A Task Space Redundancy-Based Scheme for Motion Planning[1]

Yixin Chen [2]
Dept. of Computer Science and Engineering
Penn. State University, University Park, PA 16802
E-mail: yixchen@cse.psu.edu

John E. McInroy
Dept. of Electrical Engineering
University of Wyoming, Laramie, WY 82071
E-mail: mcinroy@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, 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 locally optimize some secondary goals. The techniques are tested on a 6-DOF parallel robot performing a 2-DOF tracking task.

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 in robot planning and control [1, 2, 3, 6, 7, 8, 9, 13, 15, 16, 18, 19]. In all these references and most of the other previous work 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 (task space) [17]. 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 [12], 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 in motion planning for some particular reasons such as enhancing system reliability, avoiding obstacles and singularities in the workspace, optimizing kinematic performance, tolerating actuator failures, etc.

Following the concept of *task priority* in [17], 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.

This article focuses on finding velocity and static force mappings that exactly generate the desired MDOF motion, and optimally achieve secondary goals. Explicit treatment of MDOFs and SDOFs is important because the *degree of importance* of the end-effector's DOFs can be directly taken into account. 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 achieve secondary goals without compromising primary performance. Priority-based task decomposition is not a new idea [14, 17]. Our scheme is distinct from the previous work in the following aspects: 1) Compared with the scheme described in [17], our scheme is developed for a more general class of secondary criteria; 2) The algorithm proposed in [14] is more general than our scheme in terms of secondary criteria. Nevertheless, only one SDOF is allowed there, while our scheme is applicable for more than one SDOF.

The remainder of this paper is organized as follows. In Section 2, we derive the optimal velocity and static

[1] 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.

[2] Research was performed when the author was with the Department of Electrical Engineering, University of Wyoming.

force mappings based on several performance indices. Section 3 presents the experiment setup. Experimental results are given in Section 4. Finally, we conclude in Section 5.

2 Optimal Velocity and Static Force Mappings

At the velocity level, the end effector's motion is related to the joint motion of a manipulator by the differential kinematics model

$$\vec{v} = \mathbf{J}(\vec{\theta})\dot{\vec{\theta}} \tag{1}$$

where $\vec{v} \in \mathbb{R}^m$ ($m \leq 6$) is the spatial velocity of the end-effector, $\vec{\theta} \in \mathbb{R}^n$ denotes the generalized coordinates of active joints, $\mathbf{J} \in \mathbb{R}^{m\times n}$ is the manipulability Jacobian matrix. Following the principle of virtual work, the static force model is

$$\vec{\tau} = \mathbf{J}^T\vec{f} \tag{2}$$

where $\vec{\tau} \in \mathbb{R}^n$ 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).

For a kinematically redundant manipulator ($n > m$), there exists a space of joint velocities that give the same end-effector velocity. Different performance indices can be optimized by adding terms in the null space of $\mathbf{J}$ ($\mathcal{N}(\mathbf{J})$) to the joint velocities. For prioritized manipulation, the redundancy occurs in task space. Similarly, this extra freedom can also be used in optimizing certain kinematic performance indices. 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 $m_1 + m_2 = m$. Then models (1) and (2) can be equivalently written as

$$\begin{bmatrix} \vec{v}_m \\ \vec{v}_s \end{bmatrix} = \begin{bmatrix} \mathbf{J}_m \\ \mathbf{J}_s \end{bmatrix}\dot{\vec{\theta}}_a \tag{3}$$

$$\vec{\tau} = \begin{bmatrix} \mathbf{J}_m^T & \mathbf{J}_s^T \end{bmatrix}\begin{bmatrix} \vec{f}_m \\ \vec{f}_s \end{bmatrix} \tag{4}$$

where $\mathbf{J}_m \in \mathbb{R}^{m_1\times n}$, $\mathbf{J}_s \in \mathbb{R}^{m_2\times n}$. Here we assume that the prioritized manipulator is not at a singular position, i.e., $\mathbf{J}_m$ has full row-rank.

For a given active joint velocity $\dot{\vec{\theta}}_a$, the task space velocity of the end-effector is determined by the differential kinematics model (3). However, in many applications, only the task 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 $\mathbf{J}$ 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 task 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 2.1 *Consider the differential kinematics model (3). Given a desired task 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*

$$\vec{v}_{md} = \mathbf{J}_m\dot{\vec{\theta}}_a, \tag{5}$$

and

$$\|\mathbf{W}_1(\dot{\vec{\theta}}_a - \dot{\vec{\theta}}_{ad})\|_2^2 + \|\mathbf{W}_2(\vec{v}_s - \vec{v}_{sd})\|_2^2 \tag{6}$$

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 2.2 *In Problem 2.1, 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 (6). Note that this additional SDOF motion will not degrade MDOF motion at all. Two terms are included in (6).* $\|\mathbf{W}_1(\dot{\vec{\theta}}_a - \dot{\vec{\theta}}_{ad})\|_2^2$ *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 [5].*

Depending on the properties of the weighting matrices, Problem 2.1 is solved for three cases: 1) $\mathbf{W}_1$ and $\mathbf{W}_2$ are nonsingular weighting matrices; 2) $\mathbf{W}_1$ is nonsingular and $\mathbf{W}_2 = \mathbf{0}$; 3) $\mathbf{W}_1 = \mathbf{0}$ and $\mathbf{W}_2$ is nonsingular.

Theorem 2.3 *Let $\mathbf{W}_1$ and $\mathbf{W}_2$ be nonsingular weighting matrices. The unique solution for Problem 2.1 is*

$$\dot{\vec{\theta}}_a = \left\{ \mathbf{J}_m^+ - \widetilde{\mathbf{J}}_m \begin{bmatrix} \mathbf{W}_1\widetilde{\mathbf{J}}_m \\ \mathbf{W}_2\mathbf{J}_s\widetilde{\mathbf{J}}_m \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_1\mathbf{J}_m^+ \\ \mathbf{W}_2\mathbf{J}_s\mathbf{J}_m^+ \end{bmatrix} \right\} \vec{v}_{md}$$
$$+\widetilde{\mathbf{J}}_m \begin{bmatrix} \mathbf{W}_1\widetilde{\mathbf{J}}_m \\ \mathbf{W}_2\mathbf{J}_s\widetilde{\mathbf{J}}_m \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \dot{\vec{\theta}}_{ad} \\ \vec{v}_{sd} \end{bmatrix} \quad (7)$$

Proof: Given $\vec{v}_{md} \in \mathbb{R}^{m_1}$, all solutions of (5) are given by

$$\dot{\vec{\theta}}_a = \mathbf{J}_m^+\vec{v}_{md} + \widetilde{\mathbf{J}}_m\vec{\zeta} \quad (8)$$

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

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

It is clear that (6) can be equivalently written as

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

Since $\begin{bmatrix} \mathbf{W}_1\widetilde{\mathbf{J}}_m \\ \mathbf{W}_2\mathbf{J}_s\widetilde{\mathbf{J}}_m \end{bmatrix}$ has full column-rank, (6) is minimized if and only if

$$\vec{\zeta} = -\begin{bmatrix} \mathbf{W}_1\widetilde{\mathbf{J}}_m \\ \mathbf{W}_2\mathbf{J}_s\widetilde{\mathbf{J}}_m \end{bmatrix}^+ \left\{ \begin{bmatrix} \mathbf{W}_1\mathbf{J}_m^+ \\ \mathbf{W}_2\mathbf{J}_s\mathbf{J}_m^+ \end{bmatrix} \vec{v}_{md} \right.$$
$$\left. - \begin{bmatrix} \mathbf{W}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_2 \end{bmatrix} \begin{bmatrix} \dot{\vec{\theta}}_{ad} \\ \vec{v}_{sd} \end{bmatrix} \right\}. \quad (9)$$

Substituting (9) into (8) gives (7). □

Theorem 2.4 *Let $\mathbf{W}_1$ be nonsingular, and $\mathbf{W}_2 = \mathbf{0}$. The unique solution for Problem 2.1 is*

$$\dot{\vec{\theta}}_a = \left[\mathbf{J}_m^+ - \widetilde{\mathbf{J}}_m(\mathbf{W}_1\widetilde{\mathbf{J}}_m)^+\mathbf{W}_1\mathbf{J}_m^+\right]\vec{v}_{md} +$$
$$\widetilde{\mathbf{J}}_m(\mathbf{W}_1\widetilde{\mathbf{J}}_m)^+\mathbf{W}_1\dot{\vec{\theta}}_{ad}. \quad (10)$$

Proof: It is similar to that of Theorem 2.3. □

Theorem 2.5 *Let $\mathbf{W}_1 = \mathbf{0}$, and $\mathbf{W}_2$ be nonsingular. The minimum norm solution for Problem 2.1 is*

$$\dot{\vec{\theta}}_a = \left[\mathbf{I} - \widetilde{\mathbf{J}}_m\widetilde{\mathbf{A}}(\widetilde{\mathbf{J}}_m\widetilde{\mathbf{A}})^+\right]\left[(\mathbf{J}_m^+ - \widetilde{\mathbf{J}}_m\mathbf{A}^+\mathbf{B})\vec{v}_{md} + \right.$$
$$\left.\widetilde{\mathbf{J}}_m\mathbf{A}^+\mathbf{W}_2\vec{v}_{sd}\right] \quad (11)$$

where $\mathbf{A} = \mathbf{W}_2\mathbf{J}_s\widetilde{\mathbf{J}}_m$, $\mathbf{B} = \mathbf{W}_2\mathbf{J}_s\mathbf{J}_m^+$.

Proof: It is similar to that of Theorem 2.3. □

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

Problem 2.6 *Consider the static force model (4). 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*

$$\vec{\tau} = \mathbf{J}_m^T\vec{f}_{md} + \mathbf{J}_s^T\vec{f}_s, \quad (12)$$

and $\|\mathbf{W}_3(\vec{\tau} - \vec{\tau}_d)\|_2^2 + \|\mathbf{W}_4(\vec{f}_s - \vec{f}_{sd})\|_2^2$ is minimized. $\mathbf{W}_3$ and $\mathbf{W}_4$ are weighting matrices.

The following theorems give solutions to Problem 2.6 for different weighting strategies.

Theorem 2.7 *Let $\mathbf{W}_3$ and $\mathbf{W}_4$ be nonsingular weighting matrices. The unique solution for Problem 2.6 is*

$$\vec{\tau} = \left\{ \mathbf{J}_m^T - \mathbf{J}_s^T \begin{bmatrix} \mathbf{W}_3\mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_3\mathbf{J}_m^T \\ \mathbf{0} \end{bmatrix} \right\} \vec{f}_{md} +$$
$$\mathbf{J}_s^T \begin{bmatrix} \mathbf{W}_3\mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \begin{bmatrix} \mathbf{W}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_4 \end{bmatrix} \begin{bmatrix} \vec{\tau}_d \\ \vec{f}_{sd} \end{bmatrix}. \quad (13)$$

Proof: From (12) 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

$$\vec{f}_s = -\begin{bmatrix} \mathbf{W}_3\mathbf{J}_s^T \\ \mathbf{W}_4 \end{bmatrix}^+ \left\{ \begin{bmatrix} \mathbf{W}_3\mathbf{J}_m^T \\ \mathbf{0} \end{bmatrix} \vec{f}_{md} \right.$$
$$\left. - \begin{bmatrix} \mathbf{W}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_4 \end{bmatrix} \begin{bmatrix} \vec{\tau}_d \\ \vec{f}_{sd} \end{bmatrix} \right\}. \quad (14)$$

Substituting (14) into (12) we get (13). □

Theorem 2.8 *Let $\mathbf{W}_3$ be nonsingular, and $\mathbf{W}_4 = \mathbf{0}$. The unique solution for Problem 2.6 is*

$$\vec{\tau} = \left[\mathbf{J}_m^T - \mathbf{J}_s^T(\mathbf{W}_3\mathbf{J}_s^T)^+\mathbf{W}_3\mathbf{J}_m^T\right]\vec{f}_{md} +$$
$$\mathbf{J}_s^T(\mathbf{W}_3\mathbf{J}_s^T)^+\mathbf{W}_3\vec{\tau}_d. \quad (15)$$

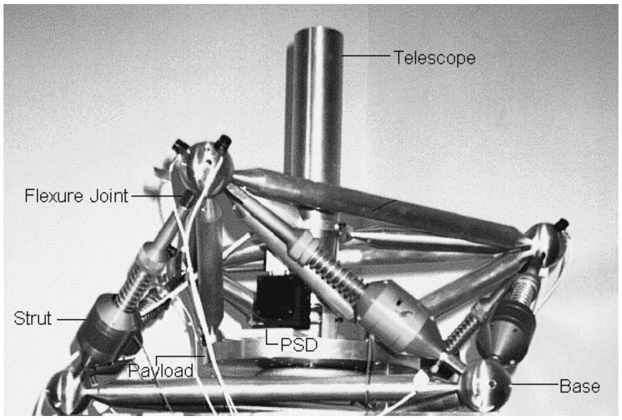

Figure 1: University of Wyoming (UW) flexure jointed hexapod

Proof: It is similar to that of Theorem 2.8. □

Theorem 2.9 *Let* $\mathbf{W}_3 = \mathbf{0}$, $\mathbf{W}_4$ *be nonsingular, and* $\mathbf{J}_s$ *has full row-rank. The unique solution for Problem 2.6 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 (12)). □

Remark 2.10 *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. Section 4 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, Section 4 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 [20] 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.*

3 Experiment Setup

The algorithms in Section 2 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 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 [10]: 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.

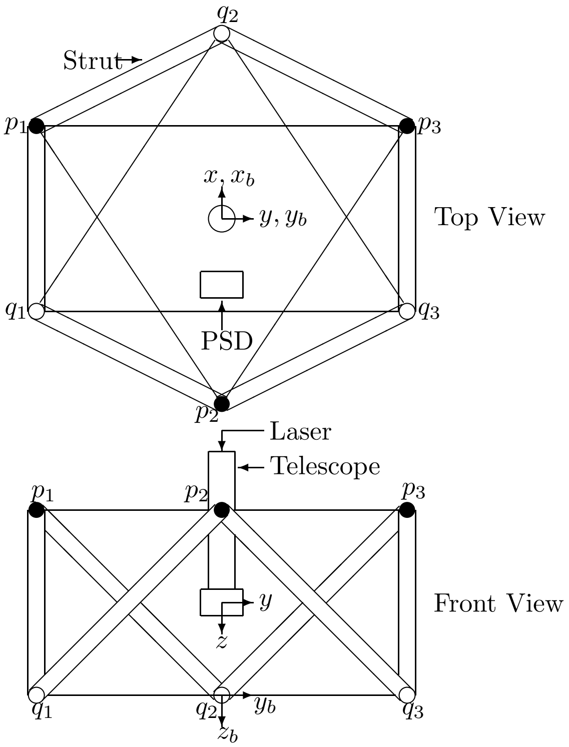

Figure 2: The top view and front view of the UW FJH. The telescope and position sensitive detector (PSD) sit on the payload, which is rigidly attached to the top nodes p_1, p_2, and p_3. For the sake of simplicity, the truss structure of the payload is not plotted in the figure. Bottom nodes q_1, q_2, 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.

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 (or reference) 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 (α and β) 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) 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 (α, β), and we want (α, β) 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 (α, β) tracks the given spiral signal (see [10, 11] 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

$$[\dot{\alpha},\, \dot{\beta},\, \dot{\gamma},\, \dot{t}_x,\, \dot{t}_y,\, \dot{t}_z]^T = \mathbf{J}[\dot{l}_1,\, \dot{l}_2,\, \dot{l}_3,\, \dot{l}_4,\, \dot{l}_5,\, \dot{l}_6]^T. \quad (16)$$

α, β, and γ 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. l_i is the length of leg i. Because 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. Numerical computation shows that the Frobenius norms of variations on $\mathbf{J}$ are more than three orders of magnitude weaker than that of $\mathbf{J}$. Moreover $\mathbf{J}$ is not ill-conditioned. Thus we can safely assume that $\mathbf{J}$ is constant across the workspace.

When pointing at a distant target, angular errors in the pointing angles (α, β) 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. γ and t_z have no effect on pointing performance. These SDOF motions are far less important, thus can be sacrificed for secondary goals. So the MDOF task space velocity $\vec{v}_m$ is defined as $\vec{v}_m = [\dot{\alpha}, \dot{\beta}]^T$, the SDOF task space velocity $\vec{v}_s$ is defined as $\vec{v}_s = [\dot{\gamma}, \dot{t}_x, \dot{t}_y, \dot{t}_z]^T$. $\mathbf{J}_m$ consists of the first two rows of $\mathbf{J}$.

However, the differential kinematics model (16) 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 (16), an approximate relationship between displacements can be derived as

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

where $\delta *$ represents the change in $*$. Although this 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 results in Section 2 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 task space and joint space coordinates) need to be small.

Figure 3 shows the block diagram of the hexapod control system for prioritized 2-DOF tracking. (α_d, β_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 $[\delta l_1, \delta l_2, \cdots, \delta l_6]^T$ is the vector of desired joint space displacements. This specific choice corresponds to pointing applications. $G_\alpha(s)$ and $G_\beta(s)$ are compensators for α and β channels, respectively. They are found using textbook single-input, single-output compensator designs [11]. Throughout the experiments, the compensators remain unchanged.

4 Experimental Results

Due to limited space, we can't include all the experimental results. So here we pick the secondary goal as *reliability enhancement*. Reliability has always been a major consideration for military, space, and some manufacturing applications. One way of enhancing the re-

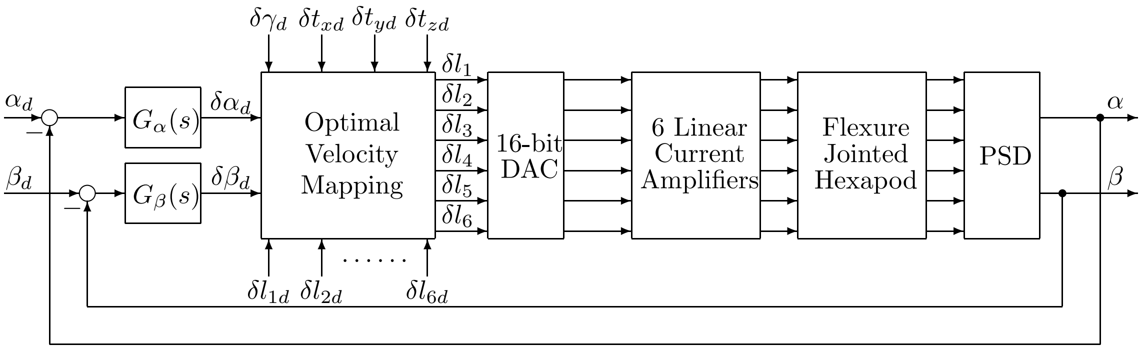

Figure 3: Block diagram for prioritized 2-DOF tracking control

liability 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 $\|\mathbf{W}_i([\delta l_1, \cdots, \delta l_6]^T - [\delta l_{1d}, \cdots, \delta l_{6d}]^T)\|_2^2$, $i \in \{a, b, c\}$. 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 $[\delta l_{1d}, \cdots, \delta l_{6d}]^T = \vec{0}$. The optimal velocity mappings are computed from (10) 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 [20] 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 1 (columns $2-4$) where α_{RMS} and β_{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 $100 \times \frac{\text{DAC input}}{\text{the maximum allowable DAC input}}$.

Table 1: Tracking errors for different weighting matrices.

Weighting Matrix	$\mathbf{W}_a$	$\mathbf{W}_b$	$\mathbf{W}_c$
α_{RMS} (micro-radians)	2.45	2.24	2.31
β_{RMS} (micro-radians)	3.09	3.27	3.25

5 Conclusions

In many applications some DOFs in the task 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. The velocity and static force mappings are derived, 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 proposed algorithms are tested on the UW FJH. Experimental results validate that the approach is practical and demonstrates good performance.

References

[1] J. Baillieul, "Avoiding obstacles and resolving redundancy," *Proc. IEEE Int'l Conf. 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 Trans. Robotics and Automation,* vol. 11, no. 2, pp. 286-292, 1995.

[3] F. Chaumette and E. Marchand, "A new

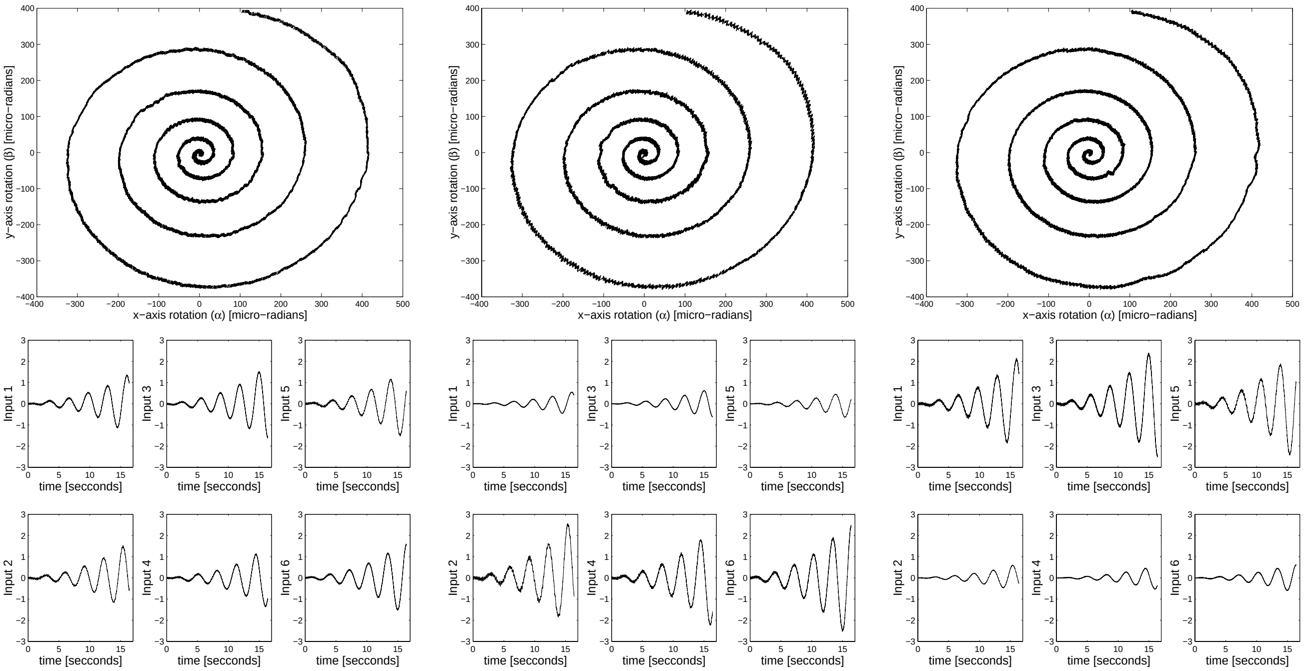

Figure 4: The tracking spirals and strut inputs when minimizing the weighted joint space displacements. The plots in the first and second rows are tracking spirals (PSD outputs) and the strut inputs, respectively. Here the Input i denotes the normalized actuator current for Strut i. From left to right,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

redundancy-based iterative scheme for avoiding joint limits application to visual servoing," *Proc. IEEE Int'l Conf. on Robotics and Automation,* pp. 1720-1725, 2000.

[4] J. J. Craig, *Introduction to Robotics: Mechanics and Control,* Reading, MA: Addison-Wesley, 1986.

[5] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverse applied to robotics," *The Int'l Journal of Robotics Research,* vol. 12, no. 1, pp. 1-19, 1993.

[6] 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.

[7] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *The Int'l Journal of Robotics Research,* vol. 6, no. 2, pp. 72-83, 1987.

[8] A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," *Proc. IEEE Int'l Conf. on Robotics and Automation,* pp. 638-642, 1990.

[9] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *The Int'l Journal of Robotics Research,* vol. 4, no. 3, pp. 1095-117, 1985.

[10] J. E. McInroy and J. C. Hamann, "Design and Control of Flexure Jointed Hexapods," *IEEE Trans. Robotics and Automation,* vol. 16, no. 4, pp. 372-381, 2000.

[11] 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. 24-31,37, 1999.

[12] J. E. McInroy, J. F. O'Brien, and G. W. Neat, "Precise, fault tolerant pointing using a Stewart platform," *IEEE/ASME Trans. Mechatronics,* vol. 4, pp. 91-95, 1999.

[13] J. E. McInroy and G. N. Saridis, "Techniques for selecting pose algorithms," *Automatica,* vol. 30, no. 3, pp. 471-487, 1994.

[14] 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.

[15] J. C. Musto, "Reliability-based inverse kinematics for redundant manipulators," *Int'l Journal of Intelligent Control and Systems,* vol. 3, no. 4, pp. 725-738, 1999.

[16] 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.

[17] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The Int'l Journal of Robotics Research,* vol. 6, no. 2, pp. 3-15, 1987.

[18] J. F. O'Brien and J. T. Wen, "Redundant actuation for improving kinematic manipulability," *Proc. IEEE Int'l Conf. on Robotics and Automation,* pp. 1520-1525, 1999.

[19] R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. Robotics and Automation,* vol. 12, pp. 543-552, 1996.

[20] H. Zhang and J. P. Ostrowski, "Visual Motion Planning for Mobile Robots," *IEEE Trans. Robotics and Automation,* vol. 18, no. 2, pp. 199-208, 2002.