Sensor Based Motion Control and Coordination of a Redundant Manipulator

S. R. Malladi, K. P. Valavanis, M. C. Mulder

Intelligent Robotic Systems Laboratory
The Center for Advanced Computer Studies
The University of Southwestern Louisiana
Lafayette LA 70504

Keywords : Sensor-based, redundant, multi-jointed, manipulator, effort factor.

1. Introduction

This research work has been motivated by the challenge to derive and explore a novel methodology for sensor based control and coordination of redundant, multi-jointed robotic appendages performing simple and/or coordinated tasks in an obstacle filled environment. Three important attributes that have motivated the research work are: i) smooth, human-like appendage joint motion, ii) concentration on the appendage end-effector location and the real-time sensor measured position of all joints that contribute to that location, and, iii) the derivation and development of an inherent mechanism to avoid collisions among robotic appendages working within an obstacle filled common work space environment. The first attribute is important due to its simplifying **impact on exploring cooperative motion**, the second allows **reduction of the computational complexity of the control problem** (very important for real-time control) and the third exploits **cooperative motion of multiple appendages** accomplishing a common task in an obstacle filled environment.

Inverse kinematics of redundant manipulators is solved by two methods. The first method is based on the optimization of certain relevant criteria (such as minimizing time, energy, joint torques or joint motions; or maximizing manipulability) and the second method requires determining the pseudo-inverse (Moore-Penrose generalized inverse) of certain matrices. The first method has been used by: Vukobratovic and Kircanski [1] to minimize energy; Hollerbach and Suh [2] to optimize torque; Nakamura and Hanafusa [3] to optimize global redundancy control using Pontryagin's maximum principle; Chang [4] who used a Lagrange multiplier method and a minimization criterion. The second method has been used by: Klein and Huang [5] who used velocity control through pseudo-inverse; Varma and Huang [6] to find a minimum norm solution. Task-directed or prioritized solutions to resolve kinematic redundancy have also been presented by Seraji and Colbaugh [7], Long [8], Lee and Lee [9], and Yoshihiko [10].

The major limitations of the existing solutions are :

1. Pseudo-inverse based solutions involve too many numerical computations and typically introduce errors.

2. Integration has to be performed to obtain position of the end-effector and it is difficult to consider the workspace constraints.
3. Most of the methods are applicable to planar robots only.
4. All of these methods consider that each joint has only one degree of freedom (DOF).

The proposed approach overcomes such limitations. A detailed case study of a 4–joint, 6–DOF redundant manipulator, is presented in [11, 12].

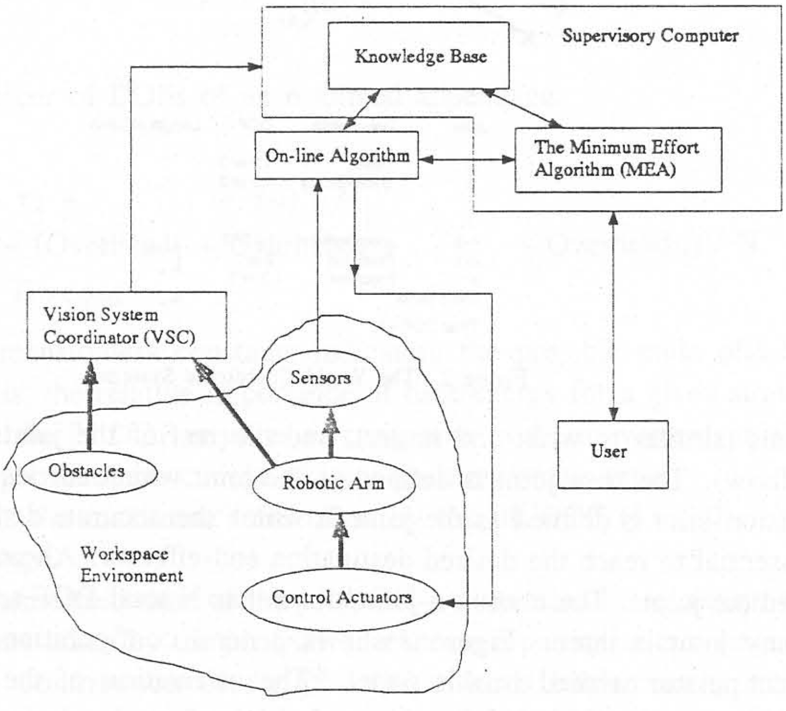
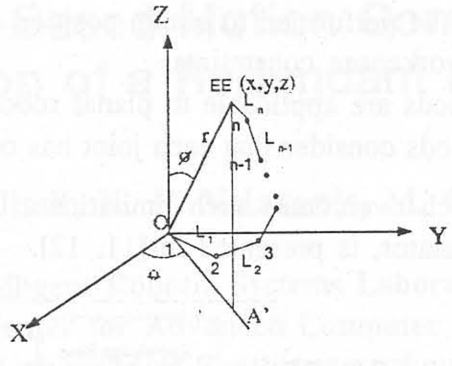


Figure 1. The Block Diagram of the Robotic System

2. The n-joint, N-DOF Redundant Manipulator

Given a **redundant** manipulator, the end-effector location (position and orientation) is most important; there are multiple joint (appendage) configurations of a redundant manipulator, which contribute to the end-effector location. Therefore, these joint configurations (joint movement strategies) are evaluated based on the total expended effort factor performance measure, which is a composite measure of time and energy during the execution of a movement strategy. The minimum effort control methodology generates several human like arm movement strategies and selects the best strategy on the basis of expendable effort. The methodology has an inherent basis to deal with obstacles in an efficient way, thus avoiding collisions. It is assumed that, at any time, the joint position, velocity and acceleration profiles are known through **sensor** readings (in some cases, the acceleration profile is computed). This **sensor based** approach significantly reduces the computational requirements for joint cooperative motion.

A block diagram of the overall robotic system is shown in Figure 1. The proposed methodology applies to any redundant manipulator. Let there be a total of n joints, where joint 1 is the base joint (similar to shoulder), the last two joints (i.e., n–1 and n joints) are



Joint	Description	DOF	Length of Limb
1	Base	1,2 or 3	
2	Intermediate	1,2 or 3	L_1
|	|	|	
|	|	|	
n-2	Intermediate	1,2 or 3	L_{n-2}
n-1	Precision	2 or 3	L_{n-1}
n end-effector	Precision	1,2 or 3	L_n

Total DOF = N

Figure 2. The World Coordinate System

precision joints (similar to wrist and finger), and the rest of the joints are intermediate joints (similar to elbow). The base joint is defined as the joint which carries the weight of the entire arm. A precision joint is defined as the joint in which the accurate determination of movement strategy is essential to reach the desired destination end-effector. All other joints are considered to be intermediate joints. The chosen n-jointed arm has N total DOF and the maximum number of DOF at any joint is three. Figure 2 shows general configuration of a n-jointed, N-DOF redundant manipulator studied in this paper. The orientation of the EE is determined after precisely positioning the EE of the arm (precise joints) at the desired destination. Consider spherical coordinates (ϕ, θ, r). The end-effector is at position (x,y,z) with respect to the world coordinate system. There is a one to one to correspondence between (x,y,z) and (ϕ, θ, r).

3. The Effort Factor Performance Measure

The EF is defined as the composite effort in terms of time and energy required by the robotic manipulator/appendage to implement and execute a movement strategy. It provides and shows possible trade offs between the energy consumed by each joint and the time required by each joint to execute a movement strategy.

The EF is a dimensionless performance measure. It is defined with the equation:

$$EF = \alpha 1 \sum_{j}^{N} |Ave.\tau - \tau j| + \alpha 2 \sum_{j}^{N} |Ave.Overhead - Overheadj| \; ; \quad j > 1$$

$$EF = \alpha 1 \tau 1 + \alpha 2 Overheadj \; ; \quad j = 1$$

When more than one movement strategies are available (which may possibly expend the same minimum effort), the problem becomes:

Minimize the EF subject to the minimization of the sum of kinetic energy expended by all joints, i.e.,

$$Min\ \{EF\},\ subject\ to$$
$$Min\ \left\{ \sum_{j=1}^{N} M_{effj} \times v_j^{\,2} \right\}$$

N: Total number of DOFs of an n-jointed appendage.
j: j^{th} DOF.

$Ave.\tau = (\tau_1 + \tau_2 + \ldots + \tau_N) / N$

Ave.Overhead = ($Overhead_1$ + $Overhead_2$ + ... + $Overhead_N$) / N

$\tau_j = Time_j \div Time_{jmax}$

α_1, α_2 : dimensionless constants indicating the possible trade offs between time and energy; that is, the relative importance of time/energy for a given strategy.

$Time_j$: time the j^{th} DOF requires to contribute to the movement; this is estimated from the open-loop calibration curves of joint position versus time.

$Time_{jmax}$: is the maximum time taken by the jth DOF (j = 1,2, . . ,N) to cover its maximum range.

τ_j: is the normalized time of the jth DOF, a dimensionless quantity, with respect to the maximum time taken by the jth DOF to execute a particular movement strategy.

$Overhead_j$: contribution of the j^{th} DOF towards implementing the movement; it is calculated based on the expended potential and kinetic energy.

All details about the EF performance measure may be found in [13].

4. The Minimum Effort Factor Algorithm

Based on the EF performance measure, the following algorithm has been derived:

1. **Determine the present end-effector location and update the motion profiles**: Receive sensor values from all joints. Compute the joint and end-effector positions. Based on the system operation time (within one iteration cycle), current joint position and end-effector location, update the motion, velocity, and acceleration profiles (trajectory profiles).

2. **Determine the end-effector (desired) destination and assess its reachability**: The required destination is determined and communicated by the vision coordination unit (in the present case, the user inputs the end-effector destination location). The necessary condition is that the destination be within the hemisphere defined by the base joint being the center with radius equal to the sum of the lengths of all limb segments. If this destination is unreachable, i.e., outside the hemisphere, send a message that the present end-effector destination is unreachable.

3. **Update obstacle information and determine the obstacle coordinates of distinct edges:** Information related to (geometrically shaped) obstacles is acquired with the help of the vision coordination unit (presently the user provides this information). An obstacle could have one of the following regular shapes: sphere, cone, cylinder, box, or pyramid. With respect to the base joint position (this is done because the base joint movement is initially determined), all distinct obstacle coordinates are computed.

4. **Compute the differences between the previous end-effector position and destination end-effector:** Let the desired destination end-effector be at θ_{dee}, ϕ_{dee}, r_{dee}. Compute the difference in θ, ϕ, and r, based on the present end-effector position (θ_{pee}, ϕ_{pee}, r_{pee}) and destination end-effector position (θ_{dee}, ϕ_{dee}, r_{dee}). Let the differences be $\Delta\theta$, $\Delta\phi$, and Δr, respectively.

5. **Check for trivial solution:** Check if $r_{dee} = \sum_{i=1}^{n} L_i$. If yes, go to step 6, else go to step 7.

6. **Unique strategy:** A unique strategy is possible. This requirement corresponds to the maximum stretch of all joints. The orientation is decided by θ_{dee} and ϕ_{dee}. Check if the movement of the arm within ϕ and θ range, interferes with that of the obstacles, and $r \leq \sum_{i=1}^{n} L_i$. If yes, send a message to the user that the destination is not reachable because of obstacles. Else, move the base joint by $\Delta\theta$ and $\Delta\phi$, and move all the rest of joints to full stretch configuration. Go back to step 1.

7. **Is the ith joint a precision joint?** If yes, do steps 200, 201 and go to step 250. Else, do steps 100, 101 and 102; then go to step 8.

8. **Is the entire feasible range explored?** If yes, go to step 250. Else, go to step 9.

9. **Select a movement strategy for the ith joint:** That is, the (i+1)th joint position is determined.

10. **Is the (i+1)th joint a precision joint?** If yes, do steps 200, 201 and then go to step 8. Else, do steps 100, 101 and 102.

11. **Is the entire feasible range explored?** If yes, go to step 8. Else, go to step 12.

12. **Select a movement strategy for the (i+1)th joint:** That is the (i+2)nd joint position is determined.

13. **Is the (i+2)nd joint a precision joint?** If yes, do steps 200, 201 and then go to step 11. Else, do steps 100, 101 and 102.

14. **Is the entire feasible range explored?** If yes, go to step 11. Else, go to step 15.

15. **Select a movement strategy for the (i+2)nd joint:** That is the (i+3)rd joint position is determined.

16. **Is the (i+3)rd joint precision joint?** If yes, do steps 200, 201 and then go to step 14.

Else, do steps 100, 101 and 102.

17. **Is the entire feasible range explored?** If yes, go to step 14. Else, go to step 18.

18. **Do steps 15, 16 and 17 until i = n–2.** Then do steps 200, 201 to obtain a strategy.

100. **Determine the obstacle coordinates with respect to the joint being considered as the origin**: Find the Cartesian difference from the joint to the desired destination end-effector.

101. **Determine the obstacle ranges, the nearest and the farthest points from the joint under consideration.**

102. If DOF = 1, Use Findfeasiblerange-I and Eliminateranges-I, to determine the feasible range avoiding obstacles. Else If DOF = 2, Use Findfeasiblerange-II and Eliminateranges-II, to determine the feasible range avoiding obstacles. Else If DOF = 3, Use Findfeasiblerange-III and Eliminateranges-III, to determine the feasible range avoiding obstacles. (Explained in detail in the next section)

200. **Fine movement strategy**: Solve the equations to precisely position the EE at the desired location. Movement required in joint(s) to orient EE to desired location is decided in this step.

201. **Check if this solution avoids obstacles**: If not, set Effort Factor (EF) to MAXINT. Else, Compute EF for this particular strategy.

250. **Select the minimum effort strategy.** Go to the Trajectory Synthesizer module.

Determination of the Joint Feasible Movement Range

Let 1, 2, ... , i-1, i, (for i < n) be the joints for which the joint movement configuration has already been decided. The (i+1)th joint feasible movement range is to be determined next. Consider that the (i+1)th joint is not a precision joint. If it is, then, a fine movement strategy is determined, instead of determining the joint movement feasible range. Each joint, but the base joint, may be considered to be the end-effector of the immediately previous joint. That is, the end-effector of the ith joint is the position of the (i+1)th joint; let this end-effector be called e_i, represented by $(\theta_{i+1}, \phi_{i+1}, r_{i+1})$ in spherical coordinates. Now the problem can be defined as follows: determine the feasible range of the (i+1)th joint movement to reach the desired destination EE. The situation can be explained using Figure 3

Let the (i+1)th joint movement feasible range in ϕ be $[\ \phi_{max}, \phi_{min}]$. To compute ϕ_{max} (as shown in Figure 3), the triangular law is applied (to determine an angle when the lengths of all sides are known) on the triangle formed with sides : 1) **0** —> e_i, length is r_{i+1}, 2) e_i —> EE, length is $\sum_{j=i+1}^{n} L_j$, and 3) EE —> **0**, length is r_{EE}, where **0** is the origin of the world coordinate frame. The initial guess for the feasible range of ϕ is from ϕ_{max} to ϕ_{EE}. This initial guess is dictated by the assumption that the multi-jointed appendage moves in a smooth anthropomorphic way (this initial guess provides ***righty*** configuration). Therefore, ϕ varies from ϕ_{EE} to ϕ_{max} to reach the desired destination end-effector. This is the feasible range of movement in the ϕ axis.

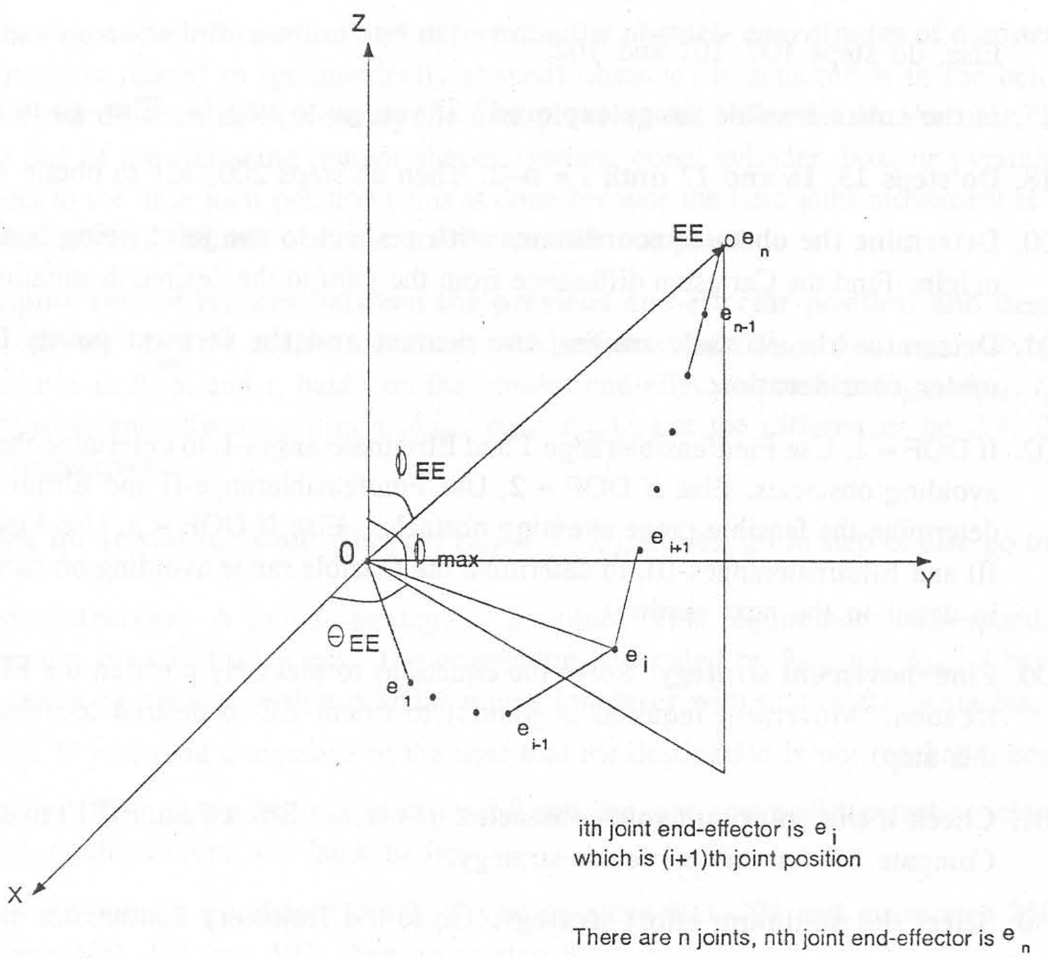


Figure 3. Determining the Feasible Range for the Movement of (i+1)th Joint

If the number of DOF of the (i+1)th joint being considered is 1, then the feasible range is obtained as explained above. If the number of DOF of the (i+1)th joint being considered is 2, for each ϕ in the range ϕ_{EE} to ϕ_{max}, the θ feasible movement range is determined. The initial guess for the (i+1)th joint feasible movement range in θ, is any value from θ_{i+1} to θ_{EE}. The exact feasible range of θ is determined by applying the modified binary search algorithm (presented in [11]) to the initially guessed range. In some cases, the closed space of ϕ and θ is parabolic as shown in Figure 4. The joint angle less than the desired end-effector angle (θ_{EE}) may lead to convex or concave shaped arm configurations, depending from which axis the angle is measured. The convention followed in this thesis, is θ from the x-axis and ϕ from the z-axis as shown in Figure 2. With this convention, the solution space for convex shaped arm configuration is selected for anthropomorphic motion (as shown in Figure 4).

If the number of DOF of the (i+1)th joint being considered is 3, one of the DOF is rotation or translation. In the case of a rotational joint, the rotation angle feasible range is determined at first. The (i+1)th joint rotational feasible range is computed with the help of modified binary search on the entire solution space (constrained by anthropomorphic movements); let this joint movement range be ψ_1 to ψ_2, to reach the desired destination end-effector. For each ψ in the

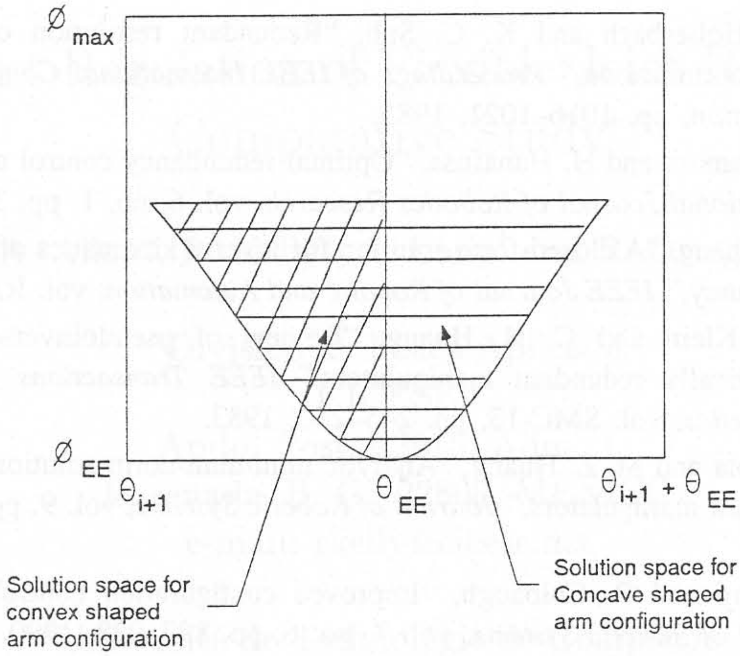


Figure 4. The Feasible Range of the (i+1)th Joint Movement in ϕ and θ Directions

above mentioned feasible rotational range, the ϕ and θ feasible movement ranges are computed as explained for the case in which the number of DOF of the joint being considered is equal to 2. If the third DOF in the (i+1)th joint is translational, the (i+1)th joint feasible movement range in the third DOF is computed at first, using the method described to determine the feasible range in the ϕ direction. Then, for each feasible movement strategy in the third DOF, the ϕ and θ feasible movement ranges are computed as already explained. For any joint i, which is not a precision joint, the above described method is applied to determine the joint feasible movement range.

5. Conclusion

Traditional performance measures used to resolve kinematic redundancy involve the computation of Jacobian and pseudo-inverse of Jacobian, thus requiring too many computations. These performance measures minimize either time or energy, but none of them considers the cooperation among the joints in the manipulator motion. Cooperation among the joints and consideration of time and energy during the execution of a movement strategy are emphasized by the proposed effort factor performance measure. Given a manipulator, the effort factor parameters can be determined based on the physical parameters, joint hardware limits of the manipulator, and open-loop position versus time characteristics. Since the EF quantizes physical stress and emphasizes cooperation among the joints, the EF is an excellent candidate for anthropomorphic (smooth) movements.

References

[1] M. Vukobratovic and M. Kircanski, "A method for optimal synthesis of manipulation robot trajectories," *Transactions of the ASME*, vol. 104, pp. 182–193, June 1982.

[2] J. M. Hollerbach and K. C. Suh, "Redundant resolution of manipulators through torque optimization," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1016–1021, 1985.

[3] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robotic manipulators," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.

[4] P. H. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," *IEEE Journal of Robotic and Automation*, vol. RA-3, pp. 393–403, 1987.

[5] C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, pp. 245–250, 1983.

[6] H. Varma and M. Z. Huang, "Analytic minimum-norm solution for rate coordination in redundant manipulators," *Journal of Robotic Systems*, vol. 9, pp. 1001–1019, December 1992.

[7] H. Seraji and R. Colbaugh, "Improved configuration control for redundant robots," *Journal of Robotic Systems*, vol. 7, no. 6, pp. 897–928, 1990.

[8] M. K. Long, "Task-directed inverse kinematics for redundant manipulators," *Journal of Intelligent Robotic Systems*, vol. 6, pp. 241–261, 1992.

[9] S. Lee and J. M. Lee, "Multiple task point control of a redundant manipulator," *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 988–993, 1990.

[10] N. Yoshihiko, H. Hideo, and Y. Tsuneo, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotic Research, Vol. 6, No. 2*, pp. 3–15, Summer 1987.

[11] M. C. Mulder and S. R. Malladi, "A minimum effort control algorithm for a cooperating sensor driven intelligent multijointed robotic arm," in *Proceedings of the 1991 IEEE conference on Decision and Control, Brighton, England, Vol.1*, 1991.

[12] S. R. Malladi, K. P. Valavanis, and M. C. Mulder, "On the validation of a sensor based robot control system using the minimum effort factor control approach," *Proceedings of the 31st IEEE International Conference on Decision and Control, Tucson, Arizona, U.S.A.*, December 1992.

[13] S. R. Malladi, K. P. Valavanis, and M. C. Mulder, "A minimum effort factor approach for sensor based control of a 4-joint 6-dof redundant manipulator," *Journal of Robotic Systems*, Fall 1993.