Robotica (2007) volume 25, pp. 511–520. © 2007 Cambridge University Press
doi:10.1017/S0263574707003402 Printed in the United Kingdom

Comparison of null-space and minimal null-space control algorithms

Bojan Nemec*, Leon Žlajpah and Damir Omrčen

Robotics Laboratory, Jožef Stefan Institute, Jamova 39, 1001 Ljubljana, Slovenia

(Received in Final Form: January 14, 2007. First published online: February 26, 2007)

SUMMARY

This paper deals with the stability of null-space velocity control algorithms in extended operational space for redundant robots. We compare the performance of the control algorithm based on the minimal null-space projection and generalized-inverse-based projection into the Jacobian null-space. We show how the null-space projection affects the performance of the null-space tracking algorithm. The results are verified with the simulation and real implementation on a redundant mobile robot composed of 3 degrees of freedom (DOFs) mobile platform and 7-DOF robot arm.

KEYWORDS: Redundant manipulator; Stability of the control algorithms; Autonomous motion; Obstacle avoidance; Mobile manipulator.

1. Introduction

One of the important issues of the new generation of service and humanoid robots is the kinematic redundancy. The kinematic redundancy is characterized by extra degrees of freedom (DOFs) with respect to the given motion posed by the assigned primary task. Most humanoid robots, as well as robotic arms mounted on a mobile platform, are kinematically redundant. A redundant manipulator has the ability to move the end-effector along the same trajectory using different configurations of the mechanical structure. This provides means for solving sophisticated motion tasks such as avoiding obstacles, avoiding singularities, optimizing manipulability, minimizing joint torques, etc. The consequence is a significant increase in the dexterity of the system, which is essential to accomplish complex tasks. Additionally, redundancy also has an important influence on the dynamic behavior of the robotic system. An appropriate control of dynamic properties is essential for higher performance. The majority of control algorithms proposed in the past decade are acceleration-based redundancy-resolution schemes, where appropriate joint accelerations are generated in order to accomplish the primary task and null-space motion as a secondary task. The behavior of a redundant manipulator is determined by the mapping of joint velocities into the Jacobian null-space. For redundant manipulators, the end-effector dynamics is only one part of the dynamics of the whole manipulator. The "rest" dynamic represents the dynamics of the internal motion of the manipulator, which has to be controlled in order to accomplish the desired secondary task. Velocity null-space control is an appropriate way to control internal motion. It is well established that certain acceleration-based control schemes exhibit instabilities, especially in instantaneous torque minimization redundancy-resolution schemes.[12,22] These instabilities were recently mathematically analyzed by O'Neil.[23] It has also been demonstrated that energy-dissipation controllers,[7] which can be interpreted as velocity null-space controllers with zero desired null-space velocity, can become unstable with low-velocity feedback gain.[23] The same result was published previously,[15] where Lyapunov functions were used to determine the range of stability of the velocity controller. We[16] proposed a modification that assures the stability of the controller for low gains of the null-space controller. Similar result was reported earlier.[14] Another approach based on dynamic decomposition of kinematically redundant manipulators into the task-space dynamics and null-space dynamics based on a minimally reparametrized homogenous velocity was proposed by Chang[2] and later by Park *et al.*[24] and Oh *et al.*[20] It has been shown earlier[25] that this approach does not exhibit instabilities, not even with instantaneous torque-minimization redundancy resolution.

This paper compares approaches based on generalized-inverse-based redundancy resolution at velocity level and minimal null-space redundancy resolution at velocity level. In both cases, we compared impedance force controller in extended task-space[20] as a more general control. We have highlighted the conceptual differences between both approaches. Furthermore, we analyze stability of algorithms based on a minimally parameterized null-space projection matrix subjected to the calculation of null-space projection matrix. We demonstrate the performance degradation of the control scheme due to the non-unique representation of null-space projection matrix. We solved this problem related to the real implementation by applying an appropriate method for calculation of the null-space projection matrix.

2. Kinematics

Robotic systems under study are n-DOF serial manipulators. We consider only redundant systems, which have more DOF than needed to accomplish the task, i.e. when the dimension of the joint-space n exceeds the dimension of the task-space m, $n > m$ and denote $r = n - m$ as the degree of redundancy. Let the configuration of the manipulator be represented

* Corresponding author. E-mail: bojan.nemec@ijs.si

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

by the vector $\boldsymbol{q}$ of n joint positions, and the end-effector position (and orientation) by m-dimensional vector $\boldsymbol{x}$ of task positions (and orientations). The relation between joint and task velocities is given by the following well-known expression:

$$\dot{\boldsymbol{x}} = \mathbf{J}\dot{\boldsymbol{q}} \tag{1}$$

where $\mathbf{J}$ is the $m \times n$ manipulator Jacobian matrix. The solution of the above equation for $\dot{\boldsymbol{q}}$ can be given as a sum of particular and homogeneous solution

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_p + \dot{\boldsymbol{q}}_h = \bar{\mathbf{J}}\dot{\boldsymbol{x}} + \mathbf{N}\xi \tag{2}$$

where

$$\bar{\mathbf{J}} = \mathbf{W}^{-1}\mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^{\mathrm{T}})^{-1} \tag{3}$$

where $\bar{\mathbf{J}}$ is the weighted generalized-inverse of $\mathbf{J}$, $\mathbf{W}$ is the weighting matrix, $\mathbf{N} = (\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})$ is $n \times n$ matrix representing the projection into the null-space of $\mathbf{J}$, and ξ is an arbitrary n-dimensional vector. We will denote this solution as the generalized-inverse-based redundancy resolution at the velocity level.[19] The homogenous part of the solution belongs to the Jacobian null-space. Therefore, we will denote it as $\dot{\boldsymbol{q}}_n$. Since the rank of the null-space matrix $\mathbf{N}$ is r, the homogeneous solution in Eq. (2) can also be presented in the form

$$\dot{\boldsymbol{q}}_n = \mathbf{N}\xi = \mathbf{V}\dot{\boldsymbol{x}}_n \tag{4}$$

where $\mathbf{V}$ is a full column rank $n \times r$ matrix which satisfies the criteria $\mathbf{J}\mathbf{V} = 0$, and $\dot{\boldsymbol{x}}_n$ is an arbitrary r-dimensional minimal null-space velocity vector. This approach will be denoted as minimal null-space redundancy resolution at velocity level.

3. Generalized-Inverse-Based Velocity Control (NSVC)

First we will derive the control law using generalized-inverse-based redundancy resolution at velocity level in the extended operational space. We denote this approach as null-space velocity control (NSVC). Extended operational space is an extension of the operational spaces introduced by Khatib[6] and offers unique approach for the analysis of both task-space and null-space.[2,20] Let us define extended-space variable $\boldsymbol{x}_e$ as

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \hdashline \xi \end{bmatrix} = \mathbf{J}_e\dot{\boldsymbol{q}} = \begin{bmatrix} \mathbf{J} \\ \hdashline \mathbf{N} \end{bmatrix}\dot{\boldsymbol{q}} \tag{5}$$

where $\mathbf{J}_e$ is the extended Jacobian. Since $\dot{\boldsymbol{q}} = \bar{\mathbf{J}}\dot{\boldsymbol{x}} + \mathbf{N}\xi$, the generalized inverse of $\mathbf{J}_e$ is defined as

$$\bar{\mathbf{J}}_e = \left[\begin{array}{c:c} \bar{\mathbf{J}} & \mathbf{N} \end{array}\right] \tag{6}$$

The proposed selection of the generalized inverse $\mathbf{J}_e^{-1}$ satisfies the generalized inversion property

$$\mathbf{J}_e\bar{\mathbf{J}}_e\mathbf{J}_e = \mathbf{J}_e \tag{7}$$

and inversion property

$$\bar{\mathbf{J}}_e\mathbf{J}_e = \bar{\mathbf{J}}\mathbf{J} + \mathbf{N}\mathbf{N} = \bar{\mathbf{J}}\mathbf{J} + \mathbf{I} - \bar{\mathbf{J}}\mathbf{J} = \mathbf{I} \tag{8}$$

The manipulator dynamics in joint-space is described by

$$\boldsymbol{\tau} = \mathbf{H}\ddot{\boldsymbol{q}} + \boldsymbol{h} + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} \tag{9}$$

where $\mathbf{H}$ is $n \times n$ inertia matrix of the manipulator, $\boldsymbol{h}$ is n-dimensional vector of centrifugal, coriolis, and gravity forces, and $\boldsymbol{F}$ is n-dimensional vector of external forces acting on the manipulator's end effector.

Premultiplying Eq. (9) by $\bar{\mathbf{J}}_e^{\mathrm{T}}$ and using $\dot{\boldsymbol{q}} = \bar{\mathbf{J}}_e\dot{\boldsymbol{x}}_e$, the equation of motion can be reformulated using the extended task-space variables as

$$\boldsymbol{f}_e = \boldsymbol{\Lambda}_e\ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e + \boldsymbol{F}_e \tag{10}$$

where

$$\boldsymbol{f}_e = \bar{\mathbf{J}}_e\boldsymbol{\tau} \tag{11}$$

$$\boldsymbol{\Lambda}_e = \bar{\mathbf{J}}_e^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}}_e = \left[\begin{array}{c:c} \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}} & \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\mathbf{N} \\ \hdashline \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\mathbf{N} & \mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N} \end{array}\right] \tag{12}$$

$$\boldsymbol{\mu}_e = \bar{\mathbf{J}}_e^{\mathrm{T}}\boldsymbol{h} - \boldsymbol{\Lambda}_e\dot{\mathbf{J}}_e\dot{\boldsymbol{q}} \tag{13}$$

and

$$\boldsymbol{F}_e = \begin{bmatrix} \boldsymbol{F} \\ \hdashline \mathbf{0} \end{bmatrix} \tag{14}$$

A straightforward approach to the controller design is to compensate nonlinear coupling terms of the system dynamics and apply a control vector, which assures the desired system dynamics. Let us define the control force in extended operational space as

$$\boldsymbol{f}_c = \boldsymbol{\Lambda}_e\ddot{\boldsymbol{x}}_c + \boldsymbol{\mu}_e + \boldsymbol{F}_e \tag{15}$$

where $\ddot{\boldsymbol{x}}_c$ denotes the control vector in the form

$$\ddot{\boldsymbol{x}}_c = \begin{bmatrix} \ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x \\ \hdashline \ddot{\boldsymbol{q}}_{nd} + \mathbf{K}_n\dot{\boldsymbol{e}}_n \end{bmatrix} \tag{16}$$

Here, $\boldsymbol{e}_x = \boldsymbol{x}_d - \boldsymbol{x}$ is the task-space tracking error and $\dot{\boldsymbol{e}}_n = \dot{\boldsymbol{q}}_{nd} - \dot{\boldsymbol{q}}_n$ is the null-space tracking error. $\boldsymbol{x}_d$ and $\dot{\boldsymbol{q}}_{nd}$ are the desired task coordinates and null-space velocity, respectively. Inserting Eq. (15) into (10) yields

$$\boldsymbol{\Lambda}_e(\ddot{\boldsymbol{x}}_c - \ddot{\boldsymbol{x}}_e) = 0 \tag{17}$$

The general form of the extended operational space inertia matrix $\boldsymbol{\Lambda}_e$ contains off-diagonal terms, which means that the task-space and the null-space are inertially coupled. By selecting $\mathbf{W} = \mathbf{H}$ in Eq. (3), we obtain

$$\begin{aligned} \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\mathbf{N} &= (\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}})^{-1}\mathbf{J}\mathbf{H}^{-1}\mathbf{H}(\mathbf{I} - \mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}})^{-1}\mathbf{J} \\ &= (\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}})^{-1}\mathbf{J} - (\mathbf{J}\mathbf{H}^{-1}\mathbf{J}^{\mathrm{T}})^{-1}\mathbf{J} = 0 \end{aligned} \tag{18}$$

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

This shows that the inertia-weighted generalized inverse is the only one that decouples task-space and null-space motion[4]. Equation (17) is thus decoupled into two equations

$$\bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}}(\ddot{\boldsymbol{e}}_x + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x) = 0$$
$$\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N}(\ddot{\boldsymbol{e}}_n + \mathbf{K}_n\dot{\boldsymbol{e}}_n) = 0 \tag{19}$$

Since matrix $\bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}}$ is positive definite, it follows that $\ddot{\boldsymbol{e}}_x + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x = 0$. In contrast, the matrix $\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N}$ is only positive semi-definite and this does not imply that $\ddot{\boldsymbol{e}}_n + \mathbf{K}_n\dot{\boldsymbol{e}}_n = 0$. Using identity $\mathbf{N}^{\mathrm{T}}\mathbf{H}\mathbf{N} = \mathbf{H}\mathbf{N}$ and a positive definitness of inertia matrix $\mathbf{H}$ yields $\mathbf{N}(\ddot{\boldsymbol{e}}_n + \mathbf{K}_n\dot{\boldsymbol{e}}_n) = 0$. Therefore, null-space projection of the null-space tracking error tends to zero, but this does not imply that null-space tracking error itself tends to zero. This is also the cause of instabilities of acceleration-based redundancy-resolution schemes which use a velocity null-space controller. Namely, O'Neil[23] proved that the energy-dissipation scheme proposed by Khatib[7] cannot guarantee the stability at low gains of the dissipation energy controller. Similar results were obtained earlier[15] by using the Lyapunov stability criterium. By setting the desired null-space velocity to zero, the null-space velocity controller reduces to an energy-dissipation controller.

The joint-space control law can be derived by inserting Eq. (16) into Eq. (15) and premultiplying by $\mathbf{J}_e$

$$\tau_c = \mathbf{H}\bar{\mathbf{J}}(\ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x - \dot{\mathbf{J}}\dot{\boldsymbol{q}})$$
$$+ \mathbf{H}\mathbf{N}(\ddot{\boldsymbol{q}}_{nd} + \mathbf{K}_n\dot{\boldsymbol{e}}_n - \dot{\mathbf{N}}\dot{\boldsymbol{q}}) + \boldsymbol{h} + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} \tag{20}$$

The first term corresponds to the task-space control τ_x, the second to the null-space control τ_n, and the third and the fourth to the compensation of the nonlinear system dynamics and external force, respectively. Using the identity $\mathbf{N}\dot{\mathbf{N}} = -\mathbf{N}\dot{\bar{\mathbf{J}}}\bar{\mathbf{J}}$, $\ddot{\boldsymbol{q}}_{nd} = \mathbf{N}\ddot{\boldsymbol{q}}_d + \dot{\mathbf{N}}\dot{\boldsymbol{q}}_d$ and introducing joint-space error $\boldsymbol{e}_q = \boldsymbol{q}_d - \boldsymbol{q}$, the null-space control law can be rewritten as

$$\tau_n = \mathbf{H}\mathbf{N}(\ddot{\boldsymbol{q}}_d + \mathbf{K}_n\dot{\boldsymbol{e}}_q + \dot{\mathbf{N}}\dot{\boldsymbol{e}}_q)$$
$$= \mathbf{H}\mathbf{N}(\ddot{\boldsymbol{q}}_d + \mathbf{K}_n\dot{\boldsymbol{e}}_q - \dot{\bar{\mathbf{J}}}\bar{\mathbf{J}}\dot{\boldsymbol{e}}_q) \tag{21}$$

4. Minimal Null-Space-Based Velocity Control (MNSVC)

The transformation from joint coordinates to minimal null-space velocities is described by

$$\dot{\boldsymbol{x}}_n = \bar{\mathbf{V}}\dot{\boldsymbol{q}} \tag{22}$$

where $\bar{\mathbf{V}}$ is the generalized inverse of $\mathbf{V}$ and is defined as

$$\bar{\mathbf{V}} = (\mathbf{V}^{\mathrm{T}}\mathbf{W}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{W} \tag{23}$$

where $\mathbf{W}$ is the weighting matrix. Using the above formulation, we can define the extended-space $\boldsymbol{x}'_e$ as

$$\dot{\boldsymbol{x}}'_e = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \hdashline \dot{\boldsymbol{x}}_n \end{bmatrix} = \mathbf{J}'_e\dot{\boldsymbol{q}} = \begin{bmatrix} \mathbf{J} \\ \hdashline \bar{\mathbf{V}} \end{bmatrix}\dot{\boldsymbol{q}} \tag{24}$$

where $\mathbf{J}'_e$ is the extended Jacobian. The symbol $'$ is used to denote that the corresponding variable is defined in minimal null-space in contrast to the variables without the suffix, which are defined in generalized-inverse-based null-space.

Since $\dot{\boldsymbol{q}} = \bar{\mathbf{J}}\dot{\boldsymbol{x}} + \mathbf{V}\dot{\boldsymbol{x}}_n$, the inverse of $\mathbf{J}'_e$ is defined as

$$\bar{\mathbf{J}}'_e = \left[\begin{array}{c:c} \bar{\mathbf{J}} & \mathbf{V} \end{array}\right] \tag{25}$$

The null-space matrix $\mathbf{N}$ and minimal null-space matrix $\mathbf{V}$ are related through

$$\mathbf{N} = \mathbf{V}\bar{\mathbf{V}} \tag{26}$$

The above relation relies on the definition of $\mathbf{V}$, and is easily verified by inserting Eq. (22) into Eq. (2). Similar to the previous case, the equation of motion can be reformulated using the extended task-space variables

$$\boldsymbol{f}'_e = \boldsymbol{\Lambda}'_e\ddot{\boldsymbol{x}}'_e + \boldsymbol{\mu}'_e + \boldsymbol{F}_e \tag{27}$$

where

$$\boldsymbol{f}'_e = \bar{\mathbf{J}}'_e\boldsymbol{\tau} \tag{28}$$

$$\boldsymbol{\Lambda}'_e = \bar{\mathbf{J}}'_e\mathbf{H}\bar{\mathbf{J}}'_e = \left[\begin{array}{c:c} \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}} & \bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\mathbf{V} \\ \hdashline \mathbf{V}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}} & \mathbf{V}^{\mathrm{T}}\mathbf{H}\mathbf{V} \end{array}\right] \tag{29}$$

$$\boldsymbol{\mu}'_e = \bar{\mathbf{J}}'^{\mathrm{T}}_e\boldsymbol{h} - \boldsymbol{\Lambda}'_e\dot{\mathbf{J}}'_e\dot{\boldsymbol{q}} \tag{30}$$

and

$$\boldsymbol{F}_e = \begin{bmatrix} \boldsymbol{F} \\ \hdashline \mathbf{0} \end{bmatrix} \tag{31}$$

Let us define control vector in extended-space in the same way we did it in generalized-inverse-based controller.

$$\boldsymbol{f}'_c = \boldsymbol{\Lambda}'_e\ddot{\boldsymbol{x}}'_c + \boldsymbol{\mu}'_e + \boldsymbol{F}_e \tag{32}$$

Inserting Eq. (32) into (27) yields

$$\boldsymbol{\Lambda}'_e(\ddot{\boldsymbol{x}}'_c - \ddot{\boldsymbol{x}}'_e) = 0 \tag{33}$$

where $\ddot{\boldsymbol{x}}'_c$ denotes the control vector in the form

$$\ddot{\boldsymbol{x}}'_c = \begin{bmatrix} \ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x \\ \hdashline \ddot{\boldsymbol{x}}_{nd} + \mathbf{K}_n\dot{\boldsymbol{e}}'_n \end{bmatrix} \tag{34}$$

Variable $\dot{\boldsymbol{e}}'_n = \dot{\boldsymbol{x}}_{nd} - \dot{\boldsymbol{x}}_n$ denotes the velocity tracking error in minimal null-space. Again, by selecting $\mathbf{W} = \mathbf{H}$ in Eq. (23), the off-diagonal elements of the extended inertia matrix are zero. Equation (34) is thus decoupled into two equations

$$\bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}}(\ddot{\boldsymbol{e}} + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x) = 0$$
$$\mathbf{V}^{\mathrm{T}}\mathbf{H}\mathbf{V}(\ddot{\boldsymbol{e}}'_n + \mathbf{K}_n\dot{\boldsymbol{e}}'_n) = 0 \tag{35}$$

Since both matrices $\bar{\mathbf{J}}^{\mathrm{T}}\mathbf{H}\bar{\mathbf{J}}$ and $\mathbf{V}^{\mathrm{T}}\mathbf{H}\mathbf{V}$ are positive definite, it follows $\ddot{\boldsymbol{e}}_x + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x = 0$ and $\ddot{\boldsymbol{e}}'_n + \mathbf{K}_n\dot{\boldsymbol{e}}'_n = 0$. The

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

error equation shows the main advantage of the minimal null-space approach. Only minimal null-space approach assures the desired dynamic behavior in the null-space, which cannot be guaranteed for the generalized-inverse-based controller. The reason for this is in the existence of the generalized inverse of minimal null-space transformation matrix $\mathbf{V}$. On the contrary, matrix $\mathbf{N}$ is rank deficient and inverse of $\mathbf{N}$ is singular. It was proved by means of O'Neil identity that minimal null-space acceleration redundancy-resolution schemes are not subjected to torque instabilities.[25] However, in the next section, we will show that performance degradation of the control algorithm arises due to the non-unique representation of null-space projection matrix.

Again, the joint-space control law can be simplified to

$$\tau_c = \mathbf{H}\bar{\mathbf{J}}(\ddot{\boldsymbol{x}}_d + \mathbf{K}_v\dot{\boldsymbol{e}}_x + \mathbf{K}_p\boldsymbol{e}_x - \dot{\mathbf{J}}\dot{\boldsymbol{q}}) + \mathbf{H}\mathbf{V}(\ddot{\boldsymbol{x}}_{nd} + \mathbf{K}_n\dot{\boldsymbol{e}}'_n - \dot{\bar{\mathbf{V}}}\dot{\boldsymbol{q}}) + \boldsymbol{h} + \mathbf{J}^{\mathrm{T}}\boldsymbol{F} \quad (36)$$

The first term corresponds to the task-space control τ_x, the second to the null-space control τ_n, and the third and the fourth to the compensation of nonlinear system dynamics and external force, respectively. Comparing Eqs. (20) and (36), we notice that MNSVC and NSVC algorithms differ only in null-space control. Let us rewrite second term of Eq. (36) using $\dot{\boldsymbol{x}}_n = \bar{\mathbf{V}}\dot{\boldsymbol{q}}$ and Eq. (26) into

$$\tau_n = \mathbf{H}(\mathbf{N}(\ddot{\boldsymbol{q}}_d + \mathbf{K}_n\dot{\boldsymbol{e}}_q) + \mathbf{V}\dot{\bar{\mathbf{V}}}\dot{\boldsymbol{e}}_q) \quad (37)$$

From the above equation, it is evident that the only difference between NSVC and MNSVC is how they compensate the projection of the joint-space error to the null-space due to the configuration change. NSVC uses term $\mathbf{N}\dot{\mathbf{N}}\dot{\boldsymbol{e}}_q$, while MNSVC compensate this effect with the term $\mathbf{V}\dot{\bar{\mathbf{V}}}\dot{\boldsymbol{e}}_q$.

5. Minimal Null-Space Calculation

There is an infinite number of possible null-space transformations. We have shown[17] that the null-space motion depends only on the criteria function and the selected weighting matrix. Therefore, it is independent of the selection of the null-space transformation $\mathbf{V}$. On the other hand, the numerical stability of the control algorithm is subjected to the selection of $\mathbf{V}$. Namely, representation of the null-space with the base vectors is not unique. There is an infinite number of orthonormal basis vectors $\mathbf{V}$ that describe the same null-space. For good control, it is necessary to obtain a smooth continuous solution of $\mathbf{V}$ during the execution of the robot's task.

There are several methods for obtaining $\mathbf{V}$. The method proposed by Park *et al.*[24] uses singular value decomposition (SVD) of $\mathbf{J}$, or alternatively, $\mathbf{J}^{\mathrm{T}}\mathbf{J}$. Singular value decomposition or $\mathbf{J}$ yields

$$\mathbf{U}\boldsymbol{\Sigma}\mathbf{Z}^{\mathrm{T}} = \mathbf{J} \quad (38)$$

where $\boldsymbol{\Sigma}$ is the diagonal n-dimensional matrix of m nonzero eigenvalues denoted by s and $n-m$ zero eigenvalues of $\mathbf{J}$. The corresponding matrices $\mathbf{Z}$ and $\mathbf{U}$ have form $\mathbf{Z} = \left[\begin{array}{c}\mathbf{R}\\ \hline \mathbf{V}\end{array}\right]$ and $\mathbf{U}^{\mathrm{T}} = [\mathbf{Q}|\mathbf{V}^{\mathrm{T}}]$. Since matrices $\mathbf{U}$ and $\mathbf{Z}$ are unitary, it follows

$$\mathbf{U}^{\mathrm{T}}\mathbf{J}\mathbf{Z} = \Sigma$$

and

$$\left[\begin{array}{c|c}\mathbf{Q} & \mathbf{V}^{\mathrm{T}}\end{array}\right]\mathbf{J}\left[\begin{array}{c}\mathbf{R}\\ \hline \mathbf{V}\end{array}\right] = \left[\begin{array}{ccc|c} s_1 & & & \\ & s_2 & & \\ & & \ddots & \mathbf{0} \\ & & & s_m \\ \hline & \mathbf{0} & & \mathbf{0}\end{array}\right] \quad (39)$$

Obviously, $\mathbf{QJV} = \mathbf{0}$, $\mathbf{V}^{\mathrm{T}}\mathbf{JR} = \mathbf{0}$, and since $\mathbf{Q}$ and $\mathbf{R}$ are nonzero matrices, $\mathbf{JV} = \mathbf{0}$, $\mathbf{V}^{\mathrm{T}}\mathbf{J} = \mathbf{0}$ and thus sub-matrix $\mathbf{V}$ forms null-space of $\mathbf{J}$. Unfortunately, matrix $\mathbf{V}$ is not unique. There is an infinite number of orthonormal basis vectors $\mathbf{V}$ that describe the same null-space.

The most popular technique for computing the SVD is the Golub–Reunsch algorithm and is available in many linear algebra software packages. It is regarded as the most efficient and numerically stable technique for computing the SVD of an arbitrary matrix. Unfortunately, it does not assure continuous solutions. For example, if matrix $\mathbf{J}$ changes continuously, this does not imply that matrices $\mathbf{U}$, $\boldsymbol{\Sigma}$, and $\mathbf{Z}$ will also change continuously. We will demonstrate this effect with the simulation of 500 instances of the kinematics of the 4-DOF planar manipulator with links of equal length. The $\mathbf{V}$ matrix was calculated using Matlab function `null`, which is based on SVD calculation using Golub–Reunsch algorithm. The null-space velocity $\boldsymbol{x}_n = \mathbf{V}[1, -1, 1, -1]$ was applied to initiate null-space motion. At $\boldsymbol{q} = [2.8274, 2.3229, 1.4770, 0.0284]$, $\mathbf{V}$ suddenly changes the set of values. If we reverse the null-space motion, $\mathbf{V}$ changes again. This is shown in Fig. 1. Discontinuity of elements in $\mathbf{V}$ causes performance degradation of the control algorithm. Namely, control algorithm 36 requires $\dot{\bar{\mathbf{V}}}$,

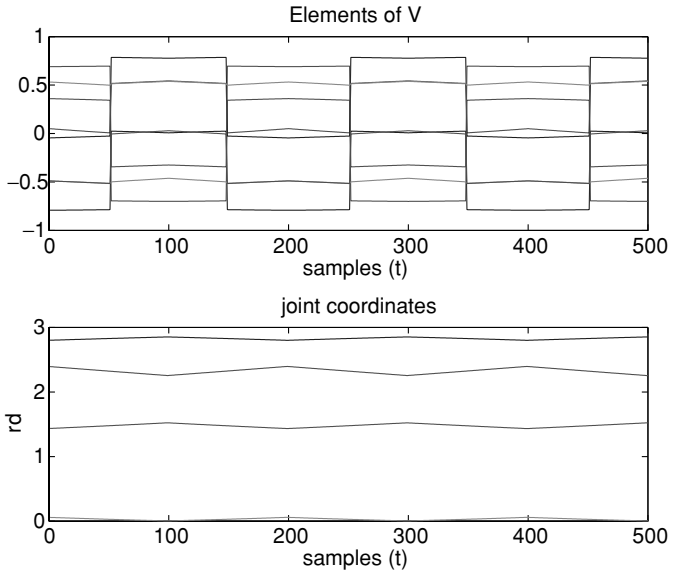


Fig. 1. Elements of $\mathbf{V}$ and joint angles $\boldsymbol{q}$.

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

which has to be differentiated numerically. Discontinuity of $\mathbf{V}$ results in an unbounded control signal.

We solved the above problem using SVD algorithm based on Givens rotations.[18] The approach was reviewed by Maciejewski and Klein[11] as an algorithm, more suited to take advantage of incremental perturbations and parallel architectures. For our purpose, we do not need to calculate all matrices of SVD. We need only the matrix $\mathbf{Z}$, which orthogonalizes the columns of $\mathbf{J}$. This matrix is usually formed as a product of successive Givens rotations, each orthogonalizes two columns. Considering the current ith and jth columns of $\mathbf{J}$, a multiplication by Givens rotation results in new columns

$$\begin{aligned}\mathbf{J}_i^* &= \mathbf{J}_i \cos(\theta) + \mathbf{J}_j \sin(\theta)\\ \mathbf{J}_j^* &= \mathbf{J}_j \cos(\theta) - \mathbf{J}_i \sin(\theta)\end{aligned} \tag{40}$$

with constraint $\mathbf{J}_i^* \mathbf{J}_j^* = 0$. The terms in the Givens rotation matrix which ortogonalizes $\mathbf{J}$ can be computed by using the following formulas:[11,13]

$$p = \mathbf{J}_i^{\mathrm{T}} \mathbf{J}_j \tag{41}$$

$$q = \mathbf{J}_i^{\mathrm{T}} \mathbf{J}_i - \mathbf{J}_j^{\mathrm{T}} \mathbf{J}_j \tag{42}$$

$$v = \sqrt{4p^2 + q^2} \tag{43}$$

For $q \geq 0$, the rotation matrix elements are

$$\cos(\theta) = \sqrt{\frac{v+q}{2v}} \tag{44}$$

$$\sin(\theta) = \frac{p}{v \cos(\theta)} \tag{45}$$

and for $q < 0$, we can use another set of elements in order to avoid ill-condition.

$$\sin(\theta) = \mathrm{sig}(p)\sqrt{\frac{v-q}{2v}} \tag{46}$$

$$\cos(\theta) = \frac{p}{v \sin(\theta)} \tag{47}$$

However, orthogonalization cannot be achieved in single sweep. In general, we need multiple sweeps, but the algorithm converges.[11,13] Perhaps, the most useful property of the algorithm is the ability to use perturbed initial values of matrix $\mathbf{Z}$. The more orthogonal are the columns of $\mathbf{JZ}$, the fewer are the number of sweeps required for the convergence, and even more important in our case, the solutions are continuous. If one considers the current $\mathbf{J}$ to be a perturbation of the previous $\mathbf{J}$, $\mathbf{J}(t+\delta t) = \mathbf{J}(t) + \delta\mathbf{J}(t)$, then the matrix $\mathbf{J}(t+\delta t)\mathbf{Z}(t)$ will have nearly orthogonal columns. Since control of the manipulator consists of subsequent calculations of Eq. (36), we can use the solution of $\mathbf{Z}$ from the previous step, which improves the convergence of the algorithm, reduces computational burden, and assures contiguous solution of the Jacobian null-space matrix $\mathbf{V}$, which is most important. In real implementation, special care should be paid to the orthogonality test of columns $\mathbf{J}$. Norm is not a good measure because $\mathbf{J}_i^{\mathrm{T}*}\mathbf{J}_j^*$ could be small simply, because of small eigenvalues. Nash[13] has proposed norm $\frac{\mathbf{J}_i^{*\mathrm{T}}\mathbf{J}_j^*}{(\mathbf{J}_i^{*\mathrm{T}}\mathbf{J}_i^*)(\mathbf{J}_j^{*\mathrm{T}}\mathbf{J}_j^*)}$. Unfortunately, when the denominator is equal to zero, Eq. (45) is also singular. In such a case, we have found an appropriate solution by perturbation of the Jacobian with small random values. For the illustration, we repeated previous experiment again, this time using the proposed algorithm. In Fig. 2, we can observe smooth transition of elements of $\mathbf{V}$.

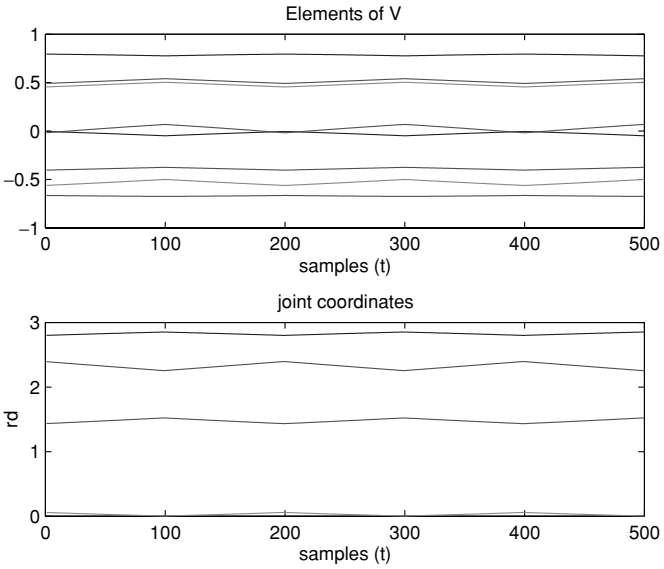


Fig. 2. Elements of $\mathbf{V}$ and joint angles $\boldsymbol{q}$.

6. Null-Space Motion Generation

The desired extended null-space velocities, which minimize the given criteria $\boldsymbol{p}$, can be obtained using the weighted gradient optimization procedure[1,15]

$$\dot{\boldsymbol{q}}_n = \mathbf{N}\mathbf{H}^{-1}\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} k_o \tag{48}$$

which assures the best optimization step in the case of inertia weighted generalized inverse. k_o is a negative constant and defines the optimization step.

The force and the position tracking are usually of the highest priority for a force-controlled robot. The selection of the subtasks with lower priority depends on the specific application. However, collision avoidance is of great importance in most applications of redundant robot systems, since it is very difficult to predict the path of all links. In most cases, the motion is not guaranteed to be conservative. Therefore, one collision-free task cycle does not imply next collision-free cycle.

Following the idea of the obstacle avoidance using the potential field,[5] we define the cost function $\boldsymbol{p} = \frac{1}{2}\mathbf{E}d_0^2$, where $\mathbf{E}$ is an $l \times l$ rotation matrix describing the direction of an artificial potential field pointing from the obstacle, l is the dimension of the position sub-space, and d_0 is the shortest distance between obstacle and the robot body. In our case, the desired objective is fulfilled, if the imaginary force is applied only on robot joints. In this case, we can obtain cost function

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

gradient in simple form as

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} = (\boldsymbol{d}_1 \mathbf{J}^{0,1} + \boldsymbol{d}_2 \mathbf{J}^{0,2} + \cdots + \boldsymbol{d}_{n-1} \mathbf{J}^{0,n-1})\mathbf{E} \qquad (49)$$

where $\boldsymbol{d}_i$ is the vector of the shortest distances between the ith joint and the obstacle and $\mathbf{J}^{0,i}$ denotes the Jacobian matrices between the base (the first index in the superscript) and the ith joint (the second index in the superscript) regarding the robot positions only.

In our experiments, we will also use singularity avoidance. A suitable measure for determining vicinity of singular point was proposed by Yoshikawa[27] and is described by

$$\boldsymbol{p}_s = \sqrt{|\mathbf{J}\mathbf{J}^{\mathrm{T}}|} \qquad (50)$$

Unfortunately, partial derivation $\frac{\partial \mathbf{J}}{\partial \boldsymbol{q}}\mathbf{J}^{\mathrm{T}}$ required by the gradient optimization (48) is generally not easy to calculate in an analytical way. Therefore, we have used numerical derivative in our experiments.

7. Experiments

The experimental setup consists of 7-DOF robot arm Mitsubishi PA10, mounted on the holonomous mobile platform Nomad XR4000 with 3 DOFs. The entire setup is presented in Fig. 3. The robot arm is torque controlled using ArcNet protocol. Unfortunately, the mobile platform has no torque input, and can only be velocity controlled. Therefore, a control algorithm for this system has to be modified.[21] Unfortunately, this modification does not allow to completely decouple task-space and null-space dynamics, and both algorithms have virtually equal response. Therefore, we compared simulation results of both algorithms. Simulation was accomplished in Matlab/Simuling and accurate dynamic models were developed using SDFast tool. The primary task of the manipulator was to track the line. The desired speed was 0.45 m/s and the initial joint configuration of the robot was $[-0.5, 0, 0, 0, \pi/2, 0, -\pi/2, 0, 0, 0]$. There was an obstacle in the robot work-space, as shown in Fig. 3. The secondary subtask was obstacle avoidance and singularity avoidance. Regarding the given task, the degree of redundancy was 7. A high degree of redundancy was selected in order to verify that the matrix $\dot{\bar{\mathbf{V}}}$ is continuous and limited during the motion. On the other hand, a high degree of redundancy combined with mobile platform requires careful selection of the secondary tasks. Tasks such as manipulability optimization or torque optimization will always lock the robot arm into the optimal position and the task motion will be performed with mobile base only. In order to avoid such a situation, singularity avoidance algorithm was activated only if the robot was close enough to the singular configuration. The simulation results of the given task using NSVC and MNSVC control algorithms are presented in Figs. 4 and 5, respectively. The task tracking error, the null-space velocity error, and the desired null-space velocity as a result of the optimization procedure. As expected, task-space errors are identical and practically equal to zero in both cases. The null-space tracking error increases at the moment when singularity avoidance, and later, obstacle avoidance generate null-space motion. We can see that, although the null-space tracking error is smaller in the case of MNSVC, there is no significant difference in robot motion in the case when NSVC and MNSVC algorithm is used. This is because the null-space velocity feedback gain $\mathbf{K}_n$ stabilizes null-space control loop. When comparing null-space tracking errors of both algorithms, we must be aware that null-space errors are not presented in the same space, and therefore, direct comparison is not possible.

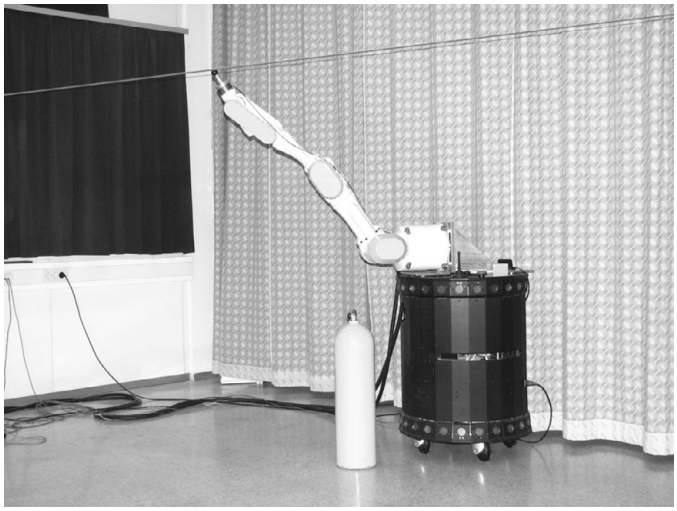


Fig. 3. Service robot during the task.

We also compared the performance of both algorithms on a real robot. As we mentioned before, the mobile platform has no torque input. Therefore, we implemented both algorithms only on PA10 robot arm. We used inertia parameters, damping and friction parameters as published previously.[8,26] The sampling time in our experiments was 0.002 s. Again, the task of the robot was to track a line with speed 0.2 m/s. In this experiment, only positions were considered in the primary task. Therefore, the degree of redundancy was 4. In order to compare the performance of both algorithms equally, we explicitly generated the secondary motion. The secondary motion was sinusoidal velocity with amplitude 0.5 rad/s and frequency 0.6 s^{-1}, applied to joints 6 and 7 of the manipulator. The trajectory was completed in 5 s, then we left the robot to perform only the secondary motion for another 5 s. The primary task tracking errors, null-space tracking errors, and compensation signals $\dot{\bar{\mathbf{N}}}\dot{\boldsymbol{q}}$ and $\dot{\bar{\mathbf{V}}}\dot{\boldsymbol{q}}$ for the NSVC and the MNSVC control algorithms are presented in Figs. 6 and 7, respectively. As expected, task errors are similar in both cases, while the null-space tracking errors are significantly lower with the MNSVC algorithm. As we mentioned previously, it is difficult to compare null-space errors of both algorithms, since they are presented in different null-spaces. Therefore, in figures we projected minimal null-space errors into the generalized-inverse-based null-space, using the equation $\dot{\boldsymbol{e}}_n = \mathbf{V}\dot{\boldsymbol{e}}'_n$. From the results, it can be clearly seen that the NSVC algorithm does not fully compensates the influence of the joint motion, which is, according to Eqs. (21) and (37), the only difference between both algorithms. We also compared the desired and the obtained secondary velocity.

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

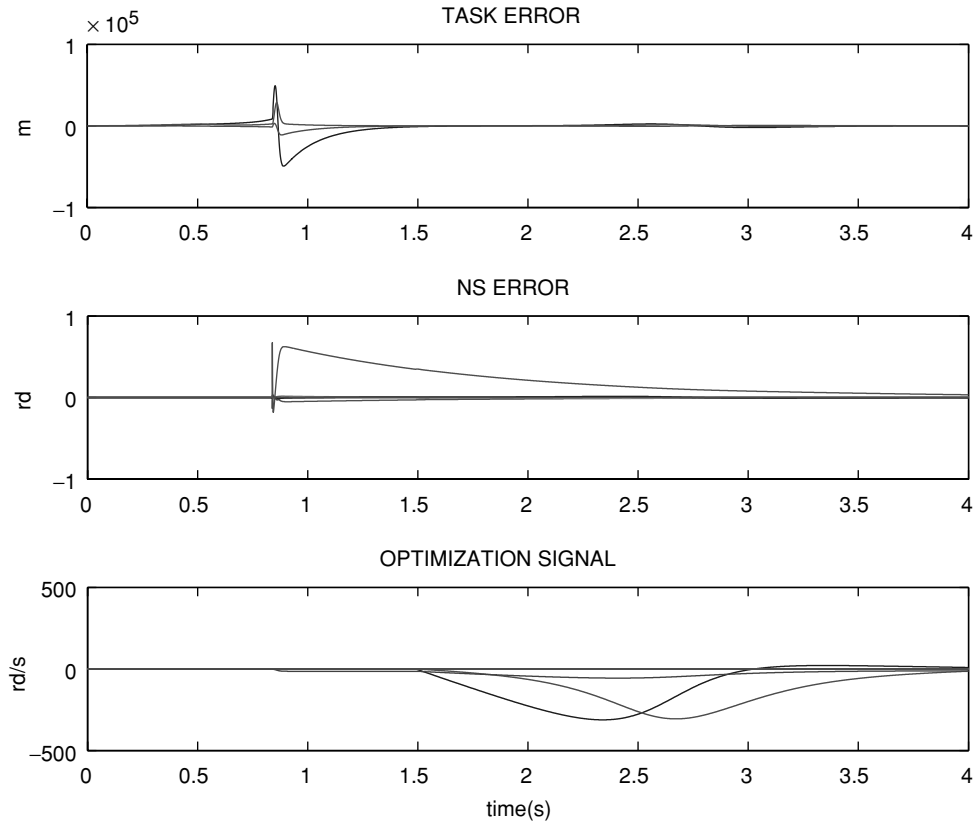


Fig. 4. Simulation results with NSVC.

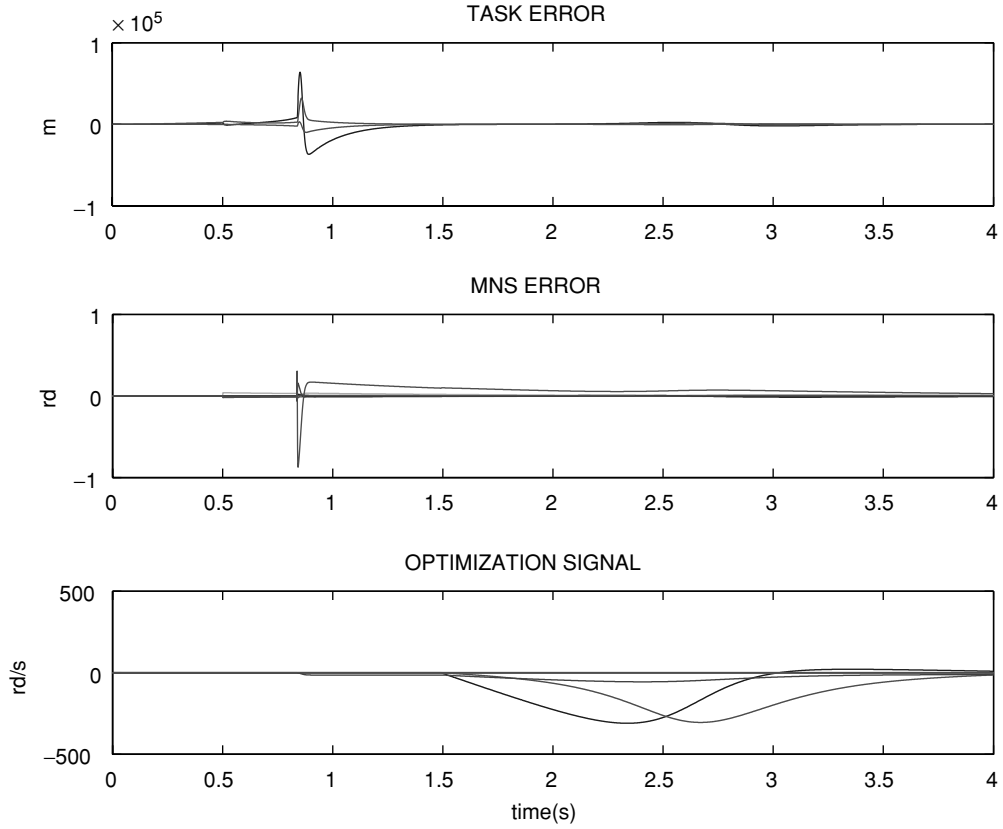


Fig. 5. Simulation results with MNSVC.

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

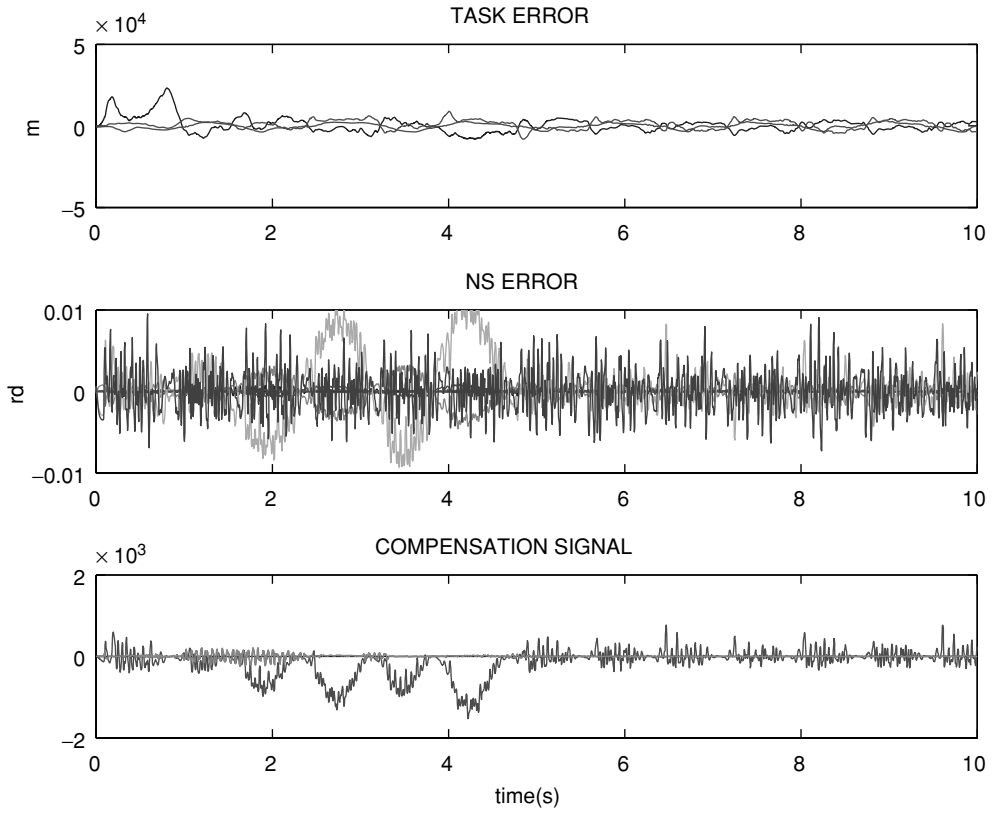


Fig. 6. Experimental results with NSVC.

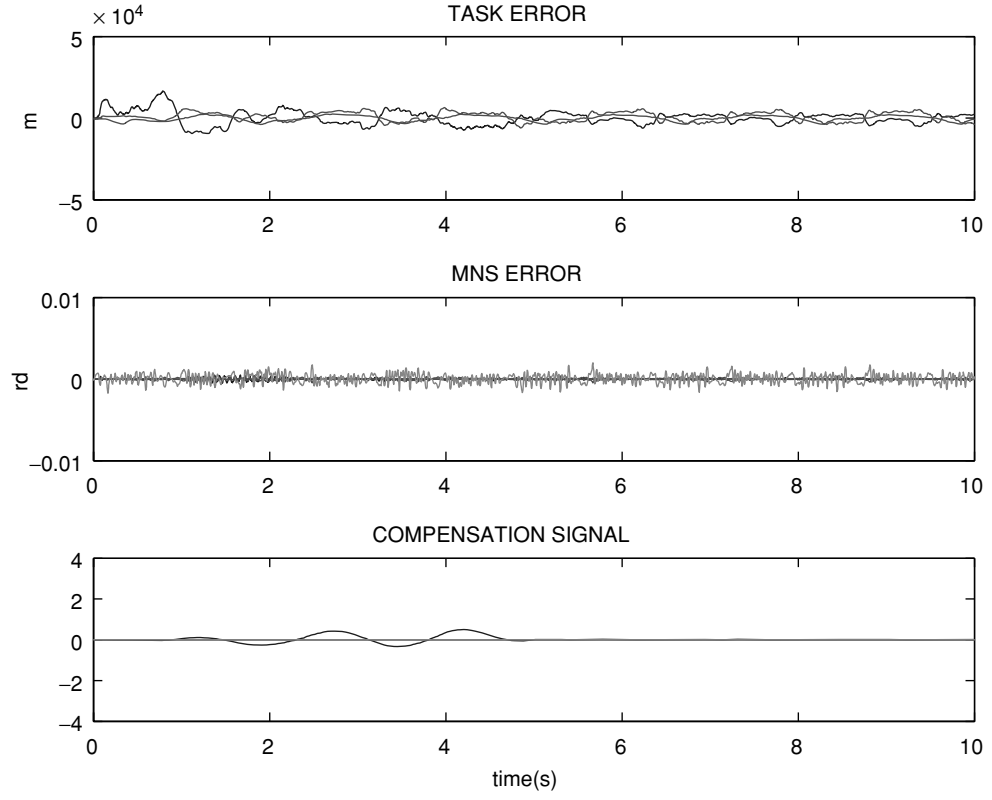


Fig. 7. Experimental results with MNSVC.

The results are presented in Fig. 8 and are almost identical for both algorithms. Although null-space tracking errors are almost zero, the control algorithms almost perfectly track the desired secondary motion of joint 7, while tracking of joint 6 is imperfect. The reason is that the motion of joint 7 is already in the null-space and does not affect the primary task.

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

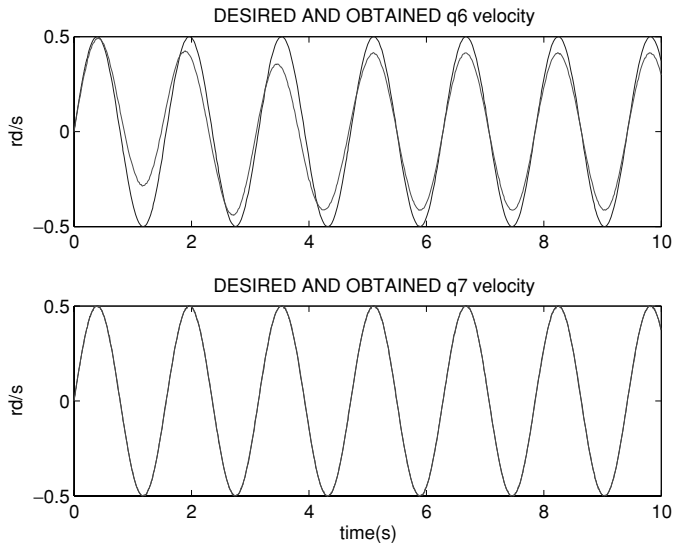


Fig. 8. Experimental results with MNSVC.

On the contrary, the motion of joint 6 also affects the primary task. The mapping to the null-space changes its amplitude. We can also see that the obtained motion is configuration-dependent. This simple example shows that perfect tracking of the secondary task is not very important, as the mapping into the null-space changes the desired secondary motion. More significant is, therefore, the closed loop stability of the algorithm. Null-space control strategies without null-space velocity feedback exhibit instabilities if generalized-inverse-based mapping of the Jacobian to the null-space is used instead of minimal null-space transformation. We repeated the same experiment with the MNSVC control also, by using the Golub–Reunsch algorithm for the SVD calculation. The results are presented in Fig. 9. We obtained similar results as with Givens-rotations-based minimal null-space calculation. However, due to the discontinuous changes of matrix $\mathbf{V}$, we can notice an increase of the null-space tracking errors. In our previous work,[18] we had claimed that discontinuous $\mathbf{V}$ can produce instability, since the differentiation of discontinuous $\mathbf{V}$ results in an unbounded signal. In practice, numerical differentiation gives bounded signal, and causes only performance degradation of the null-space tracking. However, the performance degradation increases at higher joint velocities $\dot{\boldsymbol{q}}$. In our experiment, joint velocities were rather low; therefore, larger errors can be expected at higher joint velocities. Another advantage of the Givens-rotation-based minimal null-space calculation is that it is numerically less demanding than is the Golub–Reunsch algorithm.

8. Conclusion

This paper considers the stability of the control algorithms for redundant robots using minimal null-space force. It was shown that the control algorithms which use the Golub–Reunsch based SVD, causes the performance degradation of the null-space control scheme. We proposed a solution based on SVD calculation using Givens rotations. The proposed control scheme was tested on the simulation of the 10-DOF mobile manipulator system. The primary task was the end effector trajectory tracking, while avoiding the obstacles as a secondary subtask. The results show good numerical stability and shorter computational cycle

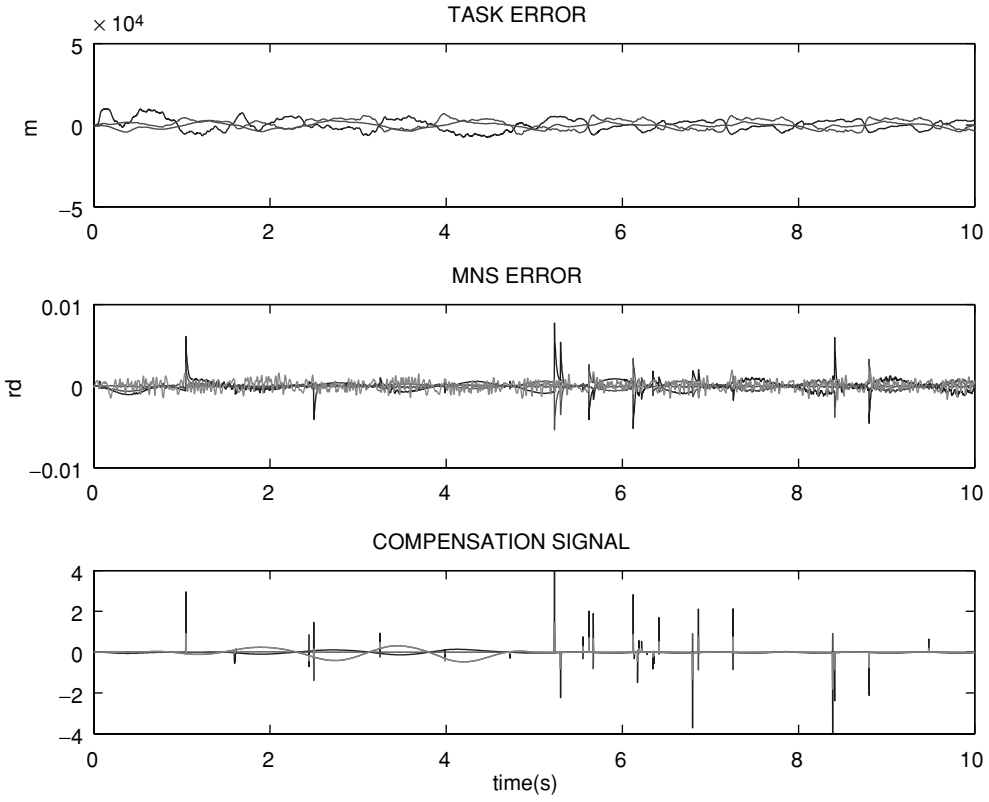


Fig. 9. Experimental results with MNSVC using Golub–Reunsch SVD algorithm.

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press

compared to the SVD based on Golub–Reunsch algorithm. Similar results were obtained with real implementation on 7-DOF robot. However, comparison of the results obtained with NSVC and MNSVC algorithm shows no significant difference. Additionally, the computational burden with MNSVC is higher. In most cases, the secondary tasks are related to obstacle avoidance, singularity avoidance, torque optimization, etc., where good tracking in the null-space is not of primary importance. More important is the closed-loop stability of the overall control scheme. Velocity-based null-space control in both schemes guarantees stability, as long as the null-space feedback gain is sufficiently high. On the contrary, if the torque-based null-space control has to be used, minimal null-space-based control algorithms is the only solution that assures stable operation.

References

1. H. Asada and J.-J. E. Slotine, *Robot Analysis and Control* (Wiley, New York, 1986).
2. P. H. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," *IEEE J. Robot. Autom.* **RA-3**, 393–403 (1987).
3. P. Hsu, J. Hauser and S. Sastry, "Dynamics control of redundant manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, Philadelphia, PA (1988) pp. 183–187.
4. R. Featherstone and O. Khatib, "Load independance of the dynamically consistent inverse of the Jacobian matrix," *Int. J. Robot. Res.* **16**(2), 168–170 (1997).
5. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**, 90–98 (1986).
6. O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Trans. Robot. Autom.* **3**(1), 43–53 (1987).
7. O. Khatib, "The argumented object and reduced effective inertia in robot systems," *Proceedings of the IEEE Conference on Robotics and Automation*, Atlanta, GA (1988) pp. 2140–2147.
8. C. W. Kennedy and J. P. Desai, "Model-based controller for the Mitsubishi PA-10 robot arm: Application to robot-assisted surgery," *Proceedings of the IEEE Conference on Robotics and Automation*, New Orleans, LA (2004).
9. J. Y. S. Luh, "Convetional controller design for industrial robots—A tutorial," *IEEE Trans. Syst., Man, Cybern.* **SMC-13**(3), 298–316 (1983).
10. J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Autom. Control* **AC-25**(3), 486–474 (1980).
11. A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and application to robotics," *Int. J. Robot. Res.* **8**(6), 63–79 (1989).
12. A. A. Maciejewski, "Kinetic limitation on the use of redundancy in robotic manipulators," *IEEE Trans. Robot. Autom.* **7**(2), 205–210 (1991).
13. J.-C. Nash, "A one-sided tranformation method for the singular value decomposition and algepraic eigenproblem," *Comput. J.* **18**(1), 74–76 (1974).
14. C. Natale, B. Siciliano and L. Vilani, "Spatial impedance control of redundant manipulators," *Proceedings of the IEEE Internatinal Symposium on Robotics and Automation (ICRA'99)*, Detroit, MI (1999).
15. B. Nemec, "Force control of redundant robots," **In**: *Preprits of the 5th IFAC Symposium on Robot Control (SYROCO'97)* (M. Guglielmi, ed.), Nantes, France (1997) pp. 215–219.
16. B. Nemec and L. Zlajpah, "Null velocity control with dinamically consistent pseudo-inverse," *Robotica* **18**, 513–518 (2000).
17. B. Nemec and L. Zlajpah, "Experiments with force control of redundant robots in unstructured environment using minimal null-space formulation," *J. Adv. Comput. Intell.* **5**(5), 263–268 (2001).
18. B. Nemec, L. Zlajpah and D. Omrcen, "Stability of null-space control algorithms," *Proceedings of the 12th RAAD Workshop*, Cassino, Italy (2003).
19. D. N. Nenchev, "Redundancy resolution through local optimization: A review," *J. Robot. Syst.* **6**(6), 769–798 (1989).
20. Y. Oh, W. K. Chung, Y. Youm and I. Suh, "Experiments on extended impedance control of redundant manipulator," *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems*, Victoria, Canada (1998) pp. 1320–1325.
21. D. Omrcen, L. Zlajpah and B. Nemec, "Autonomous motion of a mobile manipular using a combined torque and velocity control," *Robotica* **22**, 623–632 (2004).
22. K. O'Neil and Y. C. Chen, "Instability of pseudoinverse acceleration control of redundant mechanisms," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA (2000) pp. 2575–2582.
23. K. O'Neil, "Divergence of linear acceleration-based redundancy resolution schemes," *IEEE Trans. Robot. Autom.* **18**(4), 625–631 (2002).
24. J. Park, W. Chung and Y. Youm, "Weighted decomposition of kinematics and dynamics of kinematically redundant manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, Minneapolis, MN (1996) pp. 480–486.
25. J. Park, W. Chung and Y. Youm, "Characterization of instability of dynamic control for kinematically redundant manipulators," *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, DC (2002) pp. 2400–2405.
26. D. Simon, K. Kapellos and B. Espiau, "Control laws, task and procedures with ORCCAD: Application to the control of an underwater arm," *Proceedings of the International Advanced Robotics Programme Workshop on Underwater Robotics*, Toulon, France (1996) pp. 1027–1176.
27. T. Yoshikawa, *Foundations of Robotics: Analysis and Control* (MIT Press, Cambridge, MA, 1990).

https://doi.org/10.1017/S0263574707003402 Published online by Cambridge University Press