



<b>Title</b>	<b>A Jacobian-based Redundant Control Strategy for the 7-DOF WAM</b>
<b>Author(s)</b>	<b>Lau, HYK; Wai, LCC</b>
<b>Citation</b>	<b>Proceedings Of The 7Th International Conference On Control, Automation, Robotics And Vision, Icarcv 2002, 2002, p. 1060-1065</b>
<b>Issued Date</b>	<b>2002</b>
<b>URL</b>	<b><a href="http://hdl.handle.net/10722/46591">http://hdl.handle.net/10722/46591</a></b>
<b>Rights</b>	<b>Creative Commons: Attribution 3.0 Hong Kong License</b>

## A Jacobian-based Redundant Control Strategy for the 7-DOF WAM

H.Y.K. Lau and L.C.C. Wai

Department of Industrial and Manufacturing Systems Engineering  
The University of Hong Kong, Pokfulam Road, Hong Kong, PRC.  
Email: hyklau@hku.hk, lwai@graduate.hku.hk

### Abstract

The mapping between the Cartesian space and joint space of robot manipulators has long been a difficult task for redundant robots. Two main methods are used in the classical approach. One is by using direct kinematic inversion in the position regime; the other is to use Jacobian Transformation in the velocity regime. However, for a redundant robot, a non-squared Jacobian matrix is resulted when mapping between the two spaces. This results in using appropriate optimization algorithms to compute along with the Jacobian matrix. Taking the second approach, the Jacobian matrix for a redundant robot will be non-square. One approach to obtain a solution is to use pseudo inverse, this approach is however computational intensive. This paper presents a pragmatic approach by which a joint of a 7-DOF Whole Arm Manipulator (WAM) is initially fixed to facilitate the computation of the squared Jacobian matrix. Based on this approach, appropriate optimization strategies that are outlined in the paper, can then be applied to determine the optimal value of the 'fixed' joint in real time. Experiments are performed to verify the viability of this approach, and the results established that a robust and flexible Cartesian trajectory planning framework can be achieved for general redundant manipulators.

### 1. Introduction

One of the most challenging and important issues in modern robotics is to make robots more adaptable to the ever-changing unpredictable environment. In order to enhance the dexterity and sensibility of a robot, a large number of research studies have been undertaken in areas such as force control, redundant robots and their kinematics, obstacle avoidance strategies, etc. Among these studies, the solvability of the kinematics of a generic redundant manipulator is still a major hurdle to get over before a satisfactory implementation of control algorithm is achieved for a physical manipulator.

In general, the kinematics equation of a revolute manipulator in 3-D space can be represented by:

$$\begin{pmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{pmatrix} = f(\Theta) \in R^6, \quad \Theta \in R^7$$

The equation produces infinite number of solutions if the number of joints is greater than 6, this makes the determination of the redundant space being an essential building block for solving the inverse kinematics of redundant robots. Typically, to evaluate the inverse kinematics of a serial arm, two main methods are used. One approach is to find a closed form kinematic solution for the robot manipulator; the other approach is to use an appropriate Jacobian to resolve the inverse kinematics. For the first method, the robot should not be redundant and should behave classically. For example, a robot with six DOF should agree with the Pieper's principle [1]. In case of a redundant robot, the Jacobian Transformation method is a more viable approach to arrive at a practicable solution. Many inverse kinematics algorithms and optimization algorithms are based on the Jacobian transformations [2].

Despite the use of Jacobian Transformation method [3] is a typical approach for redundant robots [4, 5], the direct application on a real redundant robot will lead to a non-square Jacobian matrix, of which the problem of redundancy is still unresolved. In this paper, a pragmatic and more direct approach is used: by using a 6x6 square Jacobian matrix and fixing a sub-set of joint parameters, a pseudo-closed form solution is obtained momentarily so that other redundant strategies can be built upon. Despite the specific nature of this approach, the solution obtained enables a 7-DOF robot such as the WAM to be kinematically closed momentarily.

To implement and evaluate the performance of the proposed strategy, the WAM is used as the experimental platform. The WAM is a 7-DOF cable-driven robot that is designed to employ all the available manipulation surfaces of the robot to act upon and to sense the forces/torques resulted from interacting with its environment via direct joint torque sensing [6, 7].

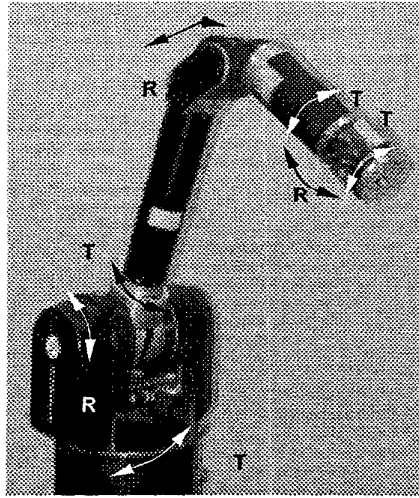


Fig. 1 The joint configuration of WAM

The WAM used in the experiment is controlled by a Pentium III machine with 256Mb memory. The robot and the computer is interfaced with a ServoToGo digital controller board under the real-time QNX operating system environment.

## 2. The Jacobian Approach

In solving for an inverse kinematics solution using the Jacobian approach, the Jacobian inverse is employed to provide a transformation from the Cartesian space velocity to Joint space velocity.

$$\dot{\Theta} = J^{-1}(\Theta)V$$

So, if  $\Theta \in R^n$ ,  $V \in R^m$  in which  $m \neq n$  then  $J^{-1}(\Theta) \in R^{n \times m}$  which is not a square matrix. In order to make a square Jacobian matrix,  $(n-m)$  number of joints must be determined before the Jacobian matrix is computed. Compare with using a non-square Jacobian method obtained for the WAM, the use of a square Jacobian method simplifies the overall computation as well as relieves the loading of the real-time control algorithm. The algorithm only needs to determine the  $(n-m)$  number of joints with an appropriate optimization algorithm. The use of a non-square matrix approach seems to be a simpler and more direct approach provided with the constraint of limited processing resources and tight cycle time.

To adopt the square Jacobian approach,  $(n-m)$  number of joints must be selected to take predetermined values for every control cycle in the computation of the inverse Jacobian. In the 7-DOF WAM case, Joint 3 is selected as the joint that is

pseudo-fixed and will take a momentarily stationary value that is known a priori.

One of the reasons that joint 3 is selected is because of the similarity between the WAM and the PUMA robot. The other reason of making Joint 3 a parameter is that the Joint 3 is the nearest joint to the elbow of the arm and it will detect most of the force acting on the elbow. This is particularly important consideration for the subsequent development of redundant control strategies that are based on force control at the elbow joints. In addition, this is important when the end effector is hung on freely, but not in contact with any other fixed objects. In these circumstances, the force that is sensed will be much localized at the first three joints, rather than at the base joints. In addition, Joint 3 has one of the largest joint ranges within the work volume of the WAM. With Joint 3 predefined for the square Jacobian method, other higher control layers such as the force control layer and the redundant strategy generation layer can then be built upon. In addition, the algorithm will be more directed. For example, by detecting the joint torque measured at Joint 3 and a control strategy is deployed to maintained a minimum force at this joint, one can make a simple obstacle avoidance algorithm base on the redundant elbow joint.

It is obvious that not only Joint 3 can be chosen as the redundant joint other joints can equally be suitable as the redundant joints for the Jacobian method. However, in the experimental studies, Joint 3 is chosen for convenience. In theory, any joint(s) can be chosen as the redundant joints.

## 3. Proposed Redundant Control Scheme

To implement the square Jacobian method, an appropriate trajectory planning scheme is constructed using the basis of a computer-based feedback controller. Suppose  $\Delta t$  is the cycle time of the digital controller.

Based on the follow equation:

$$\dot{\Theta} = J^{-1}(\Theta)V$$

The new set of joint angle can be computed as:

$$\Theta(t_{k+1}) \approx \Theta(t_k) + \dot{\Theta} \Delta t$$

Based on the approximation given above, the Jacobian path control scheme can be constructed. The control scheme must interact directly with the inner control loop. According to different performance requirements, the control scheme can either interact with the main control loop an in open loop manner or a closed loop manner, and they are implemented as in Figure 2 and Figure 3.

In the proposed scheme, various redundant control strategies can be built upon this layer, these include:

1. Force constraint control/obstacle avoidance (FCC/OA) scheme – The robot has a default joint

configuration for the redundant joint. (For example, the redundant joint angle is preset to 0 degree). A force control loop is then activated to monitor the force/torque sensed. If an obstacle is met, the reaction force acting on the robot will be higher than the specific threshold. Then the control algorithm will infer an obstacle is met, and the redundant strategy control algorithm will calculate a suitable joint configuration in order to avoid the obstacle.

a singularity can be determine by the computation of Jacobian inverse, and when

$$\det[J(\Theta)] = 0$$

the mechanism is at singularity [2], and this computation is already been executed within the normal within the normal control cycle in the Jacobian method. Thus the use of Jacobian approach provides an efficient means to perform singularity effect minimization and can further reduce the computation load of the control

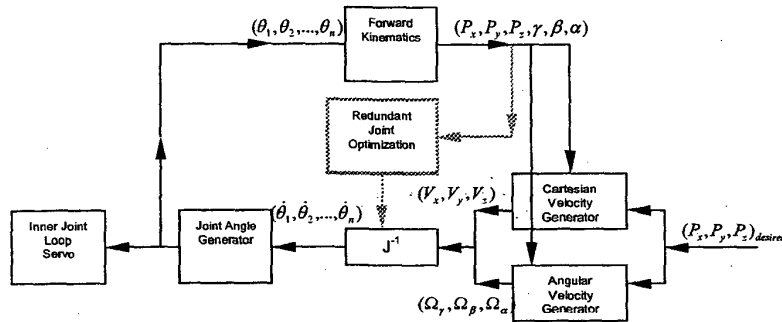


Fig 2: Open Loop Cartesian Trajectory Control Scheme

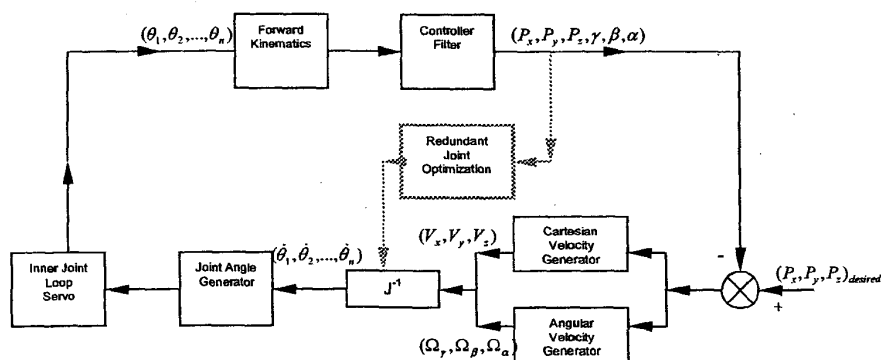


Fig 3: Close Loop Cartesian Trajectory Control Scheme

2. Singularity/Joint velocity effect minimization (S/JVEM) – singularity is a common physical problem of real robots, and this can be represented by the degeneration of the Jacobian matrix. Singularity will occur if two joint axes are aligned together. When the robot is approaching or passing near singularity. Some of the joints may generate very high velocities in order to maintain a constant Cartesian velocity. If the joint velocities exceed their physical limits, the controller may shutdown and such trajectories may not be achieved. To minimize the effect, the redundant strategy can compute and generate a set of suitable joint demand that can minimize the individual joint velocity, so that a higher Cartesian velocity motion can be achieved with an expanded the workspace in most cases.

An added advantage of the use of Jacobian approach in avoiding singularity conditions is that the reaching of

algorithm.

3. Mid-range joint angle control (MJAC) – Similar to singularity effect minimization, MJAC is used to balance the joint angle of all the joints and prevent some unnecessary reaching of joint limits during the execution of a trajectory. This strategy can maximize the usable workspace of a redundant robot and enhance the reliability of motion control.

4. Internal energy optimization (IEO) – The redundant strategy algorithm can also be used in optimizing the internal energies including the potential energy (joint configurations under the effect of gravity) and kinetic energy (joint speed) of the arm in motion.

The abovementioned policies can be selectively implemented or simultaneously used. However, when these are applied together, priorities must be stated in execution. A hierarchical classification method may be

adopted. In this hierarchy, the redundant strategies are divided into three main categories: the incident handler, incident prevention handler, and the optimized free-motion handler. In our scheme, the incident handler has the highest priority, which is followed by the preventive measures handler, and then the free-motion optimizer. Based on the four strategies stated above, their arrangement in this hierarchy is shown in Fig 4.

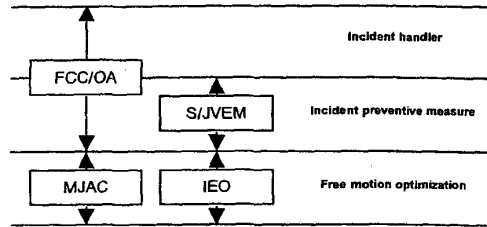


Fig. 4: Redundant strategy classifications

This redundant policy model can be implemented as a rule-based system gear around internal parameters of the robot such as joint torques, geometric constraints of the arm or with a fuzzy rule-based system. For a rule-based system, a simple if-then-else decision algorithm can be constructed by using joint torques and joint velocities as parameters:

$$\begin{aligned} & \text{if} (T_{\text{current}} > T_{\text{freeing max}}) \\ & \quad \Theta_{\text{pre}} = f_{\text{FCC/OA}}(\Theta) \\ & \text{elseif} (\dot{\Theta} > \dot{\Theta}_s) \\ & \quad \Theta_{\text{pre}} = f_{\text{S/JVEM}}(\Theta) \\ & \text{else} \\ & \quad \Theta_{\text{pre}} = f_{\text{IEO}}(\Theta) \end{aligned}$$

where

$$\begin{aligned} \mathbf{T} &= \text{Manipulator Joint Torque vector} \\ \Theta_{\text{pre}} &= \text{Predefined joint angles} \end{aligned}$$

Another implementation is to use geometric constraints such as joint limits and the inner and outer core limits of the robot to determine the appropriate algorithms to be used in different quadrants within the workspace. This is the most direct approach and required the least computation loading for the controller. However, the rule-based system will give a better-optimized performance than the geometrical constraints approach according to our experimental studies.

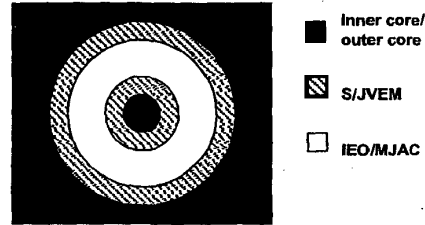


Fig. 5: Example of using geometrical constraints

Indeed, the performance of the rule-based system can be enhanced by the incorporation of fuzzy functions near the regions of transition of various strategies to prevent sudden 'jumping' of joint parameters.

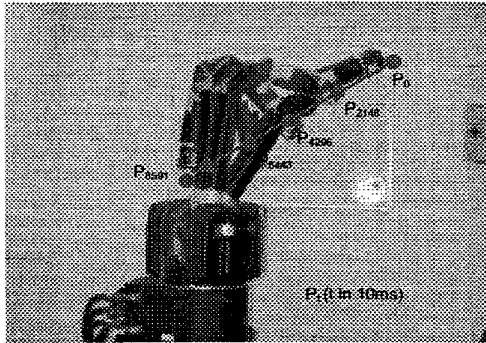
#### 4. Experiment Results

The square Jacobian method with joint 3 fixed has been implemented using a 4-DOF WAM system and making Joint 3 momentarily fixed in the Jacobian computation. This results in the following Jacobian matrix for the WAM:

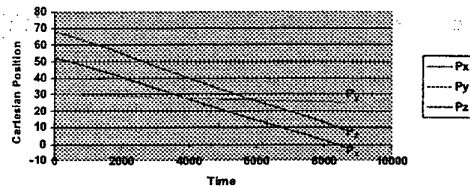
$$J(\theta) = \begin{pmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} & \frac{\partial P_x}{\partial \theta_4} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} & \frac{\partial P_y}{\partial \theta_4} \\ \frac{\partial P_z}{\partial \theta_1} & \frac{\partial P_z}{\partial \theta_2} & \frac{\partial P_z}{\partial \theta_4} \end{pmatrix}$$

where

$$\begin{aligned} \frac{\partial P_x}{\partial \theta_1} &= -(-s_1c_2c_3 - c_1s_1)c_4 + s_1s_2s_4d_3 + \\ & \quad [(s_1c_2c_3 + c_1s_1)s_4 + s_1s_2c_4](-L_4 - L_7) + \\ & \quad (c_1c_2c_3 - s_1s_1)d_4 + c_1s_2L_3 \\ \frac{\partial P_x}{\partial \theta_2} &= -(c_1s_2c_3c_4 - c_1c_2s_4)d_3 + (c_1s_2c_3s_4 + c_1c_2c_4)(-L_4 - L_7) - \\ & \quad (c_1s_2c_3 + s_1s_1)d_4 + c_1c_2L_3 \\ \frac{\partial P_x}{\partial \theta_4} &= -(c_1c_2c_3 - s_1s_1)s_4 - c_1s_2c_4d_3 + \\ & \quad [-(c_1c_2c_3 - s_1s_1)c_4 + c_1s_2c_4](-L_4 - L_7) \\ \frac{\partial P_y}{\partial \theta_1} &= -(c_1c_2c_3 - s_1s_1)c_4 - c_1s_2s_4d_3 + \\ & \quad [-(c_1c_2c_3 - s_1s_1)s_4 - c_1s_2c_4](-L_4 - L_7) + \\ & \quad (c_1c_2c_3 - s_1s_1)d_4 + c_1s_2L_3 \\ \frac{\partial P_y}{\partial \theta_2} &= -(s_1s_2c_3c_4 - s_1c_2s_4)d_3 + (s_1s_2c_3s_4 - s_1s_2c_4)(-L_4 - L_7) - \\ & \quad s_1s_2c_3d_4 + s_1c_2L_3 \\ \frac{\partial P_y}{\partial \theta_4} &= -(s_1c_2c_3 + c_1s_1)s_4 - s_1s_2c_4d_3 + \\ & \quad [-(s_1c_2c_3 + c_1s_1)c_4 + s_1s_2s_4](-L_4 - L_7) \\ \frac{\partial P_z}{\partial \theta_1} &= 0 \\ \frac{\partial P_z}{\partial \theta_2} &= -(c_1c_2c_4 + s_2s_4)d_3 + (c_1c_2s_4 + s_2c_4)(-L_4 - L_7) - c_2c_1d_3 - s_2L_3 \\ \frac{\partial P_z}{\partial \theta_4} &= -(s_1c_2s_4 - c_1c_4)d_3 + (s_1c_2c_4 + c_2s_1)(-L_4 - L_7) \end{aligned}$$



Change of Cartesian Position of straight line motion using Jacobian



Change of Joint Angles of straight line motion using Jacobian

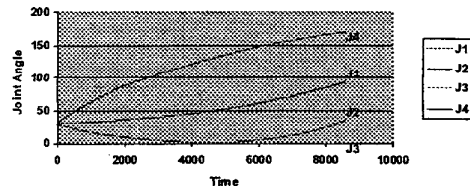


Fig. 6, 7, 8. Straight line motion control

With Joint 3 fixed, several linear trajectories are commanded at different velocities to evaluate the performance of the kinematics and control algorithms. For example, a straight path between points (52, 30, 67) to point (-1, 25, 9) (in cm) is commanded at a low velocity ( $9.186 \times 10^{-3} \text{ m/s}$ ), and it is observed that the WAM completes the trajectory successfully. The performance of the trajectory planner and the effectiveness of using Jacobian in solving redundant kinematics is established.

The performance of the Cartesian Trajectory Generator is proved to operate satisfactorily for a control cycle of 500Hz, and a Cartesian speed of 0.5 m/s.

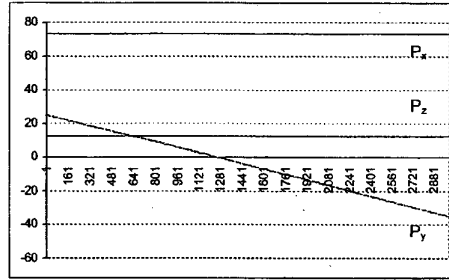


Fig 9a. Cartesian trajectory – coordinated motion of the redundant joint under explicit control (Y-axis is the world coordinate in cm, and the X-axis is the control steps)

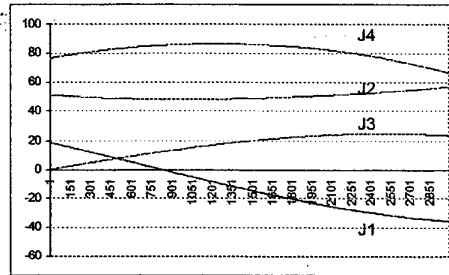


Fig 9b. Joint travels (Jt 1-4) – coordinated motion of the redundant joint under explicit control (Y-axis is the joint angle in degree, and the X-axis is the steps)

From the data obtained, it can be shown that the Jacobian method is effective for solving redundant kinematics problems in which a linear and stable behavior is observed.

An example of redundant strategy decision-making by using geometric constraint is also presented in the following experiment. Two policies are defined: Policy 1 is a policy with no explicit optimization, which means the redundant joint angle is fixed; Policy 2 injects a Sine wave motion to the redundant joint to simulate a continual change of geometric constraints.

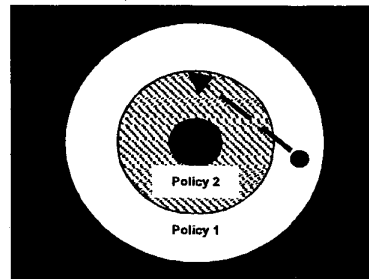


Fig 10. Cartesian trajectory of the WAM traversing the two different control policies

The robot is set to move from (72,0,13.5) to (0,55,23.5)(in cm) so that the end effector traverses the two regions so that the transition behavior can be observed.

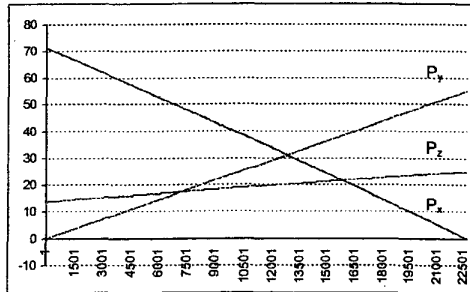


Fig 11a. Cartesian trajectory of the robot tool under the control of the two policies (Y-axis is the world coordinate in cm, and the X-axis is the steps)

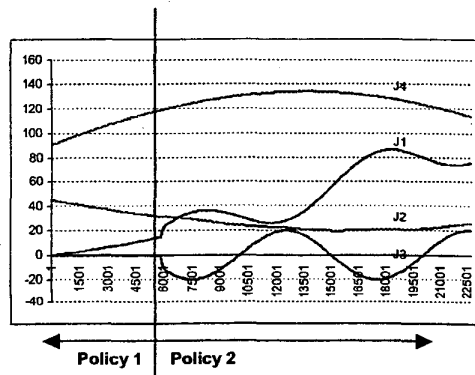


Fig. 11b. The variation of Joint angles under the control of the two control policies (Y-axis is the joint angle in degree, and the X-axis is the steps)

This experiment can be regarded as a simulation that is performed to observe the performance of redundant strategy when two or more optimization algorithms are deployed in a single trajectory. From the result obtained, the motion did not show any irregularities when the policy is changed from one to the other.

## 5. Conclusion

Jacobian method has long been used as a common method for tackling the inverse kinematics problem. However, compare with the analytical method, Jacobian method is less computational economical especially when computation resources are restricted.

Jacobian method has some advantages over the analytical method, especially in solving redundant kinematics problems. It provides a fast track for formulating a serial inverse kinematics problem as in

the case of the WAM. Also, Jacobian method is flexible to allow most optimization algorithm to build upon. With these advantages, Jacobian method promotes various optimization algorithms to build upon, thus a robust and flexible redundant control strategy mechanism must be constructed to effectively switch the appropriate algorithm that is most suitable to control the arm under different situations. In this paper, a hierarchical redundant strategy control scheme is presented for this purpose and experiments are carried to demonstrate the performance of these strategies. Based on our experimental studies, it is expected that such hierarchical-type strategy control scheme can be successfully deployed in real-world redundant robot controllers.

## References

- [1] J.J. Craig, Introduction to Robotics: Mechanics and Control, Addison-Wesley, 1985.
- [2] K. Lesser and W. Townsend, "Control and Exploitation of Kinematic Redundancy in Torque-Controllable Manipulators via Multiple-Jacobian Superposition", *Proc. Int. Conf. On Field and Service Robotics*, 1997.
- [3] W. Kaplan, Advanced Calculus, Addison-Wesley, 1983.
- [4] Y. Nakamura, Advanced Robotics: Redundancy and Optimization, Addison-Wesley, 1991.
- [5] V.R. Dubey, J.A. Euler, and S. M. Babcock, "Real-time implementation of optimization scheme for seven-degree-of-freedom redundant manipulator", *IEEE Trans. Rob. Auto.*, vol. 7, no. 5, pp. 579-588, 1991.
- [6] K. Salisbury, W. Townsend, B. Eberman, D. DiPietro, "Preliminary Design of a Whole-Arm Manipulation System (WAMS)", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 254-260, 1988.
- [7] W. Townsend and J. Guertin, "Teleoperator slave - WAM design methodology", *Industrial Robot*, pp. 167 - 177, 1995.