SINGULARITY ANALYSIS AND HANDLING TOWARDS MOBILE MANIPULATION

DENNY NURJANTO OETOMO
B.Eng

A DISSERTATION SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Associate Professor Marcelo H. Ang Jr., for all the guidance and support beyond academic aspects, throughout my period of candidature. Also to Professor Oussama Khatib from Stanford University, for his guidance and inspiration. I would also like to thank Dr. Lim Ser Yong from the Singapore Institute of Manufacturing Technology and my panel of advisors: Associate Professor Teo Chee Leong and Assistant Professor Etienne Burdet from the National University of Singapore. Also to my fellow members of the project U98A031 of the Singapore Institute of Manufacturing Technology: Rodrigo Jamisola, the father of my goddaughter Maanyag, for helping me get started in robotics when I first joined the project and for all the help from the start till the end of the project, Mana Saedan, who was also my flatmate, for his help and his cool handling of even the worst crises, Lim Tao Ming, the computer authority in the lab, for all the help from writing codes to troubleshooting the Sensable PHANToM haptic device, Lim Chee Wang, who is always ready to help, especially in mobile base issues. Not to forget Liaw Hwee Choo for the fundamental work in dynamics and help in machine maintenance, and Leow Yong Peng, who was a lot of help in kinematic analysis of wheel modules, a great source of advice and, most important of all, for introducing me to LaTeX. Last but not the least, my parents for everything I have, and my wife, Lois, for her support and encouragement in everything I strive to achieve in life.

TABLE OF CONTENTS

Page

SUMMARY

Topics covered in this dissertation fall mainly in the general framework of *Mobile Manipulation*. A control algorithm that is capable of a unified force and motion control, based on the Operational Space Formulation [1], was set as the starting platform in the project.

The thesis focused on the problem of singularity. Issues in identification of singularities and singular directions were discussed. These issues are not new, however, certain simplification process is often introduced to reduce the complexity of the identification techniques. Analysis was performed on these simplified methods to evaluate the completeness of resulting solutions.

Two concepts of singularity handling methods were presented. The first was by removing the degenerate components of the task. Certain discontinuity issues associated with this method were analyzed. This method belongs to the category that introduces a division in workspace. The second was to supplement the DOFs lost in singularity with extra "virtual" joints. There is no division of workspace in this category.

The last chapter presents the example of the application of the operational space formulation with singularity compensation, performing an industrial task of polishing the curved surface of an aircraft canopy with no prior knowledge of the surface profile.

The workspace of the manipulator was extended by mounting it on a mobile base. The result is presented in graphs and in videos that are available on the Internet.

NOMENCLATURE

${}^{i}\mathbf{J}$	Jacobian matrix expressed in Frame i
${}^{A}\mathbf{R}_{B}$	3×3 rotation matrix describing the orientation of Frame{B} in Frame{A}. The 1^{st}, 2^{nd}, and 3^{rd} column represent the rotation of X, Y, Z axes of Frame{B} expressed in Frame{A}
$\boldsymbol{\tau}$	Torque command vector to the manipulator joints
$\boldsymbol{\tau}_0$	Torque command vector to be projected into the null space of the Jacobian of the manipulator
$\mathbf{f}$	Operational space force command vector of the operational point in a manipulator
$\mathbf{A}$	Joint space inertia matrix
$\mathbf{B}$	Coriolis and Centrifugal force in joint space
$\mathbf{g}$	Gravity vector in joint space
$\hat{\mathbf{\Lambda}}$	Inertia matrix in operational space
$\boldsymbol{\mu}$	The Coriolis and centrifugal forces in operational space
$\mathbf{p}$	Gravitational vector in operational space
$\mathbf{\Omega}$	Task specification matrix, on which axes are in force and which in motion control
$\mathbf{J}^{\#}$	Generalised inverse of a Jacobian matrix
$\bar{\mathbf{J}}$	Dynamically Consistent Inverse of a Jacobian matrix

s_i	$sin(q_i)$
c_{ab}	$cos(q_a + q_b)$
$\mathbf{J}_V$	Top half of Jacobian matrix J, mapping joint rate to task space velocity of the operational point
$\mathbf{J}_\omega$	Bottom half of Jacobian matrix J, mapping joint rate to the angular velocity of the operational point.
$\mathbf{M}_i$	The $i-th$ minor of a matrix
$s(\mathbf{q})$	The factor in determinant of the Jacobian matrix which is zero at specific singularity
s_0	The threshold value that defines the singular region
$-\nabla v_0(\mathbf{q})$	Gradient descent of potential function $v_0(\mathbf{q})$
$\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$	Columns of the rotation matrix which represent the orientation of each axis of the end-effector with respect to Base Frame
$\mathbf{s}_{1d}$, $\mathbf{s}_{2d}$, $\mathbf{s}_{3d}$	represent the desired values of $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$
$\{A\}$	Frame $\{A\}$ as a label in the diagram.
$\widehat{\mathbf{A}}$	skew symmetric matrix equivalent to a cross product operator, see Section 2.3.
a	Scalar variable a (lower case, regular font).
$\mathbf{a}$	Vector $\mathbf{a}$ (lower case, bold font).
$\mathbf{A}$	Matrix $\mathbf{A}$ (upper case, bold font).
$f(\mathbf{q}, \dot{\mathbf{q}})$	Function f, a function of $\mathbf{q}$ and $\dot{\mathbf{q}}$.

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

The field of robotics has experienced a major boost in the recent years, thanks mainly to the rapid development of computer technology. Now, the field has developed and branched into many different exciting fronts, away from the industrial robotics for which it was originally designed for. Examples include the haptic technology, computer dynamic simulation, vision systems, humanoids (Figure 1.1), and various advanced control algorithms.

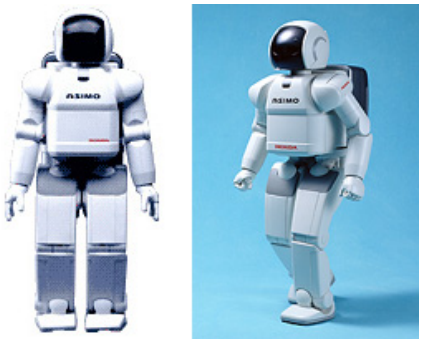


Figure 1.1: Honda Asimo - an example of the recent development in the front of humanoid robots, spearheaded mainly by the Japanese. (From http://www.honda.co.jp/ASIMO/)

Created with the purpose of assisting humans, robot development has more than ever focused on its interaction with the human world. While the usage of robots in the industry has been successful to a certain degree, its implementation is simplified by structuring the environment to suit the operation of the robots. The assembly line in the automotive industry is the classic example, where the cars in production would be placed on a conveyer belt and stationed at exact locations for the robot to operate on. Human operators are kept outside the workspace of the robots for safety reasons and because the robots are not programmed to handle any additional obstacles such as human traffic.

Today, robots are to operate in the "unstructured environment" of the human world. By "unstructured" we mean that we do not always have a static and well defined environment, where the position of everything is known and predefined. There is a need to adapt to the changing environment, to deal with new obstacles (e.g., human traffic) and changing condition (e.g., lighting, temperature, friction, stiffness of environment). It is also necessary to have the versatility to operate on objects with different characteristics, not just identical objects such as in mass production. The system also needs to be safe for human interaction. All these requirements have opened up a wide variety of exciting challenges in the field of robotics.

This dissertation deals with "Mobile Manipulation". A mobile manipulator is a manipulator that is mounted on a mobile base. Mobile manipulation means manipulation while the base is in motion. The robot can now cover a larger workspace due to the increased mobility. It also deals with force and motion control of the manipulator, enabling the robot to interact with the environment through touching and manipulation. Force control enables the robot to have a stable impact with the

environment, and a controlled force to be exerted. While all these are taking place, the dynamics of both arm and base are also compensated to produce a smooth and accurate performance.

This dissertation concentrates mainly on the topic of *singularity*. Singularity is a configuration whereby in its vicinity, robots would have unbounded joint rate for a finite motion in operational space. This is a problem that reduces the usable workspace of a lot of robots. As a part of the "Mobile Manipulation" project, the problem of singularity needs to be addressed. The contribution to the effort of creating a mobile manipulator in the unstructured human environment is by ensuring the robustness and stability of the control algorithm. This improves the safety of the human coming into interaction with the robots. It also improves the dexterity of the manipulator by claiming back workspace originally rendered unusable by singular configurations.

The objective of this Ph.D work is a complete treatment of singularity on existing manipulators, i.e. this work does not involve manipulator designs that minimise singularity. This includes the issues of singularity identification and handling of these singularities. In identification, we evaluate the existing techniques for completeness of solution, especially for redundant manipulators. Some singularity handling techniques are reviewed and proposed. These techniques can mainly be divided into two categories. One category divided the manipulator workspace into singular and non-singular regions, and applies a different control scheme inside the singular regions. The other has a uniform control scheme across the entire workspace that is designed to be singularity robust. Examples were given and issues from both categories were discussed. Chapter 2 covers general background on robotic theory necessary for the

understanding of the work reported in this dissertation. Chapter 3 covers the background on various types of singularities and various documented works in this field. Chapter 4 covers the singularity identification techniques and the issues in the identification of singular direction. Chapter 5 presents a method of handling singularities by removing the degenerate components. Chapter 6 presents a discussion on the reduced degree of freedom inside the singular region and the discontinuity across the boundary. Chapter 7 presents another method of handling singularities by supplying extra joints 'virtually' to supplement the lost DOFs when singularity occurs. Chapter 8 presents an application example in mobile manipulation that utilises all the material covered in the previous chapters. It involves mounting a manipulator arm on a mobile base to perform force/motion control task. Summary of the dissertation is presented in Chapter 9 including suggestions for future work.

The work on mobile manipulation is mainly derived from that of Oussama Khatib's at Stanford University. The operational space formulation [1] was chosen as the working platform in this project.

The contributions of this Ph.D work are:

- **Completeness of Solution in Singularity Identification**

  Several methods have been proposed in the past and often involves separating the Jacobian matrix into top and bottom halves for the purpose of singularity identification. Analysis was performed on the completeness of the set of solution given and some amendments to the technique are proposed.

- **Singularity Handling in Force and Motion Control**

  A formulation of singularity handling method in operational space framework,

capable of handling force and motion control. Null space motion was also utilized in assisting motion in the degenerate direction inside singular region.

- **Physical Interpretation and Usage of Singular Value Decomposition in Singular Handling of Manipulators**

  Although the usage of Singular Value Decomposition (SVD) in singularity analysis and handling is widespread, in this dissertation, we have included further analysis the topic. A short section is included to numerically define the singular directions of a manipulator in singular configuration. The SVD is also used to improve the singularity handling technique presented in Chapter 5.

- **The resulting issues from the reduced degree of freedom in the singular region and the discontinuity across the singular boundary**

  Inside the singular region, the manipulator task is specified only in the subspace of the workspace that is not degenerate. Inside this region, motion in singular direction is no longer performed, resulting in an accumulated error and discontinuity and jerkiness in motion as the manipulator leaves the singular region. The problem is analysed and handling methods were proposed.

- **Virtual Joints**

  Singularity handling was done by virtual joints. It was proposed that extra "virtual joints" are added to the system to compensate for the lost DOFs during singularity. The concept was implemented and verified in real-time experiments on PUMA 560.

- **Integration of Arm and Base Units with Different Control Modes**

  A method was proposed in mixing a torque controlled arm and a velocity controlled base. This can be thought of a "macro-mini" structure where the velocity controlled base is of slower servo loop and therefore slower response compared to the torque controlled arm. Also velocity control in itself compromises the performance of the manipulator, as it is not capable of force/motion control.

The next chapter will briefly go over the necessary background on the theories and methods used in this dissertation.

CHAPTER 2

BACKGROUND THEORY I: FORCE AND MOTION CONTROL OF MANIPULATORS

2.1 Chapter Overview

This chapter covers the necessary background theory that formed the starting point of the project. The main ideas were derived from the work of Oussama Khatib's from Stanford University [1, 3, 4], who was also the external advisor to this project. This provided a good starting platform to the project and ideas were developed to expand the theories and to implement the ideas into real tasks.

The main sections of the chapter include a brief summary of Operational Space Formulation [1] and Redundancy and Null Space Theory [4, 5, 6]. Other ideas in redundancy resolutions were also explored and presented in this chapter.

This section covers only the important parts of robotics background theory that are crucial in this dissertation. Further reading can be found in textbooks such as [2, 7, 8, 9, 10, 11].

2.2 Operational Space Formulation

The Operational Space Formulation [1] is a control approach where free motion and contact forces are expressed in operational space (Cartesian space as seen from

the end-effector or tool), generated by including the dynamic parameters of the manipulator in the desired control forces. This force is then transformed into equivalent torque values to be exerted by each joint to describe the desired force at the end effector.

The force is obtained by multiplying mass/inertia of the robot with the desired acceleration. The mass/inertia of the robot can be obtained by experiments as described in [12, 13] and can also be verified in [14]. In free motion, the desired acceleration is generated by the control law that minimises the error between the desired and the actual trajectories. Other dynamic parameters can be included into the generated force, such as the Gravity, Coriolis, and Centrifugal forces to better model the dynamics of the robot.

An obvious advantage of this formulation is that it is a very natural framework for combined position and force control, which is used when the end effector comes into contact with the environment. Forces are generally expressed in the Cartesian space, and having free motion generated as forces that draws the mass of the manipulator in the Cartesian space provides an elegant framework for a hybrid motion/force control.

The total force $\mathbf{f}$ is therefore a combination of the force for free motion control and force for constrained motion (force control). It is then converted to joint torques by

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{J}^T\mathbf{f} + \mathcal{N}^T\boldsymbol{\tau}_0 \\ \boldsymbol{\mathcal{N}} &= [\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}] \end{aligned} \tag{2.1}$$

where $\boldsymbol{\tau}$ is the joint torque command vector, and $\mathbf{J}$ is the Jacobian matrix. $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\tau}_0$ are used to control the null space motion of the Jacobian and is useful when the manipulator is redundant with respect to the task. They will be elaborated in the later parts of the chapter. $\mathbf{J}^{\#}$ is a generalised inverse of the $\mathbf{J}$ matrix. In our

experiment, the dynamically consistent inverse $\bar{\mathbf{J}}$ [4] is utilised. The definition of generalised inverse and the dynamically consistent inverse and its further explanation are given in Section 2.5.

2.2.1 Motion Control

Free motion, in the Operational Space Formulation, is generated by a virtual force vector from the present to the desired position, taking into account the dynamics of the manipulator. This is similar to the well-known computed-torque control [15] except that it is now done in operational space. The control law to generate the required force is computed from the required acceleration, $\mathbf{f}^*_{motion}$:

$$\mathbf{f}^*_{motion} = \mathbf{I}\ddot{\mathbf{x}}_d - \mathbf{K}_{v,motion}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - \mathbf{K}_{p,motion}(\mathbf{x} - \mathbf{x}_d) \tag{2.2}$$

$\mathbf{f}^*_{motion}$ is then multiplied with the inertia matrix $\hat{\mathbf{\Lambda}}$, and added with the Coriolis and Centrifugal forces $\hat{\boldsymbol{\mu}}$ and gravitational vector $\hat{\mathbf{p}}$, to yield the required force:

$$\mathbf{f}_{motion} = \hat{\mathbf{\Lambda}}(\mathbf{x})\mathbf{f}^*_{motion} + \hat{\boldsymbol{\mu}}(\mathbf{x}, \dot{\mathbf{x}}) + \hat{\mathbf{p}}(\mathbf{x}) \tag{2.3}$$

The operational space control can be compared with computed-torque control in joint space, which is described as:

$$\boldsymbol{\tau} = \hat{\mathbf{A}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q}) \tag{2.4}$$

where $\hat{\mathbf{A}}$ is the joint space inertial matrix of the manipulator, $\hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centrifugal vector, and $\hat{\mathbf{g}}$ is the gravity compensation vector in joint space. Methods of dynamics identification can be found in [12] and [16, 13]. In the work involved in this dissertation, we use the PUMA 560 manipulator as a test bed. The dynamic model of PUMA 560 is obtained from [13].

These dynamic parameters can then be translated into task space for use in Equation 2.3 by:

$$\begin{array}{lcl} \hat{\mathbf{\Lambda}}(\mathbf{x}) & = & [\mathbf{J}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{J}^T(\mathbf{q})]^{-1} \\ \hat{\boldsymbol{\mu}}(\mathbf{x},\dot{\mathbf{x}}) & = & [\mathbf{J}^{-T}(\mathbf{q})\hat{\mathbf{B}}(\mathbf{q}) - \mathbf{\Lambda}(\mathbf{q})\hat{\mathbf{H}}(\mathbf{q})][\dot{\mathbf{q}}\dot{\mathbf{q}}] \\ \hat{\mathbf{p}}(\mathbf{x}) & = & \mathbf{J}^{-T}(\mathbf{q})\hat{\mathbf{g}}(\mathbf{q}) \end{array} \tag{2.5}$$

where $\hat{\mathbf{H}}(\mathbf{q})[\dot{\mathbf{q}}\dot{\mathbf{q}}] = \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}$. This is valid for all serial, including redundant, manipulators. Note that Equation 2.5 is only valid if the manipulator is at a non-singular configuration.

The " ˆ " above the parameter represents our estimate of actual dynamic parameters. The actual dynamic model of the robot is represented by:

$$\mathbf{f}_{motion} = \mathbf{\Lambda}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}(\mathbf{x}) \tag{2.6}$$

2.2.2 Force Control

As the robot end-effector is in contact with the environment, reaction forces and moments are generated at the end-effector. These forces/moments are then transmitted to the robot joints where the drive torques can be generated to impose the desired contact forces/moments to the robot environment.

The force control in operational space can be transformed to the robot joint space by the same transformation as the operational space motion control.

The operational space force applied at the end-effector can be expressed as,

$$\mathbf{f}_{force} = \hat{\Lambda}(x)\mathbf{f}^*_{force} + \hat{\boldsymbol{\mu}}(\mathbf{x},\dot{\mathbf{x}}) + \hat{\mathbf{p}}(x) + \mathbf{f}_{contact} \tag{2.7}$$

where,

$$\mathbf{f}^*_{force} = \mathbf{K}_{p,force}(\mathbf{f}_d - \mathbf{f}_{contact}) + \mathbf{K}_{i,force}\sum(\mathbf{f}_d - \mathbf{f}_{contact}). \tag{2.8}$$

is the control law and $\mathbf{f}_{contact}$ is the force exerted on the environment and is related to the force sensor reading, $\mathbf{f}_{sensor}$, by

$$\mathbf{f}_{contact} = -\mathbf{f}_{sensor} \tag{2.9}$$

Note that the force sensor reading is the force exerted by the environment on the end-effector.

The $\hat{\boldsymbol{\mu}}$ and $\hat{\mathbf{p}}$ vectors are the Coriolis and centrifugal vector and gravitational vector as defined in motion control. With contact to the environment, the actual dynamic model becomes

$$\mathbf{f}_{force} = \boldsymbol{\Lambda}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{p}(\mathbf{x}) + \mathbf{f}_{contact} \tag{2.10}$$

2.2.3 Unified Force and Motion Control

In unified force and motion control, first the task is defined: which degrees-of-freedom are assigned to force control and which are to motion control. Appropriate control algorithms are then applied respectively.

The resulting force and motion control is done by selecting the desired force or motion response of the robot and adding them together to get the effective robot response (Figure 2.1). This is expressed as,

$$\mathbf{f} = \mathbf{f}_{motion} + \mathbf{f}_{force} \tag{2.11}$$

where

$$\mathbf{f}_{motion} = \hat{\boldsymbol{\Lambda}}(\mathbf{x})\boldsymbol{\Omega}\mathbf{f}^*_{motion} + \hat{\boldsymbol{\mu}}(\mathbf{x}, \dot{\mathbf{x}}) + \hat{\mathbf{p}}(\mathbf{x}) \tag{2.12}$$

and

$$\mathbf{f}_{force} = \hat{\boldsymbol{\Lambda}}(\mathbf{x})\bar{\boldsymbol{\Omega}}\mathbf{f}^*_{force} - \mathbf{f}_{sensor}. \tag{2.13}$$

$\mathbf{f}^*_{motion}$ (Equation 2.2) and $\mathbf{f}^*_{force}$ (Equation 2.8) are the force applied for motion and force control respectively. $\mathbf{\Omega}$ and $\bar{\mathbf{\Omega}}$ are the selection matrices to switch the application between force or motion whichever is desired and to specify the direction of application. $\hat{\boldsymbol{\mu}}(x, \vartheta)$ represents the Coriolis and centrifugal forces and $\hat{\mathbf{p}}(x)$ the Gravitational force, which are the same for force and motion control, and are therefore only included once.

To specify the selection matrices, consider a reference Frame $\{P\}$ at the operational point that is always parallel to the base (global) reference Frame $\{O\}$ (see Figure 2.1). We then consider an operational space (tool) force Frame $\{T\}$ whose orientation is obtained from Frame {P} by the 3×3 rotation matrix ${}^P\mathbf{R}_T$. Frame $\{T\}$ is attached to the end-effector while the origin of Frame $\{P\}$ translates with the operational point and always coincides with the origin of Frame $\{T\}$.

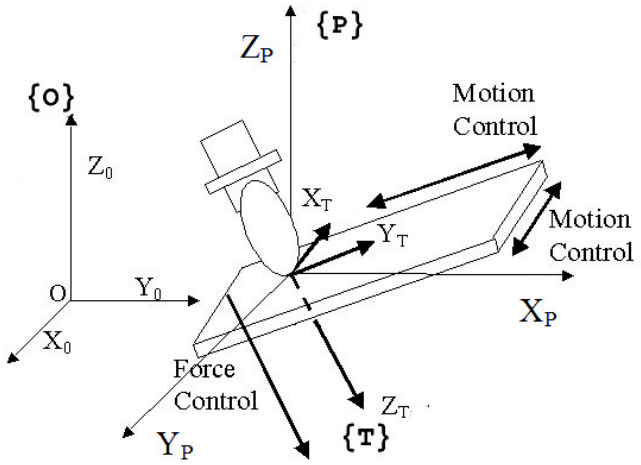


Figure 2.1: Tool frame assignment

The generalized *task specification matrices* $\mathbf{\Omega}$ is then defined as

$$\mathbf{\Omega} = \begin{pmatrix} {}^{P}\mathbf{R}_{T}^{T}\boldsymbol{\mathcal{S}}_{\mathcal{F}}^{T}\mathbf{R}_{P} & 0 \\ 0 & {}^{P}\mathbf{R}_{T}^{T}\boldsymbol{\mathcal{S}}_{\mathcal{M}}^{T}\mathbf{R}_{P} \end{pmatrix} \tag{2.14}$$

where

$$\boldsymbol{\mathcal{S}}_{\mathcal{F}} = \begin{pmatrix} \sigma_{FX} & 0 & 0 \\ 0 & \sigma_{FY} & 0 \\ 0 & 0 & \sigma_{FZ} \end{pmatrix} \tag{2.15}$$

$$\boldsymbol{\mathcal{S}}_{\mathcal{M}} = \begin{pmatrix} \sigma_{MX} & 0 & 0 \\ 0 & \sigma_{MY} & 0 \\ 0 & 0 & \sigma_{MZ} \end{pmatrix} \tag{2.16}$$

and

σ_{FX}, σ_{FY}, σ_{FZ}, σ_{MX}, σ_{MY}, σ_{MZ} are binary values where "1" signifies application of free motion (motion control) along the corresponding axis and "0" for constraint motion (force control) along the corresponding axis.

Equation 2.14 was derived to consistently match the frames that different components are expressed in. $\boldsymbol{\mathcal{S}}_{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}_{\mathcal{F}}$ are expressed in the end-effector frame (Frame{T}. However, $\mathbf{f}_{motion}^{*}$ and $\mathbf{f}_{force}^{*}$ are all expressed in Frame{0}, consistent with system dynamics expressed in Frame {P} (which is parallel to Frame{0})(Equations 2.6 to 2.10). Therefore, they have to be first transformed to Frame{T} (by ${}^{T}\mathbf{R}_{P}$) before the application of $\boldsymbol{\mathcal{S}}$ 2.14. They are then transformed back to Frame{P} by ${}^{P}\mathbf{R}_{T}$ after the application of $\boldsymbol{\mathcal{S}}$.

$\bar{\mathbf{\Omega}}$ is obtained using $\bar{\boldsymbol{\mathcal{S}}}_{\mathcal{F}}$ and $\bar{\boldsymbol{\mathcal{S}}}_{\mathcal{M}}$ which are the complements of $\boldsymbol{\mathcal{S}}_{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}_{\mathcal{M}}$.

When $\mathbf{f}_{motion}$ and $\mathbf{f}_{force}$ are obtained, the combined $\mathbf{f}$ is sent as torque by Equation 2.1.

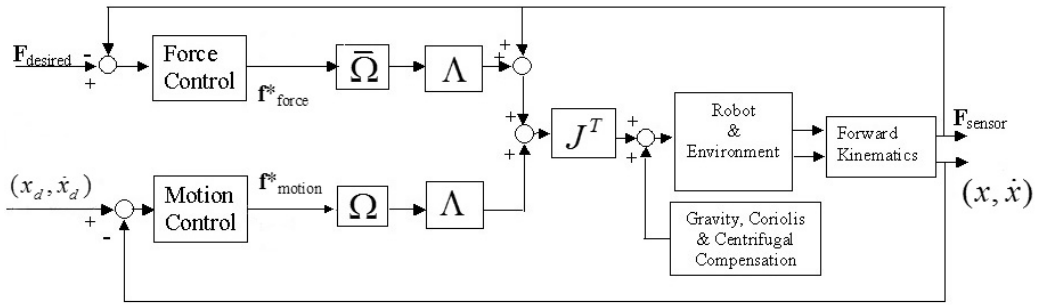


Figure 2.2: The schematic diagram of the operational space formulation

The schematic diagram of the operational space formulation is summarised in Figure 2.2. The operational space formulation is capable of unified force and motion control. Desired contact force with the environment and the desired end-effector motion is generated by task specification. The task specification also includes the description of which degrees-of-freedom are to be assigned to force control and which to motion control. The control law that compares the input and the generated output forces and motion at the end-effector provides the actuation command in task space required to close the tracking error. The equations are reproduced below for convenience.

$$\mathbf{f}^*_{motion} = \mathbf{I}\ddot{\mathbf{x}}_d - \mathbf{K}_{v,motion}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - \mathbf{K}_{p,motion}(\mathbf{x} - \mathbf{x}_d) \tag{2.2}$$

$$\mathbf{f}^*_{force} = \mathbf{K}_{p,force}(\mathbf{f}_d - \mathbf{f}_{contact}) + \mathbf{K}_{i,force}\sum(\mathbf{f}_d - \mathbf{f}_{contact}) \tag{2.8}$$

These commands are then compensated for the dynamic effect according to the dynamic model of the manipulator and assigned to its associated degrees-of-freedom. The resulting forces are added together to form the total force to be displayed at the

end-effector.

$$\mathbf{f}_{motion} = \hat{\mathbf{\Lambda}}(\mathbf{x})\mathbf{\Omega}\mathbf{f}^*_{motion} + \hat{\boldsymbol{\mu}}(\mathbf{x}, \dot{\mathbf{x}}) + \hat{\mathbf{p}}(\mathbf{x}) \tag{2.12}$$

$$\mathbf{f}_{force} = \hat{\mathbf{\Lambda}}(\mathbf{x})\bar{\mathbf{\Omega}}\mathbf{f}^*_{force} - \mathbf{f}_{sensor} \tag{2.13}$$

$$\mathbf{f} = \mathbf{f}_{motion} + \mathbf{f}_{force} \tag{2.11}$$

The force is then converted to joint space command to be sent to the respective joints, by:

$$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{f} + \mathcal{N}^T\boldsymbol{\tau}_0 \tag{2.1}$$

For more details in unified motion and force control, please refer to [1].

2.3 Decoupling of the Jacobian Matrix

The Jacobian matrix is defined as the mapping from joint velocities to the end effector linear velocity $\dot{\mathbf{p}}$ and angular velocity $\boldsymbol{\omega}$ [8], where:

$$\begin{pmatrix} \dot{\mathbf{p}} \\ --- \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_1 \\ --- \\ \mathbf{J}_2 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{pmatrix} \tag{2.17}$$

where $\dot{q}_1, \dot{q}_2, ...\dot{q}_n$ are the joint velocities, and $\mathbf{J}_1$ $(3 \times n)$ and $\mathbf{J}_2$ $(3 \times n)$ are the Jacobian matrices for the translation and orientation of the end-effector.

In an anthropomorphic arm, a spherical wrist is attached at the end of the manipulator. This is a very common design in manipulator arms, in industrial or research laboratory robots. A spherical wrist is one with 3 DoF where the three axes of rotation intersect one another at one point. This point is often called the *wrist point*, and is often used as the operational point of the manipulator arm in the analysis of

the Jacobian matrix. Locating the operational point at the wrist would simplify the Jacobian matrix to:

$$\begin{pmatrix} \dot{\mathbf{p}} \\ - - - \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{11} & | & \mathbf{0}_{3\times 3} \\ - - - - - & | & - - - - - \\ \mathbf{J}_{21} & | & \mathbf{J}_{22} \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} \tag{2.18}$$

This is because any motion in the last three joints which form the spherical wrist does not contribute to changing the position of the operational point, since the operational point is at the wrist.

When a tool is attached to the end-effector and the operational point is now at the tip of the tool, a transformation matrix can be defined to extend the Jacobian to include the extra distance.

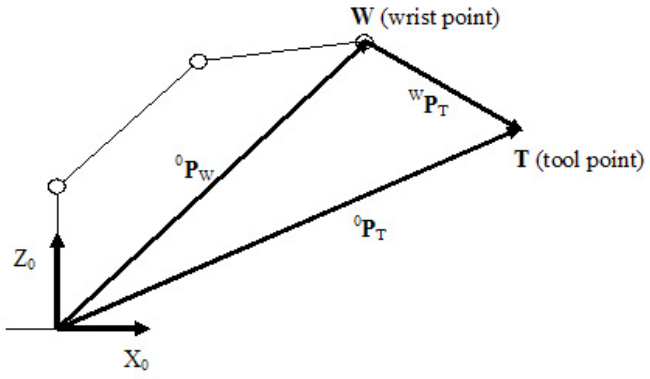


Figure 2.3: When its operating point is extended from the wrist point (W) to the tool point (T), a simple transformation matrix can be defined to relate the Jacobian matrices at the wrist point and the tool point.

Supposed ${}^{0}\mathbf{p}_{W}$ and ${}^{0}\mathbf{p}_{T}$ are the position vectors of the wrist point and the tool point with respect to Frame $\{0\}$ respectively, and ${}^{W}\mathbf{p}_{T}$ is the position vector of the

tool point with respect to the wrist (see Figure 2.3), then

$$\begin{pmatrix} {}^0\dot{\mathbf{p}}_T \\ {}^0\boldsymbol{\omega}_T \end{pmatrix} = \begin{pmatrix} {}^0\dot{\mathbf{p}}_W + {}^0\boldsymbol{\omega}_W \times ({}^0\mathbf{R}_W{}^W\mathbf{p}_T) \\ {}^0\boldsymbol{\omega}_W \end{pmatrix} \tag{2.19}$$

where ${}^0\mathbf{R}_W$ is the rotation matrix of the end-effector with respect to the Frame $\{0\}$ and ${}^0\boldsymbol{\omega}_W$ is the angular velocity of the wrist point W with respect to Frame $\{0\}$.

Using the cross product property $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$, then ${}^0\boldsymbol{\omega}_W \times ({}^0\mathbf{R}_W{}^W\mathbf{p}_T)$ can be written as $-{}^0\mathbf{R}_W{}^W\mathbf{p}_T \times {}^0\boldsymbol{\omega}_W$. Using the skew symmetric matrix notation, it can be re-written as: $({}^0\mathbf{R}_W\widehat{.}{}^W\mathbf{p}_T).{}^0\boldsymbol{\omega}_T$

$\widehat{\boldsymbol{\omega}}$ is the skew symmetric matrix equivalent to a cross product operator where:

$$\widehat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{pmatrix} \tag{2.20}$$

After rearranging, we obtain:

$$\begin{pmatrix} {}^0\dot{\mathbf{p}}_T \\ {}^0\boldsymbol{\omega}_T \end{pmatrix} = \begin{pmatrix} \mathbf{I} & -(\widehat{{}^0\mathbf{R}_W{}^W\mathbf{p}_T}) \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} {}^0\dot{\mathbf{p}}_W \\ {}^0\boldsymbol{\omega}_W \end{pmatrix} \tag{2.21}$$

or

$$\begin{aligned} {}^0\mathbf{J}_T &= \underbrace{\begin{pmatrix} \mathbf{I} & -(\widehat{({}^0R_W)^W\mathbf{p}_T}) \\ 0 & \mathbf{I} \end{pmatrix}}_{\mathbf{J}^*} {}^0\mathbf{J}_W \\ {}^0\mathbf{J}_T &= \mathbf{J}^* \quad {}^0\mathbf{J}_W \end{aligned} \tag{2.22}$$

$\mathbf{J}^*$ is the transformation matrix that converts the operational point from the wrist point to any arbitrary point rigidly attached to the end-effector.

This decoupling technique is widely used in the analysis of Jacobian matrix. The usage is shown in Chapter 3 and 4 of this dissertation.

2.4 Redundancy

2.4.1 Redundancy Definition

If a manipulator has more degrees of motion than required by the task, then the manipulator is redundant. From Equation 2.17, it can be seen that the Jacobian matrix would have more columns than rows (n > m). In this case, there exist infinite sets of solution to the inverse kinematics.

Figure 2.4 is taken from [9] to illustrate the Redundant space in the Jacobian matrix.

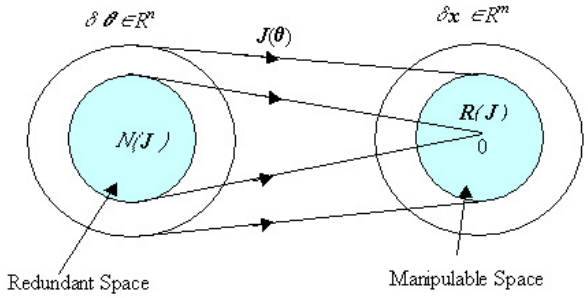


Figure 2.4: The relationship between the manipulable and redundant space

If the Jacobian matrix is defined as of size m by n, then the rank of the matrix is the degree of manipulability (DoM) of the manipulator, and the dimension of the redundant space is the degree of redundancy (DoR) [17]. It also follows that:

$$DoM + DoR = n. \tag{2.23}$$

At singular configurations, rank of the Jacobian matrix degenerates, therefore DoM would decrease, while DoR increases by the same number that DoM decreases (by 2.23).

2.4.2 The Jacobian Matrix

Most robot control algorithms use the Jacobian matrix as a coefficient to the linear equations. Therefore, kinematic properties can be analyzed on the basis of the Jacobian matrix in light of matrix theory [18].

The definition of the Jacobian matrix is given in Equation 2.17. It shows the mapping between task space and joint space velocities. The Range space of the Jacobian matrix, $\mathcal{R}(J)$, is the subspace made up of kinematically realizable motions in task space. The dimension of the Range space is the rank of $\mathbf{J}(\mathbf{q}) = min(m, n)$, where $\mathbf{J}$ is $m \times n$.

The orthogonal complement of the Range space is the Null space, made up of all the kinematically unrealizable motion of $\delta\mathbf{x}$.

2.4.3 Redundancy Resolution

Null Space control uses the fact that the orthogonal projection of the Jacobian to the null space $(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})$ is a subspace of $\dot{\mathbf{q}}$ that produces zero motion at the end effector. This is used in the the control of null space motion.

There is also a null space associated with the transpose of Jacobian matrix. The relationship between end-effector forces and joint torques is expressed as:

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{f} + [\mathbf{I} - \mathbf{J}^T(\mathbf{q})\mathbf{J}^{T\#}(\mathbf{q})]\boldsymbol{\tau}_0 \tag{2.24}$$

where $\mathbf{J}^{T\#}$ is the generalised inverse of $\mathbf{J}^T$, and $\boldsymbol{\tau}_0$ is the arbitrary joint torque vector to be projected to the null space of $\mathbf{J}^T$. Throughout this dissertation, the *dynamically consistent inverse* $(\bar{\mathbf{J}})$ [4] is used as the inverse that's consistent with the system's dynamics (Section 2.5 covers the dynamically consistent inverse). The summary table for force/position duality is shown in Table 2.1.

Table 2.1: The position/force duality

Position	Force
$\delta\mathbf{q} = \bar{\mathbf{J}}\delta\mathbf{x} + [I - \bar{J}J]\delta\mathbf{q}_0$	$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{f} + [I - \bar{J}J]^T\boldsymbol{\tau}_0$

Equation 2.24 shows that joint torques can be decomposed into those corresponding to force acting on the end-effector (the operational forces, represented by the $\mathbf{J}^T\mathbf{f}$ part of $\boldsymbol{\tau}$), and those that affect only the internal joint motion ($[\mathbf{I} - \mathbf{J}^T(\mathbf{q})\mathbf{J}^{T\#}(\mathbf{q})]\boldsymbol{\tau}_0$).

In resolving redundancies, a performance measure is usually defined so that the set of solution selected would minimise or maximise this measure. The maximum number of independent sets of performance measure that can be imposed is equal to the degrees of redundancy of the manipulator with respect to the task. Performance measure can be designed to avoid singularities [19, 20], to avoid obstacles [21, 17], to avoid joint limits [22], or to ensure that the torque is within a certain limit [19].

Let $v(\mathbf{q})$ be the performance measure as a function of joint displacement. One way to resolve redundancy, is to define a gradient descent function of the performance measure $\boldsymbol{\tau}_0 = -\nabla v(\mathbf{q})$. This is then projected into the null space by Equation 2.24. This is the method used in this dissertation.

An alternative way of obtaining the effect of null space projection is by using the extended Jacobian matrix [23, 24, 25].

The conventional method expresses the range and null space projection of the task explicitly as shown in Equation 2.25.

$$\begin{array}{rcccccc} \boldsymbol{\tau} & = & \mathbf{J}^T & \mathbf{f} & +[\mathbf{I} - \bar{\mathbf{J}}\mathbf{J}]^T & \boldsymbol{\tau}_0 \\ \delta\mathbf{q} & = & \underbrace{\mathbf{J}^{\#}}_{Manipulable\ Space} & \delta\mathbf{x} & +\underbrace{[\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}]}_{Null\ Space} & \delta\mathbf{q}_0 \end{array} \tag{2.25}$$

The idea in the Extended Jacobian Matrix (EJM) is to incorporate the desired null space behaviour into the Jacobian matrix. By doing this, the method extends the redundant Jacobian matrix into a square matrix. When the inverse kinematics is performed, this square Jacobian matrix will yield a unique solution that satisfies both the required range and null space behaviours.

The extended Jacobian matrix is defined as

$$\mathbf{J} = [\mathbf{J}_m | \mathbf{J}_{n-m}] \tag{2.26}$$

where $\mathbf{J}_m$ is the $m \times m$ square minor of the Jacobian matrix that is full rank (when the Jacobian matrix is not singular) and $\mathbf{J}_{n-m}$ is of size $m \times (n-m)$. $\mathbf{J}$ can be rewritten as:

$$\mathbf{J} = \mathbf{J}_m [\mathbf{I}_m | \mathbf{J}'_{n-m}] \tag{2.27}$$

where $\mathbf{J}'_{n-m} = \mathbf{J}_m^{-1}.\mathbf{J}_{n-m}$ and $\mathbf{I}_m$ is an identity matrix of size $m \times m$.

A matrix $\boldsymbol{\Upsilon}$ is defined so that

$$\boldsymbol{\Upsilon} = \begin{pmatrix} \mathbf{J}'_{n-m} \\ -\mathbf{I}_{n-m} \end{pmatrix} \tag{2.28}$$

The matrix $\boldsymbol{\Upsilon}$ an orthogonal matrix to the Jacobian matrix $\mathbf{J}$. This is shown by:

$$\mathbf{J}.\boldsymbol{\Upsilon} = (\mathbf{J}_m | \mathbf{J}_{n-m}) . \begin{pmatrix} \mathbf{J}_m^{-1}.\mathbf{J}_{n-m} \\ -\mathbf{I}_{n-m} \end{pmatrix} = \mathbf{0} \tag{2.29}$$

A method of utilising the matrix $\boldsymbol{\Upsilon}$ is shown by the Extended Jacobian Matrix (EJM) [23, 24, 26, 27, 28].

The EJM is a method of resolving redundancy by adding (extending) the Jacobian matrix $\mathbf{J}$ with additional number of rows $\mathbf{J}_a$ so that the extended Jacobian matrix

$\mathbf{J}_e$ is square. The additional rows $\mathbf{J}_a$ contains the extra constraints that need to be satisfied for the redundancy resolution to arrive at a particular set of solution. These extra rows are designed so that $\mathbf{J}_a.\dot{\mathbf{q}} = \mathbf{0}$ so that the internal joint motion will not affect the motion of the end-effector.

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \\ --- \\ \mathbf{0} \end{pmatrix} &= \mathbf{J}_e(\mathbf{q})\dot{\mathbf{q}} \\ &= \begin{pmatrix} \mathbf{J} \\ --- \\ \mathbf{J}_a \end{pmatrix} \dot{\mathbf{q}} \end{aligned} \tag{2.30}$$

where $\mathbf{J}_e$ is the extended or padded square Jacobian matrix, $\mathbf{J}$ is the original non-square Jacobian for the redundant manipulator, and $\mathbf{J}_a$ is the additional constraints in the form of extra rows to patch the Jacobian matrix to a square shape.

Let the performance measure to be optimised again be $v_0(\mathbf{q})$. Then a matrix $\mathbf{G}(\mathbf{q})$ can be defined as:

$$\mathbf{G}(\mathbf{q}) = \left(\frac{\delta v_0(\mathbf{q})}{\delta \mathbf{q}} . \boldsymbol{\Upsilon} \right)^T \tag{2.31}$$

The non-square Jacobian matrix can now be padded to a square matrix by:

$$\mathbf{J}_e = \begin{pmatrix} \mathbf{J} \\ \frac{\delta \mathbf{G}(\mathbf{q})}{\delta \mathbf{q}} \end{pmatrix} \tag{2.32}$$

This, in effect, projects the performance measure into the null space of the Jacobian matrix, so that:

$$\begin{pmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \\ --- \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{J} \\ --- \\ \frac{\delta \mathbf{G}(\mathbf{q})}{\delta \mathbf{q}} \end{pmatrix} \dot{\mathbf{q}} \tag{2.33}$$

When the inverse kinematics is performed, the (square) extended Jacobian $\mathbf{J}_e$ will yield a set of solution $\dot{\mathbf{q}}$ that satisfies both the required task space velocity $\dot{\mathbf{x}}$ and the null space behaviour as dictated by the potential function $v_0(\mathbf{q})$.

2.4.4 When Null Space Projection Conflicts with End-Effector Motion

Sometimes, the desired null space behaviour is in conflict with the required end-effector motion. By definition, any function projected into the null space of the Jacobian matrix does not produce any effect at the end-effector. In other words, the redundant space is mapped onto a zero point in the manipulable space (Section 2.4.1). Therefore, when there is a conflict between the desired null space behaviour and the required end-effector motion, the null space behaviour would not be satisfied.

This is also reflected mathematically on the equations. Equation 2.34 shows the *manipulable* and *null space* of a manipulator. The null space torque ($\boldsymbol{\tau}_0$) and the null space joint velocity ($\delta\mathbf{q}_0$) are as explained in Section 2.4.3.

$$\begin{array}{rcccccc} \boldsymbol{\tau} & = & \mathbf{J}^T & \mathbf{f} & +[\mathbf{I}-\bar{\mathbf{J}}\mathbf{J}]^T & \boldsymbol{\tau}_0 \\ \delta\mathbf{q} & = & \underbrace{\mathbf{J}^{\#}}_{Manipulable\ Space} & \delta\mathbf{x} & +\underbrace{[\mathbf{I}-\mathbf{J}^{\#}\mathbf{J}]}_{Null\ Space} & \delta\mathbf{q}_0 \end{array} \tag{2.34}$$

An example of three-link planar (R-R-R) manipulator is shown in Figure 2.5. The desired end-effector motion is to move along the X_0 axis away from the origin. The manipulator is set to maintain a desired null space behaviour of $q_2 = -30^0$. The null space torque or velocity vector in this case is a 3×1 vector:

$$-\nabla v_0(\mathbf{q}) = \begin{pmatrix} 0 \\ \frac{\delta}{\delta q} v_0(\mathbf{q}) \\ 0 \end{pmatrix} \tag{2.35}$$

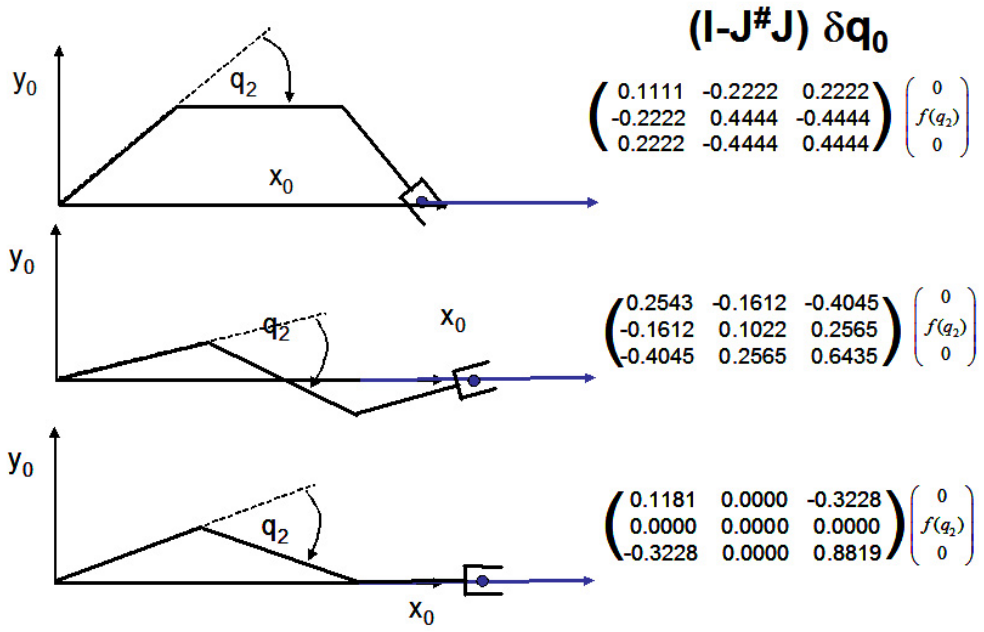


Figure 2.5: The three-link planar *(R-R-R)* manipulator above is given a task of following a trajectory in 2D. The extra degree of freedom is assigned to the null space behaviour of keeping $q_2 = -30^0$. The desired end-effector motion is to move along the X_0 axis away from the origin. The top and middle pictures show the possible configurations where both the desired end-effector motion and required null space behaviour are satisfied. The bottom picture shows the configuration where the null space behaviour can no longer be satisfied. Notice the $(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})$ projection matrices for all the three cases.

The manipulator is shown in the top and middle pictures (2.5) to be in configurations where it is possible to satisfy both the desired null space behaviour and the required end-effector motion. A numerical example is substituted into the equation. Notice the matrix $(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})$ has non-zero values in the rows and columns corresponding to the the null space behaviour of joint 2.

In the bottom picture, however, the manipulator is shown to be in a configuration where in order to satisfy the required end-effector motion, the desired null space behaviour has to be abandoned. This shows in the $(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})$ matrix where the second row and column now contain only zeros. This means any function in the second element of vector $-\nabla v_0(\mathbf{q})$ will not have any effect on the motion of the end-effector. This confirms that null space behaviour does not produce any additional effect on the end-effector motion and it will not be satisfied in the case where it is in conflict with the required end-effector motion.

2.5 Generalised Inverses

As shown by the previous sections, a generalised inverse is required to invert a non-square matrix for redundant manipulators. A well-known generalised inverse is the Moore-Penrose generalised inverse. Others can be found in [29, 30, 31, 19] among many.

The matrix $\mathbf{J}^{\#}$ is the generalised inverse of $\mathbf{J}$ if and only if [32]:

$$\mathbf{J}\mathbf{J}^{\#}\mathbf{J} = \mathbf{J} \tag{2.36}$$

Each generalised inverse has a property of minimising a certain property in selecting one solution out of the infinite number of sets. For example, pseudo inverse is one of the solutions, whereby it minimizes the norm of the vector / solution in a least-squared fashion.

2.5.1 Dynamically Consistent Inverse

The natural criterion in designing the motion of a robotic mechanism is to have a solution with minimum kinetic energy. It is the natural way because it conforms with the dynamics of the mechanism and is physically intuitive with how human and animals utilise their bodies.

While projection of an arbitrary vector into the null space of the Jacobian matrix will not affect the task of the end-effector [9, 32, 33], only one inverse would guarantee that it is consistent with the dynamics and it is where the kinetic energy is minimised.

This is the dynamically consistent inverse as shown in [34, 4], and [35]. It is found by first equating the torque from the dynamic equation of a manipulator (Equation 2.4) with the torque from the generalised-forces equation (Equation 2.1):

$$\hat{\mathbf{A}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q}) = \mathbf{J}^T\mathbf{f} + [\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}]^T\boldsymbol{\tau} \tag{2.37}$$

and then premultiplying the equation by $\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})$ to yield:

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})(\hat{\mathbf{b}}(\mathbf{q}, \dot{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q})) = \mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})\mathbf{J}^T\mathbf{f} + \mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})[\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}]^T\boldsymbol{\tau} \tag{2.38}$$

As the aim of the inverse is not to introduce additional acceleration at the end-effector, $\mathbf{J}\ddot{\mathbf{q}}$ is substituted by the derivative of $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$:

$$\begin{aligned} \frac{d}{dt}\dot{\mathbf{x}} &= \dot{J}\dot{\mathbf{q}} + J\ddot{\mathbf{q}} \\ \ddot{\mathbf{x}} &= \dot{J}\dot{\mathbf{q}} + J\ddot{\mathbf{q}} \\ \mathbf{J}\ddot{\mathbf{q}} &= \ddot{\mathbf{x}} - \dot{J}\dot{\mathbf{q}} \end{aligned} \tag{2.39}$$

The resulting equation is written as:

$$\begin{aligned}\ddot{\mathbf{x}} + [\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})\mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}] + \mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})\mathbf{g}(\mathbf{q}) = \\ (\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})\mathbf{J}^{T}(\mathbf{q})\mathbf{f} + \mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})[\mathbf{I} - \mathbf{J}^{T}(\mathbf{q})\mathbf{J}^{\#T}(\mathbf{q})]\boldsymbol{\tau}_0\end{aligned} \tag{2.40}$$

Equation 2.40 shows that for the acceleration at the operational point not to be affected by $\boldsymbol{\tau}_0$, it is necessary for the term involving $\boldsymbol{\tau}_0$ to be zero, or:

$$\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})[\mathbf{I} - \mathbf{J}^{T}(\mathbf{q})\mathbf{J}^{\#T}(\mathbf{q})]\boldsymbol{\tau}_0 = 0 \tag{2.41}$$

A generalised inverse of $\mathbf{J}(\mathbf{q})$ satisfying the above constraint is said to be *dynamically consistent* [36]. This inverse is the generalised inverse of the Jacobian matrix corresponding to the solution of $\delta\mathbf{x} = \mathbf{J}(\mathbf{q})\delta\mathbf{q}$ that minimises the manipulator's instantaneous kinetic energy. $\mathbf{J}^{\#T}$ can then be solved from 2.41 as:

$$\mathbf{J}\mathbf{A}^{-1} - \mathbf{J}\mathbf{A}^{-1}\mathbf{J}^{T}\mathbf{J}^{\#T} = 0 \tag{2.42}$$

Therefore

$$\mathbf{J}^{\#T} = (\mathbf{J}\mathbf{A}^{-1}\mathbf{J}^{T})^{-1}\mathbf{J}\mathbf{A}^{-1} \tag{2.43}$$

or

$$\bar{\mathbf{J}}^{T} = \boldsymbol{\Lambda}(\mathbf{q})\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q}) \tag{2.44}$$

where

$$\boldsymbol{\Lambda}(\mathbf{q}) = (\mathbf{J}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q})\mathbf{J}^{T}(\mathbf{q}))^{-1} \tag{2.45}$$

2.6 Measure of Orientation Error

In this dissertation, orientation error of the end-effector is measured in $d\boldsymbol{\phi}$, which is defined as [7]:

$$d\boldsymbol{\phi} = -\frac{1}{2}(\hat{\mathbf{s}}_1\mathbf{s}_{1d} + \hat{\mathbf{s}}_2\mathbf{s}_{2d} + \hat{\mathbf{s}}_2\mathbf{s}_{2d}) \tag{2.46}$$

where $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ are the columns which represent the orientation of the end-effector with respect to Base Frame, and $\mathbf{s}_{1d}$, $\mathbf{s}_{2d}$, $\mathbf{s}_{3d}$ represent the desired orientation, at each sampling period.

The "$\hat{}$" symbol is the skew symmetric matrix equivalent to the cross product of the vector. For example, if

$$\mathbf{s} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \tag{2.47}$$

then

$$\hat{\mathbf{s}} = \begin{bmatrix} 0 & -s_z & s_y \\ s_z & 0 & -s_x \\ -s_y & s_x & 0 \end{bmatrix} \tag{2.48}$$

CHAPTER 3

BACKGROUND THEORY II: SINGULARITIES

3.1 Types of Singularities

In [37], singularities are divided into real and artificial singularities. Real singularities are those inherent in the physical configuration of the robot or the environment. Artificial (or sometimes called mathematical) singularities are those resulting from the control algorithms of the manipulator.

Real singularities include *kinematic* singularities and *semi* singularities. Artificial singularities include *algorithmic* and *semi-algorithmic* singularities.

Algorithmic, semi, and semi-algorithmic singularities can occur only to redundant manipulators [38]. In a redundant system, self-motion is possible as there are more number of joints than necessary for the task. A performance measure optimisation is usually imposed to regulate the behaviour of the self motion to arrive to a particular solution.

3.1.1 Real singularity

Kinematic singularities are those where the manipulator loses the ability to execute tasks in full set of DoFs. This means, there is no solution in the inverse kinematics that would produce the required motion in task space. Semi singularities are the

result of kinematic inequality such as obstacles and joint angle limits. These are all inherent part of the robot's physical configuration.

3.1.2 Artificial singularity

Algorithmic singularities occur when the main task of controlling the end-effector contradicts the secondary task of optimising a performance measure.

The semi-algorithmic singularity is the combined version of semi and algorithmic singularities. It occurs as the result of kinematic inequality such as joint limits and obstacles (semi singularities) and at the same time the main and secondary tasks of the control algorithm contradict each other.

3.2 Kinematic Singularity

Singularity is generally defined as the configurations where the manipulator does not posses its full DoFs or when the Jacobian matrix is rank deficient. This means a drop in the degree of manipulability (and an increase in degree of redundancy, as in Equation 2.25). At these configurations, the manipulator loses the ability to perform tasks in the maximum number of DoF that it normally possesses in non-singular configurations.

It is also reflected by the determinant of the Jacobian matrix going to zero ($det(\mathbf{J}) = 0$) in the case of square matrices or $det(\mathbf{J}\mathbf{J}^T) = 0$ in the case of a redundant manipulator [39].

3.2.1 A Two-link Example of Singularity

To briefly illustrate a kinematically singular configuration, a two-link manipulator example as shown in Figure 3.1 is presented here. While in singular configuration

(Figure 3.1 (right)), any joint command sent to joint 1 or 2 will not produce any instantaneous motion in the $\mathbf{x}_2$ direction. In this case, $\mathbf{x}_2$ represents the degenerate or singular direction.

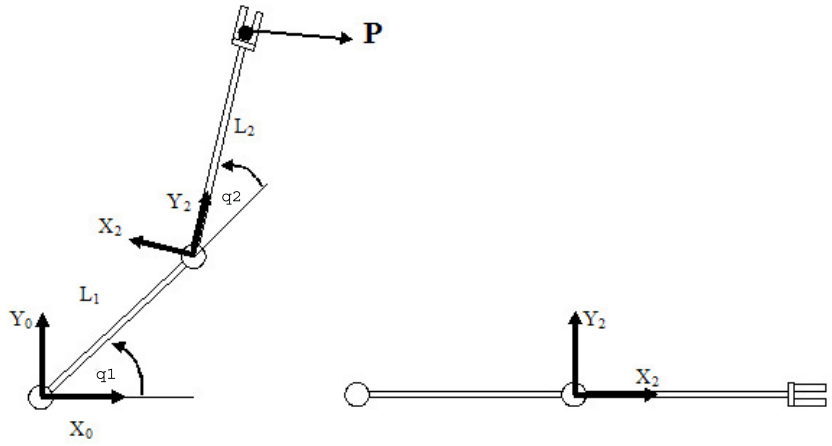


Figure 3.1: Example of singular configuration: a two-link planar manipulator has 2 degrees of freedom (left). At singular configuration (right), any joint command sent to joint 1 and/or joint 2 will not produce any instantaneous motion in X_2 direction.

If the length of the two links were l_1 and l_2 and angular displacements of the two joints were q_1 and q_2 as shown in Figure 3.1 (left), then the velocity of point P of the manipulator *with respect to Frame* $\{2\}$ can be described as:

$$\begin{array}{ccc} \delta\mathbf{x} & = & \mathbf{J} & \delta\mathbf{q} \\ \begin{pmatrix} \delta x_2 \\ \delta y_2 \end{pmatrix} & = & \begin{pmatrix} l_1 s_2 & 0 \\ l_2 + l_1 c_2 & a_2 \end{pmatrix} & \begin{pmatrix} \delta q_1 \\ \delta q_2 \end{pmatrix} \end{array} \tag{3.1}$$

At the kinematically singular configurations, as described in Figure 3.1(right), $q_2 = 0$ (therefore $sin(q_2) = 0$ and $cos(q_2) = 1$, and the Jacobian matrix becomes:

$$\begin{array}{ccc} \delta\mathbf{x} & = & \mathbf{J} & \delta\mathbf{q} \\ \begin{pmatrix} \delta x_2 \\ \delta y_2 \end{pmatrix} & = & \begin{pmatrix} 0 & 0 \\ l_2 + l_1 & a_2 \end{pmatrix} & \begin{pmatrix} \delta q_1 \\ \delta q_2 \end{pmatrix} \end{array} \tag{3.2}$$

which shows that for any value of δq_1 and δq_2, there will be no resulting instantaneous change in the $\mathbf{x}_2$ direction. Thus, the manipulator is in singular configuration, as instantaneously it can only move in $\mathbf{y}_2$ direction. The direction along $\mathbf{x}_2$ in this case represents the singular direction.

However, it is not always correct to say that a manipulator is singular when it loses one or more of its degrees of freedom. Rather, it is when it does not have the full DoFs or when the Jacobian matrix is rank-deficient. The reason will be explained in Chapter 4 (Section 4.5).

3.2.2 Singular Value Decomposition

Singular Value Decomposition of the Jacobian matrix are often used in computation to determine the inverse of the matrix and to analyse singularity issues of the manipulator. In Chapter 4, the SVD of the Jacobian matrix is utilised to determine the singular direction(s) of the manipulator. These identified singular directions are removed in the singularity handling methods in Chapter 5 and 6 to allow the end-effector to operate in the near singular region in a stable manner.

The SVD of the Jacobian matrix $\mathbf{J}$ is defined as:

$$\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3.3}$$

where $\mathbf{U}\epsilon R^{m\times m}$ and $\mathbf{V}\epsilon R^{n\times n}$ are proper orthogonal matrices and $\mathbf{\Sigma}$ is a diagonal matrix with values $(\sigma_1, \sigma_2, ..., \sigma_m)$ and $\mathbf{\Sigma}\epsilon R^{m\times n}$ [17], where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0$ are the singular values of the matrix.

The inverse of the Jacobian matrix can hence be expressed as:

$$\mathbf{J}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T \tag{3.4}$$

Since

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \tag{3.5}$$

then

$$\dot{\mathbf{x}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\dot{\mathbf{q}} \tag{3.6}$$

or

$$\mathbf{U}^T\dot{\mathbf{x}} = \mathbf{\Sigma}\mathbf{V}^T\dot{\mathbf{q}} \tag{3.7}$$

The columns of the square and orthonormal matrix $\mathbf{U}$ contains the singular vectors of the Jacobian matrix and they describe the principal axes of motion in a task space, expressed in the same frame as that of the Jacobian matrix. Since it is orthonormal, it can be utilised as a rotational matrix that transforms the Jacobian matrix from it's current frame to a *singular Frame* S, as shown in Equation 3.8.

$$^S\mathbf{J} = \mathbf{U}^T\dot{\mathbf{x}} = \mathbf{\Sigma}\mathbf{V}^T\dot{\mathbf{q}} \tag{3.8}$$

The $\mathbf{\Sigma}\mathbf{V}^T$ can therefore be thought of as the Jacobian matrix expressed in the *Singular Frame*{S}. It relates the joint space velocity to the task space velocity expressed in the *Singular Frame* {S}.

When the manipulator is in singular configuration, one or more of the singular values (σ) will be zero. Equation 3.9 shows the example of the matrix $\mathbf{\Sigma}$ for a 3×4 (redundant) Jacobian matrix, where $\sigma_1 \geq \sigma_2 \geq \sigma_3$. In singular configuration, the last k singular values (σ) would be zero, where k is the number of rank that the Jacobian matrix loses.

$$
\begin{aligned}
{}^{S}\mathbf{J} &= \mathbf{\Sigma}\mathbf{V}^{T}\dot{\mathbf{q}} \\
&= \begin{pmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \end{pmatrix} \mathbf{V}^{T}\dot{\mathbf{q}}
\end{aligned} \tag{3.9}
$$

When $\mathbf{\Sigma}$ is multiplied with $\mathbf{V}^T$, it results in a Jacobian matrix expressed in *Singular Frame* {S} that has only zeros in its last k rows. An example of a 3×4 Jacobian matrix is shown in Equation 3.10 for the case of $k = 1$ or when it possesses a rank of 2.

$$
\begin{aligned}
{}^{S}\mathbf{J} &= \mathbf{\Sigma}\mathbf{V}^{T}\dot{\mathbf{q}} \\
&= \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ 0 & 0 & 0 & 0 \end{pmatrix} \dot{\mathbf{q}}
\end{aligned} \tag{3.10}
$$

where $\{x\ x\ x\ x\}$ is a non-zero row in the matrix.

This way, the singular direction of the manipulator is revealed. However, this can only be done numerically, not symbolically.

The example of SVD application on the Jacobian matrix on a two link manipulator such as one in Figure 3.1 and on PUMA is given in Section 4.6.5 when the topic of identification of singular direction is discussed. Further reading on SVD can be found in [32].

3.3 Semi Singularity

Semi singularities are the result of kinematic inequality constraint such as obstacles and joint angle limits [37]. It is a real singularity, like kinematic singularity, as it is inherent in the physical configuration of the robot or the condition of the environment. Unlike kinematic singularity, where the manipulator loses the ability to move in a particular axis in both directions, in semi singular configurations, the

manipulator loses the ability to move in one direction but is still capable of moving in the opposite direction.

Suppose joint i is at joint limit. A minor of the Jacobian matrix $\mathbf{J}_m$ is defined by removing the column corresponding to joint i. If $\mathbf{J}_m$ is rank deficient, then the manipulator is at semi singularity [40] [41]. In short, it is a kinematic singularity that happens as a manipulator loses the use of one or more of its joints due to joint limits or obstacles.

3.4 Algorithmic Singularity

Algorithmic singularities occur when the main task of controlling the end-effector motion is interlocked with the secondary task of optimising a performance measure. When this happens, the manipulator is not degenerate physically (mechanically), but mathematically. Two examples are shown to illustrate the problem.

3.4.1 Example with Extended Jacobian Matrix method

Algorithmic singularity is a typical problem with the popular EJM technique (Extended Jacobian Matrix) The EJM method is explained in Section 2.4.3 [23, 24, 26, 27, 28].

In a redundant manipulator, the Jacobian matrix is not square but of size $m \times n$, where m is the number of DoFs in task space, while n is the number of joints. The kinematic control techniques that require the inverse of the Jacobian matrix is therefore faced with the problem of inverting a non-square matrix. Various generalised inverses have been used in the past to resolve the issue. EJM technique, as explained in Section 2.4.3 pads the Jacobian matrix into a square matrix so that:

$$\begin{aligned}\begin{pmatrix} \dot{\mathbf{p}} \\ \dot{\omega} \\ --- \\ \mathbf{0} \end{pmatrix} &= \mathbf{J}_e(\mathbf{q})\dot{\mathbf{q}} \\ &= \begin{pmatrix} \mathbf{J} \\ ------- \\ \mathbf{J}_a \end{pmatrix} \dot{\mathbf{q}}\end{aligned} \tag{3.11}$$

where $\mathbf{J}_e$ is the extended or padded square Jacobian matrix, $\mathbf{J}$ is the original non-square Jacobian for the redundant manipulator, and $\mathbf{J}_a$ is the additional constraints in the form of extra rows to patch the Jacobian matrix to a square shape.

The number of rows added is the same as the dimension of null space of the Jacobian matrix, which is $(n - m)$. The addition of rows into the Jacobian matrix to provide constraints in redundancy resolution introduces more conditions in which the Jacobian matrix could become degenerate.

When $\mathbf{J}$ is rank-deficient, the manipulator is in kinematic singularity. Algorithmic singularity is when $\mathbf{J}_a$ is rank deficient while $\mathbf{J}$ is not. This results in rank-deficient overall Jacobian matrix $\mathbf{J}_e$.

3.4.2 Example:Mobile Bases with Powered Caster Wheel

In mobile bases with powered caster wheels (Figure 3.2), a Jacobian matrix can be defined by finding the relationship between the contact point of the wheel with the floor and the center of the mobile platform. The study on designs and development of the kinematics and dynamics of mobile bases has been conducted quite extensively in the past. Specifically for powered-cater wheels, the literature can be found in [42, 43, 44, 45, 46, 47], among many others.

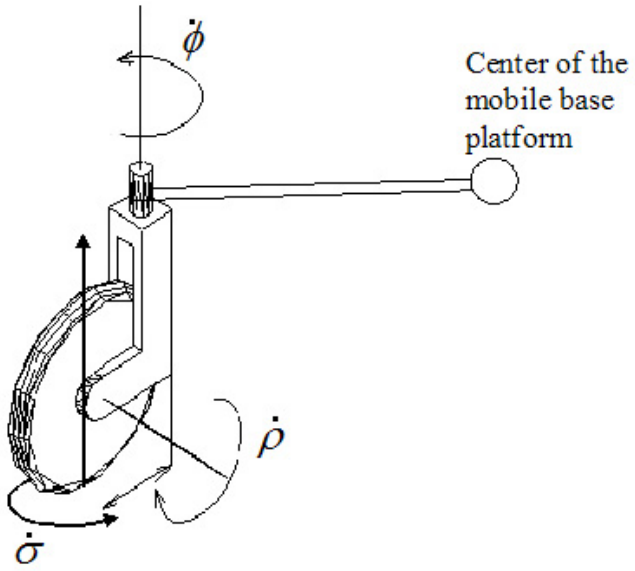


Figure 3.2: The powered caster wheel - with three angular velocities, two of which are measurable and actuated with motors, while $\dot{\sigma}$ is not measurable nor actuated.

The Jacobian can then be expressed relating the joint rate commands $[\dot{\phi}_1, \dot{\rho}_1, \dot{\sigma}_1, ...\dot{\phi}_N, \dot{\rho}_N, \dot{\sigma}_N]^T$ to task space velocity vector of the center point of the mobile platform $[\dot{x}, \dot{y}, \dot{\theta}]^T$. N is the number of wheel modules the base has.

Let a mobile platform have N wheel modules, separated evenly from each other by angle $\Delta\beta = 2\pi/N$. Let γ be the angle between the first wheel module and the X axis of the mobile base frame (the frame attached to the center of operational point of the mobile platform that moves and rotates with the mobile base) (Figure 3.3). Let β be the angle each wheel module makes with wheel module number 1, therefore $\beta_i = \Delta\beta(i-1) + \gamma$. The joint actuators are defined as: $\dot{\phi}$ is the angular velocity between the wheel and the body of the mobile base, $\dot{\rho}$ is the angular velocity of the driving wheel, and $\dot{\sigma}$ is the angular velocity between the absolute frame of the floor

with the wheel (where the rotation is around an axis normal to the floor intersecting the contact point of the wheel and the floor). (see Figure 3.3).

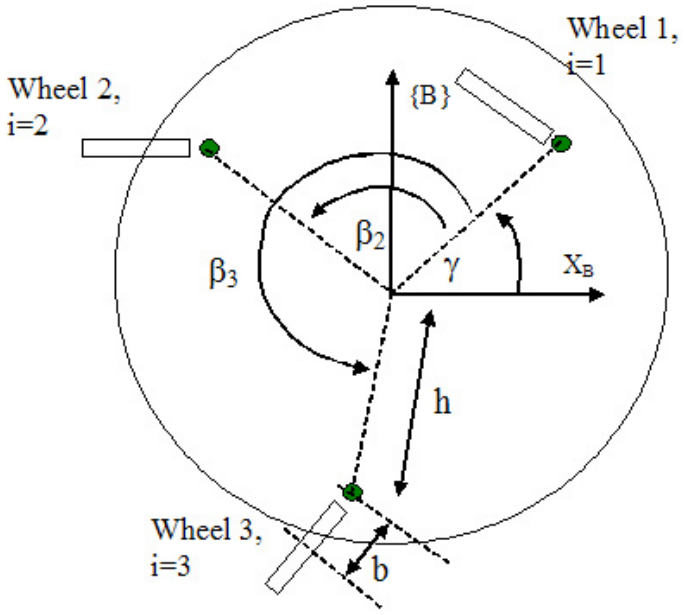


Figure 3.3: The variables used to describe the configuration of the mobile base.

The inverse kinematics of the individual wheel module is:

$$\begin{pmatrix} \dot{\phi}_i \\ \dot{\rho}_i \\ \dot{\sigma}_i \end{pmatrix} = \frac{1}{rb} \begin{pmatrix} rs_{\phi_i} & -rc_{\phi_i} & r(b - hc_{\alpha_i}) \\ bc_{\phi_i} & bs_{\phi_i} & -bhc_{\alpha_i} \\ -rs_{\phi_i} & rc_{\phi_i} & rhc_{\alpha_i} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{3.12}$$

where $\alpha = \beta_i - \phi_i$, i denotes the wheel module of interest, r is the radius of the wheels, h is the radius of the platform and b is the offset on the caster wheels.

Equation 3.12 shows the inverse kinematics that has been obtained symbolically. It can be said that as long as the radius of the wheel r and the offset b is not zero, the mapping from task space to joint rate vector always exists. This implies that for

any desired task space velocity of the operational point attached to the mobile base, there always exists a set of joint rate command to provide the motion.

Joint rate command (the inverse kinematic solution) can be obtained directly by:

$$\begin{pmatrix} \dot{\phi}_1 \\ \dot{\rho}_1 \\ \dot{\sigma}_1 \\ \vdots \\ \dot{\phi}_N \\ \dot{\rho}_N \\ \dot{\sigma}_N \end{pmatrix} = \begin{pmatrix} \mathbf{J}_1^{-1} \\ \\ \vdots \\ \\ \mathbf{J}_N^{-1} \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \tag{3.13}$$

where $\mathbf{J}_i^{-1}$ is the inverse Jacobian matrix for the individual wheel module such as shown in Equation 3.12.

However, in some control methods, such as in [44], a set of joint rate variables can be defined as active and the rest as passive. This is useful when not all the joints in the mobile platform are powered (actuated), for example, when only three joints are powered, while the rest are passive caster wheels. Supposed the active variables are chosen as $\theta_a = [\phi_1, \rho_2, \rho_3]^T$. Through rearranging of the independent equations between joint rate command and task space command, we can define two Jacobian matrices such that

$$\mathbf{J}_a \dot{\boldsymbol{\theta}}_a = \mathbf{J}_p \dot{\boldsymbol{\theta}}_p \tag{3.14}$$

The active joint rate command θ_a can be obtained from selecting the correct rows of the inverse kinematics equations, while the passive wheel joint rate command from:

$$\boldsymbol{\theta}_p = \mathbf{J}_p^{-1} \mathbf{J}_a \dot{\boldsymbol{\theta}}_a \tag{3.15}$$

Obtaining the joint rate command this way, however, introduces an extra step which involves taking the inverse of $\mathbf{J}_p$. It is possible to have configurations where $\mathbf{J}_p$ is rank-deficient. These configurations represents the singularities that occurs when only the joints in $\boldsymbol{\theta}_a$ are actuated in the mobile platform. When this happens, it is termed *algorithmic singularity* in [44]. However, when all the joints are actuated, the mobile platform is capable of moving in all directions at all time.

The resulting singular configuration is dependent on the choice of $\boldsymbol{\theta}_a$. A different set of active joints would result in a different singular configurations. The singular configurations for this choice of active joint actuators ($\boldsymbol{\theta}_a = [\phi_1, \rho_2, \rho_3]^T$, namely the steering angle of wheel 1, driving angles of wheels 2 and 3) are shown in Figure 3.4.

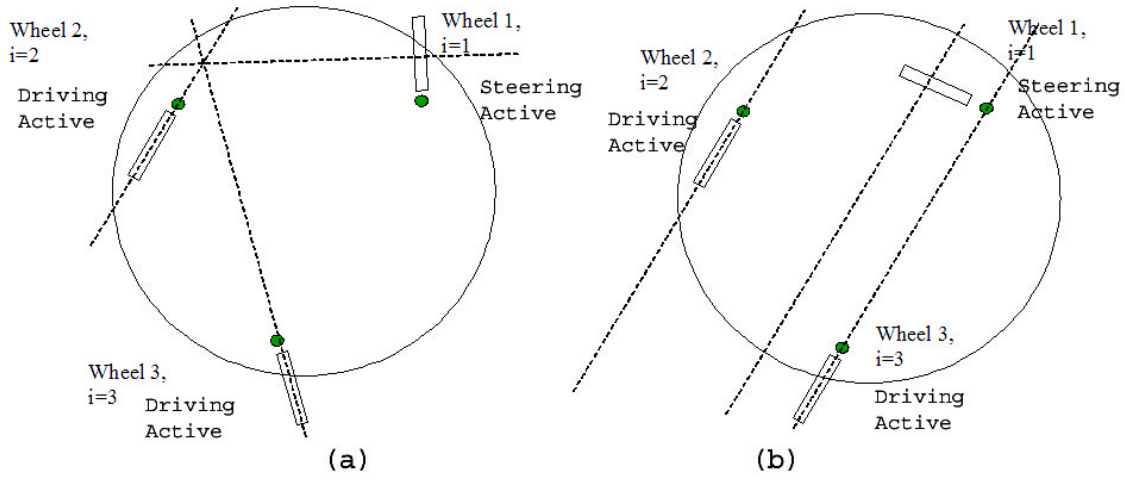


Figure 3.4: The singular configurations of a three wheeled mobile base, assuming active joint commands are: ϕ_1 (steering angle of wheel 1), ρ_2 and ρ_3 (driving angles of wheels 2 and 3). In (a) the mobile platform cannot rotate in around its Z axis and (b) it cannot translate in the direction perpendicular to the parallel lines.

This algorithmic singularity occurs because by first mapping the task space command into three active variables, it was assumed that the mobile platform only has those three actuated joints, instead of six. If instead of three, four active joint variables were made active, then the singularities would be removed. The significance of this analysis is that it is possible to design a three-wheeled mobile platform (with powered-caster wheels, each capable of steer and drive actuation) and power only four of the six joints. Theoretically, it should be able to move in any direction instantaneously.

3.5 Semi-Algorithmic Singularity

Semi-algorithmic singularities occur when a semi singularity and an algorithmic singularity happens at the same time. This can be distinguished from algorithmic singularity in that it is unidirectional, whether upon maximisation or minimisation of the performance criteria.

This class of singularities is quite new, in comparison to the rest, as reported in [37].

3.6 Summary

Types of singularities are summarised in this chapter. While mathematical or artificial singularities can be handled or removed efficiently, it is important to note that real singularities, especially kinematic singularities can not be removed, as it is inherent in the design of the mechanisms. Certain approximation or clever mathematical manipulation may be devised to handle the configuration in an efficient and

stable manner, however, certain sacrifice such as in tracking performance is still necessary. The only way to completely remove kinematic singularities is by designing a singularity free manipulators.

The subsequent chapters in this dissertation will explain the issues in the identifications of singular configurations and singular directions with their proposed solutions. Once identified, the singular configurations are to be handled so that the manipulator will be able to move through them in a robust and stable manner. The last chapter will present an industrial application where mobile manipulator is shown at work with singularity robust algorithm.

CHAPTER 4

NEW INSIGHTS INTO THE IDENTIFICATION OF KINEMATIC SINGULARITIES AND ITS DEGENERATE DIRECTIONS

4.1 Chapter Overview

This chapter discusses the methods of identifying singular configurations and singular directions of a serial manipulator. First it covers a technique proposed in [48], and evaluates the completeness of the solution given. A simple check is proposed to make sure that some singular configurations, relating to the wrist of the manipulators, are not left out. This chapter then covers the singularity identification methods for non-redundant and redundant manipulators, with examples given. A discussion on the identification of singular direction is given at the end.

The new contribution in this chapter is the insight into the completeness of the solution in the identification of singularities, obtained by the currently popular method. The method simplifies the identification process by dividing the Jacobian into the *translational* and the *orientational* halves, which was found to leave out a small subset of possible solutions. Another contribution is the analysis of identification of singular directions.

4.2 Introduction and Background

As explained in Chapter 2 in the section on singularity, singular configuration can be determined mathematically as the configuration where $det(\mathbf{J}) = 0$ for square Jacobian matrix and $det(\mathbf{J}.\mathbf{J}^T) = 0$ for redundant manipulator where the $\mathbf{J}$ matrix is not square. This property can be used to determine symbolically the condition of singular configuration in a manipulator.

Khatib [3] categorizes singularities into two main groups: type 1 and type 2 based of the types of motion generated by null space motion while the robot is in singular configuration. Type 1 is when null space torque moves the robot out of the singularity. Type 2 is when null space torques affects only the internal joint motions, or when the null space motion moves (shifts) the degenerate direction.

Section 4.3 will cover the singularity identification in a 6DoF manipulator with the PUMA 560 as an example. Section 4.4 covers the popular methods of identifying the singularities in a redundant manipulator. To simplify the process, the Jacobian is often divided into the top and bottom halves as the translation and orientation portions. These halves are then evaluated separately to yield the position and orientation singularities respectively. This method has its drawback, as presented in Section 4.5, in that it may miss certain singularities resulting from the dependencies between the top and bottom halves of the Jacobian matrix. This is the new contribution to the identification process of singularities. Section 4.6 covers the issues in the identification of singular directions once the singularities are identified. Section 4.7 offers a simple check to the problem of incomplete solutions covered in Section 4.5.

4.3 Singularity Identification for a 6 DoF Manipulator

As a 6 DoF manipulator has a square Jacobian matrix, it is very simple to identify the singularities. Setting $det(\mathbf{J}) = 0$ will yield the complete solution. If this turns out to be computationally expensive, a simplification can be done such as in Section 2.3 for those manipulators with spherical wrists. The Jacobian can be expressed as:

$$\begin{aligned} \mathbf{J}_T &= \mathbf{J}^* \quad \mathbf{J}_W \\ \mathbf{J}_T &= \mathbf{J}^* \begin{pmatrix} \mathbf{J}_{11} & \mathbf{0}_{3\times3} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix} \end{aligned} \tag{4.1}$$

$\mathbf{J}$ is a transformation matrix whose expression is given in Equation 2.20. It can be established that $det(\mathbf{J}^*) = 1$ at all time. Therefore, $det(\mathbf{J}_T) = det(\mathbf{J}_W)$. Because the Jacobian is a square matrix with a zero top right quadrant, then $det(\mathbf{J}_T) = det(\mathbf{J}_W) = det(\mathbf{J}_{11}).det(\mathbf{J}_{22})$. This reduces the identification problem to solving the determinants of 3×3 matrices. $\mathbf{J}_{11}$ and $\mathbf{J}_{22}$ determine the position and orientation singularities, respectively.

4.3.1 Singularity Identification in PUMA

In PUMA, or any PUMA-like manipulators (or referred to as anthropomorphic with spherical wrist manipulator in [8]), there are typically three singularities: wrist lock (type 2), elbow lock (type 1), and head lock (type 2). These singularities can happen individually, or as a combination of two of even three at the same time.

The frame assignment and the table of Denavit-Hartenberg parameters used in this experiment can be found in Appendix A.

With the frame assignment shown in Figure A.1, and modified DH parameters according to Craig's [2](see Table A.1 in Appendix)(other sets of DH parameters can

be found in [7] or [49]) , the determinant of PUMA is derived as:

$$\begin{aligned} det(\mathbf{J}) &= det(\mathbf{J}_{11})det(\mathbf{J}_{22}); \\ det(\mathbf{J}_{11}) &= -a_2(d_4c_3 - a_3s_3)(d_4s_{23} + a_2c_2 + a_3c_{23}); \\ det(\mathbf{J}_{22}) &= -s_5; \end{aligned} \tag{4.2}$$

(Cheng et al[50]) where the parameters are defined in Appendix A.

Singularity therefore occurs when:

- $-a_2(d_4c_3 - a_3s_3) = 0$,which yields the elbow lock with $q_3 = 92.69^o$
- $(d_4s_{23} + a_2c_2 + a_3c_{23}) = 0$, which yields the head lock and
- $-s_5 = 0$, which yields the wrist singularity.

(4.3)

After the singular configurations are identified, a handling algorithm is then designed to handle the motion through these singularities in a stable manner. This is covered in Chapters 5-7.

4.4 Singularity Identification for Redundant Manipulator

In a redundant manipulator, identifying singularities has to be done by evaluating $det(\mathbf{J}.\mathbf{J}^T)$, due to the non-square Jacobian matrix.

A method of simplifying the identification process for redundant manipulator is presented in [48, 51]. On top of that, the Jacobian decoupling method for an anthropomorphic arm as explained in Section 2.3 was utilised to simplify the process further by placing the operational point of a serial arm at the intersection point at the wrist point [52]. It is summarised in Section 4.4.1.

4.4.1 Separating Jacobian into Position and Orientation

The Jacobian matrix is split into $\mathbf{J}_V$ (the top three rows of the Jacobian, which map the joint space velocities to operational space velocities) and $\mathbf{J}_\omega$ (bottom three rows, map joint velocities to angular velocity of the end-effector).

$$\begin{pmatrix} \mathbf{v}_p \\ \boldsymbol{\omega}_p \end{pmatrix} = \begin{pmatrix} \mathbf{J}_V \\ \mathbf{J}_\omega \end{pmatrix} \begin{pmatrix} \delta q_1 \\ \delta q_2 \\ ... \\ \delta q_n \end{pmatrix} \tag{4.4}$$

These top and bottom halves of the Jacobian matrices were solved separately to determine the symbolical expression for singularities. The determinant of $\mathbf{J}_V$ was calculated symbolically. When this expression goes to zero, $\mathbf{J}_V$ and therefore $\mathbf{J}$ is rank deficient. The resulting configurations are termed *position singularity*. When $\mathbf{J}_\omega$ is rank-deficient, the resulting configurations are called *orientation singularity*. The exact conditions presented in [48] for orientation singularity in a 3D space task specification were:

$$\begin{array}{ll} 1. & rank(\mathbf{J}_{21}) < 3 \\ 2. & rank(\mathbf{J}_{22}) < 3 \\ 3. & rank(\mathbf{J}_{21}|\mathbf{J}_{22}) < 3 \end{array} \tag{4.5}$$

where $\mathbf{J}_{21}$, $\mathbf{J}_{22}$ are as defined in (2.18) and in (4.1).

This process has its drawback, which is explained in Section 4.5.

4.4.2 Utilising the Minors of the Jacobian Matrix

The technique of separating the Jacobian into top and bottom halves and evaluating position and orientation singularities reduces the complexity of the problem. Instead of solving for a $6 \times n$ Jacobian matrix, it is reduced to solving for two $3 \times n$ sub-matrices.

To obtain the expression that causes the determinant of an $m \times n$ matrix to go to zero, where $n > m$, the $m \times m$ minors of the Jacobian matrix are utilised, as in [48]. The minors of the Jacobian matrix were also used as performance measure of a manipulator in [20].

The method used is to identify the expression that causes all the minors of a redundant matrix to have determinants of zero. This expression will cause the original matrix to be rank deficient. This is the result of the Binet-Cauchy identity of:

$$det(\mathbf{J}\mathbf{J}^T) = \sum_{i=1}^{p} m_i \tag{4.6}$$

where m_i are the determinant of the minors of $\mathbf{J}$ and p is the number of $m \times m$ minors of the Jacobian matrix.

4.4.3 Example: Mitsubishi PA-10 (7 DoF Articulated Robot)

The method will be illustrated with example of the PA-10 manipulator. To simplify the expression, the Jacobian was expressed as its simplest form in the fourth frame as the following:

$${}^4\mathbf{J}_{11} = \begin{bmatrix} (d_5 + d_3c_4)s_2s_3 & c_3(d_5 + d_3c_4) & 0 & d_5 \\ c_3(d_3 + d_5c_4)s_2 + d_5c_2s_4 & (-d_3 - d_5c_4)s_3 & d_5s_4 & 0 \\ d_3s_2s_3s_4 & d_3c_3s_4 & 0 & 0 \end{bmatrix} \tag{4.7}$$

$${}^4\mathbf{J}_{21} = \begin{bmatrix} -c_3c_4s_2 - c_2s_4 & c_4s_3 & -s_4 & 0 \\ s_2s_3 & c_3 & 0 & 1 \\ c_2c_4 - c_3s_2s_4 & s_3s_4 & c_4 & 0 \end{bmatrix} \tag{4.8}$$

$${}^4\mathbf{J}_{22} = \begin{bmatrix} 0 & -s_5 & c_5s_6 \\ 0 & c_5 & s_5s_6 \\ 1 & 0 & c_6 \end{bmatrix} \tag{4.9}$$

where

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{0}_{3\times3} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix} \tag{4.10}$$

We can now separately analyze the position and the orientation singularities. The frame assignment of the manipulator is shown in Figure 4.1

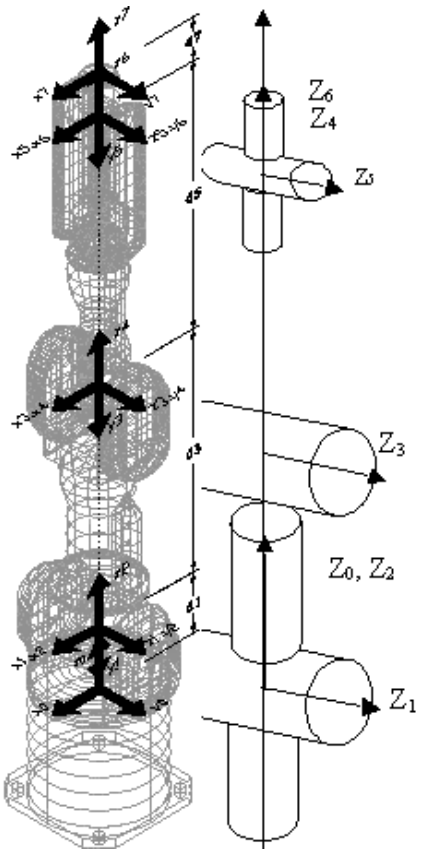


Figure 4.1: Seven dof Mitsubishi PA-10 and its frame assignment

Position Singularity

To identify position singularities, it is necessary to evaluate if sub-matrix $\mathbf{J}_{11}$ is rank-deficient.

The decoupling process has simplified the process from finding the determinant of a 6×7 Jacobian to 3×4 Jacobian matrix symbolically. However, the Jacobian is still

not square and it is still computationally intensive to obtain the symbolic expression for $det(\mathbf{J}\mathbf{J}^T) = 0$ to determine singularity.

To utilise the Binet-Cauchy identity (Equation 4.6), distinct order 3 minors of the Jacobian sub-matrix ${}^4\mathbf{J}_{11}$ are defined as follows:

Let ${}^4\mathbf{J}_{11} = [{}^4\mathbf{j}_{11}^1 \ {}^4\mathbf{j}_{11}^2 \ {}^4\mathbf{j}_{11}^3 \ {}^4\mathbf{j}_{11}^4]$, where ${}^4\mathbf{j}_{11}^k$ are the columns of ${}^4J_{11}$ matrix. We declare the minors as:

$m_1 = det[{}^4\mathbf{j}_{11}^1 \ {}^4\mathbf{j}_{11}^2 \ {}^4\mathbf{j}_{11}^3]$

$m_2 = det[{}^4\mathbf{j}_{11}^1 \ {}^4\mathbf{j}_{11}^2 \ {}^4\mathbf{j}_{11}^4]$

$m_3 = det[{}^4\mathbf{j}_{11}^1 \ {}^4\mathbf{j}_{11}^3 \ {}^4\mathbf{j}_{11}^4]$

$m_4 = det[{}^4\mathbf{j}_{11}^2 \ {}^4\mathbf{j}_{11}^3 \ {}^4\mathbf{j}_{11}^4]$

The resulting expressions are:

$$\begin{aligned} m_1 &= 0 \\ m_2 &= d_3 d_5 S_4((d_3 + d_5 C_4)S_2 + d_5 C_2 C_3 S_4) \\ m_3 &= -d_3 d_5^2 S_2 S_3 S_4^2 \\ m_4 &= -d_3 d_5^2 C_3 S_4^2 \end{aligned} \tag{4.11}$$

To obtain the condition of $det(\mathbf{J}\mathbf{J}^T) = 0$, we need to find the expressions that would make all of m_1 to m_4 equal to 0.

From Table 4.1, it is shown that there are two position singularities:

- when $sin(q_4) = 0$, therefore $q_4 = 0$. It should also be noted that since $sin(q_4)$ in itself causes $\mathbf{J}_{11}$ to be rank deficient, any combination of $sin(q_4)$ with other possible terms would also produce rank deficiency, namely: ($s_2 = 0$ AND $s_4 = 0$), ($s_3 = 0$ AND $s_4 = 0$), ($c_3 = 0$ AND $s_4 = 0$), ($s_2 = 0$ AND $s_3 = 0$ AND $s_4 = 0$), and ($s_2 = 0$ AND $c_3 = 0$ AND $s_4 = 0$). This may sound trivial,

Table 4.1: This table shows which terms in the determinants of the minors set M_1 to M_4 to zero.

	m_1	m_2	m_3	m_4
$c_3 = 0$	$\surd$			$\surd$
$s_2 = 0$	$\surd$		$\surd$	
$s_3 = 0$	$\surd$		$\surd$	
$s_4 = 0$	$\surd$	$\surd$	$\surd$	$\surd$
$((d_3 + d_5c_4)S_2 + d_5c_2C_3s_4) = 0$				
$\bullet c_4 = -(d_3/d_5)$ AND $c_2 = 0$	$\surd$	$\surd$	$\surd$	$\surd$
$\bullet c_4 = -(d_3/d_5)$ AND $c_3 = 0$	$\surd$	$\surd$		$\surd$
$\bullet s_2 = 0$ AND $c_3 = 0$	$\surd$	$\surd$	$\surd$	$\surd$
$\bullet s_2 = 0$ AND $s_4 = 0$	$\surd$	$\surd$	$\surd$	$\surd$

because all these configurations are the same family of singularities caused by $sin(q_4) = 0$. However, these may have additional singular directions, which will be covered in Section 4.6.6.

- when $sin(q_2) = 0$ AND $cos(q_3) = 0$, ie. $q_2 = 0$ or 180^o AND $q_3 = \pm 90^o$.

It is not possible for $q_4 = 180^o$ because of joint limits.

Orientation Singularity

Referring to (4.5) and [48], orientation singularity happens when $\mathbf{J}_{21}$ and $\mathbf{J}_{22}$ are rank-deficient, and that the linear combination of $\mathbf{J}_{21}$ and $\mathbf{J}_{22}$ are rank-deficient.

For $\mathbf{J}_{21}$ to be rank deficient

Using the same method to determine position singularities, it was found that for $det(\mathbf{J}_{21}\mathbf{J}_{21}^T) = 0$, it was required that $sin(q_2) = 0$ and $sin(q_3) = 0$.

For $\mathbf{J}_{22}$ to be rank deficient

$det(\mathbf{J}_{22}) = -sin(q_6)$. For it to be zero, then $q_6 = 0, \pm 180$.

For the lower half of ${}^4\mathbf{J}$ rank deficient

Let $\mathbf{J}_\omega$ be the combined matrix of $\mathbf{J}_{21}$ and $\mathbf{J}_{22}$ (*i.e.* the lower half of the full Jacobian ${}^4\mathbf{J}$). Then

$$\begin{aligned} det(\mathbf{J}_\omega \mathbf{J}_\omega^T) = 8 - 4cos(2q_4) - cos(2(q_4 - q_5)) \\ -2cos(2q_5) - cos(2(q_4 + q_5)) \end{aligned} \tag{4.12}$$

For (4.12) to be zero, then $q_4 = 0$ and $q_5 = 0$, which works out to be a special case of elbow singularity of $q_4 = 0$.

4.4.4 Summary of singularities in PA-10

The 7 DOF Mitsubishi PA-10 has singularities at:

- boundary or elbow singularity, when $q_4 = 0$.
- internal position singularity, when $q_2 = 0$ or 180^o and $q_3 = \pm 90^o$ or it can be written as: $sin^2(q_2) + cos^2(q_3) = 0$
- orientation singularity, when the arm is vertically pointing up, *i.e.* when q_2 to $q_5 = 0$. The robot is in both position and orientation singularity, therefore there would be more than one degenerate directions.

4.5 Completeness of Solution

Splitting the Jacobian into $\mathbf{J}_V$ and $\mathbf{J}_\omega$ and evaluating the Jacobian separately to identify singularities present a possibility of an incomplete set of solutions, meaning it may not identify all the possible singular configurations.

Although it is well known in the case of PUMA, it often escapes the attention when evaluating the singularities for redundant manipulator.

For PUMA 6DoF manipulator, it was shown that the determinant can be evaluated as $det(\mathbf{J}) = det(\mathbf{J}_{11}).det(\mathbf{J}_{22})$. This means that the Jacobian could be rank-deficient regardless of $\mathbf{J}_{21}$. If the top and bottom halves of the Jacobian matrix were evaluated separately, then the singularity case of $det(\mathbf{J}_{22}) = 0$ would be missed, because if $\mathbf{J}_{21}$ was of full-rank, then $[\mathbf{J}_{21}|\mathbf{J}_{22}]$ would also be of full-rank.

This case is also extended in redundant manipulators. It is often assumed that because by definition the position and orientation halves of each column of the Jacobian matrix is derived orthogonally, that entire top and bottom halves of the Jacobian matrix are orthogonal to each other. See the definition of column i of the Jacobian matrix 4.13 [8]:

$$\begin{array}{cc} \begin{pmatrix} \mathbf{z}_i \times (\mathbf{p} - \mathbf{p}_0) \\ \mathbf{z}_i \end{pmatrix} & ; \quad for \quad revolute \quad joint \quad i \\ \begin{pmatrix} \mathbf{z}_i \\ \mathbf{0} \end{pmatrix} & ; \quad for \quad prismatic \quad joint \quad i \end{array} \tag{4.13}$$

Looking at the definition of Jacobian for revolute joints, column $\mathbf{J}_{V,i} = \mathbf{z}_i \times (\mathbf{p} - \mathbf{p}_0)$ is orthogonal to $\mathbf{J}_{\omega,i} = \mathbf{z}_i$. Whereas for prismatic joint, $\mathbf{J}_{V,i} = \mathbf{z}_i$ and $\mathbf{J}_{\omega,i} = \mathbf{0}$.

However, when the top and bottom half of the columns are orthogonal, it does not mean that the entire top and bottom halves of the Jacobian matrix are orthogonal to each other. Dependant equations can still exist on the top and bottom halves, which will present a problem when singularities are evaluated separately for position and orientation. When dependent rows occur in top and bottom halves of the Jacobian matrix, it is not simply a position nor orientation singularity, it is a *coupled singularity*, resulted from the coupling between the joints used for end-effector tasks in position and in orientation. This is a conflict of 'resources' and the manipulator loses the ability to move in certain direction because the degree of mobility required is occupied. The

manipulator is able to perform task in the singular direction if all the available joints were dedicated to only orientation or only position of the end-effector, but not both.

Following this finding, it is therefore not correct to describe singularity as the configuration where the manipulator loses one or more of its degrees of freedom.

Singularity should be defined as *the configuration whereby the manipulator no longer possesses the full 6 DoFs* (or just the "full DoFs", for the case of manipulators with less than 6 DoFs).

To illustrate the problem better, the example below on PUMA 560 is shown. Let the PUMA be at the wrist singular configuration:

$q_1 = 0$, $q_2 = -45^o$, $q_3 = 135^o$, $q_4 = q_5 = q_5 = 0$ (Figure 4.2)

At this singular configuration, the Jacobian matrix with respect to Frame{4} (DH frame attached to joint 4) is:

$$ {}^4\mathbf{J} = \begin{bmatrix} 0 & 0.74 & 0.43 & 0 & 0 & 0 \\ 0.74 & 0 & 0 & 0 & 0 & 0 \\ -0.06 & 0.33 & 0.02 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4.14} $$

As shown in Equation 4.14, the rows of the Jacobian matrix that represent the translation in Y axis of Frame {4} and the rotation around X axis of Frame{4} are dependant. The robot is not capable of translating in Y direction of Frame{4} and rotating around X axis of Frame{4} at the same time (see Figure 4.2).

While it is clear-cut for a non-redundant manipulator, it is not so for a redundant serial manipulator with a spherical wrist, where there are 4 joints assigned to actuate

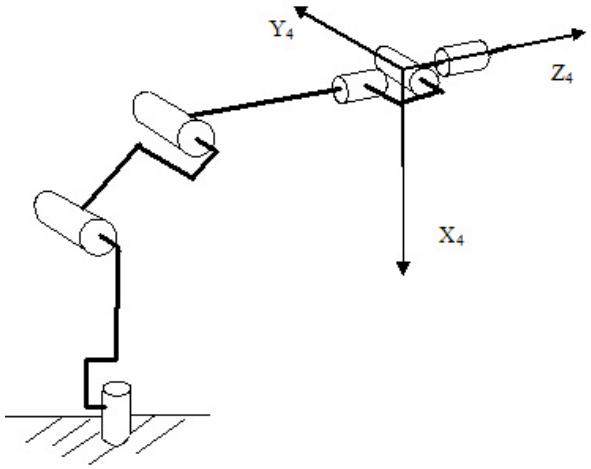


Figure 4.2: The wrist singularity expressed in Frame 4. It is not able to translate in Y and rotate in X axis of Frame 4 simultaneously.

the position of the end-effector. There is, therefore, a redundant degree of freedom in the task which can then be used to supplement the orientation task when singular. This is the assumption made in [48] which define the condition of determining the orientation singular configuration as shown in Equation 4.5.

However, this may not hold true for all cases. The important thing to note is that the extra degree of freedom in the first four joints would have to have a component in the singular direction of the wrist singularity for these condition to hold. A simple example is given below to illustrate the rare cases where this condition may not be true.

Supposed the 7 DoF manipulator has a PUMA-like structure, with an addition of a revolute joint as indicated in Figure 4.3.

When joint 6 is straightened ($q_6 = 0$), the spherical wrist loses one degree of freedom, which is the rotation around X_5 (see Figure 4.4). The position task, however,

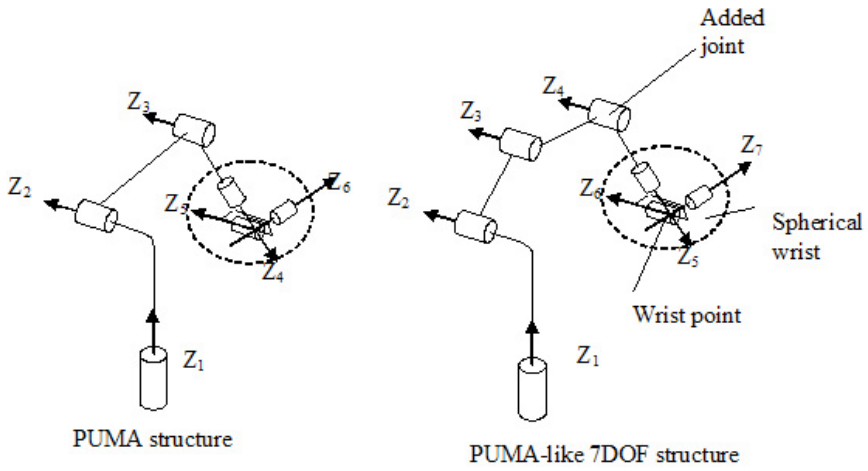


Figure 4.3: The structure of a PUMA manipulator (left) and an example of a 7 DoF PUMA-like manipulator with spherical wrist (right).

possesses 4 joints and therefore redundant. This extra DOF can be used to supplement the lost DoF at the wrist.

In some manipulator designs, however, the additional joint in may not guarantee that it would supplement the loss DoF when the wrist is straightened. The example shown in Figure 4.3 (right). When the wrist is straightened, $q_5 = 0$ or 180^o (see Figure 4.4), the additional joint does not help in the direction of lost degree of freedom. In this configuration, the manipulator is still not able to rotate around X_5. It is only possible to provide motion in that direction through joint 1. However, motion in joint 1 would affect the position of the end-effector. This manipulator is therefore in a singular configuration, as it is still lacking the capability to rotate around X_5.

Evaluating only the bottom three rows of the Jacobian matrix would ignore this fact and therefore fail to identify the singularity at $sin^2(q_5) + sin^2(q_6) = 0$. Here, $\mathbf{J}_\omega$ is full rank while the complete Jacobian $\mathbf{J}$ is rank deficient or singular.

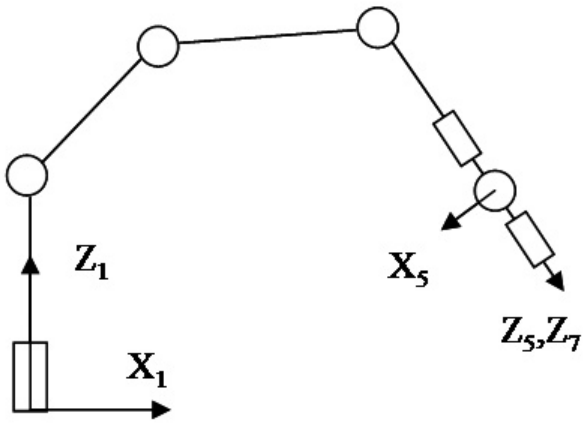


Figure 4.4: The 7DOF manipulator, side view (planar). The wrist is straightened ($q_6 = 0$) with $q_5 = 0$. It is not able to rotate around axis X_5.

With the Jacobian matrix expressed symbolically in Appendix A, evaluating the orientation singularity following the technique described in [48] would yield:

subcondition 1: $rank(\mathbf{J}_{21}) < 3$ $\mathbf{J}_{21}$ expressed in Frame$\{5\}$ is shown below:

$$ {}^5\mathbf{J}_{21} = \begin{pmatrix} -c_5 s_{234} & s_5 & s_5 & s_5 \\ -s_5 s_{234} & c_5 & c_5 & c_5 \\ c_{234} & 0 & 0 & 0 \end{pmatrix} \tag{4.15} $$

The determinant of $\mathbf{J}_{21}$ is zero when:

- $s_{234} = 0$ AND $s_5 = 0$
- $s_{234} = 0$ AND $c_5 = 0$
- $c_{234} = 0$

subcondition 2: $rank(\mathbf{J}_{22}) < 3$ The determinant is evaluated to be $sin(q_6)$. Setting $q_6 = 0$ or 180^o,

$$ {}^5\mathbf{J}_{22} = \begin{pmatrix} 0 & 0 & S_6 \\ 0 & 1 & 0 \\ 1 & 0 & C_6 \end{pmatrix} \tag{4.16} $$

The determinant is zero when $s_6 = 0$

subcondition 3: $rank({}^5\mathbf{J}_{21}|{}^5\mathbf{J}_{22}) < 3$ This is labelled as *Combined Orientation Singularity*. Setting $s_6 = 0$, for all three conditions to be true, then for $[{}^5\mathbf{J}_{21}|{}^5\mathbf{J}_{22}]$ to be rank deficient, the first row of the matrix has to be all zeros.

$$[{}^4\mathbf{J}_{21}|{}^4\mathbf{J}_{22}] = \begin{pmatrix} -c_5 s_{234} & s_5 & s_5 & s_5 & 0 & 0 & 0 \\ -s_5 s_{234} & c_5 & c_5 & c_5 & 0 & 1 & 0 \\ c_{234} & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{4.17}$$

In order to have [4.17] rank deficient, we need to choose the condition of singularity of $\mathbf{J}_{21}$ to be ($s_{234} = 0$ AND $s_5 = 0$ AND $s_6 = 0$).

Setting $s_5 = 0$ and $s_6 = 0$ (without $s_{234} = 0$) is enough to make the Jacobian matrix singular. Therefore the solution is incomplete.

Although this is a singularity that exists on conventional robots, which have been around for a long time, to the best of our knowledge, it has not been classified in the literature. This is not a semi singularity or an algorithmic singularity. It is still a kinematic singularity, it is inherent in the mechanical design of the manipulator, it is not caused by control algorithm or any joint limits or obstacles in the environment. In this dissertation, it is termed *coupled singularity*.

To properly address this issue, the idea of singular direction must be covered first. The following section will cover the idea of singular direction before discussing about how to check the completeness of the solution to the set of singularities in redundant manipulator.

4.6 Identifying the Singular Direction

When singularity occurs, there is a row(s) in the Jacobian - when it is transformed onto the correct frame - that contains only zeros. By definition that $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, a zero

row in the Jacobian means that there is a direction (or a degree of freedom) in task space that the manipulator is unable to move in, regardless of the joint rate command - hence the singular direction.

Singular direction is difficult to determine symbolically. Most of the time, when a singularity is identified, the singular direction is obtained just by observation.

For a singular direction to be described mathematically and to be physically meaningful, it has to be expressed as a vector with respect to a frame that has physical meanings in terms of the robot configuration. To declare any arbitrary frames usually results in very complicated expression to evaluate. The natural solution is to define singular directions in relation to the Denavit-Hartenberg frames of the robot. For example, a singular direction can be described as "the translation along X axis of Frame{1} of the DH frames". The following subsections discuss the singular directions of different singular configurations in PUMA.

4.6.1 Head Lock

When PUMA manipulator *head lock* (see Section 3.3.1) occurs, $a_2C_2 + a_3C_{23} + d_4S_{23} = 0$ (Equation 4.3). Equation(4.18) shows the top half of the Jacobian ${}^1\mathbf{J}$, which is the Jacobian matrix in Frame{1}. From this equation, it is shown that the second row of the Jacobian is all zeros, which corresponds to the translation along Y-axis of Frame{1} which is the degenerate direction.

$$ {}^1\mathbf{J} = \begin{bmatrix} -d_2 - d_3 & d_4c_{23} - a_2s_2 - a_3s_{23} & d_4c_{23} - a_3s_{23} & 0 & 0 & 0 \\ a_2c_2 + a_3c_{23} + d_4s_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_2c_2 - a_3c_{23} - d_4s_{23} & -a_3s_{23} - d_4s_{23} & 0 & 0 & 0 \end{bmatrix} \quad (4.18) $$

4.6.2 Elbow lock

There are some singular directions that cannot be described by any of the DH frames of the robots. This commonly happens with robots with offset joints, like the PUMA. Robots with intersecting rotational axes such as the Mitsubishi PA-10 does not have such problem. An example of this is the elbow singularity of PUMA. It is shown by projecting the Jacobian and the task space forces onto Frame{B} which is not one of the frames in our DH assignment (see Figure 4.5 for the frame assignment, and Equation (4.19) for the resulting Jacobian). At this configuration, the singular direction is found to fall along the line connecting the wrist point to the origin of base frame.

$$
\begin{aligned}
{}^{B}\mathbf{J}[1][1] &= \tfrac{(d_2+d_3)s_2(d_4c_3-a_3s_3)}{(a_2+a_3c_3+d_4s_3)h} \\
{}^{B}\mathbf{J}[1][2] &= \tfrac{(d_4c_3-a_3s_3)(a_2+a_3c_3+d_4s_3)^2h}{a_2^2+d_2^2+2d_2d_3+d_3^2+2a_2a_3c_3+a_3^2c_3^2+2a_2d_4s_3+d_4^2s_3^2+a_3d_4Sin[2q3]} \\
{}^{B}\mathbf{J}[1][3] &= \tfrac{d_4c_3-a_3s_3}{h} \\
{}^{B}\mathbf{J}[2][1] &= (d_2+d_3)s_2 \\
{}^{B}\mathbf{J}[2][2] &= a_2+a_3c_3+d_4s_3 \\
{}^{B}\mathbf{J}[2][3] &= a_3c_3+d_4s_3 \\
{}^{B}\mathbf{J}[3][1] &= \tfrac{c_2(a_2^2+d_2^2+2d_2d_3+d_3^2+a_2a_3c_3+a_2d_4s_3)+(a_2+a_3c_3+d_4s_3)(a_3c_{23}+d_4s_{23})}{(a_2+a_3c_3+d_4s_3)h} \\
{}^{B}\mathbf{J}[3][2] &= \tfrac{(d_2+d_3)(d_4c_3-a_3s_3)}{(a_2+a_3c_3+d_4s_3)h} \\
{}^{B}\mathbf{J}[3][3] &= \tfrac{(d_2+d_3)(d_4c_3-a_3s_3)}{(a_2+a_3c_3+d_4s_3)h}
\end{aligned}
\tag{4.19}
$$

where: $h = \sqrt{1+\frac{(d_2+d_3)^2}{(a_2+a_3C_3+d_4S_3)^2}}$

Frame{B} is obtained by rotating Frame{2} by angle β, (see Figure 4.5) which is defined as:

$$\beta = tan^{-1}\left[\frac{d_2+d_3}{a_2+a_3c_3+d_4s_3}\right] \tag{4.20}$$

From Equation 4.3, it is shown that $-a_2(d_4c_3 - a_3s_3) = 0$ at elbow singularity. Therefore, the first row of ${}^{B}\mathbf{J}_{11}$ is a zero row (${}^{B}\mathbf{J}[1][1] = {}^{B}\mathbf{J}[1][2] = {}^{B}\mathbf{J}[1][3] = 0$). This shows that the singular (degenerate) direction lies along the X-axis of Frame{B}

(see Figure 4.5). Equation 4.19 only shows the elements ${}^{B}\mathbf{J}$ from the top left quadrant ($\mathbf{J}_{11}$), because the top right quadrant is a zero matrix.

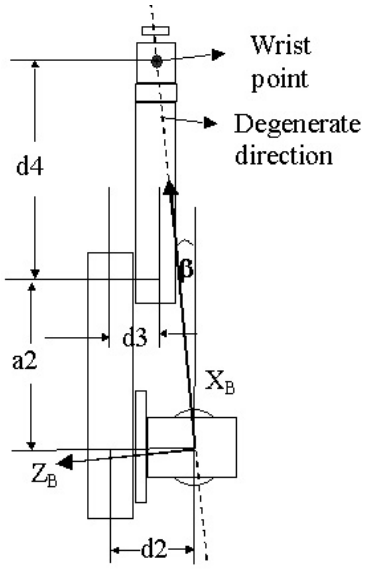


Figure 4.5: PUMA, from top view, shows the degenerate direction at elbow singularity, expressed in Frame{B}, which is derived from rotating Frame{2} by angle {b}

4.6.3 Wrist lock

In PUMA wrist lock, there is no entire row of zeros in the Jacobian matrix when it is expressed in any physically meaningful frame (such as the DH frames or one obtained by rotating one of the DH frames). This is the same case as that described in Section 4.5, where a certain DoF in the wrist is disabled, and where there is or are other joints available to supplement that lost DOF, however it is being used for another task.

The singular direction in PUMA wrist lock is the rotation around the X axis of Frame {4}. This singularity occurs when $\mathbf{J}_{22}$ is rank deficient, regardless of what $\mathbf{J}_{21}$ could be. Therefore a row of zeros may not exist when the Jacobian matrix is expressed with respect to Frame {4}. Section 4.6.4 shows that $\mathbf{J}$ will still have a row of zero when transformed to a particular frame, however, this frame no longer has any physical meaning.

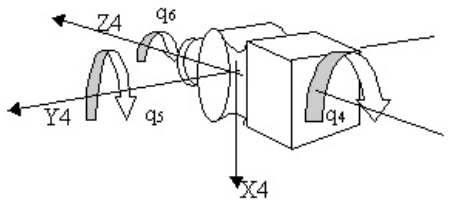


Figure 4.6: The wrist of PUMA 560, showing $Frame\{4\}$

4.6.4 On Whether There is Always a Zero Row

The last case of singularity, such as in PUMA wrist, shows that when expressed in the singular frame, there is a possibility that there is no entire row of zeros. This shows that when in wrist singularity, it is possible to rotate around X_4, but it has to be performed by joints that are already being used to perform position control of the end-effector.

As elaborated in Section 3.2.2, when the Jacobian matrix is rank deficient, then the last k entries of σ_i in the diagonal matrix Σ will be zero. k is the number of rank the Jacobian loses at the singularity. When this happens, $\mathbf{\Sigma V}^T$ will result in a $6 \times n$ matrix with the last k rows being zero.

Taking $\mathbf{U}^T$ as a transformation matrix, it can be said that there is always a way to transform the Jacobian matrix to another frame that guarantees the existence of a zero row(s) during singularity. However, this frame may not have any physical meaning as this process mixes up the translational and the rotational halves of the Jacobian matrix.

When the SVD is performed on a Jacobian matrix that maps the joint velocities to only translation velocity in task space, then $\mathbf{U}$ will be a 3×3 rotational matrix and its columns describe the direction principal axes of motion of the end-effector, and the last k columns describe of the translation singular direction. k is the number of rank that the manipulator loses in the singular configuration. Similarly, when performed on Jacobian matrix that maps the joint velocities only to angular velocity in task space, matrix $\mathbf{U}$ is a rotational matrix and its columns describe the axes of rotation of the end-effector motion.

When the Jacobian matrix is evaluated as a $6 \times n$ matrix, the translational and rotational halves are mixed up and any physical meaning is lost. It should also be noted that for orientation of some serial manipulator such as PUMA, the orientation singularity (the wrist singularity) cannot be identified fully just by taking the rotational (lower) half of the Jacobian matrix. For the specific case of the PUMA robot, it can be evaluated by taking the SVD of only $\mathbf{J}_{22}$.

Despite having no physical meaning, matrix $\mathbf{U}$ is still a valid transformation. We use this in our control strategy to perform the singularity handling.

4.6.5 On Identification of Singular Direction

As mentioned above, the singular directions in the PUMA robot were identified by observation after the independent singular configurations were identified. These singular directions are expressed as a vector in a certain frame, such as "rotation around the X axis of Frame {4}".

Searching for it symbolically would be difficult. One way to obtain an orthonormal transformation matrix to transform the Jacobian matrix into a singular frame, whereby one of the axes is the degenerate direction is by utilising SVD. This is done as shown by Equation 3.3. However, this process will lose the physical meaning of the singular frame and direction when it is performed on a 6 DoF task.

If the singular direction is one of the axes of the Denavit Hartenberg frames defined for the manipulator joints, it can be identified with the following method. When a singular configuration is identified, the singular condition is substituted back into the Jacobian matrix. The idea is then to transform the Jacobian matrix into all the DH frames of the robot (*i.e.* premultiplying the Jacobian matrix with respect to base Frame (${}^0\mathbf{J}$) with ${}^0\mathbf{R}_i^T$, where i is the i-th Frame of the DH assignment) and check for the existence of zero rows (Equation 4.21). The zero row is then the singular direction expressed in terms of one of the DH frames of the robot.

$$ {}^i\mathbf{J} = ({}^o\mathbf{R}_i)^T . {}^0\mathbf{J} \tag{4.21} $$

${}^i\mathbf{J}$ is the Jacobian matrix with respect to the i-th frame. When there are n joints in the robot, then the evaluation is performed $n \times k$ times, once at each DH frame, where n is the number of DH frames and k is the number of distinct singularities. One that yields a row of zero is a singular direction.

Searching for singular directions that don't lie on any axes of any DH frames yields very complicated expressions, such as shown in Equation 4.19 for PUMA elbow singularity. Cases of coupled singularities, such as discussed in wrist singularity of PUMA would also not yield a row of zeros.

4.6.6 Singular Value Decomposition in Determining Singular Directions

As shown in Section 4.6.4, SVD can be used to decompose the Jacobian matrix to show the row of zeros when the manipulator is in singular configurations. This section intends to expand more on the role of SVD and other information that can be gained from its usage.

Let us examine the example of a two link planar manipulator, such as one illustrated in Figure 3.1. As shown earlier, the Jacobian matrix of such manipulators can be expressed as:

$$ {}^{0}\mathbf{J} = \begin{pmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{pmatrix} \tag{4.22} $$

The Jacobian above is expressed with respect to the Frame $\{0\}$. When $q_2 = 0$, the manipulator is in singular configuration. For a condition where $q_1 = 0$ and $q_2 = 0$ and defining $l_1 = 5$ and $l_2 = 3$, then the SVD of the Jacobian matrix becomes:

$$ \mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.23} $$

$$ \mathbf{J} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 8.5440 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0.9363 & -0.3511 \\ 0.3511 & 0.9363 \end{pmatrix} \tag{4.24} $$

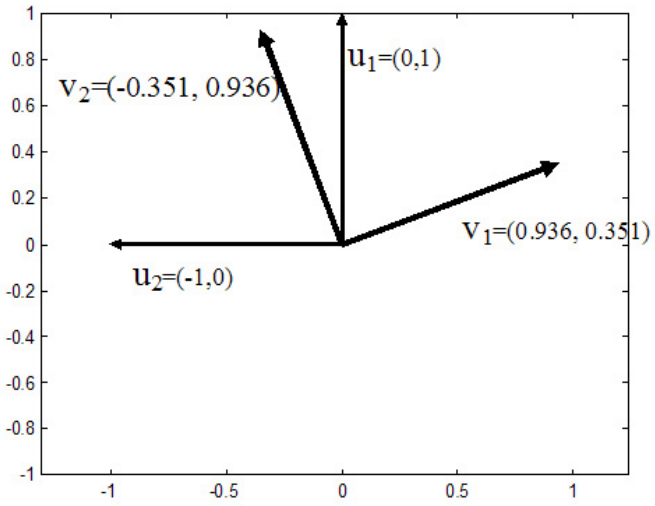


Figure 4.7: The singular vectors which form the columns of matrix **U** and **V** in the SVD of the Jacobian matrix in the example.

The matrix **Σ** contains the singular values along its diagonal while the columns of matrix **U** and **V** contain the singular vectors. It is shown that the manipulator is singular as the smallest singular values is zero.

In the configuration chosen for the example ($q_1 = 0$ and $q_2 = 0$), the singular direction of the manipulator is the translation along the X axis of Frame $\{0\}$. It is also shown that the singular vectors associated with the singular value of 0, which is the second column of the matrix **U** which is $(-1, 0)^T$, is the vector describing the singular direction of the manipulator. The first column of the matrix **U** which is $(0, 1)^T$ is associated with the singular value 8.5440 and therefore described the direction perpendicular to the singular direction.

The vectors in columns of matrix **U** and matrix **V** is shown in Figure 4.7.

For another configuration, for example $q_1 = 150^0$ and $q_2 = 0$, the SVD of the Jacobian matrix becomes:

$$\mathbf{J} = \begin{pmatrix} -0.5000 & -0.8660 \\ -0.8660 & 0.5000 \end{pmatrix} \begin{pmatrix} 8.5440 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0.9363 & -0.3511 \\ 0.3511 & 0.9363 \end{pmatrix} \tag{4.25}$$

In this configuration, the elbow of the manipulator is still straight ($q_2 = 0$) and the manipulator arm is pointing at 150^o from the X axis of Frame $\{0\}$. The singular direction is still the translation along X axis of Frame $\{1\}$, or the translation along the line forming an angle of 150^o from the X axis of Frame$\{0\}$.

Again, it is shown that the first column of the matrix $\mathbf{U}$ describes the direction perpendicular to the singular direction while the second column describes the singular direction. To see that, notice that $atan2(\frac{0.5000}{-0.8660}) = 150^o$.

Another important point is that the SVD has the same values for matrices $\mathbf{\Sigma}$ and $\mathbf{V}$. It can be said that matrix $\mathbf{U}$ describes the task space behaviour of the manipulator. Its transpose is also a valid transformation matrix that would transform the Jacobian matrix into the singular frame $\{S\}$ (see Equation 4.26.) The expression $\mathbf{\Sigma V}$ is, therefore, the Jacobian matrix of the manipulator expressed in the singular frame $\{S\}$.

$$\begin{array}{rcl} {}^0\mathbf{J} & = & \mathbf{U\Sigma V}^T \\ \mathbf{U}^T{}^0\mathbf{J} & =^S \mathbf{J} = & \mathbf{\Sigma V}^T \end{array} \tag{4.26}$$

Application on PUMA singular direction

The exercise above is applied to the head singularity of PUMA, where $a_2.C_2 + a_3.C_{23} + d_4.S_{23} = 0$. The lost degree of freedom, as explained earlier, is the translation along the Y axis of Frame $\{1\}$.

Let $q_1 = 0$, $q_2 = -45^o$, and $q_3 = 2.9177^o$. This combination of q_2 and q_3 causes $a_2.C_2 + a_3.C_{23} + d_4.S_{23} = 0$ and hence head singularity. As explained in Section 4.6.4, that the SVD of the Jacobian matrix in 6DoF will cause the transformation matrix to lose its physical meaning. The analysis in this section will be performed on the position and orientation halves of the Jacobian matrix separately.

Analysing the top half of the Jacobian only, taking the top half as a 3×6 matrix:

$$\mathbf{J}_V = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T; \tag{4.27}$$

$$\mathbf{U} = \begin{pmatrix} -0.8952 & 0.4457 & 0.0000 \\ -0.0000 & 0.0000 & -1.0000 \\ 0.4457 & 0.8952 & 0.0000 \end{pmatrix} \tag{4.28}$$

$$\mathbf{\Sigma} = \begin{pmatrix} 1.4216 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1187 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0000 & 0 & 0 & 0 \end{pmatrix} \tag{4.29}$$

$$\mathbf{V} = \begin{pmatrix} 0.0945 & -0.5637 & 0.8205 & 0 & 0 & 0 \\ -0.8331 & -0.4961 & -0.2448 & 0 & 0 & 0 \\ -0.5451 & 0.6604 & 0.5165 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix} \tag{4.30}$$

The Jacobian above is expressed with respect to Frame$\{0\}$. As $q_1 = 0$ in the example, the arm is aligned with the X axis of Frame$\{0\}$, which means that the lost degree of freedom is the translation along the Y axis of Frame$\{0\}$.

In matrix $\mathbf{U}$, the third column correspond to the σ value of 0. The value of the vector is $\{0, -1, 0\}$, still expressed in Frame $\{0\}$, which corresponds to the Y axis (or negative Y axis, to be exact) of Frame $\{0\}$ or the singular direction.

The SVD can also be performed on the Jacobian matrix expressed in the singular frames. For the example of head singularity, the Jacobian matrix is then expressed

in Frame $\{1\}$. The result is that the vector associated with the singular direction will always show $\{0, -1, 0\}$ or $\{0, 1, 0\}$ because the singular direction is always the translation along Y axis when expressed in Frame$\{1\}$.

The same analysis can be performed on the elbow singularity. However, as highlighted in this chapter, wrist singularity is not contained only within the lower half of the Jacobian matrix. Analysis only the lower half of the Jacobian matrix $\mathbf{J}_\omega$ will not yield the complete solution of the singular configurations. When SVD is performed on the 6×6 Jacobian matrix, it will lose the physical meaning of the singular direction as highlighted in Section 4.6.4.

4.6.7 Families of Singularities with Additional Singular Direction: Mitsubishi PA-10

In PUMA manipulator, it was shown that there were three distinct singular configurations, each would reduce the rank of the Jacobian matrix by one. When all three happen together, which is when the arm is point vertically upwards, then the rank of the Jacobian matrix is reduced by 3, leaving only 3 available DOFs.

As summarised in Section 4.4.4, the Mitsubishi PA-10 has three singularities: when $q_4 = 0$ (elbow singularity) (and all the combinations: ($s_2 = 0$ AND $s_4 = 0$), ($s_3 = 0$ AND $s_4 = 0$), ($c_3 = 0$ AND $s_4 = 0$), ($s_2 = 0$ AND $s_3 = 0$ AND $s_4 = 0$), and ($s_2 = 0$ AND $c_3 = 0$ AND $s_4 = 0$), when q_2 to q_5 are all zero (orientation singularity), and when ($q_2 = 0$ and $q = \pm 90^o$) (internal singularity).

Observing the combinations of elbow singularities visually, one can identify the singular directions as shown in Table 4.2.

Table 4.2: Singular directions of Mitsubishi PA-10 at singular configurations where $q_4 = 0$

	Singular direction
$s_4 = 0$	translation along Z_4
$s_2 = 0$ AND $s_4 = 0$	translation along Z_4
$s_3 = 0$ AND $s_4 = 0$	translation along Z_4
$c_3 = 0$ AND $s_4 = 0$	translation along Z_4
$s_2 = 0$ AND $s_3 = 0$ AND $S_4 = 0$	translation along Z_4 AND Y_4
$s_2 = 0$ AND $c_3 = 0$ AND $S_4 = 0$	translation along Z_4
$s_2 = s_3 = s_4 = s_5 = 0$	translation along Z_4 AND Y_4 AND rotation around X_4

The point to pay attention to is that although $s_4 = 0$ would cause the Jacobian matrix to the rank deficient, further precaution has to be taken because its combination with other conditions can result in further reduction in the rank of the Jacobian.

Certain singularity handling algorithms such as the ones described in Chapter 5, resolve singularities by removing the singular direction from the task, therefore providing a stable control over the singular region. Therefore, in the case of $s_4 = 0$, removing the singular direction of translation along Z_4, would provide a stable control over the singular region as the Jacobian matrix is now of full-rank. However, when $s_4 = 0$ AND ($s_2 = 0$ AND $s_3 = 0$), the manipulator will further lose another degree of freedom. When this happens, there is an extra singular direction to be removed from the Jacobian, otherwise the manipulator performance over the singular region would be unstable.

When q_2 to q_5 are all zero, the manipulator also loses the rotational DOF around X axis of Frame{4}. In this configuration, the manipulator loses a total of 3 DoFs.

In internal singularity where ($q_2 = 0$ and $q_3 = \pm 90^o$), the manipulator loses the translational motion along Z axis of Frame{4} (see Figure 4.8)

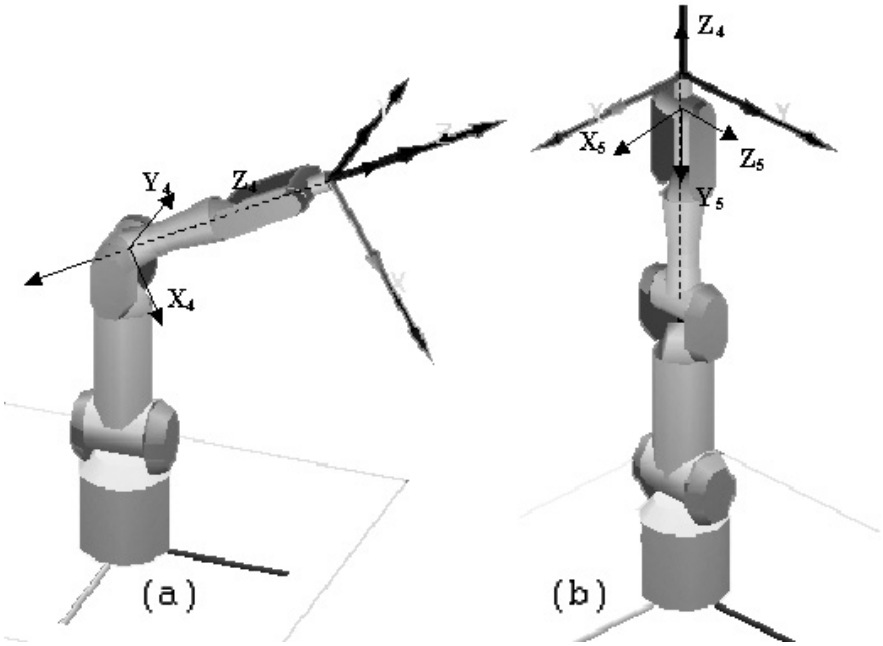


Figure 4.8: Mitsubishi PA-10 in singular configurations: left in internal singularity and right when the arm and the wrist are straightened, where the manipulator loses 3 DOFs

It is therefore important in the design of algorithm to handle singularities in a manipulator to thoroughly list not just all the possible singularities but also all the possible singular directions.

4.7 A Simple Check to the Complete Set of Solution to Singular Configurations of Redundant Manipulator

The simplification that the technique of dividing the Jacobian into position and orientation halves reduces the computational complexity tremendously. For some

cases, the redundant manipulator has a very complex Jacobian matrix, that solving for $det(\mathbf{J}.\mathbf{J}^T)$ is too difficult even for the modern computers and softwares. It is good to be able to use the simplification, while making sure we still get the complete set of solutions.

For the case of serial manipulators with a wrist attached to the end, a simple check is introduced below. These wrists are usually spherical. For these manipulators, placing the operational point at the wrist point (the point of intersection of all the rotational axes of the wrist) will cause only the first few joints to affect the position of the operational point. These joints also affect the orientation of the end-effector. The last few joints, the ones that belong to the wrist, only affects the orientation of end-effector.

The problem of possible incomplete solution can be summarized as follows: as the wrist loses rank ($rank(\mathbf{J}_{22}) < 3$), a redundant manipulator will have extra DOF(s) to supplement this shortage. When it is successfully supplemented, then it is not a singularity. If it is not supplemented, then the Jacobian is rank deficient, even when $\mathbf{J}_\omega = [\mathbf{J}_{21}|\mathbf{J}_{22}]$ is full rank.

The search for the incomplete or the missing singular configuration is therefore confined to that of orientation singularity. It is proposed that one method of checking is to be able to confine the possible configurations that the earlier method of identification had missed. These configurations are then substituted to the complete Jacobian matrix and its rank calculated to see if it becomes singular. The problem now is to determine what are the possible conditions of orientation singularity.

The lower half of the Jacobian matrix contains the $\mathbf{z}_i$ rotational axes of all the revolute joints in the manipulator (see definition of Jacobian matrix (see Equation 4.13).

When $\mathbf{J}_{22}$ is singular, only the joints in $\mathbf{J}_{21}$ whose axes of rotation has components in the singular direction can help supplement the lost degree of freedom. Taking a dot product between the $\mathbf{z}$ axes of the joints in $\mathbf{J}_{21}$ with the singular direction will show all the possible factors that can supplement the lost DOF. If the dot product is zero, the axis is completely orthogonal to the singular direction and is incapable of assisting the motion in that direction. These are the only factors, that when set to zero, (together with the factor that cause $\mathbf{J}_{22}$ to be zero) would form the possible missing singular configurations in orientation singularity.

Let's take a look at the example used in Section 4.5 (Figure 4.3). To verify that the columns of $\mathbf{J}_\omega$ are the $\mathbf{z}_i$ axes of the revolute joints i, it can be seen that when $[\mathbf{J}_{21}|\mathbf{J}_{22}]$ is expressed in the wrist frame (Frame $\{5\}$), the fifth column of the lower Jacobian now contains $\{0,0,1\}^T$ (Equation 4.16), which is the $\mathbf{z}$ axis of Frame$\{5\}$, expressed in Frame $\{5\}$. $\mathbf{J}_{22}$ is singular when $sin(q_6) = 0$, which makes top row of $\mathbf{J}_{22}$ all zeros. The singular direction in this case is the rotation around X axis of Frame $\{5\}$, or $\{1,0,0\}^T$ when expressed in Frame $\{5\}$.

We would like to find the possibility that the redundancy in $\mathbf{J}_{21}$ is able to supplement the DoF lost by $\mathbf{J}_{22}$. To obtain all the possible joint contribution to this direction of motion, a dot product is taken between this singular direction and all the axes of rotation in $\mathbf{J}_{21}$. This is to say, between $\{1,0,0\}^T$ and all the columns in $\mathbf{J}_{21}$. When Jacobian is expressed in the singular frame, then the dot product results are all the elements in the same row as the singular direction of the wrist - in our example: the first row of the $[\mathbf{J}_{21}|\mathbf{J}_{22}]$.

When the resulting expressions from the dot product go to zero, there is a possibility that the Jacobian matrix loses a rank, because when the dot product of these

axes of rotation with the singular direction is zero, they are orthogonal to each other and therefore will not contribute to the motion in the singular direction.

The $[\mathbf{J}_{21}|\mathbf{J}_{22}]$ is reproduced below for convenience. In this example, the dot product between the $\mathbf{z}$ axes in $\mathbf{J}_{21}$ with the singular direction yield $c_5 s_{234}$ and s_5.

$$[{}^5\mathbf{J}_{21}|{}^5\mathbf{J}_{22}] = \begin{pmatrix} -c_5 s_{234} & s_5 & s_5 & s_5 & 0 & 0 & s_6 \\ -s_5 s_{234} & c_5 & c_5 & c_5 & 0 & 1 & 0 \\ c_{234} & 0 & 0 & 0 & 1 & 0 & c_6 \end{pmatrix} \tag{4.31}$$

This yields three possible cases:

- $s_{234} = 0$ AND $s_6 = 0$. This was substituted and found not to cause the complete Jacobian $\mathbf{J}$ to be rank deficient.

- $s_{234} = 0$ AND $s_5 = 0$. This was substituted and found not to cause the complete Jacobian $\mathbf{J}$ to be rank deficient.

- $c_5 = 0$ and $s_6 = 0$. This is also not the term that would cause $\mathbf{J}$ to be rank-deficient.

- $s_5 = 0$ AND $s_6 = 0$. This condition causes $\mathbf{J}$ to be rank-deficient. This is consistent with what we found in Section 4.5

- s_{234} AND $s_5 = 0$ AND $s_6 = 0$. This configuration would cause the $\mathbf{J}$ matrix to be rank-deficient and also the $\mathbf{J}_\omega = [\mathbf{J}_{21}|\mathbf{J}_{22}]$ to be rank-deficient.

It is therefore detected that the configuration ($s_5 = 0$ AND $s_6 = 0$) is singular, and it can not be detected by evaluating $det(\mathbf{J}_\omega) = 0$.

4.8 Summary

The method of evaluating position and orientation singularities separately from the top and bottom halves of the Jacobian matrix helped to reduce the complexity of the problem. Instead of having to solve the determinant of $6 \times n$ matrices, it is reduced to $3 \times n$ matrices. A simple check was proposed to make sure that some wrist singularities in redundant manipulators are not overlooked. The proposed check was valid only for serial manipulator with spherical arm. This is done by narrowing the possible missing singular configuration to be the components that might contribute to the singular direction in $\mathbf{J}_{21}$. The method of identifying singularities are covered and the importance of identifying singular directions, not just the singularities, is discussed. Singular directions are difficult to evaluate and express symbolically. A simple searching method is proposed by transforming the rank deficient Jacobian matrix into all the DH frames of the robot and identifying a row of zero. For the few rare cases that don't fit into this method, it is easier to identify singular direction by inspection, especially after the singular configuration has been identified.

CHAPTER 5

SINGULARITY HANDLING: BY REMOVAL OF DEGENERATE COMPONENTS

5.1 Chapter Overview

Singularity handling methods are divided into two categories. The first category consists of the methods with divisions of workspace. A singular region is declared around a singular configuration where a different control algorithm designed to handle the singularity is applied. The second category comprises the technique with a uniform algorithm throughout the entire workspace that has the capability of going through singularities in a stable manner. There is no division of workspace with different control algorithms in these methods.

The singularity handling method presented in this chapter is one of those in the first category, while Chapter 7 covers an example of a singularity handling method from the second category.

In this chapter, various existing methods are reviewed. The method proposed in this chapter involves identifying the degenerate directions of the singularity and removing the degenerate component of the task while the end-effector is inside the singular region. Null motion was utilised in assisting and creating motion in the

degenerate direction. Implementation example in a PUMA 560 manipulator is presented.

5.2 Related Works

Chapter 4 explored singularities and methods of identifying them. In this chapter, a method of singularity handling is presented. It is well-known that singularities inherently limit the manipulator's capability to complete its task. As the manipulator approaches and enters singular configuration, the inverse of the Jacobian matrix would be ill-conditioned. The resulting joint rate command for a finite motion in task space would be unbounded. That is why it is necessary to handle singularities in a stable manner. Many methods have been proposed to handle singularities. There are generally divided into two main categories: the first involves division of workspace and a different control algorithm is applied to the region around the singularities, while the second applies a uniform control strategy, that is capable of handling singularities, throughout the entire workspace.

Techniques without workspace division generally introduces a continuous modification either to the Jacobian matrix or to the trajectory of the end-effector. This continuous function is close to zero when the end-effector is further away from a singular configuration. In the vicinity of singularity, it introduces a slight error to avoid the ill-conditioned inverse to the Jacobian matrix. The methods found in [5, 53, 54, 55, 56, 57], among others are designed within this concept. In [5] and [58], damped least-squares method was used to obtain a modified Jacobian that is not singular. With higher damping variables, tracking error introduced could be quite high. Kircanski [53] utilized the Singular Value Decomposition (SVD) and replaced

the zero value at the diagonal matrix at singularity with a continuous function of non-zero value. Other approach resolved singularity by reparameterizing the desired path $\mathbf{s}(\lambda)$ so that $\boldsymbol{\theta}(\lambda)$ and its higher derivatives are well-behaved [54, 59]. The idea was extended to desingularizing the workspace in [55]. The enhanced Jacobian transpose has also been used in place of Jacobian inverse [56] to provide a stable control over the vicinity of singularity.

There are also many examples of techniques belonging to the category where the singularity robust algorithm is appplied in a sub-space defined in the vicinity of singularities. Aboaf and Paul [60] handled spherical wrist singularity by eliminating the singular direction and the contribution of the roll joint of the wrist. This approach results in a reduced (5 by 5) Jacobian which is of full rank. The velocity of the eliminated joint is then bounded at some maximum joint rate to avoid excessive joint velocity. The effect is compensated in task space. Chiaverini and Egeland [61] identified and removed the degenerate components of motion, and applied pseudo inverse with the collapsed Jacobian in the kinematics-based approach. Cheng [50] performed an analysis and handling method on PUMA 560 also by releasing the exactness in the singular direction and providing the extra redundancy to the achievable direction. Their *Compact QP method* would then attempt to minimize the tracking error in singular direction. Another approach of removing the degenerate direction in the vicinity of singularity was implemented in operational space formulation. Null space motion was utilized to assist motion in the degenerate direction [62]. A similar approach by utilising null space motion to create motion in degenerate direction is also presented in [63].

Other classes of related techniques drives the end-effector away from the singular configurations, therefore avoiding the need to handle the phenomenon [64, 26, 65, 66, 67, 38, 64]. In these techniques, redundant motion was utilized to follow a certain optimization function that maximizes the manipulability of the robot, hence avoiding singular configuration (where manipulability is zero). This can only be done with extra degree of freedoms in the manipulator. A measure of manipulability is presented in [6].

Others incorporate the handling of singularities into the design of their manipulators, for example by introducing redundancies [68], by placing singular configuration outside of usable workspace using spherical wrist [69] or offset wrist [70], [71], [72], [73]. Other novel designs forms a Jacobian that is always invertible [74]. However, we are not always given the opportunities to design the manipulator the way the wish them to be.

The idea presented in this chapter, along with ideas presented in other papers such as [61] and [50], divides the whole workspace into two sub-spaces: the singular region (in the vicinity of singularity) and the non-singular region. An issue with this approach is the switching of control algorithms causing discontinuities as the manipulator goes in and out of the singular regions. While inside the singular region, a singularity handling algorithm is applied to the manipulator. This involves making use of the identified singularities as in Chapter 4, and removing the degenerate components of motion.

5.3 Handling Singularity by Removing Degenerate Components

This section describes the singularity handling method by removal of degenerate component(s) of the task. First, a region is defined in the vicinity of each singular configuration. This is called the *singular region*. While inside the region, the Jacobian matrix $\mathbf{J}$ and the generalised forces $\mathbf{f}$ (see Equation 5.1) are transformed into the singular frame. This is a frame where the degenerate direction spans one of the rows of the Jacobian matrix. The component is then removed and the Jacobian is collapsed into a full rank matrix of less number of rows.

The method is proposed based on the operational space formulation. The equation for the generalised forces at the end-effector is reproduced below for convenience:

$$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{f} + [\mathbf{I} - \bar{\mathbf{J}}\mathbf{J}]^T\boldsymbol{\tau}_0 \tag{5.1}$$

5.3.1 The Singular Region

The singular region around the singular configuration can be specified as the region where the determinant of the Jacobian matrix is less than a certain threshold value. The determinant can be found symbolically, and each factor $s(q)$ represents one singularity in the system. A region $\mathcal{D}$ is defined around each of these factors [1]:

$$\mathcal{D} = \{\mathbf{q} \quad ||s(\mathbf{q})| \leq s_0\} \tag{5.2}$$

where $s(\mathbf{q})$ is the factor in determinant of the Jacobian matrix which is zero at specific singularity and s_0 is the threshold value that defines the singular region.

This divides the entire workspace into two subspaces: one with full Jacobian and the other with reduced Jacobian matrix (but full rank) with the degenerate components removed.

5.3.2 Removing Degenerate Components

There are usually more than one singular configurations in a system. For example, the PUMA robot that we use for case study has wrist, elbow, and head singularities.

Once the manipulator enters a singular region $\mathcal{D}$ as in (5.2), the corresponding degenerate directions are identified and removed.

The singularity handling process is done by first transforming the Jacobian matrix $\mathbf{J}$ and the operational space force $\mathbf{f}$ to a frame where the singular direction is aligned with one of the axes of the frame. We call these frames the "singular frames". Transforming the Jacobian into the singular frame can be done by using a symbolically predetermined rotational matrix from the base frame to the singular frame.

After being transformed to the singular frame, the row(s) of zeros in the Jacobian matrix which represents the singular direction, and the corresponding component in ${}^{S}\mathbf{f}$ are then removed. ${}^{S}\mathbf{f}$ is the operational space force vector expressed in the singular frame. This process collapses the Jacobian into a reduced Jacobian (with fewer rows) but of full rank.

In the singular region, the manipulator is treated as a redundant system with end-effector subspace being orthogonal to the singular direction. The end-effector task in the subspace is controlled using the operational space redundant manipulator control, while null space joint torques are used to deal with the control in the singular direction according to the singularity type (Equation 2.1)(see Section 5.3.4). Dynamically

consistent force/torque relationship guarantees decoupled operational space and null space behaviour [4].

5.3.3 Utilising the SVD

In removing the degenerate components of the task, the matrix $\mathbf{J}$ and vector $\mathbf{f}$ need to be transformed to the singular frame. These frames can be pre-defined according to the singular direction. For example, for the wrist singularity in PUMA, the singular direction is the rotation around X axis of Frame $\{4\}$. Therefore, Frame $\{4\}$ is defined as the singular frame. For head singularity of PUMA, the singular direction is the translation along the Y axis of Frame$\{1\}$. Therefore Frame $\{1\}$ is defined as singular frame. When it is identified that the manipulator is close to head singularity, for example, then the Jacobian matrix and the operational space force vector are transformed to the corresponding singular frame, i.e. ${}^{1}\mathbf{J}$ and ${}^{1}\mathbf{f}$.

Utilising the SVD, as shown in Section 3.2.2, the Jacobian matrix and the operational space force vector can be transformed to the singular frame by pre-multiplying them with matrix $\mathbf{U}^T$. For a 6DoF system, this transformation matrix $\mathbf{U}^T$ may not display any physical meaning of the singular directions, but it is still a valid transformation to the singular frame. This will reduce the required effort of identifying various rotational matrices to all the unique set of singularities in a manipulator.

5.3.4 Null Space Control

Once inside the singular region, the task is momentarily reduced to a lower number of DOFs and motion/force in the degenerate direction that has been removed is no longer possible until the manipulator exits the singular region.

If the desired task has a component along the degenerate direction, null space motions are then utilized to assist the motion. Motion in degenerate direction was studied in [75, 76, 77]. Null space task can be realized by constructing a potential function, $v_0(\mathbf{q})$, whose minimum corresponds to the goal of the desired task. The null space torque $\boldsymbol{\tau}_0$ (see Equation (5.3)) is formulated as the gradient descent of the potential function:

$$\boldsymbol{\tau}_0 = -\mathbf{A}(\mathbf{q})\nabla v_0(\mathbf{q}); \tag{5.3}$$

where $\mathbf{A}(\mathbf{q})$ is the joint space inertia matrix as in Equation 2.5. The joint space inertia matrix $\mathbf{A}$ is included here to scale the joint space null motion torque $\boldsymbol{\tau}_0$ to include the dynamics of the manipulator.

Singularities could be categorized into type 1 and type 2, based on the effects of null space torques $\boldsymbol{\tau}_0$. When in singular configuration, null space motion in type 1 singularity moves the manipulator out of singularity [3]. Null space motion in type 2 singularities affects only internal joint motion, and changes the singular directions without affecting the end-effector motion/forces. This is sometimes termed as *escapable singularities* in some literatures [76], because the singular direction can be shifted away from the desired path.

To escape a type 1 singularity, the end-effector motion in singular direction can be generated directly through the associated null space torque by a potential function to move it from a singular configuration $s(\mathbf{q}) = 0$ to the boundary of singular region, $s(\mathbf{q}) = s_0$ (as declared in 5.2). As motion in the singular direction has been removed from the task as explained in the previous sub-section, motion in the singular direction

Table 5.1: Singularity by the effect of null space torque have on it

	Type 1	Type 2
Effect	moves end-effector out of singularity	shifts the singular direction
Potential function	$v_0(\mathbf{q}) = \frac{1}{2}k_N(s(\mathbf{q}) - s_0)^2$	$v_0(\mathbf{q}) = \frac{1}{2}k_N(\boldsymbol{\zeta}(\mathbf{q}) - \boldsymbol{\zeta}(\mathbf{q})_0)^2$

is no longer a part of the task, and is therefore a null space motion. Further example is shown in the case study of the elbow singularity in PUMA.

In type 2 singularity, the null motion does not move the end-effector but shifts the singular direction. Potential function can be constructed such that its minimum occurs when the singular direction is orthogonal to the direction of motion of the desired path ($\boldsymbol{\zeta}(\mathbf{q}) = \boldsymbol{\zeta}_0$). $\boldsymbol{\zeta}(\mathbf{q})$ is the joint configuration vector and $\boldsymbol{\zeta}_0$ is the configuration vector where singular direction is orthogonal to the desired path. Examples of type 2 singularities can be found in the wrist and head singularities of PUMA.

5.4 Application on PUMA Robot

5.4.1 Removing the Degenerate Components

The idea of removing the degenerate component is to remove the row(s) of the Jacobian matrix and elements of the task space force $\mathbf{f}$ (see Equation 2.1) that represent the degenerate direction(s) of motion. By doing so, we do not consider the motion in the degenerate direction as a part of the task space anymore. To do so, the Jacobian matrix and force vector need to be expressed in the frame in which one of the axes represents the direction of singularity (degenerate direction).

Force vector in task space is obtained from the control law to represent the virtual force that "pulls" the end effector to the desired position and orientation (see

Equations 2.2-2.3). The force vector is then resolved into the correct frame, that represents the singular direction as one of its axes. The Jacobian is also resolved into the same frame. In singular configuration, instantaneous motion in joint space does not produce any instantaneous motion in task space. This manifests itself into a row of zeros in the Jacobian matrix when expressed in the singular frame.

The row of zeros (degenerate components) in the Jacobian matrix and the force vector are then removed. The resulting Jacobian has its dimension reduced from $m \times n$ to $(m-k) \times n$, where k is the number of DOFs lost by the manipulator at the singularity. The transpose of the new collapsed Jacobian and the force vector are multiplied to obtain the joint torque vector (Equation (2.1)).

For the case of PUMA 560, the degenerate direction of motion in wrist singularity, for example, is represented by the rotation around the X-axis of Frame{4}. The relationship between the generalized forces, expressed in the frame that best represent the singular direction (without the null space component), is expressed as:

$$\boldsymbol{\tau} = ({}^4\mathbf{J})^T . {}^4\mathbf{f} \tag{5.4}$$

where

$$\begin{aligned} {}^4\mathbf{J}^T &= \begin{pmatrix} {}^0\mathbf{R}_4^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^0\mathbf{R}_4^T \end{pmatrix} . {}^0\mathbf{J} \\ {}^4\mathbf{f} &= \begin{pmatrix} {}^0\mathbf{R}_4^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & {}^0\mathbf{R}_4^T \end{pmatrix} . {}^0\mathbf{J} \end{aligned} \tag{5.5}$$

${}^4\mathbf{f}$ is the force in the Cartesian axis represented in Frame $\{4\}$, obtained from the control law, with the fourth element (rotation around X-axis) removed, $\boldsymbol{\tau}$ is the torque sent out to each joint, and ${}^4\mathbf{J}$ is the Jacobian expressed in Frame $\{4\}$, with the fourth row removed. Note that tasks considered are in 6 DoF, which is why the rotational matrices above are defined as 6×6.

As for head singularity, the degenerate direction is the translation along Y axis of Frame{1}, as shown in Equation 4.18. To express the force vector and the Jacobian vector in Frame{1}:

$$\begin{aligned} {}^{1}\mathbf{J}^{T} &= {}^{0}\mathbf{R}_{1}^{T}.{}^{0}\mathbf{J} \\ {}^{1}\mathbf{f} &= {}^{0}\mathbf{R}_{1}^{T}.{}^{0}\mathbf{J} \end{aligned} \tag{5.6}$$

The first row is then removed from the Jacobian matrix and the force vector. The resulting torque is calculated by $\boldsymbol{\tau} =^{1} \mathbf{J}^{T}.{}^{1}\mathbf{f}$.

Elbow singularity is handled in the same way by expressing the Jacobian and force vector in Frame{B}. The degenerate direction in this case is the translation along X axis of Frame $\{B\}$, as shown in Equation 4.19. The Jacobian and the force vector are expressed as:

$$\begin{aligned} {}^{B}\mathbf{J}^{T} &= {}^{0}\mathbf{R}_{B}^{T}.{}^{0}\mathbf{J} \\ {}^{B}\mathbf{f} &= {}^{0}\mathbf{R}_{B}^{T}.{}^{0}\mathbf{J} \end{aligned} \tag{5.7}$$

5.4.2 The Use of Singular Value Decomposition

As explain in Section 5.3.3, SVD can be utilised to reduce the effort of identifying each rotational matrices. Section 5.4.1 shows the need to identify ${}^{4}\mathbf{R}_{0}$, ${}^{B}\mathbf{R}_{0}$, and ${}^{1}\mathbf{R}_{0}$ to transform the Jacobian matrix and the operational space force vector to the singular frames for wrist, elbow and head singularities. This can be replaced with just matrix $\mathbf{U}^{T}$, such that:

$$\begin{aligned} {}^{0}\mathbf{J} &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{T} \\ \mathbf{U}^{T}{}^{0}\mathbf{J} &= {}^{S}\mathbf{J} = \boldsymbol{\Sigma}\mathbf{V}^{T} \end{aligned} \tag{5.8}$$

When applied to a 6 DoF system such as PUMA, the rotational matrix loses its physical meaning, however, it is still a valid transformation to the singular frame.

The application of null space motion in singularity handling, however, requires the knowledge of which singularity the manipulator is in, as treatment is case specific. In this case, the use of the $\mathbf{U}^T$ matrix of SVD is limited to reducing the effort of identifying the rotational matrices to each singular configuration. It is still necessary to know which case of singularity the manipulator is in (Equation 4.3).

5.4.3 The Use of Null Motion

Motion from a singular configuration can be divided into of feasible and non-feasible directions. Non-feasible direction is one that requires motion in the singular (degenerate) direction.

In this method, as explained above, the motion in the degenerate direction has been disabled through the removal of appropriate elements of the Jacobian matrix and task space forces.

A path is non-feasible when the desired trajectory lies along the degenerate direction of the manipulator. In our experiment, the null space motion was then utilised in handling such motion. Null space torque can then be generated to reconfigure the manipulator to move in the non-feasible direction. Different functions can be used to determine the null space torque.

Type 1 Singularity

A type 1 singularity is when null space motion moves robot out of singular configuration. This can be illustrated by a two-link planar manipulator, such as shown in Figure 5.1. In the configuration shown in Figure 5.1, the manipulator loses the ability to translate in the X_0 direction. This component is then removed from the

task, resulting in a 1×2 Jacobian matrix that maps the joint velocities into motion in Y_0.

Suppose it is desired to move the end-effector in the X_0 direction. Null motion is generated by exciting motion in either joint 1 or 2. When this happens, motion at the end-effector is generated. However, translation in Y_0 direction is still under motion control, while the translation in X_0 is free. The null motion is then translated into a motion that moves the end-effector out of singularity (Figure 5.2.)

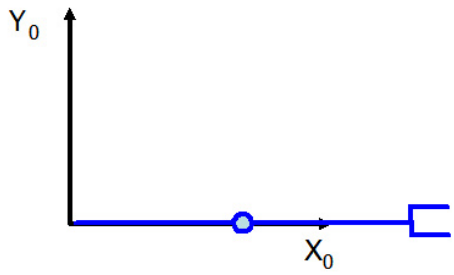


Figure 5.1: A two-link planar manipulator is a type 1 singular configuration.

The elbow lock in PUMA is an example of a type 1 singularity. As explained, the associated null space torque will create end-effector motion in the singular direction. The singular direction, as shown in Chapter 4, is the translation along the line that connects the wrist point to the origin of Frame{1}

This means, for the case of PUMA, null space motion of joint 3 would generate motion in the singular direction (see Figure 4.5 for singular direction).

In Table 5.1, it is shown that when we would like to move from one configuration to another within the singular region, the potential function is given as

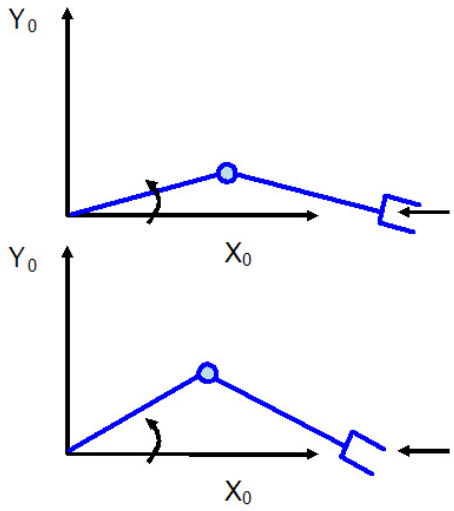


Figure 5.2: The two link manipulator from the previous example, moving out of singularity, due to the application of null space motion.

$v(\mathbf{q}) = \frac{1}{2}k_N(s(\mathbf{q}) - s_d)^2$ where $s(\mathbf{q})$ is the factor of the determinant of the Jacobian matrix associated with the singular configuration of interest. The value s_d is the value of the determinant term that we would like the end-effector to move to. If we wish to move the end-effector out of the singular region, then it is desired to drive the end-effector to the boundary of this region. In this case, the value s_d is set as the threshold value that divides singular and non-singular regions, s_0.

For elbow singularity, $s(\mathbf{q}) = d_4c_3 - a_3s_3$ from Equation 4.3. To decide on the value of s_0, Figure 5.4 was utilised. This is the plot of $1/s(\mathbf{q})$ vs q_3. The unstable motion around singularity is caused by the unbounded inverse of the Jacobian matrix, which is then used to calculate the joint rate command. The inverse of matrix $\mathbf{J}$ is proportional to $\frac{1}{det(\mathbf{J})}.Adjoint(\mathbf{J})$. The unbounded inverse can be thought of as the result of division by zero, when the determinant of $\mathbf{J}$ approaches zero at singularity.

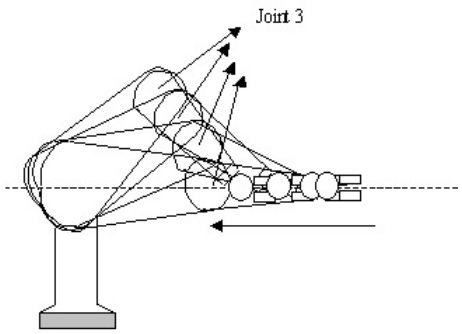


Figure 5.3: shows Puma-like manipulator moving out of elbow (and wrist) singularity, following the path which lies in the degenerate direction

The plot of $1/s_0$ (Figure 5.4 can therefore be used to approximate the threshold value. The plot shows that for most part away from singularity, the plot was almost linear. The value $1/s(\mathbf{q})$ starts to deviate significantly and approach $\pm\infty$ at around 88^o and 96^o. This is about 4-5 degrees away from the exact singularity point of $q_3 = 92.69^o$. It is necessary to verify this threshold value experimentally. The value of s_0 was therefore be approximated at ± 0.03.

So an example of potential function designed to move the end-effector from the elbow singularity ($s(\mathbf{q}) = 0$) to the boundary of the singular region would be:

$$v(\mathbf{q}) = \frac{1}{2} k_N ((d_4 c_3 - a_3 s_3) - (0.03))^2 \tag{5.9}$$

Setting $s_0 = 0.03$ would cause the robot to escape the singularity by assuming an elbow down position, while setting $s(\mathbf{q}) = -0.03$ would cause it to try to assume an elbow up position. Singularity are often considered as the intersection point between different motion manifolds. In this case, it is shown as the point where elbow up and elbow down position meet [38].

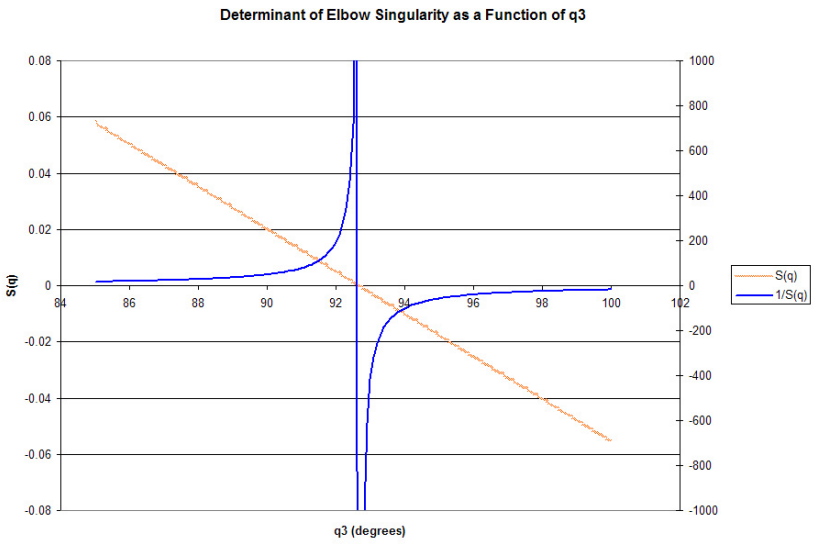


Figure 5.4: The value of $s(\mathbf{q}) = d_4c_3 - a_3s_3$ as a function of q_3 and $1/s(\mathbf{q})$.

The resulting null space torque $\boldsymbol{\tau}_0$ is:

$$\boldsymbol{\tau}_0 = -\nabla v(\mathbf{q}) = [0, 0, k_N[(-d_4s_3 - a_3c_3) - 0.03], 0, 0, 0]^T \tag{5.10}$$

Type 2 Singularity

An example of type 2 singularity in PUMA is when the wrist joint is straightened ($q_5 = 0$). The non-feasible path in this case is when the desired path contains the degenerate direction, *i.e.* for the end-effector to turn around the X-axis of Frame{4} (see Figure 5.5). As discussed in Section 4.5, PUMA wrist singularity is a special case whereby there is singular frame with physical meaning that the Jacobian matrix can be expressed on that would show a row of zeros. This is because it is a *coupled singularity* between position and orientation singularity.

The other one is the head lock, where the wrist point lies along the z axis of the base frame. The degenerate direction is the Y-axis of Frame{1}.

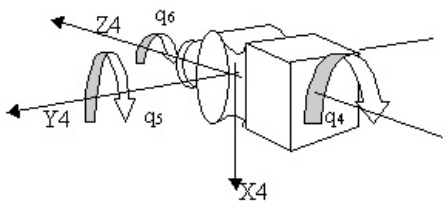


Figure 5.5: The wrist of PUMA 560, showing $Frame\{4\}$

It is described that, as singularity occurs, at least two frames of the manipulator would line up. One of these two frames can therefore be used to create a null space motion to escape the singular position.

In the case of PUMA's wrist, null space torque was used to move joint 4 (see Figure 5.6). This would then shift the plane that contains the non-feasible-path (the YZ plane) out of the way of the desired path (Figure 5).

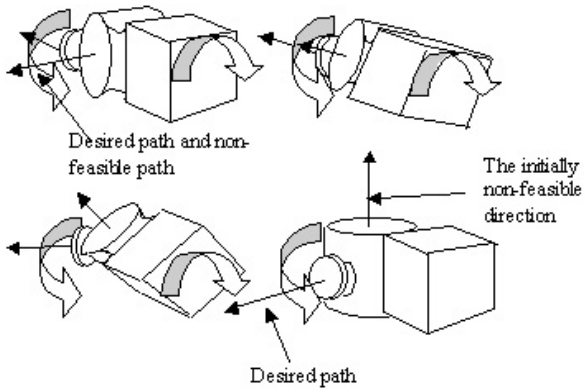


Figure 5.6: Null space torque is used to turn joint 4, so that the YZ plane of $Frame\{4\}$ is shifted out of the way of the desired trajectory.

According to Table 5.1, the potential function for this case is $v(\mathbf{q}) = \frac{1}{2}k_N(\zeta(\mathbf{q}) - \zeta_0)^2$. The joint configuration is expressed as ζ and the desired configuration $\boldsymbol{\zeta}_0$ is one where the singular direction is orthogonal to the desired path.

Supposed the manipulator is currently in wrist singularity and that the desired motion is aligned with the degenerate direction, which is rotation around X axis of Frame{4}. Then to create an internal joint motion in joint 4 to shift the degenerate direction away from the desired path, the potential function can be designed as:

$$v(\mathbf{q}) = \frac{1}{2}k_N(q_4 - (q_{4S} + 90^o))^2 \tag{5.11}$$

which rotate joint 4 (and the singular direction) 90^o away from the desired path, in the positive Z_4 direction. The variable q_{4S} is the value of q_4 at the start of the motion when the desired path is aligned with singular direction.

As motion in the singular direction is disabled, any motion with a component in the singular direction will experience difficulty in tracking the desired trajectory. It is therefore necessary in type 2 singularity to shift the singular direction so that it is completely orthogonal to the direction of the desired motion.

$$\boldsymbol{\tau}_0 = -\nabla v(\mathbf{q}) = [0, 0, 0, k_N(q_4 - (q_{4S} + 90^o)), 0, 0]^T \tag{5.12}$$

Similarly, in the case of PUMA head lock, a null space motion of joint 1 is required to shift the manipulator to go into the initially non-feasible path.

5.5 Implementation Result

The results of the experiments are shown in Figures 5.7 to 5.11.

Figures 5.7 show the position and orientation error in end-effector tracking, in non-singular motion. Figure 5.8, shows the same set of result drawn in the same

scale, of the end-effector in a trajectory through a singular, but feasible path. This was implemented by using the rotational matrix ${}^0\mathbf{R}_4$ as transformation matrix to the singular frame. The matrix $\mathbf{U^T}$ is utilised as the transformation matrix to the singular frame and the result is shown in Figure 5.9. It is shown that the transformation matrix $\mathbf{U}^T$ is not only a valid transformation to the singular frame, but also produces the same effects as those produced by transformation matrix with physical meaning. The tracking error produced by the two methods (Figure 5.8 and Figure 5.9) are of the same profile. The difference is minimal and is mainly due to variations between different runs that is inherent in the running of the robot. Utilising the $\mathbf{U}$ from the SVD of Jacobian matrix $\mathbf{J}$ as a transformation matrix to the singular frame is however, very much simpler and does not require the expressions of various rotational matrices to be included in the program code, thus, reducing the clutters. Figure 5.10 is tracking error of the end-effector while escaping the wrist lock (type 2 singularity) into a non-feasible path, and Figure 5.11 of the end-effector moving from an elbow singularity into a non-feasible path. All the graphs are drawn to the same Y axis scale. The only significantly larger error encountered is in the orientation error of the manipulator as it escapes from a wrist lock into a path that lies in degenerate direction. It is because null space motion was required to turn joint 4 before the end-effector was able to trace the desired trajectory, hence the large initial orientation error (see Figure 5.6).

There is an issue of discontinuity as the end-effector crosses between singular and non-singular region. The result in Figure 5.11 was obtained by tuning the gains in the control to provide a smooth trajectory. To obtain the same performance of smooth trajectory, the null motion gains need to be re-tuned for a different trajectory.

To obtain a better performance, discontinuity between the two regions needs to be properly addressed. This is done in Chapter 6.

The expression for orientation error presented in the graphs can be found in Section 2.7.

The videos of some of the experiments can be found in:
http://guppy.mpe.nus.edu.sg/~mpeangh/iser00/singularity-1-iser00.mpg

5.6 Summary

In this experiment, singularity handling in torque-controlled manipulator, based on Operation Space Formulation was explored. It was shown that by removing the degenerate component(s) of motion, control of the manipulator through singular configuration was possible, with tracking error no larger than that of a motion through non-singular path. As degenerate degrees-of-freedom in the task are removed in the singular region, the manipulator is redundant with respect to the task. Motion in the degenerate direction was made possible by motion of redundant joint to move the degenerate direction away from the desired path for type 2 singularities. For type 1 singularity, motion generated in the null space moves the end-effector out of singular configurations. A certain trade off between exactness and achievability was necessary in moving out of a singular configuration into a non-feasible path.

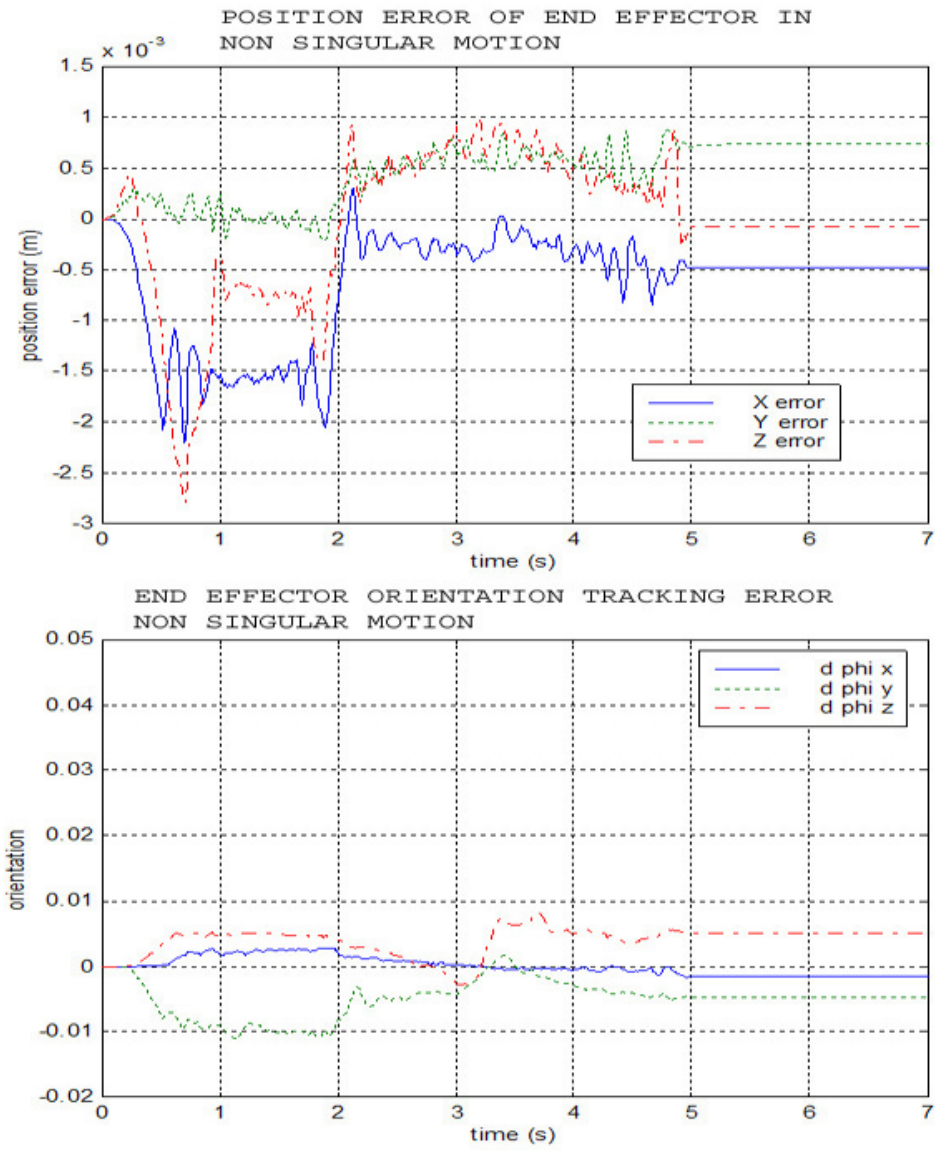


Figure 5.7: Trajectory tracking error of end-effector position as it traces a desired path that is not singular.

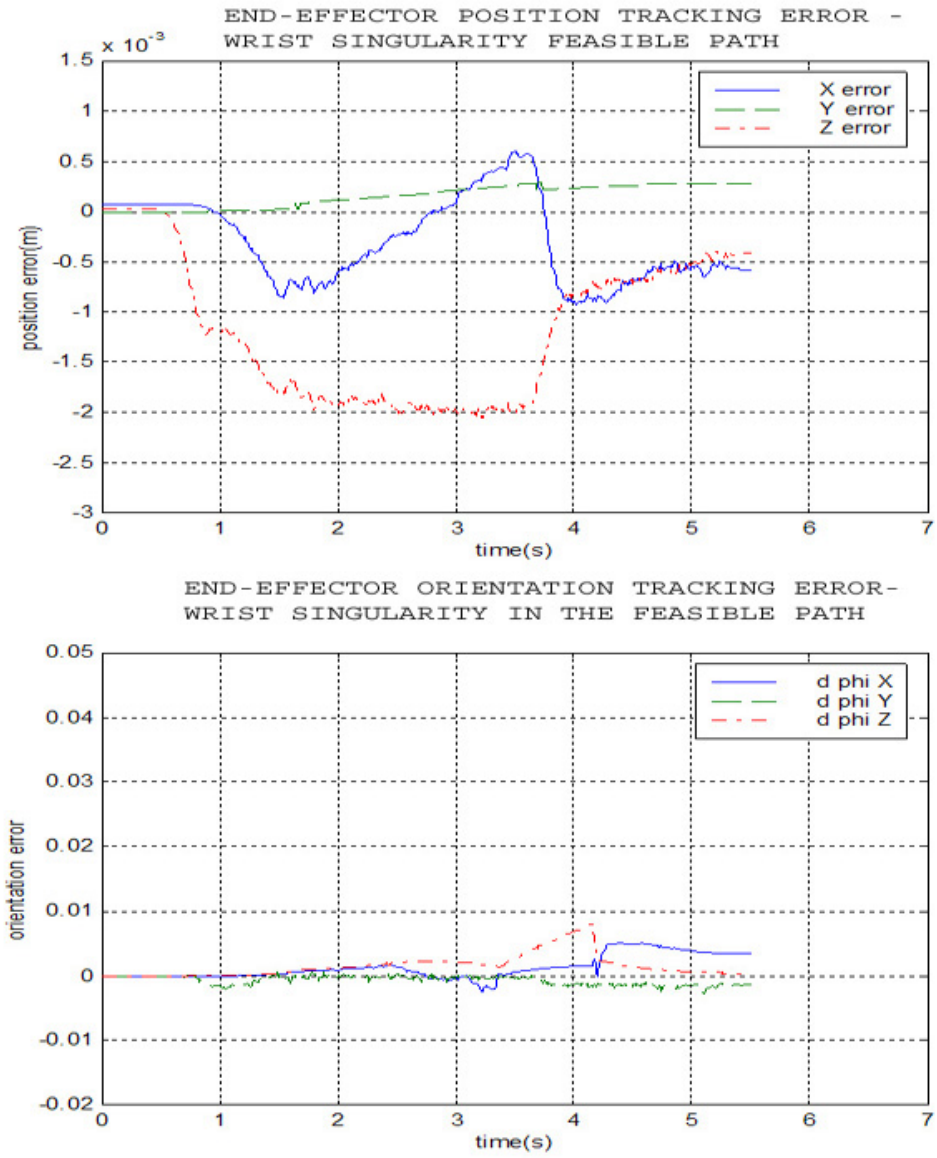


Figure 5.8: Tracking tracking error as the end-effector moved in feasible path through wrist singularity. Here rotation matrix ${}^4\mathbf{R}_0$ is used as transformation matrix to the singular frame.

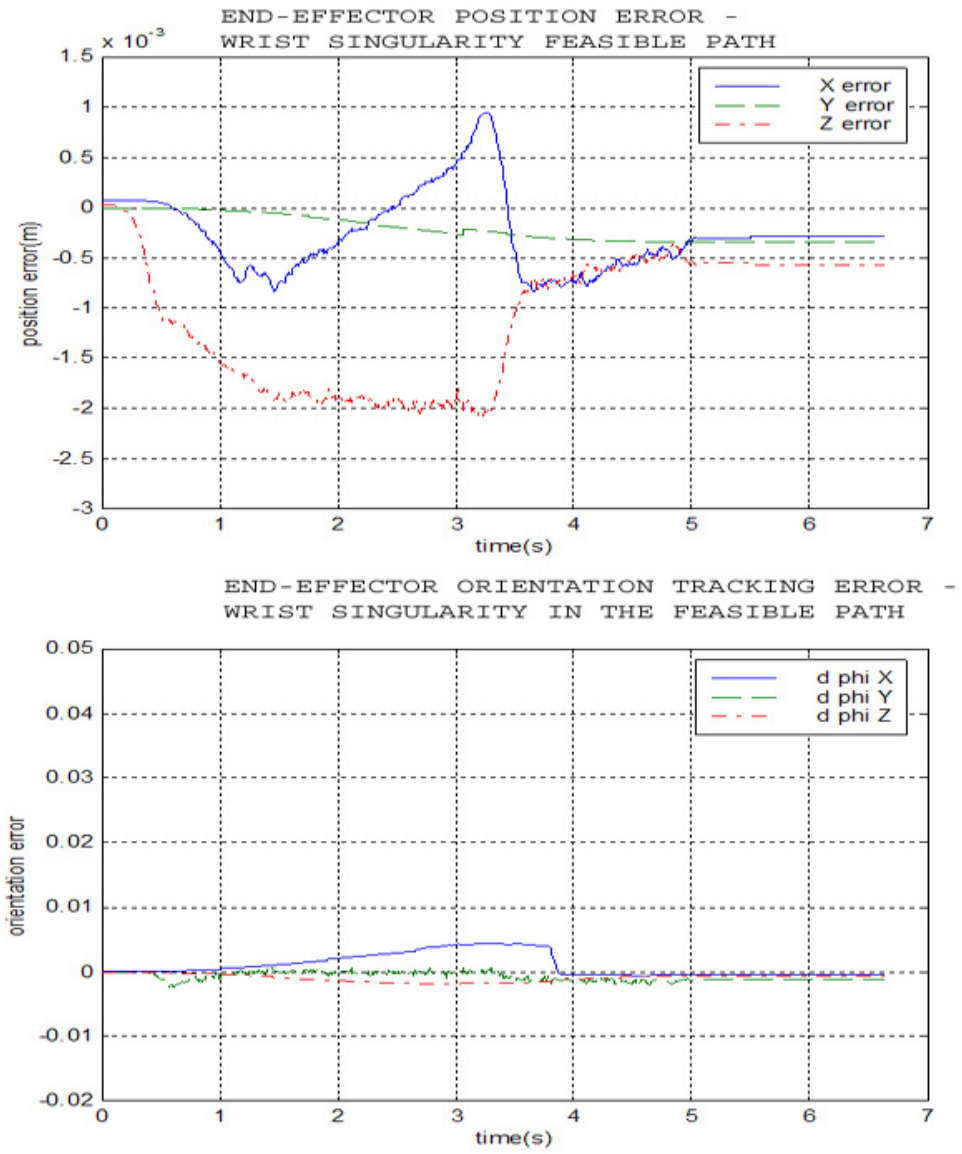


Figure 5.9: Tracking tracking error as the end-effector moved in feasible path through wrist singularity. This time, the matrix $\mathbf{U}^T$ is used as transformation matrix to the singular frame.

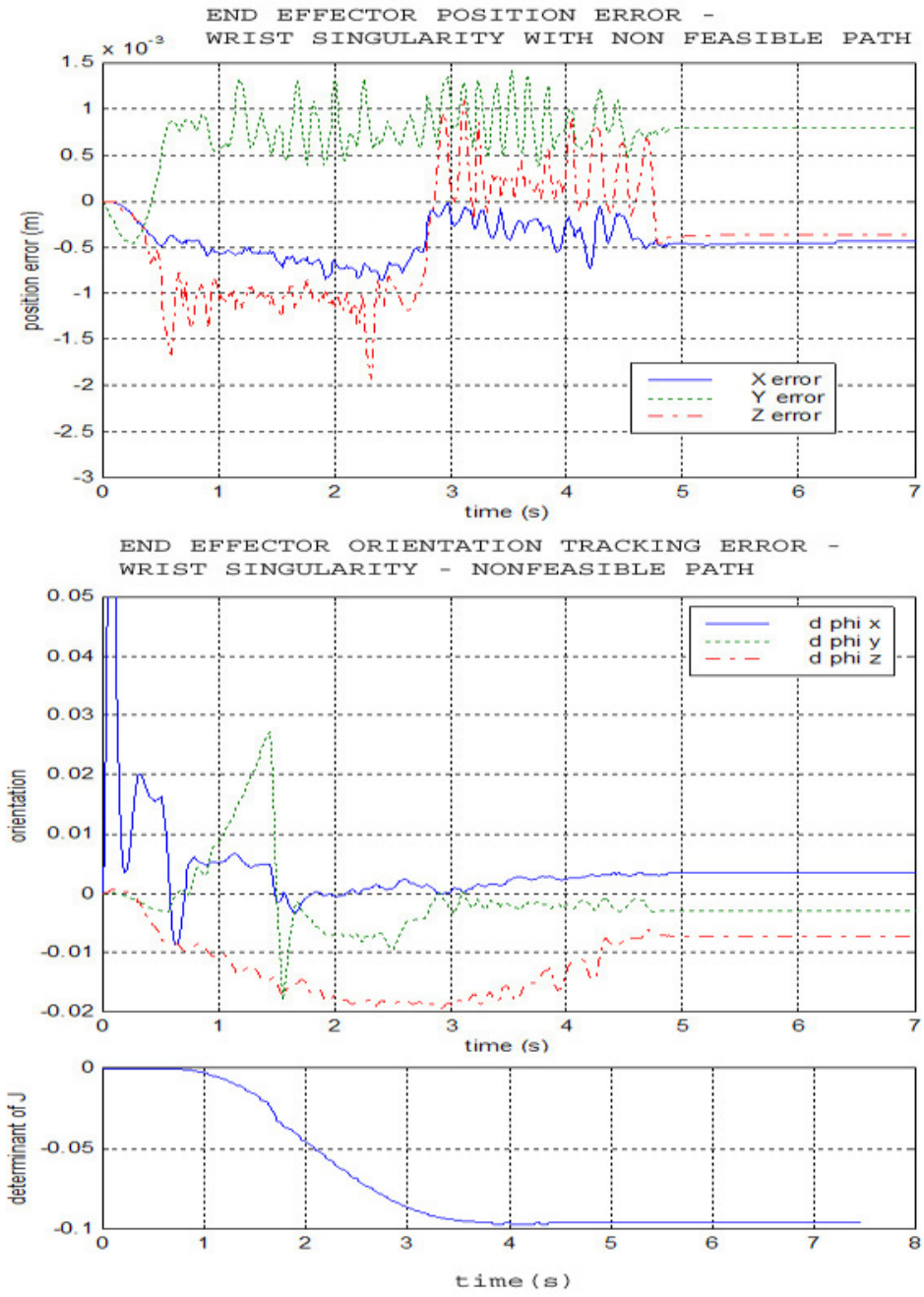


Figure 5.10: Trajectory tracking error of end-effector as it goes from stationary position within singular region (in wrist singularity) into the degenerate direction

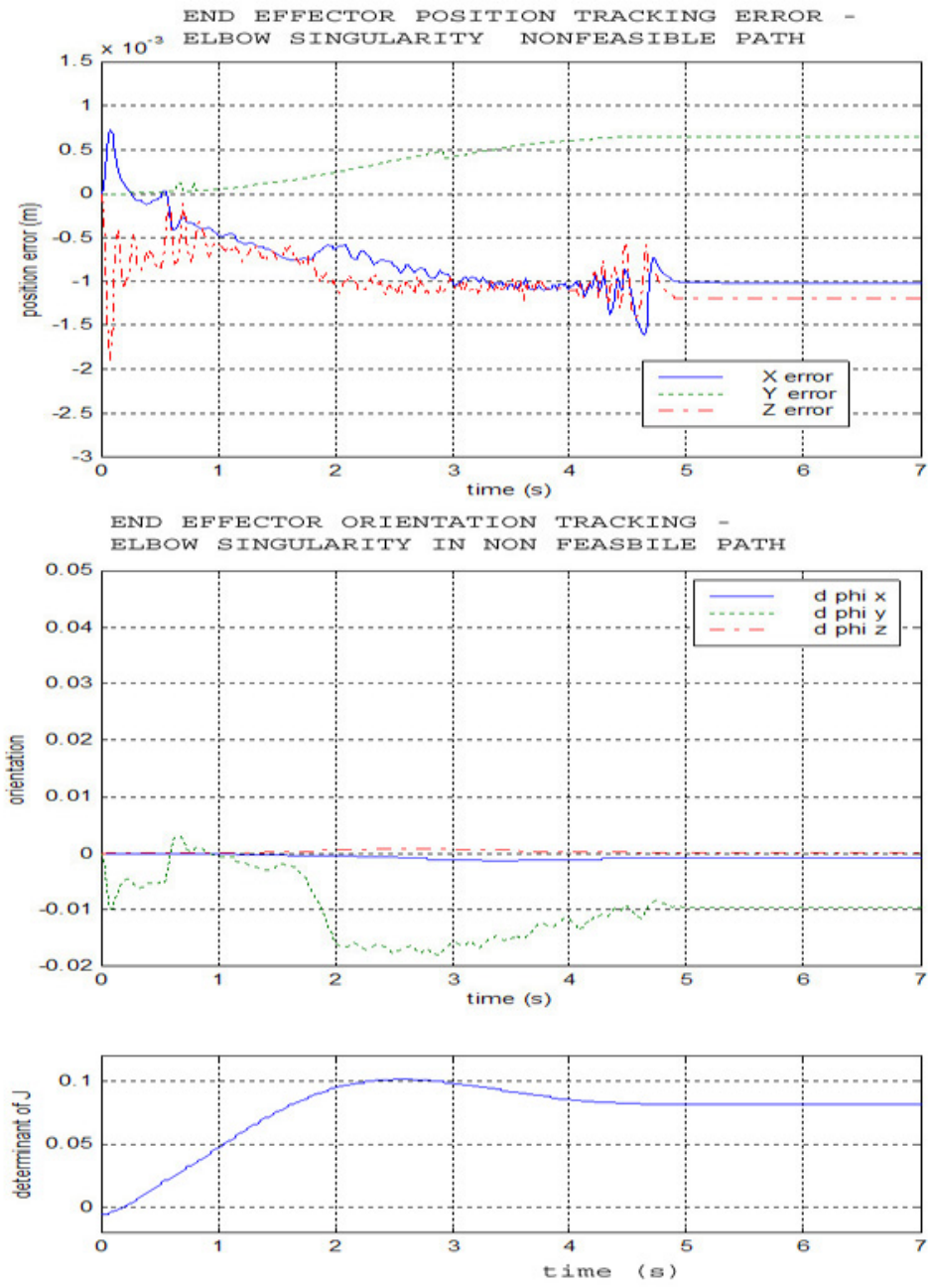


Figure 5.11: Trajectory tracking error of end-effector as it goes from stationary position within singular region (elbow singularity) into the degenerate direction

CHAPTER 6

THE REDUCED DOF WITHIN SINGULAR REGION AND DISCONTINUITY ISSUES ACROSS THE BOUNDARY

6.1 Chapter Overview

There have been many singularity handling techniques proposed in the past. Many of these techniques, such as the one presented in Chapter 5, divide the manipulator workspace into singular and non-singular regions. A singular region is defined in the vicinity of a singular configuration, within which, a different control algorithm is applied to handle the singularity. Generally, it involves removing the degenerate component(s) of motion. This results in discontinuity in the motion of the end-effector as it moves in and out of the singular region. This chapter discusses the motion of the end-effector inside the singular region and the disturbances in motion as it crosses the boundary of the singular region. Types of disturbance, the causes, and proposed solutions are presented in this chapter. The techniques were tested on PUMA 560 robot.

6.2 Introduction

Many approaches have been suggested in the past in handling the singular configuration of a manipulator. In the interest of this chapter, there are mainly two types of singularity handling methods: ones that introduces workspace division and ones that do not.

As discussed in the introduction to Chapter 5, many examples have been published on both categories of techniques. This chapter deals with the issues related to the second category of techniques, where a division of workspace is defined and different control algorithm is applied within this region to handle the singular configuration. The example of techniques in this category include [60, 61, 50, 62, 63], among many others.

The one common thing in these techniques is that within the singular region, the element that causes the singularity is removed from the control algorithm. This is commonly done by removing the component of motion in the degenerate (singular) direction. To be specific, the singular direction is identified and the Jacobian matrix is transformed onto a frame where one of the axes of this frame is aligned with the singular direction. This frame is termed the *singular frame*. The row of the Jacobian matrix in this frame will be zero at the exact singular configuration. Within the singular region, the manipulator is assumed as singular and the row is removed. The Jacobian is collapsed into a full-rank matrix with less number of rows.

In this chapter, we look into practical issues of the end-effector motion which result from the removal of the degenerate direction. This motion is removed not just at the exact point of singularity, but throughout the period the end-effector spends

inside the singular region. As the result, error would occur and accumulate in the singular direction while tracking the desired trajectory.

6.3 Effects of Removal of Singular Direction

This section will discuss the effect of the removal of the singular direction in the singular region on both type 1 and type 2 singularities [1]. Type 1 and 2 singularities are as discussed in Chapter 5, which is the division of singularities based on the effect of associated null motion on the motion of the end-effector near the singularities.

Projection into null motion is utilised in our experiment to minimise errors and disturbance by creating some 'component of motion' in the lost degree-of-freedom. The following subsections covers the motion of the end-effector around the singular configuration.

6.3.1 Upon entry into the singular region

Upon entry into the singular region, the control in the singular direction is removed. Since the end-effector is still in the correct path according to the desired trajectory (no accumulated error due to drifting away from the desired trajectory), removing the motion control in the degenerate direction causes no major disturbance to the end-effector motion. Other cause of discontinuity is the torque discontinuity at the boundary. As explained above, there is a sudden removal of the contribution of one of the degrees of freedom, which is not fully zero yet at the boundary of singular region. There is also an addition of the torque component to generate the null motion. When implemented whether on simulation or on the real robot, the effect of crossing the boundary upon entry into the singular region is hardly noticeable. However, as robots get lighter and smaller, it is necessary to handle such cases. Lighter

and smaller robots are designed for future use, such as to address the issue of human safety [78] and micro-scale robots [79].

6.3.2 Motion in singular region

Singularity happens only at an exact point in the workspace. However, behaviour of the manipulator control in the vicinity of the singularity is not stable. Therefore singularity is handled within a region declared around the exact singular configuration. This introduces some approximation and therefore some discrepancies as the end-effector will not be able to perform the task along the singular direction in the singular region. Motion is only available in the direction perpendicular to the singular direction (see Figure 6.1). Tracking error in the singular direction builds up within the singular region.

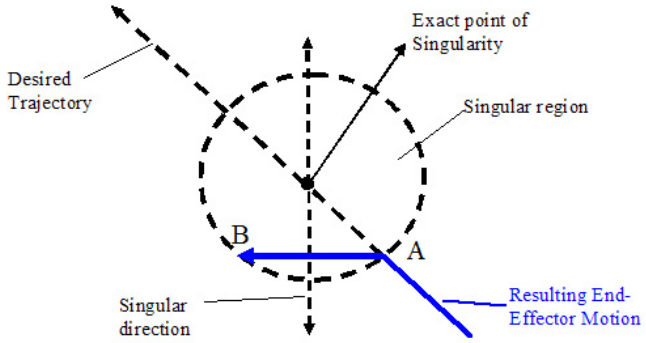


Figure 6.1: As the end-effector reaches point A at the boundary of singular region, it can no longer tracks the task in the singular direction. It can only perform motion in the directions perpendicular to the singular direction.

Type 1 singularity

In type 1 singularity, null motion creates motion that causes the end-effector to exit the singular configuration. Since a small end-effector motion in the singular direction can cause a large internal joint motion in the null space, following a time-dependent trajectory is difficult. Also, the error between the desired and actual trajectory in task space is usually too little to generate any motion in the control law, which makes the tuning of the gain difficult. Another way to put it is that the manipulability of the end-effector is low near the singular configuration [6].

Type 2 singularity

For type 2 singularity, null space motion shifts the singular direction. We therefore shift the singular direction to be exactly orthogonal to the desired trajectory while inside the singular region.

6.3.3 Exiting the Singular Region

Upon exiting the singular region, the control algorithm switches back to its full degrees-of-freedom. This is where the error accumulated inside the singular region, if not handled, manifests itself into disturbance or jerkiness in the motion.

Type 1 singularity

If the end-effector exits the singular region at the same point ($x, \dot{x}, \ddot{x}$, and t) as the desired trajectory ($x_d, \dot{x}_d, \ddot{x}_d$, and t_d), then the motion will be continuous. However, as mentioned earlier, following a time-dependent trajectory near singularity is difficult. This is a problem for type 1 singularity because the end-effector exits the singular

region according to a potential function, and hence the exit motion is not controllable according to the desired trajectory.

Motion in singular region is generated by the potential function of that drives the null motion, which does not contain any time element.

To illustrate the problem better, the following example is used. The elbow singularity of PUMA 560 is a good example. When the elbow is straightened, the singular direction is the motion of retracting the elbow from the straight posture. This motion is not controllable by a time-dependent trajectory. As the result, the following cases could happen (see Figure 6.2).

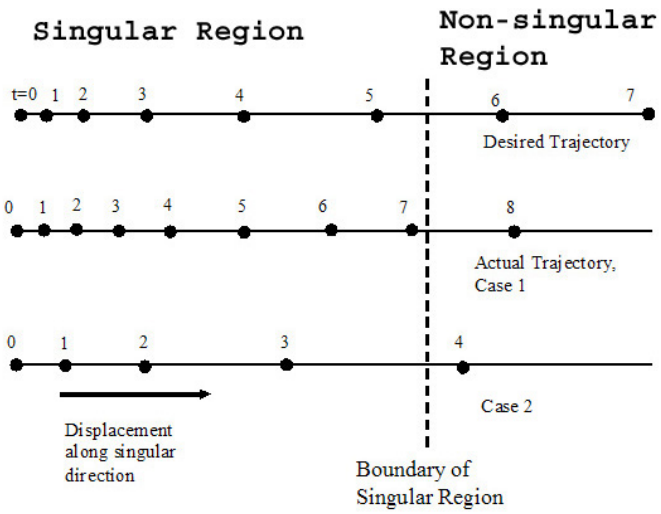


Figure 6.2: Case 1 is when the end-effector exits the singular region slower than the desired trajectory. As it exits, it has to catch up with the desired trajectory. Case 2 is when the end-effector exits the singular region ahead of the desired trajectory. Note that the points are separated by the same time duration which is the sampling period.

In case 1 of Figure 6.2, the null motion drove the end-effector to move slower than the desired trajectory while inside the singular region. As it exits the singular region (at $t = 8$ in the example in Figure 6.2), the desired trajectory is far ahead. As the full degrees of freedom is returned to the control algorithm, the large error between the desired and actual end-effector position causes a jump in the torque being sent to the manipulator.

In case 2, the end-effector exits the singular region ahead of the desired trajectory. As the control algorithm returns to full degrees of freedom (at t=4 in the example in Figure 6.2) the desired end-effector position is still inside the singular region. The motion control then pushes the end-effector back into the singular region, where the null motion would try to push it out again. This would cause a rattling at the boundary of singular region.

As seen in the explanation for Figure 6.2, it is easier to handle case 1 than case 2, *i.e.* it is easier to handle the case where the end-effector moves too slow compared to the desired trajectory in the singular region. A simple damping technique can be used to prevent the end-effector from sudden large acceleration due to large accumulated tracking error. To dictate the condition to be always of case 1, then the end-effector has to move at equal or slower speed than the desired trajectory inside the singular region. However, this proved not to be a good method, because it is difficult to implement control in joint space based on any constraint in task space while inside the singular region.

The suggested solutions to the problem is to plan a new trajectory. In this method, a new trajectory is planned when the end-effector exits the singular region. This

method assumes negligible volume for the singular region and takes the point immediately outside of the singular boundary as the starting of the new task space trajectory. The current position and velocity of the end-effector at that point are taken as the initial conditions. The final desired goal is kept the same as that of the original trajectory. The desired length of time for the newly planned trajectory is the desired length of time of the original trajectory minus the time spent inside the singular region. This method produces the smoothest result of all methods so far, with minimum effort.

Type 2 singularity

While type 1 singularities have problem with the time that the end-effector reaches the boundary of the singular region upon exit, type 2 has a problem of exiting at the wrong position from the desired trajectory (if not handled). Figure 6.3 shows the end-effector entering the singular region at point A. The singular direction is shown and the end-effector is not in motion control in this direction. It therefore still follows the desired trajectory only in the direction perpendicular to the singular direction. It exits the singular reigon at point B, while at this point, the desired trajectory is at point D. The motion control in returned to the full 6 DoF control and the accumulated error in the singular direction pushes the end-effector back into the singular region (towards D) where again the motion control in the singular direction is removed. This happens until it reaches point C, where motion control in full 6 DoF manages to push the end-effector directly towards the desired trajectory.

The solution is to utilise the null motion to change the configuration of the manipulator so that the singular direction is shifted orthogonal to the desired motion vector. When this is implemented, the end-effector will exit the singular region not

in singular direction, but in the component of motion that is under motion control. Motion is therefore continuous.

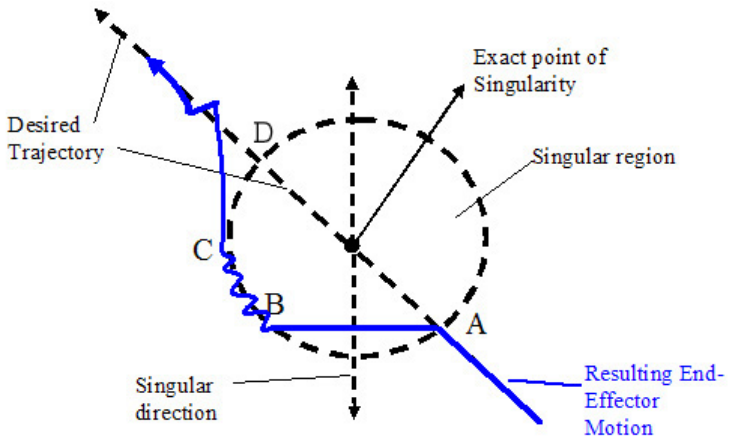


Figure 6.3: As motion is disabled in the singular direction, the end-effector will move only perpendicular to singular direction. Error accumulated inside the singular region in the singular direction causes the end-effector exit the singular region not according to the desired trajectory.

6.4 Implementation on PUMA560

The singularities of PUMA manipulators, such as shown in Equation 4.2, are identified as:

Wrist lock when the wrist is straightened, $q_5 = 0$. The singular direction is the rotation around X axis of Frame $\{4\}$.

Elbow lock when the elbow is straightened, $q_3 = 92.69^o$

Head lock when the wrist point is immediately above the Z axis of joint 1, or when $(d_4 s_{23} + a_2 c_2 + a_3 c_{23}) = 0$.

6.4.1 Wrist Singularity

This is an example of a type 2 singularity. An example of the singularity handling for the PUMA wrist is as follows:

- Upon entry, the Jacobian and the task space command vector ($\dot{\mathbf{x}}$ in velocity or kinematic control, or $\mathbf{f}$ in torque control) are transformed to the singular frame, in this case, Frame {4} (see Figure 6.4). The singular direction is the rotation around X axis of Frame {4}, so the fourth row of the Jacobian matrix and the associated element in the command vector is removed. The rest of the calculation is performed using the reduced Jacobian.

- The null motion is implemented to shift the singular direction to be perpendicular to the desired trajectory.

- The end-effector exits the region perpendicular to the singular direction. It's exit trajectory is in motion control according to the desired trajectory, therefore jerking is minimum. A simple damping technique can be implemented to eliminate any small disturbances caused by small numerical error in computation.

When implemented on PUMA, exiting the singular region without the assistance of the null motion to shift the singular direction would create jerkiness due to accumulated tracking error in the singular direction while inside the singular region (as shown in Figure 6.3). The result of the experimental run is shown in Figure 6.5. In this experiment, the PUMA robot was made to move from point ${}^{B}(x, y, z) = (0.681, 0.149, 0.013)$ to ${}^{B}(x, y, z) = (0.681, 0.150, 0.50)$. This defines a trajectory that moves vertical upwards (in positive Z direction), with a very slight motion in Y direction while passing through the singular region. The slight motion

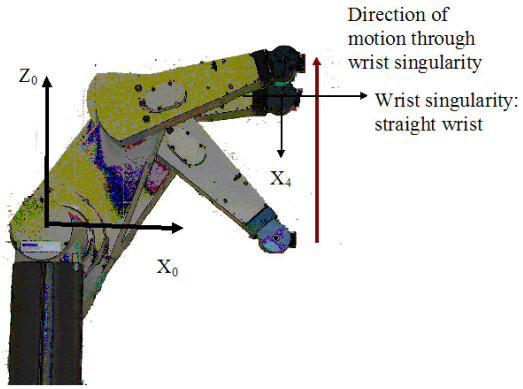


Figure 6.4: Motion of the end-effector through the wrist singularity. The singular direction: rotation around axis X_4 (shown) coupled with translation along Y_4.

in the Y direction is intentional. This is because it is difficult to obtain a trajectory that would excite a slight rotational motion around the X axis of Frame {4} within the singular region of the wrist singularity. Since the singular direction is the rotation around X axis of Frame {4} is coupled with the translation along Y axis of Frame {4} (in this specific case), it was decided to induce a slight motion in Y to create the same effect. The experimental motion of end-effector is shown in Figure 6.4. The slight motion along Y direction can not be clearly shown in the figure.

Figure 6.5 (top) shows the end-effector position tracking error. The translation in Y direction and the singular direction of rotation around X axis of Frame {4} are the dependant rows in the Jacobian matrix (in this particular example). This shows in the drift in the motion control of Y axis with respect to Base Frame while inside the singular region. As it exits the region, it snaps back into the desired trajectory, causing a sudden disturbance to the motion.

The result shows that it does not take a large motion in the singular direction to create such disturbance. In this experiment, motion in Y direction was only for

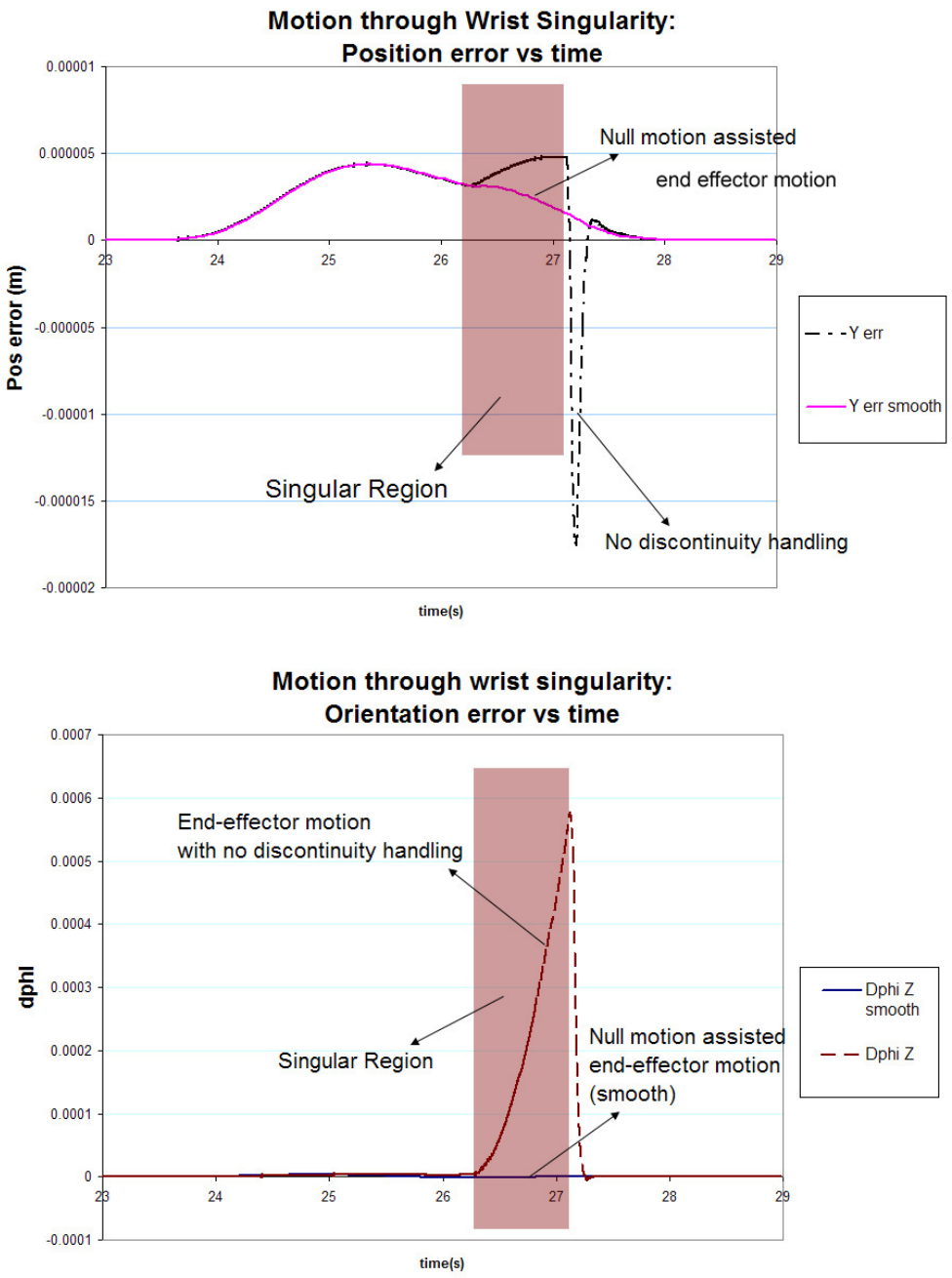


Figure 6.5: The discontinuity at the boundary of PUMA wrist singular region. Top figure shows the end-effector position error with respect to time. It is shown that motion in Y direction loses its control inside the singular region because it is coupled with the singular direction. Upon exit, it jerks back into its desired position. Center graph shows the end-effector orientation error, which also shows the drift from desired orientation in the singular direction, which snaps back in place upon exit. A smoother curve is shown as the result of the handling strategy.

a total displacement of 1 mm. In a real application, this is a reasonable amount of tracking error. This illustrates the significance of handling the motion inside singular region properly.

The middle graph shows the end-effector orientation tracking error. In the graph, the result is shown with respect to the Base Frame. The singular direction is therefore reflected as the rotation around Z axis. It is shown that the motion in this direction drifts from the desired trajectory and only snaps back into the desired trajectory after full 6 DOF motion control is returned to the control algorithm, which is when the end-effector exits the singular region.

Both graphs also show the effect of null motion in shifting the singular direction to be orthogonal to the desired direction of motion. When this is done properly, the discontinuity is reduced significantly and the resulting tracking motion is smooth.

6.4.2 Elbow Singularity

For a type 1 singularity such as the elbow lock in PUMA 560:

- Upon entry, the null motion is implemented to create motion to move the end-effector out of the singular region.
- The end-effector exits the singular region in the singular direction. Discontinuity was handled to obtain a smooth motion.

Figure 6.6 shows the experimental setup. The singular direction is the motion of retracting the straightened arm (elbow). Null motion is applied either joint 2 or joint 3 to assist the motion in the singular direction.

In case 1, (such as introduced in Figure 6.2) the end-effector moves slower than the desired trajectory when inside the singular region. A built up of tracking error in the

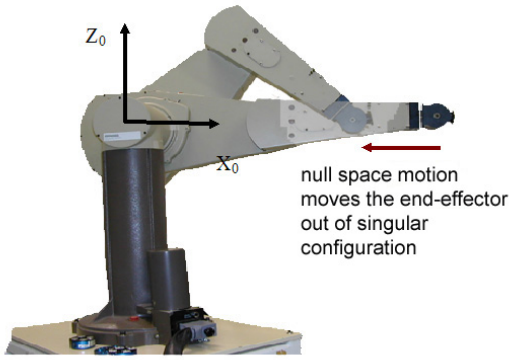


Figure 6.6: The trajectory in the experiment where the null space motion is utilised to assist the motion of retracting the straightened arm in the singular direction.

singular direction due to lag causes a sudden jump as the end-effector regains its full 6 DoF motion control. As expected theoretically, this effect was easily reduced with techniques such as including simple damping terms to make the trajectory smoother. Damping has the side effect of lengthening the time the end-effector took to close the tracking error.

The result for case 2 is shown in Figure 6.7. It can be observed that the initial position error in singular direction is negative, because the end-effector lead the desired trajectory. As it leaves the singular region, the full 6 DoF motion control pushes the end-effector back into the singular region to close the tracking error. The end-effector rattled back and forth across the boundary of the singular region several times before the desired trajectory caught up and lead ahead. Defining a new intermediate desired trajectory with current position and velocity as initial condition ensures the smooth transition. However, the new (intermediate) trajectory will not match the original

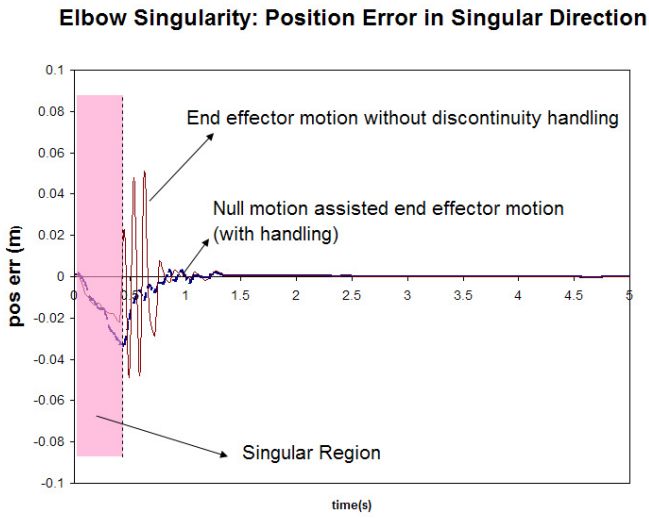


Figure 6.7: Motion of the end-effector as it exits the elbow singularity. The singular direction: translation along X axis of Frame $\{B\}$, which is the line connecting wrist point to the origin of the Base Frame.

trajectory 100%, so the control law will not attempt to close the tracking error immediately. The effect will be more significant if the portion inside the singular region constitute a large portion of the desired trajectory.

6.5 Conclusion

When a division in workspace is defined to handle singular configurations, discontinuities arise at the boundary between the sub-spaces. Error was observed especially when exiting the singular region, when the end-effector regains its full 6 DoF motion control. Handling methods were suggested and implemented on simulation and on PUMA 560 manipulator. Results show that although it is possible to obtain smooth trajectory while moving across singular regions, in the vicinity of singularity, certain tracking performance still has to be sacrificed to achieve motion in singular direction.

This is unavoidable, as singularity is a physical phenomenon of the manipulator, which can only be truly removed through the mechanical design of the robot.

CHAPTER 7

SINGULARITY HANDLING: BY VIRTUAL JOINTS

7.1 Chapter Overview

While the method presented in Chapter 5 involves removing the degenerate components of the task, the method in this chapter proposed adding "virtual" joints into the manipulator so that it can compensate for the lost DOFs when singularity occurs. The method is uniform throughout the entire workspace and does not divide the workspace into different regions. The chapter covers the method proposed and the issues involved in the technique, including the use of null space to ensure that the torque command vector assigned to carry out the task is only confined to the physical joints and not the virtual joints. The method was implemented on PUMA 560 manipulator and the results are presented.

7.2 Introduction

As explained in Chapter 5, there are two main categories of singularity handling method. The first method involves defining a singular region around the singularity, within which a singularity robust control algorithm is applied. This in effect divides the workspace into two sub-spaces: singular and non-singular region. The second method does not have a division of workspace, and the singularity robust algorithm

is applied uniformly across the entire workspace. The method presented in Chapter 5 is of the first category. While inside the singular region, the degenerate components of the tasks is removed, therefore collapsing the Jacobian into a matrix of less number of rows, but of full rank.

In this chapter, a second method of singularity handling is presented. This method is of the second category where there is no division of workspace with different control algorithms. Instead of removing the degenerate components, this method proposed to add extra "joints" into the manipulator, in the directions that are lost during singularity. By supplying extra "virtual joints" into the system, the system no longer has any singularities [57]. This method is implemented into the entire workspace, therefore there are no divisions on workspace into sub-regions. Because this method adds only virtual joints, not real or physical joints, motion in the degenerate direction at singularities are still not achievable. There is also an immediate issue that the robot might rely on the virtual joints to complete its tasks, a case that would adversely affect the tracking performance, since the virtual joints do not actually exist. Null space projection is then used to force the torque command assigned to the virtual joints to zero, therefore fully utilising the physically existent joints.

7.3 Virtual Joints

There are several methods whereby no division of into singular and non-singular regions were made for example in [5], damped least-squares method was used to obtain a modified Jacobian that is not singular. Kircanski [53] utilized the Singular Value Decomposition (SVD) and replaced the zero value at the diagonal matrix at singularity with a continuous function of non-zero value.

Rather than removing the component of the task that contains singularity, this method proposes to supply additional column(s) to the Jacobian matrix to maintain its rank, even in singular configurations. This can also be thought of as introducing virtual joint(s) to handle the lost degree(s)-of-freedom in the vicinity of singularity. These joints only exist computationally, not physically. Therefore, at singular configuration, the manipulator is still not able to perform any tasks in the degenerate directions although the inverse of the Jacobian matrix exists. This would prevent the manipulator from going into excessive joint rates (or unstable region). The advantage of this method is that there is no division of workspace or switching of control algorithms, therefore resulting in smoother performance and simpler method.

An immediate issue with this method is that we do not want the robot to rely on the virtual joints to complete its tasks, a case that would adversely affect the tracking performance, since the virtual joints do not actually exist. Null space projection is then used to force the actuation command assigned to the virtual joints to zero, therefore fully utilizing the physically existent joints.

The algorithm is evaluated in simulation and implemented on the PUMA 560 manipulator. The results of the real-time experiments are presented in this chapter.

7.3.1 Supplying Virtual Joints

When singularity happens, the manipulator loses its degree(s)-of-freedom, and the Jacobian matrix loses its rank(s) accordingly. In this condition, the inverse of the Jacobian matrix no longer exists. As the inverse of Jacobian matrix is essential in

transforming control commands from Cartesian space (in which our tasks are specified) to joint space (in which the robot joints are controlled), it is desired to have a Jacobian matrix whose inverse always exists.

The method proposed is to supply the Jacobian matrix with extra column(s), that would guarantee that the Jacobian always has full rank even when the manipulator is in singular configuration. This is done by adding "virtual joints" into the Jacobian matrix to replace the lost DOF when singularity occurs. This expands the Jacobian matrix from its original size of $m \times n$ matrix to $m \times (n+v)$ matrix. m is the dimension of the tasks specified, which is usually 6, representing 3 DOFs for translation (position) and 3 DOFs for orientation. n is the number of joints the manipulator possesses. v is the number of extra (virtual) joints that is to be added into the system.

Figure 7.1 shows a two-link planar robot in singular configuration (left). This is a boundary singularity. The degree of freedom lost in this case is the ability to translate along the line described by the straight arm. However, if two more degrees of freedom were added to the system, for example: two prismatic joints (Figure 7.1 bottom left), the robot can now translate anywhere even when the arm is straightened (assuming the prismatic joints do not reach their limits). A more efficient way would be to identify the lost degree of freedom at such singular configuration and to supply an extra DOF in this direction with a virtual joint (Figure 7.1 bottom right), which is a prismatic joint to allow the end-effector to translate along the lost DOF.

As the modification is done permanently to the Jacobian matrix to prevent it from ever becoming rank-deficient, the control algorithm is uniform throughout the workspace. This eliminates the switching of control algorithm as found in some other methods, as mentioned in the introduction.

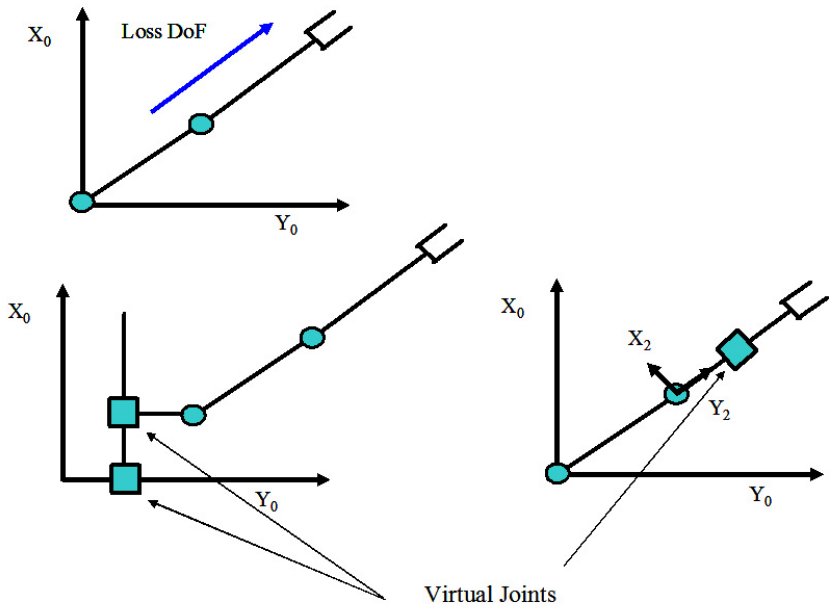


Figure 7.1: Example of a two-link planar manipulator in singular configuration and its lost DOF (top), and two ways of supplementing virtual joints into the system, where circles represent the revolute joints and squares represent the prismatic (virtual) joints

It should be noted that although the Jacobian matrix is made invertible at singular configuration, the virtual joints still does not exist physically, therefore the manipulator still can not perform its tasks in the lost degree-of-freedom.

The control algorithm would produce a command vector of size $(n+v)$ by 1, containing commands to be sent to individual joints (motor). In the case of operational space formulation, it is a torque vector, while in higher level control such as velocity control, it is a joint velocity vector. The elements corresponding to the virtual joints are therefore ignored and only those corresponding to physically existing joints would be sent accordingly.

7.3.2 Avoiding Assignment of Command to Virtual Joints

The Jacobian matrix is a mapping from the joint space velocities to the task space velocity. When there are more joints than the number of degrees-of-freedom required by the task, the robot is considered "redundant". In a redundant system, there are many solutions in the joint space that would map onto the desired path of the end-effector in task space. We can choose one set of the solution by having the desired behavior of the extra joints projected into the null space of the Jacobian.

In our case, the joints added to the system are virtual, and they only exist computationally and not physically. As the manipulator is now described as having real and virtual joints, there is an issue whereby the manipulator might rely on the non-existent joints to complete the specified task. Some non-zero torque values may be assigned to the virtual joints in its attempt to track the given trajectory. This would have an adverse effect on the tracking performance of the robot.

Since the Jacobian matrix is now redundant, the problem mentioned above can be prevented by choosing a set of solution where the desired null space behavior is to have the virtual joints stationary at zero position. This keeps the virtual joints unused in completing the specified task and assigns the roles of completing the desired trajectory to the existing joints.

This is different from simply setting the torque or velocity command values to the virtual joints to zero. If a solution in the joint space (joint command) has been obtained by including the virtual joints as real joints, setting command values to these joints to zero would produce incomplete actuation for the end-effector to follow the task space trajectory. Using null space projection, however, the velocity of the

virtual joints can be made zero, while maintaining the resulting forces/velocity at the end-effector.

The null space behavior, that has been used extensively in the earlier chapters, is specified by defining the gradient descent of a potential function to be projected into the null space of the Jacobian matrix. For our case of maintaining the virtual joints unused, *i.e.* maintaining the displacement of these "joints" as zero, null space potential function can be defined as follows:

$$v_0(\mathbf{q}) = \frac{1}{2}\sum_i k_i[f(\mathbf{q}_i) - f(\mathbf{q}_{i(desired)})]^2 \tag{7.1}$$

where $v_0(\mathbf{q})$ is the potential function, k_i is a constant gain. To obtain the desired null space behavior, which is to keep the virtual joints stationary at zero, the potential functions are defined as:

$$v_0(\mathbf{q}) = \frac{1}{2}\sum_i k_i[q_i - q_{i(desired)}]^2 \tag{7.2}$$

where q_i are the joints that we would like to control in null space, which in this case, are the virtual joints. $q_{i(desired)}$ are the desired values of these joints, which are set to zero.

The gradient descent of the potential functions are then used as null space torque or velocity to be projected into the null space of the Jacobian.

Example: 3-link planar manipulator (R-P-R)

Supposed there is a three-link planar manipulator, with two revolute joints q_1 and q_2 and a prismatic joint d_3 (revolute- prismatic- revolute). If the task was to control the position of the end-effector, then the manipulator is redundant.

When in velocity control, the motion of the end-effector can be described as:

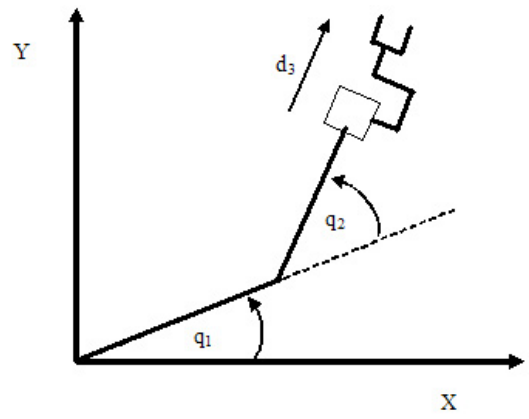


Figure 7.2: An example of a three link planar manipulator, with two revolute joints q_1 and q_2 and prismatic joint d_3. This simple example is used to illustrate the points in the null space control.

$$d\mathbf{q} = \mathbf{J}^{\#} d\mathbf{x} + [\mathbf{I} - \mathbf{J}^{\#}\mathbf{J}] d\mathbf{q}_0 \tag{7.3}$$

where $d\mathbf{q}_0$ is the gradient descent of the potential function $v_0(\mathbf{q})$, which is used to described the desired behaviour of the internal joint motion.

$$d\mathbf{q}_0 = -\nabla v_0(\mathbf{q}) \tag{7.4}$$

The desired internal joint motion is so that d_3 is stationary at position $d_3 = 0$, therefore:

$$v_0(\mathbf{q}) = \frac{1}{2} k (d_3 - d_{3,desired})^2 \tag{7.5}$$

where $d_{3,desired} = 0$.

When the end-effector was commanded to perform a straight line trajectory, there are many possible sets of solutions. Figure 7.3 shows the several sets of possible joint

displacement to the same end-effector motion. Gain k is of the null space equation 7.5. When set to zero, the resulting motion is dictated entirely by the $\mathbf{J}^{\#}\dot{\mathbf{x}}$ term. In this case, it can be seen that d_3 is free to move to produce end-effector motion that tracks the desired trajectory. When $k = 100$, then d_3 is forced to be zero at all time.

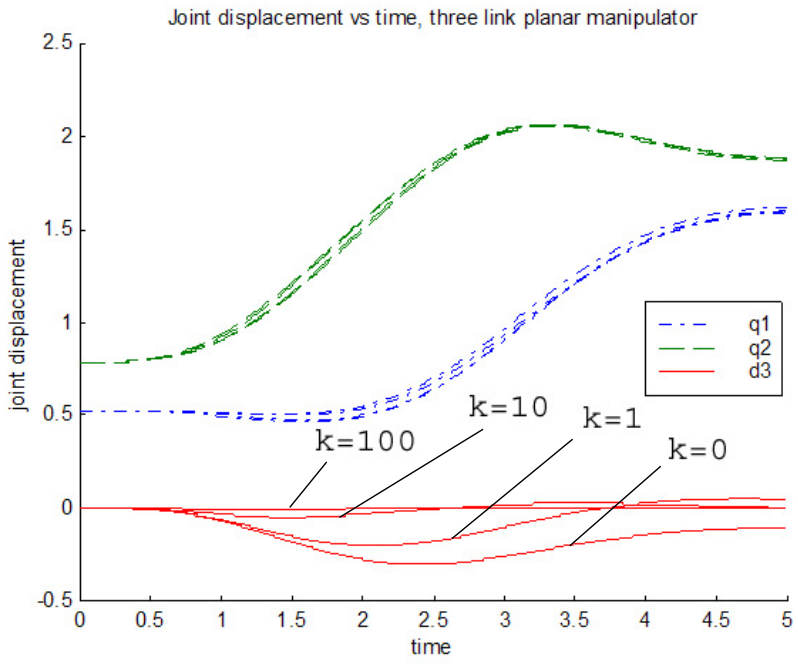


Figure 7.3: Different sets of joint displacements solution for the same end-effector motion, generated by different gain 'k' on the desired null space behaviour, which is keeping $d_3 = 0$.

If d_3 was a virtual joint, there will be no encoder feedback for d_3, and the reading for the value of d_3 will always be zero. When this happens, the term $d_3 - d_{3,desired}$ of Equation 7.5 will always be zero, which produce the same effect as having $k = 0$. The motion of the end-effector will be dictated by the $\mathbf{J}^{\#}\dot{\mathbf{x}}$ term and there will be motion assigned to d_3 in order to track the desired trajectory.

In the real time implementation, it is necessary to numerically integrate the command $d\mathbf{q}$ sent to virtual joint d_3 to obtain a "virtual" reading for d_3. This will allow the control law for the null space behaviour (Equation 7.5) to function.

7.3.3 Inclusion of Dynamic Model for Torque Control

The subsection above described the method to apply the virtual joints with velocity controller. That way, command is sent to the manipulator as joint velocity command vector.

For the low level control methods such as torque control, the dynamics of the manipulator needs to be included. The torque control method such as the operational space formulation is described in Chapter 2.

In torque control, it is necessary to obtain the inertia matrix $\mathbf{A}(\mathbf{q})$. The $\mathbf{A}$ matrix is derived by:

$$\mathbf{A}(\mathbf{q}) = \sum_{i}^{n} m_i \mathbf{J}_{vi}^T \mathbf{J}_{vi} + \mathbf{J}_{\omega i} \mathbf{I}_C \mathbf{J}_{\omega i} \tag{7.6}$$

where i the subscript denoting the centers of mass of individual links and n is the number of links in the manipulator. $\mathbf{J}_{vi}$ and $\mathbf{J}_{\omega i}$ are the translational and orientational Jacobian matrix of the individual centers of mass, and $\mathbf{I}_C$ is the inertia tensor of the individual links taken at the center of mass.

Virtual links were modelled as having no mass or inertia. In fact, in our model, it is not assigned an individual link. However, it is taken that the dynamics of the links after a virtual joint is affected by the "motion" of the virtual joint.

Let us take an example of a two-link planar manipulator with a prismatic virtual joint inserted between the two revolute joints (Figure 7.4). The center of mass of each link are set at a distance a_1 and a_2 away from joint 1 and 2 respectively. The

links are of length l_1 and l_2. The masses are denoted as m_1 and m_2, while the inertia tensor at m_1 and m_2 are defined as $\mathbf{I}_{C1}$ and $\mathbf{I}_{C2}$ where

$$\mathbf{I}_{Ci} = \begin{bmatrix} I_{xxi} & 0 & 0 \\ 0 & I_{yyi} & 0 \\ 0 & 0 & I_{zzi} \end{bmatrix} \tag{7.7}$$

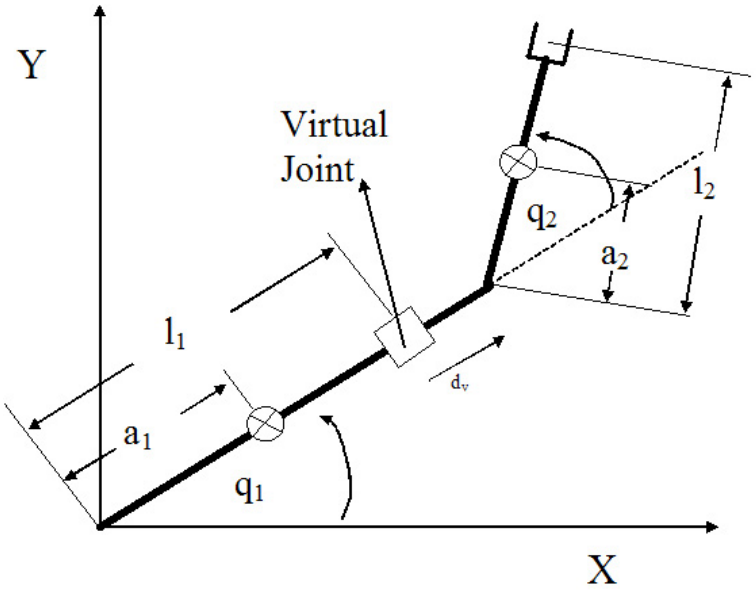


Figure 7.4: An example of a two-link planar manipulator, with a prismatic virtual joint inserted between the two revolute joints.

Although normally in serial manipulator, having three joints would equal having three links, and therefore three centers of mass, in this case, only the physical joints are modelled. Hence only m_1 and m_2.

The Jacobian matrices of the centers of mass can be calculated as:

$$\mathbf{J}_{V1} = \begin{bmatrix} -a_1 s_1 & 0 & 0 \\ a_1 c_1 & 0 & 0 \end{bmatrix} \tag{7.8}$$

$$\mathbf{J}_{V2} = \begin{bmatrix} -(l_1 + d_v)S_1 - a_2 S_{12} & c_1 & -a_2 s_{12} \\ (l_1 + d_v)c_1 + a_2 c_{12} & s_1 & a_2 c_{12} \end{bmatrix} \tag{7.9}$$

$$\mathbf{J}_{\omega 1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{7.10}$$

$$\mathbf{J}_{\omega 2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{7.11}$$

then the $\mathbf{A}(\mathbf{q})$ is calculated as:

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} I_{zz1} + I_{zz2} + a_1^2 m_1 + m_2(a_2^2 + l_1^2 + 2a_2 l_1 c_2) & I_{zz2} + a_2 m_2(a_2 + l_1 c_2) \\ I_{zz2} + a_2 m_2(a_2 + l_1 c_2) & I_{zz2} + a_2^2 m_2 \end{bmatrix} \tag{7.12}$$

With the $\mathbf{A}$ matrix defined, the $\mathbf{\Lambda}(\mathbf{q})$ matrix (kinetic energy matrix) is defined as covered in Chapter 2 and the torque command for the manipulator is obtained.

In the real-time implementation, the torque command to the physical joints will result in encoder feedback of joint displacement and velocity reading. The feedback of the virtual joint is obtained by simulation. Since the joint mass matrix $\mathbf{A}$ and its inverse is already calculated for control purposes, joint acceleration vector can be obtained from the torque command sent to the virtual joint. The integration of the vector yields the virtual joint displacement and velocity.

The simulation result of this three link planar manipulator (revolute-prismatic-revolute) is shown in Figure 7.5. It is the different sets of solution of joint displacement for the same end-effector trajectory. When the gain for the null space control is large enough, it will keep the virtual joint close to zero and hence the motion is executed with only the two physical revolute joints. The comparison of the joint motion of the

two link robot (controlled as two-link robot) and of the same two-link robot controlled as a three-link robot but with the virtual joint kept stationary is shown in Figure 7.6.

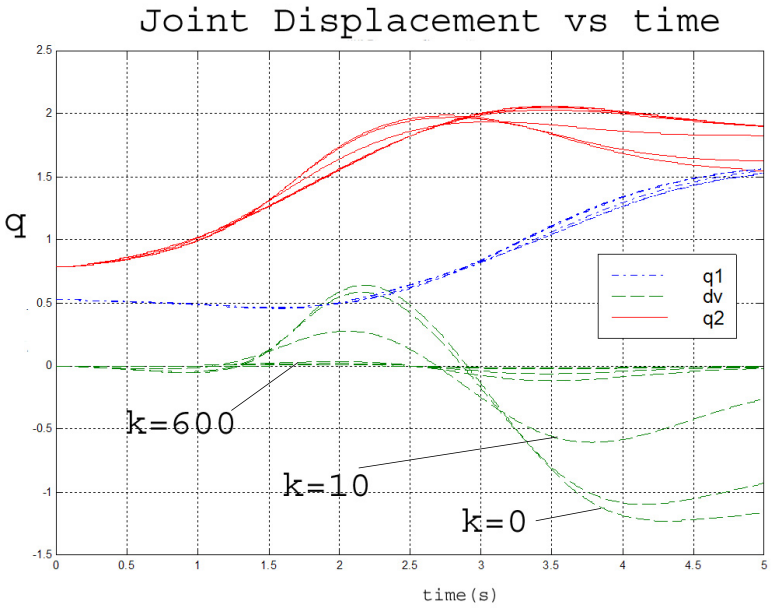


Figure 7.5: The joint displacement for the example of two link planar manipulator with a virtual prismatic joint inserted between the two revolute joints. This graph shows different possible sets of solution for the same end-effector trajectory in torque control.

7.3.4 Effect of Simulated Joint Feedback

At this point, it is necessary to analyse the validity of using a simulated result as feedbacks of the virtual joints. Since these feedback are simulated, it is necessary to analyse to what extent any error in the integration affects the performance of the algorithm.

In kinematic control, such as the example shown in Figure 7.3, a three link planar manipulator is compared to a two-link planar manipulator. The length of the links of

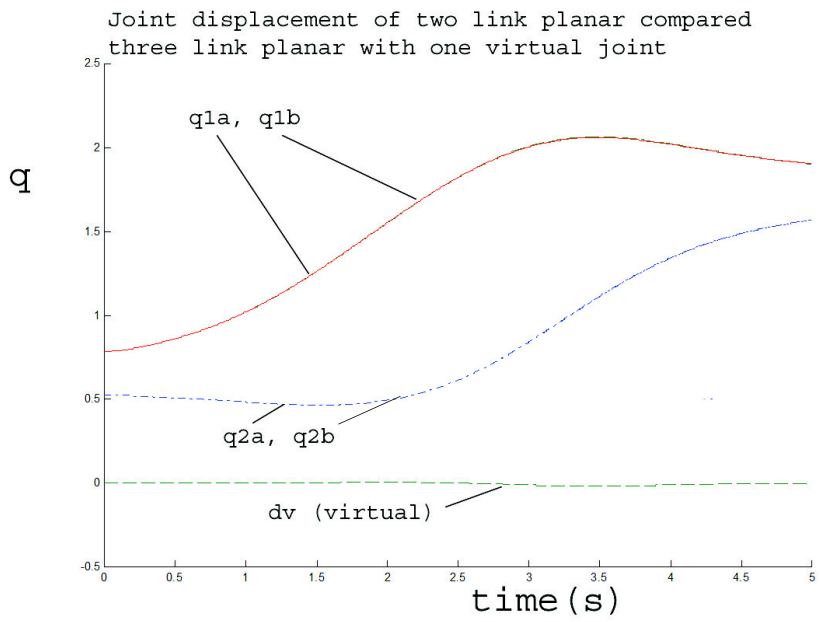


Figure 7.6: The comparison of the joint motion of a two-link manipulator (revolute-revolute) controlled as a two link robot ($q1a$, $q2a$) and as three link robot whose virtual joint is kept stationary ($q1b$, $q2b$, d_V). The control algorithm with virtual joint successfully emulate the joint motion of the two link robot.

these manipulator are set so that when the virtual joint of the three-link manipulator is fixed at zero position, the manipulator is identical to the two link case. When the three link manipulator is commanded to track a trajectory with joint 3 (the virtual link) kept stationary by null space, the resulting trajectory is almost identical to that of a two link manipulator's. Referring to Equation 7.3, the inverse kinematic solution produced by the $\mathbf{J}^{\#}d\mathbf{x}$ is the solution that minimises the norm of the error (the least square solution). To select a solution that produces zero joint velocity at the virtual joint, the null space half of the equation is used. Therefore, the role of the null space projection of the potential function is to shift the solution produced by the $\mathbf{J}^{\#}d\mathbf{x}$ term to another set that keeps the virtual joint stationary. When all the redundant

(virtual) joints are kept "stationary", The manipulator is no longer redundant. A unique solution to the inverse kinematic exists using only the physically existent joints (in the three link planar example, when the virtual joint is kept stationary, then there is a unique solution for the inverse kinematics of 2DoF task for the remaining two joints). Since the virtual joint is kept at zero at all time, the value of feedback from the virtual joint in the form of simulated joint position is not crucial, as long as it serves the purpose of shifting all the necessary joint motion to satisfy the desired trajectory to the physically existent joint. The only crucial thing is the sign $(+ \setminus -)$ of the simulated joint position for the (feedback) control law to work.

Similarly, for torque control, the simulated feedback of the virtual joint velocity and position is obtained by integrating the joint acceleration, which in turn is obtained from the torque command to the virtual joint(s) and the inverse of the mass matrix $\mathbf{A}$. The three link planar manipulator with the virtual joint (d_V) kept stationary is compared to the performance of the two-link planar manipulator (with identical parameters as the three link when the virtual joint is locked at zero position). The result is shown in Figure 7.6. Again, the mass assigned to the virtual joint does not affect the performance of the algorithm, as long as the sign of the command produced by the null space potential function is correct for the control law to function. In a way, a virtual model is constructed treating the dynamics of the virtual joint as a unit mass whose value is tuned by the gain k_i in the potential function.

7.3.5 In Singular Configuration

Because the additional joints are virtual, motion in the singular direction is still not possible when the end-effector is close to singular configuration.

If the desired motion requires the end-effector to move in the singular direction while in the vicinity of the singular configuration, then it is no longer possible to keep the virtual joint at $q_v = 0$ (Section 2.4.4). When this happens, the simulated joint displacement would grow, because these virtual joints represent motion in the singular direction.

When the end-effector is at perfect singularity, then there is no possible motion in the singular direction. However, as the manipulator moves slightly off the singular configuration, the null space behaviour will attempt to keep the virtual joint (representing the singular direction) at $q_v = 0$ and therefore shift the task to other joints. Since the manipulability of these physical joints are low near singularity, the response could be rather slow initially till a large enough error is accumulated and/or the end-effector move further away from the singularity.

The implementation result will be shown in PUMA 560 below.

7.4 Application on PUMA robot: the method of virtual joint

The algorithm above was implemented on the PUMA 560 manipulator. The result comprises of two parts: for the non-singular motion and for singular motion.

For non-singular motion, it is important to verify that the algorithm is capable of assigning no component of the task to the virtual joint, so the task will be wholly performed by the physical joints. The result is compared to the performance of PUMA under normal control algorithm, which is one without the addition of virtual joints.

For the singular motion, it is necessary to verify the stability of the control algorithm when going through the singular configurations.

The singular configurations were identified in Section 4.3.1. The following example shows the handling of wrist singularity in detail. As defined in Equation 2.18, Equation 7.13 shows the lower right quadrant of the Jacobian matrix (${}^4\mathbf{J}_{22}$), expressed in Frame{4}. When wrist singularity occurs, $s_5 = 0$, and the first row of this matrix is all zeros, showing the singular direction, and the determinant of the Jacobian is zero.

$$ {}^4\mathbf{J}_{22} = \begin{pmatrix} 0 & 0 & s_5 \\ 0 & 1 & 0 \\ 1 & 0 & c_5 \end{pmatrix} \tag{7.13} $$

In a design by [68], a four jointed spherical wrist was designed to handle the problem of singularity. In our method, a virtual revolute joint is added to compensate for the lost DOF. The diagram of the wrist with the virtual joint added is shown in Figure 7.7 (b).

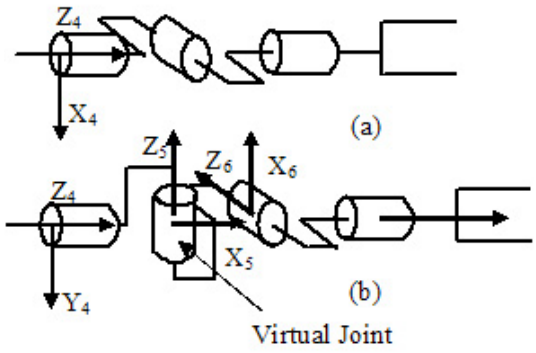


Figure 7.7: The diagram of the PUMA spherical wrist (a), and the wrist with added virtual joint (b)

Since the modified DH parameter requires choosing axis x_{i-1} along the common normal of axes z_{i-1} and z_i, (with direction from joint $i-1$ to joint i) the addition of

virtual joint requires reassignment of DH frames. The new frame assignment for the wrist is shown in Figure 7.7 (b).

The additional joint is reflected on the lower right quadrant of the Jacobian with respect to Frame{4}:

$$ {}^4\mathbf{J}_{22v} = \begin{pmatrix} 0 & 0 & -s_5 & -s_5c_6 \\ 0 & -1 & 0 & s_6 \\ 1 & 0 & C_5 & c_5c_6 \end{pmatrix} \tag{7.14} $$

where ${}^4\mathbf{J}_{22v}$ is the virtual-joint-supplemented version of ${}^4\mathbf{J}_{22}$, with the system assumed to be a 7-jointed mechanism.

The operational space formulation [1] was implemented in our system. The null space torque $\boldsymbol{\tau}_0$ which is the gradient descent of the potential functions to control the null space behavior of the system is defined as:

$$ \tau_0 = -\nabla v_0(q) = k_i(q_i - q_{i(desired)})\mathbf{z} \tag{7.15} $$

where $\mathbf{z}$ is a vector of size $n+v$: $\mathbf{z} = [z_1 z_2 ... z_{n+v}]^T$, where $z_i = 0$ if q_i is a real joint and $z_i = 1$ if q_i is a virtual joint.

When implemented in velocity control, the equation would be defined as:

$$ \delta\mathbf{q} = \mathbf{J}^{-1}\delta\mathbf{x} + \mathcal{N}\delta\mathbf{q}_0 \tag{7.16} $$

where $\delta\mathbf{q}$ is the joint velocity vector, $\delta\mathbf{x}$ is the cartesian velocity vector, and

$$ \delta\mathbf{q}_0 = -\nabla v_0(\mathbf{q}) \tag{7.17} $$

The other two singularities (head and elbow) are position singularities (causes one or more rows of $\mathbf{J}_V$ to be zero). The same method as that in handling wrist singularity can be applied there. However, it was shown in [62] and in Equation

3.18 that the lost DoF for elbow singularity (when elbow is straightened) is of a very complex expression. This is because the singular direction is not aligned with any axes of the frames defined by the DH convention. Transforming the Jacobian into this frame can result in a very complicated matrix.

Following the example described in Figure 7.1, a more straight forward method can be done by adding three prismatic (virtual) joints with respect to the absolute frame (Frame {0}), instead of supplying only two in the direction of lost degrees of freedom (for head and elbow singularities).

The resulting frame assignment is shown in Figure 7.8,

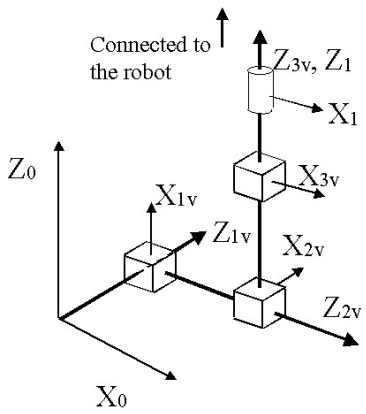


Figure 7.8: The representation of the three prismatic virtual joints with respect to the base frame to handle position singularities.

where Z_{1v}, X_{1v}, Z_{2v}, X_{2v}, Z_{3v}, and X_{3v} denotes the frames of the virtual prismatic joints. The diagram continues to the rest of the PUMA robot, with Frame{1} (denoted by Z_1, X_1) is the Frame{1} of PUMA robot as shown in Figure A.1 in the Appendix.

To incorporate the virtual joints to handle all the singularities (wrist, elbow, and head), the PUMA is now regarded as a 10-jointed mechanism, with the virtual joints labelled as joints 1, 2, 3 (prismatic) and 8 (revolute, at the wrist). This results is the following 6×10 Jacobian matrix:

$$\mathbf{J} = \left(\begin{array}{ccc|c|c} 0 & 1 & 0 & & \\ 1 & 0 & 0 & \mathbf{J}_{11} & \mathbf{0}_{3\times 4} \\ 0 & 0 & 1 & & \\ -- & -- & -- & - - - - & - - - - \\ & \mathbf{0}_{3\times 3} & & \mathbf{J}_{21} & {}^0\mathbf{J}_{22v} \end{array} \right) \tag{7.18}$$

where $\mathbf{J}_{11}$ and $\mathbf{J}_{22}$ is the as defined in (2.18). The full diagram representing all the 10 frames of the system (virtual and real) is shown in Figure 7.10.

The potential function projected into the null space to prevent the virtual joints from being relied upon to complete the task can be designed for the prismatic joints as:

$$v_0(\mathbf{q}) = \frac{1}{2} \sum_i k_i (q_i - q_{i(desired)})^2 \tag{7.19}$$

where $v_0(\mathbf{q})$ is the potential function for the virtual joints d_1, d_2, d_3 and q_8. d_i denotes a prismatic joint and q_i revolute.

7.5 Implementation Result on PUMA: by virtual joint

The result is divided into two subsections, the first covers the motion through singular configurations. The second covers the motion of the manipulator in a non-singular motion. Torque control based on the operational space formulation was used for the experiment.

7.5.1 Motion through Singular Configuration

Two sets of experimental result are presented in this section. Figure 7.11 shows the performance of PUMA as it goes through a wrist singularity, and Figure 7.12 as

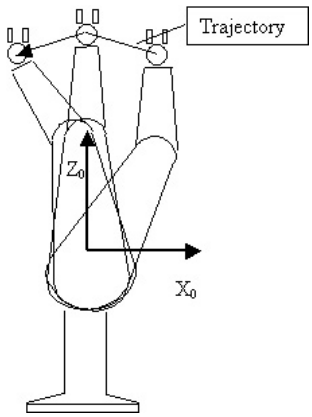


Figure 7.9: The trajectory of the PUMA in going through the combined wrist, elbow, and head singularities

it goes into a combined wrist, elbow and head singularity (when it points straight up vertically) and out of it (see Figure 7.9).

The PUMA is now regarded as a 10-jointed mechanism, with original PUMA joints labelled as joint 4,5,6,7,9, and 10, and virtual joints as joint 1,2,3, and 8 (Figure 7.10).

In going through the wrist singularity, the trajectory is a vertical path in X-Z plane with respect to Frame {0} (constant X, constant Y, increasing Z) (Figure 7.10 shows the definition of base frame).

The robot is shown to go through the wrist singularity, as in the bottom plot of Figure 7.11, where the wrist joint goes through $q_9 = 0$. The top plot shows the tracking error of the end-effector from the desired trajectory. Since the path is in increasing Z direction, while maintaining constant X and Y, little error was observed in X and Y direction. Maximum error of 0.2 mm in the Z-axis is comparable to the performance of the robot while tracking a non-singular path. Velocities for joint 7 and

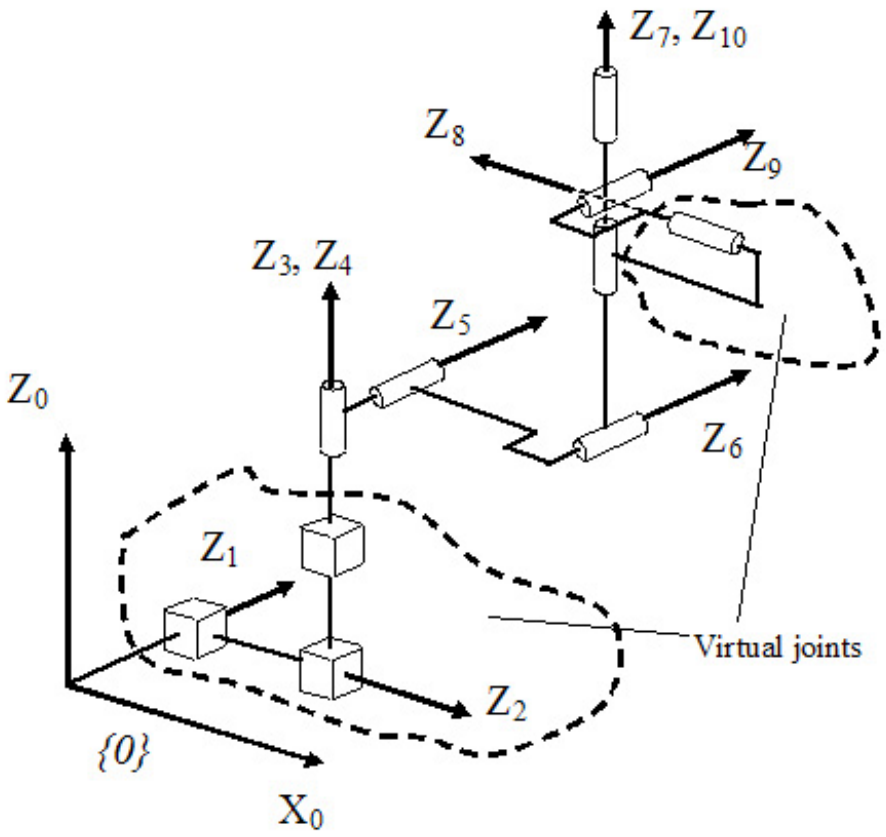


Figure 7.10: The frame assignment for PUMA in this experiment. Joint 1,2,3, and 8 are inserted as virtual, while the rest are the relabelled PUMA physical joints.

joint 10 of the PUMA robot were shown to be stable. No sudden change or excessive velocity was observed (second and third plots in Figure 7.11).

Similar result is shown in Figure 7.12, as the robot follows a trajectory in X-Z plane to go into wrist, elbow and head singularity and out again (see Figure 7.9). The task space tracking performance (Figure 7.12 top) is comparable to that in non-singular path. The second and third plots of Figure 7.12 show that there is no sudden jerks or excessive joint rates while the robot goes through singularity. The plot of the

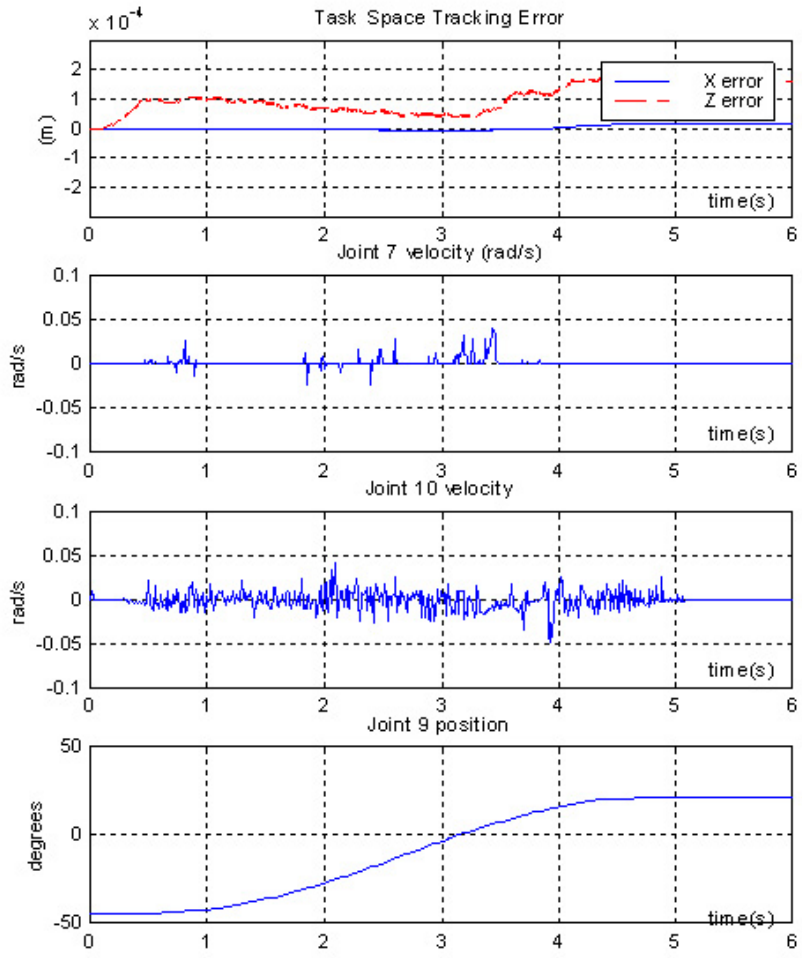


Figure 7.11: The result of the experiment, on tracking a trajectory through wrist singularity. The singular configuration is when the wrist joint (joint 9) goes through $q_9 = 0$.

determinant of the Jacobian matrix (Figure 7.12 bottom) shows that the robot went through the singularity.

The tracking error in this trajectory was shown to accumulate for a while before converging to zero. This was as predicted in Section 7.3.4, that since motion in singular direction is not possible, some accumulation of error might happens before the manipulator is able to overcome the low manipulability. When the elbow of the manipulator is straightened (elbow singularity), it required a lot of effort to bend the elbow again to follow the trajectory without the help of null motion such as the one described in Chapter 5.

The results verifies the stability of the manipulator going through the singular configurations, assisted by "virtual joints", in a real-time implementation.

7.5.2 Non-singular motion

In this sub-section, we will verify that the addition of virtual joints does not affect the tracking performance of the PUMA in real-time implementation. As explained above, the PUMA is now modelled as having 10 joints (Figure 7.8). It is necessary to show that it is the virtual joints are kept "stationary" and that the motion at the end-effector is not compromised.

Figure 7.13 and 7.14 shows the tracking performance of PUMA in terms of absolute value of the tracking error to the desired trajectory in task space. The performance in each axis is shown in separate graph for clarity. Each graph contains the performance of the PUMA manipulator when controlled in the conventional method of operational space formulation (torque control) without any singularity handling algorithm, and when controlled with the algorithm with virtually added joints. The graph shows that

the error follows the same trend for both cases. The conclusion is that the end-effector tracks the identical desired trajectory with the same margin of error and that there is no degradation in the tracking performance of the manipulator in non-singular motion due to the addition of the virtual joints.

Figure 7.13 shows the tracking error in the position of the end-effector while Figure 7.14 shows that in orientation. Again, the orientation error is displayed as $d\boldsymbol{\phi} = [d\phi_X, d\phi_Y, d\phi_Z]^T$ as explained in Section 2.7.

7.6 Conclusion

The singularity handling method proposed in this chapter is to supply 'extra joints' to the system in the lost degrees-of-freedom. This time, we introduce the idea without actually building the extra joints into the physical mechanism, but only to use it to provide extra rank into the Jacobian matrix. This prevents the Jacobian from going rank-deficient as the manipulator enters singularity. The method has been shown to work well, and is able to go through singular configuration in a stable manner. The advantage is shown in the result as having a smooth continuous motion through the singular region, as there is no switching in control algorithm in the vicinity of singularity. Task in the lost degree-of-freedom during singular configuration is still not feasible.

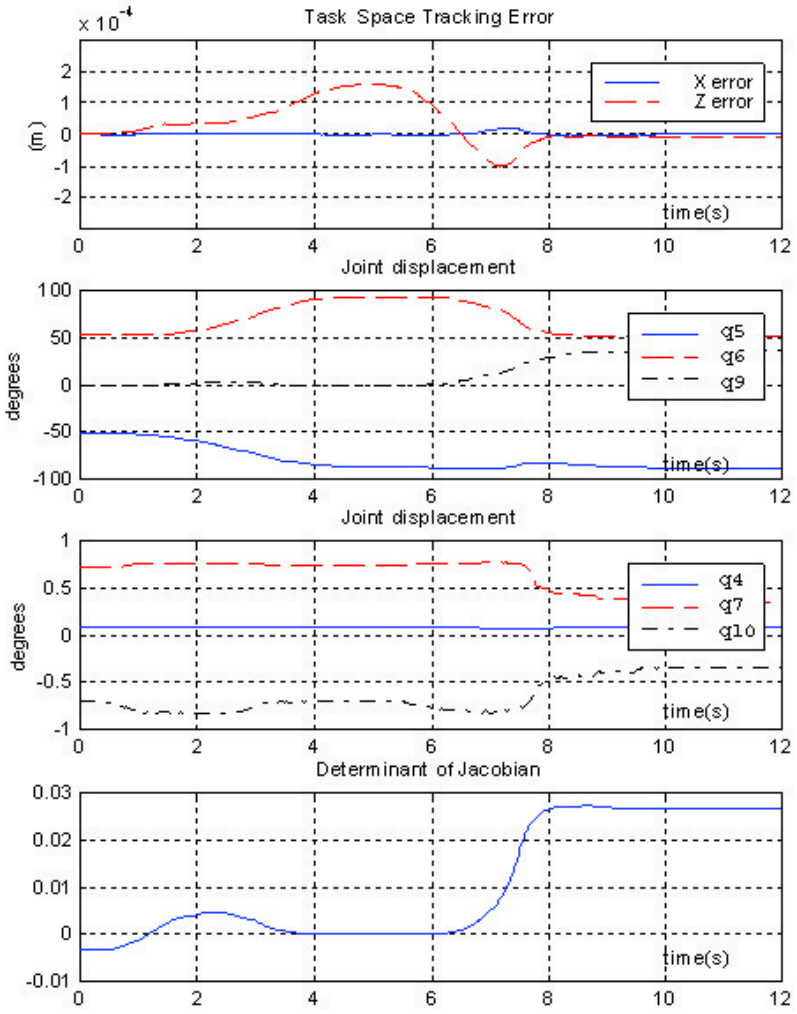


Figure 7.12: The result of the experiment, on tracking a trajectory through combined wrist, elbow, and head singularity.

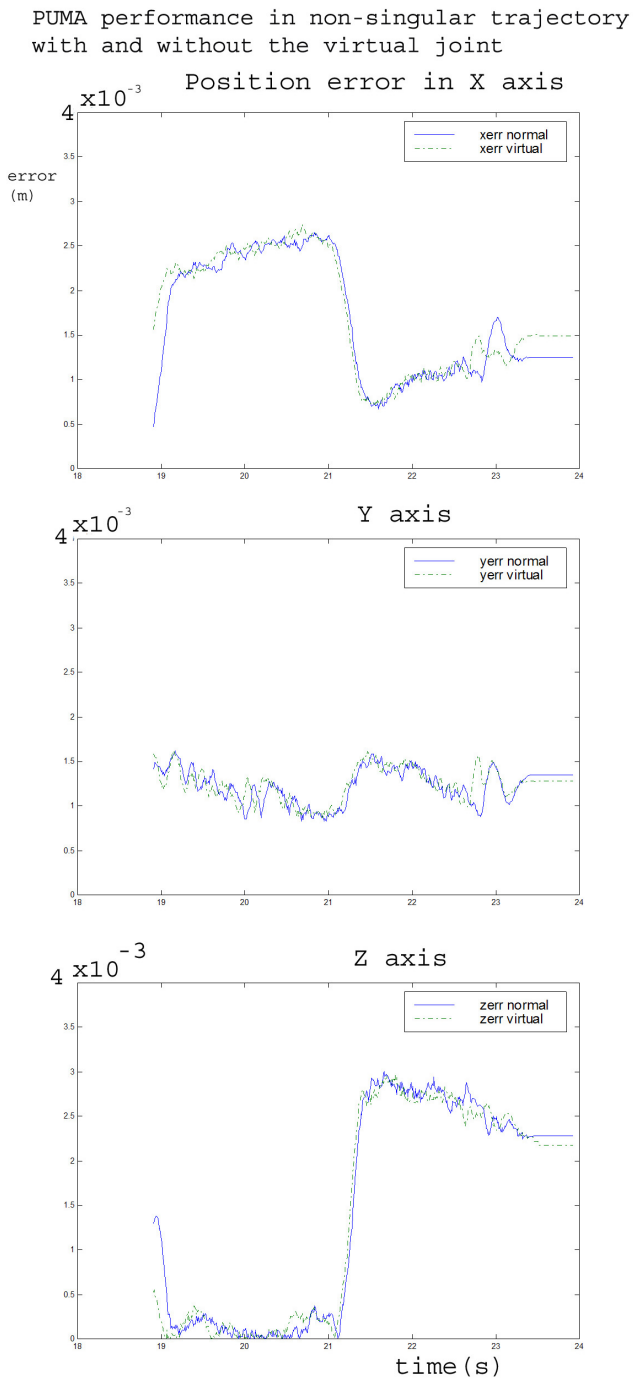


Figure 7.13: Graphs of the absolute value of the position error in the tracking performance of PUMA in non-singular trajectory: when controlled with and without virtual joint. Performance in each axis is shown in separate graph. Graph is shown in task space end-effector tracking error in meters.

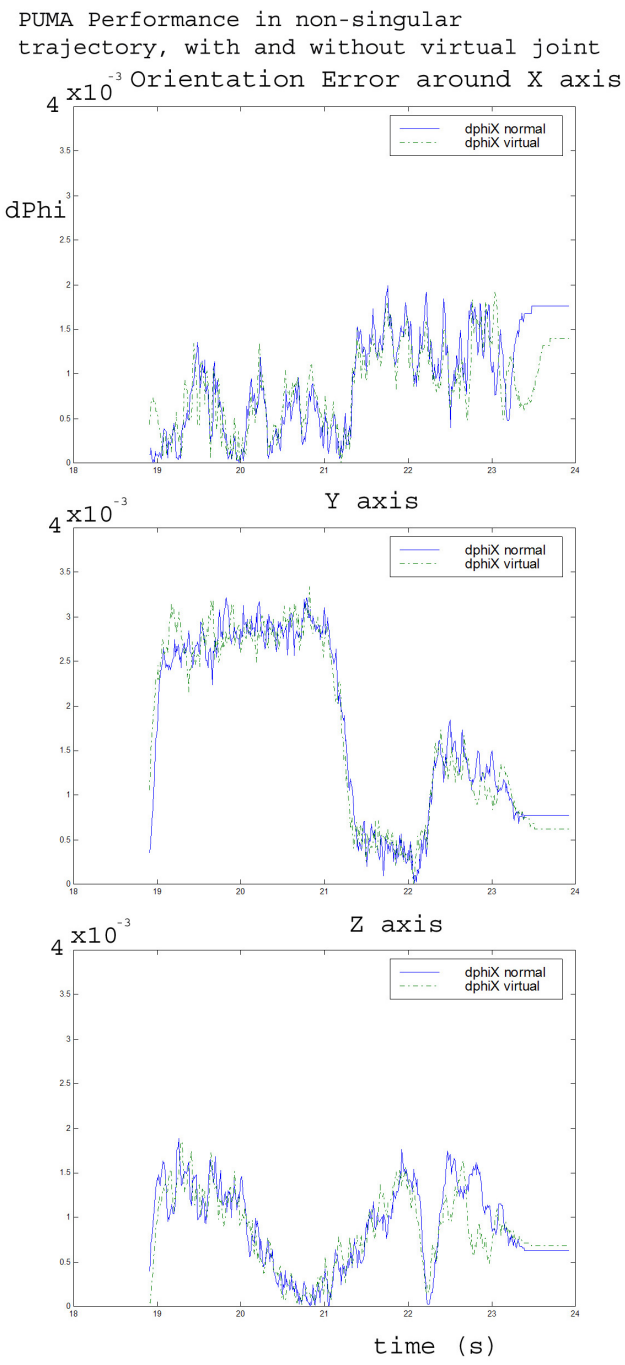


Figure 7.14: Absolute value of the orientation error in the tracking performance of PUMA in non-singular trajectory: when controlled with and without virtual joint. Graph is shown in task space end-effector tracking error in $d\boldsymbol{\phi}$ as explained in Section 2.7.

CHAPTER 8

ARM-BASE INTEGRATION TOWARDS MOBILE MANIPULATION

8.1 Chapter Overview

This chapter utilises everything that has been covered to the previous chapters in the setup and control of a mobile manipulation system. Built on the operational space formulation, the system consists of the PUMA 560 manipulator mounted on an XR4000 Nomad mobile base.

While in Chapter 7 virtual joints are added to overcome singular configurations, this time the concept is applied with addition of real (physically) joints. The extra joints come in the form of a mobile base to add extra mobility to the PUMA arm. It is shown that the extra DoF help to reduce the extent of singular configuration, leaving a smaller portion of the workspace to be affected by it.

The remaining singularity is handled by removing the degenerate components method. The mobile manipulator is now capable of larger workspace, and its implementation in a canopy polishing task has been successful [80].

This chapter presents an early stage of mobile manipulation with the singularity robust algorithm. The task for this industrial application is to polish an aircraft

canopy. The robot is required to maintain a constant 10N force normal to the unknown surface of the aircraft canopy. A mobile manipulator was used to extend the workspace of the manipulator to cover the entire piece of the canopy. The extra degrees of redundancy also gives the robot the dexterity needed to maintain a compliant motion so that the end-effector is always perpendicular to the surface of the canopy. It is shown in this chapter that the singularity robust strategy, implemented on the operational space formulation, was singularity robust in both force and motion control.

The chapter firstly covers the integration technique between the arm and the base. The problem encountered was that the base was in high level velocity control, while the operational space formulation (that has been implemented in the PUMA arm) requires torque control mode. It then discusses the singularity issues, comparing it to Chapter 4 and 5. The result of the implementation is presented.

8.2 Integration of Torque Controlled Arm and Velocity Controlled Base

In mobile manipulation [81], a manipulator arm is typically mounted on a mobile base, and the arm and base are in simultaneous motion during a manipulation task. The capabilities of the arm to manipulate and interact with the environment is very much extended with the large reachable workspace of the mobile base. When implemented in operational space formulation [1], ideally, we would like to have the resulting system as one dynamically compensated system. For that to take place, both arm and base need to be in torque control mode. However, most commercially available mobile bases are velocity-controlled. In this section, a solution to the problem

is presented by combining the operational space formulation with a kinematic-based control (velocity control) while maintaining dynamic compensation.

The dynamic model of the entire mobile manipulator is first derived. Control algorithms using the complete dynamic model are then formulated. The algorithms compute torque and velocity commands for the torque-controlled and velocity-controlled joints, respectively, to execute the desired end-effector task. The operational space formulation [1] is utilised to generate the torque command vector, taking into account the complete arm-base dynamics as seen from the operational point. Utilising the property of the dynamically consistent inverse [4], $\bar{\mathbf{J}}$ is used not only in producing the torque control command, but also the velocity control commands.

The resulting system is capable of dynamically compensated unified force and motion control. The method is implemented on a PUMA 560 arm and the NOMAD XR4000+ base (Figure 8.1). The arm provides 6 degrees of freedom for articulation while the base provides 3 degrees of freedom in the plane and extends the workspace of the manipulator arm to cover the entire canopy. All the six joints of the PUMA arm are torque-controlled. However, the NOMAD base is only capable of high level control, such as velocity control. It is independently driven by 8 DC motors that are not capable of velocity control. Furthermore, velocity command at the motor level is not possible and base motion can be only commanded using high level commands for the holomomic motion in the plane. We therefore treat the base as a 3-DOF velocity controlled system.

The system in this experiment is the PUMA 560 6 DOF manipulator arm with the XR4000 Nomadic mobile base. The performance of the system in free motion and force control task were studied. The method is then applied to an industrial

application involving an aircraft canopy polishing task. The task is similar to earlier experiment by [82], although in that experiment, the surface of the canopy is first mapped, and therefore known to the robot. The application of our method is on a task that requires the mobile manipulator to maintain a constant force perpendicular to the unknown surface of the canopy, while the polishing tool covers the entire surface to be polished.

Typically, velocity controllers operate at a slower rate, (eg. 200Hz), compared to the internal servo (500-1000Hz) at which the torque control is operating. This slower update rate has been known to affect performance. The system can be thought of as a Macro/Mini system [83] [3].

8.2.1 Combined Torque and Velocity Control for the Overall System

A mobile manipulator often consists of many joints, and often some are of torque control (typically, the arm) and some of velocity control (typically, the base). In fact, to the best of our knowledge, we are not aware of any commercially available torque-controlled mobile base today. Here a method of combining torque and velocity control is presented which is consistent with the dynamics of the entire (eg. arm and base) system. This approach utilises the force/position duality with extension to the case of redundant manipulators as shown in Table 8.1. This equation has been shown in Chapter 2, and is reproduced here for convenience.

The approach uses the dynamically consistent inverse, as introduced in Section 2.5.1 which guarantees the position/force duality when used in torque control. This property of duality as shown in Table 8.1, provides a method for us to control the velocity-controlled base in a way which is similar to the RMRC method by [84]. When

Table 8.1: The position/force duality (reproduced)

Position	Force
$\delta\mathbf{q} = \bar{\mathbf{J}}\delta\mathbf{x} + [\mathbf{I} - \bar{\mathbf{J}}\mathbf{J}]\delta\mathbf{q}_0$	$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{f} + [\mathbf{I} - \bar{\mathbf{J}}\mathbf{J}]^T\boldsymbol{\tau}_0$

used in velocity control, there will be error due to the slower and less accurate macro structure (the base), however, this is compensated by the faster and more accurate mini structure (the arm).

The combined manipulator arm and the mobile base system can now be regarded as an n DOF system, where n is the total number of joints that the two systems have combined. The Jacobian matrix generated is of size $m \times n$, where m is the number of DoF of the specified task.

Treating the combined system as an n DOF robot, a torque command vector $\boldsymbol{\tau}$ of size n and velocity command vector $\delta\mathbf{q}$, also of size n are both generated from the position/force duality tabulated in Table 8.1 to describe the desired end-effector motion trajectory and contact forces.

Since the arm is in torque control and the base in velocity control, the appropriate command for each joint is then sent accordingly. For example, let's take a combined system where the base makes up the first 3 joints, and the arm the last 6. From the resulting 9×1 velocity command vector $d\mathbf{q}$, the first 3 elements are sent to the base. Similarly, the last six elements of the torque command vector $\boldsymbol{\tau}$ are sent to the arm (see Figure 8.1).

A mobile manipulator has typically redundant joints with respect to the task. It is then necessary to control the behaviour of the manipulator posture. Many algorithms have been proposed in the past, such as those by [30, 64, 85, 17, 86].

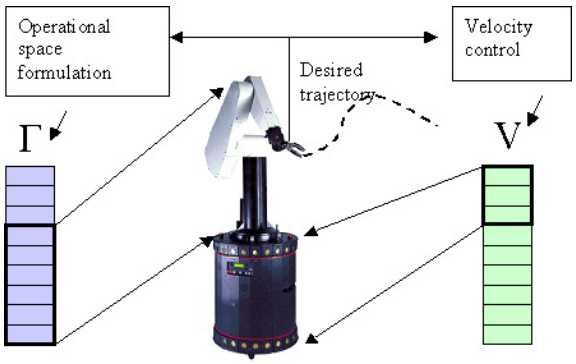


Figure 8.1: Torque vector from operational space formulation and velocity vector from velocity control, are both generated to satisfy the desired trajectory. The commands are sent to the corresponding joints.

In this application, potential functions were constructed to have minima at the desired manipulator redundant behaviour. The dynamically consistent inverse $\bar{\mathbf{J}}$ as in the equations in Table 8.1 ensures that the projection of the gradient descent of the potential function onto the null space of $\mathbf{J}$ will not interfere dynamically with the end-effector task. The potential functions and the null space torque are specified in the same way as described in earlier chapters.

8.3 Application to Aircraft Canopy Polishing

The proposed method was implemented on a torque-controlled PUMA 560 manipulator arm and a velocity-controlled Nomad XR4000 mobile base, as shown in Figure 8.1.

In this implementation, the omnidirectional Nomad XR4000 base is modelled as having 3 (planar) DoFs, namely translation along X and Y and rotation around Z axes. This is done because the control to the mobile base is available in high

level control (velocity control). The NOMAD actually has 4 wheel modules, with 2 actuators each, one for driving and one for steering [87].

The overall system is therefore thought of as a 9 jointed mechanism, with the base providing the first three velocity-controlled joints (two prismatic and one revolute), and the arm the last six torque-controlled joints (see Figure 8.2).

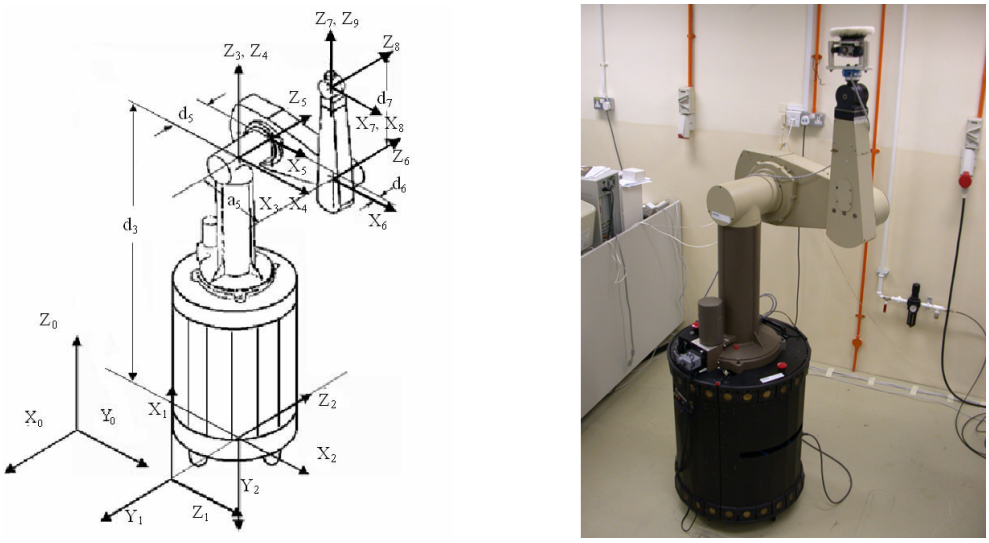


Figure 8.2: Frame assignment for the integrated arm-base system.

From this point onward, "joint 1", "joint 2", and "joint 3" would refer to the first three joints of the overall system, supplied by the base, and "joint 4" to "joint 9" would refer to joint 1 to 6 of the PUMA arm respectively.

The resulting Jacobian is of size 6×9. The resulting joint torque command and joint velocity command vectors are of size 9 by 1. The first three elements of the velocity vector are sent as joint velocity commands to the mobile base, while the last 6 elements of the torque vector to the PUMA arm.

In dynamic compensation, the full dynamic model is employed for the PUMA arm, while only a simplified model is used for the NOMAD base. The inertia matrix **A** used for the NOMAD base is only modelled as having diagonal terms.

A PD control law is used for the motion control and PI for force control. $\mathbf{K}_p$ and $\mathbf{K}_v$ gains were chosen to result in a slightly underdamped closed loop system with natural frequency $\omega_n = 45 rad/s$.

Force readings were taken with a JR3 FT (force torque) sensors, with a low pass filter. PUMA and the NOMAD base were controlled by separate computers, linked by a high speed TCP/IP connection. Control calculation is done on the processor controlling the PUMA arm. The velocity command vector $\delta\mathbf{q}$ is sent over to the NOMAD controller as high level velocity command.

8.4 On the Issue of Singularity Handling

This time, instead of adding virtual joints to the robot, we can observe the effect when real physical joints were added to the manipulator. The extra DoFs can now be utilised to supplement the DoFs lost during the manipulator arm's singularities. The following subsections compute the singular configuration of the arm and base system.

8.4.1 Position Singularity

The Jacobian matrix of the Arm-and-Base system is shown in Appendix B3.

As the mobile base is now capable of supplying motion in X and Y direction (modelled as prismatic joints), the manipulator can only have a degenerate direction in Z_0 direction, which is the boundary singularity when the arm reaches its maximum reach in Z_0 direction. It can also be considered that ${}^0\mathbf{J}_V$ has leading "1" in the matrix in the first two columns. For the motion in Z_0 to be degenerate, it is necessary that

$$(a_5c_5 + a_6c_{56} + d_7s_{56}) = 0 \tag{8.1}$$

and

$$(a_6c_{56} + d_7s_{56}) = 0 \tag{8.2}$$

Substituting (8.2) into (8.1), we have $a_5c_5 + 0 = 0$, therefore $c_5 = 0$, $q_5 = \pm 90^o$.

For $(a_6c_{56} + d_7s_{56}) = 0$, then $q_5 + q_6 = -2.68^o$.

So for position singularity to occur: $q_5 = \pm 90^o$ AND $q_5 + q_6 = 2.68^o$.

This is the position when the PUMA arm is pointing vertically. The -2.68^o is the small angle caused by the offset at the elbow of the PUMA. The degenerate direction in the translation along Z_0.

8.4.2 Orientation Singularity

When expressed in Frame{7}, which is originally Frame{4} on the stand-alone PUMA arm, the lower half of the Jacobian matrix becomes:

$${}^7\mathbf{J}_\omega = \begin{pmatrix} 0 & 0 & -c_7s_{56} & -c_7s_{56} & s_7 & s_7 & 0 & 0 & s_8 \\ 0 & 0 & s_7s_{56} & s_7s_{56} & c_7 & c_7 & 0 & 1 & 0 \\ 0 & 0 & c_{56} & c_{56} & 0 & 0 & 1 & 0 & c_8 \end{pmatrix} \tag{8.3}$$

and $\mathbf{J}_{22}$ is rank deficient when $s_8 = 0$ and the resulting $\mathbf{J}_\omega$ is:

$${}^7\mathbf{J}_\omega = \begin{pmatrix} 0 & 0 & -c_7s_{56} & -c_7s_{56} & s_7 & s_7 & 0 & 0 & 0 \\ 0 & 0 & s_7s_{56} & s_7s_{56} & c_7 & c_7 & 0 & 1 & 0 \\ 0 & 0 & c_{56} & c_{56} & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{8.4}$$

Again, we are faced with the question whether identifying the condition where $\mathbf{J}_\omega$ is rank-deficient is enough to identify the complete set of solution. $\mathbf{J}_\omega$ is rank-deficient when $s_8 = 0$ AND $s_7 = 0$ AND $s_{56} = 0$.

The singular direction for the spherical wrist is the rotation around X axis of Frame {7}, which is the reflected as the first row of ${}^7\mathbf{J}_\omega$. Following the method

proposed in Section 4.7, it is necessary to check all the terms on the first row of ${}^{7}\mathbf{J}_{\omega}$ in Equation 8.4. The possible conditions of singularities are:

- $s_8 = 0$ AND $s_7 = 0$
- $s_8 = 0$ AND $s_{56} = 0$
- $s_8 = 0$ AND $c_7 = 0$
- $s_8 = 0$ AND $s_7 = 0$ AND $s_{56} = 0$

It was found, by substituting these expression into the Jacobian matrix and evaluating the determinant, that only when ($s_8 = 0$ AND $s_7 = 0$ AND $s_{56} = 0$) the manipulator in orientation singularity, which is the same as identifying the condition when $\mathbf{J}_{\omega}$ is rank-deficient.

This singular configuration happens when the wrist of PUMA is straight and forearm of the PUMA arm is vertical and the first joint of the PUMA wrist is at zero position. The singular direction is the rotation around X_7 (see Figure 8.3).

Comparing the usable workspace of the mobile manipulation system with that of a stand-alone PUMA, we can see a significant reduction in the amount of workspace affected by singularity.

8.5 Experimental Setup and Result

Three sets of experimental result is shown in this paper, which analyse the performance of the combined arm-base system in free motion, constrained motion, and in an industrial application involving a canopy polishing task (see Figure 8.4).

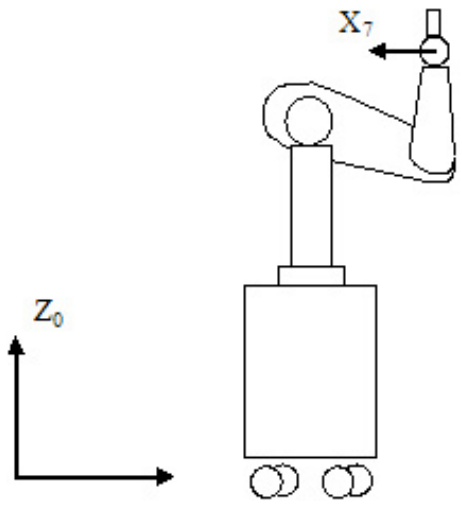


Figure 8.3: Sketch of planar view of the arm-base system. The degenerate direction is the rotation around X_7. There is no joints in the robot that can provide this degree of freedom.

8.5.1 Free motion

This task requires the mobile base to move in an elliptical trajectory, while maintaining the end-effector stationary with respect to the world frame. The error in motion tracking is shown as the result. Motion of the base while maintaining a stationary end-effector can be thought of as internal joint motion of the arm-base system. This motion is created by projecting the motion of the base into the null space of the Jacobian. The desired null space configuration $q_{i(desired)}$ for joints 1 and 2 (the translational DOF of the base) is made to follow a trajectory.

The error in motion trajectory tracking is shown in Figure 8.5. The result presents the motion tracking performance of the arm-base system. It is required to keep the end-effector stationary while the base moves in an ellipse, with major axis 40 cm and

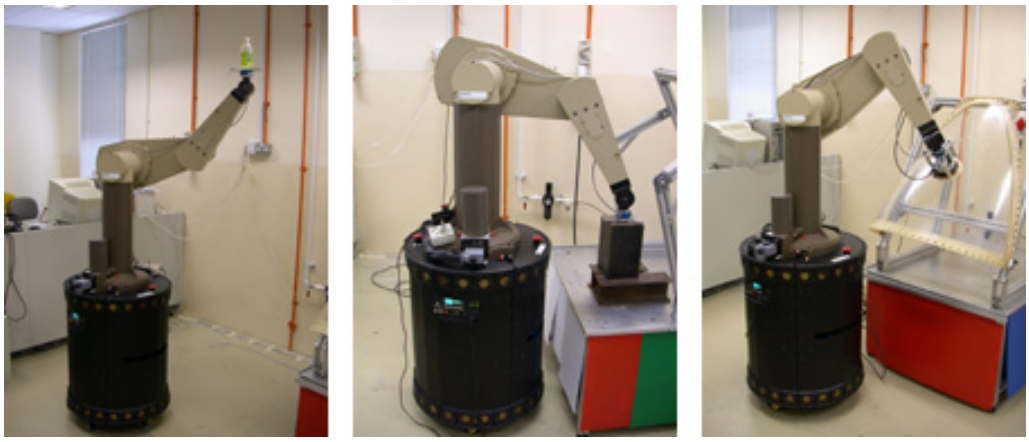


Figure 8.4: The three experiment setups: (left) maintaining stationary end-effector while the base moves in an elliptical trajectory (center) maintaining a normal force (stationary end-effector) with a moving base, and (right) polishing task, maintaining constant 10 N force normal to unknown surface with sinusoidal end-effector motion, with moving base.

minor axis 14 cm. The low speed setting was for the mobile base to complete the ellipse in 30s, and high speed in 10s, as shown in the lower graphs of Figure 8.5.

This graph shows that while velocity control of the base has been theoretically compensated for the dynamics of the structure, it is still affected by the disturbance of the base motion at high speed. Therefore, for high speed dynamic compensation, torque control architecture is still necessary.

The result only shows the tracking performance in X axis because the elliptical trajectory of the base motion has the major axis along X, and error is most prominent in this axis. Z axis has very little error because it is decoupled from the motion of the base.

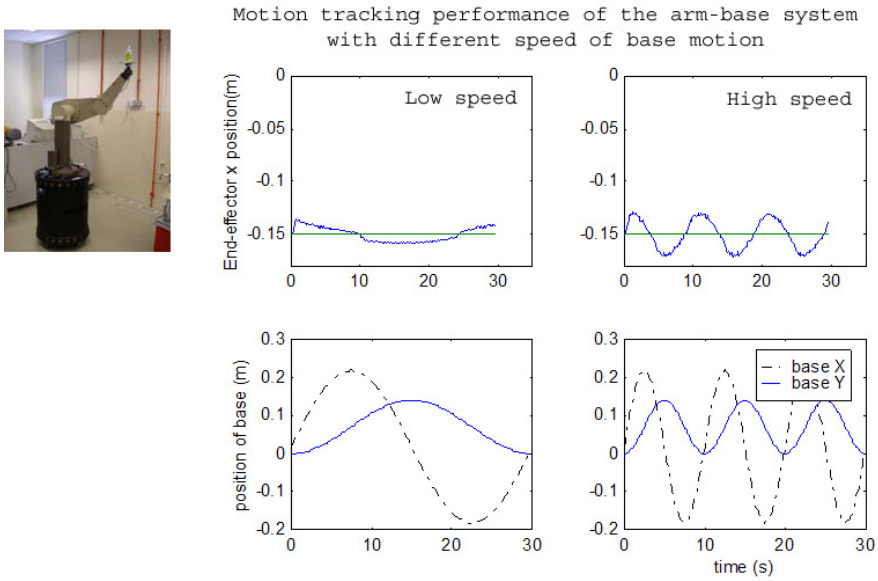


Figure 8.5: The motion tracking performance of the arm-base system. The mobile base was required to move in an elliptical trajectory of 40cm major axis and 14cm minor axis. The desired X position is -15cm. Tracking is shown with the mobile base moving in low speed (left) and high speed (right)

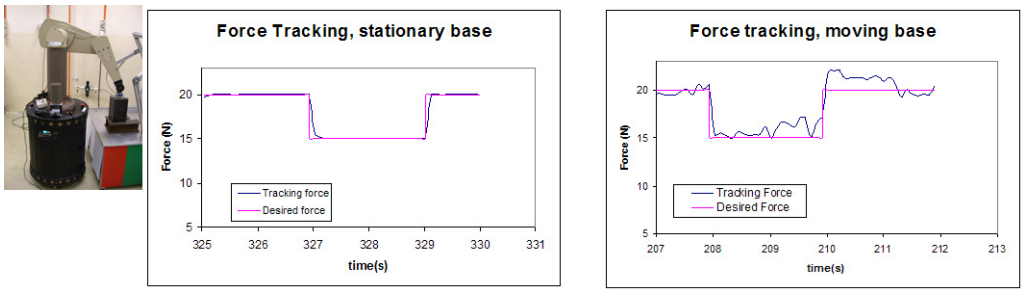


Figure 8.6: The system was required to perform force control by exerting a normal downward force to track the desired force trajectory. (left) force tracking performance of a stand alone (stationary) PUMA arm and (right) that of arm-base system with moving base.

The result in Figure 8.5 shows that the error is contained within $\pm$ 1 cm when the base moves in low speed and barely within 2 cm at high speed. It is hard to compare this performance with that of PUMA stand-alone motion tracking.

The video to this experiment can be found in:

http://guppy.mpe.nus.edu.sg/~mpeangh/robotics/water_bottle2.mpg

8.5.2 Constrained motion

This experiment compares the result of the force control ability of the system while the base is stationary and while the base is moving. The task was to track a desired force trajectory (against a surface) that switches between 20N and 15 N every 2 seconds while maintaining the end-effector position. Just like in Section 5.1, the motion of the base is created by projection to the null space of the Jacobian.

Figure 8.6 shows the performance of force control with a stationary base (left) compared to that with moving base (right). The error in force control at moving base

is shown to be within ±2.5N. The effect of dynamic disturbance of the mobile base of the performance of the arm is further discussed in [43, 42].

8.5.3 Canopy polishing

This task requires the system to maintain a normal force of 10N against an unknown surface, while the end-effector moves in a sinusoidal manner. The mobile manipulator automatically complies to the unknown canopy geometry while trying to maintain a constant force of 10 N normal to the canopy surface. The result compares the performance of the system with a stationary and with a moving base.

In this experiment, the main emphasis is placed not on force control, but on the singularity issue and the combining of the arm and the base, which are of different control modes. The operational space formulation is used to perform the unified force/motion control. Other methods can be utilized too, such as reviewed in [88]. Background in force control strategy can be found in [89] for the classic impedance control, [90, 91, 92, 93] on hybrid position and force control and [94, 95] for other force control strategies and fundamental works.

The operational point is chosen at the end of the polishing tool. Motion control is applied to the translation along X and Y axes, and the rotational around the Z axis of the tool frame. The 10N force exerted normal to the unknown canopy surface is along the Z axis of the tool frame. Moment control is applied to the other two remaining DoFs, namely rotation around X and Y of the tool frame. Maintaining the moments around these two axes at zero creates the compliant motion of the polishing tool to follow the contour of the polished surface [80].

In this experimental run, internal joint motion was set so that the mobile base always faces the end-effector. This is to prevent the PUMA arm from hitting the limit of its first joint and to increase the system manipulability [6].

The potential function used was defined as:

$$v_0(\mathbf{q}) = k_d(q_4 - q_{4,desired})^2; \tag{8.5}$$

where $q_{4-desired} = 0$. This means we would like to keep joint 4 (first joint of PUMA arm) to be at the center of its range. The projection of the gradient descent of this potential function onto the null space would cause the base to move accordingly.

The performance of the system with combined arm and base was found to be almost the same for stationary and moving base separately, therefore the graph in Figure 8.7 only shows the performance of the system with moving base. The force distribution (with the desired value being 10N) has a spread of ±2 N in both cases, and motion tracking error within 7 cm.

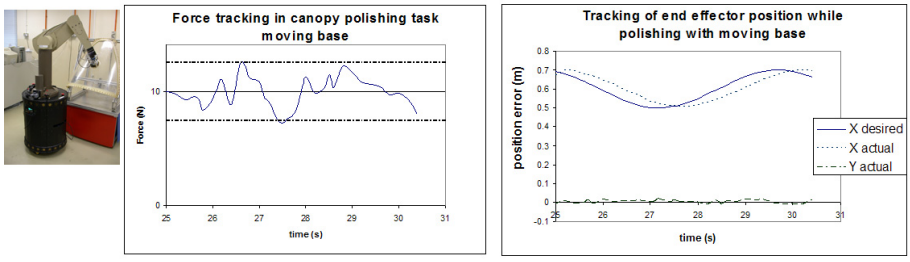


Figure 8.7: The system is required to perform sinusoidal tool motion maintaining 10N force normal to the unknown surface. The graph shows the error response of the mobile manipulator in force (right)) and position (right) tracking with moving and stationary base

When the tool moves sinusoidally while maintaining contact with the surface of the canopy, disturbance introduced by the friction with the surface seemed to overwhelm the error introduced by the dynamic disturbance of the base motion [80]. A friction compensation was implemented to reduce the effect.

As shown in Section 8.4, wrist singularity has been reduced to only when $s_8 = 0$ AND $s_7 = 0$ AND $s_{56} = 0$. When the system was performing the polishing task, it was able to move across the straight wrist configuration in a stable manner (Figure 8.8).

The polishing motion can be seen from the video in
http://guppy.mpe.nus.edu.sg/~mpeangh/robotics/polishing-base.mpg

While the polishing motion of the end-effector while going to a wrist singularity can be found in: http://guppy.mpe.nus.edu.sg/~mpeangh/robotics/polishing-wrist-singularity.mpg

8.6 Conclusion

The chapter presents an implementation result of the operational space formulation in a mobile manipulation task, where a torque controlled arm was combined with a velocity controlled base. The integration method was presented. It is also an extension of the idea in Chapter 7 of supplying the manipulator with extra DoF to compensate for the lost DoF at singular configuration. This time, however, the joints are real, in the form of a mobile base. Results of the experiments were presented.

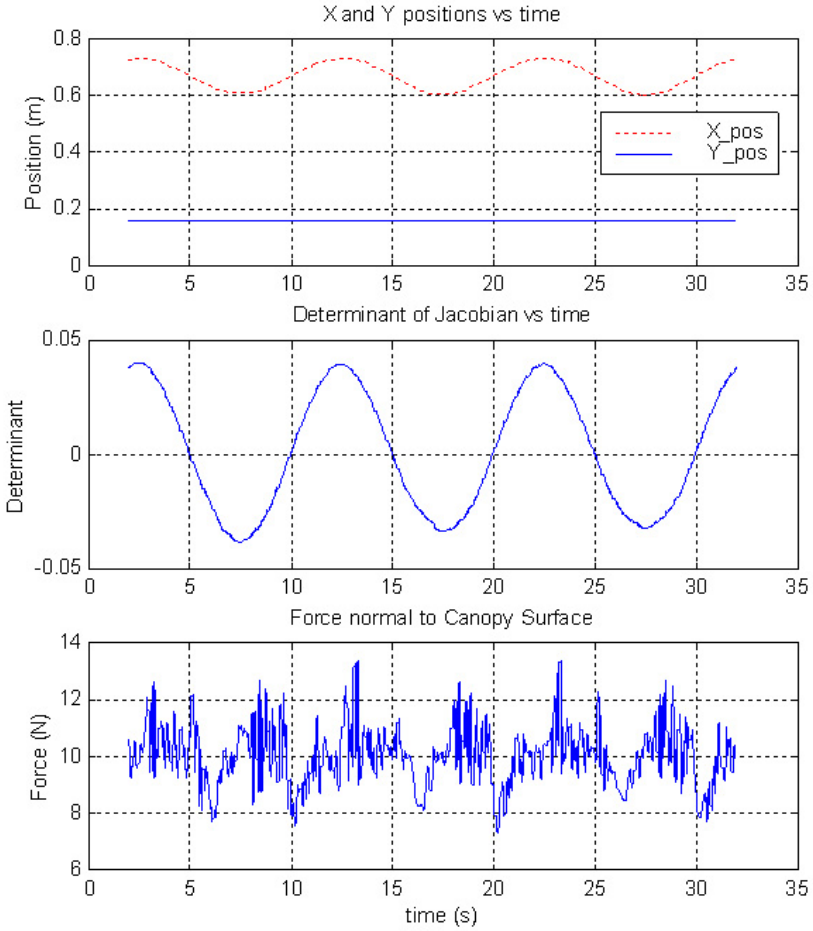


Figure 8.8: The performance of the system in polishing experiment as the manipulator was set to cross the configuration where the wrist was straight. The middle plot shows the determinant of the Jacobian of just the PUMA arm without the base. The system can now move across what used to be wrist singularity without any problem.

CHAPTER 9

CONCLUSION

This thesis concentrated mainly on the issue of *singularity*. Various issues were presented in Chapter 4 on the identification of singularities and singular direction. Completeness of solution was analysed and a simple check was proposed to eliminate possibility of incomplete solution for the case of serial manipulator with spherical wrist. Singular directions are usually easy to identify by observation, when the singular configurations have been identified. It was also highlighted that it is possible for a family of singular configurations to have different number of lost DoFs. This means that even when a manipulator has lost one DoF, at certain configurations which form a subset of the singularity it is in, it can still lose more rank. This should be taken into consideration when designing algorithms to handle singularities in a manipulator.

In the singularity handling method of removing degenerate components, null space motion was utilized to create motion to assist the end-effector in escaping singularity into degenerate direction. The reduction of degrees of freedom within the singular region causes accumulated error in the singular direction as motion in these directions is disabled. This would cause jerkiness as the end-effector tries to close the large accumulated tracking error as soon as it leaves the singular region and regains its full motion control. Methods were suggested to handle such situations.

In the method of virtual joint, the DoFs lost in the singularity was regained with extra joints that only exist computationally. The joints don't exist physically and therefore the manipulator is still not able to move in the degenerate direction. The method is then extended as the manipulator arm was mounted on a mobile base. The mobile base, capable of 3DoF planar motion was thought of as the extra joints supplied to the system. In Chapter 8, it was shown that the workspace affected by singularity was reduced significantly. The system was designed to perform mobile manipulation task such as compliant motion in canopy polishing task. The robot was to maintain a constant contact force normal to the surface of the canopy that is unknown to the robot. The result of the experiment was presented. Several videos of the experimental results can be found in:

- http://guppy.mpe.nus.edu.sg/ mpeangh/robotics/ example-polishing-force-data.mpg
- http://guppy.mpe.nus.edu.sg/ mpeangh/robotics/ polishing-wrist-singularity.mpg
- http://guppy.mpe.nus.edu.sg/ mpeangh/videos/waterbottle_2.mpg

Future Work

Although singularity is a well researched topic that has been around for a long time, there is still room for further refinement in this field. Many experts are often quoted to say that this was a hot topic "10-15 years ago" in their review. It is the author's belief that the focus in handling singularity should be directed towards the robustness of the algorithms in the real-time implementation.

For example: the issue of discontinuity at the boundary of singular region. There are not many literature at this point that address the issue in great details. It is often thought of as a secondary problem to the task of maintaining a stable trajectory around the singular configuration, and that it can be handled by introducing some damping terms. However, as robots get smaller in size and lighter, this could become a more serious problem. Works in micro or nano scale robots can not tolerate much of a jerkiness, for example.

In a more general view of mobile manipulation, it can be said that most of the fundamental theories are available to realise a decent experimental setup. However, it is often still very difficult to realise the system in a robust manner that would enable the technology to be deployed in a real human-interactive environment. The reliability and safety of the technology should be the main focus of the development effort. Experimental setup often works fine within the testing environment, however the stability windows are often not large enough to handle the uncertainty in the human (unstructured) environment.

APPENDIX A

FRAME ASSIGNMENTS

A.1 Frame Assignment for PUMA (stand-alone)

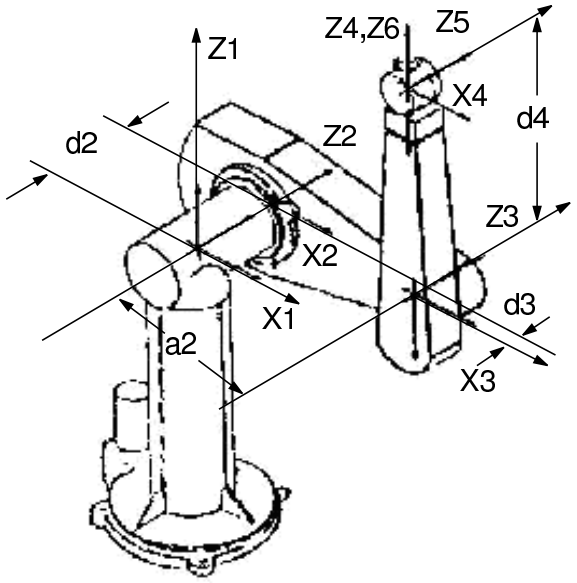


Figure A.1: Frame Assignment for PUMA 560 in the experiment, when used alone (without the mobile base).

The numerical values for the Denavit-Hartenberg parameters used are: a_2=0.4318 m, a_3=-0.0203 m, d_2=0.2435 m, d_3=-0.0934, d_4=0.4331m.

Table A.1: The modified DH parameters for PUMA manipulator (stand-alone as in [2]

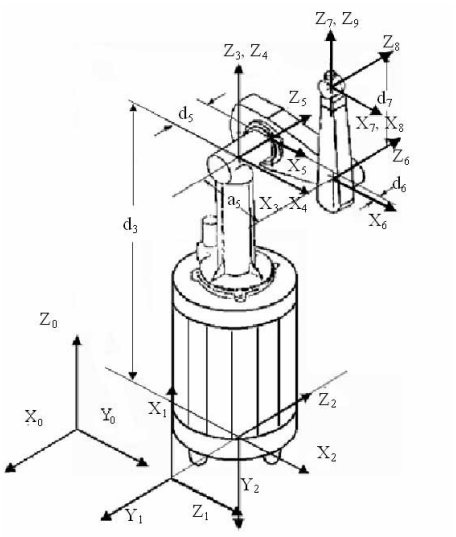
i	α_{i-1}	a_{i-1}	d_i	ϑ_i
1	0	0	0	ϑ_1
2	-90	0	d_2	ϑ_2
3	0	a_2	d_3	ϑ_3
4	90	a_3	d_4	ϑ_4
5	-90	0	0	ϑ_5
6	90	0	0	ϑ_6

A.2 Frame Assignment for PUMA-NOMAD system

The parameters used in the Denavit-Hartenberg Convention are: $d_3 = (m)$, $a_5 = 0.4318(m)$, $a_6 = -0.0203(m)$, $d_5 = 0.2435(m)$, $d_6 = -0.0934(m)$, and $d_7 = 0.4331(m)$.

Table A.2: The modified DH parameters for PUMA mounted on Nomad mobile bases system

i	α_{i-1}	a_{i-1}	d_i	ϑ_i
1	-90	0	d_1	-90
2	90	0	d_2	90
3	90	0	d_3	ϑ_3
4	0	0	0	ϑ_4
5	-90	0	d_5	ϑ_5
6	0	a_5	d_6	ϑ_6
7	90	a_6	d_7	ϑ_7
8	-90	0	0	ϑ_8
9	90	0	0	ϑ_9



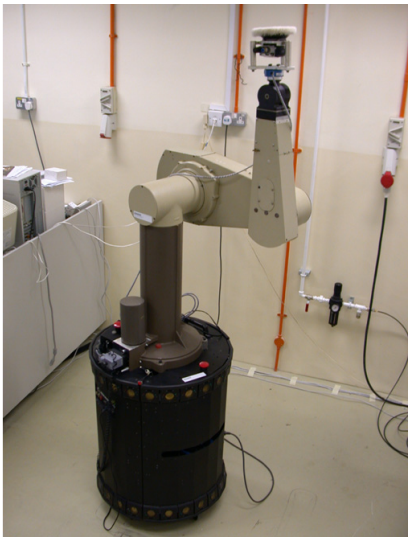


Figure A.2: This is the frame assignment for the Arm-Base System used in the experiment, involving the PUMA 560 Arm mounted on top of Nomadic XR4000 mobile robot.

APPENDIX B

JACOBIAN MATRIX

B.1 Jacobian Matrix for PUMA (stand-alone)

$$J_1 = \begin{pmatrix} -C_1(d_2+d_3) - S_1(a_2C_2 + a_3C_{23} + d_4S_{23}) \\ -S_1(d_5+d_6) + C_1(a_2C_2 + a_3C_{23} + d_4S_{23}) \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{B.1}$$

$$J_2 = \begin{pmatrix} C_1(C_{23}d_4 - a_2S_2 - a_3S_{23}) \\ S_1(C_{23}d_4 - a_2S_2 - a_3S_{23}) \\ -(a_2C_2 + a_3C_{23} - d_4S_{23}) \\ -S_1 \\ C_1 \\ 0 \end{pmatrix} \tag{B.2}$$

$$J_3 = \begin{pmatrix} C_1(C_{23}d_4 - a_3S_{23}) \\ S_1(C_{23}d_4 - a_3S_{23}) \\ -(a_3C_{23} + d_4S_{23}) \\ -S_1 \\ C_1 \\ 0 \end{pmatrix} \tag{B.3}$$

$$J_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ C_1S_{23} \\ S_1S_{23} \\ C_{23} \end{pmatrix} \tag{B.4}$$

$$J_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -C_4S_1 - C_1C_{23}S_4 \\ C_1C_4 - C_{23}S_1S_4 \\ S_{56}S_7 \end{pmatrix} \tag{B.5}$$

$$J_9 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -S_1S_4S_5 + C_1(C_5S_{23} + C_{23}C_4S_5) \\ C_5S_1S_{23} + (C_{23}C_4S_1 + C_1S_4)S_5 \\ C_{23}C_5 - C_4S_{23}S_5 \end{pmatrix} \tag{B.6}$$

B.2 Jacobian Matrix for Example Manipulator in Section 3.2.1

Below is the Jacobian of the 7 DOF Manipulator used as an example in Section 3.2.1. The diagram is reproduced below in Figure B.1 (right).

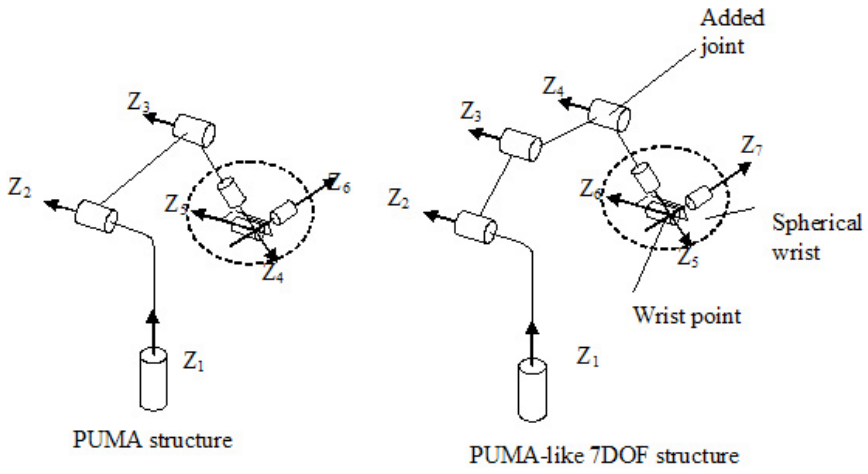


Figure B.1: Structure of the PUMA 6 DOF(left), and an example of a 7 DOF PUMA-like manipulator with spherical wrist (right). This manipulator is used as an example in Section 3.2.1.

The Jacobian is given expressed in Frame{4} where it is in its simplest form.

$$ {}^5J = \begin{pmatrix} J_{11} & \mathbf{0}_{3\times 3} \\ J_{21} & J_{22} \end{pmatrix} \tag{B.7} $$

$$ {}^5J_{11} = \left({}^5J_{11a} | {}^5J_{11b}\right) \tag{B.8} $$

$$ {}^4J_{11a} = \begin{pmatrix} (a_2C_2 + a_3C_{23} + d_5S_{234})S_5 - C_{234}((d_2+d_4)C_5 - a_4S_5) & C_5(d_5 + a_3S_4 + a_2S_{34}) \\ C_5(a_2C_2 + a_3C_{23} + a_4C_{234} + d_5S_{234}) + (d_2+d_4)C_{234}S_5 & -(d_5 + a_3S_4 + a_2S_{34})S_5 \\ -(d_2+d_4)S_{234} & -a_4 - a_3C_4 - a_2C_{34} \end{pmatrix} \tag{B.9} $$

$$ {}^4J_{11b} = \begin{pmatrix} C_5(d_5 + a_3S_4) & d_5C_5 \\ -(d_5 + a_3S_4)S_5 & -d_5S_5 \\ -a_4 - a_3C_4 & -a_4 \end{pmatrix} \tag{B.10} $$

$$ [{}^4J_{21} | {}^4J_{22}] = \begin{pmatrix} -C_5S_{234} & S_5 & S_5 & S_5 & 0 & 0 & S_6 \\ -S_5S_{234} & C_5 & C_5 & C_5 & 0 & 1 & 0 \\ C_{234} & 0 & 0 & 0 & 1 & 0 & C_6 \end{pmatrix} \tag{B.11} $$

B.3 Jacobian Matrix for PUMA-NOMAD System

$$ J_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{B.12} $$

$$ J_2 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{B.13} $$

$$ J_3 = \begin{pmatrix} S_{34}(d_5+d_6) - C_{34}(a_5C_5 + a_6C_{56} + d_7S_{56}) \\ -C_{34}(d_5+d_6) - S_{34}(a_5C_5 + a_6C_{56} + d_7S_{56}) \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{B.14} $$

$$J_4 = \begin{pmatrix} S_{34}(d_5+d_6) - C_{34}(a_5C_5 + a_6C_{56} + d_7S_{56}) \\ -C_{34}(d_5+d_6) - S_{34}(a_5C_5 + a_6C_{56} + d_7S_{56}) \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{B.15}$$

$$J_5 = \begin{pmatrix} -S_{34}(C_{56}d_7 - a_5S_5 - a_6S_{56}) \\ C_{34}(C_{56}d_7 - a_5S_5 - a_6S_{56}) \\ -(a_5C_5 + a_6C_{56} + d_7S_{56}) \\ -C_{34} \\ -S_{34} \\ 0 \end{pmatrix} \tag{B.16}$$

$$J_6 = \begin{pmatrix} -S_{34}(C_{56}d_7 - a_6S_{56}) \\ C_{34}(C_{56}d_7 - a_6S_{56}) \\ -(a_6C_{56} + d_7S_{56}) \\ -C_{34} \\ -S_{34} \\ 0 \end{pmatrix} \tag{B.17}$$

$$J_7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -S_{34}S_{56} \\ C_{34}S_{56} \\ C_{56} \end{pmatrix} \tag{B.18}$$

$$J_8 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -C_{34}C_7 + C_{56}S_{34}S_7) \\ -C_7S_{34} - C_{34}C_{56}S_7 \\ S_{56}S_7 \end{pmatrix} \tag{B.19}$$

$$J_9 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -C_8S_{34}S_{56} - (C_{56}C_7S_{34} + C_{34}S_7)S_8 \\ C_8S_{34}S_{56} + (C_{56}C_7S_{34} + C_{34}S_7)S_8 \\ C_{56}C_8 - C_7S_{56}S_8 \end{pmatrix} \tag{B.20}$$

BIBLIOGRAPHY

[1] Oussama Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics and Automation*, vol. RA-3, no. 1, pp. 43–53, 1987.

[2] John J. Craig, *Introduction to Robotics, Mechanics and Control*, Addison-Wesley, 2nd edition, 1989.

[3] Oussama Khatib, "Advanced robotics," Lecture Notes, 1997, Stanford University.

[4] K. Chang and O. Khatib, "Manipulator control at kinematic singularities: A dynamically consistent strategy," *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, vol. 3, pp. 84–88, 1995.

[5] Y. Nakamura and H. Hanafusa, "Inverse kinematics solutions with singularity robustness for robot manipulator control," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.

[6] Tsuneo Yoshikawa, "Manipulability of robotic mechanisms," *Intl. J. Robotics Research*, vol. 4, no. 2, pp. 3–9, Summer 1985.

[7] K. S. Fu, R. C. Gonzales, and C. S. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, Industrial Engineering Series. McGraw-Hill, 1987.

[8] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, 1990.

[9] Yoshihiko Nakamura, *Advanced Robotics - Redundancy and Optimization*, Addison-Wesley Series in Electrical and Computer Engineering: Control Engineering. Addison-Wesley, 1991.

[10] M.W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, NewYork: Wiley, 1989.

[11] J.M. McCarthy, *An Introduction to Theoretical Kinematics*, The MIT Press, 1990.

[12] Rodrigo Jamisola, Marcelo H. Ang, Tao Ming Lim, and Ser-Yong Lim, "Dynamics identification and control of an industrial robot," *The Ninth Intl. Conf. On Advanced Robotics*, pp. 323–328, 1999.

[13] J. Burdick B. Armstrong, O. Khatib, "The explicit dynamic model and inertial parameters of the puma 560 arm," *IEEE Intl. Conf. Robotics and Automation*, pp. 510–518, 1986.

[14] P.I. Corke, "A robotics toolbox for matlab," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24–32, March 1996.

[15] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.

[16] Rodrigo Jamisola, "Dynamics identification and control of an industrial robot," Master of Engineering, Thesis, 2001, National University of Singapore.

[17] Tsuneo Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics Research*, M. Brady and R. Paul, Eds., pp. 735–747. MIT Press, Cambridge, MA, 1984.

[18] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," *Proc. 8th Triennial IFAC World Congress*, vol. 14, pp. 78–83, 1981.

[19] Y. Nakamura and H. Hanafusa, "Task priority based redundancy control of robot manipulators," in *Second Intl. Symp. of Robotics Research*, H. Hanafusa and H. Inoue, Eds., pp. 155–162. MIT Press, 1985.

[20] P. H. Chang, "A dexterity measure for kinematic control of redundant manipulator," *Proc. American Control Conference*, pp. 496–502, June 1989.

[21] P.H. Chang, "A closed-form solution for inverse kinematics of robot manipulators with redundancy," *IEEE J. Robotics and Automation*, vol. RA3, no. 5, pp. 393–403, 1987.

[22] H. Zghal, R.V. Dubey, and J.A. Euler, "Efficient gradient projection optimization for manipulators with multiple degrees of redundancy," *Proc. IEEE Intl. conf. Robotics and Automation*, vol. 2, 1990.

[23] J. H. Park, W. K. Chung, and Y. Youm, "Behaviours of extended jacobian method for kinematic resolutions of redundancy," *Proc. IEEE Intl. Proc. on Robotics and Automation*, pp. 89–95, 1994.

[24] J. H. Park, W. K. Chung, and Y. Youm, "Characteristics of optimal solutions in kinematic resolutions of redundancy," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 471–478, June 1996.

[25] M.Z. Huang and H. Varma, "Optimal rate allocation in kinematicaly redundant manipulators - the dual projection method," *Proc. IEEE Intl. conf. Robotics and Automation*, vol. 1, pp. 702–707, April 1991.

[26] J. Bailieul, "Avoiding obstacles and resolving kinematic redundancy," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1698–1704, 1986.

[27] Yu-Che Chen and Ian D. Walker, "A consistent null-space based approach to inverse kinematics of redundant robots," *Proc. IEEE Intl. Conf. of Robotics and Automation*, vol. 3, pp. 374–381, 1993.

[28] K.A. ONeil, Yu-Che Chen, and JiaQing Sing, "On the existence and characteristics of solution paths at algorithmic singularities," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 336–342, April 1998.

[29] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multi-body mechanism," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 7, pp. 868–871, 1977.

[30] C.A. Klein and C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Sys., Man., Cyber.*, vol. SMC-13, no. 3, pp. 245–250, 1983.

[31] C.A. Klein, "Use of redundancy in design of robotics systems," *Intl. Journal of Robotic Research*, vol. 2, 1985.

[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1989.

[33] G. Strang, *Linear Algebra amd Its Application*, New York: Academic, 2nd edition, 1980.

[34] Oussama Khatib, "The augmented object and reduced effective inertia of robot system," *Proc. American Control Conference*, pp. 2140–2147, 1988, Atlanta.

[35] R. Featherstone and O. Khatib, "Load-independence of the dynamically-consistent inverse of the jacobian matrix," *International Journal of Robotics Research*, vol. 16, no. 2, pp. 168–170, 1997.

[36] Oussama Khatib, "Motion/force redundancy of manipulators," *Proc, Japan-USA Symposium on Flexible Automation, Kyoto Japan*, pp. 337–342, 1990.

[37] K.C. Park, P.H. Chang, and S. Lee, "A new kind of singularity in redundant manipulation: Semi algorithmic singularity," *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 1979–1984, May 2002.

[38] J.W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifold," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 264–270, 1989.

[39] K.J. Waldron, S.L. Wang, and S.J. Bolin, "A study of the jacobian matrix of serial manipulators," *J. Mechanisms, Transmissions, and Automation in Design*, vol. 107, pp. 230–238, 1985.

[40] C. L. Luck and S. Lee, "Self-motion topology for redundant manipulators with joint limits," *Proc. IEEE International Conference on Robotics and Automation*, pp. 626–631, 1993.

[41] C. L. Luck and S. Lee, "Global path planning of redundant manipulators based on self motion topology," *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 510–517, 1994.

[42] Robert Holmberg, "Design and development of powered-caseter holonomic mobile robots," PhD. Dissertation, August 2000, Stanford University.

[43] R. Holmberg and O. Khatib, "Development and control of a holonomic mobile robot for mobile manipulation tasks," *International Journal of Robotics Research*, vol. 19, no. 11, pp. 1066–1074, 2000.

[44] Byung-Ju Yi and Whee-Kuk Kim, "The kinematics for redundantly actuated omnidirectional mobile robots," *Journal of Robotic Systems*, vol. 19, no. 6, pp. 255–267, 2002.

[45] Guy Campion, Geore Bastin, and Brigitte D'Andrea-novel, "Structural properties and clasification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, February 1996.

[46] P.F. Muir and C. P. Newman, "Kinematic modelling of wheeled mobile robots," *Journal of Robotic Systems*, vol. 4, no. 2, pp. 281–340, 1987.

[47] S. Ostrovskaya and J. Angeles, "Non-holonomic systems revisited within the framework of analytical mechanics," *Appl. Mech. Rev.*, vol. 51, no. 7, pp. 415–431, July 1998.

[48] F.T. Cheng, J.S. Chen, and F.C. Kung, "Study and resolution of singularities for a 7-dof redundant manipulator," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 469–480, June 1998.

[49] P.I. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the puma 560 robot," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1608–1613, 1994.

[50] Fan-Tien Cheng, Tzung Liang Hour, York-Yin Sun, and Tsing-Hua Chen, "Study and resolution of singularities for a 6-dof puma manipulator," *IEEE Trans. Systems, Man and Cybernetics*, vol. 27, no. 2, pp. 332–343, 1997, Part B.

[51] Jeng-Shi Chen, Fan-Tien Cheng, and Fan-Chu Kung, "Study and resolution of singularities for a 7-dof redundant manipulator," *Intl. Conf. on Industrial Electronics, Control, and Instrumentation*, vol. 3, pp. 1438–1443, 1997.

[52] V.D. Tourassis and M.H. Ang, "Identification and analysis of robot manipulator singularities," *International Journal of Robotics Research*, vol. 11, no. 3, pp. 248–259, 1991.

[53] Manja V. Kircanski, "Symbolical singular value decomposition for a 7-dof manipulator and its application to robot control," *IEEE Conf. on Robotics and Automation*, vol. 3, pp. 895–900, 1993.

[54] Jon Kieffer, "Differential analysis of bifurcations and isolated singularities for robots and mechanisms," *IEEE Transactions on Robotics and Automation*, vol. RA-10, pp. 1–10, Feb 1994.

[55] John E. Llyod, "Removing the singularities of serial manipulators by transforming the workspace," *IEEE Proc. Intl. Conf. Robotics and Automation*, 1998.

[56] Gunter Schreiber and Gerd Hirzinger, "Singularity consistent inverse kinematics by enhancing the jacobian transpose," in *ARK 1998 Strobl/Wolfgangsee Advances in Robot Kinematics: Analysis and Control*, Jadran Lenarcic and Manfred L. Husty, Eds., vol. 271, pp. 475–483. Kluwer Academic Publishers, 1998.

[57] D.N. Oetomo, M. H. Ang, and T. M. Lim, "Singularity robust manipulator control using virtual joints," *Proc. IEEE Conf. of Robotics and Automation*, vol. 3, pp. 2418–2423, May 2002.

[58] C.W. Wampler and L.J. Leifer, "Application of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators," *J. Dynamic Systems, Measurement and Control*, vol. 110, pp. 31–38, 1988.

[59] K.A. O'Neil, Y.C. Chen, and J.Q. Seng, "Removing singularities of resolved motion rate control of mechanisms, including self-motion," *IEEE Transations on Robotics and Automation*, vol. 13, no. 5, pp. 741–751, 1997.

[60] E. W. Aboaf and R. P. Paul, "Living with the singularity of robot wrists," *IEEE Intl. Conf. for Robotics and Automation*, pp. 1713–1717, 1987.

[61] S. Chiaverini and O. Egeland, "A solution to the singularity problem for six-joint manipulators," *Proc. IEEE for Robotics and Automation*, vol. 1, pp. 644–649, 1990.

[62] D. Oetomo, M. H. Ang, and S. Y. Lim, "Singularity handling on puma in operational space formulation," in *Lecture Notes in Control and Information Sciences*, Daniela Rus and Sanjiv Singh, Eds., vol. 271, pp. 491–501. Springer Verlag, 2001.

[63] D.N. Nenchev and M. Uchiyama, "Singularity consistent path tracking: a null space based approach," *Proc. IEEE Robotics and Automation*, pp. 2482–2489, 1995.

[64] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *IEEE Intl. Conf. Robotics and Automation*, vol. 1, pp. 183–187, 1988.

[65] D. N. Nenchev, "Redundancy resolution through local optimization: A review," *J. Robotics Systems*, vol. 6, no. 6, pp. 769–798, 1989.

[66] Y. W. Sung, D. K. Cho, and M. J. Chung, "A constrained optimization approach to resolving manipulator redundancy," *J. Robotics Systems*, vol. 13, no. 5, pp. 275–288, 1996.

[67] D.K. Cho, B.W. Choi, and M.J. Chung, "Optimal conditions for inverse kinematics of a robot manipulator with redundancy," *Robotica*, vol. 13, pp. 95–101, 1995.

[68] Eric Schwartz and Keith Doty, "Derivation of redundant wrist manipulators to avoid interior workspace singularities," *IEEE Conf. Proc. of Southeastcon'88*, pp. 403–407, 1988.

[69] Y. Tsumaki, H. Naruse, D. N. Nenchev, and M. Uchiyama, "Design of a compact 6-dof haptic interface," *Proc. of IEEE Intl. Conf. on Robotics and Automation*, pp. 2580–2585, May 1998.

[70] Robert Williams, "Singularities of a manipulator with offset wrist," *Journal of Mechanical Design*, vol. 121, no. 2, pp. 315–319, June 1999.

[71] Robert Williams, "Inverse kinematics and singularities of manipulators with offset wrists," *IASTED Intl. Journal of Robotics and Automation*, vol. 14, no. 1, pp. 1–8, 1999.

[72] J.P. Trevelyan, P.D. Kovesi, M. Ong, and D. Elford, "Et: A wrist mechanism without singular positions," *International Journal of Robotic Research*, vol. 4, no. 4, pp. 71–85, 1986.

[73] V. Milenkovic, "New nonsingular robot wrist design," *Robots II Conference Proc. RI/SME*, pp. 13.29–13.42, 1987.

[74] Etemadi Zangareh and Jorge Angeles, "On the isotropic design of general six-degree-of-freedom parallel manipulators," *Proc. Computational Kinematics Workshop*, pp. 213–220, Sept 1995.

[75] C. Chevallereau, "Feasible trajectories for a non redundant robot at a singularity," *Proc. Intl. Conf. on Robotics and Automation*, vol. 2, pp. 1871 – 1876, April 1996.

[76] N.S. Bedrossian, "Classification of singular configurations for redundant manipulators," *Proc. of IEEE Intl. Conf. Robotics and Automation*, vol. 2, pp. 818–823–17, 1990.

[77] N.S Bedrossian and K. Fluckiger, "Characterizating spatial redundant manipulator singularities," *proc. IEEE Intl. Conf. Robotics and Automation*, pp. 714–719, 1991.

[78] M. Zinn, O. Khatib, B. Roth, and J.K. Salibury, "A new actuation approach for human friendly robot design," Presented at the 8th Intl. Symp. on Experimental Robotics, Sant' Angelo d'Ischia, Italy, July 2002.

[79] T. Fukuda and F. Arai, "Microrobots and applications," *Emerging Technologies and Factory Automation, Design and Operations of Intelligent Factories. Workshop Proceedings. IEEE 2nd International Workshop on*, pp. 1–8, September 1993.

[80] R. Jamisola, M. H. Ang, D.N. Oetomo, O. Khatib, T. M. Lim, and S. Y. Lim, "The operational space formulation implementation to aircraft canopy polishing using a mobile manipulator," *Proc. IEEE Conf. of Robotics and Automation*, vol. 1, pp. 400–405, May 2002.

[81] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A.Casal, "Robots in human environments: Basic autonomous capabilities," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 684–696, 1999.

[82] George A. Grabits, "Polishing aerospace transparencies," *Conference on Robotic Solutions in Aerospace Manufacturing*, pp. MS86–203 1–22, March 1986, Orlando, Florida.

[83] Alan Bowling and Oussama Khatib, "Design of macro/mini manipulators for optimal dynamic performance," *Proc. Intl. Conf. on Advance Robotics*, pp. 865–872, 1997, Monterey California.

[84] D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Sys.*, vol. MMS-10, no. 2, pp. 47–53, 1969.

[85] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.

[86] J. Angeles, M. Habib, and C. López-Cajún, "Efficient algorithms for the kinematic inversion of redundant robot manipulators," *The International Journal of Robotics and Automation*, vol. 3, no. 1, pp. 106–116, 1988.

[87] R. M. Legrand and J. C. Slater, "Method and apparatus for mobile robot motion control," US patent No. WO9812498, March 1998.

[88] Tsuneo Yoshikawa, "Force control of robot manipulators," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 220–226, April 2000, San Fransisco.

[89] N. Hogan, "Impedance control: An approach to manipulation (part i-iii)," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 107, no. 1, pp. 1–24, 1985.

[90] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulator," *Trans. ASME Journal of Dynamic Systems, Measurement and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[91] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1381–1386, 1986.

[92] T. Yoshikawa, "Hybrid control theory of robot manipulators," in *The Sixth International Symposium of Robotics Research*, T. Kanade and R. Pauls, Eds., pp. 443–452. The International Foundations for Robotics Research, 1993.

[93] R. J. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *Proc. IEEE Intl. Conf. Robotics and Automation*, vol. 4, no. 5, pp. 549–556, 1988.

[94] D.E. Whitney, "Force feedback control of manipulator fine motions," *Trans. ASME Journal of Dynamic Systems, Measurement and Control*, vol. 99, no. 2, pp. 91–97, 1977.

[95] R. Volpe and P. Khosla, "A theoretical and experimental investigation of impact control for manipulators," *Intl. Journal of Robotics Research*, vol. 12, no. 4, pp. 351–365, 1993.