Optimal Pose Estimation of Robotic Manipulator For Pick And Place Application

By

Uzair Farrukh

273527

Supervisor

Dr. Karam Dad Kallu

Department of Robotics and Artificial Intelligence

School of Mechanical and Manufacturing Engineering (SMME)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

August 2022

Optimal Pose Estimation of Robotic Manipulator For Pick And Place Application

By

Uzair Farrukh

273527

Supervisor

Dr. Karam Dad Kallu

Co-supervisor

Dr. M.Jawad Khan

A thesis submitted in conformity with the requirements for the degree of *Master of Science* in Robotics and Intelligent Machines Engineering

Department of Robotics and Artificial Intelligence

School of Mechanical and Manufacturing Engineering (SMME)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

August 2022

Declaration

I, *Uzair Farrukh* certify that this research work titled "Optimal Pose Estimation Of Robotic Manipulator For Pick And Place Application" is my own work. The work has not been presented elsewhere for assesment. The material that has been used from other sources has been properly acknowledged/referred.

Uzair Farrukh,

2018-NUST-MS-RIME-273527

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature Of Student,
Uzair Farrukh,
273527

Signature Of Supervisor
Dr Karam Dad Kallu

Copyright Notice

Acknowledgments

I am thankful to Allah for guiding me at every level of my job. I'm grateful to my Creator, Allah Subhana-Watala, for guiding me through every stage of my effort and for each fresh idea You planted in my head to make it better. Indeed, without Your invaluable assistance and direction, I would have been powerless. It was Your will that everyone who assisted me with my thesis, including my parents and others, thus only You are deserving of recognition.

I am incredibly grateful to my wonderful parents who reared me when I was unable to walk and who have supported me throughout my life in every aspect. I also want to give a special thank you to my co-supervisor Dr.M.Jawad Khan and my supervisor Dr.Karam Dad Kallu for their assistance with my thesis. Additionally, I want to express my gratitude to my wife for all of her help and collaboration.She always had the answer when I was stuck on something. I wouldn't have been able to finish my thesis without her assistance. I am grateful for her assistance and patience during the entire thesis.

Finally, I would want to offer my appreciation to everyone who has helped my research in any way.

This thesis is dedicated to my *beloved parents* , *wife and daughters*

Abstract

This thesis examines how to use robotic manipulator to pick specific object. In order to attain above objective planning of manipulator as well as grasping point calculations are studied. The process of calculating grasping pose with respect to target object,the grasping point essential for successful grasping of object is called grip synthesis.Both analytical and empirical approaches are used to calculate grasping points. While empirical uses human like strategies, analytical approaches are mainly dependent on geometric, kinematic, and/or dynamic formulations. The main goal of the thesis is to minimize joint accelerations. Constrained optimization techniques like genetic algorithm and path search algorithm are used to find grasping point and minimize joint accelerations to ensure successful grasping of object.A novel hybrid approach is also proposed which used both genetic algorithm and path search algorithm to find its fitness value. The first contribution is finding optimal grasping points of regular shaped object by calculating centroid of object and are near to centroid along minor axis. The second contribution is to minimize joint accelerations by using optimization techniques of genetic algorithm and path search algorithm in which global minima's are found and optimized trajectory is generated which helps in successful grasps with minimum joints accelerations. Third is the introduction of novel hybrid approach which uses both genetic algorithm and pattern search algorithm to find its global minima which is more robust for changing search space.

Keywords: *Robot Grasping, Robot Grasp Synthesis, Robot Grasp Planning, Genetic Algorithm,Path Search Algorithm.*

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

LIS	Locked-in Syndrome
QoL	Quality of Life
BCI	Brain-computer Interface
EEG	Electroencephalography, Electroencephalographic
NIRS	Near Infrared Spectroscopy

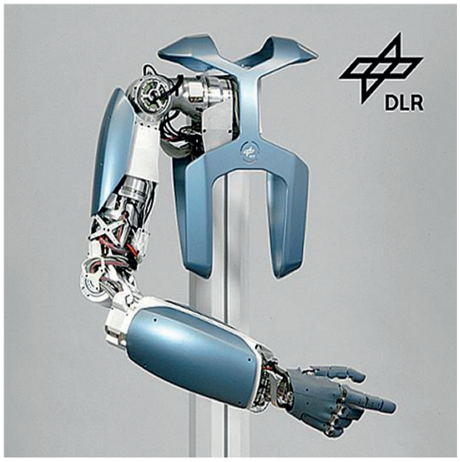
Chapter 1

Introduction

Karel Capek a czech writer published the play in 1920 which was named as Rossum's Universal Robots (R.U.R) thats deals with the ethical aspects of using cheap labor as artificially created. This work is regarded as the origin of the term robot that is derived from czech word robota thats means "hard work". Nowadays robots have become an indispensable centerpiece of automated manufacturing processes. In industrial setups where most important aspect is cost-effectiveness, they reliably carry out collection of tasks such as machining, welding,material transport, painting, assembly and packaging.machining, material transport, assembly and packaging. In this context, an often cited acronym characterizes the job the robot must execute as Dull,Dirty and Dangerous. In recent years robots have become available as consumer products in other domains such as service and entertainment.

In many applications of robot one of the important aspects in all domains is how robot interact with environment. The interface which acts as medium between robotic manipulator and environment is provided in terms of end-effector. Today, the majority of end-effectors, such as suction cups and parallel-jaw grippers, is simple and tailored to carry out specific tasks on specific objects. To avoid the need to change end-effectors on task to task basis. versatile and dextrous end-effectors are required. A solution is offered in form of articulated multi-fingered hands [10] . These are grasping devices which possess the ability to reconfigure themselves for performing different grasps. Such mechanisms were built first in the early 1980's. Among them are the Stanford/JPL hand [11] and the Utah/MIT hand [1] which is shown in Fig. 1.1(a). One line of research has focused on devising anthropomorphic (human-like) devices which attempt to mimic the human hand with its unsurpassed dexterous grasping and manipulation capabilities. The hand/arm system in [2], which is engineered by the German Aerospace Center (DLR) and depicted in Fig. 1.1(b) is one of the example. Devices in this mould are advantageous for applications such as teleoperation prosthetics and for service robots in a human environment.

(a)



(b)

(c)

(d)

(e)

Figure 1.1: Articulated Grasping Devices (a)The Utah/MIT hand, one of the first multifingered hands [1].(b)One of the most sophisticated anthropomorphic platforms available today is the DLR hand/arm system [2].(c)The 3-fingered Barrett hand features break-away transmissions in the distal joints which allows for robust grasping [3].(d)The lightweight high speed hand by Namiki et al allows for real-time visual feedback control [4].(e)The underactuated SDM hand is a low-cost compliant grasping device using a single actuator only [5].

However, anthropomorphism is neither necessary nor sufficient to achieve capability. There are many impressive grasping and manipulation devices with different mechanical structures. Examples include the Barrett hand(see Fig. 1.1(c)), which has become a popular research tool, and the high-speed hand in [4], which can perform highly dynamic tasks such as catching objects and is depicted in Fig. 1.1(d). Underactuated grippers comprising less actuators than Degrees of Freedom (DoF), such as the SDM hand [5] shown in Fig. 1.1(e), provide interesting and cost-effective alternatives. Here, the mechanisms are designed such that certain desired grasping/manipulation features are preserved.

In most robotic applications today, behaviors and motions are pre-programmed. In order for robots to leave the structured environments of industrial or laboratory settings and to succeed in uncontrolled scenarios, it has become clear that they need to be endowed with a sufficient level of autonomy. To purposefully interact with its environment, and as a prerequisite for any subsequent manipulation, a robot needs to be able to autonomously grasp objects in a robust manner which is the focus of this dissertation.

1.1 The Challenges of Autonomous Robot Grasping

The US Defense Advanced Research Projects Agency sponsors the famous DARPA Robotics Challenge (DRC), which aims to push the limits of supervised autonomy for mobile, mostly humanoid robots in emergency response situations.In this context, supervised autonomy means that there is a human teleoperator in the loop which can issue commands, albeit under the constraint of a limited bandwidth. The DRC trials held in 2013, the year before writing this thesis, included manipulation tasks such as opening a door or closing a valve. Even the most successful robots used up at least half of their 30 minutes time limit per challenge and a significant number of attempts failed. The purpose of the above example is to highlight the substantial difficulty of achieving even only partial autonomy in robot grasping and manipulation and the big gap between the capabilities of fictional robots and currently existing systems. With respect to humans, According to neuroscientific research, the control of the hand accounts for the biggest portion (30–40)percent of the motor cortex, the area of the brain responsible for planning and carrying out movements. To grab an object correctly, a robot must figure out how to see the object, synthesise the grip, and arrange the action of the hand and manipulator.

Object perception estimates the pose of the target object and, if not known a priori, its geometry from potentially incomplete and noisy sensor data. Solving this problem is aggravated by factors like occlusions of the target object by the environment or the robot itself, and varying light conditions across different scenarios which necessitate different calibrations/setups of the employed range sensing devices. Once a representation of the object is built by means of the available sensor inputs, it is necessary to address the grasp synthesis problem. Here, the goal is to determine a hand palm pose with respect to the object, as well as a joint configuration and/or grasp contact points such that a successful grasp can be achieved by an appropriate hand closing motion. This process is not trivial, especially considering uncertainties in the target object's pose and the achievable positioning accuracy of the robot platform. The purpose of hand motion planning is to generate a coordinated grasp movement which is particularly relevant when complex hands with many DoF are considered. Finally, manipulator motion planning is concerned with finding a collision-free path leading the grasping device from the initial pose to the grasping pose.

Problem Statement At this point, the general problem of interest in this dissertation can be stated.

Problem: *Given the pose and geometry of an object to be grasped with an articulated robotic grasping device, determine an appropriate set of contact points, palm pose and gripper joint configuration such that a coordinated grasp closing motion results in a stable grasp.*

Loosely speaking, the addressed question is *where* to grasp and *how* to grasp a given object. To this end, aspects of grasp synthesis and hand motion planning are investigated. The experiments presented in this work were conducted by using existing solutions for object perception and manipulator motion planning. A central tenet in this thesis is to circumvent the curse of dimensionality, which is inherent in high-dimensional planning problems, by incorporating empirical data in analytical approaches. Most of the proposed algorithms encapsulate a notion of optimality in the context of the tackled subproblem. Therefore, the use of tools from numerical optimization is a second central aspect in this dissertation.

1.2 Outline

This thesis' remaining sections are structured as follows. In Chapter 2, the dynamics, kinematics, DH table, and Lagrangian with derivations of manipulators are covered in depth. Chapter 3 provides an overview of optimization techniques to optimize trajectory and minimize joint acceleration. In-depth discussion is provided on the genetic algorithm and pattern search method. Chapter 4 is about simulation results of unoptimized trajectory with joint velocities and accelerations , optimized trajectory using genetic algorithm with joint velocities and accelerations,optimized trajectory using pattern search algorithm with joint velocities and accelerations and optimized trajectory using hybrid algorithm with joint velocities and accelerations

1.3 Contributions

As mentioned in the preceding section, the main contributions of this thesis can be summed up as follows: Grasp synthesis algorithms which extract a family of similar contact-level grasps from a provided prototype and allow to prioritize specified fingers. An open-source C++ library implementing the aforementioned algorithms. Practical applications of contact-level grasp families ranging from grasp qualification to visually guided teleoperation, interactive grasp transfer and finger gait planning. An optimization-based grasp synthesis framework which incorporates heuristics based on human grasp strategies. A grasp execution routine using the active surfaces of a gripping device for inhand manipulation to increase the stability of an initial grasp. A reactive motion generation framework whose output resembles human demonstrations. A control scheme which allows for real-time obstacle avoidance.

Chapter 2

Manipulator Structures dynamic modeling

There is a brief introduction about parameters that define robot as well as homogeneous transformations that transforms the one frames from starting point to end point, to find DH parameter of our robot and find transformation matrices using these DH parameters. The suitable introduction will thereafter be given which describes kinematics and dynamics [12].

2.1 Rigid Body Configuration

Robot in a 3-dimensional space is described by specific configuration. The configuration includes the specific joints of a particular robot. In robotics there are flexible and rigid bodies [13]. Robot is perceived as rigid entirely so our case is odf rigid body. Serial link are connected to each other by joints. Commonly used joints are revolute and prismatic joints , see Fig 2.1.

Typically, robotic manipulators have two or more serial connections, also called links.

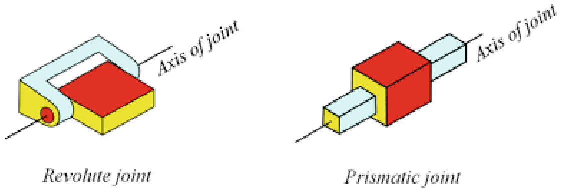


Figure 2.1: Fig A and Fig B shows revolute and prismatic joints [6]

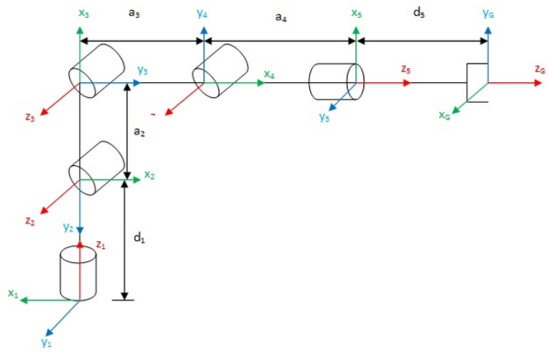


Figure 2.2: Robotic arm with links and reference frames [7].

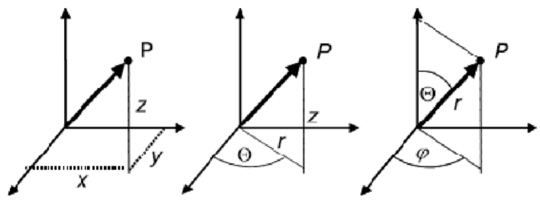


Figure 2.3: A) Cartesian, B) Cylindrical and C) Spherical coordinate systems [8].

There is a mass center, an inertial moment, etc. in each link. As shown in Figure 2.2, it is easier to define a link for a given coordinate system. The coordinate system's x, y, and z axes are represented by the colors green, blue, and red. The modification of letter an illustrates the change along the joint variable. The definition of the joint variables will be discussed in section 2.3. Any connection between two frames may be explained using transformations.

To describe a point in three dimesnsional space there exists many representations. Common representations includes Cartesian, cylindrical and/or spherical coordinate systems, as shown in Figure 2.3. We will be using cartesian coordinate system throught out our thesis.

2.2 Transformations

The associative characteristic of matrix multiplication makes rotation and translation matrices often utilized. The information on how the representation vector p1 of a certain point changes from one frame to the next is provided by the homogeneous transformation matrix, which is generally a 4x4 matrix. A general homogeneous transformation matrix is described as follows:

$$H_l^w = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2.1}$$
$$p^w = H_l^w p^l$$

The rotation matrix R is represented by the first (3 x 3) elements, while the translation T is represented by the first (3 x 3) elements in the last column. The following homogeneous transformations provide the rotations with angles (ϕ), (θ), and (ψ) around the axis (x, y, and z).

$$H_{\mathrm{R}_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$H_{\mathrm{R}_y}(\gamma) = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2.2}$$
$$H_{\mathrm{R}_z}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To calculate position of end-effector we multiply the transformations matrix. The final transformation matrix can be obtained by multiplying each joint transformation matrix with each other.

$$H_e^0 = H_1^0 H_2^1 \dots H_{i-1}^{i-2} H_e^{i-1}$$
$$H_0^e = \left(H_e^0\right)^{-1} \tag{2.2.3}$$

2.3 DH Parameter(Modified)

In [14], four DH parameters that demonstrate the custom of connecting a reference frame to a connection are mentioned. The four transformations that make up the aforementioned conventions are a rotation about z, a translation in z, a rotation about x, and a translation in x. Equations provide the results of these procedures' ultimate transformation (2.3.1).

$$
\begin{aligned}
{}^{n-1}T_n &= \mathrm{Rot}_{x_{n-1}}(\alpha_{n-1}) \cdot \mathrm{Trans}_{x_{n-1}}(a_{n-1}) \cdot \mathrm{Rot}_{z_n}(\theta_n) \cdot \mathrm{Trans}_{z_n}(d_n) \\
{}^{n-1}T_n &= \left[\begin{array}{ccc|c}
\cos\theta_n & -\sin\theta_n & 0 & a_{n-1} \\
\sin\theta_n\cos\alpha_{n-1} & \cos\theta_n\cos\alpha_{n-1} & -\sin\alpha_{n-1} & -d_n\sin\alpha_{n-1} \\
\sin\theta_n\sin\alpha_{n-1} & \cos\theta_n\sin\alpha_{n-1} & \cos\alpha_{n-1} & d_n\cos\alpha_{n-1} \\
\hline
0 & 0 & 0 & 1
\end{array}\right]
\end{aligned} \tag{2.3.1}
$$

- θ_i: Link twist angles from xi-1 to xi measured along zi.
- di : The distance from xi-1 to xi measured along zi.
- α_i : Link twist angles from zi to zi+1 measured along xi .
- ai : The distance from zi to zi+1 measured along xi .

Di or *thetai* can be either a constant or a variable. Di is a variable and *thetai*: is a constant for prismatic joints. On the other hand, di is constant whereas *thetai* is changeable for the revolute joint. Joint angle *theta* is also known as q. Both *thetai* and di would end up being constants for the fixed connection.This standard uses a set of guidelines for selecting the frame's coordinates.

1. When constructing joints, use right-hand frames.
2. Align the Joint i's zi-axis with the rotation or translation axis.
3. Choose the origin Oi at the place where the common normal and axis zi+1 cross.
4. Choose the xi -axes from zi+1 and zi in the same direction as the common normal.

2.4 The robotic manipulator's kinematics

Finding the correlation between a manipulator's joint angles (q) and end-position is the aim of effector's forward kinematics in a robotic manipulator [14]. Using forward kinematics and the kinematic equations, the end-position of the effector in reference to the base of the manipulator is computed. The base is the link that is fixed in the room if the manipulator is installed on a table or workstation; if it is placed on a moving platform, the base is the position of the moving platform.

In other words, when all joint angles (q) are known, it gives the effector's end-position (pe) in the area.These kinematic equations are obtained by computing the transformation matrices (2.3.1) using the DH parameters for each joint frame in the kinematic chain. The location of the end-effector in reference to the base frame is then determined by computing the end-effector to base transformation H0 e using (2.2.3) as indicated in (2.2.1).

$$p_e = \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = H_e^0 p_0 = \begin{bmatrix} R_e(q) & T_e \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.4.1}$$

The pe is a homogeneous vector of the end-effectors position.

2.4.1 Differential Kinematics of Robotic Manipulator

The goal of differential kinematics is to ascertain how the end-effectors' angular and linear velocities relate to the joint velocities. An equation using [14] and [12] (2.4.2).

$$\begin{aligned} \dot{p}_e &= J_{P_e}(q)\dot{q} \\ \omega_e &= J_{O_e}(q)\dot{q} \end{aligned} \tag{2.4.2}$$

Three-by-n matrices called Jp and Jo represent the joint velocity contribution to end-effector linear velocity (p), the angular velocity, and the number of joints (n). You could switch this to

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_{P_e}(q) \\ J_{O_e}(q) \end{bmatrix} \dot{q} = J_e(q)\dot{q} \tag{2.4.3}$$

where J(q) uses the "geometric Jacobian" (6 x n) matrix to represent the system differential kinematics equation. The components of the Jacobian matrix are computed as follows:

$$\begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{for a prismatic joint} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a revolute joint} \end{cases} \tag{2.4.4}$$

where pe is specified in (2.4.1) and zi-1 is taken from the third column of the rotation matrix, R^0_{i-1}.

$$z_{i-1} = R^0_1 \dots R^{i-2}_{i-1} z_0 \qquad \text{where } z_0 \text{ is } \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{2.4.5}$$

Lastly the pi-1 is given by the following equation (2.4.6).

$$p_{i-1} = H^0_1 \dots H^{i-2}_{i-1} p_0 \quad \text{where } p_0 \text{ is } \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.4.6}$$

2.5 Dynamics of Robotic Manipulator

Typically, a rigid-body system is used to model a robot's mechanism. Consequently, rigid-body dynamics can be applied to robot dynamics. Next come the forward- and inverse-dynamics challenges, which are the two fundamental issues.The joint's accelerations are determined by the forward dynamics utilizing the joint angle q, joint velocity q, and torque applied to the actuators τ. Inverse dynamics calculates the required torque (τ) for the actuators using the joint angle q, joint velocity $q^{\cdot}$, and joint accelerations $q^{\cdot}$. from the opposing side.

The following are the advantages of developing a dynamical model of a manipulator: having the capacity to simulate motions, analyse manipulator architectures, and build control algorithms Without using a real system, manipulator motion simulation enables testing of control techniques and motion planning. This is helpful if the real system is unavailable or if any risks are present for the test subjects of the system, the user, or the developer.

2.5.1 Langrange equation

The Lagrange Formulation, a variation approach based on the kinetic and potential energy of the link system, is one way for determining the dynamical model of the system.

Citations for [14] and [12] The source of the Lagrangian is: One method for figuring out the dynamical model of the system is to use the Lagrange Formulation, a variation technique based on the kinetic and potential energy of the link system. Citations for [14] and [12] The Lagrangian's origin is:

$$\mathcal{L}(q,\dot{q}) = \mathcal{T}(q,\dot{q}) - \mathcal{U}(q) \tag{2.5.1}$$

where T is the entire kinetic energy of the system and U is the total potential energy. You may use the formulas to determine the kinetic energy:

$$\mathcal{T}_i = \sum_{i=1}^{n} \frac{1}{2}\dot{p}_i^T m_i \dot{p}_i + \frac{1}{2}\omega_i^T R_i I_{\ell i}^i R_i^T \omega_i \tag{2.5.2}$$

where $p^{\cdot}$ and $omega$, respectively, represent the linear and angular velocities. In the base frame, R_i signifies the rotation matrix from link I frame to the base frame, and $I_l^i i$ specifies the inertia tensor of link I with respect to its center of mass. It might be written like follows:where $p^{\cdot}$ and $omega$, respectively, stand for the linear and angular velocities. In the base frame, $I_l^i i$ denotes the inertia tensor of link I with respect to its center of mass, and Ri denotes the rotation matrix from link I frame to the base frame. It might be formatted as follows:

$$\mathcal{T}(q,\dot{q}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}(q)\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{q}^T B(q)\dot{q} \tag{2.5.3}$$

where

$$\begin{aligned} B(q) &= \sum_{i=1}^{n}\left(m_{l_i} J_p^{(l_i)T} J_p^{(l_i)} + J_o^{(l_i)T} R_i I_{l_i}^i R_i^T J_o^{(l_i)} + m_{m_i} J_p^{(m_i)T} J_p^{(m_i)} + J_o^{(m_i)T} R_{m_i} I_{m_i}^i R_{m_i}^T J_o^{(m_i)}\right) \\ &= \sum_{i=1}^{n}\left(m_i J_p^{(i)T} J_p^{(i)} + J_o^{(i)T} R_i I_i^i R_i^T J_o^{(i)}\right) \end{aligned} \tag{2.5.4}$$

where B(q) is the (n x n)-element symmetric, positive-definite, and often configuration-dependent mass-matrix [14]. Sections 2.4 Jp and Jo of the geometric Jacobian matrix are its respective linear and angular components, and A link's mass and inertia are indicated by the indexing (l), while the motor's mass is indicated by the indexing (m) similar to inertia The combined form, which considers a motor and a connection as a single physiological part, will be employed in this thesis. To calculate the potential energy, you can use:

$$\mathcal{U}(q) = \sum_{i=1}^{n} m_i g_0^T p_{mi} \tag{2.5.5}$$

the link's (icenter) of mass is located at pmi in the world frame and at g0 in the base frame, for example, if z is the vertical axis. Therefore, the Lagrange equation is written as (2.5.6) or a vector by (2.5.7).

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial q_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \quad i = 1, 2, \ldots, n \tag{2.5.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial q}\right)^T = \xi \tag{2.5.7}$$

where qi is the generalised coordinate and xii is the generalised force. This is proven by the contributions of the viscous friction torques and the actuation torque $taui$ at the joint I. An equation might be used to calculate this (2.5.8).

$$\xi_i = \sum_{i=1}^{N_f}\left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_i}\right) + \sum_{i=1}^{N_\tau}\left(\vec{\tau}_i \cdot \frac{\partial \vec{\omega}_i}{\partial \dot{q}_i}\right) \tag{2.5.8}$$

where Nf and N_tau stand for the relative amounts of active non-conservative forces and torques. The derivatives of the lagrangian in Equation (2.5.7) lead to:

$$B(q)\ddot{q} + n(q, \dot{q}) = \xi \tag{2.5.9}$$

ξ and B(q) as mentioned above, and $n(q; q^\cdot)$ is:

$$\begin{aligned} n(q, \dot{q}) &= \dot{B}(q)\dot{q} - \frac{1}{2}\left(\frac{\partial}{\partial q}\left(\dot{q}^T B(q)\dot{q}\right)\right)^T + \left(\frac{\partial \mathcal{U}(q)}{\partial q}\right)^T \\ &= C(q, \dot{q})\dot{q} + G(q) \end{aligned} \tag{2.5.10}$$

The coriolis-matrix with (n x n) elements is known as $C(q; q^\cdot)$. Equations (2.5.11) and (2.5.12) may also be used to calculate this using the mass-matrix (2.5.4). The components of the coriolis matrix are c_{ij}, whereas the components of the mass matrix are b_{xx}.

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k \tag{2.5.11}$$

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right) \tag{2.5.12}$$

Assuming that gravity is the only source of potential energy, the term vector for gravity is G(q). The spring terms, which add to the potential energy if there are springs at the joints, will be added to G(q). Equation may be used to get this (2.5.13).

$$G(q) = \frac{\partial \mathcal{U}(q)}{\partial q} \tag{2.5.13}$$

The final equation of motion, which explains the relationship between the required torque and acceleration, is given by the equation (2.5.14). Equations (2.5.7) and are combined to produce this (2.5.8).

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + F_s\,\mathrm{sgn}(\dot{q}) + G(q) = \tau - J^T(q)h_e \tag{2.5.14}$$

where τ stands for the actuation torques. $F_v q\cdot$ represents the torques of viscous friction. The Coulomb friction torques are represented by $sgn(q)$ and the diagonal matrix Fs is (n x n). The (n x n) diagonal matrix of viscous friction coefficients is abbreviated as Fv. The single joint velocities' sign functions make up the elements of the (n x 1) vector. The end-vector effectors of force and moment acting on the environment are denoted by the symbol he.

The brushes slide on the commutator when the motor shaft spins in its bearings, creating friction torques that may also be affected by external loads. The friction torques are provided by:

$$\tau_{fric} = F_v\dot{q} + F_s\,\mathrm{sgn}(\dot{q}) = K_t I_0 \tag{2.5.15}$$

You can calculate Fv and Fs by running the motor at two different voltages while there is no load.All forces and torques stated above are taken to be equal to zero in this thesis, with the exception of the actuation torques, which represent the intended input to the manipulator. Because we don't know how much of these forces there are, this assumption is made. Therefore, the suggested simplifications were ignored. This led to the motion equation that is depicted below:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{2.5.16}$$

The joint acceleration $(q^{\cdot\cdot})$ from equation (2.5.16), commonly known as the inverse-dynamic equation, is solved to get the forward dynamical equation, which results in:

$$B(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - G(q)) = \ddot{q} \tag{2.5.17}$$

This results in a linear relationship between the actuator torques tau. This function may be controlled via the full-rank matrix B(q), which can be further inverted for any manipulator configuration. Regardless of the reference coordinate frame, the dynamical equation may be purposefully and independently determined using the Lagrange-Formulation approach. Another technique for figuring out the dynamical model of the system is the Newton-Euler Formulation, which applies F = ma to each individual connection of the robot. It defines how the connection will move in the presence of a balanced combination of stresses and forces. As a result, a set of equations with a recursive structure is produced.

Chapter 3

Optimization Techniques For Joint Acceleration Minimization

The Genetic Algorithm (GA) is introduced in this chapter in terms of both theory and implementation. The basic issues with design optimization are addressed first. In general, meta-heuristic techniques are discussed as effective problem-solving tools.

Then, GA is explained along with its foundation, uses, and future research directions.

3.1 Design Optimization and Meta-Heuristic

3.1.1 Design Optimization

Determining the group of design characteristics that will best achieve a specific target is known as design optimization. The topic of design optimization is relevant to many design issues, particularly complex ones. One must, for instance, know the percentage composition of each ingredient when creating a composite material in order to attain the best mechanical properties. To make a transportation system reliable and economical, it is important to consider the linkages between various transportation nodes when constructing the system. To ensure that the manufacturing resources work together in a dependable and effective way, one must select when to deploy which manufacturing resources [15]

Different classifications might be used to group design optimization issues. One method is to categorise them into the two classes of functional and combinatorial optimization.

A continuous or piecewise continuous function of the design parameters can typically be used to formulate the objective function in functional optimization. For instance, a composite material's mechanical property can be a continuous function of the percentage composition of each component. Contrarily, each parameter's potential value in combinatorial optimization is discrete. Different combinations of these discrete factors create a finite number of "states" of the issue, each of which will have a particular impact on the optimization objective. An example of a combinatorial optimization problem is the design of a transportation system.

Rigid mathematical techniques can be used to solve straightforward functional optimization issues. Formal approaches, however, are inadequate when the functions are complex. In certain situations, it is possible to discretize the functional optimization problem so that it becomes a combinatorial optimization problem, which can then be solved using techniques for combinatorial problems. This thesis mostly addresses issues with combinatorial optimization.

Combinatorial issues have two main challenges: 1) the solution space is too big for exhaustive search. 2) the relationship between the design parameters and the optimization aim is not fully understood, making analytical methods ineffective for solving the problem.

3.1.2 Meta-Heuristic

The use of meta-heuristic algorithms to address combinatorial optimization problems is effective and powerful. Formally speaking, meta-heuristic algorithms are iterative generation processes that direct a subordinate heuristic by intelligently fusing many concepts for exploring and utilising the search space. Information is organised using learning processes in order to quickly identify close to ideal answers.

As opposed to deterministic algorithms, meta-heuristics are approximation algorithms. Deterministic algorithms promise to solve every instance of a problem of limited size in bounded time, but they frequently provide computation durations that are too long for real-world use. Meta-heuristics, which are approximate algorithms, give up the assurance of discovering optimal answers in exchange for obtaining good solutions in a noticeably shorter amount of time. Many meta-heuristic search methods rely on probabilistic choices made along the way. The primary distinction between metaheuristic algorithms and pure random search, however, is the way in which randomness is handled in them. In addition to Simulated Annealing (SA), Tabu Search (TS), Neural Networks (NN), and Genetic Algorithms (GA), some popular meta-heuristic techniques include

3.2 Genetic Algorithm (GA)

3.2.1 Rationale

John Holland first suggested the Genetic Algorithm (GA) in 1975, and it is a class of meta-heuristic search and optimization algorithms motivated by Darwin's concept of organic selection The fittest survive is the fundamental tenet of natural selection. By way of organisms adapt as a result of the natural selection process to increase their chances of surviving in a environmental context. The genetic makeup of an organism can change at random, which is then transmitted to its offspring. If a mutation proves beneficial, these kids are more likely to reproduce and survive. The risk of injury to these kids is lower thus, the undesirable feature will perish with them.

Strings of code are used in GA to parametrically express solutions (e.g. binary)[16]. To evaluate solutions, fitness value is defined. A GA's standard operating procedures consist of: Create a population of randomly chosen people; assess each person's fitness; choose people to become parents; produce children; assess children; and then repeat steps 3 through 5 until a solution with satisfied fitness is found or a certain number of generations is reached.

3.2.2 Applications

Since its conception, GA has found use in a wide range of fields. Engineering design practitioners Yao (1992), Joines (1996), and Gold (1998) all employed GA to estimate parameters for nonlinear systems [14], manufacturing cell design, and the kinematic design of turbine blade fixtures. Timothy (1993) used GA to optimise sequencing problems in the planning and scheduling domain, while Davern (1994) used GA to design the job shop scheduling architecture. Rho (1995) utilised GA in the field of computer science when designing distributed databases. Tadikonda (1993) used GA to achieve automatic image segmentation and interpretation; Huang (1998) created face recognition detection algorithms with GA.

Since GA is not problem-specific, it can be used to solve a wide variety of combinatorial optimization problems across many fields and is not constrained by the underlying physical principles of the problems.

3.2.3 General Research Topics

The need to comprehend and improve GA performance led to a great deal of theoretical study on GA. The ability to create effective and reliable GAs is the ultimate objective. To do this, though, it is essential to thoroughly comprehend two key issues: 1. How do GAs function? 2) What kinds of issues are appropriate for GAs to resolve? Generally speaking, these two fundamental concerns form the basis of practically all theoretical work in GA.

The evolution process is modelled mathematically in order to explain how solutions improve over successive generations, which is the first of the two problems. The John Holland should be given credit for the first moderately rigorous model. He progressed Schema Theorem to explain how specific portions of code grow or shrink based on their level of fitness compared to the norm. Although Schema is subject to some criticism Theorem, on which many theoretical investigations of GA are still based.The revision of the theorem is a part of some study on the schema theorem. The GA process is also being modelled as a Markov process. The Markov models try to explain how GAs behave when they converge. Although it is more accurate, this model is typically highly complex and provides nothing in the way of useful advice for creating competent GAs. Building Block Hypothesis was presented by Goldberg on the basis of research into GA's mechanics. The design of GA was then broken down, and he offered various recommendations for creating competent GAs. From there, a number of GA with increased efficiency and/or robustness were designed.

The evolution process is modelled mathematically in order to explain how solutions improve over successive generations, which is the first of the two problems. The John Holland should be given credit for the first moderately rigorous model. He progressed Schema Theorem to explain how specific portions of code grow or shrink based on their level of fitness compared to the norm. Although Schema is subject to some criticism Theorem, on which many theoretical investigations of GA are still based. The revision of the theorem is a part of some study on the schema theorem. The GA process is also being modelled as a Markov process. The Markov models try to explain how GAs behave when they converge. Although it is more accurate, this model is typically highly complex and provides nothing in the way of useful advice for creating competent GAs. Building Block Hypothesis was presented by Goldberg on the basis of research into GA's mechanics. The design of GA was then broken down, and he offered various recommendations for creating competent GAs. From there, a number of GA with increased efficiency and/or robustness were designed.

The evolution process is modelled mathematically in order to explain how solutions improve over successive generations, which is the first of the two problems. The John Holland should be given credit for the first moderately rigorous model. He progressed Schema Theorem to explain how specific portions of code grow or shrink based on their level of fitness compared to the norm. Although Schema is subject to some criticism Theorem, on which many theoretical investigations of GA are still based. The revision of the theorem is a part of some study on the schema theorem. The GA process is also being modelled as a Markov process. The Markov models try to explain how GAs behave when they converge. Although it is more accurate, this model is typically highly complex and provides nothing in the way of useful advice for creating competent GAs. Building Block Hypothesis was presented by Goldberg on the basis of research into GA's mechanics. The design of GA was then broken down, and he offered various recommendations for creating competent GAs. From there, a number of GA with increased efficiency and/or robustness were designed.

The evolution process is modelled mathematically in order to explain how solutions improve over successive generations, which is the first of the two problems. The John Holland should be given credit for the first moderately rigorous model. He progressed Schema Theorem to explain how specific portions of code grow or shrink based on their level of fitness compared to the norm. Although Schema is subject to some criticism Theorem, on which many theoretical investigations of GA are still based. The revision of the theorem is a part of some study on the schema theorem. The GA process is also being modelled as a Markov process. The Markov models try to explain how GAs behave when they converge. Although it is more accurate, this model is typically highly complex and provides nothing in the way of useful advice for creating competent GAs. Building Block Hypothesis was presented by Goldberg on the basis of research into GA's mechanics. The design of GA was then broken down, and he offered various recommendations for creating competent GAs. From there, a number of GA with increased efficiency and/or robustness were designed.

GA doesn't always solve issues well. Understanding the types of problems that GA can solve, or alternatively, what makes it challenging for GA, is crucial. GA[17]. The concept of epistasis is the main concept to answer this query. In a nutshell, epistasis refers to the interdependency of a solution's parameters, which results in Nonlinearity makes the issue challenging for GA. Davidor suggested a strategy to epistasis measurement. Vose and Liepins demonstrated that, in theory, any problem's epistasis may be minimised using a variety of encoding techniques. However, given creating such a coding scheme might be difficult and make things more complex.

3.3 Pattern Search Algorithm

The constraint criteria in the first and an equidistant particle mesh generated between them in the second make up the two main parts of the pattern search approach. The evaluation of the particles in the function gives the guidelines for determining which direction the particles must specify by offering already chosen leaps to start their search for the whole work area

One advantage of the development of an optimization algorithm like PSM is that its fundamental feature is global convergence, which suggests that it does not produce.Since it does a comprehensive search during the search, there is less local stagnation range [18]. PSM also has the advantage of being simple to adopt, which arithmetic computations, which make implementation really easy. As an example, it can be made aware of the applications of PSM in mathematics and optimization theory.The family of optimization methods includes the PSM pattern search methodology. a metaheuristic algorithm[19]. This method makes use of a restricted search and an array performed.The aforementioned matrix is coupled to a number of survey criteria and is referred to as a mesh. The survey-generated conditions give instructions for reducing the current mesh, ensuring algorithm convergence. It is designed to be entirely compatible with NMPC techniques like those covered in this study and to provide a pleasant operation for applications with several local minimums [31]. The following enhancements were discovered as a result of using the method:

- The PSM improves the search for local minima to make the tuning more accurate values;
- The acquired results while tuning the models produced from the SIB T-S with CBMD and PSM increase in comparison to other authors;

By fine-tuning the control methods used to non-linear systems, the results from NMPC indicate outcomes with lower steady-state error than those from the majority of other writers.

One advantage of the development of an optimization algorithm like PSM is that its fundamental feature is global convergence, which suggests that it does not produce.Since it does a comprehensive search during the search, there is less local stagnation range. PSM also has the advantage of being simple to adopt, which arithmetic computations, which make implementation really easy. As an example, it can be made aware of the applications of PSM in mathematics and optimization theory[20]. The family of optimization methods includes the PSM pattern search methodology a metaheuristic algorithm.This method makes use of a restricted search and an array performed. The aforementioned matrix is coupled to a number of survey criteria and is referred to as a mesh. The survey-generated conditions give instructions for reducing the current mesh, ensuring algorithm convergence. It is designed to be entirely compatible with NMPC techniques like those covered in this study and to provide a pleasant operation for applications with several local minimums. The following enhancements were discovered as a result of using the method:

3.4 Hybrid Algorithm

A novel approach is introduced which used fitness values of both genetic algorithm and pattern search algorithm to find its own fitness value which is more robust for changing search spaces as comapared to genetic and pattern search algorithms. In real life it is generally observed that pattern search algorithm is very efficient for small search space whereas genetic algorithm is efficient in large search space.In hybrid algorithm fitness value is calculated by taking average of fitness value of genetic algorithm and pattern search algorithm. The block diagram of algorithm is shown in Fig 3.1.

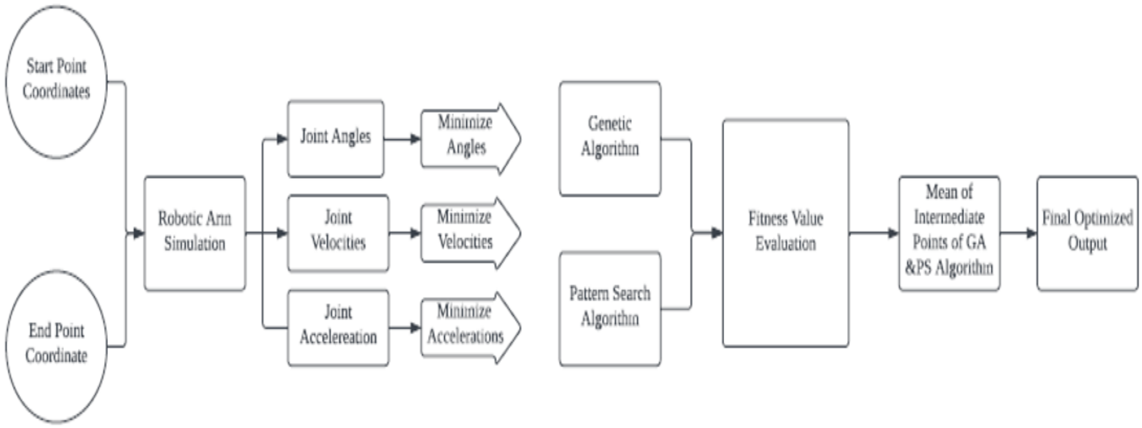


Figure 3.1: Block Diagram of Hybrid Algorithm

Chapter 4

Simualtion Results

Peter Corke toolbox of matlab is used to simulate the customized robot. Joint acceleration are minimized using genetic algorithm and pattern search algorithm.Firstly unoptimized trajectory is generated using jtraj command. Joint space trajectory is generated under constrained environment as shown in figure 4.1, 4.2 and 4.3.

Secondly genetic algorithm is applied to do optimization under constrained environment defined for a robotic manipulator. Trajectory is optimized in term of distance and global minimas are found. Then optimized trajectory is generated as shown in figure 4.4,4.5 and 4.6.

Lastly pattern search algorithm is applied and local minimas are found and optimized trajectory is generated along with joint accelerations as shown in figures 4.7,4.8 and 4.9.

Lastly hybrid algorithm which is fusion of genetic algorithm and pattern search algo-

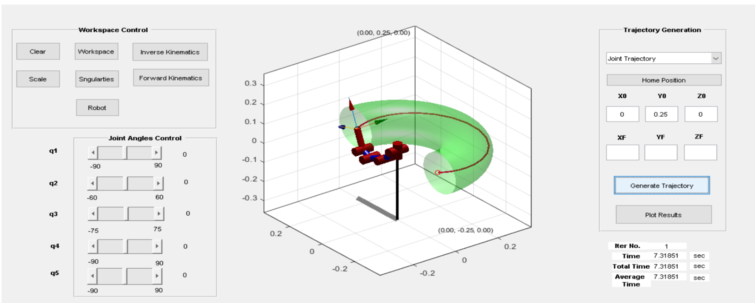
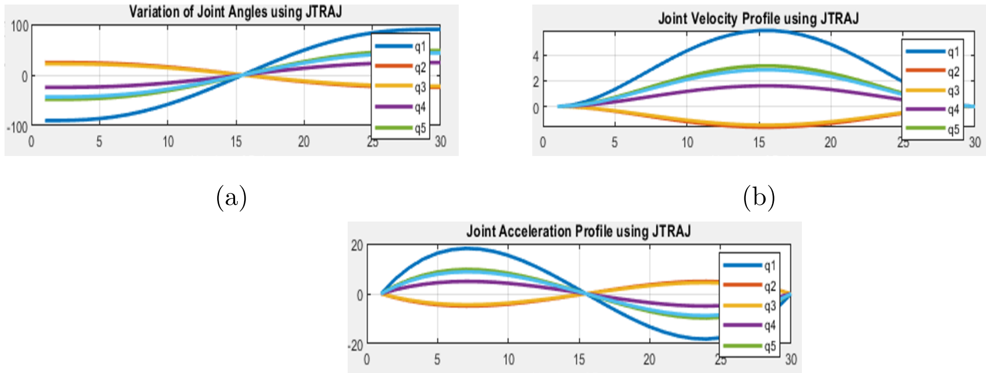


Figure 4.1: GUI of Customized Robot in Peter Corke [9] For Unoptimized Trajectory Note:As peter corke toolbox is used it may look similar to previous published work but parameters are different



(a) (b)

(c)

Figure 4.2: Unoptimized Trajectory. (a) Variation in joint angles.(b) Variation in joint velocities.(c) Variation in joint accelerations.

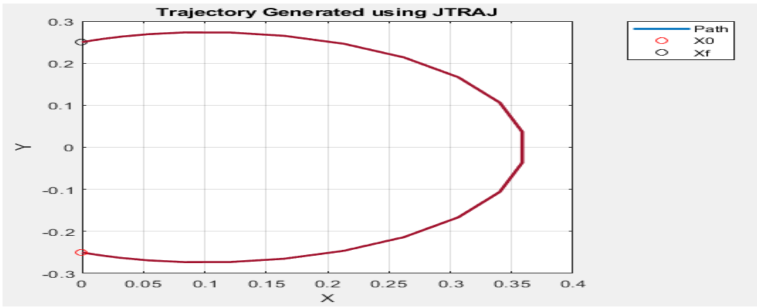


Figure 4.3: Unoptimized Trajectory plotted in matlab

rithm is applied to do optimization under constrained environment defined for a robotic manipulator. Trajectory is optimized in term of distance and global minimas are found. Then optimized trajectory is generated as shown in figure 4.10,4.11 and 4.12.

4.1 Comparison Of Hybrid Algorithm With Genetic and Pattern Search Algorithm

Following are the steps followed to do comparisons in matlab

- Initial and final point are given to the algorithm
- Intermediate points under constrained environment are calculated using genetic and pattern search algorithm.
- Trajectory is generated.
- Joint angles ,velocities and accelerations are calculated using and plotted in matlab.

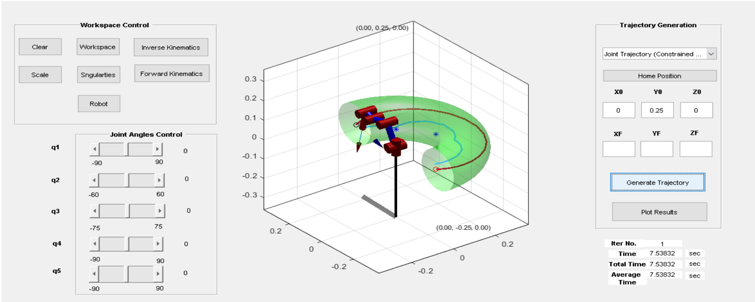
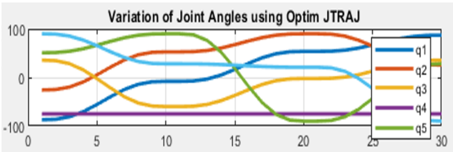
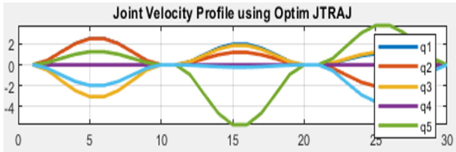


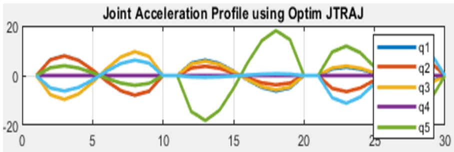
Figure 4.4: GUI of Customized Robot in Peter Corke [9] For optimized Trajectory using Genetic Algorithm Note:As peter corke toolbox is used it may look similar to previous published work but parameters are different



(a)



(b)



(c)

Figure 4.5: Optimized trajectory using genetic algorithm. (a) Variation in joint angles.(b) Variation in joint velocities.(c) Variation in joint accelerations.

- Comparison is made between hybrid algorithm ,genetic algorithm and pattern search algorithm.

Meta-Heuristic Technique	Time(sec)	Joint Ang Var	Joint Vel(deg/s)	Joint Accel(deg/s^2)
Without Optimization	7.31	10	4	20
Genetic Algorithm	7.53	8	3	18
Pattern Search Algorithm	7.85	7	2	15
Hybrid Algorithm	7.27	5	2	12

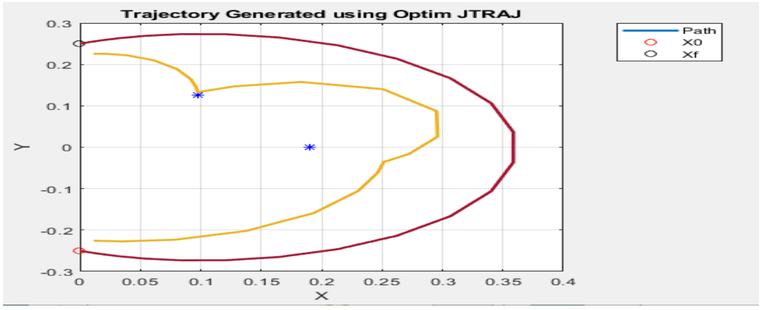


Figure 4.6: Optimized Trajectory using Genetic Algorithm plotted in matlab

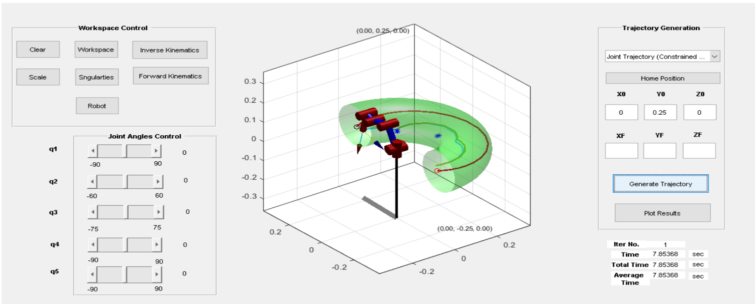
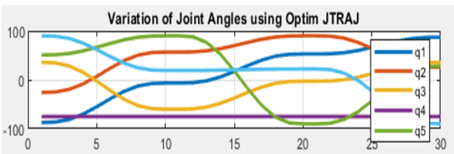
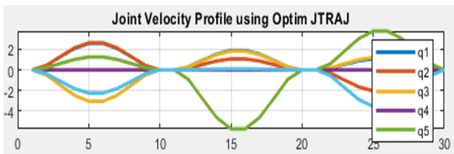


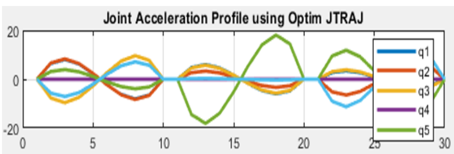
Figure 4.7: GUI of Customized Robot in Peter Corke [9] For optimized Trajectory using Pattern Search Note:As peter corke toolbox is used it may look similar to previous published work but parameters are different



(a)



(b)



(c)

Figure 4.8: Optimized Trajectory using pattern search algorithm. (a) Variation in joint angles.(b) Variation in joint velocities.(c) Variation in joint accelerations.

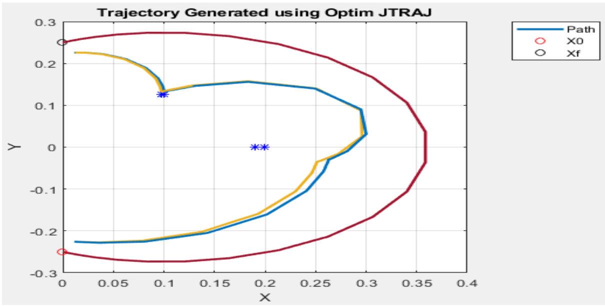


Figure 4.9: Optimized Trajectory using Pattern Search plotted in matlab

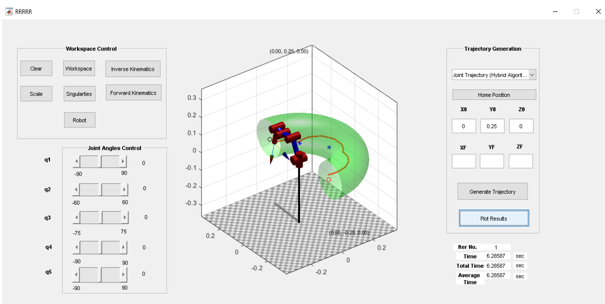
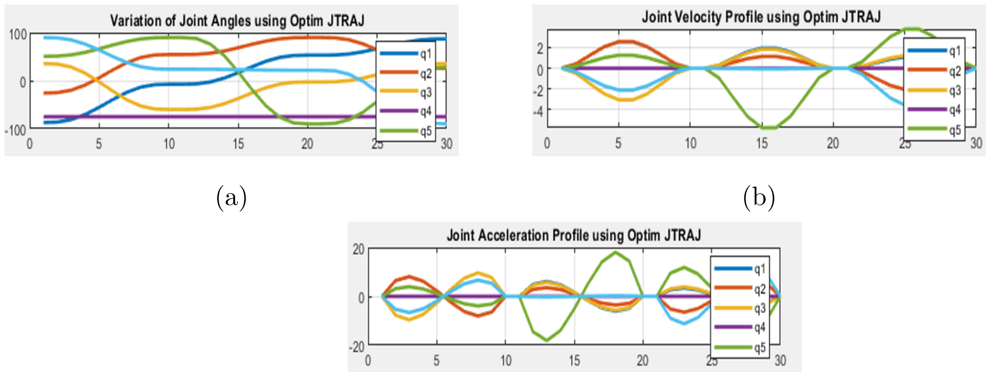


Figure 4.10: GUI of Customized Robot in Peter Corke [9] For Optimized Trajectory using Hybrid Algorithm Note:As peter corke toolbox is used it may look similar to previous published work but parameters are different



(a) (b)

(c)

Figure 4.11: Optimized Trajectory using hybrid algorithm. (a) Variation in joint angles.(b) Variation in joint velocities.(c) Variation in joint accelerations.

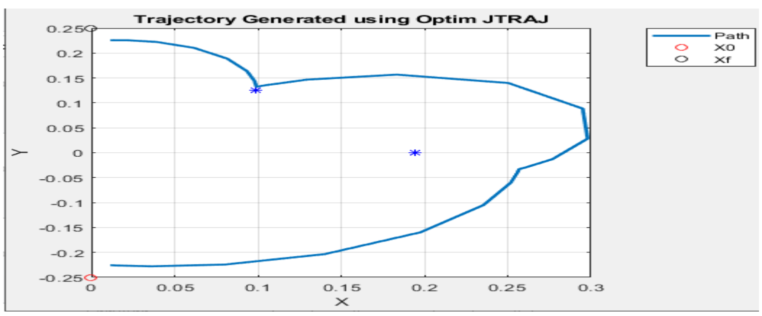


Figure 4.12: Optimized Trajectory Using Hybrid Algorithm plotted in matlab

Chapter 5

Conclusion

After comparing joint velocities and acceleration in matlab under defined constrained environment following conclusions are formed

- Overall 2 percent increase in time is achieved by optimizing trajectory with hybrid algorithm as compared to genetic algorithm and pattern search algorithm.
- Joint angles, velocities and acceleration variations are minimized by 2-3 percent as compared to genetic algorithm and pattern search algorithm.
- Joint accelerations are minimized by 2-3 percent more as compared to genetic algorithm and pattern search algorithm.

References

[1] Steve C Jacobsen, John E Wood, DF Knutti, and Klaus B Biggers. The utah/mit dextrous hand: Work in progress. *The International Journal of Robotics Research*, 3(4):21–50, 1984.

[2] Markus Grebenstein, Alin Albu-Schäffer, Thomas Bahls, Maxime Chalon, Oliver Eiberger, Werner Friedl, Robin Gruber, Sami Haddadin, Ulrich Hagn, Robert Haslinger, et al. The dlr hand arm system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3175–3182. IEEE, 2011.

[3] William Townsend. The barretthand grasper–programmably flexible part handling and assembly. *Industrial Robot: an international journal*, 2000.

[4] Akio Namiki, Yoshiro Imai, Masatoshi Ishikawa, and Makoto Kaneko. Development of a high-speed multifingered hand system and its application to catching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2666–2671. IEEE, 2003.

[5] Aaron M Dollar and Robert D Howe. The highly adaptive sdm hand: Design and performance evaluation. *The international journal of robotics research*, 29(5): 585–597, 2010.

[6] Reza N Jazar. *Theory of applied robotics*. Springer, 2010.

[7] Abdullah Titi. Can anyone help me find the inverse kinematic solution of a 5 dof robot arm?, 12 2014.

[8] Agneta Wikman Larhed, Per Artursson, Johan Gråsjö, and Erik Björk. Diffusion of drugs in native and purified gastrointestinal mucus. *Journal of pharmaceutical sciences*, 86(6):660–665, 1997.

[9] Peter I Corke. A robotics toolbox for matlab. *IEEE Robotics & Automation Magazine*, 3(1):24–32, 1996.

[10] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[11] Matthew T Mason and J Kenneth Salisbury Jr. Robot hands and the mechanics of manipulation. 1985.

[12] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.

[13] Raphael Deimel and Oliver Brock. A compliant hand based on a novel pneumatic actuator. In *2013 IEEE International Conference on Robotics and Automation*, pages 2047–2053. IEEE, 2013.

[14] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Force control*. Springer, 2009.

[15] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2-4):311–338, 2000.

[16] Chivukula A Murthy and Nirmalya Chowdhury. In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters*, 17(8):825–832, 1996.

[17] Robert Hooke and Terry A Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

[18] Charles Audet and John E Dennis Jr. Analysis of generalized pattern searches. *SIAM Journal on optimization*, 13(3):889–903, 2002.

[19] Elijah Polak and Michael Wetter. Generalized pattern search algorithms with adaptive precision function evaluations. 2003.

[20] Khaled S Al-Sultan. A tabu search approach to the clustering problem. *Pattern recognition*, 28(9):1443–1451, 1995.