A thesis report on

OBSTACLE AND SINGULARITY AVOIDANCE OF REDUNDANT SERIAL MANIPULATORS USING THE CONCEPT OF TASK PRIORITY

Submitted towards the fulfillment of the requirement for the award of degree of

MASTER OF ENGINEERING

IN

CAD/CAM & ROBOTICS

Submitted by

Himanshu Arora

Roll No: 801281009

Under the guidance of

Dr. Ashish Singla

Assistant Professor

MED, Thapar University, Patiala

MECHANICAL ENGINEERING DEPARTMENT

Thapar University, Patiala

July, 2014

CERTIFICATE

I hereby certify that the thesis entitled **"Obstacle and Singularity Avoidance of Redundant Serial Manipulators using the Concept of Task Priority"** is an authentic record of my own work carried out in fulfillment of the requirements for the award of degree of **Master of Engineering in CAD/CAM & Robotics** at **Thapar University, Patiala** under the guidance of **Dr. Ashish Singla** (Assistant Professor, MED)

Date: 08-07-2014
Place: PATIALA

Himanshu Arora
(801281009)

This is to certify that the above statement made by the student is correct and true to the best of my knowledge and belief.

Dr. Ashish Singla
Assistant Professor
MED, Thapar University, Patiala

Date: 08/07/14

Countersigned by:-

Dr. Ajay Batish
Professor and Head
Mechanical Engineering Department
Thapar University, Patiala

Dr. S. K. Mohapatra
Senior Professor and
Dean of Academic Affairs
Thapar University, Patiala

ACKNOWLEDGEMENT

First and foremost, I would like to thank God, The Almighty, for his showers of blessings throughout my thesis to complete my work successfully.

I would like to express my deep and sincere gratitude to my thesis supervisor, **Dr. Ashish Singla,** Assistant Professor, Thapar University, Patiala for his continuous support and invaluable guidance throughout my work. His enthusiasm and sincerity, vision and motivation have deeply inspired me. He has taught me the methodology to carry out my thesis and to present my work as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me.

I would like to thank all the staff members of Mechanical Engineering Department for their support and providing adequate lab facilities. Also, my thanks go to all my friends and classmates, who have supported me to complete my thesis work directly or indirectly.

Lastly, I am very grateful to my parents and family for their love, encouragement, prayers, care and sacrifices, they have offered for educating and preparing me for my future.

HIMANSHU ARORA

ABSTRACT

In this thesis work, the redundancy resolution of serial robot manipulators is performed. Redundant manipulators have the characteristics that there exists infinite solutions to the inverse kinematic problem. Out of those infinite solutions, it is possible to choose certain specific solutions on merit, which can be selected on the basis of some performance criteria or priority of some tasks over the other. These priority subtasks can be position prior to orientation, obstacle avoidance, singularity avoidance, torque minimization etc. Out of these parameters, this thesis is focused on obstacle and singularity avoidance using concept of task priority. This concept of task priority is implemented in relation to the inverse kinematics problem of redundant manipulators. A required task is divided into a number of subtask according to the order of priority. The redundancy resolution is performed to achieve the required working of manipulator under complex environment i.e. avoiding obstacle and singularity while tracing a given trajectory.

The entire procedure is formulated using the pseudoinverse of the Jacobian matrix. A number of numerical simulation are performed for different complex environments consisting of obstacles with a given trajectories to show the efficacy of the redundancy control scheme for obstacle and singularity avoidance. Also to show the validity of the formulation, these cases are discussed further by tuning the parameters. An attempt has been made in this thesis to highlight the snake-like behavior of redundant manipulators, while tracking trajectories in very narrow channels. The snake-like behavior is important in many challenging applications like under-water welding in a narrow tanks, to check the blockage of sewerage pipes, to perform laparoscopy operation inside a human body etc.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\boldsymbol{J}(\theta)$	Jacobian matrix
l_1	Length of first link of the manipulator
l_2	Length of second link of the manipulator
c_1	Cosine of the joint angle
c_{12}	Cosine of the sum of joint angle
s_1	Sine of the joint angle
s_{12}	Sine of the sum of joint angle
W	Measure of manipulability
n	Degree of Freedom
θ_i	Joint variables
m	Degree of task space
$\boldsymbol{r}$	Manipulation vector
$\dot{\boldsymbol{r}}$	Cartesian space velocity
$\dot{\boldsymbol{\theta}}$	Joint space velocity
$\ddot{\boldsymbol{r}}$	Cartesian space acceleration
$\ddot{\boldsymbol{\theta}}$	Joint space acceleration
R^m	Range of task space
R^n	Range of Cartesian space
$R(\boldsymbol{J})$	Range space of Jacobian
$R(\boldsymbol{J})'$	Orthogonal complement of range space
$N(\boldsymbol{J})$	Null space of Jacobian
$N(\boldsymbol{J})'$	Orthogonal complement of null space
$\boldsymbol{J}^{\#}(\boldsymbol{\theta})$	Pseudoinverse
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{y}, \boldsymbol{z}$	Arbitrary vectors
$\boldsymbol{\tau}$	Joint torque vector
$\boldsymbol{M}(\boldsymbol{\theta})$	Inertia matrix
$\boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$	Coriolis and centrifugal force vector

$\boldsymbol{r_1}$	Position of end-effector
$\boldsymbol{r_2}$	Orientation of end effector
$\boldsymbol{e_i}$	Error vector
$\emptyset$	Orientation of the end-effector with *x*-axis

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF REDUNDANT MANIPULATORS

Robotics is concerned with the study of machines that are expected to replace human beings in the execution of a task, as regards both physical activity and decision making. Robot makers tend to imitate the characteristics in nature/human. Redundancy is present in human or in any living creature. Human body redundancy can be observed in different levels such as; sensor/actuator level and mechanism level. Sensor/actuator level redundancy includes having two eyes, ears, some interior organs, etc. The mechanism level redundancy can be observed in multiple fingers, two arms and two kidneys etc. Furthermore, the arm itself is redundant and this is called kinematic redundancy. Kinematic redundancy provides extra capabilities for the human arm, which will be explored for robot manipulators in this thesis. The aim of this introductory chapter is to define the redundancy concept in robotic manipulators, importance of the redundancy resolution and the applications of redundant manipulators.

1.2 REDUNDANT MANIPULATORS

A manipulator is termed kinematically redundant when it possesses more degrees of freedom than it is needed to execute a given task. Since it is usually recognized that a general task space consists of six degrees-of-freedom (DOF), a robot arm with seven or more DOFs is considered as the typical example of inherently redundant manipulator, for example 7-DOF LWA4-Arm by SCHUNK. However, even in robot arms with fewer DOF, redundancy can be seen as implying extra DOF which are not needed for tasks in well-organized environments. For instance, using a six axis machine for welding or for stereotactic surgery is not kinematically required, since five axes will place a uniform cross section tool at the appropriate position with the required orientation to complete the task, no final rotation about the tool axis is required. Another example is a planar robotic arm manipulator with three or more joints that acquires to track end-effector position in a Cartesian plane.

In fact, providing robot manipulators with kinematic redundancy is mainly aimed at increasing

dexterity and using arm motion in the null space for subtask objective, during executing the main-task in the task space. A simple example given in Figs. 1.1 and 1.2.

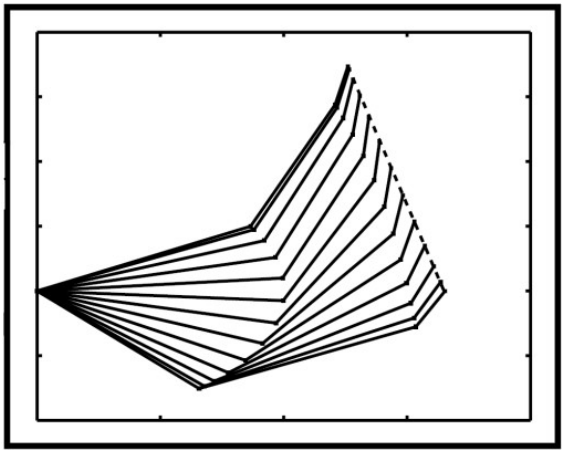

Fig.1.1: Task space motion

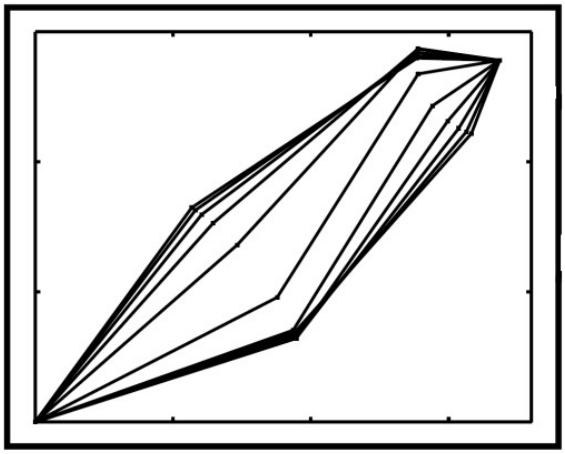

Fig.1.2: Null space motion

In addition to the distinction between non-redundant and redundant robot manipulators, it is also necessary to specify the number of redundancy. Hence the redundant manipulator has more DOF than is required to perform a task in the task space. These extra DOFs allow the robot manipulator to perform more dexterous manipulation and/or provide the robot manipulator with increased flexibility for the execution of sophisticated tasks. For this reason, highly or hyper-redundant manipulators present to improve manipulator performance in complex and unstructured environments. Hyper redundant manipulators are often used in conjunction with trunk or snake robots. The intent is clear, such devices either being planar or fully spatial should have a joint space dimension (n) that is much greater than the dimension of end-effector space (m) *i. e.*, $n >> m$.

1.3 IMPORTANCE OF REDUNDANCY RESOLUTION

Earlier manipulator designs, which are characterized by designing the manipulator with the minimum number of joints required to execute its task resulted in a serious limitation in real-world applications. The limitations are due to the singularity problem, joint limits, and workspace obstacles. These limitations increase the regions to be avoided in the joint and task space during operation, thus requiring a carefully planned task space of the manipulator. This is the case of work cells in traditional industrial applications.

On the other hand, the presence of additional DOFs, besides those strictly required to execute the task, brings the flexibility to avoid the limitations stated above. This flexibility is due to having infinite joint configurations for the same end-effector pose since there are an infinite number of possible solutions for the inverse kinematics problem of redundant robot manipulators. As a result, there exists joint motion which can be propagated in the null space of the Jacobian matrix of a robot manipulator without affecting the end-effector pose. A phenomenon commonly referred to as *self-motion* .

This phenomenon implies that the same task at the end-effector level can be executed in several ways at the joint level, giving the possibility of avoiding limitations and ultimately resulting in a more versatile mechanism. Such a feature is a key to allowing operation in unstructured or dynamically varying environments that characterize advanced industrial applications and service robotics scenarios.

Thus, redundancy can be conveniently exploited to achieve more dexterous robot motions. Formally, a functional constraint task is imposed to be satisfied along with the end-effector task. Such typical constraints include obstacle avoidance, limited joint range, manipulability measures and singularity avoidance. In practice, if properly managed, the increased dexterity of kinematically redundant manipulators may allow the robot manipulator to avoid singularities, joint limits, and workspace obstacles, but also to minimize torque/energy over a given task, ultimately meaning that the robotic manipulator can achieve a higher degree of autonomy.

1.3.1 Obstacle Avoidance using Task Priority

The inverse kinematic problem is of particular interest in case of redundant manipulators, since it admits infinite solutions. The inverse kinematic algorithms have been suitably extended to redundant manipulators by adopting a task space augmentation technique. In order to avoid the problem of obstacle constraint task specification, a task priority strategy has been applied, according to which the task with lower priority is executed only if it does not conflict with the higher priority task. This is accomplished by projecting the constraint Jacobian onto the null space of the end-effector Jacobian; the order of priority being invertible. The technique has been applied to redundancy resolution for space manipulators mounted on a free-floating spacecraft.

1.3.2 Singularity Avoidance

Singularity regions correspond to robot configurations that are close to a singular configuration, and in which the joint velocities required to achieve the end-effector motion in certain directions are extremely high. Therefore, it becomes important for a general purpose manipulator to have more than six degrees of freedom. When the robot is in a singular configuration, there is at least one direction in which the end-effector velocity is unable to move. The joint velocities required to attain the end-effector velocity component in this direction are infinitely high. The mobility of the manipulator is very poor in this direction. Thus, arbitrary directional changes of the end-effector becomes more difficult. Mathematically, this occurs when the rank of the Jacobian matrix is less than the number of its rows. There are two types of singularities:-

- Kinematic singularities are those singular configurations at which the end-effector loses all its mobility.
- Algorithmic singularities are those at which the end-effector task and the constraint task conflict despite the redundant degrees of freedom.

1.3.3 Torque Minimization

Due to infinite solutions of inverse kinematics problem, the solution needs to be chosen so that the torque required to move the motor is minimum. The next solution is chosen in reference to the previous one and thus aims at minimizing the torque of the motor.

1.3.4 Minimum Movement

The solutions are taken in such a way that there is minimum movement of the manipulator to reach the desired position. It also aims in choosing those solutions which require minor axis movements than major axis because the mobility of a robot depends on the direction of the end-effector motion desired which is represented by the manipulator velocity ratio ellipsoid (MVRE). The minor axis of the MVRE represents the minimum value of MVR and the upper limit on the joint velocities required for the end-effector to move in all directions. Thus a small minor axis implies low flexibility and proximity to singular configurations.

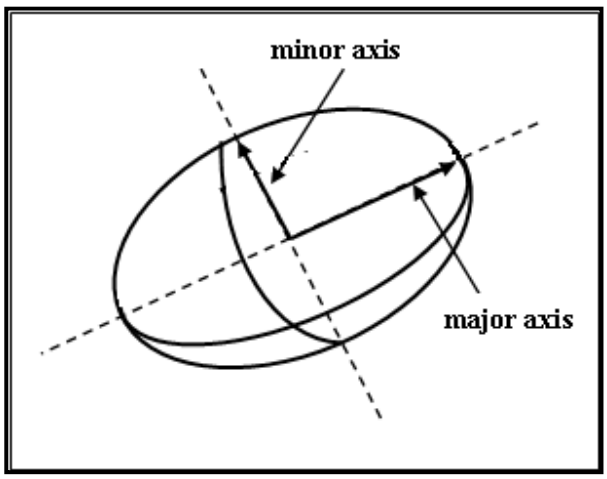

Fig.1.3: Manipulation Velocity Ratio Ellipsoid[9]

1.3.5 Flexibility and Versatility

The redundancy of robot manipulators plays an important role in increasing the flexibility and versatility. Flexibility is important so that the robot can move in any direction freely. It is the ability of the manipulator to change the direction of end-effector motion. Thus, a robot is less flexible when it is near or at a singular configuration.

1.4 APPLICATIONS OF THE REDUNDANT MANIPULATOR

The biological archetype of kinematically redundant manipulator is the human arm, which, not unexpectedly, also inspires the terminology used to characterize the structure of serial-chain manipulators. In fact, the human arm has three DOF at the shoulder, one DOF at the elbow and three DOF at the wrist. The available redundancy can easily be verified, e.g., by laying one's palm on a table and moving the elbow while keeping the shoulder motionless. The kinematic arrangement of the human arm has been replicated in a number of robots often termed as human-arm-like manipulators. This family of 7-DOF manipulators is used for many applications such as; the care-providing robotic system FRIEND presented in Fig. 1.4, which is a semi-autonomous robot designed to support disabled and elderly people in their daily life activities (Schunk 2011). Another example is the usage of Care-o-bot, a next generation of household robots, presented in Fig.1.5.

Fig.1.4: The care-providing robotic system [1]

Fig.1.5: Care-o-bot® Household robot [2]

The new application of an advanced brain-machine interface is shown in Fig. 1.6. This brain-controlled DLR redundant robot arm is used for assisting disabled people. Another application shown in Fig. 1.7 is the MODICAS assistance robot for total hip arthroplasty implantation using a 7-DOF Mitsubishi PA-10 robot arm.

Fig.1.6: Paralyzed woman drinking from a bottle using the DLR Lightweight Robot [3]

Fig.1.7: Mitsubishi PA-10 Robot arm [4]

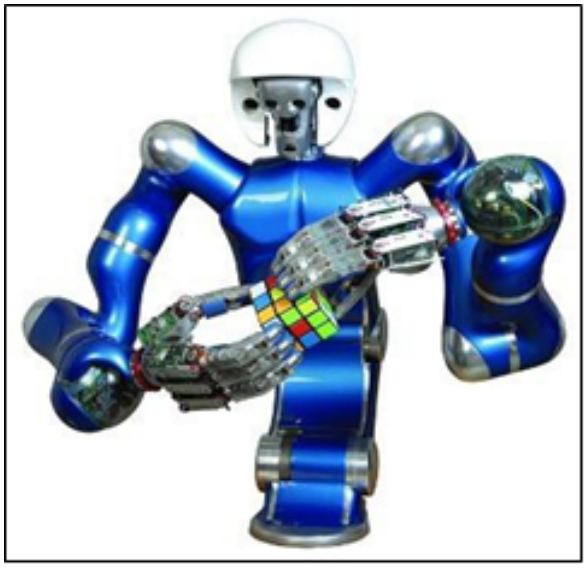

Fig.1.8: Justin the robot [5]

Fig.1.9: The Special-Purpose Dexterous Manipulator (SPDM) [6]

The use of two or more robot arms to execute a task, which is the case in the humanoid robot arms that use cooperating multi-fingered hands, is another important application for the 7-DOF arms. The dual usage of 7-DOF redundant arms can be observed in Fig. 1.8. Robot redundancy also has been recognized of major importance for manipulators in space applications. The Special-Purpose Dexterous Manipulator (SPDM), shown in Fig. 1.9, consists of two 7-DOF arms which can be interpreted as enhancing the importance of kinematic redundancy in critical tasks.

1.5 ORGANISATION OF THE THESIS

This thesis is organized as follows. In Chapter 2, the literature work that has been done so far in the area is discussed. Starting from the time, when the idea came for active utilization of redundancy in which the author used the generalized inverse of the Jacobian matrix to get the general solution of joint velocity to the time when different algorithms are developed to utilize the redundancy and use it for task priority, obstacle avoidance, singularity avoidance, toque minimization etc. Finally a systematic framework to the problem of redundancy resolution is given.

In Chapter 3, the dynamic model for the equations of motion of the robot arm is explained using the Euler-Lagrange model. Then, this approach is analytically applied to the multiple links manipulator to find the symmetric, positive-definite mass inertia matrices and the Coriolis and

centrifugal force vector for each manipulator.

In Chapter 4, the kinematic control related to redundancy resolution and task priority is given. The problem is formulated in the framework of resolved motion rate control. The basic concept of task priority including the importance of Jacobian matrix in redundant space and manipulable space is presented along with the generation of various equations associated with it.

In Chapter 5, numerical simulation are carried to show the efficacy of the formulation. The algorithm for obstacle avoidance and singularity avoidance using task priority are developed using potential function. These algorithms are coded in the MATLAB to plot the different results for obstacle avoidance and singularity avoidance of redundant manipulators.

Finally, in Chapter 6, results are summarized and possible future directions are highlighted.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

A good number of the conventional robots came into reality in the early 60's. Thereafter, the robots got vast fame and find their place in lots of industrial applications. The drift from conventional manipulators to redundant manipulators changed in late 70's, when a lot of researchers worked in the domain of dynamic and kinematic redundancy resolution. Also as explained in the previous chapter, redundant robot manipulators are used in wide range of applications. Redundancy introduced to robot manipulators gives benefit upon non-redundant robots because of the self-motion property. This chapter examines past methods which have been applied in the realization of motion planning of serial manipulators. It especially focuses on Inverse Kinematic problems, methods to resolve kinematic redundancy, obstacle avoidance and singularity avoidance techniques used in past by the researchers.

2.2 OVERVIEW OF THE INVERSE KINEMATICS PROBLEM AND ITS RESOLUTION DIFFICULTIES

As seen, the inverse kinematics method is helpful for the manipulation under complex environment. Therefore the focus is on the inverse kinematic problem and issues raised for its resolution. The basic problem is to resolve a joint configuration to achieve a desired task which is generally defined in Cartesian space (for example, the configurations of shoulder, elbow and wrist must be determined in order that a position in space is precisely reached by the end-effector). This problem usually generates a set of equations that are generally nonlinear, and are thus complicated to work out in broad-spectrum. Moreover, a resolution technique must also deal with the problems described below.

2.2.1 Dealing with Multiple Tasks, and Resolving Conflicts

Specifying a single task at a time is very impractical way for controlling a redundant manipulator. Hence, it is required for a resolution technique to deal with multiple tasks all

together. As a result, it may occur that some of them cannot be fulfilled at the same time, whereas they can separately be resolved in some way using an appropriate strategy.

Such problems come up in many other contexts where conflicting decisions have to be combined. A *Multicriterion optimization techniques* have been developed in the field of mathematical programming that must consider several criterion to deal with complex engineering problems [7]. Several strategies have also been proposed such as *weights* can be assigned to each criterion in order to define their relative importance. Besides this, so-called *hierarchical optimization techniques* set the criterion at different levels [7, 8] each criterion is then satisfied as much as possible, but with the constraint of not disturbing the satisfaction of the more important criteria. An apparent solution is to find a compromise, that however satisfies none of the criteria exactly. This is clearly a more extreme resolution of conflicts, which is preferable in some situations.

Many such similar strategies have been proposed in robotics for the positioning of complex manipulators out of which the *weighting strategy* is the most in use. However, researchers have settled the conflicts by developing *task-priority strategies* to establish a clear priority order among the tasks [9-12]. Some typical examples are given by Badler *et al.* [13, 14] for manipulation and by Phillips *et al.* [15, 16] to achieve smooth solution blending.

2.2.2 Dealing with Under-Constrained and Over-Constrained Problems

Now let's discuss the question of the number of solutions. Three situations can occur:

• **The problem has no exact solution -** It occurs when a task cannot be satisfied (i.e. when a goal is inaccessible), or when two or more task are in conflict and cannot be satisfied at the same time. This is known as an *over-constrained* problem.

• **The problem possesses a single solution** - This condition raises no conflicts, but rarely occurs in general.

• **The problem possesses two or more solutions (or even an infinity of solutions)** - This situation comes up when there are more DOF in the manipulator than tasks then, the problem is considered as *under-constrained* or *redundant*. A manipulator is not redundant by itself, but only with respect to a task that requires less DOF than those existing in the manipulator. Redundancy

is an unavoidable matter of fact when one is concerned with the serial robots, and it has to be dealt with by a proper solution selection technique. Many times, robots may be intentionally built with more DOFs than what is necessary for completing the task, the extra flexibility or DOF can be used to dynamically avoid external obstacles collision [10].
In the next section, various typical methods are presented which are required to deal with the extra DOF and manipulator flexibility.

2.2.3 Dealing with Redundancy

In robotics, typically the joint limits avoidance [17] and singularity avoidance [18] can be the criteria to exploit the redundancy.

A straightforward and practical criterion is the *minimization of the distance* with respect to a reference posture, for example to attract to the mid-range posture or a natural rest posture [19]. More complicated criteria are based on mechanical or biomechanical considerations such as *minimization of joint torques* due to the weight has been proposed by Boulic *et al.* [20]. A DOF can also be associated to each joint of manipulator, and the total DOF is a criterion that can be maximized [21, 22].

Lepoutre [23] compared the manipulator postures resulting from the minimization of different criteria such as joint torques or nearness from joint limits, under constraints imposed by a visibility and reach ability task. While appealing, this redundancy resolution method hides the complexity of determining the “right” function for obtaining realistic postures.

2.2.4 Dealing with Joint Limits

Another essential issue is the value of joint limits, since their violation may extensively affect the posture of manipulator. In robotics, joint limits are usually avoided [18], which is in contrast with animal or human figures whose rest postures are usually close to some joint limits (for example, consider the human knee during standing) but it is not advisable for a robot joint to approach its mechanical limits. Hence, a resolution method must incorporate a method to accurately implement these main constraints.

2.3 REVIEW OF REDUNDANCY RESOLUTION METHODS

In this section, an overview of the main resolution methods is discussed as reported by the different researchers in the past years.

2.3.1 Analytic Methods

Robot manipulators are often designed so that analytical solutions to the Inverse Kinematics problem exists, this is important for real-time applications because these solutions are very fast to compute. Analytical (or closed form) solutions can be establish by direct resolution of the nonlinear equations for very simple robotic manipulators with few DOF [24, 25]. Moreover, the resolution methods are robust. This is also beneficial in computer simulation, and several researchers have addressed the case of the anthropomorphic arm and leg. Korein [26] provides an interesting analytic solution for a 7-DOF arm that deals with joint limits, and Tolani *et al.* [27, 28] also discuss similar procedures.

However, analytical solutions do not exist for general redundant manipulators. In that case, iterative, numerical techniques can be used to solve the Inverse Kinematic problem.

2.3.2 The Resolved Motion-Rate Method

In the pioneering work, Whitney [29] introduced the *resolved motion-rate control*, which is the source of more complex resolution schemes. Given an initial configuration, Whitney performs a linearization of the nonlinear equations, characterized by a Jacobian matrix relating differential changes of the joint coordinates to differential changes of task coordinates. The linear system of equation is solved in order to obtain a new configuration closer to the goal. By repeated resolution of this process, the system usually converges to a solution satisfying the constraint if the initial configuration was “sufficiently close” to it. This method is inspired by the well-known iterative *Newton-Raphson method* for the resolution of nonlinear equations [30].

The main research issue has been to extend this method to exploit the redundancy of the problem at the differential level. Whitney [29] uses *a generalized inverse* [31, 32] to minimize the weighted norm of the variation of joint coordinates. Deo *et al.* [33] have proposed to use the *infinity-norm* to obtain the lowest possible magnitudes of joint variations. Liégeois [18] proposes

an extension for the optimization of a criterion expressed in joint space, by exploiting the null space of the Jacobian matrix. Klein and Huang [17] provide more insight on this topic and on the meaning of the pseudoinverse solution in terms of the *Singular Value Decomposition* representation [34] of the Jacobian matrix. Cleary *et al.* [35] and McGhee *et al.* [36] discuss the integration of multiple weighted optimization criteria in the scheme. Hanafusa, Nakamura and Yoshikawa [9, 11] further broaden the redundancy exploitation with criteria expressed in Cartesian space, hence a secondary Cartesian task that can be satisfied without affecting the primary task. This simultaneous resolution of two tasks with different priorities is known as the *task priority strategy*. Maciejewski and Klein [10] improved the resolution scheme of Hanafusa *et al.* by exploiting pseudoinverse properties. The scheme has then been generalized to an arbitrary number of priority levels by Siciliano *et al.* [12] and Garziera [37].

Another significant issue is the management of singularities of the Jacobian matrix. Merely using a pseudoinverse near singular configurations results in high joint increments that cause a lot of trouble in the resolution process. Traditionally, regularization techniques are used for solving such ill-posed problems [38]. A generalization of the pseudoinverse, known as the *damped-least squares (or singularity-robust) inverse* was independently proposed by Wampler [39] and Nakamura *et al.* [9, 11]. Afterwards, Maciejewski and Klein [40] proposed more complicated methods. Chiaverini [41, 42] introduces a new formulation that overcomes the effects of so-called *algorithmic singularities* that appears in the task-priority strategy, when two tasks are in conflict.

2.3.3 The Jacobian Transpose Method

As discussed by Welman [43], the Jacobian transpose method, only differs from the resolved motion rate method in that the transpose of the Jacobian matrix is used instead of its inverse. The method was introduced by Wolovich and Elliot [44], and extended by Sciavicco and Siciliano to redundant manipulators [45], and for the satisfaction of secondary criteria by Das *et al.* [46]. An iteration can be performed very quickly with this method, since no matrix inversion is required. However, the number of steps required to reach the goal may be much higher than with pseudoinverse-based techniques because of its poor convergence properties, especially near a solution. Hence, on the whole, the Jacobian transpose method is not necessarily more efficient

than pseudoinverse-based methods.

2.3.4 Optimization-Based Methods

Another significant elementary field of numerical mathematics is optimization. Many problems for which a large number of resolution methods have been developed can be stated as optimization problems [47]. For real-time applications, local optimization methods are preferred to the much more expensive global optimization methods [48] even if there is a risk of being stuck in a local optimum of the objective function. The Inverse Kinematic problem is no exception, it can be formulated as a constrained optimization problem and thus solved with nonlinear programming methods.

In computer simulation, Zhao and Badler [49] [14] used an optimization method [50] for the manipulation of an articulated manipulator, a nonlinear function (describing the degree of satisfaction of all constraints) is minimized under a set of linear equality and inequality constraints describing joint limits. An application of this method for the manipulation of articulated figures within the Jack system is described by Phillips *et al.* [15].

More recent optimization methods such as *genetic algorithms* have also been applied to the inverse kinematics problem [51]. It is not clear if such algorithms are valid alternative to traditional numerical resolution methods based on the gradient, especially for manipulator with a high number of DOF. However, they are able to overcome the problem of local minima due to their non-deterministic nature.

2.3.5 A Modal Approach for Hyper-Redundant Structures

Chirikjian and Burdick [52] introduce a modal approach to solve the inverse kinematics problem of hyper-redundant snake-like robots modeled as a continuous curve. Because of the high number of DOF, it is very expensive to solve the Inverse Kinematic problem with pseudo-inverse methods. Instead of this, they use an analytical curve described by a set of basis functions to form the general profile of the hyper-redundant chain. The Inverse Kinematic problem is then solved in the space of the parameters of these basis functions. Finally, a “fitting” of the discrete chain to this curve is performed in order to compute the joint parameters. This technique could

be useful to deal with the spine of a human model.

2.3.6 Other Techniques

For completeness, some other methods are also presented in this section that have not gained wide acceptance.

The *Cyclic-Coordinate-Descent method* is an iterative heuristic method that attempts to reduce the position and orientation errors by changing one joint at a time. Similarly to the Jacobian transpose method, the convergence rate towards a solution is very poor even the cost of a single iteration is very low. A comparison of both methods is done by Welman [43].

For the positioning of complex articulated figures Badler *et al.* [13] present an elegant recursive constraint satisfaction algorithm which can deal with multiple constraints simultaneously, and conflicts are resolved with a weighting strategy. To use simple inverse kinematic algorithm, the problem is recursively decomposed into smaller sub-problems until they are solvable. This approach has since been abandoned by Badler.

Next, based on the knowledge of the workspaces for each distal subchain of the whole chain is the *reach hierarchy algorithm* proposed by Korein [26]. This method indirectly deals with joint limits, as they affect the workspace data. The scope of this algorithm is limited, since the computation of these workspaces may be very complex and computationally expensive for complex chains.

Finally, by learning the nonlinear inverse task function artificial neural networks [53] have been applied to the inverse kinematics problem.

2.4 COLLISION DETECTION AND COLLISION AVOIDANCE

In a virtual environment, a manipulator must deal with the inter-penetration of objects. The first problem is the *collision detection* problem (or self-collision detection, when an object collides with itself), while the second is the *collision response* problem. Many solutions have been developed both in robotics [54] and computer graphics [55], for rigid objects as well as

deformable ones [56]. The collision response can be either kinematic (with a direct change of the position of the bodies) or dynamic (with the application of repulsive forces).

An additional complex problem is the *collision avoidance* problem. It is often wanted that a controlled body be able to execute a task without colliding with surrounding obstacles or with itself. The obstacle avoidance problem is well-known in robotics. The first solutions were based on an initial trajectory planning of a collision-free path for the robot [57]. However, such solutions are expensive and bound the interactivity of the robot with its environment. Hence new approaches have been developed to deal with dynamically varying obstacles in real-time. A classical technique proposed by Khatib [58] is the artificial potential field method, where the manipulator moves in a field of forces, that attract the end-effector to its goal, and repulse the manipulator parts from the surface of the obstacles. Similarly some kinematic methods have been proposed by Espiau and Boulic [59] and by Maciejewski and Klein [10]. A secondary task controls the point on the robot which is closest to the obstacle, in order to maximize its distance to the surface of the obstacle. Specific methods have also been developed for hyper-redundant manipulators [60].

The self-collision avoidance is especially interesting when applied to articulated configurations because this provides some extra DOF. For example, Zhao and Badler [61] used a potential fields based method for the Jack structure, together with inverse kinematics to perform collision response. In computer graphics, the robotics methods have been adopted and adapted to the human case. However, not all possible self-collisions are checked as this would be relatively expensive, only the most repeatedly occurring cases are tested, such as collisions of hands with the body and hands with hands. Koga *et al.* [62] proposed a solution to the multi-arm manipulation planning problem. In order to manipulate a movable object between two configurations, the path planner automatically computes the collision-free trajectories for several cooperating arms. Huang [63] uses a secondary task for controlling the arm in the process of reaching an object, to keep the elbow outside of the torso. While inter-penetration is avoided, the resulting motion is not necessarily sensible.

A recent challenge to deal with the self-collision avoidance difficulty for articulated manipulator is due to Nebel [64]. The objective of the system is to repeatedly generate collisions-free

trajectory sequences based on key frame interpolation. The principle of his scheme is to detect self-collisions in the sequence generated with classical methods, and then to modify these solution frames in order to remove the collisions. Finally, these new solution frames are used for a new classical interpolation. This is a potentially useful tool for simulators, but is not suited for real time applications that require on-the-fly generation of motion.

Redundant manipulators are usually addressed to an issue of obstacle avoidance. If a link structure can reach the same end-effector position by several different intermediate positions of the kinematic bodies from the chain, it is called redundant. Redundancy can be used to meet secondary tasks such as obstacle avoidance, singularity avoidance [65] or variable optimization (i.e. torque optimization [66], dexterity optimization [67], energy saving optimization and others).

One of the most used approaches to solve inverse kinematics for obstacle avoidance is based on the use of inverse of Jacobian and its relatives [68-71]. The main idea that stands behind this approach is that a Jacobian matrix ($\boldsymbol{J}$) contains all the information necessary to relate a change in any component of end-effector position vector ($\boldsymbol{r}$) to a change in any other component of the configuration angle vector ($\boldsymbol{\theta}$).

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}^{-1}(\boldsymbol{\theta})\dot{\boldsymbol{r}} \tag{2.1}$$

After imposing Eqn. (2.1), current algorithms pick a step and iterate the equation on a loop until the end-effector reaches the goal or a position which is fairly satisfying near the goal. Variations on this technique rely on constructing $\boldsymbol{J}^{-1}$, most used are the pseudo-inverse $(\boldsymbol{J}^{\#})^{-1}$ and the transposed $(\boldsymbol{J}^{T})^{-1}$.

Other studies proposed a solution that requires only standard inversion instead of pseudoinverse, by using the extended Jacobian method [72]. After the minimum norm and the homogenous solution are defined, a secondary goal can be translated mathematically [73]. However, this approach looses substance when dealing with local minima and irregularly shaped obstacles [74, 75]. To rectify this, some studies propose using task priority strategies, but these are less suitable for manipulators with a high number of DOFs [65].

Other methods related to the objective of this study, also used in robot navigation, are the

artificial potential field algorithm [73, 74] and fuzzy logic [74, 75]. Obstacles are viewed as reflecting magnetic poles while targets are viewed as attractive points. Artificial potential field algorithm is also suffering from local minima inconsistency, and because of the issues imposed by arbitrarily shaped obstacles and manipulators with multiple links, only a few studies address this approach [75-77]. One of the best approaches to the problem of manipulation in environments with obstacles is the one that uses the quadratic problem to minimize the redundancy resolution of the kinematic chain, with subject to bound constrains [72, 73, 77-80].

2.5 OBSERVATIONS FROM LITERATURE SURVEY

From the above literature survey following observations can be drawn.

1. The general solution of the joint velocity can be expressed by means of the generalized inverse of the Jacobian matrix.
2. The measure of manipulability can be used to determine the best postures of the various types of manipulator.
3. Redundancy can be utilized for keeping the joint angles within their physical limitations.
4. Redundant manipulators are capable of avoiding obstacles, avoiding singularity etc. by making use of pseudoinverse of the Jacobian matrix and finding inverse kinematic solution.
5. By dividing the task into subtasks with the order of priority, the degeneracy of degree of freedom can be overcome easily.

By taking these observations into account, the concept of task priority is found to be a very noble and versatile concept, with which a number of subtasks can be performed simultaneously such as position prior orientation, singularity avoidance, avoiding the joint limits, obstacle avoidance and minimum torque movement etc. That is why, the concept of task priority is used in this thesis, and demonstrated well with different case studies on trajectory tracking while performing obstacle avoidance and singularity avoidance.

CHAPTER 3
DYNAMIC MODELING OF REDUNDANT MANIPULATORS

3.1 INTRODUCTION

Such equations of motion are useful for computer simulation of the robot arm motion, the design of suitable control equations for a robot arm, and the evaluation of the kinematic design and structure of a robot arm.

In this chapter, the formulation, characteristics and properties of the dynamic equations of motion are considered for control purposes. The purpose of manipulator control is to maintain the dynamic response of a computer-based manipulator in accordance with some prespecified system performance and desired goals. This chapter mainly deals with the modeling and the behavior of computer-controlled robots.

The actual dynamic model of a robot arm can be obtained from known physical laws such as the laws of Newtonian mechanics and Lagrangian mechanics. This leads to the development of the dynamic equations of motion for the various articulated joints of the manipulator in the terms of specified geometric and inertial parameters of the links. The dynamic model of any system is mainly developed using two approaches – Newton-Euler approach and Euler-Lagrange approach. The first one deals with the force and moment balance, while considering each term a vector quantity like displacement, velocity, acceleration, force etc. On the other hand, the second method is based on the energy approach and treat each quantity in the formulation as a scalar quantity like kinetic energy, potential energy etc.

The Euler-Lagrange approach is more general and amenable for multi-body dynamics. That is why, this approach is being used to develop the dynamic model of redundant manipulator in this thesis.

3.2 EULER–LAGRANGE (E–L) FORMULATION

The derivation of the dynamic model of a manipulator based on the E–L formulation is simple

and systematic. The E−L equations of motion provide explicit state equations for robot dynamics and can be utilized to analyze and design advanced joint-variable space control strategies. The dynamic equations are being used to solve for the *forward dynamics* problem, that is, given the desired torques/forces. Moreover, the dynamic equations are also used for the *inverse dynamics* problem, to solve for the joint accelerations which are then integrated to solve for the generalized coordinates and their velocities. Unfortunately, the computation of the dynamic model requires a fair amount of arithmetic operations. Thus, the E−L equations are very difficult to utilize for real-time control purposes, unless they are simplified as follows.

The derivation of the dynamic equations of an *n*-DOF manipulator is based on the understanding of:

1. The 4 × 4 homogeneous coordinate transformation matrix, $\boldsymbol{A}_i^{i-1}$, which describes the spatial relationship between the i^{th} and the $(i\text{-}1)^{th}$ link coordinate frames. It relates a point fixed in link *i* expressed in homogeneous coordinates with respect to the i^{th} coordinate system to the $(i\text{-}1)^{th}$ coordinate system.
2. The Euler-Lagrange equation

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_i}\right] - \frac{\partial L}{\partial q_i} = \tau_i \qquad (i = 1, 2, \ldots, n) \tag{3.1}$$

where, L = Lagrangian function = kinetic energy (*K)* – potential energy (*P)*

K = total kinetic energy of the robot arm

P = total potential energy of the robot arm

q_i = Generalized coordinates of the robot arm

$\dot{q}_i$ = Generalized coordinate of the joint velocity.

τ_i = Generalized force (or torque) applied to the system at joint *i* to drive link *i.*

From the above Euler-Lagrange equation, one is required to properly choose a set of *generalized*

coordinates to describe the system. For a simple manipulator with rotary-prismatic joints, various sets of generalized coordinates are available to describe the manipulator. However , the angular positions of joints provide a natural correspondence with the generalized coordinates. Thus, in the case of a rotary joint, $q_i \equiv \theta_i$, the joint angle span of the joint; whereas for a prismatic joint, $q_i \equiv d_i$, the distance travelled by the joint.

The following derivation of the equations of motion of an *n* degrees of freedom manipulator is based on the homogeneous coordinate transformation matrices. The dynamics of a serial manipulator can be found in number of textbooks [25, 81], however it is reproduced here for the sake of completeness

3.2.1 Joint Velocities of a Robot Manipulator

The E−L formulation requires knowledge of the kinetic energy of the physical system, which in turn requires knowledge of the velocity of each joint. In this section, the velocity of a point fixed in link *i* is derived and the effects of the motion of other joints on all the points in this link is explored.

With reference to Fig. 3.1, let $\boldsymbol{r}_i^i$ be a point fixed and at rest in a link *i* and expressed in homogeneous coordinates with respect to the i^{th} link coordinate frame,

$$\boldsymbol{r}_i^i = (x_i, y_i, z_i, 1)^T \tag{3.2}$$

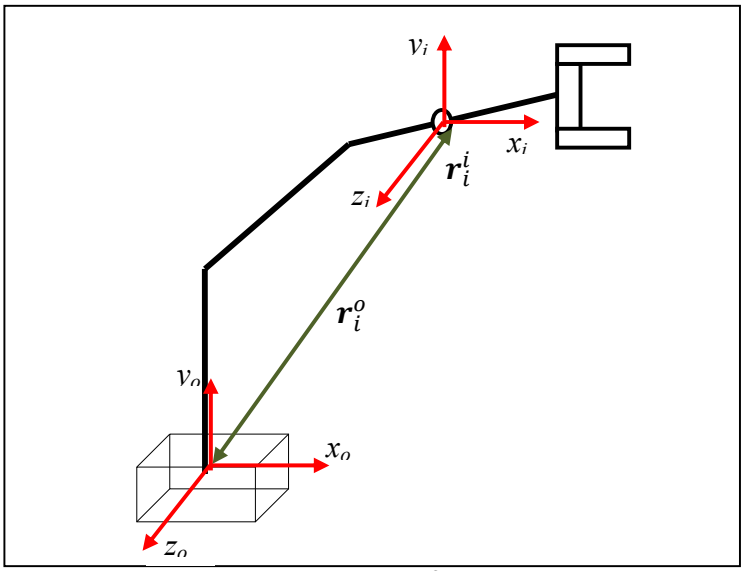

Fig. 3.1: A 3-DOF manipulator showing base coordinate frame and homogeneous coordinate frame

Let $\boldsymbol{r}_i^o$ be the same point $\boldsymbol{r}_i^i$ with respect to the base coordinate frame, $\boldsymbol{A}_i^{i-1}$ the homogeneous coordinate transformation matrix which relates the spatial displacement of the i^{th} link coordinate frame to the $(i\text{-}1)^{th}$ link coordinate frame, and $\boldsymbol{A}_i^0$ the coordinate transformation matrix which relates the i^{th} coordinate frame to the base coordinate frame, then $\boldsymbol{r}_i^o$ is related to the point $\boldsymbol{r}_i^i$ by

$$\boldsymbol{r}_i^0 = \boldsymbol{A}_i^0\, \boldsymbol{r}_i^i \tag{3.3}$$

where,

$$\boldsymbol{A}_i^0 = \boldsymbol{A}_1^0\, \boldsymbol{A}_2^1 \cdots \boldsymbol{A}_i^{i-1} \tag{3.4}$$

If joint *i* is revolute, then the general form of $\boldsymbol{A}_i^{i-1}$ is given by [81]

$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Or, if joint *i* is prismatic, then the general form of $\boldsymbol{A}_i^{i-1}$ is given by

$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & 0 \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

where, $\alpha_i, a_i, and\ d_i$ are known parameters from the kinematic structure of the arm and $\theta_i\ or\ d_i$ is the joint variable of joint *i* taken as, q_i to represent the generalized coordinate.

Since the point $\boldsymbol{r}_i^i$ is at rest in link *i*, and assuming rigid body motion, other points as well as the point $\boldsymbol{r}_i^i$ fixed in the link *i,* will have zero velocity expressed with respect to the i^{th} coordinate frame. The velocity of $\boldsymbol{r}_i^i$ expressed in the base coordinate frame can be expressed as

$$\boldsymbol{v}_i^0 \equiv \boldsymbol{v}_i = \frac{d}{dt}\left(\boldsymbol{r}_i^o\right) = \frac{d}{dt}\left(\boldsymbol{A}_i^0\, \boldsymbol{r}_i^i\right) \tag{3.7}$$

$$= \dot{\boldsymbol{A}}_1^0\, A_2^1 \cdots \boldsymbol{A}_i^{i-1}\, r_i^i + \boldsymbol{A}_1^0\, \dot{\boldsymbol{A}}_2^1 \cdots \boldsymbol{A}_i^{i-1} r_i^i + \ldots \tag{3.8}$$

$$= \left(\sum_{j=1}^{i} \frac{\partial \boldsymbol{A}_i^0}{\partial q_j} \, \dot{q}_j \right) \boldsymbol{r_i^i} \tag{3.9}$$

The above compact form is obtained because $\dot{\boldsymbol{r}}_i^i = 0$. The partial derivative of $\boldsymbol{A}_i^0$ with respect to q_j can be calculated with the help of a matrix $\boldsymbol{Q}_i$ which, for a revolute joint, is defined as

$$\boldsymbol{Q}_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.10}$$

and, for a prismatic joint, as

$$\boldsymbol{Q}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.11}$$

It then follows that

$$\frac{\partial \boldsymbol{A}_i^{i-1}}{\partial q_i} = \boldsymbol{Q}_i \, \boldsymbol{A}_i^{i-1} \tag{3.12}$$

For example, for a robot arm with all rotary joints, $q_i = \theta_i$. Therefore,

$$\frac{\partial \boldsymbol{A}_i^{i-1}}{\partial \theta_i} = \begin{bmatrix} -\sin\theta_i & -\cos\alpha_i \cos\theta_i & \sin\alpha_i \cos\theta_i & -a_i \sin\theta_i \\ \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.13}$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

$$\equiv \boldsymbol{Q}_i \, \boldsymbol{A}_i^{i-1} \tag{3.15}$$

Hence, for *i = 1, 2, ..., n.*

$$\frac{\partial \boldsymbol{A}_i^0}{\partial q_j} = \begin{cases} \boldsymbol{A}_1^0 \, \boldsymbol{A}_2^1 \, . \, . \, . \, \boldsymbol{A}_{j-1}^{j-2} \, \boldsymbol{Q}_j \, \boldsymbol{A}_j^{j-1} \, . \, . \, . \, \boldsymbol{A}_i^{i-1} & for\ j \le i \\ 0 & for\ j > i \end{cases} \tag{3.16}$$

The above equation can be interpreted as the effect of the motion of joint *j* on all the points on link *i*. To simplify, let $\boldsymbol{U}_{ij} = \frac{\partial \boldsymbol{A}_i^0}{\partial q_j}$, then the above equation can be written as follows for *i = 1, 2, . . ., n.*

$$\boldsymbol{U}_{ij} = \begin{Bmatrix} \boldsymbol{A}_{j-1}^0 \boldsymbol{Q}_j \ \boldsymbol{A}_i^{j-1} & for\ j \le i \\ 0 & for\ j > i \end{Bmatrix} \tag{3.17}$$

Using above notation, $\boldsymbol{v}_i$ can be expressed as

$$\boldsymbol{v}_i = \left(\sum_{j=1}^{i} \boldsymbol{U}_{ij} \dot{q}_j \right) \boldsymbol{r}_i^i \tag{3.18}$$

Next, the need is to find the interaction effects between joints as

$$\frac{\partial \boldsymbol{U}_{ij}}{\partial q_k} \triangleq \boldsymbol{U}_{ijk} = \begin{Bmatrix} \boldsymbol{A}_{j-1}^0 \, \boldsymbol{Q}_j \, \boldsymbol{A}_{k-1}^{j-1} \, \boldsymbol{Q}_k \, \boldsymbol{A}_i^{k-1} & for\ i \ge k \ge j \\ \boldsymbol{A}_{k-1}^0 \, \boldsymbol{Q}_k \, \boldsymbol{A}_{j-1}^{k-1} \, \boldsymbol{Q}_j \, \boldsymbol{A}_i^{j-1} & for\ \ i \ge j \ge k \\ 0 & for\ i < j, i < k \end{Bmatrix} \tag{3.19}$$

The above equation can be interpreted as the interaction effects of the motion of joint *j* and joint *k* on all the points on link *i*.

3.2.2 Kinetic Energy of a Robot Manipulator

Let K_i be the kinetic energy of link *i*, *i* = 1, 2, . . . , n, as expressed in the base coordinate system, and let dK_i be the kinetic energy of a particle with differential mass *dm* in link *i,* then

$$dK_i = \frac{1}{2}(\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm \tag{3.20}$$

$$= \frac{1}{2} \text{ trace } (\boldsymbol{v}_i \boldsymbol{v}_i{}^T) dm \tag{3.21}$$

Substituting the value of $\boldsymbol{v}_i$ from Eqn. (3.18), then the kinetic energy of the differential mass is

$$dK_i = \frac{1}{2} \text{ trace } \left[\sum_{p=1}^{i} \boldsymbol{U}_{ip} \ \dot{q}_p \, \boldsymbol{r}_i^i \left(\sum_{r=1}^{i} \boldsymbol{U}_{ir} \, \dot{q}_p \, \boldsymbol{r}_i^i \right)^T \right] dm \tag{3.22}$$

$$= \frac{1}{2}\,\text{trace}\left[\sum_{p=1}^{i}\sum_{r=1}^{i} \boldsymbol{U}_{ip}\,\boldsymbol{r}_i^i\,\boldsymbol{r}_i^{iT}\boldsymbol{U}_{ir}{}^T\dot{q}_p\dot{q}_r\right]dm \tag{3.23}$$

$$= \frac{1}{2}\,\text{trace}\left[\sum_{p=1}^{i}\sum_{r=1}^{i} \boldsymbol{U}_{ip}\,(\boldsymbol{r}_i^i\,dm\,\boldsymbol{r}_i^{iT})\boldsymbol{U}_{ir}{}^T\dot{q}_p\dot{q}_r\right] \tag{3.24}$$

The matrix $\boldsymbol{U}_{ij}$ is the rate of change of the points ($\boldsymbol{r}_i^i$) on link *i* relative to the base coordinate frame as q_j changes. Therefore, the kinetic energies of all points can be summed up as

$$K_i = \int dK_i = \frac{1}{2}\,\text{trace}\left[\sum_{p=1}^{i}\sum_{r=1}^{i} \boldsymbol{U}_{ip}\,(\int \boldsymbol{r}_i^{\,i}\,\boldsymbol{r}_i^{iT}\,dm)\boldsymbol{U}_{ir}{}^T\dot{q}_p\dot{q}_r\right] \tag{3.25}$$

The integral term inside the bracket is the inertia of all points on link *i,* hence,

$$\boldsymbol{J}_i = \int \boldsymbol{r}_i^i\,\boldsymbol{r}_i^{iT}\,dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \tag{3.26}$$

where, $\boldsymbol{r}_i^i = (x_i,\ y_i,\ z_i, 1)^T$ as defined before. If the inertia tensor I_{ij}, which is defined as

$$I_{ij} = \int \left[\delta_{ij}\left(\sum_k x_k^2\right) - x_i x_j\right] dm \tag{3.27}$$

where the indices *i, j, k* indicate principal axes of the i^{th} coordinate frame and δ_{ij} is the so-called Kronecker delta, then $\boldsymbol{J}_i$ can be expressed in inertia tensor as

$$\boldsymbol{J}_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & m_i\bar{x}_i \\ I_{xy} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & m_i\bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & m_i\bar{z}_i \\ m_i\bar{x}_i & m_i\bar{y}_i & m_i\bar{z}_i & m_i \end{bmatrix} \tag{3.28}$$

Hence, the total kinetic energy *K* of a robot arm is

$$K = \sum_{i=1}^{n} K_i = \frac{1}{2}\sum_{i=1}^{n} \text{Trace}\left(\sum_{p=1}^{i}\sum_{r=1}^{i} \boldsymbol{U}_{ip}\boldsymbol{J}_i\boldsymbol{U}_{ir}{}^T\,\dot{q}_p\,\dot{q}_r\right) \tag{3.29}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{p=1}^{i}\sum_{r=1}^{i}\left[\text{Trace}\,(\boldsymbol{U}_{ip}\boldsymbol{J}_i\boldsymbol{U}_{ir}{}^T)\,\dot{q}_p\,\dot{q}_r\right] \tag{3.30}$$

3.2.3 Potential Energy of a Robot Manipulator

Let the total potential energy of a robot arm be *P* and let each of its link's potential energy be P_i:

$$P_i = -m_i\boldsymbol{g}^{\boldsymbol{o}}\boldsymbol{r}_i^{\boldsymbol{o}} = -m_i\boldsymbol{g}^{\boldsymbol{o}}(\boldsymbol{A}_i^o\,\boldsymbol{r}_i^i) \tag{3.31}$$

and the total potential energy of the robot arm can be obtained by summing all the potential energies in each link,

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} -m_i\boldsymbol{g}^{\boldsymbol{o}}(\boldsymbol{A}_i^o\,\boldsymbol{r}_i^i) \tag{3.32}$$

where $\boldsymbol{g}^o = (g_x,\ g_y,\ g_z, 0)$ is a gravity row vector expressed in the base coordinate system. For a level system, $\boldsymbol{g}^o = (0, 0, -|g|, 0)$ and g is the gravitational constant ($g = 9.8062\ m/sec^2$).

3.2.4 Motion Equations of a Manipulator

As discussed before the Lagrangian function is given by $L = K - P$. Now using Eqns. (3.30) and (3.32), Lagrangian function is:

$$L = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i}\left[\text{Tr}\left(\boldsymbol{U}_{ij}\boldsymbol{J}_i\,\boldsymbol{U}_{ik}^T\right)\dot{q}_j\dot{q}_j\right] + \sum_{i=1}^{n} m_i\boldsymbol{g}^o(\boldsymbol{A}_i^o\,\boldsymbol{r}_i^i) \tag{3.33}$$

Applying the E- L formulation to the above equation of Lagrangian function of the robot arm, it yields the necessary generalized torque τ_i for joint *i* actuator to drive the i^{th} link of the manipulator,

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q} \tag{3.34}$$

$$= \sum_{j=1}^{n}\sum_{k=1}^{j} \mathrm{Tr}\left(\boldsymbol{U}_{jk}\boldsymbol{J}_j\,\boldsymbol{U}_{ji}^{T}\right)\ddot{q}_k + \sum_{j=i}^{n}\sum_{k=1}^{j}\sum_{m=1}^{j} \mathrm{Tr}\left(\boldsymbol{U}_{jkm}\boldsymbol{J}_j\,\boldsymbol{U}_{ji}^{T}\right)\dot{q}_k\dot{q}_m - \sum_{j=1}^{n} m_j\boldsymbol{g}^{o}\boldsymbol{U}_{ji}\,\boldsymbol{r}_j^{j} \tag{3.35}$$

for *i = 1, 2, . . ., n.* The above equation can be expressed in a much simpler matrix *notation* form as [81]

$$\tau_i = \sum_{k=1}^{n} M_{ik}\,\ddot{q}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\,\dot{q}_k\dot{q}_m + g_i \tag{3.36}$$

or in a matrix form as

$$\boldsymbol{\tau}(t) = \boldsymbol{M}\left(\boldsymbol{q}(t)\right)\ddot{\boldsymbol{q}}(t) + \boldsymbol{c}\left(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t)\right) + \boldsymbol{g}(\boldsymbol{q}(t)) \tag{3.37}$$

where

$\boldsymbol{\tau}(t) = n \times 1$ generalized torque vector applied at joints *i = 1, 2, . . ., n.* That is,

$$\boldsymbol{\tau}(t) = (\tau_1(t), \tau_2(t), \ldots, \tau_n(t))^T \tag{3.38}$$

$\boldsymbol{q}(t) = n \times 1$ vector of the joint variables of the robot arm and can be expressed as

$$\boldsymbol{q}(t) = (q_1(t), q_2(t), \ldots, q_n(t))^T \tag{3.39}$$

$\dot{\boldsymbol{q}}(t) = n \times 1$ vector of the joint velocity of the robot arm and can be expressed as

$$\dot{\boldsymbol{q}}(t) = (\dot{q}_1(t), \dot{q}_2(t), \ldots, \dot{q}_n(t))^T \tag{3.40}$$

$\ddot{\boldsymbol{q}}(t) = n \times 1$ vector of the acceleration of the joint variables $\boldsymbol{q}(t)$ and can be expressed as

$$\ddot{\boldsymbol{q}}(t) = (\ddot{q}_1(t), \ddot{q}_2(t), \ldots, \ddot{q}_n(t))^T \tag{3.41}$$

$\boldsymbol{M}(\boldsymbol{q}) = n \times n$ inertial acceleration-related symmetric matrix whose elements are

$$M_{ik} = \sum_{j=\max(i,k)}^{n} \text{Tr}\left(\boldsymbol{U}_{jk}\, \boldsymbol{J}_j\, \boldsymbol{U}_{ji}^T\right) \tag{3.42}$$

$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = n \times 1$ nonlinear Coriolis and centrifugal force vector whose elements are

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (c_1,\ c_2,\ \ldots,\ c_n)^T \tag{3.43}$$

where

$$c_i = \sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm} \dot{q}_k \dot{q}_m \tag{3.44}$$

And

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{n} \text{Trace}\left(\boldsymbol{U}_{jkm}\, \boldsymbol{J}_j\, \boldsymbol{U}_{ji}^T\right) \tag{3.45}$$

$\boldsymbol{g}(\boldsymbol{q}) = n \times 1$ gravity loading force vector whose elements are

$$\boldsymbol{g}(\boldsymbol{q}) = (g_1,\ g_2,\ \ldots,\ g_n)^T \tag{3.46}$$

where

$$C_i = \sum_{j=i}^{n} (-m_j \boldsymbol{g}^o \boldsymbol{U}_{ji}\, \boldsymbol{r}_j^j) \tag{3.47}$$

The motion equations of a manipulator as given by Eqns. (3.37) to (3.47) are coupled, nonlinear, second-order ordinary differential equations. For a given set of applied torques τ_i $(i = 1, 2, \ldots, n)$ as a function of time, Eqn. (3.37) should be integrated simultaneously to obtain the actual motion of the manipulator in terms of the time history of the joint variables $\boldsymbol{q}(t)$

Because of its matrix structure, the E-L equations of motion are appealing from the closed-loop control viewpoint in that they give a set of state equations as in Eqn. (3.37). This form allows design of a control law that easily compensates for all the nonlinear effects.

3.3 DEVELOPMENT OF THE DYNAMIC MODEL FOR MULTIPLE DOFs MANIPULATORS

To show how to use the E-L equations of motion as given by Eqns. (3.37) to (3.47), some examples of multiple link manipulators are worked out in this section, to develop their dynamic model.

3.3.1 3-Link Manipulator

Consider a 3-link manipulator with all revolute joint. The physical dimensions of the links of the manipulator and the link parameters are given in Table 3.1.

Table 3.1: Physical dimensions and parameters of 3-link manipulator

No. of Links *(i)*	Link Masses *(m) (kg)*	Link Parameters				
		Initial Joint Angle (θ_i)*(deg)*	Link Length $(l_i = a_i)$*(m)*	Joint Offset Distance (d_i) *(m)*	Twist Angle (α_i) *(deg)*	Initial Joint Angle Velocity $(\dot{\theta}_i)$ *(deg/min)*
1	20	50	1.3	0	0	1
2	20	20	0.9	0	0	1
3	8	-110	0.7	0	0	1

To develop dynamic model of a 3-link manipulator as discussed in Section 3.2, first step is to calculate the homogeneous coordinate transformation matrices using Eqn. (3.5) as described below :

$$\boldsymbol{A}_1^0 = \begin{bmatrix} \cos\theta_1 & -\cos\alpha_1 \sin\theta_1 & \sin\alpha_1 \sin\theta_1 & a_1 \cos\theta_1 \\ \sin\theta_1 & \cos\alpha_1 \cos\theta_1 & -\sin\alpha_1 \cos\theta_1 & a_1 \sin\theta_1 \\ 0 & \sin\alpha_1 & \cos\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.48}$$

$$\boldsymbol{A}_1^0 = \begin{bmatrix} 0.6428 & -0.7660 & 0 & 0.8356 \\ 0.7660 & 0.6428 & 0 & 0.9959 \\ 0 & 0 & 1 & 1.0000 \\ 0 & 0 & 1 & 1.0000 \end{bmatrix} \tag{3.49}$$

$$\boldsymbol{A}_2^1 = \begin{bmatrix} \cos\theta_2 & -\cos\alpha_2 \sin\theta_2 & \sin\alpha_2 \sin\theta_2 & a_2 \cos\theta_2 \\ \sin\theta_2 & \cos\alpha_2 \cos\theta_2 & -\sin\alpha_2 \cos\theta_2 & a_2 \sin\theta_2 \\ 0 & \sin\alpha_2 & \cos\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.50}$$

$$\boldsymbol{A}_2^1 = \begin{bmatrix} 0.9397 & -0.3420 & 0 & 0.8457 \\ 0.3420 & 0.9397 & 0 & 0.3078 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{3.51}$$

$$\boldsymbol{A}_3^2 = \begin{bmatrix} \cos\theta_3 & -\cos\alpha_3 \sin\theta_3 & \sin\alpha_3 \sin\theta_3 & a_3 \cos\theta_3 \\ \sin\theta_3 & \cos\alpha_3 \cos\theta_3 & -\sin\alpha_3 \cos\theta_3 & a_3 \sin\theta_3 \\ 0 & \sin\alpha_3 & \cos\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.52}$$

$$\boldsymbol{A}_3^2 = \begin{bmatrix} -0.3420 & 0.9397 & 0 & -0.2394 \\ -0.9397 & -0.3420 & 0 & -0.6578 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{3.53}$$

From the definition of the $\boldsymbol{Q}_i$ matrix, for a rotary joint *i,* using Eqn. (3.10)

it is taken as:

$$\boldsymbol{Q}_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.54}$$

Now, the first derivative of the coordinate transformation matrix $\boldsymbol{A}_3^0$ is calculated using Eqn. (3.17) given by $\boldsymbol{U}_{ij} = \frac{\partial \boldsymbol{A}_i^0}{\partial q_j}$

$$U_{11} = Q_1 * A_1^0; \quad = \begin{bmatrix} -0.7660 & -0.6428 & 0 & -0.9959 \\ 0.6428 & -0.7660 & 0 & 0.8356 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.55}$$

$$U_{21} = Q_1 * A_1^0 * A_2^1; \quad = \begin{bmatrix} -0.9397 & -0.3420 & 0 & -1.8416 \\ 0.3420 & -0.9397 & 0 & 1.1434 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.56}$$

$$U_{22} = A_1^0 * Q_2 * A_2^1; \quad = \begin{bmatrix} -0.9397 & -0.3420 & 0 & -0.8457 \\ 0.3420 & -0.9397 & 0 & 0.3078 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.57}$$

$$U_{31} = Q_1 * A_1^0 * A_2^1 * A_3^2; \quad = \begin{bmatrix} 0.6428 & -0.7660 & 0 & -1.3916 \\ 0.7660 & 0.6428 & 0 & 1.6797 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.58}$$

$$U_{32} = A_1^0 * Q_2 * A_2^1 * A_3^2; \quad = \begin{bmatrix} 0.6428 & -0.7660 & 0 & -0.3958 \\ 0.7660 & 0.6428 & 0 & 0.8440 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.59}$$

$$U_{33} = A_1^0 * A_2^1 * Q_3 * A_3^2; \quad = \begin{bmatrix} 0.6428 & -0.7660 & 0 & 0.4500 \\ 0.7660 & 0.6428 & 0 & 0.5362 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.60}$$

Assuming all the products of inertia are zero, the pseudo-inertia matrix $\boldsymbol{J}_i$ using Eqn. (3.26) can be derived as

$$J_1 = \begin{bmatrix} \frac{1}{3}m_1 l_1^2 & 0 & 0 & -\frac{1}{2}m_1 l_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_1 l_1 & 0 & 0 & m_1 \end{bmatrix} = \begin{bmatrix} 11.2667 & 0 & 0 & -13 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -13 & 0 & 0 & 20 \end{bmatrix}, \tag{3.61}$$

$$J_2 = \begin{bmatrix} \frac{1}{3}m_2 l_2^2 & 0 & 0 & -\frac{1}{2}m_2 l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_2 l_2 & 0 & 0 & m_2 \end{bmatrix} = \begin{bmatrix} 5.4 & 0 & 0 & -9 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -9 & 0 & 0 & 20 \end{bmatrix}, \tag{3.62}$$

$$J_3 = \begin{bmatrix} \frac{1}{3}m_3 l_3^2 & 0 & 0 & -\frac{1}{2}m_3 l_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_3 l_3 & 0 & 0 & m_3 \end{bmatrix} = \begin{bmatrix} 1.3067 & 0 & 0 & -2.8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2.8 & 0 & 0 & 8 \end{bmatrix} \tag{3.63}$$

Then, using Eqn. (3.42), the elements of symmetric, positive-definite inertia matrix are calculated as

$$M_{11} = trace(U_{11} * I_1 * U_{11}') + trace(U_{21} * I_2 * U_{21}') + trace(U_{31} * I_3 * U_{31}') = 109.6294 \tag{3.64}$$

$$M_{12} = trace(U_{22} * I_2 * U_{21}') + trace(U_{32} * I_3 * U_{31}') = 31.2528 \tag{3.65}$$

$$M_{13} = \quad trace(U_{33} * I_3 * U'_{31}) \quad = 0.4448 \tag{3.66}$$

$$M_{21} = \quad trace(U_{22} * I_2 * U'_{21}) + trace(U_{32} * I_3 * U'_{31}) \quad = 31.2528 \tag{3.67}$$

$$M_{22} = \quad trace(U_{22} * I_2 * U'_{22}) + trace(U_{32} * I_3 * U'_{32}) \quad = 11.4629 \tag{3.68}$$

$$M_{23} = \quad trace(U_{33} * I_3 * U'_{32}) \quad = 0.4448 \tag{3.69}$$

$$M_{31} = \quad trace(U_{33} * I_3 * U'_{31}) \quad = 0.4448 \tag{3.70}$$

$$M_{32} = \quad trace(U_{33} * I_3 * U'_{32}) \quad = 0.4448 \tag{3.71}$$

$$M_{33} = \quad trace(U_{33} * I_3 * U'_{33}) \quad = 1.3067 \tag{3.72}$$

Thus, on solving above equation the symmetric, positive-definite mass-inertia matrix is found to be:

$$\boldsymbol{M} = \begin{bmatrix} 109.6294 & 31.2528 & 0.4448 \\ 31.2528 & 11.4629 & 0.4448 \\ 0.4448 & 0.4448 & 1.3067 \end{bmatrix}, \tag{3.73}$$

To derive the Coriolis and Centrifugal force vector term, Eqn. (3.44) is used for *i = 1,2, and 3.*

$$c_1 = h_{111}\dot{\theta}_1^2 + h_{112}\dot{\theta}_1\dot{\theta}_2 + h_{113}\dot{\theta}_1\dot{\theta}_3 + h_{121}\dot{\theta}_2\dot{\theta}_1 + h_{122}\dot{\theta}_2^2 + h_{123}\dot{\theta}_2\dot{\theta}_3 + h_{131}\dot{\theta}_3\dot{\theta}_1 + h_{132}\dot{\theta}_3\dot{\theta}_2 + h_{133}\dot{\theta}_3^2 \tag{3.74}$$

$$c_2 = h_{211}\dot{\theta}_1^2 + h_{212}\dot{\theta}_1\dot{\theta}_2 + h_{213}\dot{\theta}_1\dot{\theta}_3 + h_{221}\dot{\theta}_2\dot{\theta}_1 + h_{222}\dot{\theta}_2^2 + h_{223}\dot{\theta}_2\dot{\theta}_3 + h_{231}\dot{\theta}_3\dot{\theta}_1 + h_{232}\dot{\theta}_3\dot{\theta}_2 + h_{233}\dot{\theta}_3^2 \tag{3.75}$$

$$c_3 = h_{311}\dot{\theta}_1^2 + h_{312}\dot{\theta}_1\dot{\theta}_2 + h_{313}\dot{\theta}_1\dot{\theta}_3 + h_{321}\dot{\theta}_2\dot{\theta}_1 + h_{322}\dot{\theta}_2^2 + h_{323}\dot{\theta}_2\dot{\theta}_3 + h_{331}\dot{\theta}_3\dot{\theta}_1 + h_{332}\dot{\theta}_3\dot{\theta}_2 + h_{333}\dot{\theta}_3^2 \tag{3.76}$$

Now, the terms h_{ikm} used in above three equations can be calculated using Eqn. (3.45),

Then, the value of the Coriolis and Centrifugal term $\boldsymbol{c}(\theta, \dot{\theta})$ is found to be

$$\boldsymbol{c}(\theta, \dot{\theta}) = [c_1, c_2, c_3]' = [19.3513 \quad 15.4031 \quad -13.1121]' \tag{3.77}$$

These values of mass-inertia matrices ***(M),*** and Coriolis and centrifugal force vector ***(c)*** are used in Chapter 5 for performing numerical simulation using MATLAB.

3.3.2 4-Link Manipulator

Consider a 4-link manipulator with all revolute joint. The physical dimensions of the links of the manipulator and the link parameters are given in Table 3.2.

Table 3.2: Physical dimensions and parameters of 4-link manipulator

No. of Links *(i)*	Link Masses *(m) (kg)*	Link Parameters				
		Initial Joint Angle (θ_i) *(deg)*	Link Length $(l_i = a_i)$ *(m)*	Joint Offset Distance (d_i) *(m)*	Twist Angle (α_i) *(deg)*	Initial Joint Angle Velocity $(\dot{\theta}_i)$ *(deg/min)*
1	30	10	0.50	0	0	1
2	25	25	0.45	0	0	1
3	20	50	0.35	0	0	1
4	10	-55	0.25	0	0	1

The dynamic model of a 4-link manipulator can be developed in a same manner as discussed in Section 3.3.1,

Thus the results obtained for mass-inertia matrices ***(M),*** and Coriolis and centrifugal force vector ***(c)*** are :

$$\boldsymbol{M} = \begin{bmatrix} 52.2541 & 25.8444 & 6.8302 & 1.6069 \\ 25.8444 & 15.6847 & 5.3370 & 1.0196 \\ 6.8302 & 5.3370 & 2.7519 & 0.4593 \\ 1.6069 & 1.0196 & 0.4593 & 0.2083 \end{bmatrix}, \tag{3.79}$$

$$\boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = [-49.0208 \quad -0.0613 \quad 16.0068 \quad -2.8886]' \tag{3.80}$$

3.3.3 6-Link Manipulator

Consider a 6-link manipulator with all revolute joint. The physical dimensions of the links of the manipulator and the link parameters are given in Table 3.3.

The dynamic model of a 6-link manipulator can be developed in a same manner as discussed in Section 3.3.1.

Table 3.3: Physical dimensions and parameters of 6-link manipulator

No. of Links *(i)*	Link Masses *(m) (kg)*	Link Parameters				
		Initial Joint Angle (θ_i) *(deg)*	Link Length $(l_i = a_i)$ *(m)*	Joint Offset Distance (d_i) *(m)*	Twist Angle (α_i) *(deg)*	Initial Joint Angle Velocity $(\dot{\theta}_i)$ *(deg/min)*
1	10	75	1.0	0	0	1
2	7	-10	0.70	0	0	1
3	6	-60	0.60	0	0	1
4	2.5	-5	0.25	0	0	1
5	2	-45	0.20	0	0	1
6	1.5	5	0.15	0	0	1

Thus the results obtained for mass-inertia matrices ***(M),*** and Coriolis and centrifugal force vector ***(c)*** are :

$$M = \begin{bmatrix} 64.0570 & 29.1818 & 9.3721 & 1.8048 & 0.0556 & 0.0350 \\ 29.1818 & 16.6399 & 7.5154 & 1.7950 & 0.3532 & 0.0826 \\ 9.3721 & 7.5154 & 5.4142 & 1.5838 & 0.4932 & 0.1029 \\ 1.8048 & 1.7950 & 1.5838 & 0.6334 & 0.2527 & 0.0552 \\ 0.0556 & 0.3532 & 0.4932 & 0.2527 & 0.1427 & 0.0337 \\ 0.0350 & 0.0826 & 0.1029 & 0.0552 & 0.0337 & 0.0113 \end{bmatrix}, \quad (3.81)$$

$$\boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = [109.8582 \quad 25.6325 \quad -23.1041 \quad -10.2536 \quad -7.1378 \quad -1.2766]' \quad (3.82)$$

3.3.4 7-Link Manipulator

Consider a 7-link manipulator with all revolute joint. The physical dimensions of the links of the manipulator and the link parameters are given in Table 3.4.

The dynamic model of a 7-link manipulator can be developed in a same manner as discussed in Section 3.3.1,

Table 3.4: Physical dimensions and parameters of 7-link manipulator

No. of Links *(i)*	Link Masses *(m) (kg)*	Link Parameters				
		Initial Joint Angle (θ_i) *(deg)*	Link Length $(l_i = a_i)$ *(m)*	Joint Offset Distance (d_i) *(m)*	Twist Angle (α_i) *(deg)*	Initial Joint Angle Velocity $(\dot{\theta}_i)$ *(deg/min)*
1	4.5	170	0.45	0	0	1
2	4.5	0	0.45	0	0	1
3	5	-160	0.5	0	0	1
4	2.5	-10	0.25	0	0	1
5	2.5	0	0.25	0	0	1
6	2.5	-20	0.25	0	0	1
7	2.5	30	0.25	0	0	1

Thus the results obtained for mass-inertia matrices ***(M),*** and Coriolis and centrifugal force vector ***(c)*** are :

$$\boldsymbol{M} = \begin{bmatrix} 6.1983 & 3.3132 & 1.3205 & 1.2716 & 0.9158 & 0.4675 & 0.1686 \\ 3.3132 & 4.6805 & 6.1806 & 3.4786 & 2.1542 & 1.0159 & 0.2995 \\ 1.3205 & 6.1806 & 11.022 & 5.6809 & 3.3906 & 1.5644 & 0.4296 \\ 1.2716 & 3.4786 & 5.6809 & 3.2564 & 2.0462 & 0.9930 & 0.2735 \\ 0.9158 & 2.1542 & 3.3906 & 2.0462 & 1.3567 & 0.6942 & 0.1968 \\ 0.4675 & 1.0159 & 1.5644 & 0.9930 & 0.6942 & 0.3962 & 0.1200 \\ 0.1686 & 0.2995 & 0.4296 & 0.2735 & 0.1968 & 0.1200 & 0.0521 \end{bmatrix}, \quad (3.83)$$

$$\boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = [22.4914 \quad 6.1745 \quad -10.1424 \quad -11.6895 \quad -8.5965 \quad -6.3699 \quad 0.4682]' \quad (3.84)$$

3.4 SUMMARY

This chapter has presented the dynamic modeling of equations of motion of the robot arm using Euler-Lagrange formulation. Then, these equations are analytically implemented on the multiple link redundant manipulator to show the effectiveness of the method. The symmetric, positive-definite, mass-inertia matrices and the Coriolis and centrifugal force vector is obtained from these equation of motion for various multiple link manipulators. The results obtained in the last section are utilized in Chapter 5 to perform various computer simulations, while controlling the manipulator under various situations

CHAPTER 4

KINEMATIC CONTROL OF REDUNDANT MANIPULATORS

4.1 INTRODUCTION

Controlling the position of a robot is not as trivial as one could think. Serial robots are composed of concatenated joints, forming an open chain structure. With this structure, the inverse kinematic problem, that is, relating the position and/or orientation of the end effector of the manipulator with the position of each joint, might not have a closed analytical solution. Not having such an analytical solution can force the user to search for alternative solutions to the kinematics of the robot. And even when a closed analytical form exists, its equations may have multiple solutions.

Ever since the first industrial robots appeared, there has been a big concern about computing kinematics in a simple, reliable and fast way in order to optimize the robot performance. In this chapter, the methodology and the fundamentals of inverse kinematics algorithm using task priority are discussed. The meaning of Jacobian matrix and pseudoinverse in mapping the Joint space to Cartesian space is described. Also, the initiative behind the trajectory development is given, which helps in setting up straight line trajectories for the manipulator to track in Cartesian space.

4.2 THE IMPORTANCE OF JACOBIAN MATRIX

The differentiation of a vector function of a vector variable is known as the Jacobian and it is a matrix quantity. Jacobian maps joint space velocity to Cartesian space velocity. For any forward kinematics problem, consider a manipulator with *n*-DOF, whose joint variables are denoted by $\boldsymbol{\theta_i}$, where $i = 1, 2, \ldots\ldots, n$. Assume that a class of tasks can be described by *m* variables $\boldsymbol{r_j}$, $j = 1, 2, \ldots\ldots, m\ (m \leq n)$, and that the relation between $\boldsymbol{\theta_i}$ and $\boldsymbol{r_j}$ is given by

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{4.1}$$

where, $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots \ldots \ldots, \theta_n]^T$ is the joint vector, $\boldsymbol{r} = [r_1, r_2, \ldots \ldots \ldots, r_m]^T$ is the manipulation vector, and the superscript *T* denotes the transpose. Differentiating Eqn. (4.1) gives,

$$\frac{\Delta \boldsymbol{r}}{\Delta t} = \frac{\partial \boldsymbol{f}}{\partial \theta} \cdot \frac{\Delta \boldsymbol{\theta}}{\Delta t} \tag{4.2}$$

If, $\Delta \rightarrow 0$,

$$\frac{d\boldsymbol{r}}{dt} = \frac{\partial \boldsymbol{f}}{\partial \theta} \cdot \frac{d\boldsymbol{\theta}}{dt}$$

So, $$\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{4.3}$$

where, $\dot{\boldsymbol{r}} \boldsymbol{\epsilon}\, R^m$, $\dot{\boldsymbol{\theta}} \boldsymbol{\epsilon}\, R^n$ and $\boldsymbol{J}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \epsilon\, R^{m \times n}$. The matrix $\boldsymbol{J}(\boldsymbol{\theta})$ is the Jacobian [9].

Assume that the following condition is satisfied: Max rank $(\boldsymbol{J}(\boldsymbol{\theta})) = m$.

Failing to satisfy this condition usually means that the selection of manipulation variables is redundant and that the number of these variables *m* can be reduced. When condition (4.3) is satisfied, it is said that the degree of redundancy of this manipulator is *(n-m)*.

If, for some $\boldsymbol{\theta}$, Rank $(\boldsymbol{J}(\boldsymbol{\theta})) < m$, then it is said that the manipulator is in a singular state. This state is not desirable because the manipulation vector $\boldsymbol{r}$ cannot move in a certain direction, meaning that the manipulability is seriously deteriorated. For redundant manipulators, as $n > m$, it has infinite solutions [9].

4.3 MANIPULABILITY AND MANIPULABLE SPACE

In Cartesian space, $\boldsymbol{r} \epsilon\, R^m$ is used to represent the kinematic output of a robot manipulator. Here $\boldsymbol{r}$ is called the manipulation variable and $\boldsymbol{J}(\boldsymbol{\theta}) \epsilon\, R^{m \times n}$ is the corresponding Jacobian matrix. As a generalized expression of Eqn. (4.2) the following equation is used:

$$\boldsymbol{\delta r} = \boldsymbol{J}(\boldsymbol{\theta})\boldsymbol{\delta \theta} \tag{4.4}$$

This equation shows that $\boldsymbol{\delta r}$ is a linear mapping of $\boldsymbol{\delta \theta}$ by the Jacobian matrix. The Fig. 4.1 explains this linear mapping. Here, ***R(J)*** denotes the range space of $\boldsymbol{J}(\boldsymbol{\theta})$, which is a subspace in the *m*-dimensional space of $\boldsymbol{\delta r}$. The dimension of ***R(J)*** is the rank $\boldsymbol{J}(\boldsymbol{\theta}) \leq \min(m, n)$. The range space is a space which the manipulator can reach by atleast one orientation.

The orthogonal complement of the range space ***R(J)'*** indicates the subspace made up of all of the kinematically unrealizable motions $\boldsymbol{\delta r}$. The range space of the Jacobian matrix and its dimension are called as **manipulable space** and **degree of manipulability (DOM)** [9], respectively.

The motion $\boldsymbol{\delta r}$ is kinematically realizable, iff it is a member of the manipulable space. The manipulable space changes as the configuration of the manipulator changes. Since, rank of Jacobian matrix $\boldsymbol{J(\theta)}$ degenerates at singular points, so does the manipulable space. At singular points, a manipulator cannot move in degenerated directions [9].

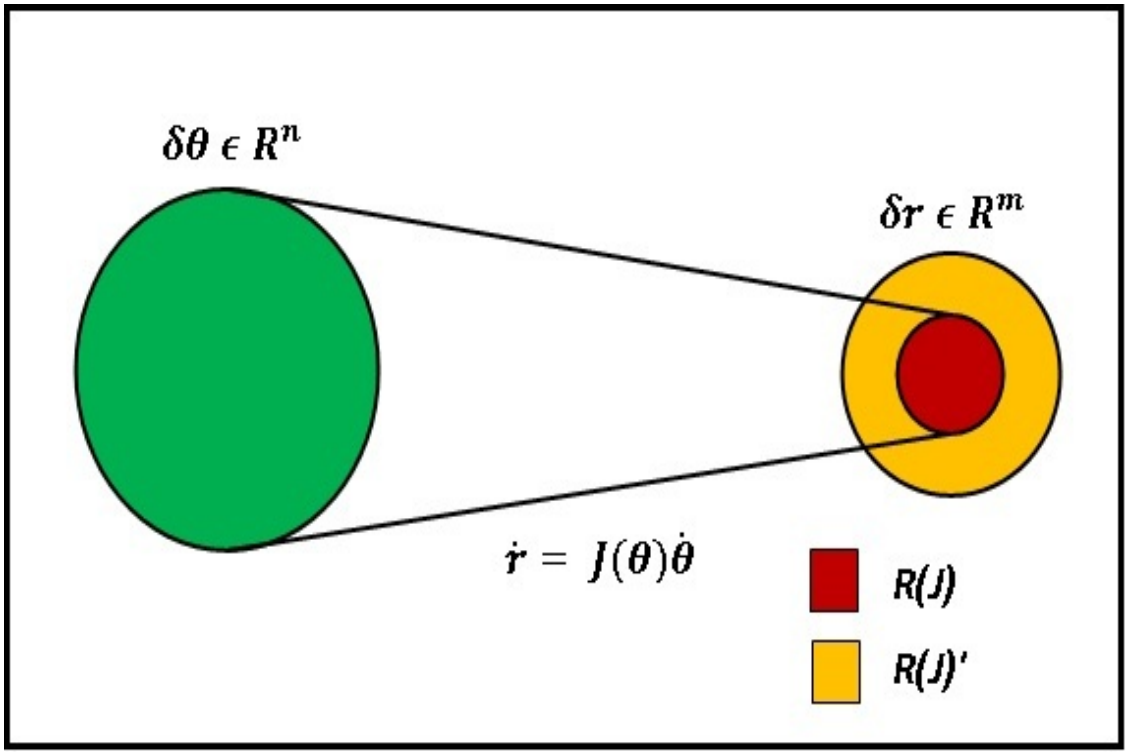

Fig. 4.1: Linear mapping of the manipulable space [9].

4.4 REDUNDANCY AND REDUNDANT SPACE

If the number of joints is greater than the dimension of the manipulation variable that is $n > m$, the manipulator is said to be redundant. Redundant manipulators have the characteristics that there exist infinite solutions of inverse kinematics problem.

Let $\boldsymbol{\partial\theta^1}$ $\boldsymbol{and}$ $\boldsymbol{\partial\theta^2}$ be two distinctive solutions of Eqn. (4.4) then,

$$\boldsymbol{J}(\theta)\partial\boldsymbol{\theta_e} = \boldsymbol{J}(\theta)\,[\partial\boldsymbol{\theta^1} - \partial\boldsymbol{\theta^2}] = 0, \tag{4.5}$$

where, $$\partial\boldsymbol{\theta_e} = \partial\boldsymbol{\theta^1} - \partial\boldsymbol{\theta^2} \neq 0 \tag{4.6}$$

These equations implies that the difference of two solutions is always mapped to the zero vector in $\boldsymbol{\delta r}$ space. The variety of solutions is represented by that of the vectors to be mapped to the

zero vectors by the Jacobian matrix. It is known that such vectors span a linear subspace in *n*-dimensional space of $\boldsymbol{\delta\theta}$, which is called as the **null space** *N(J).*

The null space of the Jacobian matrix *N(J*), and its mapping is shown in Fig. 4.2. The null space of the Jacobian matrix and its dimension are called as **redundant space** and **degree of redundancy** [9], respectively.

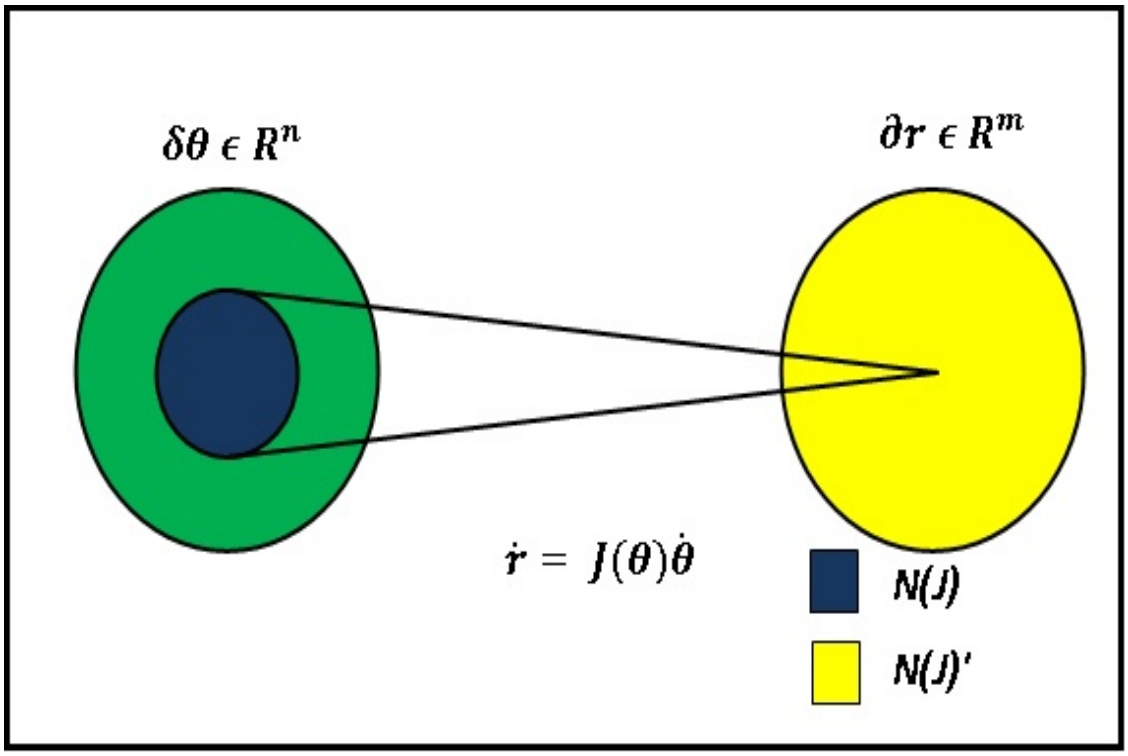

Fig.4.2: Redundant space[9].

Note that, if $n \leq m$ and rank $\boldsymbol{J(\theta)} = n$, there is no non zero entry in the null space of $\boldsymbol{J(\theta)}$, since the column vectors of $\boldsymbol{J(\theta)}$ are all independent. Accordingly, degree of redundancy (DOR) becomes zero and the solution of Eqn. (4.4) becomes unique. A more general relationship [9] between DOM and DOR is provided by using a well-known result from matrix theory that the range space and the null space of a matrix $\boldsymbol{M} \in R^{m\times n}$ should satisfy

$$dim\ R(M) + dim\ N(M) = n, \tag{4.7}$$

and

$$DOM + DOR = n. \tag{4.8}$$

This implies that degree of redundancy increases at singular points by the same number by which degree of manipulability decreases.

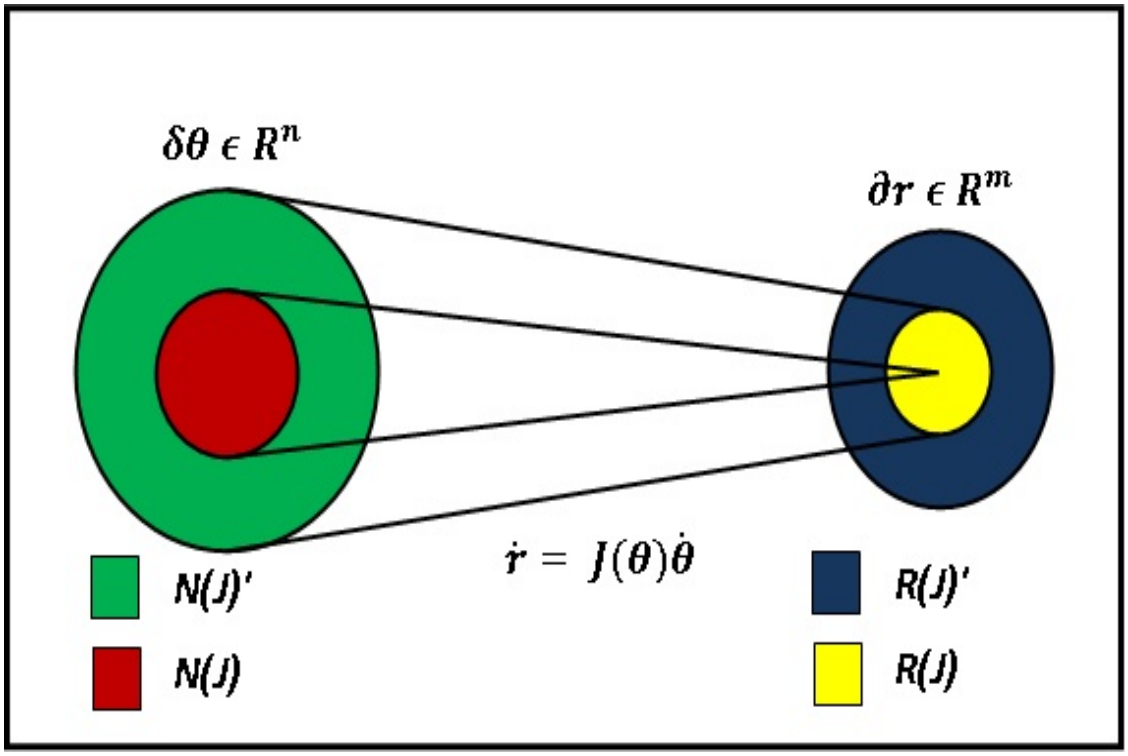

Fig.4.3: Linear mapping of the Jacobian space [9].

4.5 PSEUDOINVERSE

The inverse of a non-square matrix is known as pseudoinverse. A pseudoinverse of a matrix $\boldsymbol{J}(\boldsymbol{\theta}) \in R^{m\times n}$ is a matrix $\boldsymbol{J}^{\#}(\boldsymbol{\theta}) \in R^{n\times m}$.

Now, $$\dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{4.9}$$

Pre-multiply Eqn.(4.9) by $\boldsymbol{J}^{\#}$ gives

$$\boldsymbol{J}^{\#}\,\dot{\boldsymbol{r}} = \boldsymbol{J}^{\#}\,\boldsymbol{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{4.10}$$

It is known that $\boldsymbol{J}^{\#}\,\boldsymbol{J(\theta)}$ maps $N(\boldsymbol{J(\theta)})'$. On the similar terms, $\boldsymbol{(I - J^{\#}J(\theta))y}$ maps $N(\boldsymbol{J(\theta)})$.

The linear mapping of pseudoinverse is shown in Fig. 4.4 where, the range space and null space of the pseudoinverse matrix along with their orthogonal compliments are shown.

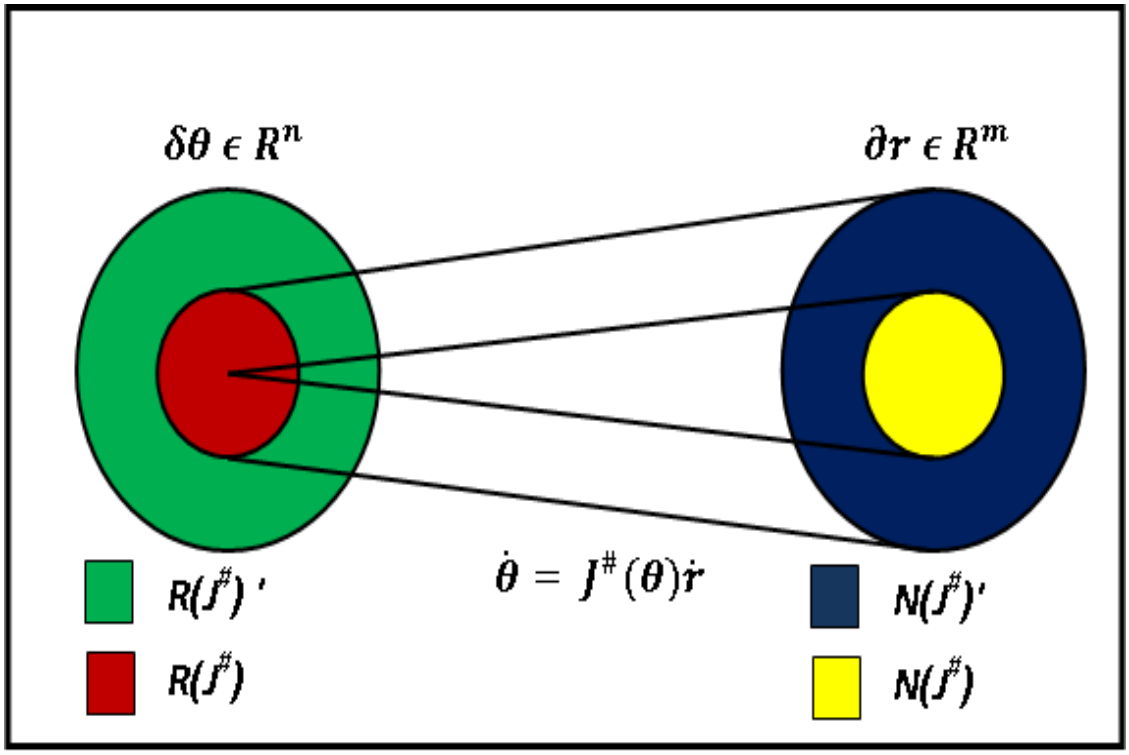

Fig.4.4: Linear mapping of the pseudoinverse $J^{\#}(\theta)$ [9].

4.6 TASK WITH THE ORDER OF PRIORITY

The problem of redundancy utilization can generally be formulated in the framework of tasks with the order of priority. When a redundant manipulator is required to follow the trajectory of the end effector, while avoiding obstacles or kinematic singular points, trajectory following is given first priority, and obstacle or singularity avoidance is given second priority.

For tasks with the order of priority, if it is impossible to perform all the subtasks completely because of the degeneracy or the shortage of DOF, it would be reasonable to perform the most significant subtasks preferentially and the less important subtasks as much as possible using the remaining DOF. Even for a six DOF manipulator, the subtask decomposition between position and orientation is advantageous, because it will enlarge the reachable workspace of the first priority subtask by allowing incompleteness for the second priority subtask. In the following section, redundancy utilization is discussed based on the concept of tasks with the order of priority

4.7 INVERSE KINEMATICS CONSIDERING THE ORDER OF PRIORITY

Assume that a task is composed of two subtasks, which will be performed according to the order of priority. The first priority subtask is specified using the first manipulation variable, $\boldsymbol{r_1} \in R^{m_1}$ and the second priority subtask by the second manipulation variable, $\boldsymbol{r_2} \in R^{m_2}$. The kinematic relationships between the joint variable $\boldsymbol{\theta} \in R^n$ and the manipulation variable [9] are expressed by:

$$\boldsymbol{r_i} = \boldsymbol{f_i}(\boldsymbol{\theta}); \qquad (\mathrm{i} = 1, 2) \tag{4.11}$$

Their differential relationships are given as follows:

$$\dot{\boldsymbol{r_i}} = \boldsymbol{J_i}(\boldsymbol{\theta})\, \dot{\boldsymbol{\theta}}; \qquad (i = 1, 2) \tag{4.12}$$

For redundant manipulators, as $n > m$, the general solution to Eqn. (4.12) for $i = 1$ is obtained using the pseudoinverse of the Jacobian matrix as follows:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_1^{\#}(\boldsymbol{\theta})\, \dot{\mathbf{r}}_1 + \{\mathbf{I} - \mathbf{J}_1^{\#}(\boldsymbol{\theta})\, \mathbf{J}_1(\boldsymbol{\theta})\}\, \mathbf{y} \tag{4.13}$$

where $\boldsymbol{J_1^{\#}}(\boldsymbol{\theta}) \in R^{n \times m_1}$ is the pseudoinverse of $\boldsymbol{J_1}(\boldsymbol{\theta})$, $\boldsymbol{y} \in R^n$ is an arbitrary vector, and $\boldsymbol{I} \in R^{n \times n}$ indicates an identity matrix. If the exact solution does not exist Eqn. (4.13) covers all the least squares solution[20] that minimize $\|\dot{\boldsymbol{r_1}} - \boldsymbol{J_1}(\boldsymbol{\theta})\, \dot{\boldsymbol{\theta}}\|$.

On substituting Eqn. (4.13) in Eqn. (4.12) for i = 2, to obtain,

$$\boldsymbol{J_2}\, \{\, \boldsymbol{I} - \widehat{\boldsymbol{J}}_1^{\#} \boldsymbol{J_1}\, \}\, \boldsymbol{y} = \dot{\boldsymbol{r}}_2 - \boldsymbol{J_2}\, \widehat{\boldsymbol{J}}_1^{\#}\, \dot{\boldsymbol{r}}_1 \tag{4.14}$$

Obtain y that minimizes $\|\dot{\boldsymbol{r_2}} - \boldsymbol{J_2}(\boldsymbol{\theta})\, \dot{\boldsymbol{\theta}}\|$, same as Eqn. (4.13) as given by

$$\boldsymbol{y} = \widehat{\boldsymbol{J}}_2^{\#} \left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J_2} \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 \right) + \left\{ \boldsymbol{I} - \widehat{\boldsymbol{J}}_2^{\#} \widehat{\boldsymbol{J}_2} \right\} \boldsymbol{z} \tag{4.15}$$

where, $\widehat{\boldsymbol{J_2}} = \boldsymbol{J_2}\, \{\, \boldsymbol{I} - \widehat{\boldsymbol{J}}_1^{\#} \boldsymbol{J_1}\, \}$ and $\boldsymbol{z} \in R^n$ is an arbitrary vector.

The solution is obtained from Eqn. (4.13) and (4.15) as follows:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 + \left\{ \boldsymbol{I} - \boldsymbol{J_1^{\#}} \boldsymbol{J_1} \right\} \widehat{\boldsymbol{J}}_2^{\#} \left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J_2} \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 \right) + \left\{ \boldsymbol{I} - \boldsymbol{J_1^{\#}} \boldsymbol{J_1} \right\} (\boldsymbol{I} - \widehat{\boldsymbol{J}}_2^{\#}\, \widehat{\boldsymbol{J_2}})\, \boldsymbol{z} \tag{4.16}$$

The second term of the Eqn. (4.16) is reduced to $\boldsymbol{J_2^{\#}} \left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J_2} \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 \right)$.

Thus, $$\dot{\boldsymbol{\theta}} = \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 + \widehat{\boldsymbol{J}}_2^{\#} \left(\dot{\boldsymbol{r}}_2 - \boldsymbol{J_2} \boldsymbol{J_1^{\#}}\, \dot{\boldsymbol{r}}_1 \right) + \left\{ \boldsymbol{I} - \boldsymbol{J_1^{\#}} \boldsymbol{J_1} \right\} (\boldsymbol{I} - \widehat{\boldsymbol{J}}_2^{\#}\, \widehat{\boldsymbol{J_2}})\, \boldsymbol{z} \tag{4.17}$$

The above equation represents the inverse kinematics solution taking account of the priority of the subtasks.

The manipulable space is the range space of the Jacobian matrix which is denoted by *R(J)*, and null space of the Jacobian matrix *N(J)* is the redundant space. Fig. 4.5 shows the general relationship between the manipulable and redundant spaces for the first and second manipulation variables. Subspace *A* shows all the possible contributions of $\dot{\boldsymbol{\theta}}$ to the first manipulation variable. Subspace *B* implies the contribution to the second manipulation variable without disturbing the first manipulation variable. Subspace *C* shows the remaining DOF that affect neither the first nor the second manipulation variables. Subspace *C* can be used for the third and higher manipulation variables, if necessary.

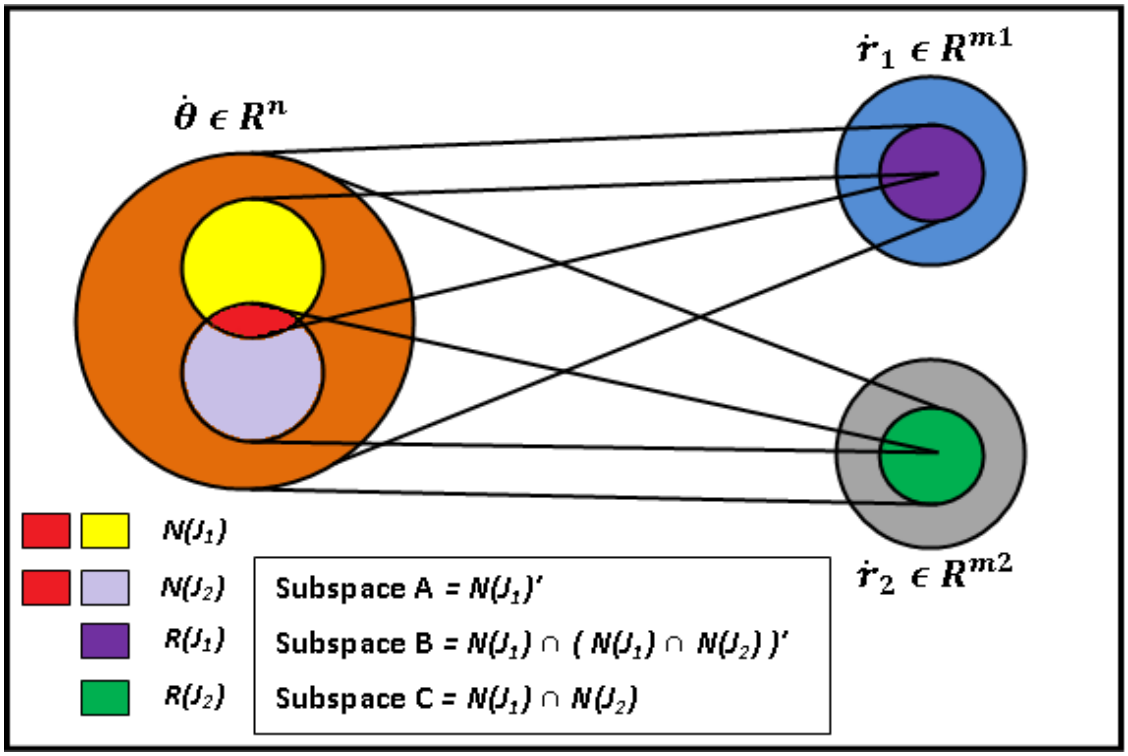

Fig.4.5: Manipulable and Redundant spaces for the first and second manipulation variables [5].

The first term in the right hand side of the Eqn. (4.17) is the least square mapping of $\dot{\boldsymbol{r}}_1$onto subspace *A*. the second term implies the least square mapping of $\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^{\#} \dot{\boldsymbol{r}}_1$ onto subspace *B*, where $\dot{\boldsymbol{r}}_2 - \boldsymbol{J}_2 \boldsymbol{J}_1^{\#} \dot{\boldsymbol{r}}_1$ is the modified desired value of the second manipulation variable due to the effect of the first term on the second manipulation variable. The third term is the orthogonal projection of the arbitrary vector $\boldsymbol{z}$ onto subspace *C*. if there is a third manipulation variable, the arbitrary vector $\boldsymbol{z}$ is determined in the same manner as $\boldsymbol{y}$ [9].

In the case of $\boldsymbol{r_2 = \theta}$, Eqn. (4.17) can be reduced to a simpler form using $\boldsymbol{J_2 = I}$ as follows:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J_1^{\#}\, \dot{r}_1} + (\boldsymbol{I} - \boldsymbol{J_1^{\#} J_1})\ \dot{\boldsymbol{r}}_2 \tag{4.18}$$

where $(\boldsymbol{I} - \boldsymbol{J_1^{\#} J_1})^{\#} = (\boldsymbol{I} - \boldsymbol{J_1^{\#} J_1})$ and the idempotency of $(\boldsymbol{I} - \boldsymbol{J_1^{\#} J_1})$ are used. In the above Eqn. (4.18), the term corresponding to the third term, in Eqn. (4.17) become intrinsically equal to zero, which means that zero DOF remains for the higher manipulation variables, because the second manipulation variable $\boldsymbol{r_2 = \theta}$ requires all the DOF remaining after being used for $\boldsymbol{r_1}$.

4.8 PATH PLANNING

A serial robot manipulator may be regarded as a variable geometrical chain of links that associate the configuration of its end-effector to the Cartesian coordinate frame in which the base frame is fixed. Forward kinematics is mathematical geometrical relations that help in finding the end effector configuration by the making use of the joint coordinates. On the other hand, the inverse kinematics are mathematical geometrical relations that provide joint coordinates from a given end effector configuration.

The velocity and acceleration of each link of the manipulator can be found by expressing the Cartesian path of motion for the end-effector as a function of time. The first applied path function that can provide a rest-to-rest motion is a cubic path for the joint variables $q_i(t)$ between two points $q_i(t_0)\ and\ q_i(t_f)$.

$$\boldsymbol{q_i(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3} \tag{4.19}$$

But to get zero acceleration or jerk at some point in our path, it is required to employ higher polynomials to satisfy the conditions. An *n* degree polynomial can satisfy *n+1* conditions.

$$\boldsymbol{q(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n} \tag{4.20}$$

It is also possible to split a multiple conditional path into some intervals with fewer conditions. The interval paths must then be connected to satisfy their boundary conditions.

A path of motion may also be defined based on different mathematical functions. Harmonic and cycloid functions are the most common paths. Non-polynomial equations introduce some advantages, due to simpler expressions, and some disadvantages due to nonlinearities.

When a path of motion either in joint or Cartesian coordinates space is defined, forward and inverse kinematics must be utilized to find the path of motion in the other space.

4.8.1 Manipulator Motion by End-Effector Path

Cartesian path planning has more application in robotics, because it can control the level of force and jerk inserted by the hand of a robot to the carrying object. Path planning in Cartesian space also determines the geometric constraints of the external world. However, a Cartesian path needs inverse kinematics to determine the time history of the joint variables.

Here for a straight line trajectory, consider a rest to rest Cartesian path from point (1,0) to point (1,1). A cubic polynomial can satisfy the position and velocity constraints at initial and final points.

$$\begin{aligned} Y(0) &= Y_0 = 0 \\ Y(1) &= Y_f = 1 \\ \dot{Y}(0) &= \dot{Y}_0 = 0 \\ \dot{Y}(1) &= \dot{Y}_f = 0 \end{aligned}$$

$$\boldsymbol{Y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3} \tag{4.21}$$

On differentiating Eqn. (4.21),

$$\boldsymbol{\dot{Y}(t) = a_1 + 2a_2 t + 3a_3 t^2} \tag{4.22}$$

Solving the above equations to get the value of coefficients as

$$\begin{aligned} a_0 &= 0 \\ a_1 &= 0 \\ a_2 &= 3 \\ a_3 &= -2 \end{aligned}$$

To get the Cartesian path as

$$\boldsymbol{X = 1} \tag{4.23}$$

$$\boldsymbol{Y = 3t^2 - 2t^3} \tag{4.24}$$

The manipulator moves from point *(1,0)* to *(1,1)* as the time changes from *0 to 1 minute* and traces a straight line at *X = 1* parallel to **y**-axis. This way any desired trajectory can be traced by forming its equations.

4.9 SUMMARY

In this chapter, the concept of inverse kinematics considering the order of priority has been discussed. According to which the task with lower priority is executed only if it does not conflict with the higher priority task. This is accomplished by projecting the constraint Jacobian onto the null space of the end-effector Jacobian; the order of priority being invertible. The main idea behind the trajectory planning is also described, which is helpful in planning different trajectories for the serial robot manipulators to follow.

On the basis of which various simulations are carried out in the next chapter. Main focus is given on obstacle avoidance under complex workspace, where the manipulator aims to follow a given trajectory along with avoiding collision with the obstacles in its path and workspace.

CHAPTER 5
RESULTS AND DISCUSSIONS

5.1 INTRODUCTION

In this chapter, the inverse kinematic algorithms have been suitably extended to redundant manipulators by adopting a task space augmentation technique. Formally, a functional constraint task (*obstacle avoidance and singularity avoidance*) is imposed to be satisfied along with the end-effector task. In order to avoid the problem of obstacle constraint task specification, a task priority strategy has been applied using potential function method.

Various simulations are conducted to show the redundancy resolution of the various manipulators, when the aim is to follow the straight line trajectory with different number of obstacles in its workspace with different parameters. All simulations demonstrate the efficacy of redundancy resolution to perform multiple task with the concept of task priority using potential function method. Moreover, extra obstacles are also placed in the path to ensure the redundancy resolution to its extent.

5.2 OBSTACLE AVOIDANCE USING POTENTIAL FUNCTION

In this work, local optimal control scheme is used, which resolves redundancy based on present information and this method is computationally economical. The Fig. 5.1 shows the physical model of a planar 4-DOF manipulator having length of each link l_1, l_2, l_3 *and* l_4 respectively. The corresponding joint angles are $\theta_1, \theta_2, \theta_3$ *and* θ_4. Two rectangular shaped-obstacles are also placed in the workspace of dimension $(L_{O_i}(m) \times H_{O_i}(m))$. The centre of gravity of the two is also marked as (CG_{xi}, CG_{yi}). The point p_i shows the mid-point of the i^{th} link of the manipulator which is used to keep the links away from the obstacle while tracking the trajectory. It is assumed that the mass of each link is uniformly distributed, and that the manipulator is constrained in a horizontal plane (planar robot). To carry out the simulation runs, the first step is to calculate the dynamic model.

As given in Eqn. (3.37), the dynamics of a general *n*-DOF manipulator is represented by

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{c}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) + \boldsymbol{g}(\boldsymbol{\theta}) \tag{5.1(a)}$$

where $\boldsymbol{M}(\boldsymbol{\theta}) \in R^{n\times n}$ is a symmetric, positive definite inertia matrix, $\boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in R^{n}$, is a torque vector, produced by the effect of Coriolis and Centrifugal forces and $\boldsymbol{g}(\boldsymbol{\theta})$ is the gravitational force vector. As, the current work is limited to planar manipulators only, the effect of gravitational forces can be neglected and Eqn. (5.1(a)) can be modified as:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{c}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right) \tag{5.1(b)}$$

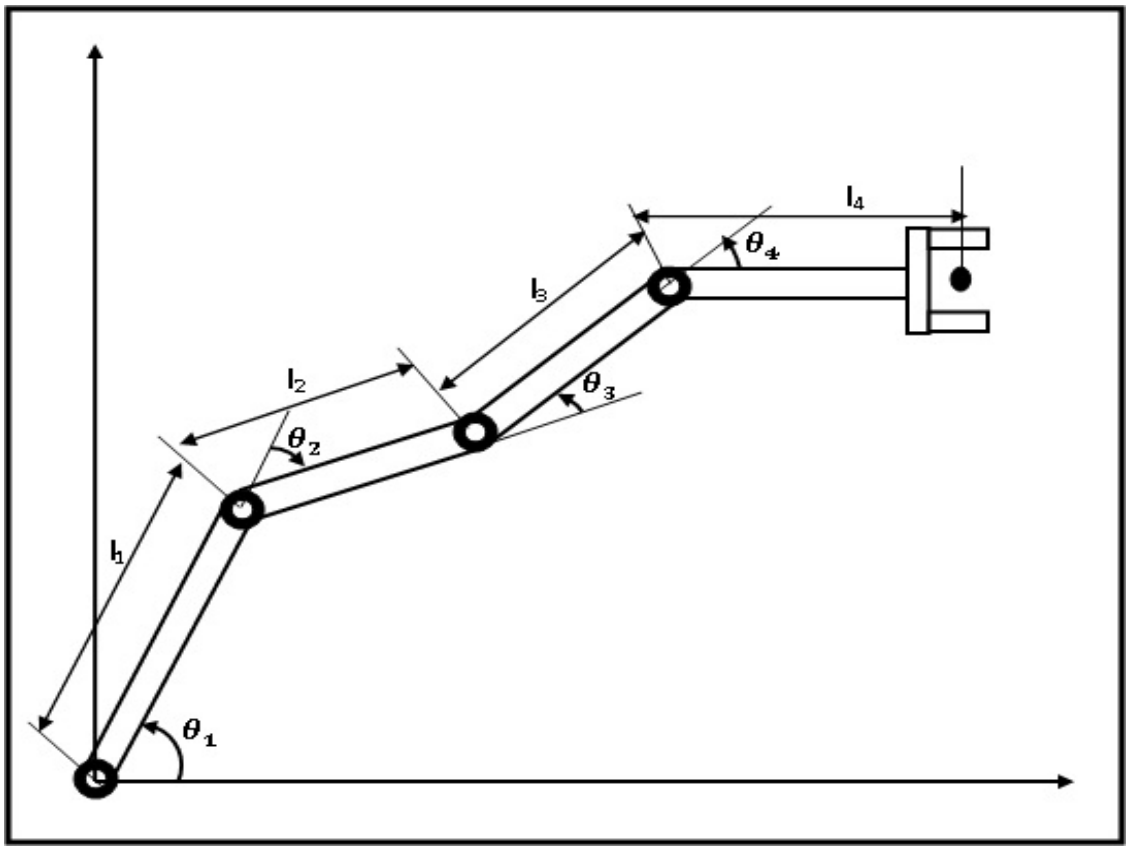

Fig. 5.1: The schematic diagram of a planar 5-DOF manipulator with two obstacles

Let the first manipulation variables is as follows:

$$\boldsymbol{r_1} = \begin{pmatrix} x_{tip} \\ y_{tip} \end{pmatrix} \tag{5.2}$$

where x_{tip} and y_{tip} represent the position of the end-effector in the Cartesian coordinates. By differentiating Eqn. (3.2) again, to obtain the following equation:

$$\ddot{\boldsymbol{r}}_{\boldsymbol{i}} = \boldsymbol{J}_{\boldsymbol{i}}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{J}}_{\boldsymbol{i}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \tag{5.3}$$

Now, a feedback control scheme is designed so that the following equation represents the closed-loop characteristics:

$$\ddot{\boldsymbol{e}}_{\boldsymbol{i}} + \boldsymbol{c}_{\boldsymbol{1i}}\dot{\boldsymbol{e}}_{\boldsymbol{i}} + \boldsymbol{c}_{\boldsymbol{2i}}\boldsymbol{e}_{\boldsymbol{i}} = \boldsymbol{0} \tag{5.4}$$

$$\boldsymbol{e}_{\boldsymbol{i}} \triangleq \boldsymbol{r}_{\boldsymbol{i}}^{\boldsymbol{d}}(\boldsymbol{t}) - \boldsymbol{r}_{\boldsymbol{i}} \tag{5.5}$$

where $\boldsymbol{r}_{\boldsymbol{i}}^{\boldsymbol{d}}(\boldsymbol{t})$ is the desired trajectory of the i^{th} manipulation variable, and $\boldsymbol{e}_{\boldsymbol{i}}$ is the error vector and c_{1i} and c_{2i} are the positive scalar feedback coefficients. Now, put the above feedback scheme in Eqn. (5.3), to get

$$\boldsymbol{J}_{\boldsymbol{i}}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} = \ddot{\boldsymbol{r}}_{\boldsymbol{i}}^{\boldsymbol{d}}(\boldsymbol{t}) - \dot{\boldsymbol{J}}_{\boldsymbol{i}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{c}_{\boldsymbol{1i}}\dot{\boldsymbol{e}}_{\boldsymbol{i}} + \boldsymbol{c}_{\boldsymbol{2i}}\boldsymbol{e}_{\boldsymbol{i}} \triangleq \boldsymbol{h}_{\boldsymbol{i}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{t}) \tag{5.6}$$

On comparing, Eqns. (3.2) and (5.6), it can be seen that both equations are similar, so the same approach is followed as before. For simplification solving Eqns. (3.18) and (5.6), to get

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}_{\boldsymbol{1}}^{\#}\boldsymbol{h}_{\boldsymbol{1}} + (\boldsymbol{E}_{\boldsymbol{n}} - \boldsymbol{J}_{\boldsymbol{1}}^{\#}\boldsymbol{J}_{\boldsymbol{1}})\ddot{\boldsymbol{r}}_{\boldsymbol{2}} \tag{5.7}$$

To calculate the expression for the acceleration of the second manipulation variable $\ddot{\boldsymbol{r}}_{\boldsymbol{2}}$, artificial potential and dissipative functions are used. These functions determine the joint torque for obstacle-avoidance problems when only the final goal of the end-effector is given and the trajectory is free.

According to this method, the joint torque is computed as follows:

$$\boldsymbol{\tau} = -\left(\frac{\partial P}{\partial \boldsymbol{\theta}} + \frac{\partial D}{\partial \dot{\boldsymbol{\theta}}}\right)^{T} \tag{5.8}$$

where P and D are the artificial potential and dissipative functions, respectively. If the joint torque of Eqn. (5.8) is applied, the following joint acceleration is generated from Eqn. (5.1(b)).

$$\ddot{\boldsymbol{\theta}} = -\boldsymbol{M}^{-\boldsymbol{1}}(\boldsymbol{\theta})\left\{\left(\frac{\partial P}{\partial \boldsymbol{\theta}} + \frac{\partial D}{\partial \dot{\boldsymbol{\theta}}}\right)^{T} + \boldsymbol{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\right\} \tag{5.9}$$

Now, the above equation is considered as the acceleration of the second manipulation variable. Hence, on combining the Eqns. (5.7) and (5.9), the joint acceleration considering both the first and second manipulation variables becomes,

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J_1^{\#}}\, \boldsymbol{h_1} - \left(\boldsymbol{E_n} - \boldsymbol{J_1^{\#} J_1}\right)\boldsymbol{M^{-1}}(\boldsymbol{\theta})\left\{\left(\frac{\partial P}{\partial \boldsymbol{\theta}} + \frac{\partial D}{\partial \dot{\boldsymbol{\theta}}}\right)^T + \boldsymbol{c}\left(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\right)\right\} \tag{5.10}$$

The artificial potential and dissipative functions used in the above equation are defined by

$$P = P_o + P_J \tag{5.11}$$

$$P_o = k_o \sum_{i=1}^{2(n-1)} (C_o(p_i) - 1)^{-1} \tag{5.12}$$

$$P_J = k_J \sum_{i=1}^{n} (\theta_{i_{max}}^2 - \theta_i^2)^{-1} \tag{5.13}$$

$$D = \frac{1}{2} k_D \sum_{i=1}^{n} \dot{\theta_i^2} \tag{5.14}$$

$$C_o(p_i) = \left(\frac{x_i - CG_{x_i}}{L_{O_i}/2}\right)^8 + \left(\frac{y_i - CG_{y_i}}{H_{O_i}/2}\right)^8 \tag{5.15}$$

where P_o and P_J are potential functions due to the obstacle and the joint limits, respectively and $C_o(p_i)$ is to approximate the contour of the obstacle. $L_{O_i}$and H_{O_i}denotes the length and height of the i^{th} obstacle in the workspace. The $p_i \triangleq (x_i, y_i)^T$ are the position of the mid points of the links of the manipulator and are used to evaluate the distance between the manipulator and the obstacle.

5.2.1 Numerical Simulations

a) 4-DOF Manipulator with One Obstacle

In the first simulation, which is shown in Fig. 5.2, a 4-DOF manipulator is considered. The desired task of which is given by $\boldsymbol{r_1^d}(t)$ to follow a straight line trajectory parallel to ***y***-axis without using concept of task priority for obstacle avoidance. The desired trajectory of the first manipulation variable $\boldsymbol{r_1^d}(t)$ is the constant velocity motion along the line from point *(1.1, 0.8)* at time *t = 0 min.* to point *(1.1, -0.2)* at time *t = 10 min* which is calculated by the cubic polynomial as discussed in section 3.8. The parameters of the i^{th} link of the manipulator are given in Table 5.1.

In Fig. 5.2, the effect of the rectangular shaped obstacle in the workspace of the manipulator is not considered. The virtual obstacle (O_1) colored with red, is considered as inactive. The

manipulator only fulfills the first task of tracing the desired straight line trajectory without concerning about the obstacle in its path.

Table 5.1: Parameters of 4-DOF manipulator

No. of Links *(i)*	Link Lengths $(l_i)(m)$	Link Masses $(m_i)(kg)$	Joint Angle $(\theta_i)(deg)$	Max. Joint Limit $\theta_{i_{max}}(deg)$
1	0.50	30	10	180
2	0.45	25	25	120
3	0.35	20	50	120
4	0.25	10	-55	150

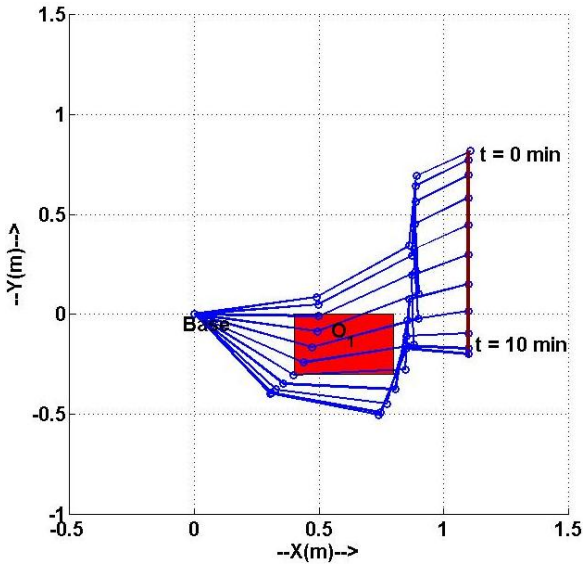

Fig.5.2: 4-DOF manipulator following straight line trajectory

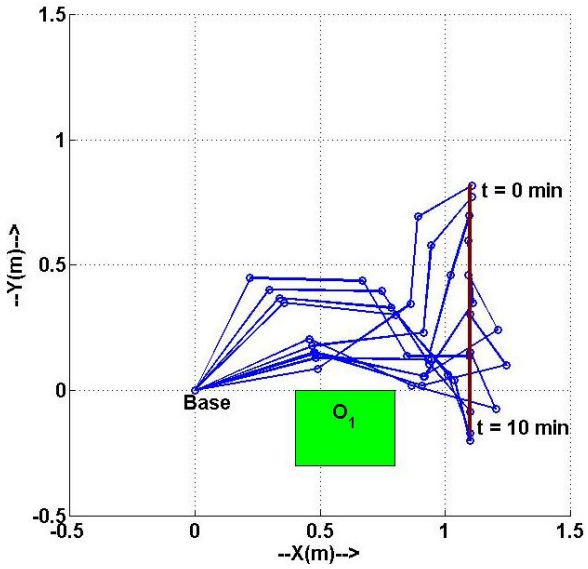

Fig.5.3: 4-DOF manipulator following straight line trajectory with obstacle avoidance

In the next simulation with same 4-DOF manipulator shown in Fig. 5.3, an rectangular shaped obstacle (O_1) is also considered in the workspace of the manipulator. The dimension of the obstacle are $(L_{O_1}(m) \times H_{O_1}(m)). = 0.4(m) \times 0.3(m)$ with centre of gravity at (CG_{x_1}, CG_{y_1}) = *(0.6, -0.15).* The feedback coefficients used in Eqn. (5.6) are chosen as $c_{11} = 50(1/min)$, $c_{21} = 100\ (1/min^2)$ and the coefficients k_o, k_j, and k_d used in Eqns. (5.12), (5.13) and (5.14) are *50(kgm^2/min^2), 6(kgm^2/min^2) and 1(kgm^2/min^2).*

It is clearly observed, that the manipulator due to redundancy, utilizes the concept of task priority for obstacle avoidance. Hence both the subtask of tracing the desired trajectory and obstacle avoidance are fulfilled by the manipulator.

The comparison between the joint trajectory of the manipulator in the above two simulation proves that there are many solution to the inverse kinematic problem, which can be used to resolve the redundancy.

b) Combination of Three Obstacles in Work Space of 3-DOF Manipulator

In this simulation, shown in Fig. 5.4, a 3-DOF manipulator is considered. The length of each link of the manipulator is taken as $\boldsymbol{l}$ = *[1.3, 0.9, 0.7](m)* with masses $\boldsymbol{m}$ = *[20, 20, 8](kg)* and initial configuration $\boldsymbol{\theta} = [50^0, 20^0, -110^0]$. The manipulator moves in a straight line trajectory from point *(1.69, 1.4)* at time *t = 0 min.* to point *(1.69, 0)* at time *t = 1 min.* So the time taken to perform the task is exactly *1 min.*

The desired first task ($\boldsymbol{r}_1^d(t)$) of the 3-DOF manipulator is to follow a straight line trajectory parallel to *y*-axis without concerning about any obstacle in its Cartesian space. In other words, the 3-DOF manipulator can maneuver its links with any joint configuration within the joint limit to trace the straight line trajectory in its workspace. In this simulation the desired second task ($\boldsymbol{r}_2^d(t)$) for collision avoidance with the obstacles in the path of maneuvering is not considered. Thus, it is considered as a minimum norm solution.

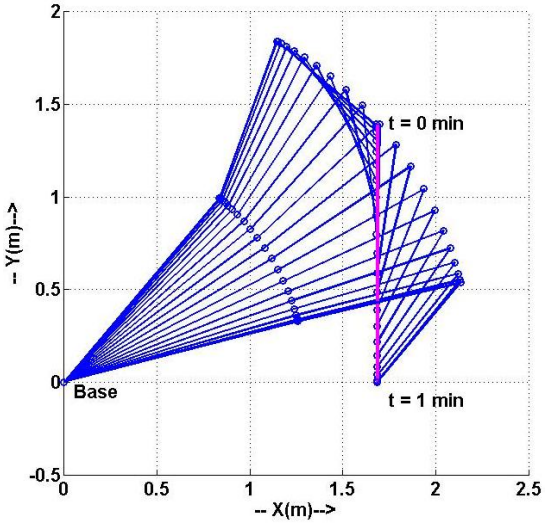

Fig. 5.4: 3-DOF manipulator following straight line trajectory with no obstacle in its path

Here, the trajectory tracing by the 3-DOF manipulator is given the higher priority. That means, the main preference is given to follow the trajectory without any deviation from the desired path. In the next simulation, three obstacles are considered in the workspace of manipulator named as O_1, O_2, *and* O_3 as shown in Fig. 5.5. The position of the i^{th} obstacle in the cartesian space can be calculated from its centre of gravity (x_i, y_i) and the size can be found from its length (l_i) and breadth (b_i) dimensions. The complete geometrical information of all the three obstacles is tabulated in Table. 5.2. The feedback coefficients used in Eqn. (5.6) are chosen as $c_{11} = 2000(1/min)$, $c_{21} = 1000\ (1/min^2)$.

Table 5.2: Geometrical data of three obstacles

Obstacle	Centre of Gravity		Dimensions *(m)*	
O_i	x_i	y_i	l_i	b_i
O_1	0.3	0.7	0.2	0.2
O_2	1.0	0.4	0.4	0.8
O_3	1.5	0.5	0.2	0.4

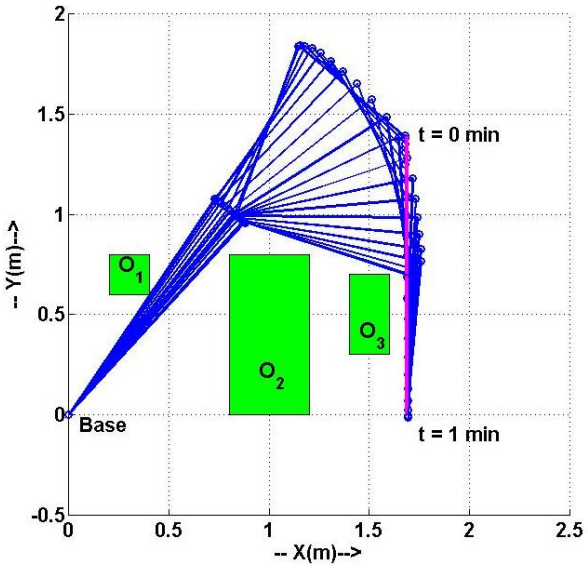

Fig. 5.5: 3-DOF manipulator following straight line trajectory with three obstacle in its path

This shows when the concept of obstacle avoidance using potential function is applied, the manipulator also fulfill the desired second task ($\boldsymbol{r}_2^d(t)$) to avoid collision with the three obstacles in its working space. The 3-DOF manipulator can take only those joint configurations at which

its links do not collide with the obstacles, while tracing the trajectory. That is, these links maintain a minimal distance with the obstacles during maneuvering. Thus, both the subtasks i.e. tracking the desired trajectory and obstacle avoidance, are fulfilled in overall part of the trajectory.

It can be clearly seen from Fig. 5.5 that the joint configurations acquired by the manipulator are different from those in Fig. 5.4. It is mainly due to the redundancy of the manipulator in nature and has infinite inverse kinematic solutions. Out of these solutions, the manipulator has chosen those solution, which can fulfill both the task efficiently.

The above simulation with three obstacles comprises of six individual simulation shown below for making comparison in the joint configuration space of links in detail by using concept of potential function to solve inverse kinematics for obstacle avoidance. Therefore, the next three simulations shown in Figs. 5.6(a)-(c), demonstrate the variation in joint trajectory, when only one obstacle is considered out of the three.

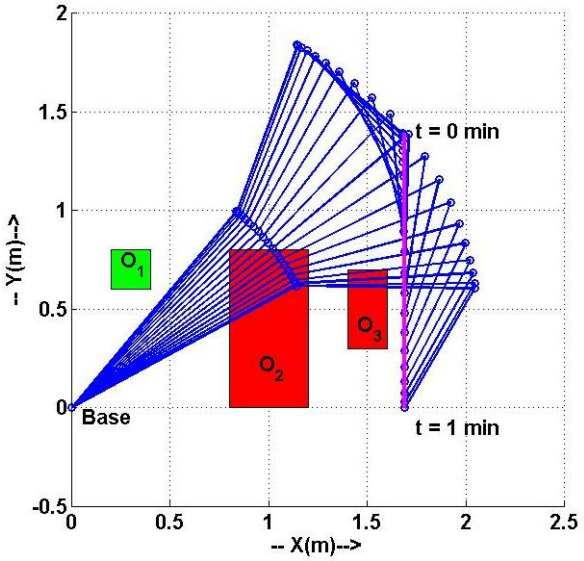

Fig. 5.6(a): 3-DOF manipulator following straight line trajectory with active obstacle1 in its path

As shown in Fig. 5.6(a), the links length, masses and initial configuration are taken same. One active obstacle which is considered during simulation is shown with green color and other two inactive obstacles with red color. This means, that the green color obstacle is contributing as a real obstacle in the workspace. Here the manipulator attains a different joint configuration so as

not to hit the active obstacle while tracing a trajectory and it is clearly seen that the manipulator is colliding with both the virtual inactive obstacles, while tracking the trajectory.

The similar trend of acquiring different joint configuration spaces with respect to one active obstacle at different locations in workspace is shown in Figs. 5.6(b) and 5.6(c).

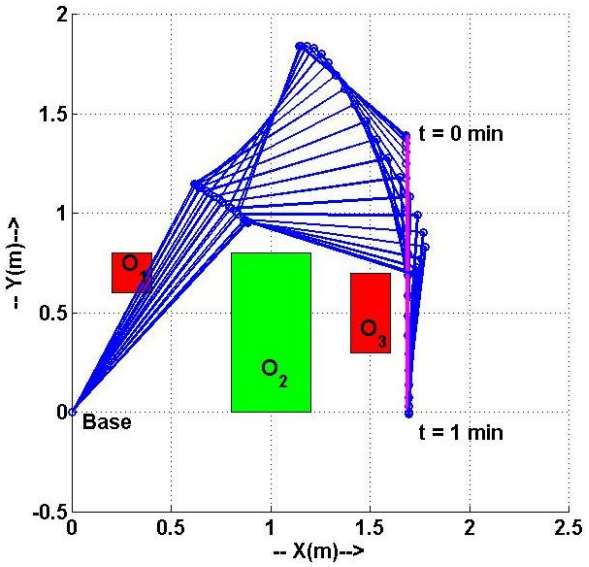

Fig. 5.6(b): 3-DOF manipulator following straight line trajectory with active obstacle1 in its path.

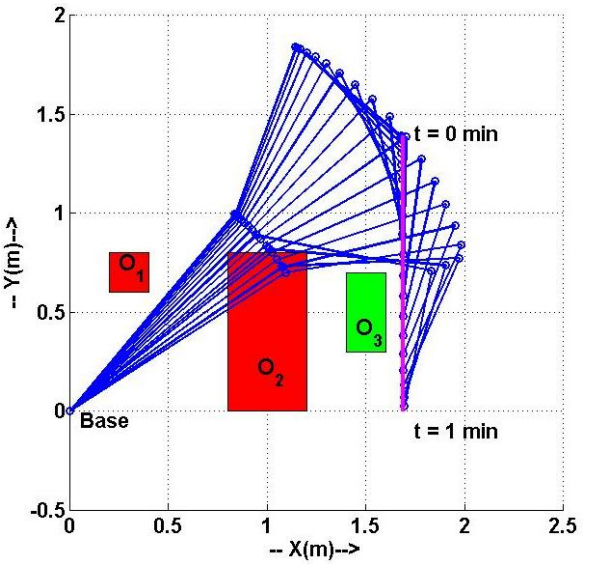

Fig. 5.6(c): 3-DOF manipulator following straight line trajectory with active obstacle2 in its path.

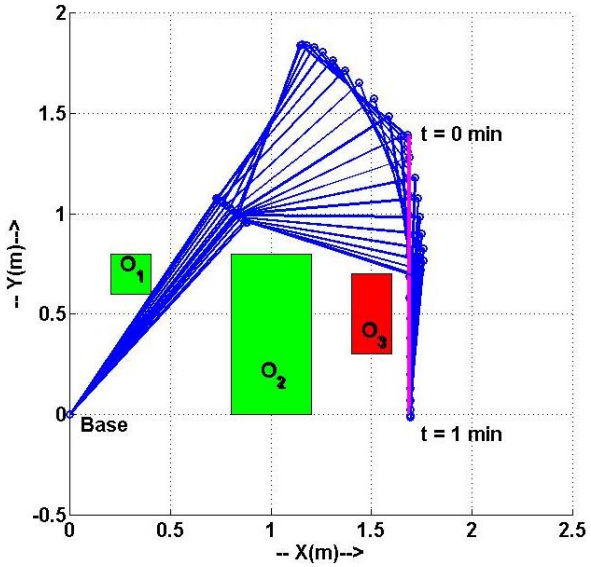

Fig. 5.7(a): 3-DOF manipulator following straight line trajectory with active obstacles1and 2 in its path

In the next set of three simulation run as shown in Fig. 5.7(a)-(c), the manipulator configuration and the obstacle configuration are same as above. In this section two active obstacles are considered in the workspace. This means, while calculating the inverse kinematic solutions using redundancy resolution, only two obstacles are active as shown in Fig. 5.7(a) with green color. The virtual third obstacle is taken inactive i.e. its contribution in the workspace is not considered. Thus the links are attaining different joint trajectories with in joint limit to avoid collision with the two active obstacle simultaneously and to trace the desired trajectory. The collision with the third inactive virtual obstacle is clearly shown in Fig. 5.7(a).

The similar trend of different joint trajectories can be observed in the next two simulation, where the two other active obstacle pairs are considered at different locations in the workspace as shown in Figs. 5.7(b) and 5.7(c). The above three simulation clearly demonstrate that how the redundant resolution scheme effectively maneuvers the joint trajectory to avoid the active obstacle, while tracing the desired trajectory

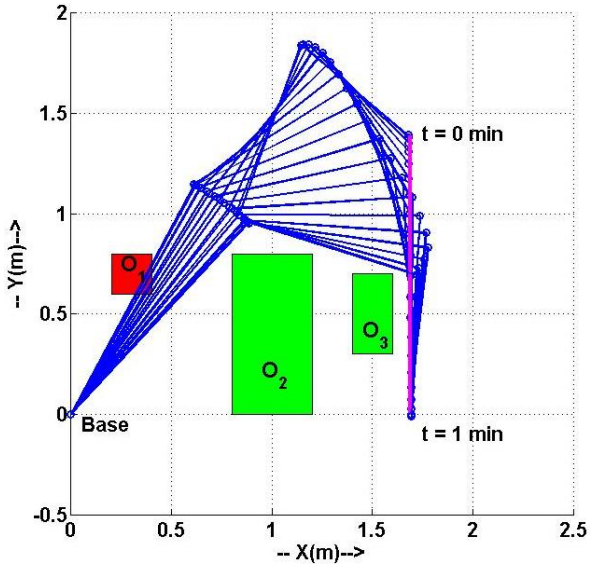

Fig. 5.7(b): 3-DOF manipulator following straight line trajectory with active obstacles 2and 3 in its path

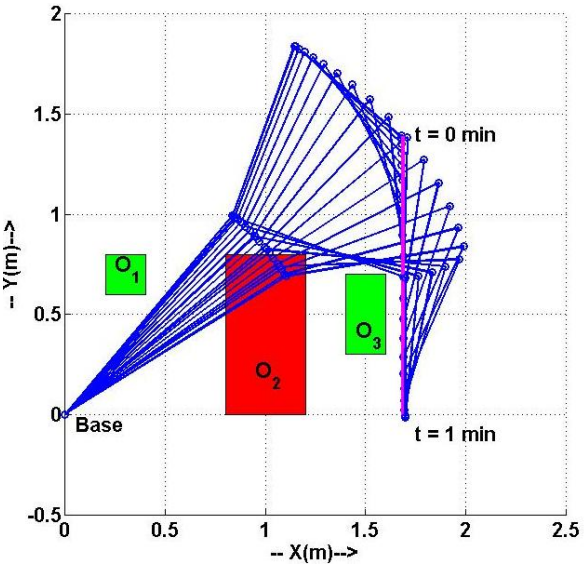

Fig. 5.7(c): 3-DOF manipulator following straight line trajectory with active obstacles 1and 3 in its path

This shows the validation of how the redundancy can be utilized to fulfill the desired first and second task simultaneously under complex workspace filled with one, two or three obstacle around the manipulator.

c) 6-DOF Manipulator with Vertical Channel Shaped Obstacle along with the Trajectory

The next simulation, demonstrates the effectiveness of redundancy resolution schemes to mimic a snake-like behavior. This behavior is required, when the robot is required to work in a water tank, or to check the blockage inside a sewerage pipe, or to do a laparoscopy operation inside a human body.

In this simulation, a 6-DOF manipulator is considered as shown in Fig 5.8. The geometrical configuration of the links of the manipulator is given in the Table 5.3. The desired trajectory of the first manipulation variable $\boldsymbol{r_1^d}(t)$ is the constant velocity motion along the line from point *(1.69, 1.39)* at time *t = 0 min.* to point *(1.69, 0)* at time *t = 1 min.* So the time taken to perform the task is exactly *1 min.*

Table 5.3: Parameters of 6-DOF manipulator

No. of Links *(i)*	Link Lengths $(l_i)(m)$	Link Masses $(m_i)(kg)$	Joint Angle $(\theta_i)(deg)$	Max. Joint Limit $\theta_{i_{max}}(deg)$
1	1	10	75	200
2	0.7	7	-10	200
3	0.6	6	-60	200
4	0.25	2.5	-5	200
5	0.2	2	-45	200
6	0.15	1.5	5	200

In Fig. 5.8, the effect of the channel shaped obstacle in the workspace of the manipulator is not considered. The virtual obstacles (O_1 and O_2) are considered as inactive colored with red. The manipulator only fulfills the first task of tracing the desired straight line trajectory without concerning about the obstacles along its path.

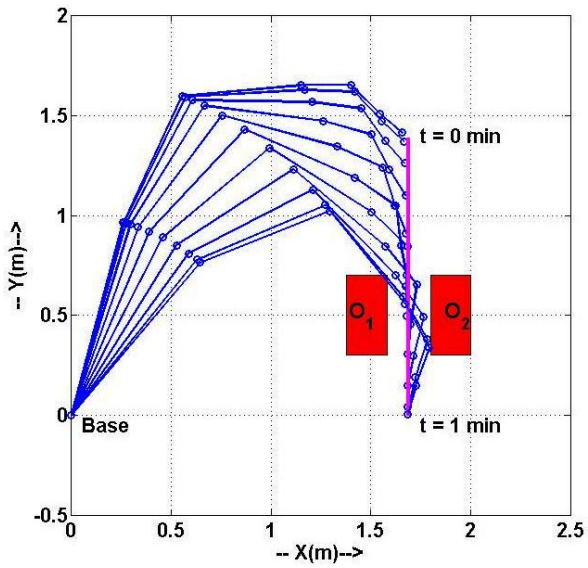

Fig.5.8: 6-DOF manipulator tracing straight line Trajectory

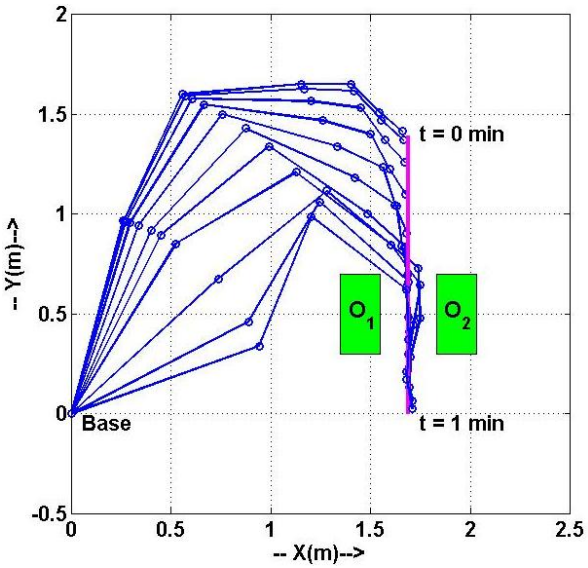

Fig. 5.9: 6-DOF manipulator with vertical channel shaped obstacle along its path

On the other hand in Fig. 5.9, vertical channel shaped obstacles, named as O_1 and O_2, are considered around the vertical trajectory. The position of the i^{th} obstacle in the Cartesian space can be calculated from its centre of gravity (x_i, y_i) and the size can be found from its length (l_i) and breadth (b_i) dimensions. The complete geometrical information of all the two obstacles is

tabulated in Table. 5.4. The feedback coefficients used in Eqn. (5.6) are chosen as $c_{11} = 1000(1/min),\ c_{21} = 2000\ (1/min^2)$.

This shows redundancy (extra DOF) of the manipulator allows its links to maneuvers in between the channel very smoothly to trace the straight line trajectory without colliding with the obstacles.

Table 5.4: Geometrical data of two vertical channel shaped obstacles

Obstacle	Centre of Gravity		Dimensions *(m)*	
O_i	x_i	y_i	l_i	b_i
O_1	1.45	0.5	0.2	0.4
O_2	1.93	0.5	0.2	0.4

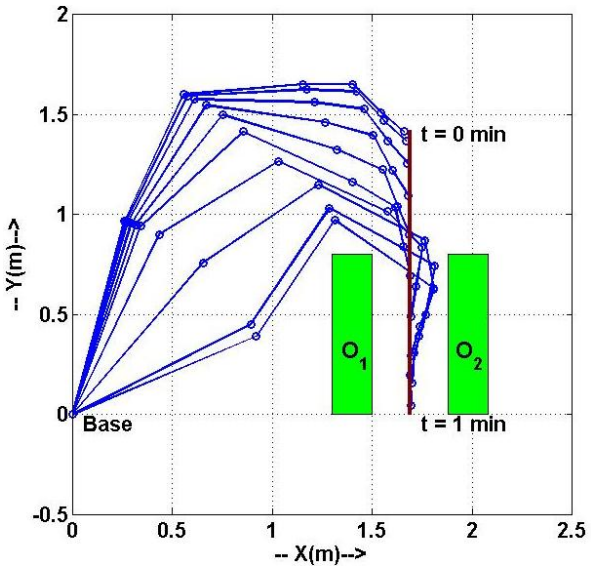

Fig. 5.10: 6-DOF manipulator with large vertical channel shaped obstacle along its path

In the next run of same simulation, the height of the vertical channel shaped obstacle is increased by *0.4(m)* as shown in Fig. 5.10. This utilizes the redundancy of manipulator to maximum extent by tracing the desired straight line trajectory without collision with the two obstacle along the path.

d) 7-DOF Manipulator with Horizontal Channel-Shaped Obstacle along the Trajectory

In this simulation, shown in Fig. 5.11, a 7-DOF manipulator is considered, which is expected to maneuver like a snake in a narrow horizontal channel. The length of each link of the manipulator is taken as $\boldsymbol{l}$ = [0.45, 0.45, 0.5, 0.25, 0.25, 0.25, 0.25] *(m)* with masses $\boldsymbol{m}$ = [4.5, 4.5, 5, 2.5, 2.5, 2.5, 2.5] *(kg)* and initial configuration $\boldsymbol{\theta}$ = [170, 0, -160, -10, 0, -20, 30] *(deg).* The manipulator moves in a straight line trajectory from point *(0.75, 0.2)* at time *t = 0 min.* to point *(2, 0.2)* at time *t = 1 min.* So the time taken to perform the task is exactly *1 min.*

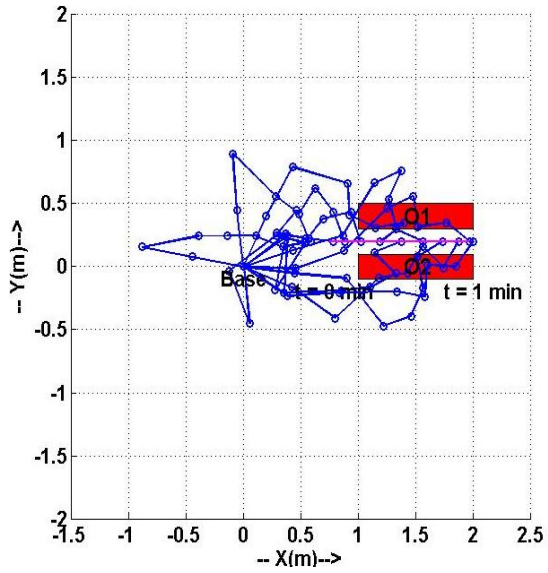

Fig. 5.11: 7-DOF manipulator following horizontal straight line trajectory

The desired first task ($\boldsymbol{r}_1^d(t)$) of the 7-DOF manipulator is to follow a straight line trajectory parallel to y-axis without concerning about any obstacle in its cartesian space. In other words, the 7-DOF manipulator can maneuver its links with any joint configuration within the joint limit to trace the straight line trajectory in its workspace. In this simulation, the desired second task ($\boldsymbol{r}_2^d(t)$) for collision avoidance with the obstacles in the path of maneuvering is not considered.

Here, the DOF of the manipulator required to complete the first task of desired trajectory tracing is less than 7-DOF that manipulator have. Thus, this redundancy in the links of the manipulator can be utilized to resolve the obstacle constraint or to fulfill the desired second task of obstacle avoidance

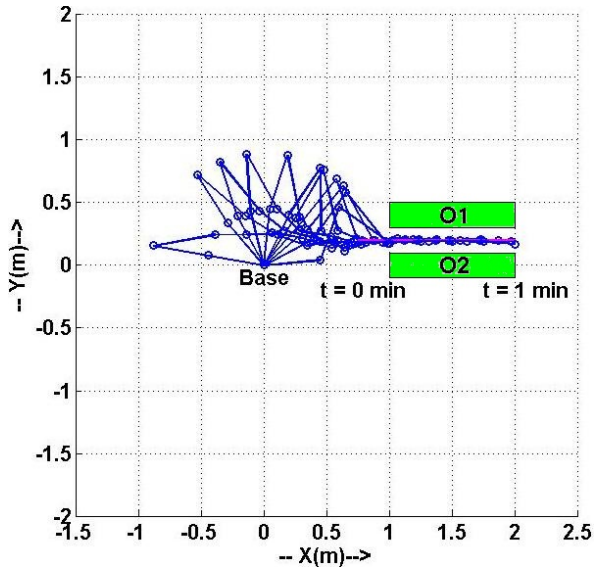

Fig. 5.12: 7-DOF manipulator following horizontal straight line trajectory along with obstacle avoidance

As shown in Fig. 5.12, horizontal channel shaped obstacle along the path of the trajectory are considered. The geometrical configuration of the obstacle are given in the Table 5.5. The feedback coefficients used in Eqn. (5.6) are chosen as $c_{11} = 300(1/min)$, $c_{21} = 90000\ (1/min^2)$. The 7-DOF manipulator resolve redundancy and fulfills both the subtasks and smoothly maneuvers between the horizontal channel shape obstacle to trace the trajectory while avoiding collision with the obstacle.

Table 5.5: Geometrical data of two horizontal channel shaped obstacles

Obstacle	Centre of Gravity		Dimensions *(m)*	
O_i	x_i	y_i	l_i	b_i
O_1	1.5	0.4	1.0	0.2
O_2	1.5	0.0	1.0	0.2

It is observed from Fig. 5.11, where the manipulator while maneuvering is colliding with the virtual obstacles, but no such collision is been reported in Fig. 5.12. In fact, as expected, the manipulator depicts a snake-like behavior and tracks the straight line trajectory in a narrow channel effectively.

It can be clearly seen in Fig 5.12 that while tracking the desired trajectory, the first and second links have very less relative movement and it appears that both are in a locked position, as if both the links were replaced by a single link of equivalent length. This set of configuration may also lead to singularity issues. As the manipulator chosen is highly redundant in nature, it is still possible for the manipulator to choose different configurations or solutions. To demonstrate this effect, one more obstacle is now placed at a appropriate location in the next simulation run to break the so-called locking posture of first and second link

In the next simulation as shown in Fig. 5.13, one more rectangular shaped obstacle is considered in the workspace of the manipulator along with the horizontal channel shaped obstacle. It is clearly observed that the manipulator does not collide with any of the obstacle while tracing the overall trajectory.

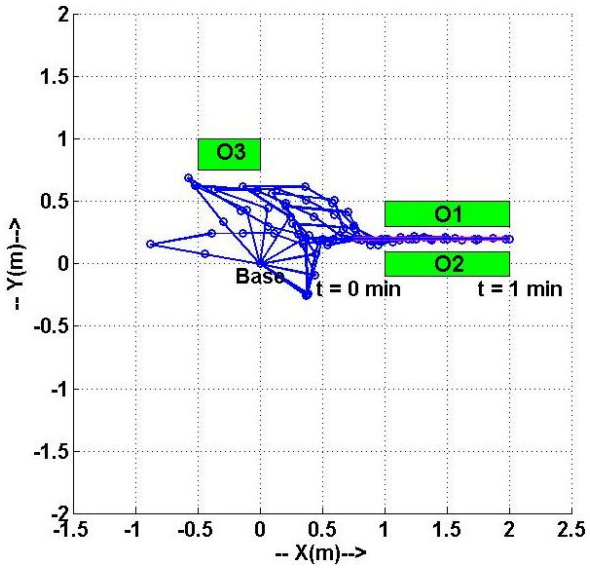

Fig. 5.13: 7-DOF manipulator with horizontal channel shaped and rectangular shaped obstacle following straight line trajectory

It can be clearly seen in Fig 5.13, that now the locking posture of first and second links is broken and the manipulator is still able to track the desired trajectory with different set of configurations, while avoiding all the obstacles.

5.3 SINGULARITY AVOIDANCE USING THE POTENTIAL FUNCTION

The measure of manipulability can be used as the potential function for avoiding singularity [9]. The potential function is now given as follows:

$$p(\theta) = -\sqrt{\det J_1 J_1^T} \tag{5.16}$$

The measure of manipulability, $\sqrt{\det J_1 J_1^T}$, is non-negative and becomes zero only at singular points. Therefore, it can be considered as a kind of distance from singular points. Choosing above equation as a potential function is expected to keep a manipulator away from singularity. The minimizing of the potential function implies not only avoiding the singularity, but also maintaining the kinematic ability of the manipulator as much as possible. The resultant motion of the manipulator is obtained by substituting derivative of Eqn. (5.16) into Eqn. (3.18) as the acceleration of the second manipulation variable.

5.3.1 Numerical Simulations

a) 3-DOF Manipulator with Singularity Avoidance

A 3-DOF manipulator is considered for numerical simulation. The link lengths are assumed to be $l_1 = 0.6,\ l_2 = 0.85\ and\ l_3 = 0.2\ (m)$. The given task for the first manipulation variable is to move in the negative x_2 direction with constant velocity 0.01(m/sec) for 10 seconds. The initial joint angles are $\theta(t_o) = (180^o\ -175^o\ \ 0^o)^T$.

Fig. 5.14(a) shows the result of simulation when the second term of Eqn. (3.18) is neglected. This is nothing but the resolved motion rate control. Fig. 5.14(b) shows the result when the second term is included in the same equation. Fig. 5.15 shows the change of the measure of manipulability. The graph with blue and red color corresponds to the two above cases of neglecting and including the second term respectively as discussed and corresponds to the cases of Figs. 5.14(a) and 5.14(b) respectively.

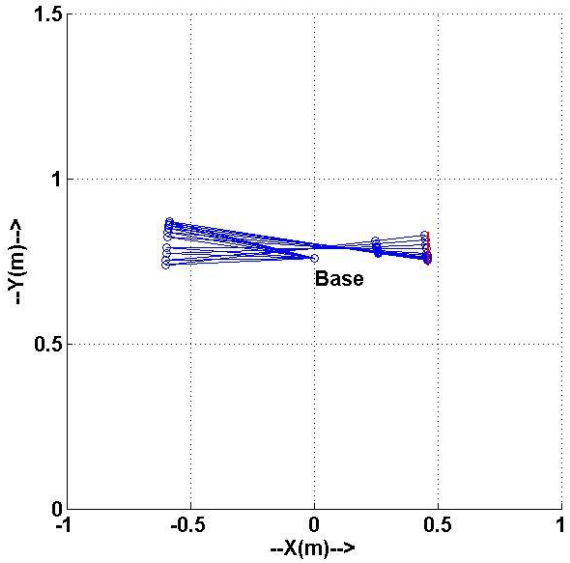

Fig. 5.14 (a) : 3-DOF manipulator without singularity avoidance

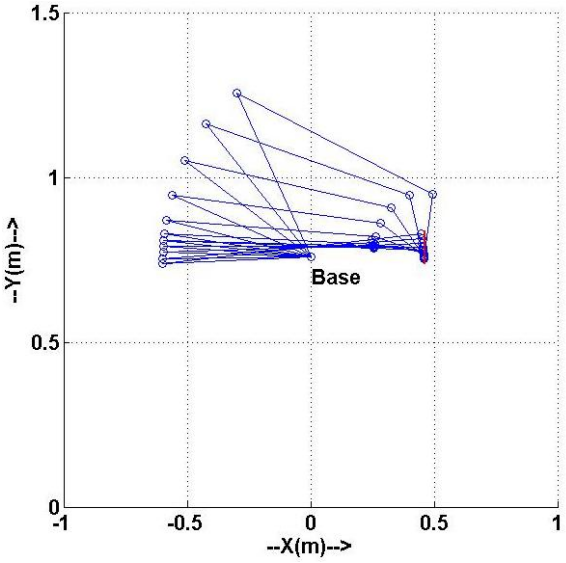

Fig. 5.14 (b) : 3-DOF manipulator with singularity avoidance

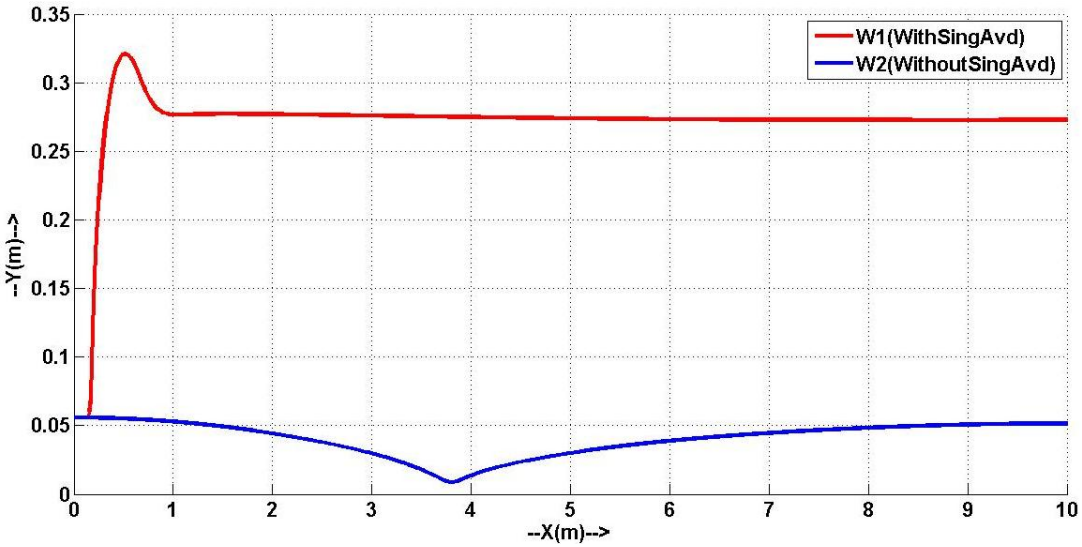

Fig. 5.15: Change of the measure of manipulability

From Figs. 5.14 and 5.15, it is seen that the conventional resolved motion rate control has no ability for moving away from the singularity, whereas utilizing redundancy by the potential function quickly drove the manipulator away from the singular points and afterward maintained the maximum value of the measure of manipulability.

5.4 SUMMARY

This chapter has presented different case studies to show the efficacy of the concept of task priority in resolving the redundancy of the serial robots. The redundancy resolution of the multiple DOFs manipulator is applied while the manipulator is following straight line trajectory. It is demonstrated that both the subtasks can be successfully completed using redundancy of the manipulator. In all the simulations, it is well demonstrated that it is possible to perform multiple tasks with the concept of task priority.

CHAPTER-6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

Various conclusions are drawn from the results that are obtained after simulating for obstacle avoidance and singularity avoidance under different complex workspace of the manipulator by utilizing the concept of task priority.

1. The purpose of quantifying the ability of redundant manipulators in manipulating their end-effector can be solved by studying the redundancy of robot manipulators.
2. The purpose of task priority is discussed and an inverse kinematics solution is derived taking into account the order of priority, which can be regarded as an optimal solution suitable for real time redundancy control.
3. This concept of task priority is then implemented for redundancy utilization of robot manipulators by dividing a task into a number of subtasks that can be used for tracing desired trajectory, obstacle avoidance and singularity avoidance.
4. Numerical simulation are performed in order to verify the effectiveness of the solution for redundancy control problems. Results shows that more degree of redundancy implies fulfillment of more subtasks (such as obstacle avoidance and singularity avoidance) according to the given priority.
5. The appropriate criteria to fulfill the lower priority subtasks is selected according to the workspace and the order of priority.
6. It is confirmed that dividing a task into subtasks with the order of priority is a remedy for overcoming the degeneracy of degree of freedom. Hence, a number of complex workspace are planned and the manipulators are aimed to trace the given trajectory along with avoiding collision with the obstacles.
7. The first priority is given to follow the desired trajectory and the second priority is to avoid collision with the obstacles while moving around the complex workspace.
8. Results shows that for a 6, 7-DOF manipulator is redundant enough, that the redundancy in the link can be utilize to avoid more number of obstacles in the workspace and can pass through very narrow long channels.

9. By adapting, the measure of manipulability as a second manipulation variable, singularity avoidance can be done. The redundancy of the manipulator droves it away from the singular points and afterward maintained the maximum value of the measure of manipulability.

6.2 FUTURE SCOPE

1. In this thesis work, the concept of task priority has been implemented on planar manipulators only. The work can be extended effectively to spatial manipulators also.
2. The work can be extended to demonstrate the effectiveness of task priority on physical prototype along with complex workspace.
3. The work can be extended to handle other subtasks like torque optimization, minimum movement etc.
4. Singularity avoidance can also be implemented on the more number of DOF manipulator to attain huge workspace.

REFERENCES

[1] Schunk Modular Robotics, http://www.schunk-modular-robotics.com/left navigation/ servicerobotics/ servicedownload/simulationcad/cad-data.html , 2011.

[2] Fraunhofer, Institute for Manufacturing Engineering and Automation, IPA. Website Accession : http://www.care-o-bot-research.org, 2010

[3] Leigh, R. H., B. Daniel and J. Beata, "*Reach and Grasp by People with Tetraplegia using a Neurally Controlled Robotic Arm*." Nature International Weekly Journal of Science, Vol. 485, No. 7398, pp. 372−375, 2012.

[4] Schneider, Hans-Christian and J. Wahrburg, "*Simulation Model for the Dynamics Analysis of a Surgical Assistance Robot, Robot Surgery*." , ISBN: 978-953-7619-77-0, 2010

[5] DLR Institute of Robotics and Mechatronics**,** German Center. Website Accession http: www.dlr.de/rm/en., 2012

[6] Canadian Space Agency. Website Accession date: 6-6-2012, http:\\www.asc-csa.gc.ca/eng/default.asp, 2012

[7] A. Osyczka, "*Multicriterion Optimization in Engineering*", Ellis Horwood Series in Engineering Science, 1984.

[8] G. Anandalingam, T. L. Friesz, "*Hierarchical Optimization*", Annals of Operations Research, Scientific Publishing Company, Basel, Vol. 34, No. 1, 1992.

[9] Y. Nakamura, Advanced Robotics, "*Redundancy and Optimization*," Addison-Welsey Publshing Company, 1991.

[10] A. Maciejewski, C. A. Klein, "*Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments*", International Journal of Robotics Research, Vol. 4, No. 3, pp. 109 − 117, 1985.

[11] Y. Nakamura, H. Hanafusa, T. Yoshikawa, "*Task-Priority Based Control of Robot Manipulators*", International Journal of Robotics Research, Vol. 6, No. 2, pp. 3−15, 1987.

[12] B. Siciliano, J. J. Slotine, "*A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems*", ICAR '91, pp. 1211 − 1216, 1991.

[13] N. I. Badler, K. H. Manoochehri, G. Walters, "*Articulated Figure Positioning by Multiple Constraints", IEEE Computer Graphics & Applications*, Vol. 7, No. 6, pp. 28 − 38, June 1987.

[14] J. Zhao, N. Badler, "*Inverse Kinematics Positioning using Nonlinear Programming for Highly Articulated Figures*", ACM Transactions on Graphics, Vol. 13, No. 4, pp. 313 – 336, Oct. 1994.

[15] C. B. Phillips, J. Zhao, N. Badler, "*Interactive Real-Time Articulated Figure Manipulation using Multiple Kinematic Constraints*", SIGGRAPH 90 Course Notes (Human figure animation : approaches and applications), 1990.

[16] C. B. Phillips, N. Badler, "*Interactive Behaviors for Bipedal Articulated Figures*", Computer Graphics, Vol. 25, No. 4, pp. 359 – 362, July 1991.

[17] C. A. Klein, C. H. Huang, "*Review of Pseudoinverse Control for use with Kinematically Redundant Manipulators*", IEEE Transactions on Systems, Man, Cybernetics, Vol. SMC-13, pp. 245 – 250, 1983.

[18] A. Liégeois, "*Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms*", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 12, pp. 868 – 871, 1977.

[19] R. Boulic, R. Mas, D. Thalmann, "*Inverse Kinetics for Center of Mass Position Control and Posture Optimization*", Race Workshop on "Combined Real and Synthetic Image Processing for Broadcast and Video Production (Monalisa Project)", Hamburg, Y. Parker & S. Wilbur Edt, Workshop in Computing Series, Springer-Verlag, 1994.

[20] R. Boulic, R. Mas, D. Thalmann, "*Complex Character Positioning Based on a Compatible Flow Model of Multiple Supports*", IEEE Transactions on Visualization and Computer Graphics, Vol. 3, No. 3, pp. 245 – 261, Sept. 1997.

[21] M. Hirose, G. Deffaux, Y. Nakagaki, "*Development of an Effective Motion Capture System Based on Data Fusion and Minimal use of Sensors*", VRST 96, July 1996.

[22] Y. Aydin, M. Nakajima, "*Database Guided Computer Animation of Human Grasping Using Forward and Inverse Kinematics*", Computers & Graphics, Vol. 23, No. 1, pp. 145 – 154, 1999.

[23] F. X. Lepoutre, "*Human Posture Modelization as a Problem of Inverse Kinematics of Redundant Robots*", Robotica, Vol. 11, pp. 339 – 343, 1993.

[24] R. Paul, "*Robot Manipulators: Mathematics, Programming and Control*", MIT Press, 1981.

[25] J. J. Craig, "*Introduction to Robotics: Mechanics and Control*", 2nd edition, Addison-

Wesley, 1989.

[26] J. U. Korein, "*A Geometric Investigation of Reach*", The MIT Press, Cambridge, 1985.

[27] D. Tolani, N. Badler, "*Real-Time Inverse Kinematics of the Human Arm*", Presence, Vol. 5, No. 4, pp. 393 – 401, 1996.

[28] D. Tolani, A. Goswami, N. Badler, "*Real-Time Inverse Kinematics Techniques for Anthropomorphic Arms*", Graphical Models, Vol. 62, pp. 353 – 388, 2000.

[29] D.E. Whitney, "*Resolved Motion Rate Control of Manipulators and Human Prostheses*", IEEE Transactions on Man-Machine Systems, Vol. MMS-10, No. 2, pp. 47 – 53, 1969.

[30] J. M. Ortega, W. C. Rheinboldt, "*Iterative Solution of Nonlinear Equations In Several Variables*", New York, Academic Press, 1970.

[31] T. Boullion, P. L. Odell, "*Generalized Inverse Matrices*", John Wiley and Sons, New York, 1971.

[32] A. Ben-Israel, T. N. E. Greville, "*Generalized Inverses: Theory and Applications*", John Wiley & Sons, 1974.

[33] A. S. Deo, I. D. Walker, "*Minimum Effort Inverse Kinematics for Redundant Manipulators*", IEEE Transactions on Robotics and Automation, Vol. 13, No. 5, pp. 767 – 775, 1997.

[34] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B.P. Flannery, "*Numerical Recipes in C*", Second edition, Cambridge University Press, 1992.

[35] K. Cleary, "*Incorporating Multiple Criteria in the Operation of Redundant Manipulators*", IEEE Proceedings, International Conference on Robotics and Automation, pp. 618 – 624, 1990.

[36] S. McGhee, T. F. Chan, R. V. Dubey, "*Probability-Based Weighting of Performance Criteria for a Redundant Manipulator*", IEEE Conference, pp. 1887 – 1894, 1994.

[37] R. Garziera, "*Recursive Formulation for the Inverse Kinematics of Redundant Robots Performing Tasks with Priority Order*", ASME Journal of Mechanical Design, Vol. 116, No. 1, pp. 337 – 338, March 1994.

[38] A. Neumaier, "*Solving Ill-Conditioned and Singular Linear Systems: A Tutorial on Regularization*", SIAM Review, Vol. 40, pp. 636 – 666, 1998.

[39] C. W. Wampler, "*Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Method*", IEEE Transactions System, Man, Cybernetics, Vol.

14, pp. 93−101, 1986.

[40] A. A. Maciejewski, C. A. Klein, "*Numerical Filtering for the Operation of Robotic Manipulators through Kinematically Singular Configurations*", Journal of Robotic Systems, Vol. 5, No. 6, pp. 527 − 552, 1988.

[41] S. Chiaverini, "*Task-Priority Redundancy Resolution with Robustness to Algorithmic Singularities*", Preprints Syroco '94, Capri, Italy, pp. 453 − 459, 1994.

[42] S. Chiaverini, "*Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators*", IEEE Transactions on Robotics and Automation, Vol. 13, No. 3, pp. 398 − 410, June 1997.

[43] C. Welman, "*Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation*", Master Thesis, Simon Fraser University, 1993.

[44] W. A. Wolovich, H. Elliot, "*A Computational Technique for Inverse Kinematics*", Proceedings of 32nd Conference on Decision and Control, pp. 1359 − 1362, Dec.1984.

[45] L. Sciavicco, B. Siciliano, "*A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators*", IEEE Journal of Robotics and Automation, Vol. 4, pp. 403 − 410, 1988.

[46] H. Das, J. J. Slotine, T. B. Sheridan, "*Inverse Kinematic Algorithms for Redundant Systems*", Proceedings of IEEE ICRA (International Conference on Robotics and Automation), pp. 43−48, 1988.

[47] G. R. Walsh, "*Methods of Optimization*", John Wiley and Sons, 1975.

[48] P. M. Pardalos, J. Ben Rosen, "*Constrained Global Optimization: Algorithms and Applications*", Springer-Verlag, 1987.

[49] J. Zhao, N. Badler, "*Real Time Inverse Kinematics with Joint Limits and Spatial Constraints*", Technical Report, University of Pennsylvania, 1989.

[50] D. Goldfarb, "*Extension of Davidon's Variable Metric Method to Maximisation under Linear Inequality and Equality Constraints*", SIAM Journal Applied Mathematics, Vol. 17, pp. 739 − 764, 1969.

[51] A. A. Khwaja, M. O. Rahman, M. G. Wagner, "*Inverse Kinematics of Arbitrary Robotic Manipulators Using Genetic Algorithms* in *Advances in Robot Kinematics: Analysis and Control*", Lenarcic and Husty (eds.), Kluwer Academic Publishers, pp. 375 − 382, 1998.

[52] G. S. Chirikjian, J. W. Burdick, "*A Modal Approach to Hyper-Redundant Manipulator*

Kinematics", IEEE Transactions on Robotics and Automation, Vol. 10, No. 3, pp. 343−354, June 1994.

[53] O. Omidvar, P. Van der Smagt (eds.), "*Neural Systems for Robotics*", Academic Press, 1997.

[54] S. Cameron, "*Collision Detection by Four-Dimensional Intersection Testing*", IEEE Transactions on Robotics and Automation, Vol. 6, No. 3, pp. 291 − 302, June 1990.

[55] A. Garcia-Alonso, N. Serrano, J. Flaquer, "*Solving the Collision Detection Problem*", IEEE Computer Graphics and Applications, pp. 36 − 43, May 1994.

[56] M. Moore, J. Wilhelms, "*Collision Detection and Response for Computer Animation*", Computer Graphics, Vol. 22, No. 4, Aug. 1988.

[57] J. C. Latombe, "*Robot Motion Planning*", Kluwer Academic Publishers, 1991.

[58] O. Khatib, "*Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*", IEEE International Conference on Robotics and Automation, St. Louis, 1985.

[59] B. Espiau, R. Boulic, "*Collision Avoidance for Redundant Robots with Proximity Sensors*", 3rd International Symposium on Robotics Research, Gouvieux, France, Oct. 1985.

[60] G. S. Chirikjian, J. W. Burdick, "*A Geometric Approach to Hyper-Redundant Manipulator Obstacle Avoidance*", ASME Journal of Mechanical Design, Vol. 114, pp. 580 − 585, Dec. 1992.

[61] X. Zhao, N. Badler, "*Interactive Body Awareness*", Computer Aided Design,Vol. 26, No. 12, pp. 861 − 868, Dec. 1994.

[62] Y. Koga, K. Kondo, J. Kuffner, J-C. Latombe, "*Planning Motions with Intentions*", Computer Graphics Proceedings, Vol. 28, pp. 395 − 408, 1994.

[63] Z. Huang, "*Motion Control for Human Animation*", Ph. D. thesis #1601, LIG - EPFL, 1997.

[64] J. C. Nebel, "*Keyframe Interpolation with Self-Collision Avoidance*", Computer Animation and Simulation '99, Springer, pp. 77 − 86, 1999.

[65] S. Chiaverini, "*Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators*," IEEE Transactions on Robotics and Automation, Vol. 13, No. 3, pp. 398−410, 1997

[66] C. Y. Chung, B. H. Lee, M. S. Kim, C. W. Lee, "*Torque Optimizing Control with Singularity-Robustness for Kinematically Redundant Robots*," Intelligent and Robotic

Systems, Vol. 28, No. 3, pp. 231−258, 2000

[67] R. S. Jamisola, "*Optimization of Failure-Tolerant Workspaces for Redundant Manipulators*," Philippine Science Letters; Vol. 3, No. 1, pp. 66−75, 2010.

[68] R. Penrose, "*A Generalized Inverse for Matrices*," In: Proceedings of the Cambridge Philosophical Society; Vol. 51, pp. 406–413, 1955.

[69] L. Sciavicco, B. Siciliano, "*A Solution Algorithm to The Inverse Kinematic Problem for Redundant Manipulators*," In: IEEE Transactions on Robotics and Automation; Vol. 4, pp. 403−410, 1998.

[70] S. Rotenberg. Inverse Kinematics Courses, available via <http://graphics.ucsd.edu/courses/cse169_w05/>, 2005.

[71] H. Ding, S. E. Chang, "*A Real-Time Planning Algorithm for Obstacle Avoidance of Redundant Robots,"* Intelligent and Robotic Systems; Vol. 16, pp. 229−243, 1996

[72] J. Baillieul, "*Kinematic Programming Alternatives for Redundant Manipulators, Robotics and Automation*," pp. 722−728, 1985.

[73] Y. Zhang, "*Minimum-Energy Redundancy Resolution Unified by Quadratic Programming*," In Proceedings of the IEEE International Conference on Robotics and Automation; pp. 2747−2752, 2003.

[74] J. Kim, P.K. Khosla, "*Real-Time Obstacle Avoidance Using Harmonic Potential Functions,"* In: IEEE Transactions on Robotics and Automation; Vol. 3, pp. 338−349, 1992.

[75] M. A. Ramalho, "*A Fuzzy Approach to Obstacle Avoidance for Small Robots,*" Bulletin of the Transivlania University Of Brasov; Vol. 15, No. 50, pp. 461−467, 2008.

[76] O. Khatib, "*Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*," Robotics Research; Vol. 5, pp. 90−99, 1986.

[77] R. Volpe, P. Khosla, "*Manipulator Control with Superquadric Artificial Potential Functions: Theory And Experiments*," In: Proceedings of IEEE Conference on Systems, Manufacturing and Cybernetics; Vol. 20, pp. 1423−1436, 1990.

[78] Y. Zhang, J. Wang, "*Obstacle Avoidance for Kinematically Redundant Manipulators Using a Dual Neural Network*," In: Proceedings of IEEE Conference on Systems, Manufacturing and Cybernetics; Vol. 34, No.1, pp. 752−759, 2004.

[79] Y. Zhang, S. Ma, "*Minimum-Energy Redundancy Resolution of Robot Manipulators*

Unified by Quadratic Programming and its Online Solution," In: Proceedings of the IEEE International Conference on Mechatronics and Automation; pp. 3232−3237, 2007.

[80] Y. Zhang et. al., "*More Illustrative Investigation on Window-Shaped Obstacle Avoidance of Robot Manipulators Using a Simplified LVI-Based Primal-Dual Neural Network*," In: Proceedings of the IEEE International Conference on Mechatronics and Automation, pp. 4240−4245, 2009.