

Multi-Priority Control in Redundant Robotic Systems

Hamid Sadeghian, Luigi Villani, Mehdi Keshmiri, Bruno Siciliano

Abstract—This paper presents a dynamic level control algorithm to meet simultaneously multiple desired tasks based on allocated priorities for redundant robotic systems. It is shown that this algorithm can be treated as a general framework to achieve control over the whole body of the robot and some of the previously developed results are formalized using this approach. Null-space impedance control is proposed as one of the main results of using this method and is evaluated by means of computer simulation.

I. INTRODUCTION

ROBOTS are termed kinematically redundant when they possess more degrees of freedom than those necessary to achieve a desired task. Redundant degrees of freedom can be conveniently used to perform some additional tasks besides the main task. These additional tasks can be a performance objective or for example a given Cartesian position of a point on the body of robot. There are plenty of papers that deal with how to use redundancy effectively to optimize some performance objective besides the main task control. This optimization is usually performed in the null-space of the main task to ensure its perfect tracking. In order to solve the conflict between tasks in the case where several objective functions are going to be satisfied simultaneously, the so-called task priority strategy developed in [1,2] is adopted. The formulation has later been extended in a general framework for managing multiple tasks by Siciliano and Slotine [3].

This formulation uses first-order differential kinematics equation and solves redundancy in the Least-Squares (LS) sense, based on the assigned priority by resorting to pseudo-inverse solution. Because of using the pseudo-inverse of the *projected Jacobians*—the Jacobians of the lower-priority tasks that are projected into the null-space of the higher-priority tasks—the formulation may suffer from high norms during transition into and out of algorithmic singularities. Usually singularity-robust pseudo-inverse that allows limiting joint velocities at the expense of small tracking

error in lower priority tasks is the first remedy to treat this problem. Efficient damping techniques have been suggested by Nakamura and Hanafusa [4] and Wampler [5] and also by Nenchev et al [6] for the case of multiple priorities.

Chiaverini [7] proposed the singularity-robust task-priority resolution without using the projected Jacobian. This formulation always involves tracking errors in the additional tasks but singularities do not occur as long as the Jacobian of each additional task is full rank. The stability of this formulation has been shown in [8].

De Santis et al [9] apply the concept of Multi Point Control and Virtual End-Effectors (VEEs) for Human-Robot Interaction (HRI). The VEEs are parts of the manipulator structure, whose positions are to be controlled in addition to the control of the end-effector of the robot manipulator. They proposed a nested closed-loop inverse kinematics algorithm, with a priority management strategy in order to control robot in a cluttered environment.

Instead of velocity-based control, acceleration-based control computes the desired joint accelerations for given task references. Synthesis of joint acceleration in a redundant robot usually requires a more involved analysis, but for second-order system such as robots this formulation is the most natural and offers improved tracking ability due to the explicit incorporation of acceleration information.

The problem of internal instability at the acceleration level was first noticed by Hsu et al [10]. De Luca et al [11] presented different methods for solving robot redundancy at the acceleration level. A complete theoretical and empirical evaluation of different dynamic methods has been investigated in [12].

There are a few papers that use multi-priority control at dynamic level. Khatib et al [13,14] proposed the extension of the operational space formulation [15] to control the behavioral primitives in a humanoid robot at torque level.

Recently Platt et al [16] proposed multi-priority Cartesian impedance control by resorting to acceleration resolution.

This paper investigates multi-priority control at the acceleration level for redundant robotic systems and establishes a general framework to achieve dynamic control over the whole body of robot. It is shown that by a proper choice of the additional tasks it is possible to derive previous results in the literature within this framework. Null-space impedance control, as a result of task prioritization, together with the possible solution to cope with singularities, is presented as a main result of this work. The paper is organized as follows. In Section II the multi-priority resolution at the velocity level is briefly described. The main results, including acceleration level multi-priority control,

Manuscript received March 21, 2011.

B. Siciliano and L. Villani (emails: {bruno.siciliano,luigi.villani}@unina.it) are with PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Italy.

H. Sadeghian is a Ph.D. candidate at Isfahan University of Technology, and currently a visiting scholar at PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Italy. (e-mail: h.sadeghian@me.iut.ac.ir).

M. Keshmiri is with the Mechanical Engineering Department, Isfahan University of Technology, Isfahan, Iran (e-mail: mehdik@cc.iut.ac.ir)

This work was partially supported by the FP7 DEXMART project, funded by the European Community under grant agreement ICT-216239.

null-space impedance and singularity treatment are presented in Sections III and IV. Section V gives a comparison between torque level and acceleration level, while some of the main results are verified by simulation in Section VI. The conclusion is given in the final section.

II. MULTI-PRIORITY INVERSE KINEMATICS

Multi-priority inverse kinematics is a well-established framework to manage the tasks in a kinematically redundant robotic system. Assume that the task is composed of two prioritized tasks. The first-priority task (main task) is specified using the first manipulation variable $X_1 \in R^{m_1}$ and the second-priority task (sub-task) by the second manipulation variable $X_2 \in R^{m_2}$. The kinematic relationships between the joint vector $\dot{q} \in R^n$ and the vectors of task variables are expressed by

$$\dot{X}_i = J_i(q)\dot{q}, \quad (i=1,2) \quad (1)$$

where $J_i(q) \in R^{m_i \times n}$ is the Jacobian matrix of the i -th task. The inverse kinematics solution considering the priority of the main task over the sub-task is given by

$$\dot{q} = J_1^\dagger \dot{X}_1 + \bar{J}_2^\dagger (\dot{X}_2 - J_2 J_1^\dagger (q) \dot{X}_1) + N_2 \eta, \quad (2)$$

where $(\cdot)^\dagger$ is the pseudo-inverse of the related matrix, $\bar{J}_2 = J_2 N_1$ is the projected Jacobian, which gives the available range for the sub-task to be executed without affecting the main task, $N_1 = (I - J_1^\dagger J_1)$, $N_2 = N_1 (I - \bar{J}_2^\dagger \bar{J}_2)$ and η is an arbitrary vector [2]. A recursive extension of (2) was proposed in [3].

In multi-priority control, an algorithmic singularity occurs whenever the projected Jacobian $\bar{J}_i$ drops rank. Two generic tasks are dependent when

$$\rho(J_1) + \rho(J_2) > \rho(\bar{J}_2),$$

$$\rho(\bar{J}_2) = \rho\left(\begin{bmatrix} J_1 \\ J_2 \end{bmatrix}\right),$$

where $\rho(\cdot)$ denotes the rank of a matrix. For more details about task independency conditions and algorithmic singularities see [17,7,8].

Thus it is obvious that singularities may occur from the lower-priority tasks. In the case of free task priority assignment, dynamic task priority allocation in [6] is crucial to the overall performance of the system. For a singularity-robust task-priority handling, Chiaverini [7] proposed the following formulation for a case with two tasks

$$\dot{q} = J_1^\dagger \dot{X}_1 + N_1 J_2^\dagger \dot{X}_2, \quad (3)$$

Comparing with (2), algorithmic singularities are absent, but there is typically a greater tracking error for the sub-task [8,18].

The above formulation has also been successfully applied for path planning of a mobile robot in [18].

III. MULTI-PRIORITY CONTROL AT THE ACCELERATION LEVEL

A. General Formulation

The goal of multi-priority control is to derive a control torque which will cause the system to track the desired main task exactly, while at the same time, system redundancy is exploited to realize a number of sub-tasks according to some desired priorities.

Dynamics of a robot manipulator can be written in compact form as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_{\text{ext}} = \tau, \quad (4)$$

with known notation. In this formulation τ_{ext} are the exerted external forces on the manipulator, including the forces applied on the body of robot. The kinematic relationships between the joint variable $q \in R^n$ and the task variables at the acceleration level are expressed by

$$\ddot{X}_i = J_i \ddot{q} + \dot{J}_i \dot{q}, \quad i=1,2, \quad (5)$$

Following the same guideline used in (2) for extracting $\dot{q}$, the corresponding LS solution for the joint space command acceleration $\ddot{q}_c$ for a given task space command accelerations $\ddot{X}_{c1}, \ddot{X}_{c2}$ is obtained by

$$\ddot{q}_c = J_1^\dagger (\ddot{X}_{1c} - \dot{J}_1 \dot{q}) + N_1 \bar{J}_2^\dagger [\ddot{X}_{2c} - \dot{J}_2 \dot{q} - J_2 J_1^\dagger (\ddot{X}_{1c} - \dot{J}_1 \dot{q})] + N_2 \eta, \quad (6)$$

The basic issue in this formulation is the differential order at which resolution take place. In the case of two or more sub-tasks, $\ddot{q}_c$ can be obtained similarly. Further investigation enables us to propose the general recursive solution for k tasks, as follows:

$$\ddot{q}_c^{(i)} = \ddot{q}_c^{(i-1)} + \bar{J}_i^\dagger (\ddot{X}_{ic} - \dot{J}_i \dot{q} - J_i \ddot{q}_c^{(i-1)}) + N_i \eta, \quad (7)$$

$$\bar{J}_i = J_i N_{i-1}$$

$$N_i = \Pi_i^\dagger (I - \bar{J}_i^\dagger \bar{J}_i)$$

$$N_0 = 0, \quad \ddot{q}_c^{(0)} = 0, \quad i=1, \dots, k$$

where the matrices $\bar{J}_i$ are termed projected Jacobians because they are obtained from the Jacobian matrix of the i -th task as projected into the null-space of the higher-priority tasks. Notice that the null-space N_i can also be written as

$$N_i = (I - J_{Ai}^\dagger J_{Ai}), \quad (8)$$

$$J_{Ai} = \begin{bmatrix} J_1 \\ \vdots \\ J_i \end{bmatrix},$$

where J_{Ai} is the augmented Jacobian.

Equation (7) can be treated as a general framework to control the whole behavior of the redundant robotic system by multiple priorities and even multi-point control. The acceleration level prioritization in contrast to velocity level prioritization, gives full trajectory planning with complete time information. A nice property of (7) is that, as desired, it can be easily used for any kind of force motion control by a

proper choice of the operational command acceleration. The method also enables us to give priority to one task over another even in a non-redundant robotic system. This kind of formulation can explain the previously proposed acceleration level resolution techniques in the framework of task prioritization by a proper choice of sub-tasks.

Regarding (7), the following remarks must be considered;

- 1) Care must be taken about internal instability. By a proper choice of the sub-task it is possible to achieve control over the internal motion as we will see in this section.
- 2) The first $i-1$ tasks influence the performance of task i , hence the way of priority allocation will be crucial to the performance.
- 3) The recursive scheme suffers from high-norm solutions in the neighborhood of singularity. This issue is treated in the next section.

Once the command acceleration $\ddot{q}_c$ is obtained, a well-known resolved acceleration control [19] can be used to find the driving torques

$$\tau = M(q)\ddot{q}_c + C(q, \dot{q})\dot{q} + g(q) + \tau_{ext}. \quad (9)$$

By the above compensation of Coriolis, centrifugal, gravity and external torques, the closed-loop behavior of the system is obtained as

$$\ddot{q} = \ddot{q}_c, \quad (10)$$

Multiplying both sides by J_1 and using (6) gives

$$\ddot{X}_1 = \ddot{X}_{1c}, \quad (11)$$

meaning that the main task is correctly executed.

Multiplying both sides of (10) by J_2 and further by $\bar{J}_2\bar{J}_2^\dagger$ and considering the idempotency of $\bar{J}_2\bar{J}_2^\dagger$ yields

$$\bar{J}_2\bar{J}_2^\dagger[\ddot{X}_{2c} - \ddot{X}_2] = 0. \quad (12)$$

Thus, if the two tasks are not in conflict and $\bar{J}_2$ is full rank, the sub-task is correctly executed. Equation (12) has been obtained under the assumption that the external forces applied to the manipulator are entirely compensated. In the case where there is no torque sensor to compensate τ_{ext} , the closed-loop equations for the main task and the sub-task are as follows:

$$\ddot{X}_{1c} - \ddot{X}_1 = J_1 M^{-1} \tau_{ext}, \quad (13)$$

$$\bar{J}_2\bar{J}_2^\dagger[\ddot{X}_{2c} - \ddot{X}_2 - J_2 M^{-1} \tau_{ext}] = 0.$$

This formulation will be used later in the case of impedance control.

To optimize an objective function $m(q)$, assume $\dot{X}_2 = N_1\dot{q}$ as a sub-task with the desired trajectory $\dot{X}_{2d} = N_1(\alpha\nabla m)$ and the command acceleration as $\ddot{X}_{2c} = \ddot{X}_{2d} + K(\dot{X}_{2d} - \dot{X})$. Substitution of the above command in (6) gives the command joint acceleration as

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1N_1^\dagger[\dot{N}_1(\alpha\nabla m) + \alpha N_1H_m\dot{q} + KN_1(\alpha\nabla m - \dot{q}) - \dot{N}_1\dot{q}]. \quad (14)$$

where H_m is the Hessian matrix of $m(q)$. Using idempotency of N_1 and $N_1N_1^\dagger = N_1$, (11) can be written as

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1[\dot{N}_1(\alpha\nabla m - \dot{q}) + \alpha H_m\dot{q} + KN_1(\alpha\nabla m - \dot{q})]. \quad (15)$$

This command acceleration, besides the correct execution of the main task will minimize

$$\|\ddot{e} + K\dot{e}\|, \dot{e} = N_1(\alpha\nabla m - \dot{q}).$$

The command acceleration in (15) seems very similar to Hsu et al [10] redundancy resolution schemes which originally proposed the command acceleration

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1[\alpha H_m\dot{q} + KN_1(\alpha\nabla m - \dot{q}) - (J_1^\dagger J_1 J_1^\dagger + J_1^\dagger)J_1(\alpha\nabla m - \dot{q})]. \quad (16)$$

In the original paper, this command acceleration was obtained through a Lyapunov analysis. To find the relation between (15) and (16), after some algebraic manipulation, (16) can be rewritten as

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1[\alpha H_m\dot{q} + KN_1(\alpha\nabla m - \dot{q})] - N_1J_1^\dagger J_1(\alpha\nabla m - \dot{q}) \quad (17)$$

Substituting $N_1\dot{N}_1 = -J_1^\dagger J_1$ gives

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1[\alpha H_m\dot{q} + KN_1(\alpha\nabla m - \dot{q}) + \dot{N}_1(\alpha\nabla m - \dot{q})], \quad (18)$$

which is the same as in (15). The above analysis reveals that by a clever choice of sub-tasks the previous results in acceleration redundancy resolution can be recovered.

B. Null-Space Impedance Control

Another interesting choice for sub-task X_2 is $X_2 = q$ and the command impedance acceleration as $\ddot{X}_{2c} = \ddot{q}_d + M_d^{-1}(B_d\dot{q} + K_d\ddot{q} - \tau_{ext})$. Here M_d , B_d and K_d are the impedance parameters. The command joint acceleration and the closed-loop behavior are obtained as follows:

$$\ddot{q}_c = J_1^\dagger(\ddot{X}_{1c} - \dot{J}_1\dot{q}) + N_1[\ddot{q}_d + M_d^{-1}(B_d\dot{q} + K_d\ddot{q} - \tau_{ext})] \quad (19)$$

$$\ddot{X}_1 = \ddot{X}_{1c},$$

$$N_1[\ddot{q} + M_d^{-1}(B_d\dot{q} + K_d\ddot{q}) - M_d^{-1}\tau_{ext}] = 0,$$

where $\ddot{q} = \ddot{q}_d - \ddot{q}$. Actually this choice lets us realize joint space impedance despite the impedance of the main task, in the null-space of the main task. By a proper choice of null-space impedance matrices, it is possible to achieve a compliant behavior for the robot's body. This compliant behavior is useful in the case where the robot works in a cluttered environment and interaction may occur. In the simulation provided in this paper, this behavior is shown by applying a disturbance to the body of robot and also during contact with environment. The desired trajectory in the above equation can be chosen as a rest point or as a gradient of a given objective function.

When the torque measurement is not available, the closed-loop behavior is obtained as

$$\ddot{X}_{lc} - \ddot{X}_1 = J_1 M^{-1} \tau_{ext} \quad (20)$$

$$N_1[\ddot{q} + M_d^{-1}(B_d \dot{q} + K_d \tilde{q}) - M^{-1} \tau_{ext}] = 0.$$

Note that by using dynamically consistent generalized inverse $J_1^\# = M^{-1} J_1^T (J_1 M^{-1} J_1^T)^{-1}$ in the above formulation and choosing $M_d = M$ and also considering $MN_1^\# = N_1^{\#T} M$ and $N_1^\# = (I - J_1^\# J_1)$, following equation for null-space is obtained

$$N_1^{\#T} [M \ddot{q} + B_d \dot{q} + K_d \tilde{q} - \tau_{ext}] = 0. \quad (21)$$

C. Remarks

As we have seen in the previous subsections, with the proposed formulation of multi-priority control, it is possible to achieve control over the whole behavior of the robot and stable internal motion just through a wise choice of the sub-tasks. These sub-tasks can be chosen as the Cartesian coordinate of some points on the body of the robot in a hierarchical manner or as a kind of performance index. It is interesting to mention that, differently from other redundancy resolution approaches where the performance index is a function of only the configuration; this approach lets us to select a dynamic performance index as sub-task.

Until now we did not make any attempts about the nature of the specified tasks. In general, position and orientation control needs to be considered separately. While position control is rather straightforward, orientation control is more complex. Indeed, the task space command acceleration is given by

$$\ddot{X}_c = \begin{bmatrix} \ddot{p}_c \\ \ddot{\omega}_c \end{bmatrix}$$

where the task space positional command $\ddot{p}_c$ and angular command acceleration $\ddot{\omega}_c$ are defined based on the nature of operational task and the definition of orientation error [19].

IV. SINGULARITY TREATMENT IN MULTI-PRIORITY CONTROL

Because the acceleration level formulation is based on LS, the first remedy to treat singularity is the Damped Least-Squares (DLS) method. A complete analysis of using DLS in second order kinematic control has been performed in [20]. Using the DLS method at the acceleration level causes oscillatory motion near singularities. Here this problem is solved by introduction of yet a lower priority sub-task $X_3 = q$ with desired singularity configuration q_{ds} with the same priority as the singular tasks. Thus the command acceleration for two tasks is generated as

$$\begin{aligned} \ddot{q}_c = & J_1^\# (\ddot{X}_{lc} - \dot{J}_1 \dot{q}) \\ & + N_1 \bar{J}_2^{\dagger, \lambda_1} [\ddot{X}_{2c} - \dot{J}_2 \dot{q} - J_2 J_1^\# (\ddot{X}_{lc} - \dot{J}_1 \dot{q})] \\ & + \lambda_2 N_1 [-B \dot{q} + K(q_{ds} - q)] \end{aligned} \quad (22)$$

In this formulation λ_1 is damping factor and λ_2 is a weighting factor which can be a smooth function of the minimum singular value of the related projected Jacobian. In this way, inside the region of singularity, damping factors increase in proportion to the closeness of singularity. The

following choice for λ_1 [20] and λ_2 can be adopted which ensures continuity and good shaping of solution.

$$\lambda_1^2 = \begin{cases} 0 & \sigma \geq \varepsilon \\ (1 - (\frac{\sigma}{\varepsilon})^2) \lambda_M^2 & \sigma < \varepsilon \end{cases}$$

$$\lambda_2 = k \lambda_1$$

where σ is the singular value of related projected Jacobian, ε defines the size of singular region, λ_M suitably shapes the solution near the singularity and k is a constant.

Another method is the use of a singularity-robust formulation, which can also be proposed for the acceleration level, like for the velocity level, as follows

$$\ddot{q}_c = J_1^\# (\ddot{X}_{lc} - \dot{J}_1 \dot{q}) + N_1 J_2^\# [\ddot{X}_{2c} - \dot{J}_2 \dot{q}], \quad (23)$$

Because the projected Jacobian is not used, the algorithmic singularity does not occur, but this formulation has typically larger error on the lower-priority sub-tasks in comparison with (6).

V. COMPARISON WITH TORQUE LEVEL MULTI-PRIORITY CONTROL

Khatib et al [13,14] developed a whole-body control framework for prioritized multiple task control in humanoid robots. Hand location, mass center control and obstacle and joint limit avoidance are the common choices for tasks in a humanoid robot. They labeled all the behaviors not affecting the main task as posture space. This formulation allows for posture objectives to be controlled without dynamically interfering with the main (operational) task.

The control law originally proposed in [13], in the absence of external torque τ_{ext} , in order to obtain a decoupled behavior for the operational task X_t and the posture behavior X_p can be written as

$$\tau = \tau_{task} + \tau_{posture}, \quad (24)$$

$$\begin{aligned} \tau_{task} = & J_t^T [A_t (\ddot{X}_{lc} - \dot{J}_t \dot{q}) + J_t^{\#T} (C + g)] \\ \tau_{posture} = & J_{p|t}^T [A_{p|t} (\ddot{X}_{pc} - \dot{J}_{p|t} \dot{q} - \ddot{X}_{p|bias}) + J_{p|t}^{\#T} (C + g)] \end{aligned}$$

where J_p and J_t are the Jacobians associated to the operational and posture tasks, $J_{p|t} = J_p N_t$, N_t being the dynamically consistent null-space of task, $\ddot{X}_{lc}$ and $\ddot{X}_{pc}$ are the command accelerations for the task and posture spaces respectively, $\ddot{X}_{p|bias}$ is the acceleration induced by τ_{task} in the posture space and $A_t = (J_t M^{-1} J_t^T)^{-1}$ is the task inertia matrix; further, $A_{p|t} = (J_{p|t} M^{-1} J_{p|t}^T)^{-1}$ is the posture related inertia matrix and $J_t^\#$ is the dynamically consistent generalized inverse of J_t . The extension of the above formulation has been proposed in [14].

In the control law (24), Coriolis and gravitational terms are compensated in the operational and posture space. It is often more suitable to perform full gravity and Coriolis compensation in the joint space, as also highlighted in [12].

In this case further investigation shows that using a dynamically consistent generalized inverse in the acceleration based controller (6) gives exactly the same result as the torque level control.

VI. COMPUTER SIMULATION

A three-link planar arm in the horizontal plan as depicted in Fig. 1 is considered in the simulation study. When only the position of the end-effector is of concern, this system is a kinematically redundant manipulator.

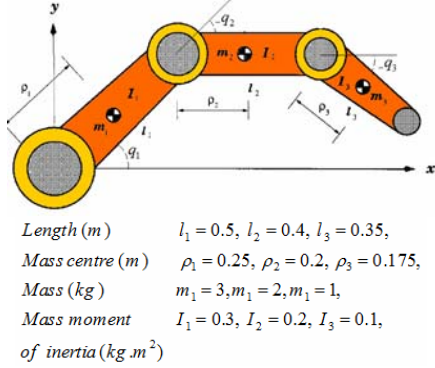


Fig 1. Planar redundant arm with the parameters of the three links

The first simulation illustrated in Fig. 2 shows multi-priority control when the arm has no contact with its environment. The main task is the end-effector trajectory tracking and the desired sub-task is the configuration $q_{\text{rest}} = [-\pi/2, \pi/2, \pi/2]$ which actually gives a sort of manipulability measure. Here the two tasks are always in conflict, yet we do not have problems with singularities. Usually a singularity may give troubles during transition from a nonsingular to a singular configuration because of the use of a pseudo-inverse in the solution.

The initial condition is $q_0 = [-0.031, 0.967, 1.934]$ and the desired trajectory is

$$x(t) = 0.4 + 0.06t$$

$$y(t) = 0.4 - 0.05t$$

Without any external disturbance and obstacle, we can see that null-space control attempts to reach the desired configuration in the LS sense.

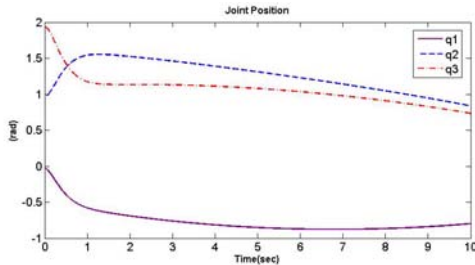


Fig 2. Joint position during null-space control

In order to show the null-space impedance control behavior, assume that the body of manipulator hits an obstacle in the presence of joint torque sensor (Fig. 3). In this situation the configuration of the arm changes to comply with the external interaction. Here the first-priority task error is zero thanks to the use of the torque sensor. Without this null-space impedance the command torques increase and the arm shows a stiff behavior regarding the obstacle.

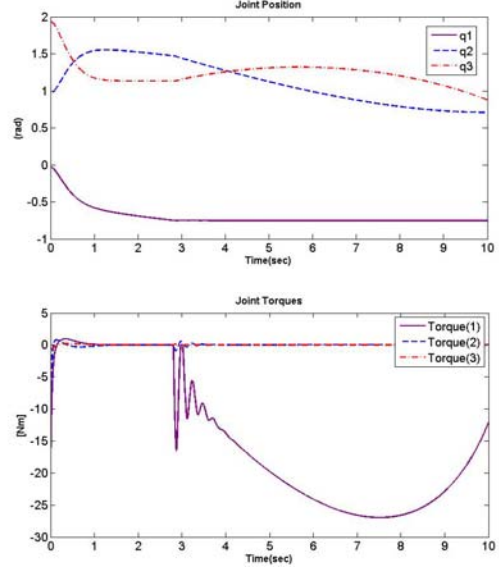


Fig 3. Joint position and joint torques under null-space impedance control during impact with an obstacle

The proportional and derivative gains for the main task are chosen as $K_{p1} = 400I, K_{d1} = 40I$, while the null-space impedance matrices are chosen as $M_d = I, B_d = 8I, K_d = 16I$. The obstacle is in the position -0.75 rad, with a stiffness of 6000Nm/rad .

In the next simulation illustrated in Fig. 4, the above analysis is performed without using torque sensor information in the controller. Equation (20) and (21) shows that in the absence of torque sensor, by a proper choice of impedance matrices and desired rest configuration, a satisfactory compliance behavior in the null-space can be achieved. Indeed there is an error on the main task during interaction which can be tolerated by using high gain for the main task control and giving high compliance to the null-space task through the impedance matrices.

In the last simulation an external disturbance force $f_{\text{ext}} = [20 \ 20]^T \text{N}$ is exerted on the body of manipulator for 0.2 second, in the middle of second link and the behavior of arm without using external disturbance information is illustrated in Fig. 5. Above simulation shows that null-space impedance control, under multi-priority control framework, enables us to have more control over the behavior of robot during interaction with environment, even in the case of unknown disturbance.

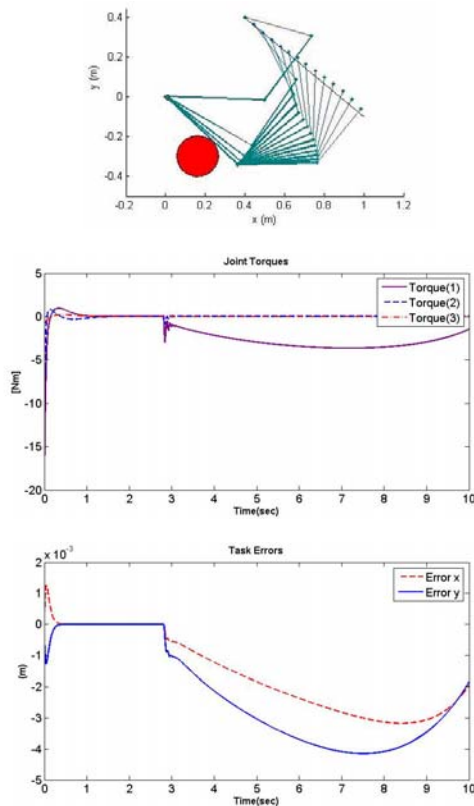


Fig 4. Arm configuration, joint torques and task errors under null impedance control during impact with obstacle without torque sensor

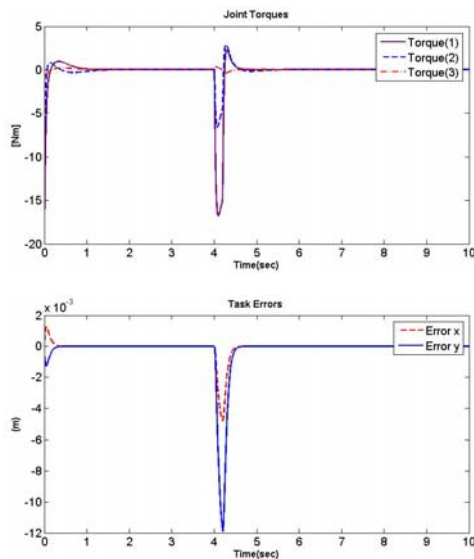


Fig 5. Joint torques and task errors under null impedance control during disturbance

VII. CONCLUSION

A new acceleration level multi-priority control algorithm has been presented in this paper. It has been shown how this formulation can be treated as a general framework to

achieve control over the whole body of robot. By a proper choice of the sub-tasks it is possible to recover some of the previously proposed results for acceleration level resolution. Null-space impedance control with possible solution to cope with singularities has been proposed as a result of task prioritization and the ability of this impedance to control the interaction of robot's body has been shown by simulation.

REFERENCES

- [1] Y. Nakamura, H. Hanafusa, T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. of Robotics Research*, 6(2), 3–15, 1987.
- [2] Y. Nakamura, *Advanced Robotics Redundancy and Optimization*, Addison-Wesley, 1991.
- [3] B. Siciliano, J.J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems", *5th International Conference on Advanced Robotics*, 1991.
- [4] Y. Nakamura, H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement, and Control*, 108, 163–171, 1986.
- [5] C. W. Wampler II, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, 16, 93–101, 1986.
- [6] D. Nenchev, Z. M. Sotirov, "Dynamic task-priority allocation for kinematically redundant robotic mechanisms," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [7] S. Chiaverini, "Singularity-robust task priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, 13, 398 – 410, 1997.
- [8] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE International Conference on Robotics and Automation*, 2008.
- [9] A. De Santis, P. Pierro, B. Siciliano, "The virtual end-effectors approach for human–robot interaction", in *Advances in Robot Kinematics*, J. Lenarcic, B. Roth (Eds.), Kluwer, Dordrecht, 133–144, 2006.
- [10] P. Hsu, J. Hauser, S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, 6, 133–148, 1989.
- [11] A. De Luca, G. Oriolo, and B. Siciliano, "Robot redundancy resolution at the acceleration level," *Laboratory Robotics and Automation*, 4, 97–106, 1992.
- [12] J. Nakanishi, R. Cory, M.J. Peters, S. Schaal, "Operational space control: a theoretical and empirical comparison," *International Journal of Robotics Research*, 27, 737–757, 2008.
- [13] O. Khatib, L. Sentis, J. H. Park, J. Warren, "Whole-body dynamic behavior and control of human-like robots," *International Journal of Humanoid Robotics*, 1, 29–43, 2004.
- [14] L. Sentis, O. Khatib, "A prioritized multi-objective dynamic controller for robots in human environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [15] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, 3, 1115–1120, 1987.
- [16] R. Platt, M. Abdallah, C. Wampler, "Multi-priority Cartesian impedance control," *Robotics Science and Systems*, 2010.
- [17] J. Park, Y. Choi, W. K. Chung, Y. Youm, "Multiple tasks kinematics using weighted pseudo-inverse for kinematically redundant manipulators," *IEEE International Conference on Robotics and Automation*, 2001.
- [18] G. Antonelli, F. Arrichiello, S. Chiaverini, "The null-space based behavioral control for autonomous robotic systems," *Intelligent Service Robotics*, 1, 27–39, 2008.
- [19] F. Caccavale, C. Natale, B. Siciliano, L. Villani, "Resolved acceleration control of robot manipulators: A critical review with experiments," *Robotica*, 16, 565–573, 1998.
- [20] F. Caccavale, S. Chiaverini, B. Siciliano, "Second-order kinematic control of robot manipulators with Jacobian damped least-squares inverse: Theory and experiments", *IEEE/ASME Transactions on Mechatronics*, 2, 188–194, 1997.