Robotica (2001) volume 19, pp. 649–662. Printed in the United Kingdom © 2001 Cambridge University Press

Analysis and control of redundant manipulator dynamics based on an extended operational space

Ki Cheol Park,* Pyung-Hun Chang† and Sukhan Lee‡

(Received in Final Form: April 23, 2001)

SUMMARY

In this paper a new concept, named the Extended Operational Space (EXOS), has been proposed for the effective analysis and the real-time control of the robot manipulators with kinematic redundancy. The EXOS consists of the operational space (OS) and the optimal null space (NS): the operational space is used to describe manipulator end-effector motion; whereas the optimal null space, described by the minimum number of *NS vectors*, is used to express the self motion.

Based upon the EXOS formulation, the kinematics, statics, and dynamics of redundant manipulators have been analyzed, and control laws based on the dynamics have been proposed. The inclusion of only the minimum number of NS vectors has changed the resulting dynamic equations into a very compact form, yet comprehensive enough to describe: not only the dynamic behavior or the end effector, but also that of the self motion; and at the same time the interaction of these two motions. The comprehensiveness is highlighted by the demonstration of the dynamic couplings between OS dynamics and NS dynamics, which are quite elusive in other approaches.

Using the proposed dynamic controls, one can optimize a performance measure while tracking a desired end-effector trajectory with a better computational efficiency than the conventional methods. The effectiveness of the proposed method has been demonstrated by simulations and experiments.

KEYWORDS: Redundant manipulator; Operational space; Null space; Kinematics; Statics; Dynamics; Dynamic control.

I. INTRODUCTION

For advanced robot applications that require precise control of the end effector, the dynamic behavior of the end effector is one of the most important considerations. For instance, the end effector motion in one direction can substantially affect the motions in its orthogonal directions. This dynamic interaction (or coupling) could become more serious in tasks where the end effector needs to move along stiff surfaces while exerting normal forces on them. To adequately describe, analyze, and control the behavior of the end effector calls for a dynamic model with respect to the end effector. To this end, the operational space (OS) formulation[1] provides a comprehensive framework that enables a better understanding and control of the end effector motion: the respective motion in each direction in the OS, as well as its interaction with other motions.

For kinematically redundant manipulators, the tasks often involve not only the end effector motion, but also the achievement of an additional performance using the extra degrees of freedom, through the so-called self-motion (or null-motion). Then, as a logical extension to the OS case, the issue of interest would be: Are there any *dynamic interactions between the end-effector motion and the self-motion*, just as there exist interactions (or couplings) among the end effector motions only? If that is indeed the case, then follow the issues: Why and how do these occur; and how can we control them? In fact, we have observed that, in most redundancy resolution schemes except a few using the inertia weighted pseudoinverse,[1] there exist dynamic interactions between the two motions; yet few research works have attempted to analyze and control it except those cited in references [2–4].* Not only the interactions of the two motions are interesting, but so are the respective dynamics of each motion as well.

The description, analysis, and control of redundant manipulators with respect to both the end effector dynamics and self-motion dynamics obviously require an even more comprehensive framework. *To propose such a framework* is one of the most important motivations for the research described in this paper. Although the OS approach[1] also addresses the dynamics of redundant manipulators, a *more descriptive framework* appears still necessary. To this end, we will formulate an extension to the operational space† (EXOS) at the *velocity level*,[2] which consists of both the OS for end effector motion and Null Space (NS) for self motion. Then, on the basis of the EXOS, the kinematics, statics, dynamics, and control of the redundant manipulator will be described and analyzed in a consistent manner.

As another comprehensive framework for redundant manipulators, the *configuration control scheme*[6] is noteworthy. In this scheme, the kinematic functions that reflect the additional performances are combined with the end effector coordinates to form a set of *configuration variables*

* FA Research Inst., Production Eng. Center, Samsung Electronics Co., Suwon (South Korea), E-mail: robopark@samsung.co.kr
† Dept. of Mechanical Eng., KAIST, Taejon (Korea), E-mail: phchang@kaist.ac.kr
‡ Samsung Advanced Institute of Technology, Kyungki-do (Korea), E-mail: lshee@weed.sait.samsung.co.kr

* This paper and [4] are based on almost verbatim translations of references [2], [3]. The difference, however, is that: in [4] we added experimental results; in this paper, both experimental results and a substantial enhancement in analysis – thus the most complete version among the three.
† Similar term was also used in reference [5] to express the inclusion of the closed-chain mechanisms.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

at the position level. The role of the kinematic function is conceptually simple, yet its utility is quite flexible and effective. For instance, the function can represent some intuitive and tangible performances like the *shape* of the manipulator configuration, in addition to conventional objective functions such as the manipulability measure. Furthermore, the scheme ensures cyclic motions provided that the singularity boundaries are not crossed. In this approach, however, the NS dynamics does not appear to be of primary concern.

The paper is structured as follows: In Section II, EXOS will be defined and both EXOS Jacobian and its property will be considered. Section III will present a new kinematic control algorithm based upon the EXOS and its relationship with the conventional algorithms. Section IV deals with the statics and dynamics based on EXOS and reviews other methods in light of the EXOS dynamics. In Section V, new dynamic control laws will be proposed and will be compared with the conventional methods. In addition, simulation and experimental results will be presented to demonstrate the effectiveness of the new control law. Finally, the conclusion will be drawn in Section VI.

II. AN EXTENDED OPERATIONAL SPACE

The kinematic equation for a redundant manipulator is given as the following kinematic equation:

$$\boldsymbol{x}=\mathbf{f}(\boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{x}$ denotes an m-dimensional vector representing the location of end effector w.r.t the base coordinate system, $\boldsymbol{\theta}$ an n-dimensional vector representing joint variables, and $\mathbf{f}$ a vector consisting of m scalar functions, with $m < n$. For dimensional consistency, it is assumed without loss of generality that all joints are revolute.*

II.A. EXOS concept

Given an end effector location, the redundant manipulator has infinite joint configurations, the set of which is described as follows:

$$\Psi(\boldsymbol{x})=\{\boldsymbol{\theta}\in\Re^n : \mathbf{f}(\boldsymbol{\theta})=\boldsymbol{x}\}, \tag{2}$$

where Ψ denotes self-motion manifold,[8] which has the dimension of r, with r being the *degree of redundancy* (DOR), namely $r=n-m>0$.

From the kinematic equation, the Jacobian equation of the manipulator is determined as

$$\dot{\boldsymbol{x}}=\boldsymbol{J}\dot{\boldsymbol{\theta}}, \tag{3}$$

where $\dot{(\cdot)}$ denotes the time derivative, and $\boldsymbol{J}=\dfrac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}\in\Re^{m\times n}$ denotes a Jacobian matrix. It is well known that this Jacobian equation characterizes *the mapping relationship from n-dimensional joint space to m-dimensional operational space at velocity level*.

* One can easily show that the formulation in this section is made applicable to prismatic joints by introducing appropriate matrices. Interested readers will find a more detailed exposition in reference [7].

Now define $\boldsymbol{Z}\in\Re^{r\times n}$ as a matrix consisting of the basis vectors spanning the null space (NS) (or the tangent space of the self-motion manifold) – $\boldsymbol{Z}$ can be obtained by the method described in Section II.B. Then $\boldsymbol{Z}$ satisfies the following relationship:

$$\boldsymbol{Z}\boldsymbol{J}^T=0. \tag{4}$$

In addition, define $\dot{\boldsymbol{x}}_{\mathrm{N}}$ as self-motion velocity or NS velocity. Then we have a *complementary mapping relationship at velocity level* between the joint space and the NS, which is

$$\dot{\boldsymbol{x}}_{\mathrm{N}}\equiv\boldsymbol{Z}\dot{\boldsymbol{\theta}}. \tag{5}$$

By using the two relationships, let us define the *extension to the operational space* (EXOS) as a space that consists of m-dimensional Operational Space (OS) for end effector motion and r dimensional NS for self motion. Then, EXOS Jacobian can be defined as

$$\boldsymbol{J}_{\mathrm{E}}\equiv\begin{bmatrix}\boldsymbol{J}\\ \boldsymbol{Z}\end{bmatrix} \tag{6}$$

and EXOS velocity as

$$\dot{\boldsymbol{x}}_{\mathrm{E}}\equiv\begin{Bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{x}}_{\mathrm{N}}\end{Bmatrix}. \tag{7}$$

Now, the EXOS Jacobian equation is determined as

$$\dot{\boldsymbol{x}}=\boldsymbol{J}_{\mathrm{E}}\dot{\boldsymbol{\theta}}. \tag{8}$$

Note that the EXOS Jacobian equation enables to treat the differential kinematics of redundant manipulators as if they were non-redundant manipulators.

The relationship (8) can be described more clearly by the visualization for a 3 DOF planar revolute manipulator. In Fig. 1, the direction of $\boldsymbol{Z}$ being perpendicular to the plane spanned by the two row vectors of $\boldsymbol{J}$, $\dot{\boldsymbol{x}}_{\mathrm{N}}$ corresponds to the projection of $\dot{\boldsymbol{\theta}}$ to that direction, the component responsible for self-motion. The remaining part of $\dot{\boldsymbol{\theta}}$ corresponds to the component contributing to the motion in the OS.

When $\boldsymbol{J}_{\mathrm{E}}$ maps $\dot{\boldsymbol{\theta}}$, $\boldsymbol{J}$ in $\boldsymbol{J}_{\mathrm{E}}$ extracts from $\dot{\boldsymbol{\theta}}$ the component, $\dot{\boldsymbol{\theta}}_{\mathrm{R}}$, and maps into the end effector velocity, $\dot{\boldsymbol{x}}$; whereas $\boldsymbol{Z}$

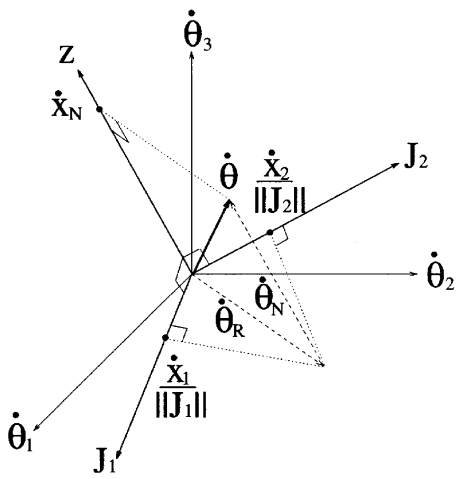


Fig. 1. Visualization of EXOS kinematic relationship.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

extracts $\dot{\boldsymbol{\theta}}_N$ and maps into the NS velocity vector $\dot{\boldsymbol{x}}_N$. This interpretation of (8), and $\boldsymbol{J}_E$ in particular, can be illustrated rather clearly with Fig. 3.7 of reference [9]. Incidentally, observing the role of $\boldsymbol{J}$ and $\boldsymbol{Z}$, one can find it very similar to that of the selection matrices, $\boldsymbol{S}$ and $\boldsymbol{I}-\boldsymbol{S}$, in the sense that they also extract and map.[10]

II.B. Selection of Null Space matrix, $\boldsymbol{Z}$

As illustrated in Fig. 1, given $\boldsymbol{J}$ at time t, there always exists the corresponding NS matrix, $\boldsymbol{Z}$. There are a few methods available including reference [11] to yield the NS matrix, $\boldsymbol{Z}$.

In reference [11], $\boldsymbol{Z}$ is obtained as

$$\boldsymbol{Z}=[\boldsymbol{J}_{n-m}Adj(\boldsymbol{J}_m) \ \vdots \ -\det(\boldsymbol{J}_m)\boldsymbol{I}_{n-m}], \tag{9}$$

where $Adj(\cdot)$ denotes an adjoint matrix. Here $\boldsymbol{J}_m$ consists of m consists of m linearly independent column vectors of $\boldsymbol{J}$ and $\boldsymbol{J}_{n-m}$ its remaining column vectors. This method, we have found, provides an efficient way of obtain $\boldsymbol{Z}$ and a convenient means to generate a symbolic form of $\boldsymbol{Z}$. Yet, it accompanies a discontinuous $\dot{\boldsymbol{Z}}$ when when $\boldsymbol{J}_m$ has to be updated with a different set of column vectors between consecutive times, thus causing discontinuous input for the dynamic control.

In comparison, the Singular Value Decomposition (SVD) method[2] yields a set of the orthogonal base vectors of $\boldsymbol{Z}$ without causing the discontinuity problem due to the switching of $\boldsymbol{J}_m$. Nevertheless, it is worth reporting that a commercially available SVD method (in MATLAB) occasionally yields base vectors with opposite directions between two consecutive times, involving a different class of discontinuity problem. Although it appears premature to provide the reason for this problem, those who need to compute $\boldsymbol{Z}$ by using the SVD should be well aware of, and cautious about, this problem. Furthermore, relatively large computational amount makes the SVD inappropriate for real-time control of redundant manipulators.

In response, we are going to propose a new method that is about 5 to 10 times more efficient than the SVD method and at the same time free from the discontinuity problem. Like the SVD, the new method is to generate a set of orthonormal base vectors satisfying the following constraints:

$$\begin{aligned} \boldsymbol{J}\boldsymbol{Z}^T &= \boldsymbol{0} \\ \boldsymbol{Z}\boldsymbol{Z}^T &= \boldsymbol{I}. \end{aligned} \tag{10}$$

Differentiating them leads to

$$\begin{aligned} \boldsymbol{J}\dot{\boldsymbol{Z}}^T &= -\dot{\boldsymbol{J}}\boldsymbol{Z}^T \\ \boldsymbol{Z}\dot{\boldsymbol{Z}}^T &= \boldsymbol{0}. \end{aligned} \tag{11}$$

From (11), the following equation results:

$$\dot{\boldsymbol{Z}}^T=\begin{bmatrix}\boldsymbol{J}\\ \boldsymbol{Z}\end{bmatrix}^{-1}\begin{bmatrix}-\dot{\boldsymbol{J}}\boldsymbol{Z}^T\\ \boldsymbol{0}\end{bmatrix}. \tag{12}$$

By using (12) as an updating rule, we can obtain $\boldsymbol{Z}$ as follows:

$$\boldsymbol{Z}(t)=\boldsymbol{Z}(t-\Delta t)+\dot{\boldsymbol{Z}}(t)\Delta t, \tag{13}$$

where the initial $\boldsymbol{Z}(0)$ is computed off-line by the SVD method. It is noteworthy that Δt, being the control sampling time has already been selected to be *sufficiently small* so that it can faithfully represent robot motion (including null-motion) up to its maximum bandwidth.

In addition, $\dot{\boldsymbol{Z}}(t)^T$ is determined from (12) as follows:

$$\dot{\boldsymbol{Z}}(t)^T=\begin{bmatrix}\boldsymbol{J}(t-\Delta t)\\ \boldsymbol{Z}(t-\Delta t)\end{bmatrix}^{-1}\begin{bmatrix}-\dot{\boldsymbol{J}}(t)\boldsymbol{Z}(t-\Delta t)^T\\ \boldsymbol{0}\end{bmatrix}, \tag{14}$$

where $\dot{\boldsymbol{J}}(t)$ is obtained by the backward difference, i.e.,

$$\dot{\boldsymbol{J}}(t)=\frac{\boldsymbol{J}_{(t)}-\boldsymbol{J}_{(t-\Delta t)}}{\Delta t}.$$

Note that the inverse of the EXOS Jacobian $\boldsymbol{J}_E^{-1}(t-\Delta t)$ in (14) does not need to be computed here because it is already available through either the EXOS kinematic control (25) or dynamic control (52). Note also that $\dot{\boldsymbol{J}}(t)$ and $\dot{\boldsymbol{Z}}(t)$ can be reused for dynamic controls in (52), (53) and (54), thereby saving additional computations for dynamic control.

It is instructive to observe that (12) is also the solution to minimize $\| \dot{\boldsymbol{Z}} \|$ under the constraint, $\boldsymbol{J}\boldsymbol{Z}^T=\boldsymbol{0}$. More specifically, one can easily prove that (12) is the solution for the following Quadratic Programming (QP) problem:

$$\text{Minimize} \qquad \frac{1}{2}\dot{\boldsymbol{Z}}\dot{\boldsymbol{Z}}^T=\frac{1}{2}\| \dot{\boldsymbol{Z}} \|^2 \tag{15}$$

$$\text{Subject to} \qquad \boldsymbol{J}\dot{\boldsymbol{Z}}^T=-\dot{\boldsymbol{J}}\boldsymbol{Z}^T. \tag{16}$$

That (12) is the solution minimizing $\| \dot{\boldsymbol{Z}} \|$ means that $\boldsymbol{Z}(t)$ is updated in such a way that its continuity is tightly preserved. As a result, we have not encountered any discontinuity problem in the new method.

The integration error due to (12) and (13) can be significantly reduced, if we make the following modification:*

$$\dot{\boldsymbol{Z}}^T=\begin{bmatrix}\boldsymbol{J}\\ \boldsymbol{Z}\end{bmatrix}^{-1}\begin{bmatrix}-\dot{\boldsymbol{J}}\boldsymbol{Z}^T-k_z\boldsymbol{J}\boldsymbol{Z}^T\\ \frac{1}{2}k_z(\boldsymbol{I}-\boldsymbol{Z}\boldsymbol{Z}^T)\end{bmatrix}, \tag{17}$$

where k_z denotes the scalar gain for correcting the errors of the constraints (10).

In (17), which we used for subsequent simulations and experiments, since k_z plays an important role for the accuracy of $\boldsymbol{Z}$, its selection is discussed in the appendix.

II.C. Important properties of EXOS Jacobian, $\boldsymbol{J}_E$

By using the definition of EXOS Jacobian, $\boldsymbol{J}_E$, in (8) and $\boldsymbol{Z}$ in (10), we can derive the determinant of $\boldsymbol{J}_E$ as follows. Since

* If k_z is selected according to the method described in the appendix, we could reduce the errors to levels that are meaninglessly small for practical purposes.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

$$\boldsymbol{J}_{\mathrm{E}}\boldsymbol{J}_{\mathrm{E}}^{T}=\begin{bmatrix}\boldsymbol{J}\boldsymbol{J}^{T} & \boldsymbol{J}\boldsymbol{Z}^{T}\\ \boldsymbol{Z}\boldsymbol{J}^{T} & \boldsymbol{Z}\boldsymbol{Z}^{T}\end{bmatrix}=\begin{bmatrix}\boldsymbol{J}\boldsymbol{J}^{T} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Z}\boldsymbol{Z}^{T}\end{bmatrix}, \tag{18}$$

it immediately follows that

$$\det(\boldsymbol{J}_{\mathrm{E}}\boldsymbol{J}_{\mathrm{E}}^{T})=\det(\boldsymbol{J}\boldsymbol{J}^{T})\det(\boldsymbol{Z}\boldsymbol{Z}^{T}) \tag{19}$$

From (19) and (10), follows the determinant of EXOS Jacobian:

$$\det(\boldsymbol{J}_{\mathrm{E}}\boldsymbol{J}_{\mathrm{E}}^{T})=\sqrt{\det(\boldsymbol{J}\boldsymbol{J}^{T})}. \tag{20}$$

Equation (20) shows that the EXOS Jacobian has the following properties:

- Firstly, EXOS Jacobian in singular only when $\det(\boldsymbol{J}\boldsymbol{J}^{T})=0$, which occurs if and only if the manipulator is *kinematically singular. The EXOS Jacobian behaves at singularity in the same way as a Jacobian matrix does for non-redundant manipulators*.
- Secondly, it is noteworthy that the determinant is the same as what is called the *manipulability measure.*[13]
- Thirdly, at any configuration without kinematic singular point, there is no other singularity. Therefore, *no algorithmic singularity exists in EXOS Jacobian*.

III. EXOS KINEMATICS

Given the desired end effector velocity and NS velocity, the *forward kinematics* is immediately available from the defining relationships for EXOS as

$$\begin{Bmatrix}\dot{\boldsymbol{x}}_{d}\\ \dot{\boldsymbol{x}}_{\mathrm{N}d}\end{Bmatrix}=\boldsymbol{J}_{\mathrm{E}}\dot{\boldsymbol{\theta}}_{d}, \tag{21}$$

where $\dot{\boldsymbol{x}}_{d}$, $\dot{\boldsymbol{x}}_{\mathrm{N}d}$, and $\dot{\boldsymbol{\theta}}_{d}$ are the desired end effector velocity, the NS velocity, and the corresponding joint velocity, respectively. The *inverse kinematics*, then, is expressed as

$$\dot{\boldsymbol{\theta}}_{d}=\boldsymbol{J}_{\mathrm{E}}^{-1}\begin{Bmatrix}\dot{\boldsymbol{x}}_{d}\\ \dot{\boldsymbol{x}}_{\mathrm{N}d}\end{Bmatrix}. \tag{22}$$

If a secondary performance is required and specified with a measure of H, then the desired NS velocity may be defined as

$$\dot{\boldsymbol{x}}_{\mathrm{N}d}=\boldsymbol{Z}\boldsymbol{h}, \tag{23}$$

where

$$\boldsymbol{h}=k_{h}\nabla H \tag{24}$$

with k_h being a constant. Combining this equation with equation (22) leads to

$$\dot{\boldsymbol{\theta}}_{d}=\boldsymbol{J}_{\mathrm{E}}^{-1}\begin{Bmatrix}\dot{\boldsymbol{x}}_{d}\\ \boldsymbol{Z}\boldsymbol{h}\end{Bmatrix}. \tag{25}$$

III.A. Relationship with Resolved Motion Method

The Resolved Motion Method[14] is given as

$$\dot{\boldsymbol{\theta}}_{d}=\boldsymbol{J}^{+}\dot{\boldsymbol{x}}_{d}+(\boldsymbol{I}_{n}-\boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{h}, \tag{26}$$

where $\boldsymbol{J}^{+}=\boldsymbol{J}^{T}(\boldsymbol{J}\boldsymbol{J}^{T})^{-1}$ is the Moore-Penrose pseudo-inverse matrix of $\boldsymbol{J}$. RMM is the most general formulation and is widely used for the kinematic control of the redundant manipulator.

In reference [15], the following have been derived:

$$\boldsymbol{J}^{+}=\boldsymbol{J}_{\mathrm{E}}^{-1}\begin{bmatrix}\boldsymbol{I}_{m}\\ \boldsymbol{0}\end{bmatrix},\quad \boldsymbol{I}_{n}-\boldsymbol{J}^{+}\boldsymbol{J}=\boldsymbol{J}_{\mathrm{E}}^{-1}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{Z}\end{bmatrix} \tag{27}$$

Combining the above two equations with equation (26) leads to equation (25).[15]

Hence, equation (26) and equation (25) are essentially the same, sharing the *same inverse kinematic solutions*. In other words, the difference between the two does not lie in *essence* but in its *form*. Yet, in addition to having more compact form using (25) may have some significant advantages stemming from its very expression as follows:

- In RMM, since the NS projection matrix, the rank of which is $n-m$, is an $n\times n$ matrix, there are m overlapping equations. In comparison, since the NS basis matrix in EXOS has exactly $n-m$ vectors, the method tends to expose the contribution of each vector rather transparently, allowing a better understanding and use of the NS. This advantage will be clearly demonstrated in Section IV, where EXOS formulation yields the dynamics equations in a remarkably succinct form, which would have been very difficult to obtain with RMM owing to its complexity.
- With EXOS method, we can treat kinematics, statics, dynamics, and control in a consistent manner as if the manipulator were non-redundant, thereby sharing insights already established, such as the duality of kinematics and statics.

III.B. Comparison with Extended Jacobian Method

As is well known, the Extended Jacobian Method (EJM)[16] is given as follows:

$$\boldsymbol{\theta}_{d}=\begin{bmatrix}\boldsymbol{J}\\ \dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\boldsymbol{\theta}}\end{bmatrix}^{-}\begin{Bmatrix}\dot{\boldsymbol{x}}_{d}\\ \boldsymbol{0}\end{Bmatrix}. \tag{28}$$

Comparing with the EXOS method in (25), one can easily observe that the two methods have similarity in their forms, yet difference in their augmented terms. In fact, $\dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\boldsymbol{\theta}}$ does not represent the NS of $\boldsymbol{J}$ at all. As the result of this difference, the EJM has the following drawbacks: the complexity and computational inefficiency due to the matrix $\dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\boldsymbol{\theta}}$; and the difficulty in recovering from the algorithmic singularity. Nevertheless, EJM shows the better performance in terms of satisfying the optimality condition ($\boldsymbol{Z}\nabla H=\boldsymbol{0}$) and achieving repeatability, the ability to yield the same joint values for a given end effector location.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

Incidentally, it is noteworthy that the RM type methods, including EXOS, also achieve this repeatability under the following conditions:[17]

- When the joint configuration belongs to and stays in a group on the Measure Constraint Locus (MCL)[18] that meets the necessary and sufficient condition for optimality.
- If a constant k_h in equation (24) has an appropriate value.

Although the repeatability achieved is in an approximate sense, it is good enough for most practical purposes, and should not be confused with the case when joint variables evolve into unpredictable states.

IV. EXOS DYNAMICS

In this section, we will derive the statics and dynamics by using the framework of EXOS. The result of EXOS statics is immediately used for the dynamics.

IV.A. EXOS statics

To derive the relationships between the statics in joint space and those in both the OS and the NS we adopted the principle of virtual work.

First of all, let $\mathcal{F}_{\mathrm{N}}$ denote the force responsible for the NS displacement $\delta \boldsymbol{x}_{\mathrm{N}}$, and term this force *null force*. In order for a redundant manipulator to stay in a perfect equilibrium state, the virtual displacement needs to exist not only in the OS but also in the NS. Virtual work for the redundant manipulator is expressed as

$$\delta W = \tau \cdot \delta \boldsymbol{\theta} - \mathcal{F} \cdot \delta \boldsymbol{x} - \mathcal{F}_{\mathrm{N}} \cdot \delta \boldsymbol{x}_{\mathrm{N}} = 0, \tag{29}$$

where $(\cdot)$ denotes the inner product of vector. Alternatively,

$$\tau^T \delta \boldsymbol{\theta} = \mathcal{F}^T \delta \boldsymbol{x} + \mathcal{F}_{\mathrm{N}}^T \delta \boldsymbol{x}_{\mathrm{N}}. \tag{30}$$

Since $\delta \boldsymbol{x}_{\mathrm{E}} = \boldsymbol{J}_{\mathrm{E}} \delta \boldsymbol{\theta}$ from (8), combining this equation with (30) and then transposing, we have

$$\begin{aligned} \tau &= \boldsymbol{J}^T \mathcal{F} + \boldsymbol{Z}^T \mathcal{F}_{\mathrm{N}} \\ &= \boldsymbol{J}_{\mathrm{E}}^T \mathcal{F}_{\mathrm{E}}, \end{aligned} \tag{31}$$

where

$$\mathcal{F}_{\mathrm{E}} = \begin{Bmatrix} \mathcal{F} \\ \mathcal{F}_{\mathrm{N}} \end{Bmatrix}. \tag{32}$$

It is noteworthy that equations (8) and (31) show the duality between the kinematics and statics in exactly the same manner as was found in the non-redundant case.

IV.B. EXOS dynamics

To provide the context for the EXOS dynamics, let us first consider the joint space dynamics and the OS dynamics[1] as follows:

$$\text{Joint Dynamics} \ : \ \tau = M_\theta(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + N_\theta(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}). \tag{33}$$

$$\text{OS Dynamics} \ : \ \mathcal{F} = M_x \ddot{\boldsymbol{x}} + N_x. \tag{34}$$

To derive the EXOS dynamics, differentiate the Jacobian equation (8) and solve for joint acceleration as follows:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}_{\mathrm{E}}^{-1} (\ddot{\boldsymbol{x}}_{\mathrm{E}} - \dot{\boldsymbol{J}}_{\mathrm{E}} \dot{\boldsymbol{\theta}}). \tag{35}$$

Then from the statics in (31) and joint dynamics in (33), we have

$$\mathcal{F}_{\mathrm{E}} = \boldsymbol{J}_{\mathrm{E}}^{-T} (M_\theta \ddot{\boldsymbol{\theta}} + N_\theta). \tag{36}$$

Substituting (35) into (36) leads to

$$\mathcal{F}_{\mathrm{E}} = M_{x_{\mathrm{E}}} \ddot{\boldsymbol{x}}_{\mathrm{E}} + N_{x_{\mathrm{E}}}, \tag{37}$$

where $\boldsymbol{M}_{x_{\mathrm{E}}} = \boldsymbol{J}_{\mathrm{E}}^{-T} \boldsymbol{M}_\theta \boldsymbol{J}_{\mathrm{E}}^{-1} (\in \Re^{n \times n})$, $\boldsymbol{N}_{x_{\mathrm{E}}} = \boldsymbol{J}_{\mathrm{E}}^{-T} (\boldsymbol{N}_\theta - \boldsymbol{M}_\theta \boldsymbol{J}_{\mathrm{E}}^{-1} \dot{\boldsymbol{J}}_{\mathrm{E}} \dot{\boldsymbol{\theta}})$ $(\in \Re^n)$.

Comparing (34) with (37), term by term, one can easily find out the *similarity* between OS dynamics and EXOS dynamics: replacing $\boldsymbol{J}$ with $\boldsymbol{J}_{\mathrm{E}}$, the OS dynamics immediately results in EXOS dynamics. The *difference*, however, is also quite obvious and significant, which is made more clear, if the EXOS dynamics is expressed as the following:

$$\ddot{\boldsymbol{x}}_{\mathrm{E}} = \boldsymbol{M}_{x_{\mathrm{E}}}^{-1} (\mathcal{F}_{\mathrm{E}} - \boldsymbol{N}_{x_{\mathrm{E}}}), \tag{38}$$

where the inverse of EXOS inertia matrix is determined as

$$\boldsymbol{M}_{x_{\mathrm{E}}}^{-1} = \begin{bmatrix} \boldsymbol{J} \boldsymbol{M}_\theta^{-1} \boldsymbol{J}^T & \boldsymbol{J} \boldsymbol{M}_\theta^{-1} \boldsymbol{Z}^T \\ \boldsymbol{Z} \boldsymbol{M}_\theta^{-1} \boldsymbol{J}^T & \boldsymbol{Z} \boldsymbol{M}_\theta^{-1} \boldsymbol{Z}^T \end{bmatrix}. \tag{39}$$

Evidently this matrix maps a set of OS force and NS force into a corresponding set of OS motion and NS motion in various combinations. In short, the two block matrices on the diagonal represents the OS and NS dynamics, respectively; the two on the off-diagonal the *interactions or couplings* of the two.

For example, let us consider a planar 3 DOF redundant manipulator. Each link of the manipulator has the length of 1 *m*, the mass of 12 *kg*, and the inertia of 1 $kg \cdot m^2$. Figure 2 shows the elements of $\boldsymbol{M}_{x_{\mathrm{E}}}^{-1}$ at each joint configuration, $\alpha = \theta_1 + \theta_2 + \theta_3$, when the end effector stays at $(x, y) = (2, 0)$ with the joint changing its configuration along the self-motion manifold. In this figure, ρ_{ij} denotes the element of i^{th} row and j^{th} column of $\boldsymbol{M}_{x_{\mathrm{E}}}^{-1}$. More specifically, Fig. 2(a) on one hand shows the OS force-acceleration relationships: ρ_{11} denote the relationship in x direction, ρ_{22} than in y direction, and ρ_{12} stands for the dynamic coupling between x-axis and y-axis. Figure 2(b), on the other hand, shows the elements of $\boldsymbol{M}_{x_{\mathrm{E}}}^{-1}$ owing to the self-motion. That is, ρ_{33} stands for the NS force-acceleration relationship, and ρ_{13}, ρ_{23} the dynamic coupling between OS and NS. Note that just as ρ_{12}, the coupling mentioned in Section I and experimentally observed in reference [19], is very difficult to identify in joint space dynamics, so are ρ_{13}, ρ_{23} and ρ_{33} in either joint space dynamics or OS dynamics; only in EXOS dynamics they become transparent.

In addition $\boldsymbol{M}_{x_{\mathrm{E}}}^{-1}$ in (39) clearly explains why OS dynamics is decoupled from NS dynamics when the inertia weighted pseudoinverse is used for redundancy resolution.[1] That is, if $\boldsymbol{Z}\boldsymbol{M}_\theta$ is used instead of $\boldsymbol{Z}$, the inverse of EXOS inertia matrix becomes

$$\boldsymbol{M}_{x_{\mathrm{E}}}^{-1} = \begin{bmatrix} \boldsymbol{J} \boldsymbol{M}_\theta^{-1} \boldsymbol{J}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z} \boldsymbol{M}_\theta \boldsymbol{Z}^T \end{bmatrix}. \tag{40}$$

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

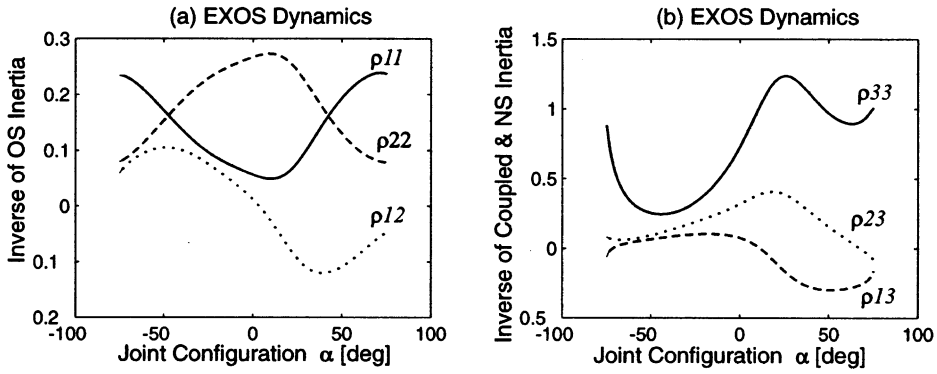


Fig. 2. Elements of inverse of EXOS inertia matrix along the self motion manifold for showing dynamic and coupling effects ($\alpha = \theta_1 + \theta_2 + \theta_3$ with the end-effector fixed at $(x, y) = (2, 0)$.

Before we begin to make use of our findings, let us summarize the relationships in EXOS framework in terms of another diagram, termed Extended Dynamic Premultiplier Diagram (EXDPD), shown in Fig. 3(b). Like DPD, EXDPD succinctly and clearly describes the relationship among the kinematics, statics, and dynamics in the EXOS. Note that EXOS Jacobian being a square matrix, its inverse is obtained as if the manipulator were nonredundant. In Fig. 3(b), $\ddot{\boldsymbol{x}}'_{\mathrm{E}} \equiv \ddot{\boldsymbol{x}}_{\mathrm{E}} - \dot{\boldsymbol{J}}_{\mathrm{E}}\dot{\boldsymbol{\theta}}$, $\tau' \equiv \tau - N_\theta$, and $\mathcal{F}'_{\mathrm{E}} \equiv \mathcal{F}_{\mathrm{E}} - \boldsymbol{J}_{\mathrm{E}}^{-T} N_\theta$.

IV.C. Comparison with other methods

Having proposed the EXOS dynamics and expounded its significance, it is relevant and instructive to compare it with other formulations in terms of the *manner and the economy with which the OS dynamics and NS dynamics are described*. The methods of interest are the dynamics due to the EJM, the OS formulation, and the RMM, respectively. To this end, we have derived each of the dynamics in a form similar to the EXOS and, in particular, the inertia matrix due to each method in the same form as (39) and (40). Although the dynamics due to the EJM has been rendered into a similar form,[6] those for the OS formulation and the RMM, to our knowledge, have not appeared in literature, yet.

IV.C.1. The dynamics due to the Extended Jacobian Method. Define $\boldsymbol{J}_a \equiv \dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial \boldsymbol{\theta}}$ and $\dot{\boldsymbol{x}}_a \equiv \boldsymbol{J}_a\dot{\boldsymbol{\theta}} \in \Re^r$, respectively. In addition, the static equilibrium condition requires that $\tau = \boldsymbol{J}^T\mathcal{F} + \boldsymbol{J}_a^T\mathcal{F}_a$. From these, we have

$$\ddot{\boldsymbol{x}}_e = \begin{bmatrix} \ddot{\boldsymbol{x}} \\ \ddot{\boldsymbol{x}}_a \end{bmatrix} \in \Re^n;\ \mathcal{F}_e = \begin{bmatrix} \mathcal{F} \\ \mathcal{F}_a \end{bmatrix} \in \Re^n;\ \boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{J} \\ \boldsymbol{J}_a \end{bmatrix} \in \Re^{n\times n}. \quad (41)$$

Then, it is easy to show that the dynamics due to EJM can be rendered into a form similar to (38) as follows:

$$\ddot{\boldsymbol{x}}_e = \boldsymbol{M}_{xe}^{-1}(\mathcal{F}_e - \boldsymbol{N}_{xe}), \quad (42)$$

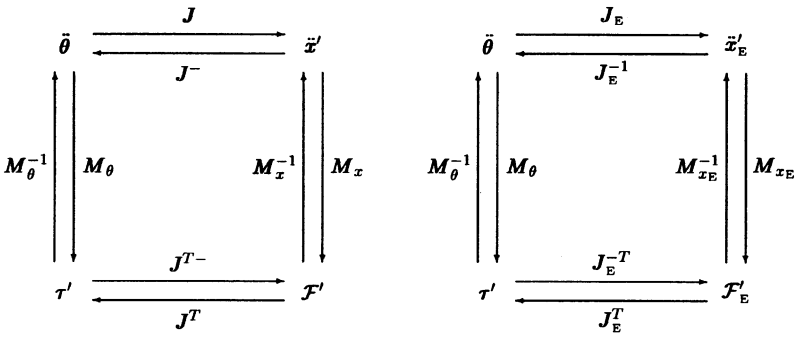
where $\boldsymbol{N}_{xe} = \boldsymbol{J}_e^{-T}(\boldsymbol{N}_\theta - \boldsymbol{M}_\theta \boldsymbol{J}_e^{-1}\dot{\boldsymbol{J}}_e\dot{\boldsymbol{\theta}})$ and the inverse inertia matrix is given as

$$\boldsymbol{M}_{xe}^{-1} = \boldsymbol{J}_e\boldsymbol{M}_\theta^{-1}\boldsymbol{J}_e^T = \begin{bmatrix} \boldsymbol{J}\boldsymbol{M}_\theta^{-1}\boldsymbol{J}^T & \boldsymbol{J}\boldsymbol{M}_\theta^{-1}\boldsymbol{J}_a^T \\ \boldsymbol{J}_a\boldsymbol{M}_\theta^{-1}\boldsymbol{J}^T & \boldsymbol{J}_a\boldsymbol{M}_\theta^{-1}\boldsymbol{J}_a^T \end{bmatrix} \in \Re^{n\times n}. \quad (43)$$

Note that $\boldsymbol{M}_{xe}^{-1}$ and its submatrices in (43) have exactly the same dimensions as their counterparts of the EXOS in (39). As a result, the dynamics is as compact and succinct as the EXOS dynamics.

Nevertheless, neither is $\boldsymbol{J}_a$ a NS matrix of $\boldsymbol{J}$, nor does $\boldsymbol{J}_a\boldsymbol{M}_\theta^{-1}\boldsymbol{J}_a^T$ *represent the NS dynamics at all*, nor do $\boldsymbol{J}\boldsymbol{M}_\theta^{-1}\boldsymbol{J}_a^T$ and $\boldsymbol{J}_a\boldsymbol{M}_\theta^{-1}\boldsymbol{J}^T$ stand for the couplings of OS dynamics and NS dynamics. Recollecting that our objective is to obtain a framework to describe both the two dynamics and their coupling, the dynamics due to the EJM does not serve this objective.

IV.C.2. The dynamics due to the Operational Space Formulation. From the well-known statics in the OS formulation,[1] $\tau = \boldsymbol{J}^T\mathcal{F} + (\boldsymbol{I}_n - \boldsymbol{J}^T\boldsymbol{J}^{T-})\Gamma$, it is clear that



(a) DPD (b) EXDPD

Fig. 3. Premultiplier diagram for kinematics, statics, and dynamics.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

$(\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J}) \in \Re^{n\times n}$ is used as its NS matrix, where $\boldsymbol{J}^{-}$ denotes the inertia-weighted pseudoinverse. In addition, let ϕ denote the acceleration in the NS, which then satisfies $\phi = (\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J})\ddot{\boldsymbol{\theta}}$.

In order to obtain a form similar to EXOS, we need to explicitly include the NS part force and acceleration as follows:

$$\ddot{\boldsymbol{x}}_{os} = \begin{bmatrix} \ddot{\boldsymbol{x}} \\ \phi \end{bmatrix} \in \Re^{m+n}; \quad \mathcal{F}_{os} = \begin{bmatrix} \mathcal{F} \\ \Gamma \end{bmatrix} \in \Re^{m+n};$$

$$\boldsymbol{J}_{os} = \begin{bmatrix} \boldsymbol{J} \\ (\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J}) \end{bmatrix} \in \Re^{(m+n)\times n}. \tag{44}$$

Then it is easy to obtain the inverse inertia matrix,

$$\boldsymbol{M}_{os}^{-1} = \boldsymbol{J}_{os}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}_{os}^{T}$$

$$= \begin{bmatrix} \boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & \boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}(\boldsymbol{I}_n - \boldsymbol{J}\boldsymbol{J}^{T-}) \\ (\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J})\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & (\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J})\boldsymbol{M}_{\theta}^{-1}(\boldsymbol{I}_n - \boldsymbol{J}\boldsymbol{J}^{T-}) \end{bmatrix} \tag{45}$$

$$= \begin{bmatrix} \boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J})\boldsymbol{M}_{\theta}^{-1}(\boldsymbol{I}_n - \boldsymbol{J}^{T}\boldsymbol{J}^{T-}) \end{bmatrix} \in \Re^{(n+m)\times(n+m)}.$$

Through (45), the original OS dynamics in (34), which deals only with the force-acceleration relationship in the OS is now made to include both the OS part and NS part, as well as their couplings. Note in (45) that $\boldsymbol{M}_{os}^{-1}$ has the same OS part as $\boldsymbol{M}_{x_E}^{-1}$ of (39), and that its NS part plainly shows the *explicit* relationship between the NS acceleration and the NS force. Further, (45) clearly confirms the decoupling of the OS part from the NS part as mentioned in reference [1]. Compared to its original form, thus, the OS formulation has become more transparent.

Note, however, that the NS part in (45) has a dimension of $n\times n$ instead of $r\times r$, implying that more relationships are provided than necessary. As a result, the inertia matrix also has the dimension of $(n+m)\times(n+m)$ instead of $n\times n$. Owing to these overlapped relationships, not only becomes the dynamic formulation larger than necessary in its size, but also is the NS dynamics difficult to interpret and understand, causing *another class of transparency problem*. For example, a 3 DOF planar manipulator with $m=2(r=1)$ has only *one*-dimensional NS. Yet the inertia matrix in (45) has a dimension of 5×5, instead of 3×3 in the EXOS; its NS part consists of *9 elements*,* instead of one, making it difficult to understand the NS dynamics. For instance, Fig. 4(b) shows the 6 elements representing the NS part of $\boldsymbol{M}_{os}^{-1}$ for the 3 DOF manipulator introduced in IV-B. Even for such a simple case, the NS dynamics is very difficult to interpret.

A close inspection of derivation procedure reveals that this problem is *inevitable* as long as $(\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J})$ is used as the NS matrix. One can observe this causality more transparently, by using the following relationship:

$$\boldsymbol{I}_n - \boldsymbol{J}^{-}\boldsymbol{J} = \boldsymbol{Z}^{T}(\boldsymbol{Z}\boldsymbol{M}_{\theta}\boldsymbol{Z}^{T})^{-1}\boldsymbol{Z}\boldsymbol{M}_{\theta}. \tag{46}$$

Clearly, the NS of the OS formulation may be interpreted as the mapping of the $\boldsymbol{Z}\boldsymbol{M}_{\theta} \in \Re^{r\times n}$ into joint space by $\boldsymbol{Z}^{T}(\boldsymbol{Z}\boldsymbol{M}_{\theta}\boldsymbol{Z}^{T})^{-1}$ – note that the essential relationships, $(\boldsymbol{Z}\boldsymbol{M}_{\theta})$,

* Being a symmetric matrix, the NS part has 6 different elements.

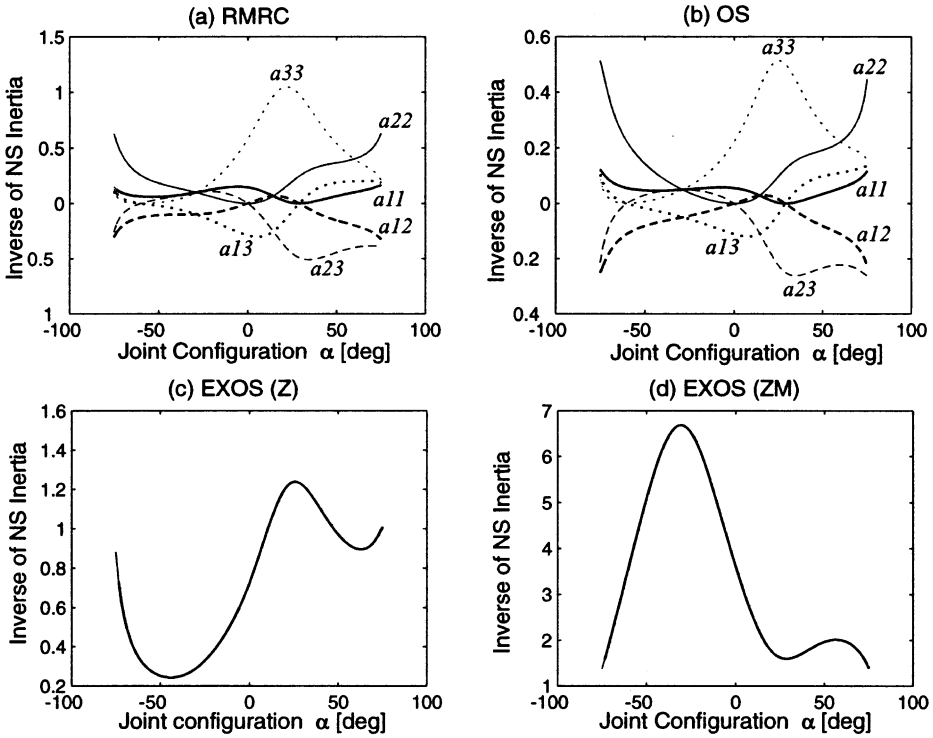


Fig. 4. Elements of inverse of inertia matrix along the self motion manifold for showing NS dynamics of (a) RMRC, (b) OS, (c) EXOS, and (d) EXOS with $\boldsymbol{Z}\boldsymbol{M}_{\theta}(\alpha = \theta_1 + \theta_2 + \theta_3$ with the end-effector fixed at $(x, y) = (2, 0))$.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

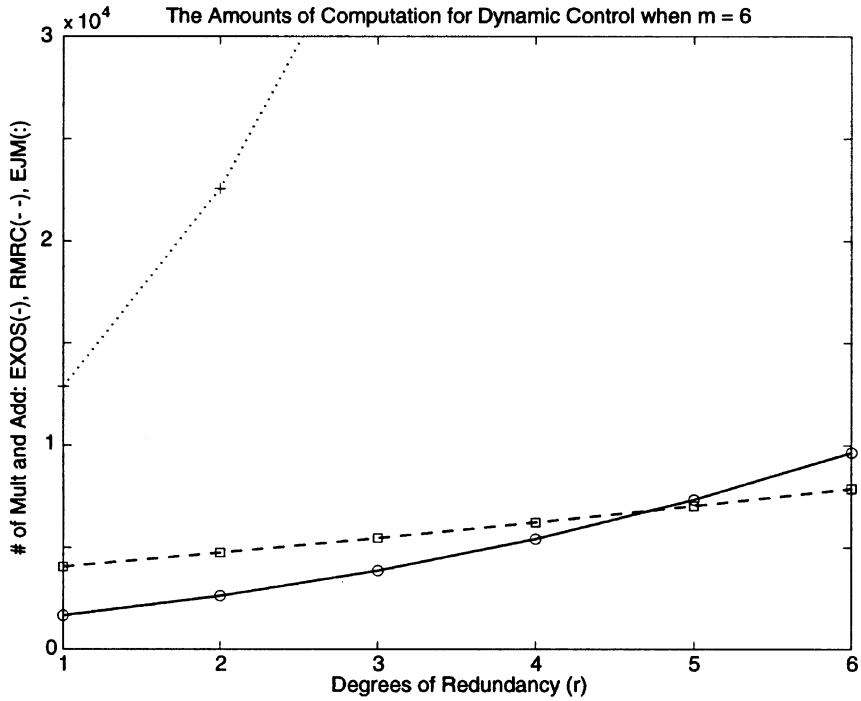


Fig. 5. The computational effort to evaluate $\ddot{\boldsymbol{\theta}}^*$'s for computed torque control based on EJM, RMRC, and EXOS, respectively.

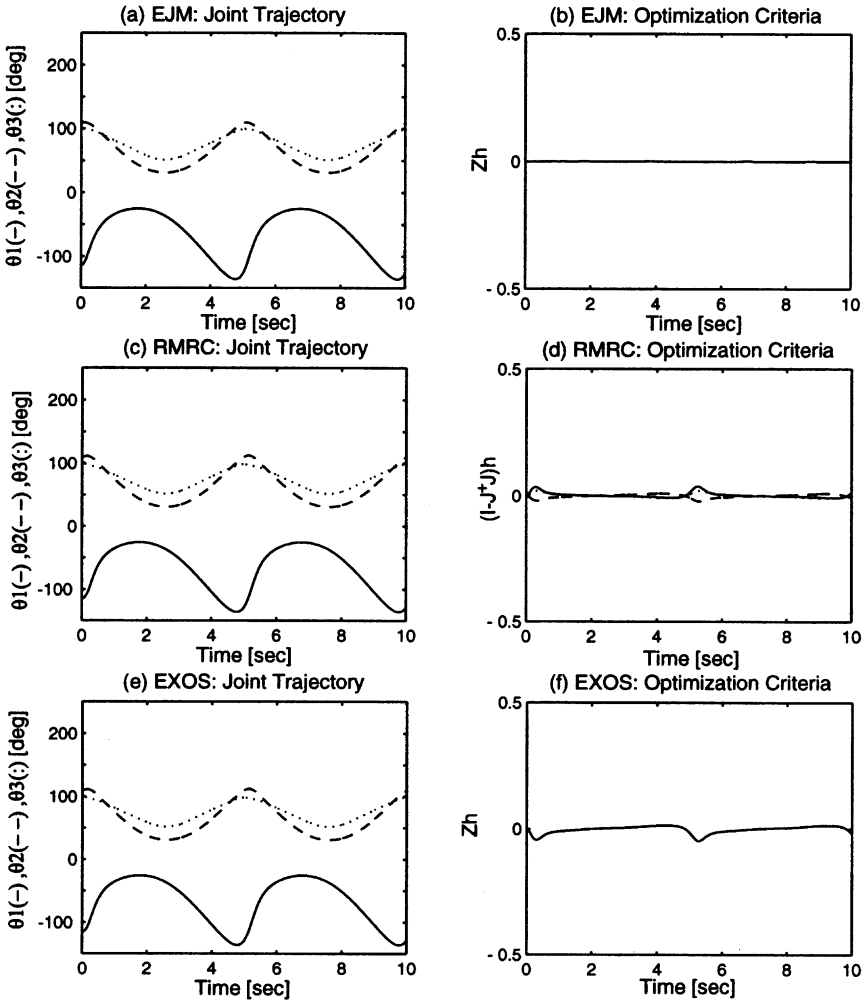


Fig. 6. Simulation results of dynamic controllers based on EJM, RMRC, and EXOS when the initial joint angles of the manipulator is $\boldsymbol{\theta}(0) = [-114.6°, 109.6°, 99.6°]$.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

are *dispersed* into more overlapping relationships. Using (46), we can easily render (45) into the following form:

$$\boldsymbol{M}_{os}^{-1}=\begin{bmatrix}\boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Z}^{T}(\boldsymbol{Z}\boldsymbol{M}_{\theta}\boldsymbol{Z}^{T})^{-1}\boldsymbol{Z}\end{bmatrix}. \tag{47}$$

Note also that the essential relationships, $\boldsymbol{Z}\boldsymbol{M}_{\theta}\boldsymbol{Z}^{T}\in\Re^{r\times r}$, are dispersed into more overlapping relationships, $\boldsymbol{Z}^{T}(\boldsymbol{Z}\boldsymbol{M}_{\theta}\boldsymbol{Z}^{T})\boldsymbol{Z}\in\Re^{n\times n}$. Comparing (46) and (47) plainly explains why the aforementioned problem occurs.

At the same time, the comparison explains why it is so important to start with a compact NS matrix such as $\boldsymbol{Z}$ or $\boldsymbol{Z}\boldsymbol{M}_{\theta}$ at kinematic level. If we had begun with $\boldsymbol{Z}\boldsymbol{M}_{\theta}$, we would have been able to come up with (40), the NS part of which is exemplified in Fig. 4(d) for the manipulator above.

IV.C.3. The Dynamics due to the Resolved Motion Method. Since the NS matrix for the RMM is given as $(\boldsymbol{I}_n-\boldsymbol{J}^{+}\boldsymbol{J})$, the NS acceleration, ϕ_{rm}, satisfies $\phi_{rm}=(\boldsymbol{I}_n-\boldsymbol{J}^{+}\boldsymbol{J})\ddot{\boldsymbol{\theta}}$. For the NS, it is natural to assume that $\tau=\boldsymbol{J}^{T}\mathcal{F}+(\boldsymbol{I}_n-\boldsymbol{J}^{T}\boldsymbol{J}^{T+})\Gamma_{rm}$, with Γ_{rm} denoting the null force. Then we have

$$\ddot{\boldsymbol{x}}_{rm}=\begin{Bmatrix}\ddot{\boldsymbol{x}}\\ \phi_{rm}\end{Bmatrix}\in\Re^{m+n};\quad \mathcal{F}_{rm}=\begin{Bmatrix}\mathcal{F}\\ \Gamma_{rm}\end{Bmatrix}\in\Re^{m+n};$$

$$\boldsymbol{J}_{rm}=\begin{bmatrix}\boldsymbol{J}\\ (\boldsymbol{I}_n-\boldsymbol{J}^{+}\boldsymbol{J})\end{bmatrix}\in\Re^{(m+n)\times n}. \tag{48}$$

By using these relationships, it is easy to obtain the inverse inertia matrix as follows:

$$\boldsymbol{M}_{rm}^{-1}=\boldsymbol{J}_{rm}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}_{rm}^{T}=$$

$$\begin{bmatrix}\boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & \boldsymbol{J}\boldsymbol{M}_{\theta}^{-1}(\boldsymbol{I}_n-\boldsymbol{J}\boldsymbol{J}^{T+})\\ (\boldsymbol{I}_n-\boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{M}_{\theta}^{-1}\boldsymbol{J}^{T} & (\boldsymbol{I}_n-\boldsymbol{J}^{+}\boldsymbol{J})\boldsymbol{M}_{\theta}^{-1}(\boldsymbol{I}_n-\boldsymbol{J}\boldsymbol{J}^{T+})\end{bmatrix}\in\Re^{(n+m)\times(n+m)}. \tag{49}$$

As with the OS method, we could separate the NS part from the OS part, thereby achieving better transparency. Yet, the NS part again has the dimension of $n\times n$, sharing the same problem with the OS case. Worse yet, the off-diagonal blocks are nonzero in this case, and have dimensions of $m\times n$ (lower part of diagonal) and $n\times m$ (upper part), instead of $m\times r$ and $r\times m$ with EXOS. Like the NS part, the

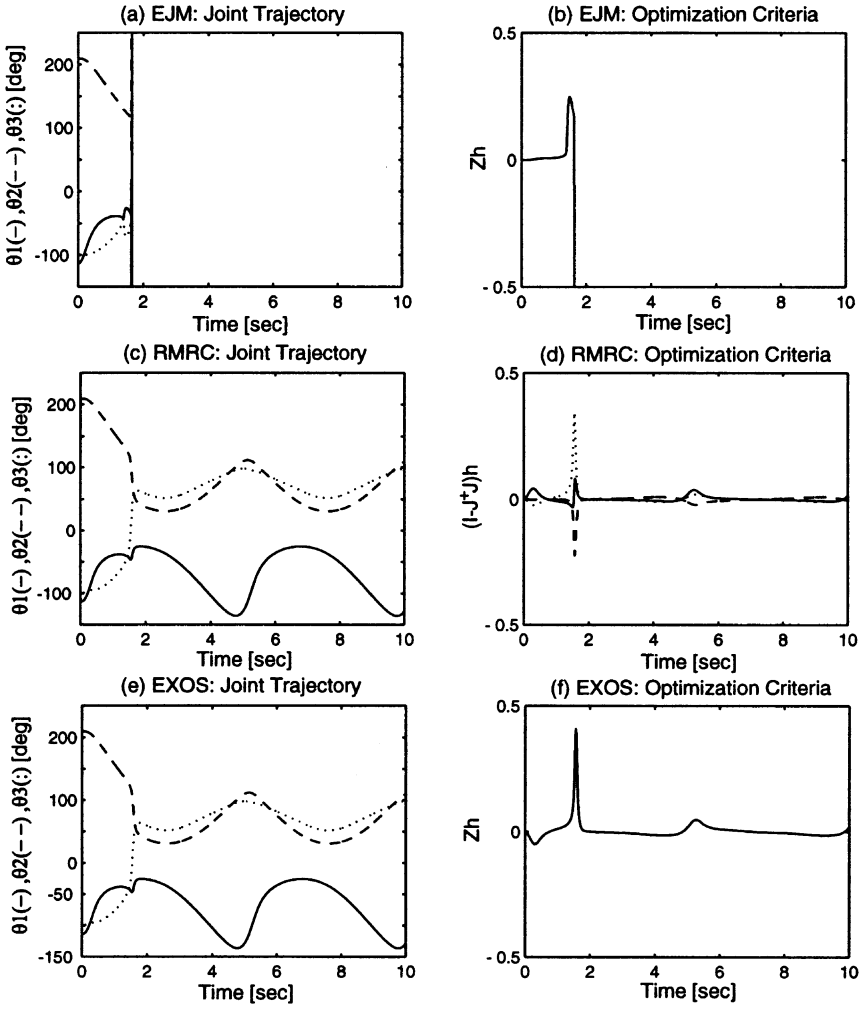


Fig. 7. Simulation results of dynamic controllers based on EJM, RMRC, and EXOS when the initial joing angles of the manipulator is $\boldsymbol{\theta}(0)=[-113.3°, 209.1°, -100.2°]$.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

overlapping relationships in the blocks also make it difficult to interpret the couplings of the NS part with the OS part.

For example, the 3 DOF manipulator above have 6 elements for the NS part, and 6 for each of the blocks. Although the 6 elements in Fig. 4(a) represent the NS dynamics, it is very difficult to find a truly meaningful relationship such as the one in Fig. 4(c).

Like the OS case, this problem comes directly and inevitably from *the way the NS is constructed*. One can see this causality more transparently, by noting the relationship between the NS of the RMM and that of EXOS as follows:

$$\boldsymbol{I}_n - \boldsymbol{J}^+\boldsymbol{J} = \boldsymbol{Z}^T\boldsymbol{Z}. \tag{50}$$

which is easy to derive by using $\boldsymbol{I}_n - \boldsymbol{J}^+\boldsymbol{J} = \boldsymbol{Z}^+\boldsymbol{Z}$ and $\boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{I}_r$. The NS of RMM may be interpreted as *the NS of EXOS mapped by* $\boldsymbol{Z}^T$ *into joint space* – note that the essential relationships are dispersed by $\boldsymbol{Z}^T$ into more overlapping relationships. As an immediate outcome, the following relationship results,

$$\boldsymbol{M}_{rm}^{-1} = \begin{bmatrix} \boldsymbol{J}\boldsymbol{M}_\theta^{-1}\boldsymbol{J}^T & \boldsymbol{J}\boldsymbol{M}_\theta^{-1}\boldsymbol{Z}^T)\boldsymbol{Z} \\ (\boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{M}_\theta^{-1}\boldsymbol{J}^T) & (\boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{M}_\theta^{-1}\boldsymbol{Z}^T)\boldsymbol{Z} \end{bmatrix} \in \Re^{n\times n}. \tag{51}$$

Note also that the essential relationships, $\boldsymbol{Z}\boldsymbol{M}_\theta^{-1}\boldsymbol{Z}^T \in \Re^{r\times r}$, are dispersed into more overlapping relationships, $\boldsymbol{Z}^T(\boldsymbol{Z}\boldsymbol{M}_\theta^{-1}\boldsymbol{Z}^T)\boldsymbol{Z} \in \Re^{n\times n}$. The examples and relationships derived clearly display how important it is to use a succinct and compact NS matrix such as $\boldsymbol{Z}$ at kinematic level, thereby demonstrating the usefulness of EXOS approach.

V. DYNAMIC CONTROL BASED ON THE EXOS

As mentioned at the end of Section IV.2, we are going to propose two dynamic compensation techniques which on one hand guarantee (in Proposition 1) the tracking of a desired trajectory in OS, and on the other hand ensure (in Proposition 2) the control of self-motion. The effectiveness of the compensation techniques is tested through both a simulation and an experiment.

V.A. Proposed dynamic control

Consider the two dynamic control laws based on the CTM – the former of which computes control torque, τ_c, in joint space, whereas the latter in EXOS – as follows:

- Type-1

$$\tau_c = \boldsymbol{M}_\theta \boldsymbol{J}_{\mathrm{E}}^{-1}(\ddot{\boldsymbol{x}}_{\mathrm{E}}^* - \dot{\boldsymbol{J}}_{\mathrm{E}}\dot{\boldsymbol{\theta}}) + N_\theta \tag{52}$$

- Type-2

$$\tau_c = \boldsymbol{J}_{\mathrm{E}}^T\mathcal{F}_{\mathrm{E}}$$
$$\mathcal{F}_{\mathrm{E}} = \boldsymbol{M}_{x_{\mathrm{E}}}\ddot{\boldsymbol{x}}_{\mathrm{E}}^* + N_{x_{\mathrm{E}}}, \tag{53}$$

where the desired EXOS acceleration is defined as

$$\ddot{\boldsymbol{x}}_{\mathrm{E}}^* = \begin{pmatrix} \ddot{\boldsymbol{x}}^* \\ \ddot{\boldsymbol{x}}_{\mathrm{N}}^* \end{pmatrix} = \begin{pmatrix} \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p e \\ \ddot{\boldsymbol{x}}_{\mathrm{N}d} + \boldsymbol{K}_{\mathrm{N}}\dot{\boldsymbol{e}}_{\mathrm{N}} \end{pmatrix}. \tag{54}$$

In (54), $\boldsymbol{e} \equiv \boldsymbol{x}_d - \boldsymbol{x} \in \Re^m$ denotes the OS tracking error vector, $\dot{\boldsymbol{e}}_{\mathrm{N}} \equiv \dot{\boldsymbol{x}}_{\mathrm{N}d} - \dot{\boldsymbol{x}}_{\mathrm{N}} \in \Re^r$ the NS velocity tracking error vector; $\boldsymbol{K}_v \in \Re^{m\times m}$, $\boldsymbol{K}_p \in \Re^{m\times m}$, and $\boldsymbol{K}_{\mathrm{N}} \in \Re^{r\times r}$ denote feedback gain matrices.

V.A.0.a Proposition 1. *Let the control input,* τ_c*, be given by either Type-1 or Type-2 control scheme. If the manipulator does not go through a kinematic singularity, then this control law guarantees that the tracking errors converges to zero exponentially* (The proof is given in the appendix.)

Note that these dynamic compensation schemes require accurate dynamic models and intensive real-time computation. These requirements can be alleviated by using indirect methods to compute torque such as the Time Delay Control.[19, 20]

The following proposition based Hsu et al.'s proposition[21] guarantees the local optimization through self-motion.

V.A.0.b Proposition 2. *Suppose that* $\dot{\boldsymbol{x}}(0) = \boldsymbol{0}_m$ *and* $\boldsymbol{x}_d(t) = \boldsymbol{x}(0)$ *for all* $t \geq 0$*. Let* $H : \Re^n \rightarrow \Re$ *be a* C^2 *function such that H restricted to the constraint set* $\Psi(\boldsymbol{x}(0))$ *has an isolated local minimum at* $\boldsymbol{\theta}^*$*. Suppose that* $\boldsymbol{h} = k_h\nabla H$ *and* $\dot{\boldsymbol{x}}_{\mathrm{N}d} = \boldsymbol{Z}\boldsymbol{h}$*, where* $k_h(<0)$ *is a constant and let the control input* τ_c *be given by the Proposition 1. If the manipulator does not go through a singularity, then there exist* $\epsilon > 0$ *and a neighborhood* $A \subset \Psi$ *of* $\boldsymbol{\theta}^*$ *such that*

$$\boldsymbol{\theta}(0) \in A \tag{55}$$

$$\| \dot{\boldsymbol{e}}_{\mathrm{N}}(0) \| < \epsilon \tag{56}$$

implies

$$\boldsymbol{\theta}(0) \rightarrow \boldsymbol{\theta}^*. \tag{57}$$

(This proposition can immediately be proved by replacing $(\boldsymbol{I}_n - \boldsymbol{J}^+\boldsymbol{J})$ in reference [21] with $\mathbf{Z}$ in (10).)

V.A.0.c Remarks

- Clearly, the size of the set A in Proposition 2 depends upon the performance measure H. If H behaves reasonably well, the set A can be quite large. Especially, a convex H enlarges the A to the entire constraint set.[21]
- If $\dot{\boldsymbol{x}}$ is relatively small for the size of k_h given, it is easy to see that the system will behave similarly. However, in case that $\dot{\boldsymbol{x}} \neq 0$, proof of convergence is much more difficult since both Ψ and $\boldsymbol{\theta}^*$ depend upon the end effector position.[21]
- As shown in Proposition 2, the constant k_h determines not only the choice between maximization and minimization of H, but also the speed and boundary of optimization. For example, with a large negative k_h, the performance measure can be minimized with a high speed and the boundary of optimization becomes small. The size of k_h, however, cannot be increased at will, because an excessive value of k_h can make the system unstable. In general, the joint torque limit is known to be one of the constraints involved with a large k_h. Accordingly, Chung et al.[22] derived the limit of k_h from the joint torque limit.

V.B. Comparison with other dynamic control laws

Among dynamic control laws for redundant manipulators, we have selected two control laws for comparison that attempt to track a given end effector trajectory while providing for the control of the self-motion: the computed torque method (CTM) based on the EJM and the CTM

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

Table I. Computation for the Three Methods.

Method	The amounts of computation
EJM	$-30 + 26m + 55m^2 + 9m^3 + 2m^4 + (25 + 110m + 31m^2 + 14m^3)r + (55 + 33m + 28m^2)r^2 + (11 + 22m)r^3 + 6r^4$
RMM	$-31 + 56m + 14m^2 + 12m^3 + (49 + 25m + 12m^2)r + (7 + 2m)r^2$
EXOS	$-31 + 53m + 7m^2 + 2m^3 + (48 + 18m + 12m^2)r + (11 + 16m)r^2 + 6r^3$

based on the RMM.[21] Hence the three methods have the CTM in common; the difference lies in how joint acceleration is computed. In other words, the control torque τ_c is determined as:

$$\tau_c = \hat{\boldsymbol{M}}_\theta(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}^* + \hat{N}_\theta(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{58}$$

where $\hat{(\cdot)}$ denotes the dynamic model and $\ddot{\boldsymbol{\theta}}^*$ stands for the joint acceleration to be determined by each of the three methods. $\ddot{\boldsymbol{\theta}}^*$ for the three methods are given as the following:

EJM

$$\ddot{\boldsymbol{\theta}}^* = \begin{bmatrix} \boldsymbol{J} \\ \dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\theta} \end{bmatrix}^{-1} \begin{pmatrix} \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} \\ -\boldsymbol{K}_1\dfrac{d}{dt}(\boldsymbol{Z}\boldsymbol{h}) - \boldsymbol{K}_2\boldsymbol{Z}\boldsymbol{h} - \dot{\boldsymbol{Z}}\dot{\boldsymbol{\theta}} \end{pmatrix} \tag{59}$$

RMM[21]

$$\begin{aligned} \ddot{\boldsymbol{\theta}}^* = {} & \boldsymbol{J}^+(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}}) + \\ & (\boldsymbol{I}_n - \boldsymbol{J}^+\boldsymbol{J})(\dot{\boldsymbol{h}} + \boldsymbol{K}_N(\boldsymbol{I}_n - \boldsymbol{J}^+\boldsymbol{J})(\boldsymbol{h} - \quad \boldsymbol{\theta})) \\ & -(\boldsymbol{J}^+(\dot{\boldsymbol{J}}\boldsymbol{J}^+ + \dot{\boldsymbol{J}}^+)\boldsymbol{J}(\boldsymbol{h} - \dot{\boldsymbol{\theta}}) \end{aligned} \tag{60}$$

EXOS

$$\ddot{\boldsymbol{\theta}}^* = \begin{bmatrix} \boldsymbol{J} \\ \boldsymbol{Z} \end{bmatrix}^{-1} \begin{pmatrix} \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}} \\ \dfrac{d}{dt}(\boldsymbol{Z}\boldsymbol{h}) + \boldsymbol{K}_N(\boldsymbol{Z}\boldsymbol{h} - \boldsymbol{Z}\dot{\boldsymbol{\theta}}) - \dot{\boldsymbol{Z}}\dot{\boldsymbol{\theta}} \end{pmatrix} \tag{61}$$

On the basis of (58)–(61), the three methods are compared, in terms of computational efficiency, repeatability, and the ability to handle algorithmic singularity. More specifically, computational efficiency is compared in terms of the effort to obtain $\ddot{\boldsymbol{\theta}}^*$ for each method; repeatability through simulations, where the ability to handle the algorithmic singularity is also compared.

The computational effort is compared in terms of the total number of additions and multiplications as functions of m, r – the computation time for addition is assumed to be the same as that of multiplication. For the comparison, we assumed* that $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{x}_d$, $\dot{\boldsymbol{x}}_d$, and $\ddot{\boldsymbol{x}}_d$ are given at each sampling time. In addition $\boldsymbol{h}$, $\boldsymbol{J}$[23] and $\boldsymbol{Z}$ are assumed to be in numerical form and all the derivatives are obtained by using the backward difference. As a performance measure, we incorporated Joint Range Availability (JRA) measure[14] given as the following:

$$H(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\left(\frac{2\theta_i - (\theta_{iu} + \theta_{il})}{\theta_{iu} - \theta_{il}}\right)^2 \tag{62}$$

where θ_{iu} and θ_{il} denote the upper and the lower limit of the i-th joint, respectively. The number of additions and multiplications is estimated for each method† as shown in Table I.

To display the comparison more clearly, we have plotted in Fig. 5 the numbers of additions and multiplications for the case of $m=6$.

As predicted, the excessive amount of computation is required to evaluate $\dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\boldsymbol{\theta}}$ in the EJM. More specifically, $\dfrac{\partial(\boldsymbol{Z}\nabla H)}{\partial\boldsymbol{\theta}}$ requires $(n+1)$ evaluations of $\boldsymbol{J}$, $\boldsymbol{Z}$, and $\boldsymbol{h}$; and $(n+1)^2$ evaluations of H. The comparison between the RMM and EXOS displays that EXOS has the better efficiency up to $r=4$ or $n=10$; when $r \geq 5$ it is the RMM that has the better efficiency. The reason for this tendency is that as r increases, $\boldsymbol{J}_E^{-1}$ requires more computation than $\boldsymbol{J}^+$.[15]

In the simulation to compare repeatability and the ability to handle algorithmic singularity, we have applied the three control laws in (58) through (61) to the planar 3 DOF serial manipulator. The end effector is made to trace a circle having the radius of $1m$ and the center at $(1.5m, 0.0m)$, while self-motion is made to optimize the Manipulability measure mentioned before. The control gains and parameters for simulations are given as $\boldsymbol{K}_v = 40\boldsymbol{I}_2$. $\boldsymbol{K}_p = 400\boldsymbol{I}_2$, $\boldsymbol{K}_1 = 20$, $\boldsymbol{K}_2 = 100$, $\boldsymbol{K}_N = 20$, $k_z = 100$, and $k_h = 50$.

Figure 6 shows the joint trajectories resulting from the three dynamic control laws with the initial joint angles of $[-114.6°, 109.6°, 99.6°]^T$, whereas Fig. 7 with $[-113.3°, 209.1°, -100.2°]^T$. In Fig. 6, EJM plainly shows the better performance in terms of satisfying the optimality condition $(\boldsymbol{Z}\nabla H = 0)$ than the other two. Yet, the repeatability performance is almost the same as the two, showing that the repeatability in the approximate sense is good enough for practical purposes. Note that in the RMM there are three elements representing the optimality condition instead of one in the other two methods.

Figure 7(a) shows that the control based on the EJM breaks down completely at an algorithmic singularity. In contrast, the controls based on the other two methods still

* Only the important assumptions are stated here. For the conditions not explicitly stated here, we adopted the best available ones.

† The details of the estimations are not covered owing to the space limit. They are available upon request to the authors.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

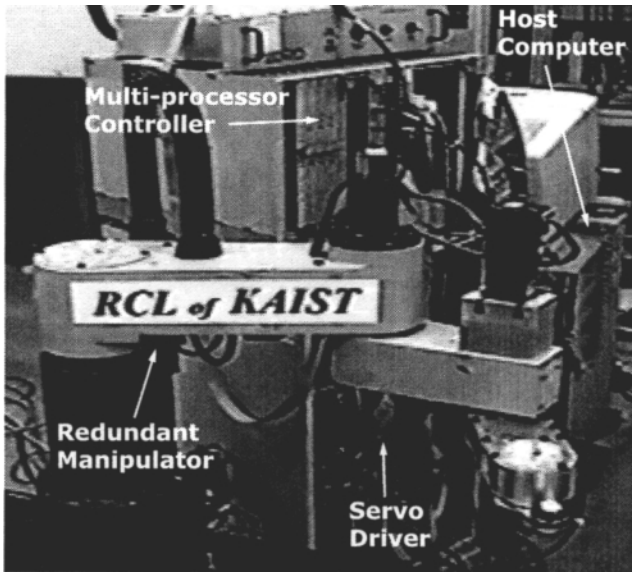


Fig. 8. Experimental setup.

work, although the initial joint configuration changes into another, establishing a different repeatable path. This result clearly shows the performance difference in handling algorithm singularity, and at the same time confirms that the RMM and EXOS are free from it.

It is noteworthy that the three methods compare in exactly the same manner as they do when used for *inverse kinematics purpose*, as mentioned in 3.1 and 3.2.

V.C. Experiment

In order to assure the validity of the control law based on EXOS in real situations, we have carried out an experiment. As shown in the experimental setup of Fig. 8, we used a real planar 3 DOF redundant manipulator, the link properties of which are: 0.35m and 11.17Kg for link 1; 0.20m and 6.82Kg for link 2; 0.15m and 2.0Kg for link 3. At each joint a Harmonic Drive (with gear reduction ratio of 100 : 1 for joint 1, 80 : 1 for joint 2; 80 : 1 for joint 3) is used to transmit power from a joint BLDC motor, which has a resolver attached to its shaft for sensing the angular displacement with the resolution of 4096 pulses/rev. TDC (Time Delay Control)[19, 20] is used to compensate for the dynamic parameter variation and to reject external disturbance. The control hardware, a multi-processing system, is capable of 250 Hz servo rates for this task. Control gains are the same as those of the simulations in the previous section.

The manipulator has joint limits, which are $|\theta_1| \leq 100°$, $|\theta_2| \leq 130°$, and $|\theta_3| \leq 140°$. As a performance measure, the JRA measure in (62) is used. In addition, $\boldsymbol{Z}$ is evaluated by using (17), where k_z is selected to be 250, in accordance with the procedure described in the appendix.

Figure 9 shows the experimental result of self-motion using the EXOS dynamic control law with $k_h = -5$. One can easily observe that $\dot{\boldsymbol{e}}_{\mathrm{N}}$ and $\boldsymbol{Z}\nabla H$ decrease exponentially to zero, yielding good self-motion capability without end-effector motion.

Figure 10 shows the experimental results for minimizing the JRA measure with $k_h = -20$ while the end effector is made to trace a circle. It is observed that repeatable joint trajectory is obtained and very small $\dot{\boldsymbol{e}}_{\mathrm{N}}$ keeps $\boldsymbol{Z}\nabla H$ near zero, indicating that the measure is well optimized.

VI. CONCLUSION

In this study, we have proposed a comprehensive framework, EXOS, which consists of OS vectors and the *minimum number* of NS vectors. In this framework, we formulated kinematics, statics, dynamics, and control of redundant manipulators in a consistent way. The inclusion of only a minimum number of NS vectors from the very

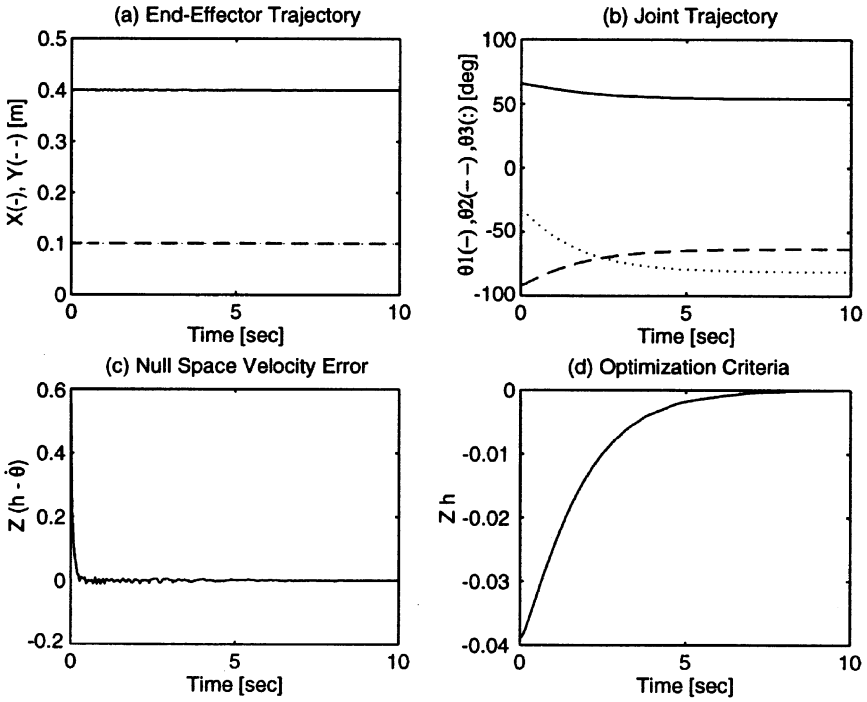


Fig. 9. Experimental results of self-motion using the EXOS dynamic control algorithm.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

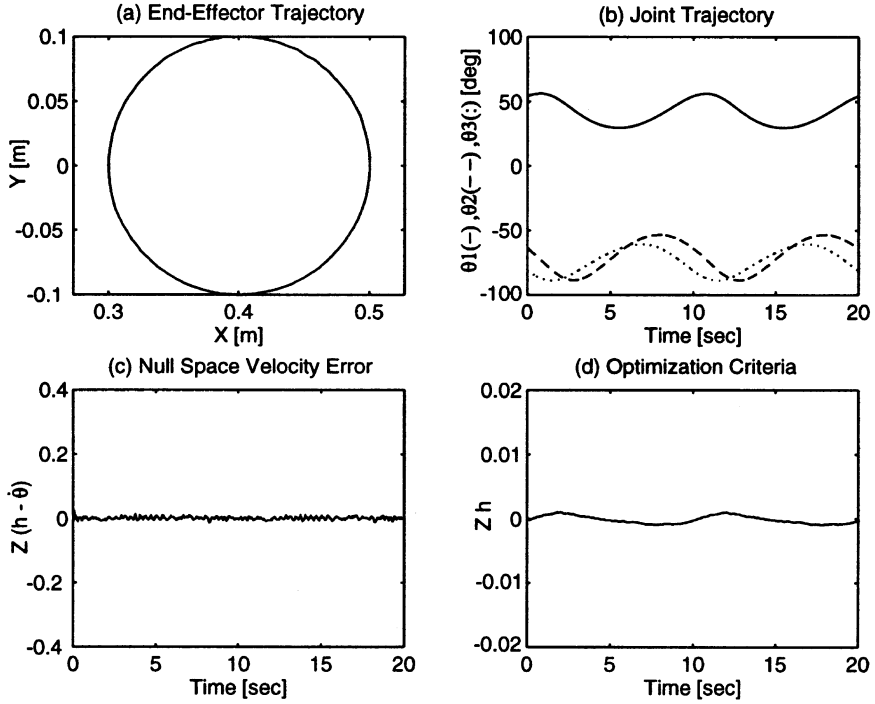


Fig. 10. Experimental results of trajectory motion using the EXOS dynamic control algorithm.

beginning has changed the resulting dynamic equations and control laws into a compact and succinct form.

As the results, the dynamic equations effectively describe not only the dynamic behavior of the end effector, but also that of the self-motion; and at the same time, the interaction of these two motions. The effectiveness was highlighted in the demonstration of the dynamic couplings between OS dynamics and NS dynamics, which are quite elusive in other approaches. Throughout theoretical analysis, simulation, and experiment, the dynamic control based on the EXOS turns out to be not only valid, but also effective and perhaps efficient.

Hence, one may conclude that just as OS formulation provides a good framework for the description and control of the end effector dynamics behavior, so does EXOS for the dynamic behaviors of both the end effector motion and the self-motion. By establishing such a framework, we will be able to merge smoothly the well-known benefits of a redundant manipulator into the OS formulation.

ACKNOWLEDGEMENT

The authors gratefully acknowledge that this work has been supported by the center for Human Welfare Robotics Systems (HWRS).

References

1. O. Khatib, "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation", *IEEE Trans. Robot. Automat.* **RA-3**, No. 1, 43–53 (1987).
2. K.C. Park and P.H. Chang, "Dynamic Control Algorithm for Redundant Manipulators Based on the Velocity-level Extended Operational Space", *Proc. KSME Spring Annual Meeting* (1993) pp. 309–314.
3. P.H. Chang, K.C. Park and S.H. Kim, "Extended Operational Space Formulation for the Kinematics, Dynamics, and Control of the Robot Manipulators with Redundancy", *Trans. Korean Society of Mech Eng.*, **18**, No. 12, 3253–3269 (1999).
4. K.C. Park and P.H. Chang, "An Extension to Operational Space Formulation for Kinematically Redundant Manipulators: Kinematics, Dynamics, and Control", *Proc. 27th Int. Symp. Ind. Robots*, Milan (1996) pp. 533–538.
5. J. Russakow, O. Khatib and S.M. Rock, "Extended Operational Space Formulation for Serial-to-Parallel Chain (Branching) Manipulators", *Proc. IEEE Int. Conf. Robot. Automat.* (1995) pp. 1056–1061.
6. H. Seraji, "Configuration Control of Redundant Manipulators: Theory and Implementation", *IEEE Trans. Robot. Automat.* **5**, No. 4, 472–490 (1989).
7. K.L. Doty, C. Melchiorri and C. Bonivento, "A Theory of Generalized Inverses Applies to Robotics", *Int. J. Robotics Research* **12**, No. 1, 1–19 (1993).
8. J.W. Burdick, "On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds", *Proc. IEEE Int. Conf. Robot. Automat.* (1989) pp. 264–270.
9. G. Strang, *Linear Algebra and Its Applications*, 2nd ed. (Academic Press, 1980).
10. J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. (Addison-Wesley Publishing Co., 1989).
11. P.H. Chang, "A Closed-Form Solution for Inverse Kinematics of Robot Manipulators with Redundancy", *IEEE J. Robot. Automat.* **RA-3**, No. 5, 393–403 (1987).
12. G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Third Edition, 1996).
13. T. Yoshikawa, "Analysis and Control of Robot Manipulators with Redundancy", *Robotics Research: The First Int. Symp.* (1984) pp. 735–748.
14. A. Liégois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms", *IEEE Trans. Syst. Man Cyber.* **SMC-17**, No. 2, 868–871 (1977).
15. P.H. Chang, "Analysis and Control of Robot Manipulators with Kinematic Redundancy", *Technical Report 1022* (MIT AI Lab., 1987).
16. J. Bailieul "Kinematic Programming Alternatives for Redundant Manipulators", *Proc. IEEE Int. Conf. Robot. Automat.* (1985) pp. 722–728.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press

17. S.J. Lee and P.H. Chang, "Analysis and Solution of Problems of Redundant Manipulator", *Proc. KSME Spring Annual Meeting* (1994) pp. 717–722.
18. B.W. Choi, J.H. Won and M.J. Chang, "Manipulability Constraints Locus for a Redundant Manipulator", *Proc IEEE/RSJ IROS* (1991) pp. 267–172.
19. P.H. Chang, B.S. Park and K.C. Park, "An Experimental Study on Improving Hybrid Position/Force Control of Robot Using Time Delay Control", *Int. J. Mechatronics* **6**, No. 8, 915–931 (1996).
20. P.H. Chang, D.S. Kim and K.C. Park, "Robust Force/Position Control of A Robot Manipulator Using Time-Delay Control", *Control Eng. Practice* **3** No. 9, 1255–1264 (1995).
21. P. Hsu, J. Hauser and S. Sastry, "Dynamic Control of Redundant Manipulators", *J. Robotic Systems* **6**, No. 2, 133–148 (1989).
22. W.J. Chung, W.K. Chung and Y. Youm, "Dynamic Control of Redundant Manipulators using Full Row-Rank Minors of Jacobian", *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.* (1993) pp. 1053–1058.
23. D.E. Orin and W.W. Schrader, "Efficient Jacobian Determination for Robot Manipulators", *Proc. The First Int. Symp. Robotics Research* (1984) pp. 727–734.

APPENDIX

Proof of the Proposition 1

Applying control input of type-1 to the joint space dynamics, the closed-loop system is given by

$$\boldsymbol{M}_\theta \ddot{\boldsymbol{\theta}} + N_\theta = \boldsymbol{M}_\theta \boldsymbol{J}_{\mathrm{E}}^{-1}(\ddot{\boldsymbol{x}}_{\mathrm{E}}^{*} - \dot{\boldsymbol{J}}_{\mathrm{E}}\dot{\boldsymbol{\theta}}) + N_\theta \tag{63}$$

and similarly applying control input of type-2 to the joint space dynamics, the closed-loop system is given by

$$\boldsymbol{M}_{x_{\mathrm{E}}}\ddot{\boldsymbol{x}}_{\mathrm{E}} + N_{x_{\mathrm{E}}} = \boldsymbol{M}_{x_{\mathrm{E}}}\ddot{\boldsymbol{x}}_{\mathrm{E}}^{*} + N_{x_{\mathrm{E}}} \tag{64}$$

which simplify to

$$\ddot{\boldsymbol{x}}_{\mathrm{E}} = \ddot{\boldsymbol{x}}_{\mathrm{E}}^{*}. \tag{65}$$

Combining equation (54) with equation (65), we have

$$\ddot{\boldsymbol{e}} + \boldsymbol{K}_v\dot{\boldsymbol{e}} + \boldsymbol{K}_p\boldsymbol{e} = \boldsymbol{0}_m \tag{66}$$

$$\ddot{\boldsymbol{e}}_{\mathrm{N}} + \boldsymbol{K}_{\mathrm{N}}\dot{\boldsymbol{e}}_{\mathrm{N}} = \boldsymbol{0}_r. \tag{67}$$

Through the proper choice of $\boldsymbol{K}_v$, $\boldsymbol{K}_p$, and $\boldsymbol{K}_{\mathrm{N}}$, $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}_{\mathrm{N}}$ decrease to zero, exponentially. (Q.E.D.)

Selection of k_z

Although (17) provides considerable improvements in accuracy over (14), using (17) does not guarantee similar accuracy for *any* Δt. We still need to make Δt as small as possible. Fortunately, Δt, being the *control sampling time*, has already been selected to be *sufficiently small* so that it can faithfully represent robot motion (including null-motion) up to its maximum bandwidth. For instance, the sampling rate for our experiment was 250 Hz and $\Delta t = 0.004s$, which was small enough, as far as accuracy of $\mathbf{Z}$ is concerned.

k_z is selected on the basis of consideration as the following:

Define

$$Y = \begin{bmatrix} \boldsymbol{J}\boldsymbol{Z}^T \\ \boldsymbol{Z}\boldsymbol{Z}^T \end{bmatrix};\ Y_d = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \text{ and } E = Y_d - Y \tag{68}$$

Then it is easy to show that (17) is equivalent to

$$\dot{\boldsymbol{E}} + k_z\boldsymbol{E} = \boldsymbol{0}, \tag{69}$$

which is asymtotically stable for positive k_z.

Obtaining its backward difference approximation,* we have

$$\frac{\boldsymbol{E}(t) - \boldsymbol{E}(t-\Delta t)}{\Delta t} + k_z\boldsymbol{E}(t-\Delta t) \approx \boldsymbol{0} \tag{70}$$

from which we have

$$\boldsymbol{E}(t) + (k_z\Delta t - 1)\boldsymbol{E}(t-\Delta t) \approx \boldsymbol{0} \tag{71}$$

In order for $\boldsymbol{E}(t)$ to converge, we need to have

$$|\,1 - k_z\Delta t\,| < 1. \tag{72}$$

In addition, the convergence rate is determined by $k_z\Delta t$. Since $k_z\Delta t > 0$, the range of $k_z\Delta t$ becomes

$$0 < k_z\Delta t < 2. \tag{73}$$

Note that the closer the value of $k_z\Delta t$ becomes to 2, the slower the convergence of $\boldsymbol{E}(t)$. In the experiment, we used $k_z\Delta t = 1$. Given Δt, therefore, $k_z = \frac{1}{\Delta t}$. In our case, $k_z = \frac{1}{\Delta t} = 250$.

It is to be noted that we can equalize to some extent the accuracy of $\mathbf{Z}$ for different sampling times by using $k_z = \frac{1}{\Delta t}$, provided that the different values of Δt are small enough.

* The rigorous background and detailed derivation, being considered to be out of the scope of this paper, will be provided upon request to the second author.

https://doi.org/10.1017/S0263574701003599 Published online by Cambridge University Press