Open access • Proceedings Article • DOI:10.1109/ROBOT.1993.291886

Task-space/joint-space damping transformations for passive redundant manipulators — Source link

Ambarish Goswami, Michael A. Peshkin

Institutions: Northwestern University

Published on: 02 May 1993 - International Conference on Robotics and Automation

Topics: Damping matrix, Matrix similarity, Jacobian matrix and determinant and Linear map

Task-space / Joint-space Damping Transformations for Passive Redundant Manipulators

Ambarish Goswami
Michael A. Peshkin

Department of Mechanical Engineering
Northwestern University
Evanston IL 60208

Abstract

We consider here passive mechanical wrists, capable of imparting a desired damping matrix to a grasped workpiece. Previous work [12, 13] has shown how to select a damping matrix such that an assembly operation can be made force-guided. The passive mechanical wrist is to be programmable – it can adopt a wide range of damping matrices – by virtue of a number of tunable dampers which interconnect the joints.

We have been studying the range of damping matrices that such a wrist can adopt, purely by tuning its dampers. We find that a redundant wrist has a broader range of realizable damping matrices than a non-redundant wrist.

A kinematic Jacobian relates the task-space damping matrix to a similar matrix in the hydraulic space of the tunable dampers (joint-space). For redundant wrists the transformation of damping matrices between task-space and joint-space is not straightforward. In this paper we identify the causal directions along which the transformations are linear. We show that the joint-space matrices which are obtained as linear transformations of desired task-space matrices are all singular. Many realizable joint-space matrices (corresponding to a desired task-space damping matrix) are shown to exist which are not discovered by linear transformations.

1.0 BACKGROUND AND MOTIVATION

The motion with which a robot responds to forces encountered during assembly may bring the workpiece closer to or farther from correct assembly. We have been studying methods of *designing* the accommodation (inverse damping) properties of a grasped workpiece, such that the forces arising during assembly always cause motions which move the workpiece closer to correct assembly. The design of an accommodation matrix for force-guided assembly was described in [13].

Implementing a suitable accommodation behavior is a form of force control [7]. Force control schemes in which the robot mimics a passive physical system are known to enjoy inherent advantages in interactive robotic tasks such as automated assembly. Colgate and Hogan showed that *only* a passive system remains stable at all frequencies when coupled to an arbitrary passive environment [3]. Robot controllers may emulate a passive system in order to take advantage of this fact [1, 11, 14].

Unfortunately, the speed of a software-controlled system is limited by the control system bandwidth [17]. This motivates the use of mechanical elements, such as springs, dampers etc., in order to implement force control.

One of the best known mechanical devices used for a class of assembly tasks is the *remote center of compliance* (RCC) device [16]. Work on the analysis and design of devices with desirable compliance, accommodation, or inertia properties suitable for different classes of interactive tasks are found in [2, 6, 9, 12, 13, 18].

A suitable force control law is task-specific. An accommodation matrix that works for a particular task is not necessarily useful (in fact it may be detrimental) for another. Therefore, robots must be able to adopt a broad range of accommodation matrices in order to perform a variety of tasks.

A disadvantage of mechanically implemented force control is the loss of simple software programmability. This motivates the need for mechanical elements with *programmable* parameters, e.g. spring stiffness, damping coefficient etc. For example, Cutkosky and Wright developed a programmable RCC wrist for introducing variable compliance in a robot [4].

We have been studying the range of accommodation matrices attainable by coupling the joints of a robot (or more practically of a wrist) via a passive network of programmable dampers (see Fig. 1). The network directly determines the wrist's *joint-space* accommodation matrix. This matrix describes the force-velocity relationship of the individual joints and does not involve their geometry or interconnections.

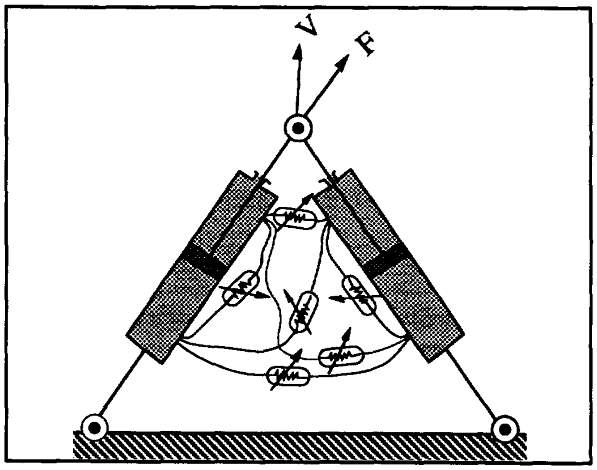

Figure 1. A simple parallel 2 DOF passive mechanism. The ports of the hydraulic cylinders are interconnected through constrictions with tunable damping.

The accommodation matrix of the workpiece as viewed by the environment is called the *task-space* accommodation matrix. This matrix relates the forces on and the velocities of a firmly grasped rigid workpiece. The task-space matrix is related to the joint-space matrix by the manipulator's Jacobian.

Our objective is to achieve a wide range of task-space accommodation matrices by programming the network of dampers that couple the joints. However we find that passive networks may adopt only a particular class of accommodation matrices [5]. We have proposed kinematic redundancy as a means of increasing the range of force control laws that may be implemented by a passive device.

2.0 OBJECTIVE AND SUMMARY

In this paper, we study the relationship between accommodation (or damping) matrices in joint-space and task-space for passive redundant manipulators. Our analysis can be immediately applied to networks of springs (imparting a compliance) or of masses (imparting an inertia matrix).

Just as we use a manipulator's Jacobian matrix to transform forces and velocities between its joint-space and task-space, we can imagine similar transformations between the spaces for its accommodation or damping matrices.

By analogy to the term "forward kinematics," the computation of the task-space accommodation matrix from a given joint-space accommodation matrix will be called the *forward transformation* problem. The problem of determining the joint-space accommodation matrix from a desired task-space matrix will be called the *inverse transformation* problem.

The inverse transformation problem is relevant when, as described above, a desired accommodation matrix is specified in task-space in order to make an assembly operation force-guided. The desired matrix is transformed to the robot's joint space. To implement the resulting joint-space accommodation matrix one still has to program the network appropriately as described in [5].

The tasks-space accommodation (or damping) matrices of a manipulator are related to their joint-space counterparts through a congruence transformation. This is a linear transformation involving the manipulator's Jacobian and is sensitive to the manipulator's pose. For non-redundant manipulators in non-singular poses, the forward and inverse transformations are simple one-to-one mappings between joint-space and task-space.

For redundant manipulators, however, the transformations are not always straightforward. Kinematic redundancy imposes constraints on joint-space velocities (in parallel manipulators) or forces (in serial manipulators). These constraints give rise to preferred causal directions along which linear transformations of accommodation and damping matrices may take place. The causal directions depend on the structure of the manipulator (serial or parallel) as well as on the type of matrix being transformed (accommodation or damping).

For example, in a parallel manipulator, an accommodation matrix maps linearly from task-space to joint-space but not in the reverse direction. A damping matrix, on the other hand, maps linearly from joint-space to task-space. Dual results exist for serial manipulators.

For redundant manipulators some of the transformations are many-to-one. For instance, for a serial redundant manipulator, many joint-space accommodation matrices map to a single task-space matrix. To implement a desired task-space matrix, one has a choice of many joint-space matrices, and hopefully some of them are realizable by a passive network of dampers.

Unfortunately the causal linear transformations do not directly identify all of the corresponding matrices in the case of many-to-one transformations. As an example, the inverse transformation (which is a linear congruence transformation) of a desired task-space accommodation matrix for a parallel manipulator yields only one matrix. However, infinitely many joint-space matrices exist that also correspond to the given task-space matrix. In order to take full advantage of redundancy, one must therefore look beyond the linear transformation. In the next section we give a simple physical example to point out some of the important characteristics exhibited by redundant passive mechanisms. Section 4.0 discusses the nature of force and velocity transformation between joint-space and task-space of redundant manipulators. An understanding of force and velocity transformation is important for identifying the causal directions in which accommodations and damping matrices transform. We discuss the latter in Section 5.0. Finally, in Section 6.0 we apply our results to passive force control.

3.0 AN EXAMPLE

Fig. 2 shows a parallel arrangement of two hydraulic cylinders that are connected to a massless cart. The cylinders have damping coefficients of d_1 and d_2. This mechanism may be thought of as a redundant parallel manipulator with a 1- DOF task-space and a 2-DOF joint-space. The joint-space damping matrix D_j is a 2×2 diagonal matrix[1] with d_1 and d_2 as the diagonal elements. The task-space damping matrix D_t (a scalar here) is the apparent damping of the cart as seen by the environment, which we know to be $(d_1 + d_2)$.

Although this example may appear a bit trivial, it exhibits some important characteristics of redundant mechanisms. For instance, we can observe many-to-one mapping of damping matrices from joint-space to task-space in this manipulator as many combinations of d_1 and d_2 give the same D_t .

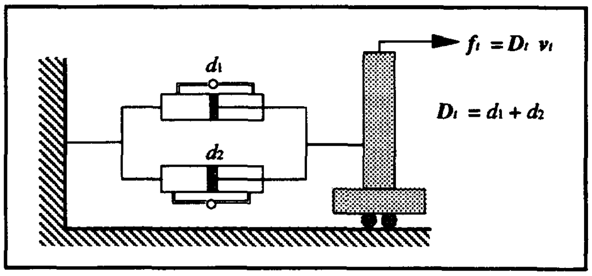

Figure 2. A simple parallel mechanism to illustrate the basic features of redundant parallel manipulators.

4.0 REVIEW OF VELOCITY AND FORCE TRANSFORMATIONS

4.1 Velocity Transformation for Parallel Manipulators

The velocity transformation relationship $v_t \rightarrow v_j$ for a parallel manipulator is expressed as:

$$J\, v_t = v_j \qquad (1)$$

where v_t is an $(m\times 1)$ task-space velocity vector and v_j is the corresponding $(n\times 1)$ joint-space velocity vector. m and n are the degrees of freedom of the task-space and the joint-space respectively. For a redundant manipulator $n>m$ and J is an $(n\times m)$ Jacobian matrix transforming a task-space velocity to a joint-space velocity.

For a redundant parallel manipulator there is one and only one joint-space velocity corresponding to a given task-space velocity. If the velocities of any m independent joints are known, the velocities of rest of the n-m joints are uniquely determined. The left nullspace of J correspond to those

[1] Cross-coupling of the hydraulic cylinders (not shown in the picture) gives rise to the off-diagonal terms in the damping matrix, see [5]

velocities which are physically impossible. The pseudo-inverse of the Jacobian $J^+ = (J^T J)^{-1} J^T$ may not be used to obtain a v_t for a given v_j unless one restricts the set of joint-space velocities to be the physically possible ones.

For the manipulator in Fig. 2, the Jacobian $J = [1 \;\; 1]^T$. The dampers must always have equal velocity. The left nullspace of J corresponds to physically impossible unequal damper velocities.

4.2 Force Transformation for Parallel Manipulators

In our ideal lossless mechanism, the virtual power in joint-space and task-space are equal. The force transformation $f_j \rightarrow f_t$ is dual to the velocity transformation expression of (1):

$$J^T f_j = f_t \quad , \tag{2}$$

where f_j is an $(n\times 1)$ joint-space force vector, f_t is the corresponding $(m\times 1)$ task-space force vector, and J^T is the Jacobian transpose.

Equation (2) maps an n-dimensional space (of f_j) to a smaller m-dimensional space (of f_t) in a many-to-one fashion. The non-uniqueness of the joint-space forces corresponds to the existence of the $(n\text{-}m)$-dimensional null-space of J^T. Joint-space forces lying entirely in the nullspace of J^T are not manifested in the task-space. These are the internal forces in a mechanical system.

Although many joint-space forces may result in a single task-space force, a force applied in the task space of a physical system results in a unique joint-space force. The pseudo-inverse $(J^T)^+$ cannot correctly depict this $f_t \rightarrow f_j$ mapping. Although a mapping through the pseudo-inverse produces a unique f_j for a given f_t, the mapping is based on an ad-hoc mathematical assumption that nullspace forces are zero, which is not necessarily satisfied by redundant manipulators.

For the redundant manipulator in Fig. 2, we may write out (2) as, $f_{j1} + f_{j2} = f_t$, where f_{j1} and f_{j2} are the forces in the two dampers. An unequal joint-space force generated in this mechanism (which happens whenever $d_1 \neq d_2$) corresponds to a non-zero nullspace component. Given only f_t, it is impossible to predict the joint-space forces from (2). However, if we know the accommodation properties of the manipulator, we can easily compute $f_{j1} = d_1\,(d_1 + d_2)^{-1} f_t$, and $f_{j2} = d_2(d_1 + d_2)^{-1} f_t$. This we do by using the velocity constraint ($v_{j1} = v_{j2}$) and the task-space accommodation matrix $(d_1 + d_2)^{-1}$. In Section 5.1 we generalize this method to obtain a physically meaningful one-to-one $f_t \rightarrow f_j$ mapping.

4.3 Serial Manipulators

The velocity and force transformation relationships in serial manipulators are dual to those in parallel manipulators. For serial manipulators the transformation equations are,

$$v_t = J\, v_j \qquad \text{and} \qquad f_j = J^T f_t\,, \tag{3}$$

where J is an $(m\times n)$ Jacobian matrix. The mappings of Eq. (4) are dual to those in Equations (1) and (2). Note that the Jacobian matrix J in a serial manipulator transforms a joint-space velocity to a task-space velocity. In a parallel manipulator J represents the inverse transformation.

Serial redundant manipulators may have infinite number of joint-space velocities, all mapping onto a single task-space velocity. Joint-space velocities lying in the nullspace of J are not manifested in the task-space. There are joint-space force constraints in serial redundant manipulators as opposed to joint-space velocity constraints in parallel manipulators. The left nullspace of J^T (in Eq. 3 above) corresponds to the physically impossible forces which would require infinite joint velocities.

5.0 TRANSFORMATIONS OF ACCOMMODATION AND DAMPING MATRICES

In this section we use the causal transformations equations of force and velocity (Equations 1, 2, and 3) to obtain the causal relationships of accommodation and damping matrices between task-space and joint space of redundant manipulators. Our discussion in this section will mainly relate to parallel manipulators.

5.1 Force-Velocity Cycle

We first start with the two causal kinematic transformations J: $v_t \rightarrow v_j$ and J^T: $f_j \rightarrow f_t$ as seen in (1) and (2), respectively. We have two other transformations, namely the task-space accommodation matrix $A_t : f_t \rightarrow v_t$ and the task-space damping matrix $D_j : v_j \rightarrow f_j$. These four transformations may be simultaneously illustrated with the help of a force-velocity cycle diagram, as shown in Fig. 3. Mussa-Ivaldi et al. described similar causality cycles in biological networks as K-nets [10]. Kim et al. described similar cycles (and named them premultiplier diagrams) to derive optimal control strategies for redundant manipulators [8].

There are two types of parameters in a force-velocity cycle. The first type corresponds to a force vector or a velocity vector. This includes f_j, f_t, v_j, and v_t. The second type of parameters includes the matrices J, J^T, A_j, D_j, A_t, and D_t. Each of these matrices represents a transformation, and is therefore associated with an input vector and an output vector. The vector parameters are located at the nodes of the cycle whereas a matrix is positioned on the directed arc connecting the input and the output vectors. The direction of the arc specifies the causality of the mapping. Notice that for non-singular damping and accommodation matrices, the relationships $f_t \leftrightarrow v_t$ and $f_j \leftrightarrow v_j$ are bi-directional. The relationships $v_t \rightarrow v_j$ and $f_j \rightarrow f_t$ are, on the other hand, unidirectional for redundant parallel manipulators. This is because the inversion of the non-square Jacobian (or its transpose) is not always physically meaningful

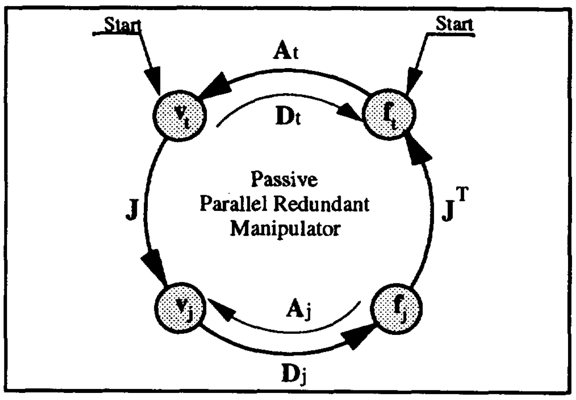

Figure 3. The force-velocity cycle for a passive parallel redundant manipulator. The cycle may start from task-space velocity or task-space force only.

as we have observed in the last section. These unidirectional branches are responsible for specifying the general direction of the force-velocity cycle.

Since passive systems consist of unpowered joints, they may only respond to forces or velocities imparted in the task-space. Thus, there are only two starting nodes in the force-velocity cycle of passive manipulators (see Fig. 3), those corresponding to $\boldsymbol{v}_t$ and $\boldsymbol{f}_t$.

Now consider the question: What joint-space force $\boldsymbol{f}_j$ results from a given task-space velocity $\boldsymbol{v}_t$? We are looking for the $\boldsymbol{v}_t \rightarrow \boldsymbol{f}_j$ mapping here. The general rule is to start from the given input vector $\boldsymbol{v}_t$, travel in the causal direction (shown by the arrowheads) of the force-velocity cycle, get pre-multiplied by the encountered matrices until the output vector $\boldsymbol{f}_j$ is reached. Following this rule we obtain the desired relationship $\boldsymbol{f}_j = (\boldsymbol{D}_j\ \boldsymbol{J})\, \boldsymbol{v}_t$.

Although any two vectors may be related this way, only those relationships which represent mapping from one of the starting vectors are physically meaningful for a passive manipulator. For instance, one may express the mapping $\boldsymbol{v}_j \rightarrow \boldsymbol{v}_t$ as $\boldsymbol{v}_t = (\boldsymbol{A}_t \boldsymbol{J}^T \boldsymbol{D}_j)\, \boldsymbol{v}_j$. Since joint-space velocities are generated only as consequences of task-space velocities, and not the other way around, $\boldsymbol{v}_j \rightarrow \boldsymbol{v}_t$ is not a meaningful mapping for passive manipulators.

The force-velocity cycle also helps in answering a typical matrix transformation question such as the following: How to compute the task-space damping matrix $\boldsymbol{D}_t$ for a given joint-space damping matrix $\boldsymbol{D}_j$? The representative mapping here is $\boldsymbol{D}_j \rightarrow \boldsymbol{D}_t$. From the force-velocity cycle of Fig. 3, we see that $\boldsymbol{D}_t$ represents the mapping $\boldsymbol{v}_t \rightarrow \boldsymbol{f}_t$. This mapping may be represented in an alternative way if we follow the route $\boldsymbol{v}_t \rightarrow \boldsymbol{v}_j \rightarrow \boldsymbol{f}_j \rightarrow \boldsymbol{f}_t$. One may obtain this alternative relationship as,

$$\boldsymbol{f}_t = (\boldsymbol{J}^T \boldsymbol{D}_j\ \boldsymbol{J})\boldsymbol{v}_t \quad . \tag{4}$$

Comparing (4) with the relationship $\boldsymbol{f}_t = \boldsymbol{D}_t\, \boldsymbol{v}_t$, we may derive the well-known expression $\boldsymbol{D}_t = \boldsymbol{J}^T \boldsymbol{D}_j\ \boldsymbol{J}$, which linearly maps a joint-space damping matrix to the corresponding task-space damping matrix.

We complete this discussion by illustrating the force-velocity diagram for serial redundant manipulators. As shown in Fig. 4 this diagram is dual to that for parallel manipulators. It is interesting to note that the general causality direction of Fig. 4 is clockwise, which is opposite to that for parallel redundant manipulators (Fig. 3).

5.2 Linear Directions of Matrix Mapping

There are four different mappings of accommodation and damping matrices in a manipulator. These are $\boldsymbol{D}_j \rightarrow \boldsymbol{D}_t$, $\boldsymbol{A}_t \rightarrow \boldsymbol{A}_j$, $\boldsymbol{A}_j \rightarrow \boldsymbol{A}_t$, and $\boldsymbol{D}_t \rightarrow \boldsymbol{D}_j$. We will show that for parallel redundant manipulators only the first two mappings are linear whereas for serial manipulators only the last two are linear.

From the force-velocity cycles we can see that only if there are two different valid routes between two adjacent vectors, we get a linear relationship between the associated matrices. For example, the mapping $\boldsymbol{D}_j \rightarrow \boldsymbol{D}_t$ was derived (in the last section) by observing that there are two different routes from $\boldsymbol{v}_t$ to $\boldsymbol{f}_t$. One may obtain an $\boldsymbol{A}_t \rightarrow \boldsymbol{A}_j$ mapping by exploiting two different routes between $\boldsymbol{f}_j$ and $\boldsymbol{v}_j$. The first route is direct and it involves $\boldsymbol{A}_j$ only (see Fig. 3). The other route is along $\boldsymbol{f}_j \rightarrow \boldsymbol{f}_t \rightarrow \boldsymbol{v}_t \rightarrow \boldsymbol{v}_j$. From Fig. 3 we get $\boldsymbol{A}_j = \boldsymbol{J}\, \boldsymbol{A}_t\, \boldsymbol{J}^T$.

One can check that for a parallel manipulator it is impossible to obtain the other two linear mappings, $\boldsymbol{A}_j \rightarrow \boldsymbol{A}_t$ and $\boldsymbol{D}_t \rightarrow \boldsymbol{D}_j$,

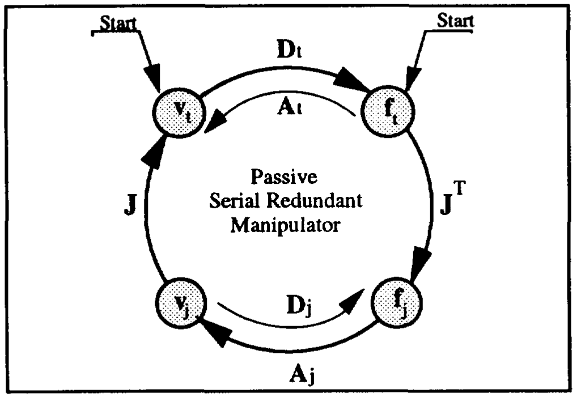

Figure 4. The force-velocity cycle for a passive serial redundant manipulator. The cycle may commence from task-space velocity or task-space force only.

without inverting the Jacobian. We will show that inversion of $\boldsymbol{J}$ or $\boldsymbol{J}^T$ produces physically meaningless results.

For the parallel manipulator in Fig. 2, we can obtain the linear relationships ,

$$D_t = d_1 + d_2 \qquad \text{and} \qquad A_j = \begin{bmatrix} a_t & a_t \\ a_t & a_t \end{bmatrix}, \tag{5}$$

where $\boldsymbol{A}_j$ and a_t are the joint-space and task-space accommodation matrices respectively. a_t is a scalar in this case.

As another example, for a given $A_j = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$, where $a_1 = 1/d_1$ and $a_2 = 1/d_2$, we can compute $\boldsymbol{A}_t = \boldsymbol{D}_t^{-1} = (\boldsymbol{J}^T \boldsymbol{D}_j\ \mathbf{J})^{-1} = (1/a_1 + 1/a_2)^{-1}$, an $\boldsymbol{A}_j \rightarrow \boldsymbol{A}_t$ relationship that is nonlinear.

If we invert $\boldsymbol{A}_j = \boldsymbol{J}\, \boldsymbol{A}_t\, \boldsymbol{J}^T$ and use $\boldsymbol{A}_t = \boldsymbol{J}^+ \boldsymbol{A}_j\, (\boldsymbol{J}^T)^+$, this would produce wrong results. This is because it assumes a linear $\boldsymbol{A}_j \rightarrow \boldsymbol{A}_t$ relationship where no such relationship really exists. Also the process of pseudo-inversion of Jacobian matrices looses the information about task-space velocity constraints. We compute $\boldsymbol{J}^+ = [0.5\ \ 0.5]$ and $(\boldsymbol{J}^T)^+ = (\boldsymbol{J}^+)^T$, and calculate $\boldsymbol{A}_t = \boldsymbol{J}^+ \boldsymbol{A}_j (\boldsymbol{J}^T)^+ = 0.25(a_1 + a_2)$. One can easily check that this result is incorrect.

For serial manipulators, we have results which follow from Fig. 4: $\boldsymbol{A}_t = \boldsymbol{J}\ \ \boldsymbol{A}_j\ \ \boldsymbol{J}^T$ and $\boldsymbol{D}_j = \boldsymbol{J}^T\ \ \boldsymbol{D}_t\ \ \boldsymbol{J}$. These are the only linear relationships between accommodation and damping matrices in a serial redundant manipulator. One should remember that the Jacobian $\boldsymbol{J}$ in serial manipulators represents the $\boldsymbol{v}_j \rightarrow \boldsymbol{v}_t$ mapping.

Notice that for non-redundant manipulators, serial or parallel, one may obtain all of the four transformations $\boldsymbol{D}_j \rightarrow \boldsymbol{D}_t$, $\boldsymbol{A}_t \rightarrow \boldsymbol{A}_j$, $\boldsymbol{A}_j \rightarrow \boldsymbol{A}_t$, and $\boldsymbol{D}_t \rightarrow \boldsymbol{D}_j$.

5.3 Flexibility in Programming a Task-Space Matrix

Flexibility in programming a task-space matrix in a redundant manipulator results from the choice one can have in selecting a joint-space force (in parallel) or a joint-space velocity (in serial) for desired task-space counterparts. In Fig. 5 we show how the existence of nullspace forces in a parallel manipulator gives rise to a set of joint-space accommodation matrices, all of which map to a single task-space accommodation matrix.

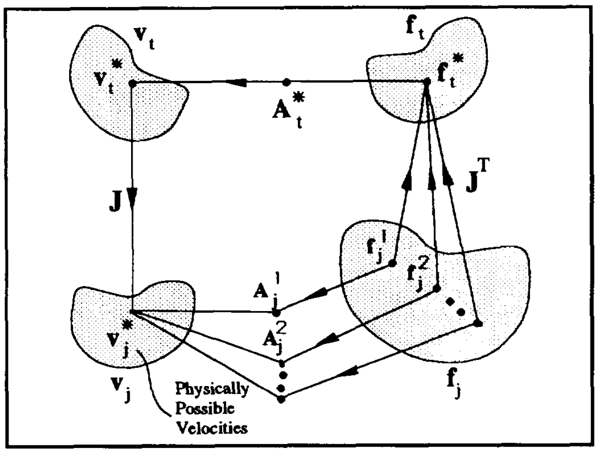

Figure 5. An abstract illustration depicting the redundancy of accommodation matrices in a parallel manipulator. An infinite number of combinations of A_j is equivalent to a specified A_t.

Consider the effect of a particular task-space accommodation matrix A_t^* on a particular task-space force f_t^*. The transformation of this force takes place according to $v_t = A_t^* f_t^*$. This task-space velocity corresponds to a unique joint-space velocity $v_t^* = J\, v_t^*$. We know that there are an infinite number of joint-space forces f_j^k, each of which maps to f_t^* under the transformation of J^T (see Section 4.2). These forces are composed of a unique row space component of J^T and arbitrary nullspace components. From Fig. 5, we can infer that any A_j^k that maps one of the f_j^k to v_t^* is equivalent to A_t^*. Consequently we have an infinite number of joint-space accommodation matrices all of which are equivalent to A_t^*. All different joint-space forces that are resulted from a given f_t^* by modifying the accommodation/damping characteristics in a fixed geometry mechanism, have a unique row space component given by $(J^T)^+ f_j = f_t^*$. This is true since all the f_j must satisfy (2) which is a consequence of invariance of power in joint-space and task-space.

For a serial manipulator the situation is reversed. For a given f_t^* and v_t^* we will have a unique f_t^* and an infinite number of combinations of v_j.

6.0 APPLICATION

Recall (from Section I) that our main interest is the implementation of passive force control laws by a low-inertia unpowered mechanical wrist. A network of programmable passive dampers interconnects the joints of this wrist. The network of dampers is equivalent to a network of resistors in the electrical domain. From electrical network theory, we learn that there is a class of conductance matrices (analogous to accommodation matrices) for which the resistor values of the network may be computed algorithmically. These matrices are called *dominant* matrices. The diagonal elements of dominant matrices are greater than or equal to the sum of the absolute values of all other elements in the same row (or column) [15]. Therefore for any desired dominant accommodation matrix in the joint-space of a manipulator, one can compute the resistor values in algorithmically.

For a specified task-space accommodation matrix, a set of equivalent joint-space matrices exist in a redundant manipulator, although not all of them are accessible by the linear transformation of the task-space matrices. In fact, we show that joint-space matrices that are images of task-space matrices under the linear transformation $A_j = J\, A_t\, J^T$ are all singular matrices of rank m, where m is task-space DOF. This is due to the fact that A_t is of rank m, and no linear congruence transformation on it (here, with full rank Jacobians) may change the rank of the resulting matrix. Fig. 6 illustrates the

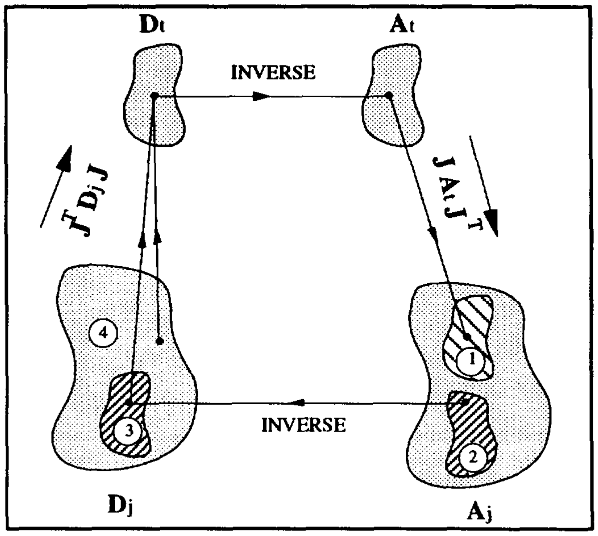

Figure 6. An illustration of the fact that the image of task-space accommodation matrices under linear mapping consists of singular joint-space accommodation matrices only. Area ① corresponds to A_j matrices of rank m. Area ② corresponds to A_j matrices of rank n. Area ③ corresponds to D_j matrices of rank n. Area ④ corresponds to D_j matrices of rank less than n. m and n are the task-space DOF and the joint-space DOF, respectively.

transformation of accommodation and damping matrices in redundant parallel manipulators. We explain this figure with the help of our simple parallel manipulator of Fig. 2. As seen in (5) every A_j which are images of task-space matrices under linear transformation are singular (they have equal elements). This is a marginally dominant matrix and is implementable. These matrices belong to area ① in Fig. 6. Any matrix of the form $A_j = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix}$ where the specified task-space matrix $A_t = (1/a_1 + 1/a_2)^{-1}$ belongs to area ② . There are other non-diagonal non-singular accommodation matrices which falls in the area ② . For the specified A_j above, area ③ will contain damping matrices of the form $D_j = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}$. Again, there will be other non-diagonal non-singular damping matrices in the area ③. For a given $A_t = D_t^{-1}$, a typical matrix of area ④ is of the form $D_j = \begin{bmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{bmatrix}$ where $D_t = d_{11} + 2d_{11} + d_{22}$ and $d_{11}d_{22} - d_{12}^2 = 0$ (to render D_t singular).

We give a simple example to demonstrate that for a given A_t equivalent dominant matrices may exist beyond area ①. This would justify the search for appropriate joint-space accommodation matrices which are not attainable by linear transformation of $A_j = J A_t J^T$. Imagine a parallel redundant mechanism with three hydraulic cylinders arranged in the same pattern as in Fig. 2. The Jacobian matrix for the mechanism is $J = [1\ 1\ 1]^T$. Now, for a given $a_t = 3$, the joint-space accommodation matrix according to (8) is $A_j = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$.

This resulting A_j is not a dominant matrix, and therefore no systematic way for its implementation exists. One can nevertheless find out a dominant $A_j = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 9 \end{bmatrix}$ which may be easily implemented. This latter matrix belongs to the area ② and it is not linearly related to any task-space matrix. We may, however, follow the transformation: $A_j \rightarrow D_j \rightarrow d_t \rightarrow a_t$ to verify that the latter A_j indeed corresponds to the given a_t.

7.0 CONCLUSIONS

This paper investigated how accommodation and damping matrices transform between task-space and joint-space of redundant manipulators.

We found that in parallel redundant manipulators infinitely many joint-space forces may map onto a single task-space force. The joint-space velocities are, however, subjected to constraints, the violation of which would require mechanical deformation of the manipulator. In serial redundant manipulators, on the other hand, infinitely many joint-space velocities may map onto a single task-space velocity. The joint-space forces, in this case, are subjected to constraints, the violation of which would require infinite joint velocities.

We have found that redundant manipulators may exhibit an increased range of task-space accommodation and damping matrices. This makes them valuable for the implementation of force control.

Redundancy dictates the causal directions in which accommodation and damping matrices may linearly transform between the joint-space and the task-space of a manipulator. These causal directions result from inherent constraints on joint-space velocity (in parallel manipulators) and force (in serial manipulators) imposed by redundancy.

Joint-space matrices are mapped in a many-to-one fashion to the task-space matrices. This provides a manipulator the flexibility to choose from a set of joint-space matrices in order to achieve a specified task-space matrix. However this is complicated by the fact that this mapping is not always linear. Also, the linear transformation that maps task-space matrices to joint-space matrices result in singular joint-space matrices only. Therefore one has to look beyond these linear mappings in order to exploit the full potential offered by kinematic redundancy.

ACKNOWLEDGMENTS

The authors wish to acknowledge useful discussion with Ed Colgate and Joe Schimmels. This work was supported by Northwestern University and NSF grant DMC-8857854.

REFERENCES

1. Anderson, R. J. *Dynamic Damping Control: Implementation Issues and Simulation Results.* Proceedings of the 1988 IEEE International Conference on Robotics and Automation. Cincinnati, OH (68-77) IEEE Press (1990)

2. Asada, H. and Y. Kakumoto. *The Dynamic RCC Hand for High-Speed Assembly.* Proceedings of the 1988 IEEE International Conference on Robotics and Automation. Philadelphia (120-) IEEE Press (1988)

3. Colgate, J. E. and N. Hogan. *Robust Control of Dynamically Interacting Systems.* International Journal of Control **48**(1):65-88 (1988)

4. Cutkosky, M. R. and P. K. Wright. *Active Control of a Compliant Wrist in Manufacturing Tasks.* Journal of Engineering for Industry ; Transactions of the ASME **108**(February):36-43 (1986)

5. Goswami, A., M. Peshkin and J. E. Colgate. *Passive Robotics: An Exploration of Mechanical Computation.* IEEE International Conference on Robotics and Automation. Cincinnati IEEE Press (1990)

6. Griffis, M. and J. Duffy. *Kinestatic Control: A Novel Theory for Simultaneously Regulating Force and Displacement.* The 1990 ASME Design Technical Conferences. Chicago (287-295) The American Society of Mechanical Engineer (1990)

7. Hogan, N. *Impedance Control: An Approach to Manipulation.* ASME Journal of Dynamic Systems, Measurement, and Control **107**:1-24 (1985)

8. Kim, J.-O., P. Khosla and W.-K. Chung. *Static Modeling and Control of Redundant Manipulators.* Robotics and Computer Integrated Manufacturing Vol. 9(No. 2):pp. 145-157 (1992)

9. Loncaric, J. *Elastic Systems and Quasistatic Grasp Stability.* Proceedings IEEE International Conference on Robotics and Automation. Sacramento IEEE Press (1991)

10. Mussa-Ivaldi, F. A., P. Morasso and R. Zaccaria. *Kinematic Networks: A Distributed Model for Representing and Regulating Motor Redundancy.* Biological Cybernetics **60**:1-16 (1988)

11. Newman, W. S. and M. E. Dohring. *Augmented Impedance Control: An Approach to Compliant Control of Kinematically Redundant Manipulators.* Proceedings of IEEE International Conference on Robotics and Automation. Sacramento, CA (30-35) IEEE Press (1991)

12. Peshkin, M. A. *Programmed Compliance for Error-Corrective Assembly.* IEEE Transactions on Robotics and Automation **6**(4):473-482 (1990)

13. Schimmels, J. M. and M. A. Peshkin. *Admittance Matrix Design for Force Guided Assembly.* IEEE Transactions on Robotics and Automation Vol. **8**(No. 2):213-227 (1992)

14. Wang, D. and M. Vidyasagar. *Passive Control of a Single Flexible Link.* Proceedings of the 1988 IEEE International Conference on Robotics and Automation. Cincinnati, OH (1432-1437) IEEE Press (1990)

15. Weinberg, L. *Network Analysis and Synthesis.* McGraw-Hill Electrical and Electronic Engineering Series. Terman ed. McGraw-Hill Book Company, Inc. New York (1962)

16. Whitney, D. E. *Quasi-Static Assembly of Compliantly Supported Rigid Parts.* ASME J. Dynamic Systems, Measurement, and Control. **104**:65-77 (1982)

17. Whitney, D. E. *Historical Perspective and State of the Art in Robot Force Control.* International Journal of Robotics Research **6**(1):3-14 (1987)

18. Yi, Byung-Ju, R. A. Freeman, D. Tesar. *Open-Loop Stiffness Control of Overconstrained Mechanisms/Robotic Linkage Systems.* 1989 IEEE International Conference on Robotics and Automation. Scottsdale, AZ IEEE Press (1989)