To appear in the IEEE Transactions on Robotics and Automation
Winter, 1999.

On the Use of Scaling Matrices for Task-Specific Robot Design

Leo J. Stocco, S. E. Salcudean and F. Sassani

Abstract—**Good robot performance often relies upon the selection of design parameters that lead to a well conditioned Jacobian or impedance "design" matrix. In this paper, a new design matrix normalization technique is presented to handle the problem of non-homogeneous physical units and to provide a means of specifying a performance based design goal. The technique pre- and post-multiplies a design matrix by scaling matrices corresponding to a range of joint and task-space variables. The task-space scale factors are used to set relative required strength or speed along any axes of end-point motion while the joint-space scale factors are treated as free design parameters to improve isotropy through non-homogeneous actuation. The effect of scaling on actual designs is illustrated by a number of design examples using a global search method previously developed by the authors.**

I. INTRODUCTION

Many robot design variables such as structure (serial vs. parallel), geometry, actuators (rotary vs. prismatic) and reduction ratios contribute to the way a robot behaves. Unfortunately, any change that enhances one performance attribute will almost always detract from another. Stiffness, for example, is improved by using a parallel robot instead of a serial robot but workspace size suffers. This trade-off that occurs with virtually every design variable suggests that a universally optimum device does not exist. Optimality only exists in the context of a specific application since different applications make different performance demands. This paper describes how a robot can be designed for a particular application by integrating application specific performance requirements into the performance function. It shows how to specify desired relative capabilities with respect to individual workspace dimensions and how to improve the solution through non-homogeneous actuation. The technique also normalizes physical units to ensure a meaningful result.

Methods for handling non-uniform workspace dimensions have been suggested by Gosselin [6], Tandirci *et al* [14], Angeles [1], Ma and Angeles [10], Angeles *et al* [2] and Doty *et al* [4]. They address the problem pointed out by Lipkin and Duffy [9] that a measure such as the condition number of the Jacobian matrix is of little practical significance in the presence of non-uniform physical units. This occurs when a robot can both translate and rotate its end-effector or when it contains both rotary and prismatic actuators. To accommodate this, Gosselin [6] defines a new Jacobian that transforms actuator velocities into the linear velocities of two points on the end-effector. He does not, however, indicate how one should choose these points. Tandirci *et al* [14] normalize the Jacobian by dividing a "Characteristic Length" (CL) out of all translational elements. The CL that produces the best performance measure is dubbed the "Natural Length" (NL) by Ma and Angeles [10] and is used for design optimization. When the NL of a platform manipulator is not derivable, it is approximated by the average platform radius. Angeles [1] calculates the NL for a serial manipulator by averaging the distances between the operating point and all active joint axes while Angeles *et al* [2] find a serial manipulator's NL by making it a free design parameter. Doty *et al* [4] propose a method of inverting non-square matrices with mixed physical units so that the solution is both unit and frame invariant. The method achieves physical unit consistency but does not differentiate between quantities with similar units but dissimilar magnitudes.

Although much past work (i.e. Doty *et al* [4]) suggests that scaling matrices possess an arbitrary quality when used to address the unit inconsistency problem, it is argued here that scaling is not arbitrary if the intended use of the device is not arbitrary. It is shown here that when a specific performance goal exists, the choice of scale factors greatly affects the performance measure and, if chosen properly, can result in a drastic improvement in performance. The scaling matrices proposed here remove all physical units and enable one to specify the desired performance of a device and solve for its optimum actuator sizes. This method is the first of its kind to simultaneously consider both geometric and actuator parameters in robot design optimization. It is demonstrated through the use of a relevant, workspace inclusive, global performance measure and an optimization algorithm previously developed by the authors to make meaningful comparisons of different robot devices.

Section II of this paper discusses the definition of global (i.e. workspace inclusive) isotropy and the optimization algorithm that is used in all design examples. Section III describes the proposed scaling matrices that are the focus of this paper. The task-space scaling matrix and its effect on robot design is described in Section IV, while the joint-space scaling matrix is described in Section V with a summary and conclusions in Section VI.

II. OPTIMIZING FOR GLOBAL ISOTROPY

Many relationships exist for quantifying robot performance. They include, but are not limited to, the Jacobian *J(x)* (1) that relates actuator rates $\dot{q}$ to end-effector velocity $\dot{x}$, its transpose (2) which relates end-effector force/torque f to actuator force/torque τ and the impedance $Z(s, x)$ (3) presented by the robot to its environment where $Z(s, x)$ might contain mass, stiffness and damping terms.

$$\dot{q} = J(x)\dot{x} \tag{1}$$

$$f = J(x)^{T}\tau \tag{2}$$

$$f = Z(s, x)\dot{x} \tag{3}$$

Relationships (1) to (3) are matrix transformations that are functions of *x* (i.e. position dependent) and are non-diagonal (i.e. direction dependent) in general. Minimizing the non-uniformity associated with these dependencies is often the primary goal in robot design optimizations. For example, one would like the velocity of a welding robot to be accurate and consistent. It should, therefore, have an isotropic (i.e. direction independent) Jacobian (1) that does not change much at any position in its workspace.

While there are many ways to measure isotropy, a number of which are discussed in [8], the most common is the condition number which describes worst-case behaviour at a position. In the case of the Jacobian, the condition number (ratio between the largest and smallest singular values) describes the ratio between the highest and lowest effective transmission ratios occurring in all directions. For consistency, accuracy, direction independence and maximum distance from kinematic singularities, this ratio should be as close as possible to unity. The condition number approaches infinity as the robot nears a singular position and has a value of one when the robot is perfectly isotropic.

Isotropy is evaluated here using the "Global Isotropy Index" or GII (4) which is a more stringent measure than the condition number since the GII is workspace-inclusive whereas the condition number is position dependent. The GII, evaluated for a robot design parameter *p*, is the ratio between the smallest and largest singular values in the workspace *W* and is defined between *0* and *1* corresponding to singular behaviour and perfect isotropy, respectively.

$$GII(p) = \min_{x,\, y \in W} \frac{\sigma_{min}(J(p, x))}{\sigma_{max}(J(p, y))} \tag{4}$$

For an underactuated or redundant robot with a non-square design matrix, a GII similar to (4) can also be defined, but this is beyond the scope of this paper.

To obtain a position independent worst-case optimum, the computational requirements of global searching and the difficulty of integrating over the workspace [7] are avoided by using the Culling algorithm described in [12], [13] to maximize the GII. The Culling algorithm is a discrete minimax optimization algorithm that requires no prior knowledge or prediction of function values but relies solely on explicit function evaluations. It finds the parameter p^* (5) that produces the best GII within a discrete parameter space *P*:

$$p^* = \underset{p \in P}{\arg max}\ GII(p) \tag{5}$$

It starts by placing conservative upper (6) and lower (7) bounds on the minimum and maximum singular values for each parameter in the parameter space. In (6) and (7) the overbar and underscore represent upper and lower bounds.

$$\overline{\sigma_{min}}(p) = \infty;\ \forall p \in P \tag{6}$$

$$\underline{\sigma_{max}}(p) = 0;\ \forall p \in P \tag{7}$$

It then computes the actual minimum and maximum singular values and GII for an arbitrarily chosen parameter p^* by computing its singular values at all positions in the workspace. The positions $\underset{\sim}{x}$ and $\tilde{x}$ which minimize and maximize the singular values of $J(p^*, \underset{\sim}{x})$ and $J(p^*, \tilde{x})$ are then used to compute the singular values for all other parameters in P. This lowers the upper bound $\overline{\sigma_{min}}(p)$ and raises the lower bound $\underline{\sigma_{max}}(p)$. All parameters whose ratio of bound values are less than the GII of p^* cannot be the global optimum (8) and are culled from the parameter space P.

$$\text{if } \frac{\overline{\sigma_{min}}(p)}{\underline{\sigma_{max}}(p)} < GII(p^*) \quad \text{then } P := P - \{p\} \tag{8}$$

The actual minimum $\sigma_{min}(J(p, x \in W))$ and maximum $\sigma_{max}(J(p, y \in W))$ singular values and GII are computed for the parameter with the largest bound ratio. If this parameter has a superior GII, it replaces p^*. Regardless of its GII, the positions $\underset{\sim}{x}$ and $\tilde{x}$ which minimize and maximize the singular values are used to adjust the upper and lower bounds of the remaining parameters and the process is repeated. When all but one parameter have been eliminated by (8), the remaining parameter is, by default, the global optimum.

Note that the discrete parameter p is a vector containing any number of physical design parameters with a parameter space P that spans all possible combinations within prescribed upper and lower limits and sampling resolutions. Therefore, the Culling algorithm places no limit on the number of free variables. It can also be used with any performance function, is insensitive to initial conditions and has been found to be extremely efficient at solving robot optimization problems [12], [13].

For all design examples in this paper, the Culling algorithm is used to optimize the GII of the Jacobian *J(x)* in (2) with task-space requirements specified in terms of forces and torques.

III. Performance Based Matrix Scaling

Lipkin and Duffy [9] point out that the condition number of the Jacobian holds little practical significance when its elements have non-uniform physical units. This occurs when a robot is capable of both translating and rotating its end-effector or when it is comprised of both rotary and prismatic actuators. Furthermore, even when physical units are uniform, the singular values only evaluate the uniformity of actuator responsibilities given a task-space event of unit magnitude and arbitrary direction. They do not address the more general case of non-uniform actuator capabilities and/or task-space responsibilities. To remove this limitation, a more general description of desired robot performance is formulated and a conformity measure is derived.

A common robot design criteria is isotropy of end-point forces. For the sake of simplicity, assume that only forces f (i.e. no torques) are produced at the end-effector and that only torques τ are generated by the actuators (i.e. all rotary actuators) so that physical units are homogeneous. The more general case will be considered later. A robot is said to have an isotropic force profile at a position if the length of the joint-space torque vector (RMS value of all joint torques) is constant for any task-space force of unit magnitude. A perfectly isotropic force profile is, therefore, illustrated by the mapping shown in Fig. 1a. The ratio of singular values of a robot's Jacobian $J(x)$ at a position x describes how closely $J(x)$ approximates the transformation shown in Fig. 1a.

Fig. 1a is representative of an ideal isotropic mapping only if the intended use of the manipulator demands forces of equal magnitude in all directions. For some applications, it may be preferable that the robot be capable of larger forces along one axis than it is along another (e.g. a device affected by gravity). The ideal force/torque transformation for a device with non-homogeneous task-space force requirements would map an ellipse in task-space into a circle in joint-space. A mapping such as this would suggest that the kinematic chain has a mechanical advantage along the direction corresponding to the major axis of the task-space ellipse. Of course, the axes of the desired task-space force ellipse may not align with the coordinate system of the design matrix task-space variables resulting in a desired transformation shown in Fig. 1b.

Fig. 1b is representative of an ideal isotropic mapping only if the actuators have uniform torque capabilities. If the actuators have different torque capabilities, the stronger of the two would be under-utilized. Fig. 1b would, therefore, be undesirable for most serial manipulators since they typically use actuators of various sizes. Full actuator utilization would require that the task-space ellipse be mapped into a joint-space ellipse as shown in Fig. 1c, where the major axis of the joint-space ellipse aligns with the axis of the stronger actuator. Note that as long as there is no cross-coupling between actuators, the axes of the joint-space torque ellipse always align with the joint-space coordinate frame. Fig. 1c, therefore, illustrates a general representation of an ideal task-space to joint-space mapping given variations in both task-space requirements along different directions and individual actuator capabilities. Note that a similar argument is easily applied to non-uniform velocity, acceleration, resolution, and other task and joint-space quantities.

Conformance of a robot's Jacobian to a desired mapping between two ellipsoids such as that shown in Fig. 1c cannot be determined by singular values alone. If, however, a new transformation matrix $\hat{J}^{-T}$ is derived that transforms percentages of maximum values (i.e. Δf, $\Delta\tau$) rather than the actual values themselves (i.e. f, τ), even the more general representation of desired isotropic behaviour such as that shown in Fig. 1c takes on the familiar form shown in Fig. 1d since percentages are always unity bounded.

The GII of the transformation in Fig. 1d is easily computed from the minimum and maximum singular values in the workspace which corresponds to the ratio of the inner to outer radius of the object formed by superimposing all of the percentage torque ellipses in the workspace on top of one another. The GII ensures that all percentage torque ellipses are similar in both shape and size.

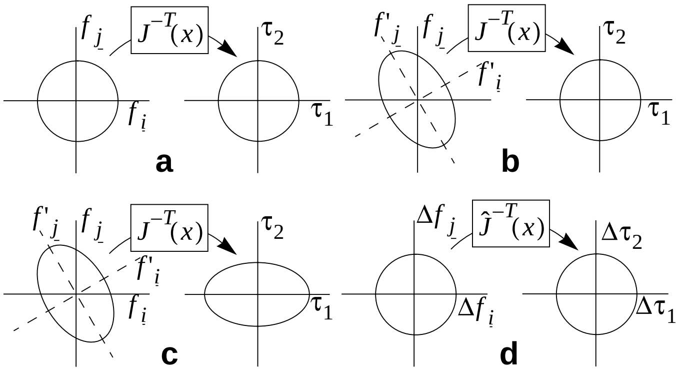

Fig. 1. Desired Force/Torque Transformations

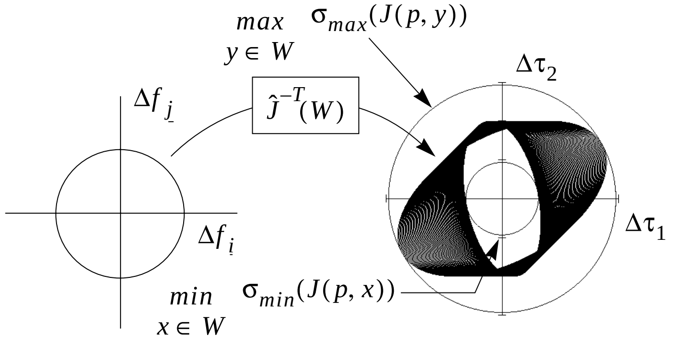
$\max_{y \in W} \sigma_{max}(J(p, y))$
$\Delta\tau_2$
Δf_j
$\hat{J}^{-T}(W)$
Δf_i
$\Delta\tau_1$
$\min_{x \in W} \sigma_{min}(J(p, x))$

Fig. 2. GII of Percent Force/Percent Torque Transformation

The normalized transformation matrix $\hat{J}(x)$ is computed by separating the vectors of task-space forces and joint-space torques (2) into diagonal scaling matrices (S'_T and S_J) with maximum values along the diagonal, a task-space rotation matrix S'_R which rotates the desired force ellipse's axes into the design matrix's task-space coordinate frame, and vectors (Δf and $\Delta\tau$) of unity bounded percentages (9). Because all scaling matrices are pulled from the task and joint-space vectors (f and τ), S'_R, S'_T and S_T are $n \times n$ matrices where n is the number of active degrees of freedom and S_J is an $m \times m$ matrix where m is the number of actuators. This holds regardless of whether $J(x)$ is square or whether the device is over or under-actuated. (9) is rearranged in (10) using definition (11) to derive the normalized transformation matrix $\hat{J}(x)$ (12).

$$S'_R S'_T \Delta f = J(x)^T S_J \Delta\tau \qquad (9)$$

$$\Delta f = S'^{-1}_T S'^{-1}_R J(x)^T S_J \Delta\tau = S^{-1}_T J(x)^T S_J \Delta\tau = \hat{J}(x)^T \Delta\tau \qquad (10)$$

$$S_T = S'_R S'_T \qquad (11)$$

$$\hat{J}(x) = S_J J(x) S_T^{-T} \qquad (12)$$

For a two dimensional robot, S'_R, S'_T, S_J, Δf and $\Delta\tau$ are expanded in (13) through (17) where $\tilde{f}'_i$ and $\tilde{f}'_j$ are the maximum desired forces along the i' and j' axes which define a reference frame rotated α radians from the i and j axes, $\tilde{\tau}_1$ and $\tilde{\tau}_2$ are the maximum torque capabilities of actuators *1* and

2, $\Delta f_{\underline{i}}$ and $\Delta f_{\underline{j}}$ are the percentages of maximum force along the $\underline{i}'$ and $\underline{j}'$ axes and $\Delta\tau_1$ and $\Delta\tau_2$ are the percentages of maximum torque of actuators *1* and *2*.

$$S'_R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{13}$$

$$S'_T = Diag\left[\tilde{f}'_{\underline{i}}\ \tilde{f}'_{\underline{j}}\right] \tag{14}$$

$$S_J = Diag\left[\tilde{\tau}_1\ \tilde{\tau}_2\right] \tag{15}$$

$$\Delta f = \left[\Delta f'_{\underline{i}}\ \Delta f'_{\underline{j}}\right]^T \tag{16}$$

$$\Delta\tau = \left[\Delta\tau_1\ \Delta\tau_2\right]^T \tag{17}$$

Note that S'_R may not be static as shown in (13). For example, it is configuration dependent if the task-space performance specifications are specified in end-effector coordinates. On the other hand, if the performance specifications are specified in base coordinates, (i.e $\alpha = 0$), S'_R becomes the identity, S_T becomes diagonal (18) and $\tilde{f}'_{\underline{i}}$ and $\tilde{f}'_{\underline{j}}$ become equal to $\tilde{f}_{\underline{i}}$ and $\tilde{f}_{\underline{j}}$ (19):

$$S_T = Diag\left[\tilde{f}_{\underline{i}}\ \tilde{f}_{\underline{j}}\right] \tag{18}$$

$$\Delta f = \left[\Delta f_{\underline{i}}\ \Delta f_{\underline{j}}\right]^T \tag{19}$$

Since the homogeneous Jacobian $\hat{J}(x)$ (12) transforms percentages into percentages, it is unitless and easily adapted to the more general case where mixed physical units are originally present. Consider, for example, a 6-DOF manipulator with rotary odd numbered joints and prismatic even numbered joints. S'_R, S'_T, S_J, Δf and $\Delta\tau$ are shown in equations (20) through (24) where R is the 3×3 rotation matrix which rotates the desired force ellipse axes into the design matrix task-space coordinate frame. Note that desired torques specified about a point non-collocated with the reference end-point results in off-diagonal terms in S'_R (20).

$$S'_R = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \tag{20}$$

$$S'_T = Diag\left[\tilde{f}'_{\underline{i}}\ \tilde{f}'_{\underline{j}}\ \tilde{f}'_{\underline{k}}\ \tilde{\tau}'_{\underline{i}}\ \tilde{\tau}'_{\underline{j}}\ \tilde{\tau}'_{\underline{k}}\right] \tag{21}$$

$$S_J = Diag\left[\tilde{\tau}_1\ \tilde{f}_2\ \tilde{\tau}_3\ \tilde{f}_4\ \tilde{\tau}_5\ \tilde{f}_6\right] \tag{22}$$

$$\Delta f = \left[\Delta f'_{\underline{i}}\ \Delta f'_{\underline{j}}\ \Delta f'_{\underline{k}}\ \Delta\tau'_{\underline{i}}\ \Delta\tau'_{\underline{j}}\ \Delta\tau'_{\underline{k}}\right]^T \tag{23}$$

$$\Delta\tau = \left[\Delta\tau_1\ \Delta f_2\ \Delta\tau_3\ \Delta f_4\ \Delta\tau_5\ \Delta f_6\right]^T \tag{24}$$

Although the Jacobian is used here for design examples, the S_T and S_J matrices can be used to normalize and scale any transformation matrix such as (1) or (3) but will contain quantities other than forces and torques. For example, to normalize a mass matrix *D(x)* (25) (i.e. $\hat{D}(x)$ (26)), S_J contains maximum end-effector forces/torques and S_T contains maximum end-effector accelerations (27). Note that in (27), S_J contains maximum end-effector forces which are task-space quantities. Therefore, S_J is really a second task-space matrix. Its notation is maintained for the sake of simplicity. Similarly, to normalize the velocity Jacobian $\hat{J}(x)$ (1), S_J contains maximum joint rates while S_T contains maximum end-effector velocities.

$$f = D(x)\ddot{x} \tag{25}$$

$$\Delta f = \hat{D}(x)\Delta\ddot{x} \tag{26}$$

$$\hat{D}(x) = S_J^{-1} D(x) S_T \tag{27}$$

Note that scalar multiples do not affect ratios of singular values so one element of S'_T and S_J can be factored out with the remaining elements representing relative values. Also note that these matrices are easily adapted to serial manipulators by rearranging their order (30) to account for the way the Jacobian is normally defined for a serial mechanism (28):

$$\tau = J(x)^T f \tag{28}$$

$$\Delta\tau = \hat{J}(x)^T \Delta f \tag{29}$$

$$\hat{J}(x) = S_T J(x) S_J^{-1} \tag{30}$$

IV. THE TASK-SPACE SCALING MATRIX: S_T

The task-space scaling matrix S_T is used to define the desired task-space performance capabilities of the device. Consider, for example, the three degree-of-freedom (3-DOF) parallel planar manipulator shown in Fig. 3. The geometry of the device is described by five design parameters, l_1 through l_4 and θ_0 and the workspace of the device is constrained by two translational, $\langle x \rangle$, $\langle y \rangle$ and one rotational limit $\langle \theta \rangle$. Although additional geometric parameters exist for this robot, they are fixed (i.e. *120°* separation between actuators on the platform and the base) to make parameter optimization more manageable. The Jacobian matrix of the manipulator in Fig. 3 is defined by (31) and is computed using equations (48) through (59) in Appendix A, part A.

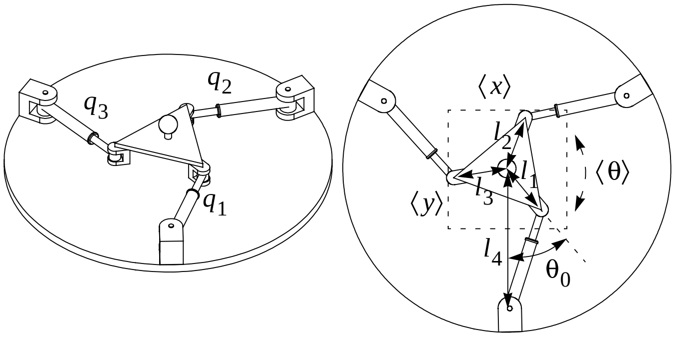

Fig. 3. 3-DOF Planar Parallel Manipulator

$$\left[f_{\underline{i}}\ f_{\underline{j}}\ \tau_{\underline{k}}\right]^T = J^T\left[f_1\ f_2\ f_3\right]^T \tag{31}$$

Since isotropy always improves when the radius of the base is increased, l_4 is fixed at *20cm*. The robot parameters l_1 through l_3 and θ_0 are found to optimize the GII inside a square workspace ($\langle x \rangle$, $\langle y \rangle$ = ±*5cm*, $\langle \theta \rangle$ = ±*30°*). To observe the effect of different task-space requirements on the optimum geometry, the optimization is carried out many times using different S_T matrices. In all cases, the force requirements are kept equal in both directions while the torque requirements are varied. Since equal force requirements make S'_R inconsequential, it is set to the identity. Substituting $\tilde{f}_{\underline{i}} = \tilde{f}_{\underline{j}} = \tilde{f}_{\underline{i},\underline{j}}$ into the maximum force/torque vector $\tilde{f}$ in (32) and dividing through by $\tilde{f}_{\underline{i},\underline{j}}$ (ratios of singular values are invariant to scalar multiples) produces the S_T matrix shown in (33). For $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$ varied between *0.1cm* and *10.0cm*, the optimum geometries and GIIs are shown in Fig. 4.

$$\tilde{f} = \begin{bmatrix}\tilde{f}_{\underline{i}} & \tilde{f}_{\underline{j}} & \tilde{\tau}_{\underline{k}}\end{bmatrix}^T \qquad (32)$$

$$S_T = Diag\begin{bmatrix}1 & 1 & \tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}\end{bmatrix} \qquad (33)$$

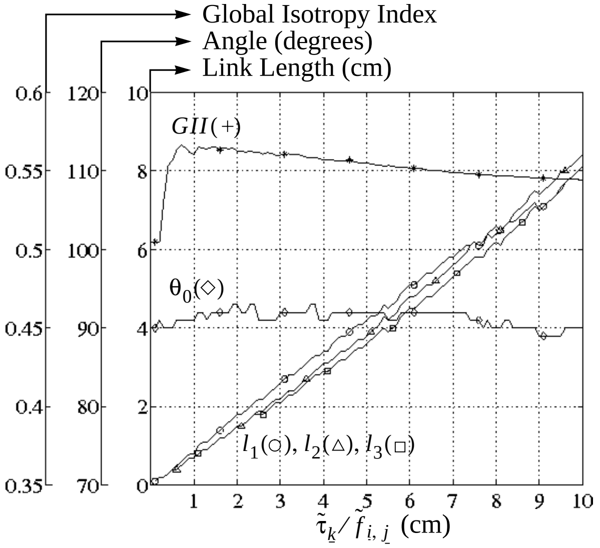

Fig. 4. Optimum Parallel Manipulator

The optimum platform offset angle θ_0 stays relatively constant at approximately *90°* and the platform is nearly equilateral in all cases. Its size increases linearly with $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$ which is, on average, *1.27* times the mean platform radius. This is expected since a larger platform radius provides the actuators with additional leverage on the platform centre and increases the task-space torque capabilities $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$. Fig. 4, therefore, supports the proposition that assigning values to the S_T matrix in accordance with a desired task-space performance specification produces a device that is particularly well suited to that performance specification.

Note that since S_T (33) places equal weighting upon all translational elements and all rotational elements, it is a special case that is mathematically equivalent to scaling by a Characteristic Length (CL) [14] equal to $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$. If one was to disregard the scale factor (i.e. the CL or $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$), as done in the past, one would be led to conclude that the platform radius corresponding to the NL (CL of *0.7cm* which maximizes the GII in Fig. 4) represents a globally optimum solution and that any two designs with similar GIIs should have similar performance characteristics. This obviously cannot be true since a platform radius of *5.3cm* is clearly capable of higher torques than a platform radius of *0.4cm*, but both produce the same GII. Thus, as expected, the optimum mechanism design varies significantly with the performance specification that is described by the scale factor(s) which, therefore, must not be ignored.

Since the S_T matrix does not just normalize physical units but also specifies a performance goal, it has practical application in all task-space dimensions, not just those with dissimilar physical units. Consider, for example, an assembly robot that lifts a part onto a shaft and locks it in place. Translational force requirements are not homogeneous since horizontal positioning forces need only address the mass of the payload while vertical positioning forces must also overcome gravity. The torque requirements are different still and correspond to the torque needed to lock the part into place. An example S'_R matrix that accounts for an angle of $\alpha = 30°$ between the manipulator's $\underline{j}$ axis and the real world's vertical ($\underline{j}'$) axis is shown in (34) and an example S'_T matrix for the requirements in (35) is shown in (36) which are used to re-optimize the robot in Fig. 3. Note that the S_T matrix (37) produced from S'_R (34) and S'_T (36) scales both translational and rotational elements and is not mathematically equivalent to multiplication by a CL.

$$S'_R = \begin{bmatrix}0.866 & 0.5 & 0\\ -0.5 & 0.866 & 0\\ 0 & 0 & 1\end{bmatrix} \quad \text{for} \quad \alpha = 30° \qquad (34)$$

$$\begin{bmatrix}\tilde{f}'_{\underline{i}} & \tilde{f}'_{\underline{j}} & \tilde{\tau}'_{\underline{k}}\end{bmatrix} = \begin{bmatrix}5N & 25N & 50Ncm\end{bmatrix} \qquad (35)$$

$$S'_T = Diag\begin{bmatrix}1 & \tilde{f}'_{\underline{j}}/\tilde{f}'_{\underline{i}} & \tilde{\tau}'_{\underline{k}}/\tilde{f}'_{\underline{i}}\end{bmatrix} = Diag\begin{bmatrix}1 & 5 & 10cm\end{bmatrix} \qquad (36)$$

$$S_T = \begin{bmatrix}0.866 & 2.5 & 0\\ -0.5 & 4.33 & 0\\ 0 & 0 & 10cm\end{bmatrix} \qquad (37)$$

The optimum robot geometry has $l_1 = 4.75$, $l_2 = 1.75$, $l_3 = 7.75$ and $\theta_0 = 77°$ with a GII of *0.158*. Unlike the devices described in Fig. 4, the optimum geometry for this application (shown in Fig. 5a) has asymmetric platform dimensions. This solution is not obvious and illustrates the power of the S_T matrix to tailor robot design parameters toward satisfying arbitrary task-space requirements. Note from Fig. 5b that if the task-space requirements are mirrored across the manipulator's $\underline{j}$ axis ($\alpha = -30°$), the optimum geometry $l_1 = 4.75$, $l_2 = 7.75$, $l_3 = 1.75$, $\theta_0 = -77°$ produces an identical GII (*0.158*). This is the mirror image of the original solution, just as one would expect. Any other rotation angle, however, produces a completely different optimum geometry (e.g. $l_1 = 3.25$, $l_2 = 8.5$, $l_3 = 7.75$, $\theta_0 = 97°$, GII = *0.155* for $\alpha = 0°$) as shown in Fig. 5c.

V. THE JOINT-SPACE SCALING MATRIX: S_J

The joint-space scaling matrix S_J is used to find the optimal actuators for a device. This accounts for a fundamental difference between S_T and S_J. S_T is a means of defining good performance while S_J is a means of achieving good performance. The S_J matrix, therefore, contains free design

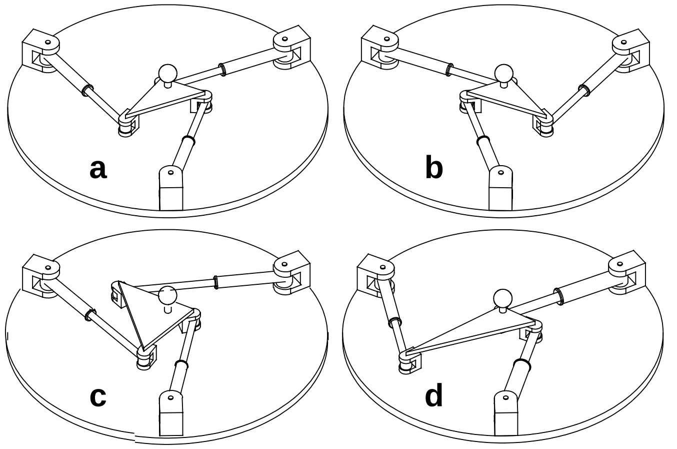

Fig. 5. Asymmetric 3-DOF Planar Parallel Manipulator

parameters that can be chosen simultaneously with geometric design parameters to arrive at the optimum solution.

To illustrate this, the robot design of Fig. 3 is re-visited using the S_T matrix in (37) with S_J derived from the maximum actuator force vector $\tilde{\tau}$ shown in (38) where $\tilde{f}_1$ through $\tilde{f}_3$ are the maximum force capabilities of actuators *1* through *3*. Dividing out a constant $\tilde{f}_1$ produces the S_J matrix in (39). With $\tilde{f}_2/\tilde{f}_1$ and $\tilde{f}_3/\tilde{f}_1$ treated as free design parameters, an optimum geometry with $l_1 = 4.5$, $l_2 = 1.0$, $l_3 = 14.5$ and $\theta_0 = 64°$, optimum actuator scale factors of $\tilde{f}_2/\tilde{f}_1 = 0.9$ and $\tilde{f}_3/\tilde{f}_1 = 0.5$, and a GII of *0.22* is obtained. Recall that the optimum GII was only *0.158* when the S_J matrix was not included (i.e. homogeneous actuators). A diagram of the optimized device with the proposed actuator scaling is shown in Fig. 5d.

$$\tilde{\tau} = \begin{bmatrix}\tilde{f}_1 & \tilde{f}_2 & \tilde{f}_3\end{bmatrix}^T \qquad (38)$$

$$S_J = Diag\begin{bmatrix}1 & \tilde{f}_2/\tilde{f}_1 & \tilde{f}_3/\tilde{f}_1\end{bmatrix} \qquad (39)$$

Although the S_J matrix (i.e. non-homogeneous actuation) improved the GII of the parallel manipulator by a substantial *39%*, it is even more effective with serial manipulators. In practice, serial manipulators rarely use the same actuators throughout because torque requirements tend to diminish for actuators distal to the base. It is for this same reason that they often produce dismal condition numbers since neglecting to include an S_J matrix evaluates the device as though it is fitted with homogeneous actuators. Consider, for example, the 3-DOF planar serial manipulator shown in Fig. 6 which has three design parameters, l_1 through l_3. The Jacobian matrix of the manipulator in Fig. 6 is defined by (40) and is computed using equations (60) through (66) in Appendix A, part B.

$$\begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^T = J^T\begin{bmatrix}f_{\underline{i}} & f_{\underline{j}} & \tau_{\underline{k}}\end{bmatrix}^T \qquad (40)$$

The workspace is centred a fixed distance y_0 of *5cm* above the base actuator q_1 and the robot parameters l_1 through l_3 are found to optimize the GII inside a square workspace ($\langle x\rangle$, $\langle y\rangle$ = *±5cm*) at any angle ($\langle\theta\rangle$ = *±180°*). The force requirements along the $\underline{i}$ and $\underline{j}$ axes are kept equal ($\tilde{f}_{\underline{i}} = \tilde{f}_{\underline{j}} = \tilde{f}_{\underline{i},\underline{j}}$) while the torque requirements $\tilde{\tau}_{\underline{k}}$ are varied using the S_T matrix in (33) (i.e. identity S'_R matrix). The results are shown in Fig. 7.

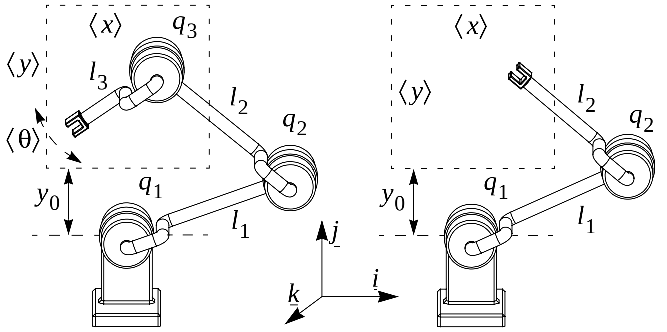

Fig. 6. Planar RRR and RR Serial Manipulators

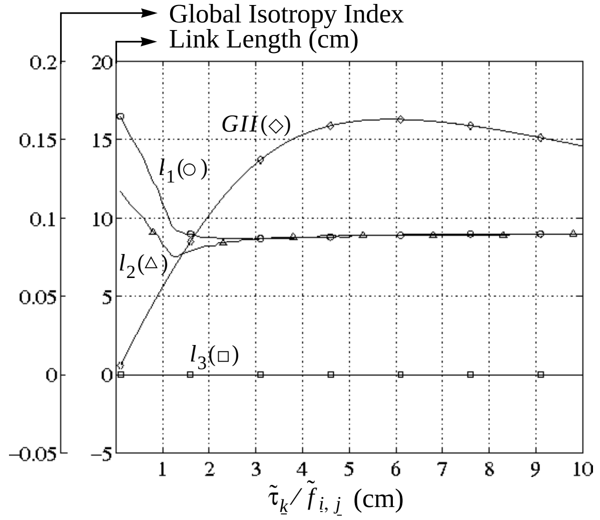

Fig. 7. Optimum Serial Manipulator with Homogeneous Actuation

The GII in Fig. 7 peaks at *0.163* and drops to *0.006* at low $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$ ratios. These results suggest that the serial device is unsuitable for applications with low torque requirements regardless of its geometry. These poor results, however, are not due to an intrinsic deficiency of the device but are largely due to inappropriate actuation. Performance is significantly improved by including the S_J matrix with free variables along the diagonal as shown in (41). Note that the S_J matrix in (41) is normalized with respect to $\tilde{\tau}_3$ since q_3 is likely to have the lowest torque requirements since it is furthest from the base.

$$S_J = Diag\begin{bmatrix}\tilde{\tau}_1/\tilde{\tau}_3 & \tilde{\tau}_2/\tilde{\tau}_3 & 1\end{bmatrix} \qquad (41)$$

Re-optimizing the serial robot with the two additional free parameters in (41) results in the design presented in Fig. 8. Notice the improvement in GII values which now vary from *0.17* to *0.28* with stronger actuators at the q_1 and q_2 joints. The non-homogeneously actuated device actually turns in its best results at low $\tilde{\tau}_{\underline{k}}/\tilde{f}_{\underline{i},\underline{j}}$ ratios with an up to *46*-fold improvement in its GII over its homogeneously actuated counterpart.

Actuator scaling can cause physical dimensions (i.e. l_1 and l_2) to grow without bound since this is analogous to shrinking the workspace size which has a favourable effect on isotropy. To avoid this when designing devices with inhomogeneous actuators, physical constraints must be imposed. In Fig. 8,

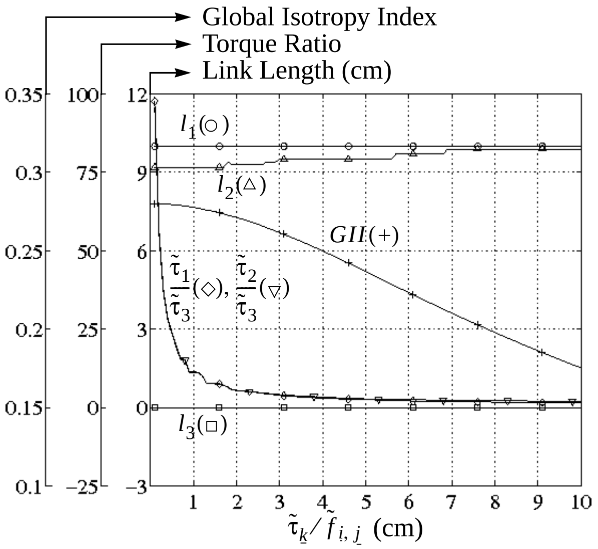

Fig. 8. Optimum Scaled Serial Manipulator

physical dimensions are limited to *10cm* so that a fair comparison can be made with Fig. 7. In practice, one might choose to fix the actuator ratios in accordance with practical considerations such as availability, size or weight. For example, $\tilde{\tau}_2$ is fixed at twice $\tilde{\tau}_3$ and $\tilde{\tau}_1$ is fixed at twice $\tilde{\tau}_2$ by using the S_J matrix shown in (43). Using the non-homogenous task-space requirements described by the S_T matrix in (42), the optimum geometry has l_1 = 9.6, l_2 = 6.6 and l_3 = 0 with a GII of *0.104*.

$$S_T = Diag\begin{bmatrix}1 & 3 & 5\end{bmatrix} \tag{42}$$

$$S_J = Diag\begin{bmatrix}4 & 2 & 1\end{bmatrix} \tag{43}$$

One might also fix both S_J and S_T when optimizing an impedance (3) or mass matrix (26) since both sides of the equation contain task-space quantities. Consider, for example, the 2-DOF planar serial manipulator shown in Fig. 6 which has two design parameters, l_1 and l_2. The mass matrix D is computed using equations (67) through (71) in Appendix A, part C and is scaled using (44) and (45).

$$S_J = Diag\begin{bmatrix}\tilde{f}_{\underline{i}} & \tilde{f}_{\underline{j}}\end{bmatrix} \tag{44}$$

$$S_T = Diag\begin{bmatrix}\tilde{\ddot{x}}_{\underline{i}} & \tilde{\ddot{x}}_{\underline{j}}\end{bmatrix} \tag{45}$$

By assigning S_J to the identity, the desired effective masses along $\underline{i}$ and $\underline{j}$ can be specified by assigning S_T to the desired acceleration for an applied unit force. The results of three example optimizations are shown in Table I with $\langle x\rangle$, $\langle y\rangle$ = ±*5cm* and y_0=*5cm*. A linkage mass of *10 g/cm* and an actuator mass of *50 g* are used to compute *D*.

Table I
Inertial Optimization Results for RR Manipulator

Desired Acceleration		Optimum		
i axis	j axis	l_1	l_2	GII
1.0	1.0	7.9	8.7	0.096
1.5	1.0	7.5	8.9	0.069
1.0	1.5	8.7	8.3	0.082

The resulting GIIs show that the device is well suited to having uniform mass or to be lighter along $\underline{j}$ than it is along $\underline{i}$ but is not easily made lighter along $\underline{i}$ than it is along $\underline{j}$.

Because S_J is typically used to remove the physical units from joint-space, it ensures a meaningful result even when a manipulator combines rotary and prismatic actuators. Consider, for example, the 3-DOF serial manipulator shown in Fig. 9 which has only one geometric design parameter l_1. Since q_2 applies a force rather than a torque, it is optimized using the S_J matrix shown in (46). The Jacobian matrix of the manipulator in Fig. 9 is defined by (47) and is computed using equations (72) through (76) in Appendix A, part D.

$$S_J = Diag\begin{bmatrix}\tilde{\tau}_1/\tilde{\tau}_3 & \tilde{f}_2/\tilde{\tau}_3 & 1\end{bmatrix} \tag{46}$$

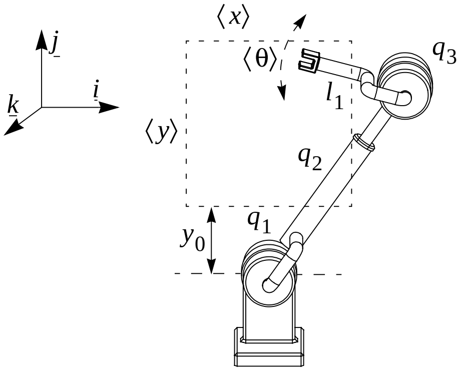

Fig. 9. 3-DOF Planar RPR Serial Manipulator

$$\begin{bmatrix}\tau_1 & f_2 & \tau_3\end{bmatrix}^T = J^T\begin{bmatrix}f_{\underline{i}} & f_{\underline{j}} & \tau_{\underline{k}}\end{bmatrix}^T \tag{47}$$

For $\langle x\rangle$, $\langle y\rangle$ = ±*5cm*, $\langle\theta\rangle$ = ±*180°*, y_0=*5cm* and the force/torque requirements in (42), the optimum robot has l_1 = *0cm*, $\tilde{\tau}_1/\tilde{\tau}_3$ = 3.6 and $\tilde{f}_2/\tilde{\tau}_3$ = $0.9cm^{-1}$ with a GII of *0.203*. This robot shows a significantly (95%) better GII than the homogeneous serial arm (Fig. 6) but in practice may suffer from larger inertia due to its prismatic upper-arm.

VI. Discussion

By converting explicit joint and task-space values into scaling matrices of maximum values and vectors of percentages, a meaningful and dimensionless condition number is derived from the singular values of the normalized Jacobian or other (i.e. mass, impedance) performance matrix. The scaling matrices, S_T and S_J, remove physical units from both the task and actuation spaces, provide a means of specifying desired behaviour along each task-space dimension and provide a means of scaling actuators for improved performance. Optimization of the scaled, unitless performance matrix (i.e. $\hat{J}$, $\hat{D}$, $\hat{Z}$, etc.) ensures that the task-space specifications are optimally satisfied using the best choice of robot actuators.

It is shown by example that the optimum geometry is adjusted predictably by different S_T matrices when the performance implications of different geometries are self-evident. It is, therefore, concluded that the S_T matrix does, in fact, have the proposed effect on robot designs and shows great potential for selecting robot parameters in cases where no such obviousness exists.

It is also shown by example that the S_J matrix can be used to scale individual actuator capabilities to improve performance. S_J is shown to be insensitive to mixed physical units in joint-space and is demonstrated to be most effective when designing serial devices. Improvements in the isotropy index of up to two orders of magnitude have been observed as a direct result of actuator scaling.

VII. Appendix A

A. Jacobian of Planar Parallel Manipulator

The Jacobian (31) of the planar parallel manipulator in Fig. 3 is computed from (48) using the terms defined in (49) through (59) and the origin and frame assignments shown in Fig. 10.

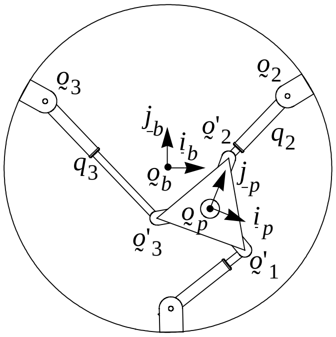

Fig. 10. Frames of Planar Parallel Manipulator

$$J=\begin{bmatrix} \frac{1}{q_1}({}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_1+{}^bC_p{}^p\underset{\sim}{o}'_1)^T & \frac{1}{q_1}det\begin{bmatrix} {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_1 & {}^bC_p{}^p\underset{\sim}{o}'_1 \end{bmatrix} \\ \frac{1}{q_2}({}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_2+{}^bC_p{}^p\underset{\sim}{o}'_2)^T & \frac{1}{q_2}det\begin{bmatrix} {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_2 & {}^bC_p{}^p\underset{\sim}{o}'_2 \end{bmatrix} \\ \frac{1}{q_3}({}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_3+{}^bC_p{}^p\underset{\sim}{o}'_3)^T & \frac{1}{q_3}det\begin{bmatrix} {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_3 & {}^bC_p{}^p\underset{\sim}{o}'_3 \end{bmatrix} \end{bmatrix} \quad (48)$$

$${}^b\underset{\sim}{o}_p = \begin{bmatrix} x & y \end{bmatrix}^T \quad (49)$$

$${}^b\underset{\sim}{o}_1 = \begin{bmatrix} 0 & -l_4 \end{bmatrix}^T \quad (50)$$

$${}^b\underset{\sim}{o}_2 = \begin{bmatrix} 0.866l_4 & 0.5l_4 \end{bmatrix}^T \quad (51)$$

$${}^b\underset{\sim}{o}_3 = \begin{bmatrix} -0.866l_4 & 0.5l_4 \end{bmatrix}^T \quad (52)$$

$${}^bC_p = \begin{bmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{bmatrix} \quad (53)$$

$${}^p\underset{\sim}{o}'_1 = \begin{bmatrix} 0 & -l_1 \end{bmatrix}^T \quad (54)$$

$${}^p\underset{\sim}{o}'_2 = \begin{bmatrix} 0.866l_2 & 0.5l_2 \end{bmatrix}^T \quad (55)$$

$${}^p\underset{\sim}{o}'_3 = \begin{bmatrix} -0.866l_2 & 0.5l_2 \end{bmatrix}^T \quad (56)$$

$$q_1 = \left\| {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_1+{}^bC_p{}^p\underset{\sim}{o}'_1 \right\| \quad (57)$$

$$q_2 = \left\| {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_2+{}^bC_p{}^p\underset{\sim}{o}'_2 \right\| \quad (58)$$

$$q_3 = \left\| {}^b\underset{\sim}{o}_p-{}^b\underset{\sim}{o}_3+{}^bC_p{}^p\underset{\sim}{o}'_3 \right\| \quad (59)$$

B. Jacobian of Planar RRR Serial Manipulator

The Jacobian (40) of the planar serial manipulator in Fig. 6 is computed from (60) using the terms defined in (61) through (66) and the origin and frame assignments shown in Fig. 11 where all joint angles are set to zero.

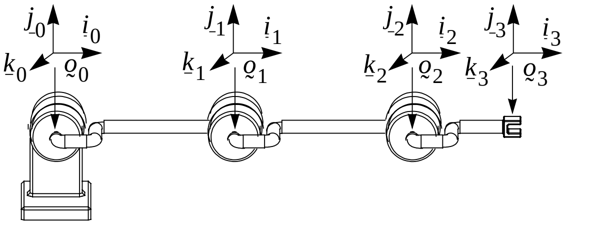

Fig. 11. Frames of Planar RRR Serial Manipulator

$$J = \begin{bmatrix} \underline{j}^T(\underset{\sim}{o}_0-\underset{\sim}{o}_3) & \underline{j}^T(\underset{\sim}{o}_1-\underset{\sim}{o}_3) & \underline{j}^T(\underset{\sim}{o}_2-\underset{\sim}{o}_3) \\ \underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_0) & \underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_1) & \underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_2) \\ 1 & 1 & 1 \end{bmatrix} \quad (60)$$

$$\begin{aligned} \underline{j}^T(\underset{\sim}{o}_0-\underset{\sim}{o}_3) &= -l_1\sin(q_1)-l_2\sin(q_1+q_2) \\ &\quad -l_3\sin(q_1+q_2+q_3) \end{aligned} \quad (61)$$

$$\underline{j}^T(\underset{\sim}{o}_1-\underset{\sim}{o}_3) = -l_2\sin(q_1+q_2)-l_3\sin(q_1+q_2+q_3) \quad (62)$$

$$\underline{j}^T(\underset{\sim}{o}_2-\underset{\sim}{o}_3) = -l_3\sin(q_1+q_2+q_3) \quad (63)$$

$$\begin{aligned} \underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_0) &= l_1\cos(q_1)+l_2\cos(q_1+q_2) \\ &\quad +l_3\cos(q_1+q_2+q_3) \end{aligned} \quad (64)$$

$$\underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_1) = l_2\cos(q_1+q_2)+l_3\cos(q_1+q_2+q_3) \quad (65)$$

$$\underline{i}^T(\underset{\sim}{o}_3-\underset{\sim}{o}_2) = l_3\cos(q_1+q_2+q_3) \quad (66)$$

C. Mass Matrix of Planar RR Serial Manipulator

In [3] a mass matrix D' (67) is computed for an RR serial elbow manipulator (68) where m_1 and m_2 are point masses located at the ends of link l_1 and l_2 respectively. D' is transformed from joint space (67) to task-space (69), (70) using the Jacobian (71) from [11] and the assumption that the derivative of the Jacobian is small enough to be neglected.

$$\tau = D'\ddot{q} \tag{67}$$

$$D' = \begin{bmatrix} d'_{11} & d'_{12} \\ d'_{21} & d'_{22} \end{bmatrix} \tag{68}$$

$$d'_{11} = l_2^2 m_2 + 2 l_1 l_2 m_2 \cos(q_2) + l_1^2 (m_1 + m_2)$$

$$d'_{12} = d'_{21} = l_2^2 m_2 + l_1 l_2 m_2 \cos(q_2)$$

$$d'_{22} = l_2^2 m_2$$

$$f = D\ddot{x} \tag{69}$$

$$D = J^{-T} D' J^{-1} \tag{70}$$

$$J = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix} \tag{71}$$

D. Jacobian of Planar RPR Serial Manipulator

The Jacobian (47) of the planar serial manipulator in Fig. 9 is computed from (72) using the terms defined in (73) through (76) and the origin and frame assignments shown in Fig. 12 where all joint angles are set to zero.

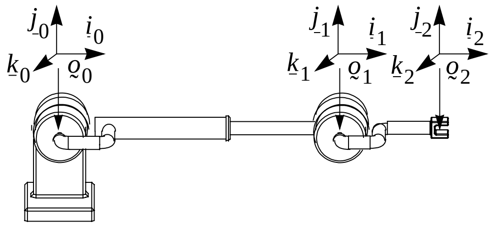

Fig. 12. Frames of Planar RPR Serial Manipulator

$$J = \begin{bmatrix} \underline{j}^T(\underset{\sim}{o}_0 - \underset{\sim}{o}_2) & \cos(q_1) & \underline{j}^T(\underset{\sim}{o}_1 - \underset{\sim}{o}_2) \\ \underline{i}^T(\underset{\sim}{o}_2 - \underset{\sim}{o}_0) & \sin(q_1) & \underline{i}^T(\underset{\sim}{o}_2 - \underset{\sim}{o}_1) \\ 1 & 0 & 1 \end{bmatrix} \tag{72}$$

$$\underline{j}^T(\underset{\sim}{o}_0 - \underset{\sim}{o}_2) = -q_2 \sin(q_1) - l \sin(q_1 + q_3) \tag{73}$$

$$\underline{j}^T(\underset{\sim}{o}_1 - \underset{\sim}{o}_2) = -l \sin(q_1 + q_3) \tag{74}$$

$$\underline{i}^T(\underset{\sim}{o}_2 - \underset{\sim}{o}_0) = q_2 \cos(q_1) + l \cos(q_1 + q_3) \tag{75}$$

$$\underline{i}^T(\underset{\sim}{o}_2 - \underset{\sim}{o}_1) = \cos(q_1 + q_3) \tag{76}$$

ACKNOWLEDGEMENT

This work was supported by the Canadian IRIS Network of Centres of Excellence projects, HMI-6 and IS-8 and a scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] J. Angeles, “Kinematic Isotropy in Humans and Machines”, *Proc. IFToMM 9th World Cong. Theory of Mach. & Mech.* (Milan, Italy), V. 1, pp. XLII-XLIX, Aug. 29 - Sept. 2, 1995.

[2] J. Angeles, F. Ranjbaran, R.V. Patel, “On the Design of the Kinematic Structure of Seven-Axes Redundant Manipulators for Maximum Conditioning”, *Proc. IEEE Int. Conf. Robotics & Auto.* (Nice, France), pp. 494-499, May 10-15, 1992.

[3] J.J. Craig, “Introduction to Robotics Mechanics and Control”, 2nd Ed., Addison-Wesley, 1989.

[4] K. Doty, C. Melchiorri, C. Bonevento, “A Theory of Generalized Inverses Applied to Robotics”, *The Int. J. of Robotics Res.*, V. 12, No. 1, pp. 1-19, Feb. 1993.

[5] J. Doyle, “Analysis of Feedback Systems with Structured Uncertainties”, *IEE Proc.*, V. 129, Pt. D, No. 6, Nov. 1982.

[6] C. Gosselin, “Dexterity Indices for Planar and Spatial Robotic Manipulators”, *Proc. IEEE Int. Conf. Robotics & Auto.* (Cincinnati, Ohio), pp. 650-655, May 13-18, 1990.

[7] C. Gosselin, J. Angeles, “A Global Performance Index for the Kinematic Optimization of Robot Manipulators”, *Trans. ASME, J. Mech. Des.*, V. 113, pp. 220-226, Sept. 1991.

[8] J-O. Kim, P.K. Khosla, “Dexterity Measures for Design and Control of Manipulators”, *Proc. IROS ‘91, IEEE/RSJ Int. Workshop Intell. Robots & Sys.* (Osaka, Japan), pp. 758-763, Nov. 3-5, 1991.

[9] H. Lipkin, J. Duffy, “Hybrid Twist and Wrench Control for a Robotic Manipulator”, *Trans. ASME, J. Mech., Trans. & Auto. in Design*, V. 110, pp. 138-144, June 1988.

[10] O. Ma, J. Angeles, “Optimum Architecture Design of Platform Manipulators”, *Proc. IEEE Int. Conf. Advanced Robotics*, 1991.

[11] M.W. Spong, M. Vidyasagar, “Robot Dynamics and Control”, John Wiley & Sons, 1989.

[12] L. Stocco, S.E. Salcudean, F. Sassani “Fast Constrained Global Minimax Optimization of Robot Parameters”, *Robotica*, V. 16, pp. 595-605, 1998.

[13] L. Stocco, S.E. Salcudean, F. Sassani “Mechanism Design for Global Isotropy with Applications to Haptic Interfaces”, *Proc. ASME Winter Annual Meeting.* (Dallas, Texas), V. 61, pp. 115-122, Nov. 15-21, 1997.

[14] M. Tandirci, J. Angeles, F. Ranjbaran, “The Characterstic Point and the Characteristic Length of Robotic Manipulators”, *Proc. ASME 22nd Biennial Conf. Robotics, Spatial Mechanisms & Mech. Sys.* (Scotsdale, Arizona), V. 45, pp. 203-208, Sept. 13-16, 1992.