8-2005

A Cartesian Space Approach to Teleoperate a Slave Robot with a Kinematically Dissimilar Redundant Manipulator

Hariharan Ananthanarayanan
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Hariharan Ananthanarayanan entitled "A Cartesian Space Approach to Teleoperate a Slave Robot with a Kinematically Dissimilar Redundant Manipulator." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

William. R. Hamel, Major Professor

We have read this thesis and recommend its acceptance:

Gary. V. Smith, Vijay Chellaboina

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Hariharan Ananthanarayanan entitled " A Cartesian Space Approach to Teleoperate a Slave Robot with a Kinematically Dissimilar Redundant Manipulator." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

William. R. Hamel

_____

Major Professor

We have read this thesis
And recommend its acceptance:

Gary. V. Smith

_____

Vijay Chellaboina

_____

Accepted for the Council:

Anne Mayhew

_____

Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records)

A Cartesian Space Approach to Teleoperate a Slave Robot with a Kinematically Dissimilar Redundant Manipulator

A Thesis
Presented for the
Master Of Science Degree
The University Of Tennessee, Knoxville

Hariharan Ananthanarayanan
August 2005

Dedicated to my beloved parents

and my hard work!

Acknowledgement

As per a famous quote in the Hindu religion, Mother, Father, Teacher and God is the order in which a person has to pay homage to. It is with immense gratitude and love that I thank my parents for the incessant support and love they have extended me over the years. I am indebted to them for the confidence they laid on me, which made me reach greater heights. My parents personify sacrifice and love. My two brothers have mentally and financially supported me during each stage of my stay in the United States. I take this opportunity to acknowledge their support. Needless to say, this work would not have been possible for me to accomplish without my teacher and mentor Dr. Hamel. The support he extended for the past two years in identifying and bringing out my research skills is unparalleled. His intellectual and practical skills astonish me to this day. I sincerely thank him for his patience in tolerating my inexperience in research and for his confidence and faith in me, which led me to this stage. He has lived up to my expectations for a teacher. God helps those who help themselves and we both have done our duties.

The research group at REMSL too has been supportive in all my endeavors. A special thanks to Robin who has come to my rescue during all crises. Prem is a friend, a companion, an entertainer and a competitor. His presence had always been an encouraging and entertaining one. I value friendship more than anything else in the world. My friends Ravi, Balaji, and Damu have given me the confidence and courage to take up this task, with their constant enquiries and solutions. Last and definitely not the least the “G” factor has had a great influence in my life. With his constant enthusiasm, egoism and never-ending confrontations with me has open many doors for me in life. I take this opportunity to thank every other person, not mentioned, who has contributed directly or indirectly for my success here.

Abstract

Due to the inability of humans to interact with certain unstructured environments, telemanipulation of robots have gained immense importance. One of the primary tasks in telemanipulating robots remotely, is the effective manipulation of the slave robot using the master manipulator. Ideally a kinematic replica of the slave manipulator is used as the master to provide a joint-to-joint control to the slave. This research uses the 7-DOF Whole Arm Manipulator© (WAM) as the master manipulator and a 6-DOF Titan as the slave manipulator. Due to the kinematic dissimilarity between the two, a Cartesian space position mapping technique is adapted in which the slave is made to follow the same trajectory as the end effector of the master with respect to its reference frame. The main criterion in undertaking this mapping approach is to provide a convenient region of operation to the human operator. Various methods like pseudo inverse, Jacobian transpose and Damped least squares have been used to perform the inverse kinematics for the Titan. Joint limit avoidance and obstacle avoidance constraints were used to perform the inverse kinematics for the WAM and thereby remove the redundancy. Finally a joint volume limitation constraint (JVLC) was adopted which aims at providing the operator, a comfortable operational space in union with the master manipulator. Each inverse method for the Titan was experimentally tested and the best method identified from the simulation results and the error analysis. Various experiments were also performed for the constrained inverse kinematics for the WAM and results were simulated. RoboWorks© was used for simulation purposes.

Contents

List Of Tables

List Of Figures

Chapter 1

Introduction

On the road to achieve technological advancements, humans have been posed with great challenges over the past few decades. The inability of humans to physically execute tasks in various specific environments has been the greatest challenge of all. There are many places in this world, which are inaccessible to humans, due to various reasons. Dismantling of radioactive sites **[Ham01]**, extraterrestrial explorations, under water applications, oil drilling, tele-surgery are some of the tasks in which the environment is unsuitable for human intervention. As a result in the above cases and in many other applications, a robot being teleoperated from a remote location is performing the task. This is typically made possible by a robot capable of performing various physical manipulations. Most of these tasks are variable in the sense of the actual location and the kind of action to be performed. An ideal system to perform these tasks at the remote site would be one equivalent to a human, which can recognize what tasks to be performed, and which has the physical flexibility as humans. Since most systems performing remote tasks are far beyond idealism, they require human intelligence to command the robot as to how to proceed with the tasks. So a combination of human and robot system should be used to perform such tasks. Since it is either infeasible or impractical to send the human directly to the remote site, the human performs the task indirectly through a remotely operated machine. This is popularly referred to as telemanipulation. The basic idea behind telemanipulation is to avoid human presence and at the same time to enable robots perform human like activities with human assistance. This is made possible by visual and tactile feedback. This kind of system enables interaction between humans and environment without direct contact between them. The feel of the environment is preserved and transferred to the human operator. There are many issues involved with teleoperating robots at remote locations, such as telemanipulation of the slave manipulator, time delay between the master and slave

motions, dynamics of the system and the human operator and certain other. Of all these, the compatibility between the master and slave manipulator is taken into task in this research and an approach is attempted to imitate the motions of the master manipulator in Cartesian space.

1.1 Motivation

One critical issue involved in telemanipulation is the control of the slave manipulator interacting with the environment. The one at the human side used to control the slave is called the master. In most of the applications a simple slave manipulator is being employed to carry out the tasks and in majority of those cases a kinematically similar master is used to control it. Ideally this manipulation system should be capable of high dexterity and flexibility in taking up the commands from the operator. This is enabled through mounting sensors at various locations, which would sense and feel the environment. A video camera might capture the physical movement of the robot and feed it back to the operator interface to enable the operator to make decisions based on the current state of the system. A force/torque sensor mounted on the end effector (EE) of the manipulator sends information about the force interaction between the robot and the environment. Teleoperation is suitable for performing tasks in environments, which have low accessibility for human presence and high variability, where the variability refers to changes in the teleoperation environment from one instance of performing the teleoperation to another. The teleoperation environment can change because of many reasons: the teleoperation site can change, the location of the objects in the environment might change from one instance to another even if teleoperation is performed on the same site, or the task changes from one instance to another. In contrast, industrial robots perform very similar routine physical manipulation tasks with low variability. The specified task procedures are executed a large number of times, within a very controlled environment. The robotic task space can be organized such that the automated process can also be carried out successfully. The position of objects to be manipulated also

usually remains fixed from one operation cycle to another and if a correction is needed, the process can be interrupted by an operator. In contrast, since most teleoperation tasks have a high variability, it makes them unsuitable to be performed by completely automated systems, as is the case in industrial robotics. Teleoperation tasks need the additional capability of the human operator to understand and modify the task.

Another characteristic of teleoperation tasks is the degree of structure in the remote environment. It refers to the degree of certainty with which the task and its environment at the remote site are "known" at the local site. Environments about which there is only a little previous knowledge available are referred to as unstructured environments. This characteristic has important implications for the extent to which remote task can be automated. A teleoperation task requires the control of the end effector of the remote manipulator, which might have different degrees of freedom. There are a couple of ways of performing this, one is the direct joint control and the other is the control of the end effector position and orientation. There are various other issues involved in carrying out a stable teleoperation as well.

A teleoperation system requires various components and knowledge to perform a given task. To begin with, the remote manipulator should have added compliance to carry out even the minor change in position of the master. Various methods of communication are adopted to transfer the data to and from the slave, depending on the application. Numerous sensors are used to project the environment to the operator and the data from these sensors analyzed for its information about the remote environment. Of all this, the most important part of the teleoperation system is the operator interface which decides the extent to which the operator can sense the remote environment and consequently control the slave robot. The information provided to the operator should be informative about the environment. The controller should be such that the operator can effectively control the slave manipulator.

In presenting the environment to the operator, every teleoperation system tries to achieve a transparency state where the operator feels his/her virtual presence in the operational site. Various teleoperation methods have been adopted over the years to achieve this state. Various feedback systems have been analyzed to make every telemanipulation ideal and transparent. All the above stated issues are attributed towards effective control of the slave robot, which makes the teleoperator one of the most important constituents of any telerobotic system. In order to stay within the scope of this work let us take up the critical task of physically controlling the slave robot with the teleoperator.

A teleoperator is a device, which enables human to manipulate objects remotely Fig 1.1. It also enables certain functionalities, which could not be done alone by either people or robots. It mainly consists of

a. A master which the human manipulates with.
b. A slave, which imitates the movements of the master.
c. A controller, which communicates between the master and the slave

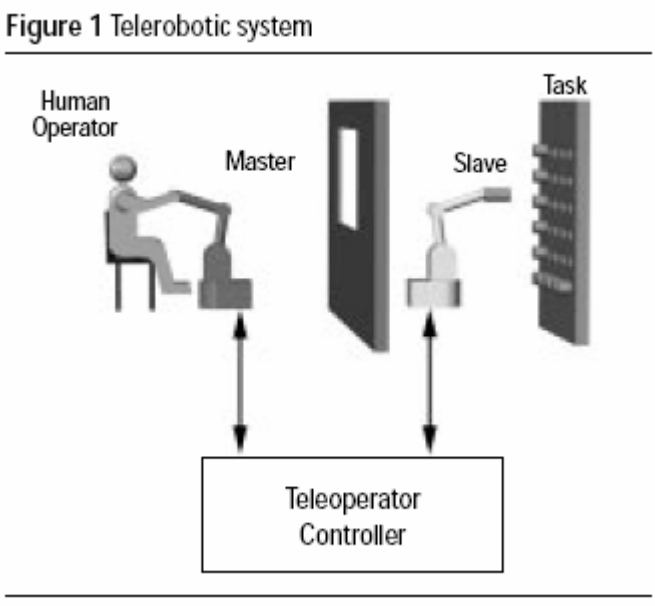

Fig 1.1 A Typical Telerobotic system

Irrespective of the complexity of the master, the human interacts with it by grasping a structure at its end and the slave interacts with the environment with the tool tip. The main purpose of the master is to transmit the movements of its end-tip to that of the slave despite its kinematic dissimilarity with the slave manipulator. In a teleoperator system, the kinematics of the master and its corresponding slave may be identical, scaled or totally different. The current system uses a master, which is kinematically dissimilar from that of the slave.

The master manipulator is a 7 DOF back-drivable Whole Arm Manipulator with human-like kinematics Fig 1.2. It has a centralized cable system mounted at its base, which transmits power to all other joints through cables **[Tow99]**. It is capable of being back-driven since it uses pulleys and cables for power transmission. It has inbuilt sensors to measure angles, forces and torques at every joint. This force and torque measuring capability enables it to feedback forces from the slave to the human operator. Force feedback is not of interest for this work as the master tries to command the slave in terms of the Cartesian co-ordinate of its end effector. The WAM is a redundant robot with an extra degree of freedom at the elbow.

Fig 1.2 Master manipulator WAM

Usually the redundant joint is used constructively in avoiding obstacles. Since WAM is the master the redundant DOF does not matter much. The slave robot is a 6-DOF Titan manipulator from Schilling Robotics. It was primarily designed and manufactured for under water applications. It comes with a conventional parallel jaw gripper at the end of the sixth joint. It is hydraulically powered. It has built in encoders for each joint, which gives the value of each joint at every instant. It is a non-back drivable manipulator since it is powered hydraulically. The seven functions that the manipulator can perform are Yaw at the azimuth, Pitch at the shoulder, Pitch at the elbow, Pitch at the wrist, Yaw at the wrist, wrist Rotation and Jaw open and close. With these seven functionalities the Titan has a wide range of operability.

The task of instructing the slave robot to perform defined tasks as defined by the operator on the master is identified as the primary task in a teleoperation system. Manipulating or teleoperating a robot in an unstructured environment greatly depends on the type of teleoperator used and the type of mapping between them. In certain cases the teleoperator may be a kinematic replica of the slave and in certain other might be kinematically dissimilar. In the earlier case the control may be at the joint level, transferring the joint angles from the master to the slave. In the latter case the mapping is made easier in the Cartesian space (the position of the EE).

In the current work since the master and the slave are kinematically dissimilar, a Cartesian co-ordinate mapping is undertaken for the effective telemanipulation of the slave. That is the basic idea is to achieve spatial correspondence between the two. In this approach, forward kinematics is performed at the master side and inverse kinematics at the slave side. The forward kinematics transforms the joint angles of the master to its EE positions. At this stage a scaled mapping is performed from the velocities of the master to the velocities of the slave. The inverse kinematics then transforms the EE position to joint angles. Later these results are tested in a simulation package, where the trajectory of the slave is verified against the trajectory of the master.

1.2 Thesis Outline

This thesis explains a method to achieve spatial correspondence between the master and the slave manipulator irrespective of the kinematic dissimilarity between them. That is, the position and orientation of the two end effectors are mapped in the Cartesian co-ordinate space. Future chapters explain the D-H parameters, kinematic transformations for both the manipulators and also the formulation of the corresponding Jacobian. The forward and inverse kinematics is performed to achieve the end effector position and orientation. End effector velocities of the two manipulators are mapped in order to achieve spatial correspondence. It means that since the slave manipulator has a longer reach and bigger workspace, the end effector position is just a scaled up value of that of the master with respect to their respective reference frames. In doing so various constraints on the slave manipulator kinematics and dynamics are taken into consideration. The main criterion was that, the slave should be continuous and reliable in imitating the movements of the master within its workspace.

The inverse kinematics was done through many methods including Jacobian inverse method though it failed at singular points. Singularities occurred when the slave tried to move outside its operational space and also at certain points within the workspace. To avoid singularities within the workspace and to provide a good EE tracking error a Damped Least Square (DLS) method was adopted which would always keep the desired and actual target point close to each other. Though this method is more effective in case of autonomous operation it does not affect the performance much in teleoperation as the operator always has a visionary feedback. Various constraints were introduced to resolve the redundancy of WAM and thereby perform the inverse kinematics. Finally the best constraint was identified therough error analysis and simulations. A RoboWorks model of both the master and the slave are made to simulate

the trajectories of both to analyze its performance. This simulation takes in joint variables as input and executes the trajectory based on the kinematics of the manipulator.

1.3 Contribution of the Work

As this research uses a Cartesian space approach to map the position and orientation of the EE, it can be used for a teleoperator system with any two kinematically dissimilar manipulators irrespective of the number of degrees of freedom available at the master side, though there is a trade-off with the available workspace to operate on the master side. The DLS method is a powerful tool to perform the inverse kinematics, which avoids complex inverse calculation and saves operation time. This enhances the performance of the system in terms of reduced time delay. The joint volume limitation constraint is unique for the master manipulator used. The redundancy and the mounting of the WAM in the operators' console posed a great challenge in providing the operator with a comfortable workspace in union with the manipulator. This approach attempts to solve one of the major issues in the current teleorobotic system, as it provides a practical solution to control a slave robot using a redundant manipulator. This logic can be extended conveniently to a higher degree redundant manipulator. Also this method can be conveniently used to control a much simpler slave robot than the one used now. It provides a very low EE tracking error for the slave and the master in addition to which satisfies the constraint on the master. To sum up, the Cartesian space position mapping in addition to the two inverse kinematics methods provides a reliable and continuous teleoperation in the range of operation of the human operator.

Chapter 2

Kinematics

Kinematics is the science of motion without giving regard to the forces that cause them. A manipulator can be considered as a chain of kinematic links from a mechanical standpoint. Typically a manipulator is made of revolute and prismatic joints. A revolute joint is subject to pure rotation and a prismatic joint is subject to pure translation. One end of the kinematic chain is fixed to a base and an end effector is mounted at the other end. The effective movement of the end effector is due to the relative movements of the joints. There are two types of kinematic chain: Open and Closed. If the series of joints lead finally to the end effector then it is an open kinematic chain. If the kinematic chain ends back at the base it is a closed kinematic chain, which is unpopular for use. In order to manipulate an object in space the position and orientation of the end effector has to be defined with respect to its base. The locus of points which could be reached by the end effector in 3D space defines the workspace of the manipulator. Six co-ordinates, three for position and three for orientation are required to define a point in space. The final position and orientation of the end effector is represented as a function of its joints. Therefore a manipulator with more number of degrees of freedom than required to define a point in space is called a redundant manipulator. The task of determining the position and orientation from the joint angles is called the Forward kinematics and the task of determining the joints angles from the position and orientation is called the Inverse kinematics.

The forward kinematics always has a unique solution for a given configuration of the manipulator, whereas the Inverse kinematics is not unique at all times. There exist singularity cases where the inverse kinematics fails and in case of redundancy infinite

solution exists. The key to perform the forward kinematics is the homogenous transformation matrix. Chapter 2 of **[Sci00]** gives an introduction to basics of defining the position and orientation of rigid body, rotation matrices and its properties, Euler and Roll-Pitch-Yaw angles and homogenous transformations. However an important convention to be followed to obtain the transformation matrix is the Denavit Hartenberg (D-H) convention. The D-H convention **[Den55]** defines a transformation from one joint to another and finally the transformation between the end effector and the base of the manipulator.

2.1 Denavit Hartenberg Convention

In order to determine the transformation between the tip and the base of the manipulator, a general method has to be followed to define the relative position and orientation of two consecutive links. The challenge lies in attaching two frames to the two links and finding the transformation between them. This is followed from the base to the end effector and a final (4×4) transformation matrix is obtained to define the EE with respect to the base for an *n* degrees of freedom manipulator.

In compliance to the Fig 2.1 let axis *i* be the axis along the joint connecting the link *i-1* and link *i.* The frame *i* is defined according to *Denavit Hartenberg Convention* **[Sci00]** as

- Select axis z_i along the axis of joint $i+1$
- Locate the origin of the frame at the intersection of axis z_i and the common normal to the axis z_i from the axis z_{i-1}.
- Choose the axis x_i along the common normal between the axis z_i and z_{i-1} and in the direction from Joint *i* to Joint *i+1*.
- Choose axis y_i so as to complete the right-handed frame.

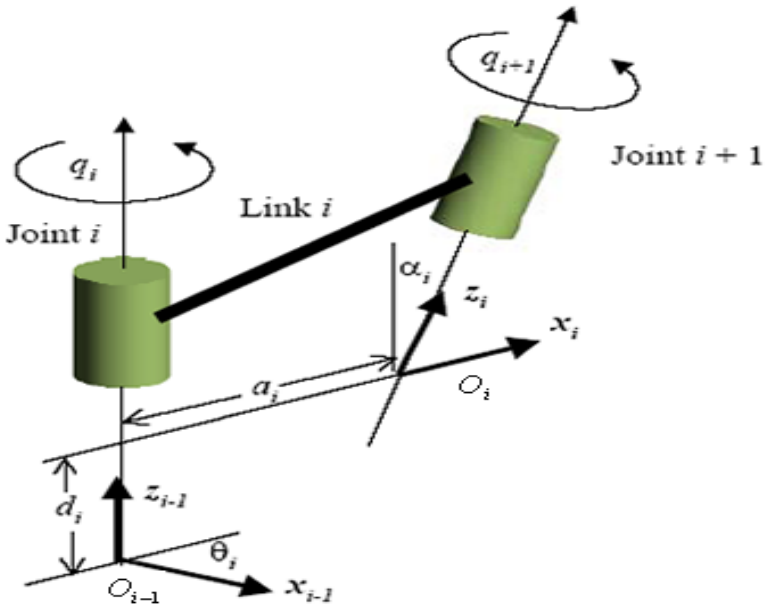

Fig 2.1 Co-ordinate frames and joint parameters

Though the Denavit Hartenberg Convention provides the definition for a frame there are certain ambiguities involved in it. They are

- For the first joint, only the axis z_0 of Frame 0 is defined and the location of O_0 and x_0 are arbitrary.
- For the last joint *n*, since there is no joint *n+1*, z_n is not uniquely defined but x_n should be perpendicular to the axis z_{n-1}.
- When two consecutive joints are parallel then the common normal between them to locate the origin is not specifically defined.
- When two consecutive joints intersect then the direction of axis *x* is arbitrary.
- For a prismatic joint the direction of z_{i-1} is arbitrary.

After having laid the link frames the transformation from one frame to the other can be established using four DH parameters.

1. a_i, the distance between the *z* axis of both the frames(z_i and z_{i-1}).

2. α_i, angle between the axis z_i and z_{i-1} about the axis x_i. Angle is positive for counter clockwise rotation.

3. d_i, the distance between the *x* axis of both the frames(x_i and x_{i-1}).

4. θ_i, the angle between the axis x_i and x_{i-1} about the axis z_{i-1}. Angle to be taken as positive for counter clockwise rotation.

In all the cases two of the four parameters a_i and α_i remains constant depending on the geometry of the manipulator. Of the remaining two only one parameter vary depending on the type of the joint.

- If the joint is revolute the variable is θ_i
- If the joint is prismatic the variable is d_i.

In order to express the transformation between Frame *i* and Frame *i-1* the following steps are followed.

- Choose a frame aligned with Frame *i-1*.
- Translate the frame by a distance of d_i along the axis z_{i-1} and rotate it by an angle θ_i about the z_{i-1} axis. This will align the original frame with frame i' and the transformation is given by

$$A_{i'}^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Translate the frame aligned with i' by a distance a_i along the axis $x_{i'}$ and rotate it by an angle α_i about the axis $x_{i'}$. This will align the current frame with frame *I* and the transformation is given by

$$A_i^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final transformation matrix between the two frames is given by

$$A_i^{i-1}(q_i) = A_{i'}^{i-1} A_i^{i'} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.2 Direct Kinematics

As stated earlier a manipulator is constituted of many links connected to each other in series. The position and orientation of the tip of the manipulator is defined with respect to the base in Cartesian co-ordinates. Direct Kinematics determines this position and orientation as a function of the joint variables, which uniquely define the configuration of the manipulator. The position is given by the co-ordinates of the origin of the tip and the orientation is given by the unit vectors of the frame at the tip. It has been stated that the final transformation can be represented in a (4×4) matrix with the first (3×3) elements giving the rotation matrix and the fourth column giving the position.

$$T_e^b(q) = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where *n, s, a* are the unit vectors attached to the frame of the end effector and *p* is the position of the origin of the frame from the base. The vector *a* is in the directionof approach of the end effector, *s* is normal to *a* in the sliding direction and *n* completes the right-handed frame. The position can be given by a minimal number of co-ordinates and

the orientation can be given by the minimal representation (Euler Angles) describing the rotation of the end effector with respect to the base frame. Therefore the posture of the end effector can be given by

$$x = \begin{bmatrix} p \\ \phi \end{bmatrix}$$

where p is the position and ϕ is the orientation. X is an $(m \times 1)$ vector with $m \leq n$ (number of degrees of freedom). The space defined by x is called the operational space and on the other hand the joint space defines the $(n \times 1)$ joint variables.

$$q = \begin{bmatrix} q_1 \\ . \\ . \\ q_n \end{bmatrix}$$

2.3 Inverse Kinematics

Inverse kinematics is the determination of the joint variables for a given configuration of the manipulator i.e., the position and orientation of the end effector. This is a very important task in commanding a manipulator to execute a motion as position defined in the operational space are transformed into joint space variables. Generally the Inverse kinematics equations are non-linear unlike direct kinematics. Due to their nonlinearity

- Multiple solutions exists
- Infinite solutions in case of kinematically redundant manipulators.
- No solution in certain manipulator configurations.

For simple manipulators with less number of degrees of freedom the inverse kinematics solution is fairly simple too. But for manipulators with more number

of degrees of freedom the solution becomes highly complicated due to the matrix inversion. Though the inverse kinematics is certain to converge if the given position and orientation falls in the dexterous workspace of the manipulator.

The homogenous transformation matrix transforms the joint angles to the EE position and orientation and vice versa. However an important matrix called the Jacobian matrix relates the joint and the EE velocities.

$$\dot{q} = J\,\dot{x}$$

The Jacobian matrix can be separated into position and orientation jacobian, which determines the contribution of the joint velocities to the EE linear and angular velocities respectively. Chapter 3 of **[Sci00]** gives a detailed description on the geometric and analytical jacobian and its derivation. Moreover it also gives an introduction to different inverse kinematic schemes for both redundant and non-redundant manipulators. It is very important for the reader to understand the basic kinematics and differential kinematics of a manipulator to perform any sort of operation with it. This research does not concentrate too much on the basics, but on how to use them to our convenience to fulfill the objective. After a thorough understanding of the basics involved the, future chapters explain on how they have been used to perform the Cartesian space position mapping, forward kinematics and the inverse kinematics.

Chapter 3

WAM & Titan Kinematics

3.1 WAM Kinematics

The master device used to control the slave is a 4-DOF WAM (shown in Fig 3.1) with a 3-DOF-gimbal structure attached to its end. The gimbal attaches itself to the WAM at the end of the second link. The above picture shows the joints and links of a 4-DOF WAM. It is a redundant manipulator wherein any joint can act as the redundant degree of freedom based on the task being executed. WAM uses a state of the art cable drive system to actuate its joints. The main drive system is housed at the base of the manipulator. The main feature of WAM is it's back drivability, i.e., the manipulator joints can be actuated by external forces too.

Back drivability is also the measure of how well a force or motion applied at the output end is transmitted to the input end. This manipulator can therefore be used for force or motion feedback in bilateral telemanipulation systems. The name suggests that it allows control of forces and torques all along the links and not just at the end effector. The final link, the gimbal cylinder allows the operator to manipulate the WAM by holding it. WAM is uniquely designed to exhibit the advantages of redundant manipulators. It finds extended usability in various applications. Various features like back drivability, gravity compensation and force feedback capability makes WAM perfectly suitable for a master device. WAM has been used extensively for various autonomous operations and this application as a master device is the first of its kind.

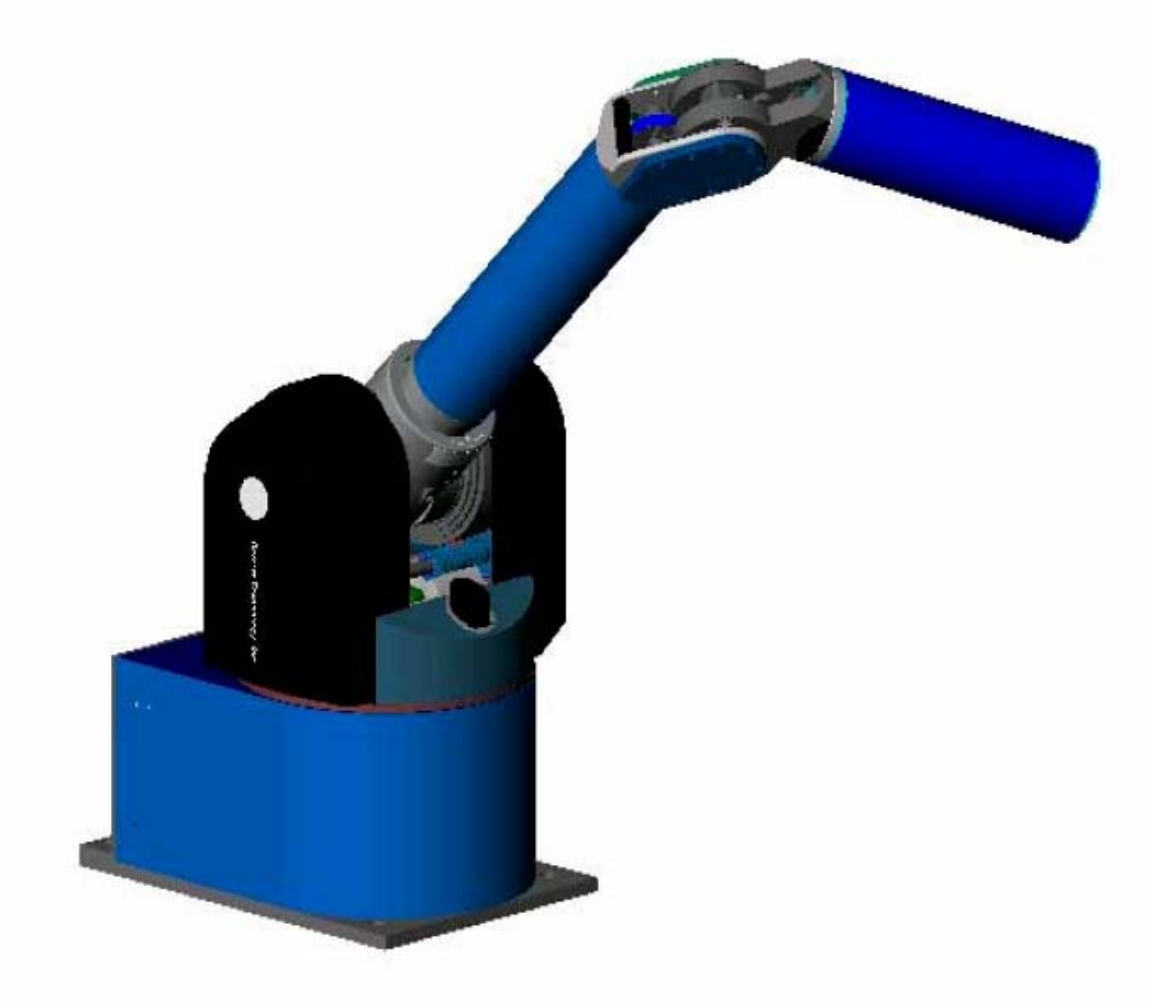

Fig 3.1 Master manipulator used in current teleoperational system

The reason for using WAM in controlling the slave in this telerobotic task is for its back-drivability and its dexterous workspace. Since it is manually operated the redundant DOF can be used efficiently in manipulating its end-effector to proximal and distal points from its base.

3.1.1 WAM Specifications

The original WAM has four revolute joints without a wrist. The first three joints intersect at the base and the distal joint is located at 0.6 meters from the base. The two cylindrical links have a combined reach of one meter and weighs 4 kg. The total mass of the WAM including the base and the motors is 35Kg. The angle limits for all the four joints, ranges from zero to three quarters of a revolution. Four brushless DC motors and a stiff back-drivable multistage cable transmission are used to actuate all the four joints. A 12-bit resolution resolver mounted at each motor shaft gives the joint angles and a Hall effect sensor gives the joint torque from the motor winding current. These position and torque measurements enable effective control of the arm. Though direct drive is possible with the current size of WAM, small motors with speed reducing cable transmissions are used to improve torque ripple, back driven inertia, and the effective motor constant while simultaneously reducing the weight and cost. The links are covered with a 3-mm thick dense foam to adjust to the contact forces of the manipulated objects. The two links are made of aluminum to absorb local impacts with a wall thickness of 5-mm and a radius of 38-mm. The arm is made stronger to lift its payload against gravity and stiffer than the servomotors driving it.

Denavit Hartenberg Convention for WAM

The three dimensional diagram of WAM in Fig 3.2 shows the type of joint for all the degrees of freedom and its arrangement with respect to each other. For simplicity

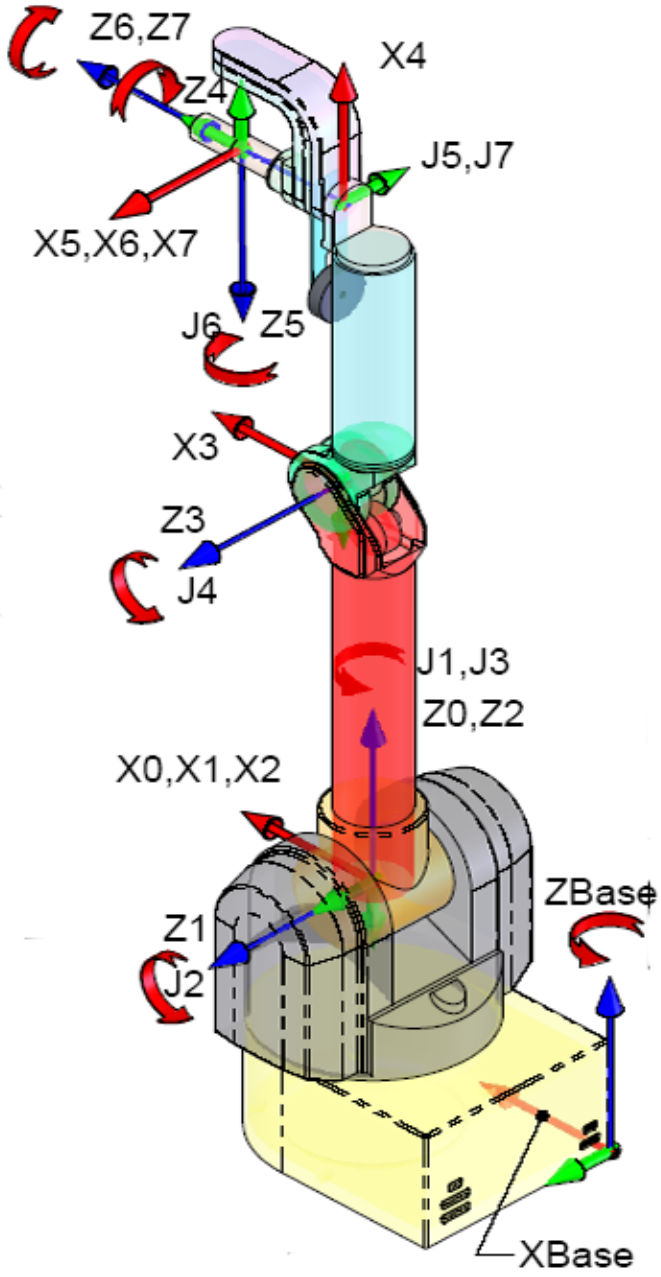

Fig 3.2 3D Model of master manipulator WAM+Gimbal showing D-H frames

the first three joints are made to intersect with each other at the base and the last three joints intersect with each other at the gimbals' cylinder. This makes the D-H table for the WAM pretty simple. The co-ordinate frames for each joint is indicated in Fig 3.3 along with the link lengths and joint offsets. The D-H table for the WAM is as follows:

This D-H table 3.1 gives the parameters associated with each frame of the manipulator with respect to the previous. As mentioned in the previous chapter the transformation matrix from one frame to the other is

$$T_i^{i-1}(q_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{- (3.1)}$$

In accordance to the transformation matrix and the D-H parameters the transformation matrices are,

- From Frame 1 to Frame 0

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From Frame 2 to Frame 1

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & 0 \\ \sin\theta_2 & 0 & -\cos\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

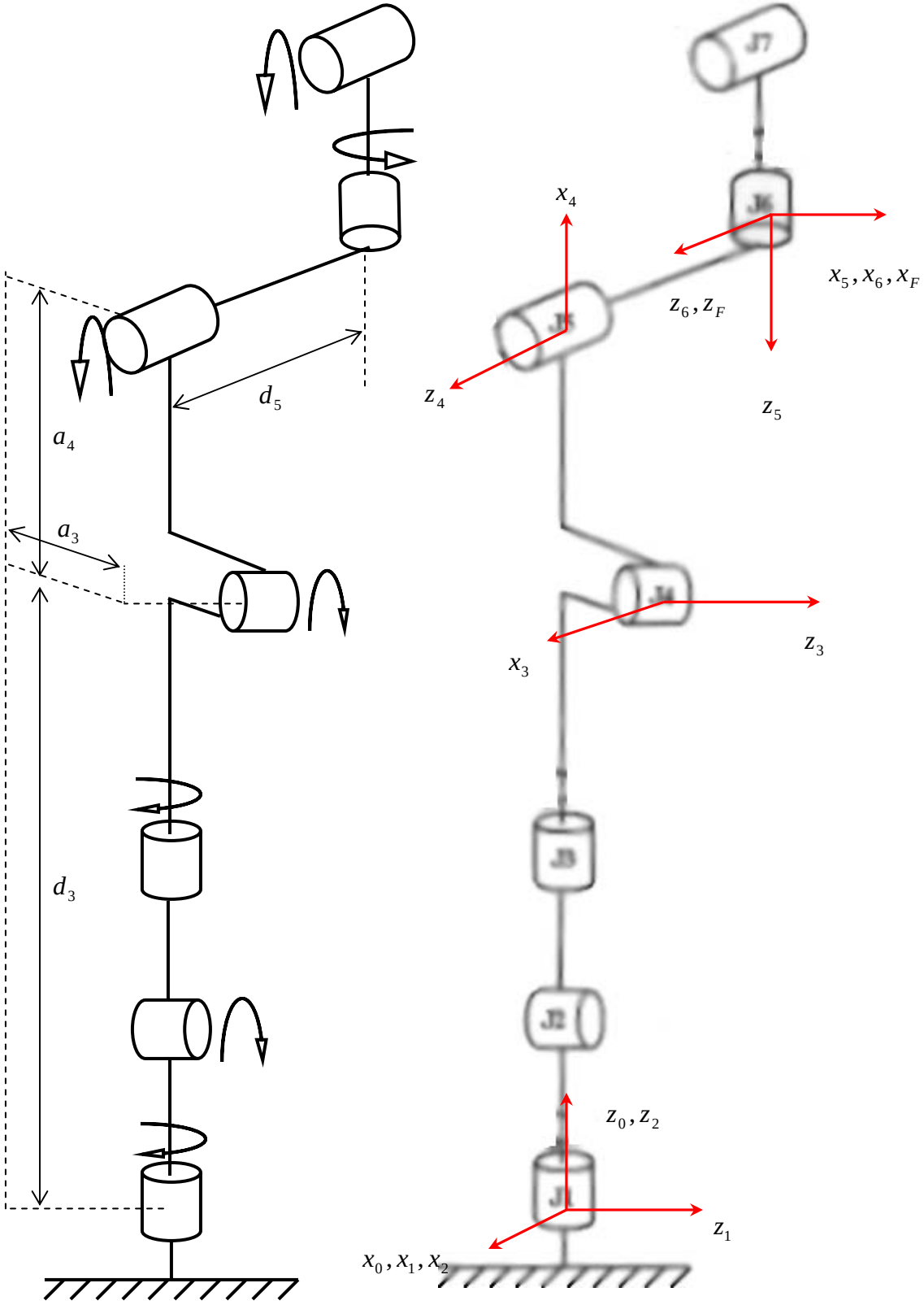

Fig 3.3 Kinematic model of WAM and Gimbal with the D-H parameters

Table 3.1 D-H table for WAM

4 DOF + G Link Frame	**Z-axis of this frame is joint**	To get from the previous frame to this frame			
		a	α	**d**	θ
0	1				
1	2	0	$-\pi/2$	0	θ_1
2	3	0	$\pi/2$	0	θ_2
3	4	**0.045**	$-\pi/2$	**0.55**	θ_3
4	5	**0.4**	$-\pi/2$	0	$\theta_4 - \pi/2$
5	6	0	$\pi/2$	**0.1547**	$\theta_5 - \pi/2$
6	7	0	$-\pi/2$	0	θ_6
Final		0	0	0	θ_7

- From Frame 3 to frame 2

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & 0 & -\sin\theta_3 & 0.045\cos\theta_3 \\ \sin\theta_3 & 0 & \cos\theta_3 & 0.045\sin\theta_3 \\ 0 & -1 & 0 & 0.55 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From Frame 4 to Frame 3

$$T_4^3 = \begin{bmatrix} \sin\theta_4 & 0 & \cos\theta_4 & 0.4\sin\theta_4 \\ -\cos\theta_4 & 0 & \sin\theta_4 & -0.4\cos\theta_5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From Frame 5 to Frame 4

$$T_5^4 = \begin{bmatrix} \sin\theta_5 & 0 & -\cos\theta_5 & 0 \\ -\cos\theta_5 & 0 & -\sin\theta_5 & 0 \\ 0 & 1 & 0 & 0.1547 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- From Frame 6 to Frame 5

$$T_6^5 = \begin{bmatrix} \cos\theta_6 & 0 & -\sin\theta_6 & 0 \\ \sin\theta_6 & 0 & \cos\theta_6 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Final Frame to Frame 6

$$T_F^6 = \begin{bmatrix} \cos\theta_7 & -\sin\theta_3 & 0 & 0 \\ \sin\theta_7 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Transformation matrix from the final frame to the Frame 0 is

$$T_F^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 = \begin{bmatrix} n1 & s1 & a1 & px \\ n2 & s2 & a2 & py \\ n3 & s3 & a3 & pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} n1 = {} & ((((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3)\sin\theta_4 + \cos\theta_1 \sin\theta_2 \cos\theta_4)\sin\theta_5 - \\ & (\cos\theta_1 \cos\theta_2 \sin\theta_3 + \sin\theta_1 \cos\theta_3)\cos\theta_5)\cos\theta_6 + ((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \\ & \sin\theta_1 \sin\theta_3)\cos\theta_4 - \cos\theta_1 \sin\theta_2 \sin\theta_4)\sin\theta_6)\cos\theta_7 + (((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \\ & \sin\theta_1 \sin\theta_3)\sin\theta_4 + \cos\theta_1 \sin\theta_2 \cos\theta_4)\cos\theta_5 + (\cos\theta_1 \cos\theta_2 \sin\theta_3 + \\ & \sin\theta_1 \cos\theta_3)\sin\theta_5)\sin\theta_7 \end{aligned}$$

$$\begin{aligned} s1 = {} & -((((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3)\sin\theta_4 + \cos\theta_1 \sin\theta_2 \cos\theta_4)\sin\theta_5 - \\ & (\cos\theta_1 \cos\theta_2 \sin\theta_3 + \sin\theta_1 \cos\theta_3)\cos\theta_5)\cos\theta_6 + ((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \\ & \sin\theta_1 \sin\theta_3)\cos\theta_4 - \cos\theta_1 \sin\theta_2 \sin\theta_4)\sin\theta_6)\sin\theta_7 + (((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \\ & \sin\theta_1 \sin\theta_3)\sin\theta_4 + \cos\theta_1 \sin\theta_2 \cos\theta_4)\cos\theta_5 + (\cos\theta_1 \cos\theta_2 \sin\theta_3 + \\ & \sin\theta_1 \cos\theta_3)\sin\theta_5)\cos\theta_7 \end{aligned}$$

$$\begin{aligned} a1 = {} & -(((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3)\sin\theta_4 + \cos\theta_1 \sin\theta_2 \cos\theta_4)\sin\theta_5 - \\ & (\cos\theta_1 \cos\theta_2 \sin\theta_3 + \sin\theta_1 \cos\theta_3)\cos\theta_5)\sin\theta_6 + ((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3)\cos\theta_4 \\ & - \cos\theta_1 \sin\theta_2 \sin\theta_4)\cos\theta_6 \end{aligned}$$

$$\begin{aligned} px = {} & ((\cos\theta_1 \cos\theta_2 \cos\theta_3 - \sin\theta_1 \sin\theta_3)\cos\theta_4 - \cos\theta_1 \sin\theta_2 \sin\theta_4)d_5 + (\cos\theta_1 \cos\theta_2 \cos\theta_3 \\ & - \sin\theta_1 \sin\theta_3)a_4 \sin\theta_4 + \cos\theta_1 \sin\theta_2 a_4 \cos\theta_4 + \cos\theta_1 \cos\theta_2 a_3 \cos\theta_3 - \sin\theta_1 a_3 \sin\theta_3 \\ & + \cos\theta_1 \sin\theta_2 d_3 \end{aligned}$$

$$\begin{aligned} n2 = {} & ((((\sin\theta_1 \cos\theta_2 \cos\theta_3 - \cos\theta_1 \sin\theta_3)\sin\theta_4 + \sin\theta_1 \sin\theta_2 \cos\theta_4)\sin\theta_5 - (\sin\theta_1 \cos\theta_2 \sin\theta_3 \\ & - \cos\theta_1 \cos\theta_3)\cos\theta_5)\cos\theta_6 + ((\sin\theta_1 \cos\theta_2 \cos\theta_3 + \cos\theta_1 \sin\theta_3)\cos\theta_4 - \sin\theta_1 \sin\theta_2 \sin\theta_4) \\ & \sin\theta_6)\cos\theta_7 + (((\sin\theta_1 \cos\theta_2 \cos\theta_3 + \cos\theta_1 \sin\theta_3)\sin\theta_4 + \sin\theta_1 \sin\theta_2 \cos\theta_4)\cos\theta_5 + \\ & (\sin\theta_1 \cos\theta_2 \sin\theta_3 - \cos\theta_1 \cos\theta_3)\sin\theta_5)\sin\theta_7 \end{aligned}$$

$$s2 = -((((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\sin\theta_4 + \sin\theta_1\sin\theta_2\cos\theta_4)\sin\theta_5 - (\sin\theta_1\cos\theta_2\sin\theta_3 - \cos\theta_1\cos\theta_3)\cos\theta_5)\cos\theta_6 + ((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\cos\theta_4 - \sin\theta_1\sin\theta_2\sin\theta_4)\sin\theta_6)\sin\theta_7 + (((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\sin\theta_4 + \sin\theta_1\sin\theta_2\cos\theta_4)\cos\theta_5 + (\sin\theta_1\cos\theta_2\sin\theta_3 - \cos\theta_1\cos\theta_3)\sin\theta_5)\cos\theta_7$$

$$a2 = -(((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\sin\theta_4 + \sin\theta_1\sin\theta_2\cos\theta_4)\sin\theta_5 - (\sin\theta_1\cos\theta_2\sin\theta_3 - \cos\theta_1\cos\theta_3)\cos\theta_5)\sin\theta_6 + ((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\cos\theta_4 - \sin\theta_1\sin\theta_2\sin\theta_4)\cos\theta_6$$

$$py = ((\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\sin\theta_3)\cos\theta_4 - \sin\theta_1\sin\theta_2\sin\theta_4)d_5 + (\sin\theta_1\cos\theta_2\cos\theta_3 + \cos\theta_1\cos\theta_3)a_4\sin\theta_4 + \sin\theta_1\sin\theta_2 a_4\cos\theta_4 + \sin\theta_1\cos\theta_2 a_3\cos\theta_3 + \cos\theta_1 a_3\sin\theta_3 + \sin\theta_1\sin\theta_2 d_3$$

$$n3 = (((-\sin\theta_2\cos\theta_3\sin\theta_4 + \cos\theta_2\cos\theta_4)\sin\theta_5 + \sin\theta_2\sin\theta_3\cos\theta_5)\cos\theta_6 + (-\sin\theta_2\cos\theta_3\cos\theta_4 - \cos\theta_2\sin\theta_4)\sin\theta_6)\cos\theta_7 + ((-\sin\theta_2\cos\theta_3\sin\theta_4 + \cos\theta_2\cos\theta_4)\cos\theta_5 - \sin\theta_2\sin\theta_3\sin\theta_5)\sin\theta_7$$

$$s3 = -(((-\sin\theta_2\cos\theta_3\sin\theta_4 + \cos\theta_2\cos\theta_4)\sin\theta_5 + \sin\theta_2\sin\theta_3\cos\theta_5)\cos\theta_6 + (-\sin\theta_2\cos\theta_3\cos\theta_4 - \cos\theta_2\sin\theta_4)\sin\theta_6)\sin\theta_7 + ((-\sin\theta_2\cos\theta_3\sin\theta_4 + \cos\theta_2\cos\theta_4)\cos\theta_5 - \sin\theta_2\sin\theta_3\sin\theta_5)\cos\theta_7$$

$$a3 = -((-\sin\theta_2\cos\theta_3\sin\theta_4 + \cos\theta_2\cos\theta_4)\sin\theta_5 + \sin\theta_2\sin\theta_3\cos\theta_5)\sin\theta_6 + (-\sin\theta_2\cos\theta_3\cos\theta_4 - \cos\theta_2\sin\theta_4)\cos\theta_6$$

$$pz = (-\sin\theta_2\cos\theta_3\cos\theta_4 - \cos\theta_2\sin\theta_4)d_5 - \sin\theta_2\cos\theta_3 a_4\sin\theta_4 + \cos\theta_2 a_4\cos\theta_4 - \sin\theta_2\cos\theta_3 a_3 + \cos\theta_2 d_3$$

3.2 TITAN Kinematics

The slave manipulator, which interacts with the real environment, is the Titan II manipulator from Schilling Robotics. The Schilling Titan II is a commercially available robot that was developed originally for under sea applications. Commonly used on submersibles, it has also found favor in the nuclear industry due to its titanium construction, and high performance. The Titan II has a 6-foot reach, and a *109 kg*. payload at full extension. It is also a very agile robot, and precise in operations. The Titan II is a very versatile robot for manipulating heavy objects and demolition. For this reason it is preferred in unstructured environments such as Nuclear Waste disposal sites. It is a 6-DOF with jaw open/close and hydraulically actuated manipulator. The six degrees of freedom are Yaw at the azimuth (rotation at the horizontal plane), shoulder pitch, Pitch at the elbow, Pitch at the wrist, Yaw at the wrist and wrist rotation. The Titan II is a high velocity, spatially correspondent remote manipulator system. It consists of

- A hydraulically actuated slave arm, which manipulates the objects using a jaw tool.
- The slave controller, which provides power to the slave arm and transmits process signals between the master controller and the slave arm.
- A master controller, which contains
    - A master arm kinematically similar to the slave arm.
    - A control panel for the master arm.
    - A master controller, which transmits process signals between the master arm and the slave controller.

Fig 3.4 shows the sketch of the Titan with all the six degrees of freedom. In its outstretched position it has a reach of 2 meters with the center of the gripper at its open position taken as the end point for the Cartesian space position mapping.

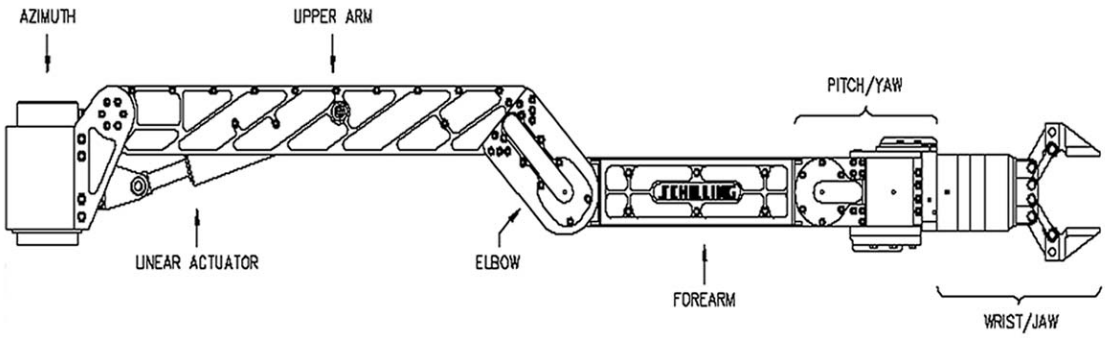

Fig 3. 4 Titan II displaying its six degrees of freedom

The slave arm imitates the master arm in its movements. The signals from the master arm either activate the servo valves to actuate the joints or drive a solenoid valve, which would activate or deactivate the hydraulics. Resolvers fixed at each joint give the joint angle values wit great accuracy. The stator of the resolver is fixed firmly to one link and the rotor to the other. As the joint moves the rotor rotates with respect to the stator. The resolver outputs are communicated to the slave controller where they are converted into digital position data for the joint. The position of the jaw is known through an Linear variable differential transformer (LVDT) in the jaw assembly. There is a provision to mount a force/torque sensor in the wrist of the Titan. The sensor consists of four rings each having a strain gauge. These strain gauges sense the forces in the x, y and z directions and the torques about them. These data are then amplified and fed to the slave controller where they can be interpreted into forces at each joints. A pressure sensor can also be fixed to the jaw to measure the force applied on the tools by the jaw. The wiring for these sensor and other electronic components housed and the hydraulic pipes are totally concealed within the arm.

Fig 3.5 shows the kinematic diagram of Titan with the co-ordinate frames and the D-H parameters. The base reference frame can be moved anywhere along the first link in which case the value of d_1 depend on the position of the frame. For convenience the base frame is taken at the bottom of the base structure. If the base frame is moved to the top of the base structure Frame 1' is avoided and a direct transformation from the base frame to Frame 1 is done. Similarly a direct transformation could be done from Frame 1 to Frame 2 without going through frame 2' in which case the d_2 has two components. The frames are split for convenience and an orderly and smooth transformation between frames. The D-H table for the Titan is given in Table 3.2

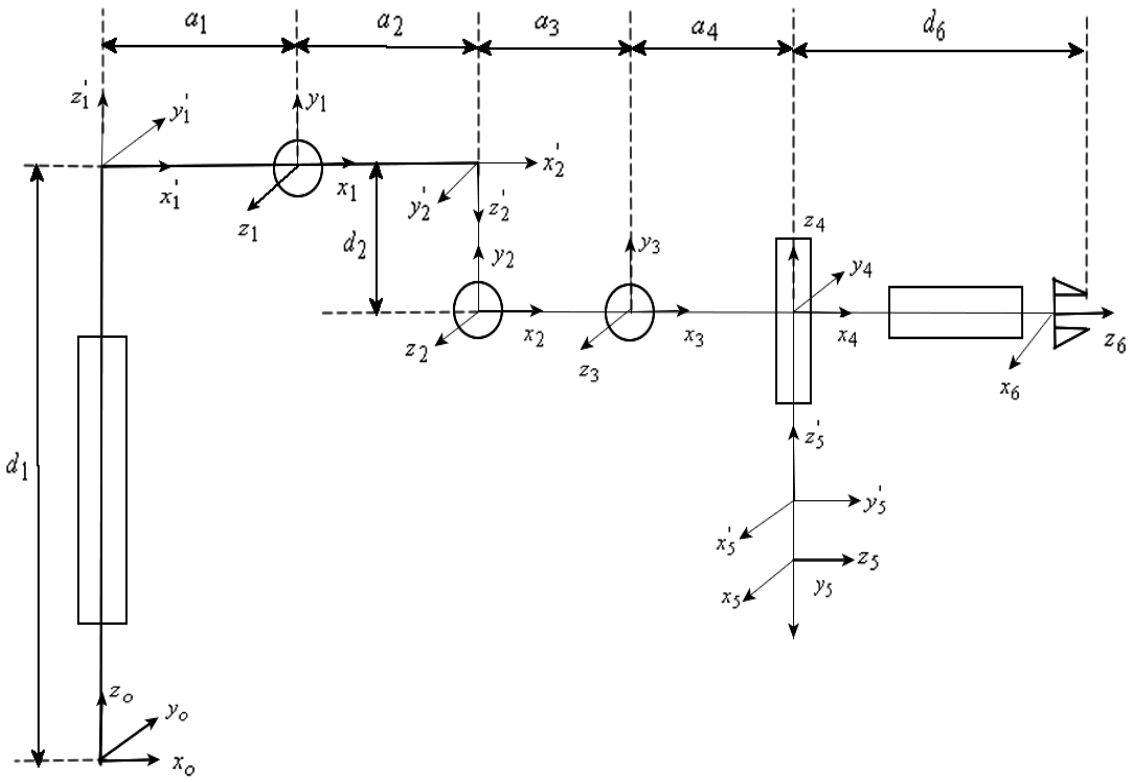

Fig 3.5 D-H frames and parameters of Titan

Table 3.2 D-H table for Titan

D-H Frames	**D-H parameters to get from Previous frame to the current frame**			
	A	α	**d**	θ
$1^{'}$	0	0	**0.19482**	θ_1
1	**0.1216**	$\pi/2$	0	0
$2^{'}$	**0.843**	$\pi/2$	0	θ_2
2	0	$-\pi/2$	**0.11557**	0
3	**0.4826**	0	0	θ_2
4	**0.13335**	$-\pi/2$	0	θ_4
$5^{'}$	0	0	0	$-\pi/2$
5	0	$-\pi/2$	0	θ_5
6	0	0	**0.4168**	θ_6

In accordance to the transformation matrix described in (3.1) the homegenous transformation matrix from one frame to the other is defined as.

$$A_1^0 = A_{1'}^0 A_1^{1'} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & a_1\sin\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = A_{2'}^1 A_2^{2'} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & a_2\cos\theta_2 \\ \sin\theta_2 & 0 & -\cos\theta_2 & a_2\sin\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & d_2\sin\theta_2 + a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & -d_2\cos\theta_2 + a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & a_4\cos\theta_4 \\ \sin\theta_4 & 0 & \cos\theta_4 & a_4\sin\theta_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = A_{5'}^4 A_5^{5'} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_5 & 0 & -\sin\theta_5 & 0 \\ \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ -\cos\theta_5 & 0 & \sin\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^5 = \begin{bmatrix} \cos\theta_6 & -\sin\theta_6 & 0 & 0 \\ \sin\theta_6 & \cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The overall transformation from the base frame to the EE frame is given by

$$A_6^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 = \begin{bmatrix} n1 & s1 & a1 & px \\ n2 & s2 & a2 & py \\ n3 & s3 & a3 & pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n1 = \cos\theta_1 \cos\theta_6 \sin\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) + \sin\theta_1 \cos\theta_5 \cos\theta_6 + \cos\theta_1 \sin\theta_6 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$s1 = -\cos\theta_1 \sin\theta_5 \sin\theta_6 \cos(\theta_2 + \theta_3 + \theta_4) - \sin\theta_1 \sin\theta_6 \cos\theta_5 + \cos\theta_1 \cos\theta_6 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$a1 = \cos\theta_1 \cos\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) - \sin\theta_1 \sin\theta_5$$

$$px = d_6(\cos\theta_1\cos\theta_5\cos(\theta_2+\theta_3+\theta_4)-\sin\theta_1\sin\theta_5)+a_4\cos\theta_1\cos(\theta_2+\theta_3+\theta_4)+ a_3\cos\theta_1\cos(\theta_2+\theta_3)+a_2\cos\theta_1\cos\theta_2+d_2\cos\theta_1\sin\theta_2+a_1\cos\theta_1$$

$$n2 = \sin\theta_1\cos\theta_6\sin\theta_5\cos(\theta_2+\theta_3+\theta_4)-\cos\theta_1\cos\theta_5\cos\theta_6+ \sin\theta_1\sin\theta_6\sin(\theta_2+\theta_3+\theta_4)$$

$$s2 = -\sin\theta_1\sin\theta_5\sin\theta_6\cos(\theta_2+\theta_3+\theta_4)-\cos\theta_1\sin\theta_6\cos\theta_5+ \sin\theta_1\cos\theta_6\sin(\theta_2+\theta_3+\theta_4)$$

$$a2 = \sin\theta_1\cos\theta_5\cos(\theta_2+\theta_3+\theta_4)+\cos\theta_1\sin\theta_5$$

$$py = d_6(\sin\theta_1\cos\theta_5\cos(\theta_2+\theta_3+\theta_4)+\cos\theta_1\sin\theta_5)+a_4\sin\theta_1\cos(\theta_2+\theta_3+\theta_4)+ a_3\sin\theta_1\cos(\theta_2+\theta_3)+a_2\sin\theta_1\cos\theta_2+d_2\sin\theta_1\sin\theta_2+a_1\sin\theta_1$$

$$n3 = \sin\theta_5\cos\theta_6\sin(\theta_2+\theta_3+\theta_4)-\sin\theta_6\cos(\theta_2+\theta_3+\theta_4)$$

$$s3 = -\sin\theta_5\sin\theta_6\sin(\theta_2+\theta_3+\theta_4)-\cos\theta_6\cos(\theta_2+\theta_3+\theta_4)$$

$$a3 = \cos\theta_5\sin(\theta_2+\theta_3+\theta_4)$$

$$pz = d_6\sin(\theta_2+\theta_3+\theta_4)\cos\theta_5+a_4\sin(\theta_2+\theta_3+\theta_4)+a_3\sin(\theta_2+\theta_3)+a_2\sin\theta_2 - d_2\cos\theta_2+d_1$$

After having established the individual and the final transformation matrices, it is easy to determine the Jacobian matrix according to the procedure explained in the previous chapter.

Chapter 4

Cartesian Space Position Mapping

After having explained the kinematics of both the master and slave manipulators it is now time to devise a method to control the movements of the slave with the master. The approach used to imitate the slave manipulator with the motions of the master is the Cartesian space position mapping (CSPM). In this method the position of the EE of the master with respect to its base frame is transformed suitably to that of the slave manipulator EE. The primary factor to be considered here is the compatibility of both the workspaces. However it is very important that the Titan EE does not exceed its operational space while the master EE is operating in its workspace. This depends on the kinematic dissimilarity of both the manipulators and the joint angle limits. CSPM takes all these factors into account while transforming the position of the EE. Unlike ideal method of joint level control, CSPM maps every point in the trajectory of the master to the corresponding point in the slave workspace so that the slave trajectory is same as that of the master. In short the entire CSPM procedure can be explained as.

1. Obtain the position of the EE of the master at ever time step in its trajectory.
2. Determine the EE operational space linear and angular velocities for the master.
3. Scale the velocities suitably to the operational space velocities of the slave. Ideally this scaling factor is one and varies depending on the operating characteristics.
4. Determine the joint space velocities for the Titan through Jacobian inverse.
5. Determine the joint angles at every time step for all the six joints of the slave.
6. Check for constraint violation for WAM.

However a unique inverse kinematic solution does not exist for the WAM, as it is a redundant manipulator. Therefore redundancy resolution has to be done for the WAM to

obtain unique set of joint angle values for each position and orientation of the EE. The entire CSPM procedure in explained in detail in this chapter splitting it into simple modules each performing a certain operation. Initially only the inverse kinematics of the slave is taken into consideration. After successfully executing it, the inverse kinematics of the WAM is considered. Though pure teleoperation occurs where the accuracy of position tracking of the slave EE is not of great importance, a Jacobian transpose method is adopted to improve its accuracy significantly. Fig 4.1 explains the data flow in the CSPM method. The equations following the diagram explain the same architecture of the CSPM in terms of implementing in software. The following figure basically shows the flow of the algorithm.

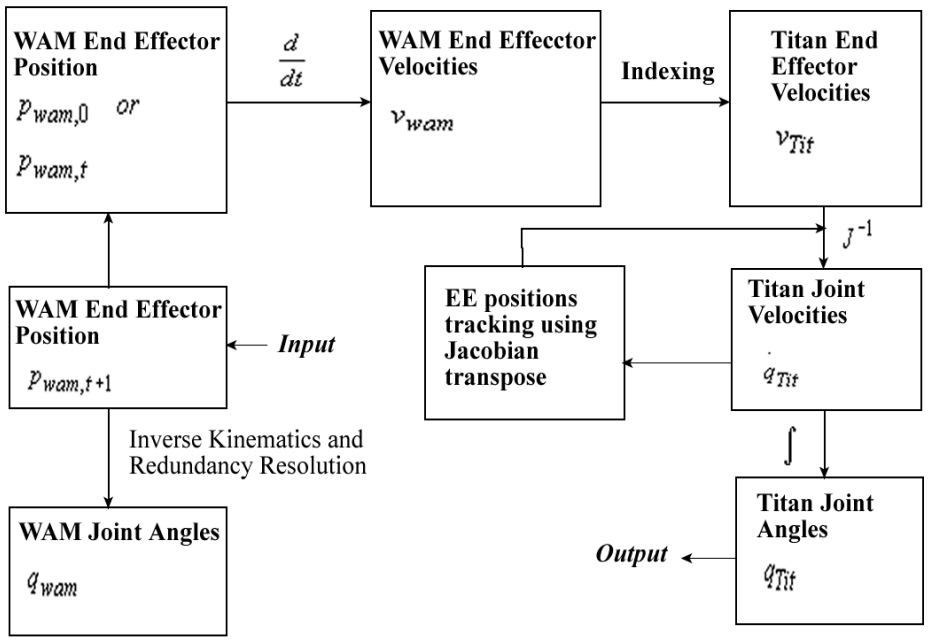

Fig 4.1 Functional architecture of CSPM

Forward Kinematics of WAM

$$\dot{x}_{wam} = J_{wam}\,\dot{q}_{wam} \qquad \text{- (4.1)}$$

$\dot{x}_{wam}$ - EE linear and angular velocity (6×1)

$$\dot{x}_{wam} = \frac{p_{t+1} - p_t}{dt} \qquad \text{- (4.2)}$$

p_{t+1}, p_t - EE position and orientation at consecutive time steps.

J_{wam} - Jacobian matrix (6×7)

$\dot{q}_{wam}$ - Joint angular velocity (7×1)

Velocity Scaling (Indexing)

$$\dot{x}_{Tit} = C\,\dot{x}_{wam} \qquad \text{- (4.3)}$$

$\dot{x}_{Tit}$ - EE linear and angular velocities of Titan (6×1)

C – Velocity scaling (usually 1).

Inverse Kinematics of Titan

$$\dot{q}_{Tit} = J_{Tit}^{-1}\,\dot{x}_{Tit} \qquad \text{- (4.4)}$$

J_{Tit}^{-1} - Jacobian Inverse (6×6)

$$\theta_i = \theta_o + \sum_{k=1}^{n} \dot{q}_i(k)\,dt \qquad \text{- (4.5)}$$

The CSPM algorithm consists of these main tasks carried out by various modules. The input to the CSPM algorithm is the position of the master EE and the output is the Joint angles of the slave. The unique feature of CSPM is that it can be used to control any degrees of freedom manipulator using any degrees of freedom master. However the only alteration occurs at the inverse kinematics part of the slave if it has redundant degrees of

freedom. The CSPM applied to this research work attempts to map the EE position of WAM to the Titan. WAM has a relatively less workspace than the Titan, which plays a major role in scaling the positions. Also the velocity limits for each joints of the manipulators differs significantly. Following is the explanation for each module of the CSPM algorithm, which ultimately performs the required operation.

The entire CSPM procedure can be divided into three modules. They are

1. Forward kinematics
2. Inverse kinematics and EE position tracking for Titan.
3. Redundancy resolution and inverse kinematics of the master manipulator.

4.1. Forward Kinematics

All the three modules require standard functions like vector subtraction, vector cross product, vector dot product, matrix transpose, matrix subtraction and matrix dot product. Forward kinematics represents the relation between the position and orientation of the EE with the joint angles. The function that relates these two is called the transformation matrix. Similarly the Jacobian matrix relates the joint space velocities to the operational space velocities. The function in CSPM, which carries out these operations, is the HomoJac and HomoJacTitan. This function takes in joint angles as input and determines their respective transformation matrix and Jacobian matrix. The inverse Jacobian is calculated in HomoJacTitan alone for the Titan manipulator as a different approach is handled for the WAM inverse kinematics. The transformation matrices are defined according to the kinematic arrangement of the manipulator links. The final transformation matrix is determined by a series of matrix multiplication, which has sines and cosines of the input angles and the position of the joint with respect to the base frame. The first three rows of the Jacobian matrix is a cross product between two vectors. This is performed by the vector cross product function, which takes in two vectors as input and outputs the final vector (cross product of the two input vectors). This function including all other functions mentioned earlier uses floating pointers for

mathematical calculation. This operation is performed separately in HomoJac and HomoJacTitan as the Jacobian matrix size varies for WAM and Titan. It is a square matrix (6×6) for Titan and rectangular matrix (6×7) for the WAM. Once the Jacobian matrices are determined they are then used in the main function to calculate the EE velocities in Cartesian space as explained in equation (4.1).

4.2 Inverse Kinematics

The entire inverse kinematics of the Titan is divided into two main sub modules. The first is the actual inverse of the Jacobian matrix using Singular Value Decomposition (SVD). The second module performs the real inverse kinematics using Jacobian transpose method.

4.2.1 Jacobian Inverse

Various methods have been proposed to perform the Jacobian inverse in many literatures **[Mer]**. Two methods have been employed to perform the inverse kinematics for the Titan manipulator. Firstly pseudo inverse of the Jacobian is computed by factorizing it using SVD and finally damped least square method is used to perform the inverse kinematics. Both these approaches are discussed in detail in this section and the best method is identified and discussed in the results chapter by performing the actual simulation using RoboWorks.

Singular Value Decomposition and the Pseudo Inverse

SVD is the most popular method used to determine the Inverse. SVD is a powerful tool in calculating the inverse for singular matrices or matrices numerically nearing singularity **[Str88]**. It also shows the applications of SVD and how it transforms

a vector from one dimension to other. SVD is used to perform the Jacobian inverse for Titan because at certain points near the workspace boundaries the Jacobian tends to reach singularity.

The Jacobian matrix $(m \times n)$ (in this case $m = n$) can be factorized into product of three matrices, a $(m \times m)$ orthogonal matrix U, an $(m \times n)$ diagonal matrix W of positive elements or zero and transpose of an $(n \times n)$ orthogonal matrix V.

$$J = U.W.V^T$$

The matrices U and V are orthogonal matrices, which are the left and right singular vectors of J. The diagonal matrix W contains the singular values of J. They are normally arranged in the order of decreasing magnitude.

$$W = \begin{bmatrix} w_1 & 0 & . & 0 & 0 \\ 0 & w_2 & . & 0 & 0 \\ . & . & . & . & . \\ 0 & 0 & . & w_{n-1} & 0 \\ 0 & 0 & 0 & 0 & w_n \end{bmatrix}$$

Since U and V are orthogonal matrices

$UU^T = VV^T = 1$ and $w_1, w_2, ...w_n \geq 0$

Since the Jacobian matrix has been factorized into three matrices it now easy to define its inverse which is

$$\begin{aligned} J^{-1} &= (UWV^T)^{-1} \\ &= (V^T)^{-1} W^{-1} U^{-1} \\ &= V.W^{-1}.U^T \end{aligned} \qquad \text{- (4.6)}$$

The final inverse jacobian in eq (4.6) is due to the orthogonality property of the two matrices U and V, which says that the inverse of an orthogonal matrix is equal to its transpose. The inverse of the diagonal matrix W is defined as

$$W^{-1} = \begin{bmatrix} \frac{1}{w_1} & 0 & . & 0 & 0 \\ 0 & \frac{1}{w_2} & . & 0 & 0 \\ . & . & . & . & . \\ 0 & 0 & . & \frac{1}{w_{n-1}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{w_n} \end{bmatrix}$$

In an ideal case where the Jacobian matrix is full rank all the elements of the diagonal matrix are positive and greater than zero. But when the manipulator crosses certain singular points some of the elements of matrix *W* might become zero or close to zero. Due to that effect, the corresponding diagonal elements in W^{-1} blows up (tends to reach infinity), which ultimately blows the entire inverse matrix. That is when the pseudo inverse is performed with the help of SVD. Pseudo inverse determines the matrix inverse wherever possible. i.e. wherever Jacobian matrix is invertible. It takes in the matrices V, W^{-1} *and* U^T and determines the Jacobian inverse. In order to determine which rows of the Jacobian actually contributes to the inverse, pseudo inverse uses the condition number. This is very important because singular values are so significant in determining how error is magnified since they specify how a transformation scales different vectors from the input space to the output space. Condition number is the ratio of the largest singular value to the smallest of the Jacobian matrix. i.e. the ratio of the first diagonal element to the last diagonal element in matrix *W*. It gives a measure of the sensitivity of the matrix to numerical calculations. For example a matrix with condition number 1 is well-conditioned and not very sensitive to change in outputs. A matrix with condition number to the order of 10^{-5} is ill-conditioned and is very sensitive to even very small change in outputs. It is important that we set an optimal condition number while employing Pseudo Inverse to perform the inverse calculations. The inverse calculations in

CSPM were performed setting the condition number to 25. The advantage of fixing the condition to a defined value is that any singular value of the Jacobian matrix, which does not contribute to its inverse or makes the inverse more sensitive, can be removed. The disadvantage of a high condition number can be seen in the joint velocities. If the condition number were to be 100 or more, then it would result in a highly sensitive Jacobian inverse, which will produce a prodigious change in the joint velocities for a meager change in the EE position. This is highly undesirable for a teleoperation. Hence there is a tradeoff in the performance even when the best method of matrix inverse is used. Once the pseudo inverse of the Jacobian is determined the inverse kinematics is done according to eq (4.4), which is a mere multiplication of the Jacobian inverse and the EE velocities. The disadvantage with this method is that it fails at singular points and complex inverse calculations have to be performed at every time step. Additional to the inverse kinematics, pseudo inverse was also used to optimize the joint angles, which gave the minimum position tracking error **[Mer]**. This EE position tracking error reduction was done in the following way.

- Define the allowable error *e* for the EE position.
- Calculate the difference between the EE actual position and the desired position.

$$dx = x_d - x_a$$

- Calculate the Jacobian using the current joint angles. Use the initial angles for the first time step.
- Calculate the pseudo inverse of the Jacobian using SVD.
- Determine the error of the pseudo inverse

$$e_p = |(I - JJ^{-1})dx|$$

- If $e_p > e$ then $dx = \frac{dx}{2}$. Restart from the step 5.
- If the error is satisfied then calculate the updated joint angles and use these as the new set of angles.

$$\theta_o = \theta_n + J^{-1} dx$$

- Determine the EE position and orientation from the forward kinematics and repeat the entire procedure from step 2 for every time step.

The disadvantage with pseudo inverse method is that the least square criteria of achieving the EE trajectory minimizing the error $|\dot{x} - J\dot{q}|$ is given priority over minimizing the joint velocity $|\dot{q}|$.

4.2.2 Damped Least Squares Method

The unsatisfactory performance of SVD at certain singular points in the workspace of Titan paved way to the use of DLS method to perform the inverse kinematics. The DLS method avoids the pseudo inverse problem at singular points and gives a good approximation to update the joint angles **[Bus04]**. This method performs the inverse kinematics and at the same time optimizes the joint velocities to give the minimum position tracking error. DLS was first used in 1986 by Wampler, Nakamura and Hanufa to perform inverse kinematics. As mentioned earlier in the previous section there exists a discontinuity between the EE postion error and joint velocities. The one way to remove this discontinuity is to consider them both simultaneously. DLS method solves this discontinuity **[Ge00]**. Its criteria is based on finding the solution that minimizes the sum

$$\| \dot{x} - J\dot{q} \|^2 + \lambda^2 \| \dot{q} \|^2 \qquad \text{- (4.7)}$$

where λ is the damping factor which weights the importance of minimizing the joint velocities as well minimizing the residual error. The above equation can be equivalently represented as

$$\| J\Delta\theta - e \|^2 + \lambda^2 \| \Delta\theta \|^2 \qquad \text{- (4.8)}$$

where

$J\Delta\theta$ - represents the actual error in the EE position.

e - the error desired to be achieved

The eq (4.8) is equivalent to minimizing $\left| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta - \begin{pmatrix} e \\ 0 \end{pmatrix} \right|$.

The corresponding normal equation is

$$\begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta = \begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} e \\ 0 \end{pmatrix} \qquad \text{- (4.9)}$$

$$\Rightarrow \left(J^T J + \lambda^2 I \right) \Delta\theta = J^T e$$

The term $(J^T J + \lambda I)$ in equation (4.9) can be proved to be a nonsingular matrix. Thus the DLS solution is

$$\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T e \qquad \text{- (4.10)}$$

From eq (4.10)

$$(J^T J + \lambda^2 I)^{-1} J^T = a$$
$$(J^T J)^{-1} J^T + \lambda^2 I J^T = a$$
$$(J^{T^{-1}} (J^T J))^{-1} + \lambda^2 I J^T = a$$
$$J^T J^{T^{-1}} (J^{T^{-1}} J^T J)^{-1} + \lambda^2 I J^T = a$$
$$J^T J^{T^{-1}} J^{-1} + J^T \lambda^2 I = a$$
$$J^T (J J^T)^{-1} + J^T \lambda^2 I = a$$
$$J^T (J J^T + \lambda^2 I)^{-1} = a$$

Therefore Eq (4.10) is equivalent to

$$\Delta\theta = J^T (J J^T + \lambda^2 I)^{-1} e \qquad \text{- (4.11)}$$

In equation (4.10) $J^T J$ is of size $(n \times n)$, where *n* is the number of degrees of freedom. Factorizing the Jacobian matrix using SVD and rewriting equation (4.10) gives

$$\Delta\theta = \sum \frac{\sigma}{\sigma^2 + \lambda^2} V U^T e \qquad \text{- (4.12)}$$

which gives the unique solution achieving the desired EE trajectory from all possible joint angles. It is thus evident from eq (4.12) that pseudo inverse is one particular case of the DLS method. If the singular value is much larger than the damping factor then it is a clear case of pseudo inverse.

$$\frac{\sigma}{\sigma^2 + \lambda^2} \approx \frac{1}{\sigma}$$

If the singular value is of the order of λ, then λ in the denominator limits the high norm of that component in the solution. However the maximum value is limited to $\frac{1}{2\lambda}$ Fig 4.2

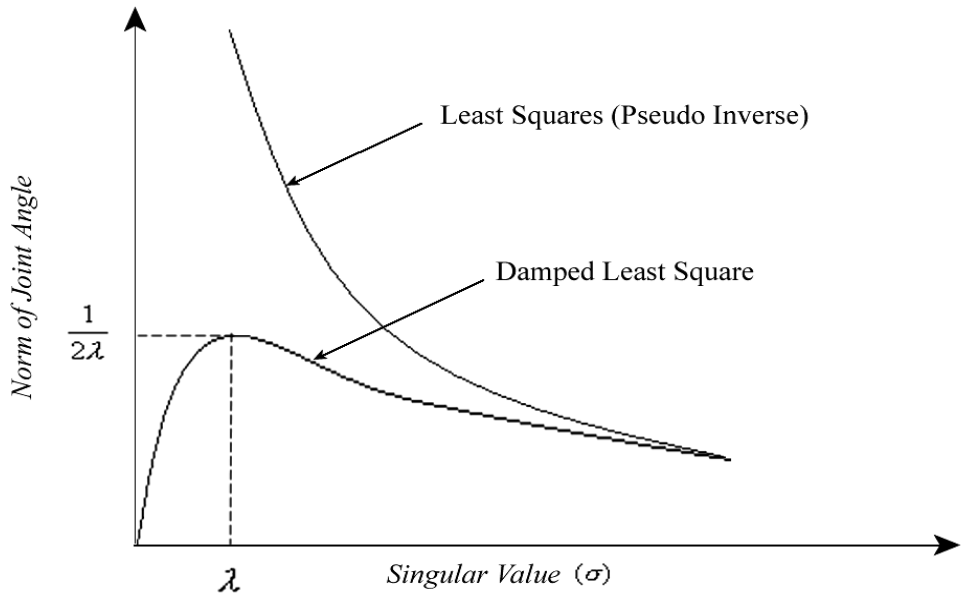

Fig 4.2. Damped Least Square Solution versus Least Square Solution

If the singular value if much lesser than the damping factor, then

$$\frac{\sigma}{\sigma^2 + \lambda^2} \approx \frac{\sigma}{\lambda^2}$$

approaches zero as σ approaches zero. This ensures the continuity in the solution despite the change in rank of the Jacobian at singular points. If the damping factor is large then the solution for $\Delta\theta$ would behave well near the singularities but if it is too large then the convergence of error might be slow. Appropriate selection of the damping factor can provide very good convergence of the error within relatively short period of time. A damping factor of *3.5* was chosen for the inverse kinematics of Titan. The DLS method is implemented in the following way to optimize the Titan Joint angles

- Select an initial position to the joint angles with $q_i = \{\theta_{1i}, \theta_{2i}, \theta_{3i}, \theta_{4i}, \theta_{5i}, \theta_{6i}\}^T$.
- From the EE velocities obtained from the WAM determine the position and orientation of the end effector.
- Determine the Jacobian matrix using the existing set of joint angles q_i.
- Determine the error matrix *e* from the desired and actual location of the EE.
- Determine $\Delta\theta$ according to equation (4.11) and find the new set of joint angles $q_{t+1} = q_t + \Delta\theta$.
- Through the forward kinematics determine the actual position and orientation for this new set of angles.
- If the error is not within desired range repeat procedure from step 4 until the error is within limits.

This entire procedure is executed for every time step so that the joint angles are optimized to their closest actual value for their current EE position and orientation.

4.3 Redundancy Resolution and Inverse Kinematics of the Master Manipulator

The most interesting part of the CSPM is the redundancy resolution of the master manipulator WAM. A redundant manipulator is one, which has more degrees of freedom than the number of variables required to define a point at its EE. So a manipulator can be intrinsically redundant when it has more than six degrees of freedom (six variables required to define a point in space). Redundancy can also result due to the task performed by the manipulator. For example WAM is a redundant manipulator of the first kind and SCARA is a redundant manipulator of the second kind. Normally the extra degrees of freedom in such manipulators are effectively used to improve the ability of the manipulator to interact with the environment, avoid obstacles, and avoid singular points. Another use of redundancy is to keep the manipulator in a configuration, which makes it as dexterous as possible. On the other hand the control of redundant manipulators is more complex than for a non-redundant manipulator.

The inverse kinematics for a redundant manipulator is not a unique solution as the number of variables is less than the number of degrees of freedom. Normally a functional constraint is introduced to perform the inverse kinematics. The number of constraints equals the number of extra degrees of freedom. There are two popular constraints used to solve the redundancy and hence find the inverse kinematics. Firstly joint limit avoidance is used as a constraint where a joint is always prevented from reaching its maximum or minimum limit at the same time the EE following the trajectory **[Sci88]**, **[Hwa97]**, **[Lau02]**. Secondly obstacle avoidance constraint is used where all the points in the manipulator are maintained at a minimum distance from the obstacle **[Bai86]**, **[Chu97]**, **[Gal01]**, **[Sci88]**. Traditionally Moore-Penrose pseudo inverse is used to solve the inverse kinematics, which gives a local minimum norm of joint velocity vector.

$$\dot{q} = J^{+}\dot{x} \qquad \text{- (4.13)}$$

The equation (4.13) can be modified by adding an extra term as

$$\dot{q} = J^{+}\dot{x} + (I - J^{+}J)\dot{q}_o \qquad \text{- (4.14)}$$

where the projection operator $(I - J^{+}J)$ selects the component of $\dot{q}_o$ in the homogenous solution space of $\dot{x} = J\dot{q}$. Therefore $\dot{q}_o$ can be selected in such a way such that it optimizes the joint velocities in eq (4.14). Let us now discuss about the inverse kinematics of WAM using different constraints.

Mapping between the operational space and joint space has always been difficult for redundant manipulators due to the presence of non-square Jacobian matrix. Since a direct pseudo inverse doesn't work at all times, an addition optimization algorithm is required along with the jacobian calculation. That is some constraints needs to be implemented, which would make the Jacobian square and non-singular. Since WAM has seven degrees of freedom against six variables (three position and three orientation variables) required to define a point in 3D space, functional constraints like joint limit avoidance and obstacle avoidance constraints are implemented **[Kle95]**, **[Chi97]**, and an extended jacobian method is followed henceforth to perform the inverse kinematics and optimize the joint angles **[Kle95]**. The inverse kinematics is considered as a dynamic problem to determine a general solution algorithm, which would require only the calculation of the forward kinematics. Many researchers have successfully performed the inverse kinematics with just the calculation of the Jacobian transpose, thus avoiding complex inverse calculations. The theory behind using Jacobian transpose to perform inverse kinematics is known and simple.

In an inverse kinematics task there is always a drift between the desired and actual trajectory obtained. In order to settle this difference it is better to resort to a solution which would take into account the operational space error. Let

$$e(t) = x_d(t) - x(t) = x_d(t) - A(q) \qquad \text{- (4.15)}$$

where

$x_d(t)$ - The desired position of the end effector at time *t*

$x(t)$ - The current position of the end effector at time *t*

$e(t)$ - Error along the time varying trajectory

Considering the time derivative of (4.15)

$$\dot{e} = \dot{x}_d - \dot{x} = \dot{x}_d - J(q)\dot{q} \qquad \text{- (4.16)}$$

For the equation (4.16) to lead to an inverse kinematics solution, it is worth relating the joint velocities $\dot{q}$ to the error *e*, so that (4.16) gives a differential equation in error. It is also important to choose this relation between joint velocities and error which will ensure convergence of error to zero. This can be derived from the principle of virtual work and generalized forces. The external force $F = \{f_x, f_y, f_z, m_x, m_y, m_z\}$ (consisting of the forces f and torques m) applied to the end-effector of the manipulator results in internal force and torque at the joints. The relation between the force *f* and the generalized torque τ is

$$\tau = J^T F \qquad \text{-(4.15)}$$

From Fig 4.3

- The range of J^T is the subspace $R(J^T)$ in R^n of the joint torques which balance the EE forces for the given manipulator posture.
- The null space of J^T is the subspace $N(J^T)$ in R^r of the EE forces which do not require any balancing torques at the joints for the given posture.

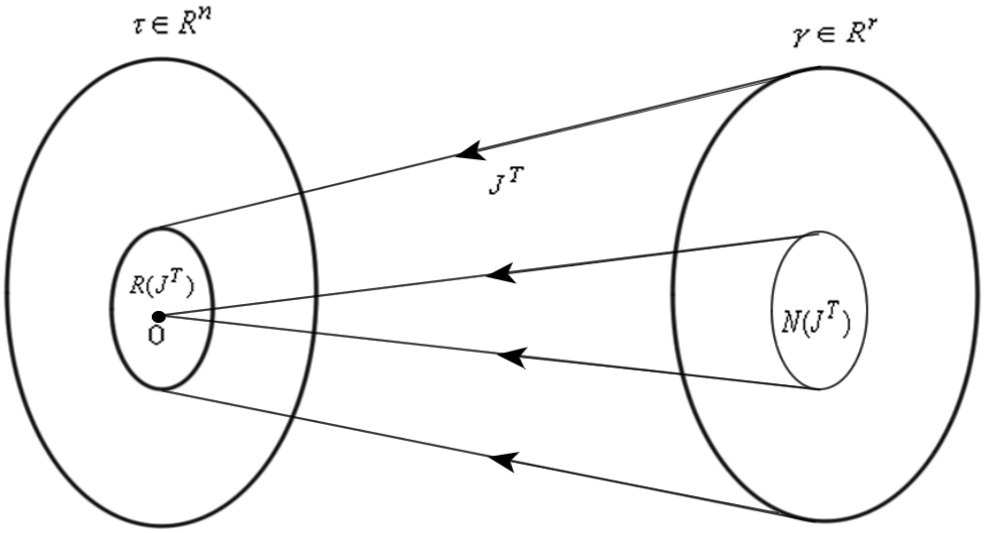

Fig 4.3 Mapping between the End Effector force space and joint torque space

It is an important fact to note that the EE forces $\gamma \in N(J^T)$ are self-absorbed in the mechanical structure so that the mechanical constraint reaction forces balance them perfectly. Therefore in a singular configuration the manipulator stays in the given posture for any EE force applied such that $\gamma \in N(J^T)$. The entire relation between the joint torques and EE forces can be stated as

$$N(J) \equiv R^{\perp}(J^T) \text{ and } R(J) \equiv N^{\perp}(J^T) \qquad \text{- (4.17)}$$

On the basis of the above duality rule an inverse kinematic solution relating the joint velocities and the error can be formulated. Consider a manipulator with ideal dynamics $\tau = \dot{q}$ (null masses and unit viscous friction co-efficients). The inverse kinematic solution $\dot{q} = J^T Ke$ plays the role of a generalized spring of stiffness constant α generating a force Ke that pulls the EE towards the desired trajectory in the

operational space. If the manipulator moves $Ke \notin N(J^T)$, then the EE follows the desired trajectory and the joint angles are determined. Therefore the inverse kinematic solution is

$$\dot{q} = J^T Ke \qquad \text{- (4.18)}$$

Since a computationally simpler algorithm has been derived to relate the joint velocities and the error without linearizing (4.16) the error dynamics is governed by a nonlinear equation. The Lyapunov error criteria can be used to determine $\dot{q}(e)$ which will ensure convergence of error **[Sci00]**, **[Sci88]**.

$$V(e) = \frac{1}{2} e^T Ke \qquad \text{- (4.19)}$$

where *K* is a positive definite matrix (usually diagonal) and

$$V(e) > 0, \quad \forall e \neq 0, \quad V(0) = 0$$

Differentiating (4.19) with respect to time and accounting for $\dot{e}$

$$\dot{V} = e^T K \dot{x}_d - e^T K \dot{x}$$

$$\dot{V} = e^T K \dot{x}_d - e^T KJ(q) \dot{q}$$

With the inverse kinematic solution derived in (4.18)

$$\dot{V} = e^T K \dot{x}_d - e^T KJ(q) J^T(q) Ke \qquad \text{- (4.20)}$$

Fig 4.4 schematically explains the inverse kinematics solution in (4.18). The current master system has a constant reference which means $\dot{x}_d = 0$. The function in (4.20) is negative definite assuming the manipulator is not in a singular configuration and

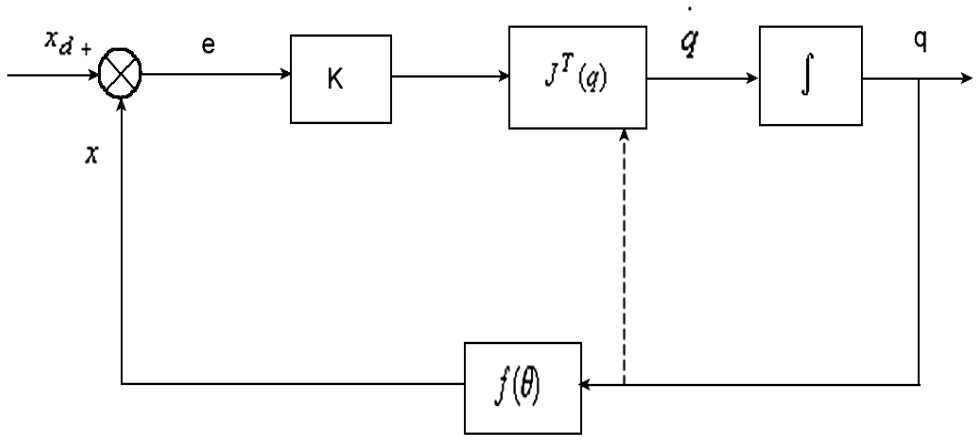

Fig 4.4. Closed loop inverse kinematics with Jacobian transpose

the Jacobian is full rank. This condition $\dot{V} < 0$ with $V > 0$ clearly shows that the system attains asymptotic stability with error approaching zero as the EE follows the trajectory. However if the error falls in the null space of J^T then the manipulator goes into a dead lock state. That is $\dot{V} = 0$ *for* $e \neq 0$ since $Ke \in N(J^T)$. At this point $\dot{q} = 0$ with $e \neq 0$. However this happens only at points not reachable by the EE from its current configuration.

4.3.1 Requirement from Constraints

One of the main concerns in using the WAM to control the slave manipulator is that the workspace of the WAM EE should comply with the human operator's convenient region of operation. The operator would be manipulating the WAM from a seated position in the Compact Remote Console (CRC) Fig 4.5.

Fig 4.5. Compact Remote Console

Therefore the region of operation is going to be restricted for the operator from that seated position. The WAM is mounted on a special tailor-made fixture attaching itself to the CRC platform on to the left side of the operator as shown in Fig 4.5. It is mounted at such a height from the ground such that the EE is at a convenient height for the operator. Therefore the constraints have to be implemented keeping in mind the required outcome of it, which is placing the EE at a convenient position to operate. This constraint could be either avoiding joint limit or avoiding an obstacle.

4.3.2 Joint Limit Avoidance

Redundancy has been effectively used to avoid joint angles from reaching their limits while the manipulator EE is chasing its trajectory. In this approach any joint of the WAM can be identified as the redundant joint and constrained **[Lau02]**. Though it is wise to select a joint, which would keep the elbow joint out of the way of the operator. That is the elbow is outstretched for all the trajectories. Hence the shoulder pitch was identified as the joint to be constrained. This occurrence of joint limit constraint can be systematically implemented in the solution algorithm of (1.4) such that it expands the task space properly. A set of operational space variables describing the current configuration of the manipulator satisfying the limits on the joints can be described. The ultimate aim now is to represent these variables as a function of the joint variables so as to obtain the augmented forward kinematics.

Let $\hat{q}(t)(n\times 1)$ be the solution to (4.4) describing a trajectory $\hat{x}(t)(m\times 1)$.

Let $e(t)$ be the error vector, which is the difference between the desired trajectory $\hat{x}(t)$ and the actual trajectory $x(t)$ obtained from the algorithm variables $q(t)$.

$$e(t) = \hat{x}(t) - x(t) \qquad \text{- (4.21)}$$

With the proper choice of joint velocities according to (4.18) the closed loop Inverse Kinematics system in Fig 4.3 can ensure that as $e \to 0$, $q \to \hat{q}$.

For the joint limit constraint a joint angle q_i is kinematically constrained between two constant extreme values. That is

$$q_{i\min} \le q_i \le q_{i\max}$$

If the joint angle exceeds the limit while the EE is following the defined trajectory then the solution explained in (4.18) is to be modified. A threshold distance $\hat{d}_q$ is defined as the allowable diversion of the joint q_i from either of its limits. So the control in (4.18) needs to be modified if the distance of the current value of the joint q_i from either of its limits becomes less than the threshold distance. The error vector is defined as

$$e_q = \hat{d}_q - d_q \qquad \text{- (4.22)}$$

where $d_q = q_i - q_{i\min}$ or $d_q = q_{i\max} - q_i$ depending on which joint limit is involved in the constraint. The error calculation involves the dynamic calculation of the distance vector at all time steps in the trajectory. Differentiating the error and analyze the error dynamics we get,

$$\begin{aligned} \dot{e}_q &= \dot{\hat{d}}_q - \dot{d}_q \\ \dot{e}_q &= \dot{\hat{d}}_q - u_i^T \dot{q} \end{aligned} \qquad \text{- (4.23)}$$

where,

$$u_i^T = \{0\ 0\ 0 \ldots\ldots \pm 1 \ldots\ldots 0\} \qquad \text{- (4.24)}$$
$$(i)$$

is the matrix which multiplies the joint velocities vector, to give the change in velocity of just the joint constrained. The + sign is used when the joint avoids $q_{\min}$ and the – sign is

used when it avoids $q_{\max}$. From (4.21) the error vector will have an additional element e_q along with other elements concerned with position and orientation of the EE, and the Jacobian matrix will have an added row u_i^T as given in (4.23).

4.3.3. Obstacle Avoidance Constraint

One of the main advantages of a redundant manipulator is that it can maneuver well in an environment using its redundant degree of freedom and avoid many obstacles in its trajectory. When the EE is tracking a collision free trajectory one or more of the links might be very close to an obstacle. In such situations the inverse kinematics in (4.18) gives joint angles, which are very close to each other. As the manipulator moves along the trajectory one or more constraints needs to be implemented in order to avoid obstacles. For this purpose obstacles are modeled as convex volumes in 3D space. That is they are assumed to be made up of simple structures like cubes, spheres and cylinders so that the exact distance between the obstacle and the manipulator link can be determined. A link is said to have avoided an obstacle if the distance from that link to the obstacle is over a predefined threshold distance just as described in joint limit avoidance. If all the links satisfy this condition then equation (4.18) can be used to perform the inverse kinematics. However if the distance is less than the threshold distance then the current solution is to be modified by setting *n-m* constraints.

Let us consider the example of a four link planar arm to exemplify the obstacle avoidance criteria. Fig 4.6 elucidates the geometry of the four link planar manipulator with a link nearest to the obstacle. The objective here is for the link l_i to avoid the obstacle while the EE is tracking its trajectory.

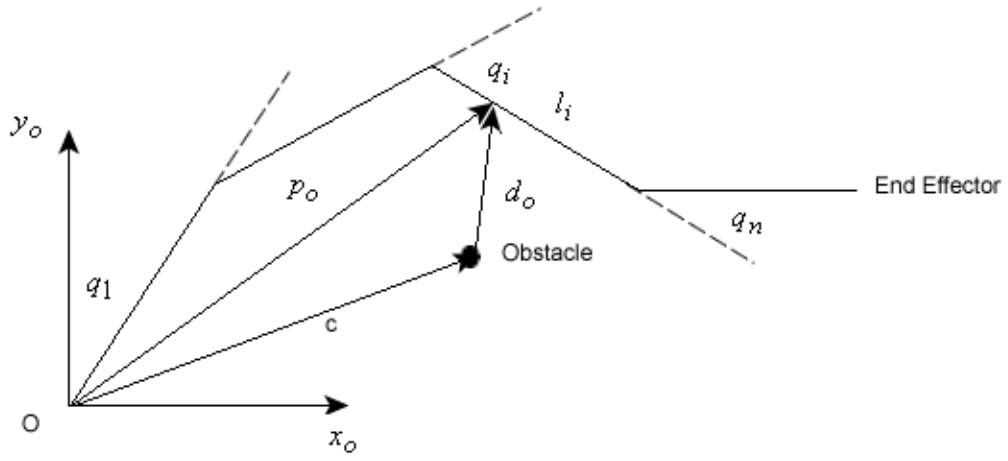

Fig 4.6 A planar manipulator near an obstacle

Let $\hat{d}_o$ be the threshold distance the link always has to maintain from the obstacle. Let *c* be the co-ordinates of the obstacle. It is always possible to calculate the minimum distance of the link from the obstacle at all times. Let p_o be the position vector of the obstacle avoidance point. Both points p_o and *c* are with respect to the same base frame as the kinematics of the manipulator are described. The position of the minimum distance point from the obstacle keeps changing as the manipulator moves around the obstacle. Therefore p_o has to be dynamically calculated throughout the trajectory. Let d_o be the distance vector between the obstacle and the obstacle avoidance point and it is given by

$$d_o = p_o - c \qquad \text{- (4.25)}$$

Once $d_o < \hat{d}_o$, then there is a peril of collision, and the joint velocities have to be changed according to the new constraints. The error based on which the joint velocities are updated is given by

$$e_o = 0.5(\hat{d}_o^2 - d_o^T d_o) \qquad \text{- (4.26)}$$

Differentiating (4.26) we get,

$$\dot{e}_o = \hat{d}_o \dot{\hat{d}}_o - d_o^T \dot{c} - J_{do}^T \dot{q} \qquad \text{- (4.27)}$$

where

$$J_{do}^T = d_o^T J_{po} \qquad \text{- (4.28)}$$

where $J_{po} = \frac{\partial p_o}{\partial q_i}$ is the Jacobian of the obstacle avoidance point. In order to find the solution to the inverse kinematics problem *e* will indicate an extended error vector with the additional component e_o and the Jacobian indicates an extended matrix with an additional row given by J_{do}^T. By doing this the movement of those degrees of freedom, which decides the position of the obstacle, avoidance point p_o is broken preventing the link from hitting the obstacle. In fact the link is forced to move in a tangential path as if around a sphere centered at *c* and with radius $\hat{d}_o$.

4.3.4 Joint Volume Limitation Constraint

The Joint Volume Limitation Constraint (JVLC) as it is called is based on the obstacle avoidance criteria. It is yet another method to input a constraint in to the inverse kinematics solution just like the joint limit avoidance and obstacle avoidance. The

main concern with manipulating the WAM is to keep the EE in a convenient region of operation for the operator. One observation made from manipulating the WAM is that, it is very difficult for the operator to hold the shoulder from falling. In order to provide an effortless manipulation to the operator it is very important to prevent the shoulder from falling. One disadvantage with implementing Obstacle avoidance mentioned in the previous section in the master manipulation control is that it prevents the free falling shoulder only in that direction towards the obstacle. In all other directions beyond a certain shoulder pitch angle the gravity pulls the arm. One way to solve this is to restrict the region of movement of the Joint four in all directions. That is exactly what the name JVLC suggests. JVLC is very much similar to the obstacle avoidance criteria except for the error definition and criteria for activating the constraint.

Let us consider Fig 4.6 to explain the method. Let us assume that the joint q_i volume is to be constrained. Let *c* be the initial location if this joint for the given set of initial angles for all the joints. Let $\hat{d}_o$ be the threshold distance, which the joint q_i should never exceed from its initial location *c* and d_o be the distance of the joint position from the initial location $d_o = p_o - c$. The error vector is defined as the difference between the threshold distance and the dynamic distance calculated $e_o = \hat{d}_o - d_o$. As long as $e_o \geq 0$ the joint angles are determined from (4.18) but once the error becomes negative then the same set of constraints as explained in the previous section are applied to the inverse kinematics solution. This approach is further explained in the results chapter. The following section describes about the extended Jacobian and error.

4.3.5 Extended Jacobian and Error Calculations

On the basis of the constraints discussed above, the solution algorithm for the inverse kinematics is established for WAM in this section. For this purpose the task space vector *x* is extended into a $((m+v)\times 1)$ vector *y* as

$$y = \begin{bmatrix} x \\ x_o \\ x_q \end{bmatrix} \qquad \text{- (4.29)}$$

where x_o is a $(k \times 1)$ vector whose components are of the kind $d_o^T d_o$ as explained in (4.26) for every obstacle avoidance constraint and x_q is a $(r \times 1)$ vector whose components are of the kind d_q as explained in (4.22) for every joint limit avoidance constraint. Correspondingly the task space reference vectors are given by

$$\hat{y} = \begin{bmatrix} \hat{x} \\ \hat{x}_o \\ \hat{x}_q \end{bmatrix} \qquad \text{- (4.30)}$$

with $\hat{x}_o$ *and* $\hat{x}_q$ carrying obvious meanings and $v = (k + r) \leq (n - m)$. The joint velocities are then derived in a similar fashion to (4.18) as

$$\dot{q} = \alpha J_e^T e_y \qquad \text{- (4.31)}$$

where

$$J_e = \begin{bmatrix} J \\ J_{do} \\ U \end{bmatrix} \qquad \text{- (4.32)}$$

is the extended Jacobian with the end effector Jacobian matrix *J*, the Jacobian matrix $J_{do}(k \times n)$ as defined in (4.28) and the matrix $U(r \times n)$ as defined in (4.24). The corresponding extended error matrix is

$$e_y = \begin{bmatrix} e \\ e_o \\ e_q \end{bmatrix} \qquad \text{- (4.33)}$$

which includes the EE error vector *e*, the obstacle avoidance error vector e_o as explained in (4.26) and the joint limit avoidance error vector e_q as explained in (4.22). The inverse kinematics solution given in (4.31) causes the EE to track the trajectory at the same time use the extra degrees of freedom to avoid an obstacle in the workspace or a joint limit, which is not desirable.

Though this method provides good solution to joint angles further research is involved on when a constraint should be activated. Usually a constraint is activated when the threshold distance is exceeded in both the cases. But when a threshold is not wide enough then an obstacle could be hit or a joint could have exceeded its limit. Therefore a proper and a practical way of selecting the range for the threshold should be adopted for the best use of this algorithm. Moreover a proper criterion to activate and deactivate the constraints should also be adopted because introduction of constraints at a wrong time might actually blow up the errors. Therefore this selection depends more on the practical application of the manipulator. Further to this all the three constraints are discussed and the performance analyzed in the results chapter.

Chapter 5

RoboWorks

RoboWorks is a simulation package, which can be used to model 3D mechanical objects and animate them. It is a user-friendly software which allows the user to model 3D objects conveniently. It is a useful tool for robotic engineers as there is a need to simulate the robots' movements from a control program. Especially for this particular task of controlling the mobility of a manipulator using another, RoboWorks is of immense use, as a prior knowledge of the response of the slave manipulator can be known from the simulations. This chapter explains how the two manipulators are modeled and simulated in RoboWorks.

5.1 Creating Models in RoboWorks

The graphical interface of RoboWorks consists of

- 3D view, which shows the 3D model of the parts created and their position with respect to each other.
- A tree view showing in order, the functions performed in modeling the entire manipulator
- A menu bar providing all the basic functions and editing commands to enable modeling an object.

There are three basic functions used to create a 3D object. They are the Shape, transformation and other. The shape consists of the basic shapes like cube, cylinder,

sphere and discs used to model objects. The transformation consists of the possible movement an object can undergo such as rotation and translation. The other consists of the miscellaneous functions like selecting the material for the object and modifying its appearance. The shape feature allows the user to select the basic shape to be modeled and its size. Once the size is decided the location of the object can be selected too from the same window. Fig 5.1 and 5.2 shows the windows which allows the user to select the shape, size and location of the object. The location of the 3D object refers to the location of the center of gravity (COG) of the object. So if an object has to be placed adjacent to a previously modeled 3D object, then the location of the COG of the latter should be appropriately offset from the former. Any complex shape can be modeled by assembling the basic shapes appropriately. The entire model can be made more appealing by choosing different materials for different structures.

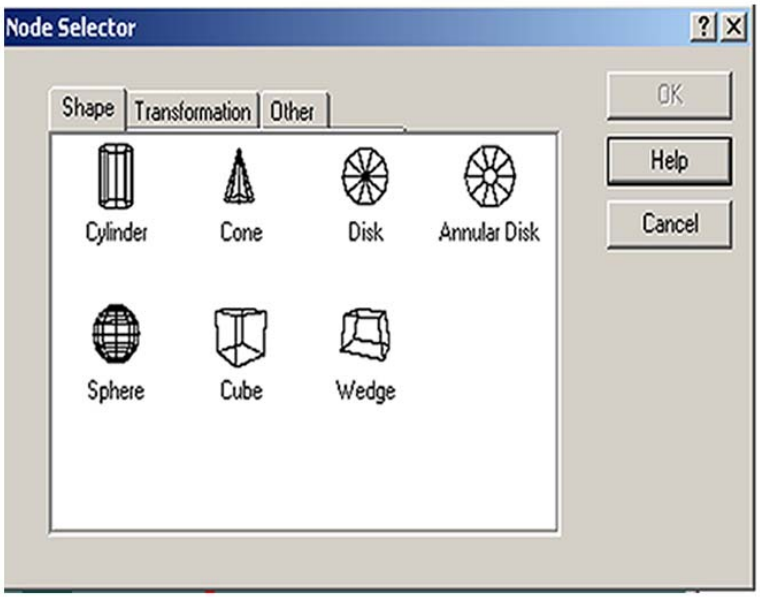

Fig 5.1 Selection of the basic 3D shape

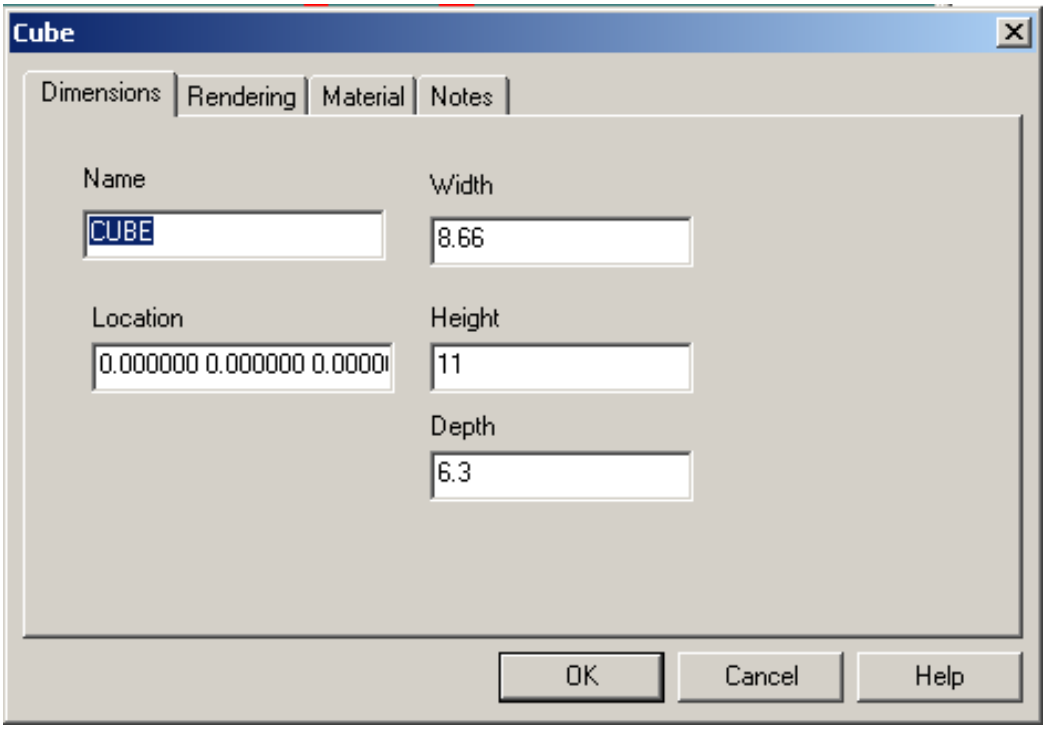

Fig 5.2 Size and location of the object

A normal procedure before modeling the object is to select the material for it. Selection of material is done from the “Other” pane in Fig 5.1. The material selection can also be done from the “Material” pane in Fig 5.2. Once the object has been selected it appears on the 3D view window and the name of the object appears on the “Tree View” Fig 5.3. As the models are built, the tree grows. A proper procedure is to group the objects which constitute each link. In this way, other 3D structures which constitute the entire manipulator are modeled and placed accordingly with respect to each other. The important function, which should be considered here, is the transformation. The transformation enables the user to define a joint, revolute or prismatic, and also placement of the links with respect to each other. Further insight into the transformations is provide in the next section.

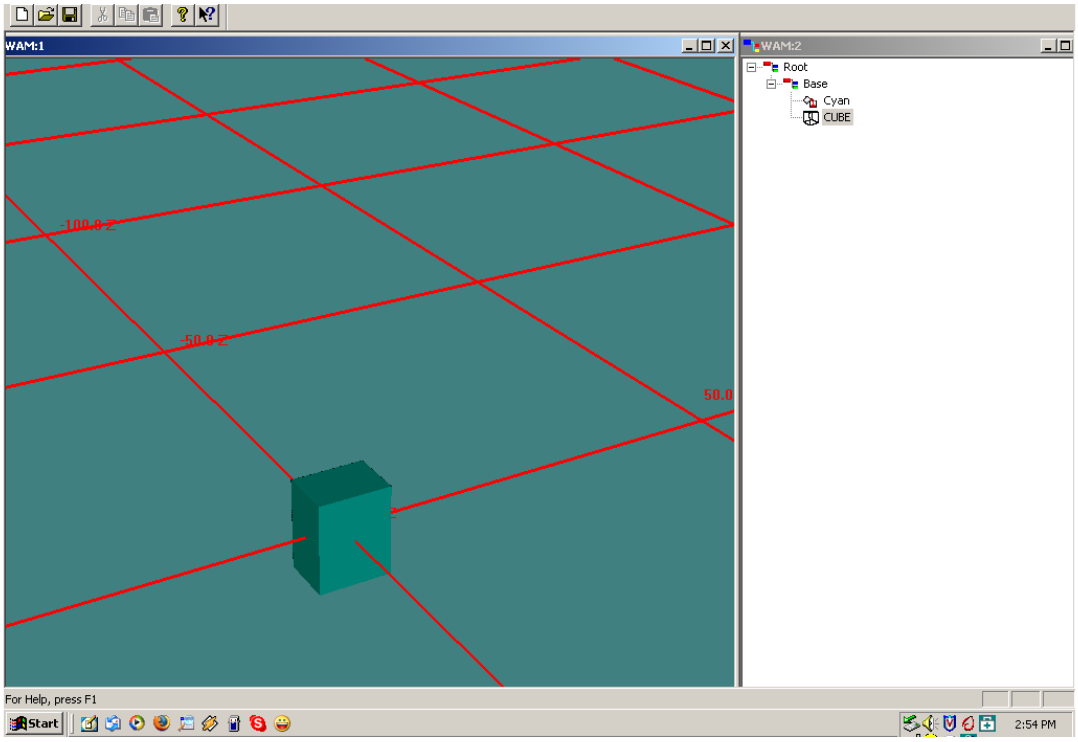

Fig 5.3 3D View and Tree View of the object created

5.2 Transformations

Transformations in RoboWorks are of two types, static and dynamic. Every object created in RoboWorks has a defined origin for itself. In order to place a second object with origin displaced from the previous static transformations are used. The two basic static transformations are rotation and translation. The translation displaces the origin in x, y, and z directions. However if the origin is displaced in all the three directions the translation should be done separately for the each co-ordinate displacement. Secondly when an object is rotated permanently in 3D space with respect to the other rotation is used to effect it. Fig 5.4 shows the static rotation and translation functions. The static rotation enables the object to be rotated about all the three co-ordinate axes and with

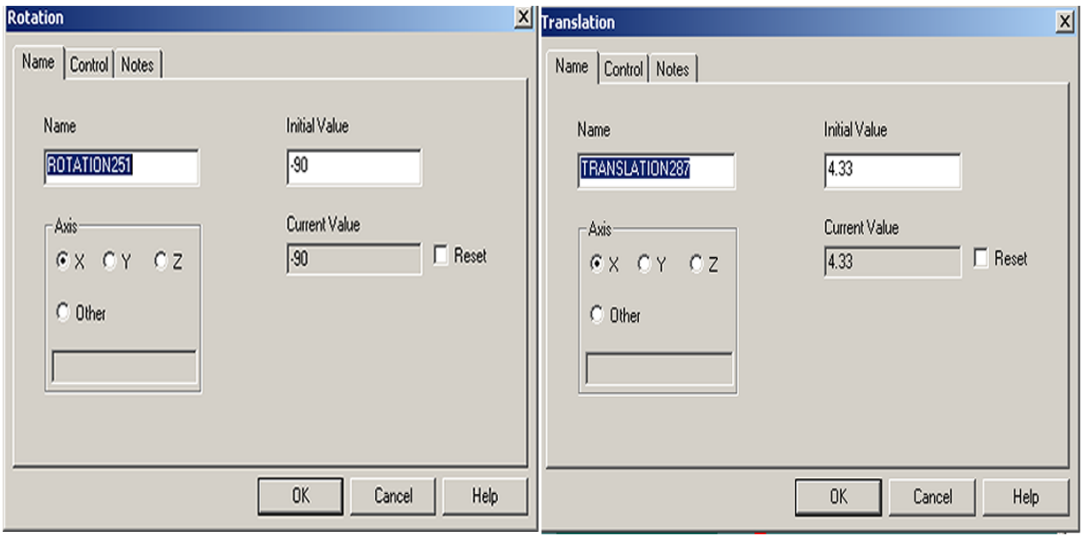

Fig 5.4 Rotation and translation of 3D objects

any defined angle. Since objects are grouped for each link a transformation start is placed at the beginning and a transformation stop is placed at the end. In this way any rotation or translation is made effective only to that group objects and not the rest.

5.2.1 Dynamic Transformations

Dynamic transformations do not have a specific value associated with them like static transformations. For dynamic transformations the magnitude of the transformation is a variable whose value is input by the user. This allows us to animate the objects by means of another program, the keyboard or a file generated by the user. As with static transformations, a dynamic transformation can translate, rotate and scale the model. The options “Transformation Start” and “Transformation Stop” again allow us to return to a point of reference given before the transformation and they allow us to animate several objects in the model independently. For example, the first joint in the WAM, Fig 5.5, has a dynamic transformation attached to it, which allows it to rotate with respect to the base structure.

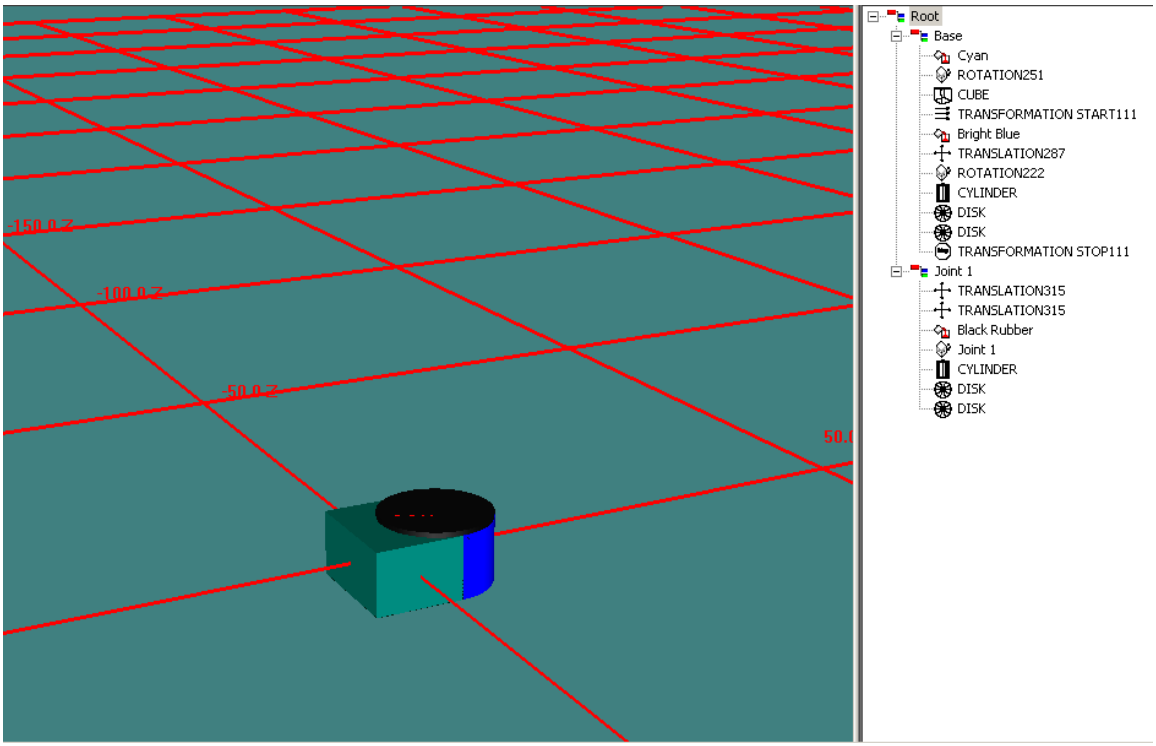

Fig 5.5 Dynamic transformation applied to a joint

Fig 5.5 depicts how a rotational joint of WAM is modeled in RoboWorks. The black cylindrical structure represents the first joint rotation. Usually a "Transformation Stop" is not used for a dynamic transformation since the other structures following this has to obey the same transformation. That is, the links which are on top of the first joint should rotate along with the joint.

5.3 Animation

There are three ways in which a model can be animated in RoboWorks: The keyboard, a file and data through another program. The simplest of all the three is through keyboard. Keyboard animation involves assigning a key for every joint. Joints

are animated when the corresponding key is pushed after selecting the 3D view. The second method is to feed the dynamic transformation parameters through a file. A file player reads the parameters from the files and feeds it to the corresponding joints. The third method is directly from another program. One of the advantages of RoboWorks is that it can communicate with other applications running simultaneously. RoboTalk enables this feature through a set of library functions that allows the user to generate trajectories from another program. The program creating the trajectory includes the RoboTalk header file in it. The main commands of RoboTalk which enables this feature are

- Connect: Which sets up the communication between the RoboWorks and the control program.
- Disconnect: Which disconnects the program from RoboWorks.
- Get Tag Values: Gets the values of the tags defined in the transformation while creating the model.
- Set Tag Values: Sends the corresponding values to the tags to animate the model.

Fig 5.6 shows the arrangement of the links and joints in WAM as modeled in RoboWorks. The model conforms to the real dimension of the WAM in order to simulate and understand the redundancy of the arm perfectly.

Fig 5.7 shows the joints and the arrangement of the links in Titan as modeled in RoboWorks. The black colored 3D structures represent the joints of the manipulator. This manipulator has been modeled as holding a tool with its gripper. This is to clearly view the trajectory of the tip of the tool as the joints move with respect to each other.

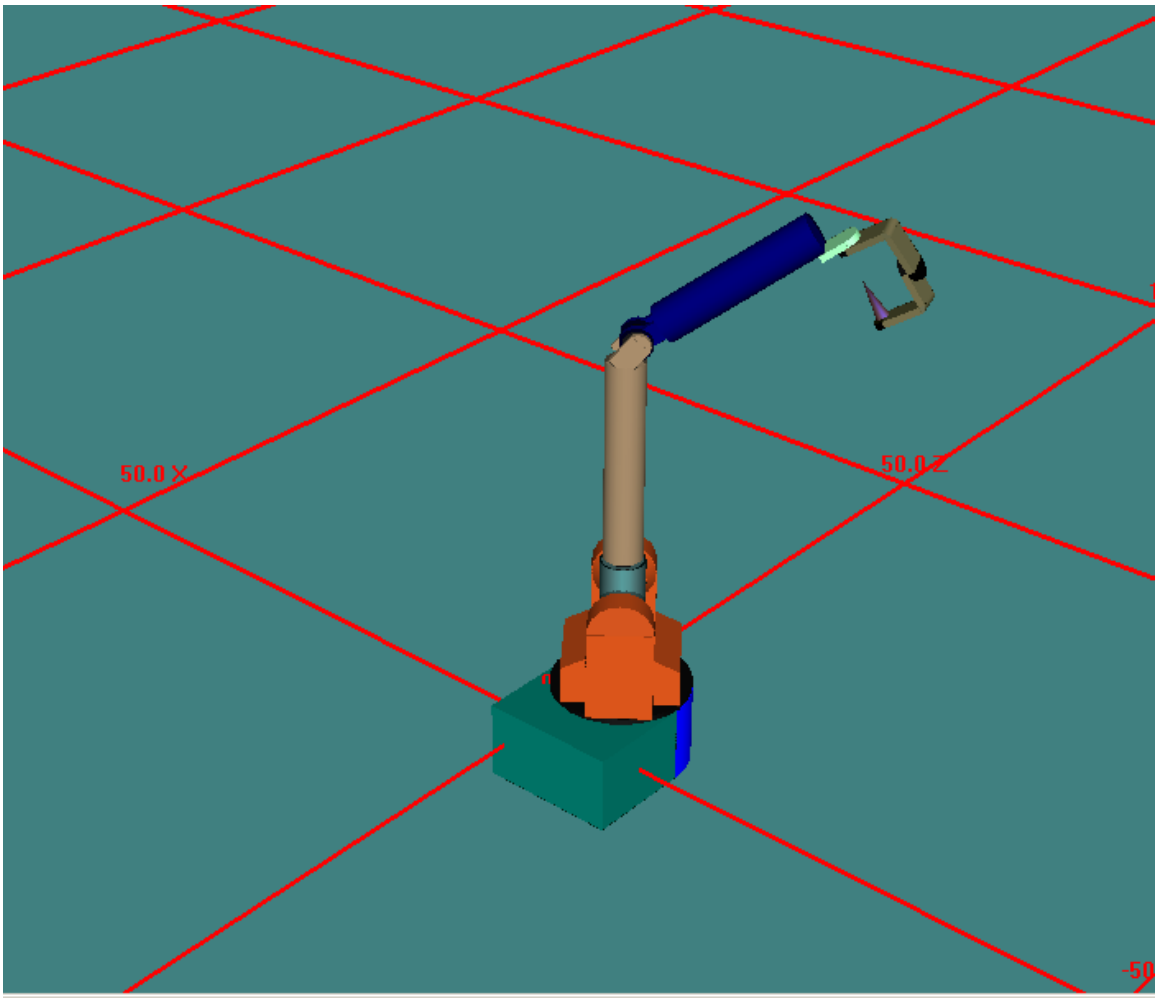

Fig 5.6 RoboWorks model of WAM

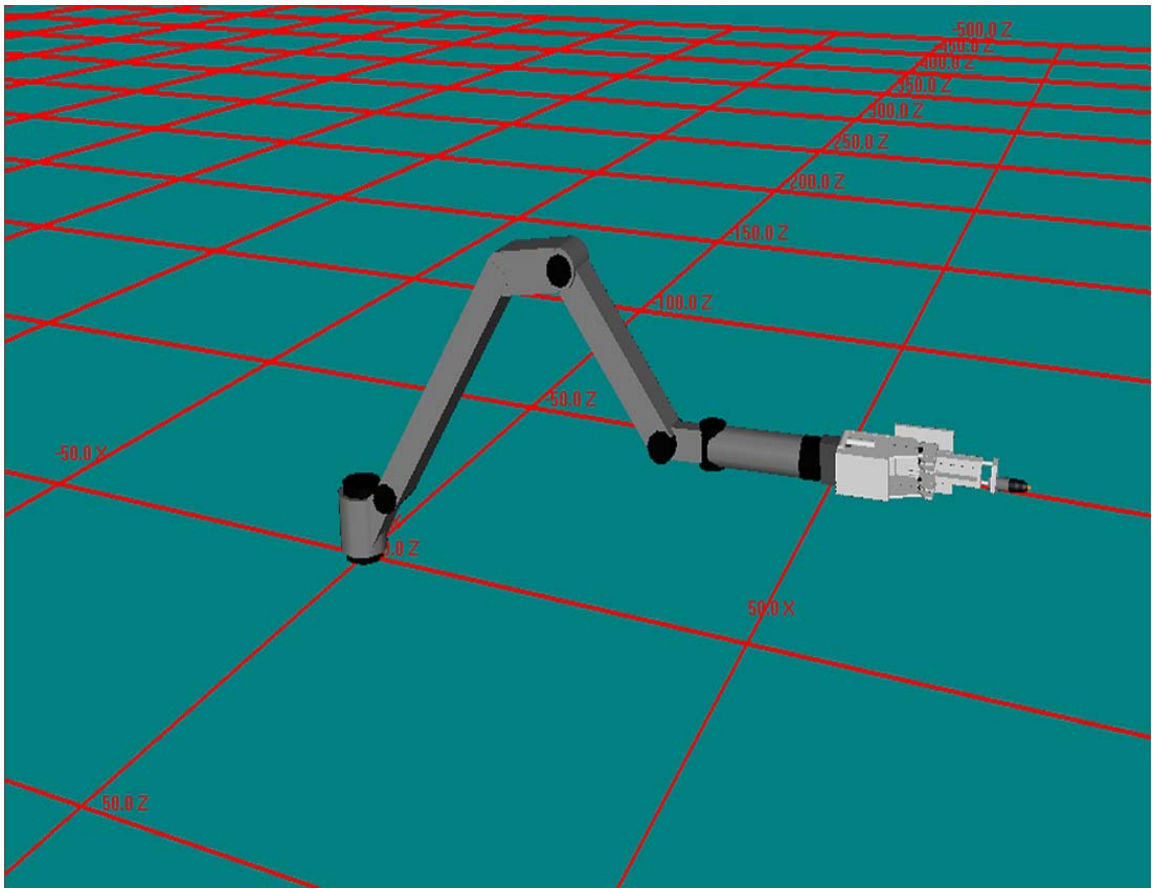

Fig 5.7 RoboWorks model of Titan

Chapter 6

Simulation Results

The previous chapters provided an insight view on the kinematics of the manipulators and various methods to perform the forward and inverse kinematics to obtain a Cartesian space position mapping. The ultimate goal was to manipulate the Titan with the help of WAM. In this chapter, we will discuss about each proposed method to perform the inverse kinematics for both the manipulators, based on their performance and error, and identify the best out of them. Various trajectories have been generated and simulated in RoboWorks. It is very important to notice that even the master manipulator has error graphs, because for the sake of animating the master trajectory, it was commanded to move autonomously. Ideally in case of teleoperation, there is no error involved with the master trajectory. This was done to identify the best criterion to perform the Inverse Kinematics for WAM. However error graphs for the Titan makes perfect sense as it depicts how well the Inverse Kinematics performs in tracking the desired trajectory. Before analyzing the results, let us discuss about the performance criteria.

6.1 Performance Criteria For WAM

The idea of setting up constraints to perform the inverse kinematics for WAM is to resolve redundancy and simultaneously provide a convenient range of operation for the operator. One of the main concerns in setting up the constraint is the placement of the WAM in its fixture in the CRC relative to the operator's seat. This creates a restrictive range of operation as far as the WAM is concerned. The operator seated in CRC, will be unable to sweep the entire workspace of WAM. This depends on the range of operation

of the operator's arm, when seated in that position. There exist a three dimensional space in front of the operator, which he/she is comfortable working in. Given this, our goal is to set up constraints on the WAM such that, the operator is comfortable manipulating the EE of WAM without encountering any redundancy problems. That is, this three dimensional space should be the union of the workspace of the EE of WAM and the operator's arm.

Fig 6.1 shows the CRC with the WAM mounted on it. It is clear from the figure that the WAM has a restricted workspace to operate on. This is one reason to have indexing between the two manipulators, since the master manipulator has a smaller workspace compared to the slave. One other major factor in setting up the constraint is the friction at joints. Since WAM is a redundant manipulator, certain joints are free to move for a fixed end effector location, which means there are infinite number of configurations available for a given EE position and orientation. This causes a great inconvenience to the operator in effectively manipulating the arm. This might also cause high joint velocities at the slave side. In order to avoid this unfavorable scenario, constraints have to be set up to restrict certain joints within a specified region or joint limits. For this reason various constraints like joint limit avoidance, obstacle avoidance and joint volume limitation have been considered. This chapter analyses the effect of each constraint in providing a convenient workspace to the operator.

6.1.1 Joint Limit Avoidance

One of the constraints to maintain the EE of WAM at a convenient location is to limit a joint to a certain angle. Generally the redundant degree of freedom is limited to a certain range. In WAM any joint can be taken as the redundant joint **[Lau02]**. Though it is pragmatic to limit the shoulder pitch and avoid free falling of the first link. Therefore a suitable angle was identified through experimentation with the WAM and set as limit. The shoulder pitch was constrained not to exceed -30^{o}. The Inverse Kinematics

Fig 6.1 Compact Remote Console with the master manipulator WAM

explained in (4.18) was used as long as the constraint was satisfied. Once the constraint was violated, the control algorithm in (4.31) applied with (4.22) and (4.24) was used to determine the change in the joint velocities. This process continues throughout the entire trajectory. The following graphs depict the variation of EE position error and the shoulder pitch angle with time.

Fig 6.2 depicts the variation of the actual trajectory from the desired trajectory for WAM when joint limit avoidance constraint was used. It is evident that the performance was not good in terms of EE position tracking. The EE was commanded to move in the positive Y direction at 5cm/sec. Each times step was *10* milliseconds long. The error at the beginning of the trajectory is too high to fall in the acceptable limits.

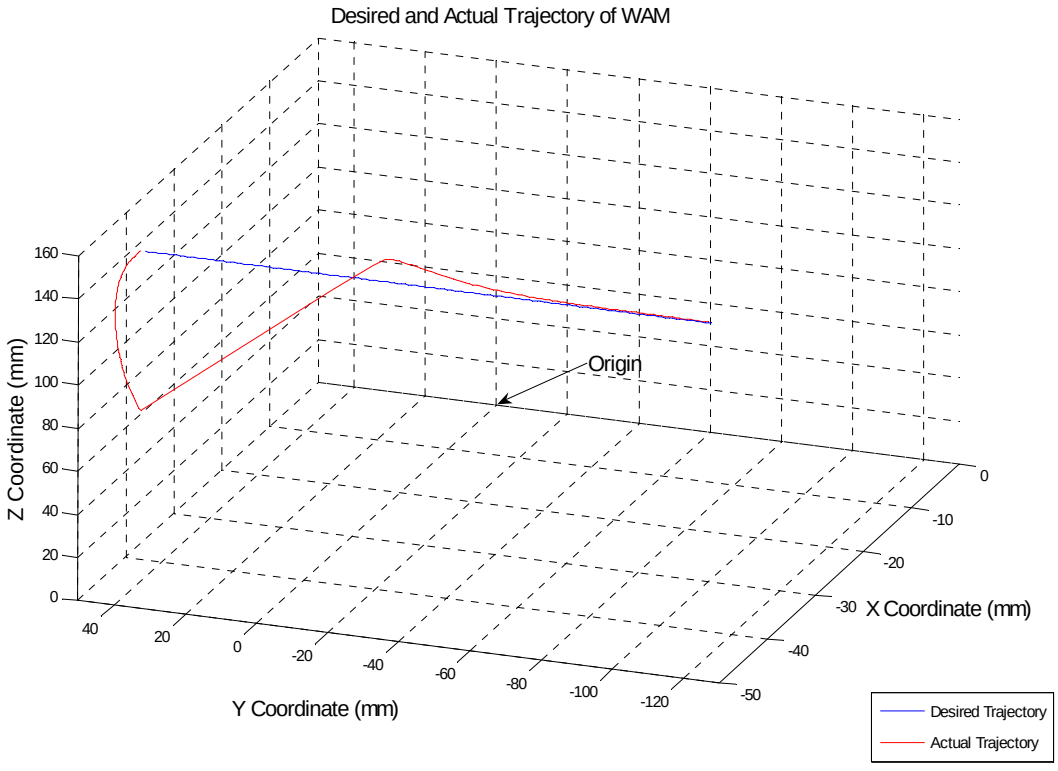

Fig 6.2 Comparison of desired and actual trajectory for WAM

Fig 6.3 shows the variation of the position error of the WAM End Effector in all the three co-ordinate axes. It can be seen from the graph that error initially started from zero for the first time step, when the manipulator started from its initial location. The initial location was selected keeping in mind that none of the joints would reach their physical limits for all kinds of possible trajectories except for the case of the constraint. As the EE followed the trajectory simultaneously satisfying the constraint, the error increased. Once the constraint was violated the extended Jacobian and error method took over locking the shoulder pitch at -30^0. That is when the error immediately tries to reach zero and thereafter remain there. This is evidently seen in Fig 6.3 where all the three co-ordinate errors change direction suddenly.

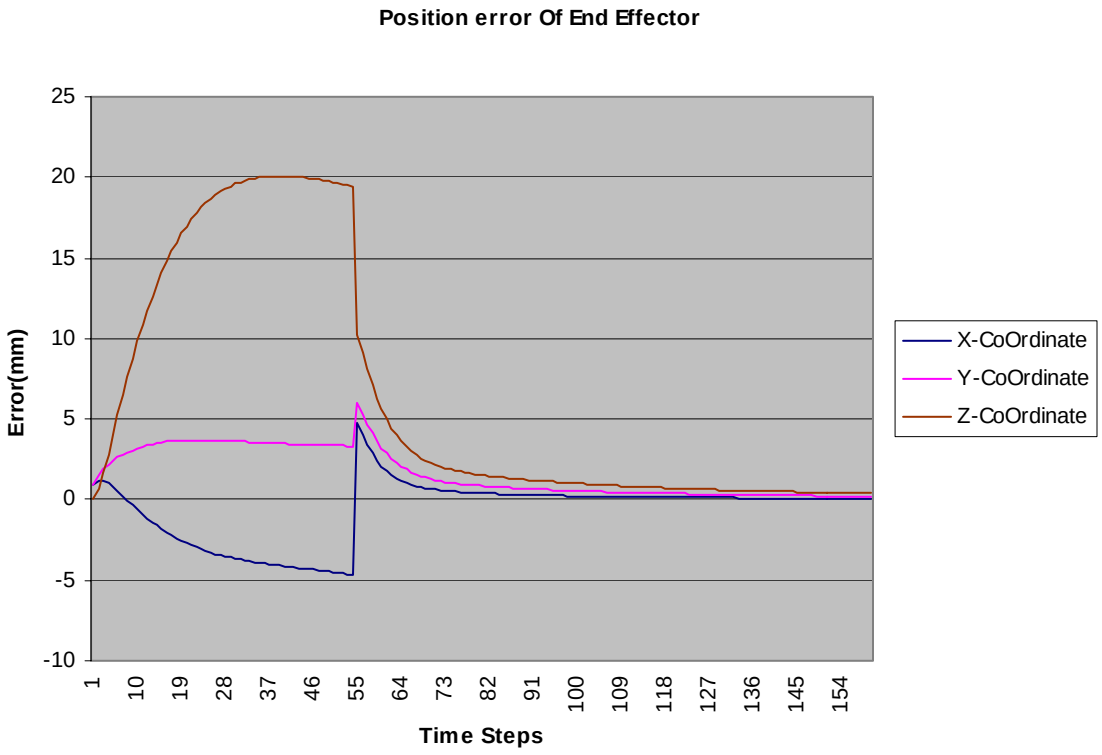

Fig 6.3 Variation of position error of WAM EE throughout the trajectory

Fig 6.4 shows the variation of the shoulder pitch angle with time. This corresponds to the same trajectory and time step as in Fig 6.3. As long as the constraint on the shoulder pitch angle is satisfied, the shoulder pitch angle approaches the threshold value. Once it crosses the threshold slightly, the revised algorithm maintains the joint at that threshold value. This is similar to locking a joint and making WAM a six-degree of freedom manipulator. The overshoot in the graph is because; within one time step the joint value exceeded -30^{o} and immediately falls back to it.

The graphs Fig 6.5 and 6.6 show the error and joint angle variance for a different trajectory. The EE was commanded to move in the negative X direction at 5 cm/sec. The threshold value for the shoulder pitch was the same as in previous case.

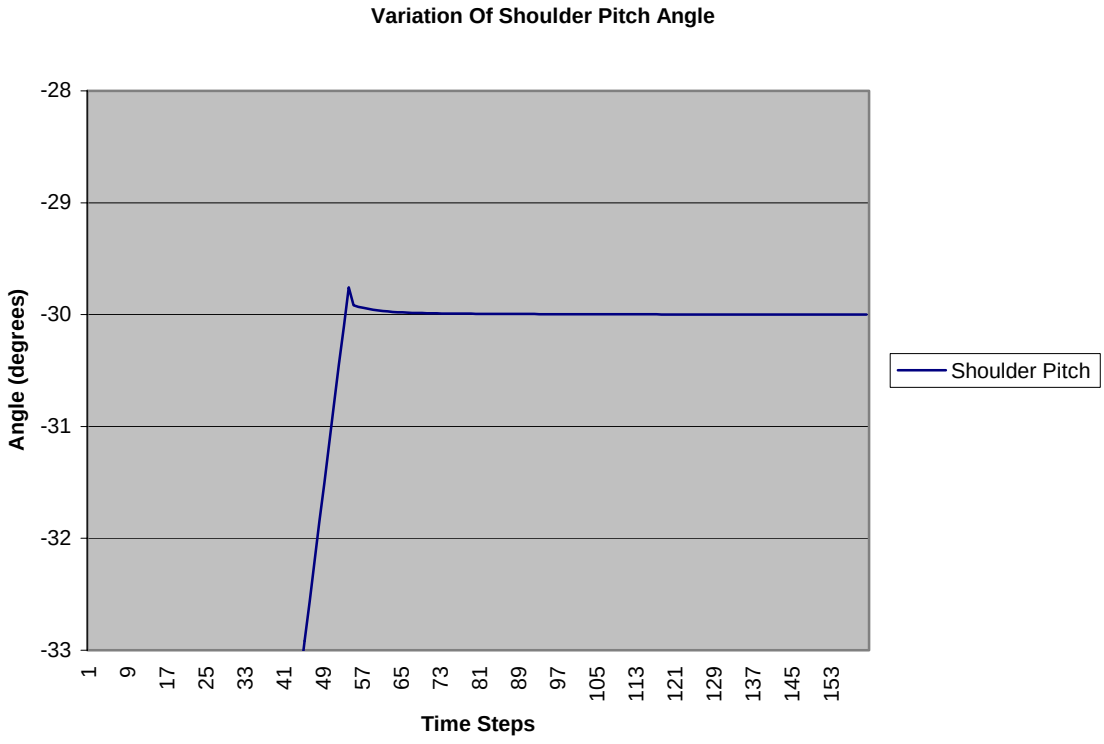

Fig 6.4 Variation of shoulder pitch angle throughout the trajectory

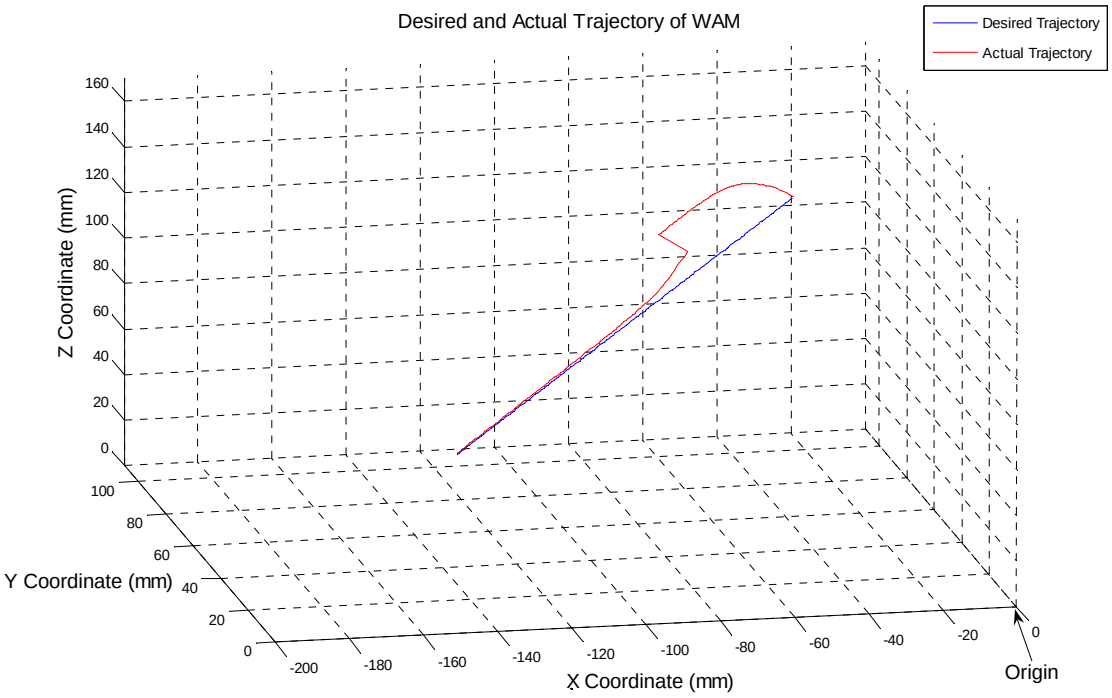

Fig 6.5 Comparison of desired and actual trajectory for WAM

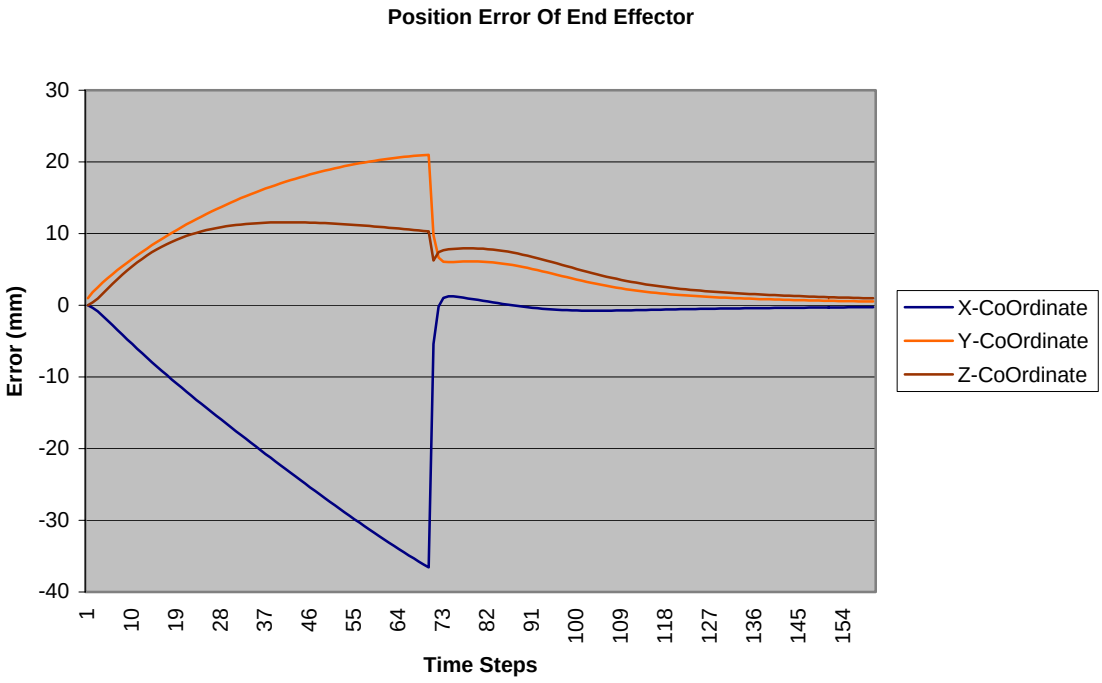

Fig 6.6 EE position error variation for the second trajectory

It can clearly be seen from the joint limit avoidance constraint performance that there is a drastic difference in the EE error within a very short period of time. It is dangerous to have such a quick change as it reflects badly on the joint velocities except the joint, which is constrained. Fig 6.4 confirms this occurrence. The joint velocities have to be very high to accommodate such a change in the EE position. This is an undesirable response, as the operator will experience a sudden shock while operating the manipulator. Moreover, as the Titan is imitating the movements of the WAM continuously, this high velocity occurrence may damage the manipulator. For this reason the joint limit avoidance constraint is not used to perform the inverse kinematics.

6.1.2 Obstacle Avoidance Criteria

Obstacle avoidance has been traditionally used to resolve redundancy for many years. In order to maintain the EE in a three-dimensional space in front of the operator, an obstacle is virtually placed at the armrest in the operator's seat in the CRC. The idea is that as the manipulator link avoids the obstacle, the EE is at a convenient location to the operator. Also as the obstacle is placed at the armrest, the free falling of the link towards the operator due to less joint friction, is avoided. The obstacle can be placed anywhere in the workspace of the WAM. From repeated experiments, the first link was observed to fall towards the operator as he/she tried to pull the arm.

One good criterion to avoid this would be to restrict the shoulder pitch angle just like in the previous method. However the joint was not locked but rather restricted in one direction. The obstacle is assumed to be a sphere of certain radius. Fig 6.6 shows the virtual sphere centered at the geometric center of the armrest. The radius of the sphere is however decided based on the shoulder pitch angle. That is the angle at which the shoulder pitch needs to be restricted decides the radius. After few experiments on the arm

the radius of the sphere was fixed at eight inches. As per Fig 6.7, with the current location of the chair, WAM and the armrest, the shoulder pitch does not exceed beyond 15^{o}.

The point on the manipulator which avoids the obstacle at all times, is at a distance of 0.45 meters from the intersection of the first three joints. That is, this point lies in between the joint four and the J1, J2 and J3 intersection. This method is slightly different from the conventional obstacle avoidance method. In the latter the minimum distance between the manipulator links and the obstacle is not to go lesser than the threshold distance so that all the links avoid the obstacle. In the former case, only one point on the manipulator is maintained above the threshold distance from the obstacle. The threshold distance in our case is the radius of the sphere. Therefore as the obstacle avoidance point approaches the obstacle it is directed to move around the sphere smoothly. The position of the obstacle avoidance point in the manipulator depends on the position of the WAM relative to the chair. The obstacle is considered as a sphere traditionally because, the idea is to avoid the obstacle in all directions. The performance of this method is analyzed from the following graphs. The performance analysis is again done based on the EE position tracking error and the simulation just like in the previous method.

Fig 6.8 shows the difference in the desired and the actual trajectory of the WAM when it was commanded to move in the positive Z direction at *4 cm/sec* with each time step maintained at *10* millisecond. It is clear from the graph that the EE tracking was very good compared to the previous method. It is evident from the graph that the actual trajectory followed the desired trajectory with minimum error. The desired trajectory alone can be seen in the graph as it got superimposed on the actual trajectory. The error analysis is done on the master manipulator to determine how good the method performs in EE tracking in the autonomous mode so that it behaves the same way in the teleoperation mode. It is just to identify the best method to perform the redundancy resolution.

Fig 6.7 Pictorial representation of obstacle avoidance constraint

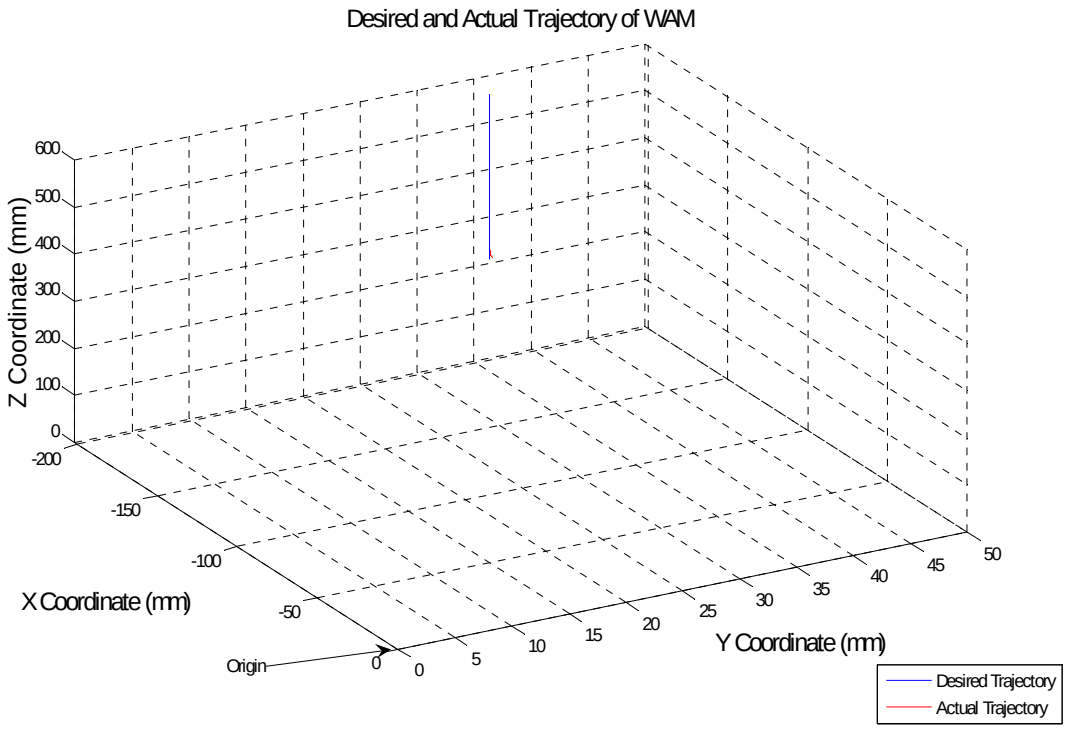

Fig 6.8 Comparison of the desired and actual trajectory of WAM for obstacle avoidance

Fig 6.9 depicts the variation of the EE position error in all the three co-ordinates with time. It can be clearly seen that the overall error is very small compared to the previous approach. The error started with a higher value and asymptotically approached zero. Another important observation is that, there is no drastic change in the error as there was for the joint limit approach. This causes a slow change in the manipulator configuration. The slave manipulator is also safe from vibrations and high joint velocities. The operator would feel comfortable working with for such a gradual shift in error.

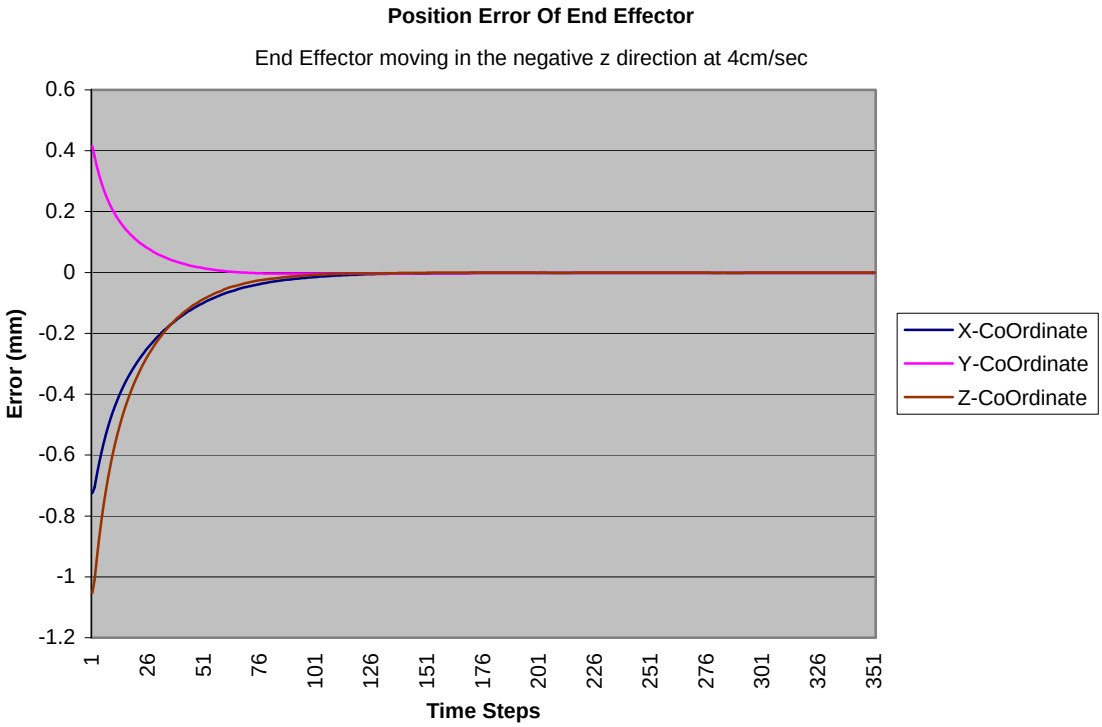

Fig 6.9 Variation of EE position error throughout the trajectory

Fig 6.10 shows the variation of the distance between the obstacle avoidance point and the obstacle, with time. The obstacle is the center of the sphere, which falls on the geometric center of the armrest. Initially when this distance was less than the threshold distance the extended Jacobian algorithm explained in (4.31) applied with (4.16) and (4.28) performed the inverse kinematics, thereby increasing the distance to the threshold. Once it has reached the threshold then the control algorithm in (4.18) takes care in maintaining the distance at the threshold. In this way the EE follows the commanded trajectory simultaneously satisfying the constraint. The graphs Fig 6.11, 6.12 and 6.13 depict the performance of this method for a different trajectory. The End Effector was commanded to move in the positive Y direction at *5 cm/sec.*

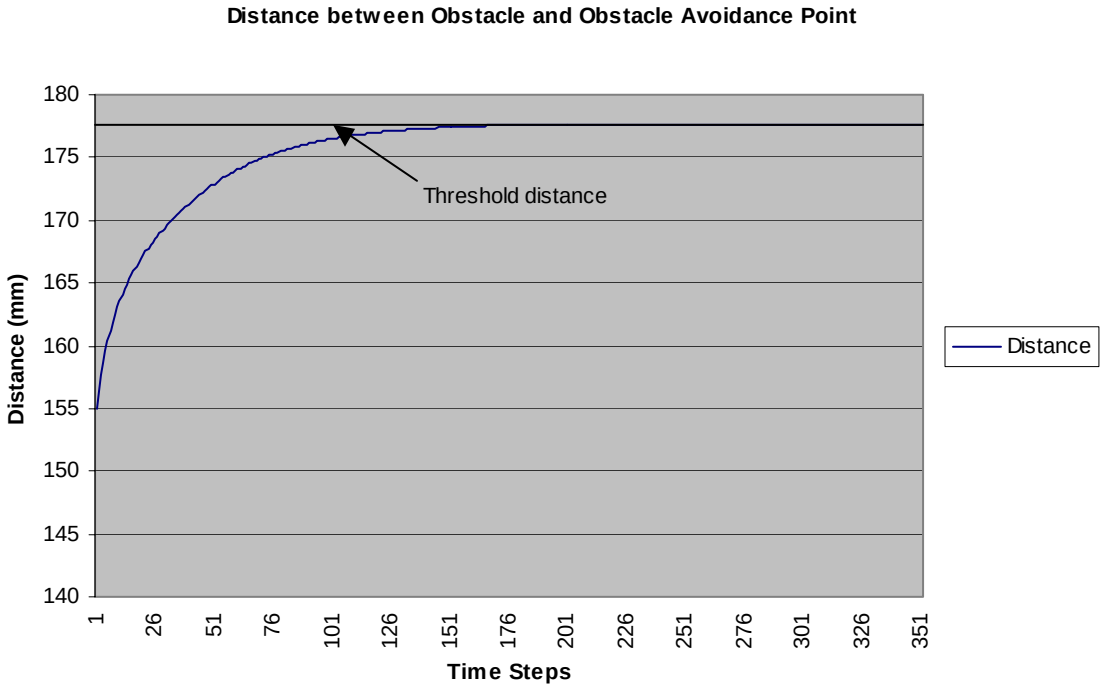

Fig 6.10 Variation of distance between obstacle and obstacle avoidance point

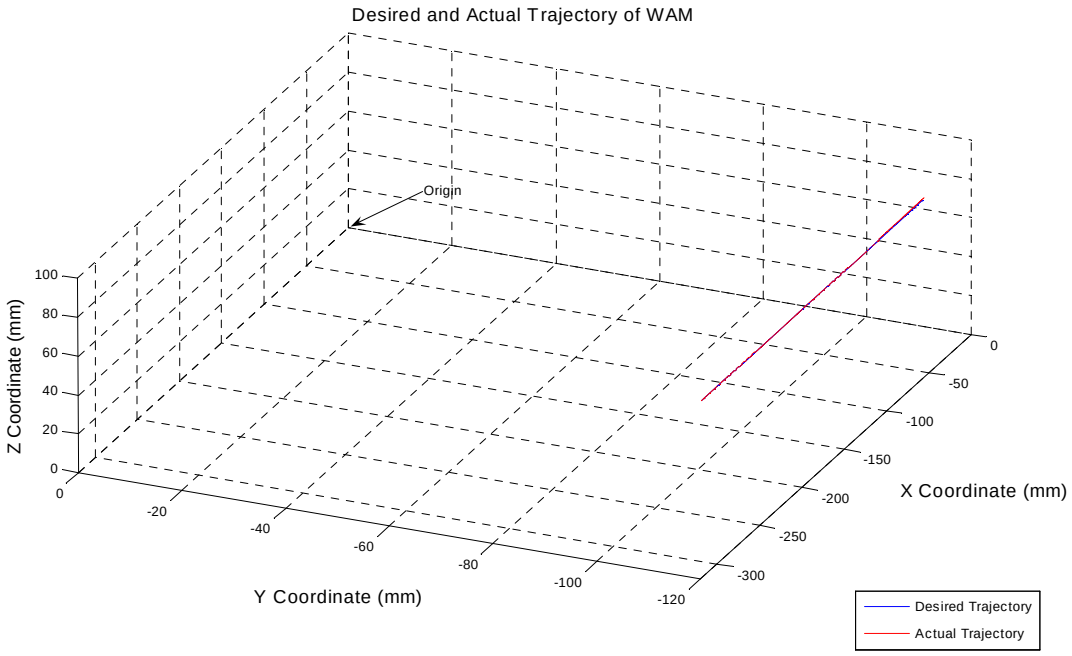

Fig 6.11 Desired and actual trajectory of WAM for obstacle avoidance

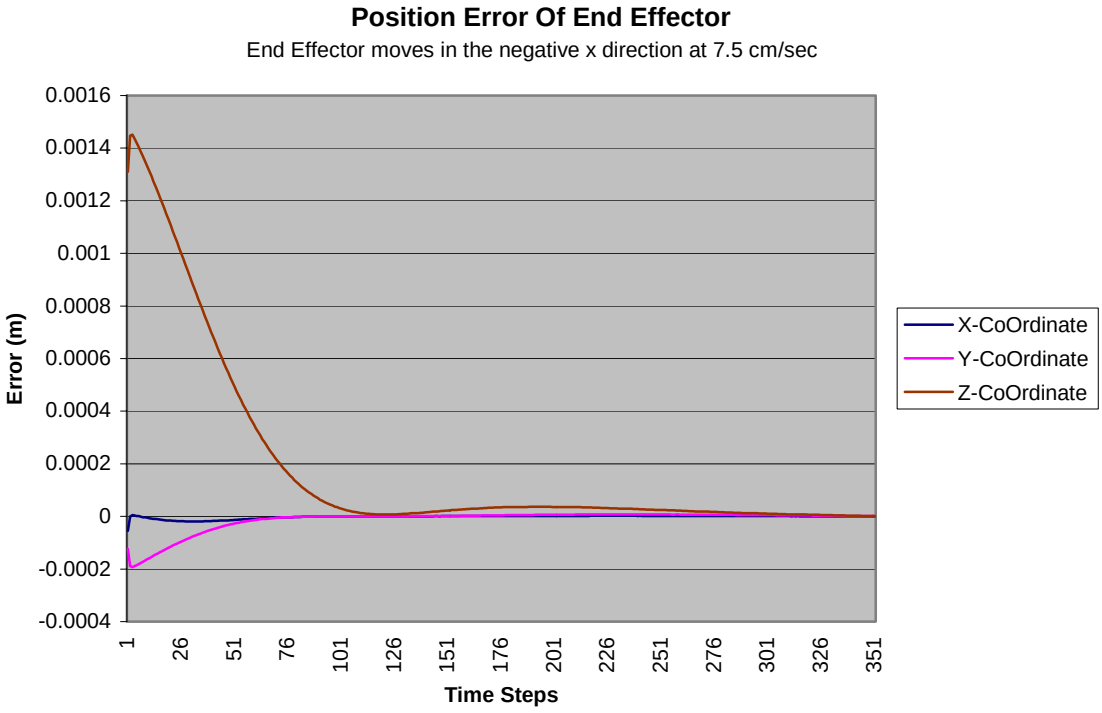

Fig 6.12 End Effector position tracking error for a different trajectory

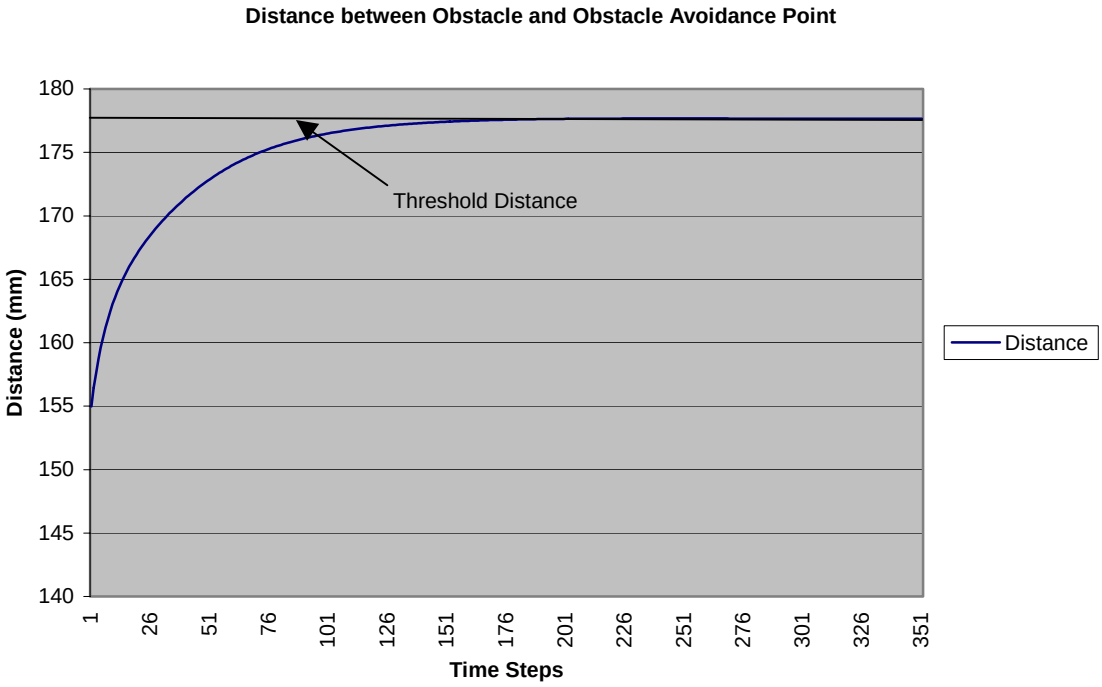

Fig 6.13 Variation of distance between obstacle and obstacle avoidance point

The error graph for this trajectory is shown in meters in order to show a smooth transition. It is again evident for Fig 6.12 that the error varies smooth over time. Fig 6.13 again shows how the distance between the obstacle and the obstacle avoidance point slowly approaches the threshold value. Based on these graphs it is evident that the obstacle avoidance criterion definitely performs better than the joint limit avoidance criteria. Though this constraint performed well with respect to the EE tracking error, it managed to control the manipulator in only one direction. That is, the link was prevented from falling freely only in one direction. If the arm was manipulated away from the operator the problem of less joint friction and free falling shoulder occurred again. In order to avoid this inconvenience, many obstacles needed to be placed in the workspace of WAM such that the shoulder pitch angle always stayed within limits for all kinds of trajectory. Hence the idea of obstacle avoidance was used to draft a new constraint which would keep the shoulder pitch within limits at all times. This is the joint volume limit constraint.

6.1.3 Joint Volume Limitation Constraint

The JVLC is yet another method to perform inverse kinematics on the WAM. It has proved to be effective and most practical for the current scenario. The aim of maintaining the EE of WAM in a three dimensional space in front of the operator, is achieved by limiting the range of movement of joint four to a sphere. That is the locus of points of the joint four geometric center should always lie within a sphere for all trajectories. This in pictorially represented in Fig 6.14. The blue colored disc represents the boundaries of the sphere in the plane parallel to the ground. Since the WAM is fixed and the links are rigid, majority of the movements of joint four lies in that plane. Though, the constraint tries to keep the joint four within a sphere.

Fig 6.14 Pictorial representation of joint volume limitation

The radius of this virtual sphere within which the joint four position stays, is decided again based on the shoulder pitch limitation. Firstly a suitable initial configuration for the arm is selected such that none of the joints reach their physical limits within the operational range of the operator. However, the initial shoulder pitch angle is zero keeping the first link perpendicular to the base. This current position of the joint four with respect to the base co-ordinates marks the center of the sphere. Then by moving the shoulder pitch to various angles, the configuration at which it starts to freely fall under gravity is identified.

For this joint angle the location of the joint four is noted. This would be true for any base rotation. That is, the shoulder pitch angle at which the link starts to fall would be the same for any angle of the base rotation. This criterion reflects on the locus of points for the joint four. Joint four refers to the center of the joint axis. This forms a virtual circular disc in a plane parallel to the base plane of the WAM. The radius of the circular disc (or the sphere) was identified to be *6* inches. The selection of the radius also depends on the relative position of the WAM with respect to the operator's position. At each time step in the trajectory, the position of the joint four is calculated and its distance from the center is calculated and the constraint activated appropriately.

For this criterion, the EE error for the Titan is also analyzed in order to observe its reaction to the movements of the WAM. Based on the error for both the manipulators the algorithm was improved to give a better performance. The following graphs show the performance of this approach for different trajectories. Fig 6.16 shows the variation of the actual trajectory from the desired, when the EE moved in the positive Z direction at *5cm/sec.* It can be seen that initially there is an offset from the desired trajectory but the EE later catches up with it as time passes. It is evidently shown in Fig 6.15. The actual trajectory initially moves away from the desired trajectory but later coincides with it once the constraint is violated.

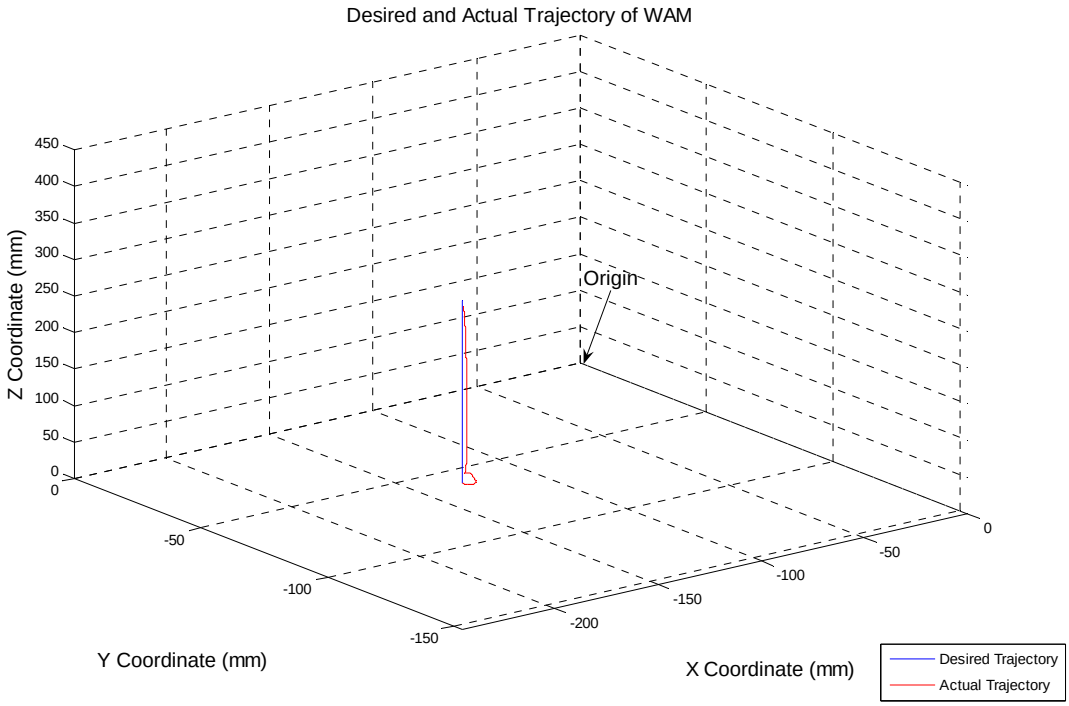

Fig 6.15. Desired and actual trajectory of WAM for JVLC

Fig 6.16 depicts the EE position tracking error for the WAM It can be observed from the graph that the error is little high for this constraint than for the obstacle avoidance criteria, though it performed in the same way as it did for the latter. The distance of the joint four from its center is constantly monitored. As long as it is within the spherical domain the normal Jacobian transpose method performs the inverse kinematics. But once the position tries to move out of the sphere the extended Jacobian method takes over. Thereafter the joint four position is kept within the sphere or it moves along the boundary of the sphere. Unlike the joint limit avoidance criterion where one joint is always locked to prevent it from reaching its limit, JVLC restricts the movements of joints which would keep the Joint four position inside the sphere. Fig 6.16 clearly shows when the constraint is being violated. However the credibility of this performance can be completely understood from the EE tracking error of the Titan which can be seen in Fig 6.17.

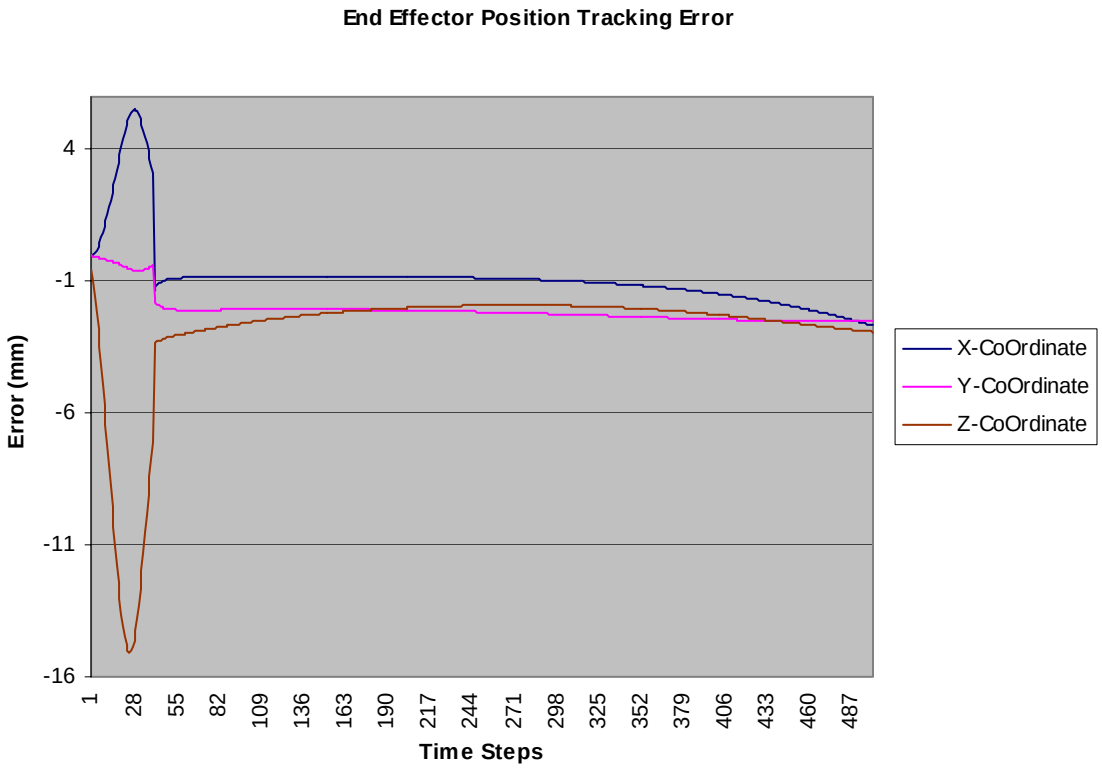

Fig 6.16 End Effector position tracking error for WAM

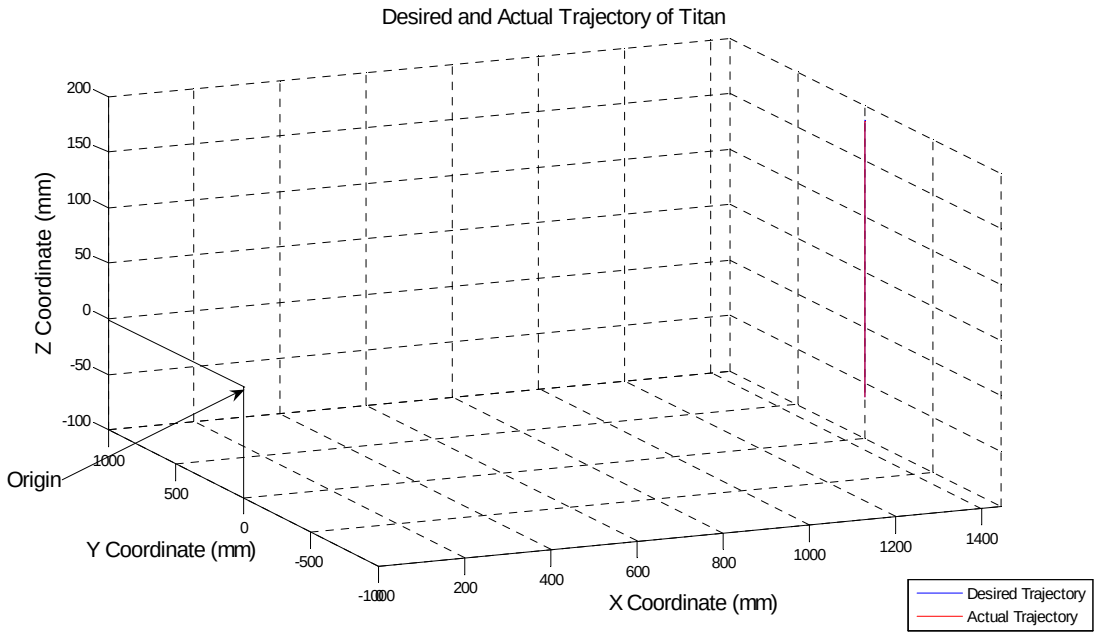

Fig 6.17 Desired and actual trajectory of Titan as commanded by WAM

Fig 6.18 shows that, though the EE tracking error was high for WAM initially, it is much less for the Titan. This is partly because of the inverse kinematics algorithm for the Titan. It can be seen that the error initially increases, but after a certain point, it asymptotically reaches a constant value. The simulations performed better than what could be expected from the error graph. However the important observation made was the closeness between the actual and the desired trajectory. Fig 6.17 shows the comparison between them. The EE tracked the desired trajectory with minimum error which is also evident from Fig 6.18. The maximum error was *0.5 mm*. The performance of this method could also be understood from the distance graph. Fig 6.19 shows the variation of the distance of the joint-four position from its virtual center. When the EE was commanded to move in the positive X direction at *5cm/sec*, the joint-four started from the center and moved towards the boundary of the sphere. Once it hit the boundary the extended Jacobian method maintained the joint at the boundary. This prevented the shoulder to fall under the influence of gravity. The following graphs show the performance of this method for a different trajectory.

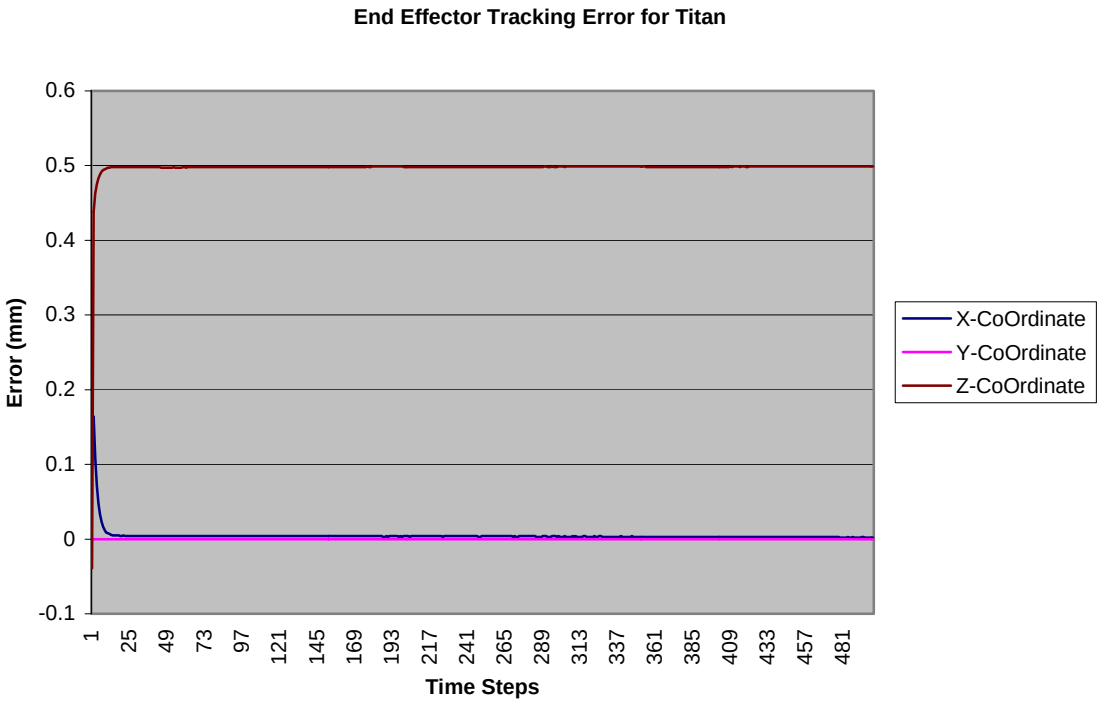

Fig 6.18 End Effector tracking error for Titan

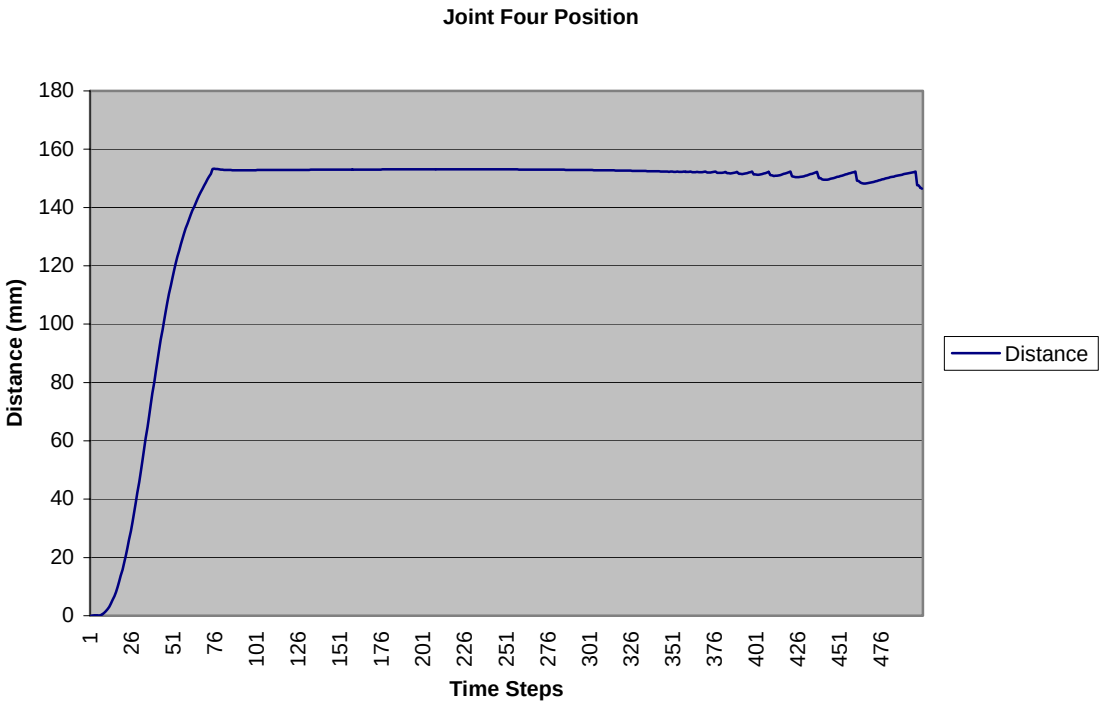

Fig 6.19 Distance of joint-four from its center

Fig 6.20, 6.21, 6.22 and 6.23 again show the desirable performance of the JVLC for a different trajectory when the EE moved in the positive *X* direction at *5 cm/sec*. The EE tracking error increases initially as observed in Fig 6.16 but later reaches a constant value asymptotically. The tracking error for the Titan is also very less. Due to it Fig 6.23 shows only the desired trajectory as it superimposed on the other. If the manipulator can be operated in a convenient region, then this method provides the most practical constraint for resolving the redundancy and simultaneously providing a convenient region of operation for the human operator. It can thus be concluded from all the above graphs that the JVLC is chosen over the other two to resolve redundancy and thus perform the Inverse Kinematics for the WAM. Certain oscillations are absent for this trajectory but appeared for certain other trajectories when one of the joints was approaching its physical limit. This would have caused a undesirable oscillation to appear in the slave joint velocities too.

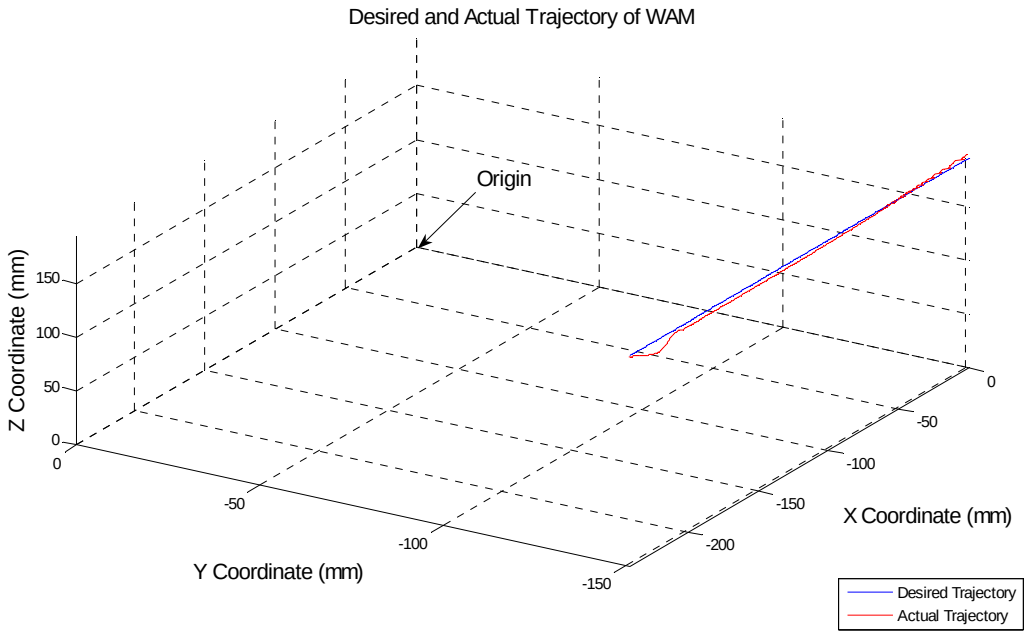

Fig 6.20 Desired and actual trajectory of WAM for a different trajectory

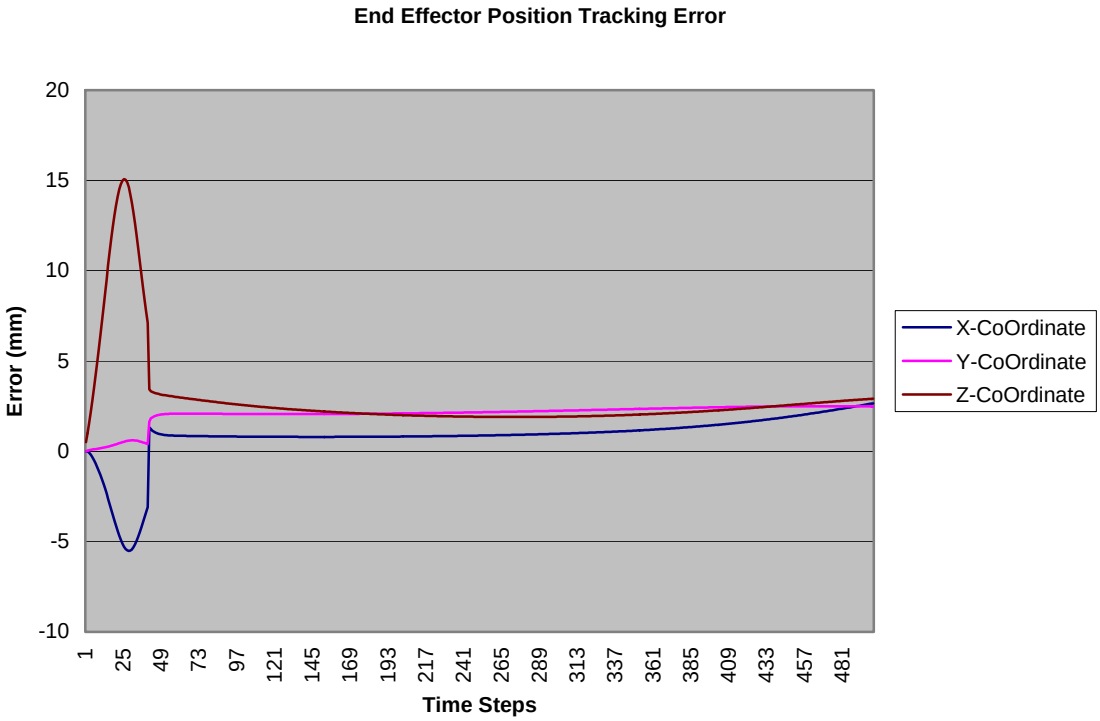

Fig 6.21 End Effector position tracking error for WAM

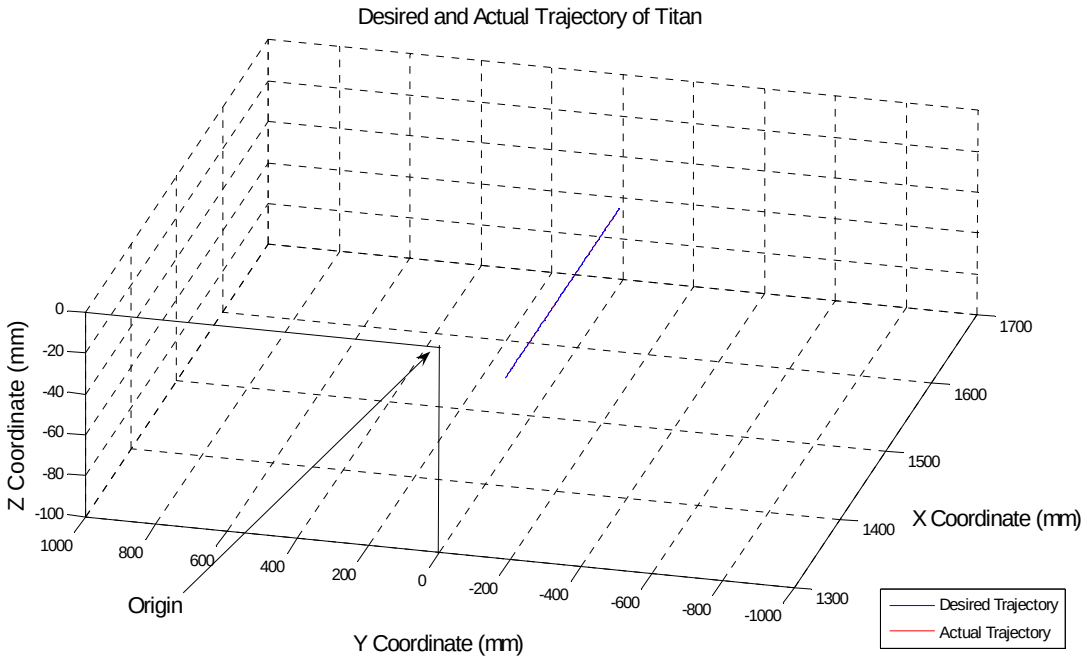

Fig 6.22 Desired and actual trajectory of Titan

Fig 6.23 Distance of joint-four from its center

6.2 Inverse Kinematics for the Titan

The inverse kinematics for Titan is much simpler than it is for the WAM since the Jacobian for Titan is a square matrix. Though the pseudo inverse performs very well in terms of EE position tracking, it is non-invertible at singular points and joint limits. Therefore a pseudo inverse method cannot be used, as it is not reliable for all the points. It is very difficult to predict when singularity would occur in a certain trajectory. Therefore a different approach has to be followed to perform the inverse kinematics. A DLS method derived from the pseudo inverse method is used. The DLS method uses a damping constant to avoid singularity of the Jacobian matrix at points where the pseudo inverse fails. The effect of the damping constant however affects the position tracking accuracy and the convergence rate. This method was well explained in chapter 4. A damping constant of *3*.5 was used for all the trajectories. Though the DLS involves inverse calculation, it is only after making the Jacobian non-singular. So this method is expected to perform better than the pseudo inverse. However any conclusion or comparisons are made after analyzing the error graphs for this method.

Figs 6.24 and 6.25 show the EE tracking error for Titan for two different trajectories. The first trajectory was with the EE moving in the positive Z direction at *4cm/sec* and the second with the EE moving in the X-Z plane with *4cm/sec* in the positive Z and *4cm/sec* in the negative X direction. It is very clear from the above two error graphs that the error reaches its peak early in the trajectory and then moved towards the null error line slowly. Though the error did not reach zero within the trajectory, the direction in which it approaches would definitely take it to zero at some point. Moreover the position transition is also smooth without any oscillations. This was also evident from the graphs explained in the previous section.

Fig 6.24 End Effector position tracking error for Titan

Fig 6.25 End Effector position tracking error for a different trajectory

6.3 Conclusions

From the performance charts it can be concluded that DLS method and JVLC were adopted to perform the inverse kinematics for Titan and WAM respectively. Though the master was operated in autonomous mode for the purpose of simulation, it can be inferred from the graphs that the JVLC performed satisfactorily in terms of EE position tracking. However one should understand that there is no position error involved in the master side, as it is teleoperated. The error graphs were basically to identify the most practical and still the best performing constraint for redundancy. Both these methods together provided a reliable and continuous teleoperation in the operational space of the master manipulator. Though implementation of these algorithms into hardware might involve certain other control issues, it can definitely be inferred from the performance graphs that it provides a comfortable teleoperation to the human operator. The simulation of these trajectories in RoboWorks also proved to be satisfactory. The idea of Cartesian space position mapping was a perfect substitute to joint level control in case of kinematically dissimilar master and slave manipulators.

6.4 Future Work

Only one of the primary objectives of a teleoperation was addressed in this thesis work. There remain many other functions like force feedback and Haptic interface which would ensure a safe and flexible teleoperation. The current Titan manipulator uses a three-fingered Barrett hand instead of its traditional Parallel jaw gripper. Control of the Barrett hand is another major work, which needs analysis of grasping techniques. Issues regarding the haptic control needs to be solved in order to enable sensor-based actuation of the three-fingered hand. In addition to this, the versatility of the WAM could be used to provide force feedback to the human operator to ensure his/her virtual presence in the environment. As already mentioned the force torque sensor mounted on the Titan would

aid to this. In that case, the dynamics of the system has to be analyzed in addition to kinetics and kinematics. The current telerobotic system operates with dual arm acting simultaneously. This thesis work addresses only half the problem involved with coordinated teleoperation. A high level control theory needs to be developed to enable smooth coordinated teleoperation.

Bibliography

Bibliography

[Bai86] John Baillieul, *"Avoiding Obstacle and Resolving Kinematic redundancy"* Proceedings of IEEE, International Conference on Robotics and Automation, v. 3, pp. 1698-1704, Apr 1986.

[Bus04] Samuel. R. Buss, "*Introduction to Inverse Kinematics with Jacobian transpose, Pseudo Inverse and Damped Least Squares Methods*", 2004

[Chi97] Stefano Chiaverini, "*Singularity-Robust Task-Priority redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators*". IEEE transactions on Robotics and Automation, v. 13, no. 3, June 1997.

[Chu97] Chi Youn Chung, Beom Hee Lee and Jung Hoon Lee, "*Obstacle Avoidance for Kinematically Redundant Robots using Distance Algorithm*". Proceedings of the IEEE/RSJ, International Conference on Intelligent Robots and Systems, v. 3, Sep 1997.

[Den55] J. Denavit and R. S. Hartenberg, *"A kinematic notation for lower-pair mechanisms based on matrices"*, ASME Journal of Applied Mechanics, v. 22, pp. 215-221, June 1955.

[Gal01] Miroslaw Galicki, "*Real-Time Trajectory Generation for redundant Manipulators with Path constraints*". The International Journal of Robotics Research, v. 20, no. 8, Aug 2001.

[Ge00] Kang Teresa Ge, "*Solving Inverse Kinematics Constraint Problems for Highly Articulated Models*", Master's Thesis In Computer Science at University of Waterloo, 2000.

[Ham01] William R. Hamel and Reid L. Kress, "*Elements of Telerobotics Necessary for Waste Cleanup Automation*", Proceedings of the 2001 IEEE International Conference on Robotics & Automation, pp. 393-400, 2001.

[Hwa97] Dal-Yeon Hwang and Blake Hannaford, "*Teleoperation Performance with a Kinematically Redundant Slave Robot*", International Journal of Robotics Research, 1997.

[Kle95] Charles. A. Klein, Caroline Chu-Jenq and Shamim Ahmed, "*A New Formulation of the Extended Jacobian Method and its use in Mapping Algorithmic Singularities For Kinematically Redundant Manipulators*", IEEE Transactions on Robotics and Automation, v. 11, no. 1, Feb 1995.

[Lau02] H.Y.K. Lau and L.C.C. Wai, "*A Jacobian based Redundant Control Strategy for the 7-DOF WAM*". International Conference on Control, Automation, Robotics and Vision, pp. 1060-1065, Dec 2002.

[Mer] Michael Meredith and Steve Maddock, "*Real-Time Inverse Kinematics: The Return of the Jacobiani*",

[Sci88] Lorenzo Sciavicco and Bruno Siciliano, "*A Solution Algorithm to the inverse Kinematic Problem for Redundant manipulators*". IEEE Journal of Robotics and Automation, v. 4, no. 4, pp. 403-410, Aug 1988.

[Sci00] Lorenzo Sciavicco and Bruno Siciliano, "*Modeling and Control of Robot Manipulators*", 2nd Edition, McGraw Hill Publications, 2000.

[Str88] Gilbert Strang, " *Linear Algebra and its Application*". 3rd Edition, Academic Press, New York, 1988.

[Tow99] William. T. Townsend and Jeffry. A. Guertin, "*Teleoperator Slave-WAM design methodology*", Industrial Robot, v. 26, no. 3, pp. 167-177, 1999.

Vita

Hariharan was born on March 15, 1982 in the southern state of India, TamilNadu. He was born and brought up in Chennai, the capital of TamilNadu. He finished his entire schooling in the same city and went on to do his undergraduate degree in the University of Madras. He majored in Mechanical Engineering where he was introduced to all the fundamentals of engineering. He took up his career in Robotics in the final year of his Bachelor's. His bachelor's project was "Inverse Kinematic Analysis and Trajectory Generation for a Six axes Spherical Robot using Neural Networks". He designed and manufactured a prototype of the robot and successfully implemented inverse kinematics and trajectory generation in it. After successful completion of his Bachelor's he joined the University of Tennessee, Knoxville in Fall 2003 to do his Master's degree again in Mechanical Engineering. Due to his persistent interest in the field of robotics he joined the research group at REMSL, UTK under Dr. Hamel. Since his inception, he has been exploring many researches in the field of robotics and finally took up the task of teleoperating a 6-DOF manipulator using a redundant master. His efforts and ideas finally culminated in this research. He did not want to put an end to his research career with the Master's. With motivation and attitude he is ready to take a step further and pursue a Doctoral research in the same field.