

# Matrix Normalization for Optimal Robot Design

L. Stocco, S. E. Salcudean and F. Sassani\*

Department of Electrical and Computer Engineering, \*Department of Mechanical Engineering,  
The University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4  
*leos@ece.ubc.ca, tims@ece.ubc.ca, sassani@mech.ubc.ca*

## 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 cope with the problem of non-homogeneous physical units. The technique pre and post-multiplies a design matrix by diagonal scaling matrices corresponding to the range of joint and task space variables. In the case of the Jacobian, normalization leads to a practical interpretation of a robot’s “Characteristic Length” as the desired ratio between maximum linear and angular force or velocity. The scale factors can also be used to set relative required strength or speed along any axes of end-point motion and/or can be treated as free design parameters to improve isotropy through asymmetric 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.*

## 1 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 asymmetric actuation. The technique also normalizes physical units to ensure a meaningful result.

Methods for handling non-uniform workspace dimensions have been suggested by Gosselin [4], Tandirci et. al. [10], Angeles [1], Ma and Angeles [8] and Angeles et. al. [2]. They address the problem pointed out by Lipkin and Duffy [7] 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 which appear

when a robot can both translate and rotate its end-effector. To accommodate this, Gosselin [4] 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. [10] 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 [8] 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.

It is shown here that the CL, in fact, represents a robot’s relative capability to translate and rotate its end-effector. It should be chosen to best satisfy the demands of the application and should not be a free design parameter. The CL is extended to a more general diagonal matrix which scales all workspace dimensions, not just those with dissimilar physical units. A similar type of scaling is also applied to joint-space to simultaneously remove mixed physical units that appear when dissimilar types of actuators are used (e.g. the Stanford Arm) and improve isotropy through asymmetric actuation.

Section 2 of this paper discusses the definition of isotropy and the optimization algorithm used in subsequent design examples. Section 3 describes the proposed scaling matrices that are the focus of this paper. The task-space scaling matrix and its relationship with the CL is described in Section 4 while the joint-space scaling matrix is described in Section 5 with a summary and conclusions in Section 6.

## 2 Isotropy and Optimization

Many different relationships are used to quantify robot performance. They include, but are not limited to, the Jacobian  $J(x)$  (1) that relates actuator velocity  $\dot{q}$  to end-effector velocity  $\dot{x}$ , its transpose (2) which relates end-effector force/torque  $f$  to actuator force/torque  $\tau$  and the equivalent impedance  $Z_E$  (3) presented by the robot to its environment where robot impedance  $Z_R(x)$  contains mass, stiffness and damping terms.

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

$$f = J(x)^T \tau \quad (2)$$

$$Z_E = J(x)^T Z_R(x) J(x) \quad (3)$$

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 (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 [6], the most common is the condition number which describes worst-case behaviour at a position. The condition number is the ratio between the largest and smallest singular values which, in the case of the Jacobian, are the highest and lowest effective transmission ratios occurring in all directions. For consistency, accuracy, and direction independence, this ratio should be as close as possible to unity. When the condition number has a value of zero, the robot is uncontrollable and is said to be in a singular position. When it has a value of one, the robot is considered to be perfectly isotropic.

For all design examples in this paper, the goal is an optimally isotropic Jacobian matrix  $J(x)$  with task-space requirements specified in terms of forces and torques (2). To obtain a position independent worst-case optimum, the computational intensity of global searching and the difficulty of integrating over the workspace [5] are avoided by using the ‘‘Global Isotropy Index’’ or GII (4) and Culling algorithm proposed in [9]. 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. Due to its workspace-inclusive nature, the GII is a more stringent measure than the condition number.

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

The Culling algorithm is a discrete minimax optimization algorithm that requires no prior knowledge or estimation of function values. It finds the parameter  $p^*$  (5) that produces the best GII within a discrete parameter space  $P$ . A simpler version also exists and is presented in [9] to optimize the more common minimax problem (6) for an arbitrary function  $F(p, x)$  bounded by 0 and 1.

$$p^* = \operatorname{argmax}_{p \in P} GII(p) \quad (5)$$

$$F^* = \max_{p \in P} \min_{x \in W} F(p, x) \quad (6)$$

It starts by placing a conservative lower bound on the optimum function value ( $F^* = 0$ ) and a non-conservative upper bound on the function value of each parameter ( $\overline{F}(p, x^*) = 1$ ) where the underscore represents a lower bound, the overbar represents an upper bound and  $x^*$  is the position that minimizes the function value for a parameter  $p$ . It then iteratively pulls up on  $\underline{F}^*$  by searching one parameter for all positions and pushes down on each element of  $\overline{F}(p, x^*)$  by searching one position for all parameters. When the upper bound  $\overline{F}(p, x^*)$  of a parameter  $p$  drops below the optimum lower bound  $\underline{F}^*$ , that parameter is eliminated from the parameter space  $P$ . Consider the example search space in Figure 1 where each discrete workspace location is assigned a row and each discrete parameter value is assigned a column.

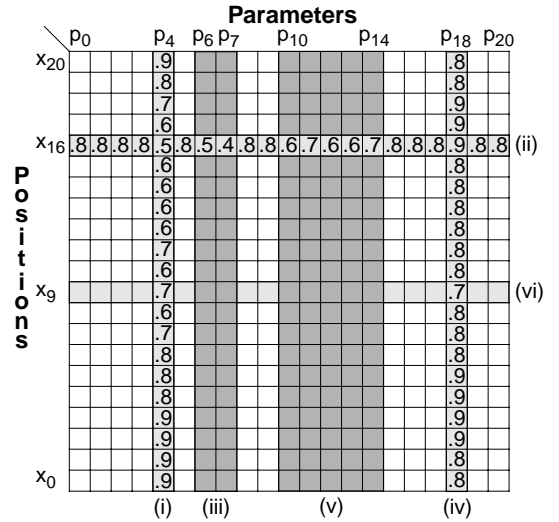


Figure 1: Culling Example

The Culling algorithm searches all positions of an arbitrarily chosen initial parameter  $p_4$  (i) to find that its worst case value  $F^* = 0.5$  occurs at  $x_{16}$ . All parameters are searched at  $x_{16}$  (ii) to find that  $F(p, x^*) \leq F^*$  for  $p_6$  and  $p_7$  and that a maximum is obtained at  $p_{18}$ . Parameters  $p_6$  and  $p_7$  are eliminated from  $P$  (iii) and all positions of  $p_{18}$  are searched (iv) to find that its worst case value of 0.7 occurs at  $x_9$ .  $F^*$  is updated to 0.7 and parameters  $p_4$  and  $p_{10} \rightarrow p_{14}$  are eliminated from  $P$  (v). All parameters are searched at  $x_9$  (vi) and so on until all but one parameter is eliminated. The remaining parameter  $p^*$  is the global optimum which produces the value  $F^* = F(p^*, x^*)$ . The Culling algorithm guarantees a global optimum with significantly less computational effort than a global search since many function evaluations are avoided (e.g. darkened areas in Figure 1). It can be used with any performance function, places no limitation on the number of free variables, is insensitive to initial conditions and has been found to be extremely efficient at solving robot optimization problems [9].

### 3 Physical Unit Normalization

Lipkin and Duffy [7] point out that the condition number of the Jacobian is of little practical significance when its elements have non-uniform physical units. This occurs when the robot can both translate and rotate its end-effector or when the robot is comprised of both rotary and prismatic actuators. For a homogeneous Jacobian, the physical units must be removed from both sides of the equation. This can be accomplished by separating each force/torque vector ( $f$  and  $\tau$ ) into a diagonal scaling matrix with positive maximum force/torque values on the diagonal and a vector of unity bounded percentages. This is shown in (7) where  $S_J$  is the joint-space scaling matrix,  $S_T$  is the task-space scaling matrix,  $\Delta\tau$  is the actuator force/torque percentage vector and  $\Delta f$  is the end-effector force/torque percentage vector. Substituting (7) into (2) produces (8) which contains no physical units.

$$\tau = S_J \Delta\tau \quad f = S_T \Delta f \quad (7)$$

$$\Delta f = \hat{J}^T \Delta\tau \quad (8)$$

$$\hat{J} = S_J J S_T^{-1} \quad (9)$$

The homogeneous Jacobian  $\hat{J}$  (9) relates the percentage of maximum actuator capabilities  $\Delta\tau$  to the percentage of maximum end-effector requirements  $\Delta f$ , both of which are unity bounded. Because  $\hat{J}$  transforms percentages into percentages, it is dimensionless and its condition number is meaningful regardless of the actuators used or the workspace dimensions involved.  $\hat{J}$  bears a strong similarity to structured singular values which are used in controller design to measure stability and robustness of systems with structured uncertainties. Doyle [3] shows that a non-conservative estimate of stability can be made for a compensated system containing various uncorrelated disturbances of known bound by the structured singular value which is the solution to the minimax problem shown in (10) where  $D$  is a diagonal matrix of positive real values.

$$\mu = \min_D \sigma_{\max}(D Q D^{-1}) \quad (10)$$

There is, however, a fundamental difference between the  $D$  matrix used in structured singular values and the  $S_T$  and  $S_J$  matrices used here to model robot performance. In robot design,  $S_T$  is a diagonal matrix of maximum end-effector force/torque while  $S_J$  is a diagonal matrix of maximum actuator force/torque. End-effector requirements are dictated by the application so  $S_T$  partially describes the goal whereas actuators can be chosen to best satisfy the requirements so  $S_J$  partially describes the solution. Because of this intrinsic dissimilarity, the two matrices must be handled differently. In control, a common  $D$  matrix multiplies both sides of the system matrix ( $Q$ ) and its elements are assigned by the optimization algorithm.

Although  $S_T$  and  $S_J$  are shown in (7) through (9) to normalize the force/torque Jacobian (2), they can be used to normalize and scale any transformation matrix such as (1) or (3). Since different transformations involve different physical quantities,  $S_T$  and  $S_J$  will not always contain forces and torques. For example,  $Z_E$  (3) transforms end-effector velocity  $\dot{x}$  into end-effector force/torque  $f$  (11). To normalize  $Z_E$  (12),  $S_J$  should contain maximum end-effector velocities and  $S_T$  should contain maximum end-effector forces/torques (13). Note that with  $Z_E$ ,  $S_J$  contains task-space quantities and is really a second  $S_T$  matrix ( $S_{T2}$ ). The original notation ( $S_J$ ) is maintained purely for the sake of convenience. Similarly, it is easily shown that  $J(x)$  from (1) is normalized and scaled with an  $S_J$  containing maximum joint velocities and an  $S_T$  containing maximum end-effector velocities.

$$f = Z_E \dot{x} \quad (11)$$

$$\Delta f = \hat{Z}_E \Delta \dot{x} \quad (12)$$

$$\hat{Z}_E = S_T^{-1} Z_E S_J \quad (13)$$

Note that scalar multiples do not affect ratios of singular values so one element of  $S_T$  and  $S_J$  can be assigned to unity with the remaining elements representing relative values. Also note that these matrices are easily adapted to serial manipulators by rearranging the order (16) to account for the way the Jacobian is normally defined for a serial mechanism (14).

$$\dot{x} = J \dot{q} \quad \tau = J^T f \quad (14)$$

$$\Delta \tau = \hat{J}^T \Delta f \quad (15)$$

$$\hat{J} = S_T J S_J^{-1} \quad (16)$$

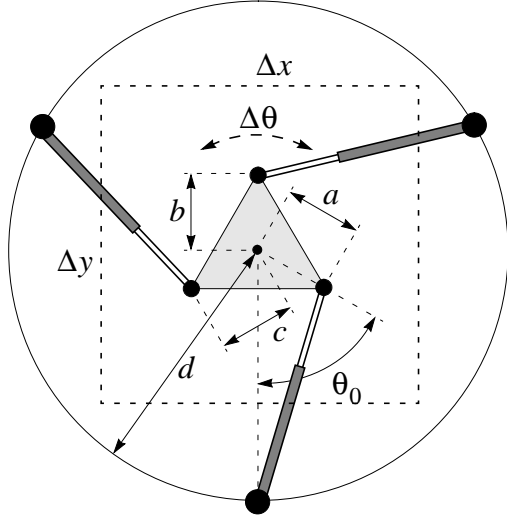
### 4 The Task-Space Scaling Matrix: $S_T$

The task-space scaling matrix  $S_T$  is a generalized extension of the CL. It is shown in (17) that the two are analogous if maximum desired forces along all axes are identical and equal to  $\bar{f}_{lin}$ , maximum desired torques about all axes are identical and equal to  $\bar{f}_{ang}$  and the CL is taken to represent  $\bar{f}_{ang}/\bar{f}_{lin}$ . Multiplication by a CL is, therefore, a special case of the  $S_T$  matrix.

$$\begin{bmatrix} 1 & 0 \\ 0 & CL \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{\bar{f}_{ang}}{\bar{f}_{lin}} \end{bmatrix} = \frac{1}{\bar{f}_{lin}} \begin{bmatrix} \bar{f}_{lin} & 0 \\ 0 & \bar{f}_{ang} \end{bmatrix} = S_T \quad (17)$$

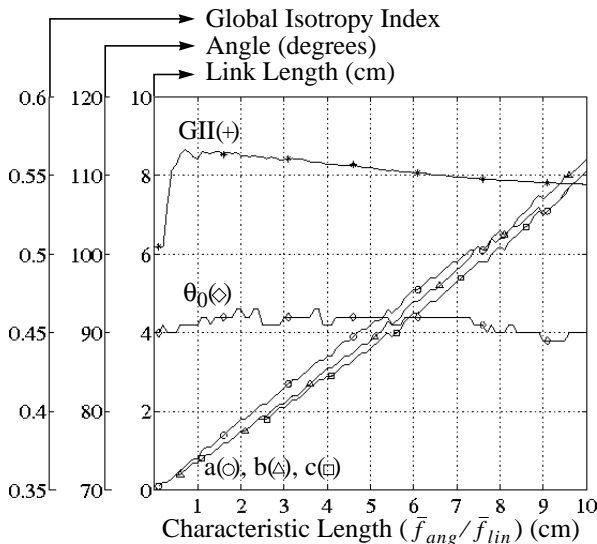
In the past, the value of the CL has not been associated with any physical meaning. Common practice has been to choose it freely for optimum conditioning and then discard it. In (17) it is shown that the CL does have physical meaning and, in particular, that it represents the robot's relative ability to apply forces and torques with its end-effector. Adjusting the CL to improve isotropy alters the goal to fit the solution. While this approach may be reasonable if the application is poorly defined, it is not practical in general. Consider, for example, the three

degree-of-freedom (3-DOF) parallel planar manipulator shown in Figure 2. The geometry of the device is described here by five design parameters  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $\theta_0$ .



**Figure 2: 3-DOF Planar Parallel Manipulator**

While additional geometric parameters exist for this robot, they are fixed (i.e.  $120^\circ$  separation between actuators on the platform and the base) for the sake of simplicity since the five parameters are adequate to illustrate the relationship between the CL and corresponding optimum geometries. Since isotropy always improves when  $d$  is increased, it is fixed at  $20\text{cm}$ . The robot parameters  $a$ ,  $b$ ,  $c$  and  $\theta_0$  are found to optimize the GII inside a square workspace ( $\Delta x, \Delta y = \pm 5\text{cm}$ ,  $\Delta\theta = \pm 30^\circ$ ) for CLs ranging between  $0.1\text{cm}$  and  $10.0\text{cm}$ . The optimum geometries and GIIs are shown in Figure 3.



**Figure 3: Optimum Parallel Manipulator**

The optimum platform is nearly equilateral and the optimum  $\theta_0$  is approximately  $90^\circ$  in all cases. Platform size increases linearly with the CL which is, on average,

$1.27$  times the mean platform radius. The NL is  $0.7\text{cm}$ , however the GII only fluctuates by  $12\%$  ( $0.505 @ \text{CL}=0.1\text{cm} \rightarrow 0.567 @ \text{CL}=0.7\text{cm}$ ) even though the platform radius changes by over a factor of  $80$  ( $0.1 @ \text{CL}=0.1\text{cm} \rightarrow 8.1 @ \text{CL}=10.0\text{cm}$ ). Note that many highly dissimilar geometries are shown to have identical GIIs in Figure 3. For example, platform radii of  $0.4\text{cm}$  and  $5.3\text{cm}$  both produce a GII of  $0.554$ . The reason why two devices whose platform sizes differ by a factor of  $14$  are evaluated as equals is because they are evaluated against two different specifications. The CL sets the physical property  $\bar{f}_{ang}/\bar{f}_{lin}$  so larger CL values suggest higher torque capabilities and result in a larger platform radius. A larger platform does not affect linear force capabilities much but provides the actuators with additional leverage for inducing torques. A relatively constant level of isotropy is, therefore, maintained over a wide range of relative force/torque requirements provided that the appropriate geometry is chosen.

Assigning desired force ratios clearly does more than remove physical units and is, therefore, applicable to all dimensions. Consider, for example, an assembly robot that places a part on a shaft and turns it to lock it in place. Linear force requirements are not identical since vertical positioning forces must overcome the additional burden of gravity. The robot in Figure 2 can be optimized for this purpose by assuming a maximum horizontal force  $\bar{f}_x$  of  $5\text{N}$ , a maximum vertical force  $\bar{f}_y$  of  $15\text{N}$  and a maximum torque  $\bar{f}_\theta$  of  $25\text{Ncm}$ . The corresponding  $S_T$  matrix is shown in (19) after factoring out a  $5\text{N}$  scalar.

$$\bar{f}_x = 5\text{N} \quad \bar{f}_y = 15\text{N} \quad \bar{f}_\theta = 25\text{Ncm} \quad (18)$$

$$S_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{f}_y/\bar{f}_x & 0 \\ 0 & 0 & \bar{f}_\theta/\bar{f}_x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5\text{cm} \end{bmatrix} \quad (19)$$

After normalizing  $J$  (2) with  $S_T$  (19), the optimum robot geometry is  $a=2.0$ ,  $b=4.2$ ,  $c=4.0$  and  $\theta=103^\circ$  with a GII of  $0.247$ . Notice that the resulting optimum geometry is different from those in Figure 3 since its platform is not equilateral.

## 5 The Joint-Space Scaling Matrix: $S_J$

Unlike the task-space scaling matrix  $S_T$  which specifies relative end-effector capabilities, the joint-space scaling matrix  $S_J$  specifies relative actuator capabilities. Usually, actuators are chosen to satisfy the needs of the application so  $S_J$  can contain free design parameters with constraints that reflect practical limitations. Actuator scaling is particularly beneficial to serial manipulators which typically have diminishing torque requirements for actuators more distal to the base. Consequently, they will often produce dismal condition numbers since not including  $S_J$  is the same as setting it to the identity which implies symmetric actuation. Consider, for example, the 3-

DOF planar serial manipulator shown in Figure 4 which has three design parameters  $a$ ,  $b$ , and  $c$ .

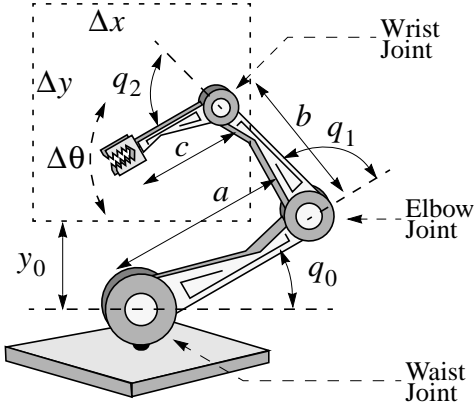


Figure 4: 3-DOF Planar Serial Manipulator

The workspace position  $y_0$  is fixed at  $5\text{cm}$  and the robot parameters  $a$ ,  $b$  and  $c$  are found to optimize the GII inside a square workspace ( $\Delta x, \Delta y = \pm 5\text{cm}$ ) at any angle ( $\Delta\theta = \pm 180^\circ$ ). Linear force requirements are made equal but many  $\bar{f}_{ang}/\bar{f}_{lin}$  ratios are explored to see how well the device is suited to different angular requirements. The results are shown in Figure 5.

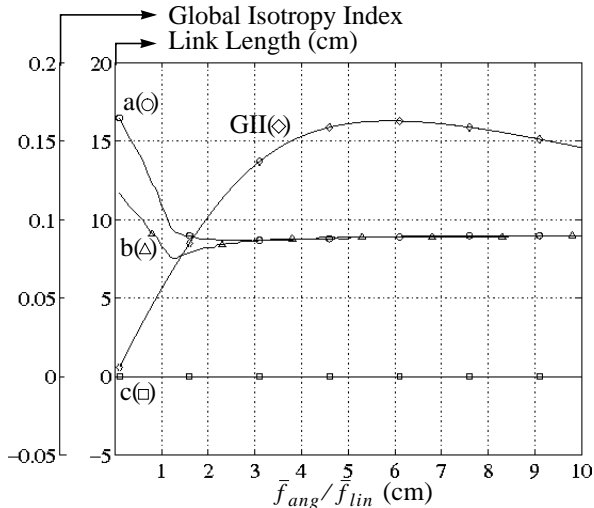


Figure 5: Optimum Serial Manipulator

The GII in Figure 5 peaks at  $0.163$  and drops to  $0.006$  at low  $\bar{f}_{ang}/\bar{f}_{lin}$  ratios. The device appears to be extremely unsuitable for applications with high force and low torque requirements. These poor results, however, are largely due to inappropriate actuation and are significantly improved by including the  $S_J$  matrix with free variables along the diagonal as shown in (20) where  $\bar{\tau}_{q_n}$  is the maximum torque capability of actuator  $n \in \{0, 1, 2\}$ .

$$S_J = \begin{bmatrix} \bar{\tau}_{q_0}/\bar{\tau}_{q_2} & 0 & 0 \\ 0 & \bar{\tau}_{q_1}/\bar{\tau}_{q_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Optimizing the serial robot again with the two additional free parameters in (20) produces the geometries and GIIs shown in Figure 6. Notice the improvement in GII values which now vary between  $0.17$  and  $0.28$  with stronger actuators at the  $q_0$  and  $q_1$  joints. The asymmetrically actuated device actually turns in its best results at low  $\bar{f}_{ang}/\bar{f}_{lin}$  ratios with as much as a 46 fold improvement over its symmetrically actuated counterpart.

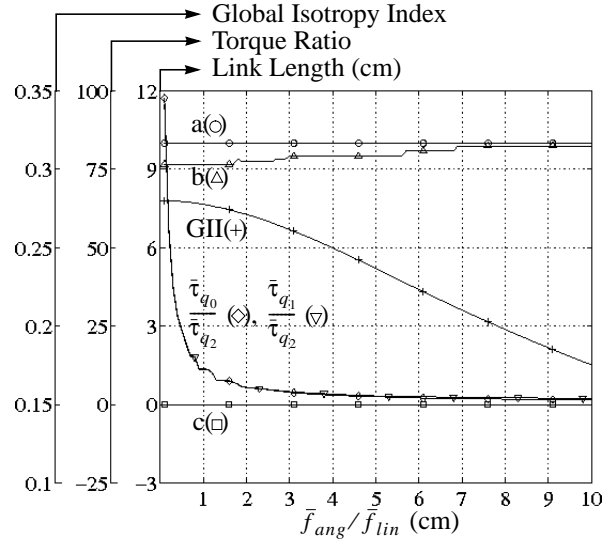


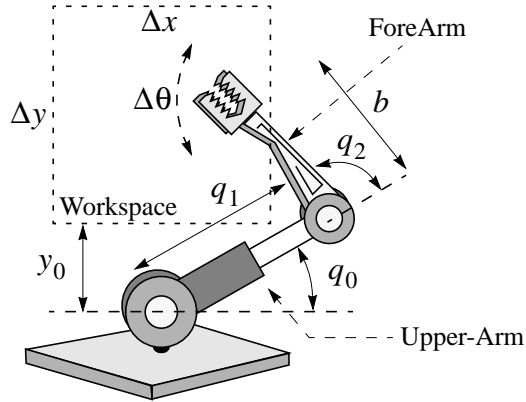
Figure 6: Optimum Scaled Serial Manipulator

However, actuator scaling can make  $a$  and  $b$  grow without bound since this is similar to positioning the workspace further from the base which improves isotropy. To avoid this, physical dimensions are constrained to  $10\text{cm}$  so that a fair comparison can be made with Figure 5. In practice, one might instead fix the actuator ratios due to practical constraints such as availability, size or weight. For example, using an elbow actuator that is twice as large as the wrist actuator, a shoulder actuator that is twice as large as the elbow actuator and the task-space requirements in (18), the optimum geometry is found using the  $S_T$  and  $S_J$  matrices shown in (21). The optimum geometry is  $a=9.6\text{cm}$ ,  $b=6.6\text{cm}$  and  $c=0\text{cm}$  with a GII of  $0.104$ .

$$S_T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix} \quad S_J = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

It also makes sense to fix  $S_J$  when it holds task-space quantities such as when it is used to normalize an impedance matrix (11). In cases such as this,  $S_J$  should be treated as a second  $S_T$  matrix and be fixed to reflect the demands of the application. For example, if the goal is to design a haptic interface to present impedance  $Z_E$  to its environment (a human hand),  $S_T$  should contain the ratios of maximum forces and  $S_J$  should contain the ratio of maximum velocities that could be expected of a human hand when acting in different directions.

Neither task-space nor joint-space units remain in the dimensionless, normalized Jacobian  $\hat{J}$ , so there are never dissimilar units, even if the manipulator has both rotary and prismatic actuators. Consider, for example, the 3-DOF serial manipulator shown in Figure 7 which has only one geometric design parameter  $b$ .



**Figure 7: 3-DOF Mixed Serial Manipulator**

For  $\Delta x, \Delta y = \pm 5cm$ ,  $\Delta \theta = \pm 180^\circ$ ,  $y_0 = 5cm$  and the force/torque requirements in (18), the optimum robot has  $b = 0cm$ ,  $\bar{\tau}_{q_0} / \bar{\tau}_{q_2} = 3.6$  and  $\bar{\tau}_{q_1} / \bar{\tau}_{q_2} = 0.9cm^{-1}$  with a GII of 0.203. This robot uses a similar shoulder:wrist actuator ratio (3.6:1) as the rotary serial arm (4:1) but produces a significantly (95%) better GII. In practice, however, it may suffer from added inertia due to its prismatic upper-arm.

While actuator scaling produces substantial improvements with serial manipulators, it is also effective with parallel manipulators, especially those that are geometrically asymmetric. The actuator scaling matrix in (20) used with the parallel manipulator in Figure 2 and the force/torque requirements in (18) produces the geometry  $a = 1.25cm$ ,  $b = 4.0cm$ ,  $c = 3.25cm$ ,  $\theta_0 = 96^\circ$ ,  $\bar{\tau}_{q_0} / \bar{\tau}_{q_2} = 2.1$ ,  $\bar{\tau}_{q_1} / \bar{\tau}_{q_2} = 1.0$  with a GII of 0.378. This is a 53% improvement over the GII of 0.247 obtained using identical actuators even though the geometries of the two devices are very similar.

## 6 Summary and Conclusions

By converting explicit joint and task-space values into diagonal 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. It is shown that multiplication by a "Characteristic Length" which has been done in the past to eliminate dissimilar physical units from task-space is a special case of the task-space scaling matrix  $S_T$ . The more general scaling matrices,  $S_T$  and  $S_J$ , remove physical units from both task-space and joint-space and make it possible to specify requirements for all task-space dimensions and scale actuators for improved isotropy. Optimization of the scaled, unitless performance matrix (i.e.  $J$  or  $D$ ) ensures that the robot actuators are fully utilized in satisfying the task-space requirements defined.

It is shown that although task-space scale factors greatly affect the optimum geometry, their effect on isotropy is less pronounced. This occurs because a device can often maintain consistent performance levels for a wide range of requirements as long as the geometry is adjusted accordingly. The elements of  $S_T$  should, therefore, be assigned to reflect the demands of the application.

The joint-space scaling matrix  $S_J$  is similar in nature to the task-space matrix  $S_T$  but describes relative actuator capabilities. It improves isotropy through asymmetric actuation which has been shown to be particularly effective with serial devices. Improvements in the isotropy index of up to two orders of magnitude have been observed as a direct result of actuator scaling.

## Acknowledgements

This work is 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 9<sup>th</sup> World Cong. Theory of Mach. & Mech., (Milano, Italia), 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. Doyle, "Analysis of Feedback Systems with Structured Uncertainties", IEE Proc., V. 129, Pt. D, No. 6, Nov. 1982.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] O. Ma, J. Angeles, "Optimum Architecture Design of Platform Manipulators", Proc. IEEE Int. Conf. Advanced Robotics, 1991.
- [9] 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.
- [10] M. Tandirci, J. Angeles, F. Ranjbaran, "The Characteristic Point and the Characteristic Length of Robotic Manipulators", Proc. ASME 22<sup>nd</sup> Biennial Conf. Robotics, Spatial Mechanisms & Mech. Sys. (Scottsdale, Arizona), V. 45, pp. 203-208, Sept. 13-16, 1992.