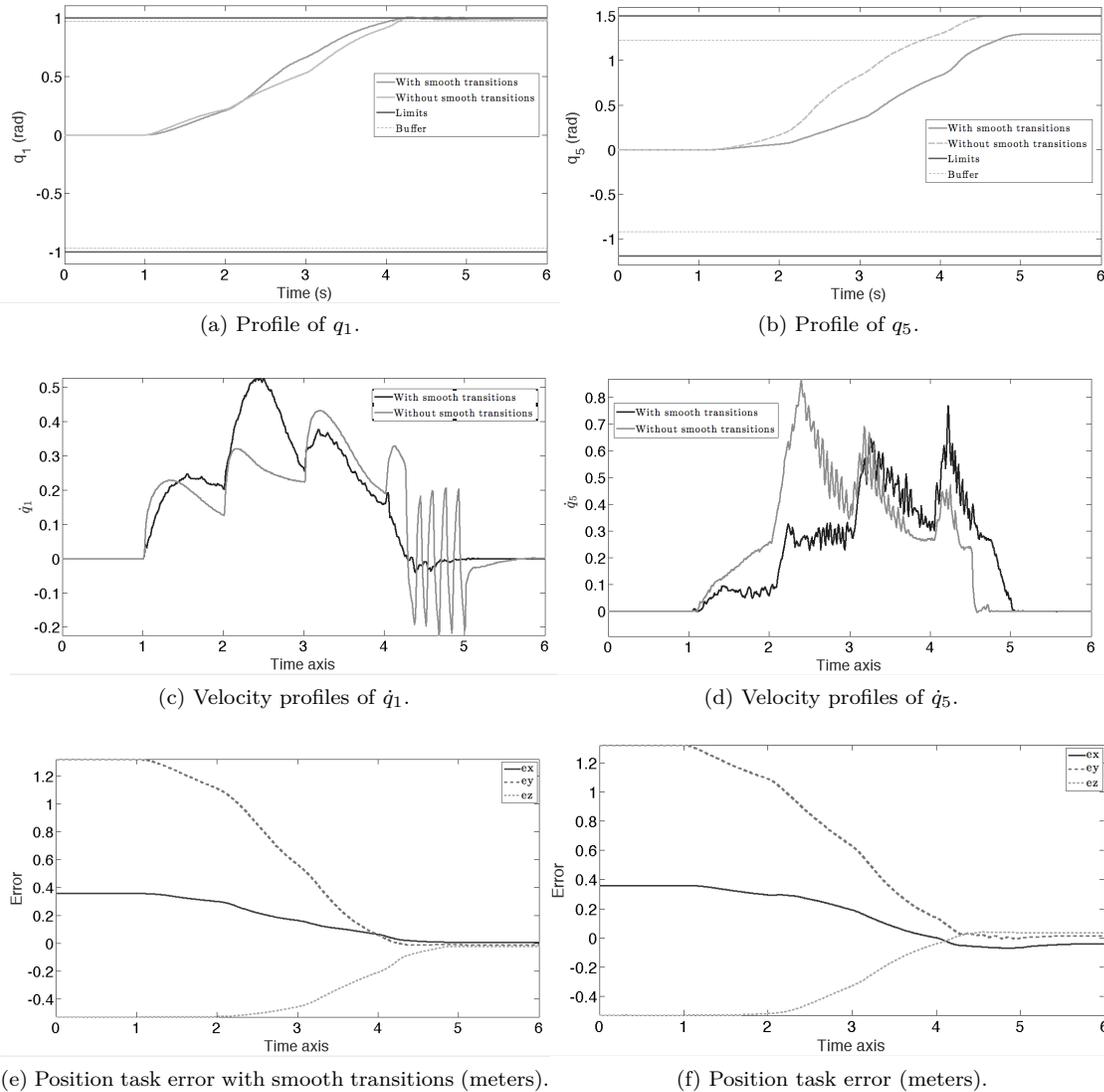


Figure 5. **Comparison of the TSC for the mobile manipulator to solve hierarchical kinematic tasks with and without smooth joint limit transitions.** (a) and (b) show the activation of the joint limits for  $q_1$  and  $q_5$ , respectively. The gray and black profiles refer to the first and second experiments, respectively. Notice that the first experiment did not consider smooth joint limit transitions. (c) and (d) show the joint velocity profiles of the same robot joints. (e) and (f) show the error convergence behavior for the reaching task of end-effector with smooth joint limit transitions and with repulsive potential fields, respectively

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1), 43–53.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1), 90–98.

Lee, J., Mansard, N., and Park, J. (2012). Intermediate desired value approach for task transition of robots in kinematic control. *IEEE Transactions on Robotics*, 28(6), 1260–1277.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12), 868–871.

Saab, L., Ramos, O.E., Keith, F., Mansard, N., Soueres, P., and Fourquet, J.Y. (2013). Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2), 346–362.

Samson, C., Borgne, M.L., and Espiau, B. (1991). *Robot Control: The Task Function Approach*, volume 22 of *Oxford Engineering Science Series*. Oxford University Press, New York, USA, first edition.

Siciliano, B. and Slotine, J.J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *IEEE International Conference on Advanced Robot*, 1211–1216. Pisa, Italy.

Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer.