JAUME I UNIVERSITY

MASTER'S THESIS

Kinematic Control of Redundant Manipulators

Author:
Dipendra SUBEDI

Supervisors:
Dr. Enric CERVERA
Prof. Giuseppe CASALINO

A thesis submitted in fulfillment of the requirements
for the degree of European Master on Advanced Robotics Plus (EMARO+)

in the

Robotic Intelligence Laboratory
Department of Computer Science and Engineering

July 7, 2017

Declaration of Authorship

I, Dipendra SUBEDI, declare that this thesis titled, “Kinematic Control of Redundant Manipulators” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

"A robot may not injure a human being, or, through inaction, allow a human being to come to harm."

Isaac Asimov

Jaume I University

Abstract

School of Technology and Experimental Sciences
Department of Computer Science and Engineering

European Master on Advanced Robotics Plus (EMARO+)

Kinematic Control of Redundant Manipulators

by Dipendra SUBEDI

While most manipulators have enough degrees of freedom (DOFs) to perform tasks in their end-effector task space (desired position and orientation), their workspace is limited due to mechanical joint constraints and obstacles that may be present in the work space. Redundant manipulators have extra DOFs than required for reaching the desired position and orientation of the end-effector. This allows the redundant manipulators to use the extra DOFs to avoid their joint limits and obstacles in the workspace, while still reaching a desired end-effector pose in the task space.

The objective of the thesis is to implement and analyze the performance of most common methods for redundancy resolution with respect to algorithmic and kinematic singularity, and incorporation of different additional desired tasks, e.g., joint limit avoidance, singularity avoidance. Moreover, wrist mounted force/torque sensor is calibrated and used to detect the contact with the objects in the environment and for the robust extraction of the object during picking and stowing operations. The thesis is a part of work done for Amazon Robotics Challenge 2017.

Acknowledgements

I would first like to thank my thesis supervisor Dr. Enric Cervera of the Department of Computer Science and Engineering at Jaume I University. The door to Dr. Cervera's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to acknowledge Prof. Giuseppe Casalino of Department of Informatics, Bioengineering, Robotics, and System Engineering at University of Genoa, as the co-supervisor of this thesis, and I am gratefully indebted to him for his very valuable comments on this thesis.

I would also like to thank Prof. Angel P. del Pobil of Department of Computer Science and Engineering as the director of the Robotics Intelligence Laboratory at Jaume I University for providing me an opportunity to join his team for Amazon Robotics Challenge 2017, and who gave access to the laboratory and research facilities. Without his precious support it would not be possible to conduct this research.

Finally, I must express my very profound gratitude to my parents and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. In particular, I am grateful to my fellow lab-mates Mr. Angel J. Durán for enlightening me the first glance of research and providing me useful suggestions throughout my research. Thank you.

Contents

List of Figures

List of Tables

List of Abbreviations

APC	**A**mazon **P**icking **C**hallenge
ARC	**A**mazon **R**obotics **C**hallenge
CAN	**C**ontroller **A**rea **N**etwork
CLIK	**C**losed **L**oop **I**nverse **K**inematics
DH	**D**enavit-**H**artenberg
DOF	**D**egree **O**f **F**reedom
EE	**E**nd-**E**ffector
F/T	**Force**/**Torque** (Sensor)
IK	**I**nverse **K**inematics
IP	**I**nternet **P**rotocol
JLA	**J**oint **L**imit **A**voidance
KDL	**K**inematics and **D**ynamics **L**ibrary
PD	**P**roportional-**D**erivative
PyKDL	**Py**thon **K**inematics and **D**ynamics **L**ibrary
RoboCup	**Robo**t (Soccer World) **Cup**
ROS	**R**obot **O**perating **S**ystem
SVD	**S**ingular **V**alue **D**ecomposition

List of Symbols

e	vector of position and orientation error in 3-D space
$\dot{e}$	vector of derivative of position and orientation error in 3-D space
F_x	force corresponding to x-axis
F_y	force corresponding to y-axis
F_z	force corresponding to z-axis
I	identity matrix $n \times n$
J_e	Jacobian of end-effector
$J_e^\dagger$	pseudo-inverse of Jacobian matrix J_e
J_e^T	transpose of Jacobian matrix J_e
K	diagonal weighting matrix ($n \times n$)
k_0	positive scalar
K_p	proportional gain
m	Cartesian degrees of freedom (degrees of freedom in task space)
n	number of joints (degrees of freedom in joint space)
$\mathbb{N}$	null space
q	vector of joint positions
$\dot{q}$	vector of joint variables (velocities)
$\dot{q}^d$	vector of desired joint velocities
q_0	vector of initial joint configuration
q_i	position of i^{th} joint
$\overline{q}_i$	middle value of joint range of i^{th} joint
q_{im}	minimum joint limit of i^{th} joint
q_{iM}	maximum joint limit of i^{th} joint
$\dot{q}_N$	vector of joint variables (velocities) belonging to null space
$\dot{q}_p$	vector of joint velocities corresponding to main (primary) task
r	degree of redundancy
R	revolute joint
$\mathbb{R}$	range
$\mathbb{R}^m$	Cartesian (task) space
$\mathbb{R}^n$	joint space
S	spherical joint
T_x	torque corresponding to x-axis
T_y	torque corresponding to y-axis
T_z	torque corresponding to z-axis
U	universal joint
x	vector of position and orientation in 3-D space
x^d	vector of desired end-effector position and orientation in 3-D space
$\dot{x}$	vector of end-effector linear and angular velocities
$\dot{x}^d$	vector of desired end-effector linear and angular velocities
α	scalar multiplier
$\dot{\varphi}$	arbitrary n-dimensional vector
$\Phi(q)$	scalar objective function of the joint variable q

λ	damping or singularity robustness factor (number)

Dedicated to my parents

Chapter 1

Introduction

At present, robots play an active role in many fields like industrial use, use in hazardous environments, use for collaboration in human society and so on. Robotic manipulator based automation has gained increasing deployment in the industry. Industrial robots are used but not limited to material handling, machine tending, arc welding, spot welding, cutting, painting and gluing. They are capable of lifting massive objects, moving with high speed and repeating complex performances with very high precision. Typically, in industrial application, a robotic manipulator performs a repetitive sequence of movements. A robot task generally consists of a sequence of end-effector movements. It can be defined by a path consisting of a set of robot positions either in joint positions or end effector position and the corresponding set of motion definitions between adjacent robot positions. The position and orientation of the robot's end-effector are not directly measured but instead computed using the individual joint position readings and kinematics of the robot. On the contrary, inverse kinematics of the robot is used to obtain the joint positions required for desired end-effector position and orientation.

Generally, robot manipulators can be categorized into two, one that is designed for general purpose and and the other for exclusive (specific) work. General purpose manipulator is generally designed with more than 6 degrees of freedom (DOF) to allow free positioning and posturing of end-effector and to provide capabilities to handle more real-time applications. More the DOF, the manipulator can handle tasks that are more complicated. However, their structures and control problem become more complicated with the difficulty of the task. Kinematic redundancy occurs if a manipulator has more DOF than the minimum required to execute a given task. Although it was once claimed that a 6-DOF nonredundant manipulator is a general purpose manipulator, since it can freely position and orient its end-effector in the Cartesian workspace, it can no longer be considered a general purpose manipulator [47]. This is because, the geometry of 6-DOF manipulator has a number of kinematic flaws such as limited joint ranges, workspace obstruction and kinematic singularities which prevent it from attaining arbitrarily assigned end-effector position and orientation in its workspace. 7-DOF human arm posses an excellent model of a dexterous redundant structure to overcome the limitations discussed above with its additional degrees of freedom.The exclusive use manipulators are designed specifically with structures for tasks that can be accomplished even in the presence of restrictions on the posture and working range like 5-DOF robotic arm used for welding applications.

Robot redundancy occurs with respect to the given task when the number of active joints exceeds the number of variables which define the task. For example, 6 DOF manipulator becomes redundant with respect to all the five-dimensional tasks, such as, arc welding, laser cutting, which do not require sixth roll angle specifications. Recently, the scientific and technological perspective have been greatly enhanced by considering the redundancy of manipulator which provide greater flexibility,versatility, and dexterity.

1.1 Redundancy Resolution for Redundant Manipulator

There are infinite solution for the inverse kinematic problem of a redundant manipulator. Redundancy resolution means selecting particular solution out of many to accomplish the desired end-effector task. The solution can be selected in such a way that the assigned task requirements can be satisfied with the best posture of the manipulator. This can be done by inducing a self-motion of the structure without changing the position and orientation of the end-effector. In principle, if a desired task trajectory is given to the end-effector, it is possible to assign the joint motion in such a way that both the end-effector task and a suitable constraint task are accomplished at the best.

Researchers have suggested a number of solution for solving the inverse kinematic problem for redundant manipulators. Redundant manipulators have been more exploited for avoiding joint limits [1] and obstacles [60, 39]. However, extra-DOF of the redundant manipulators can also be used to satisfy any desired kinematic or dynamic characteristics [36, 24]. The objective of this thesis is to implement and analyze the performance of exact and approximate redundancy resolution techniques to satisfy the additional desired tasks (singularity avoidance and joint limit avoidance) for the specific application.

1.1.1 Baxter

The Baxter Robot System is human-sized humanoid robot with two 7-DOF arms manufactured by Rethink Robotics.[1] It also contains stationary pedestal, torso, 2-DOF head, vision system, robot control system, safety system, gravity-offload controller and collision detection routine where last two are optional.

The Baxter arms are kinematically redundant, i.e. possessing more joint freedoms than necessary to operate fully in the desired Cartesian space. More specifically, Baxter has $n = 7$ single-DOF revolute (**R**) joints which means that it has one-DOF greater than $m = 6$ Cartesian DOF (3 translations and 3 rotations) required for general trajectories in the task space. As $n > m$, the Baxter qualifies as a kinematically-redundant robot, and thus in addition to following the desired trajectories in the task space, the 7-DOF arm can be used for optimizing the performance by using some constraints in the robot arm. Each Baxter arm has a 2-DOF (offset-**U**-joint) shoulder joint, a 2-DOF (offset-**U**-joint) elbow joint and a 3-DOF (offset-**S**-joint) wrist joint as shown in Figure 1.1. Associated Denavit-Hartenberg (DH) parameters for the 7-DOF left and right arms are summarized in Appendix A. Joint motions corresponding to the individual joints of each Baxter arms are summarized in Table 1.1 and the range of joint motion (joint limits) for each joint is presented in Table 1.2.

According to Pieper [40], if three adjacent joint axes are parallel to one another or they intersect at a single point, a closed-form joint solution to a robot manipulator is guaranteed to exist for coupled nonlinear inverse kinematics problem. This does not occur for Baxter, i.e. there are no parallel **R** joint axes anywhere in each arm nor a

[1] More details and technical specifications can be found in `http://www.rethinkrobotics.com/baxter/`

TABLE 1.1: Baxter Joint Motions

Joint Name	Joint Motion
S_o	shoulder roll
S_1	shoulder pitch
E_o	elbow roll
E_1	elbow pitch
W_o	wrist roll
W_1	wrist pitch
W_2	wrist roll

TABLE 1.2: Baxter Joint Range

Joint Name	Min Limit(Radians)	Max Limit(Radians)	Range(Radians)
S_o	-1.7016	+1.7016	3.4033
S_1	-2.147	+1.047	3.194
E_o	-3.0541	+3.0541	6.1083
E_1	-0.05	+2.618	2.67
W_o	-3.059	+3.059	6.117
W_1	-1.5707	+2.094	3.6647
W_2	-3.059	+3.059	6.117

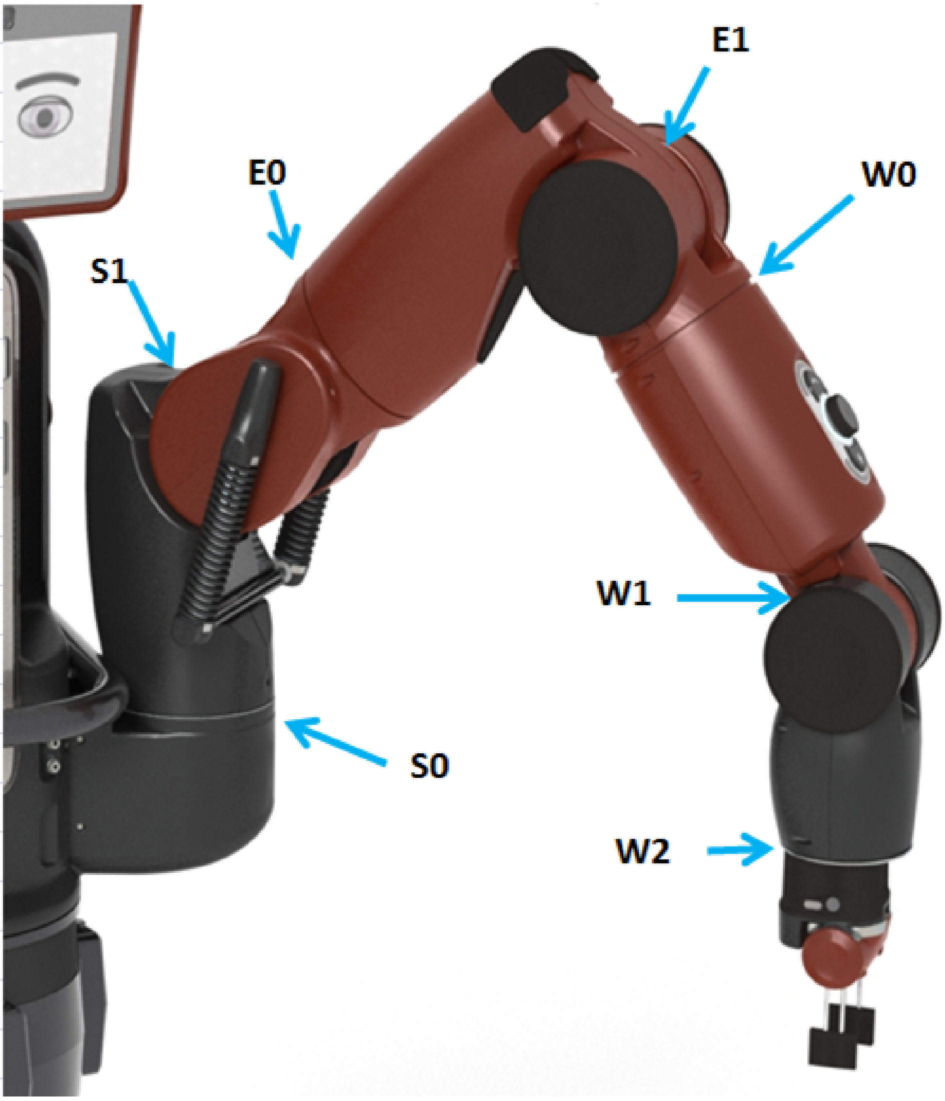


FIGURE 1.1: Baxter Left Arm Joints[2](S=Shoulder, E=Elbow, W=Wrist)

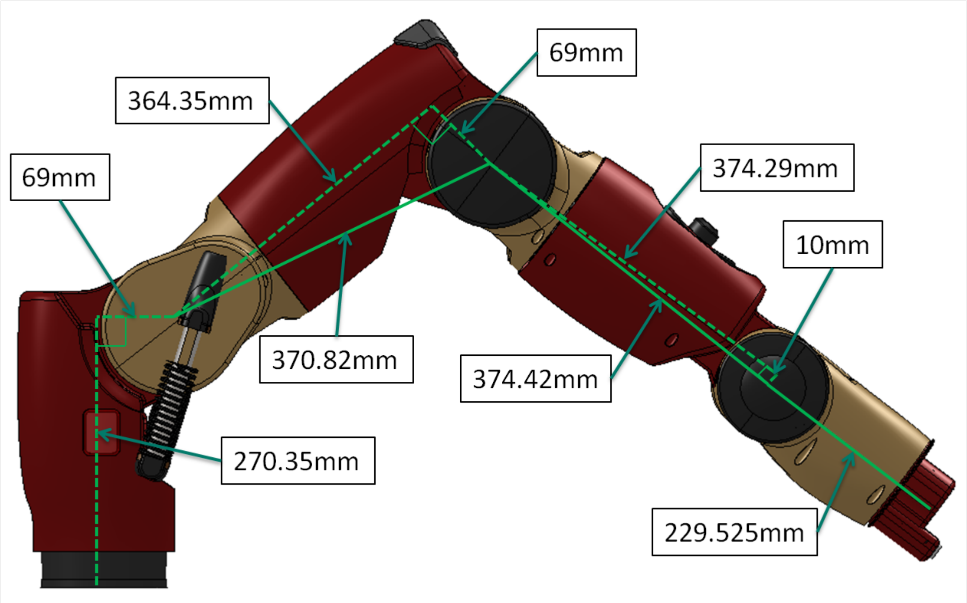


FIGURE 1.2: Link- and Offset- lengths for Baxter's joints[2]

series of three adjacent **R** joints sharing a common origin. Because of these reasons, there are significant complications to the inverse kinematic problem for Baxter arms. Due to the arm offsets, as shown in Figure 1.2, there is no closed-form analytical solution to the inverse kinematic problem for Baxter arm. A numerical iterative method is thus required to solve the inverse kinematics problem but is a challenge for real-time control.

An attractive alternative method for control of redundant robot manipulator is by resolved joint rate control method based on inverse velocity solution which uses a linear set of equations which can easily be solved in a control loop at real-time rates. The work in this thesis includes exploration of detail methods of Cartesian velocity control of redundant arms using redundancy resolution techniques which are discussed in Chapter 3.

1.2 Amazon Robotics Challenge (ARC) 2017

The first *Amazon Robotics Challenge*[3] *(ARC)*, then called the *Amazon Picking Challenge (APC)*, was held at the 2015 International Conference on Robotics and Automation in Seattle Washington, May 26-27. The second APC was held at RoboCup 2016 in Leipzig, Germany from June 30 to July 3. It featured two tasks: a picking task to remove 12 specific items from an Amazon Robotics shelf and place them into a tote, and a stowing task to move 12 items from a tote and place them into a partially full shelf. The 2017 ARC will be held in conjunction with RoboCup 2017 in Nagoya, Japan, July 27-30 and will feature enhanced versions of the *Pick* and *Stow* tasks. Some of the changes from previous challenges include introducing new items at the challenge event and placing picked items into Amazon boxes. The 2017 ARC is a skill challenge sponsored by Amazon Robotics LLC to strengthen the ties between the industrial and academic robotic communities and promote shared and open solutions to some of the big problems in unstructured automation.The Challenge combines

[2] Source: `http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications`

[3] More details about the Challenge can be found at `https://www.amazonrobotics.com/#/roboticschallenge`

object recognition, pose recognition, grasp planning, compliant manipulation, motion planning, task planning, task execution, and error detection and recovery.

Perception, motion planning, grasping, and robotic system engineering have reached a level of maturity that makes it possible to explore automating simple warehouse tasks in semistructured environments that involve high-mix, low-volume picking applications [9]. However, commercially viable automated picking and stowing in unstructured environments by a robot arm remains a difficult challenge. The objective of the challenge is to address the problem of automating the picking and stowing process in online shopping warehouses.

The Challenge consists of three tasks:

- A pick task to remove target items from storage and place them into boxes,
- A stow task to take target items from totes and place them into storage, and
- A Final Round task where all items are first stowed and then selected items are picked into boxes.

The thesis is a part of work done for Amazon Robotics Challenge 2017. It includes kinematic control of Baxter arms using different redundancy resolution techniques at velocity level and selecting the best control scheme for the challenge after implementation and evaluation of different control (redundancy resolution) algorithms.

1.3 Motivations

In context to the ARC 2017, the robots are scored by how many items are picked and stowed in a fixed amount of time. The time consumed for picking and placing an item from the bin to the box and from the tote to the bin, is the critical part of the challenge. This is because, lesser the time consumed per item, more items can be picked and stowed in fixed amount of time provided in the challenge. Based on the experience gained by the UJI RobInLab team[4] in APC 2015, the time needed for motion planning and execution has to be reduced in order to score high points during the challenge. Use of *MoveIt!* motion planning framework consumes more time to move the Baxter arm from initial pose to a goal pose. This is because the MoveIt! framework requires additional time to generate, optimize and execute the path from initial pose to a goal pose. This additional time can be reduced by using the kinematic controller. However, the obstacle avoidance problem should be addressed separately within the kinematic control, which in contrast, can be incorporated in a single planning problem in MoveIt!. Therefore, the obstacle avoidance problem has been minimized during the design of the robotic system (see Chapter 4) for the challenge. Based on these considerations, the kinematic controller is used for picking and stowing operations of the challenge.

The gripper of the manipulator (Baxter) must be able to grasp the object precisely for picking and stowing operations of the challenge. The pose of an object to be grasped, as estimated by the vision system, might possess some errors. Because of this, the object might not be grasped properly even if the gripper reaches the estimated pose. Thus, to compensate the errors in the vision system and ensure robust grasping of the object (through contact detection), the force information of the wrist mounted force/torque (F/T) sensor is used.

[4] The team from the Robotic Intelligence Laboratory, Jaume I University for Amazon Robotics Challenge 2017

1.4 Objectives

Researchers have proposed a number of algorithms for the kinematic control of redundant manipulators. Each of the approaches has its own advantages and limitations. The main objective of this work is to implement and analyze the performance of most common (exact and approximate solution) methods of redundancy resolutions to select the best algorithm of kinematic control of Baxter arms with respect to the challenge. The objectives also include the consideration and analysis of algorithmic and kinematic singularities of the control algorithms. Singularity avoidance and joint limit avoidance are other important objectives of this work. Use of force information for contact detection and robust extraction of objects in picking and stowing operations of the challenge is another goal of this work.

1.5 Contributions and Summary

The thesis is a part of work done for the 2017 ARC. Although, the challenge includes object recognition, grasp planning, task planning, task execution, error detection and recovery; the work done in this thesis is focused on the motion planning and control of the Baxter arms for robust picking and stowing operations in the challenge. The contributions of this thesis work include implementation of the robust low-level kinematic controller for reaching the grasp pose and place pose while avoiding the joint limits and singularities during the motion.

Exact(pseudo-inverse) and approximate(damped least-squares) solution methods of redundancy resolution are considered for the kinematic control of the Baxter arms. Both the methods are implemented and analyzed with respect to kinematic and algorithmic singularities, and incorporation of joint limit avoidance and singularity avoidance tasks. Approximate solution method with joint limit avoidance by kinematic optimization is finally selected for the control of Baxter arms for the challenge. Additionally, the wrist mounted F/T sensor is first calibrated to compensate the weight of the gripper. The force information is then used to detect contact with the object (item) in the environment ensuring robust picking and placing operations.

1.6 Organization of Thesis

The thesis is organized as follows. Chapter 2 reviews the related works in inverse kinematic control of redundant manipulator, and discusses their advantages and shortcomings in general. In Chapter 3, the redundancy resolution problem of a redundant manipulator is stated and common approaches for solving redundancy are described. In Chapter 4, the experimental setup carried out for the project is described and results are presented and analyzed. Chapter 5 describes the conclusions and future works.

Chapter 2

Related Works

Kinematics is considered as the fundamental part of the study of the motion of any manipulator. Manipulator Kinematics consists of forward kinematics and inverse kinematics where the former refers to the computation of end-effector's pose (position and orientation) using the joint variables, whereas the latter refers to the use of end-effector's position and orientation to determine the joint variables of the robotic manipulator [10]. For a redundant manipulator, inverse kinematics solution is more challenging compared to the forward kinematics as it has multiple solutions [46, 15].

A redundant robotic manipulator is characterized by having a greater number of degrees of freedom (DOF) than required to accomplish certain tasks. In order to deal with the real life complexities, the redundant systems have been designed to achieve more dexterous motions.

The extra DOFs of the redundant manipulator can be exploited to satisfy various additional constraints such as singularity avoidance, joint limit avoidance and obstacle avoidance on the kinematic control problem in order to achieve greater manipulability in terms of arm configurations and interaction with the environment [30, 22, 3]. Researchers have proposed a number of solution techniques for solving the kinematic control problem for redundant manipulators. The additional constraints can be satisfied by optimizing a suitable scalar objective function in the null space of the Jacobian matrix based on the requirements.

Suppose the robot is required to move in a cluttered environment, then it is more important (required) to avoid the obstacles [30] and mechanical joint limits [27]. In another case, like considering energy efficiency, the interest could be minimizing the power consumption of joint actuators of the manipulator [52].

In a different point of view, redundancy is purposely adopted in manipulator to avoid kinematic singularities. In the case of the kinematic singularities at some configuration, the manipulator loses its ability to move along or rotate about some direction of the task space, meaning that its manipulability is reduced. Yoshikawa [62] defined the manipulability measure as $\sqrt{det(J_e J_e^T)}$, and as proposed by Klein and Blaho [21], the dexterity measures, i.e., the matrix condition number and the minimum singular value of the matrix $J_e J_e^T$, represent indices of the ability of a manipulator to arbitrarily position and orient its end-effector. Moreover, the dynamic manipulability measure as proposed by Yoshikawa [61] instead, takes the arm dynamics into account. In addition to these measures, Chiu [7] introduced the concept of task compatibility where the matrix $J_e J_e^T$ is used to compute quantitative indices of the ability to perform an exertion/control task along a given direction of the task space.

For resolving the redundancy of redundant manipulator in inverse kinematic control, Whitney [56] proposed resolved motion rate control using the Moore-Penrose *pseudo-inverse* of the Jacobian matrix J_e.

Although, the solution using pseudo-inverse has a least squares property which generates the minimum norm joint velocities, Baillieul, Hollerbach, and Brockett [4] showed that kinematic singularities are not avoided using this method. That is, joint velocities can become arbitrarily very high near singular configurations. *Damped least-square inverse* method was then proposed independently by Wampler [53] and Nakamura and Hanafusa [33] using a modified Jacobian that is nonsingular in the whole workspace. Under this control, only an approximate inverse kinematic solution is obtained and the problem is in selecting the suitable value of the damping factor λ which sets the weight of the minimum norm solution with respect to the minimum task tracking error. A technique to calculate a good estimate of the minimum singular value to set λ and a refinement of the technique which performs selective filtering only in the direction of the singular components for a given task trajectory was proposed by Maciejewski and Klein [29].

One of the widely used approaches for solving redundancy is by optimizing a scalar cost function using the *gradient projection* method where any differentiable cost function can be used based on the requirements. Some of the examples of cost functions can be found in Liégeois [27] for avoiding mechanical joint limits, in Yoshikawa [62, 61] for maximizing kineto-static and dynamic manipulability measures and in Dubey, Euler, and Babcock [11] for maximizing various criteria. Moreover, Mayorga and Wong [32] proposed another solution for optimizing rate of change of Jacobian based on proper bounds.

Another approach of redundancy resolution is by *task space augmentation* where an additional constraint task to be executed is imposed along with the original main task [44, 12]. However, it is unlikely that, for any augmented task, the joint paths will satisfy both tasks. An indirect way of choosing the constraint task is by optimizing some cost function of the type discussed above like as suggested by Baillieul [3] which was later formalized by Chang [6].

The augmented task space method has a nice feature of repeatability for any initial joint setting if the space of redundancy is fully exploited and the paths are chosen in a simply connected subset of the workspace [5]. However, a major problem that may be encountered in the application task space augmentation method is the occurrence of algorithmic singularities [2] which are the singularities associated with the augmented Jacobian matrix.

An effective solution for handling multiple tasks is task priority strategy formulated by Nakamura, Hanafusa, and Yoshikawa [35] where the original main task and the constraint task are assigned different priorities in that the lower priority task is satisfied only if it does not conflict with higher priority task.

However, in the complex environment, the number of constraints may be very high and prioritizing the tasks may be a challenging task. For such context, Xiang, Zhong, and Wei [59] proposed a varied weight method for coping the multiple constraints where the main task and the constraint subtasks are treated equally but with varied weights. In a framework for sensor-based robot control proposed by Kermorgant and Chaumette [19], the features coming from several sensors are smoothly injected into the control law allowing multiple constraints into account. Another approach has been proposed by Huang et al. [16] which uses the concept of virtual joints to cope with the multiple constraints.

Recently, researchers have proposed other approaches for solving the redundancy resolution problem for control of robotic manipulators. Specially, because of the parallel processing capability of neural networks, the problem of control algorithms which use conventional serial processing techniques are addressed by using neural networks for the control of redundant manipulators [42, 25, 37, 26]. However, development of neural networks for the real time control of manipulators with high accuracy is still challenging.

In some applications like robotic polishing, the control of end-effector force is necessary [50]. Von Sternberg [51] has proposed a generalized contact control framework for the end-effector force control. For this, the force/torque (F/T) sensor mounted on the wrist of the manipulator needs to be calibrated. This can be done through least squares estimation using the accelerometer signalsKubus, Kroger, and Wahl [23]. Lu, Chung, and Velinsky [28] have proposed collision detection techniques based on the wrist mounted F/T sensor.

In this thesis work, the redundancy resolution problem has been explained, most common solutions have been implemented and evaluated with respect to the Cartesian velocity control of Baxter. Moreover, F/T sensor mounted on the wrist of the Baxter is used for the contact detection of the gripper with the object to be grasped and the environment.

Chapter 3

Methodologies

3.1 Kinematics of Redundant Manipulators

For a manipulator, the task space is the space that defines the position and orientation of the end-effector whereas the joint space for a manipulator is comprised of all the variables that define the configuration of the joints. In the case of a redundant manipulator, number of joint variables are more than the DOFs of the end-effector. In other words, the dimension of the joint space n exceeds the dimension of the task space m in the redundant systems. The difference between n and m is denoted as the degree of redundancy r. In this definition, the redundancy is not only a characteristic of the manipulator itself but also of the task which means that a non redundant manipulator may also become a redundant manipulator for certain task.

Let the end-effector pose (position and orientation) be represented by the m-dimensional vector x and the configuration of the manipulator by the n-dimensional vector q of joint positions. The degree of redundancy r is given by Equation 3.1.

$$r = n - m, \text{ where } (r \geq 1) \tag{3.1}$$

The pose of the end-effector depends on the configuration of the joints which can be mathematically represented as a functional relation between the end-effector pose vector x and the joint variables vector q as

$$x = f(q) \tag{3.2}$$

where f is a m-dimensional vector function representing the manipulator's forward kinematics.

Similarly, the linear and angular velocity components for an end-effector can be related to the rate of change of the joint variables which can be expressed mathematically as

$$\dot{x} = J_e(q)\dot{q} \tag{3.3}$$

where $\dot{x}$ contains the linear and angular velocity components of the end-effector, and J_e is the $(m \times n)$ Jacobian of the end-effector. Equation 3.3 is known as the differential kinematics of the manipulator. The differential kinematics equation (equation 3.3), in terms of either the geometric or the analytical Jacobian, establishes a linear mapping between joint space velocities and task space velocities, and it can be utilized to solve for joint velocities using kinematic equation.

3.1.1 Mathematical Interpretation of Differential Kinematics Equation

Equation 3.3 has an interesting mathematical interpretation. $n \times 1$- dimensional mathematical space is formed by all possible joint variable (velocities) $\dot{q}$ that is a subset of $\mathbb{R}^n$. Also, $m \times 1$-dimensional mathematical space is formed by all possible end-effector velocity vectors $\dot{x}$ that is a subset of $\mathbb{R}^m$. From these definitions, at any fixed manipulator configuration q, the Jacobian matrix $J_e(q)$ can be interpreted as a linear transformation that maps vectors from the space $\mathbb{R}^n$ into the space $\mathbb{R}^m$.

Like any other linear transformation, the input space $\mathbb{R}^n$ of the Jacobian matrix has two important associated subspaces: the range and the null space(Figure 3.1). The range of the Jacobian matrix is the subspace of $\mathbb{R}^n$ that is covered by the transformation and physically, these are joint velocities that are mechanically possible to be generated by the manipulator's drive mechanism. The range denoted by $\mathbb{R}(J_e)$ can be mathematically defined as

$$\mathbb{R}(J_e) = \{J_e\dot{q} \mid \dot{q} \in \mathbb{R}^n\} \tag{3.4}$$

A subset of the input space $\mathbb{R}^n$ that is mapped to a zero vector in the output space $\mathbb{R}^m$ by the Jacobian matrix is the null space of the Jacobian matrix and physically, these are the joint velocities that do not generate any velocity at the end-effector. The null space of the Jacobian matrix is denoted by $\mathbb{N}(J_e)$ and can be mathematically defined as

$$\mathbb{N}(J_e) = \{\dot{q} \in \mathbb{R}^n \mid J_e\dot{q} = 0\} \tag{3.5}$$

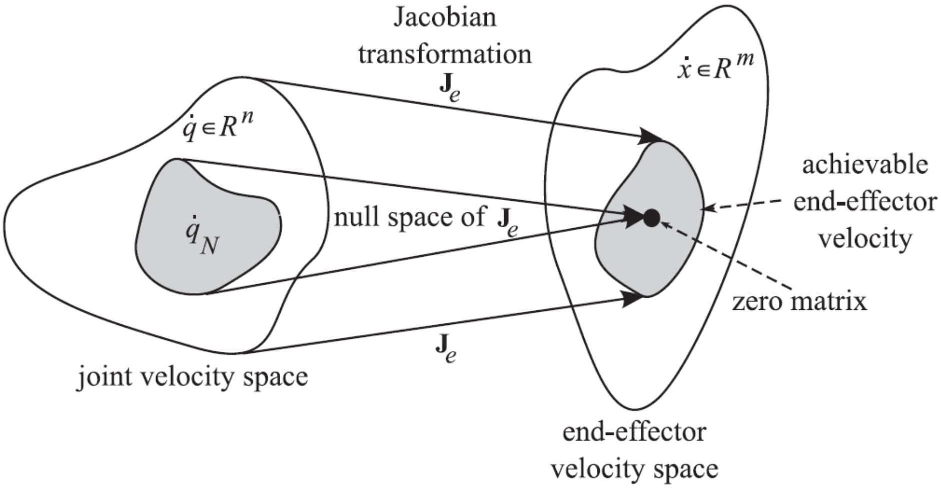


FIGURE 3.1: The Jacobian matrix J_e maps the joint velocity space onto the end-effector velocity space. The null space of the Jacobian matrix $\mathbb{N}(J_e)$ maps a portion of the joint velocity space $\dot{q}_N$ onto zero end-effector velocity, adopted from Fahimi [13, p. 17]

3.1.2 Mathematical and Physical Basis for Redundant Manipulators

The mathematical basis for redundant manipulators is the existence of the null space for the Jacobian matrix as defined in Equation 3.5. Intuitively, Equation 3.5 implies that the velocities $\dot{q}_N$ belonging to the null space $\mathbb{N}(J_e)$ do not generate any velocity $\dot{x}$ at the task space (end-effector), i.e.,

$$J_e\dot{q}_N = 0 \tag{3.6}$$

Although the joint velocities $\dot{q}_N$ do not generate any motion at the end-effector, they generate internal joint motions of the manipulator which can be used to satisfy some desired criteria that the redundant manipulator must meet, for example, joint limit avoidance, while the end effector is performing its main task.

Mathematically, a desired end-effector velocity $\dot{x}^d$ that can be generated by applying the joint rates $\dot{q}^d$ can be represented as

$$\dot{x}^d = J_e \dot{q}^d \tag{3.7}$$

If the joint velocities $\dot{q}_N$ are selected from the null space $\mathbb{N}(J_e)$ by an algorithm, the joint velocities $\dot{q}^d + \alpha\dot{q}_N$, where α is a scalar multiplier, still generate the desired end-effector velocity because

$$J_e(\dot{q}^d + \alpha\dot{q}_N) = J_e\dot{q}^d = \dot{x}^d \tag{3.8}$$

The rank of the Jacobian matrix determines the dimension of the null space from which $\dot{q}_N$'s can be selected. If the Jacobian matrix $J_e(q)$ has full column rank at a given joint position q, then the dimension of the null space $\mathbb{N}(J_e)$ is equal to the degree of redundancy. If the Jacobian matrix has a rank of $m' < m$, the dimension of $\mathbb{N}(J_e)$ is equal to $(n - m')$.

Since the choice of velocities that belong to the null space is not unique, there are several ways in which the desired main task $\dot{x}^d$ can be achieved, that is to say, there are an infinite number of solutions to the inverse kinematics problem for a redundant manipulator. This is the major advantage of redundant manipulators. Additional useful constraints can be satisfied by wisely using these multiple solutions while executing the main task specified via positions and orientations of the end-effector. Additional constraints can be defined and incorporated using two different approaches: global and local. Global approaches achieve optimal behavior along the whole trajectory which ensures superior performance over local methods [18, 34, 49]. However, their computational burden makes them unsuitable for real-time sensor-based manipulator control applications.

3.2 Redundancy Resolution at the Velocity Level

In most real world applications of manipulators, the main task is the desired trajectory of the end effector, i.e., timed position and orientation. For controlling the manipulator, these control inputs should be projected into joint space. In other words, the trajectory of joint variables is required. Therefore, the solution to the inverse kinematics problem (redundancy resolution for redundant manipulators) is necessary. That is, the desired joint rates that generate a desired end effector velocity should be calculated which is called redundancy resolution at the velocity level. Because of the kinematic redundancy in redundant manipulators, there are always more unknown joint velocities than there are equations which make the redundancy resolution for a redundant manipulator little challenging. In this project, different mathematical methods that allow the solution for redundant manipulators at velocity levels are implemented and analyzed.

3.2.1 Pseudo-Inverse (Exact Solution) Method

In this method, for a given desired instantaneous velocity $\dot{x}$ at the end-effector of a redundant manipulator, a solution $\dot{q}$ is selected which exactly satisfies Equation 3.3. One of the mostly used methods for obtaining this exact solution is based on the pseudo-inverse of the matrix J_e, denoted by $J_e^\dagger$:

$$\dot{q}_p = J_e^\dagger \dot{x} \tag{3.9}$$

where the subscript p indicates the primary solution (corresponding to the main task) to Equation 3.3. This solution can be later enhanced by adding solutions $\dot{q}_N$ from the null space of the Jacobian matrix J_e.

The pseudo inverse of J_e can be expressed as

$$J_e^\dagger = v\sigma^* u^T \tag{3.10}$$

where σ, v, and u are obtained from the singular-value decomposition (SVD) of J_e [14], and σ^* is the transpose of σ with all the non-zero values reciprocated. Equation 3.10 can also be written as

$$J_e^\dagger = \sum_{i=1}^{m'} \frac{1}{\sigma_i} \hat{v}_i \hat{u}_i^T \tag{3.11}$$

where m' is the number of nonzero diagonal components of the matrix σ, and σ_i is the i-th nonzero diagonal element of the matrix σ, and $\hat{v}_i$ and $\hat{u}_i$ are the i-th column of the matrices v and u, respectively. If J_e has full row rank, then its pseudo inverse is given by:

$$J_e^\dagger = J_e^T (J_e J_e^T)^{-1} \tag{3.12}$$

Equation 3.9 represents the general form of a minimum 2-norm solution to the following least-square problem:

$$min_{\dot{q}}\{\|J_e \dot{q} - \dot{x}\|\} \tag{3.13}$$

The main advantage of the pseudo-inverse method of redundancy resolution is its ability to provide a meaningful solution in the least-squares sense regardless of whether the forward kinematics equation, i.e., Equation 3.3 is under-specified, square, or over-specified. However, there are some drawbacks associated with this solution like the solution given by Equation 3.9 does not guarantee generation of joint motions which avoid singular configurations - configurations at which the Jacobian matrix J_e does not have full rank [31]. At singular positions, end-effector cannot generate velocity components in certain directions, whereas close to singular position, very large joint rates are required to generate an end-effector velocity in certain directions which is not desirable. This can be seen mathematically by Equation 3.10 or Equation 3.11, in which even the largest element of matrix σ is very close to zero at a singular posture. Therefore, there are some velocities in task space which require very large joint rates resulting from Equation 3.9.

Another problem with the pseudo-inverse method of redundancy resolution is that repeatability and cyclicity condition are not preserved (a closed path in Cartesian space may not result in a closed path in joint space) by the joint motions generated by this approach [22].

Another difficulty with this method is that the extra DOFs (when $dim(q) > dim(x)$) are not exploited for any useful purpose to satisfy user-defined additional tasks because Equation 3.9 only provides the primary solution, which is not in the null space of the Jacobian J_e.

To overcome this problem, a joint velocity vector $\dot{q}_N$ that belongs to the null space of the Jacobian matrix J_e can be added to the primary solution (Equation 3.9) as [11]

$$\dot{q} = \dot{q}_p + \dot{q}_N \tag{3.14}$$

As shown in Equation 3.8, the new joint velocity $\dot{q}$ still satisfies Equation 3.3. The term $\dot{q}_N$ can be obtained by projection of arbitrary n-dimensional vector $\dot{\varphi}$ to the null space of the Jacobian J_e as

$$\dot{q}_N = (I - J_e^{\dagger} J_e)\dot{\varphi} \tag{3.15}$$

where $\dot{\varphi}$ can be chosen as [58, 27]

$$\dot{\varphi} = -k_0 \nabla \Phi(q) = -k_0 (\frac{\partial \Phi(q)}{\partial q})^T \tag{3.16}$$

with $k_0 > 0$, $\Phi(q)$ is a cost function (scalar objective function of the joint variables) that can be selected to satisfy different objectives whose optimal value would ensure the desired additional task and $(\frac{\partial \Phi(q)}{\partial q})^T$ is the vector function representing the gradient of Φ. The desired additional task can be torque and acceleration minimization [45], singularity avoidance [33], or obstacle avoidance [8, 2].

3.2.2 Singularity Avoidance (Approximate Solution) Method

The exact solution method as discussed in section 3.2.1 like the solution given by Equation 3.9 does not guarantee the generation of joint motions which avoid singular configurations. An alternative method to deal with this type of problems of artificial/kinematic singularities and large joint rates is to solve the problem for an approximate solution. For a redundant manipulator, singularity avoidance can be achieved by exploiting its extra DOFs than that required to perform the primary task.

In this method, the exact solution of a linear equation (Equation 3.3) is replaced by a solution that takes into account both the accuracy and the norm of the solution at the same time which is often referred as the damped least-squares solution. The least-squares criterion that is used in different forms for redundancy resolution [53, 33] can be used for solving Equation 3.3 for avoiding singularities.

Considering $\dot{x}^d$ as a desired main task (end-effector tracking), the joint rate $\dot{q}$ can be determined that approximately satisfies Equation 3.3 by minimizing the cost function.

$$F = \|J_e \dot{q} - \dot{x}^d\|^2 \tag{3.17}$$

For avoiding the singularities, the weighted norm of the joint rates can be added to the cost function above because of which the high joint rates are penalized causing the manipulator to avoid the singular posture. So the cost function can be rewritten as

$$F = \|J_e \dot{q} - \dot{x}^d\|^2 + \|\lambda \dot{q}\|^2 \tag{3.18}$$

where λ, the damping or singularity robustness factor, is used to specify the relative importance of the norms of joint rates and the tracking accuracy. This can be equivalently represented by replacing Equation 3.3 by a new augmented system of equations as

$$\begin{bmatrix} J_e \\ \lambda I \end{bmatrix} \dot{q} = \begin{bmatrix} \dot{x}^d \\ 0 \end{bmatrix} \tag{3.19}$$

The least-squares solution for the new system of equations (Equation 3.19) can be determined by solving the following consistent set of equations:

$$(J_e^T J_e + \lambda^2 I)\dot{q} = J_e^T \dot{x}^d \tag{3.20}$$

The least square solution can be thus written as

$$\dot{q}_\lambda = (J_e^T J_e + \lambda^2 I)^{-1} J_e^T \dot{x}^d \tag{3.21}$$

The Equation 3.21 gives a unique solution which most closely approximates the desired task velocity (exact solution) while avoiding high joint rates.

The singular value decomposition (SVD) of the matrix $(J_e^T J_e + \lambda^2 I)^{-1} J_e^T$ in Equation 3.21 is given by

$$(J_e^T J_e + \lambda^2 I)^{-1} J_e^T = \sum_{i=1}^{m'} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \hat{v}_i \hat{u}_i^T \tag{3.22}$$

where m' ,σ_i, $\hat{v}_i$ and $\hat{u}_i$ are as in Equation 3.11. By comparing Equations 3.22 and 3.11, it can be seen that the weight λ is what makes the actual difference between the singular values of the two solutions. That is to say the singular values of the exact and the approximate solutions are $\frac{1}{\sigma_i}$ and $\frac{\sigma_i}{\sigma_i^2+\lambda^2}$ respectively. Assigning $\lambda = 0$, we can obtain the pseudo-inverse soution from Equation 3.22

When manipulator is far from singular configuration and singular values are much larger than λ then there is little difference between the two solutions that is to say the singular values of the exact and approximate solutions are very close, causing close solutions for $\dot{q}$,i.e.,

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

Moreover, when the manipulator is close to singular configuration, i.e., if the singular values are of the order of λ or smaller, then the weight λ^2 in the denominator tends to reduce the potentially high norm joint rates. In all the cases discussed above, the norm of joint rates will be bounded by

$$\|\dot{q}_\lambda\| \leq \frac{1}{2\lambda} \|\dot{x}^d\| \tag{3.23}$$

The solutions obtained by exact solution (pseudo-inverse) method and approximate solution (damped least-squares) method can be compared diagrammatically in Figure 3.2.

From Figure 3.2 it can be clearly seen that, the problems of discontinuity at singular configurations and large solution norms near singularities associated with exact solution (pseudo-inverse) method are changed in the approximate solution(damped least-squares) method.

3.3 Joint Limit Avoidance (JLA) by Kinematic Optimization

Joint Limit Avoidance (JLA) is an important issue that needs to be addressed while planning the trajectory of the manipulator joints to perform certain tasks. This is done by defining a kinematic objective function and minimizing it to prevent the joints from reaching their limits by taking the advantage of the null space of the Jacobian matrix J_e of the redundant manipulator.

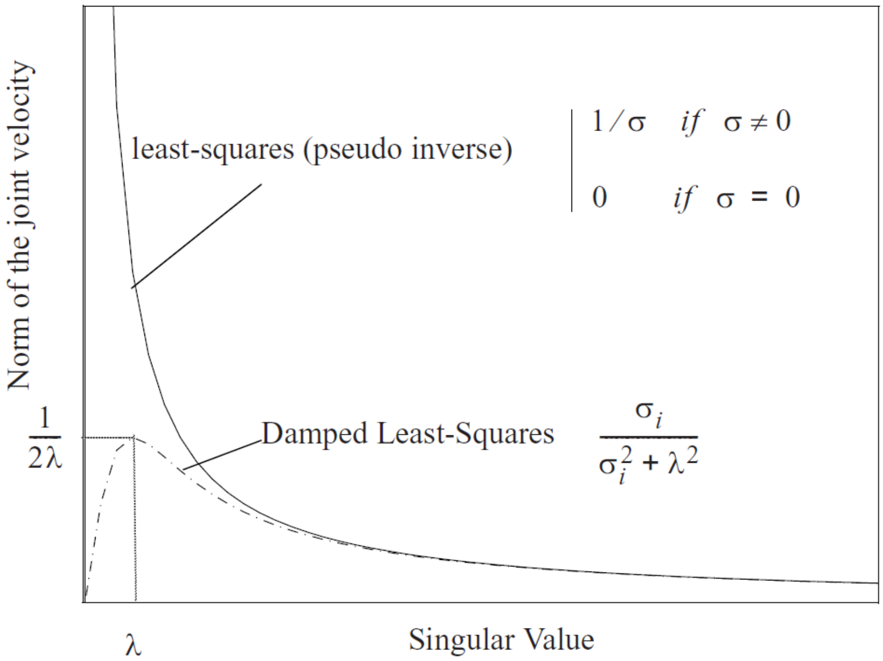


FIGURE 3.2: Damped versus undamped least-square solution, adopted from Patel and Shadpey [38, p. 15]

At first, an exact or approximate solution to the redundancy resolution problem is determined from Equation 3.9 or Equation 3.21. Then the final joint velocities are calculated by adding joint velocities solution belonging to the null space of the manipulator Jacobian matrix J_e as in Equation 3.14. Null space solution is found from Equations 3.15 and 3.16.

There are many objective functions that have been proposed for joint limit avoidance, i.e., to keep the joint far from their limits. Taking distance from the mechanical joint limits as the objective function [58, 27], a quadratic cost function that is to be minimized can be written as,

$$\Phi(q) = \frac{1}{n}\sum_{i=1}^{n}(\frac{q_i - \overline{q_i}}{q_{iM} - q_{im}})^2 \tag{3.24}$$

where q_{iM} denotes the maximum limit, q_{im} denotes the minimum limit and $\overline{q_i}$ denotes the middle value of the joint range for joint i.

Thus, Joint limit avoidance can be achieved by finding $\dot{\varphi}$ that optimizes Equation 3.24 and using Equations 3.14 and 3.15, redundancy can be exploited to keep the manipulator joints away from their mechanical limits.

Another approach of optimization which reflects the joint limit avoidance objective has been proposed using p-norm [54, 20], where the p-norm of a vector can be defined as

$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p} \tag{3.25}$$

Using the p-norm, the cost function for joint limit avoidance can be defined as

$$\Phi(q) = max\frac{|q_i - \overline{q_i}|}{q_{iM} - q_{im}} = \|\frac{q - \overline{q}}{q_M - q_m}\|_p \tag{3.26}$$

where $p \geq 2$ and in most practical applications $p = 6$ gives good results. As it is visible from Equation 3.26 that all the joints have the same importance in the objective function. In some practical applications, it might be more important for certain joints to avoid joint limits. In such case, a $n \times n$ diagonal weighting matrix K can be introduced to form a new objective function for kinematic optimization as

$$\Phi(q) = \|K\Big(\frac{q-\overline{q}}{q_M - q_m}\Big)\|_p \tag{3.27}$$

$\dot{\varphi}$ that optimizes Equation 3.27 is calculated using Equations 3.16 and 3.27, and using Equations 3.14 and 3.15, the joint velocities are determined which tends to keep the trajectory for the joints around the center of their ranges at all times by exploiting the redundancy of the manipulator.

3.4 Closed Loop Inverse Kinematics (CLIK) Algorithm

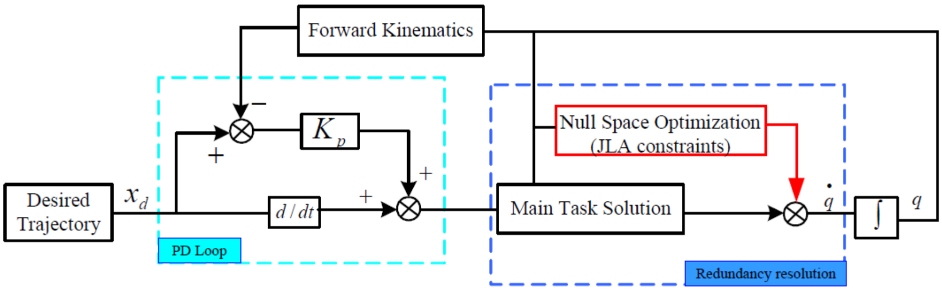


FIGURE 3.3: CLIK scheme with redundancy resolution at the velocity level, adopted from Wang, Li, and Zhao [55, p. 3]

Figure 3.3 shows the CLIK algorithm with redundancy resolution at the velocity level where the Proportional Derivative (PD) feedback loop takes the desired trajectory and current position and orientation of end-effector as input and regulates the input of the redundancy resolution.

Considering vectors x and x^d represent the current and desired position (and orientation) of end-effector and $\dot{x}$ and $\dot{x}^d$ are their derivatives respectively, then the position (and orientation) error and its derivative can be given by

$$e = x^d - x, \dot{e} = \dot{x}^d - \dot{x}, \tag{3.28}$$

The exact or approximate solutions as discussed in previous sections resolves the redundancy at velocity level along with null-space optimization for joint limit avoidance. Thus, the general inverse solution of a kinematically redundant manipulator at a velocity level based on CLIK algorithms can be written as

$$\dot{q} = J^{\dagger}(q)(\dot{x}^d + K_p(x^d - x)) + (I - J^{\dagger}(q)J(q))\dot{q}_N \tag{3.29}$$

For using exact solution (pseudo-inverse) method $J^{\dagger}$ can be calculated using Equations discussed in Section 3.2.1 and for using the singularity avoidance (approximate solution) method, $J^{\dagger}$ can be replace by $(J^T J + \lambda^2 I)^{-1} J^T$ in Equation 3.29 as discussed in Section 3.2.2.

Using Equation 3.29, joint velocity vector can be obtained for any task and the tracking error along the given trajectory converges to zero with a rate depending on the eigenvalues of K_p, that is,

$$\dot{e} + K_p e = 0 \tag{3.30}$$

where K_p is a symmetric positive definite matrix, and the choice of K_p guarantees that the error uniformly converges to zero.

3.5 Force/Torque Sensor for Contact (Collision) Detection

While the end-effector is following the desired trajectory, the contact (collision) of the end-effector with any object (obstacle) can be detected using the wrist mounted force/torque (F/T) sensor [28]. It can be also used to control the forces at the end-effector of a manipulator if the F/T sensor is calibrated [51]. Wrist mounted F/T sensor parameters can be calibrated through least squares estimation using the F/T sensor and accelerometer signals by moving the manipulator into a number of different poses [23]. If the parameters (bias of the F/T sensor, mass of the gripper location of the center of mass of the attached gripper) related to F/T Sensor are known, the gravity forces measured by a F/T sensor can be compensated and hence the filtered readings of the sensor can be used for contact (collision) detection, end-effector force control [43, 41, 50].

Chapter 4

Experimentation and Results

4.1 Experimental Setup

The overall hardware and software systems for the Amazon Robotics Challenge 2017 consists of the following main components:

4.1.1 Hardware System

The hardware system used for the challenge is shown in Figure 4.1. It consists of robot, storage system, totes, boxes and items (objects to be picked and placed). However, the work done within this thesis uses mainly the robot arm to test and verify the control algorithms explained in Chapter 3.

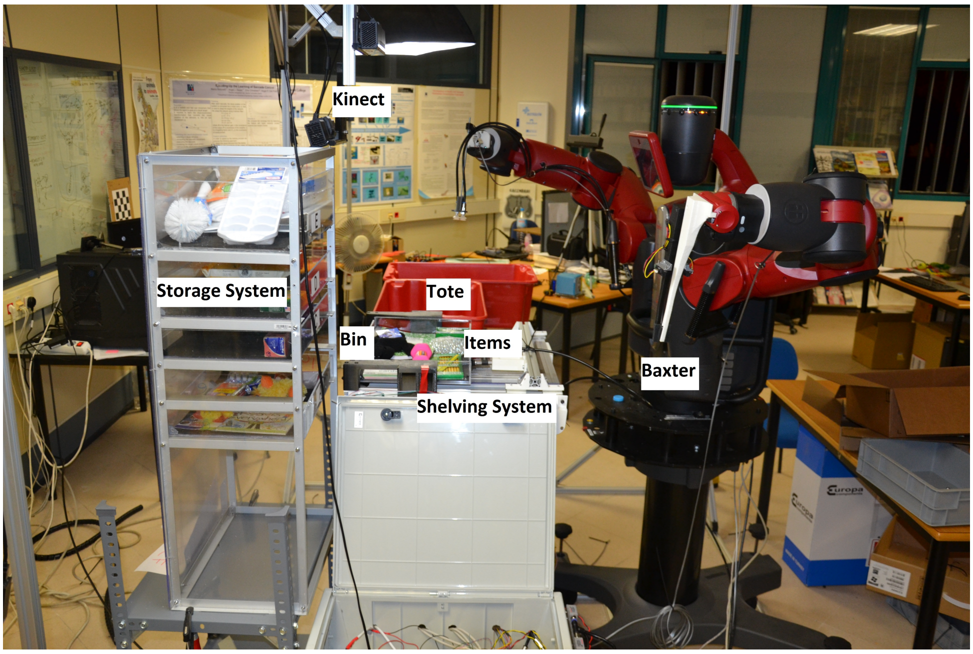


FIGURE 4.1: Hardware System

(a) **Robot** - Baxter with customized grippers is used as a robot which is extended with a *mobile shelving system* to carry the bin from/to the storage system to/from just in front of Baxter. The vision system comprising of Kinect is mounted vertically up in front of Baxter.

(b) **Storage System** - consists of 6 distinct internal compartments or bins that are clearly labeled with letters. It is used for storing the items (within the bins).

(c) **Totes and Boxes**

(d) **Items** - contains popular kinds of products and may include books, cubic boxes, clothing, soft objects, objects with handles, and irregularly shaped objects.

4.1.2 Software System

The work done in this thesis includes the implementation of software module for kinematic control of Baxter arms. The overall picking and placing operations with respect to arm control are shown in Figure 4.3. The software system used for the challenge is shown in Figure 4.2 and consists of the following modules.

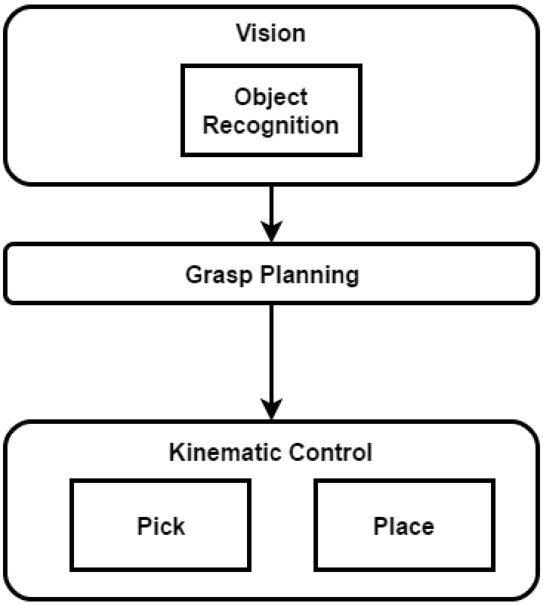
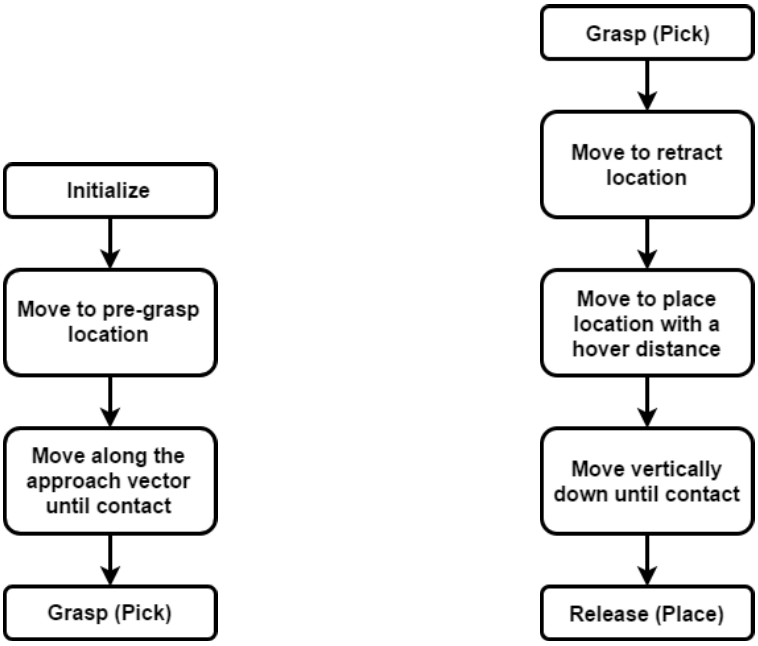


FIGURE 4.2: Software Architecture

(a) **Vision Pipeline** - dedicated for object recognition.

(b) **Grasp Planning Pipeline** - gives the approach vector necessary for grasping the items.

(c) **Kinematic Control** - moves the arm for picking and placing operations.


Initialize
Move to pre-grasp location
Move along the approach vector until contact
Grasp (Pick)
Grasp (Pick)
Move to retract location
Move to place location with a hover distance
Move vertically down until contact
Release (Place)

(A) Pick

(B) Place

FIGURE 4.3: Pick and place operations

4.2 Experimentation

The motion planning and control of robotic arm is simplified by adding the mobile shelving system with the robot system. The mobile shelving system facilitates the movement of manipulator by providing easily feasible position of the object to be picked. Thus, the motion planning and control of the arm is limited to picking/placing the items from/to just in front of the robot.

For testing the Cartesian velocity control algorithms as described in Chapter 3, the Jacobian matrix J_e and the pseudo-inverse of Jacobian matrix $J_e^\dagger$ are obtained using the Baxter PyKDL[1] library. The right arm of the Baxter is used at its initial joint configuration q_0 where,

$$q_0 = \begin{bmatrix} -1.17 & -1.11 & 0.92 & 1.16 & 1.14 & 0.38 & -1.44 \end{bmatrix} \tag{4.1}$$

The initial joint angles in Equation 4.1 correspond to joints $(S_0, S_1, E_0, E_1, W_0, W_1, W_2)$ respectively. From this configuration, the arm is moved to a desired pose $x^d = \begin{bmatrix} 0.80 & 0.09 & 0.43 & 0.86 & 0.50 & 0.01 & -0.03 \end{bmatrix}$ where first 3 numbers represent positions (x,y,z) and last 4 numbers represent orientation using quaternions(x,y,z,w). Kinematic control with different redundancy resolution techniques are used for the motion and the results are compared to select the best control technique suitable for the Challenge.

The joint velocities of each joint are limited to ± 0.5 radians per second via normalization to avoid excessive joint velocities and actuators saturation. The maximum velocity limits are later varied to compare the performances of different algorithms of redundancy resolutions in Section 4.4.

[1] http://sdk.rethinkrobotics.com/wiki/Baxter_PyKDL

4.2.1 Calibration of Force/Torque Sensor

6-axis force/torque (F/T) sensor manufactured by Schunk is mounted on the wrist of the manipulator. The customized gripper is mounded at the tool side of the sensor. The sensor is connected to the system using ethernet via Net Box as shown in Figure 4.4.

The Net F/T sensor system simultaneously measures forces Fx, Fy, Fz, and torques Tx, Ty, and Tz. The Net F/T system provides EtherNet/IP, CAN bus, and Ethernet communication interfaces and is compatible with DeviceNet. In this project, the F/T sensor is used with the driver developed as a ROS node[2].

The F/T sensor is calibrated using the equations adopted from Kubus, Kroger, and Wahl [23] and software tools available at `https://github.com/kth-ros-pkg/force_torque_tools` with some modifications. The software calibrates the F/T sensor by moving the arm into a number of different poses using Moveit![3] and combining the accelerometer/imu and F/T sensor signals. The inverse kinematics solver of Moveit! is changed from default KDL IK solver to TRAC-IK[4] solver for better performance.

The wrist mounted calibrated force torque sensor is used for contact detection with objects in the environment and for the robust extraction of the object during picking and stowing operations.

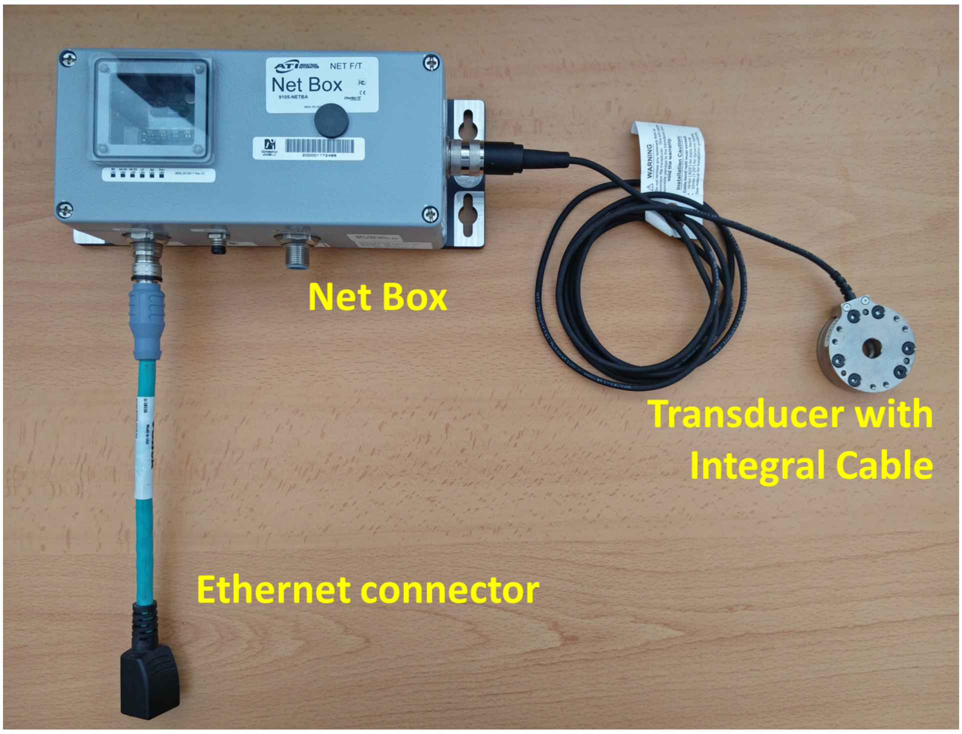


FIGURE 4.4: Net F/T System Components

[2] The driver is adopted from `https://github.com/UTNuclearRoboticsPublic/netft_utils`

[3] `http://moveit.ros.org/`

[4] `https://bitbucket.org/traclabs/trac_ik`

4.3 Results

4.3.1 Redundancy Resolution at Velocity level using Pseudo-Inverse Method

The joint rates required to move the end-effector from current position x to desired position x^d can be given by (see Chapter 3):

$$\dot{q}_p = J_e^{\dagger}\dot{x} \tag{4.2}$$

where $J_e^{\dagger}$ is the pseudo-inverse of Jacobian matrix J_e and $\dot{x}$ consists of the desired twist and is given by

$$\dot{x} = K_p(x^d - x) \tag{4.3}$$

where K_p represents the proportional gain.

In this case, only the exact solution (pseudo-inverse) is considered with no kinematic optimization. Application of this algorithm with $K_p = 2$ results in the joint trajectories shown in Figure 4.9. As can be seen in Figures 4.7 and 4.8, these joint trajectories cause the end-effector of the manipulator to move from the current pose x to the desired pose x^d. The end-effector position and orientation error converges to zero as shown in Figures 4.5 and 4.6.

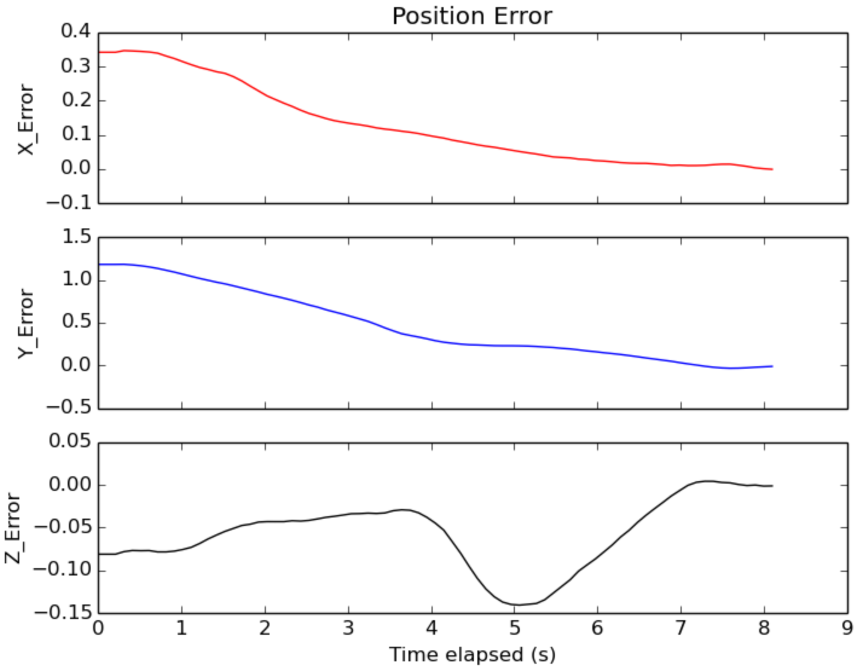


FIGURE 4.5: End-effector position error (Exact solution method)

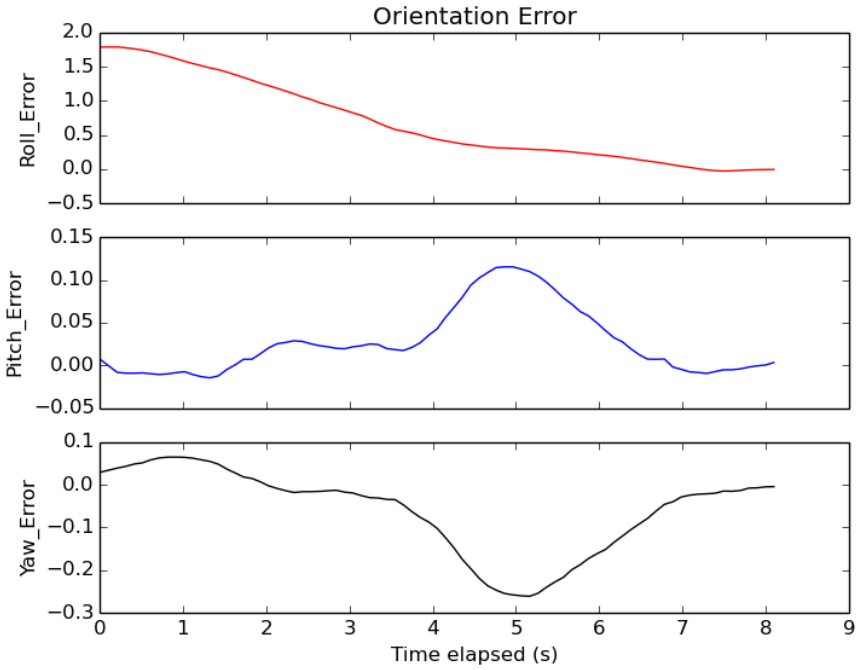


FIGURE 4.6: End-effector orientation error (Exact solution method)

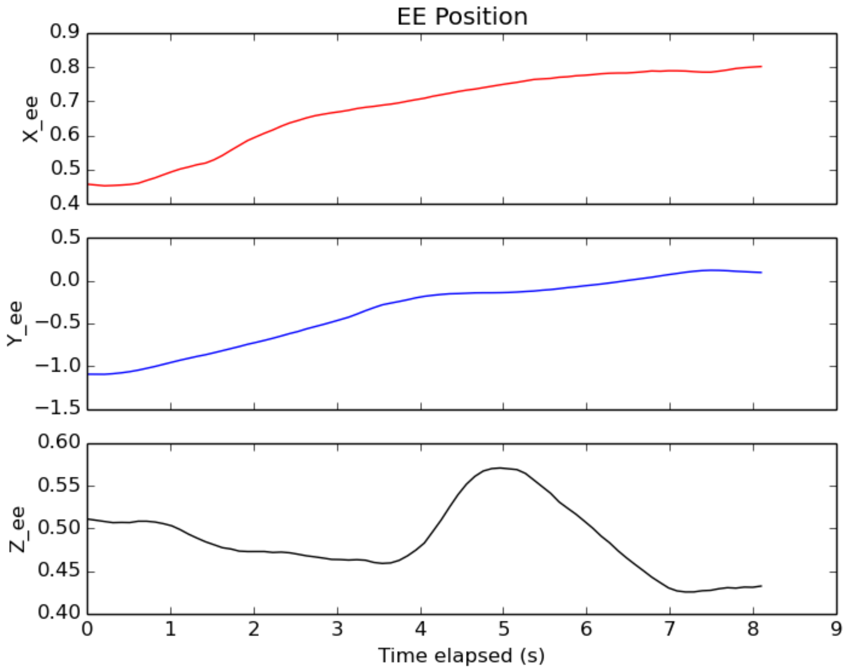


FIGURE 4.7: End-effector position trajectory (Exact solution method)

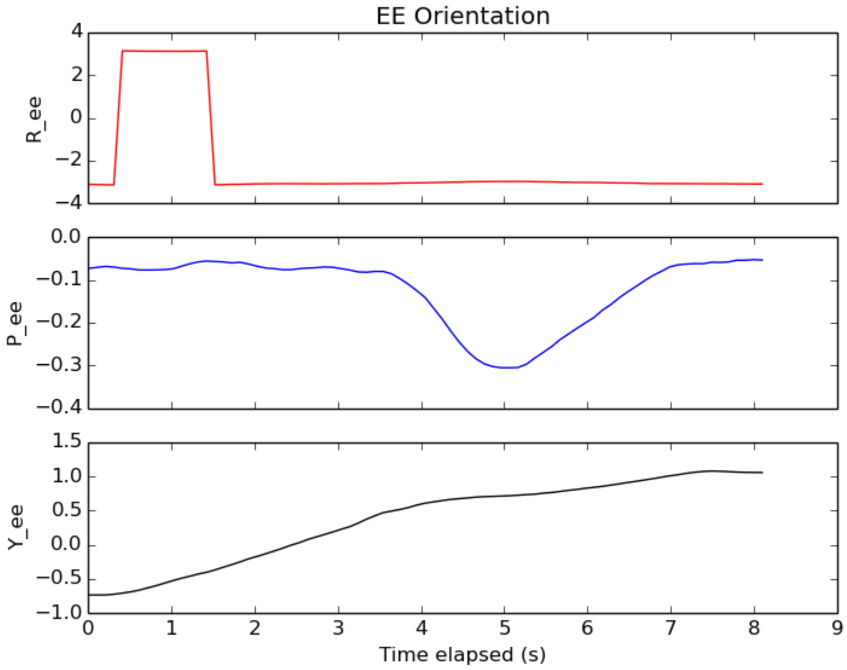


FIGURE 4.8: End-effector orientation trajectory (Exact solution method)

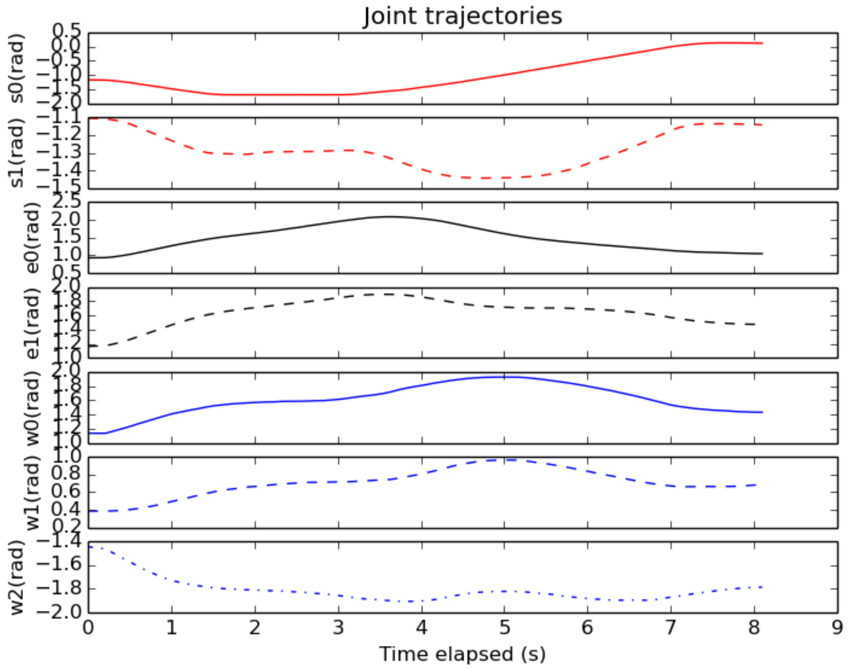


FIGURE 4.9: Joint trajectories (Exact solution method)

4.3.2 Redundancy Resolution at Velocity level with Singularity Avoidance using Approximate Solution Method

The joint velocities required to move the end-effector from current position x to desired position x^d is given by (see Chapter 3):

$$\dot{q}_\lambda = (J_e^T J_e + \lambda^2 I)^{-1} J_e^T \dot{x} \tag{4.4}$$

where $\lambda = 0.1$ is the damping or singularity robustness factor and $\dot{x}$ is same as in Equation 4.3

Here, an approximate solution is considered with no kinematic optimization. Application of this algorithm with $K_p = 2$ results in the joint trajectories shown in Figure 4.14. As can be seen in Figures 4.12 and 4.13, these joint trajectories cause the end-effector of the manipulator to move from the current pose x to the desired pose x^d. The end-effector position and orientation error converges to zero as shown in Figures 4.10 and 4.11.

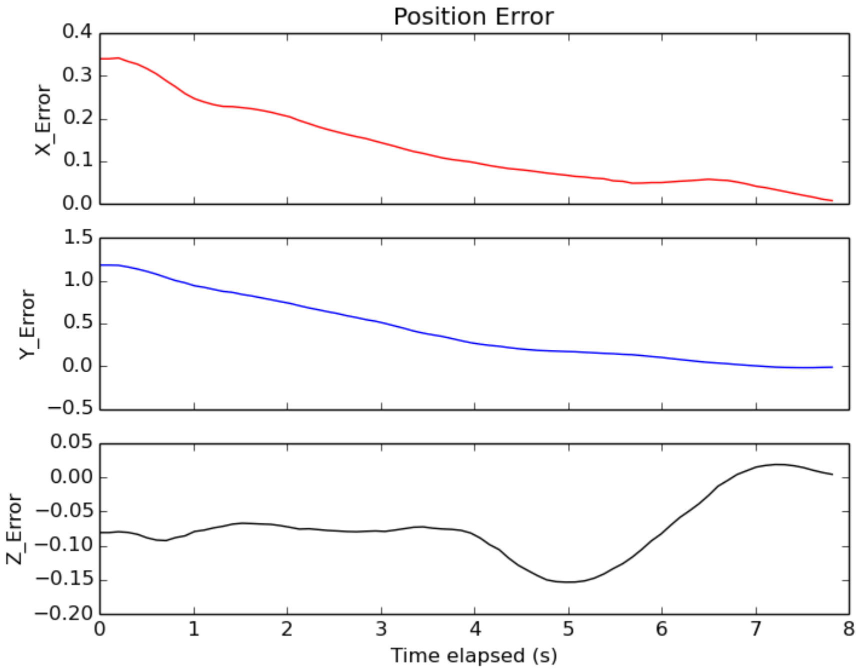


FIGURE 4.10: End-effector position error (Approximate solution method)

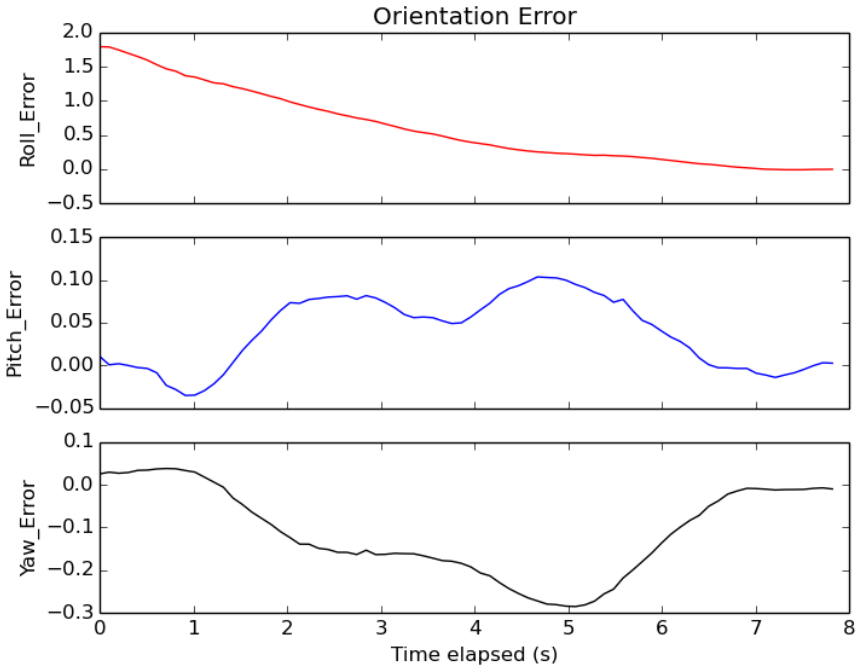


FIGURE 4.11: End-effector orientation error (Approximate solution method)

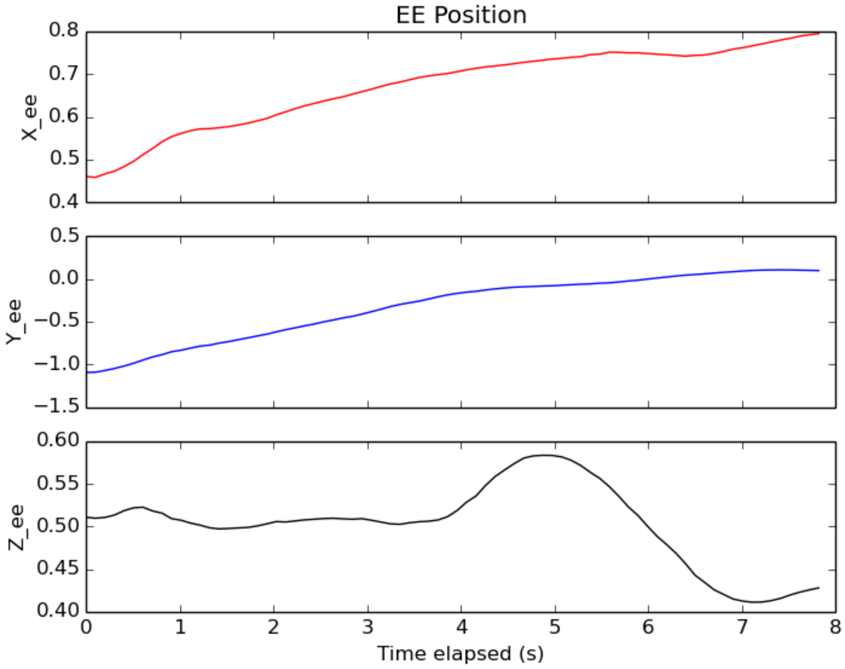


FIGURE 4.12: End-effector position trajectory (Approximate solution method)

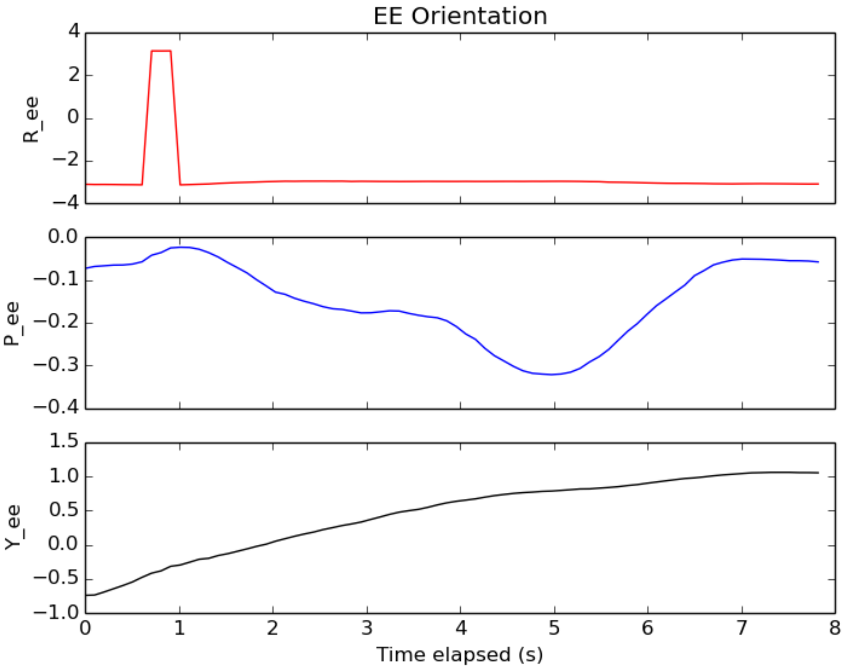


FIGURE 4.13: End-effector orientation trajectory (Approximate solution method)

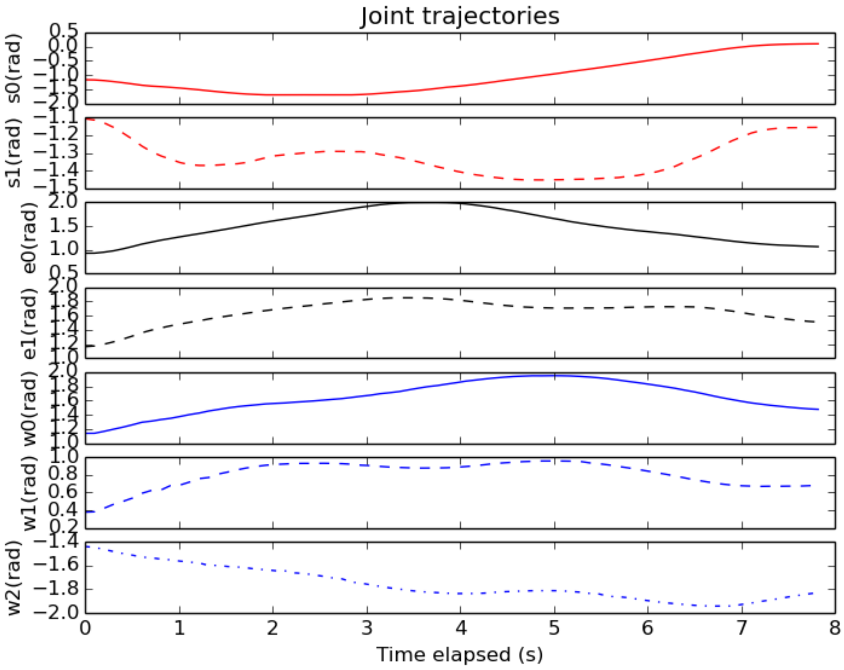


FIGURE 4.14: Joint trajectories (Approximate solution method)

4.3.3 Redundancy Resolution at Velocity level with Joint Limit Avoidance using Pseudo-Inverse Method Method

In the above two cases, joint limit avoidance task is not considered. In this case, the joint limit avoidance task is considered along with the primary task of the end effector to reach the desired position.

The joint rates required to move the end-effector from current position x to desired position x^d and avoid joint limits during the motion can be given by (see Chapter 3 for more details):

$$\dot{q} = J_e^\dagger \dot{x} + (I - J_e^\dagger J_e)\dot{\varphi} \tag{4.5}$$

where $J_e^\dagger$ and $\dot{x}$ are same as in Equation 4.2, and $\dot{\varphi}$ is determined by Equation 3.16 with $k_0 = 1$ and Equation 3.27 with $p = 2$ and K as,

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

In Equation 4.5, the first part of the solution, i.e., $J_e^\dagger \dot{x}$ generates the desired motion at the end effector and the second part of the solution, i.e., $(I - J_e^\dagger J_e)\dot{\varphi}$, belonging to the null space of the manipulator's Jacobian, keep the joints far from their limits.

Application of this algorithm with $K_p = 2$ results in the joint trajectories shown in Figure 4.19.

The joint limits are avoided and specially visible for $S0$ joint in Figure 4.19, because the joint limit avoidance for $S0$ joint is intensionally given greater importance (higher weight) than the other (it has an application specific advantage). One can change the weights for individual joints to avoid joint limits based on the requirements.

As can be seen in Figures 4.17 and 4.18, these joint trajectories cause the end-effector of the manipulator to move from the current pose x to the desired pose x^d while avoiding the joint limits. The end-effector position and orientation error converges to zero as shown in Figures 4.15 and 4.16.

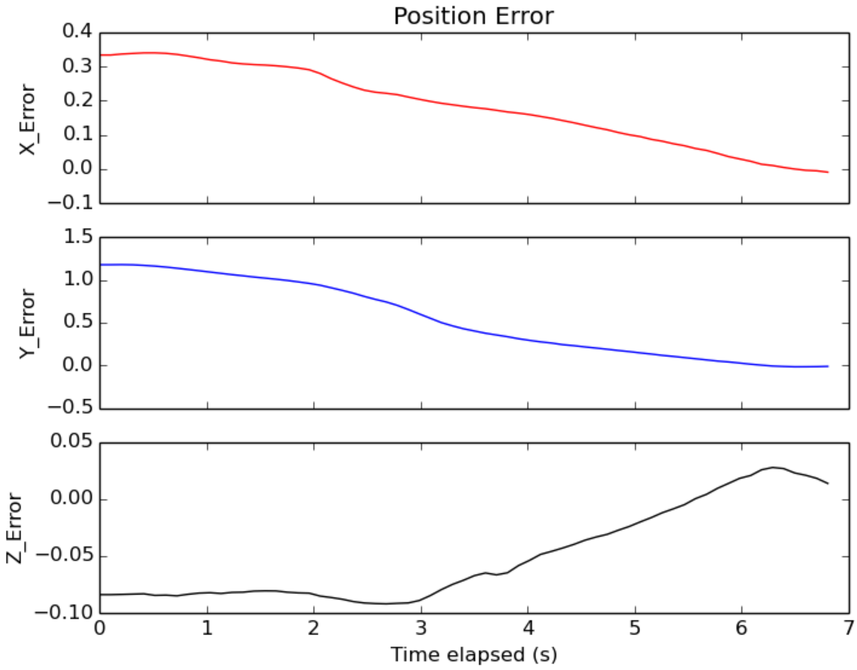


FIGURE 4.15: End-effector position error (Exact solution method with joint limit avoidance)

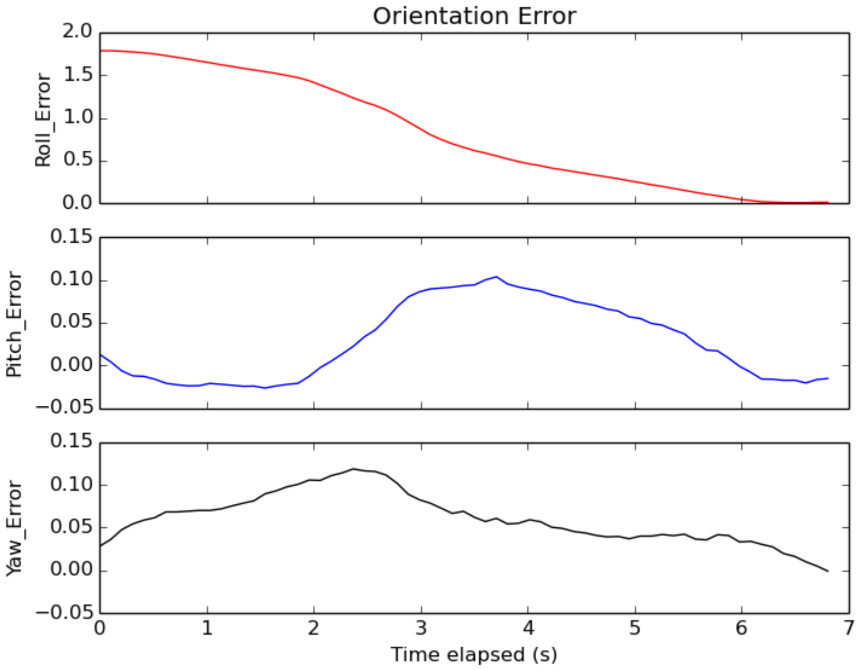


FIGURE 4.16: End-effector orientation error (Exact solution method with joint limit avoidance)

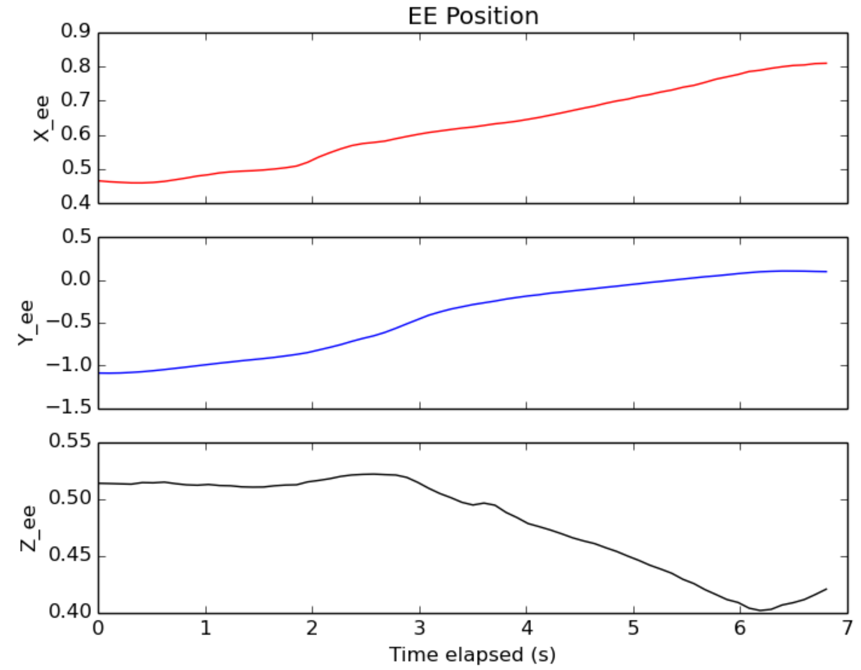


FIGURE 4.17: End-effector position trajectory (Exact solution method with joint limit avoidance)

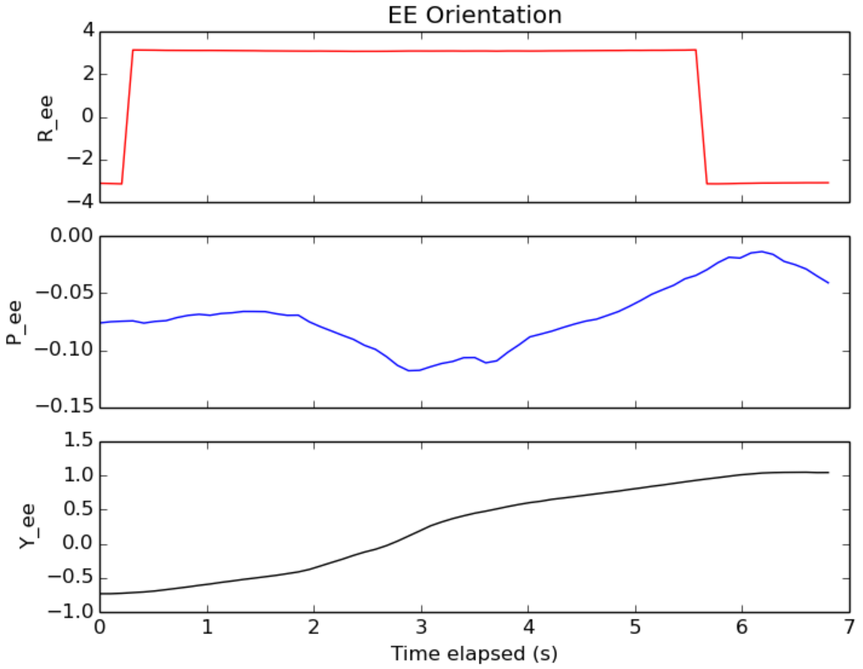


FIGURE 4.18: End-effector orientation trajectory (Exact solution method with joint limit avoidance)

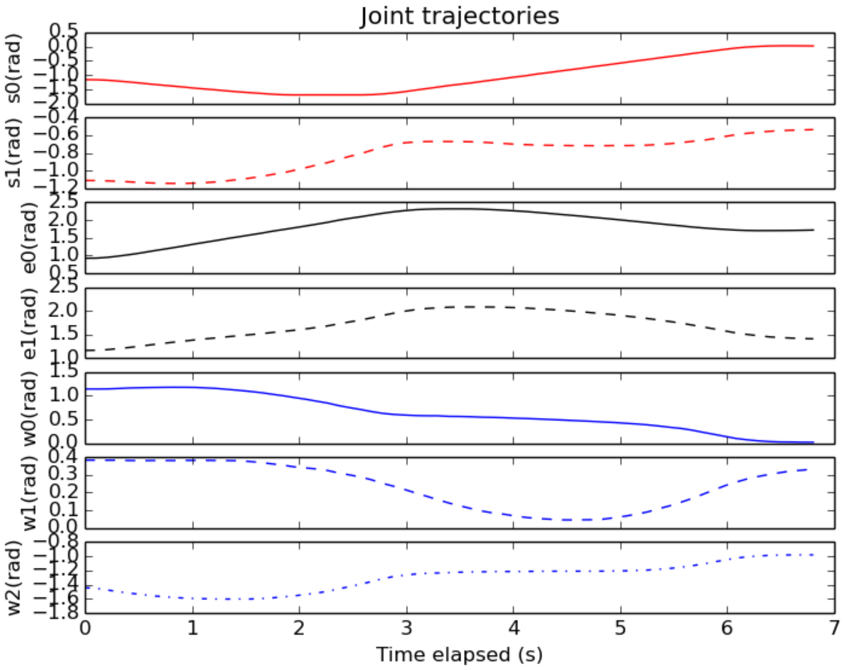


FIGURE 4.19: Joint trajectories (Exact solution method with joint limit avoidance)

4.3.4 Redundancy Resolution at Velocity level with Joint Limit Avoidance using Approximate Solution Method

In section 4.3.3, joint limit avoidance task is considered by adding it to the exact solution, where the joint limit avoidance task is achieved but without avoiding the singularity.

In this case, the joint limits are avoided, while the end effector moves from current position to the desired position. In addition, the approximate solution is used for the redundancy resolution which includes singularity avoidance.

The joint rates required to move the end-effector from current position x to desired position x^d and avoid joint limits during the motion can be given by (see Chapter 3 for more details):

$$\dot{q} = (J_e^T J_e + \lambda^2 I)^{-1} J_e^T \dot{x} + (I - (J_e^T J_e + \lambda^2 I)^{-1} J_e^T J_e)\dot{\varphi} \tag{4.7}$$

where λ and $\dot{x}$ are same as in Equations 4.4, and $\dot{\varphi}$ is determined as in Equation 4.5.

In Equation 4.7, the first part of the solution, i.e., $(J_e^T J_e + \lambda^2 I)^{-1} J_e^T \dot{x}$ generates the desired motion at the end effector with singularity avoidance and the second part of the solution, i.e., $(I - (J_e^T J_e + \lambda^2 I)^{-1} J_e^T J_e)\dot{\varphi}$, belonging to the null space of the manipulator's Jacobian, keep the joints far from their limits.

Application of this algorithm with $K_p = 2$ results in the joint trajectories shown in Figure 4.24.

The joint limits are avoided and specially visible for $S0$ joint in Figure 4.24, because the joint limit avoidance for $S0$ joint is intensionally given greater importance (higher weight) than the other (it has an application specific advantage). One can change the weights for individual joints to avoid joint limits based on the requirements.

As can be seen in Figures 4.22 and 4.23, these joint trajectories cause the end-effector of the manipulator to move from the current pose x to the desired pose x^d while avoiding the joint limits. The end-effector position and orientation error converges to zero as shown in Figures 4.20 and 4.21.

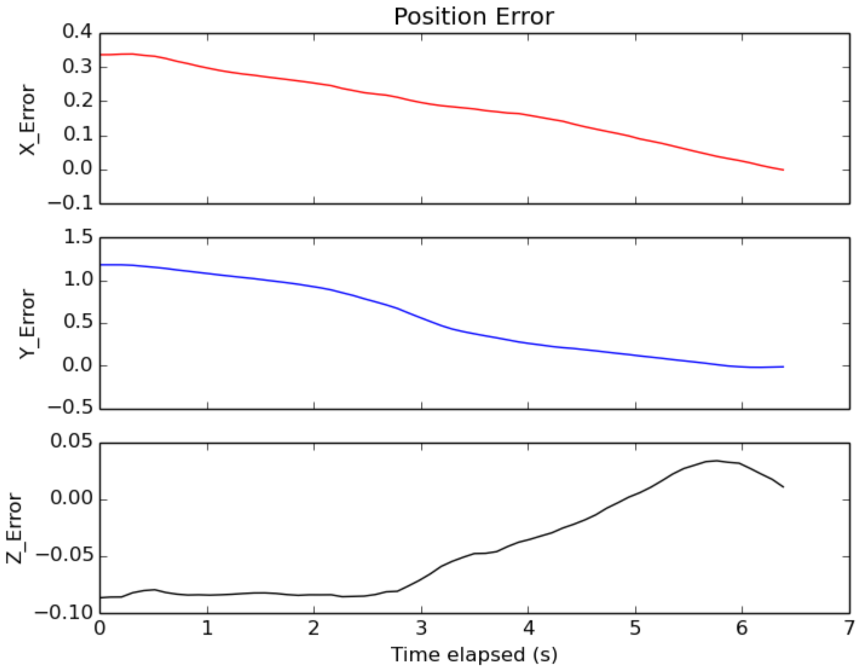


FIGURE 4.20: End-effector position error (Approximate solution method with joint limit avoidance)

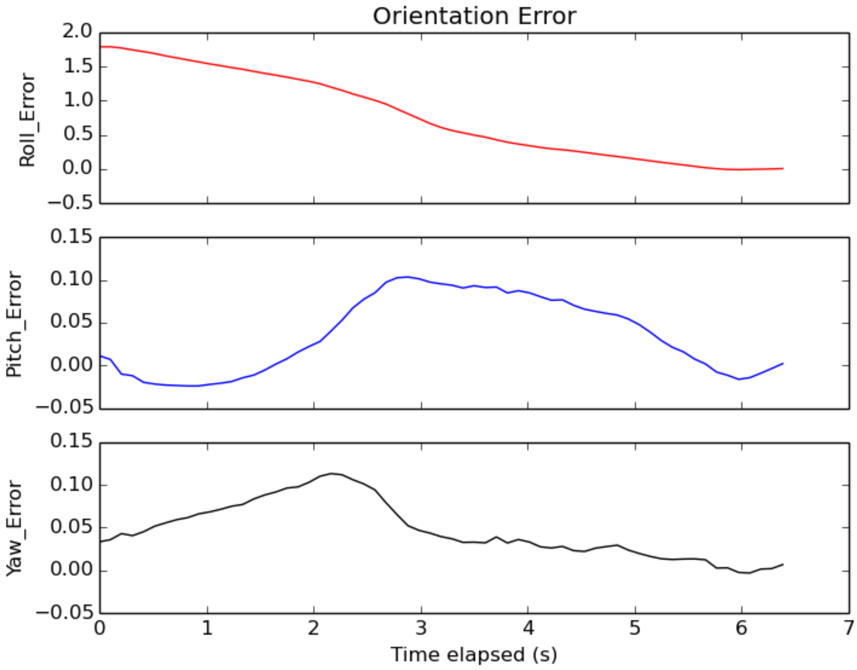


FIGURE 4.21: End-effector orientation error (Approximate solution method with joint limit avoidance)

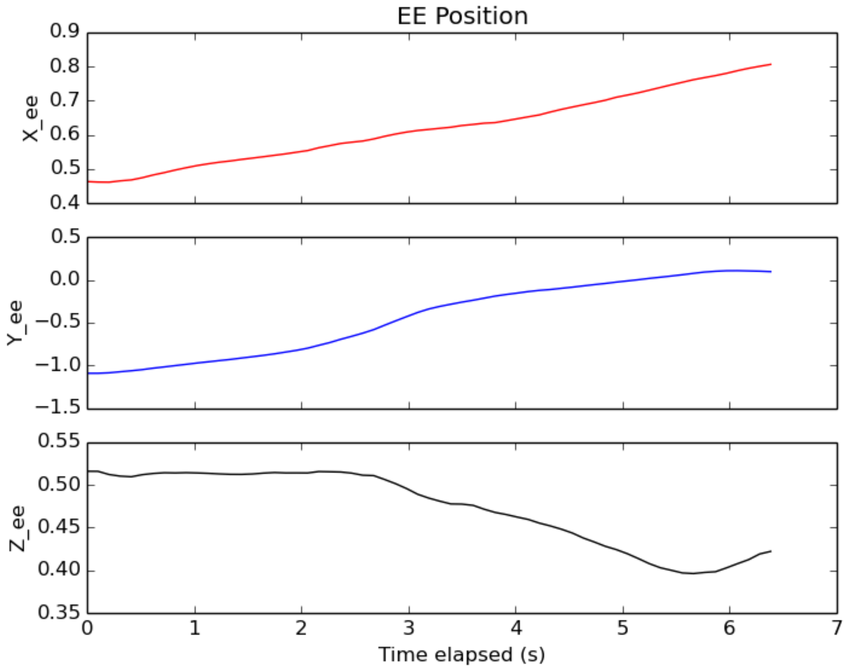


FIGURE 4.22: End-effector position trajectory (Approximate solution method with joint limit avoidance)

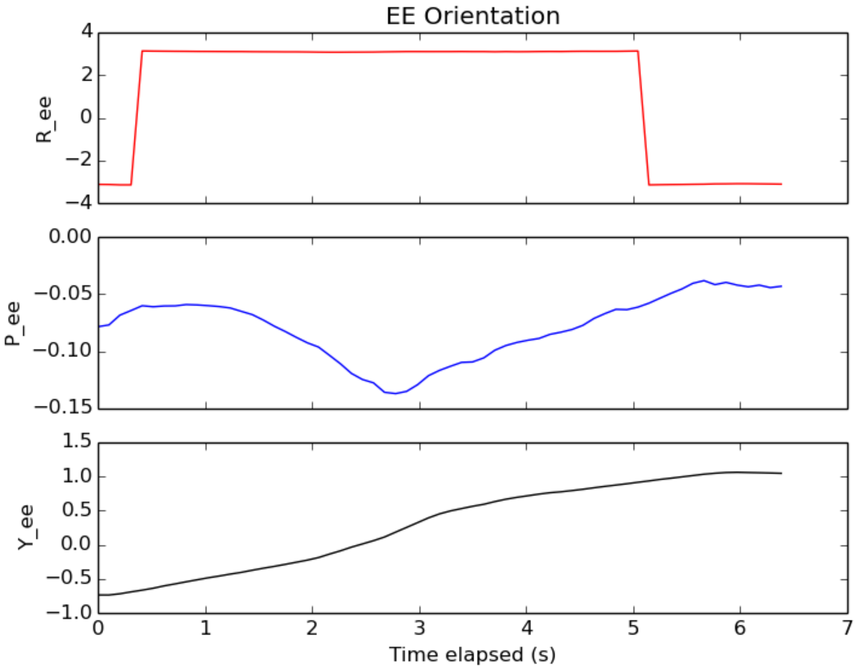


FIGURE 4.23: End-effector orientation trajectory (Approximate solution method with joint limit avoidance)

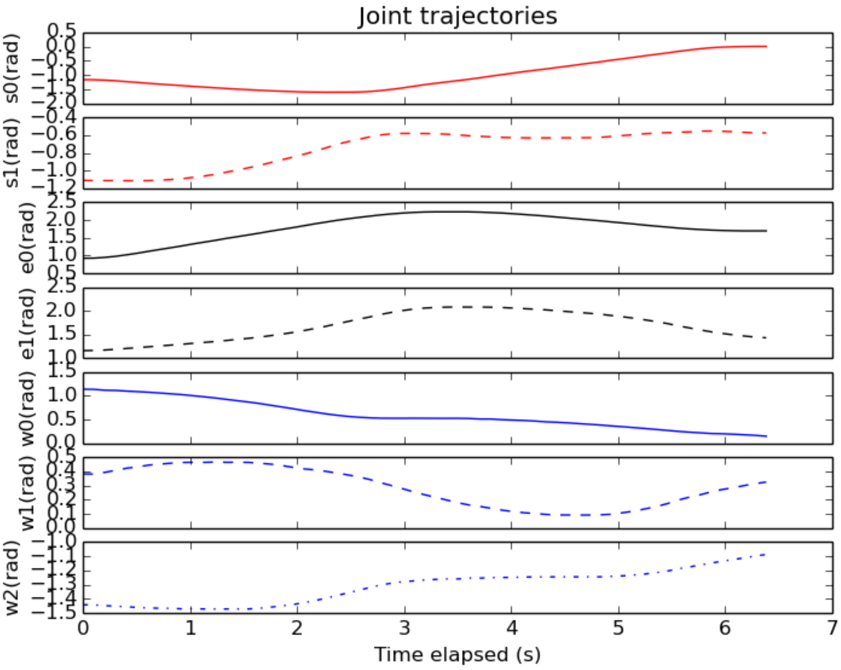
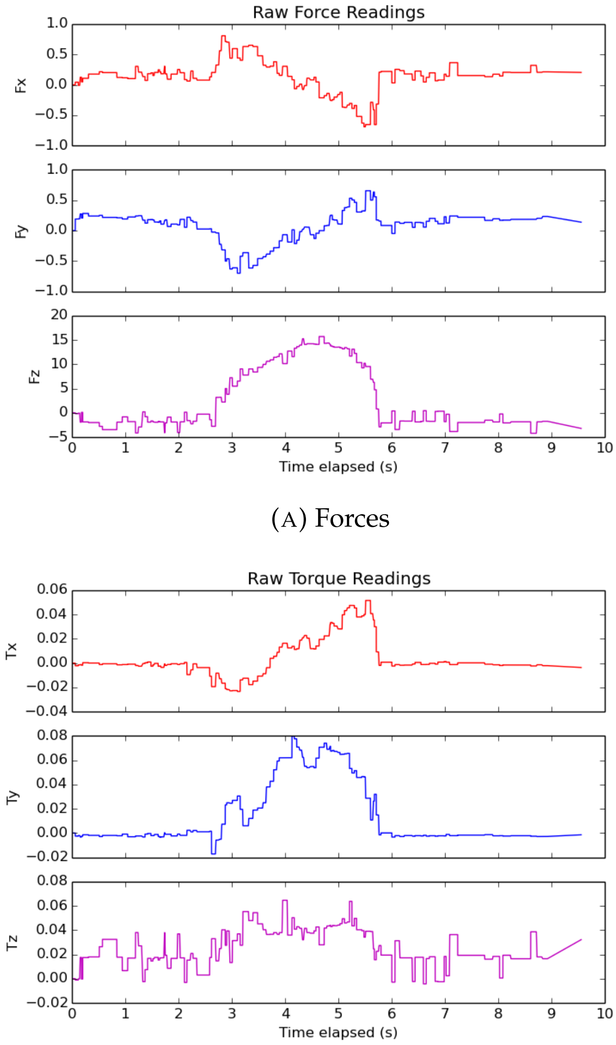


FIGURE 4.24: Joint trajectories (Approximate solution method with joint limit avoidance)

4.3.5 Contact Detection Using Wrist Mounted Force/Torque Sensor

The F/T sensor is calibrated and the raw sensor readings are obtained after gravity compensation, i.e. after compensating the weight of the gripper attached at the tool side of the sensor. Figure 4.25 shows the raw forces and torques reading from the wrist mounted F/T sensor. In Figure 4.25a, Fx, Fy, and Fz represent the forces sensed along x, y, and z axes of the sensor frame respectively . Similarly, in Figure 4.25b, Tx, Ty, and Tz represent the corresponding torques sensed along x, y, and z directions respectively. The raw force sensor readings are again filtered using some threshold (5 in this case) to compensate noises and then used in the contact detection along x, y, and z axes. As shown in Figure 4.26, the sensor is used to detect the contact of the gripper with the object in the environment along the z-axis of the sensor frame, i.e. the peak readings of Fz represent contact along the z-axis.



(A) Forces

(B) Torques

FIGURE 4.25: Raw force and torque readings of wrist mounted F/T sensor

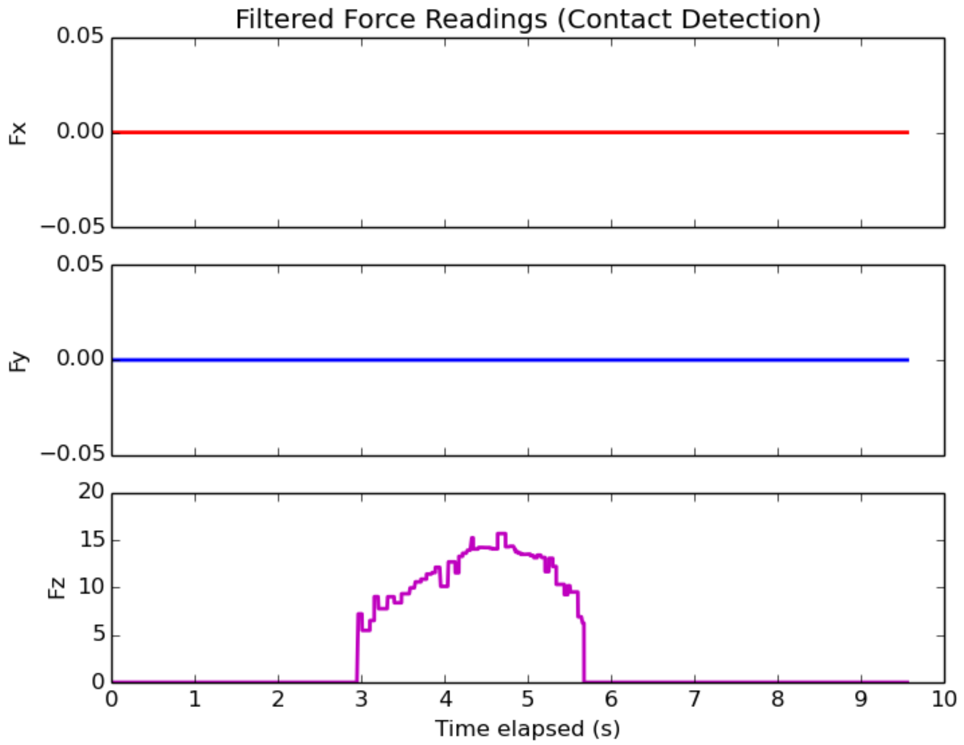


FIGURE 4.26: Contact detection using filtered force readings of F/T sensor

4.4 Comparisons and Discussions

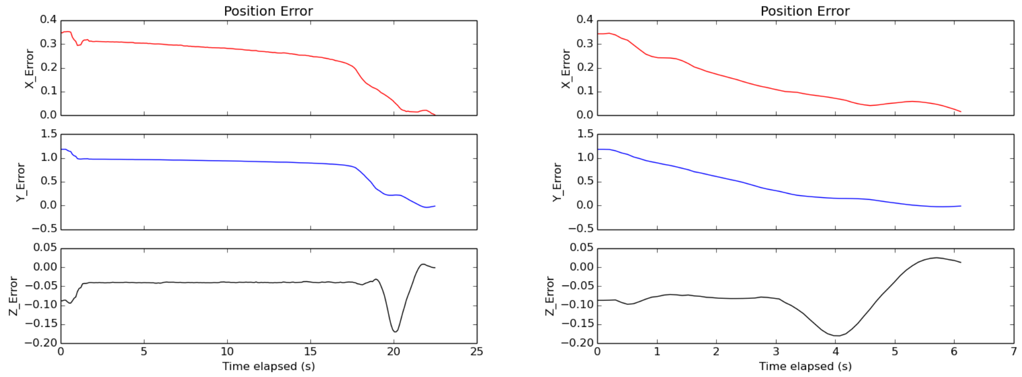
The results obtained by limiting the joint velocities to ± 0.5 radians per second via normalization are presented in Section 4.3. It can be seen that the end-effector reaches its goal position in all the cases, i.e, using exact or approximate solution methods with and without joint limit avoidance. No easily visible differences are noticed between the exact solution method and approximate solution method of redundancy resolution. This is due to the fact that the singularity configuration does not occur in both the cases during the motion from initial pose to goal pose. However, changing the maximum joint velocity limit from 0.5 to 0.8 radians per second keeping other parameters same as in Section 4.3, the results obtained with different redundancy resolution techniques are interesting which are compared as follows.

From Figures 4.27, 4.28, 4.29, 4.30 and 4.31, it can be easily noticed that the time required by the manipulator to reach from initial pose to goal pose is significantly high with exact solution method than with approximate solution method. This is due to the reason that during the motion using exact solution method, the manipulator reaches its singular configuration (or near singular configuration) before reaching the goal. At singular configuration, the end-effector of the manipulator cannot generate velocity components in certain directions, which is not desirable. More specifically, close to a singular position, very large joint rates are needed to generate an end-effector velocity in certain directions which are not physically possible for the joints to afford. Moreover, in the experiment, if any of the joint velocities is greater than the maximum velocity limits of the joints (0.8 radians per sec in this experiment), all the joint velocities are normalized with respect to the maximum joint velocity. This results in the very minimum joint velocities until the manipulator moves away from the singular configuration. Thus, the time taken to reach the goal pose is very high with exact solution method.

However, when using approximate solution method, singularities are avoided by penalizing the high joint rates causing the manipulator not to move close to the singularity posture. Thus, the end effector of the manipulator reaches its goal pose in normal time.

When incorporating joint limit avoidance mechanism within the exact solution and approximate solution methods of redundancy resolution, like in Sections 4.3.4 and 4.3.3, the singular configuration does not occur (in this particular experiment) during the motion from initial pose to goal pose. So, the results obtained from both the methods are similar in this experiment. However, the singular configuration is avoided by default in approximate solution method but might occur during the motion using exact solution method if some parameters are varied. The joint limits are avoided and specially visible for $S0$ joint in Figure 4.35, because the joint limit avoidance for $S0$ joint is intensionally given greater importance (higher weight) than the other (it has an application specific advantage). One can change the weights for individual joints to avoid joint limits based on the requirements.

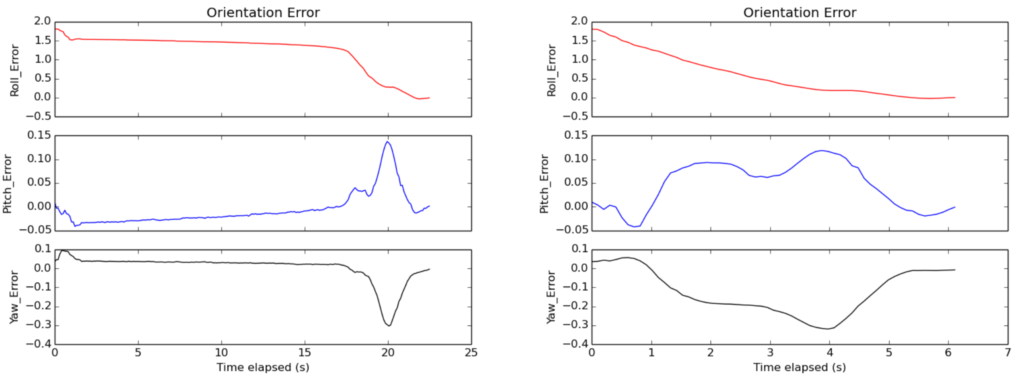
In context to ARC 2017, the kinematic controller that uses approximate solution method of redundancy resolution is used with the functionality of joint limit avoidance. The implemented controller is robust with respect to avoidance of kinematic and algorithmic singularities, and joint limits. The picking and stowing tasks are successfully tested using the controller and are performed within the time frame similar to ARC 2017. The items are grasped and placed in a robust way by ensuring the contact of the gripper with the object using the force feedback from F/T sensor.



(A) Exact Solution Method

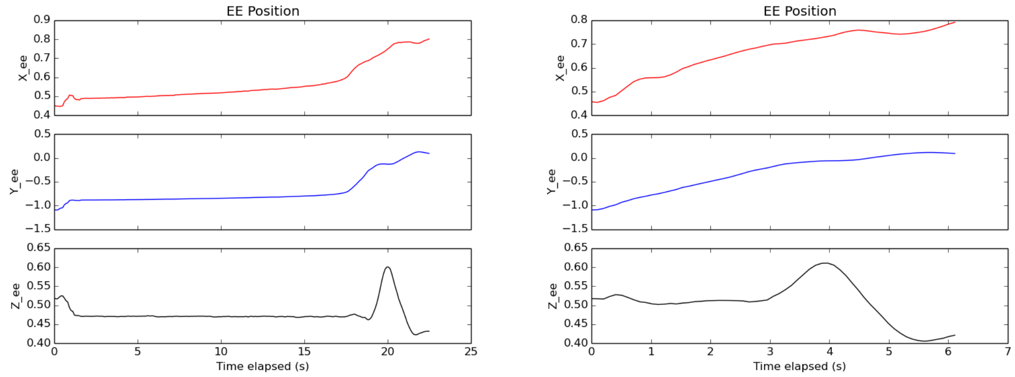
(B) Approximate solution method

FIGURE 4.27: Comparison of end-effector position errors



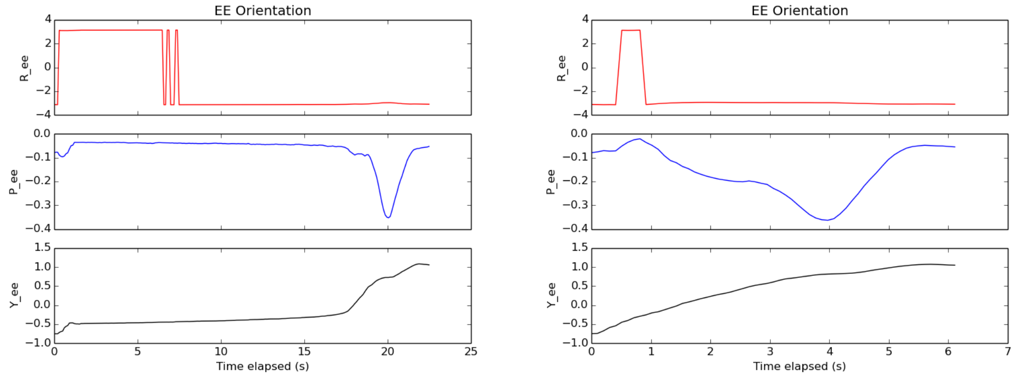
(A) Exact Solution Method (B) Approximate solution method

FIGURE 4.28: Comparison of end-effector orientation errors



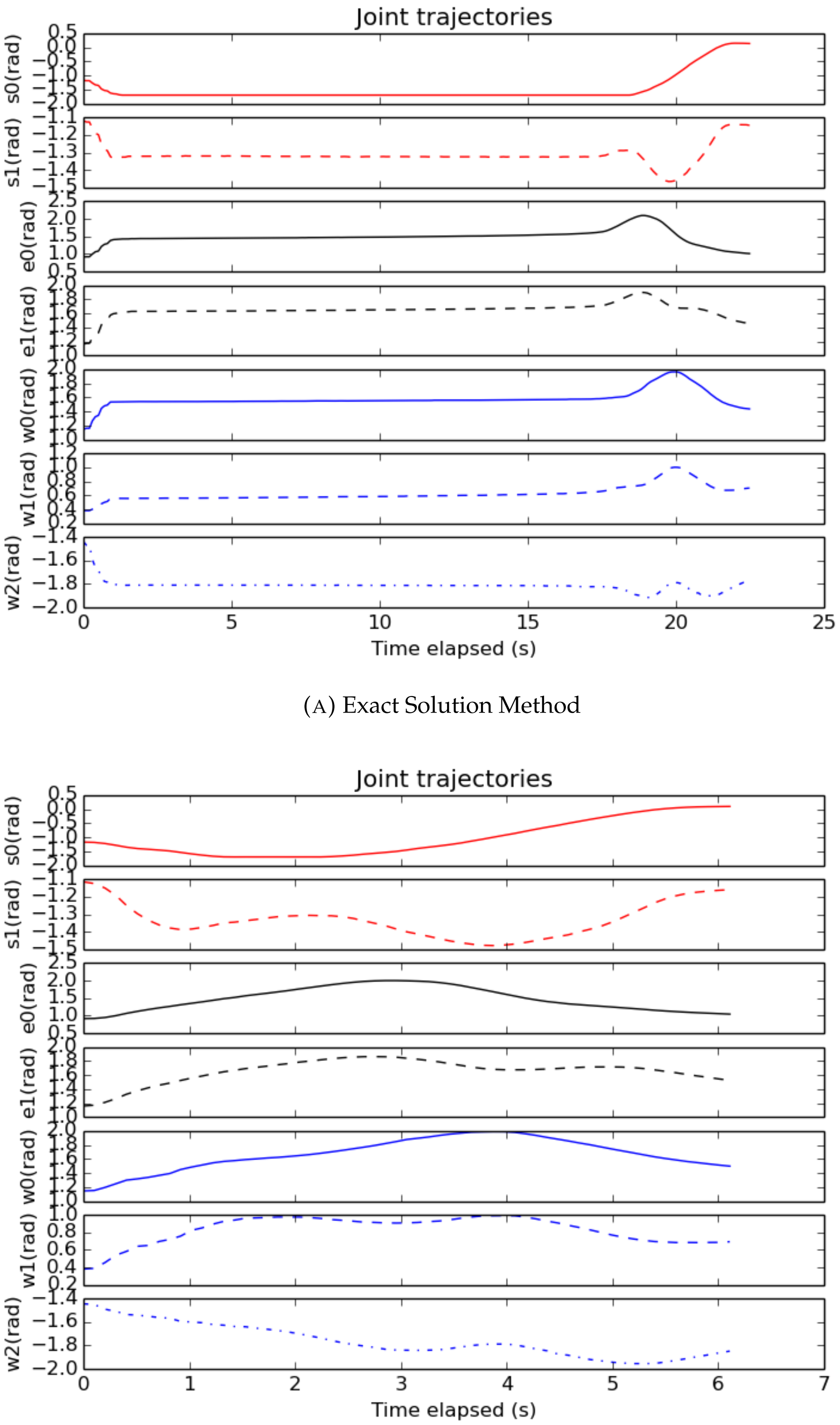
(A) Exact Solution Method (B) Approximate solution method

FIGURE 4.29: Comparison of end-effector position trajectories



(A) Exact Solution Method (B) Approximate solution method

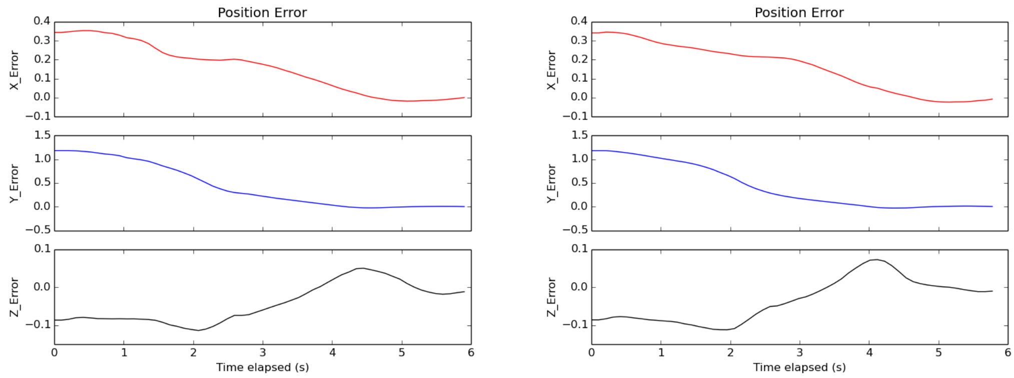
FIGURE 4.30: Comparison of end-effector orientation trajectories



(A) Exact Solution Method

(B) Approximate solution method

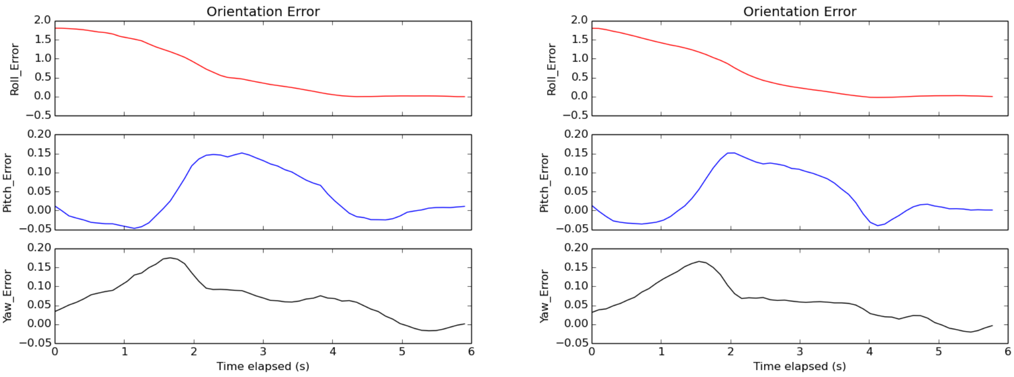
FIGURE 4.31: Comparison of joint trajectories



(A) Exact Solution Method

(B) Approximate solution method

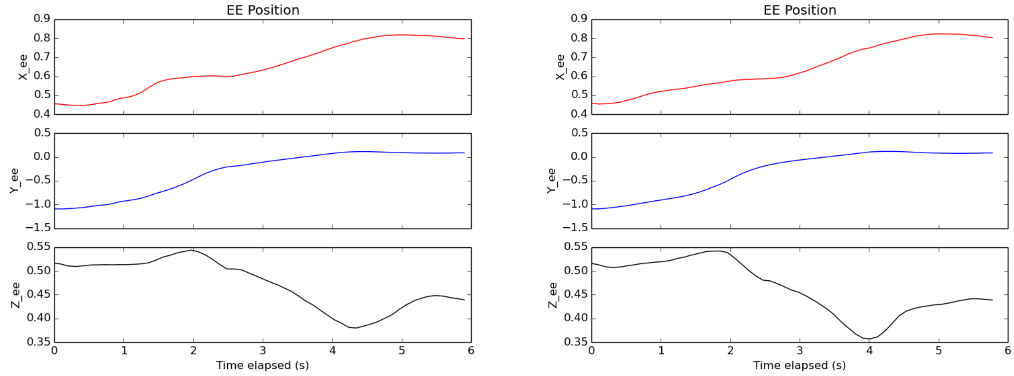
FIGURE 4.32: Comparison of end-effector position errors with joint limit avoidance



(A) Exact Solution Method

(B) Approximate solution method

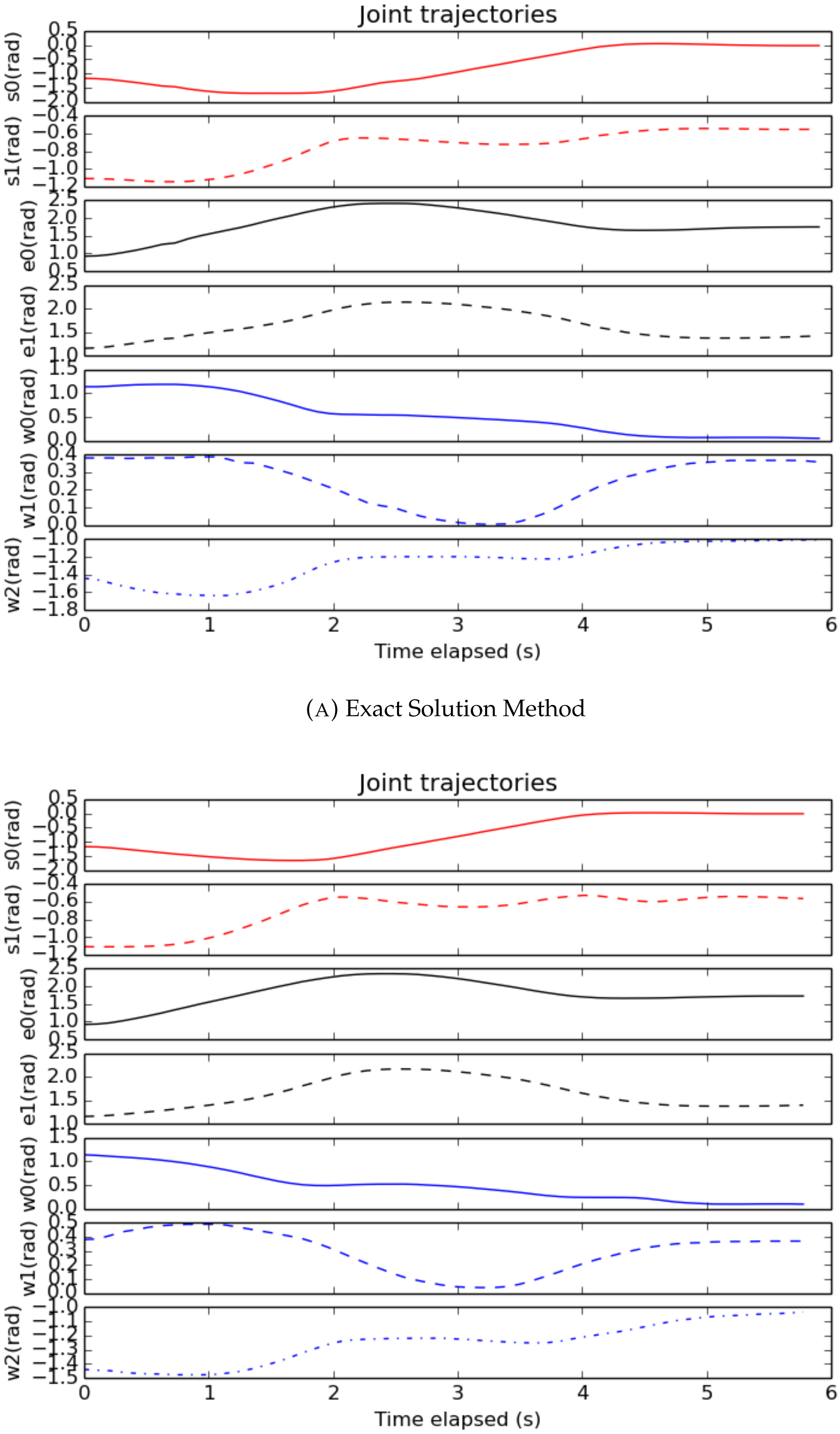
FIGURE 4.33: Comparison of end-effector orientation errors with joint limit avoidance



(A) Exact Solution Method

(B) Approximate solution method

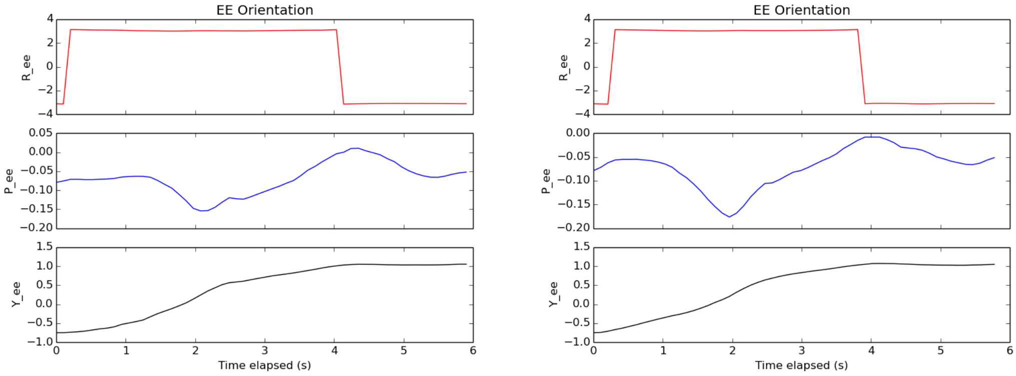
FIGURE 4.34: Comparison of end-effector position trajectories with joint limit avoidance



(A) Exact Solution Method

(B) Approximate solution method

FIGURE 4.35: Comparison of joint trajectories with joint limit avoidance



(A) Exact Solution Method

(B) Approximate solution method

FIGURE 4.36: Comparison of end-effector orientation trajectories with joint limit avoidance

Chapter 5

Conclusions and Future Works

5.1 Conclusions

Different techniques of redundancy resolution at velocity level for the inverse kinematic control of the redundant manipulators are reviewed and most common (exact and approximate solutions) methods are implemented and analyzed with respect to the specific use in the Amazon Robotics Challenge 2017. The approximate solution method of redundancy resolution is selected to avoid singularities and joint limit avoidance is achieved via kinematic optimization. A kinematic controller for redundant manipulator is implemented with the feature of singularity avoidance and joint limit avoidance. The wrist mounted force/torque (F/T) sensor has been calibrated and used for contact detection with objects in the environment and for the robust extraction of the object during picking and stowing operations.

The implemented kinematic controller is successful in reaching the grasp pose and able to perform picking and stowing operations in the scenario similar to ARC 2017. The joint limits and singularities are avoided successfully by the controller during the motion. Contact detection of the gripper with the object (and the environment) is successful using the force feedback from the F/T sensor, resulting in the robust grasping and placing of items. The time taken by the robot to perform picking and stowing operations is acceptable for use in the ARC 2017. More specifically, the implemented kinematic controller is able to perform picking and stowing tasks within the fixed time frame similar to ARC 2017.

5.2 Future Works

In this work, only the redundancy resolution at velocity level is considered. This can be extended to the acceleration level to meet some applications that require joint accelerations. The task of obstacle avoidance can be included in the control loop for kinematically redundant manipulators for use in complex working environments, where obstacle avoidance emerges as an important issue to be addressed in robot motion planning. The calibrated force torque sensor can be used for the force/contact control of robot end-effector in many applications like robot assembly.

Appendix A

Baxter Denavit-Hartenberg (DH) Parameters

The Cartesian reference frame definitions for Baxter′s 7-DOF left and right arms are shown in Figure A.1 and Figure A.2 respectively. Table A.1 and Table A.2 give the associated DH parameters according to Craig [10] convention, known as 'modified DH parameters' for the 7-DOF left and right arms respectively. It can be seen from Tables A.1 and A.2 that the Baxter was designed so that the DH Parameters are identical for both left and right arms so that all kinematics and dynamics equations for one arm apply equally to the other.

TABLE A.1: 7-DOF Left Arm DH Parameters

$\boldsymbol{i}$	$\boldsymbol{\alpha_{i-1}}$	$\boldsymbol{a_{i-1}}$	$\boldsymbol{d_i}$	$\boldsymbol{\theta_i}$
1	0	0	0	θ_1
2	-90°	L_1	0	$\theta_2 + 90°$
3	90°	0	L_2	θ_3
4	-90°	L_3	0	θ_4
5	90°	0	L_4	θ_5
6	-90°	L_5	0	θ_6
7	90°	0	0	θ_7

TABLE A.2: 7-DOF Right Arm DH Parameters

$\boldsymbol{i}$	$\boldsymbol{\alpha_{i-1}}$	$\boldsymbol{a_{i-1}}$	$\boldsymbol{d_i}$	$\boldsymbol{\theta_i}$
1	0	0	0	θ_1
2	-90°	L_1	0	$\theta_2 + 90°$
3	90°	0	L_2	θ_3
4	-90°	L_3	0	θ_4
5	90°	0	L_4	θ_5
6	-90°	L_5	0	θ_6
7	90°	0	0	θ_7

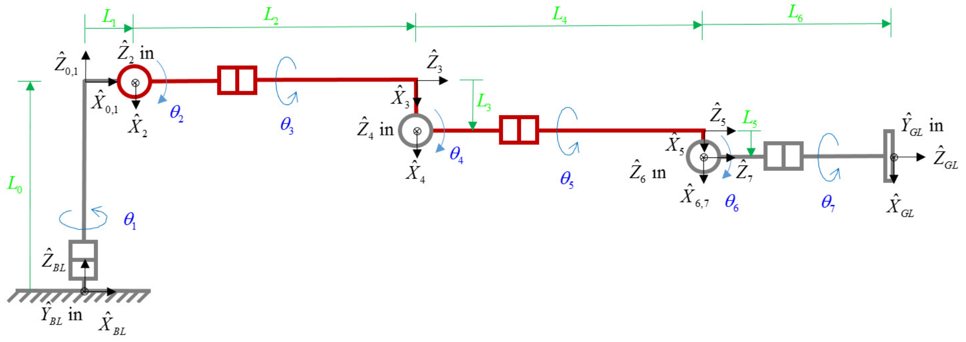


FIGURE A.1: Baxter Left Arm Kinematic Diagram with Coordinate Frames[1]

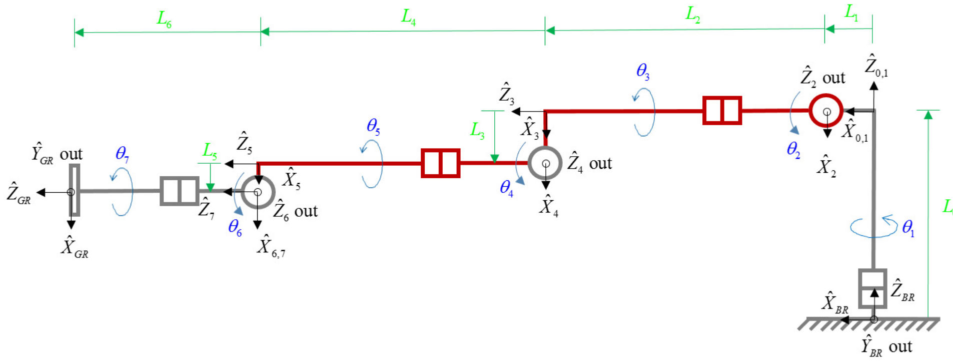


FIGURE A.2: Baxter Right Arm Kinematic Diagram with Coordinate Frames[1]

[1] Source: [57]

Bibliography

[1] Abdelrahem Atawnih, Dimitrios Papageorgiou, and Zoe Doulgeri. "Kinematic control of redundant robots with guaranteed joint limit avoidance". In: *Robotics and Autonomous Systems* 79 (2016), pp. 122–131.

[2] John Baillieul. "Avoiding obstacles and resolving kinematic redundancy". In: *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*. Vol. 3. IEEE. 1986, pp. 1698–1704.

[3] John Baillieul. "Kinematic programming alternatives for redundant manipulators". In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. Vol. 2. IEEE. 1985, pp. 722–728.

[4] John Baillieul, John Hollerbach, and Roger Brockett. "Programming and control of kinematically redundant manipulators". In: *Decision and Control, 1984. The 23rd IEEE Conference on*. Vol. 23. IEEE. 1984, pp. 768–774.

[5] Daniel R Baker and Charles W Wampler. "On the inverse kinematics of redundant manipulators". In: *The International Journal of Robotics Research* 7.2 (1988), pp. 3–21.

[6] Pyung Chang. "A closed-form solution for inverse kinematics of robot manipulators with redundancy". In: *IEEE Journal on Robotics and Automation* 3.5 (1987), pp. 393–403.

[7] Stephen L Chiu. "Task compatibility of manipulator postures". In: *The International Journal of Robotics Research* 7.5 (1988), pp. 13–21.

[8] Richard Colbaugh, Homayoun Seraji, and KL Glass. "Obstacle avoidance for redundant robots using configuration control". In: *Journal of Field Robotics* 6.6 (1989), pp. 721–744.

[9] Nikolaus Correll et al. "Analysis and Observations From the First Amazon Picking Challenge". In: *IEEE Transactions on Automation Science and Engineering* (2016).

[10] John J Craig. *Introduction to robotics: mechanics and control*. Vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.

[11] RV Dubey, JA Euler, and SM Babcock. "An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist". In: *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*. IEEE. 1988, pp. 28–36.

[12] Olav Egeland. "Task-space tracking with redundant manipulators". In: *IEEE Journal on Robotics and Automation* 3.5 (1987), pp. 471–475.

[13] Farbod Fahimi. *Autonomous robots: modeling, path planning, and control*. Vol. 107. Springer Science & Business Media, 2008.

[14] Gene H Golub and Charles F Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[15] Guo-Shing Huang et al. “Inverse kinematics analysis trajectory planning for a robot arm”. In: *Control Conference (ASCC), 2011 8th Asian*. IEEE. 2011, pp. 965–970.

[16] Shuihua Huang et al. “On the Virtual Joints for Kinematic Control of Redundant Manipulators With Multiple Constraints”. In: *IEEE Transactions on Control Systems Technology* (2017).

[17] Pei Jiang, Ji Xiang, and Wei Wei. “Kinematic control for redundant manipulators with time dependent constraints: General-Weighted Least-Norm method”. In: *Control and Decision Conference (2014 CCDC), The 26th Chinese*. IEEE. 2014, pp. 1478–1483.

[18] Kazem Kazerounian and Zhaoyu Wang. “Global versus local optimization in redundancy resolution of robotic manipulators”. In: *The International Journal of Robotics Research* 7.5 (1988), pp. 3–12.

[19] Olivier Kermorgant and François Chaumette. “Dealing with constraints in sensor-based robot control”. In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 244–257.

[20] Charles A Klein. “Use of redundancy in the design of robotic system”. In: *Robotics Research: 2nd Int. Symp., 1985*. Mit Press. 1985, pp. 207–214.

[21] Charles A Klein and Bruce E Blaho. “Dexterity measures for the design and control of kinematically redundant manipulators”. In: *The international journal of robotics research* 6.2 (1987), pp. 72–83.

[22] Charles A Klein and Ching-Hsiang Huang. “Review of pseudoinverse control for use with kinematically redundant manipulators”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1983), pp. 245–250.

[23] Daniel Kubus, Torsten Kroger, and Friedrich M Wahl. “On-line rigid object recognition and pose estimation based on inertial parameters”. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, pp. 1402–1408.

[24] I Kuhlemann et al. “Robust inverse kinematics by configuration control for redundant manipulators with seven DoF”. In: *Control, Automation and Robotics (ICCAR), 2016 2nd International Conference on*. IEEE. 2016, pp. 49–55.

[25] Naveen Kumar et al. “Neural network-based nonlinear tracking control of kinematically redundant robot manipulators”. In: *Mathematical and Computer Modelling* 53.9 (2011), pp. 1889–1901.

[26] Shuai Li, Yunong Zhang, and Long Jin. “Kinematic control of redundant manipulators using neural networks”. In: *IEEE transactions on neural networks and learning systems* (2016).

[27] Alain Liégeois. “Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.12 (1977), pp. 868–871.

[28] Shujun Lu, Jae Heon Chung, and Steven A Velinsky. “Human-robot collision detection and identification based on wrist and base force/torque sensors”. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE. 2005, pp. 3796–3801.

[29] Anthony A Maciejewski and Charles A Klein. “Numerical filtering for the operation of robotic manipulators through kinematically singular configurations”. In: *Journal of Field Robotics* 5.6 (1988), pp. 527–552.

[30] Anthony A Maciejewski and Charles A Klein. “Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments”. In: *The international journal of robotics research* 4.3 (1985), pp. 109–117.

[31] Anthony A Maciejewski and Charles A Klein. “The singular value decomposition: Computation and applications to robotics”. In: *The International journal of robotics research* 8.6 (1989), pp. 63–79.

[32] René V Mayorga and Andrew KC Wong. “A singularities avoidance approach for the optimal local path generation of redundant manipulators”. In: *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*. IEEE. 1988, pp. 49–54.

[33] Yoshihiko Nakamura and Hideo Hanafusa. “Inverse kinematic solutions with singularity robustness for robot manipulator control”. In: *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control* 108 (1986), pp. 163–171.

[34] Yoshihiko Nakamura and Hideo Hanafusa. “Optimal redundancy control of robot manipulators”. In: *The International Journal of Robotics Research* 6.1 (1987), pp. 32–42.

[35] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. “Task-priority based redundancy control of robot manipulators”. In: *The International Journal of Robotics Research* 6.2 (1987), pp. 3–15.

[36] Christian Ott, Alexander Dietrich, and Alin Albu-Schäffer. “Prioritized multitask compliance control of redundant manipulators”. In: *Automatica* 53 (2015), pp. 416–423.

[37] Prem Kumar Patchaikani, Laxmidhar Behera, and Girijesh Prasad. “A single network adaptive critic-based redundancy resolution scheme for robot manipulators”. In: *IEEE Transactions on Industrial Electronics* 59.8 (2012), pp. 3241–3253.

[38] Rajni V Patel and Farshid Shadpey. *Control of redundant robot manipulators: theory and experiments*. Vol. 316. Springer Science & Business Media, 2005.

[39] Tadej Petrič and Leon Žlajpah. “Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem”. In: *Robotics and Autonomous Systems* 61.9 (2013), pp. 948–959.

[40] D.L. Pieper. “The Kinematics of manipulators under Computer control”. In: *Ph. D. Thesis, Stanford University,Department of Mechanical Engineering* (1968).

[41] Marc H Raibert and John J Craig. “Hybrid position/force control of manipulators”. In: *Journal of Dynamic Systems, Measurement, and Control* 102.127 (1981), pp. 126–133.

[42] H Sadjadian, HD Taghirad, and A Fatehi. “Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator”. In: *International Journal of Computational Intelligence* 2.1 (2005), pp. 40–47.

[43] J Kenneth Salisbury. “Active stiffness control of a manipulator in Cartesian coordinates”. In: *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*. Vol. 19. IEEE. 1980, pp. 95–100.

[44] L Sciavicco and B Siciliano. “Solving the inverse kinematic problem for robotic manipulators”. In: *RoManSy 6*. Springer, 1987, pp. 107–114.

[45] Homayoun Seraji. “Task options for redundancy resolution using configuration control”. In: *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*. IEEE. 1991, pp. 2793–2798.

[46] Weimin Shen, Jason Gu, and Evangelos E Milios. “Self-configuration fuzzy system for inverse kinematics of robot manipulators”. In: *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American*. IEEE. 2006, pp. 41–45.

[47] Bruno Siciliano. “Kinematic control of redundant robot manipulators: A tutorial”. In: *Journal of Intelligent & Robotic Systems* 3.3 (1990), pp. 201–212.

[48] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.

[49] Ki Suh and J Hollerbach. “Local versus global torque optimization of redundant manipulators”. In: *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*. Vol. 4. IEEE. 1987, pp. 619–624.

[50] Luigi Villani and Joris De Schutter. “Force control”. In: *Springer handbook of robotics*. Springer, 2016, pp. 195–220.

[51] Rusty Alexander Von Sternberg et al. “GCCF: a generalized contact control framework”. PhD thesis. 2016.

[52] M. Vukobratovic and M. Kirćanski. “A dynamic approach to nominal trajectory synthesis for redundant manipulators”. In: *IEEE transactions on systems, man, and cybernetics* 4 (1984), pp. 580–586.

[53] Charles W Wampler. “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 16.1 (1986), pp. 93–101.

[54] J. Wang and Y. Li. “Comparative analysis for the inverse kinematics of redundant manipulators based on repetitive tracking tasks”. In: *2009 IEEE International Conference on Automation and Logistics*. 2009, pp. 164–169.

[55] Jingguo Wang, Yangmin Li, and Xinhua Zhao. “Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm”. In: *International Journal of Advanced Robotic Systems* 7.4 (2010), pp. 1–9.

[56] Daniel E Whitney. “Resolved motion rate control of manipulators and human prostheses”. In: *IEEE Transactions on man-machine systems* 10.2 (1969), pp. 47–53.

[57] R.L. Williams II. *Baxter Humanoid Robot Kinematics*. Ed. by Internet Publication. 2017. URL: `https://www.ohio.edu/mechanical-faculty/williams/html/pdf/BaxterKinematics.pdf`.

[58] Carlos Canudas de Wit, Bruno Siciliano, and Georges Bastin. *Theory of robot control*. Springer Science & Business Media, 2012.

[59] Ji Xiang, Congwei Zhong, and Wei Wei. “A varied weights method for the kinematic control of redundant manipulators with multiple constraints”. In: *IEEE Transactions on Robotics* 28.2 (2012), pp. 330–340.

[60] Chenguang Yang et al. “Visual servoing control of baxter robot arms with obstacle avoidance using kinematic redundancy”. In: *International Conference on Intelligent Robotics and Applications*. Springer. 2015, pp. 568–580.

[61] Tsuneo Yoshikawa. “Dynamic manipulability of robot manipulators”. In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. Vol. 2. IEEE. 1985, pp. 1033–1038.

[62] Tsuneo Yoshikawa. “Manipulability of robotic mechanisms”. In: *The international journal of Robotics Research* 4.2 (1985), pp. 3–9.