NOTE TO USERS

Page(s) missing in number only; text follows. Page(s) were microfilmed as received.

29
68,69

This reproduction is the best copy available.
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-59017-8
Performance Analysis of the Neural Network Control Robot System


Mingwei Wang

Department of Mechanical and Industrial Engineering

University of Toronto

Abstract

The advantage of neural network controllers to address robot trajectory tracking errors due to dynamic parameter uncertainties has been widely recognized. Despite many publications in the field of neural networks applied to repetitive robotic task execution, the lack of quantitative performance analysis impedes the application of neural networks in industry. The objective of this thesis is to formulate a systematic approach to analyze the mathematical relation between the performance and parameters of the neural network control robot system, i.e., the sensitivity of the system performance to parameters. This provides a better understanding how the neural network controller performs after the neural network is trained, a condition in which robot systems typically operate. First, the complete dynamic system is treated as a two-time-scale dynamics system, based on the perturbation theory. The learning rate is considered as a small parameter, the robot and neural network dynamics are regarded as fast and slow dynamic systems respectively. Then, the solution bound approach is employed to bound the norm of the system solutions. Finally, the influence of the system parameters, such as the robot payload and link mass etc., on the system performance, measured by the system error norm, is investigated using sensitivity analysis. Experimental results obtained with an industrial robot demonstrate the validity of the proposed approach. Above all, the quantification of the performance issue provides further justification for the commercialization of neural networks to robot systems. The future impact of this thesis is that it gives the basis to formulate an approach for synthesis of neural network control systems with guaranteed performance.
Acknowledgement

I would like to thank Professor J.K. Mills for giving me the opportunity to explore the exciting neural network control robot system field, and for his guidance and support with the work presented here. His dedication to the research and vision of the future direction are always an inspiration throughout my study at the University of Toronto.

I wish to thank the people in Laboratory For Nonlinear Systems Control, Hugh, Dong, Chris, Andrew, Edward, Jiaxin, Benny, Wuwei, Andy, Ke, Gary and others. They made my lab life more fun and my experience becomes richer though various discussions with them.

Finally, I would like to thank my parents for their continuous support and patience. Especially, I want to express my gratitude to my mother.
Contents

Abstract ................................. ii
Acknowledgement ....................... iii
Table of Contents ....................... iv
List of Figures ........................... vii
List of Tables ........................... ix
Nomenclature ............................ x

1 Introduction .......................... 1
  1.1 Thesis Scope ................................................................. 1
  1.2 Literature Survey .......................................................... 5
    1.2.1 Development of neural network structures ....................... 5
    1.2.2 Applications ............................................................ 6
    1.2.3 Performance analysis .................................................. 7
  1.3 Thesis Objective .......................... 9
  1.4 Thesis Contributions .................... 10
  1.5 Thesis Overview .......................... 11

2 Dynamic Modeling ........................ 13
  2.1 Introduction ............................................................... 13
  2.2 Robot Dynamics ........................................................... 14
  2.3 PID and Computed Torque Feedback ......................... 14
5.2.2 Solution bound ................................................................. 53
5.2.3 Influence of link mass on the system error bound ............ 55

5.3 Experimental Results ............................................................ 59
5.3.1 Experiment process ........................................................... 59
5.3.2 Experimental parameters ................................................... 60
5.3.3 Experimental results ......................................................... 60

5.4 Summary ............................................................................. 62

6 Influence of Trajectory Variation on System Performance 70
6.1 Introduction ......................................................................... 70
6.2 Theoretical Analysis ............................................................ 71
6.2.1 Separation of solutions ..................................................... 72
6.2.2 Solution bound ................................................................. 72
6.2.3 Influence of trajectory variation on the system error bound .... 74

6.3 Summary ............................................................................. 78

7 Conclusions and Discussions 80
7.1 Conclusions ......................................................................... 80
7.2 Suggestions for Further Research ......................................... 82

Bibliography .............................................................................. 83

Appendix-1 Bounds of Weights 89
Appendix-2 Norm of the Sensitivity of System Error with Respect to Payload 91
Appendix-3 Norm of Partial Derivative with respect to Payload 97
Appendix-4 Norm of Partial Derivative with respect to Link Mass 105
Appendix-5 Norm of Partial Derivative with respect to Trajectory 110
List of Figures

1.1 Architecture of neural network controlled robot system ........................................ 12
2.1 Back-Propagation Multi-layer Feed-forward Neural Network ............................... 20
3.1 The CRS Robotics A460 Robot ................................................................. 28
3.2 Robot Experiment Hardware Configuration ....................................................... 28
3.3 Experiment Software Architecture ....................................................................... 29
4.1 System error norm when the neural network is trained ......................................... 47
4.2 System error norm with payload of 2 kg ............................................................ 47
4.3 System error norm with payload of 1 kg ............................................................ 48
4.4 System error norm with the payload of 0 kg ....................................................... 48
4.5 System error norm with payload of 3 kg ............................................................ 49
4.6 System error norm with payload of 4 kg ............................................................ 49
4.7 Experimental result of system error norm ........................................................... 50
4.8 The comparison of system error norm ............................................................... 50
5.1 CRS A460 robot with the link mass variation ....................................................... 64
5.2 System error norm when the neural network is trained ....................................... 64
5.3 System error norm with link mass variation vector of [2kg, 2kg, 2kg] .................. 65
5.4 System error norm with link mass variation vector of [2kg, 2kg, 3kg] ............... 65
5.5 System error norm with the link mass variation vector of [2kg, 2kg, 4kg]...........66
5.6 System error norm with the link mass variation vector of [0kg, 0kg, 0kg]...........66
5.7 Experimental result of system error norm..........................................................67
5.8 The comparison of system error norm.................................................................67
6.1 Trajectory variation in three-dimensional space..............................................79
List of Tables

4.1. Bounds of robot and neural network parameters ........................................... 46
4.2 Predicted value of system error norm with varied payload ................................ 46
4.3 Experimental result of system error norm with varied payload ........................... 46
5.1 Bounds of robot and neural network parameters ............................................. 63
5.2 Predicted value of system error norm with varied link mass ............................. 63
5.3 Experimental result of system error norm with varied payload ........................... 63
Nomenclature

\( a_i \) coefficients of the given robot trajectory equation

\( A \) linear uncoupled matrix, together with \( B \) to represent the linear part of the complete system

\( B \) linear uncoupled matrix

\( c_i \) \( \cos(q_i) \)

\( C(q, \dot{q}) \in \mathbb{R}^{n \times l} \) centrifugal, Coriolis and gravitational term

\( \hat{C}(q, \dot{q}) \in \mathbb{R}^{n \times l} \) estimate of centrifugal, Coriolis and gravitational term

\( \Delta C \) variation between estimated and real Coriolis and centrifugal term

\( d_i \) distance between the axis of joint \( i \) and joint \( i+1 \)

\( e_i \) first order of solution

\( e_2 \) second order of solution

\( e_{r1} \in \mathbb{R}^{n \times l} \) error state of joint angle

\( e_{r2} \in \mathbb{R}^{n \times l} \) error state of joint velocity

\( e_s \in \mathbb{R}^{2 \times n \times l} \) vector of error state

\( E \) error square sum of neural network output
\( H(q) \in R^{nxn} \) inertia matrix of robot

\( \hat{H}(q) \in R^{nxn} \) estimate of inertia matrix

\( I_{ij} \) element of the inertia matrix of CRS A460 robot

\( J_{ijk} \) pseudo-inertia element in raw \( j \) and column \( k \) for link \( i \)

\( k_p \in R^{nxn} \) proportional gain matrix of PD control

\( k_v \in R^{nxn} \) derivative gain matrix of PD control

\( k_m \) maximum of PD gains

\( K \) matrix of PD gains

\( l \) numbers of neurons in the output layer of neural network

\( l_{ic} \) signed distance from the origin of link \( i \) along the \( a_i \) axis to the center of mass of the link \( i \)

\( L_m \) link mass vector at which neural network is trained, \( L_m = [m_1, m_2, m_3]^T \)

\( \Delta L_m \) link mass variation, \( \Delta L_m = [\Delta m_1, \Delta m_2, \Delta m_3]^T \)

\( m \) numbers of neurons in the hidden layer of neural network

\( m_i \) mass of link \( i \)

\( \Delta m_i \) variation of the mass of link \( i \)

\( m^* \) payload mass at which the neural network is trained

\( \Delta m \) payload variation, \( \|\Delta m\|_\infty = |\Delta m| \)

\( M_1 \) non-negative

\( M_2 \) positive constants
numbers of neurons in the input layer of neural network and also the degree-of-freedom of the robot

\( q = R^{nxl} \) vector of generalized coordinates, i.e., joint angles

\( \dot{q} = R^{nxl} \) vector of joint velocities

\( \ddot{q} = R^{nxl} \) vector of joint accelerations

\( q^{d} = R^{nxl} \) vector of desired joint angles

\( \dot{q}^{d} = R^{nxl} \) vector of desired joint velocities

\( \ddot{q}^{d} = R^{nxl} \) vector of desired joint accelerations

\( \rightarrow q \) trajectory variable vector at which the neural network is trained,

\( \rightarrow q = [q_1, q_2, q_3]^T \)

\( \Delta q \) trajectory variation vector, \( \Delta q = [\Delta q_1, \Delta q_2, \Delta q_3]^T \)

\( q_i \) joint rotate angle of link \( i \)

\( \Delta q_i \) variation of the joint rotate angle of link \( i \)

\( R \) radius of the round plate form payload

\( s_{ij} = \sin(q_i + q_j) \)

\( u = R^{nxl} \) input signal of computed torque technique

\( v = R^{nxl} \) neural network compensation signal, i.e., neural network output

\( v_i \) input-output mapping function of the neural network

\( v^{d}_i \) desired neural network output
\( w^i \in R^{nxm} \) weights between input and hidden layers

\( w^o \in R^{m-xl} \) weights between hidden and output layers

\( \dot{w}^{ih} \in R^{nom} \) change rates of weights between input and hidden layers

\( \dot{w}^{ho} \in R^{m-xl} \) change rates of weights between hidden and output layers

\( w \in R^{2nxl} \) weight states, \( w = [w_{s1} \quad w_{s2}]^T = [w_{kj} \quad w_{ji}]^T \)

\( X \) robot state, \( X = [q \quad \dot{q}]^T \)

\( \beta \) coefficients relevant with PD term in the solution bound process

\( \chi \) bound of first order solution

\( \delta \) non-negative constants

\( \varepsilon \) positive constant \( 0 < \varepsilon < 1 \)

\( \phi \) upper bound on the neural network output

\( \Phi \) scale factor of the neural network output

\( \eta \) uncertainty of the transformed equivalent system in solution bound process caused by estimation of robot parameters

\( \eta_\sigma \) uncertainty of the sensitivity system

\( \varphi(\cdot) \) derivative of the neural network activation function

\( \lambda \) learning rate

\( \mu \) control of the equivalent system in solution bound process

\( \Theta \) empty matrix

\( \rho \) nonnegative constants

\( \sigma \) sensitivity of system error with respect to the payload mass
\( \tau \in \mathbb{R}^{n \times l} \) vector of joint torque

\( \omega, \omega \) positive constants

\( \zeta(e,w) \) nonlinear part of the complete system

**Notation**

\[ \| \cdot \|_{\infty} \] \( L^\infty \) infinite norm \( L_{\infty} \)

\[ \| \cdot \|_{T\infty} \] truncated \( L^\infty \) infinite norm \( L_{T\infty} \)

**Acronyms**

- **BIBO** bounded-input bounded-output
- **BP** Back-Propagation Multi-layer Feed-forward Neural Network
- **CT** computed torque technology
- **DSP** Digital Signal Processor
- **MLP** Multilayer Perceptron
- **NN** neural network
- **PD** proportional and derivative control technology
- **PID** proportional, integral and derivative control technology
Chapter 1

Introduction

1.1 Thesis Scope

The robotics industry has experienced a sustained growth surge for the past few years. With more than 10,000 new robot orders annually received by U.S. based robotics companies [RIA, 1999], and enormous untapped opportunities still available throughout the world, the robotics industry will be one of the most important global industries of this century. In order to stay on the leading edge in today’s global market and competition, robotics companies must offer better performance to customers. Among the many factors influence the robot performance, the control system of robot is of great importance. The goal of a robot control system is to determine the time history of the joint inputs required to drive the end-effector to execute a commanded task [Spong and Vidyasagar, 1989]. Since the appearance of the programmable robot in 1950’s, the proportional, integral and derivative (PID) control has been the dominant control technology in robot industry. With the increasing demands of robot tasks, conventional PID method is inadequate. For example, using the end-effector of a robot to follow a given trajectory in a
repetitive manner is one of the important robot applications in industry. As we know, under PID control, the higher the trajectory following speed is, the more inaccurately the robot follows the desired trajectory. The reason for the above behavior is the non-linearity and complexity of the robot system dynamics. In order to solve the problem mentioned above, several modern control technologies have been proposed. These methods can solve the robot control problems successfully in some cases, but the results may be unsatisfactory when an unknown payload is added, for example. The reason for this is that these control methods rely on the exact dynamic model of the control plant. The mathematical model of robot system is not known exactly due to uncertain factors such as the tolerance, which is caused by the "wear-and-tear" (e.g., joint friction, wear of gears, etc.) of the robot and the unknown payload.

A control architecture, called a neural network [Haykin, 1994], has been developed as a possible solution to a number of commonly encountered problems in the control of robots. A neural network (NN) is an interconnection of a set of neurons, usually in a layered pattern. A neuron, which performs a nonlinear transformation on its inputs, is the fundamental operational unit of a neural network. The connecting links between the neurons are characterized by weights that represent the link strength. By adjusting the value of each weight, a mapping from the input to the output of neural network can be obtained. This mapping can be used to approximate smooth functions to arbitrary accuracy [Volga, 1989].

Neural network controllers have several properties that make this control method particularly suitable for applications to robot control in industrial manufacturing environments. The main attribute of neural network controllers is their ability to learn to generate a corrective signal applied on a system's actuator to correct the disturbances that act on the system for a long period.
of time, i.e., neural network can adjust the outputs of the system based on the relation of inputs and outputs from former trials. Thus the neural network doesn't depend on the exact robotic system model. Also the structure of the neural network controllers enables themselves an iterative learning capability that is well fitted to the repetitive task environment in which industrial robots are commonly used.

Research works have shown that the neural network controller, when applied in an appropriate manner to the control of robot manipulators, leads to less trajectory following errors than is possible with other control approaches, such as the computed torque method [Abdallah and Heileman, 1996]. Although there have been many neural network controllers with good simulation or even experiment results, the lack of a theoretical basis and quantitative analysis of neural network system performance prevents the neural network controllers from commercialization in the robot industry. Since it is difficult to promote the neural network controlled robot in commercial industry by simply stating that the new robot with neural network controller has a better performance then the conventional PID robot, without stating how much better. The performance of the robot control system usually represents how well the robot follows the given reference track and can be defined in terms of exactness, quickness and stability of the system output. Hence, regardless the abundant research results in terms of neural network controllers for robot systems, the fact is that neural network controllers are rarely applied to commercial robots industry. The following reasons may be considered to explain the above phenomenon:

1. Much of the research work, which focuses on neural network controllers, is not performance oriented. There are a number of aspects of the neural network controllers that are investigated,
such as the construction of the architecture of a neural network and the acceleration of the learning process [Srideep and Peng, 1996].

2. Most research results relevant to the performance of the neural network controllers are qualitative.

3. The parameters in the neural network performance studies are chosen heuristically [Abdallah and Heileman, 1996]. Thereafter no exact and deterministic outputs of neural network control system can be expected.

In light of the above reasons, the main objective of this thesis is to investigate the quantitative relation between the performance and parameters of the neural network control system. When a neural network controller is implemented on robot systems, the neural network is trained for a certain number of repetitive trials first. Then the system may be operated with the weights fixed. The system parameters may still vary in some cases, for example, when robots grasp different payload or change of end-effector tools, etc. Consequently, such changes to the operational status will affect the system performance. Though through the generalization ability of neural networks, such changes can be compensated to some extent. Hence, if the system performance degradation can be quantified, i.e., a quantitative performance analysis is obtained, it will provide better information to potential users of neural network controllers on the overall performance of these controllers. In a further sense, this research is of assistant to the commercialization of the neural networks in robot industry.

A neural network controller, which is used in conjunction with existing PID controls, is developed, as shown in the Figure 1.1. We define the error between the output of robot joint vector and the output of the given reference model as the neural network input and use the output
of neural network to compensate the system along with the PID feedback term. The model for the neural network here is the multilayer perceptron with error backpropagation learning algorithm, i.e. backpropagation (BP) neural network, since it is the most widely adopted model when applied to control systems [Haykin, 1994]. The back-propagation algorithm adjusts the values of weights according to a manner similar to that of a gradient search in the neural network output error space [Thimm, 1994].

The parameters of the neural network control robot system refer to the variable coefficients in the system equation, not the variables of the system dynamics themselves, though the variation of the parameters also cause the change in the system performance. The performance of the neural network is usually referred as the mapping and generalization of the neural network. Mapping is the goodness of the neural network's mapping ability from the input space to the output space. Generalization defines how well the neural network's mapping holds when dealing with a new situation, such as a task that the neural network hasn't been trained before or a case in which the neural network system is operating under unknown disturbance from the environment.

The analytical relation between the performance and a specific parameter of the neural network can be acquired by employing the perturbation, solution bound and sensitivity theories.

By achieving the above goal, the specific quantitative performance analysis of the neural network control robot system with respect to the neural network parameters is originally developed. Also the above analysis gives a theoretical base for the further study of the performance issue of the neural network control system, such as optimization etc.

1.2 Literature Survey
Many research works have been published about the neural networks and they can be roughly categorized into the following three areas.

1.2.1 Development of neural network structures

Many efforts had been put into the study of developing new neural network architectures. For example, the evolutionary method uses a generic algorithm to evolve the neural network architecture automatically [Sndee and Peng, 1996]. Though all research concerning neural networks address the system performance to some extent, and every paper declares the improvement in the neural network control system performance, few directly investigate the performance issue of the neural network control system, such as the universal learning method [Galan, 1994] and modular neural network [Lifford, 1994].

1.2.2 Applications

Presently, neural networks are mostly adopted in pattern recognition, signal processing, specialist system and robotics control, etc. The application of neural networks in robot control is of our research interest here. As previously discussed, neural networks are suitable to be applied in the robot control systems and the applications can loosely be classified into the following categories.

- A neural network is used to identify the robot inverse dynamics and then acts as the model for other control approaches, such as optimal control, which requires the exact robot dynamic model [Murata, 1994].
- A neural network is used as a feedback controller in the robot control system to improve the
system performance [Yasuyoshi, 1994].

- A neural network is used as an optimal computation method for trajectory planning in the robot control system [Martínez, 1994].

- A neural network is integrated with other control methods to form a coordinated control strategy so as to utilize the merits of both control methods, such as fuzzy neurocontrol [Sutton, 1994].

The purpose of the above applications is to improve the performance of robot system through the use of neural networks, but the performance issue are typically just investigated qualitatively, i.e., the exact amount of performance improvement of robot system after adopting the neural networks is not addressed.

1.2.3 Performance analysis

Some existed publications address the performance issue of the neural network control system, but the adopted methodologies are mainly tuning approaches. For instance, it has been shown that Multilayer Perceptron (MLP) can form arbitrarily close approximations to any continuous nonlinear mapping [Abdallah and Heileman, 1996], but only as the size, i.e. the number of neurons, of neural network becomes arbitrarily large. The fact that every neural network when applied to a real system has a finite size leads to the consideration of the proper size for a neural network. In order to solve the size problem, two methods have been proposed. One is growing approach which is proposed to construct a neural network starting with the smallest possible network and gradually increase the size of the neural network until the performance becomes worse, i.e., the output error of neural network begins to increase [Bagart, 1994]. On the other hand, a pruning approach is proposed to construct neural network starting
with a large network and then apply a cutting technique that eliminates the neurons or weights that contribute little or nothing to the performance of neural network [Hagn, 1994]. Both methods are based on the "try and see" principle, thus lack of a theoretical basis. It is impossible to get a deterministic answer to the questions such as "How much performance improvement of the neural network can be expected with respect to a variation in the number of neurons?" Above all, the conclusion is that the present studies of neural network control robot systems are either not performance analysis orientated or simply provide qualitative performance analysis.

On the other hand, there are promising results that permit the quantitative performance analysis, even if the influence of the parameters on the system performance hasn’t been addressed. For example, Chen and Mills [Chen and Mills, 1997] have proved the bounded-input bounded-output (BIBO) stability of neural networks for robot control systems, i.e., if the input of the system is bounded, the output of the system is bounded. The close-loop tracking with internal stability of neural networks for robot system is guaranteed [Lewis, 1998], under the condition that a robust term is added to the learning algorithm.

Moreover, some research results may be of assistance to formulate a systematic methodology for the quantitative performance analysis of neural network robot control system. For instance, the bound of solution method [Spong and Vidyasagar, 1987], which utilizes the physical features of robotic systems, is suitable for our performance analysis. Since the dynamics of neural network control robot system are highly nonlinear and complex, the exact solutions of the system dynamic equations are likely impossible to obtain, while the bound of solution will give theoretical insight into the system performance as well. Perturbation theory [Gardin, 1991] investigates the dynamic system with a small parameter which can characterize part of the system dynamics as much
slower than that of the other part of the system. The learning rate of a stable neural network is always small, and consequently can be considered as the small parameter of the complete dynamic system. Hence, the neural network and robot dynamics can be treated as a two time-scale system. The sensitivity theory [Frank, 1978], which studies the system output variation with respect to system parameters, provides an appropriate tool to analyze the influence of the system parameters. Finally, a systematic methodology investigating the performance of neural network control robot system is possible to be constructed with the assistance of perturbation, solution bound and sensitivity theories.

No work on the performance analysis of neural network system by perturbation and solution bound methods has been reported in the literature yet. This, and the feasibility of the insight into performance issue, as well as the achievements on the perturbation and solution bound theories, motivate this Ph.D. research.

1.3 Thesis Objective

Based on the literature survey, we find there are virtually no research publications that quantitatively address the performance issue of neural network control robot systems. Hence in order to address this aspect of neural network control, the objective of this research is to investigate the mathematical relations between the performance, which is measured by the system error norm, and the parameters (such as the payload or link mass of robot) of neural network robot control systems as shown in Figure 1.1. The specific objectives of the thesis are as follows:

- Determine the functional relations between neural network control robot system error norm and the variation payload. This determines the influence of the robot payload on the
performance of the neural network control robot system

- Determine the quantitative relations between the performance of neural network control robot system and the link mass of robot.
- Determine influence of the trajectory variation on the neural network control robot system performance.
- Conduct experiments on the industrial standard equipments to verify the above theoretical justification.

1.4 Thesis Contributions

In this research work, the specific performance related theoretical work of neural network control robot system, formerly not found in the literature, is being developed. This systematic methodology gives the basis for the further investigation of the performance issue of neural network control systems. The developments are significant, since it is the quantification of neural network controller performance that is critical for commercialization of neural network controllers in robot industry. The detailed contributions of the thesis are:

- A systematic methodology for the quantitative performance analysis of the neural network robot control system is established. Insight of how the payload mass influence the complete system performance, measured by system error norm, is obtained through rigorous mathematical relations.
- The rigorous relation between the performance of the neural network control robot system and the link mass variation is determined. As link mass variation is normally caused by the update of new equipments, this research result helps us to have a better sense what the system
performance will be affected when the new facilities are attached to robots.

- The influence of the trajectory variation on the neural network control robot system performance is obtained. This result permits us to predict the scale of performance degradation caused by the variation of the trajectory, which may be altered due to different execution tasks.
- The neural network controller is used conjunction with the widely existed PID control, thus leads to easy implementation of neural network in the industrial robot systems.
- The experiments are conducted according to the industrial standard, thus the validity of the results should not be a problem.

1.5 Thesis Overview

The outline of the thesis is as follows. In Chapter 2, the dynamic models of the neural network control robot system are constructed. Chapter 3 introduces the experimental test bed, on which all experiments are performed. In Chapter 4, the methodology of quantitative analysis of the payload influence on the complete system performance is developed. Chapter 5 investigates the influence of the robot link mass on the complete system performance. The trajectory variation issue is discussed in Chapter 6. Finally, Chapter 7 provides conclusions to the work and suggestions for further research.
Figure 1.1 Architecture of neural network controlled robot system
Chapter 2

Dynamic Modeling

2.1 Introduction

In this chapter, the dynamic model of the neural network controllers and robot system are formulated. The robot and neural network models are first given separately, then the complete system model is developed and finally expressed in error dynamics form. The rest of the chapter is organized as follows. Section 2.2 gives the dynamic model of a three degree-of-freedom robot. In Section 2.3, the PID and computed torque feedback algorithm are provided. Section 2.4 introduces the Back-Propagation neural network model. In Section 2.5, the complete system model is constructed and expressed in error state form. A summary is provided in the last section of this chapter.
2.2 Robot Dynamics

The dynamic equation of motion for a three degree-of-freedom manipulator can be expressed as

\[ H(q)\ddot{q} + C(q,\dot{q}) = \tau \]  

(2.1)

where:

- \( \dot{q} \) = vector of joint velocities, \( \dot{q} \in R^{n \times 1} \)
- \( \ddot{q} \) = vector of joint accelerations, \( \ddot{q} \in R^{n \times 1} \)
- \( H(q) \) = inertia matrix, \( H(q) \in R^{nxn} \)
- \( C(q,\dot{q}) \) = Centrifugal, Coriolis and gravitational term, \( C(q,\dot{q}) \in R^{n \times 1} \)
- \( \tau \) = vector of joint torque, \( \tau \in R^{n \times 1} \)
- \( n \) = degree-of-freedom.

2.3 PID and Computed Torque Feedback

As discussed previously, the PID controller and neural network controller are used in conjunction with each other. The control feedback output signal is joint torque, which is generated by the computed torque technique. The input signal of the computed torque method is in turn generated from both PID and neural network controllers.

First, the computed torque control is given as  [Asada and Slotine, 1986]:
\[ \tau = \hat{H}(q)u + \hat{C}(q, \dot{q}) \]  
\hspace{1cm} (2.2)

where:

\[ \hat{H}(q) = \text{estimate of inertia matrix } H(q), \hat{H}(q) \in \mathbb{R}^{n \times n} \]

\[ \hat{C}(q, \dot{q}) = \text{estimate of Centrifugal, Coriolis and gravitational term } C(q, \dot{q}), \hat{C}(q, \dot{q}) \in \mathbb{R}^{n \times l} \]

\[ u = \text{input signal of computed torque technique, } u \in \mathbb{R}^{n \times l}. \]

The control input signal is generated from both PID and neural network controllers. To facilitate the theoretical analysis, only Proportional and Derivative terms of PID control are adopted

\[ u = \ddot{q}^d + k_p (q - q^d) + k_v (\dot{q} - \dot{q}^d) + v \]  
\hspace{1cm} (2.3)

where:

\[ k_p = \text{proportional gain matrix of PID control, } k_p \in \mathbb{R}^{n \times n} \]

\[ k_v = \text{derivative gain matrix of PD control, } k_v \in \mathbb{R}^{n \times n} \]

\[ q^d = \text{vector of desired joint angles, } q^d \in \mathbb{R}^{n \times l} \]

\[ \dot{q}^d = \text{vector of desired joint velocities, } \dot{q}^d \in \mathbb{R}^{n \times l} \]

\[ v = \text{neural network compensation signal, } v \in \mathbb{R}^{n \times l}. \]

### 2.4 BP Neural Network

The Back-Propagation neural network is the most widely employed model in the application of neural networks to control systems. The BP network is a feed-forward multi-layer network, as shown in Figure 2.1, with the weights adjusted in such a way that a gradient search approach in output error space is generated to minimize a performance criteria, to be given in the
following. The input of the BP neural network is the complete system error state $e$, which will be defined later. The output of the BP neural network is $v$.

The input-output mapping function of the BP neural network is

$$
v_k = \varphi(-\sum_{j=1}^{m} w_{kj}^b \cdot \varphi(-\sum_{i=1}^{n} w_{ji}^h \cdot e))
$$

(2.4)

where:

$v_k$ = input-output mapping function of the neural network

$w^b$ = weights between input and hidden layers, $w^b \in \mathbb{R}^{n \times m}$

$w^h$ = weights between hidden and output layers, $w^h \in \mathbb{R}^{m \times l}$

$n,m,l$ = numbers of neurons in the input, hidden and output layer respectively.

$\varphi$ = hyperbolic activation function, which is given as

$$
\varphi(x) = a \tanh(x) = \frac{1-e^{-x}}{1+e^{-x}}
$$

(2.5)

The neural network dynamics can be adjusted by the BP learning algorithm, which is given as [Haykin, 1994]

$$
\dot{w}_{kj}^b = -\lambda \frac{\partial E}{\partial w_{kj}^b} = \lambda (v^d - v^k) \varphi'(v_k) b_j
$$

(2.6.1)

$$
\dot{w}_{ji}^h = -\lambda \frac{\partial E}{\partial w_{ji}^h}
$$

(2.6.2)

where:

$\lambda$ = learning rate,

$\dot{w}_{ij}^h$ = change rates of weights between input and hidden layers, $\dot{w}_{ij}^h \in \mathbb{R}^{n \times m}$
\( \dot{w}^{ho} \) = change rates of weights between hidden and output layers, \( \dot{w}^{ho} \in R^{n_{hl}} \)

\( E \) = error square sum of neural network output, which is defined as

\[
E = \sum_{k=1}^{l}(v_k^d - v_k)
\]

(2.7)

where:

\( v_k^d \) = desired neural network output.

### 2.5 Complete System Dynamic Model

Finally, we formulate the complete system dynamics by synthesizing the individual system dynamics. Since the error norm will be used as the performance criterion, the complete system dynamics is constructed on the error state. Define the error state \( e_s \), as

\[
e_{s1} = q - q^d \\
e_{s2} = \dot{q} - \dot{q}^d
\]

(2.8.1, 2.8.2)

then

\[
e_s = [e_{s1} \ e_{s2}]^T
\]

(2.8.3)

where:

\( e_{s1} \) = error state of joint angle, \( e_{s1} \in R^{n_{hl}} \)

\( e_{s2} \) = error state of joint velocity, \( e_{s2} \in R^{n_{hl}} \)

\( e_s \) = vector of error state, \( e_s \in R^{2n_{hl}} \)

The robot dynamics with PD control expressed in error dynamics form is
\[ \dot{e}_{s1} = e_{s2} \]
\[ \dot{e}_{s2} = \ddot{\hat{q}} - \ddot{\bar{q}}^d = H^{-1}(q) \cdot \tau - H^{-1}(q)C(q, \dot{q}) - \ddot{\bar{q}}^d \] (2.9)

where:

\[ \ddot{\bar{q}}^d = \text{vector of desired joint accelerations}, \quad \ddot{\bar{q}}^d \in R^{n \times 1}. \]

Considering the neural network dynamics (2.6) together, the complete system dynamics is obtained as

\[ \dot{e}_s = Ae_s + B \zeta(e_s, w_s) \]
\[ \dot{w}_s = \lambda \cdot g(e_s, w_s) \] (2.10)

where:

\[ w \in R^{2n \times 1} = \text{weight states}, \quad w = [w_{s1} \quad w_{s2}]^T = [w_{k1} \quad w_{k2}]^T, \]

\[ B = \begin{bmatrix} 0 \\ \lambda \end{bmatrix} = \text{linear uncoupled matrix}, \]

\[ A = \begin{bmatrix} 0 & I \\ k_p & k_v \end{bmatrix} = \text{linear uncoupled matrix, together with } B \text{ to represent the linear part} \]

\[ 0 = \text{zero matrix}. \]

\[ \zeta(e, w) = \text{nonlinear part of the complete system, and is expressed as} \]

\[ \zeta(e, w) = E(\ddot{\bar{q}}^d + (k_p e_1 + k_v e_2)) + H^{-1} \Delta C + H^{-1} \hat{H} \nu \] (2.11)

where:

\[ \Delta C = \text{variation between estimated and real Corolis and Centrifugal term}, \]

\[ \Delta C = \bar{C}(q, \dot{q}) - C(q, \dot{q}) \]
2.6 Summary

In this chapter, the dynamic model of complete system with neural network, PID and computed torque technique used in conjunction in feedback loop is constructed. In favor of the simplicity in mathematical manipulation, only proportional and derivative parts of PID are adopted in our analysis.
Figure 2.1 Back-Propagation Multi-layer Feed-forward Neural Network
Chapter 3

Experimental Test Bed

3.1 Introduction

This chapter provides a description of the experimental test bed on which all experiments are run. Each experiment is comprised of a set number of repeated trials. A trial consists of routine steps that take the robot through a desired trajectory. The rest of chapter is organized as follows. Section 3.2 gives a description of the system hardware construction. In Section 3.3, the system software structure is introduced. A summary is provided in Section 3.4.

3.2 Experiment Hardware Structure

The hardware of neural network control robot system is composed of the following several components.

- CRS Robotics A465 Robot
- CRS Robotics C500 Controller and transputer network
- DSPACE Digital Signal Processor
- Two host PCs
In the following, the hardware components listed above are described. Modifications made to the test bed to facilitate the performance of real industrial tasks are discussed.

**Robot**

A CRS Robotics A460 robot, as shown in Figure 3.1, is available in the Laboratory for Nonlinear Systems Control at the Department of Mechanical and Industrial Engineering, University of Toronto. The robot has six degrees of freedom, realized by six revolute joints. Each joint is actuated by a DC motor via a harmonic drive, which provides a transmission ratio of 100:1. The last three joints form a spherical wrist, a configuration that arises when the three consecutive joint axes intersect at a point. The point of intersection is referred to as the wrist center. Such a configuration allows a decomposition of the six degree of freedom (DOF) kinematic computation into two sets, three DOF each, of kinematic computations. It separates the orientational coordinates, controlled by the local PID controllers of the last three joints, from the positional coordinates, controlled by the local user defined control laws of the first three joints. Since we focus on the positioning trajectory in our experiments, the controller is designed and implemented on the first three joints, leaving the local controllers on the last three joints as PID controllers.

**C500 Controller and Transputer Network**

The CRS robotic system includes the commercial C 500 robot controller. It's a multi-processor real time controller that makes use of an INTEL 80286 processor and two INMOS 805 transputers. The transputers are capable of hosting multiple process simultaneously, working on one process at a time according to the specified priority of the process. Each transputer has four communication links that allow easy expansion of the system by adding more transputers to the network.
The industrial robotic system provided by CRS has undergone many modifications to facilitate the implementation of different control laws, and help with the research performed at the Laboratory for Nonlinear Systems Control. Transputers have been added to the controllers, as well as a Windows based interface that runs on a 486-66 MHz personal computer (PC). Another transputer was implemented in the PC circuitry to handle the communication between the PC and the controllers. This expansion of the network is necessary to handle the large number of computation generally required when implementing complex control laws used in research, such as neural network and force control.

The first three joints have been equipped with the torque sensors constructed of two pairs of strain gauges mounted on the outer perimeter of the flexsplines of the harmonic drives. The torque sensors measure the actual torque output from the harmonic drive, which is used in an inner torque feedback loop. This work is also reported in [Mills, 1995]. The torque sensors feed their signals to a torque signal conditioner, thus each controller network has been equipped with an A/D module to sample the joint torque data. The detailed description of the system can be found in [Baines, 1995].

**Digital Signal Processor (DSP)**

A dSPACE Inc. C40 Digital Signal Processor (DSP) is used to execute neural network programs. The C40 processor is capable of performing 40 million floating point operations per second, hence is well suited for the complicated neural network calculations.

The DSP is contained in a dSPACE PX-6 expansion box, external to the C500 controller. The Input/Output (I/O) functionality of the DSP is implemented by DS4001, a digital I/O and timer board. The DSP has been interfaced to the C500 controller via a parallel cable connection, which connects the DS 4001 directly to the on board I/O module of the IOT336.
Control transputer, where the user specified control program is executed. A schematic showing the hardware configuration of the experimental test bed after addition of the DSP is presented in Figure 3.2. In this figure, the TMS320C40 DSP and DS 4001 are shown in the dSPACE System block. In the C500 Controller block, just the IOT336 transputer is shown separately. The transputers comprising the rest of the C500 transputer network are grouped together and referred to as “Other TRAMs”. The development of this hardware configuration was first reported in [Chen, 1998], and further details are provided in [Zhou, 1997].

Two Host PCs

Two host computers serve as the user interfaces for the DSP and C500 respectively. These two host computers are required for initiating all experiments, running control software, data recording, and selecting experiment parameters.

Through the above hardware installation, the neural network control law can be executed. The neural network control law is programmed with C language, compiled and downloaded to the TMS320C40 DSP. The serial connection enables communication between the C500 and the neural network on DSP. The setup allows real time communications such that information is passed between the DSP and C500 while the robot and neural network are running.

3.3 Experiment Software Structure

The neural network control robot system experiment software architecture is shown in Figure 3.3. The DSP Host enables a user to compile and download any type of neural network to the DSP. MATLAB software is employed for the “front-end” programming and enables neural network parameter selection as well as the generation of initial weights.
The Neural Network Module consists of following subroutines

- The network output map generation.
- The neural network learning procedure, i.e., weight adjustments.

A different C program containing these subroutines has been developed for each type of neural network.

The DSP Communication Module sends and receives data to and from the C500 Communication Module. The DSP Communication Module must execute the following functions

- Establish communication with the C500 controller.
- Send neural network output signals to the C500 controller to provide the compensating torque signal.
- Receive the actual and desired values of joint position, velocity and acceleration from the C500 controller.

The DSP Execution Manager supervises the execution of subroutines in the DSP. It ensures a proper schedule for real-time execution. The subroutines include

- Neural network output generation.
- Neural network learning.
- Communication with the C500 controller.
- Data recording.

The C500 Host allows the user to select parameters for the PID control system. Important selections include, robot trajectory, PID gains, maximum robot velocity and accelerations,
and the number of trials in an experiment. Experiment execution and termination are also controlled via the C500 Host.

The C500 Communication Module works with the DSP Communication Module to carry out data transfer. Its responsibilities are as follows

- Establishing communication with the DSP.
- Sending actual and desired values of joint position, velocity and acceleration.
- Receiving the neural network output signals from the DSP.

The User Control Program contains the PID control algorithm. Modifications were made to include the addition of the neural network compensating torque signal into the control scheme.

The C500 Execution Manager is responsible for supervising the other modules while incorporating the real-time neural network communication and execution. The detail description of the system software structure can be found in [Clark, 1998].

3.4 Summary

In this chapter, the construction of the neural network control test bed is introduced. A DSP is interfaced to the commercial C500 controller, allowing for the addition of a neural network signal to the commanded PID torques for the first three joints of the CRS robot. The various hardware and software components of the system are described.
Figure 3.1. The CRS Robotics A460 Robot

Figure 3.2. Robot Experiment Hardware Configuration
Figure 3.3. Experiment Software Architecture
NOTE TO USERS

Page(s) missing in number only; text follows. Page(s) were microfilmed as received.

This reproduction is the best copy available.
Chapter 4

Influence of Payload Variation on System Performance

4.1 Introduction

In this chapter, the influence of the robot payload variation on the neural network control robot system performance is investigated. Payloads of robot are often varied in the execution process due to the demands of different tasks, and consequently this variation causes difference in the trajectory tracking performance of the robot system. Hence, the payload should be addressed as a parameter of neural network control robot system. In the following, we formulate a systematic methodology to analyze the influence of payload on the performance of robotics system with a neural network control, based on the perturbation, solution bound and sensitivity theories. Perturbation theory [Gardin, 1992] investigates dynamic systems with a small parameter, which can characterize part of the system dynamics as much slower than that of the other part of the system. The learning rate of a stable neural network is typically small, and consequently can
be considered as the small parameter of the dynamic system. Hence, the neural network and robot dynamics can be treated as a two time-scale system, which leads to two relatively simple dynamics. The solution bound method [Spong and Vidyasagar, 1987] is suitable for performance analysis. Since the system dynamics are highly nonlinear, the exact solution of system equation is likely impossible to obtain, while the bound of solution will also give theoretical insight into the system performance. The sensitivity approach [Frank, 1978], which studies the system output variation with respect to system parameters, provides an appropriate tool to analyze the influence of the payload variation.

The rest of this chapter is organized as follows. In Section 4.2, a systematic methodology to investigate the influence of payload on the system performance is developed. Section 4.3 illustrates the experimental results to verify the theoretical analysis of previous section. A summary is provided in Section 4.4.

### 4.2 Theoretical Analysis

The methodology is developed in the following order to achieve our objectives. In Section 4.2.1, the solution of system equation is expressed in a power series of learning rate parameter \( \lambda \), based on perturbation theory. The bound of system solutions is obtained by employing the solution bound approach in Section 4.2.2. In Section 4.2.3, we explore the relation between the performance and the payload of the neural network control robot system. For purposes of conciseness, three Appendixes are provided to give details of the calculations.
4.2.1 Separation of Solutions

In this section, the solution of the system equation is expressed in a power series in $\lambda$. The learning rate of a stable neural network can be treated as a small parameter of the complete system with two time-scale dynamics, i.e., fast robot dynamics and slow neural network dynamics. By adopting this approach, the influence of learning rate on the error bound of the system solution can be shown explicitly.

From equation (2.10), the complete system dynamics can be expressed as

$$
\begin{align*}
\dot{e}_s &= \lambda e_s + B \xi(e_s, w_s) = f(e_s, w_s) \\
\dot{w}_s &= \lambda \cdot g(e_s, w_s)
\end{align*}
$$

(4.1)

We first transform the time variable in order to express the equations as a regular form in perturbation theory by defining $\tau = \lambda t$, then

$$
\frac{d}{dt}(\cdot) = \lambda \frac{d}{d\tau}(\cdot) \Rightarrow \frac{de}{dt} = \lambda \frac{de}{d\tau} = \lambda \cdot \dot{e}, \quad \frac{dw}{dt} = \lambda \frac{dw}{d\tau} = \lambda \cdot \dot{w}
$$

(4.2)

Substituting this into the system dynamics (4.1), we obtain

$$
\begin{align*}
\lambda \dot{e} &= f(e, w) \\
\dot{w} &= g(e, w)
\end{align*}
$$

(4.3)

Note the time variable in the above equation is $\tau$ instead of $t$, while the states $e$ and $w$ still represent fast and slow dynamics respectively.

The system solution can be expressed as a power series in $\lambda$

$$
\begin{align*}
e(t) &= e_0(t) + \lambda e_1(t) + 0.5 \lambda^2 e_2(t) + \ldots \\
w(t) &= w_0(t) + \lambda w_1(t) + 0.5 \lambda^2 w_2(t) + \ldots
\end{align*}
$$

(4.4)

With the above series truncated to the first two terms, we substitute the assumed solutions given by (4.4) into the system dynamic equation (4.3). Comparing terms of the same order in $\lambda$ on both
Chapter 4  The Influence of Payload Variation on System Performance

sides of these equations, the zero order dynamics and first order dynamics are obtained

\[ \dot{e}_0 = f(e_0, w_0) \quad (4.5) \]
\[ \dot{w}_0 = 0 \quad (4.6) \]
\[ \dot{e}_1 = \frac{\partial f}{\partial e_0} e_1 + \frac{\partial f}{\partial w_0} w_1 \quad (4.7) \]
\[ \dot{w}_1 = g(e_0, w_0) \quad (4.8) \]

where \( f_0 = f(e_0, w_0) \). Now we can find the bound on \( e, e \), separately and obtain the bound on the error with

\[ \|e\| \leq \|e_0\| + \lambda \cdot \|e_1\| \quad (4.9) \]

This perturbation approach enables us to transform the complete system into two relatively simple dynamic systems.

4.2.2 Solution Bound

In this section, the norm bound on the zero and first order of solution will be given first, and then the bound of the solution will be obtained by the inequality given by equation (4.9). Here the zero order dynamics is the robot control system with PD and neural network control as feedback. Note that there are no weight dynamics in the zero order dynamics, i.e., the feedback gain is fixed, but still nonlinear.

Zero Order Solution Bound

An assumption is made first, i.e., the inertia matrix and the Coriolis and centrifugal term are bounded as follows [Spong and Vidyasagar, 1987].

\[ M_1 \leq \|H^{-1}(q)\|_\infty \leq M_2 \leq \infty \]
\[ \|\Delta C\|_\infty \leq \delta \|X\|_\infty + \rho \quad (4.10) \]
where:

\[ M_1, M_2 = \text{positive constants}, \]

\[ \delta, \rho = \text{nonnegative constants}, \]

\[ E = H \hat{H} - I \text{ and } \|E\| \leq \alpha, \]

\[ X = \text{robot state, } X = [q \dot{q}]^T \]

The norms of the vector and matrix quantities in the above equations are the truncated \( L_\infty \) norm, as defined in [Spong and Vidyasagar, 1987]

\[ \|f(t)\|_{L_\infty} = \begin{cases} \text{ess sup}_{t \in [0, T]} \|f(t)\|, & t < T \\ 0, & t \geq T \end{cases} \]

From equation (4.1) and (4.5), the zero dynamics can be expressed as follows through mathematical manipulation

\[ e_0 = Ge, \quad (4.11) \]

\[ \varepsilon = \eta + u \]

\[ u = Ke_0 + v \]

\[ \eta = Eu + \eta_0 \]

where:

\[ G = (sI - A)^{-1}B, \]

\[ \eta_0 = E\dot{q}^d + H^{-1}\Delta C. \]

Through algebraic operations, the following is obtained

\[ e_0 = P_1\eta + P_1v \]

\[ u = P_2\eta + P_2v \]

where:
\[ \eta = \text{system uncertainty caused by estimation of robot parameters}, \]
\[ \nu = \text{neural network control output}, \]
\[ P_1 = (I - Gk)^{-1} G, \]
\[ P_2 = K(I - Gk)^{-1} G, \]
\[ P_3 = K(I - Gk)^{-1} G + I, \]
\[ K = \begin{bmatrix} k_p & \Theta \\ \Theta & k_\nu \end{bmatrix} \]

Since the \( L_\infty \) norm of a transfer matrix is \( \|P\|_\infty \leq \sup \frac{\|Px\|_\infty}{\|x\|_\infty} \), then \( \|Px\|_\infty \leq \|P\|_\infty \cdot \|x\|_\infty \).

Consequently, we have
\[ \begin{align*}
\|e_0\|_{T_0} & \leq \beta_1 \|\eta\|_{T_0} + \beta_2 \|\nu\|_{T_0} \\
\|u\|_{T_0} & \leq \beta_2 \|\eta\|_{T_0} + \beta_3 \|\nu\|_{T_0}
\end{align*} \quad (4.16) \]
\[ \quad (4.17) \]
where \( \beta_i = \|P_i\|_\infty \). Further, through mathematical manipulation, the following is obtained
\[ \begin{bmatrix} \|\eta\|_{T_0} \\ \|u\|_{T_0} \end{bmatrix} \leq \begin{bmatrix} M_2 \delta \beta_1 & \alpha \\ \beta_2 & 0 \end{bmatrix} \begin{bmatrix} \|\eta\|_{T_0} \\ \|u\|_{T_0} \end{bmatrix} + \begin{bmatrix} M_2 \delta \beta_1 \phi + b \\ \beta_2 \phi \end{bmatrix} \quad (4.18) \]

Denote \( Q = \begin{bmatrix} M_2 \delta \beta_1 & \alpha \\ \beta_2 & 0 \end{bmatrix} \), then if \( \Delta = \det(I - Q) = (1 - M_2 \delta \beta_1 - \alpha \beta_2) > 0 \), i.e.,
\[ k_m > \frac{M_2 \delta}{1 - \alpha} \quad (4.19) \]
the uncertainty \( \eta \) and control \( u \) are bounded under the small gain theory [Spong and Vidyasagar, 1987], where \( k_m \) is the maximum of PID gains (See details in Appendix-2). Therefore
\[ \|\eta\|_{T_0} \leq \frac{M_2 \delta \beta_1 \phi + b + \beta_2 \beta_2 \phi}{\Delta} \quad (4.20) \]
Substituting back into (4.16), we obtained the zero order error bound

$$\|e_0\|_{r_0} \leq \frac{\beta_1}{\Delta} (M_2 \delta \beta_1 \phi + b + \beta_2 \beta_1 \phi) + \beta \phi$$  

(4.21)

where:

$$b = \alpha \cdot \|\tilde{x}_0\|_{r_0} + M_2 \cdot \delta \cdot \|x_0\|_{r_0} + M_2 \cdot \rho$$

$$\phi = \text{upper bound on the neural network output.}$$

First Order Solution Bound

The first order solution is also norm bounded. In order to show this, the weights have to be shown to be bounded first. From equation (4.8), we have $\dot{w}_i = g(e_o, w_i)$. The derivative of the weights is bounded first, and consequently the weights themselves are bounded. Through mathematical manipulation (See details in Appendix-1), the weights are bounded as

$$\|w^{th}\|_{r_0} \leq \omega_1, \quad \|w^{ho}\|_{r_0} \leq \omega_2$$  

(4.22)

where:

$$\omega_1, \omega_2 = \text{constants}, \quad \omega = [\omega_1 \omega_2]^T$$

By defining $G_e = \frac{\partial f_o}{\partial e_o}$ and $G_w = \frac{\partial f_o}{\partial w_o}$, equation (16) is transformed to be

$$e_i = G_e G_w w_i$$  

(4.23)

where $G_e = (sI - G_e)^{-1}$ and $\|G_i\|$ is bounded. Since the original system is a (BIBO) stable system, the first order system is a BIBO stable system and is bounded. Denote this bound as $\chi$.

Since $G_w = \frac{\partial f_0}{\partial w_0} = (E + 1) \phi(v) b_j$ (Here $\phi(v)$ is the derivative of the neural network activation function and is bounded as $0.5 \phi$ [Haykin, 1994]), thus $\|G_w\| \leq 0.5 \cdot (\alpha + 1) \cdot \phi^2$, and also weights
are bounded in (4.22), hence the first order of solution $e_i$ is bounded by combining the bounds of each term in (4.23).

$$\|e_i\|_{\text{F}_\infty} \leq \|G_i\| \cdot \|G_w\| \cdot \|w\|_{\text{F}_\infty} = 0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega$$

(4.24)

**Solution Bound**

Finally the bound of the system solution is obtained by combining the results from zero order norm bound (4.21) and first order norm bound (4.24)

$$\|e\|_{\text{F}_\infty} \leq \left\{ \frac{\beta_1}{\Delta} \left( M_2 \delta \beta_2 \phi + b + \beta_2 \beta_3 \phi \right) + \lambda \cdot (0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega) \right\}$$

(4.25)

From the above equation, it is clear that the learning rate parameter, $\lambda$, has no influence on the first term of the system error norm, which is obtained from the static dynamics, to be defined later. The learning rate affects only the second term, which is obtained from the system dynamics when the weights are not fixed. Through mathematical analysis, the partial derivative of system error norm with respect to learning rate is positive, hence an increase of learning rate leads to an increase of system error norm.

### 4.2.3 Influence of System Parameters on the System Error Bound

In the following discussion, the influence of the payload mass parameter on the system error bound is investigated. The system dynamics we consider is the static state dynamics after the neural network is trained, i.e., the weights are fixed after the training process. Hence, there is no weight dynamics, and consequently the system can be represented by the same dynamical equation that represents the zero order dynamics (4.5). First, the system error norm is bounded by triangle inequality (4.26). Then the partial derivative of system error with respect to the payload
mass is defined as the sensitivity function and the associated sensitivity equation (4.27.2) is formulated. Furthermore, the solution bound approach is employed again to bound the sensitivity function (4.32). Finally the system error norm bound (4.33) will be obtained from the triangle inequality.

The system error norm can be expressed as

$$\left\| e(m^* + \Delta m) \right\|_{\tau_\infty} \leq \left\| e(m^*) \right\|_{\tau_\infty} + \left\| \frac{\partial e}{\partial m} \right\| \cdot \left\| \Delta m \right\|_{\tau_\infty}$$

where:

$m^*$ = payload mass at which the neural network is trained,

$\Delta m$ = payload variation, $\left\| \Delta m \right\|_{\tau_\infty} = |\Delta m|$

$$\left\| e(m^*) \right\|_{\tau_\infty} \equiv \left\| e_0 \right\|_{\tau_\infty} = \frac{B}{\Delta} (M_1 \beta \phi + b + \beta_2 \beta_3 \phi)$$

Now the task is to calculate the truncated $L_\infty$ norm of the sensitivity of system error $e$ with respect to $m$. The sensitivity analysis and the solution bound method are employed to accomplish the above task (See details in Appendix-2).

Taking the partial derivative $\frac{\partial (\cdot)}{\partial m}$ on both sides of the system error equation $\dot{e} = Ae + B\dot{\xi}$, we have

$$\frac{\partial \dot{e}}{\partial m} = A \frac{\partial e}{\partial m} + B \frac{\partial \dot{\xi}}{\partial m}$$

Define $\sigma = \frac{\partial e}{\partial m}$, then

$$\frac{\partial \dot{e}}{\partial m} = \frac{\partial}{\partial m} (de / dt) = \frac{d}{dt} (\partial e / \partial m) = \dot{\sigma}$$

hence the sensitivity equation is obtained.
\[ \dot{\sigma} = A \sigma + B \frac{\partial \xi}{\partial m} \]  \hspace{1cm} (4.27.2)

where:

\[ \frac{\partial \xi}{\partial m} = (E \dot{K} + H^{-1} \dot{H} \phi \phi \dot{w}^h \dot{w}^{h*}) \sigma + \frac{\partial H^{-1}}{\partial m} (\dot{H} \dot{q}^d + Ke + v) + \Delta C \]

and \( \frac{\partial H^{-1}}{\partial m} \) will be calculated by payload dynamics later.

The sensitivity equation can be transformed to the following form

\[ \sigma = G \varepsilon \]
\[ \varepsilon = \eta_\sigma + u_\sigma \]
\[ u_\sigma = K_\sigma \sigma + K_{\sigma v} v \]
\[ \eta_\sigma = E \dot{K} \sigma + \eta_{\sigma 0} \] \hspace{1cm} (4.28)

Through mathematical manipulation, the following is obtained

\[ \| \sigma \|_{T_\sigma} \leq \beta_{1 \sigma} \| \eta_\sigma \|_{T_\sigma} + \beta_{2 \sigma} \| v \|_{T_\sigma} \] \hspace{1cm} (4.29)
\[ \| u_\sigma \|_{T_\sigma} \leq \beta_{3 \sigma} \| \eta_\sigma \|_{T_\sigma} + \beta_{4 \sigma} \| v \|_{T_\sigma} \] \hspace{1cm} (4.30)

In the following, CRS A460 robot parameters are used for calculation purpose. Through mathematical manipulation, we obtain

\[ \frac{k_m}{\phi^2} < \frac{0.9375(\alpha + 1)}{\alpha} \] \hspace{1cm} (4.31)

Then, the uncertainty \( \eta_\sigma \) of the sensitivity system is bounded, and consequently the sensitivity function is bounded as

\[ \| \sigma \|_{T_\sigma} \leq \frac{\beta_{1 \sigma} (\alpha k_m \beta_{2 \sigma} \phi + \| \eta_{\sigma 0} \|_{T_\sigma} + \beta_3 \sigma \beta_4 \sigma )}{1 - \alpha k_m \beta_1} + \beta_{2 \sigma} \phi \] \hspace{1cm} (4.32)

where:
The Influence of Payload Variation on System Performance

\[ \|e\|_{T_0} = \left\| \frac{\partial H^{-1}}{\partial m} \right\|_{T_0} (\|\dot{\mathbf{q}}\|_{T_0} + \|K\mathbf{e}\|_{T_0} + \|\mathbf{v}\|_{T_0}) + \|\Delta C\| \]

can be obtained from equation (29), which is robot system error norm.

\[ \left\| \frac{\partial H^{-1}}{\partial m} \right\|_{T_0} \]

can be calculated through payload dynamics (See details in Appendix-3), and the result is given as

\[ \left\| \frac{\partial H^{-1}}{\partial m} \right\|_{T_0} = \frac{R^2[I_{23}^2 + (I_{22} + 0.25m^*R^2)^2 + I_{22} + I_{33} + 0.5m^*R^2]}{2[(I_{22} + 0.25m^*R^2)(I_{23} + 0.25m^*R^2) - I_{23}^2]^2} \]

where:

- \( R \) = radius of the round plate form payload,
- \( m^* \) = payload mass at which the neural network is trained,
- \( I_{ij} \) = element of the inertia matrix of CRS A460 robot.

Substituting (4.32) into (4.26), we obtain the bound on system error norm with the payload variation

\[ \|e(m^* + \Delta m)\|_{T_0} \leq \frac{\beta_1}{\Delta} (M_2 \delta \beta_1 \phi + b + \beta_2 \beta_3 \phi) + \beta_1 \phi + \frac{\beta_1 \sigma (\alpha_k \beta_2 \phi + \|\mathbf{e}\|_{T_0} + \beta_3 \sigma \beta_4 \phi)}{1 - \alpha_k \beta_1 \sigma} + \beta_2 \phi \] (4.33)

From above analysis, the conclusion that the variation of payload increases the norm bound on the system error regardless of the sign of the payload variation (positive or negative) is obtained. This is coincident with the neural network theory, i.e., the neural network performs the best at the training point. Any deviation from the training point will lead to increase of neural network output error, which in turn increases the complete system error. Note the maximum of the PD gains must
satisfy both equation (4.19) and (4.31) simultaneously

\[
\frac{M_2 \delta}{1 - \alpha} < k_m < \frac{0.9375(\alpha + 1)\phi^2}{\alpha}
\]  

(4.34)
4.3 Experimental Results

In this section, we present the results of experiments, which are carried out on the CRS A460 industrial robot as described in Chapter 3, to verify the theoretical analysis. First a theoretical value of error norm of the complete system is calculated according to the proposed methodology. Then the value of system error norm, which is obtained from experiments on the robot, will be compared to the predicted value. Reasons for discrepancies between the theoretical prediction and the experimental results are provided at the end of section.

4.3.1. Experiment Process

The robot arm, which holds a circular plate as payload, is controlled to follow a desired trajectory, given as a power series in t as in [Ogilvie, 1998]

\[ q^d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \] 

(4.35)

where:

\[ a_0 = a_1 = 0, \]
\[ a_2 = -3.11 \times 10^{-2}, \]
\[ a_3 = 2.44 \times 10^{-3}. \]

The experiments start by training the neural network with payload of 2 kg. After 20 trials, the system error becomes stable, i.e., the learning process finished, the weights are fixed. The experiment are continued for another 10 trials with the payload of 2 kg. From 31 to 80 trials, the experiment is conducted with varied payload masses in the order of 1 kg, 0 kg, 3 kg, 4 kg. The data
is saved and the error norm is calculated, and shown in Table 4.3, with the experimental results shown in Figure 4.4.

4.3.2. Experimental Parameters

From equation (4.33), we see the system error norm is influenced by two groups of parameters, the robot parameters and the neural network parameters. The values of system parameters are given in Table 1. The learning rate of neural network is chosen as $3.0 \times 10^{-8}$. The scale factor of the neural network output is $\Phi = 40$, since under this value the influence of the neural network on the error reduction of the whole system is better than other cases [Ogilvie, 1998]. The weights between input and hidden layers are limited within the range of 2.5 after training, while the absolute value of the weights between the hidden and output layers are smaller than 1.5. The PID parameters are bounded as $\beta_1 = \| (I - GK)^{-1} G \|_\infty \approx 1 / \| K \|_\infty = 0.00133$ and $\beta_3 = \| K(I - GK)^{-1} G \|_\infty \approx 0.992$. The theoretical system error norms calculated by MATHEMATICA software for each payload are given in Table 4.2.

4.3.3 Experimental Results

The experimental results show that the system error norm increases when the robot payload deviates from the training point as shown in Figure 4.1 to Figure 4.6, which is coincident with our prediction. Furthermore, after plotting the experimental result and the theoretical prediction simultaneously in Figure 4.8, we notice that the difference between the above two values lies within the range of 7%. Hence we can conclude that the proposed methodology of performance analysis is valid for the neural network robot control system.
There are two reasons that result in discrepancies between the predicted value and the experimental results. From theoretical point of view, the approximation employed to calculate the sensitivity function norm leads to part of the difference. Since \( m^* \) is used to represent \((m^* + \varepsilon \cdot \Delta m)\) in the calculation (4.26)

\[
\|e(m^* + \Delta m)\| \leq \|e(m^*)\| + \| \frac{\partial e}{\partial m}(m^* + \varepsilon \cdot \Delta m) \| \cdot \| \Delta m \| \equiv \|e(m^*)\| + \| \frac{\partial e}{\partial m}(m^*) \| \cdot \| \Delta m \| \quad (4.36)
\]

where \( 0 < \varepsilon < 1 \), the prediction we make is not the exact value that the system error norm should be, theoretically. In terms of experimental considerations, the values of parameters we use for the theoretical prediction will not be exactly the same as those in the experimental process, and this also contributes to the discrepancy.

**Remark:**

The values of parameters in our experiment can be divided into three groups: the robot parameter values (such as Coriolis term), the neural network parameter values (such as output scale factor) and the PID gains. The robot parameter values are determined according to the work of [Baines, 1995]. The neural network and PID parameter values are determined according to the experiments performed on the same neural network control system by [Ogilvie, 1998]. The sensitivity test of the parameter value picking has been done. The system error norm is calculated when the values of the parameters are perturbed around the values, which are used in the experiments. The results show that all the error norms obtained with varied parameter values generate the same shape of curve as shown in Figure 4.5. Besides the maximum variation of system error norm is less than 10% when the perturbation of parameter value is 5%. The results of sensitivity test indicate that the parameter values of the experiment are not fortuitous choice.
4.4 Summary

In this chapter, a systematic methodology for the quantitative performance analysis of the neural network robot control system is established. Insight of how the payload mass influence the complete system performance, measured by the system error norm, is obtained through rigorous mathematical relations. The developments are significant, since it is the quantification of neural network controller performance that is critical for commercialization of neural network controllers in the robot industry.

The experimental results prove the validity of this methodology. The use of industrial standard equipment in the experimental process brings considerable weight to the results achieved. These results form the basis for the justification that the results could be applied in a real task execution.

Moreover, the approach can be applied to the performance analysis of other system parameters, associated with Coriolis and centrifugal terms, since the theoretical mechanism to investigate the influence of these parameters on the system performance is the same or similar as that of payload mass.
Table 4.1. Bounds of robot and neural network parameters

<table>
<thead>
<tr>
<th>Neural network parameters</th>
<th>Robot parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>3.0 \times 10^{-8}</td>
</tr>
<tr>
<td>Scale factor</td>
<td>40</td>
</tr>
<tr>
<td>Bound of weights (ω₁)</td>
<td>2.5</td>
</tr>
<tr>
<td>Bound of weights (ω₂)</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 4.2. Predicted value of system error norm

<table>
<thead>
<tr>
<th>Payload Mass (kg)</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass Variation Δm</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>System Error Norm  (|e|_w)</td>
<td>0.2998</td>
<td>0.2612</td>
<td>0.2241</td>
<td>0.2612</td>
<td>0.2998</td>
</tr>
</tbody>
</table>

Table 4.3. Experimental result of system error norm

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>00-30</th>
<th>30-50</th>
<th>50-60</th>
<th>60-70</th>
<th>70-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload (kg)</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>3.0e-8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>State of Weights</td>
<td>Training</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Fixed</td>
</tr>
<tr>
<td>Static Error Norm</td>
<td>0.2208</td>
<td>0.2667</td>
<td>0.2956</td>
<td>.02808</td>
<td>0.3226</td>
</tr>
</tbody>
</table>
Figure 4.1. System error norm when the neural network is trained

Figure 4.2. System error norm with payload of 2 kg
Chapter 4  The Influence of Payload Variation on System Performance

Figure 4.3. System error norm with payload of 1 kg

Figure 4.4. System error norm with the payload of 0 kg
Chapter 4  
*The Influence of Payload Variation on System Performance*

![Graph of System Error Norm vs Trial Number for 3 kg payload](image1.png)

*Figure 4.5. System error norm with payload of 3 kg*

![Graph of System Error Norm vs Trial Number for 4 kg payload](image2.png)

*Figure 4.6. System error norm with payload of 4 kg*
Figure 4.7. Experimental result of system error norm

Figure 4.8. The comparison of system error norm
Chapter 5

Influence of Link Mass Variation on System Performance

5.1 Introduction

In this chapter, the influence of the robot link mass variation on the neural network control robot system performance is investigated. The inertia uncertainty of robot system is caused by two factors. One factor is the payload mass, whose influence on the complete system error norm has been investigated in Chapter 4. Another is the link mass variation. As we know, the link mass itself remains the same once a robot is manufactured. However, some attached elements may be added, changing the effective mass. Since sensors and other equipments are normally attached to links, this leads to link mass variation, and consequently affects the robot dynamics and in turn the system performance. Another possibility is the adoption of new grippers, which is a kind of end-of-arm tooling update, and certainly also causes link mass variation. Hence, the link mass should be addressed as a parameter of neural network control robot system. In the following, perturbation
theory [Gardin, 1992], the solution bound method [Spong and Vidyasagar, 1987] and the sensitivity approach [Frank, 1978] are employed to formulate a systematic methodology to investigate the influence of link mass on the performance of the robotics system with a neural network control.

The rest of this chapter is organized as follows. In Section 5.2, a systematic methodology to investigate the influence of payload on the system performance is developed. Section 5.3 illustrates the experimental results to verify the theoretical analysis of previous section. A summary is provided in Section 5.4.

5.2 Theoretical Analysis

The methodology is developed in the following order to achieve our objectives. In Section 5.2.1, the solution of system equation is expressed in a power series of learning rate parameter \( \lambda \), based on perturbation theory. The bound of system solutions is obtained by employing the solution bound approach in Section 5.2.2. In Section 5.2.3, we explore the relation between the performance and the link mass of the neural network control robot system. For purposes of conciseness, Appendix-4 is provided to give details of the calculations.

5.2.1 Separation of Solutions

In this section, the solution of the system equation is expressed in a power series in \( \lambda \). The learning rate of a stable neural network can be treated as a small parameter of the complete system with two time-scale dynamics, i.e., fast robot dynamics and slow neural network
dynamics. By adopting this approach, the influence of learning rate on the error bound of the system solution can be shown explicitly.

From equation (2.10), the complete system dynamics can be expressed as

\[
\begin{align*}
\dot{e}_s &= A e_s + B \xi(e_s, w_s) = f(e_s, w_s) \\
\dot{w}_s &= \lambda \cdot g(e_s, w_s)
\end{align*}
\] (5.1)

Through mathematical manipulations (See details in Section 4.2.1), the zero order dynamics and first order dynamics are obtained

\[
\begin{align*}
\dot{e}_0 &= f(e_0, w_0) \quad (5.2) \\
\dot{w}_0 &= 0 \quad (5.3)
\end{align*}
\]

\[
\begin{align*}
\dot{e}_1 &= \frac{\partial f}{\partial e_0} e_1 + \frac{\partial f}{\partial w_0} w_1 \quad (5.4) \\
\dot{w}_1 &= g(e_0, w_0) \quad (5.5)
\end{align*}
\]

where \( f_s = f(e_s, w_s) \). Now we can find the bound on \( e_s, e_i \) separately and obtain the bound on the error with

\[
\|e\| \leq \|e_0\| + \lambda \cdot \|e_1\| \quad (5.6)
\]

This perturbation approach enables us to transform the complete system into two relatively simple dynamic systems.

5.2.2 Solution Bound

In this section, the norm bound on the zero and first order of solution will be given first, and then the bound of the solution will be obtained by the inequality given by equation (5.6). Here the zero order dynamics is the robot control system with PD and neural network control as feedback. Note that there are no weight dynamics in the zero order dynamics, i.e., the feedback gain is fixed,
but still nonlinear.

**Zero Order Solution Bound**

After the inertia and centrifugal matrices are bounded using several assumptions, as in Section 4.2.2, the zero dynamics can be expressed as follows through mathematical manipulation

\[
e_0 = P_1\eta + P_1\nu
\]

\[
u = P_2\eta + P_2\nu
\]

where:

\(\eta\) = system uncertainty caused by estimation of robot parameters,

\(\nu\) = neural network control output,

\[P_1 = (I - GK)^{-1}G,\]

\[P_2 = K(I - GK)^{-1}G,\]

\[P_3 = K(I - GK)^{-1}G + I,\]

\[K = \begin{bmatrix} k_p & \Theta \\ \Theta & k_v \end{bmatrix}\]

Taking the \(L_\infty\) norm of a transfer matrix on both side of the above equation, we obtain

\[
\|e_0\|_{L_\infty} \leq \beta_1\|\eta\|_{L_\infty} + \beta_2\|\nu\|_{L_\infty}
\]

(5.9)

\[
\|\nu\|_{L_\infty} \leq \beta_2\|\eta\|_{L_\infty} + \beta_3\|\nu\|_{L_\infty}
\]

(5.10)

where \(\beta_i = \|P_i\|_{\infty}\). Though mathematical manipulations that given,

\[
k_m > \frac{M_2\delta}{1 - \alpha}
\]

(5.11)

the zero order error is bounded as

54
where:

\[ b = \alpha \cdot \| \tilde{q} \|_{\tau_0} + M_2 \cdot \delta \cdot \| x_d \|_{\tau_0} + M_2 \cdot \rho \]

\( \phi \) = upper bound on the neural network output.

**First Order Solution Bound**

Through a similar process (See details in Section 4.2.2), the first order solution is bounded as

\[ \| e_1 \|_{\tau_0} \leq \| G_1 \| \cdot \| G_w \| \cdot w \|_{\tau_0} = 0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega \]  

(5.13)

**Solution Bound**

Finally the bound of the system solution is obtained by combining the results from zero order norm bound (5.12) and first order norm bound (5.13)

\[ \| e \|_{\tau_0} \leq \left\{ \frac{\beta_1}{\Delta} (M_1 \beta_1 \phi + b + \beta_2 \beta_3 \phi) + \beta_1 \phi \right\} + \lambda \cdot (0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega) \]  

(5.14)

### 5.2.3 Influence of Link Mass on the System Error Bound

In the following discussion, the influence of the link mass parameter on the system error bound is investigated. The system dynamics we consider is the static state dynamics after the neural network is trained, i.e., the weights are fixed after the training process. Hence, there is no weight dynamics, and consequently the system can be represented by the same dynamical equation that represents the zero order dynamics (5.2). First, the system error norm is bounded by triangle inequality (5.6). Then the partial derivative of system error with respect to the link mass is defined as the sensitivity function and the associated sensitivity equation (5.16) is formulated. Furthermore, the solution bound approach is employed again to bound the sensitivity
function (5.17). Finally the system error norm bound (5.19) will be obtained from the triangle inequality.

In the process of calculating \( \left\| \frac{\partial H^{-1}}{\partial L_m} \right\|_a \), the sensitivity of system error with respect to the link mass \( \frac{\partial H^{-1}}{\partial L_m} \) must be obtained first. This sensitivity is a partial derivative of a matrix with respect to a vector, thus leading to a tensor. Also the sensitivity of system error with respect to the link mass variation \( \frac{\partial e}{\partial L_m} \) is a matrix instead of vector. Since the norms of either a matrix or a vector are scalars, our results are scalars.

The system error norm considering the influence of the link mass variation can be expressed as

\[
\left\| e(L_m + \Delta L_m) \right\|_{\tau_0} = \left\| e(L_m) + \frac{\partial e(L_m)}{\partial L_m} \Delta L_m \right\|_{\tau_0} \leq \left\| e(L_m) \right\|_{\tau_0} + \left\| \frac{\partial e}{\partial L_m} \right\|_{\tau_0} \left\| \Delta L_m \right\|_{\tau_0} (5.15)
\]

where:

- \( L_m = \) link mass vector at which the neural network is trained, \( L_m = [m_1, m_2, m_3]^T \)
- \( \Delta L_m = \) link mass variation, \( \Delta L_m = [\Delta m_1, \Delta m_2, \Delta m_3]^T \)
- \( m_i = \) mass of link \( i \),
- \( \Delta m_i = \) variation of the mass of link \( i \),
- \( \left\| e(L_m) \right\|_{\tau_0} \equiv \| e_0 \|_{\tau_0} \)

Now the task is to calculate the \( L_{\tau_0} \) norm of the sensitivity of system error \( e \) with respect to \( L_m \). The sensitivity analysis and the solution bound method are employed to accomplish this.
Define $\sigma = \frac{\partial e}{\partial L_m}$, then take partial derivative with respect to $L_m$ on both sides of the system error equation (2.10), the sensitivity equation is obtained

$$\dot{\sigma} = A\sigma + B\frac{\partial \xi}{\partial L_m} \quad (5.16)$$

where:

$$\frac{\partial \xi}{\partial L_m} = (NK + H^{-1}\dot{H}\phi \phi' + W^h W^h)\sigma + \frac{\partial H^{-1}}{\partial L_m}(\dot{\xi}^d + Ke + v + \Delta C)$$

$\phi'$ = derivative of the activation function,

$$\frac{\partial H^{-1}}{\partial L_m}$$ will be calculated later through link mass dynamics.

Through similar mathematical manipulation as that of bounding the zero order solution, the sensitivity function is bounded as

$$\|\sigma\|_{\tau_\infty} \leq \beta_{1\sigma} \left(\frac{\alpha_k \beta_2 \phi + \|\eta_\sigma\|_{\tau_0} + \beta_3 \beta_1 \phi}{1 - \alpha_k \beta_1 \phi} + \beta_2 \phi\right) \quad (5.17)$$

where:

$$\|\eta_\sigma\|_{\tau_0} = \left\|\frac{\partial H^{-1}}{\partial L_m}\right\|_{\tau_{\infty}} \left\|\dot{H}\right\| \left(\|\dot{\xi}^d\|_{\tau_0} + \|K\|\|e_0\|_{\tau_0} + \|v\|\|e_0\|_{\tau_0} + \|\Delta C\|\right) \quad (5.18)$$

$$\|e_0\|_{\tau_0}$$ can be obtained from equation (5.12).

Substituting (5.18) into (5.17), we obtain the bound on the system error norm including link mass variation

$$\|\epsilon\|_{\tau_\infty} \leq \left\{\frac{\beta_1}{\Delta} (M_2 \phi + b + \beta_2 \phi) + \beta_1 \phi\right\} + \left\{\frac{\beta_{1\sigma} (\alpha_k \beta_2 \phi + \beta_3 \beta_1 \phi)}{1 - \alpha_k \beta_1 \phi} + \beta_2 \phi\right\} \quad (5.19)$$
Chapter 5  \textit{The Influence of Link Mass Variation on System Performance}

\begin{equation}
\frac{4d_2 \cdot l_{2c} + 2d_2^2 l_{2c}^2 - 2J_{111}^2 + J_{222}^2 - d_1^2 m_f l_{2c}^2 + 2J_{333}}{1 - d_1 k_m \beta_{1\sigma}} \cdot |\Delta L_m| \tag{5.19}
\end{equation}

where:

\[ |\Delta L_m| = \text{maximum of link mass variation}, \quad \max_i |\Delta L_m| = \|\Delta L_m\|_{T_0} \]

\[ l_{ic} = \text{signed distance from the origin of link} \ i \ \text{along the} \ a_i \ \text{axis to the center of mass of the link} \ i, \]

\[ J_{jk} = \text{pseudo-inertia element in row} \ j \ \text{and column} \ k \ \text{for link} \ i. \]

\[ d_i = \text{distance between the axes of joint} \ i \ \text{and joint} \ i+1. \]

From the above analysis, the conclusion is reached that the variation of link mass increases the norm bound on the system error regardless of the sign of the link mass variation. This is coincident with the neural network theory, i.e., the neural network performs at its best at the training point, any deviation from the training point will lead to an increase of neural network output error, which in turn increases the overall system error.

\textit{Remark:}

The basic methodology used for link mass vector analysis is similar to that of the payload mass, while since the variable here is a vector (link mass \( L_m \)) instead of scalar (payload \( m \)), the mathematical process is much more complicated. But as we mentioned, norms of either matrices or vectors are scalars, so our results are still in scalar from, which again illustrates the advantage of using error norm as the system performance measure.
5.3 Experimental Results

In this section, we present the results of experiments, which are carried out on the CRS A460 industrial robot as described in Chapter 3, to verify the theoretical analysis. First a theoretical value of error norm of the complete system is calculated according to the proposed methodology. Then the value of system error norm, which is obtained from experiments on the robot, will be compared to the predicted value.

5.3.1. Experiment Process

The robot arm, with a circular plate mass attached to each link as shown in Figure 5.1, is controlled to follow a desired trajectory, given as a power series in $t$ as in [Ogilvie and Mills, 1998]

$$q^d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

where:

$$a_0 = a_1 = 0,$$

$$a_2 = -3.11 \times 10^{-2},$$

$$a_3 = 2.44 \times 10^{-3}.$$  

The experiments start by training the neural network with link mass vector of $[2\text{kg}, 2\text{kg}, 2\text{kg}]^T$. After 20 trials, the system error becomes stable, i.e., the learning process finished, the weights are fixed. From 21 to 90 trials, the experiment is conducted with varied link mass vector in the order of $[2\text{kg}, 2\text{kg}, 2\text{kg}], [2\text{kg}, 2\text{kg}, 3\text{kg}], [2\text{kg}, 2\text{kg}, 4\text{kg}], [0\text{kg}, 0\text{kg}, 0\text{kg}]$. The system error norm is
calculated and shown in Table 5.3, with the experimental results shown in Figure 5.4.

5.3.2. Experimental Parameters

From equation (5.19), we see the system error norm is influenced by two groups of parameters, the robot parameters and the neural network parameters. The values of system parameters are given in Table 5.1. The learning rate of neural network is chosen as \(3.0 \times 10^{-8}\). The scale factor of the neural network output is \(\Phi = 40\), since under this value the influence of the neural network on the error reduction of the whole system is better than other cases [Ogilvie, 1998]. The weights between input and hidden layers are limited within the range of 2.5 after training, while the absolute value of the weights between the hidden and output layers are smaller than 1.5. The PID parameters are bounded as 
\[
\beta_1 = \| (I - GK)^{-1} G \|_\infty \leq 1 / \| K \|_\infty = 0.00133 \\
\beta_3 = \| K(I - GK)^{-1} G \|_\infty \leq 0.992 
\]
The theoretical system error norms calculated by MATHEMATICA software for each link mass variation are given in Table 5.2.

5.3.3 Experimental Results

The experimental results show that the system error norm increases when the robot link mass deviates from the training point as shown in Figure 5.2 to Figure 5.6, which is coincident with our prediction. Furthermore, after plotting the experimental result and the theoretical prediction simultaneously in Figure 5.8, we notice that the difference between the above two values lies within the range of 21%. Hence we can conclude that the proposed methodology of performance analysis is valid for the neural network robot control system.

There are two reasons that result in discrepancies between the predicted value and
the experimental results. From a theoretical point of view, the approximation employed to calculate the sensitivity function norm leads to part of the difference. Since \( L_m^\ast \) is used to represent \((L_m^\ast + \varepsilon \cdot \Delta L_m)\) in the calculation (5.15)

\[
\left\| e(L_m^\ast + \Delta L_m) \right\| \leq \left\| e(L_m^\ast) \right\| + \left\| \frac{\partial e}{\partial L_m} \right\| (L_m^\ast + \varepsilon \cdot \Delta L_m) \cdot \left\| \Delta L_m \right\| = \left\| e(L_m^\ast) \right\| + \left\| \frac{\partial e}{\partial L_m} \right\| L_m^\ast \cdot \left\| \Delta L_m \right\| \tag{5.22}
\]

where \( 0 < \varepsilon < 1 \), the prediction we make is not the exact value that the system error norm should be, theoretically. In terms of experimental considerations, the values of parameters we use for the theoretical prediction will not be exactly the same as those in the experimental process, and this also contributes to the discrepancy.

**Remark:**

The values of parameters in our experiment can be divided into three groups: the robot parameter values (such as Coriolis term), the neural network parameter values (such as output scale factor) and the PID gains. The robot parameter values are determined according to the work of [Baines, 1995]. The neural network and PID parameter values are determined according to the experiments performed on the same neural network control system by [Ogilvie, 1998]. The sensitivity test of the parameter value picking has been done, such as the choice of scale factor of the neural network output. The system error norm is calculated when the values of the parameters are perturbed around their nominal values, which are used in the experiments. The results show that all the error norms obtained with varied parameter values generate the same shape of curve as shown in Figure 5. Besides, the maximum variation of the system error norm is less than 10% when the perturbation of parameter value is 5%. The results of the sensitivity test indicate that the
parameter values of the experiment are not simply the result of a fortuitous choice.

5.4 Summary

In this chapter, a systematic methodology for the quantitative performance analysis of the neural network robot control system is established. Insight of how the variation of link mass parameter influence the complete system performance is obtained through rigorous mathematical relations. The developments are significant, since it is the quantification of neural network controller performance, which is critical for commercialization of neural network controllers in robot industry. As link mass variation is caused by the update of new equipments, this research result helps us to have a better sense what the system performance will be affected when the new facilities are attached to robots.

The experimental results prove the validity of this methodology. The use of industrial standard equipment in the experimental process brings considerable weight to the results achieved.

Also the approach, proved to be successful in the other parameters too, such as the payload mass [Wang and Mills, 1999], is optimistic to be applied to the performance analysis of other system parameters, such as the coefficients of Coriolis and centrifugal term. Since the theoretical mechanism to investigate the influence of these parameters on the system performance is the same or similar as that of link mass.

Moreover, comparing the system errors from the link mass with those from the payloads, we find that the system error is more sensitive with respect to the link mass variation than to the payload variation. This is a result of the use of the $L_{\infty}$ norm as the measure of parameter
variation. Changes in link mass lead to greater variation in the robot inertia distribution.

Table 5.1. Bounds of robot and neural network parameters

<table>
<thead>
<tr>
<th>Neural network parameters</th>
<th>Robot parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>3.0 \times 10^{-8}</td>
</tr>
<tr>
<td>Scale factor</td>
<td>40</td>
</tr>
<tr>
<td>Bound of weights($\omega$)</td>
<td>2.5</td>
</tr>
<tr>
<td>Bound of weights($\omega$)</td>
<td>1.5</td>
</tr>
<tr>
<td>Link Mass Variation Unit</td>
<td>Upper bound on inertia</td>
</tr>
<tr>
<td>Bound on Coriolis and centrifugal term</td>
<td>0.412</td>
</tr>
</tbody>
</table>

Table 5.2. Predicted value of system error norm

<table>
<thead>
<tr>
<th>Link Mass Vector</th>
<th>[2kg, 2kg, 2kg]</th>
<th>[2kg, 2kg, 3kg]</th>
<th>[2kg, 2kg, 4kg]</th>
<th>[0kg, 0kg, 0kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Mass Variation $A_m$</td>
<td>2kg</td>
<td>3kg</td>
<td>4kg</td>
<td>0kg</td>
</tr>
<tr>
<td>System Error Norm $|e|_\infty$</td>
<td>0.2724</td>
<td>0.3109</td>
<td>0.3500</td>
<td>0.3500</td>
</tr>
</tbody>
</table>

Table 5.3. Experimental result of system error norm

<table>
<thead>
<tr>
<th>Trial No.</th>
<th>00-20</th>
<th>21-30</th>
<th>21-30</th>
<th>31-50</th>
<th>51-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Mass Vector (kg)</td>
<td>[2kg, 2kg, 2kg]</td>
<td>[2kg, 2kg, 2kg]</td>
<td>[2kg, 2kg, 3kg]</td>
<td>[2kg, 2kg, 4kg]</td>
<td>[0kg, 0kg, 0kg]</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>3.0e-8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>State of Weights</td>
<td>Training</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Fixed</td>
</tr>
<tr>
<td>Static Error Norm</td>
<td>0.2454</td>
<td>0.3039</td>
<td>0.3920</td>
<td>0.3326</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.1. CRS A460 robot with the link mass variation

Figure 5.2. System error norm when the neural network is trained
Chapter 5  The Influence of Link Mass Variation on System Performance

Figure 5.3. System error norm with link mass variation vector of [2kg, 2kg, 2kg]

Figure 5.4. System error norm with link mass variation vector of [2kg, 2kg, 3kg]
Chapter 5  The Influence of Link Mass Variation on System Performance

Figure 5.5. System error norm with the link mass variation vector of [2kg, 2kg, 4kg]

Figure 5.6. System error norm within the link mass variation vector of [0kg, 0kg, 0kg]
Chapter 5 The Influence of Link Mass Variation on System Performance

Figure 5.7. Experimental result of system error norm

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Link Mass Variation (kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-30</td>
<td>2.2, 2</td>
</tr>
<tr>
<td>31-40</td>
<td>2.2, 3</td>
</tr>
<tr>
<td>41-60</td>
<td>2.2, 4</td>
</tr>
<tr>
<td>61-90</td>
<td>0.0, 0</td>
</tr>
</tbody>
</table>

Figure 5.8. The comparison of system error norm
NOTE TO USERS

Page(s) missing in number only; text follows. Page(s) were microfilmed as received.

68,69

This reproduction is the best copy available.
Chapter 6

Influence of Trajectory Variation on System Performance

6.1 Introduction

In this chapter, the influence of the robot trajectory variation on the neural network control robot system performance is investigated. There are several factors, which influence the system performance. The influence of the inertia uncertainty of robot system, which is caused by the payload mass and link mass, has been investigated in Chapter 4 and 5. Trajectory variation is the other reason causes the degradation of the system performance. In the trajectory following problem we consider, the neural network is trained on the original trajectory. While the trajectory may be altered due to different execution tasks, this variation leads to trajectory following error, i.e., has an impact on the performance of the neural network control robot system. The research objective of this chapter is to investigate this impact. Now the performance varies with respect to the variable of the system dynamic equation directly, not the parameters of the system. Also, the
trajectory varies after training, the case we study is a generalization problem as shown in Figure 6.1. The solid line represents the trajectory with which the neural network is trained, and the stared line represents the varied path.

In the following, perturbation theory [Gardin, 1992], the solution bound method [Spong and Vidyasagar, 1987] and the sensitivity approach [Frank, 1978] are employed to formulate a systematic methodology to investigate the influence of trajectory variation on the performance of robotics system with a neural network control.

The rest of this chapter is organized as follows. In Section 6.2, a systematic methodology to investigate the influence of payload on the system performance is developed. A summary is provided in Section 6.3.

### 6.2 Theoretical Analysis

The methodology is developed in the following order to achieve our objectives. The complete system equation is provided in Chapter 2. Hence in Section 6.2.1, the solution of the system equation is expressed in a power series of learning rate parameter $\lambda$, based on perturbation theory. The bound of system solutions is obtained by employing the solution bound approach in Section 6.2.2. In Section 6.2.3, we explore the relation between the performance and the trajectory of the neural network control robot system. For purposes of conciseness, Appendix 5 is provided to give details of the calculations.
6.2.1 Separation of Solutions

In this section, the solution of the system equation is expressed in a power series in \( \lambda \). The learning rate of a stable neural network can be treated as a small parameter of the complete system with two time-scale dynamics, i.e., fast robot dynamics and slow neural network dynamics.

From equation (2.10), the complete system dynamics can be expressed as

\[
\dot{e}_s = Ae_s + B \xi(e_s, w_s) = f(e_s, w_s) \\
\dot{w}_s = \lambda \cdot g(e_s, w_s) 
\]

(6.1)

Through mathematical manipulations (See details in Section 4.2.1), the zero order dynamics and first order dynamics are obtained

\[
\dot{e}_0 = f(e_0, w_0) \\
\dot{w}_0 = 0 
\]

(6.2)

\[
\dot{e}_1 = \frac{\partial f}{\partial e_0} e_1 + \frac{\partial f}{\partial w_0} w_1 \\
\dot{w}_1 = g(e_0, w_0) 
\]

(6.3)

where \( f_s = f(e_s, w_s) \). Now we can find the bound on \( e_s, e_1 \) separately and obtain the bound on the error with

\[
\|e\| \leq \|e_0\| + \lambda \cdot \|e_1\| 
\]

(6.6)

This perturbation approach enables us to transform the complete system into two relatively simple dynamic systems.

6.2.2 Solution Bound

In this section, the norm bound on the zero and first order of solution will be given first, and
then the bound of the solution will be obtained by the inequality given by equation (6.6). Here the zero order dynamics is the robot control system with PD and neural network control as feedback. Note that there are no weight dynamics in the zero order dynamics, i.e., the feedback gain is fixed, but still nonlinear.

**Zero Order Solution Bound**

After the inertia and centrifugal matrices are bounded with assumptions as given in Section 4.2.2, the zero order dynamics can be expressed as follows, through mathematical manipulation

\[
\begin{align*}
  e_0 &= P_1 \eta + P_1 \nu \\
  u &= P_2 \eta + P_2 \nu
\end{align*}
\]

(6.7) (6.8)

where:

- \( \eta \) = system uncertainty caused by estimation of robot parameters,
- \( \nu \) = neural network control output,
- \( P_i \) = coefficients defined in Section 5.2.1

Taking the \( L_{\infty} \) norm of a transfer matrix on both sides of the above equations and through mathematical manipulations, we obtain the zero order error bounded

\[
\|e_0\|_{\infty} \leq \frac{\beta_1}{\Delta} (M_2 \partial \phi + b + \beta_2 \beta_3 \phi) + \beta_1 \phi
\]

(6.9)

where:

\[
b = \alpha \cdot \|\dot{q}_d\|_{\infty} + M_2 \cdot \delta \cdot \|x_d\|_{\infty} + M_2 \cdot \rho
\]

\( \phi \) = upper bound on the neural network output.
First Order Solution Bound

Through a similar process as before, the first order solution is bounded as

$$\|e_1\|_{T_n} \leq \|G_1\| \cdot \|G_w\| \cdot \|w\|_{T_n} = 0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega$$

(6.10)

Solution Bound

Finally the bound of the system solution is obtained by combining the results from zero order norm bound (6.9) and first order norm bound (6.10)

$$\|e\|_{T_n} \leq \left\{ \frac{\beta_1}{\Delta} (M_2 \delta \beta_1 \phi + b + \beta_2 \beta_3 \phi) + \beta \phi \right\} + \lambda \cdot (0.5 \cdot \chi \cdot (\alpha + 1) \cdot \phi^2 \cdot \omega)$$

(6.11)

6.2.3 Influence of Trajectory Variation on the System Error Bound

In the following discussion, the influence of the trajectory variation on the system error bound is investigated. The system dynamics we consider is the static state dynamics after the neural network is trained, i.e., the weights are fixed after the training process. Hence, there are no weight dynamics, and consequently the system can be represented by the same dynamical equation that represents the zero order dynamics (6.2). First, the system error is expressed in Lagrange form and the system error norm is bounded by triangle inequality (6.12). Then the partial derivative of system error with respect to the trajectory variable \( q \) is defined as the sensitivity function and the associated sensitivity equation (6.13) is formulated. Furthermore, the solution bound approach is employed again to bound the sensitivity function (6.14). Finally the system error norm bound (6.15) will be obtained from the triangle inequality.

In the process of calculating \( \left\| \frac{\partial H^{-1}}{\partial q} \right\|_{T_n} \), the sensitivity of system error with respect to the
trajectory variable $\frac{\partial H^{-1}}{\partial q}$ must be obtained first. Symbol $\vec{q}$ is the trajectory variable vector at which the neural network is trained, which will be defined later. This sensitivity is a partial derivative of a matrix with respect to a vector, thus leading to a tensor. Also the sensitivity of system error with respect to the trajectory variation $\frac{\partial e}{\partial q}$ is a matrix. Since the norms of either a matrix or a vector are scalars, our results are scalars.

The system error considering the influence of the trajectory variation can be expressed by Lagrange theorem as

$$e(\vec{q} + \Delta \vec{q}) = e(\vec{q}) + \frac{\partial e}{\partial q} \bigg|_{(\vec{q} + \xi \Delta \vec{q})} \cdot \Delta \vec{q}$$

(6.12)

where:

$\xi$ = positive constant, $0 < \xi < 1$

Taking $L_{\infty}$ on both side of the above equation, the system error norm is depicted as

$$\|e(\vec{q} + \Delta \vec{q})\|_{L_{\infty}} \leq \|e(\vec{q})\|_{L_{\infty}} + \left\| \frac{\partial e}{\partial q} \right\|_{L_{\infty}} \cdot \|\Delta \vec{q}\|_{L_{\infty}}$$

(6.13)

where:

$\vec{q}$ = trajectory variable vector at which the neural network is trained, $\vec{q} = [q_1, q_2, q_3]^T$

$\Delta \vec{q}$ = trajectory variation vector, $\Delta \vec{q} = [\Delta q_1, \Delta q_2, \Delta q_3]^T$

$q_i$ = joint rotate angle of link $i$,

$\Delta q_i$ = variation of the joint rotate angle of link $i$,
Chapter 6 \hspace{1cm} The Influence of Trajectory Variation on System Performance

\[ \| e(q) \|_{T_{\infty}} \approx \| e_0 \|_{T_{\infty}} \]

Now the task is to calculate the $L_{T_{\infty}}$ norm of the sensitivity of system error $e$ with respect to $q$. The sensitivity analysis and the solution bound method are employed to accomplish this.

Define $\sigma = \frac{\partial e}{\partial q}$, then take partial derivative with respect to $q$ on both sides of the system error equation (2.10), the sensitivity equation is obtained

\[ \dot{\sigma} = A \sigma + B \frac{\partial \xi}{\partial q} \] (6.14)

where:

\[ \frac{\partial \xi}{\partial q} = (NK + H^{-1} \dot{H} \phi \phi' w^0 w^0) \sigma + \frac{\partial H^{-1}}{\partial q} (\dot{H} (q^d + Ke + v) + \Delta C) \]

$\phi'$ = derivative of the activation function,

$\frac{\partial H^{-1}}{\partial q}$ will be calculated later through robot dynamics (See details in Appendix-5).

In this way, the system error norm can be expressed as

\[ \left\| e(q + \Delta q) \right\|_{T_{\infty}} \leq \left\| e(q) \right\|_{T_{\infty}} + \left\| \sigma \right\|_{T_{\infty}} \cdot \left\| \Delta q \right\|_{T_{\infty}} \] (6.15)

Through mathematical manipulation, the sensitivity function is bounded as

\[ \left\| \sigma \right\|_{T_{\infty}} \leq \frac{\beta_{1\sigma} (\alpha_k \beta_{2\sigma} \phi + \| \eta_{\sigma 0} \|_{T_{\infty}} + \beta_{3\sigma} \beta_{4\sigma})}{1 - \alpha_k \beta_{1\sigma}} + \beta_{5\sigma} \phi \] (6.16)

Substituting (6.16) into (6.15), we obtain the bound on the system error norm considering the trajectory variation.
\[ \|e\|_{\|\Delta q\|} \leq \left\{ \frac{\beta_{1\sigma}(\alpha \kappa_\tau \beta_{2\sigma} \phi + 2 d_2 l_2 c_2 + 2 d_2 l_2 s_2 - d_2^2 m_1 l_2 s_2 c_2 + 2 J_{333} c_2 + \beta_{3\sigma} \beta_{4\sigma}) + \beta_{2\alpha} \phi}{1 - \alpha \kappa_\tau \beta_{1\sigma}} \right\} \cdot |\Delta q| + \left\{ \frac{\beta_\Delta}{\Delta} (M_2 \beta_\phi + b + \beta_2 \beta_\phi) + \beta_\phi \right\} \]  
(6.17)

where:

\[ |\Delta q| = \text{maximum of trajectory variation}, \quad \max_i |\Delta q| = \|\Delta q\|_{\|\Delta q\|_{\|\Delta q\|}} \]

\[ c_i = \cos(q_i) \]

\[ s_{ij} = \sin(q_i + q_j) \]

\[ l_{ic} = \text{signed distance from the origin of link } i \text{ along the } a_i \text{ axis to the center of mass of the link } i, \]

\[ J_{jk} = \text{pseudo-inertia element in row } j \text{ and column } k \text{ for link } i. \]

\[ d_i = \text{distance between the axis of joint } i \text{ and joint } i+1. \]

From the above analysis, we conclude that the variation of trajectory increases the norm bound on the system error, regardless of the sign of the trajectory variation. This is coincident with the neural network theory, i.e., the neural network performs at its best at the training point, any deviation from the training point will lead to an increase of neural network output error, which in turn increases the overall system error.

**Remark:**

In the calculation process of the sensitivity of system error with respect to the trajectory
variation, \( \left\| \frac{\partial H^{-1}}{\partial q} \right\| \) is needed to be calculated first. As we know, the form of the inertia matrix \( H \) is a function of \( \sin(q) \) or \( \cos(q) \) with the coefficients of these terms being link mass of link etc. Our previous partial derivative of inertia matrix is with respect to the coefficients (as in Chapter 5), while here the partial derivative of inertia matrix is with respect to the trajectory variable \( q \) itself. Still the maximum of the PD gains should satisfy the following inequality in the above theoretical analysis process

\[
\frac{M \delta}{1 - \alpha} < k_m < \frac{0.9375(\alpha + 1)\phi^2}{\alpha}
\]

(6.18)

where \( k_m \) is the maximum of PD gains, all the other symbols are the same as previously defined.

### 6.3 Summary

In this chapter, the influence of the robot trajectory variation on the neural network control robot system is investigated. The rigorous relation between the trajectory variation and the system performance, measured by the system error norm, is established. The developments are significant, since it permits us to predict the scale of performance degradation caused by the variation of the trajectory, which may be altered due to different execution tasks.
Figure 6.1. Trajectory variation in three-dimensional space
Chapter 7

Conclusions and Discussions

This thesis is concerned with the developing a systematic methodology to analyze the performance of neural network controllers as applied to robot systems.

7.1 Conclusions

The rigorous mathematical relation between the performance and parameters of the neural network control system is investigated here.

- The system dynamic model is formulated when the backpropagation neural network, used in conjunction with the PD and computed torque control, is employed as the controller of the robot trajectory following system, as shown in Figure 1.1.

- The highly nonlinear and complex dynamics of neural network control robot system are separated into two relatively simple dynamics based on perturbation theory. This is possible since the time constant of neural network dynamics is much slower than that of the robot dynamics for any stable neural network.

- The solution of the separated system dynamics are bounded by adopting the solution bound approach, which gives insight into the system performance, resulting from the
simplicity of the separated system dynamics.

- The influence of the parameters on the system performance is obtained through the sensitivity analysis.

- The system performance is more sensitive with respect to the link mass variation than to that of the payload when the $L_{in}$ norm is employed as measurement.

The system performance is measured with the system error norm. The parameters considered are the robot payload and link mass. Since payload is not part of the robot body, the robot dynamics is employed to obtain the inertia matrix considering the influence of payload. This is necessary when calculating the influence of the payload on the system performance.

The influence of the robot trajectory variation on the system performance is explored in Chapter 6. The partial derivative of system error norm, which characterizes the system performance, is with respect to the trajectory variable itself, not the coefficients of the trajectory variable, for instance, the mass of link in chapter 5. Hence the trajectory variation is a generalization case.

The proposed methodology permits us to predict the scale of the system performance degradation caused by the variation of the system parameters, which may be altered due to different execution tasks or update of equipment. This provides a better understanding of how the system works when the weights of neural networks are fixed, a state in which the robot system often operates.

The experiments are conducted on industrial standard equipments to verify the proposed methodology. The neural network controller is used conjunction with widely adopted PID control, thus leading to easy implementation of neural network in industrial robot systems.
Above all, the quantification of the performance issue, which predicts the rigorous range of system error, provides further justification for the commercialization of neural networks to robot systems.

7.2 Suggestions for Further Research

Based on the systematic approach we proposed here, several future research directions are natural considerations.

- Performance issue of similar parameters of the neural network control robot system, such as the centrifugal and Coriolis terms, and comparison of the system sensitivity with respect to variations of different parameters, as was carried out with link mass and payload.

- Quantification of the performance of control systems embedded with artificial intelligence technologies, for instance, the fuzzy control robot system, etc.

- Performance analysis of other systems with complex or nonlinear dynamics, as well as use of neural networks as controller, such as internal combustion engine systems.

- Performance analysis of the system with two time scale dynamics, such as the learning control system.

- Quantified performance comparison of the neural network controller with other control methods, such as self-adaptive technology, etc.

- Formulate an approach for synthesis of neural network control systems with guaranteed performance.
Bibliography


Appendix-1

Bounds of Weights

In this Appendix, the weights between the hidden and the output layers are to be shown bounded first. Then the weights between the input and hidden layers are shown bounded.

**Bounds of weights from hidden to output layer**

Taking truncated $L_{\infty}$ norm on both sides of the weight dynamics of the hidden to the output layers, as given in equation 2.6.1 (Section 2.4), we obtain

$$\| \hat{w} \|_{T_{\infty}} \leq \lambda \cdot \| v^d - v \|_{T_{\infty}} \cdot \| \varphi'(v) \|_{T_{\infty}} \cdot \| b_j \|_{T_{\infty}}$$

(1)

where:

$$\| v^d - v \|_{T_{\infty}} \leq \| v^d \|_{T_{\infty}} + \| v \|_{T_{\infty}} = \phi^d + \phi = \gamma,$$

$$\| \varphi'(v) \|_{T_{\infty}} \leq 0.5\phi \quad \text{and} \quad \| b_j \|_{T_{\infty}} \leq \phi$$

$\phi^d$ = ideal neural network output which is determined by

$$\| v^d \|_{T_{\infty}} \leq (\alpha + 1) \cdot (\alpha \| \hat{\alpha}^d \|_{T_{\infty}} + M_1(\delta \| X \|_{T_{\infty}} + \rho) = \phi^d$$

(2)

Hence $\| \hat{w} \|_{T_{\infty}} \leq 0.5 \cdot \lambda \cdot \gamma \cdot \phi$, and consequently, the weights between hidden and output of neural network $w^{ho}$ are bounded [Chen, 1996].
Bounds of weights from input to hidden layer

Taking truncated $L_{\infty}$ norm on both sides of weight dynamics of input to hidden (2.6.2), we obtain

$$\|\psi^h\|_{\tau_0} \leq \lambda \cdot \|v^d - v\|_{\tau_0} \cdot \|\varphi'(v)\|_{\tau_0} \cdot \|w^{ho}\|_{\tau_0} \cdot \|\varphi'(b)\|_{\tau_0} \cdot (\|e_0\|_{\tau_0} + \|q^d\|_{\tau_0})$$

(3)

where:

- $\|w^{ho}\|_{\tau_0}$ is bound of weights from hidden to output of neural network
- $\|e_0\|_{\tau_0}$ is bounded from equation (4.21) in Chapter 4,
- $\|q^d\|_{\tau_0}$ is also bounded, for it is a designed parameter,
- $\|\varphi'(b)\|_{\tau_0}$ is bounded at 0.5.

All the other terms are bounded the same as the above step.

Hence $\|\psi^h\|_{\tau_0}$ is also bounded and consequently $\|w^{ho}\|_{\tau_0}$ [Chen, 1996]. Now these two bounds on weights are denoted as two constants $\omega_1, \omega_2$. 

90
Appendix-2

Truncated $L_\infty$ Norm of the Sensitivity of System Error $e$ with respect to Payload $m$

In this Appendix, the truncated $L_\infty$ norm of the sensitivity of system error $e$ with respect to payload $m$ is obtained by employing the sensitivity approach and the solution bound method. First, the sensitivity equation of system error $e$ with respect to $m$ is constructed with the sensitivity definition. Then the truncated $L_\infty$ norm of the sensitivity equation is acquired through the solution bound approach.

The complete system error norm can be expressed as

$$\|e(m^* + \Delta m)\| = \|e(m^*) + \frac{\partial e(m)}{\partial m}(m^* \cdot \Delta m)\| \leq \|e(m^*)\| + \|\frac{\partial e}{\partial m}\| \cdot \|\Delta m\|$$

(1)

where:

$$\|e(m^*)\| = \|e_0\| \leq \frac{\beta_1}{\Delta} (M_2 \delta \beta_1 \phi + b + \beta_2 \beta_3 \phi) + \beta_1 \phi$$

$$\|\Delta m\| = |\Delta m|$$
Now the task is to calculate the partial differentiation \( \| \frac{\partial e}{\partial m} \| \), which is actually the norm of the sensitivity of system error \( e \) with respect to \( m \). We propose to use the sensitivity equation and the solution bound method to obtain the bound of the sensitivity equation.

**Sensitivity Equation**

Taking partial derivative of the system error equation with respect to the payload \( m \), we obtain

\[
\frac{\partial \dot{e}}{\partial m} = A \frac{\partial e}{\partial m} + B \frac{\partial \xi}{\partial m} \tag{2}
\]

Define \( \sigma = \frac{\partial e}{\partial m} \), then

\[
\frac{\partial \dot{e}}{\partial m} = \frac{\partial}{\partial m} (de/dt) = \frac{d}{dt} (\partial e/\partial m) = \dot{\sigma}
\]

Then the sensitivity equation is given as

\[
\dot{\sigma} = A\sigma + B \frac{\partial \xi}{\partial m} \tag{3}
\]

Since \( \zeta = E(\dot{q}^d + Ke) + H^{-1}\Delta C + H^{-1}\dot{H}v \)

Then the partial differential of non-linear term \( \zeta \) with respect to payload \( m \) is

\[
\frac{\partial \xi}{\partial m} = EK \frac{\partial e}{\partial m} + \frac{\partial E}{\partial m} (\dot{q}^d + Ke) + \frac{\partial H^{-1}}{\partial m} \Delta C + \frac{\partial H^{-1}}{\partial m} H^{-1} \dot{H}v + H^{-1} \dot{H} \frac{\partial v}{\partial m} \tag{4}
\]

where:

\[
E = H^{-1} \dot{H} - I \quad \text{and} \quad \|E\| \leq \alpha, \quad \text{then} \quad \frac{\partial E}{\partial m} = \frac{\partial H^{-1}}{\partial m} \dot{H}
\]

\[
v = \phi \cdot \varphi (w^{ha} \cdot \varphi (w^{ih} \cdot e)) \quad \text{then} \quad \frac{\partial v}{\partial m} = \phi \cdot \varphi (w^{ha} \varphi (w^{ih} \cdot e)) \frac{\partial e}{\partial m} = \phi \cdot \varphi (w^{ha} \varphi (w^{ih} \cdot e)) \frac{\partial e}{\partial m}
\]

\[
\frac{\partial H^{-1}}{\partial m} \quad \text{will be calculated by payload dynamics later, and note} \quad K, \dot{q}^d, \Delta C, \dot{H} \text{ are not functions of} \quad m.
\]
Now we have

$$\frac{\partial \xi}{\partial m} = (EK + H^{-1} \hat{H} \phi \phi' w^{ho} w^{ih}) \sigma + \frac{\partial H^{-1}}{\partial m} (\hat{H} (\dot{q}^d + Ke + v) + \Delta C) \quad (5)$$

### Solution bound of sensitivity equation

The sensitivity equation can be transformed to be the following form

$$\sigma = G \varepsilon \quad (6.1)$$
$$\varepsilon = \eta_\sigma + u_\sigma \quad (6.2)$$
$$u_\sigma = K_\sigma \sigma + K_\sigma^\nu \quad (6.3)$$
$$\eta_\sigma = EK \sigma + \eta_\sigma^0 \quad (6.4)$$

where:

$$G = (sI - A)^{-1} B$$

$$K_\sigma = H^{-1} \hat{H} \phi \phi' w^{ho} w^{ih}$$

$$K_\sigma^\nu = \frac{\partial H^{-1}}{\partial m} \hat{H}$$

$$\eta_\sigma^0 = \frac{\partial H^{-1}}{\partial m} (\hat{H} (\dot{q}^d + Ke + v) + \Delta C)$$

Through mathematical manipulation, the following is obtained

$$\sigma = P_{1\sigma} \eta_\sigma + P_{2\sigma} \nu \quad (7.1)$$
$$u = P_{3\sigma} \eta_\sigma + P_{4\sigma} \nu \quad (7.2)$$

where:

$$\eta_\sigma = \text{uncertainty caused by estimation of robot parameters,}$$

$$\nu = \text{neural network work control output,}$$

$$P_{1\sigma} = (I - GK_\sigma)^{-1} G$$

$$P_{2\sigma} = (I - GK_\sigma)^{-1} GK_\sigma^\nu$$
$P_{3\sigma} = K_\sigma (I - G K_\sigma)^{-1} G$

$P_{4\sigma} = K_\sigma (I - G K_\sigma)^{-1} G + K_{\sigma}$

The infinity norm ($L_\infty$) of a transfer matrix and a function in time domain is defined as

$$\|f(t)\|_{L_\infty} = \begin{cases} \text{ess sup}_{t \in [0, T]} \|f(t)\|, & t < T \\ 0, & t > T \end{cases}$$

Thus we have

$$\|\sigma\| \leq \beta_{1\sigma} \|\eta_\sigma\| + \beta_{2\sigma} \|\nu\| \quad (8.1)$$

$$\|\mu_\sigma\| \leq \beta_{3\sigma} \|\eta_\sigma\| + \beta_{4\sigma} \|\nu\| \quad (8.2)$$

where:

$$\beta_{1\sigma} = \|P_{1\sigma}\|$$

$$\beta_{1\sigma} = \|P_{1\sigma}\| \approx \frac{1}{K_\sigma}$$

$$\beta_{2\sigma} = \|P_{2\sigma}\| \approx \frac{K_{\sigma}}{K_\sigma}$$

Substitute (8.1) to (6.4) and take $\|\|$ on both side, we get

$$\|\eta_\sigma\| \leq \|E K\| \beta_{1\sigma} \|\eta_\sigma\| + \|E K\| \beta_{2\sigma} \|\nu\| + \|\eta_{\sigma 0}\| \quad (9)$$

Put (9) and (8.1) together

$$\begin{bmatrix} \|\eta_\sigma\| \\ \|\mu_\sigma\| \end{bmatrix} \leq \begin{bmatrix} \|E K\| \beta_{1\sigma} & 0 \\ \beta_{3\sigma} & 0 \end{bmatrix} \begin{bmatrix} \|\eta_\sigma\| \\ \|\mu_\sigma\| \end{bmatrix} + \begin{bmatrix} \|E K\| \beta_{2\sigma} \|\nu\| + \|\eta_{\sigma 0}\| \\ \beta_{4\sigma} \|\nu\| + \|\eta_{\sigma 0}\| \end{bmatrix} \quad (10)$$

Denote $Q = \begin{bmatrix} \|E K\| \beta_{1\sigma} & 0 \\ \beta_{3\sigma} & 0 \end{bmatrix}$, then $(I - Q) = \begin{bmatrix} 1 - \|E K\| \beta_{1\sigma} & 0 \\ -\beta_{3\sigma} & 1 \end{bmatrix}$

$$\Delta = \det(I - Q) = (1 - \|E K\| \beta_{1\sigma})$$

94
Since \( K = \begin{bmatrix} k_p & 0 \\ 0 & k_v \end{bmatrix} \) and \( \| K \|_\infty = \max \{ k_p, K_v \} = k_m \),

\[
\| K \| = H^{-1} H \| \phi \| \phi' \| \| \phi' \| \| \phi' \| \| \phi' \| \| \phi' \| = (\alpha + 1) \phi \* 0.5 * \phi \* 0.5 \* 1.5 \* 2.5 = 0.9375(\alpha + 1)\phi^2
\]

\[
\| E \| = \| E \| \| K \| \text{ and } \| E \| = \alpha, \text{ then } \Delta = 1 - \| E \| \beta_{1\sigma} \approx 1 - \frac{\alpha \| K \|}{\| K \|} = 1 - \frac{\alpha k_m}{0.9375(\alpha + 1)\phi^2}.
\]

So if

\[
\alpha k_m < 0.9375(\alpha + 1)\phi^2 \tag{11}
\]

then

\[
\Delta = \det(I - Q) \geq 0 \tag{12}
\]

which is the sufficient condition that the uncertainty \( \eta_\sigma \) is bounded [Spong and Vidyasagar, 1987]. In order to satisfy the above condition (11), the maximum PD gain must satisfy the condition

\[
\frac{k_m}{\phi^2} < \frac{0.9375(\alpha + 1)}{\alpha} \tag{13}
\]

which is true in our case, since \( \| K \|_\infty = \max \{ k_p, K_v \} = 6250 \) and \( \phi = 40 \).

Now we can say \( \Delta = \det(I - Q) = (1 - \| E \| \beta_{1\sigma}) \geq 0 \), then

\[
(I - Q)^{-1} = \frac{1}{\Delta} \begin{bmatrix} 1 & \beta_{3\sigma} \\ 0 & 1 - \alpha k_m \beta_{1\sigma} \end{bmatrix} \tag{14}
\]

\[
\begin{bmatrix} \| \eta_\sigma \| \\ \| \mu_\sigma \| \\ \| \eta_\sigma \| \end{bmatrix} \leq (I - Q)^{-1} \begin{bmatrix} \| E \| \beta_{1\sigma} \| \eta_\sigma \| + \| \eta_\sigma \| \\ \| E \| \beta_{1\sigma} \| \eta_\sigma \| + \| \eta_\sigma \| \end{bmatrix} = \frac{1}{1 - \alpha k_m \beta_{1\sigma}} \begin{bmatrix} 1 & \beta_{3\sigma} \\ 0 & 1 - \alpha k_m \beta_{1\sigma} \end{bmatrix} \begin{bmatrix} \| E \| \beta_{1\sigma} \| \eta_\sigma \| + \| \eta_\sigma \| \\ \| E \| \beta_{1\sigma} \| \eta_\sigma \| + \| \eta_\sigma \| \end{bmatrix} \tag{15}
\]

\[
\| \eta_\sigma \| \leq \frac{\alpha k_m \beta_{1\sigma} \phi + \| \eta_\sigma \| + \beta_{3\sigma} \beta_{4\sigma}}{1 - \alpha k_m \beta_{1\sigma}} \tag{16}
\]

where:
Here $\|e\|$ can be obtained from equation (4.21), which is robot system error norm; other terms have been given before, except $\left\|\frac{\partial H^{-1}}{\partial m}\right\|$. From payload dynamics [See details in Appendix-3], we have

$$\left\|\frac{\partial H^{-1}}{\partial m}\right\| = \frac{R^2[I_2^2 + (I_2 + 0.25m^*R^2)^2 + I_3 + 0.5m^*R^2]}{2[(I_2 + 0.25m^*R^2)(I_2 + 0.25m^*R^2) - I_2^2]}$$

(18)

Hence substitute $\left\|\frac{\partial H^{-1}}{\partial m}\right\|$ into (17), the bound on uncertainty $\|\eta_\sigma\|$ is obtained; then into (8.1), the bound on sensitivity function $\|\sigma\|$ is obtained.

$$\|\sigma\| \leq \frac{\beta_{1\sigma}(\alpha k_m\beta_{2\sigma}\phi + \|\eta_{0\sigma}\| + \beta_{3\sigma}\beta_{4\sigma}) + \beta_{2\sigma}\phi}{1 - \alpha k_m\beta_{1\sigma}}$$

(19)
Appendix-3

Norm of Partial Derivative of the Inverse of Inertia Matrix with respect to Payload

In this Appendix, the norm of the partial derivative of the inverse of inertia matrix with respect to the payload is determined. The analysis is carried out by the following three steps. First, the relation between the payload and the inertia matrix is established by payload dynamics. Second, we determine the partial derivative of inverse of inertia matrix with respect to the payload. Finally, the norm of the partial derivative is calculated.

Payload Dynamics

First, the influence of an inertia payload on the inertia matrix of robot is derived by robot and payload dynamics. Consider a solid object with six degrees of freedom as a payload to be grasped by a rigid body robot arm, shown in Figure 1. The end-effector frame is taken as the basic reference frame for analysis, i.e., the force and velocity of the robot end-effector and payload object are transformed with respect to the end-effector frame.

The robot dynamics with the payload is

\[ H(q)\ddot{q} + C(q, \dot{q}) = \tau - J^T f_e \]  

(1)
where:

- $q = \text{vector of generalized coordinates, i.e., joint angles, } q \in \mathbb{R}^{n \times 1}$

- $\dot{q} = \text{vector of joint velocities, } \dot{q} \in \mathbb{R}^{n \times 1}$

- $\ddot{q} = \text{vector of joint accelerations, } \ddot{q} \in \mathbb{R}^{n \times 1}$

- $H(q) = \text{inertia matrix, } H(q) \in \mathbb{R}^{n \times n}$

- $C(q, \dot{q}) = \text{Centrifugal, Coriolis and gravitational term, } C(q, \dot{q}) \in \mathbb{R}^{n \times l}$

- $\tau = \text{vector of joint torque, } \tau \in \mathbb{R}^{6 \times 1}$

- $J^T = \text{transpose of the Jacobian matrix of robot at the origin with respect to the end-effector frame, } J^T \in \mathbb{R}^{6 \times 6}$

- $f_e = \text{generalized force applied to the robot by the payload with respect to the end-effector frame, } f_e \in \mathbb{R}^{6 \times 1}.$

Let's consider the payload dynamics expressed in the payload frame

$$f_p = \begin{bmatrix} n \\ g \\ I_p \ddot{\theta}_p \end{bmatrix} = M_p \dot{\psi}$$

where:

- $x_p = \text{position and orientation vectors of the payload object expressed in payload frame, } x_p \in \mathbb{R}^{3 \times 1}$

- $\theta_p = \text{position and orientation vectors of the payload object expressed in payload frame, } \theta_p \in \mathbb{R}^{3 \times 1}$

- $\psi = \text{generalized vector of payload, } \psi \in \mathbb{R}^{6 \times 1}$

- $n = \text{force vector that applied on the payload object, } n \in \mathbb{R}^{1 \times 1}$
\( g \) = moment vector that applied on the payload object, \( g \in \mathbb{R}^{3 \times 1} \)

\( f_p \) = generalized force vector applied to payload by the robot end-effector expressed in payload frame, \( f_p \in \mathbb{R}^{6 \times 1} \)

\( m_p \) = mass matrix of payload object, \( m_p \in \mathbb{R}^{3 \times 3} \)

\( I_p \) = inertia matrix of payload object, \( I_p \in \mathbb{R}^{3 \times 3} \)

\( M_p \) = generalized inertia matrix of payload, \( M_p \in \mathbb{R}^{6 \times 6} \)

\( \Theta \) = zero matrix.

\[ \psi = \begin{bmatrix} x_p \\ \theta_p \end{bmatrix} \]

\[ m_p = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \]

\[ I_p = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \]

\[ M_p = \begin{bmatrix} m_p & \Theta \\ \Theta & I_p \end{bmatrix} \]

The linking equations are the force and velocity equation of the end-effector expressed in \( F_e \)

\[ f_e = T_{ep} f_p \quad (2) \]

\[ J(q) \dot{q} = T_{ep} \dot{\psi} \quad (3) \]

where:

\( T_{ep} \) = transformation matrix that transform the generalized force from payload frame to end-effector frame, \( T_{ep} \in \mathbb{R}^{6 \times 6} \)
\[ \dot{q} = \text{velocity vector of end-effector frame, } \dot{q} \in R^{6 \times 1} \]

\[ \dot{\psi} = \text{velocity vector of payload frame, } \dot{\psi} \in R^{6 \times 1} \]

**Inertia Matrix Considering the Payload Influence**

Through mathematical manipulation, we have

\[
(H(q) + J^T T_{ep} M_p T_{ep}^T J) \cdot \dot{q} + C(q, \dot{q}) + J^T T_{ep} M_p T_{ep}^T (\dot{J}q - \ddot{T}_{ep} \dot{\psi}) = \tau
\]

(4)

From the above equation, the variation of the inertia matrix influenced by the payload is

\[
\Delta H = J^T T_{ep} M_p T_{ep}^T J
\]

(5)

where:

\[ \Delta H = \text{variation of inertia matrix, } \Delta H \in R^{6 \times 6} \]

Here two plausible assumptions are made to simplify the otherwise complex mathematical calculation, which will be encountered in the subsequent analysis. The reason why the mathematical calculations are complicated is that from the above equation we see that \( \Delta H \) is a 6×6 matrix with non-zero off-diagonal elements. Hence when the eigenvalue of the inverse of this matrix is calculated, the solution to a six-order equation of eigenvalue must be found. While there is no closed form solution to a six-order equation in the mathematical literature, some reasonable simplification becomes necessary.

**Assumption 1:** The payload frame is chosen to be parallel to that of end-effector frame, since the axis direction of payload frame is chosen arbitrarily. Under this assumption, there will be no orientation transformation from the payload frame to the end-effector frame. Therefore, the transformation matrix \( T_{ep} \) from payload frame to end-effector frame has a diagonal form
where the $R_x, R_y, R_z$ are the coordinates of payload object vector in $x,y,z$ directions and the inertia matrix of payload can be expressed directly in the end-effector frame (where $m$ is the mass of payload object).

$$I_p = \begin{bmatrix}
I_{xx} + mR_x^2 & I_{xy} & I_{xz} \\
I_{yx} & I_{yy} + mR_y^2 & I_{yz} \\
I_{zx} & I_{zy} & I_{zz} + mR_z^2
\end{bmatrix} \quad (7)$$

**Assumption 2:** The payload object is a solid round plate with mass $m$, radius $R$ and height $h$.

This assumption means that $I_{ij} = \begin{cases} I_m, & i = j \\ 0, & i \neq j \end{cases}$ thus the generalized inertia matrix of payload has the diagonal form

$$M_p = \begin{bmatrix}
m_p & \Theta \\
\Theta & I_{xx} & 0 & 0 \\
0 & 0 & I_{yy} & 0 \\
0 & 0 & 0 & I_{zz}
\end{bmatrix} \quad (8)$$

Since $T_{ep}^T T_{ep} = I, \quad J^T J = I$, then

$$\Delta H = J^T T_{ep} M_p T_{ep}^T J = M_p J^T T_{ep}^T T_{ep}^T J = M_p \quad (9)$$

The problem we are concerned with is a trajectory tracking problem, i.e., only the position of the payload object is to be regulated, hence the first three degrees of freedom of robot joints are considered in the following analysis. And consequently the form of variation of inertia matrix of robot is
The inertia matrix of the robot varies with the robot geometry structure and is bounded. In this section, our goal is to find this bound using the CRS A460 robot as our model. Since the position of end effector is determined only by the first three joints, we consider the inertia matrix of first three joints only. The inertia matrix of CRS A460 robot [Bains, 1995] is

\[
H = \begin{bmatrix}
I_{11} & 0 & 0 \\
0 & I_{22} & I_{23} \\
0 & I_{32} & I_{33}
\end{bmatrix}
\]

Then the equivalent inertia matrix considering the influence of payload is obtained

\[
H + \Delta H = \begin{bmatrix}
I_{11} + m & 0 & 0 \\
0 & I_{22} + m & I_{23} \\
0 & I_{32} & I_{33} + m
\end{bmatrix}
\]

**Inverse of the Inertia Matrix Considering the Payload Influence**

Through mathematical manipulation, the inverse of inertia matrix is

\[
(H + \Delta H)^{-1} = \frac{1}{\det(A)} A^* = \frac{1}{a_{11}(a_{22}a_{33} - a_{23}^2)} \begin{bmatrix}
a_{11}(a_{22}a_{33} - a_{23}^2) & 0 & 0 \\
0 & a_{11}a_{33} - a_{13}a_{23} \\
0 & -a_{13}a_{33} + a_{11}a_{23}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{1}{a_{11}} & 0 & 0 \\
0 & \frac{a_{11}a_{33}}{\det(A)} & -\frac{a_{11}a_{23}}{\det(A)} \\
0 & \frac{a_{13}a_{33}}{\det(A)} & \frac{a_{11}a_{23}}{\det(A)}
\end{bmatrix}
\]

\[
\frac{\partial (H + \Delta H)^{-1}}{\partial m} = \begin{bmatrix}
-R^2 & 0 & 0 \\
0 & -R^2(a_{22}^2 + a_{33}^2) & -R^2(a_{22} + a_{33}) \\
0 & 0 & 0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\frac{-R^2}{2a_{11}} & 0 & 0 \\
0 & \frac{-R^2(a_{22}^2 + a_{33}^2)}{2(a_{22}a_{33} - a_{23}^2)^2} & \frac{-R^2(a_{22} + a_{33})}{2(a_{22}a_{33} - a_{23}^2)^2} \\
0 & 0 & \frac{-R^2(a_{22}^2 + a_{33}^2)}{2(a_{22}a_{33} - a_{23}^2)^2} & \frac{-R^2(a_{22} + a_{33})}{2(a_{22}a_{33} - a_{23}^2)^2}
\end{bmatrix}
\]

where:
\[ a_{11} = I_{11} + I_{xx}^p = I_{11} + 0.5m^*R^2 \]
\[ a_{22} = I_{22} + I_{yy}^p = I_{22} + 0.25m^*R^2 \]
\[ a_{33} = I_{33} + I_{zz}^p = I_{33} + 0.25m^*R^2 \]
\[ a_{23} = a_{32} = I_{23} = I_{32} \]

**Norm of the Partial Derivative of the Inverse Inertial Matrix with respect to the Payload**

Substitute the values of parameters [Bains, 1995] into the above equation and calculate the infinity norm

\[
\frac{\partial H^{-1}}{\partial m} = \frac{R^2 (a_{23}^2 + a_{32}^2 + a_{22} + a_{33})}{2(a_{22}a_{33} - a_{23}^2)} = \frac{R^2 [I_{23}^2 + (I_{22} + 0.25m^*R^2)^2 + I_{22} + I_{33} + 0.5m^*R^2]}{2[(I_{22} + 0.25m^*R^2)(I_{22} + 0.25m^*R^2) - I_{23}^2]^2}
\]

(13)

where

\[ R = \text{radius of the round plate form payload}. \]
Figure-1 The rigid body payload dynamics where $F_b$: base frame, $F_e$: end-effector frame, $F_p$: payload frame.
Appendix-4

Norm of Partial Derivative of the Inverse of Inertia Matrix with respect to Link Mass

In this Appendix, the norm of partial derivative of inverse of inertia matrix with respect to link mass is determined. First, the partial derivative of inverse of inertia matrix with respect to the link mass is calculated. Then, the norm of the partial derivative is obtained.

Inverse of Inertial Matrix

Here $H(q)$ is a three dimension matrix, $H(q) \in R^{3\times3}$, $L_m$ is a vector $L_m \in R^{3\times1}$, hence the partial derivative will lead to an $9 \times 3$ Jacobian matrix. (Note this Jacobian matrix is the term in mathematical literature, i.e., matrix derivative [Rudin, 1991].)

The inertia matrix of CRSA460 has the form [Baines, 1995]

$$
H = \begin{bmatrix}
    I_{11} & 0 & 0 \\
    0 & I_{22} & I_{23} \\
    0 & I_{32} & I_{33}
\end{bmatrix}
$$

(1)

where:

$$
I_{11} = K_{i2} + c_2^2 \cdot K_{i3} - 2c_2s_{23} + c_{23}^2 \cdot K_2
$$

$$
I_{22} = K_{i4} - 2s_3 \cdot K_i
$$
\[ I_{23} = I_{32} = K_4 - s_3 \cdot K_1 \]

\[ I_{33} = K_2 \]

in which

\[ K_{12} = J_{111} + J_{113} + J_{m1} + J_{222} + J_{333} + J_{332} \]

\[ K_{13} = 2a_2m_2l_{2c} + a_3^2m_2 + a_2^2m_3 + J_{211} - J_{222} \]

\[ K_{14} = 2a_2m_2l_{2c} + a_3^2m_2 + a_2^2m_3 + J_{221} + J_{222} + J_{m2} + J_{311} + J_{333} \]

where:

\[ m_i = \text{mass of } i\text{th link}, \]

\[ l_{ic} = \text{signed distance from the orign of link } i \text{ along the } a_i \text{ axis to the center of mass of the link } i, \]

Inverse of the inertia matrix is obtained through mathematical manipulation

\[
H^{-1} = \frac{1}{\det(A)} \begin{bmatrix}
    a_{11}(a_{22}a_{33} - a_{23}^2) & 0 & 0 \\
    0 & a_{11}a_{33} - a_{11}a_{23} & a_{11}a_{22} \\
    0 & -a_{11}a_{23} & a_{11}a_{22}
\end{bmatrix}
\]

\[
A^* = \begin{bmatrix}
    1 & 0 & 0 \\
    a_{11}a_{23} & \frac{a_{11}a_{33}}{\det(A)} & -a_{11}a_{23} \\
    0 & -a_{11}a_{23} & \frac{a_{11}a_{22}}{\det(A)}
\end{bmatrix}
\]

**Calculation of the partial derivative** \( \frac{\partial H^{-1}}{\partial L_m} \)

Here inertia matrix \( H^{-1}(q) \) is a three dimension matrix, \( H^{-1}(q) \in \mathbb{R}^{3 \times 3} \), \( L_m \) is a vector \( L_m \in \mathbb{R}^{3 \times 1} \), hence the partial derivative will lead to an 9 X 3 Jacobin matrix. (Note this Jacobian matrix is the term in mathematical literature, i.e., matrix derivative [Rudin, 1991].)

The link mass vector is defined as

\[ L_m = [m_1, m_2, m_3]^T \]

where:
\[ m_i = \text{mass of link } i, \]

Then the partial derivative has the form of following

\[
\frac{\partial H}{\partial L} = \begin{bmatrix}
\frac{\partial I_{11}}{\partial m_1} & \frac{\partial I_{11}}{\partial m_2} & \frac{\partial I_{11}}{\partial m_3} \\
\frac{\partial I_{21}}{\partial m_1} & \frac{\partial I_{21}}{\partial m_2} & \frac{\partial I_{21}}{\partial m_3} \\
\frac{\partial I_{31}}{\partial m_1} & \frac{\partial I_{31}}{\partial m_2} & \frac{\partial I_{31}}{\partial m_3} \\
\frac{\partial I_{12}}{\partial m_1} & \frac{\partial I_{12}}{\partial m_2} & \frac{\partial I_{12}}{\partial m_3} \\
\frac{\partial I_{22}}{\partial m_1} & \frac{\partial I_{22}}{\partial m_2} & \frac{\partial I_{22}}{\partial m_3} \\
\frac{\partial I_{32}}{\partial m_1} & \frac{\partial I_{32}}{\partial m_2} & \frac{\partial I_{32}}{\partial m_3} \\
\frac{\partial I_{13}}{\partial m_1} & \frac{\partial I_{13}}{\partial m_2} & \frac{\partial I_{13}}{\partial m_3} \\
\frac{\partial I_{23}}{\partial m_1} & \frac{\partial I_{23}}{\partial m_2} & \frac{\partial I_{23}}{\partial m_3} \\
\frac{\partial I_{33}}{\partial m_1} & \frac{\partial I_{33}}{\partial m_2} & \frac{\partial I_{33}}{\partial m_3} \\
\frac{\partial I_{13}}{\partial m_1} & \frac{\partial I_{13}}{\partial m_2} & \frac{\partial I_{13}}{\partial m_3} \\
\end{bmatrix}
\]

where:

\[
\frac{\partial I_{11}}{\partial m_1} = \frac{\partial K_{12}}{\partial m_1} + c_2^2 \cdot \frac{\partial K_{13}}{\partial m_1} + c_2^2 \cdot \frac{\partial K_2}{\partial m_1}
\]

\[
\frac{\partial I_{22}}{\partial m_1} = \frac{\partial K_{14}}{\partial m_1} - 2c_3 \cdot \frac{\partial K_1}{\partial m_1}
\]

\[
\frac{\partial I_{23}}{\partial m_1} = \frac{\partial K_4}{\partial m_1} - s_3 \cdot \frac{\partial K_1}{\partial m_1}
\]

\[
\frac{\partial I_{11}}{\partial m_2} = \frac{\partial K_{12}}{\partial m_2} + c_2^2 \cdot \frac{\partial K_{13}}{\partial m_2} + c_2^2 \cdot \frac{\partial K_2}{\partial m_2}
\]

\[
\frac{\partial I_{22}}{\partial m_2} = \frac{\partial K_{14}}{\partial m_2} - 2c_3 \cdot \frac{\partial K_1}{\partial m_2}
\]
\[ \frac{\partial I_{23}}{\partial m_2} = \frac{\partial K_4}{\partial m_2} - s_3 \cdot \frac{\partial K_1}{\partial m_2} \]

\[ \frac{\partial I_{11}}{\partial m_3} = \frac{\partial K_{12}}{\partial m_3} + c_2^2 \cdot \frac{\partial K_{13}}{\partial m_3} + c_3^2 \cdot \frac{\partial K_2}{\partial m_3} \]

\[ \frac{\partial I_{22}}{\partial m_3} = \frac{\partial K_{14}}{\partial m_3} - 2c_3 \cdot \frac{\partial K_1}{\partial m_3} \]

\[ \frac{\partial I_{33}}{\partial m_2} = \frac{\partial K_4}{\partial m_2} - s_3 \cdot \frac{\partial K_1}{\partial m_2} \]

Since for all other \( i, j \), \( I_{ij} = 0 \), then \( \frac{\partial I_{ij}}{\partial m_i} = 0 \).

In turn

\[ \frac{\partial K_{12}}{\partial m_1} = -\frac{\partial J_{m1}}{\partial m_1} \]

\[ \frac{\partial K_{13}}{\partial m_2} = 2a_2 l_{2e} + a_2^2 \]

\[ \frac{\partial K_{14}}{\partial m_3} = a_2^2 m_2 + a_2^2 + \frac{\partial J_{m1}}{\partial m_1} \]

\[ s_i = \sin(q_i) \]

\[ c_i = \cos(q_i) \]

\[ s_{ij} = \sin(q_i + q_j) \]

\[ c_{ij} = \cos(q_i + q_j) \]

So, substituting the above results into the partial derivative, we obtain the partial derivative of the inverse of inertia matrix with respect to the link mass.
**Norm of the Partial Derivative**

\[ \left\| \frac{\partial H^{-1}}{\partial L_m} \right\| \]

Now through the mathematical manipulation, the \( L_m \) norm of the Jacobian Matrix, which is the partial derivative \( \frac{\partial H^{-1}}{\partial L_m} \), is obtained

\[ \left\| \frac{\partial H^{-1}}{\partial L_m} \right\|_m = 4a_1^2 l_{2c} + 2a_1^2 l_{2e}^2 - 2J_{113} + J_{11}^2 - a_2^2 m_1 l_{2c}^2 + 2J_{333} \]  \( \text{(3)} \)

The basic methodology used for link mass analysis is the same as that of the payload mass, while since the variable here is a vector \( L_m \) instead of scalar \( m \), the mathematical process turns out to be more complicated in the calculation of the sensitivity function. But as previously mentioned, the norm of either a matrix or a vector must be a scalar, so the final result is still a scalar form, which again illustrates the advantage of using the error norm as the system performance measure.
Appendix-5

Norm of Partial Derivative of the Inverse of Inertia Matrix With Respect to Trajectory

In this Appendix, the norm of partial derivative of inverse of inertia matrix with respect to trajectory is determined. First, the partial derivative of inverse of inertia matrix with respect to the trajectory is calculated. Then, the norm of the partial derivative is obtained.

Inverse of Inertial Matrix

The inertia matrix of CRSA460 has the form [Baines, 1995]

\[
H = \begin{bmatrix}
I_{11} & 0 & 0 \\
0 & I_{22} & I_{23} \\
0 & I_{32} & I_{33}
\end{bmatrix}
\] (1)

where:

\[
I_{11} = K_{12} + c_2^2 \cdot K_{13} - 2c_2s_{23} + c_{23}^2 \cdot K_2
\]

\[
I_{22} = K_{14} - 2s_3 \cdot K_1
\]

\[
I_{23} = I_{32} = K_4 - s_3 \cdot K_1
\]
\[ I_{33} = K_2 \]

in which

\[ K_{12} = J_{111} + J_{113} + J_{m1} + J_{222} + J_{332} + J_{333} \]

\[ K_{13} = 2a_2m_2l_{2c} + a_2^2m_2 + a_2^2m_3 + J_{211} - J_{222} \]

\[ K_{14} = 2a_2m_2l_{2c} + a_2^2m_2 + a_2^2m_3 + J_{221} + J_{222} + J_{m2} + J_{311} + J_{333} \]

where:

\[ m_i = \text{mass of } i\text{th link}, \]

\[ l_{ic} = \text{signed distance from the origin of link } i \text{ along the } a_i \text{ axis to the center of mass of the link } i, \]

Inverse of the inertia matrix is obtained through mathematical manipulation

\[ H^{-1} = \frac{1}{\det(A) a_{11}^2} \begin{bmatrix} a_{11}(a_{22}a_{33} - a_{23}^2) & 0 & 0 \\ 0 & a_{11}a_{23} & -a_{11}a_{23} \\ 0 & -a_{11}a_{23} & a_{11}a_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{a_{11}a_{33}}{\det(A)} & -\frac{a_{11}a_{23}}{\det(A)} \\ 0 & -\frac{a_{11}a_{23}}{\det(A)} & \frac{a_{11}a_{22}}{\det(A)} \end{bmatrix} \quad (2) \]

**Calculation of the partial derivative** \( \frac{\partial H^{-1}}{\partial L_m} \)

Here inertia matrix \( H^{-1}(q) \) is a three dimension matrix, \( H^{-1}(q) \in \mathbb{R}^{3 \times 3} \), \( \vec{q} \) is a vector \( \vec{q} \in \mathbb{R}^{3 \times 1} \), hence the partial derivative will lead to an 9 X 3 Jacobin matrix. The trajectory vector is defined as

\[ \vec{q} = [q_1, q_2, q_3]^T \quad (3) \]

where:

\[ q_i = \text{rotation angle of } i\text{th joint} \]
Then, the partial derivative of $H^{-1}(q)$ with respect to $\mathbf{q}$ has the form of following

\[
\frac{\partial H^{-1}}{\partial \mathbf{q}} = 
\begin{bmatrix}
\frac{\partial I_{11}}{\partial q_1} & \frac{\partial I_{11}}{\partial q_2} & \frac{\partial I_{11}}{\partial q_3} \\
\frac{\partial I_{21}}{\partial q_1} & \frac{\partial I_{21}}{\partial q_2} & \frac{\partial I_{21}}{\partial q_3} \\
\frac{\partial I_{31}}{\partial q_1} & \frac{\partial I_{31}}{\partial q_2} & \frac{\partial I_{31}}{\partial q_3} \\
\frac{\partial I_{12}}{\partial q_1} & \frac{\partial I_{12}}{\partial q_2} & \frac{\partial I_{12}}{\partial q_3} \\
\frac{\partial I_{22}}{\partial q_1} & \frac{\partial I_{22}}{\partial q_2} & \frac{\partial I_{22}}{\partial q_3} \\
\frac{\partial I_{32}}{\partial q_1} & \frac{\partial I_{32}}{\partial q_2} & \frac{\partial I_{32}}{\partial q_3} \\
\frac{\partial I_{13}}{\partial q_1} & \frac{\partial I_{13}}{\partial q_2} & \frac{\partial I_{13}}{\partial q_3} \\
\frac{\partial I_{23}}{\partial q_1} & \frac{\partial I_{23}}{\partial q_2} & \frac{\partial I_{23}}{\partial q_3} \\
\frac{\partial I_{33}}{\partial q_1} & \frac{\partial I_{33}}{\partial q_2} & \frac{\partial I_{33}}{\partial q_3}
\end{bmatrix}
\]

where:

\[
\frac{I_{11}}{q_1} = \frac{\partial K_{12}}{\partial q_1} + c_2^2 \frac{\partial K_{13}}{\partial q_1} + c_2^2 \frac{\partial K_2}{\partial q_1}
\]

\[
\frac{\partial I_{22}}{\partial q_1} = \frac{\partial K_{14}}{\partial q_1} - 2c_3 \frac{\partial K_1}{\partial q_1}
\]

\[
\frac{\partial I_{23}}{\partial q_1} = \frac{\partial K_{4}}{\partial q_1} - s_3 \frac{\partial K_1}{\partial q_1}
\]

\[
\frac{\partial I_{11}}{q_2} = \frac{\partial K_{12}}{\partial q_2} + c_2^2 \frac{\partial K_{13}}{\partial q_2} + c_2^2 \frac{\partial K_2}{\partial q_2}
\]

\[
\frac{\partial I_{22}}{\partial q_2} = \frac{\partial K_{14}}{\partial q_2} - 2c_3 \frac{\partial K_1}{\partial q_2}
\]

\[
\frac{\partial I_{23}}{\partial q_2} = \frac{\partial K_{4}}{\partial q_2} - s_3 \frac{\partial K_1}{\partial q_2}
\]

\[
\frac{\partial I_{11}}{q_3} = \frac{\partial K_{12}}{\partial q_3} + c_2^2 \frac{\partial K_{13}}{\partial q_3} + c_2^2 \frac{\partial K_2}{\partial q_3}
\]
\[
\frac{\partial I_{22}}{\partial q_3} = \frac{\partial K_{14}}{\partial q_3} - 2c_3 \cdot \frac{\partial K_1}{\partial q_3} \\
\frac{\partial I_{23}}{\partial q_2} = \frac{\partial K_{4}}{\partial q_2} - s_3 \cdot \frac{\partial K_1}{\partial q_2}
\]

Since for all other \(i,j\), \(I_{ij} = 0\), then \(\frac{\partial I_{ij}}{\partial q_i} = 0\).

In turn

\[
\frac{\partial K_{12}}{\partial q_1} = -\frac{\partial J_{ml}}{\partial q_1} \\
\frac{\partial K_{13}}{\partial q_2} = 2a_2l_2c + a_2^3c_3s_2 \\
\frac{\partial K_{14}}{\partial q_3} = a_2^2m_2 + a_2^3s_3 + \frac{\partial J_{ml}}{\partial q_3}
\]

\[
s_i = \sin(q_i) \\
c_i = \cos(q_i) \\
s_{ij} = \sin(q_i + q_j) \\
c_{ij} = \cos(q_i + q_j)
\]

So, substituting the above results into the partial derivative equation (4), we obtain the partial derivative of the inverse of inertia matrix with respect to the trajectory.

**Norm of the Partial Derivative**\[
\left\| \frac{\partial H^{-1}}{\partial L_m} \right\|_{\infty}
\]

Now through the mathematical manipulation, the \(L_{\infty}\) norm of the Jacobin matrix, which is the partial derivative \(\frac{\partial H^{-1}}{\partial q}\), is obtained.
The result of this partial derivative resemble that of link mass. While as we know, the form of inertia matrix $H$ is a function of $\sin(q)$ or $\cos(q)$ and the coefficients of these terms, such as link mass etc. Our previous partial derivative of inertia matrix is with respect to the coefficients (such as in Chapter 5), while here the partial derivative of inertia matrix is with respect to the trajectory variable $q$ itself.