A thesis report on

STATIC FORCE ANALYSIS OF KINEMATICALLY REDUNDANT SERIAL MANIPULATORS ALONG WITH ACTUATOR WEIGHT COMPENSATION

Submitted in the partial fulfilment of the requirement for

the award of the degree of

MASTER OF ENGINEERING

IN

CAD/CAM & ROBOTICS

Submitted by

Ankit Gupta

Roll No.: 801181003

Under the guidance of

Dr. Ashish Singla

Assistant Professor

Mechanical Engineering Department

Thapar University, Patiala.

MECHANICAL ENGINEERING DEPARTMENT

Thapar University, Patiala

July 2013

CERTIFICATE

I hereby certify that the thesis entitled "**STATIC FORCE ANALYSIS OF KINEMATICALLY REDUNDANT SERIAL MANIPULATORS ALONG WITH ACTUATOR WEIGHT COMPENSATION**" is an authentic record of my own work carried out in partial 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, Mechanical Engineering Department, Thapar University. The matter in this report has not been submitted in part or full to any other university or any institution for the award of any other degree.

Ankit Gupta
(801181003)

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

Dr. Ashish Singla
Assistant Professor
Mechanical Engineering Department
Thapar University, Patiala

Countersigned By:-

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

Dr. Ajay Batish
Professor and Head
Mechanical Engineering Department
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 a deep sense of gratitude and thank profusely to my guide **Dr. Ashish Singla** for his sincere & invaluable guidance, suggestions and attitude, which inspired me to submit dissertation in the present form. His dynamism and diligent enthusiasm have been highly instrumental in keeping my spirits high. His flawless and forthright suggestions blended with an innate intelligent application have crowned my task with success.

I take pride of myself being son of ideal parents for their everlasting desire, sacrifice, affectionate blessings, and help, without which it would not have been possible for me to complete my studies.

I would like to thank employees of **Mechanical Engineering Department, Thapar University, Patiala** for their everlasting support and helping advices. Above all, I express my indebtedness to the **almighty** for all his blessing and kindness.

Ankit Gupta

Abstract

Dynamic modelling of a serial manipultor is a challenging task, complexity of the problem increases with increase in number of links in serial manipulator system. But, problem becomes more critical, if the actuator weight compensation is taken into account.

In this thesis, MATLAB code is developed to generate the dynamic model of serial manipulators with any number of degrees-of-freedom or links. Taking actuator weight compensation into account, a lookup table is developed, which is helpful in selecting the appropriate actuator at each joint axis.

Static force analysis is done on the solid model generated, using Solid Works, to caculate stress, strain and deflection. Static force analysis is done for various cases like, wihtout actuator weight compensation, with actuator weight compensation and changing orientation of the actuators.

Moreover, the worst configuration torque values are compared with three random configurations, which proves that static force analysis done at worst configuration is sufficient to draw conclusion for the whole system.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_i	Link length of link *i*
a_i	Link twist
d_i	Joint offset
θ_i	Joint angle
$\dot{\theta}$	Joint angular velocity
$\ddot{\boldsymbol{\theta}}$	Joint angular acceleration
$\boldsymbol{F}$	Force
m	Mass
$\boldsymbol{N}$	Moment of inertia
l_1	Length of first link of the manipulator
l_2	Length of second link of the manipulator
n	Degree of freedom
$\boldsymbol{\tau}$	Joint torque vector
$\boldsymbol{M}(\boldsymbol{q})$	Inertia matix
$\boldsymbol{\omega}$	Angular velocity vector
$\dot{\boldsymbol{\omega}}$	Angular acceleration vector
${}_{i}^{i+1}\boldsymbol{R}$	Rotation matix
$\dot{\boldsymbol{v}}_{C_i}$	Acceleration at centre of the i^{th} link
$\boldsymbol{I}$	Inertia tensor
$\mathcal{L}$	Lagrangian function
KE	Kinetic energy
PE	Potential energy
${}^{i-1}\boldsymbol{A}_i$	Transformation Matrix
$\boldsymbol{P}_i$	Permutation Matrix
g	Gravity constant (9.81 m/sec^2)

Chapter 1 INTRODUCTION

1.1 A ROBOT

A robot can be defined as a programmable, self-controlled device consisting of electronic, electrical, or mechanical units. It is a computer-controlled machine that is programmed to move, manipulates objects, and accomplishes work while interacting with its environment. Robots are able to perform repetitive tasks more quickly, economically, and accurately than humans. The term robot originates from the Czech word 'robota', meaning 'compulsory labor'. The word robot has been used to refer to a machine that can work with humans, assist them or sometimes replace them in many engineering applications.

More generally, it is a machine that functions in place of a living agent. Robots are especially desirable for certain works as, unlike humans, they never get tired, they can endure physical conditions that are uncomfortable or even dangerous, they can operate in airless conditions, they do not get bored by repetition and they cannot be distracted from the task at hand. The present chapter gives the details of various terms and types of robots.

1.2 MANIPULATOR

A mechanical device which can handle remote objects or material even in the absence of any worker is called Manipulator. It consists of links and joints to make long chain, which can manipulate in its workspace. The number of joints gives the degrees-of-freedom (DOF). Manipulators are broadly classified into two categories namely: Parallel manipulators and Serial manipulators.

1.2.1 Parallel Manipulators

These manipulators use some base to move the arms parallelly, which are attached to it. These are different from the serial manipulator in the respect that the end effectors are connected to its base by a number of separate and independent linkages working in parallel. The parallel word is taken from the topological point of view rather than geometrical. These links works altogether in parallel but it does not mean that they always remain parallel to each other, they may be skewed to each other. A parallel manipulator is shown in Fig. 1.1.

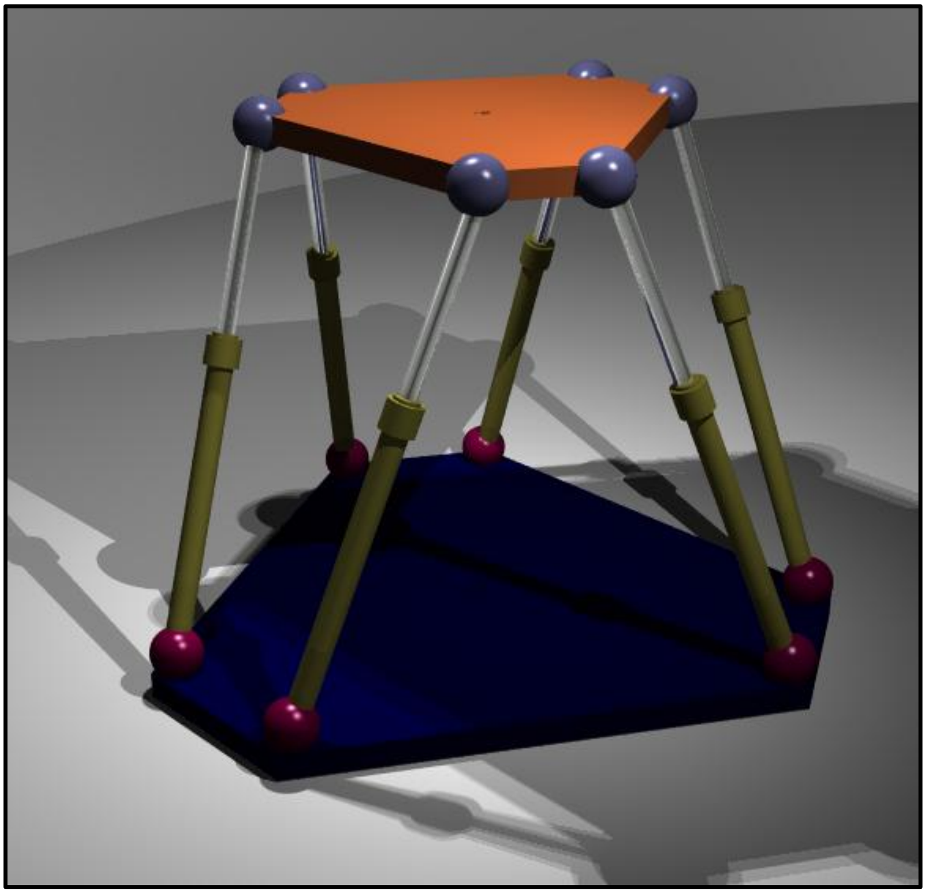

Fig. 1.1: Stewart Platform is an example of a parallel manipulator [1].

When a joint moves then there should be a proper movement of all parallel joints, which are controlled by actuators. If there is unwanted sloppiness then the other joints are also affected.

Parallel robots are alternate approach to manipulation. Each chain is usually short, simple and can thus be rigid against unwanted movement as compared to a serial arm. Errors in chain positioning are averaged in conjunction with the others, rather than being cumulative. Each actuator must still move within its own degree-of-freedom, as for a serial robot; however in the parallel robot the off-axis flexibility of a joint is also constrained by the effect of the other chains. It is this closed-loop stiffness that makes the overall parallel manipulator stiff relative to its components, unlike the serial chain that becomes progressively less rigid with more components.

One more advantage of the parallel manipulator is that the heavy actuators may often be centrally mounted on a single base platform, the movement of the arm taking place through struts and joints alone. This reduction in mass along the arm permits a lighter arm construction, thus lighter actuators and faster movements. This centralization of mass also reduces the robot's overall moment of inertia, which is an advantage for a mobile or walking robot.

Limited workspace is their one of main disadvantage, because the legs can collide and in addition each leg has some passive joints, each has their own mechanical limits. Also, they loose stiffness in singular positions completely. This means that the mapping from joint space to Euclidian space becomes singular.

1.2.2 Serial Manipulators

These are the most common industrial robots. Often they have links joined one to another in an open chain like structure, as shown in Fig. 1.2. One of their major advantages is that, they have large workspace with respect to their own volume and occupied floor space.

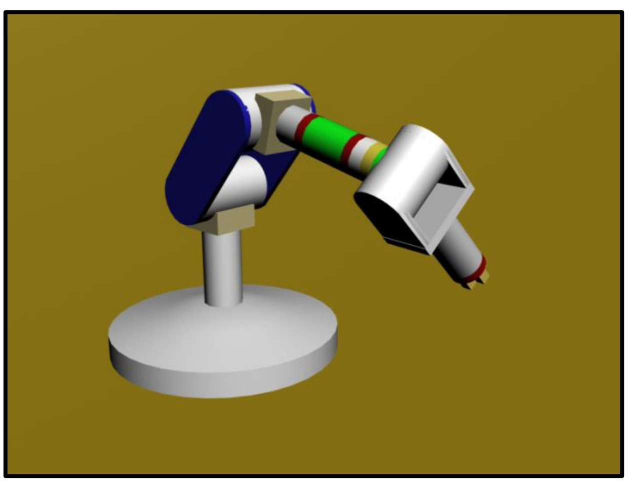

Fig. 1.2: An example of a serial manipulator with six DOF in a kinematic chain [2].

A popular application for serial robots in modern industry is the pick-and-place assembly robot, called a SCARA robot, which has four degrees of freedom as shown in Fig. 1.3.

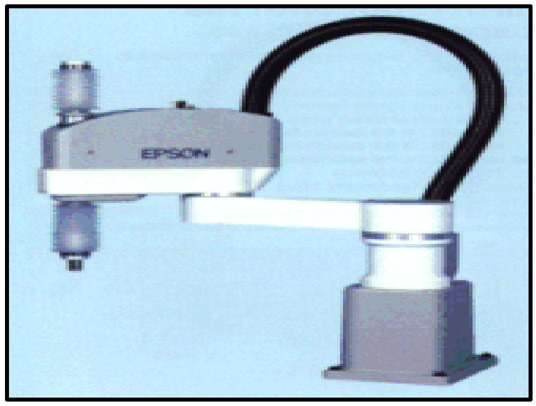

Fig. 1.3: SCARA Robot [3].

Fig. 1.4: Robotic Arm performing operation in space [4].

Another popular applications of serial manipulator are, to perform operations in environment which is not suitable for humans, like toxic industrial operations or in space operations as shown in Fig. 1.4.

Some of their limitations are:

- Have low stiffness due to an open kinematic structure.
- If there is any error present at the base joint, it gets accumulated and amplified along the chain length.

- These manipulators have to carry and move the large weight of most of the actuators.
- They can manipulate relatively small loads.

1.3 ABOUT SERIAL MANIPULATORS

1.3.1 General Structure

In its most general form, a serial robot consists of a number of rigid links connected with joints. Simplicity considerations in manufacturing and control have led to robots with only revolute or prismatic joints.

To reach at any position and orientation in workspace, minimum of six degrees-of-freedom (3 for position and 3 for orientation) are required. Hence, most of the manipulators are made with six degree-of-freedom. The four degree-of-freedom robots are only used for pick and place applications.

1.3.2 Redundant Manipulator

A redundant manipulator has more than six degrees of freedom which means that it has additional joint parameters that allow the configuration of the robot to change while it holds its end-effector in a fixed position and orientation.

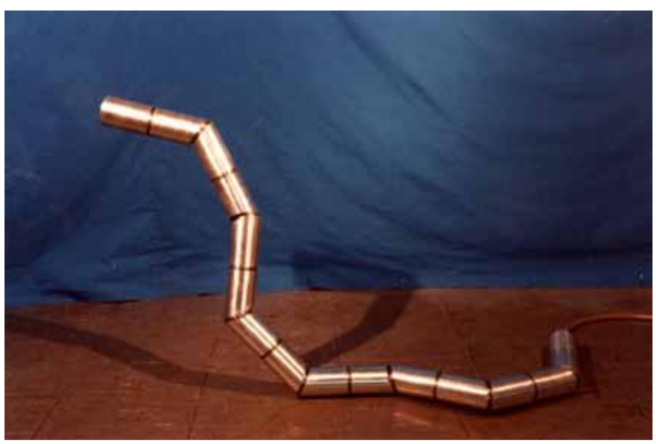

Fig. 1.5: An example for a hyper-redundant manipulator [5].

A snake robot has much more than six degrees of freedom and is often called hyper-redundant, as shown in Fig. 1.5.

A typical redundant manipulator has seven joints, for example three at the shoulder, one elbow joint and three at the wrist. This manipulator can move its elbow around a circle while it maintains a specific position and orientation of its end-effector.

1.3.3 Robot Arm Dynamics

Robot arm dynamics deals with the mathematical formulation of the equations of robot arm motion. The dynamic equations of motion of a manipulator are a set of mathematical equations describing the dynamic behavior of the manipulator. Such equations of motion are useful for computer simulation of kinematic design and structure of a robotic arm.

The purpose of manipulator control is to maintain the dynamic response of a computer-based manipulator in accordance with some pre-specified system performance and desired goals. In general, the dynamic performance of a manipulator directly depends on the efficiency of the control algorithms and the dynamic model of the manipulator. The control problem consists of obtaining dynamic models of the physical robot arm system and then specifying corresponding control laws or strategies to achieve the desired system response and performance.

This development is important in several ways, namely,

- A dynamical model can be used to develop the suitable control strategies. A sophisticated controller requires the use of a realistic dynamical model to achieve an optimal performance of the robot under high-speed operations. Some control schemes rely directly on a dynamic model to computer actuator torques and forces required to follow a desired trajectory.
- The dynamical model can be used for computer simulation of a robotic system. By examining the behaviour of the model under various operating conditions, it is possible to predict how a robotic system will behave when it will be built.

- The dynamic analysis of a robot gives all the joint reaction forces and moments needed for the design and sizing of links, bearings, and actuators.

1.4 SCOPE OF THE THESIS

In this thesis, focus has been laid on the development of dynamic model of serial manipulators. In most of the conventional text available, dynamic model of two, three or maximum four-links are available. Calculating the dynamic model of a general robotic manipulator, which is extremely nonlinear in nature, is a challenging task. In this thesis, a complete code in MATLAB environment is developed, which can generate dynamic model of any general serial manipulator having arbitrary number of links.

The second challenge lies in the proper selection of actuator at each joint. The choice of correct actuator depends upon number of factors, eg: its position in the kinematic chain, its orientation, weight and the cost of the motor. In this thesis, with the help of static force analysis (performed in Solid Works Environment) of a redundant manipulator, a look-up table is developed, which can help in selecting the optimal actuator. Optimal selection means the motor with minimum weight that can satisfies the torque requirements even in worst configurations and not to choose the motor, which can produce very high torque but at the cost of higher weight.

1.5 ORGANISATION OF THE THESIS

This thesis is organised as follows.

In Chapter 2, the literature work that has been done so far in this area discussed. Different challenges faced by authors in development of dynamic modeling of serial manipulator and its static force analysis. Approaches opted by the researcher to solve the different, are also discussed.

In Chapter 3, Dynamic modeling of the serial manipulator and it's different approaches is discussed. Based on that approachs, a code is developed in MATLAB environment. This code has no such limitation in terms of Degree-of-freedom. Further, code is validated through varies examples from published literatures and software developed at IIT-Delhi.

In Chapter 4, developed MATLAB code is used to generate the dynamic model for three case studies. And a lookup table is developed for the appropriate selection of actuators. Further static force analysis is done in Solid Works environment for the calculation of stress, strain and deflection with and without actuator weight compensation.

Finally, in Chapter, the conclusions are drawn and future directions are out.

CHAPTER 2 LITERATURE SURVEY

2.1 INTRODUCTION

This chapter covers the literature on the work done so far in the field of Static force analysis of serial manipulators, dynamic modeling and different approaches used so far. Different aspects shown by researches to solve the dynamic models and how important is to solve.

2.2 LITERATURE REVIEW

S.K. Saha et. al [10], Simplicity in the dynamics model of a serial robot manipulator greatly enhances the speed of its control and the associated hardware implementation. Since the motion of one link influences the torque or force required at the other joints, the control becomes difficult. This is referred as dynamic coupling. In this paper, it is proposed to simplify the robot's dynamic coupling by suitably choosing the manipulator's kinematic and dynamic parameters. The intention is to make the Generalized Inertia Matrix (GIM) of the serial manipulator associated with its dynamic equations of motion diagonal and/or constant. Such choice automatically ensures the associated convective inertia terms vanish. Such simplifications are carried out by investigating the expression of each element of the GIM. The concepts of the twist propagation matrices and the joint motion propagation vectors are used to obtain the analytical expressions of the GIM elements that allow one to investigate the elements for simplifications. The methodology is illustrated with a 3-link spatial manipulator arm.

G. Zeng, et. al [11], This paper reports on the existing robot force control algorithms. The objective is to provide a pragmatic exposition with speciality on their differences and different application conditions, and to give a guide of the existing robot force control algorithms. The previous work can be categorized into discussion , design and / or application of fundamental force control techniques , stability analysis of the various control algorithms , and the advanced methods . Advanced methods combine the fundamental force control techniques with advanced control algorithms such as adaptive , robust and learning control strategies .

A. Singla et. al.[12] This paper presents some guidelines for analyzing, manipulator parameters, resulting from an optimal design problem. An automated procedure for the static force analysis of the synthesized manipulators is discussed and illustrated through the case study of an 8 link manipulator. The methodology includes a strategy to compute an approximation for the maximum torque required at each joint and also for the weight estimation of the motor at each joint. The underlying idea to complete the data required for the modeling and analysis through professional software and to examine the results for the prescribed width of the links.

K. Mahindra. et. al.[13] This paper discusses a methodology and an algorithm for the analysis of dynamics of bio-mechanical systems, and in particular of optimal movement patterns between initial and target configurations. The basic formulation utilizes a finite element time discretization and establishes a large set of equations in displacements and forces. These are solved simultaneously for the whole time interval considered. The algorithm allows different optimization criteria for the movement, based on either the smoothness of the movement or the minimization of needed controls or control rates. It is primarily aimed at musculoskeletal simulations with either the joint resultant moments or the redundant muscular tensions as unknowns. Kinetic and kinematic constraints can be introduced for the obtained movement. Examples show that the obtained results are strongly dependent on the optimality criterion used. Systematic usage of the algorithm can improve knowledge about optimal motion planning.

Yang Zhao, Zheng Feng[14] A computational methodology for analysis of space robot manipulator systems, considering the effects of the clearances in the joint, is presented. The contact dynamics model in joint clearance is established using the nonlinear equivalent spring-damp model and the friction effect is considered using the Coulomb friction model. The space robot system dynamic equation of manipulator with clearance is established. Then the dynamics simulation is presented and the dynamics characteristics of robot manipulator with clearance are analyzed. This work provides a practical method to analyze the dynamics characteristics of space robot manipulator with joint clearance and improves the engineering application. The computational methodology can predict the effects of clearance on space robot manipulator preferably, which is the basis of space robot manipulator design, precision analysis and ground test.

Shin-Min Song [15] The computational efficiency of inverse dynamics of a manipulator is important to the real-time control of the system. For serial manipulators, the recursive Newton-Euler method has been proven to be the most efficient. However, for more general manipulators, such as serial manipulators with closed kinematic loops or parallel manipulators, it must be modified accordingly and the resultant computational efficiency is degraded. This article presents a computationally efficient scheme based on the virtual work principle for inverse dynamics of general manipulators. The present method uses a forward recursive scheme to compute velocities and accelerations, the Newton-Euler equation to calculate inertia forces/torque, and the virtual work principle to formulate the dynamic equations of motion. This method is equally effective for serial and parallel manipulators. For serial manipulators, its computational efficiency is comparable to the recursive Newton-Euler method. For parallel manipulators or serial manipulators with closed kinematic loops, it is more efficient than the existing methods

R. Tapia Herrera et. al. [16] In this paper, they presented method, based on dual-number representation, has demonstrated be a powerful tool for solving a great variety of problems, that imply motions simultaneity off rotation and translation of rigid bodies in the space; the aforementioned, allows establishing dual rotation matrices. Robotics is a field wherein dual numbers have been employed to describe the motion of a rigid body, in particular of serial robotic arms. The methodology proposed is useful for robotic arms with cylindrical, prismatic and rotational joints. Once established the dual angles θ and α , if the dual part of θ is zero, the mechanism has only revolute joints, otherwise if the primary part of θ is zero, only exist prismatic joints. So the developed methodology can be generalized to different topologies, which is a great advantage that allows that only one program solves a great variety of topologies.

S.F. Chan, R. Kwan [17], shows that Post-processors can be system- or application-dependent. Some of these post-processors have been designed for use with an application-dependent robot simulator. They proposed two post-processing methodologies for off-line robot programming, namely the Hierarchical "Top-Down" and "Bottom-Up" approaches which appropriately process data in-line with data structure adopted in CAD model. It would be logical to develop the processor in two modules, pre-processor and post-processor which would substantially improve the efficiency. The successful experience gained proved the validity of the approaches and indicated that similar effort can be applied to different systems.

Dae-Jin Kim et. al [18] ; In his paper, they have presented a three-week user study with UCF-MANUS to help individuals post-SCI to perform pick-and-place tasks with six items on bi-level shelves. Based on the pre-evaluation assessment by OTs, none of the ten subjects could perform grasping tasks without help from a caregiver. After a three-week training/testing period, subjects were able to manipulate the robotic arm to perform ADL tasks with speed and command efficiency. Compared with manual (Cartesian) control mode, Auto mode was seen to enable the users to perform the given tasks faster and with lesser effort, however, the manual mode operation was perceived to be better by the users. For both manual and auto modes, users felt that UCF-MANUS would improve their functional abilities, quality of life and overall well being. During semi-structured exit interviews, users felt that they could benefit from the robot's autonomy, however, they indicated satisfaction with being in charge during the interactive manual mode operation. It can therefore be concluded that the autonomy provided by UCF-MANUS or any other assistive technology needs to be appropriately channeled so that user satisfaction can be enhanced at the same time as their objective performance. Since there is great variability in performance of populations with disabilities as compared to healthy individuals, flexible interfaces need to be designed that are capable of providing a tailored amount of feedback to the user based on an estimate of the specific bottlenecks in their performance.

Ramesh Kolluru et al [19]; They have given design and modeling fundamentals of a reconfigurable robotic gripper. The design of the system has been analytically validated for static and dynamic behaviour that the RGS may be subjected to in normal operational conditions. The design of the system has been proven mechanically robust and stable. Further validation of the system design has been provided by the use of I-DEAS simulation software. A fully validated design for the RGS mechanism has been derived. The overall reconfigurable gripper system design has been proven kinematically and dynamically robust, indicating that the gripper, once developed, will be capable of reliable manipulation of limp material. The analytical study has resulted in defining parameters for the fabrication of a reconfigurable gripper, currently under development.

Denny Oetomo, et. al. [20]; Presented , a certified workspace evaluation algorithm and a serial-chain design algorithm were proposed and presented. In the process, an effective interval inverse kinematics algorithm for serial chain, without explicit inverse kinematic expressions, was also proposed and presented. These algorithms are shown to be effective in certifying that the various design constraints, given in the form of equality or inequality constraints, are satisfied. The proposed algorithm is also demonstrated to be effective in obtaining all variations of the kinematic topology of a serial manipulator such that the given constraints are satisfied in all points within the desired workspace. Further work is required to improve the efficiency of the algorithm in admitting or rejecting the various interval boxes, such as by exploring an efficient interval representation of serial kinematic transformations, hence reducing the necessity for further bisections in the process. As this study was carried out under European Union project ARES on endoluminal surgery through reconfigurable modular robots, future work also includes improving and adapting the proposed strategy to the challenging biomedical environment and requirements to compute the most suitable topology for the task

Christiaan J.J. et.al.[21]; In his article, they have shown that making a manipulator fault tolerant by adding redundant DOFs is an effective way to increase the reliability of a manipulator. However, not every redundant manipulator is fault tolerant. Thus, an important problem for the design of fault tolerant manipulators is: How many DOFs are necessary and sufficient for fault tolerance and how should these DOFs be distributed along the length of the manipulator? They have shown that, depending on the assumptions that are made about the task, the answer to this question varies. For general purpose fault tolerant manipulators without joint limits, two degrees-of-redundancy are necessary and sufficient to sustain one fault. This conclusion was illustrated with two spatial general purpose fault tolerant manipulator designs: a 5-DOF positional manipulator and an 8-DOF positional and orientational manipulator. Both manipulators have a fault tolerant workspace without any holes or voids so that one can scale the designs to fit any task. For task specific fault tolerant manipulators, only one degree-of-redundancy is necessary and sufficient for 1-fault tolerance. However, one might have to use a different manipulator and recompute the fault tolerant joint intervals, for every task. This drawback can be partially overcome by using a modular manipulator system that can be quickly reconfigured to suit a particular task.

2.3 OBSERVATIONS FROM LITERATURE REVIEW

Survey of the previous works on dynamics modeling and static force analysis of serial manipulators is presented in this chapter. The following key observations can be made are:

- The dynamic modeling of serial manipulators is a very challenging task. The problem becomes more challenging with increase in number of links.
- Another challenge is the degrees-of-freedom of a serial manipulator. Published literature is available for maximum 4, 5 or 6 link manipulators. Not much significant work is found in the literature for any general serial manipulators, with no limitation on degrees-of-freedom.
- Problem becomes more complex, when actuator weight compensation is added to system. Actuator weight has important role in modeling, which is more closer to realistic world.
- Choice of the right actuator becomes more important. As more powered motors/actuators add unnecessary load on the manipulator system, and lesser powered motors can fail the serial manipulator.

Chapter 3 DYNAMIC MODELING OF SERIAL ROBOTIC MANIPULATORS

3.1 INTRODUCTION

In this Chapter, Dynamic modeling of a serial manipulators is discussed. Different modeling techniques – Newton-Euler approach and Euler-Lagrange approach, are also discussed to generate the dynamic model. The first one is based on energy conservation and later one is based on force-moment balance.

Further, the developed dynamic model is validated with various solved problems of two-link manipulator from published literatures. As, it is difficult to find the solution for more than four degrees-of-freedom in published literature. A software named as "RoboAnalyzer6" is used for the validation of code for six-link serial manipulator.

3.2 KINEMATIC DESCRIPTION OF A SERIAL MANIPULATOR

Manipulators are made up of links and joints. They have degree-of-freedom in between each link which depends upon the type of joint. The links of manipulators are connected by joints allowing either rotational (such as in an articulated robot) or translational (linear displacement). The links of the manipulator are connected in a manner to form an open kinematic chain. The end of the kinematic chain of the manipulator is called the end effector, which is analogous to the human hand. The end effector can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application.

The links and joints are completely defined by the use of Denavit-Hartenberg (DH) parameters. These parameters are divided into two categories:

- Link parameters
- Joint parameters

Link parameters defines the link itself and Joint parameters defines the connection of link to its neighboring next link, as shown in Fig. 1.6, Link *'i-1'* is defined between axis '*i-1'* and axis *'i'* and it is connected to link *'i'* at axis *'i'*.

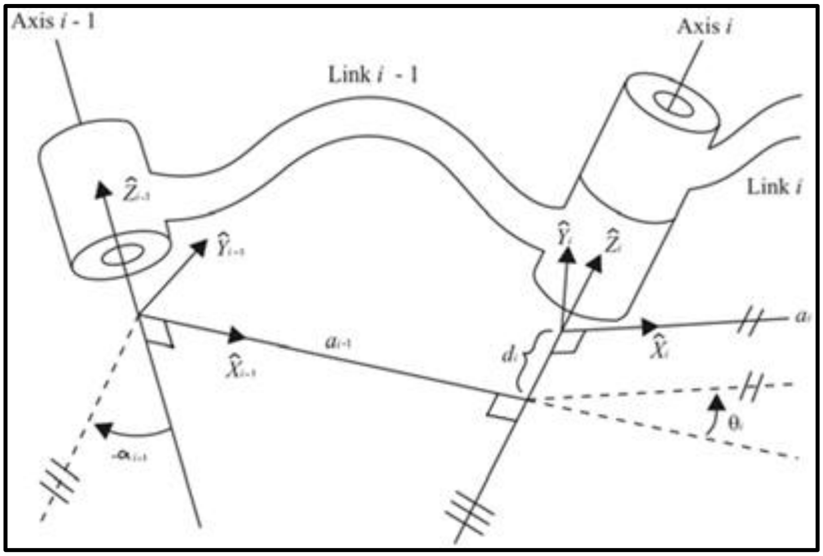

Fig. 3.1: Schematic representation of DH Parameters [6].

Link Parameters:

- Link Length (a_i) is the distance from Z_i to Z_{i+1} measured along X_i
- Link Twist (a_i) is the angle between Z_i to Z_{i+1} measured about X_i

Joint Parameters:

- Joint Offset (d_i) is the distance from X_{i-1} to X_i measured along Z_i
- Joint Angle (θ_i) is the angle between X_{i-1} to X_i measured about Z_i

In case of revolute joint, only Joint Angle (θ_i) is variable and other parameters remain fixed. And for prismatic joint, only Joint Offset (d_i) is variable and other parameters remain fixed.

3.3 DYNAMIC MODELING TECHNIQUES

The dynamic equations of motion can be formulated by several methods.

One approach, named as Newton-Euler Approach, is based on the application of Newton and Euler laws for linear and rotational motions, respectively. Writing Newton's and Euler's equations of motion for each link of the robot results in a system of equations that contains both the applied forces and the forces of constraints. These equations can then be solved simultaneously for all the forces, including those due to the constraints which do not contribute to the motion of the links but are required for link design.

Another approach, known as Euler-Lagrange approach, is based on energy principles. The advantage of using the Lagrange approach is that it eliminates the forces of constraint from the dynamic equations of motion if the generalized coordinates are independently chosen. The elimination makes it suitable for motion control and simulation. However, these eliminated constraint forces can be recovered using Lagrange multiplies, if they are to be used for the purpose of design.

Euler-Lagrange approach treats each variable as a scalar quantity, where as in Newton-Euler approach, every variable is treated as a vector quantity.

3.3.1 Newton-Euler Approach

Each link of a manipulator is considered as a rigid body. If the location of the center of mass and the inertia tensor of the link are known, then its mass distribution is completely characterized. In order to move the links, accelerate and decelerate them. The forces required for such motion are a function of the acceleration desired and of the mass distribution of the links. Newton's equation, along with its rotational analog, Euler's equation, describes how forces and inertia related to acceleration.

Newton's equation

A rigid body of mass *m* is considered, whose center of mass is accelerating with $\dot{v}_C$ as shown in Fig. 3.2. In such a situation, the force ***F***, acting at the center of mass and causing this acceleration is given by Newton's equation.

$$\boldsymbol{F} = m\dot{v}_C \tag{3.1}$$

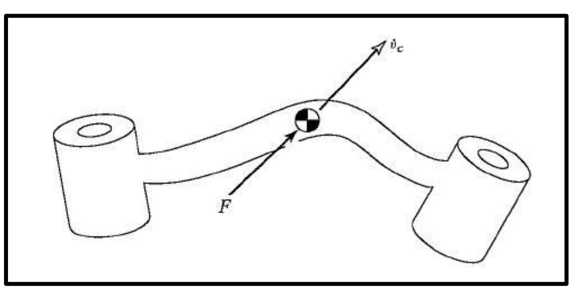

Fig. 3.2: Force-moment balance shown on i^th link [7].

Euler's equation

A rigid body rotating with angular velocity ω and with angular acceleration $\dot{\omega}$. In such a situation, the moment $\boldsymbol{N}$, which must be acting on the body to cause this motion, is given by Euler's equation.

$$\boldsymbol{N} = {}^{C}\boldsymbol{I}\dot{\omega} + \omega \times {}^{C}\boldsymbol{I}\dot{\omega} \tag{3.2}$$

where ${}^{C}\boldsymbol{I}$ is the inertia tensor of the body written in a frame, {*C*}, whose origin is located at the center of mass.

Iterative Newton - Euler Dynamic Formulation

In this section, the problem of inverse dynamics of a serial manipulator is presented. Inverse dynamics means to compute the joint torques correspond to a given trajectory of a manipulator. Given the position, velocity and acceleration of the joints, $(\theta, \dot{\theta}, \ddot{\theta})$, the problem of computing joint torques is known as Inverse Dynamics. With knowledge of the kinematics and the mass-distribution information of the robot, calculate the joint torques, required to cause this motion, can be calculated.

Outward Iterations to Compute Velocities and Accelerations

In order to compute inertial forces acting on the links, it is necessary to compute the rotational velocity and linear and rotational acceleration of the center of mass of each link of the manipulator at any given instant. These computations are done in an iterative way, starting with link *1* and moving successively, link by link, outward to link *n*.

The "propagation" of rotational velocity from link to link is given (for joint *i + 1* rotational) by

$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}\boldsymbol{R}\,^{i}\omega_i + \dot{\theta}_{i+1}\,^{i+1}\hat{\boldsymbol{Z}}_{i+1} \tag{3.3}$$

The equation for transforming angular acceleration from one link to the next.

$$^{i+1}\dot{\omega}_{i+1} = {}^{i+1}_{i}\boldsymbol{R}\,^{i}\dot{\omega}_i + {}^{i+1}_{i}\boldsymbol{R}\,^{i}\omega_{i1} \times \dot{\theta}_{i+1}\,^{i+1}\hat{\boldsymbol{Z}}_{i+1} + \ddot{\theta}_{i+1}\,^{i+1}\hat{\boldsymbol{Z}}_{i+1} \tag{3.4}$$

When joint $i+\ 1$ is prismatic, this simplifies to

$$^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}_{i}\boldsymbol{R}\,{}^{i}\boldsymbol{\omega}_{i} \tag{3.5}$$

The linear acceleration of each link – frame origin is

$$^{i+1}\dot{\boldsymbol{v}}_{i+1} = {}^{i+1}_{i}\boldsymbol{R}\left({}^{i}\dot{\boldsymbol{\omega}}_{i} \times {}^{i}\boldsymbol{P}_{i+1} + {}^{i}\boldsymbol{\omega}_{i} \times \left({}^{i}\boldsymbol{\omega}_{i} \times {}^{i}\boldsymbol{P}_{i+1}\right) + {}^{i}\dot{\boldsymbol{v}}_{i}\right) + 2\,{}^{i+1}\boldsymbol{\omega}_{i+1} \times \dot{d}_{i+1}\,{}^{i+1}\widehat{\boldsymbol{Z}}_{i+1} + \ddot{d}_{i+1}\,{}^{i+1}\widehat{\boldsymbol{Z}}_{i+1} \tag{3.6}$$

The linear acceleration of the center of mass of each link, which also can be found by applying

$$^{i}\dot{\boldsymbol{v}}_{C_i} = {}^{i}\dot{\boldsymbol{\omega}}_{i} \times {}^{i}\boldsymbol{P}_{C_i} + {}^{i}\boldsymbol{\omega}_{i} \times \left({}^{i}\boldsymbol{\omega}_{i} + {}^{i}\boldsymbol{P}_{C_i}\right) + {}^{i}\dot{\boldsymbol{v}}_{i}, \tag{3.7}$$

Here, imagine a frame, $\{C_i\}$, attached to each link, having its origin located at the center of mass of the link and having the same orientation as the link frame, $\{i\}$. Equation doesn't involve joint motion at all and so is valid for joint $i + 1$, regardless of whether it is revolute or prismatic.

The application of the equations to link 1 is especially simple, because ${}^{0}\boldsymbol{\omega}_{0} = {}^{0}\dot{\boldsymbol{\omega}}_{0} = 0$

The Force and Torque Acting an a Link

Having computed the linear and angular accelerations of the mass center of each link, apply the Newton—Euler equations to compute the inertial force and torque acting at the center of mass of each link.

Thus,

$$\boldsymbol{F}_i = m\,\dot{\boldsymbol{v}}_{C_i}, \tag{3.8}$$

$$\boldsymbol{N}_i = {}^{C_i}I\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times {}^{C_i}I\boldsymbol{\omega}_i, \tag{3.9}$$

where (C_i) has its origin at the center of mass of the link and has the same orientation as the link frame, $\{i\}$.

Inward iterations to compute forces and torques

Having computed the forces and torques acting on each link, now need to calculate the joint torques that will result in these net forces and torques being applied to each link.

It can be done by writing a force-balance and moment-balance equation based on a free-body diagram of a typical link. Each link has forces and torques exerted on it by its neighbors and in addition experiences an inertial force and torque.

$\boldsymbol{f_i}$ = force exerted on link i by link $i - 1$,

$\boldsymbol{n_i}$ = torque exerted on link i by link $i - 1$.

By summing the forces acting on link i, force-balance relationship:

$$ {}^{i}\boldsymbol{F}_i = {}^{i}\boldsymbol{f}_i - {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{f}_{i+1}. \tag{3.10} $$

By summing torques about the center of mass and setting them equal to zero, the torque-balance equation is:

$$ {}^{i}\boldsymbol{N}_i = {}^{i}\boldsymbol{n}_i - {}^{i}\boldsymbol{n}_{i+1} + \left(-{}^{i}\boldsymbol{P}_{C_i}\right) \times {}^{i}\boldsymbol{f}_i - \left({}^{i}\boldsymbol{P}_{i+1} - {}^{i}\boldsymbol{P}_{C_i}\right) \times {}^{i}\boldsymbol{f}_{i+1}. \tag{3.11} $$

Using the result from the force-balance relation and adding a few rotation matrices, above equation can be written as:

$$ {}^{i}\boldsymbol{N}_i = {}^{i}\boldsymbol{n}_i - {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{n}_{i+1} - {}^{i}\boldsymbol{P}_{C_i} \times {}^{i}\boldsymbol{F}_i - {}^{i}\boldsymbol{P}_{i+1} \times {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{f}_{i+1}. \tag{3.12} $$

Finally, by rearranging the force and torque equations so that they appear as iterative relationships:

$$ {}^{i}\boldsymbol{f}_i = {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{f}_{i+1} + {}^{i}\boldsymbol{F}_{i,} \tag{3.13} $$

$$ {}^{i}\boldsymbol{n}_i = {}^{i}\boldsymbol{N}_i + {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{n}_{i+1} + {}^{i}\boldsymbol{P}_{C_i} \times {}^{i}\boldsymbol{F}_i + {}^{i}\boldsymbol{P}_{i+1} \times {}^{i}_{i+1}\boldsymbol{R}\, {}^{i+1}\boldsymbol{F}_{i+1}. \tag{3.14} $$

These equations are evaluated link by link, starting from link *n* and working inward toward the base of the robot.

As in the static case, the requires joint torques are found by taking the $\hat{Z}$ component of the torque applied by one link on its neighbor:

$$\tau_i = {}^i\mathbf{n}_i^T \, {}^i\hat{Z}_i. \tag{3.15}$$

For joint *i* prismatic,

$$F_i = {}^i f_i^T \, {}^i\hat{Z}_i. \tag{3.16}$$

where F_i symbol is used for a linear actuator force.

For a robot moving in free space, ${}^{N+1}f_{N+1}$ and ${}^{N+1}n_{N+1}$ are set equal to zero, and so the first application of the equations for link *n* is very simple. If the robot is in contact with the environment, the forces and torques due to this contact can be included in the force balance by having nonzero ${}^{N+1}f_{N+1}$ and ${}^{N+1}n_{N+1}$.

The iterative Newton – Euler dynamics algorithm

The complete algorithm for computing joint torques from the motion of the joints is composed of two parts.

First, link velocities and accelerations are iteratively computed from link 1 out -to-link *n* and the Newton–Euler equations are applied to each link.

Second, forces and torques of interaction and joint actuator torques are computed recursively from link *n* back to link 1

3.3.2 Euler Lagrange Approach

The general motion equations of a manipulator can conveniently be expressed thought the direct application of the Euler-Lagrange formulation of non-conservative systems. Many investigator utilize the Denavit – Hartenberg matrix representation to describe the spatial displacement between the neighboring link coordinate frames to obtain the link kinematic information, and they employ the Lagrange dynamics technique to derive the dynamic equations of a manipulator. The direct application of the Lagrange dynamics formulation, together with the Denavit–Hartenberg link coordinate representation, results in a convenient and compact alogorithmic description of the manipulator equations of motion. The algorithm is expressed by matrix operations and facilitates both analysis and computer implementation. The evaluation of the dynamic and control equations in functionally explicit terms will be based on the compact matrix algorithm presented in this section.

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

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

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right)-\frac{\partial \mathcal{L}}{\partial \dot{q}_i}=\tau_i \qquad \forall \quad i=1,\ 2\ ,\dots,n \tag{3.17}$$

$\mathcal{L}$ = Lagrangian function = kinetic energy (*KE*) – potential energy (*PE*)

KE = total kinetic energy of the robot arm

PE = total potential energy of the robot arm

q_i = generalized coordinates of the robot arm

$\dot{q}_i$ = first time derivative of the generalized coordinate, q_i

τ_i = generalized force (or torque) applied 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. Generalized coordinates are used as a convenient set of coordinates which completely describe the location (position and orientation) of a system with respect to a reference coordinate frame. For a simple manipulator with rotary-prismatic joints, various sets of generalized coordinates are available to describe the manipulator. However, since the angular positions of the joints are readily available because they can be measured by potentiometers or encoders or other sensing devices, they provide a natural correspondence with the generalized coordinates. This, in effect, corresponds to the generalized coordinates with the joint variable defined in each of the 4 × 4 link coordinate transformation matrices. Thus, in the case of a rotary joint, q_i is the joint angle span of the joint; whereas for a prismatic joint, q_i is , the distance traveled by the joint.

Joint Velocities of a Robot Manipulator

The Euler-Lagrange formulation requires knowledge of the kinetic energy of the physical system, which in turn requires knowledge of the velocity of each joint. 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 can be explored.

Let ${}^{i}\boldsymbol{r}_i$ 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,

$$ {}^{i}\boldsymbol{r}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = (x_i, y_i, z_i, 1)^T \tag{3.18} $$

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

$$ {}^{0}\boldsymbol{r}_i = {}^{0}\boldsymbol{A}_i \, {}^{i}\boldsymbol{r}_i \tag{3.19} $$

Where, ${}^{0}\boldsymbol{A}_i = {}^{0}\boldsymbol{A}_1 \, {}^{1}\boldsymbol{A}_2 \ldots {}^{i-1}\boldsymbol{A}_i$

If joint i is revolute, it follows that the general form[9] of ${}^{i-1}\boldsymbol{A}_i$ is given by

$$ {}^{i-1}\boldsymbol{A}_i = \begin{bmatrix} cos\theta_i & -cos\alpha_i \, sin\theta_i & sin\alpha_i \, sin\theta_i & a_i cos\theta_i \\ sin\theta_i & cos\alpha_i \, cos\theta_i & -sin\alpha_i \, cos\theta_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\,\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20} $$

or, if joint i is prismatic, the general form of ${}^{i-1}\boldsymbol{A}_i$ is

$$ {}^{i-1}\boldsymbol{A}_i = \begin{bmatrix} cos\theta_i & -cos\alpha_i \, sin\theta_i & sin\alpha_i \, sin\theta_i & 0 \\ sin\theta_i & cos\alpha_i \, cos\theta_i & -sin\alpha_i \, cos\theta_i & 0 \\ 0 & sin\alpha_i & cos\,\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21} $$

In general, all the nonzero elements in the matrix ${}^{0}\boldsymbol{A}_i$ are a function of link parameters and joint variables. In order to derive the equations of motion that are application to both revolute and prismatic joints, use the variable q_i to represent the generalized coordinate of joint i which is either q_i (for a rotary joint) or d_i (for a prismatic joint).

Since the point ${}^{i}\boldsymbol{r}_i$ is at rest in link i, and assuming rigid body motion, other points as well as the point ${}^{i}\boldsymbol{r}_i$ fixed in the link i and expressed with respect to the i^{th} coordinate frame will have zero velocity with respect to the i^{th} coordinate frame (which is not an inertial frame). The velocity of ${}^{i}\boldsymbol{r}_i$ expressed in the base coordinate frame (which is an inertial frame) can be expressed as

$$ {}^0\boldsymbol{v}_i = \boldsymbol{v}_i = \frac{d}{dt}\left({}^0\boldsymbol{r}_i\right) = \frac{d}{dt}\left({}^0\boldsymbol{A}_i {}^i\boldsymbol{r}_i\right) \tag{3.22} $$

$$ = {}^0\boldsymbol{A}_1 {}^1\boldsymbol{A}_2 \ldots {}^{i-1}\boldsymbol{A}_i {}^i\boldsymbol{r}_i + {}^0\boldsymbol{A}_1 {}^i\dot{\boldsymbol{A}}_2 \ldots i^{i-1}\boldsymbol{A}_i {}^i\boldsymbol{r}_i + {}^0\boldsymbol{A}_1 \ldots {}^{i-1}\dot{\boldsymbol{A}}_i {}^i\boldsymbol{r}_i + {}^0\boldsymbol{A}_i {}^i\dot{\boldsymbol{r}}_i = \left(\sum_{j=1}^{i} \frac{\partial^0\boldsymbol{A}_i}{\partial q_j}\dot{q}_j\right){}^i\boldsymbol{r}_i \tag{3.23} $$

The above compact form is obtained because ${}^i\dot{\boldsymbol{r}}_i = 0$. The partial derivative of ${}^0\boldsymbol{A}_{\mathrm{i}}$ with respect to q_{j} can be easily calculated with the help of a matrix $\boldsymbol{P}_{\mathrm{i}}$ (Permutation Matrix) which for a revolute joint, is defined as

$$ \boldsymbol{P}_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.24} $$

and, for a prismatic joint, as

$$ \boldsymbol{P}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.25} $$

It then follows that

$$ \frac{\partial^{i-1}\boldsymbol{A}_i}{\partial q_i} = \boldsymbol{P}_i \quad {}^{i-1}\boldsymbol{A}_i \tag{3.26} $$

Eq.(3.26) can be interpreted as the effect of the motion of joint j on all the points on link i. In order to simplify notations, $\boldsymbol{U}_{ij}$ can be defined as $\boldsymbol{U}_{ij} \overset{\Delta}{=} \partial^0\mathbf{A}_i / \partial q_j$, then Eq.(3.26) can be written as follows for $i = 1, 2 \ldots, n$,

$$ \boldsymbol{U}_{ij} = \begin{cases} {}^0\boldsymbol{A}_{j-1} \ \boldsymbol{P}_j {}^{j-1}\boldsymbol{A}_i & \text{for } \ j \le i \\ 0 & \text{for } \ j > i \end{cases} \tag{3.27} $$

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

$$\boldsymbol{v_i} = \left(\sum_{j=1}^{i} \boldsymbol{U}_{ij} \dot{q}_j \right) {}^{i}\boldsymbol{r}_i \tag{3.28}$$

It is worth pointing out that the partial derivative of ${}^{i-1}\boldsymbol{A}_i$ with respect to q_i results in a matrix that does not retain the structure of a homogeneous coordinate transformation matrix. For a rotary joint, the effect of premultiplying ${}^{i-1}\boldsymbol{A}_i$ by $\boldsymbol{P}_i$ is equivalent to interchanging the elements of the first two rows of ${}^{i-1}\boldsymbol{A}_i$, negating all the elements of the first row, and zeroing out all the elements of the third and fourth rows. For a prismatic joint, the effect is to replace the elements of the third and fourth rows. For a prismatic joint, the effect is to replace the elements of the third row with the fourth row of ${}^{i-1}\boldsymbol{A}_i$ and zeroing out the elements in the other rows. The advantage of using the $\boldsymbol{P}_i$ matrices is that ${}^{i-1}\boldsymbol{A}_i$ matrices can be used again and apply the above operations to ${}^{i-1}\boldsymbol{A}_i$ when premultiplying it with the $\boldsymbol{P}_i$.

Next, interaction effects [9] between joints can be calculated as :

$$\frac{\partial \boldsymbol{U}_{ij}}{\partial q_k} \stackrel{\Delta}{=} \boldsymbol{U}_{ijk} = \begin{cases} {}^{0}\boldsymbol{A}_{j-1} \boldsymbol{P}_j {}^{j-1}\boldsymbol{A}_{k-1} \boldsymbol{P}_k {}^{k-1}\boldsymbol{A}_i & i \geq k \geq j \\ {}^{0}\boldsymbol{A}_{k-1} \boldsymbol{P}_k {}^{k-1}\boldsymbol{A}_{j-l} \boldsymbol{P}_j {}^{j-1}\boldsymbol{A}_i & i \geq j \geq k \\ 0 & i < j \, or \, i < k \end{cases} \tag{3.29}$$

Eq. (3.29) can be interpreted as the interaction effects of the motion of joint *j* and joint *k* on all the points on link *i*.

Kinetic Energy of a Robot Manipulator

After obtaining the joint velocity of each link, the kinetic energy of link *i* can be calculated. Let K_i be the kinetic energy of link *i*, $i = 1, 2, \ldots\ldots, n$, 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

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

$$= \frac{1}{2}\ trace\ \left(\boldsymbol{v}_i \boldsymbol{v}_i^T\right)\ dm = \frac{1}{2} Tr\ \left(\boldsymbol{v}_i \boldsymbol{v}_i^T\right)\ dm \tag{3.30}$$

where a trace operator instead of a vector dot product is used in the above equation to form the tensor from which the link inertia matrix (or pseudo-inertia matrix) J_i can be obtained. Substituting $\boldsymbol{v}_i$ from Eq. (3.28), the kinetic energy of the differential mass is

$$dK_i = \frac{1}{2} Tr\left[\sum_{p=1}^{i} \boldsymbol{U}_{ip} \dot{q}_p {}^i\boldsymbol{r}_i \left(\sum_{r=1}^{i} \boldsymbol{R}_{ir} \dot{\boldsymbol{q}}_r {}^i\boldsymbol{r}_i\right)^T\right] dm$$

$$= \frac{1}{2}\ Tr\ \left[\sum_{p=1}^{i}\ \sum_{r=1}^{i} \boldsymbol{U}_{ip} {}^i\boldsymbol{r}_i {}^i\boldsymbol{r}_{ii}^T \boldsymbol{U}_{ir}^T \dot{\boldsymbol{q}}_p \dot{\boldsymbol{q}}_r\right] dm$$

$$= \frac{1}{2}\ Tr\ \left[\sum_{p=1}^{i}\ \sum_{r=1}^{i} \boldsymbol{U}_{ip} \left({}^i\boldsymbol{r}_i dm {}^i\boldsymbol{r}_{ii}^T\right) T_{ir}^T \dot{\boldsymbol{q}}_p \dot{\boldsymbol{q}}_r\right] \tag{3.31}$$

The matrix U_{ij} is the rate of change of the points (${}^i r_i$) on link i relative to the base coordinate frame as q_j changes. It is constant for all points on link i and independent of the mass distribution of the link i. Also $\dot{q}_i$ are independent of the mass distribution of link i, so summing all the kinetic energies of all links and putting the integral inside the bracket,

$$K_i = \int dK_i = \frac{1}{2} Tr\left[\sum_{p=1}^{i}\ \sum_{r=1}^{i} \boldsymbol{U}_{ip} \left(\int {}^i\boldsymbol{r}_i {}^i\boldsymbol{r}_i^T dm\right) \boldsymbol{U}_{ir}^T \dot{q}_p \dot{q}_r\right] \tag{3.32}$$

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

$$\boldsymbol{J}_i = \int {}^i\boldsymbol{r}_i {}^i\boldsymbol{r}_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i\ dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \tag{3.33}$$

where ${}^{i}\boldsymbol{r}_i = (x_i, y_i, z_i, 1)^T$ as defined before.

Inertia tensor I_{ij}, which is defined as

$$\boldsymbol{I}_{ij} = \int \left[\delta_{ij} \left(\sum_k x_k^2 \right) - x_i x_j \right] dm \tag{3.34}$$

where the *j, k* indicate principal axes of the i^{th} coordinate frame and I_{ij} is the so-called Kronecker delta, then J_i can be expressed in inertia tensor as

$$\boldsymbol{J}_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & m_i\bar{x}_i \\ I_{xy} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & m_i\bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & m_i\bar{z}_i \\ m_i\bar{x}_i & m_i\bar{y}_i & m_i\bar{z}_i & m_i \end{bmatrix} \tag{3.35}$$

Note that the $\boldsymbol{J}_i$ are dependent on the mass distribution of link *i* and not their position or rate of motion and are expressed with respect to the i^{th} coordinate frame. Hence, the $\boldsymbol{J}_i$ need be computed only once for evaluating the kinetic energy of a robot arm.

Potential Energy of a Robot Manipulator

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

$$PE_i = -m_i g^0 \bar{\boldsymbol{r}}_i = -m_i g \left({}^0\boldsymbol{A}_i {}^i\bar{\boldsymbol{r}}_i \right) \qquad i = 1, 2, ..., n \tag{3.36}$$

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

$$PE = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} -m_i \boldsymbol{g} \left({}^0\boldsymbol{A}_i {}^i\bar{\boldsymbol{r}}_i \right) \tag{3.37}$$

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

Motion Equations of a Manipulator

The Lagrangian function $\mathcal{L} = KE - PE$ is given by

$$\mathcal{L} = \tfrac{1}{2}\sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=1}^{i} \left[Tr\left(\boldsymbol{U}_{ij} \boldsymbol{J}_i \boldsymbol{U}_{ik}^T \right) \dot{q}_j \dot{q}_k \right] + \sum_{i=1}^{n} m_i g \left({}^0\boldsymbol{A}_i {}^i\bar{\boldsymbol{r}}_i \right) \tag{3.38}$$

Applying the Lagrange-Euler formulation to the Lagrangian function of the robot arm yields the necessary generalized torque τ_i for joint i actuator to drive the i^{th} link of the manipulator,

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

$$= \sum_{j=i}^{n} \sum_{k=1}^{j} Tr\left(\boldsymbol{U}_{jk} \boldsymbol{J}_j \boldsymbol{U}_{ji}^T \right) \ddot{q}_k + \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{m=1}^{j} Tr\left(\boldsymbol{U}_{jkm} \boldsymbol{J}_j \boldsymbol{U}_{ji}^T \right) \dot{q}_k \dot{q}_m - \sum_{j=i}^{n} m_j g \boldsymbol{U}_{ji} {}^j\bar{\boldsymbol{r}}_j \tag{3.39}$$

for $i = 1, 2, \ldots, n$.

The above equation can be expressed in a much simpler matrix notation form as

$$\tau_i = \sum_{k=1}^{n} \boldsymbol{D}_{ik} \ddot{q}_k + \sum_{k=1}^{n} \sum_{m=1}^{n} h_{ikm} \dot{q}_k \dot{q}_m + g_i \quad i = 1, 2, \ldots, n \tag{3.40}$$

or in a matrix form as

$$\boldsymbol{\tau}(t) = \boldsymbol{M}(\boldsymbol{q}(t))\, \ddot{\boldsymbol{q}}(t) + \boldsymbol{c}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) + \boldsymbol{g}(\boldsymbol{q}(t)) \tag{3.41}$$

where

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

$$\tau(t) = (\tau_1(t), \tau_2(t), \ldots \tau_n(t))^T \tag{3.42}$$

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

$$\mathbf{q}(t) = (q_1(t), q_2(t), \ldots, q_n(t))^T \tag{3.43}$$

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

$$\dot{\boldsymbol{q}}(t) = (\dot{q}_1(t), \dot{q}_2(t), \ldots, \dot{q}_n(t))^T \tag{3.44}$$

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

$$\ddot{\boldsymbol{q}}(t) = (\ddot{q}_1(t), \ddot{q}_2(t), \ldots, \ddot{q}_n(t))^T \tag{3.45}$$

M(q) = an n × n inertial acceleration-related symmetric matrix whose elements are

$$\mathbf{M}_{ik} = \sum_{j=\max(i,k)}^{n} Tr\left(\boldsymbol{U}_{jk}\boldsymbol{J}_j\boldsymbol{U}_{ji}^T\right) \qquad i,k = 1, 2, \ldots n \tag{3.46}$$

c (q, $\dot{q}$) = an n × 1 nonlinear coriolis and centrifugal force vector whose elements are

$$\mathbf{c}(q, \dot{q}) = (c_1, c_2, \ldots, c_n)^T \tag{3.47}$$

where

$$c_i = \sum_{k=1}^{n}\sum_{m=1}^{n} c_{ikm}\dot{q}_k\dot{q}_m \quad \forall \quad i = 1, 2, \ldots, n \tag{3.48}$$

and

$$\mathbf{c}_{ikm} = \sum_{j=\max(i,k,m)}^{n} Tr\left(\boldsymbol{U}_{jkm}\boldsymbol{J}_j\boldsymbol{U}_{ji}^T\right) \quad i, k, m = 1, 2 \ldots, n \tag{3.49}$$

g(q) = an n × 1 gravity loading force vector whose elements are

$$\boldsymbol{g}(q) = (g_1, g_2, \ldots, g_n)^T \tag{3.50}$$

where

$$g_i = \sum_{j=i}^{n}\left(-m_j g \boldsymbol{U}^j{}_{ji}\bar{\boldsymbol{r}}_j\right) \qquad i = 1, 2, \ldots, n \tag{3.51}$$

Total torque is given by:

$$\tau = \boldsymbol{M}\ \ddot{\boldsymbol{q}}(t) + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) \tag{3.52}$$

3.4 MODEL VALIDATION

A MATLAB code is developed to obtain dynamic model of a general serial manipulator for any number of Degree-of-freedom (DOF) using the Euler Lagrange approach, is validated with the two examples.

- Two-Link Serial Manipulator
- Six-Link Serial Manipulator

3.4.1 Case Study 1: Two-Link Serial Manipulator

In this section, the case study of a two-link manipulator is considered to validate the MATLAB code. The robot is made to follow a cycloidal trajectory. Using the inverse dynamics, the joint torques are calculated and compared with the corresponding values in published literature.

For the validation part, the robot parameters are considered as given in Table 3.1:

Table 3.1: Parameters for Two-link manipulator

Parameters	Symbols	Value	Units
Mass	m_1	1	Kg
Mass	m_2	1	Kg
Length	l_1	1	m
Length	l_2	1	M
Inertia	I_1, I_2	0	N-m

Moreover, the cycloidal trajectory, which the robot is supposed to follow can be represented as:

Joint angle:
$$\theta = \theta(0) + \frac{\theta(T) - \theta(0)}{T}\left[t - \frac{T}{2\pi}\sin\left(\frac{2\pi}{T}t\right)\right] \quad (3.53)$$

Angular velocity:
$$\dot{\theta} = \frac{\theta(T) - \theta(0)}{T}\left[1 - cos\left(\frac{2\pi}{T}t\right)\right] \quad (3.54)$$

Angular acceleration: $$\ddot{\theta} = \frac{\theta(T) - \theta(0)}{T}\left[\frac{2\pi}{T}\sin\left(\frac{2\pi}{T}t\right)\right] \quad (3.55)$$

Where $\theta(0)$ is the initial angle and $\theta(T)$ is the final angle, which varies in time T=10 secs, taking 10 steps in each second.

Angle θ for the first angle varies from 0° degree to 180° and 2nd angle varies from 0° to 90° in this given time T=10 secs.

Torque data comparison

In this test run, centre of mass of each link is kept at its end and other data as mentioned above. Results for joint torques for 1st and 2nd joint are shown in these Figs. 3.3 and 3.4.

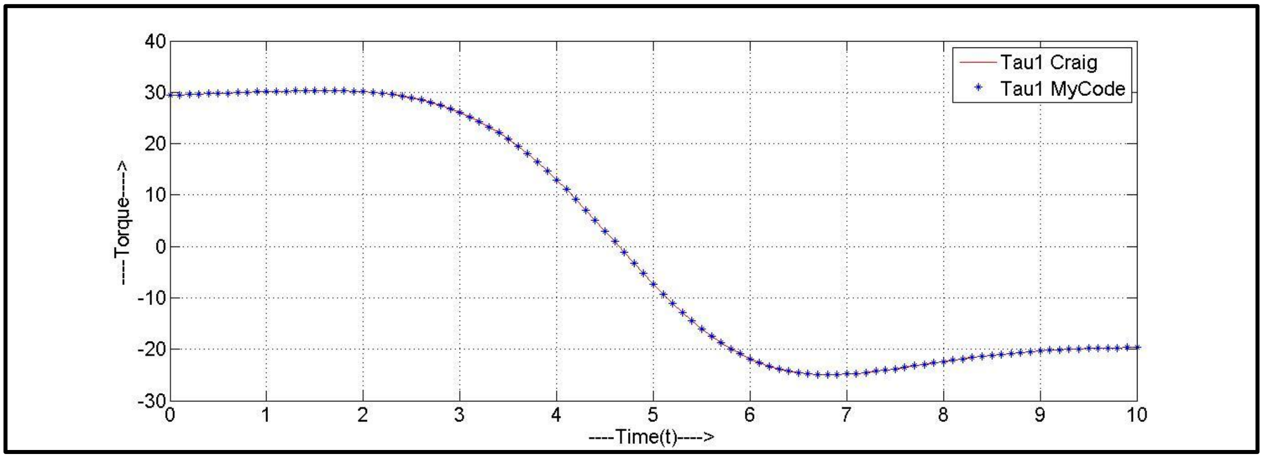

Fig. 3.3: Joint Torque at 1st Joint.

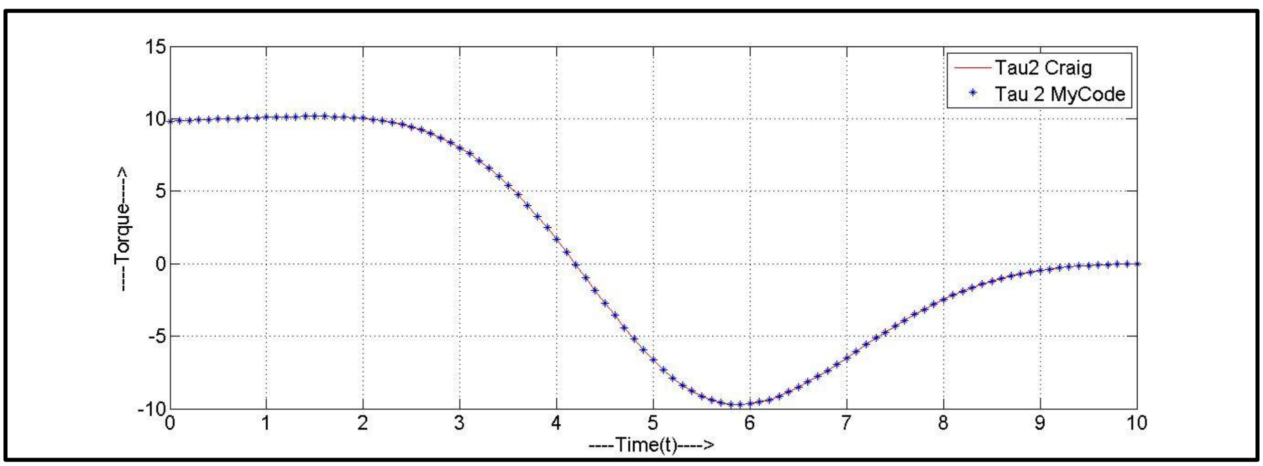

Fig. 3.4: Joint Torque at 2nd Joint.

In the next test run, the joint torques are calculated, while centre of mass of each link is kept at its centre and other data as mentioned above. Results for joint torques for 1st and 2nd joint are shown in Figs. 3.5 and 3.6, and compared with published literature [8]. It can be clearly seen from the figures that the results calculated by the MATLAB code are found in close agreement with that reported by Saha [8]. This process validated the dynamic model and authenticates the code developed to use for more complicated cases.

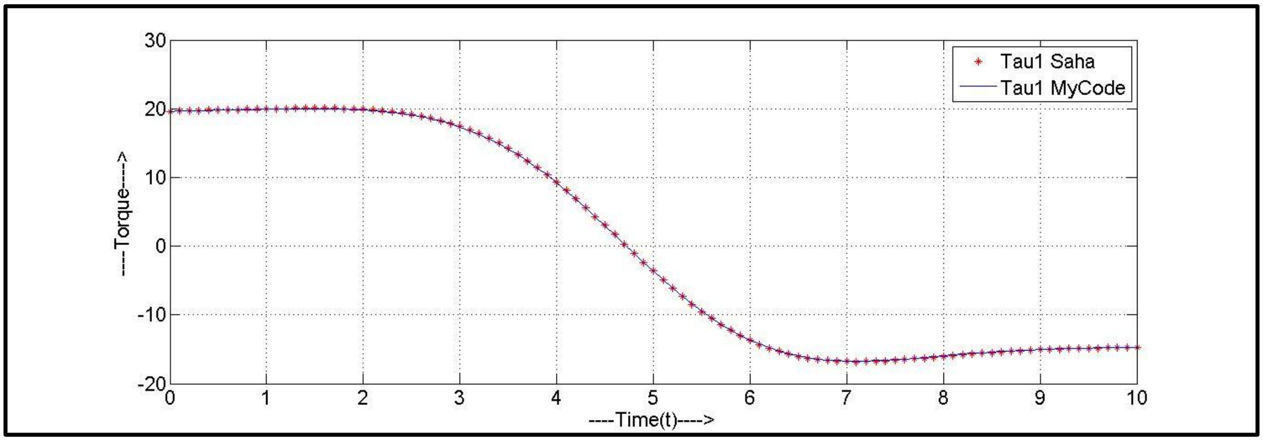

Fig. 3.5: Joint Torque at 1st Joint.

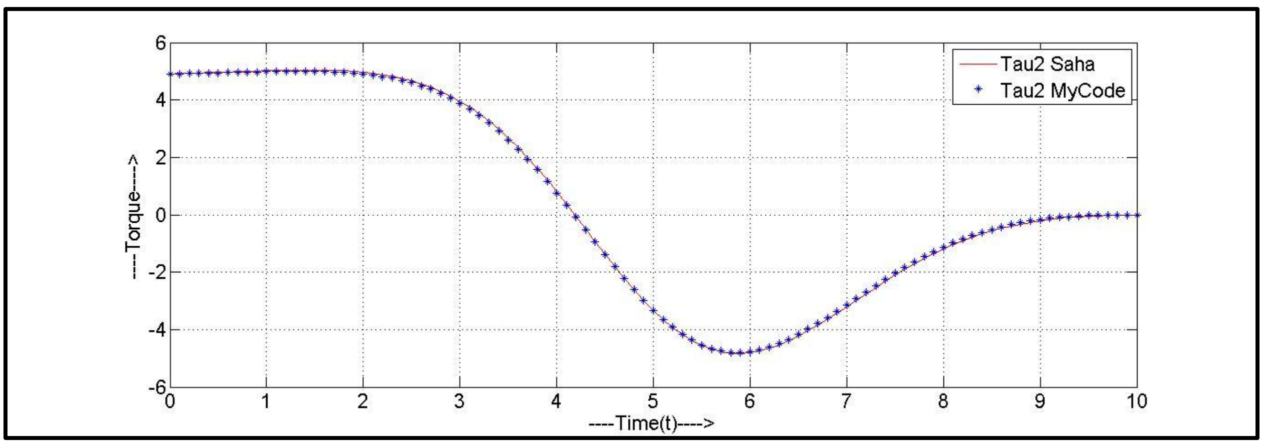

Fig. 3.6: Joint Torque at 2nd Joint.

To draw better comparison, the joint torques plotted by three different approaches are compared in Figs. 3.7 and 3.8., which clearly demonstrates the accuracy of MATLAB code developed.

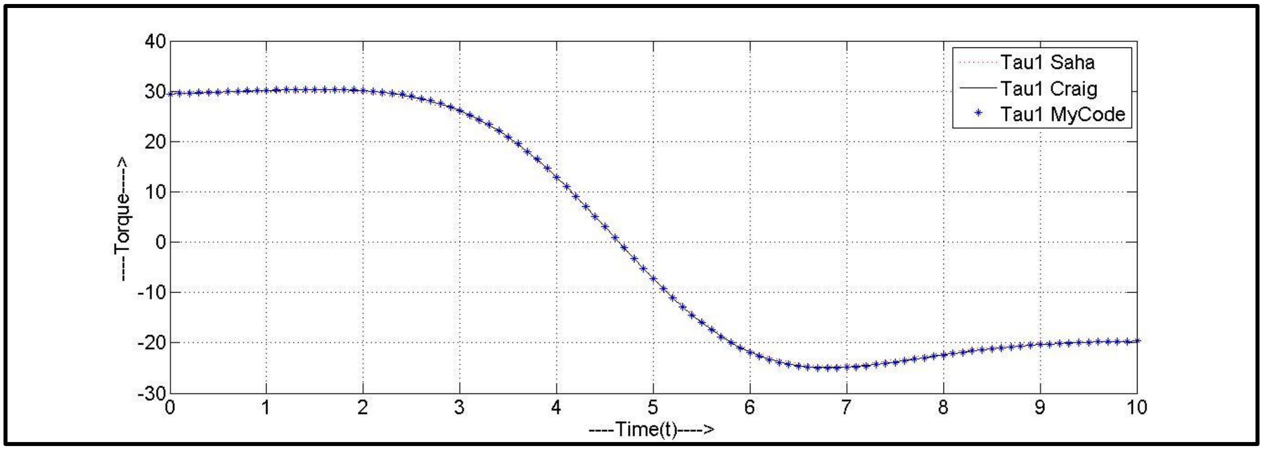

Fig. 3.7: Joint Torque at 1st Joint.

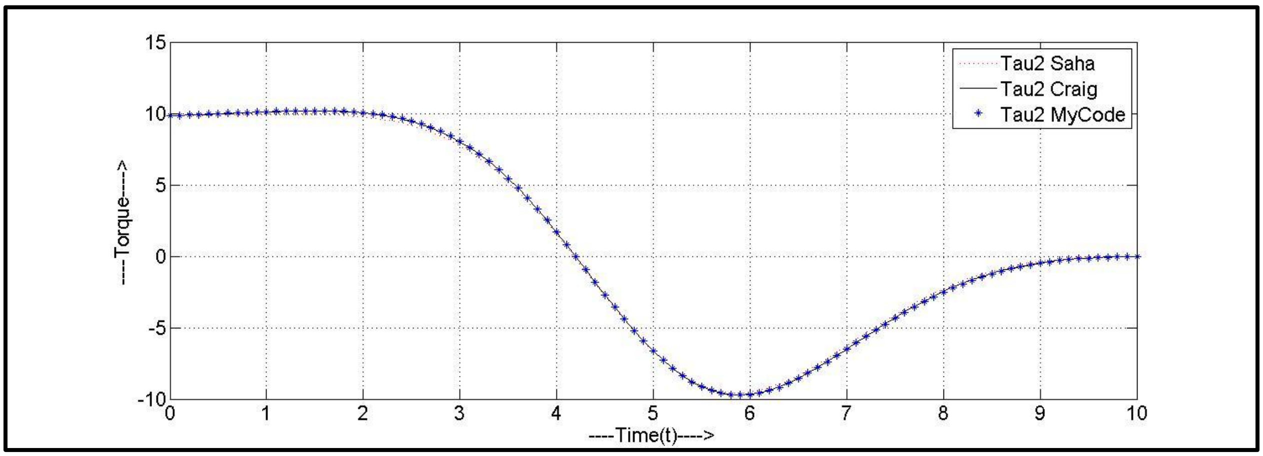

Fig. 3.8: Joint Torque at 2nd Joint.

3.4.2 Case Study 2: Six-Link Serial Manipulator

As most of the industrial robots are not limited to two-links only, it is very important to develop the accurate mathematical model for a manipulator having 4, 6 or any arbitrary number of links.

In this section, the dynamic model of 6-link serial manipulator is validated. As it is very difficult to find the analytical dynamic model for more than four-links in published literature, the problem boils down to validate the model with the help of a software, named as 'RoboAnalyzer6' developed by S. K. Saha in IIT, Delhi. This software is being used to generate as well as validate the dynamic model of a Six-Link serial Manipulator.

Taking six-link manipulator parameters as, the length of all links as 1 m, mass of each link as 1 Kg and zero inertia for all the links.

Joint variable, joint angles, joint velocity and joint acceleration are following cycloidal trajectory, as given ins Eqn. 5.3 – Eqn. 5.5

Where $\theta(0)$ is the initial angle and $\theta(T)$ is the final angle, which varies in time T=10 secs, taking 10 steps in each second.

Joint angle θ_1, θ_3, θ_5, varies from 0° degree to 180° and θ_2, θ_4, θ_6, angle varies from 0° to 90° in this given time T=10 secs.

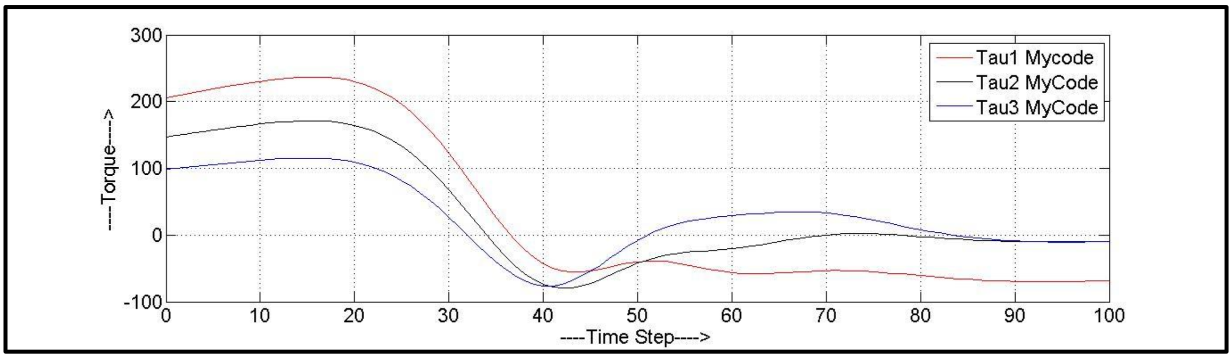

Fig. 3.9: Joint Torque plot of first-three axis obtained with MATLAB code.

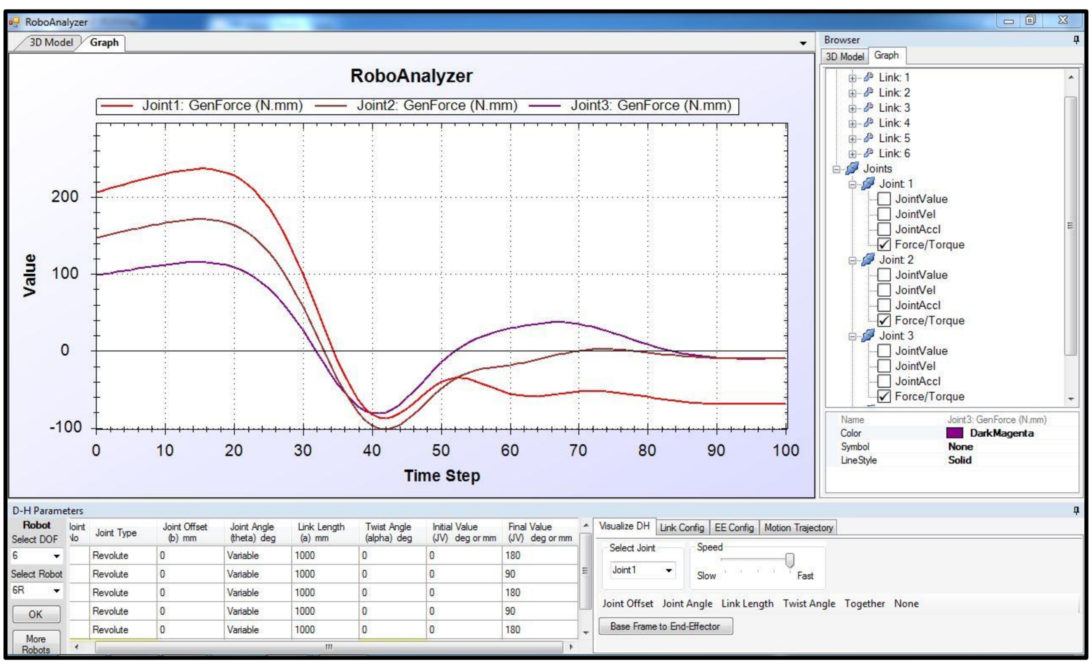

Fig. 3.10: Joint Torque plot of first-three axis obtained with "Robo Analyzer6".

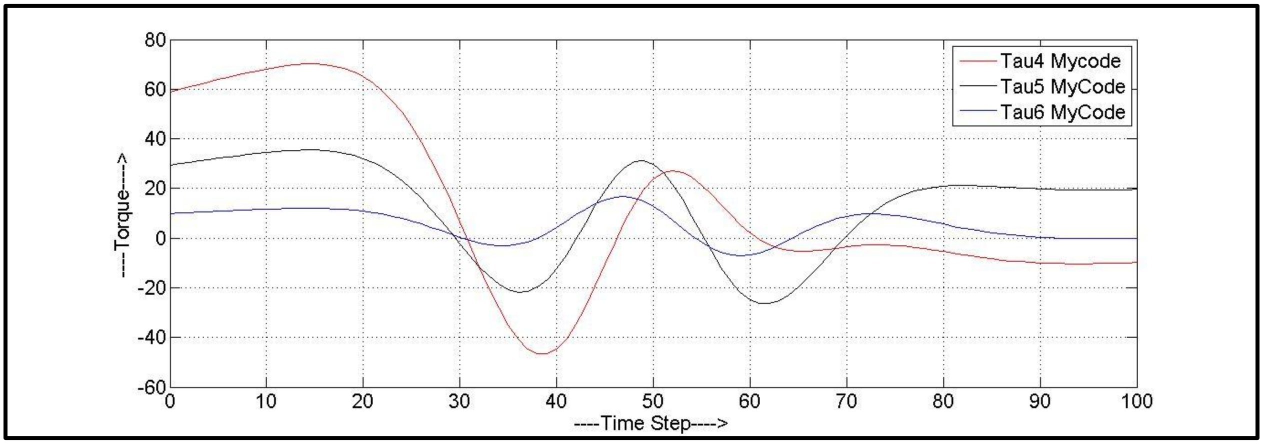

Fig. 3.11: Joint Torque plot of four-six axis obtained with MATLAB code.

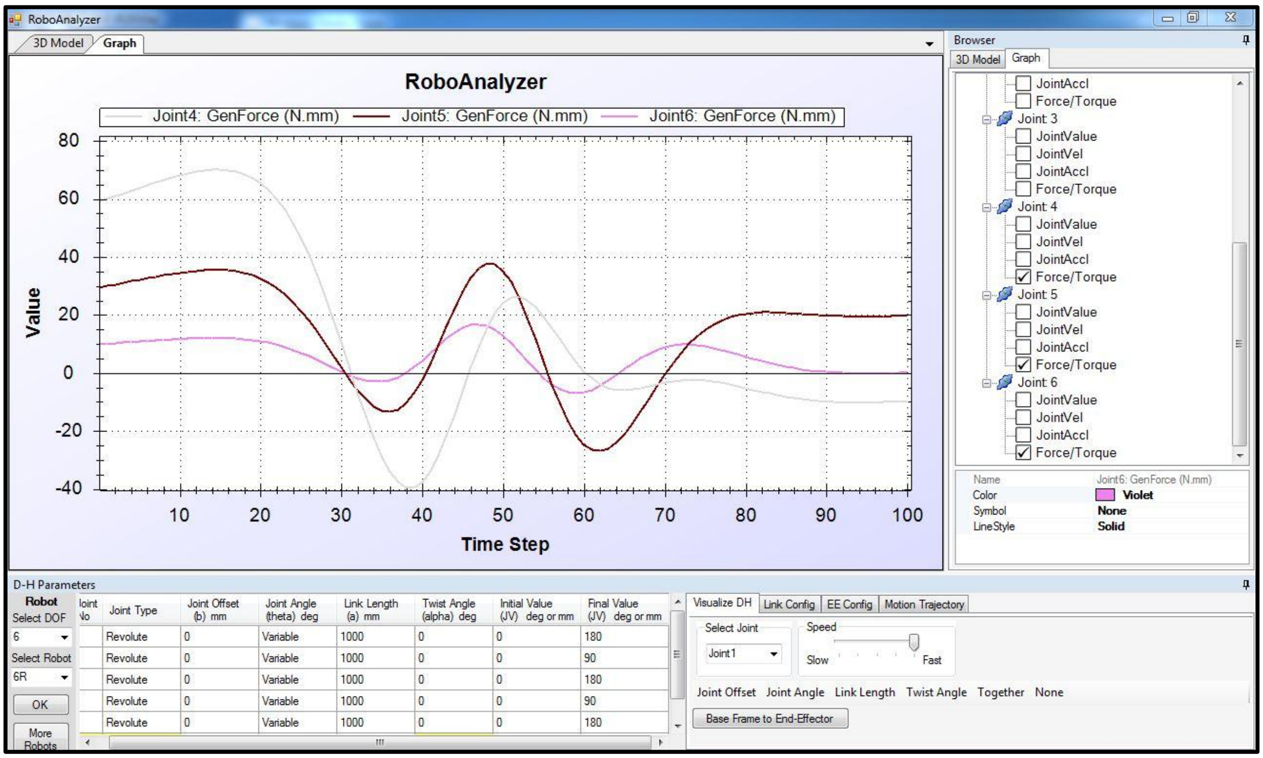

Fig. 3.12: Joint Torque plot of four-six axis obtained with "Robo Analyzer6".

3.5 SUMMARY

In this Chapter, it has been discussed, how to develop a dynamic model of a serial manipulators. Different modeling techniques – Newton-Euler approach and Euler-Lagrange approach, has been discussed to generate the dynamic model. The first one is based on energy conservation and later one is based on force-moment balance.

Further the developed dynamic model has been validated with various examples like two-link and six-link manipulator. Solved problems of two-link manipulator are taken from published literatures. Taking this validation beyond this level with six-link manipulator, modeled on a software "RoboAnalyzer6" available freeware on IIT-Delhi website. Results from the software and MATLAB were found in close agreement.

Authenticity of the code has been proved with these validation, this code can be used for a system with higher DOF. There is no limitation in terms of DOF with this MATLAB code. The model developed in this chapter is helpful to generate static force analysis of serial manipulators, which is performed is Chapter 4.

Chapter 4 STATIC FORCE ANALYSIS OF REDUNDANT MANIPULATORS

4.1 INTRODUCTION

It is concluded in Chapter 3 that it is difficult to find the of dynamic model for manipulators having higher degrees-of-freedom. Dynamic model in analytical form are available for 2, 3 or maximum upto 4-Link manipulators in published literature. Even software developed at IIT-Delhi is limited to handle 6-Links only. This code developed in MATLAB environment, has no such limit in term of degrees-of-freedom. Code can be implemented on any serial manipulator with arbitrary number of links.

So far, the MATLAB code is developed and validated through various examples. Now in this Chapter, the developed code is deployed on different cases to study the forward kinematics, inverse dynamics, static force analysis, actuator weight compensation. The chapter concludes with a lookup table, which is helpful in selecting the proper actuator

Moreover, torque comparison is done between some random configurations and the worst configuration. And importance of the worst configuration to facilitate the static force analysis throughout the whole workspace is highlighted.

4.2 PROCEDURE

The following procedure is opted to carried out the static force analysis:

- First of all, define all the parameter for a manipulator including DH parameters, mass inertias, Joint limits and derived trajectories.
- Put these values in MATLAB Code to calculate the dynamic model.
- Taking worst configuration into account, calculate the joint-torques requirement at each joint.
- Now, taking link parameter and motor weights, make a model on Solid Works for static force analysis.

- For the actuator weight compensation, the joint torques are calculated iteratively, starting backward from the distal end. This is required, because in a serial manipulator, the base joint and any successive joint thereafter has to carry the load of each link along with the actuator weight.
- The process is iterative, which means that first of all. The joint-torque is calculated for the distal link, the appropriate motor is selected. Thereafter, the weight of the actuator is compensated for the previous link torque calculation.
- The next step is to perform the static force analysis in Solid Works environment after completing the actuator weight compensation. Using the same parameters the manipulators are designed at worst configuration and stress, strain and deflection plots are obtained.
- Both the process, i.e. actuator weight compensation and static force analysis, are performed for three different cases:
    - With no actuator weight
    - With actuator attached perpendicular to the link axis.
    - With actuator placed along the link

4.3 MOTOR SELECTION

From previous sections, it can be clearly understood that how important is to select the right motor. A lookup table is developed using Maxon Motor Manual [25], by selecting the lowest weight motor falling in appropriate Torque Range. This table can be used to select motor ranging from 0 to 192 N-m Torque. One can choose motor by torque value from Table 4.1. Motor Weight, Power consumption, Motor diameter, Lenght and Brush material data are given along Model Number of DC Motors, manufactured by Maxon Co.

Table 4.1: Lookup table for motor Selection [25].

Torque Range (Nm)	Model No.	Motor specifications				Recomm-ended Motor
		Weight (grams)	Power (W)	Diameter (mm)	Length (mm)	
0 - 0.8	RE 10 256090	7	0.75	10	17	RE 10 256090
	A-max 12 200937	11	0.75	12	22	
	RE-max 13 268340	15	0.75	13	21	
0.8 - 1.2	A-max 16 352870	22	2	16	25	RE 13 118423
	RE 13 118423	15	1.2	13	21	
	RE-max 13 203938	24	2.5	13	31	
1.2 - 3.0	RE 13 113514	21	2	13	24	RE 13 113514
	A-max 19 110083	33	2.5	19	29	
	RE-max 17 214899	26	4	17	25	
3.0 - 6.1	A-max 22 110019	54	5	22	31	RE-max 21 221011
	RE-max 21 221011	42	5	21	28	
	RE 22 110120	54	5	22	32	
6.1 - 12.2	RE-max 24 222050	71	11	24	32	RE-max 24 222050
	RE 25 118741	130	10	25	54	
12.2 - 17.8	A-max 26 353613	119	11	26	45	A-max 26 353613
17.8 - 30	RE 25 302009	130	20	25	55	RE 25 302009
30 - 45.1	A-max 32 353233	240	20	32	64	A-max 32 353233
45.1 - 89.7	RE 30 268216	260	60	30	68	RE 30 268216
89.7 - 106	RE 35 273759	340	90	35	71	RE 35 273759
106 - 192	RE 40 218013	480	150	40	71	RE 40 218013

4.4 CASE STUDIES

Case study is done on three different cases, by following above procedure, calculations are done with MATLAB Code. Assuming the all parameter, which are essential to define a link completely. for the following cases:

- Four–Link serial manipulator
- Six–Link serial manipulator
- Eight–Link serial manipulator

4.4.1 Four–Link Manipulator

First of all consider a four-link serial manipulator. Each link is assumed to be rectangular with length 300 mm, width 30 mm and height 6 mm, and centre of mass at the centre of each link. Material of the link is assumed to be alloy steel. All the calculation are done for the worst configuration, as its importance is discussed in Section 4.4.

By following the above procedure, joint torques for worst configuration are obtained from code and motor weights from lookup table. Using these values, a model is created on 'Solid Works' and static force analysis is performed to stress, strain and deflection plots.

Stress, strain and deflection analysis results are shown in Figs. 4.1, 4.2 and 4.3 without consideration of motor weight.

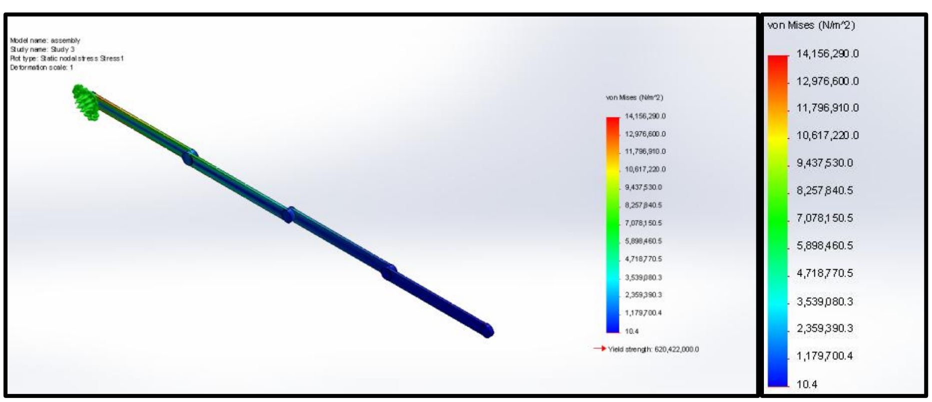

Fig. 4.1: Stress Analysis for 4-Link without motor weight compensation.

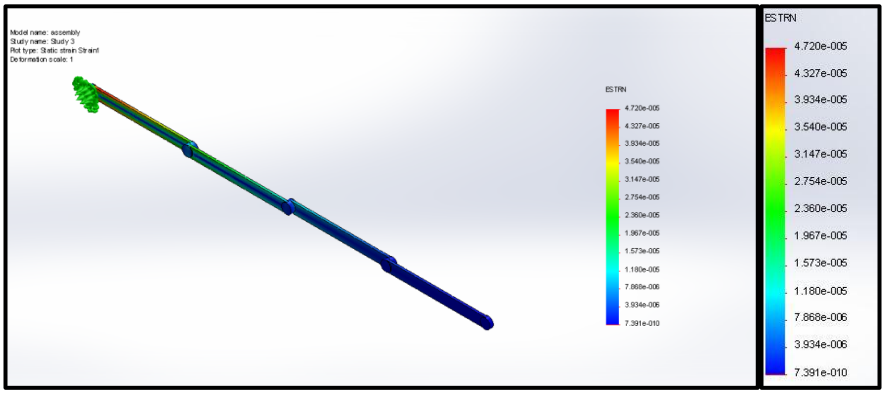

Fig. 4.2: Strain Analysis for 4-Link without motor weight compensation.

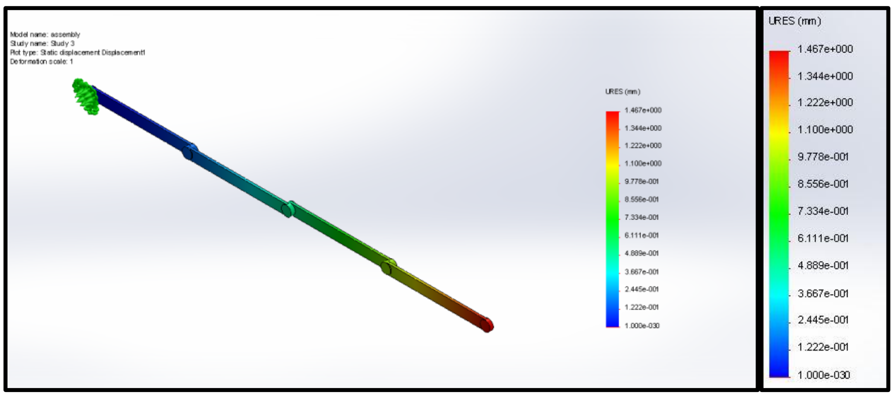

Fig. 4.3: Deflection Analysis for 4-Link without motor weight compensation.

In the next test run, all the parameter kept same along with actuator weights at each joint. Motors are kept perpendicular to link. Motor weight is assumed to be a point mass, at one end of the motors. Stress, strain and deflection analysis results are shown in Figs. 4.4, 4.5 and 4.6 .

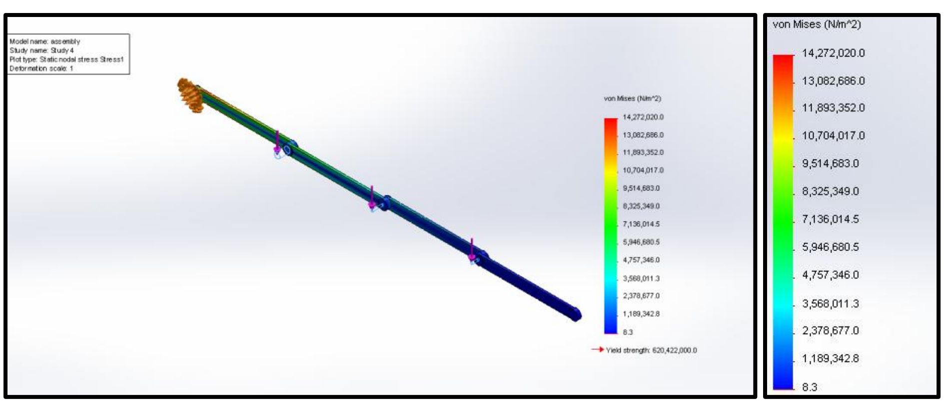

Fig. 4.4: Stress Analysis for 4-Link with motor weight compensation perpendicular to link.

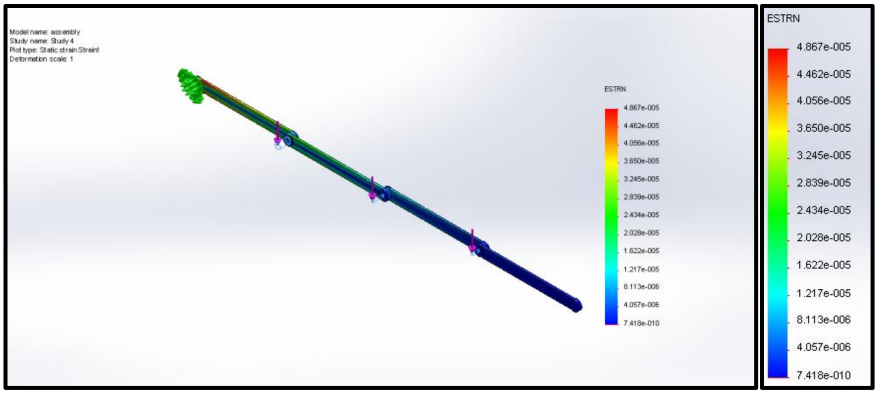

Fig. 4.5: Strain Analysis for 4-Link with motor weight compensation perpendicular to link.

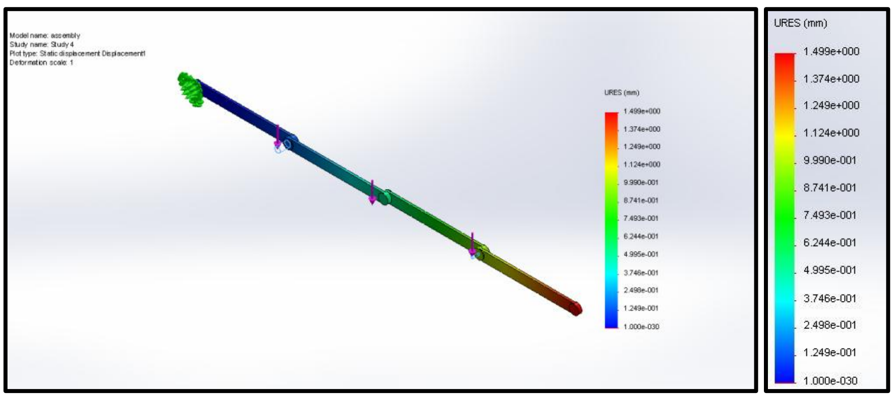

Fig. 4.6: Deflection Analysis for 4-Link with motor weight compensation perpendicular to link.

Unlike the previous run, where all the motor are perpendicular to respective joint axis, the motors are kept along the links in this test run. The previous case has a potential demerit that perpendicular motor can act as a obstacle and can hinder the robot movement within it's workspace. Now, considering all the parameter and actuator weights same as above, at each. But the orientation of motor is changed to along the link. Motor weights are assumed to be a point mass at the outer end of the motor. Stress, strain and deflection analysis results are shown in Figs. 4.7, 4.8 and 4.9.

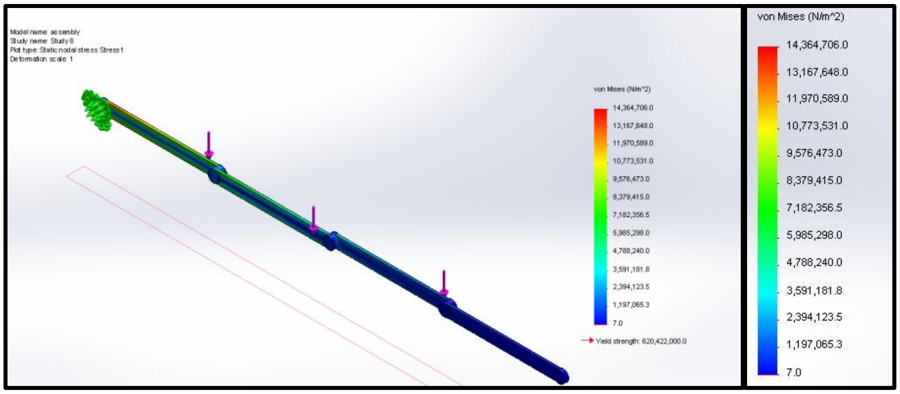

Fig. 4.7: Stress Analysis for 4-Link with motor weight compensation along the link.

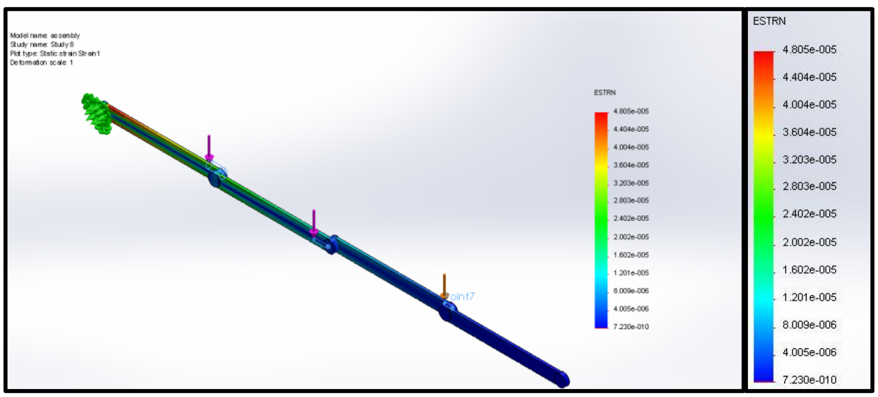

Fig. 4.8: Strain Analysis for 4-Link with motor weight compensation along the link.

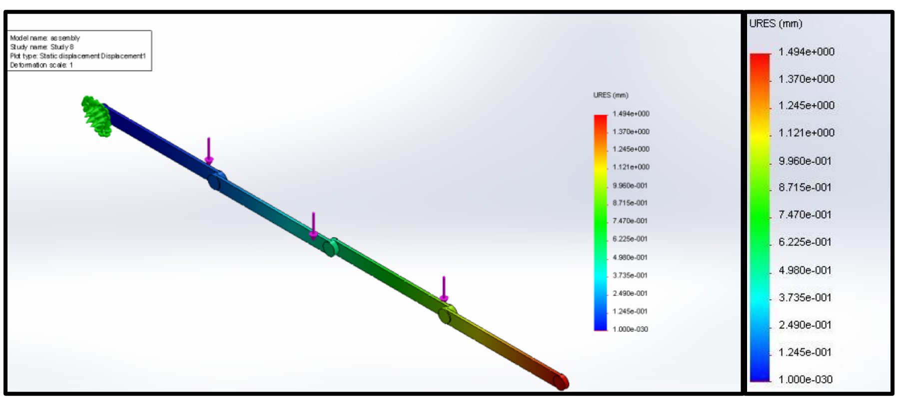

Fig. 4.9: Deflection Analysis for 4-Link with motor weight compensation along the link.

4.4.2 Six Link Manipulator

In the next test run, consider a six-link serial manipulator. All the data is taken same as in four-link case. Each link is a rectangular in geometry with length 300 mm, width

30 mm and height 6 mm, and centre of mass at the centre of each link. Material of the link is assumed to be alloy steel. Again, all the calculation are done for the worst configuration.

Opting the above procedure as discussed in section 4.2, joint torques for worst configuration are obtained from code and motor weights from lookup table. Using these values, a model is created on 'Solid Works' and static force analysis is performed to stress, strain and deflection plots.

Stress, strain and deflection analysis results for six link are shown in Figs. 4.10, 4.11 and 4.12 without consideration of motor weight.

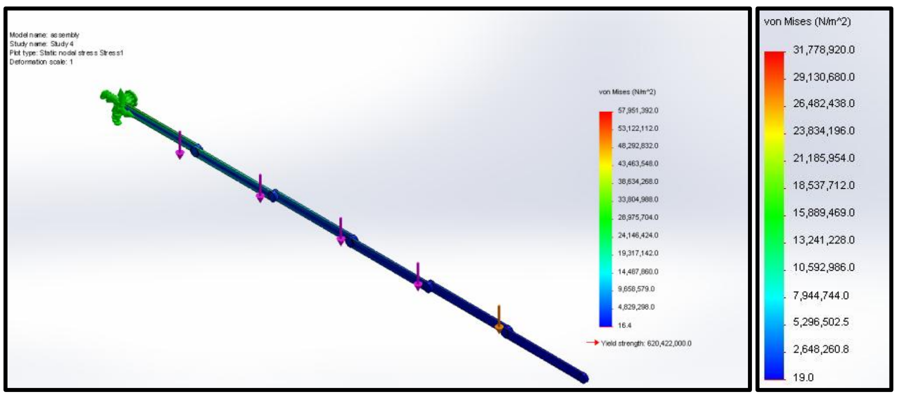

Fig. 4.10: Stress Analysis for 6-Link without motor weight compensation.

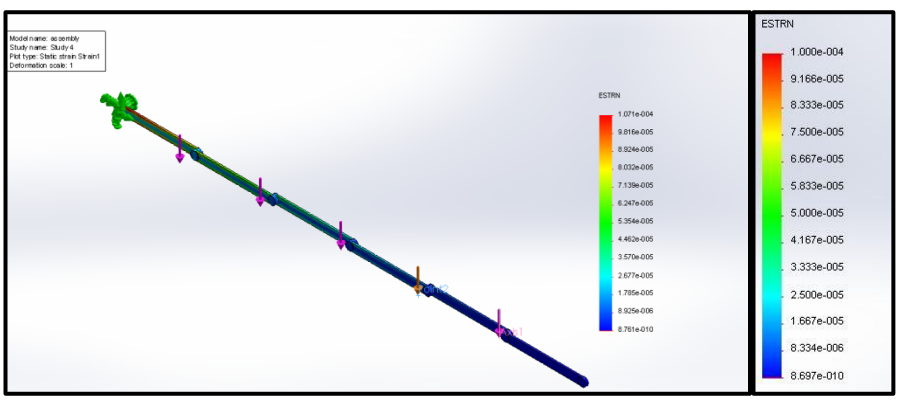

Fig. 4.11: Strain Analysis for 6-Link without motor weight compensation.

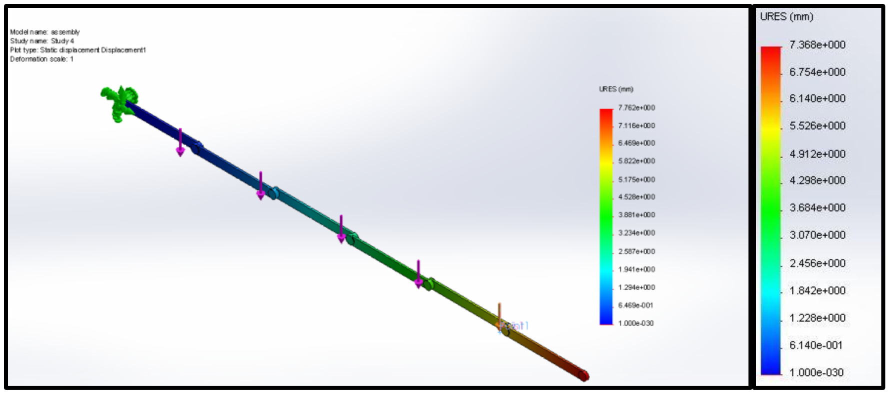

Fig. 4.12: Deflection Analysis for 6-Link without motor weight compensation.

Now, adding the motor weight to the links. Motor weights are assumed to be a point mass at the outer end of the motor. Motors are kept perpendicular to link. Stress, strain and deflection analysis plots are shown in Figs. 4.13, 4.14 and 4.15 .

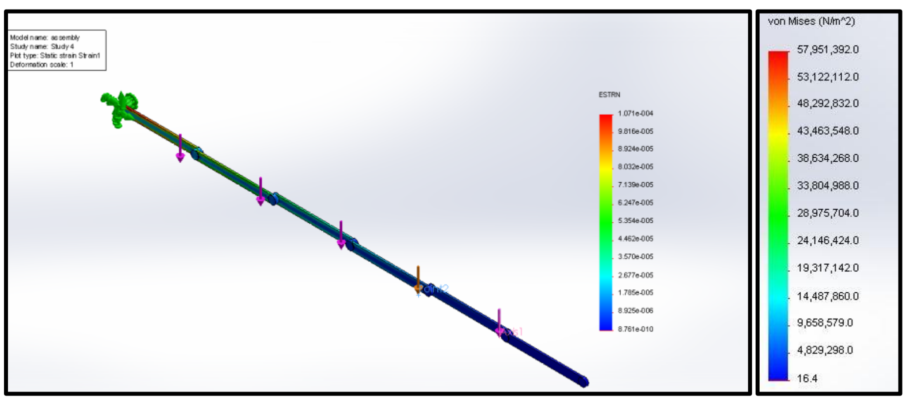

Fig. 4.13: Stress Analysis for 6-Link with motor weight compensation perpendicular to link.

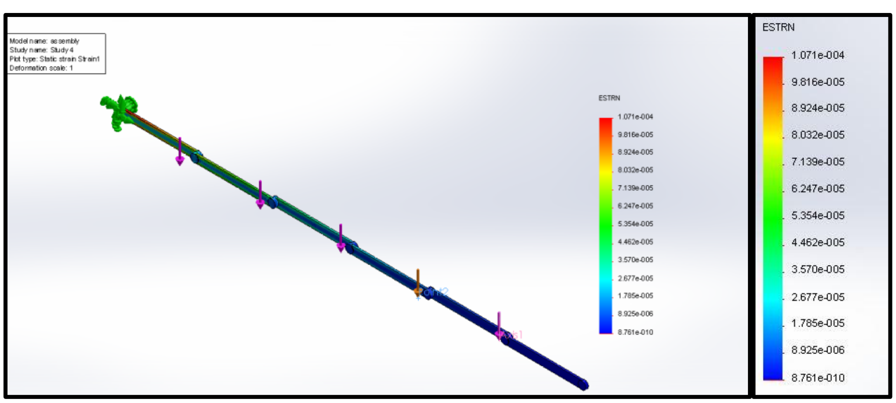

Fig. 4.13: Strain Analysis for 6-Link with motor weight compensation perpendicular to link.

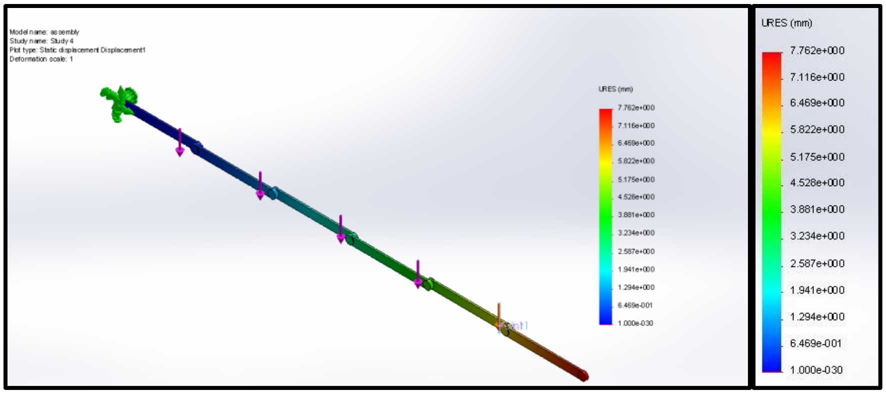

Fig. 4.15: Deflection Analysis for 6-Link with motor weight compensation perpendicular to link.

Again considering all the parameter and actuator weights same as above, at each joint with a change in orientation of the motors, motor are now kept along the Link. Motor weights are assumed to be a point mass at the outer end of the motor. Stress, strain and deflection analysis for Six-Links results are shown in Figs. 4.16-4.18:

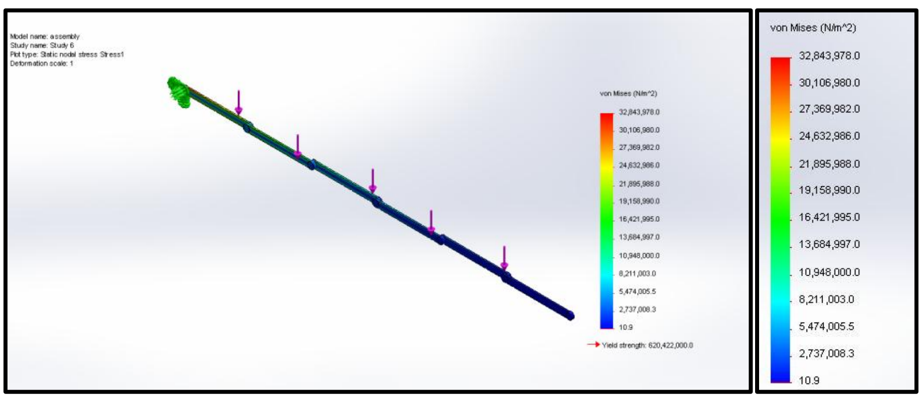

Fig. 4.16: Stress Analysis for 6-Link with motor weight compensation along the link.

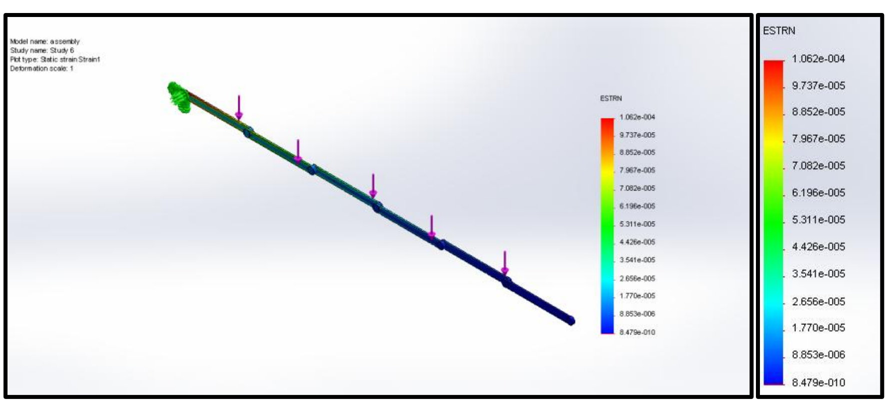

Fig. 4.17: Strain Analysis for 6-Link with motor weight compensation along the link.

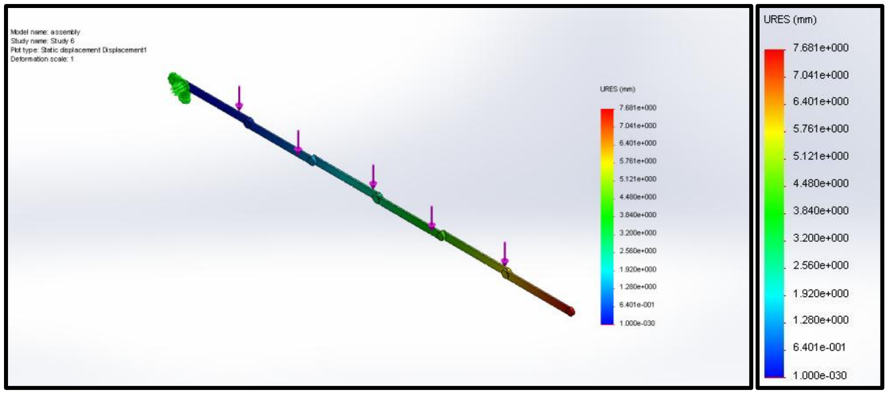

Fig. 4.18: Deflection Analysis for 6-Link with motor weight compensation along the link.

4.4.3 Eight-Link Manipulator

Taking this test run further, with analysis on eight-link manipulator. All the input data ach link is assumed to be same as in four-link case study, with rectangular links of length 300 mm, width 30 mm and height 6 mm, and centre of mass at the centre of each link. Material of the link is assumed to be alloy steel. All the calculation are done for the worst configuration.

Following the above procedure again, joint torques for worst configuration are obtained from code and motor weights from lookup table. Using these values, a model is created on 'Solid Works' and static force analysis is performed to stress, strain and deflection plots.

Stress, strain and deflection analysis results for six link are shown in Figs. 4.19, 4.20 and 4.21 without consideration of motor weight.

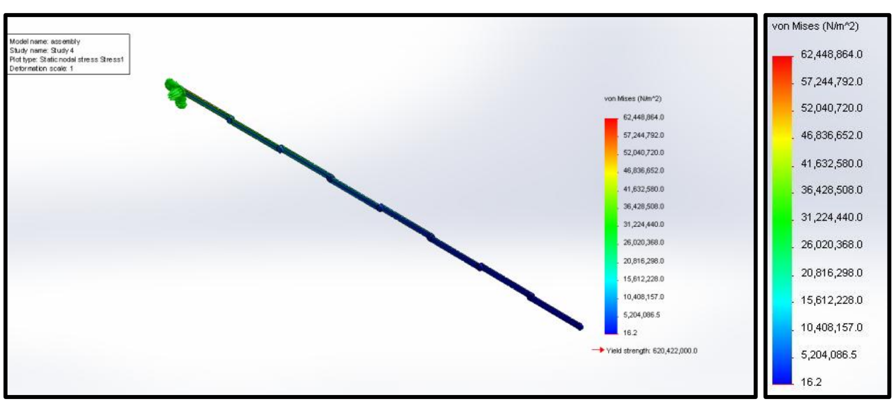

Fig. 4.19: Stress Analysis for 8-Link without motor weight compensation.

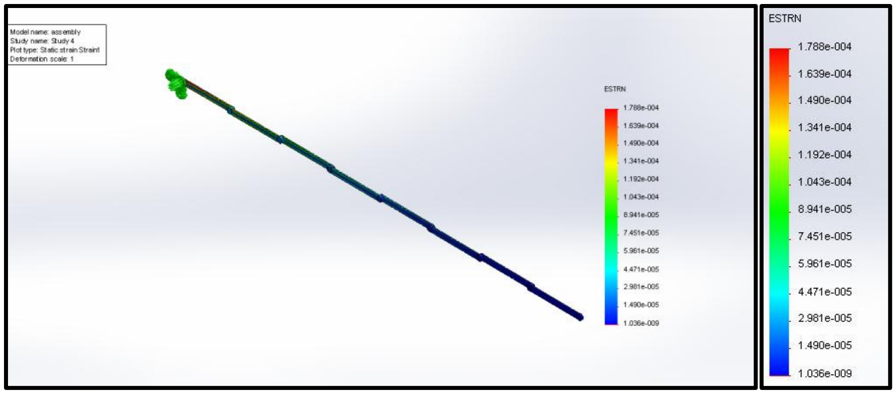

Fig. 4.20: Strain Analysis for 8-Link without motor weight compensation.

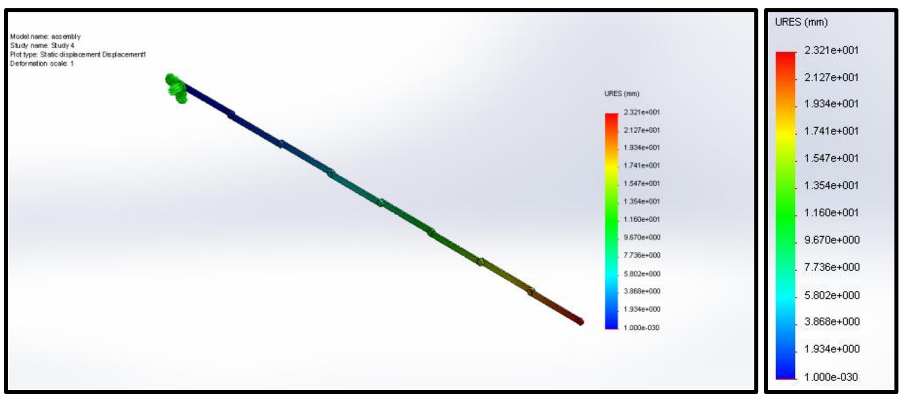

Fig. 4.21: Deflection Analysis for 8-Link without motor weight compensation.

In this next test run, all the parameter kept same along with actuator weights at each joint. Motors are kept perpendicular to link. Motor weight is assumed to be a point mass, at one end of the motors. Stress, strain and deflection analysis results are shown in Figs. 4.22, 4.23 and 4.24 .

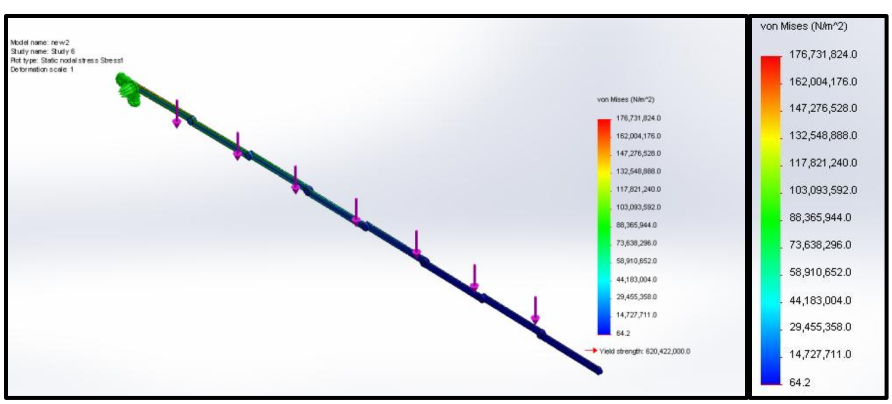

Fig. 4.22: Stress Analysis for 8-Link with motor weight compensation perpendicular to link.

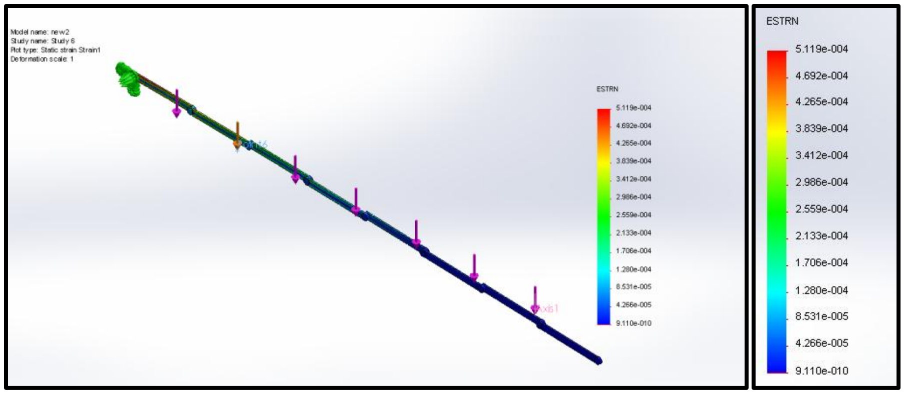

Fig. 4.23: Strain Analysis for 8-Link with motor weight compensation perpendicular to link.

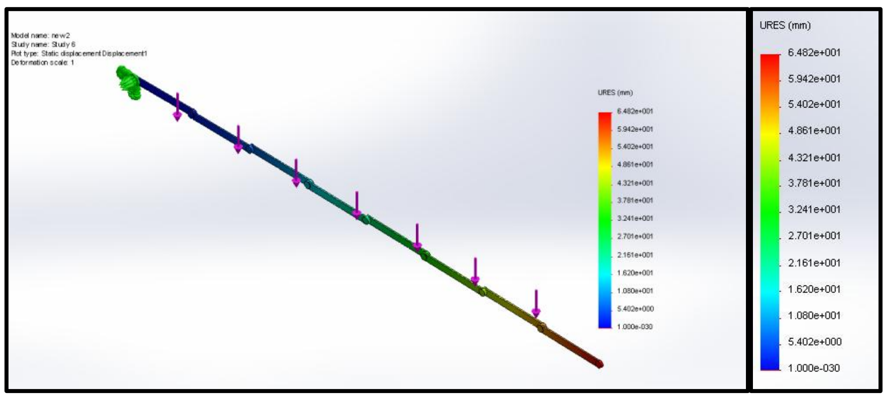

Fig. 4.24: Deflection Analysis for 8-Link with motor weight compensation perpendicular to link.

Again considering all the parameter and actuator weights same as above, at each joint with a change in orientation of the motors, motor are now kept along the Link. Motor weights are assumed to be a point mass at the outer end of the motor. Static force analysis for Stress, strain and deflection, on Eight-Links results are shown in Figs. 4.25, 4.26 and 4.27.

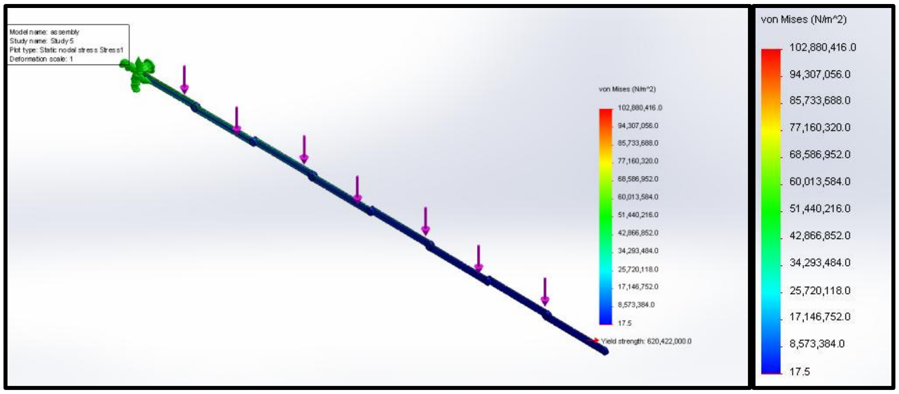

Fig. 4.25: Stress Analysis for 8-Link with motor weight compensation along the link.

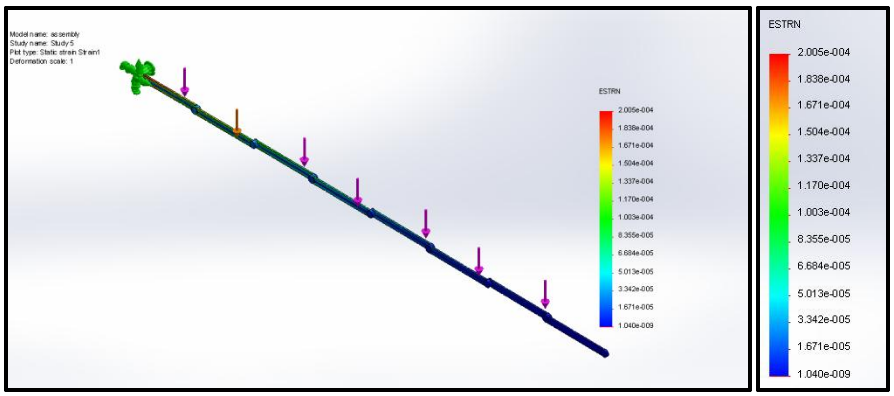

Fig. 4.26: Strain Analysis for 8-Link with motor weight compensation along the link.

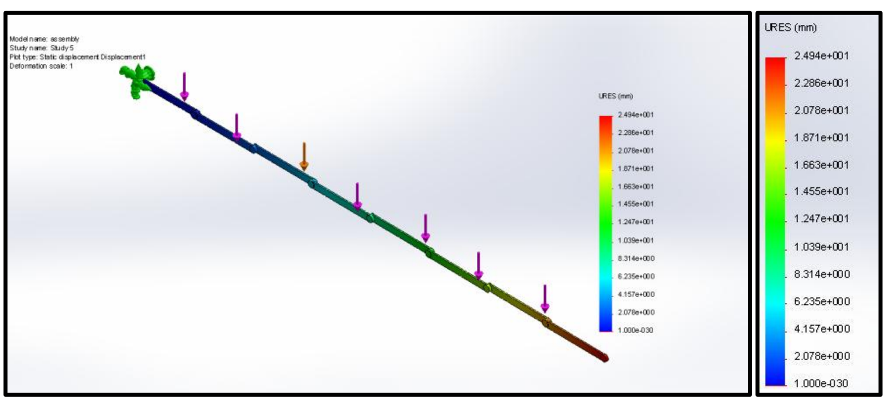

Fig. 4.27: Deflection Analysis for 8-Link with motor weight compensation along the link.

Put the static force analysis from Figs. 4.1-4.27, into Table 4.2, 4.3 and 4.4, to make a comparison chat between the different orientations.

Table.4.2: Comparison Table for 4 Link Manipulator.

Analysis Type	**Without Motor Weight**	**With Motor Perpendicular to Link**	**With motor along the Link**
Deflection(mm)	1.467	1.499	1.494
Stress(N/m^2)	1.415×10^7	1.427×10^7	1.436×10^7
Strain	4.720×10^{-5}	4.867×10^{-5}	4.805×10^{-5}

Table.4.3: Comparison Chart for 6 Link Manipulator.

Analysis Type	**Without Motor Weight**	**With Motor Perpendicular to Link**	**With motor along the Link**
Deflection(mm)	7.668	7.762	7.681
Stress(N/m^2)	3.177×10^7	5.795×10^7	3.284×10^7
Strain	1.000×10^{-4}	1.071×10^{-4}	1.062×10^{-4}

Table.4.4: Comparison Chart for 8Link Manipulator.

Analysis Type	**Without Motor Weight**	**With Motor Perpendicular to Link**	**With motor along the Link**
Deflection(mm)	23.21	64.82	24.92
Stress(N/m^2)	6.244×10^7	1.767×10^8	1.028×10^8
Strain	1.788×10^{-4}	5.119×10^{-4}	2.005×10^{-4}

It can be clearly seen from Tables 4.2, 4.3 and 4,4, that the orientation of motor have major effect on the link in term of Static force analysis. Like in Table 4.4 maximum deflection is 64.82 mm with motors perpendicular to links where as deflection is just 24.92mm which is very near to deflection 23.21mm without motor weight case. The same is for the stress and strain analysis.

Motor placed along the link shows very good results in terms of static force analysis. But the placed of motor along the links is another challenge, whether to place the motor in side the link or some mountings can be used. Now motor placed inside the link can affect the strength of the link and in another case some mountings used to hold the motor outside the link, these mountings can increase the weight on the system and can create obstacle to the manipulator, work space can be affected.

4.5 WORST CONFIGURATION VS RANDOM CONFIGURATIONS

Motor or actuator applied on every joint should satisfy the joint torque requirements at any configuration within the workspace.

To verify that torque produced at each joint can satisfy torque requirement for any general configuration, three random configurations of a Six-Link Manipulator are taken, with motor weights as shown in figure 4.28. All the link parameters are considered as same as in previous case study of Six-Link case.

Its clearly shown in figures, torque that can be produced by the actuators at each joint axis is all ways lesser than the requirement by Fig.4.29.

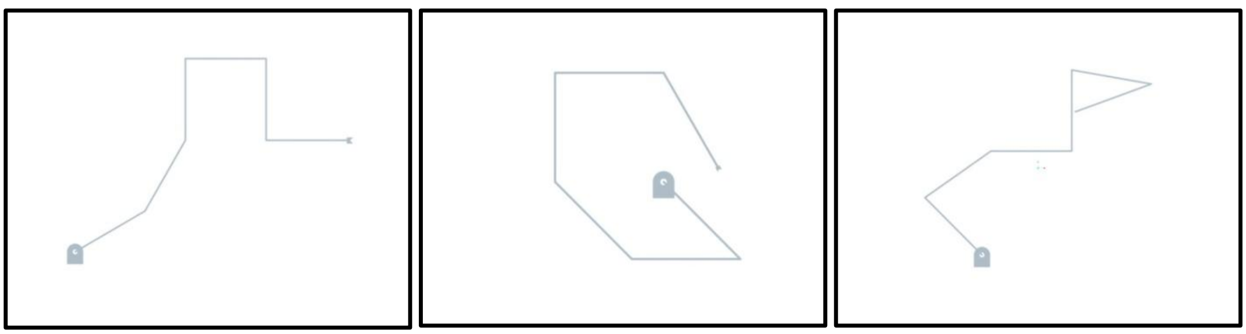

Fig 4.28 : Random configurations

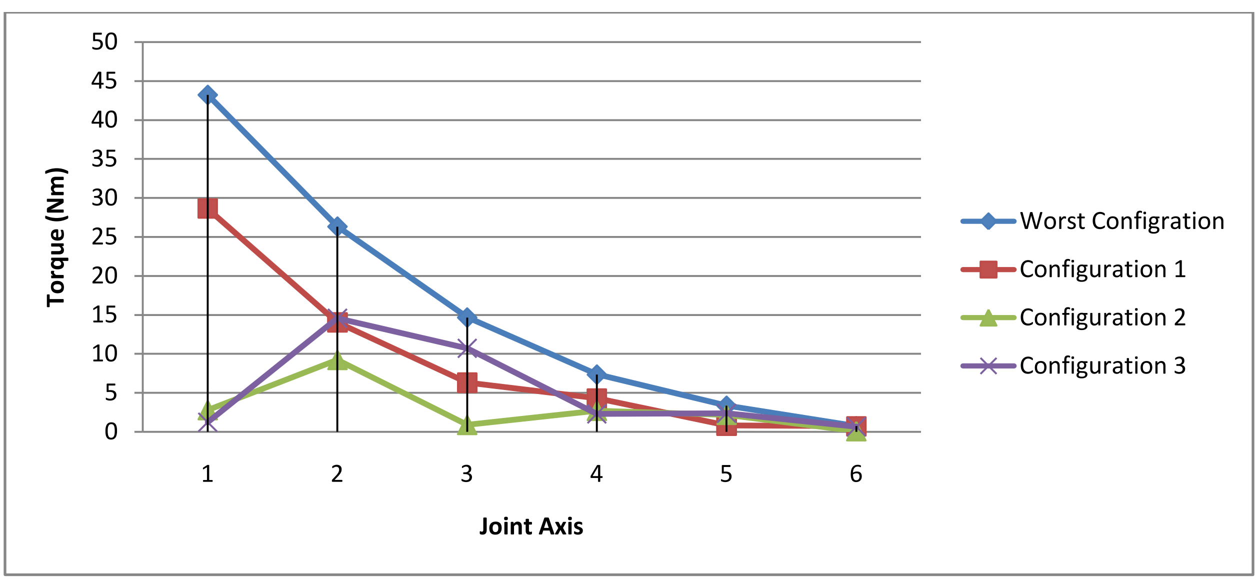

Fig 4.29: Comparison chart Worst configuration and Random configurations

4.6 SUMMARY

A MATLAB code has developed and validated in Chapter 3, for the dynamic modeling of serial manipulators. This Code can be implemented on any serial manipulator with arbitrary number of links. In this Chapter, the developed MATLAB code has deployed on different cases to study the forward kinematics, inverse dynamics, static force analysis, actuator weight compensation. This chapter concludes with a lookup table, which is helpful in selecting the proper actuator

Moreover, torque comparison has done between some random configurations and the worst configuration. And importance of the worst configuration to facilitate the static force analysis throughout the whole workspace has been highlighted.

Chapter 5 CONCLUSIONS AND FUTURE DIRECTIONS

5.1 CONCLUSIONS

The following conclusions can be drawn from this thesis work:

- Calculating the dynamic model of a general robotic manipulator, which is extremely nonlinear in nature, is a challenging task. The problem becomes more complex with increase in number of links. In this thesis, a complete code in MATLAB environment has developed, which can generate dynamic model of any general serial manipulator having arbitrary number of links.
- Another big challenge is the selection of appropriate actuator for each joint. Scope of this thesis is extended to the development of a lookup table, which is helpful in selecting the appropriate actuator at each joint axis, and plotting results for static force analysis with and without actuator weight compensation.
- Orientation of the motor also a main concern, as its effects can be seen from the Tables 4.2, 4,3 and 4.4. Change in the orientation effect on workspace of robot also. These motors can act as a obstacle in movement of robotic arm or can affect the strength of serial link if placed inside the links.
- Motor placed along the link shows very good results in terms of static force analysis. But the placement of motor along the links is another challenge, whether to place the motor in side the link or some mountings can be used. Now motor placed inside the link can affect the strength of the link and in another case some mountings used to hold the motor outside the link, these mountings can increase the weight on the system and can create obstacle to the manipulator, work space can be affected.
- Moreover, the worst configuration torque values are compared with three random configurations, which proves that static force analysis done at worst configuration is sufficient to draw conclusion for the whole system.

5.2 FUTURE DIRECTIONS

The current work is an initiative to develop such type of Dynamic Modeling of Serial Manipulators, there exist a wide range of possibilities for extensions. Several interesting openings for future research and development are:

- Along with the motor direction, the work can be extended to include gears and mounting designs.
- The work can be extended to develop a laboratory prototype.
- The work can be extended to include prismatic links also.
- Software can be developed for virtual simulation of such dynamic model.

REFERENCES

[1] http://en.wikipedia.org/wiki/File:Hexapod0a.png

[2] http://upload.wikimedia.org/wikipedia/commons/thumb/9/9b/Robot_arm_model_1.png

[3] http://upload.wikimedia.org/wikipedia/commons/0/00/Scara.gif

[4] http://la.epfl.ch/files/content/sites/la/files/users/139973/public/photo-project/NASAmanip.png

[5] http://www-robot.mes.titech.ac.jp/hirose/robot/snake/oblix/photo1.jpg

[6] http://2.bp.blogspot.com/82bSwi4S5_0/UM_ENxDceqI/AAAAAAAAAFY/0qJPycUPyDo/s1600/1.png

[7] CraigJ.J., Introduction to Robotics Mechanics and Control, *Addison Wesley Longman*,ISBN0-201-09528-9, pp. 1–450, 2002

[8] S. K. Saha, Intorduction to Robotics, *Tata McGraw-Hill*, ISBN-13:978-0-07—066900-0,pp. 1-371, 2012

[9] K. S. Fu, Ralph Gonzalez and C. S. G. Lee, *Tata McGraw-Hill Education*, pp.1-580 1987

[10] S.K. Saha, B. Shirinzadeh and Gürsel Alici, Dynamic Model Simplification of Serial Manipulators, *Int. Symp. On Robotics & Automation,ISRA*, pp. 1-6. 2006

[11] Ganwen Zeng and Ahmad Hemami, An overview of robot force control, *Robotica*, Vol.15 , pp 473 – 482, 1997

[12] Ekta Singla, Ashish Singla and Bhaskar Dasgupta, An Posterior Analysis Tool for Redundant Manipulators, *2nd Indo-Russian Ioint Workshop on Computing Intelligence and Modern Heuristics in Automation and Robotics,* pp. 213-318, 2011

[13] Manindra Kaphle, Anders Eriksson, Optimality in forward dynamics simulations, *Journal of Biomechanics* Vol. 41, pp. 1213–1221, 2008

[14] Yang Zhao, Zheng Feng Bai, Dynamics analysis of space robot manipulator with joint clearance, *Acta Astronautica* Vol. 68, pp. 1147–1155, 2011

[15] Shin-Min Song, An efficient method for inverse dynamics of manipulators based on the virtual work principle, *Journal of Filed Robotics,* Vol. 10, No. 2, pp. 605-627.

[16] R. Tapia Herrera, Samuel M. Alcántara, J.A. Meda-Campaña and Alejandro S. Velázquez, Serial and Parallel Robot Manipulators - Kinematics, Dynamics Control and Optimization, ISBN 978-953-51-0437-7, pp. 67-83, 2012

[17] S.F. Chan, R. Kwan , Post-processing methodologies for off-line robot programming within computer integrated manufacture, *Journal of Materials Processing Technology* Vol. 139, pp. 8–14, 2003

[18] Dae-Jin Kim, Rebekah Hazlett, Heather Godfrey, Greta Rucks, David Portee, John Bricout, Tara Cunningham, and Aman Behal, On the Relationship between Autonomy, Performance, and Satisfaction: Lessons from a Three-Week User Study with post-SCI Patients using a Smart 6DOF Assistive Robotic Manipulator, *International Conference on Robotics and Automation*,

pp. 217- 222, 2010

[19] Ramesh Kolluru, Kimon P. Valavanis, Stanford A. Smith, and Nikos Tsourveloudis, Design Fundamentals of a ReconfigurableRobotic Gripper System, *IEEE transactions on systems, man, and cybernetics—part a: systems and humans*, vol. 30, no. 2, pp. 181-187, march 2000

[20] Denny Oetomo, David Daney, and Jean-Pierre Merlet, Design Strategy of Serial Manipulators With Certified Constraint Satisfaction, IEEE Transactions On Robotics, Vol. 25, No. 1, pp. 1-11,2009

[21] Christiaan J.J. Paredis, Pradeep Khosla Kinematic Design of Serial Link Manipulators from Task Specifications, *The International Journal of Robotics Research*, 1996.

[22] Maxon DC Motor Manual (http://www.kwapil.com/downloads/maxon-dc-motor.pdf)