

# Optimal, Fault-Tolerant Mappings to Achieve Secondary Goals without Compromising Primary Performance

Yixin Chen, *Student Member, IEEE*, John E. McInroy, *Member, IEEE*, and Yong Yi

Yixin Chen is with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA. Research was performed when the author was with the Department of Electrical Engineering, University of Wyoming. E-mail: [yixchen@cse.psu.edu](mailto:yixchen@cse.psu.edu).

John E. McInroy and Yong Yi are with the Department of Electrical Engineering, University of Wyoming, Laramie, WY 82071, USA. E-mail: [mcinroy@uwyo.edu](mailto:mcinroy@uwyo.edu); [yongyi@uwyo.edu](mailto:yongyi@uwyo.edu).

## Abstract

In many applications, the manipulations require only part of the degrees of freedom (DOFs) of the end-effector, or some DOFs are more important than the rest. We name these applications prioritized manipulations. The end-effector's DOFs are divided into those which are critical and must be controlled as precisely as possible, and those which have loose specifications, so their tracking performance can be traded-off to achieve other needs. In this paper, for the class of general constrained rigid multibody systems (including passive joints and multiple closed kinematic loops), we derive a formulation for partitioning the task space into major and secondary task directions and finding the velocity and static force mappings that precisely accomplish the major task and optimize some secondary goals such as reliability enhancement, obstacle and singularity avoidance, fault tolerance, or joint limit avoidance. The major task and secondary goals need to be specified in term of velocities/forces. In addition, a framework is developed to handle two kinds of common actuator failures, torque failure and position failure, by reconfiguring the differential kinematics and static force models. The techniques are tested on a 6-DOF parallel robot. Experimental results illustrate that the approach is practical and yields good performance.

**Index Terms**— Robot kinematics, task decomposition, fault tolerance, parallel robots, multibody systems.

## 1 Introduction

A manipulator is kinematically redundant if the number of active joints is greater than the number of degrees of freedom (DOFs) of its end-effector. This extra freedom offers many advantages over conventional nonredundant manipulators including reliability improvement [26], [27], dexterity improvement [30], [12], plan selection [32], [23], fault tolerance [31], [17], singularity and workspace obstacle avoidance [1], [18], joint limit avoidance [2], [3], and energy (joint torque) minimization [10].

In all the references listed above and most of the other previous works on redundant manipulators, the redundancy comes from the joint space (redundant joints). However, in many applications the redundancy can also occur in the Cartesian space (operational space) [28]. For example, in welding, the positions of the welding rod are crucial to the performance while rotations about the welding rod may be irrelevant. Similarly, when pointing a camera at distant objects [16],[22], orientation is more important than position. Thus the DOFs related to orientation (for welding) or position (for pointing a camera) can be viewed as redundant in the sense that they can be sacrificed for some particular reasons such as enhancing system reliability, avoiding obstacles and singularities in the workspace,

optimizing kinematic performance indices, tolerating actuator failures, etc.

Following the concept of *task priority* in [28], we call these manipulations *prioritized manipulations*. A manipulator performing prioritized manipulation tasks is called a *prioritized manipulator*. The end-effector’s DOFs during a prioritized manipulation can be divided into major DOFs (MDOFs), which are critical in performing a task, and secondary DOFs (SDOFs), which are less important. For example, when a manipulator executes a pointing task, tracing the trajectory of the object is given higher priority than avoiding obstacles in the workspace because the object’s trajectory must be tracked exactly, whereas loose tolerances are typically sufficient for avoiding obstacles. Thus the MDOFs are those DOFs related to orientation of the camera (or end-effector), and the SDOFs are the remaining DOFs. In general, both MDOFs and SDOFs can be a mixture of position and orientation.

Explicit treatment of MDOFs and SDOFs is important because the *degree of importance* of the end-effector’s DOFs can be directly taken into consideration. Prior methods compromise MDOF performance to achieve secondary goals. This makes weight selection tricky—weighting an SDOF incorrectly can result in MDOF failure. In contrast, a prioritized manipulator can be controlled to accomplish a given MDOF motion with maximal performance. At the same time, an optimal SDOF motion is found to achieve, as “closely” as possible, secondary goals including reliability enhancement, workspace obstacle and singularity avoidance, fault tolerance, or joint limit avoidance. Since prioritized manipulations may be performed either by kinematically redundant or “ordinary” manipulators, the theory developed here is applicable to both.

This article focuses on the problems of finding velocity and static force mappings that exactly generate the desired MDOF motion, optimally achieve secondary goals, and can handle actuator failures. The problem is solved in two steps: constructing the differential kinematics and static force models (with or without actuator failures) followed by finding the optimal mappings for the given differential kinematics and static force models. In the literature, there are rich resources on velocity and static force analysis [5], [9], [14], [29], [34], [35], priority-based task decomposition [24], [28], and fault tolerance [7], [8], [16], [17], [22], [31], [33]. Our scheme is distinct from the previous work in the following aspects:

- It derives a formulation for partitioning the task space into major and secondary task directions and finding the velocity and static force mappings that achieve the major task and optimize some secondary criteria. Although the concept of priority-based task decomposition is not new [24], [28], the scheme described in [28] is applicable only to serial manipulators, while our scheme is developed for a more general class of robots, namely, general constrained rigid multibody systems [34], [35] (including serial manipulators, a multi-finger hand, multiple cooperative robots, and Gough-Stewart platform) with a

more general class of secondary criteria. Merlet et al. [24] propose an algorithm to execute the major task and numerically optimize an arbitrary secondary criterion, which is more general than our scheme in terms of secondary criteria. Nevertheless, their approach only allows one SDOF, while our scheme is applicable for more than one SDOF.

- It develops a framework for incorporating two kinds of common actuator failures (torque and position failures). The torque failure occurs when the actuator can't provide any torque actively [7], [8] (it is also referred to as free swing). The position failure acts as if the actuator is locked [16], [17], [22], [31]. Unlike the previous approaches, which are restricted to serial manipulators with torque failures [7], [8], serial manipulators with position failures [17], [31], or a particular parallel manipulator with position failure [16], [22], our method is applicable to a more general class of applications, namely general constrained rigid multibody systems, with torque failures, position failures, or a mixture of two types failures.

The remainder of this paper is organized as follows. In Section 2, we first summarize the differential kinematics and static force models for a general constrained rigid multibody system performing prioritized manipulation based on Wen and Wilfinger's work [34], [35]. Then, we develop a reconfiguration method to treat the torque and position failures. Section 3 derives the optimal velocity and static force mappings based on several performance indices. Section 4 applies the techniques in Section 2 and 3 to a Gough-Stewart platform performing a 2-DOF tracking task. Finally, we conclude in Section 5 together with a discussion of limitations of the proposed methods.

## 2 Differential Kinematics and Static Force Models

A prioritized manipulation can be performed by a serial robot, a parallel robot, or in a more general sense a general constrained rigid multibody system. Any of these manipulators may also be kinematically redundant. In Section 2.1 the differential kinematics and static force models for general constrained rigid multibody systems are summarized based on [34], [35]. These earlier works are extended by considering constrained active joints and prioritized manipulation. The velocity and force manipulability is discussed in Section 2.2. In the case of actuator failures, reconfiguration methods are discussed in Section 2.3. To be consistent with [34], [35], we will use the following terminology and notation:

- Spatial (general) force at a given frame is an  $m \times 1$  vector  $\begin{bmatrix} torque \\ force \end{bmatrix}$  ( $m \leq 6$ ).
- Spatial (Cartesian) velocity at a given frame is an  $m \times 1$  vector  $\begin{bmatrix} angular\ velocity \\ linear\ velocity \end{bmatrix}$  ( $m \leq 6$ ).

- For a given matrix  $\mathbf{G}$ ,  $\tilde{\mathbf{G}}$  denotes a right annihilator of  $\mathbf{G}$ , i.e.,  $\mathbf{G}\tilde{\mathbf{G}} = \mathbf{0}$ .

## 2.1 Nominal Model

Consider general constrained rigid multibody systems without any actuator failures. Let  $\vec{\theta}$  denote the generalized joint coordinates,  $\vec{\theta}_a \in \mathbb{R}^n$  denote the coordinates of active joints,  $\vec{\theta}_p$  denote the coordinates of passive joints, and  $\vec{\tau}$  represent the torque exerted by active joints. The joint coordinates are partitioned as

$$\vec{\theta} = \begin{bmatrix} \vec{\theta}_a \\ \vec{\theta}_p \end{bmatrix}. \quad (1)$$

For a general kinematically constrained rigid multibody system, the differential kinematics model is given by [34], [35]

$$\vec{v} = \mathbf{J}_T(\vec{\theta})\dot{\vec{\theta}} \quad (2)$$

with a general constraint written in terms of the joint velocity vector

$$\mathbf{J}_C(\vec{\theta})\dot{\vec{\theta}} = \vec{0} \quad (3)$$

where  $\vec{v} \in \mathbb{R}^m$  is the spatial velocity of the end-effector,  $\vec{\theta}$  is partitioned according to (1). Clearly the differential kinematics model of serial manipulators is a special case of the above model since we can let  $\mathbf{J}_C = \mathbf{0}$  and  $\vec{\theta} = \vec{\theta}_a$ . For a general constrained rigid multibody system with passive joints and multiple closed kinematic loops,  $\mathbf{J}_C$  reflects the geometric constraints, imposed by the configuration of the system, on the joint (active and passive) velocities. Examples can be found in [34] and [35].

From the principle of virtual work, we have

$$[\vec{\tau}^T \quad \vec{0}^T] \dot{\vec{\theta}} = \vec{f}^T \vec{v} \quad (4)$$

where  $\vec{\tau}$  is the torque applied by active joints,  $\vec{f} \in \mathbb{R}^m$  is the spatial force (the force that the manipulator exerts at the end-effector). Since the passive joints can't apply any torque actively, this is denoted by  $\vec{0}$  on the left side of equation (4). From (3), we have  $\dot{\vec{\theta}} = \tilde{\mathbf{J}}_C \vec{\zeta}$  where  $\vec{\zeta}$  is an arbitrary vector parameterizing the null space of  $\mathbf{J}_C$ . Thus (4) becomes

$$[\vec{\tau}^T \quad \vec{0}^T] \tilde{\mathbf{J}}_C \vec{\zeta} = \vec{f}^T \mathbf{J}_T \tilde{\mathbf{J}}_C \vec{\zeta} .$$

Because this holds true for any  $\vec{\zeta}$ , we have the force balance equation

$$[\vec{\tau}^T \quad \vec{0}^T] \tilde{\mathbf{J}}_C = \vec{f}^T \mathbf{J}_T \tilde{\mathbf{J}}_C$$

which can be equivalently stated as

$$\begin{bmatrix} \vec{\tau} \\ \vec{0} \end{bmatrix} = \mathbf{J}_T^T \vec{f} + \mathbf{J}_C^T \vec{f}_C \quad (5)$$

where  $\vec{f}_C$  is the ‘‘internal force’’ (the force that enforces the constraint (3)). Equation (5) is the static force model for general constrained rigid multibody systems [34], [35].

In order to find a direct relationship between  $\vec{v}$  and  $\dot{\vec{\theta}}_a$  ( $\vec{\tau}$  and  $\vec{f}$ ), we partition  $\mathbf{J}_T$  and  $\mathbf{J}_C$  according to the dimension of  $\vec{\theta}_a$  and  $\vec{\theta}_p$ :

$$\begin{aligned} \mathbf{J}_T &= \begin{bmatrix} \mathbf{J}_{T_a} & \mathbf{J}_{T_p} \end{bmatrix} \\ \mathbf{J}_C &= \begin{bmatrix} \mathbf{J}_{C_a} & \mathbf{J}_{C_p} \end{bmatrix}. \end{aligned}$$

Thus equations (2), (3), and (5) can be equivalently written as

$$\vec{v} = \mathbf{J}_{T_a} \dot{\vec{\theta}}_a + \mathbf{J}_{T_p} \dot{\vec{\theta}}_p \quad (6)$$

$$\vec{0} = \mathbf{J}_{C_a} \dot{\vec{\theta}}_a + \mathbf{J}_{C_p} \dot{\vec{\theta}}_p \quad (7)$$

$$\vec{\tau} = \mathbf{J}_{T_a}^T \vec{f} + \mathbf{J}_{C_a}^T \vec{f}_C \quad (8)$$

$$\vec{0} = \mathbf{J}_{T_p}^T \vec{f} + \mathbf{J}_{C_p}^T \vec{f}_C. \quad (9)$$

Solving equations (7) and (9) for  $\dot{\vec{\theta}}_p$  and  $\vec{f}_C$  yields

$$\dot{\vec{\theta}}_p = -\mathbf{J}_{C_p}^+ \mathbf{J}_{C_a} \dot{\vec{\theta}}_a + \widetilde{\mathbf{J}}_{C_p} \vec{\xi} \quad (10)$$

$$\vec{f}_C = -\mathbf{J}_{C_p}^{+T} \mathbf{J}_{T_p}^T \vec{f} + \widetilde{\mathbf{J}}_{C_p}^T \vec{\eta} \quad (11)$$

where  $^+$  denotes the Moore-Penrose inverse,  $\vec{\xi}$  and  $\vec{\eta}$  are arbitrary vectors parameterizing the null space of  $\mathbf{J}_{C_p}$  ( $\mathcal{N}(\mathbf{J}_{C_p})$ ) and  $\mathbf{J}_{C_p}^T$  ( $\mathcal{N}(\mathbf{J}_{C_p}^T)$ ), respectively. Substituting (10) and (11) into (6) and (8), we get the differential kinematics and static force models [34], [35]

$$\vec{v} = \mathbf{J} \dot{\vec{\theta}}_a + \mathbf{J}_{T_p} \widetilde{\mathbf{J}}_{C_p} \vec{\xi} \quad (12)$$

$$\vec{\tau} = \mathbf{J}^T \vec{f} + \mathbf{J}_{C_a}^T \widetilde{\mathbf{J}}_{C_p}^T \vec{\eta} \quad (13)$$

where  $\mathbf{J} \in \mathbb{R}^{m \times n}$  is the *manipulability Jacobian* (or in short the Jacobian matrix) defined as

$$\mathbf{J} = \mathbf{J}_{T_a} - \mathbf{J}_{T_p} \mathbf{J}_{C_p}^+ \mathbf{J}_{C_a}. \quad (14)$$

When the end-effector DOF priority is considered,  $\vec{v}$  and  $\vec{f}$  can be ordered such that

$$\vec{v} = \begin{bmatrix} \vec{v}_m \\ \vec{v}_s \end{bmatrix}$$

$$\vec{f} = \begin{bmatrix} \vec{f}_m \\ \vec{f}_s \end{bmatrix}$$

with  $\vec{v}_m \in \mathbb{R}^{m_1}$  ( $\vec{v}_s \in \mathbb{R}^{m_2}$ ) denoting the end-effector velocity in the directions or subspace of MDOFs (SDOFs),  $\vec{f}_m \in \mathbb{R}^{m_1}$  ( $\vec{f}_s \in \mathbb{R}^{m_2}$ ) denoting the end-effector force in the directions (or subspace) of MDOFs (SDOFs), and  $m_1 + m_2 = m$ . Rearranging (12) and (13), we have

$$\begin{bmatrix} \vec{v}_m \\ \vec{v}_s \end{bmatrix} = \begin{bmatrix} \mathbf{J}_m \\ \mathbf{J}_s \end{bmatrix} \dot{\vec{\theta}}_a + \mathbf{J}_{T_p} \widetilde{\mathbf{J}}_{C_p} \vec{\xi} \quad (15)$$

$$\vec{\tau} = \begin{bmatrix} \mathbf{J}_m^T & \mathbf{J}_s^T \end{bmatrix} \begin{bmatrix} \vec{f}_m \\ \vec{f}_s \end{bmatrix} + \mathbf{J}_{C_a}^T \widetilde{\mathbf{J}}_{C_p}^T \vec{\eta} \quad (16)$$

where  $\mathbf{J}_m \in \mathbb{R}^{m_1 \times n}$ ,  $\mathbf{J}_s \in \mathbb{R}^{m_2 \times n}$ .

## 2.2 Velocity and Force Manipulability

Regarding the velocity and force manipulability, we emphasize the following

- *Unmanipulable singularity* [34], [35] corresponds to configurations at which  $\mathbf{J}$  loses row rank. In the serial manipulator case, this is termed a general singularity configuration. It means that there is some direction or subspace in Cartesian space along which it is impossible to move the end-effector no matter which joint rates are selected. In the same direction (or subspace), the end-effector cannot actively exert static forces as desired. That is,  $\vec{f}$  could be increased or decreased in some directions, which define the null space of  $\mathbf{J}$ , with no effect on the value calculated for  $\vec{\tau}$ . A prioritized constrained rigid multibody system is in an unmanipulable singular configuration if  $\mathbf{J}_m$  loses row rank, i.e., there exists unfeasible MDOF Cartesian space motion.
- *Unstable singularity* [34], [35] corresponds to configurations at which  $\mathbf{J}_{T_p} \widetilde{\mathbf{J}}_{C_p} \neq \mathbf{0}$ . Physically an unstable singularity means that the end-effector can move even when all active joints are locked. This is also referred to as *self-motion* or *unactuated task motion*.
- *Irresolvable internal force* [34], [35] corresponds to  $\widetilde{\mathbf{J}}_{C_p}^T \vec{\eta}$  with  $\widetilde{\mathbf{J}}_{C_p}^T \neq \mathbf{0}$ . This occurs when the manipulator is over-constrained, i.e., there are more constraints than the number of passive joints. Physically this implies that some of the constraints may be removed and the constraint forces cannot be uniquely solved through rigid body formulation alone. Detailed discussions of general over-constrained mechanisms can be found in [11], [13], [15], [19], [36].
- *Constrained active joint motion* occurs if  $\mathbf{J}_{C_a}^T \widetilde{\mathbf{J}}_{C_p}^T \neq \mathbf{0}$ . This can be proven by the following derivations. First, multiplying both sides of equation (7) by  $\widetilde{\mathbf{J}}_{C_p}^T$  gives

$$\vec{0} = \widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_a} \dot{\vec{\theta}}_a + \widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_p} \dot{\vec{\theta}}_p. \quad (17)$$

Since  $\mathbf{J}_{C_p}^T \widetilde{\mathbf{J}}_{C_p}^T = \mathbf{0}$ ,  $\widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_p} = \mathbf{0}$ . Then (17) becomes

$$\vec{0} = \widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_a} \dot{\vec{\theta}}_a.$$

Thus the active joint velocity vector  $\dot{\vec{\theta}}_a$  is restricted to  $\mathcal{N}(\widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_a})$  where  $\mathcal{N}(\widetilde{\mathbf{J}}_{C_p}^T \mathbf{J}_{C_a}) = \mathbb{R}^n$  if and only if  $\mathbf{J}_{C_a}^T \widetilde{\mathbf{J}}_{C_p}^T = \mathbf{0}$ . Physically, constrained active joint motion implies possible reduction in the number of end-effector's DOFs.

## 2.3 Actuator Failures

A common failure for a robotic manipulator is due to actuators [25]. Examples include motor winding failure, servo controller power failure, bearing failure, gearbox failure, brake failure, etc. Most of the actuator failures can be characterized as either torque failures or position failures. A torque failure occurs when the actuator can't actively exert any torque (or force). It is named the free swinging failure in [7], [8]. A position failure acts as if the actuator is locked [16], [17], [22], [31], i.e., the joint can't change its angle or length. Previous work, including [7], [8], [16], [17], [22], [31], focuses on either serial manipulators or a particular parallel manipulator, and only one type of failures is allowed. In this section, we introduce a method to reconfigure the differential kinematics and static force models of general constrained rigid multibody systems when position and/or torque failures occur.

### 2.3.1 Torque Failures

Torque failure refers to the loss of torque (or force) on an active joint, for example: a ruptured seal on a hydraulic actuator, the loss of electric power and brakes on an electric actuator, and a mechanical failure in a drive system [7]. After a torque failure, the actuator cannot exert any torque actively. However, the failed joint moves freely under the influence of internal forces (exerted by other active joints), external forces, and gravity, i.e., the failed joint can move not actively but passively<sup>1</sup>. Therefore, kinematically, an active joint with torque failure can be viewed as a passive joint that can only move passively. Not surprisingly, for serial manipulators, torque failure leads to unstable singularity (unactuated task motion). That is why previous work on serial manipulators, such as [7], [8], discusses the minimization of errors, which are induced by torque failures, based on various measures. For general constrained rigid multibody systems, however, a torque failure doesn't necessarily cause an unstable singularity. One can easily validate the velocity and force manipulability by reconstructing the differential kinematics and static force models as follows.

<sup>1</sup>The torque failure presented here is not the only case: for linear actuators a torque failure may result in the joint being locked. Such a situation is defined as a position failure in Section 2.3.2.

Without loss of generality, we suppose torque failures occur to the first  $q$  active joints. This can always be done by arranging  $\vec{\theta}_a$ ,  $\vec{\tau}$ , and the corresponding columns of  $\mathbf{J}_T$  and  $\mathbf{J}_C$ . As mentioned above, we can change the first  $q$  active joints (with torque failures) to passive joints. So we define  $\hat{\theta}_a = [\theta_{q+1}, \dots, \theta_n]^T$ ,  $\hat{\tau} = [\tau_{q+1}, \dots, \tau_n]^T$  consisting of angles and torques of active joints without torque failures ( $\tau_i = 0$  for  $i = 1, \dots, q$ ),  $\hat{\theta}_p = [\theta_1, \dots, \theta_q, \theta_p^T]^T$  to be the angles of passive joints. Then we partition  $\mathbf{J}_T$  and  $\mathbf{J}_C$  according to the dimension of  $\hat{\theta}_a$  and  $\hat{\theta}_p$ :

$$\mathbf{J}_T = \begin{bmatrix} \hat{\mathbf{J}}_{T_a} & \hat{\mathbf{J}}_{T_p} \end{bmatrix} \quad (18)$$

$$\mathbf{J}_C = \begin{bmatrix} \hat{\mathbf{J}}_{C_a} & \hat{\mathbf{J}}_{C_p} \end{bmatrix} \quad (19)$$

where  $\hat{\mathbf{J}}_{T_a}$  consists of the last  $n - q$  columns of  $\mathbf{J}_{T_a}$ ,  $\hat{\mathbf{J}}_{C_a}$  consists of the last  $n - q$  columns of  $\mathbf{J}_{C_a}$ ,  $\hat{\mathbf{J}}_{T_p} = [\vec{J}_{T_{p1}}, \dots, \vec{J}_{T_{pq}}, \mathbf{J}_{T_p}]$ ,  $\hat{\mathbf{J}}_{C_p} = [\vec{J}_{C_{p1}}, \dots, \vec{J}_{C_{pq}}, \mathbf{J}_{C_p}]$ ,  $\vec{J}_{T_{pi}}$  and  $\vec{J}_{C_{pi}}$  are the  $i$ th column of  $\mathbf{J}_{T_p}$  and  $\mathbf{J}_{C_p}$ , respectively. Finally we can derive the new differential kinematics and static force models by replacing  $\vec{\theta}_a$ ,  $\vec{\tau}$ ,  $\mathbf{J}_{T_a}$ ,  $\mathbf{J}_{T_p}$ ,  $\mathbf{J}_{C_a}$ ,  $\mathbf{J}_{C_p}^+$ ,  $\widetilde{\mathbf{J}}_{C_p}$ , and  $\widetilde{\mathbf{J}}_{C_p}^T$  in (14,15,16) with  $\hat{\theta}_a$ ,  $\hat{\tau}$ ,  $\hat{\mathbf{J}}_{T_a}$ ,  $\hat{\mathbf{J}}_{T_p}$ ,  $\hat{\mathbf{J}}_{C_a}$ ,  $\hat{\mathbf{J}}_{C_p}^+$ ,  $\widetilde{\hat{\mathbf{J}}}_{C_p}$ , and  $\widetilde{\hat{\mathbf{J}}}_{C_p}^T$  respectively.

### 2.3.2 Position Failures

Physically, a joint can't change its angle or length when position failure happens to the actuator. This can be mathematically interpreted as

$$\dot{\theta}_i = 0 \quad (20)$$

for some  $i \in \{1, \dots, n\}$ . Thus an active joint with position failure can be viewed as a passive joint with an extra constraint that its joint velocity equals 0. Suppose  $q$  position failures occur. Without loss of generality, we assume the failure happens in the first  $q$  active joints. If a passive joint locks, then the analysis is similar so it is excluded for brevity.

Let  $\hat{\theta}_a = [\theta_{q+1}, \dots, \theta_n]^T$ ,  $\hat{\tau} = [\tau_{q+1}, \dots, \tau_n]^T$  consisting of angles and torques of active joints without position failures,  $\hat{\theta}_p = [\theta_1, \dots, \theta_q, \theta_p^T]^T$  be the angles of passive joints. Then according to the dimension of  $\hat{\theta}_a$  and  $\hat{\theta}_p$ ,  $\mathbf{J}_C$  is partitioned exactly the same as (18). Considering  $q$  constraints in the form of (20) with  $i = 1, \dots, q$ , the general constraint (3) is redefined as

$$\mathbf{J}'_C \begin{bmatrix} \dot{\hat{\theta}}_a \\ \dot{\hat{\theta}}_p \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{J}}_{C_a} & \hat{\mathbf{J}}_{C_p} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \begin{bmatrix} \dot{\hat{\theta}}_a \\ \dot{\hat{\theta}}_p \end{bmatrix} = \vec{0}$$

where  $\hat{\mathbf{J}}_{C_a}$  and  $\hat{\mathbf{J}}_{C_p}$  are given by (19),  $\mathbf{E} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{I}$  is a  $q \times q$  identity matrix. Then we partition  $\mathbf{J}'_C$  according to the dimension of  $\hat{\theta}_a$  and  $\hat{\theta}_p$ :

$$\mathbf{J}'_C = \begin{bmatrix} \mathbf{J}'_{C_a} & \mathbf{J}'_{C_p} \end{bmatrix}$$

with  $\mathbf{J}'_{C_a} = \begin{bmatrix} \hat{\mathbf{J}}_{C_a} \\ \mathbf{0} \end{bmatrix}$ ,  $\mathbf{J}'_{C_p} = \begin{bmatrix} \hat{\mathbf{J}}_{C_p} \\ \mathbf{E} \end{bmatrix}$ . Finally we can derive the new differential kinematics and static force models by replacing  $\vec{\theta}_a$ ,  $\vec{\tau}$ ,  $\mathbf{J}_{T_a}$ ,  $\mathbf{J}_{T_p}$ ,  $\mathbf{J}_{C_a}$ ,  $\mathbf{J}_{C_p}^+$ ,  $\tilde{\mathbf{J}}_{C_p}$ , and  $\widetilde{\mathbf{J}}_{C_p}^T$  in (14,15,16) with  $\hat{\theta}_a$ ,  $\hat{\tau}$ ,  $\hat{\mathbf{J}}_{T_a}$ ,  $\hat{\mathbf{J}}_{T_p}$ ,  $\mathbf{J}'_{C_a}$ ,  $\mathbf{J}'_{C_p} + \widetilde{\mathbf{J}}_{C_p}$ , and  $\widetilde{\mathbf{J}}_{C_p}^T$  respectively.

Mixture of torque and position failures can also be handled by combining the methods in Section 2.3.1 and 2.3.2. For serial manipulators, if position failures occur, the differential kinematics model can be reconstructed by removing the columns of the Jacobian matrix ( $\mathbf{J}$  matrix) corresponding to the failed joints [17], [31]. Nevertheless, this approach may not lead to correct models for general constrained rigid multibody systems because there could be constrained active joint motion as defined in Section 2.2.

### 3 Optimal Velocity and Static Force Mappings

For a kinematically redundant manipulator, there exists a space of joint velocities that give the same end-effector velocity. Utilizing this property, many different performance indices can be optimized by adding terms in  $\mathcal{N}(\mathbf{J})$  to the joint velocities [17], [2], [10], [28]. For prioritized manipulation, the redundancy occurs in Cartesian space. Similarly, this extra freedom can also be used in optimizing certain kinematic performance indices.

In this section, we assume that the general constrained rigid multibody system is not at a singular position, i.e.,  $\mathbf{J}_m$  in (15) and (16) has full row-rank and  $\mathbf{J}_{T_p} \tilde{\mathbf{J}}_{C_p} = \mathbf{0}$ . In addition, there are no irresolvable internal forces i.e.  $\mathcal{N}(\mathbf{J}_{C_p}^T) = \{\vec{0}\}$ . These assumptions can be easily satisfied for most applications. Thus the differential kinematics and static force models given by (15) and (16) can be simplified as

$$\begin{bmatrix} \vec{v}_m \\ \vec{v}_s \end{bmatrix} = \begin{bmatrix} \mathbf{J}_m \\ \mathbf{J}_s \end{bmatrix} \dot{\vec{\theta}}_a \quad (21)$$

$$\vec{\tau} = \begin{bmatrix} \mathbf{J}_m^T & \mathbf{J}_s^T \end{bmatrix} \begin{bmatrix} \vec{f}_m \\ \vec{f}_s \end{bmatrix} \quad (22)$$

If there are joint failures then the above models are reconfigured using the methods in Section 2.3.

For a given active joint velocity  $\dot{\vec{\theta}}_a$ , the Cartesian space velocity of the end-effector is determined by the differential kinematics model (21). However, in many applications, only the Cartesian space trajectory (in terms of  $\vec{v}$ ) of the end-effector is specified. We need to compute  $\dot{\vec{\theta}}_a$  which can generate the desired  $\vec{v}$ . If  $\mathbf{J} = \begin{bmatrix} \mathbf{J}_m \\ \mathbf{J}_s \end{bmatrix}$  is invertible, the solution is given by  $\dot{\vec{\theta}}_a = \mathbf{J}^{-1} \vec{v}$ . If  $\mathbf{J}$  is not invertible, then  $\dot{\vec{\theta}}_a = \mathbf{J}^+ \vec{v}$  gives the active joint velocity with minimum length (2-norm) that produces a Cartesian

space velocity closest to, in the least squares sense, the desired end-effector velocity  $\vec{v}$ .

However, for prioritized manipulations, achieving desired MDOF motion is more important than the accomplishment of SDOF motion. Consequently, instead of using the classical approach ( $\mathbf{J}^+$ ) which minimizes the errors across all DOFs, we propose a method of handling the MDOF and SDOF motions separately. MDOF motions will, if possible, be exactly tracked. If impossible, a solution minimizing MDOF errors will be found. SDOF motions will be traded-off with other needs *without compromising MDOF motion*. This is in stark contrast to a conventional weighted pseudo inverse approach, which compromises MDOF motion in accordance with the weight.

**Problem 1:** Consider the differential kinematics model (21). Given a desired Cartesian space velocity  $\vec{v}_d = \begin{bmatrix} \vec{v}_{md} \\ \vec{v}_{sd} \end{bmatrix} \in \mathbb{R}^m$  and a desired active joint velocity  $\dot{\vec{\theta}}_{ad}$ , find an actual active joint velocity  $\dot{\vec{\theta}}_a \in \mathbb{R}^n$  such that

$$\vec{v}_{md} = \mathbf{J}_m \dot{\vec{\theta}}_a, \quad (23)$$

and

$$\|\mathbf{W}_1(\dot{\vec{\theta}}_a - \dot{\vec{\theta}}_{ad})\|_2^2 + \|\mathbf{W}_2(\vec{v}_s - \vec{v}_{sd})\|_2^2 \quad (24)$$

is minimized.  $\mathbf{W}_1 \in \mathbb{R}^{n \times n}$  and  $\mathbf{W}_2 \in \mathbb{R}^{m_2 \times m_2}$  are weighting matrices.

**Remark 2:** In Problem 1, we try to find a joint velocity which will produce the desired MDOF velocity. At the same time, the secondary goal is optimally accomplished by minimizing the performance criterion (24). Note that this additional SDOF motion will not degrade MDOF motion at all. Two terms are included in (24).  $\|\mathbf{W}_1(\dot{\vec{\theta}}_a - \dot{\vec{\theta}}_{ad})\|_2^2$  denotes the magnitude of the joint space error where  $\dot{\vec{\theta}}_{ad}$  can be specified for joint limits avoidance, reliability enhancement, or energy minimization, etc. The error in SDOF motion is measured by  $\|\mathbf{W}_2(\vec{v}_s - \vec{v}_{sd})\|_2^2$  where  $\vec{v}_{sd}$  may be specified for workspace obstacle avoidance, dexterity improvement, etc. Note that in general  $\dot{\vec{\theta}}_a$  and  $\vec{v}$  have elements with different physical units. Adding terms with different units gives a physically meaningless sum. The weighting matrices are used to avoid this kind of inconsistent operation. Methods of finding appropriate weighting matrices can be found in [6].

Depending on the properties of the weighting matrices, Problem 1 is solved for three cases

- $\mathbf{W}_1$  and  $\mathbf{W}_2$  are nonsingular weighting matrices.
- $\mathbf{W}_1$  is nonsingular and  $\mathbf{W}_2 = \mathbf{0}$ .
- $\mathbf{W}_1 = \mathbf{0}$  and  $\mathbf{W}_2$  is nonsingular.

**Theorem 3:** Let  $\mathbf{W}_1$  and  $\mathbf{W}_2$  be nonsingular weighting matrices. The unique solution for Problem 1

is

$$\begin{aligned} \dot{\theta}_a = & \left\{ \mathbf{J}_m^+ - \tilde{\mathbf{J}}_m \begin{bmatrix} \mathbf{W}_1 \tilde{\mathbf{J}}_m \\ \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_1 \mathbf{J}_m^+ \\ \mathbf{W}_2 \mathbf{J}_s \mathbf{J}_m^+ \end{bmatrix} \right\} \vec{v}_{md} \\ & + \tilde{\mathbf{J}}_m \begin{bmatrix} \mathbf{W}_1 \tilde{\mathbf{J}}_m \\ \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ad} \\ \vec{v}_{sd} \end{bmatrix}. \end{aligned} \quad (25)$$

**Proof:** Given  $\vec{v}_{md} \in \mathbb{R}^{m_1}$ , all solutions of (23) are given by

$$\dot{\theta}_a = \mathbf{J}_m^+ \vec{v}_{md} + \tilde{\mathbf{J}}_m \vec{\zeta} \quad (26)$$

where  $\vec{\zeta} \in \mathbb{R}^{n-m_1}$  is arbitrary. Since  $\vec{v}_s = \mathbf{J}_s \dot{\theta}_a$  we have

$$\begin{aligned} \begin{bmatrix} \mathbf{W}_1(\dot{\theta}_a - \dot{\theta}_{ad}) \\ \mathbf{W}_2(\mathbf{J}_s \dot{\theta}_a - \vec{v}_{sd}) \end{bmatrix} &= \begin{bmatrix} \mathbf{W}_1 \mathbf{J}_m^+ \\ \mathbf{W}_2 \mathbf{J}_s \mathbf{J}_m^+ \end{bmatrix} \vec{v}_{md} + \\ & \begin{bmatrix} \mathbf{W}_1 \tilde{\mathbf{J}}_m \\ \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \end{bmatrix} \vec{\zeta} - \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ad} \\ \vec{v}_{sd} \end{bmatrix}. \end{aligned}$$

It is clear that (24) can be equivalently written as

$$\left\| \begin{bmatrix} \mathbf{W}_1(\dot{\theta}_a - \dot{\theta}_{ad}) \\ \mathbf{W}_2(\mathbf{J}_s \dot{\theta}_a - \vec{v}_{sd}) \end{bmatrix} \right\|_2^2.$$

Since  $\begin{bmatrix} \mathbf{W}_1 \tilde{\mathbf{J}}_m \\ \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \end{bmatrix}$  has full column-rank, (24) is minimized if and only if

$$\begin{aligned} \vec{\zeta} = & - \begin{bmatrix} \mathbf{W}_1 \tilde{\mathbf{J}}_m \\ \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \end{bmatrix}^+ \left\{ \begin{bmatrix} \mathbf{W}_1 \mathbf{J}_m^+ \\ \mathbf{W}_2 \mathbf{J}_s \mathbf{J}_m^+ \end{bmatrix} \vec{v}_{md} \right. \\ & \left. - \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ad} \\ \vec{v}_{sd} \end{bmatrix} \right\}. \end{aligned} \quad (27)$$

Substituting (27) into (26) gives (25).  $\square$

**Theorem 4:** Let  $\mathbf{W}_1$  be nonsingular, and  $\mathbf{W}_2 = \mathbf{0}$ . The unique solution for Problem 1 is

$$\begin{aligned} \dot{\theta}_a = & \left[ \mathbf{J}_m^+ - \tilde{\mathbf{J}}_m (\mathbf{W}_1 \tilde{\mathbf{J}}_m)^+ \mathbf{W}_1 \mathbf{J}_m^+ \right] \vec{v}_{md} + \\ & \tilde{\mathbf{J}}_m (\mathbf{W}_1 \tilde{\mathbf{J}}_m)^+ \mathbf{W}_1 \dot{\theta}_{ad}. \end{aligned} \quad (28)$$

**Proof:** From (26) we have

$$\mathbf{W}_1(\dot{\theta}_a - \dot{\theta}_{ad}) = \mathbf{W}_1 \mathbf{J}_m^+ \vec{v}_{md} + \mathbf{W}_1 \tilde{\mathbf{J}}_m \vec{\zeta} - \mathbf{W}_1 \dot{\theta}_{ad}.$$

Since  $\mathbf{W}_1 \tilde{\mathbf{J}}_m$  has full column-rank,  $\|\mathbf{W}_1(\dot{\theta}_a - \dot{\theta}_{ad})\|_2^2$  is minimized if and only if

$$\vec{\zeta} = -(\mathbf{W}_1 \tilde{\mathbf{J}}_m)^+ \left( \mathbf{W}_1 \mathbf{J}_m^+ \vec{v}_{md} - \mathbf{W}_1 \dot{\theta}_{ad} \right). \quad (29)$$

Substituting (29) into (26) we get (28).  $\square$

**Theorem 5:** Let  $\mathbf{W}_1 = \mathbf{0}$ , and  $\mathbf{W}_2$  be nonsingular. The minimum norm solution for Problem 1 is

$$\begin{aligned} \dot{\theta}_a = & \left[ \mathbf{I} - \tilde{\mathbf{J}}_m \tilde{\mathbf{A}} (\tilde{\mathbf{J}}_m \tilde{\mathbf{A}})^+ \right] \left[ (\mathbf{J}_m^+ - \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{B}) \vec{v}_{md} + \right. \\ & \left. \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{W}_2 \vec{v}_{sd} \right] \end{aligned} \quad (30)$$

where

$$\begin{aligned} \mathbf{A} &= \mathbf{W}_2 \mathbf{J}_s \tilde{\mathbf{J}}_m \\ \mathbf{B} &= \mathbf{W}_2 \mathbf{J}_s \mathbf{J}_m^+. \end{aligned}$$

**Proof:** From (26) we have

$$\mathbf{W}_2 (\mathbf{J}_s \dot{\theta}_a - \vec{v}_{sd}) = \mathbf{B} \vec{v}_{md} + \mathbf{A} \vec{\zeta} - \mathbf{W}_2 \vec{v}_{sd}.$$

It is clear that  $\|\mathbf{W}_2 (\mathbf{J}_s \dot{\theta}_a - \vec{v}_{sd})\|_2^2$  is minimized if and only if

$$\vec{\zeta} = -\mathbf{A}^+ (\mathbf{B} \vec{v}_{md} - \mathbf{W}_2 \vec{v}_{sd}) + \tilde{\mathbf{A}} \vec{\zeta} \quad (31)$$

where  $\vec{\zeta}$  is arbitrary. Substituting (31) into (26) yields

$$\dot{\theta}_a = (\mathbf{J}_m^+ - \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{B}) \vec{v}_{md} + \tilde{\mathbf{J}}_m \tilde{\mathbf{A}} \vec{\zeta} + \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{W}_2 \vec{v}_{sd}, \quad (32)$$

which tells that  $\|\dot{\theta}_a\|_2^2$  is minimized if and only if

$$\vec{\zeta} = -(\tilde{\mathbf{J}}_m \tilde{\mathbf{A}})^+ \left[ (\mathbf{J}_m^+ - \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{B}) \vec{v}_{md} + \tilde{\mathbf{J}}_m \mathbf{A}^+ \mathbf{W}_2 \vec{v}_{sd} \right] + \vec{y}$$

where  $\vec{y} \in \mathcal{N}(\tilde{\mathbf{J}}_m \tilde{\mathbf{A}})$ . Substituting into (32) and rearranging terms, we get (30).  $\square$

A similar optimization problem can be formulated using joint torque and spatial force as follows

**Problem 6:** Consider the static force model (22). Given desired spatial force  $\vec{f}_d = \begin{bmatrix} \vec{f}_{md} \\ \vec{f}_{sd} \end{bmatrix} \in \mathbb{R}^m$  and desired joint torque  $\vec{\tau}_d$ , find an actual joint torque  $\vec{\tau} \in \mathbb{R}^n$  such that

$$\vec{\tau} = \mathbf{J}_m^T \vec{f}_{md} + \mathbf{J}_s^T \vec{f}_s, \quad (33)$$

and

$$\|\mathbf{W}_3 (\vec{\tau} - \vec{\tau}_d)\|_2^2 + \|\mathbf{W}_4 (\vec{f}_s - \vec{f}_{sd})\|_2^2$$

is minimized.  $\mathbf{W}_3 \in \mathbb{R}^{n \times n}$  and  $\mathbf{W}_4 \in \mathbb{R}^{m_2 \times m_2}$  are weighting matrices.

The following theorems give solutions to Problem 6 for different weighting strategies.

**Theorem 7:** Let  $\mathbf{W}_3$  and  $\mathbf{W}_4$  be nonsingular weighting matrices. The unique solution for Problem 6 is

$$\vec{\tau} = \left\{ \mathbf{J}_m^T - \mathbf{J}_s^T \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_m^T \\ \mathbf{0} \end{bmatrix} \right\} \vec{f}_{md} + \mathbf{J}_s^T \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_4 \end{bmatrix} \begin{bmatrix} \vec{\tau}_d \\ \vec{f}_{sd} \end{bmatrix}. \quad (34)$$

**Proof:** From (33) we have

$$\begin{bmatrix} \mathbf{W}_3(\vec{\tau} - \vec{\tau}_d) \\ \mathbf{W}_4(\vec{f}_s - \vec{f}_{sd}) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_m^T \\ \mathbf{0} \end{bmatrix} \vec{f}_{md} + \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix} \vec{f}_s - \begin{bmatrix} \mathbf{W}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_4 \end{bmatrix} \begin{bmatrix} \vec{\tau}_d \\ \vec{f}_{sd} \end{bmatrix}.$$

It is clear that

$$\|\mathbf{W}_3(\vec{\tau} - \vec{\tau}_d)\|_2^2 + \|\mathbf{W}_4(\vec{f}_s - \vec{f}_{sd})\|_2^2 = \left\| \begin{bmatrix} \mathbf{W}_3(\vec{\tau} - \vec{\tau}_d) \\ \mathbf{W}_4(\vec{f}_s - \vec{f}_{sd}) \end{bmatrix} \right\|_2^2.$$

Since  $\begin{bmatrix} \mathbf{W}_3 \mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}$  has full column-rank,  $\|\mathbf{W}_3(\vec{\tau} - \vec{\tau}_d)\|_2^2 + \|\mathbf{W}_4(\vec{f}_s - \vec{f}_{sd})\|_2^2$  is minimized if and only if

$$\vec{f}_s = - \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \left\{ \begin{bmatrix} \mathbf{W}_3 \mathbf{J}_m^T \\ \mathbf{0} \end{bmatrix} \vec{f}_{md} - \begin{bmatrix} \mathbf{W}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_4 \end{bmatrix} \begin{bmatrix} \vec{\tau}_d \\ \vec{f}_{sd} \end{bmatrix} \right\}. \quad (35)$$

Substituting (35) into (33) we get (34).  $\square$

**Theorem 8:** Let  $\mathbf{W}_3$  be nonsingular, and  $\mathbf{W}_4 = \mathbf{0}$ . The unique solution for Problem 6 is

$$\vec{\tau} = [\mathbf{J}_m^T - \mathbf{J}_s^T(\mathbf{W}_3 \mathbf{J}_s^T)^+ \mathbf{W}_3 \mathbf{J}_m^T] \vec{f}_{md} + \mathbf{J}_s^T(\mathbf{W}_3 \mathbf{J}_s^T)^+ \mathbf{W}_3 \vec{\tau}_d. \quad (36)$$

**Proof:** From (33) we have

$$\mathbf{W}_3(\vec{\tau} - \vec{\tau}_d) = \mathbf{W}_3 \mathbf{J}_m^T \vec{f}_{md} + \mathbf{W}_3 \mathbf{J}_s^T \vec{f}_s - \mathbf{W}_3 \vec{\tau}_d.$$

Then  $\|\mathbf{W}_3(\vec{\tau} - \vec{\tau}_d)\|_2^2$  is minimized if and only if

$$\vec{f}_s = -(\mathbf{W}_3 \mathbf{J}_s^T)^+ (\mathbf{W}_3 \mathbf{J}_m^T \vec{f}_{md} - \mathbf{W}_3 \vec{\tau}_d) + \vec{z} \quad (37)$$

where  $\vec{z} \in \mathcal{N}(\mathbf{W}_3 \mathbf{J}_s^T)$ . Since  $\mathbf{W}_3$  is invertible,  $\mathcal{N}(\mathbf{W}_3 \mathbf{J}_s^T) = \mathcal{N}(\mathbf{J}_s^T)$ . Thus substituting (37) into (33) yields (36).  $\square$

**Theorem 9:** Let  $\mathbf{W}_3 = \mathbf{0}$ ,  $\mathbf{W}_4$  be nonsingular, and  $\mathbf{J}_s$  has full row-rank. The unique solution for Problem 6 is

$$\vec{\tau} = \mathbf{J}_m^T \vec{f}_{md} + \mathbf{J}_s^T \vec{f}_{sd}.$$

**Proof:**  $\|\mathbf{W}_4(\vec{f}_s - \vec{f}_{sd})\|_2^2 = 0$  if and only if  $\vec{f}_s = \vec{f}_{sd}$  (since  $\mathbf{W}_4$  is invertible) if and only if  $\mathbf{J}_s^T \vec{f}_s = \mathbf{J}_s^T \vec{f}_{sd}$  (since  $\mathbf{J}_s$  has full row-rank) if and only if  $\vec{\tau} = \mathbf{J}_m^T \vec{f}_{md} + \mathbf{J}_s^T \vec{f}_{sd}$  (from (33)).  $\square$

**Remark 10:** Formulating the goals in terms of velocity and/or force has the advantage that it is suited for instantaneous, on the fly modification of goals. The next section shows, for example, how the ideas can be incorporated into a feedback control loop. A restriction of this approach is that inequality constraints (which may arise due to joint limits, etc.) cannot be directly included. On the other hand, the next section also illustrates how these constraints can be indirectly included by weight and desired velocity selection. To directly include inequality constraints, motion planning techniques such as those developed by Zhang and Ostrowski [37] can be used to generate feedforward commands. The techniques in this paper can then be used in a feedback loop to desensitize the feedforward control.

## 4 Experimental Results

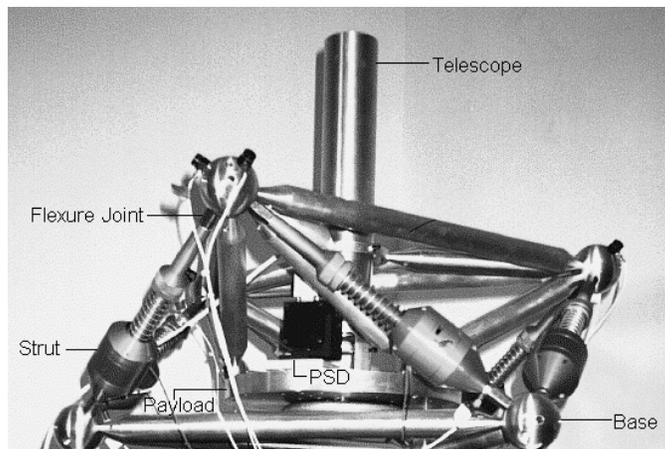


Fig. 1. University of Wyoming (UW) flexure jointed hexapod

The algorithms in Section 2.3 and Section 3 are verified on a University of Wyoming (UW) flexure jointed hexapod (FJH). FJHs are great candidates for micro-precision applications including micro-manipulation, laser weapon pointing, space-based interferometers, and optical communication, etc. Figure 1 shows a photo of the UW FJH in the pointing configuration. The schematic view of the hexapod is given in Figure 2. Like any hexapod, it consists of a base attached to a base plate, a

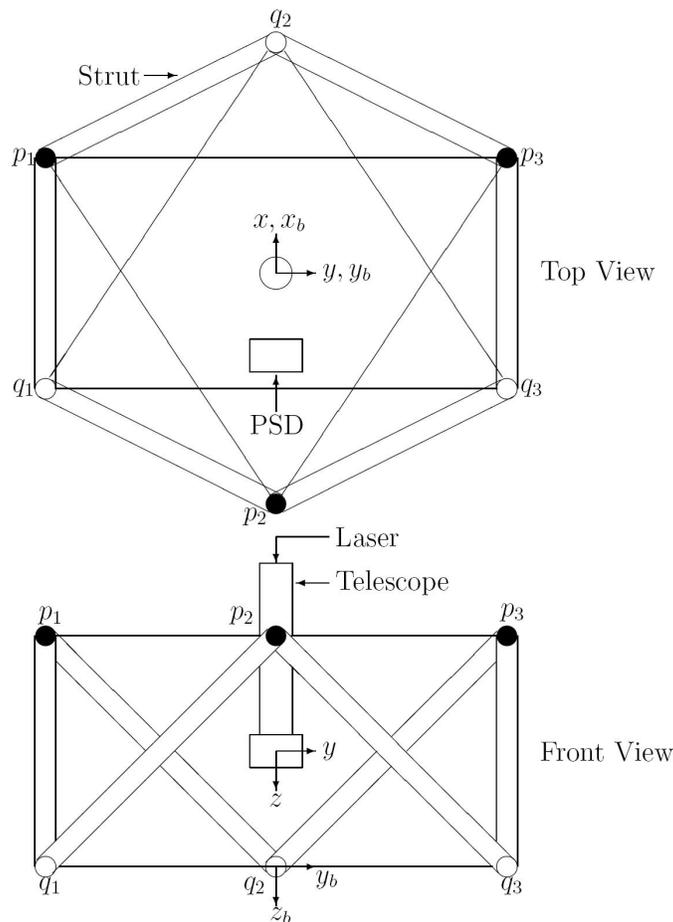


Fig. 2. The top view and front view of the UW FJH. The telescope and position sensitive detector (PSD) sit on the payload, which is rigidly attached to the top nodes  $p_1$ ,  $p_2$ , and  $p_3$ . For the sake of simplicity, the truss structure of the payload is not plotted in the figure. Bottom nodes  $q_1$ ,  $q_2$ , and  $q_3$  are attached to the base. Six struts connect the top nodes to the bottom nodes. The coordinate systems  $\{x, y, z\}$  and  $\{x_b, y_b, z_b\}$  denote the payload frame and base frame, respectively.

payload, and six struts (also called legs) connecting the payload to the base. Each strut contains springs which passively reduce vibrations from the base to the payload plate. A voice coil motor is also embedded into each strut. Thus the hexapod can slightly change the length of its legs to allow precise pose control of the payload in up to six DOFs. The UW FJH employs the mutually orthogonal geometry [20]: each pair of struts meets at a right angle. The system is configured such that the center of mass is located at an equal distance between the top and bottom nodes. As shown in Figure 2, this is the location to which the payload frame is attached.

#### 4.1 Experimental Setup

In this experiment, the hexapod is programmed to perform a *target acquisition* task used to establish space-based two-way laser communications link. In order to set up a laser communications link between

two satellites (“A” and “B”), satellite “A” must first hit satellite “B” with its laser. Satellite “B” must then point its laser at satellite “A”, and send a laser signal back. During this process, satellite “A” is controlled to spirally steer the laser beam toward the “best known” location of satellite “B”, i.e., satellite “A” is commanded to track a spiral signal.

The hexapod in Figure 1 is configured for this 2-DOF tracking task. The laser light passes through a telescope with an effective focal length of 1 *m*. In our experiment, the injecting laser beam is kept stationary with respect to the base frame ( $\{x_b, y_b, z_b\}$ ) of the hexapod. On the payload, a position-sensitive detector (PSD) manufactured by On-Trak Photonics measures the movements of the laser in the payload frame ( $\{x, y, z\}$ ). These measurements are converted to angles of rotation ( $\alpha$  and  $\beta$ ) of the laser beam around the *x* and *y* axes. These angles are essentially the angles of rotation of the *z* axis of the payload frame (or the direction of outgoing laser beam <sup>2</sup>) around the  $x_b$  and  $y_b$  axes of the base frame. The tracking task is to control the pointing direction (*z* axis) to follow control commands. Since the rotation around the  $z_b$  axis doesn’t affect the pointing performance, the pointing direction is defined by the pair of angles  $(\alpha, \beta)$ , and we want  $(\alpha, \beta)$  to track a spiral command signal.

A Pentium-II based computer running the QNX real time operating system sends control commands through Computer Boards 16-bit DAC converters to Techron linear current amplifiers. These activate BEI voice coil actuators which change the length of the legs such that  $(\alpha, \beta)$  tracks the given spiral signal (see [20], [21] for test bed details). Each strut has a nominal length of 0.4064 *m*, and a maximum stroke of  $\pm 0.000635$  *m*.

The differential kinematics model of the hexapod presented in Figure 1 and Figure 2 is given as

$$\begin{bmatrix} \dot{\alpha} & \dot{\beta} & \dot{\gamma} & \dot{t}_x & \dot{t}_y & \dot{t}_z \end{bmatrix}^T = \mathbf{J} \begin{bmatrix} \dot{l}_1 & \dot{l}_2 & \dot{l}_3 & \dot{l}_4 & \dot{l}_5 & \dot{l}_6 \end{bmatrix}^T. \quad (38)$$

$\alpha$ ,  $\beta$ , and  $\gamma$  are the amount of rotation along the *x*, *y*, and *z* axes, respectively.  $t_x$ ,  $t_y$ , and  $t_z$  are the amount of translation along *x*, *y*, and *z* axes, respectively. According to [20], [21], at the configuration that all struts have nominal length of 0.4064 *m*, the Jacobian matrix is computed as

$$\mathbf{J} = \begin{bmatrix} 1.740 & 1.740 & 0.000 & -1.740 & -1.740 & 0.000 \\ 1.004 & -1.004 & -2.009 & -1.004 & 1.004 & 2.009 \\ 0.710 & -0.710 & 0.710 & -0.710 & 0.710 & -0.710 \\ 0.354 & -0.354 & -0.000 & 0.354 & -0.354 & 0.000 \\ 0.204 & 0.204 & -0.408 & 0.204 & 0.204 & -0.408 \\ -0.289 & -0.289 & -0.289 & -0.289 & -0.289 & -0.289 \end{bmatrix}. \quad (39)$$

Since the magnitude of maximum stroke of struts ( $\pm 0.000635$  *m*) is much less than the nominal strut length (0.4064 *m*), the FJH has a very small workspace, and  $\mathbf{J}$  can be closely approximated as constant

<sup>2</sup>The outgoing laser light is not included in our experiment since it is not needed for the closed-loop control of the tracking task.

throughout this workspace. This is validated as follows. As given in [20], all entries of the Jacobian matrix are continuous functions of the leg lengths. Therefore, we use numerical optimization method <sup>3</sup> to find the maximum percentage change, i.e.

$$e = \frac{\sup_{\text{Strut Positions}} \|\mathbf{J}(\text{Strut Position}) - \mathbf{J}_0\|}{\|\mathbf{J}_0\|} \quad (40)$$

where  $\mathbf{J}_0$  is the nominal Jacobian given in (39). Note that the division of these norms tends to cancel out effects caused purely by choice of units. Using the Frobenius norm in (40) gives  $e = 0.0012$ , while using the 2-norm gives  $e = 0.0018$ . Considering that the condition number of  $\mathbf{J}$  equals 4.92, i.e.,  $\mathbf{J}$  is not ill-conditioned, it seems reasonable to assume that  $\mathbf{J}$  is constant across the workspace. An analytic confirmation is theoretically possible, but practically intractable, as it involves symbolically solving the inverse of a  $6 \times 6$  matrix and the notoriously difficult Stewart platform forward kinematics.

Since the tracking task only requires the control of the rotations along the  $x$  and  $y$  axes, the task is clearly a prioritized manipulation with the MDOF Cartesian space velocity  $\vec{v}_m$  defined as  $\vec{v}_m = [\dot{\alpha}, \dot{\beta}]^T$ , the SDOF Cartesian space velocity  $\vec{v}_s$  defined as  $\vec{v}_s = [\dot{\gamma}, \dot{t}_x, \dot{t}_y, \dot{t}_z]^T$ . Consequently the Jacobian matrix  $\mathbf{J}$  is partitioned as  $\mathbf{J} = \begin{bmatrix} \mathbf{J}_m \\ \mathbf{J}_s \end{bmatrix}$  where  $\mathbf{J}_m$  consists of the first two rows of  $\mathbf{J}$ .

However, the differential kinematics model (38) and the algorithms proposed in the above sections can't be directly applied to our closed-loop tracking control for two reasons.

1. The PSD only gives absolute rotation measurements, not rotational velocities.
2. We want to get closed-loop control, with the loop closed with respect to the desired rotations.

Although the loop would be closed with respect to velocities if we have proper measurements, we wouldn't really want to do this in many cases (including this tracking task), because the system becomes sensitive to any kinematic errors. So we use the following strategy, which can also be viewed as a way to handle the case that the task is specified by displacements. From (38), an approximate relationship between displacements can be derived as

$$[\delta\alpha, \delta\beta, \delta\gamma, \delta t_x, \delta t_y, \delta t_z]^T = \mathbf{J} [\delta l_1, \delta l_2, \delta l_3, \delta l_4, \delta l_5, \delta l_6]^T \quad (41)$$

where  $\delta*$  represents the change in  $*$ . Although (41) is only an approximation of the differential kinematics model, it is in fact highly accurate when the control system of the manipulator does its job (i.e.,  $\delta*$ 's can be assumed small), and works well in many applications [4]. All the algorithms

<sup>3</sup>We use the constrained nonlinear optimization procedure `fmincon`, provided by Matlab's Optimization Toolbox, to solve the optimization problem. Finite-differencing derivatives are employed. To alleviate the problem of local minima, `fmincon` is applied with 1000 random initial points.

in Section 2.3 and Section 3 still hold by replacing velocities with displacements (we still call them velocity mappings to simplify notations). Note that, to make the approximation reasonably accurate, only the displacements (not the absolute Cartesian space and joint space coordinates) need to be small.

## 4.2 Reliable and Fault-Tolerant Tracking

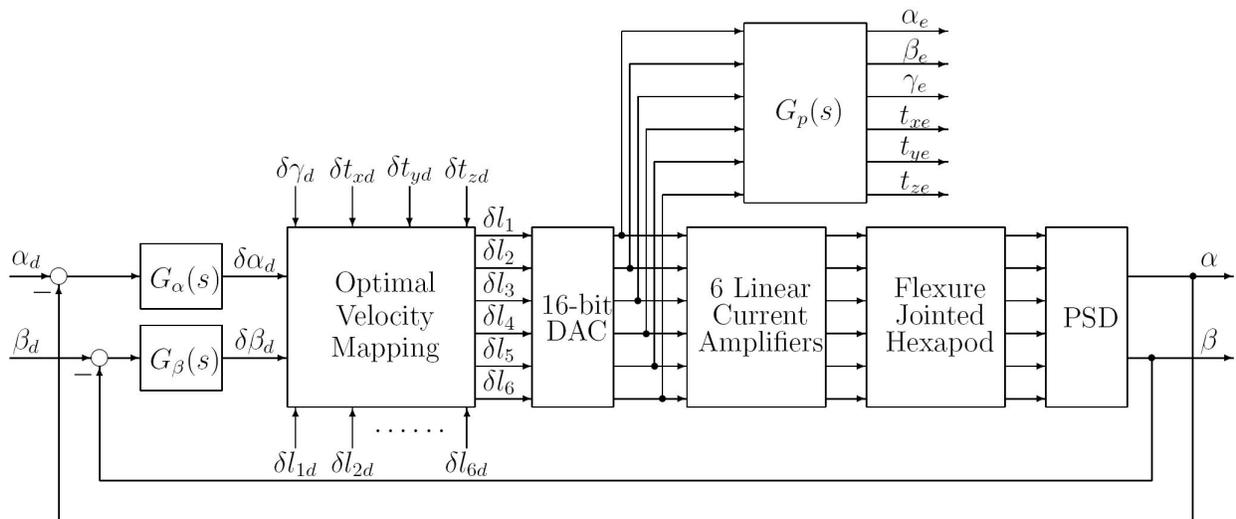


Fig. 3. Block diagram for prioritized 2-DOF tracking control

Figure 3 shows the block diagram of the hexapod control system for prioritized 2-DOF tracking.  $(\alpha_d, \beta_d)$  specifies the command spiral signal.  $[\delta\alpha_d, \delta\beta_d]^T$  is the vector of desired MDOF displacements,  $[\delta\gamma_d, \delta t_{xd}, \delta t_{yd}, \delta t_{zd}]^T$  is the vector of desired SDOF displacements, and  $[\delta l_1, \delta l_2, \dots, \delta l_6]^T$  is the vector of desired joint space displacements. This specific choice corresponds to pointing applications. When pointing at a distant target, angular errors in the pointing angles  $(\alpha, \beta)$  are multiplied by the distance to produce translational errors on the target. Thus the pointing angles are the MDOF. Translational errors in the image plane  $(t_x, t_y)$  are not multiplied by the distance—the same errors appear at the target. These SDOF motions are far less important, but it is desirable to minimize them without compromising MDOF performance.  $G_\alpha(s)$  and  $G_\beta(s)$  are compensators for  $\alpha$  and  $\beta$  channels, respectively. They are found using textbook single-input, single-output compensator designs [21]. Throughout the experiments, the compensators remain unchanged. The values of  $\alpha_e$ ,  $\beta_e$ ,  $\gamma_e$ ,  $t_{xe}$ ,  $t_{ye}$ , and  $t_{ze}$  are estimates of the amount of rotation and translation. We need them only for comparison purposes (the PSD sensor only measures  $\alpha$  and  $\beta$ ).

Next we experimentally demonstrate that, using the optimal velocity mappings given by the algorithms in Section 2.3 and Section 3, the tracking task is accomplished, and at the same time, a secondary goal is also achieved. Here the secondary goals are *reliability enhancement*, and *SDOF*

*motion minimization* (under position failure).

#### 4.2.1 Reliability Enhancement

Reliability has always been a major consideration for military, space, and some manufacturing applications. One way of enhancing the reliability of a manipulator system is to control the actuator's inputs to avoid mechanical fatigue, actuator overheating, etc. For example, if the temperature of a motor is close to its critical value then the input current should be decreased. Using the techniques in Section 3, this can be easily implemented by adjusting the weights on joint displacements. Larger weights imply smaller actuator inputs.

To demonstrate and validate this idea, we define the secondary goal as minimizing  $\|\mathbf{W}_i([\delta l_1, \dots, \delta l_6]^T - [\delta l_{1d}, \dots, \delta l_{6d}]^T)\|_2^2$ ,  $i \in \{a, b, c\}$ . Three weighting matrices:  $\mathbf{W}_a = \mathbf{I}$ ,  $\mathbf{W}_b = \text{Diag}[2, 1, 2, 1, 2, 1]$ , and  $\mathbf{W}_c = \text{Diag}[1, 2, 1, 2, 1, 2]$  are compared. In the experiments, we let  $[\delta l_{1d}, \dots, \delta l_{6d}]^T = \vec{0}$ . The optimal velocity mappings are computed from (28) for all three weighting matrices. We want to demonstrate that, by changing the relative weights on joint space displacements (or equivalently actuator currents), one can manipulate the input current of the actuators accordingly without damaging the tracking performance. Namely, when  $\mathbf{W}_b$  is used, the input currents for actuator 1, 3, and 5 are expected to be relatively smaller than those under  $\mathbf{W}_a$ . Similarly, when  $\mathbf{W}_c$  is used, the input currents for actuator 2, 4, and 6 should be smaller than those under  $\mathbf{W}_b$ . Note that this same technique can be used to indirectly avoid joint limits as it is minimizing the joint displacement required to achieve the goal. To directly avoid joint limits and obstacles, motion pre-planning [37] combined with these methods can be employed.

Figure 4 shows the tracking spirals (plots of the PSD outputs) with starting points centered at the plots and strut inputs for a period of 16 seconds. The tracking errors in terms of the root mean square (RMS) errors are listed in Table I (columns 2 – 4) where  $\alpha_{RMS}$  and  $\beta_{RMS}$  are the RMS values of  $\alpha_d - \alpha$  and  $\beta_d - \beta$ , respectively (i.e., the angular errors in the pointing angles). As we can see, the tracking performance is almost identical for all three weighting matrices. Figure 4 also illustrates that strut inputs decrease when corresponding joint displacements are relatively higher weighted. Note however that, the price paid for this decrease is an increase in the remaining strut inputs, which have lower weights. Here the strut input is the normalized actuator current, which is defined as  $100 \times \frac{\text{DAC input}}{\text{the maximum allowable DAC input}}$ .

#### 4.2.2 Fault-Tolerant with SDOF Motion Minimization

A single actuator failure can put the entire task at risk or cause excessive downtime expenses. The algorithms in Section 2.3 can be utilized to greatly reduce these adverse effects caused by position and

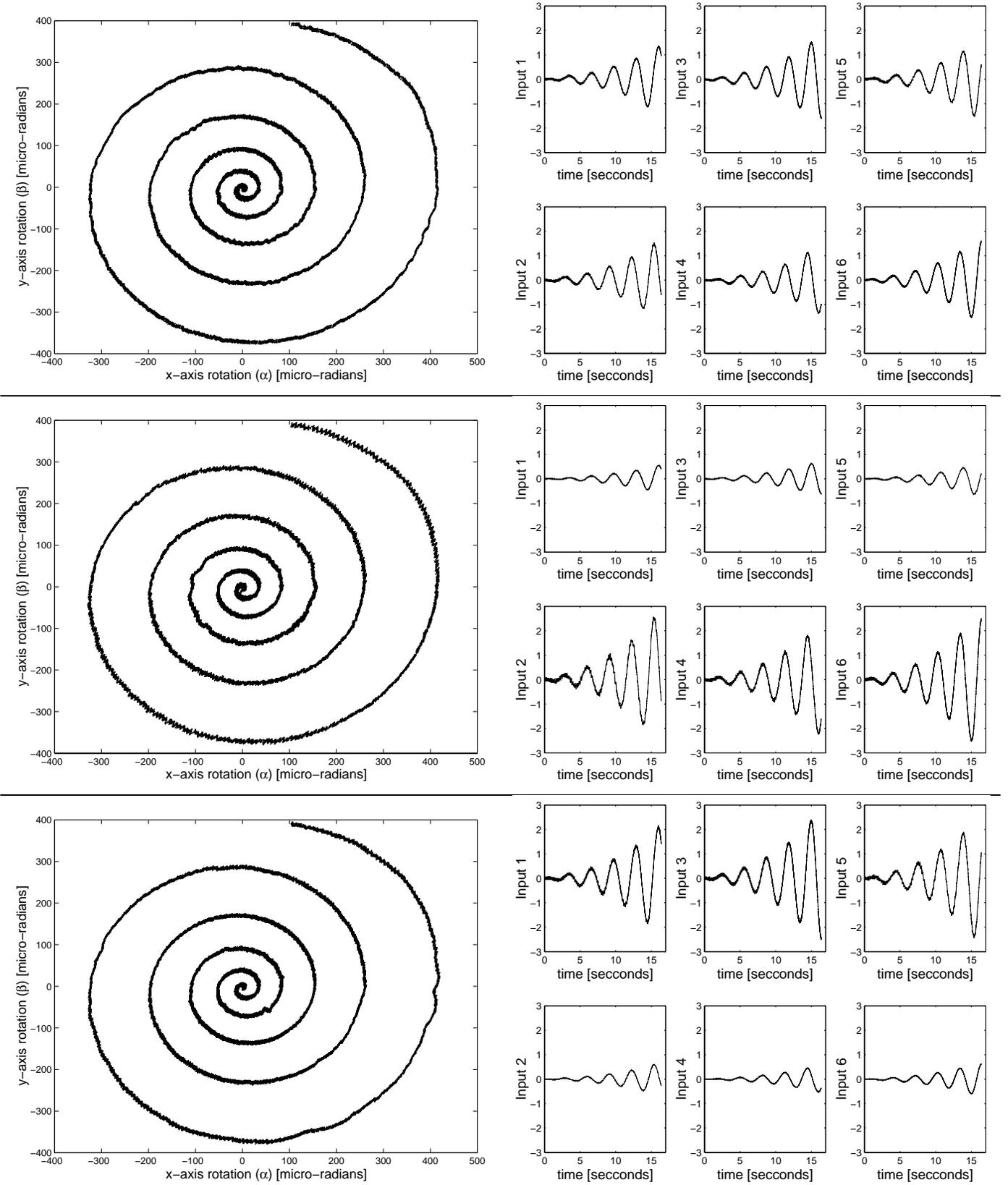


Fig. 4. The tracking spirals and strut inputs when minimizing the weighted joint space displacements. The plots in the left and right columns are tracking spirals (PSD outputs) and the strut inputs, respectively. Here the Input  $i$  denotes the normalized actuator current for Strut  $i$ . From top to bottom, the rows correspond to the weighting matrices  $\mathbf{W}_a$ ,  $\mathbf{W}_b$ , and  $\mathbf{W}_c$ , respectively.

TABLE I

TRACKING ERRORS FOR DIFFERENT WEIGHTING MATRICES.

Weighting Matrix	$\mathbf{W}_a$	$\mathbf{W}_b$	$\mathbf{W}_c$	$\mathbf{W}_d$	$\mathbf{W}_e$	$\mathbf{W}_a$ without Correction
$\alpha_{RMS}$ (micro-radians)	2.4476	2.2444	2.3115	2.3634	2.8324	9.8167
$\beta_{RMS}$ (micro-radians)	3.0916	3.2712	3.2528	3.5601	3.2868	4.7903

torque failures.

Here we assume that position failure occurs in the first strut of the hexapod. When the failure is not taken into consideration and the optimal velocity mapping in Section 4.2.1 for  $\mathbf{W}_a$  is used, the tracking result is shown in Figure 5. The tracking errors are listed in Table I (the last column). The performance degradation is obvious: the  $\alpha_{RMS}$  and  $\beta_{RMS}$  are increased by 300% and 50%, respectively.

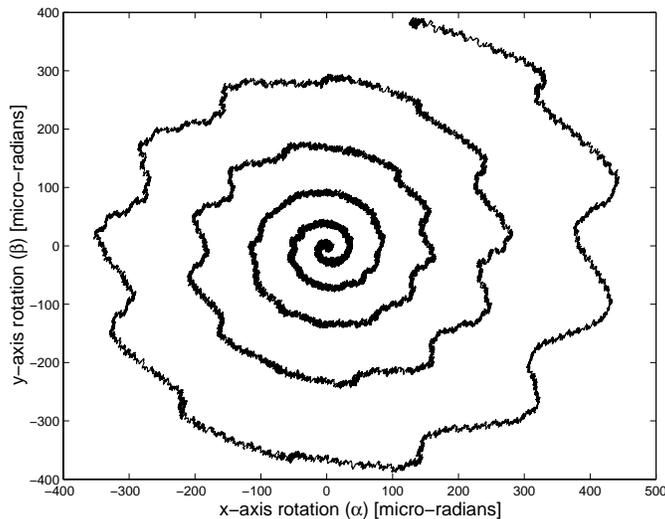


Fig. 5. Tracking spiral without any correction on the position failure in Strut 1.

Using the reconfiguration methods in Section 2.3, and defining the secondary goal as minimizing  $\|\mathbf{W}_i([t_x, t_y, t_z, \gamma]^T - [t_{xd}, t_{yd}, t_{zd}, \gamma_d]^T)\|_2^2$ ,  $i \in \{d, e\}$  where  $\mathbf{W}_d = \mathbf{I}$  and  $\mathbf{W}_e = \text{Diag}[10, 10, 1, 1]$ , and letting  $[t_{xd}, t_{yd}, t_{zd}, \gamma_d]^T = \vec{0}$  in the experiments, we derive the optimal velocity mappings for weighting matrices  $\mathbf{W}_d$  and  $\mathbf{W}_e$ . This choice emphasizes translational errors in the image plane while still preserving the same pointing  $(\alpha, \beta)$  performance.

As shown in Figure 6 and Table I (column 5 and column 6), the tracking performance is comparable to that without strut failures. Moreover, by assigning different weights, SDOF motions can be minimized according to their importance. In the above example, the estimated maximal translation along the  $x$  and  $y$  axes are reduced by more than a factor of 20 when the corresponding weights are increased

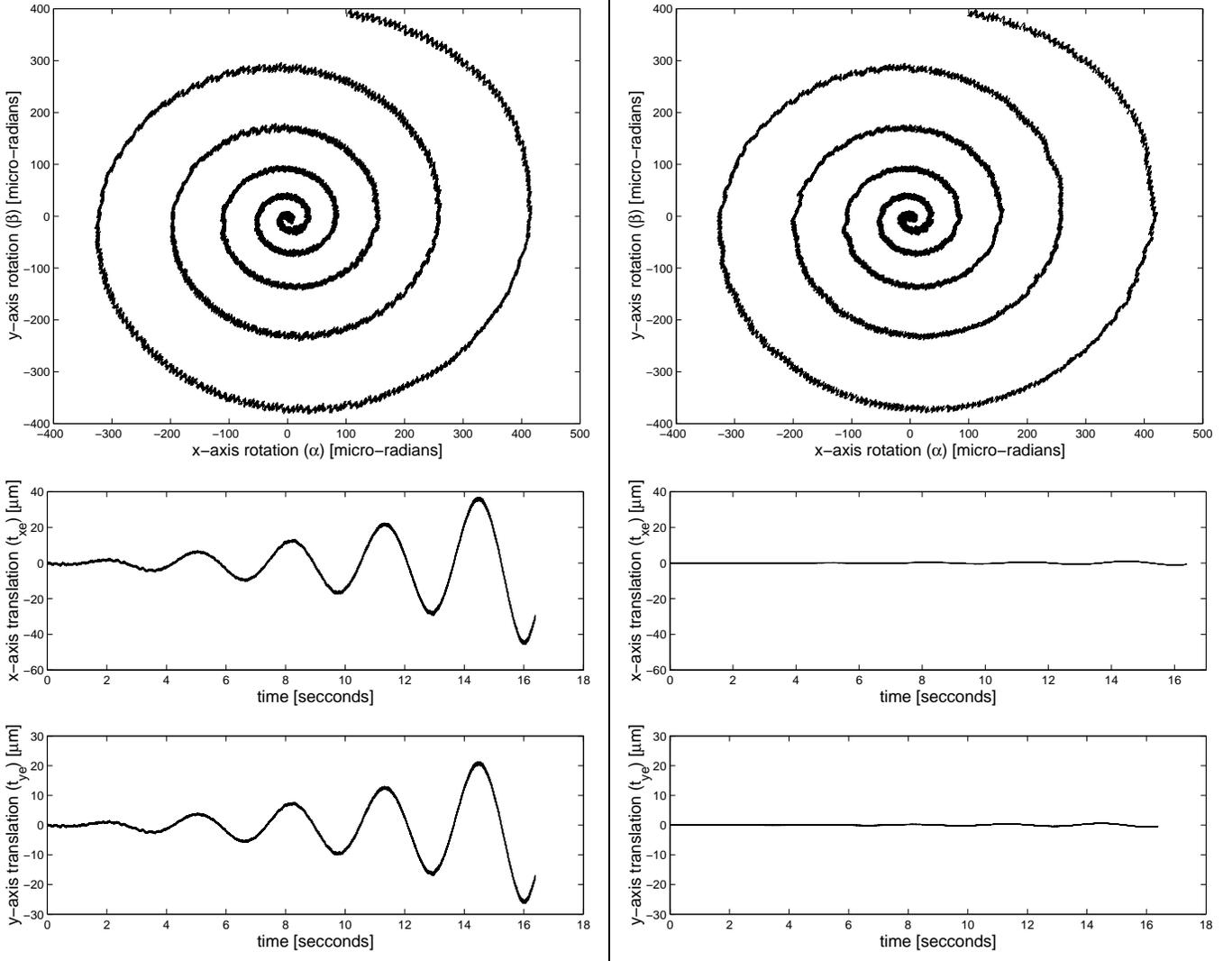


Fig. 6. Tracking spirals and minimizing SDOF displacements with correction on position failure in Strut 1. The left and right columns correspond to weighting matrices  $\mathbf{W}_d$  and  $\mathbf{W}_e$ , respectively.

10-fold.

## 5 Conclusions

In many applications some DOFs in the Cartesian space are critical and must be controlled as precisely as possible. Other DOFs may have quite loose specifications, so their tracking performance can be traded-off to achieve other needs. We call these applications prioritized manipulation, and divide the end-effector's DOFs into MDOFs and SDOFs according to their importance. Based on the differential kinematics and static force models, we derive the velocity and static force mappings which accomplish the given task expressed in MDOFs and, at the same time, optimally complete a secondary goal by picking an appropriate SDOF motion. The secondary goal can be specified for reliability enhancement, obstacle and singularity avoidance, fault tolerance, or joint limit avoidance.

Two common actuator failures, position failure and torque failure, are also considered. During these failures, the differential kinematics and static force models are reconfigured so that the optimal velocity and static force mappings can be calculated using reconfigured models. The proposed algorithms are tested on the UW FJH. Experimental results validate that the approach is practical and demonstrates good performance.

The major limitation of the proposed method is that the task and secondary goals need to be described in terms of velocities. This formulation facilitates instantaneous, on the fly modification of goals. Thus it is useful for some applications (such as trajectory planning for obstacle avoidance) where the task is typically expressed only by velocities. In these cases, even though the proposed method can still achieve the goals locally, it may fail on the whole because a collection of local optimal movements do not necessarily lead to a global solution. In this sense, the mappings found are only locally optimal.

## Acknowledgments

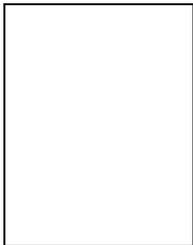
This work was supported by the Ballistic Missile Defense Organization and Army Research Office under grants DAAG55-98-1-0007 and DAAD19-00-1-0153. The authors would like to thank anonymous reviewers and the associate editor for their comments and constructive suggestions.

## References

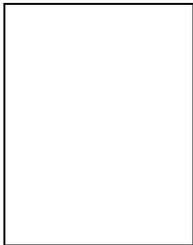
- [1] J. Baillieul, "Avoiding Obstacles and Resolving Redundancy," in *Proceedings of IEEE Conference on Robotics and Automation*, pp. 1698-1703, 1996.
- [2] T. F. Chang and R. V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joints Limits for Redundant Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286-292, 1995.
- [3] F. Chaumette and E. Marchand, "A New Redundancy-Based Iterative Scheme for Avoiding Joint Limits Application to Visual Servoing," in *Proceedings of International Conference on Robotics and Automation*, pp. 1720-1725, 2000.
- [4] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Reading, MA: Addison-Wesley, 1986.
- [5] B. Dasgupta and T. S. Mruthyunjaya, "Force Redundancy in Parallel Manipulators: Theoretical and Practical Issues," *Mechanism and Machine Theory*, vol. 33, no. 6, pp. 727-742, 1998.
- [6] K. L. Doty, C. Melchiorri, and C. Bonivento, "A Theory of Generalized Inverse Applied to Robotics," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 1-19, 1993.
- [7] J. D. English and A. A. Maciejewski, "Fault Tolerance for Kinematically Redundant Manipulators: Anticipating Free-Swinging Joint Failures," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 566-575, 1998.
- [8] J. D. English and A. A. Maciejewski, "Measuring and Reducing the Euclidean-Space Effects of Robotic Joint Failures," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 1, pp. 20-28, 2000.
- [9] C. Gosselin and J. Angeles, "Singularity Analysis of Closed-Loop Kinematic Chains," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281-290, 1990.
- [10] J. M. Hollerbach and K. C. Suh, "Redundancy Resolution of Manipulators through Torque Optimization," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 308-316, 1987.

- [11] J. Kerr and B. Roth, "Analysis of Multifingered Hands," *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 3-17, 1986.
- [12] C. A. Klein and B. E. Blaho, "Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 72-83, 1987.
- [13] C. A. Klein and S. Kittivatcharapong, "Optimal Force Distribution for the Legs of a Walking Machine with Friction Cone Constraints," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 73-85, 1990.
- [14] V. Kumar, "Instantaneous Kinematics of Parallel-Chain Robotic Mechanisms," *ASME Journal of Mechanical Design*, vol. 114, no. 3, pp. 349-358, 1992.
- [15] V. R. Kumar and K. J. Waldron, "Force Distribution in Closed Kinematic Chains," *IEEE Journal of Robotics and Automation*, vol. 4, no. 6, pp. 657-664, 1988.
- [16] X. Li, J. E. McInroy, and J. C. Hamann, "Optimal Fault Tolerant Control of Flexure Jointed Hexapods for Applications Requiring Less Than Six Degrees of Freedom," in *Proceedings of the 39th Conference on Decision and Control*, pp. 3337-3338, 2000.
- [17] A. A. Maciejewski, "Fault Tolerant Properties of Kinematically Redundant Manipulators," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 638-642, 1990.
- [18] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 1095-117, 1985.
- [19] C. Mavroidis, and B. Roth, "Analysis of Overconstrained Mechanisms," *ASME Journal of Mechanical Design*, vol. 117, no. 1, pp. 69-74, 1995.
- [20] J. E. McInroy and J. C. Hamann, "Design and Control of Flexure Jointed Hexapods," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 4, pp. 372-381, 2000.
- [21] J. E. McInroy, G. W. Neat, and J. F. O'Brien, "A Robotic Approach to Fault Tolerant, Precision Pointing," *IEEE Robotics and Automation Magazine*, vol. 6, no. 4, pp. 24-31, 1999.
- [22] J. E. McInroy, J. F. O'Brien, and G. W. Neat, "Precise, Fault Tolerant Pointing using a Stewart Platform," *IEEE/ASME Transactions on Mechatronics*, vol. 4, pp. 91-95, 1999.
- [23] J. E. McInroy and G. N. Saridis, "Techniques for Selecting Pose Algorithms," *Automatica*, vol. 30, no. 3, pp. 471-487, 1994.
- [24] J-P. Merlet, M-W. Perng, and D. Daney, "Optimal Trajectory Planning of a 5-axis Machine-Tool Based on a 6-axis Parallel Manipulator," *Advances in Robot Kinematics*, edited by J. Lenarčič and M.M. Stanišić, Kluwer Academic Publishers, pp. 315-322, 2000.
- [25] V. Monteverde and S. Tosunoglu, "Fault Tolerance in Robotics and Mechanical Systems: an Introduction Survey," in *ASME, PD, Proceedings of the 1996 3rd Biennial Joint Conference on Engineering Systems Design and Analysis, ESDA, part 2 (of 9)*, vol. 74, pp. 259-264, 1996.
- [26] J. C. Musto, "Reliability-Based Inverse Kinematics for Redundant Manipulators," *International Journal of Intelligent Control and Systems*, vol. 3, no. 4, pp. 725-738, 1999.
- [27] J. C. Musto and G. N. Saridis, "Task Reliability Assessment in Robotic Systems," *Journal of Robotic Systems*, vol. 12, no. 9, pp. 583-598, 1995.
- [28] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3-15, 1987.
- [29] L. Notash, "Uncertainty Configurations of Parallel Manipulators," *Mechanism and Machine Theory*, vol. 33, no. 1/2, pp. 123-138, 1998.
- [30] J. F. O'Brien and J. T. Wen, "Redundant Actuation for Improving Kinematic Manipulability," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1520-1525, 1999.
- [31] R. G. Roberts and A. A. Maciejewski, "A Local Measure of Fault Tolerance for Kinematically Redundant Manipulators," in *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 543-552, 1996.
- [32] S. Seereeram and J. T. Wen, "A Global Approach to Path Planning for Redundant Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 152-160, 1995.

- [33] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A Dynamic Fault Tolerance Framework for Remote Robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 477-490, 1995.
- [34] J. T. Wen and L. S. Wilfinger, "Kinematic Manipulability of General Constrained Rigid Multibody Systems," in *Proceedings of International Conference on Robotics and Automation*, pp. 1020-1025, 1998.
- [35] J. T. Wen and L. S. Wilfinger, "Kinematic Manipulability of General Constrained Rigid Multibody Systems," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 558-567, 1999.
- [36] Y. Yi, J. E. McInroy, and Y. Chen, "General Over-Constrained Rigid Multibody Systems: Differential Kinematics, and Fault Tolerance," *SPIE International Symposium on Smart Structure and Materials*, vol. 4701, pp. 189-199, 2002.
- [37] H. Zhang and J. P. Ostrowski, "Visual Motion Planning for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 199-208, 2002.

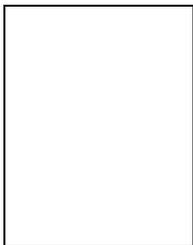


**Yixin Chen** (SM'99) received the B.S. degree in control theory from Beijing Polytechnic University, China, in 1995, the M.S. degree in control theory and application from Tsinghua University, China, in 1998, and the M.S. and Ph.D. degrees in electrical engineering from the University of Wyoming, Laramie, WY, in 1999 and 2001, respectively. Since August 2000, he has been a Ph.D. student in the Department of Computer Science and Engineering, The Pennsylvania State University, University Park. He is expected to receive the Ph.D. degree in 2003. His research interests include machine learning, content-based image retrieval, computer vision, precision and fault tolerant robotic control, and soft computing. He is a student member of the IEEE Computer Society, the IEEE Neural Networks Society, and the IEEE Robotics and Automation Society.



**John E. McInroy** (M'91) received the B.S. degree in electrical engineering from the University of Wyoming, Laramie, in 1986, and both the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute, Troy, NY, in 1991. During his graduate study, he was a Rensselaer Fellow, Xerox Fellow, NASA Graduate Student Researchers Fellow.

He is currently a Professor of Electrical and Computer Engineering at the University of Wyoming, Laramie. His research interests are in the areas of precision robotic control, sensor fusion, machine intelligence, wind surfing, and cross-country skiing.



**Yong Yi** received the B.S. degree in electrical engineering from HUAZHONG University of Science and Technology, China, in July 1993, and M.S. degree in electrical engineering from the University of Wyoming, Laramie, in 2001. Since then, she has been a Ph.D. student in the Department of Electrical and Computer Engineering at the University of Wyoming, Laramie. Her research is in the area of fault tolerant parallel manipulators.