AN INTELLIGENT FORCE CONTROL SCHEME FOR ROBOTIC APPLICATIONS: CONTACT WITH NON-RIGID ENVIRONMENTS

CHRISTIAN JAY KORDICH B.Eng.(Hons), A.M.I.E.E.

A thesis submitted in partial fulfilment of the
requirements of Liverpool John Moores University for
the degree of Doctor of Philosophy

This research programme was carried out
in collaboration with British Nuclear Fuels Limited .

September 1998

ABSTRACT

This thesis describes an investigation into the use of a novel intelligent force control scheme that was developed to control the contact force between a mechanical manipulator's end-effector and a range of non-rigid contact environments. The scheme uses a Radial Basis Function (RBF) neural network to model idealised reaction to a range of environments, each with differing degrees of rigidity.

During the development of the intelligent force control scheme's neural network, factors that may affect network performance were investigated, including aspects relating to network topology selection and RBF centre placement. Results presented show that a single RBF network was capable of modelling idealised reaction to a range of non-rigid environments to a high degree of accuracy.

Once trained, the RBF network was incorporated into a single degree of freedom mechanical manipulator simulation that was developed in the Advanced Continuous Simulation Language and the control system's ability to apply forces to a range of non-rigid environments was investigated by simulation. Results presented demonstrate that satisfactory contact was achievable with a range of non-rigid environments without *a priori* knowledge of the contact environment's mechanical properties. The intelligent force control scheme's suitability for force application to varying environments was also investigated.

ACKNOWLEDGEMENTS

I would like to acknowledge the financial assistance provided by Liverpool John Moores University.

In addition, I would like to acknowledge the assistance and support of both the Process Control and Optimisation Group and the Robotics Research Group at British Nuclear Fuels, Sellafield. In particular, I wish to thank Bill Harper, Dr. David Woodhead, and Gulab Mistry.

Finally, I could not have embarked on a study of this magnitude without the love and support of my family and friends. Special thanks go to my mother and brother, Barbara and Darren, Suzie, and my daughter Sophie Katarina, who was born during the compilation of this thesis.

Christian Jay Kordich.

CONTENTS

CHAPTER 5

CONTACT WITH A SINGLE NON-RIGID ENVIRONMENT

CHAPTER 6

CONTACT WITH A RANGE OF NON-RIGID ENVIRONMENTS

LIST OF SYMBOLS AND ABBREVIATIONS

a	Spring hardening exponent
AFPE	Akaikes Final Prediction Error
ANN	Artificial Neural Network
b	Shock absorption exponent
B	Effective system friction referred to the motor shaft
BTestNet	Best performing network with test data set
BTrainNet	Best performing network with training data set
c_i	RBF centre vector for node i
D_{sat}	Depth at which positional saturation occurs
e	Error signal
EMF	Electromotive Force
F	Contact force
I/O	Input-output
J	Effective system inertia referred to the motor shaft
J_m	Motor inertia
J_L	Load inertia
k	Kilo
k_{dist}	K-means clustering minimum distance
k_1	Contact environment spring constant
k_2	Contact environment damping coefficient
k_3	Contact environment spring hardening coefficient

K_a	Current amplifier gain
K_p	Controller gain
K_v	Tacho-feedback gain
L	Motor armature inductance
LF	Loss Function
m	Rotation to translation gain factor
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
n	Reduction gear ratio
n_h	Number of hidden layer nodes
N	Number of vectors in test data set
N_w	Number of adjustable model parameters
PI	Proportional plus Integral
PID	Proportional plus Integral plus Derivative
R	Motor armature resistance
RBF	Radial Basis Function
RLS	Recursive Least Squares
s	Laplacian operator
SNR	Signal to Noise Ratio
t	Time
T_m	Motor torque acting at the motor shaft
V	Controller output
w_{ij}	Weight connecting the *i*th hidden node to the *j*th output node
x	Network input vector

x_0	Zero depth reference position
x_{env}	Depth into the contact environment
$\dot{x}_{env}$	End-effector velocity through the contact environment
X_1	Displacement of robot end-effector due to contact
X_2	Displacement of force sensor due to contact
X_{spt}	Positional setpoint
X_{out}	Measured position in x direction
$X_{ref}, Y_{ref}, Z_{ref}$	Robot reference frame axis labels
$X_{task}, Y_{task}, Z_{task}$	Task aligned reference frame axis labels
y	Target output vector
y_j	Network output for output node j
$\hat{y}$	Network estimation
z	Activation function input
α_c	K-means clustering learning rate
β	K-means clustering decay rate
β_j	Bias parameter for the *j*th output node
ρ	RBF centre width
$\phi(z)$	Activation function output
θ_L	Load shaft position
θ_m	Motor shaft position
θ_m	Motor shaft velocity
σ_n	Standard deviation of noise
σ_s	Standard deviation of signal (measured variable)

$\phi_i(*)$	Output of the *i*th hidden node
$\|\|*\|\|$	Euclidean norm of *

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Robots are most commonly used to increase productivity and improve product quality in manufacturing environments where they perform tasks such as paint spraying, spot welding, and component placement. When performing such tasks, a robot is used as a positioning device where it is commanded to either move its end-effector (and most frequently tooling attached to the end-effector) to desired points within the robot workspace or to the follow a desired trajectory. In order that these 'positioning' tasks can be performed, the robot programmer must supply the robot control system with task oriented positioning data (i.e. positions, velocity profiles, etc.) and the robot is controlled using a positional control scheme.

Many 'real world' tasks require that a robot system applies a desired force to an object or surface. Examples of these 'contact' tasks are object handling, cutting, grinding, and drilling. Although positional control schemes are appropriate for non-contact tasks, they are unsuitable for contact tasks and dedicated force control strategies are required if robots are to have a force control capability. Research in the area of robotic force control has been particularly active over the past two decades but despite the high level of research activity, there remains a need for fast acting, stable, and widely applicable force control methodologies.

Robot force control is a multifaceted problem that requires the creation of each of the following items: a task description, the definition of force-motion relations, a task execution strategy, command logic, control, and stability analysis [Whitney, 1987]. However a wide range of non-trivial problems still exist and many robot researchers regard force control to be one of the most sophisticated and challenging problems in robot control [Pei, 1992].

Most of the work in robot force control to date has been directed towards contact with rigid objects [Hopcroft *et al.*, 1991] and numerous problems remain unsolved in the rigid manipulator/rigid environment force control problem. However, 'real world' contact environments can be either compliant or rigid, and if robots are to perform 'real tasks in real environments' (e.g. street cleaning, refuse collecting, etc.) then they must be endowed with the capability of autonomously controlling interaction with both types of environment (and the transitions between the two).

At the outset of this project, robot contact with non-rigid environments was poor [Wada *et al.*, 1993] and this shortfall ultimately limits the range of tasks that robots can perform. The research presented in this thesis attempts to partially address this shortfall by investigating the use of a novel force control methodology that was developed to control the contact force between a rigid end-effector and a range of non-rigid environments. The control scheme uses an Artificial Neural Network (ANN) to model 'idealised reaction' (a concept that is explained in Chapter 4) to a range of environments, each with differing degrees of rigidity.

1.2 PROBLEMS ASSOCIATED WITH ROBOT FORCE CONTROL FOR CONTACT WITH NON-RIGID ENVIRONMENTS

Control of the contact force between a robot and non-rigid environments is made difficult by several factors, including:

- high levels of uncertainty in the contact environment's mechanical properties (i.e. spring constant of the contact material, frictional components, etc.).
- non-linear contact characteristics.
- for some tasks, variance in the degree of contact environment rigidity.

Non-rigid environments deform upon contact and the mechanical properties of the environment are generally unknown prior to contact occurring. Many 'real' non-rigid environments have non-linear contact characteristics that tend towards positional saturation [Caldwell and Gosney, 1993], and these characteristics may vary with environmental conditions (i.e. humidity, flexing, etc.). Additionally, considerable variance in the contact environments mechanical properties may be experienced during tasks that require a robot to track a non-rigid surface while simultaneously applying a force to the surface.

The mechanical properties of the contact environment have been shown to have a significant effect on the contact quality and, in some instances, force control system stability [Fukuda and Kitamura, 1986]. Thus, for robotic contact tasks, the environmental model should be central to the development of any force control scheme [Lewis *et al.*, 1993] and it is essential that forces are applied to the environment with consideration of the environment's mechanical properties.

However, the development of a widely representative contact model and a mechanism that can accurately estimate the model parameters in 'real time' is a non-trivial task, and the most frequently adopted contact models are linear in their parameters and time-invariant. Investigations into contact model parameter estimation have been reported [Venkataraman *et al.,* 1992; Fukuda *et al.*, 1987] and some of the significant studies are presented in Chapter 2.

1.3 THE ROLE OF INTELLIGENCE DURING HUMAN FORCE APPLICATION

At present, human perception and cognition far exceeds the capabilities of our most developed 'intelligent' robot systems. Humans mundanely perform a wide range of tasks that are far beyond the abilities of our most advanced robot systems, and these tasks are autonomously performed in diverse, unpredictable, and varying environments [Annaswamy and Seto, 1993]. Nature has evolved human intelligence over many thousands of years and a facility of 'real intelligence' (as opposed to artificial intelligence which, by comparison, is in its infancy) allows humans to adapt their behaviour to new task domains based on previous experience. Artificial intelligence is not so well developed and this shortfall leaves considerable scope for future development.

Humans (and many animals) have excellent manipulation and force application skills, which are acquired by learning through repeated interaction with objects of differing rigidity, shape, and orientation. The mechanism by which human 'motor skills' are acquired is not fully understood, but the acquisition of such skills has been attributed

to the formation and strengthening of neuronal connections within the brain [Patterson, 1996]. These connections are believed to be 'formed' in response to the human experiencing his environment (input stimulus).

It is worth noting that several aspects of human anatomy play a significant role in our excellent force application skills, namely excellent force sensing capabilities and advantageous mechanical characteristics such as compliance in key areas of the body (e.g. fingertips, feet, etc.) [Annaswamy and Seto, 1993]. At present, the mechanical characteristics, sensing capabilities, and most significantly 'intelligence' of our most advanced robots fall short of those of the human capability, and the accumulation of these shortfalls has limited the range of tasks that robots can perform. Advances in the areas of robot vision, mechanical design, and machine intelligence are required if robots are to perform tasks in wider domains.

Intelligence (or at least behaviour acquired through learning) plays an important role in the human force application capability and facets of human intelligence that are evident during force application are an ability to:

i. recognise when and where contact occurs.

ii. make the distinction between rigid and compliant environments and adjust behavior patterns accordingly when interacting with environments of differing rigidity.

iii. recognise constraint, both rotational and translational.

iv. simultaneously apply forces in some directions while moving in others (profile tracking with force application).

v. perform contact tasks without *a priori* knowledge of the degree of contact environment rigidity.

1.4 MOTIVATION FOR AN INTELLIGENT APPROACH TO FORCE CONTROL

A widely applicable force control strategy must be capable of performing tasks in challenging environments where the manipulator is faced with uncertainties and variations in its environment [Raibert and Craig, 1981]. Thus a force application robot must be capable of sensing variance in the contact environment and it must react to it in a safe and efficient manner.

Facets of human intelligence that are evident during human force application were outlined in section 1.3 and it is clear that without this intelligence, many tasks would be beyond the human capability. Thus strategies that incorporate aspects of human 'intelligence' into force control methodologies have potential for expanding the range of tasks that a robot can perform and the range of environments in which they can be performed. An approach of learning via experience and adaptation to new environments based on this experience is not only akin to the human force application learning mechanism but it is also characteristic of Artificial Neural Network (ANN) learning and generalisation.

1.5 PROJECT DESCRIPTION

1.5.1 INVESTIGATION APPROACH

The force control scheme investigated attempts to mimic the human force application learning mechanism and uses a connectionist model (an Artificial Neural Network) to 'learn' idealised reaction to a range of non-rigid environments. This was to be achieved by 'training' a neural network with data that specified ideal responses to several non-rigid environments. Once trained, the ANN was tested for its ability to perform accurate estimations with environments that were represented in the network training data set and 'unseen' environments. The trained ANN was then incorporated into a force control strategy and the control scheme's ability to apply forces to a range of non-rigid environments was investigated by simulation.

1.5.2 RESEARCH OBJECTIVES

The research objectives were to:

1. investigate factors that may influence an ANNs ability to accurately model 'idealised reaction' to a single non-rigid environment.
2. investigate the effect of noise on network performance.
3. develop a force control strategy that utilises the ANNs knowledge of 'idealised reaction' to a single environment so that a position controlled mechanical manipulator may be endowed with an ability to apply a desired force to the environment.
4. investigate, by simulation, the force control scheme's ability to apply forces to a range of non-rigid environments, each with differing degrees of rigidity. A force application range of 1N to 15N was investigated.

5. extend the ANN's knowledge by investigating the use of a single neural network to accurately model idealised reaction to a range of environments.
6. incorporate the 'multi-environment trained network' into the force control scheme and investigate, by simulation, the scheme's ability to acquire and maintain a specified contact with a wide range of non-rigid environments. Again, a force application range of 1N to 15N was investigated.
7. to investigate, by simulation, the intelligent force control system's ability to maintain contact with the environment when the degree of environmental rigidity is varied.

Control of contact with rigid environments was not within the scope of this investigation.

1.6 ORIGINALITY OF THE RESEARCH

The originality of the work lies in three areas, namely:

- the method by which the intelligent force control scheme applies forces to its environment. The control scheme applies forces to a range of non-rigid environments using a novel force control methodology.
- the use of a novel data extraction technique that was used to generate neural network training data.
- the use of a novel network training/selection procedure.

1.7 THESIS OUTLINE

This thesis is organised into eight chapters. Following this introductory chapter, Chapter 2 presents an overview of force control methodologies and contact models that are frequently used to represent non-rigid contact environments. Several previous studies of contact model parameter identification are presented.

Chapter 3 describes ANN principles that were considered relevant to the development of the intelligent force control scheme's ANN. The network topologies and training algorithms/methodologies investigated are also described.

Chapter 4 presents the development of the intelligent force control scheme and a single degree of freedom mechanical manipulator simulation that was implemented in the Advanced Continuous Simulation Language (ACSL). A novel data extraction technique that was used to extract ANN training data from a parameterised contact model is also presented.

Chapter 5 introduces an investigation into the use of a Radial Basis Function (RBF) network to model idealised reaction to a single non-rigid environment. Training methodologies that were used to obtain a network topology that was capable of performing a single environment input-output mapping to a high degree of accuracy are presented. Two RBF centre placement/optimisation methodologies were considered, random placement of the RBF centres and k-means clustering. Once trained, the RBF network was incorporated into the ACSL simulation developed in

Chapter 4 and the intelligent force control scheme's ability to apply forces to a range of non-rigid environments was investigated by simulation.

Chapter 6 extends the 'single environment' investigation to present an investigation into the use of an RBF network to model idealised reaction to a range of non-rigid environments, each with differing degrees of rigidity. Once trained, the RBF network was incorporated into the ACSL simulation developed in Chapter 4 and the intelligent force control schemes ability to apply forces to a range of environments was again investigated by simulation. The network's ability to perform accurate estimations with data extracted from a range of real environments was also investigated.

Chapter 7 presents an investigation into the intelligent force control system's ability to apply forces to varying environments. Conclusions obtained from the research and suggestions for further work are presented in Chapter 8.

Appendix A lists the contact model parameters that were used for the simulated non-rigid environments. Appendix B provides a list of parameters used for the ACSL simulation while Appendix C shows hardness profiles for simulated environments that were used to test network performance with unseen environments. Appendix D shows hardness profiles for the experimentally measured environments. Appendix E summarises the papers that were produced through this research.

CHAPTER 2

OVERVIEW OF FORCE CONTROL METHODOLOGIES AND CONTACT MODELS

2.1 INTRODUCTION

A considerable amount of research has been directed towards controlling the interaction force between a robot and its environment. Numerous force control strategies of varying complexity have been proposed and force control implementations with practical robot systems have been reported (albeit in controlled environments). However, force control applications in real environments (i.e. uncontrolled, uncertain, and varying environments) have been noticeably scarce, and robot force control has not yet reached real world requirements.

The complexity of the force control problem and the diversity of its potential applications has resulted in an abundance of research publications and, as such, this overview presents work that is regarded by the author as posing most relevance to the development of the work outlined in this thesis. This overview should not be taken to be a complete and comprehensive review of robotic force control techniques. However, several 'traditional' force control methodologies (i.e. early studies that laid foundations for robot force control research) have been included. The strengths and weaknesses of the control methodologies are highlighted where possible.

The intelligent control scheme was developed using a mathematical model that represented contact between a robot end-effector and it's contact environment. As such, a review of frequently adopted contact models is also presented. The environmental model has been shown to be central to any force control scheme [Lewis *et al.,* 1993] and the limitations of the contact models are highlighted. Methods for contact environment parameter identification are also presented.

2.2 ROBOT FORCE CONTROL

2.2.1 TRADITIONAL FORCE CONTROL METHODOLOGIES

The most notable of the early attempts to simultaneously control robot force and motion was the work by Paul and Shimano [1976] and Whitney [1977].

The control scheme proposed by Paul and Shimano was centred around the switching of certain robot joints to force control in response to Cartesian demands while the other joints remained under positional control. The joint based control scheme suffered from severe coupling problems and the transient switching between control states (i.e. position to force and vice versa) made the control strategy impractical for all but the simplest of applications.

Whitney presented his resolved motion rate control scheme that allowed a force control outer loop to be added to a velocity controlled inner loop (termed resolved rate force control). The technique was significant in that force demand could be specified in Cartesian co-ordinates.

Mason [1981] formalised the general manipulation task by proposing the use of a task aligned co-ordinate system (commonly known as the 'C' or 'local' co-ordinate system). The control scheme used the 'C' co-ordinate system to switch the appropriate degrees of freedom in and out of compliant control in response to forces sensed at the point of contact.

After the formalisation by Mason, two distinct trends in decoupled force control were proposed: namely hybrid position/force control and impedance control. It is worth noting that the vast majority of the 'new' force control methodologies extend the underlying principles of these force control methodologies, and intelligent and/or extended adaptations of both strategies are regularly published.

2.2.2 HYBRID POSITION/FORCE CONTROL

Hybrid position/force control (also known as 'hybrid control') was first proposed by Raibert and Craig [1981] but has its origins based in the early work by Mason. Hybrid control utilised a compliance selection matrix to switch local Cartesian co-ordinate axes to force or position control depending on the task. The control scheme uses separate position and force control loops and has been developed from a conceptual force control strategy to a practically feasible scheme (albeit under controlled conditions!).

Hybrid control has been proposed for a wide range of force control applications including: a multiple manipulation problem [West and Asada, 1992], door opening [Pujas *et al.*, 1993], and biped foot/floor control [Tsai and Orin, 1986].

However, hybrid control schemes have been reported to exhibit problems in mode switching [Mayeda and Ikeda, 1993] and force control stability [Steven, 1989].

2.2.3 IMPEDANCE CONTROL

Impedance control was first proposed by Hogan [1985]. The technique differed from previous schemes in that it did not require the specification of the end-effector force, but instead used the relationship between the end-effector position and force (known as the 'impedance' or mechanical stiffness). However, the control scheme was practically difficult to implement since it required complete knowledge of the robot dynamic model.

Impedance control has been proposed for applications as diverse as force application using dextrous space manipulators [Colbaugh *et al.,* 1992] and force tracking [Lasky and Hsai, 1991].

Kazerooni, *et al.* [1986] proposed a robust impedance controller which was achieved by choosing state feedback and force feedforward gains. Anderson and Spong [1988] proposed a hybrid impedance control scheme that combined an impedance controller with a hybrid position/force controller.

The main limitation of impedance control for compliant motion is that the manipulator-environment contact force is controlled indirectly by an appropriate choice of position trajectory [Seraji *et al.,* 1993]. However, in practical situations, the environmental parameters (i.e. stiffness and exact location of the contact surface) cannot be accurately specified, and as a result, impedance based force control schemes

exhibit poor force tracking characteristics [Jankowski and ElMaraghy, 1992].

Impedance control and hybrid control principles are highly significant in that their control structure is at the heart of the vast majority of intelligent and adaptive force control methodologies.

2.2.4 POSITION BASED FORCE CONTROL

A position based force control scheme was proposed by Maples and Becker [1986]. The control scheme used a position controlled manipulator (the vast majority of industrial robots, as supplied by the robot manufacturer, are position controlled devices) which took its positional demands from an outer force controlled loop. A compliant force sensor was added to the robot end-effector and the measured force was used to generate a force error. A model of the force sensor was used to convert the force error into an increment in the positional demand, and this was then passed to the positional control loop. The force control scheme was implemented on an Adept One robot and the resulting force response was stable but oscillatory. However, position based force control has been shown to be very sensitive to compliance in the manipulator/contact environment [Elosegui *et al.*, 1990].

2.2.5 INTELLIGENT FORCE CONTROL

There have been several attempts at controlling robotic interaction using artificial intelligence (AI) based control schemes. Two main AI methodologies have been employed, namely Artificial Neural Network (ANN) and Fuzzy Logic based force controllers.

Fukuda *et al.* [1990] proposed a Neural Servo Control scheme that had time delay elements in the networks first hidden layer node (two hidden layers were used) so that the network could learn the dynamics of the robot and the contact object. A 'fuzzy turbo', which is based on fuzzy set theory, was used to avoid stagnation during training. The proposed scheme used the 'hybrid' control structure proposed by Raibert and Craig [1981] with PID control applied to the position control loops and PI control applied to the force control loops. The ANN was used to decrease the error between the output of the hybrid controller and the torque measured at the point of contact. Simulation results applied to a two dimensional robot manipulator showed that the system could acquire and maintain a specified contact with the environment.

Tokita *et al.* [1991] proposed an ANN based 'hybrid' force control scheme that used a neural network to adjust the PID controller gains depending on the contact object type (i.e. soft or hard) and manipulator orientation. Simulation and experimental results were presented for a two degree of freedom mechanical manipulator. Results presented show that stable force control with both 'soft' and 'hard' objects was achieved using this control strategy.

Yabuta *et al.* [1990] proposed an ANN based control scheme that used a neural network to directly control a single degree of freedom force control servomechanism. Experimental results presented show that the system can acquire a specified contact, although stability problems were highlighted.

Pei [1992] proposed an alternative direct ANN force control strategy based on two cascading neural networks. Simulation results for a two link robot show that the force control scheme could acquire and maintain a desired contact with the environment.

Hollinger *et al.* [1993] developed a fuzzy logic controller that was used for the 'hard contact' problem. The control scheme was implemented on a MERLIN 6540 robot and the robustness of the controller was demonstrated by commanding the robot to follow a spiral trajectory on a table top while maintaining a four pound contact force. The force response was stable but oscillatory.

Dote *et al.* [1990] used a fuzzy force controller to control the grasping force of a manipulator hand. The control scheme was tested in its ability to apply forces to four objects of differing rigidity: a steel ball, a tennis ball, a sponge ball, and a soft tennis ball. The results presented show that the control scheme could acquire a desired contact with each of the objects.

2.3 CONTACT MODELLING

2.3.1 THE NON-RIGID CONTACT MODEL

Compliant (non-rigid) contact occurs when a robot end-effector (or, most frequently, tooling attached to the end effector) comes into contact with a surface/object that deforms upon contact (generally assumed to be local elastic deformation) and/or the compliances in the robot/force sensor/tooling have not been completely absorbed.

When in the compliant contact phase, the end-effector is not constrained in the direction of the sensed force, and can therefore move further into the contact surface, thereby increasing the contact force.

Whitney [1987] reported that the control of robots in contact with an environment requires at least the management of compliances rather than inertias. However, the task of compliance modelling and management is non-trivial since compliance modes may exist in the robot structure (some robots are designed to have mechanical flexibility [Surdhar *et al.*, 1996]), force sensor [Volpe and Khosla, 1994], tooling, and/or the contact environment itself. Each of the compliance modes may have non-linear and time-variant characteristics and the presence of several compliance modes thus makes accurate modelling of the total 'system' compliances a non-trivial task.

A contact model that is frequently used to represent contact with compliant environments is based on the spring damper equation:

$$F = k_1 x_{env} + k_2 \dot{x}_{env} \tag{2.1}$$

where F is the contact force, k_1 is the contact environment spring coefficient, k_2 is the contact environment damping coefficient, x_{env} is the depth into the environment, and $\dot{x}_{env}$ is the end-effector velocity through the environment.

The contact model shown in equation 2.1 is frequently used to represent not only compliance modes in the contact environment, but also compliance modes that may exist in the robot structure and force sensor [Sinha *et al.*, 1993]. This is illustrated in figure 2.1 which shows compliance modes in the contact environment, force sensor

exist in the robot structure and force sensor [Sinha *et al.,* 1993]. This is illustrated in figure 2.1 which shows compliance modes in the contact environment, force sensor and a lumped mass/spring/damper robot model.

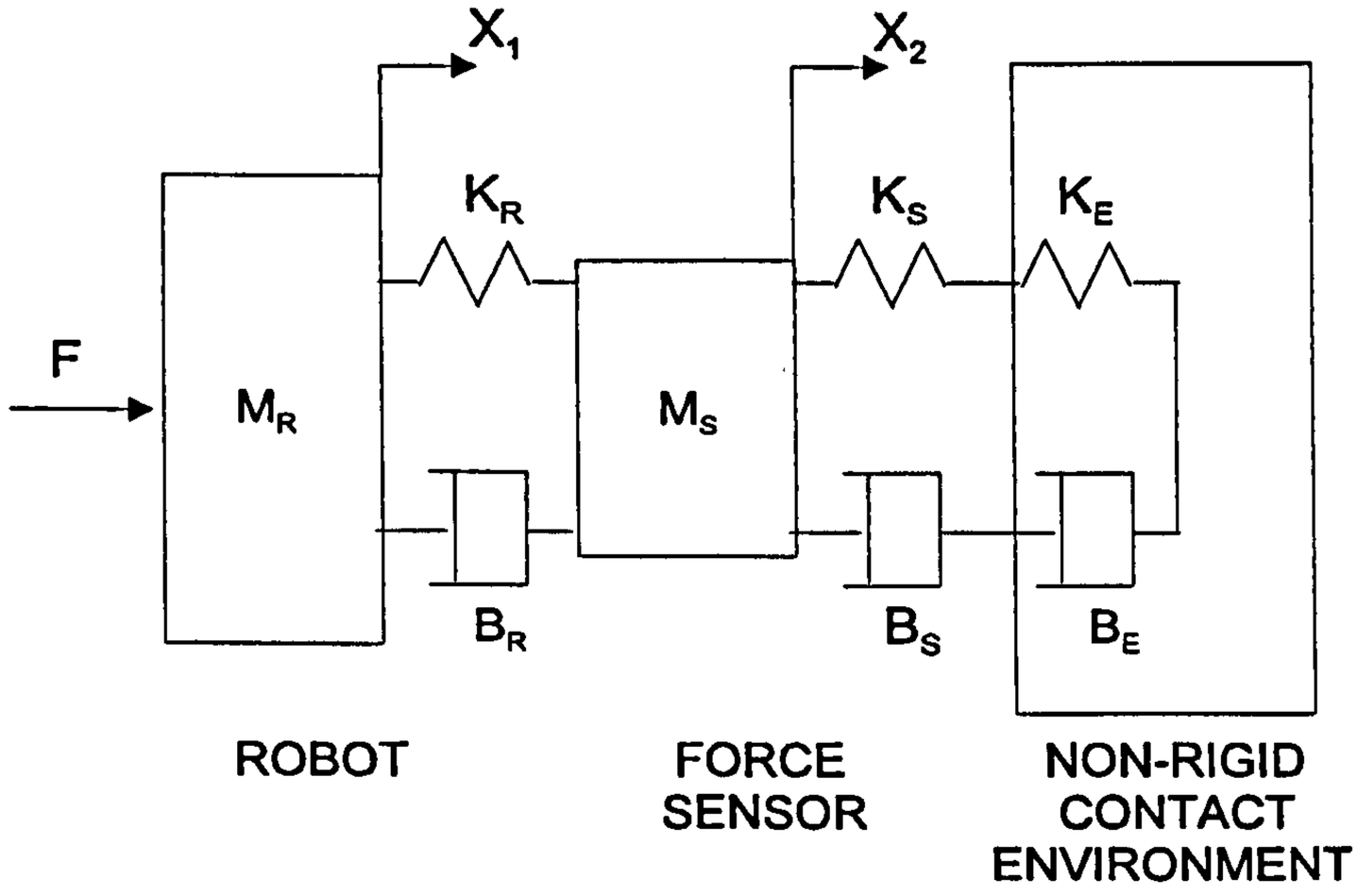


Figure 2.1 Compliance modes in the robot structure, force sensor, and contact environment. K_R, K_S, K_E are spring coefficients of the robot, force sensor, and contact environment respectively. B_R, B_S, B_E are damping coefficients of the robot, force sensor, and contact environment respectively. M_R and M_S are the respective masses of the robot link and the force sensor. X_1 and X_2 are the respective displacements of the robot end-effector and force sensor that resulted from contact with the environment.

Volpe and Khosla [1994] proposed a fourth order model that considered compliance modes in the robot arm, force sensor, and contact environment. The 'plant' model was validated experimentally and showed that the under 'controlled' conditions, the model parameters could be determined to a reasonable degree of accuracy. However, it is worth noting that for practical contact tasks it is difficult to estimate the contact model parameters in real time [Mayeda and Ikeda., 1993].

Forces generated by contact actually include impact dynamics, inertial forces, frictional forces, and reaction forces which are normally modelled as elastic deformation. However, at low speeds typical of robot contact, the dynamics are usually ignored and frictional forces are assumed to be proportional to elastically induced normal forces [Whitney, 1987]. Thus the contact model in equation 2.1 is frequently reduced to the linear spring contact model shown in equation 2.2.

$$F = k_1 x_{env} \tag{2.2}$$

where F is the contact force, k_1 is the contact environment spring coefficient, and x_{env} is the depth into the contact environment.

2.3.2 PRACTICAL LIMITATIONS OF NON-RIGID CONTACT MODELS

Hardness profiles of non-rigid contact materials [Caldwell and Gosney, 1993] have illustrated that many practical non-rigid materials have non-linear contact characteristics that tend towards positional saturation as the depth into the material increases (spring hardening in the contact environment). This is illustrated by the hardness profile shown in figure 2.2, where a tendency towards positional saturation can be seen to occur at a depth of D*sat*. At this point, the contact phase changes from compliant contact to rigid contact and the robot cannot move further into the contact environment. Control of contact in the rigid contact phase is beyond the scope of this work.

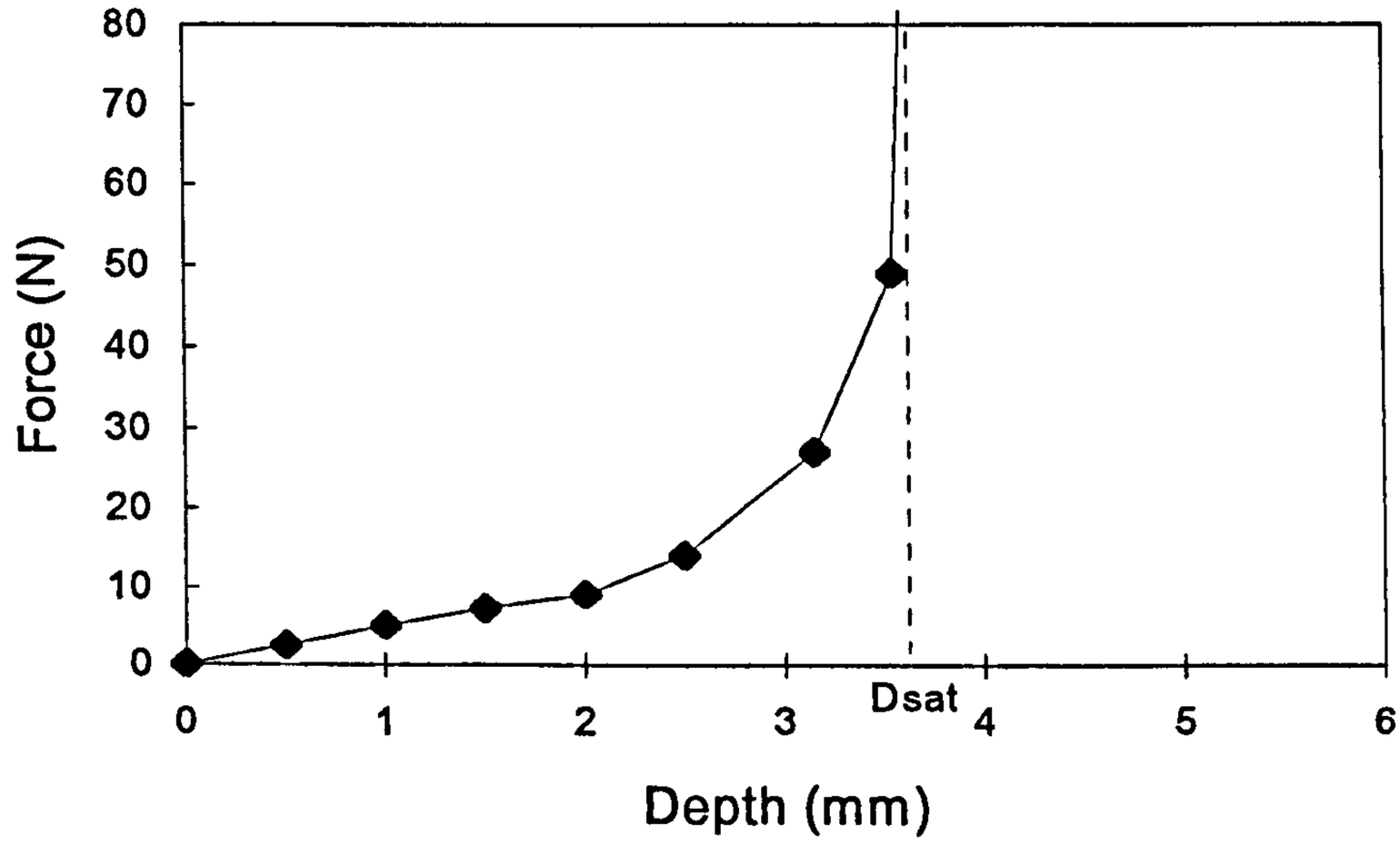


Figure 2.2 Hardness profile for a single non-rigid environment

It is worth noting that the contact environment's mechanical properties may vary not only with material type, but other factors such as material thickness, flexing, temperature, and humidity can significantly affect the contact environment's mechanical characteristics. Thus, many 'real' contact environments have not only non-linear characteristics, but, depending on the contact task and the environmental conditions in which the task is to be performed, they may also have time-varying properties [Guglielmo and Sadegh, 1994]. In light of these considerations, the fixed parameter linear contact model is unrepresentative of contact with many material types, since contact non-linearity, positional saturation, and variance in the contact characteristics are not represented in the model.

2.3.3 ENVIRONMENTAL PARAMETER IDENTIFICATION

An and Hollerbach [1987] recognised the possibility of explicitly identifying the environment stiffness for tuning a force controller. Fukuda and Kitamura [1986] used a *continuous* time model reference adaptive system to identify object dynamics, but

only considered a fixed linear model for a one degree of freedom prismatic mechanism. Yabuta and Yamada [1990] applied discrete time model reference adaptive control to the force control of a fixed linear single degree of freedom manipulator model and utilised identification of the object stiffness [Carelli *et al.*, 1990].

Lu and Goldberg [1995] proposed an impedance controller that estimated the stiffness parameter of a linear model of the contact environment. However, they reported that it was difficult to obtain exact environmental parameters and that the environment should be known as precisely as possible to reduce force error.

Venkataraman *et al.* [1992] addressed the problem of identifying uncertain environments by extending the contact model shown in equation 2.1 to include characteristics of environmental spring hardening and shock absorption. The work was targeted at controlling the contact force during rock coring and assumed that the dynamics of the contact environment had a fixed nonlinear structure. The form of the environmental model adopted was:

$$F = k_1 x_{env}^{\ a} + k_2 \dot{x}_{env}^{\ b} \tag{2.3}$$

where F is the contact force, k_1 is the contact environment spring coefficient, k_2 is the contact environment damping coefficient, x_{env} is the depth into the environment, $\dot{x}_{env}$ is the end-effector velocity through the environment and a and b were exponents that were included to represent spring hardening and shock absorption effects respectively. Compliance modes in the robot structure and tooling were not considered. The control scheme used a neural network to estimate the contact parameters in real time with a reasonable degree of success.

Compliance modes in the robot structure and tooling were not considered. The control scheme used a neural network to estimate the contact parameters in real time with a reasonable degree of success.

2.4 SUMMARY

Although there has been a significant amount of research activity in the area of robot force control, there still remains a need for fast, stable, robust, and widely applicable force control algorithms. A large proportion of 'new' and 'intelligent' force control strategies are based around either hybrid control or impedance control structures, with each scheme having their own inherent limitations.

Conventional hybrid control schemes suffer from a range of problems, in particular the switching of position and force control modes [Mayeda and Ikeda, 1993]. Impedance control schemes have practical limitations in that the manipulator-environment contact force can only be controlled indirectly by an appropriate choice of reference position trajectory, which is difficult to specify owing to uncertainty in the contact environment model form parameters. Position based control schemes have been shown to be highly sensitive to unmodelled compliance in the manipulator and/or contact environment.

Intelligent force control schemes generally extend traditional force control methodologies and the integration of 'intelligence' into traditional force control structures shows potential for producing practically useful force control methodologies.

Characteristics and limitations of frequently adopted contact models that are used to represent contact with non-rigid environments were presented and previous investigations of contact model parameter identification were highlighted.

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

3.1 INTRODUCTION

This chapter introduces the main characteristics of ANNs including the network architectures and training methodologies that were considered during the development of the intelligent force control scheme's ANN. The main emphasis of this chapter is the presentation of the structure, characteristics, and parameters of the Radial Basis Function network, the network architecture that was used to incorporate knowledge of idealised reaction into the force control scheme. Two methods of RBF centre placement/optimisation are introduced, namely random placement of the RBF centres and k-means clustering.

3.2 ARTIFICIAL NEURAL NETWORKS

ANNs were developed in an attempt to reproduce the learning capability and adaptability of the human brain, and they have been successfully used for a variety of applications including speech recognition [Christodoulou *et al.,* 1996;], financial forecasting [Dutta and Shekhar, 1988; Malliaris and Salchenberger, 1994] and process modelling and control [Doherty *et al.*, 1997; Zamarreno and Vega, 1996].

The human brain has been shown to comprise a highly interconnected array of processing units called 'neurons' (approximately 10^{10} neurons). Biological neurons receive inputs from sensory cells and a single neuron may be connected to as many as tens of thousands of other neurons [Patterson, 1996]. The neuron is said to 'fire' if the input stimulus is sufficient to overcome the neurons threshold. Although not fully understood, human learning and memory has been attributed to the formation and strengthening of connections between the neurons, and these connections are formed based on input stimulus.

Information in neural networks is not stored explicitly as in traditional computing techniques, but instead the input/output (I/O) relationship of the training data set is stored in interconnection weights that exist between the networks nodes. The network acquires knowledge of the I/O mapping via training, whereby the network training data is presented to the network for several complete iterations of the training data set. Network weights are adjusted by an optimisation algorithm that attempts to minimise the sum of the squared network prediction errors over the complete training data set. Several ANN architectures have been proposed, including:

- Multi-Layer Perceptron [Rumelhart *et al.*, 1986]
- Radial Basis Function network [Broomhead and Lowe, 1988; Moody and Darken, 1989]
- Recurrent Network [Kodogiannis, 1994]
- Kohonen Network [Kohonen, 1990]

3.3 THE MULTI-LAYER PERCEPTRON

Limitations in the usefulness of the single neuron model proposed by McCulloch and Pitts [1943] were realised with the publication of 'Perceptrons' (a term given to a logical thresholding neuron) [Minsky and Papert, 1969]. However, such limitations were overcome with the development of the backpropagation algorithm [Rumelhart *et al.*, 1986] which allowed neurons to be interconnected to form arrays or networks. The backpropagation algorithm allowed a 'neural network' to be trained to model complex I/O mappings.

The Multi-Layer Perceptron (MLP) trained using the backpropagation algorithm is one of the most frequently used neural network architectures. The feedforward structure of the MLP network is illustrated in figure 3.1, which shows an MLP architecture that comprises a single hidden layer.

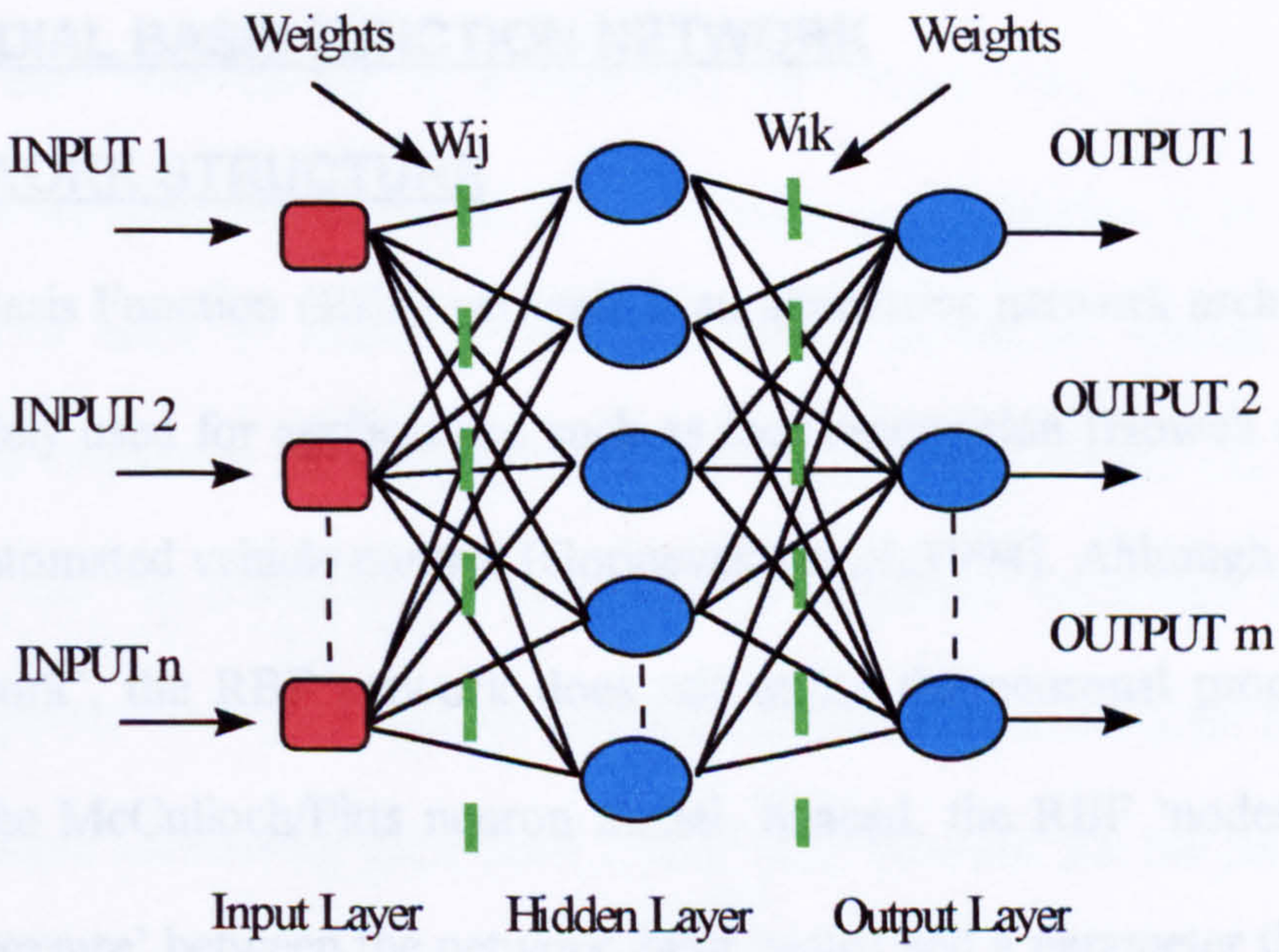


Figure 3.1. Multi-Layer Perceptron. ■ passive unit ● active processing unit.

As its name suggests, the MLP comprises several layers of neural processing units which are highly interconnected to form a network. The input neurons are passive and do not perform any processing on the inputs (i.e. they pass information directly to the hidden neurons via connection weights). The hidden layer neurons pass information to the next layer via a second set of connection weights. Both the hidden layer and output layer neurons are active and they process information through non-linear activation functions. Although the MLP is a widely used network architecture, MLPs trained using the backpropagation algorithm have several shortcomings, namely:

- a slowness to converge. The non-linear to-and-fro nature of the backpropagation algorithm results in long training times.
- a tendency to get stuck in local minima. With the MLP network, there is no guarantee that the global minimum will be reached.

3.4 THE RADIAL BASIS FUNCTION NETWORK

3.4.1 NETWORK STRUCTURE

The Radial Basis Function (RBF) network is an alternative network architecture that has been widely used for applications such as face recognition [Howell and Buxton, 1996] and automated vehicle control [Gorinevsky *et al.*,1994]. Although defined as a 'neural network', the RBF network does not utilise the neuronal processing units defined by the McCulloch/Pitts neuron model. Instead, the RBF 'nodes' perform a 'euclidean measure' between the network input vector and a parameter that is known as the nodes 'centre'. Additionally, the RBF architecture differs from the MLP in that

it comprises a single layer of non-linearities and a linear output layer. The RBF network structure is illustrated in figure 3.2.

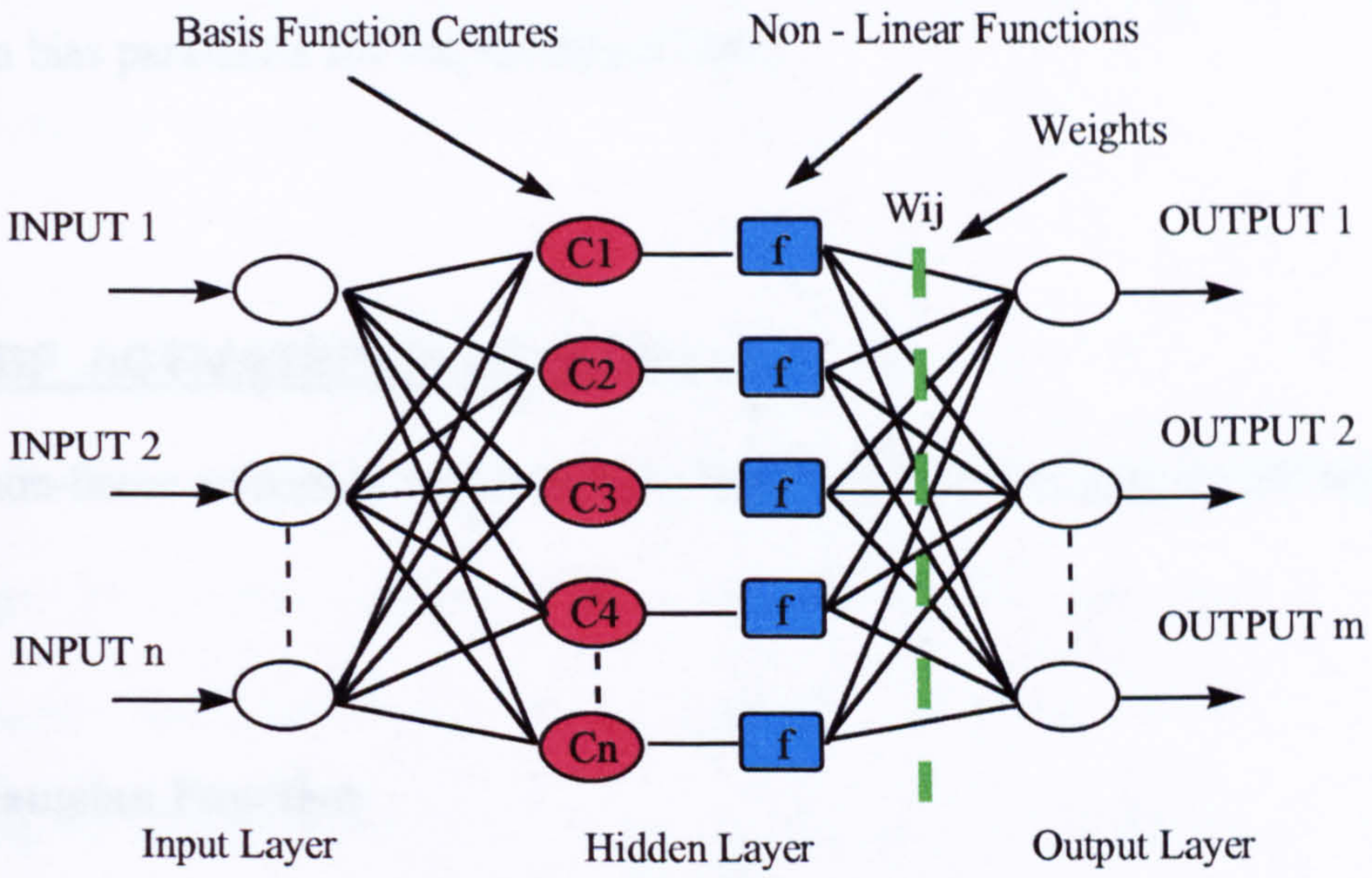


Figure 3.2 The Radial Basis Function network

The function of the input layer is to distribute the inputs unaltered to the hidden layer. Weights on links between the input layer and the hidden layer are set to unity and these weights are not updated during network training. The hidden layer comprises an array of nodes each of which contains a centre parameter vector. Each node calculates the Euclidean distance between the nodes centre vector and the network input vector and the result is passed through a non-linear activation function. The network outputs are computed as the weighted sum of the hidden layer node outputs. The input-output processing performed by the RBF network for the *j*th output node is expressed mathematically by equation 3.1.

$$y_j = \beta_j + \sum_{i=1}^{n_h} w_{ij} \phi_i \left(\| x - c_i \| \right) \tag{3.1}$$

where x is the input vector, c_i and $\phi_i(*)$ are the centre vector and output of the ith hidden node respectively, w_{ij} is the weight connecting the ith hidden node to the jth output node, n_h is the number of hidden layer nodes, $\|*\|$ denotes the Euclidean norm, and β_j is a bias parameter for the jth output node.

3.4.2 RBF ACTIVATION FUNCTIONS

Several non-linear activation functions have been proposed for use in RBF networks, including:

- **the Gaussian Function**

$$\phi(z) = \exp\left(z^2 / \rho^2\right) \tag{3.2}$$

- **the Thin Plate Spline**

$$\phi(z) = z^2 \log(z) \tag{3.3}$$

- **the Multi-Quadratic Function**

$$\phi(z) = \left(z^2 + \rho^2\right)^{1/2} \tag{3.4}$$

- **the Inverse Multi-Quadratic**

$$\phi(z) = \frac{1}{(z^2+\rho^2)^{1/2}} \tag{3.5}$$

where $\phi(z)$ is the non linear function output, z is the function input, and ρ is a parameter called the width.

Several algorithms for selecting appropriate values for the RBF centre width have been proposed including the p-nearest neighbour algorithm [Leonard and Kramer, 1991] and genetic algorithms [Kuo and Melsheimer, 1994]. However it has been reported that for non-linear function approximation the choice of basis function is not crucial to the performance of an RBF network [Chen *et al.*, 1990].

The network output layer is linear in its parameters with regard to the network weights thus, for a given set of hidden node centre parameters, the output layer weights can be computed using established linear regression algorithms. A recursive least squares algorithm [Ljung and Soderstrom, 1983] is frequently used for computing the weights during network training. Methods for partitioning the network training vectors to ensure that the network training data set is well conditioned (and therefore numerically robust) include LU factorisation [Bierman, G., 1977] and UD factorisation [Niu and Fisher, 1991].

3.4.3 RBF CENTRE PLACEMENT

A factor that can significantly influence RBF network performance is the placement of the hidden layer node centres. Researchers have reported that the RBF centres should sample the network input domain [Moody and Darken, 1989] but, at present, a method for initialising RBF centres that guarantees optimum centre placements does not exist.

Several algorithms for RBF centre placement have been proposed, including Random Placement [Broomhead and Lowe, 1988] and K-means Clustering [Chen and Billings, 1992; Hofland *et al.*, 1992], two centre placement methodologies that were investigated during the development of the intelligent force control scheme's ANN. Alternative centre placement algorithms that have been proposed include:

- Orthogonal Forward Regression [Chen *et al.*, 1992]
- Stepwise Regression [Pottmann and Seborg, 1992]
- Mean-Tracking Clustering [Warwick *et al.*, 1995]
- K-medeoids clustering [Kaufman and Rousseeuw, 1990]
- Branch and Bound [Eikens and Karim, 1994].

3.4.3.1 RANDOM PLACEMENT OF THE RBF CENTRES

The nature of the error surface that the optimisation algorithm searches during network training is dependent on the placement of the RBF centres. Random placement of the RBF centres involves the random placement of centres within the training data set input domain. However, RBF networks that have centre initialisations bounded by the input domain may not yield networks with optimum performance. Additionally, the use of a single centre randomisation is unlikely to produce an optimum centre placement.

3.4.3.2 K-MEANS CLUSTERING

One popular approach to determining the network centres is to use a learning method where the RBF centres are initialised to random points within the network training data set input domain and a clustering algorithm, such as k-means clustering, is then applied to place the centres in more optimal positions [Moody and Darken, 1989].

The k-means clustering algorithm partitions the input data set into k clusters and yields k cluster centres by minimizing the total squared error incurred in representing the data set by the k cluster centres. The method of using a recursive version of the k-means clustering algorithm is presented below:

- **Initialise Centres to Random Values within the Data Set**

$$c_j(0), \quad 1 < j < n_h \tag{3.6}$$

Following the initialisation step, the following computational steps are performed at each sample t:

- **Compute Distances and find Minimum Distance**

$$d_j(t) = \| x(t) - c_j(t-1) \|, \; 1 < j < n_h \tag{3.7}$$

$$k_{dist} = \arg[\min\{d_j(t), 1 < j < n_h\}] \tag{3.8}$$

- **Update Centres and Re-Calculate the kth distance**

$$c_j(t) = c_j(t-1),\ 1 < \mathrm{j} < \mathrm{n_h} \text{ and } \mathrm{j} \neq \mathrm{k} \tag{3.9}$$

$$c_k(t) = c_k(t-1) + \alpha_c(t)(x(t) - c_k(t-1)) \tag{3.10}$$

$$d_k(t) = \| x(t) - c_k(t) \| \tag{3.11}$$

where n_h is the number of hidden layer nodes and the learning rate α_c is :

$$\alpha_c(t) = \beta\alpha_c(t\text{-}1) \tag{3.12}$$

and β is the decay rate.

Figure 3.3 illustrates the k-means clustering algorithm redistributing the centres in an attempt to find more optimal centre positions. The centres are moved towards the input data space by minimising the sum of distances squared from each data point to the nearest centre.

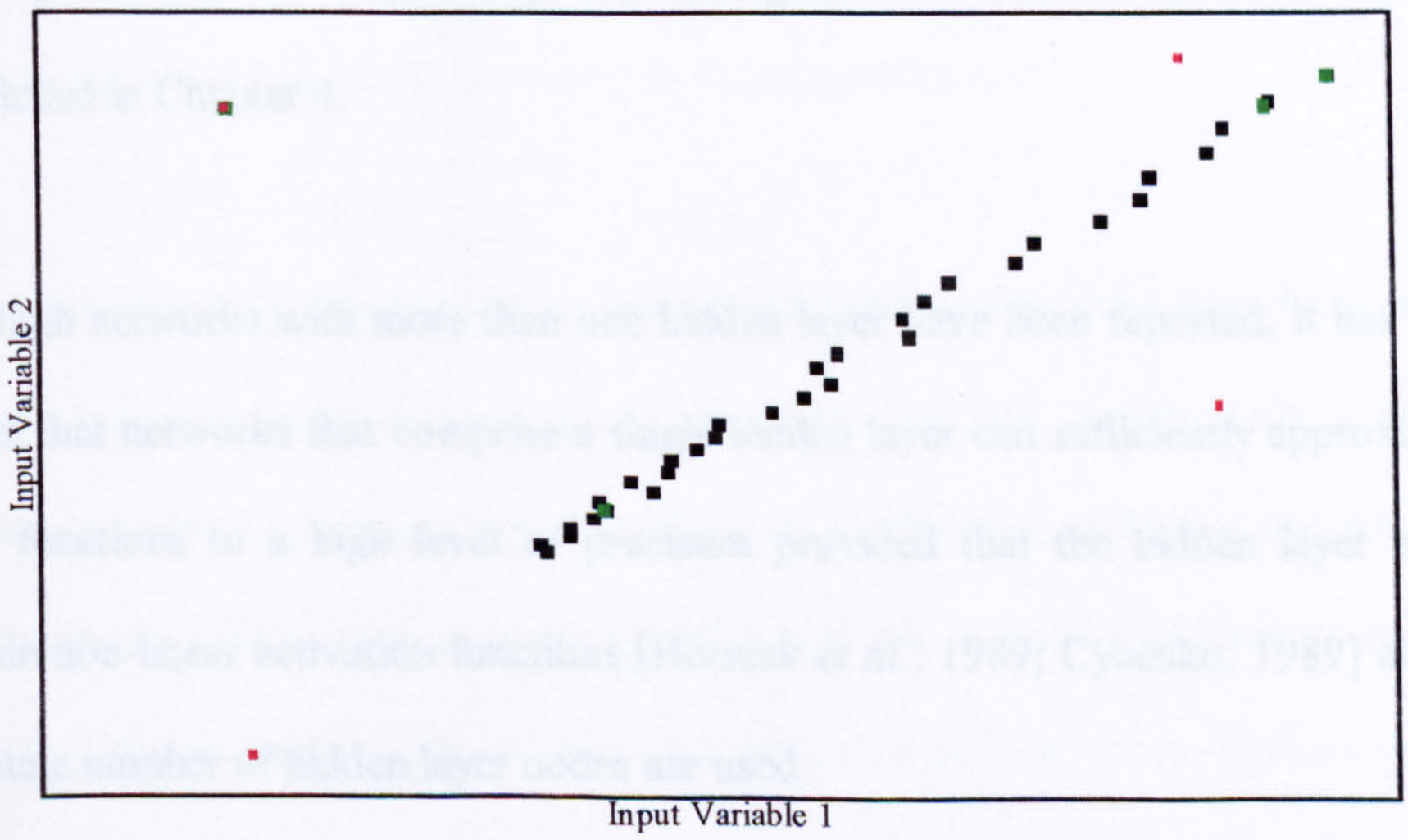


Figure 3.3. Redistribution of the RBF centres using the K-means Clustering algorithm. ■ Input data points ■ Initial centres placement ■ Final Centre position after k-means clustering.

It should be noted however that the k-means clustering does not guarantee that the final centre positions are optimally placed. Additionally, the final placement of the centres is dependent on the centre initialisation.

3.5 NETWORK TOPOLOGY SELECTION

Selecting a network topology to perform a particular I/O mapping requires choices to be made regarding:

- the number of network inputs and outputs.
- the number of hidden layers.
- the number of nodes used in each hidden layer.
- the type of activation function.

The number of inputs and outputs used in a network are dependent on the desired I/O mapping and the I/O structure of the intelligent force control scheme's ANN is introduced in Chapter 4.

Although networks with more than one hidden layer have been reported, it has been shown that networks that comprise a single hidden layer can sufficiently approximate most functions to a high level of precision provided that the hidden layer nodes contain non-linear activation functions [Hornick *et al.*, 1989; Cybenko, 1989] and an adequate number of hidden layer nodes are used.

A factor that can significantly influence RBF network performance is the number of nodes included in the network's hidden layer. If too few hidden nodes are used, then a network's ability to model a non-linear mapping between the input and output variables may be limited. Whereas, the use of too many hidden neurons can result in over parameterisation of the network, which can result in the learning of the process noise characteristics and consequently poor generalisation properties are exhibited. At present, empirical methods for selecting an optimum number of hidden layer nodes to perform a particular input-output mapping do not exist.

3.6 DURATION OF NETWORK TRAINING

Training a neural network to perform an I/O mapping involves the presentation of the complete training data set to the network for several epochs. An optimisation algorithm adjusts the network weights in an attempt to map the data set inputs to the corresponding data set outputs, for the complete data set. Updating of the weights is performed after each epoch by the optimisation algorithm.

If the neural network experiences too few passes of the training data set, the network will not acquire suitable knowledge of the data sets I/O mapping. As a result, estimations with the training data set will be poor. However, if the neural network experiences too many passes of the training data set, network estimations will be accurate with the training data set but poor with data not used to train the network [Evans, 1994]. Figure 3.4 illustrates the effect of overtraining on the network output Mean Square Error (MSE).

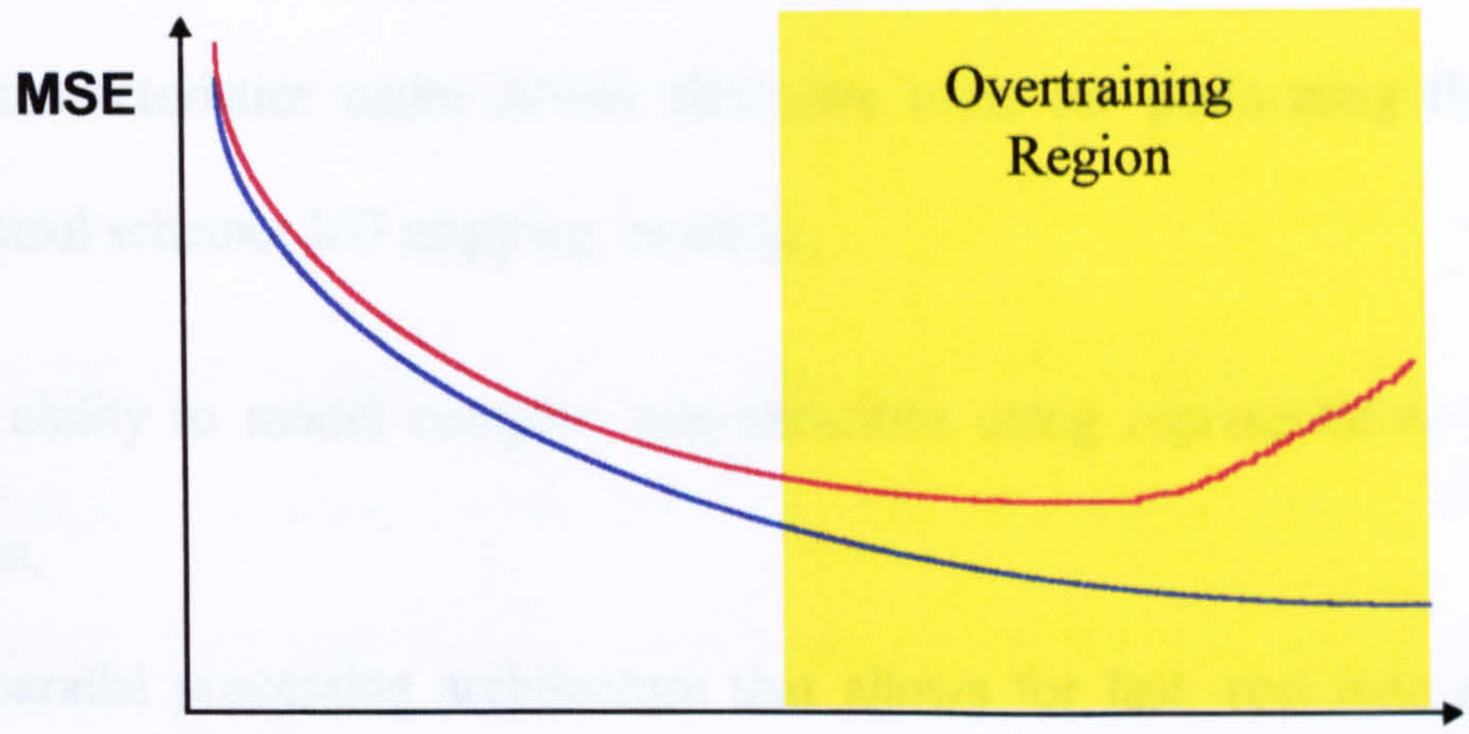


Figure 3.4 Illustration of ANN overtraining ■ Training data set ■ Test data set

Initially a large MSE is found since the network has not experienced enough passes of the training data set and therefore has insufficient knowledge of the training data I/O mapping. With further training, the MSE obtained with the training data set can be seen to decrease (although the decrease is minimal after a certain number of training epochs). However, network estimations with 'unseen' data can be seen to deteriorate (as characterised by an increase in MSE). Network 'overtraining' occurs because the weights have been adjusted such that the network has 'over-fitted' the training data set and the network has 'memorised' the specific data vectors in the training data set. A 'correctly' trained network does not store knowledge of individual data vectors, but instead learns the overall I/O relationship of the training data set. Generalisation to unseen data sets is of paramount importance since a network must be capable of performing accurate estimations with data not represented in the network training data set.

3.7 ARTIFICIAL NEURAL NETWORK CHARACTERISTICS

Several characteristics make ANNs attractive tools for performing the intelligent force control schemes I/O mapping, namely:

i. an ability to model complex non-linearities using representative input-output data.

ii. a parallel processing architecture that allows for fast, real time operation (in hardware). This factor is highly significant since practical force control strategies must be capable of sensing and reacting to the contact environment in 'real time'.

iii. an ability to operate in noisy environments.

iv. the ability to generalise to unseen inputs within the network training data range.

v. the ability to learn by example.

3.7.1 ARTIFICIAL NEURAL NETWORK MODELLING OF NON-STATIONARY SIGNALS

Although ANNs have several characteristics that make them desirable for modelling poorly defined and/or non-linear systems, it has been difficult to apply them to nonstationary (i.e. time-varying) signals.

Neural adaptation to time-varying domains requires an additional mechanism to enable the ANN to model systems with inherent time-varying characteristics [Levin, 1993]. This generally requires that the ANN learning process is kept active throughout the control operation [Takahashi, 1993]. However, an approach of 'on-line learning' is not feasible for the intelligent force control scheme since:

- for many practical force application tasks, the speed of the variation in the contact environments mechanical properties does not allow for the network weights to be updated in real time.
- on-line network training results in a 'forgetting' of previous knowledge when new knowledge is attained by the network. The non-cyclic nature of most force application tasks and the non-predictable nature of 'real world' environments requires that the force control scheme's ANN has 'metaknowledge' of contact with a range of environments with differing degrees of rigidity. The network can then instantaneously draw upon this knowledge when contact with the environment is sensed.

An investigation of the storage of knowledge of idealised reaction to a range of non-rigid environments is presented in Chapter 6.

3.8 SUMMARY

The network architectures that were considered during the development of the intelligent force control scheme were introduced. The emphasis of this chapter was the presentation of the structure, parameters, and characteristics of the Radial Basis Function network, the network architecture that was used for the intelligent force control scheme.

Factors that were considered in the choice of network architecture were presented. Methods of initialising the RBF centres were introduced, namely random placement of the RBF centres and k-means clustering.

The main characteristics of ANNs were highlighted, several of which made ANNs ideal 'tools' for incorporating knowledge of idealised reaction with a range of environments into the intelligent force control scheme. The suitability of ANNs for modelling non-stationary signals was also discussed.

CHAPTER 4

INTELLIGENT FORCE CONTROL SCHEME AND SIMULATION DEVELOPMENT

4.1 INTRODUCTION

This chapter introduces the development of the intelligent force control scheme. Included is the definition of the ANN input/output structure (i.e. the selection of the ANN inputs and outputs) and the ANN training/test data generation methodology. The latter required the use of a contact model to describe the 'behaviour' of the contact environment during contact. The ANN training/test data sets were generated from the parameterised contact model using a novel data extraction technique. Methodologies that were used for training the intelligent force control scheme's ANN are introduced.

A simulation of a single degree of freedom (DOF) position controlled mechanical manipulator is developed along with the method by which the 'trained' ANN was incorporated into the position based control scheme. The simulation was developed in the Advanced Continuous Simulation Language (ACSL) so that the force control system's ability to apply forces to a range of non-rigid environments could be investigated.

4.2 INTELLIGENT FORCE CONTROL SCHEME DEVELOPMENT

4.2.1 CONTROL SCHEME OBJECTIVES

The intelligent force control scheme objectives were to:

1. apply forces to a range of non-rigid environments, each with differing degrees of rigidity. This was to be achieved without *a priori* knowledge of the contact environment's mechanical properties. A force application range of 1 to 15N was chosen for the investigation. Contact with rigid environments was not within the scope of the investigation.
2. maintain performance when noise was superimposed onto the force and depth measurements.
3. maintain a desired contact with a range of non-rigid environments when the degree of environmental rigidity changes.

4.2.2 FORCE CONTROL SCHEME RATIONALE

In the case of contact with non-rigid environments, it is possible to control the contact force by regulating the end-effectors position [Zhou, 1991]. Movement into a non-rigid environment results in an increase in the contact force and conversely movement out of the environment results in a decrease in the contact force. However, the degree of movement required to achieve a required force is dependent on the environments mechanical properties, which have been shown to have a significant effect on the quality of the force control. The mechanical properties of 'real' contact environments have non-linear and, for certain tasks, time-varying characteristics which are difficult to accurately model. Additionally, the contact characteristics are unknown prior to contact.

The intelligent force control scheme uses an Artificial Neural Network (ANN) to vary the position of the robot end-effector relative to the contact surface such that the contact force is as required. The ANN's role in the intelligent force control scheme is illustrated in figure 4.1.

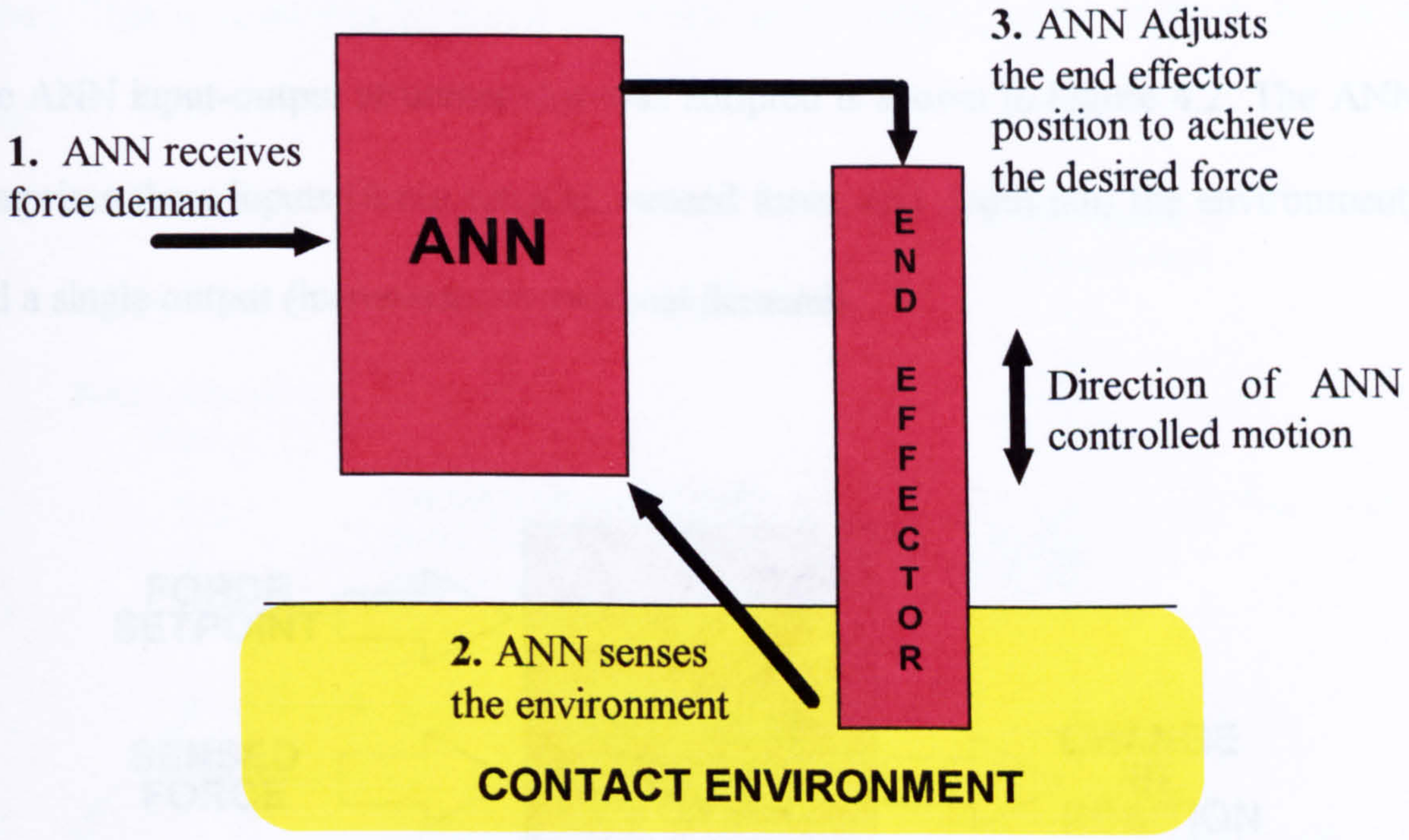


Figure 4.1 Role of the ANN in the intelligent force control scheme

To achieve its goal, the ANN must be capable of learning 'idealised reaction' to a range of environments. This is further explained in section 4.3.2.

4.2.3 DEFINING THE ANN INPUT/ OUTPUT STRUCTURE

In order that the ANN could be trained, it was necessary to define the network input/output (I/O) structure. Initially, several candidate ANN inputs were considered based on physical quantities that can be practically measured (or derived from such measurements) at the point of contact. The quantities that were considered were:

- contact force
- contact torque
- displacement (depth) into the contact material
- velocity through the material

The ANN input-output structure that was adopted is shown in Figure 4.2. The ANN comprises three inputs (force setpoint, sensed force, and depth into the environment) and a single output (increment in positional demand).

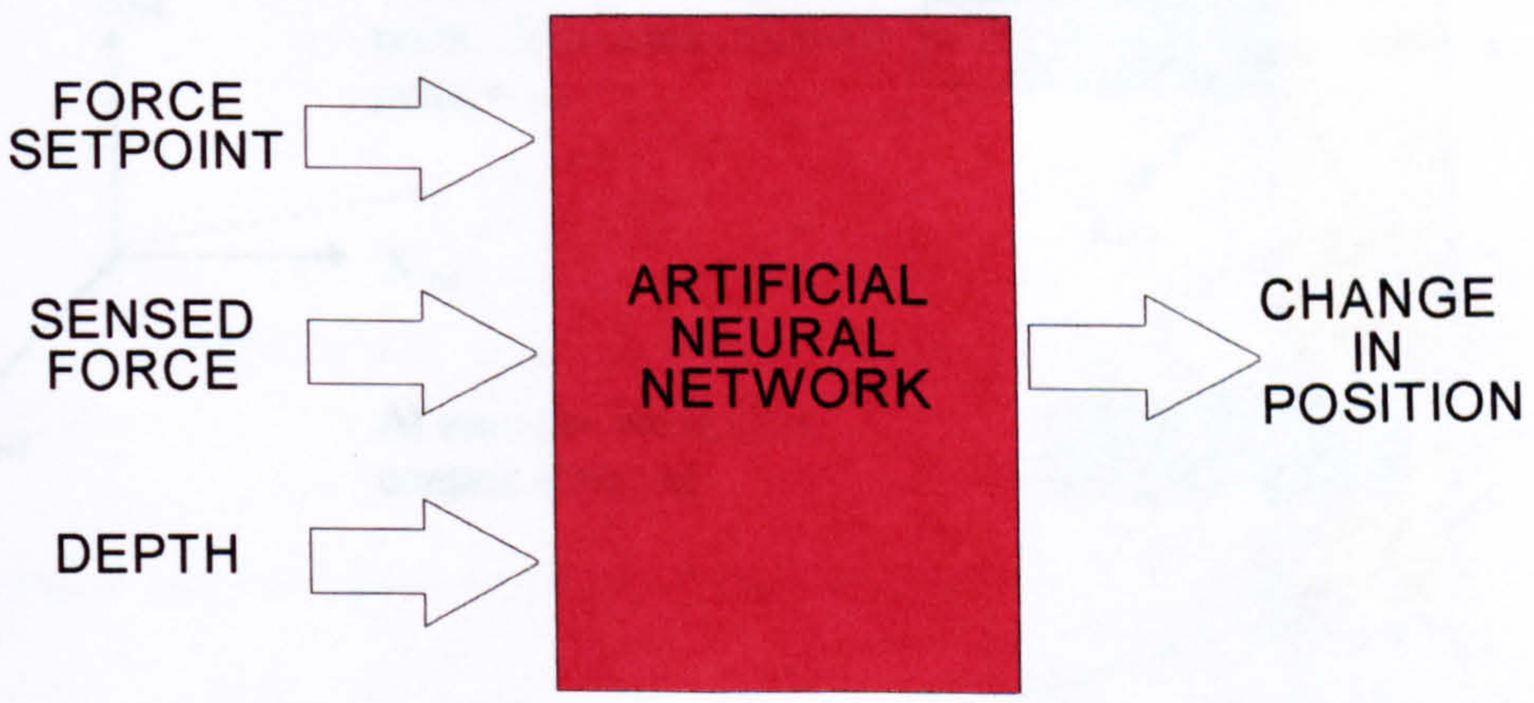


Figure 4.2 Intelligent force control scheme's ANN input/output structure

Practically, the contact force could be measured directly by a force/torque sensor attached to the robot end-effector. An alternative method of force sensing is the monitoring of motor currents which has been shown to be inaccurate and dependent on robot loading and configuration [Luk, 1991]. Thus the directly measured contact force can be used as an ANN input.

Depth measurements can be obtained by using a backward difference between the present position into the environment and the point at which the contact was originally sensed. These positions can be calculated relative to the robots reference frame using a task aligned reference frame. The backward difference between these two stored values would give an absolute measurement of the depth at a given instant. This is illustrated by figure 4.3 which shows contact at two discrete points in time.

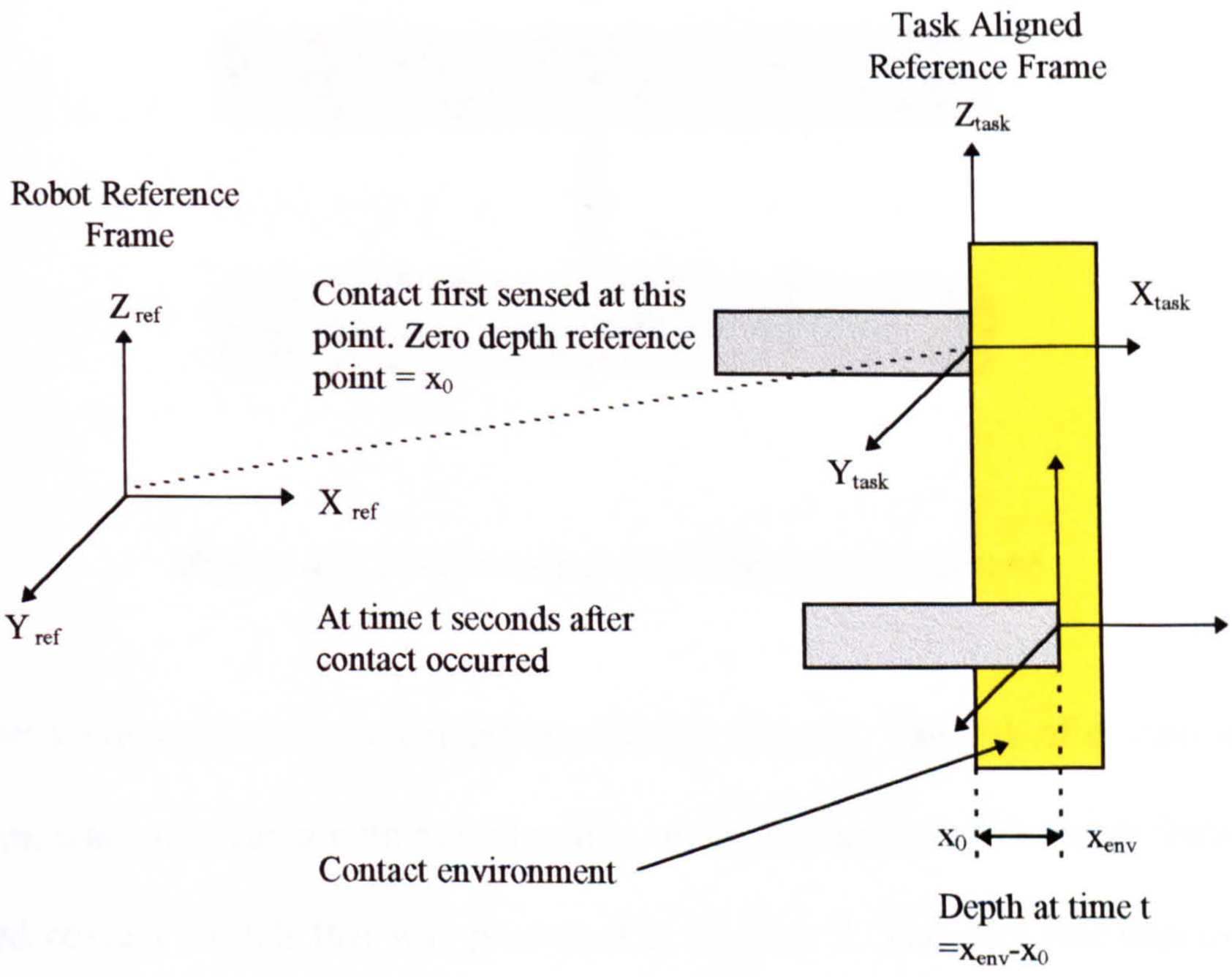


Figure 4.3 Depth measurement using backward difference

It should be noted that the development of 'real time' depth measurement techniques was not within the scope of this work.

4.3 NETWORK TRAINING DATA GENERATION

4.3.1 DATA GENERATION METHODOLGY OVERVIEW

The ANN training data sets were generated using a novel 3 stage data extraction procedure which is illustrated in figure 4.4.

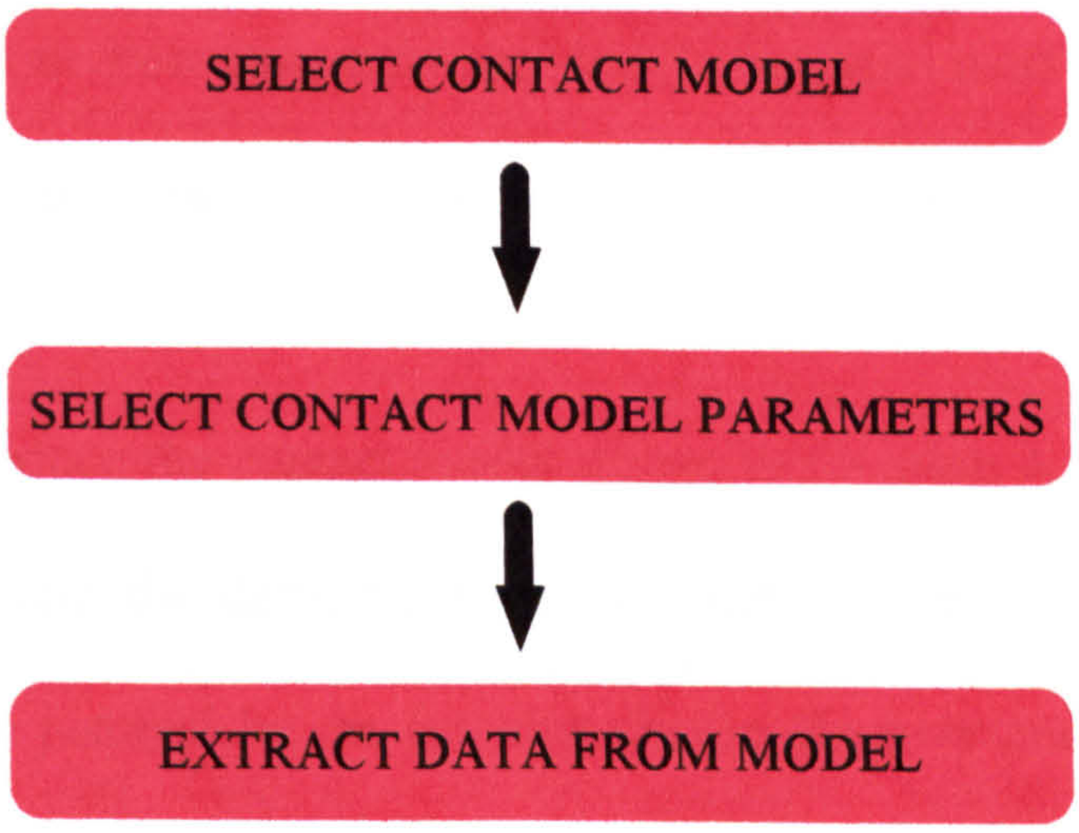


Figure 4.4 ANN training data extraction procedure

The first stage required that a contact model was selected. The task of contact model selection was undertaken with consideration of the limitations of the most frequently adopted contact models that was presented in Chapter 2. The data that was used to train the ANN was generated from the parameterised contact model using a novel data extraction technique which is described in section 4.3.

4.3.1.1 CONTACT MODEL SELECTION

The contact model that was initially considered to model compliance in the contact environment was based on the non-linear contact model proposed by Venkataraman *et al.*, [1992]:

$$F=k_1 x_{env}{}^{a} + k_2 \dot{x}_{env}{}^{b} \qquad (4.1)$$

where F is the contact force, k_1 is the contact environment spring coefficient, k_2 is the contact environment damping coefficient, x_{env} is the depth into the material, $\dot{x}_{env}$ is the velocity through the environment, and a and b are exponents used to represent spring hardening in the contact environment and shock absorption respectively.

It was decided to reduce the contact model shown in equation 4.1 to generate static contact data by setting the damping coefficient (i.e. k_2) equal to zero. The static investigation can be justified by the fact that during the execution of contact tasks, end-effector velocities are significantly lower than for unconstrained motion [Whitney, 1987] and, as such, the dynamic components of the contact model will be small in relation to the static model components.

The effects of spring hardening in the contact environment were modelled as being proportional to the square of the depth into the environment. Separate coefficients were used for the spring and spring hardening model components. Hence, the non-linear contact model structure that was adopted for the intelligent force control scheme investigation was:

$$F=k_1 x_{env} + k_3 x_{env}{}^{2} \qquad (4.2)$$

where F is the contact force, k_1 is the contact environment spring coefficient, k_3 is the contact environment spring hardening coefficient, and x_{env} is the depth into the environment.

4.3.1.2 GENERATING SIMULATED ENVIRONMENTS

Simulated contact environments were generated to represent a range of non-rigid environments, each with differing degrees of rigidity. Twelve simulated environments were generated in total by varying the contact model coefficients (i.e. the values of k_1 and k_3) in equation 4.2. Six of the simulated environments were to be used as the source of data used to train the ANN and the remaining six data sets were to be used as the source of data used to test the network estimations with 'unseen' environments. The twelve sets of simulated contact environment parameters are tabulated in Appendix A.

4.3.1.3 DATA EXTRACTION

Once a contact model form and parameters were selected, it was possible to extract the ANN training data from the parameterised contact model. The data extraction technique involved calculating the change in position, δpos, that would make the sensed force equal to the force setpoint, for a range of force setpoints and sensed forces. The force application range was chosen to be between 1N and 15N. However, data for contact up to 20N was included to ensure that 'bounded' estimations were obtained should the force control system 'overshoot' the upper limit of the force application range (i.e. >15N).

The data extraction technique is illustrated in figure 4.5 which shows the hardness profile for a single non-rigid environment.

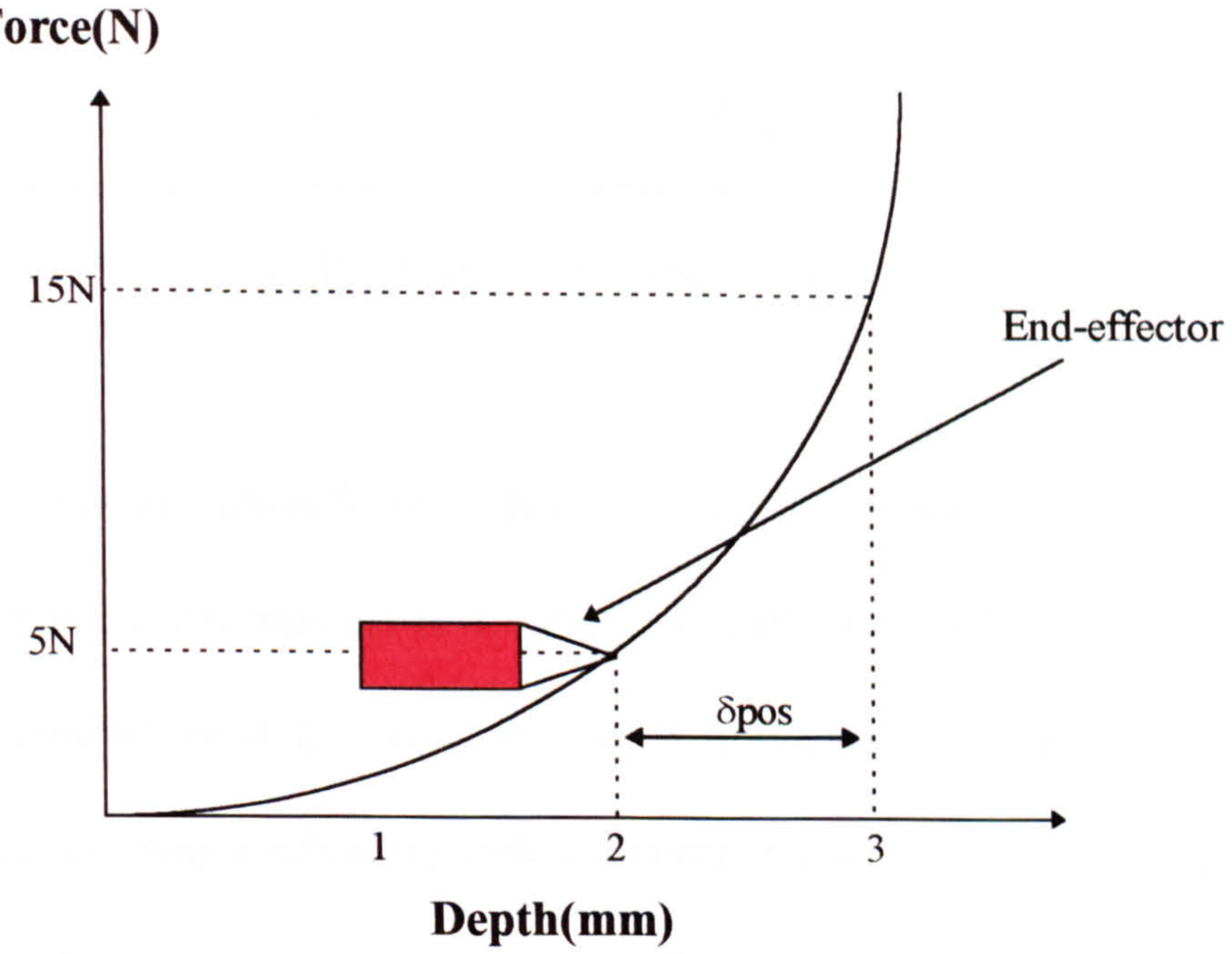


Figure 4.5 Data extraction from a hardness profile

In the example shown, the desired contact force (i.e. the force setpoint) is 15N which, for this environment, would occur at a depth of 3mm into the environment. The sensed force after the initial impact is 5N which, for this environment, would occur at a depth of 2mm into the environment. Thus the robot end-effector must be moved 1 mm further into the contact environment to make the desired contact. The idealised positional increment (which is also referred to as 'idealised reaction'), δpos, was tabulated for a force setpoint/sensing range of 1N to 20N, with both ranges being incremented in 1N steps. Thus the form of the ANN training data sets is illustrated in figure 4.6.

FORCE SETPOINT(N)	SENSED FORCE(N)	DEPTH(mm)	CHANGE IN POSITION(mm)
1	1	0.75	0
2	1	0.75	1.75
3	1	0.75	2.22
⋮	⋮	⋮	⋮
20	20	5.92	0

Figure 4.6 Illustration of ANN training data set

Two data sets were extracted from each of the simulated environments: a network training data set and a test data set that comprised data extracted at random from the parameterised contact model, across the force application range. Since the data extraction procedure was a laborious task a computer program was developed, using the Pascal programming language, to extract the ANN training/test data from the parameterised contact model. Twelve ANN training data sets and twelve test data sets were generated in total.

4.3.2 NATURE OF THE ANN INPUT-OUTPUT MAPPING

Although the ANN maps 3 input variables (2 force quantities and one positional quantity) to a single output (a positional quantity), the ANN output can be considered as being a function of two distinct input pairings. By considering the ANN input-output structure, the increment in the end-effector position (i.e. the ANNs output) can be seen to be a function of :

- the magnitude and direction of the force error. This is characterised by the difference between the input pair: sensed force and force setpoint.

- the contact environment rigidity. This is characterised by the non-linear relationship between the input pair: sensed force and depth into the environment.

Thus the knowledge contained in the training data set that is extracted from a particular environment comprises knowledge of 'idealised reaction' to the environment, across the force application range represented in the training data set. This knowledge is stored in the network weights when the ANN is trained.

4.4 ARTIFICIAL NEURAL NETWORK DESIGN

4.4.1 ANN ARCHITECTURE SELECTION

Once the ANN I/O structure had been defined, it was necessary to decide upon the network type. The Multi-Layer Perceptron (MLP) trained using the back-propagation algorithm was initially considered, but the MLP is notoriously slow to converge and it does not always converge to a global minimum. The Radial Basis Function (RBF) network is an alternative network architecture and the RBF's main characteristics were introduced in Chapter 3. The RBF network is fast to converge with training times a fraction of those required to train a similarly sized MLP. This can be attributed to the fact that the RBF output layer is a set of linear combiners that uses a linear regression algorithm, a factor which guarantees fast convergence that always reaches the global minimum. Additionally, the RBF network has been shown to have excellent non-linear input-output mapping capabilities and as such, the RBF network was used throughout the investigation.

4.4.2 RBF ACTIVATION FUNCTION SELECTION

Once the network type had been decided, it was necessary to select the non-linear activation function to be used in the RBF hidden layer nodes. Several non-linear functions have been proposed (e.g. Gaussian, multi-quadratic, etc.) and all but one (the thin plate spline) contain an additional parameter known as the 'width'. Chen and Billings [1992] reported that the type of non-linear function is not critical, and as such, the thin plate spline was chosen since it eliminates the need to specify the width parameter. The thin plate spline activation function can be mathematically expressed as:

$$\phi(z) = z^2 \log(z) \tag{5.3}$$

where ϕ is the output of the non-linear function and z is the function input.

4.4.3 ANN TRAINING ENVIRONMENT

The RBF network training and testing was performed using a software package that was specifically developed for the investigation. The package was developed in the Pascal programming language and allowed for the following options to be specified for network training:

- selection of the network topology.
- the number of training epochs.
- the spread of the RBF centre initialisation as a multiple of the training data set input domain.
- the inclusion of a network bias node.

- choice of random or sequential selection of the network training data vectors.
- choice of ANN training methodology.

The training methodologies that were implemented in the software were random placement of the RBF centres and k-means clustering.

4.4.3.1 REPEATED RANDOM PLACEMENT

In an attempt to find suitable centre placements, a repeated random centre placement methodology was employed since preliminary experiments showed that a single random centre placement did not produce networks with adequate performance. The repeated random placement training/testing/selection procedure adopted is illustrated in figure 4.7.

The steps involved in implementing the procedure are:

1. define the network topology.
2. initialise the RBF centres to sample the training data set input domain.
3. train the network. This involved the presentation of the network training data to the network for several complete iterations of the training data set. The network weights were updated after each iteration.
4. test the network with two data sets: the network training data set and a test data set.
5. re-initialise the RBF centres and repeat the training procedure for 500 centre randomisations.

6. retain 2 networks (out of the 500 networks trained), the network that had the smallest Mean Square Error (MSE) with the training data set and the network that had the smallest MSE for the test data set.

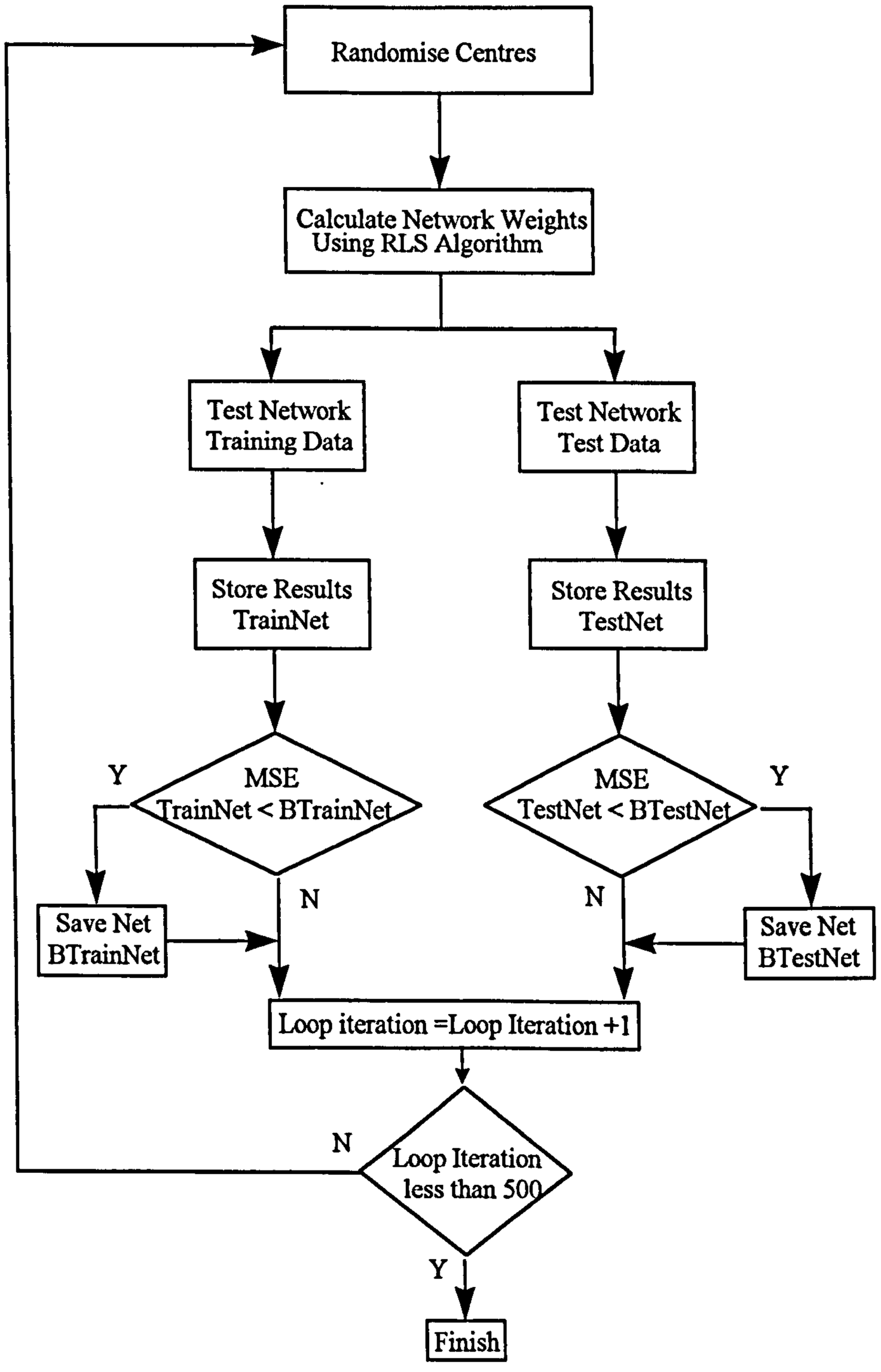


Figure 4.7 Repeated random placement training/selection procedure

4.4.3.2 K-MEANS CLUSTERING

An alternative centre placement/optimisation technique was considered during the investigation into the training of the intelligent force control scheme's ANN. The k-means clustering algorithm introduced in Chapter 3 was investigated. The k-means clustering algorithm attempts to redistribute the RBF centres so that they are suitably placed around the training data set input vectors. However, the use of k-means clustering does not guarantee optimum centre placements and the final centre positions are dependent on the RBF centre initialisation. As such, the repeated random placement training/selection procedure described in the previous section was repeated, but k-means clustering was performed prior to network training.

4.5 SIMULATION DEVELOPMENT

4.5.1 FORCE CONTROL SCHEME ARCHITECTURE

Once the ANN was developed it was incorporated into a positioning control scheme. The addition of the ANN would give the positioning device an autonomous force control capability. Figure 4.8 illustrates the ANN's integration into a positional control scheme. The scheme comprises an inner position control loop with an 'intelligent' positional adjustment loop. Translational motion is controlled in a single degree of freedom. The ANN senses the environment and selects an appropriate increment in the end-effector position to achieve a desired force. The 'intelligent' outer loop increments the positional setpoint X_{spt} by the amount δpos (the ANN output) in response to depth and force measurements taken at the point of contact (two of the ANN inputs). The force demand (the force setpoint) was also a network input.

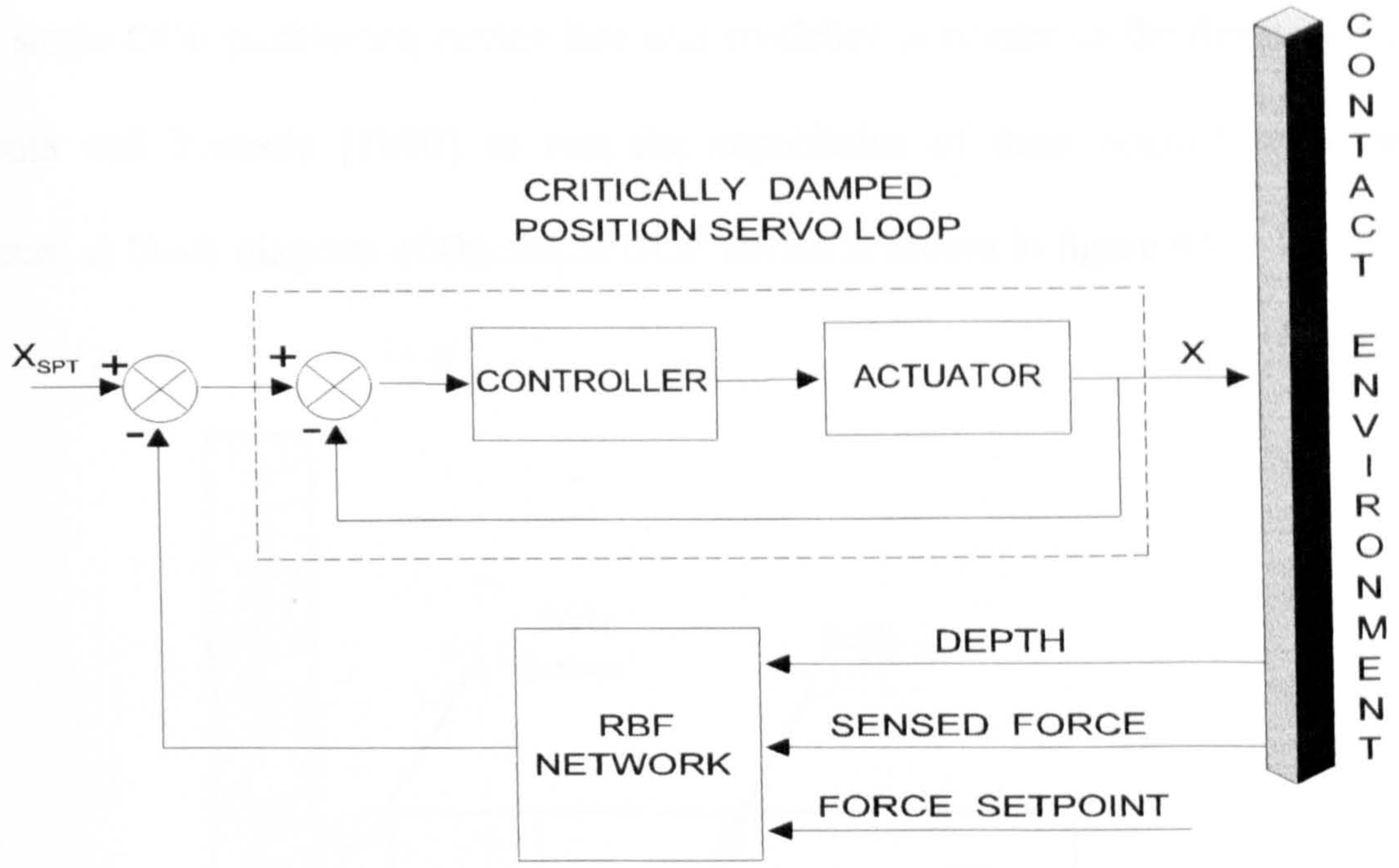


Figure 4.8 Single degree of freedom neuro-force control scheme

4.5.2 FIRST PRINCIPLE MODEL OF THE SINGLE DEGREE OF FREEDOM MANIPULATOR

A simulation of a single degree of freedom (DOF) mechanical manipulator was developed in the Advanced Continuous Simulation Language (ACSL) so that the intelligent force control scheme's ability to apply forces to a range of non-rigid environments could be investigated. The use of a simple mechanical testbed (in preference to a complex multi-DOF robot model) allowed the control scheme's force application mechanism to be investigated without the need to consider:

- interaction within a robot structure that would result from forces generated by the contact.
- compliance modes in the robot structure.
- methods for establishing and partitioning the task aligned reference frame into orthogonal force and position controlled degrees of freedom.

The single DOF positioning device that was modelled is similar to the device used by Yabuta and Yamada [1990] to test the capabilities of their neuro-force control scheme. A block diagram of the single DOF device is shown in figure 4.9.

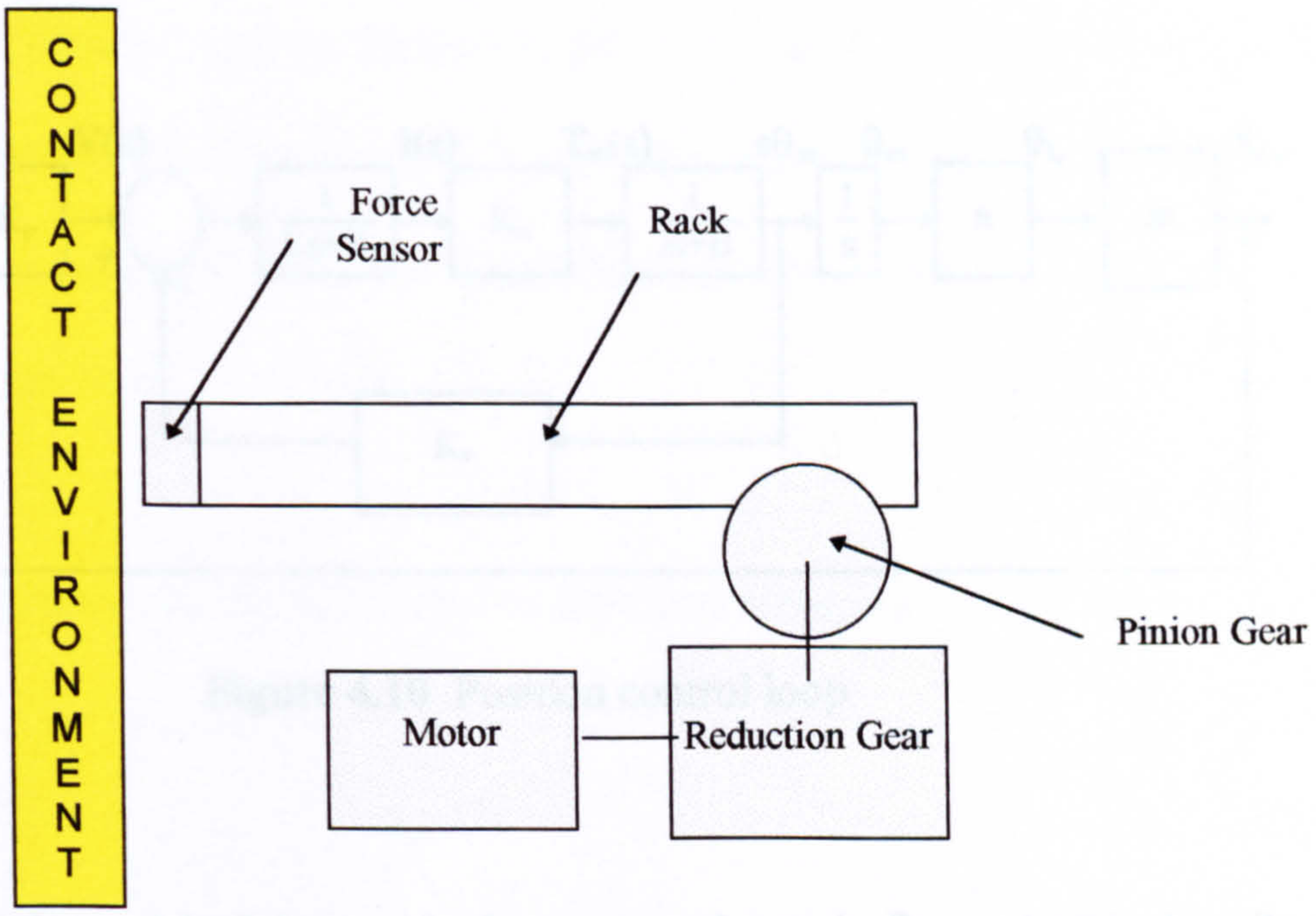


Figure 4.9 Single DOF position controlled mechanical testbed

The device shown is a single DOF servo controlled mechanical manipulator that comprises a rack and pinion which is coupled to a DC motor via a reduction gearbox. Torque supplied by the motor causes the pinion gear to rotate and this results in translation of the rack. The motor may be fitted with a shaft encoder to provide positional feedback and a force/torque sensor may be rigidly coupled to the end-effector so that the contact force between the end-effector and the environment could be measured directly.

A first principle model of the servo controlled manipulator was implemented in the Advanced Continuous Simulation Language (ACSL). The positioning control system was based on a position controller model proposed by Luh [1983] and the block diagram of the system is illustrated in figure 4.10.

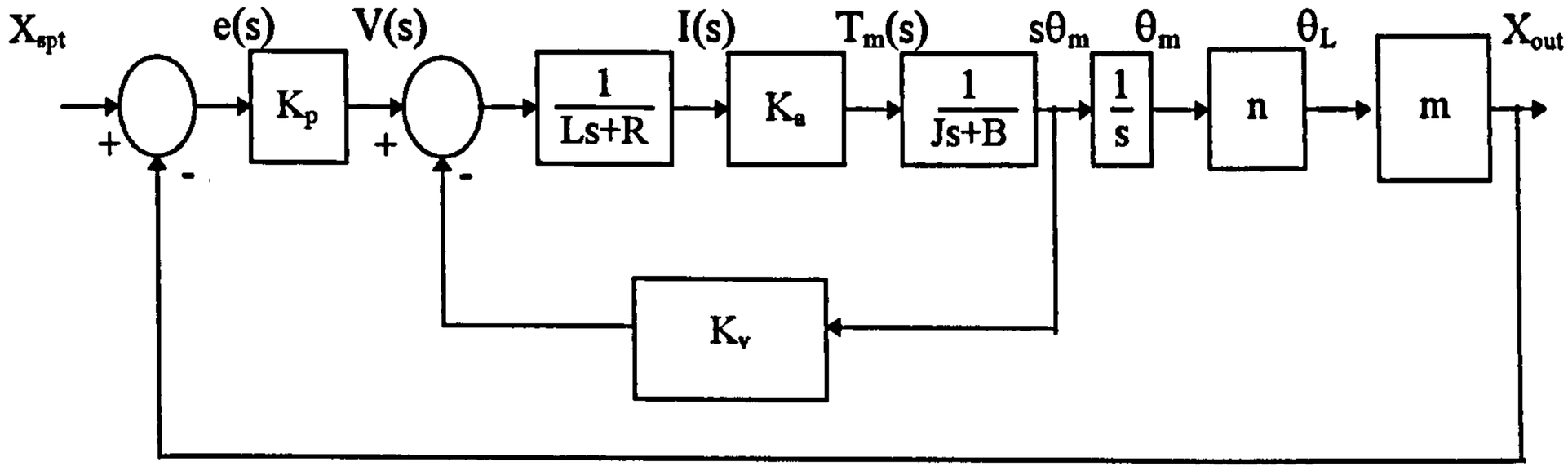


Figure 4.10 Position control loop

The first principle model of the positioning system shown in figure 4.10 is described by the equations:

$$e(s)=X_{spt}-X_{out} \tag{4.3}$$

$$V(s) = K_p\, e \tag{4.4}$$

$$I(s) = \frac{V(s)-K_v s\theta_m}{Ls+R} \tag{4.5}$$

$$T_m = K_a\, I(s) \tag{4.6}$$

$$T_m = (Js^2+Bs)\theta_m \tag{4.7}$$

$$\theta_L = n\theta_m \tag{4.8}$$

$$X_{out} = m\theta_L \tag{4.9}$$

X_{spt} is the positional setpoint, X_{out} is the measured position, e is the error signal, V is the controller output, K_p is the controller proportional gain, K_a is the current amplifier gain, L is the motor armature inductance, R is the motor armature resistance, T_m is the motor torque acting at the motor side, J is the effective system inertia at the motor shaft, B is the effective system friction at the motor shaft, θ_m is the motor shaft position, θ_L is the load shaft position, K_v is the tacho-feedback gain, n is the reduction gear ratio, m is a rotational motion to translational motion conversion factor, and s is the Laplacian operator.

The system's effective inertia J, referred to the motor shaft is expressed as:

$$J=J_m+n^2J_L \tag{4.10}$$

where J_m is the motor inertia and, n is the gearbox ratio ($n<1$) and J_L is the combined inertia of the manipulator link and inertial loading that results from the contact.

The intelligent force control scheme was developed to make non-excessive contact with the environment (an upper force application bound of 15N was adopted) and, as such, changes in the systems effective inertia as a result of inertial loading will be negligible. Thus inertial loading resulting from contact with the environment was not considered. Effects of motor inductance and motor back EMF were also not considered.

The controller comprised a proportional gain and velocity feedback was chosen for system damping in preference to a controller derivative term. The positioning system's model parameters are listed in Appendix B. The controller's proportional gain, K_p and the tacho-feedback gain K_v were manually tuned such that a critically damped unit step response was obtained. Figure 4.11 shows the positioning system's transient response to a unit step input.

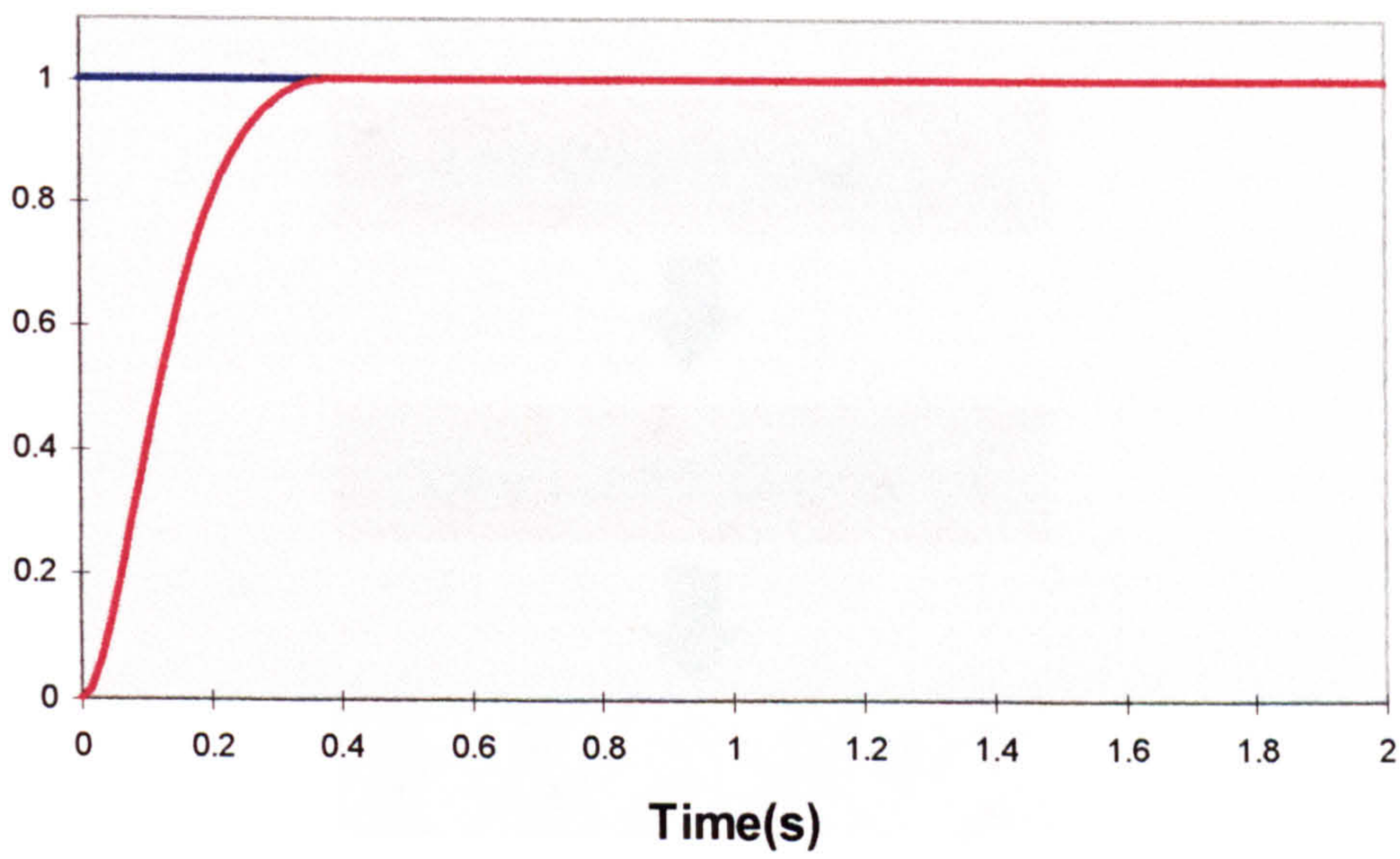


Figure 4.11 Position controlled manipulator transient response

The critically damped response shown has a settling time of 0.28 seconds (within 5% of the steady state value) and zero offset. The result was produced with a controller proportional gain of 160 and a tacho-feedback gain of 0.02.

The trained RBF network was incorporated into the ACSL simulation by adding Fortran subroutines to the ACSL code.

4.6 ROBOT CONTROL STATES

For many force control tasks, a robot needs to frequently make and break contact with the environment and the robot system must be capable of managing two states - free motion and contact (and transitions between the two) [Xu *et al.*, 1995]. However, the contact state can be further subdivided into a compliant control phase and a rigid contact phase. This is illustrated in figure 4.12 which shows the three robot control states.

Figure 4.12 Robot control states

Prior to contact, a robot end-effector is unconstrained, and can therefore be controlled using a positional control scheme. When contact does occur, the contact can be described as being either compliant or rigid, with each contact phase having distinctly different dynamic characteristics.

Rigid contact occurs when a robot end effector is in contact with a rigid material (i.e. steel, hardwood, etc.) *and* compliances in the robot/force sensor have been completely absorbed. Rigid contact can also occur with non-rigid contact

environments if the compliance in the environment is completely absorbed. In the rigid contact phase, the end-effector is totally constrained in the direction of the sensed force, and cannot move further into the contact surface.

Compliant contact occurs when the contact surface/object deforms upon contact (generally assumed to be local elastic deformation) and/or compliance's in the robot/force sensor have not been completely absorbed. When in the compliant contact phase, the end-effector is not constrained in the direction of the sensed force, and can therefore move further into the material, thereby increasing the contact force.

4.7 COMMAND LOGIC

In addition to the servo model control equations presented in section 4.5.2, the simulation required command logic to represent transitions between the contact state and the free motion state (and vice versa). The command logic that was used to model the contact state transitions comprised an if-then statement which monitored the force sensor reading. A reading that was below the force sensor threshold placed the control system in the free motion control state. When in this state, the system behaves as a positional control scheme and a switch placed on the ANN output was opened, thereby setting the ANN output to zero. When contact was sensed (i.e.when the force sensor reading was greater than a threshold) the switch placed on the ANN output was closed and the intelligent force control scheme was activated. This methodology was proposed and practically adopted by Hyde and Cutkosky [1993].

Problems of 'jumps' in the actuator input as a result of switching between states were highlighted but the use of a low level command filter was recommended for smoothing out the transients. Thus the force control scheme operation can be summarised as follows:

- prior to contact, the end-effector is moving in free space and the neural network output is set to zero (i.e. the ANN is inactive).
- when the end-effector makes contact with a surface, a non-zero contact force is sensed and the neural network is activated. For contact with non-rigid environments, the end-effector is now in contact but it is not constrained in the direction of the sensed force, and as such it is free to move further into the environment. When contact is sensed, the positional setpoint is set equal to the present sensed position and position of the end-effector is autonomously adjusted by the network output such that the desired force is applied to the environment.

4.8 SUMMARY

This chapter presented the reasoning behind the development of the intelligent force control scheme. The ANN input/output structure and the novel ANN training/test data generation procedure were also introduced. The contact model that was used to represent interaction with compliant environments was described and the method by which the ANN training data was extracted from the parameterised contact model was introduced. Methodologies that were used for training the intelligent force control scheme's ANN were also highlighted.

An ACSL simulation model of a single DOF servo controlled mechanical manipulator was developed from first principles and the method by which the trained ANN was incorporated into the positional control scheme was described. The main features of the RBF network training package that was developed for training and testing the intelligent force control scheme's ANN were presented. The need to include command logic within the simulation model was also highlighted.

CHAPTER 5

CONTACT WITH A SINGLE NON-RIGID ENVIRONMENT

5.1 INTRODUCTION

This chapter describes an investigation of the use of the intelligent force control scheme developed in Chapter 4 to apply forces to a single non-rigid environment. The control scheme uses an Artificial Neural Network (ANN) to increment the position of a mechanical manipulator's end-effector relative to its contact environment such that a desired force is applied to the environment. The positional increments made by the ANN were based on depth and force measurements taken at the point of contact, and the ANN can be said to issue a correct action (by issuing a command that will adjust the end-effector's position) in response to the sensed environment.

The ANN architecture that was investigated was the Radial Basis Function (RBF) network and RBF training methodologies that were adopted for the investigation are presented. Once trained, model selection and validation tests (Mean Square Error tests and, in some instances, Akaikes Final Prediction Error tests) were used to give empirical measures of network performance. The ANN model was then tested for its ability to perform accurate estimations across a force application range of 1N to 15N with noise free data and data that had noise added to the force and depth measurements.

The investigation was centered around the development of an ANN model that could accurately perform a 'single environment' input-output mapping. Several factors were considered when developing the ANN including the selection of a suitable network topology and training methodologies. The effects that the following training considerations had on network performance were investigated during the development of the intelligent force control scheme's ANN:

- RBF centre placement.
- RBF centre spread.
- the use of k-means clustering.
- the number of training epochs.
- the use of a network bias node.
- the number of hidden layer nodes.

Once trained, the ANN model was incorporated into a simulation of a single degree of freedom (DOF) mechanical manipulator that was developed in the Advanced Continuous Simulation Language (ACSL), as described in Chapter 4. The addition of the ANN allowed a manipulator that could previously perform only positioning tasks to autonomously apply forces to its environment. The intelligent force control scheme's ability to apply forces to the environment was investigated by simulation. Contact with several environments, each with differing degrees of rigidity, was investigated. The effect of noise on control system performance was also investigated.

5.2 NETWORK TRAINING DATA GENERATION FOR CONTACT WITH A SINGLE NON-RIGID ENVIRONMENT

5.2.1 MODELLING THE NON-RIGID ENVIRONMENT

Initially, contact with a single simulated environment was investigated and a contact model and contact model parameters were required to represent the environment. The contact model that was adopted was introduced in Chapter 4 and is shown in equation 5.1.

$$F=k_1x_{env} + k_3x_{env}{}^{a} \tag{5.1}$$

where F is the contact force, k_1 is the contact environment spring constant, k_3 is the contact environment spring hardening coefficient, x_{env} is the depth into the environment, and a is an exponent included to represent spring hardening in the contact environment.

The model represents a single compliance mode in the contact environment. Compliance modes within the robot/force sensor were not considered (i.e. the robot and sensor were assumed to be rigid). The model parameters that were chosen to represent the environment under investigation are shown in table 5.1.

	k_1 (N/m)	k_3 (kN/m^2)	a
Environment 1	100	100	2

Table 5.1 Model parameters for environment 1

The contact model shown in equation 5.1 was parameterised with the model parameters shown in table 5.1 and the force experienced at different depths into the environment was tabulated. Additional simulated environments were generated by varying the contact model parameters so that the repeatability of the ANN training methodologies could be tested and validated with environments that had differing degrees of rigidity.

Twelve simulated environments were generated for the investigation. Six environments were used as training environments and the remaining six environments were used as validation environments. The contact model parameters that were used to generate the twelve data sets are shown in Appendix A. The characteristics of the simulated hardness profiles (a plot of the contact force at different depths into the environment) produced were similar to practically measured hardness profiles reported by Caldwell and Gosney [1993].

Hardness profiles for the simulated environment under investigation (environment 1) and a second simulated environment (environment 2) are shown in figure 5.1. The contact model parameters for environment 2, which can be seen to be more rigid than environment 1, are shown in Appendix A. Results of tests with networks trained to respond to environment 2 have been included to illustrate the repeatability of the training methodologies with a different non-rigid environment. Although results are presented for contact with two environments, the training methodologies were investigated with each of the six training environments.

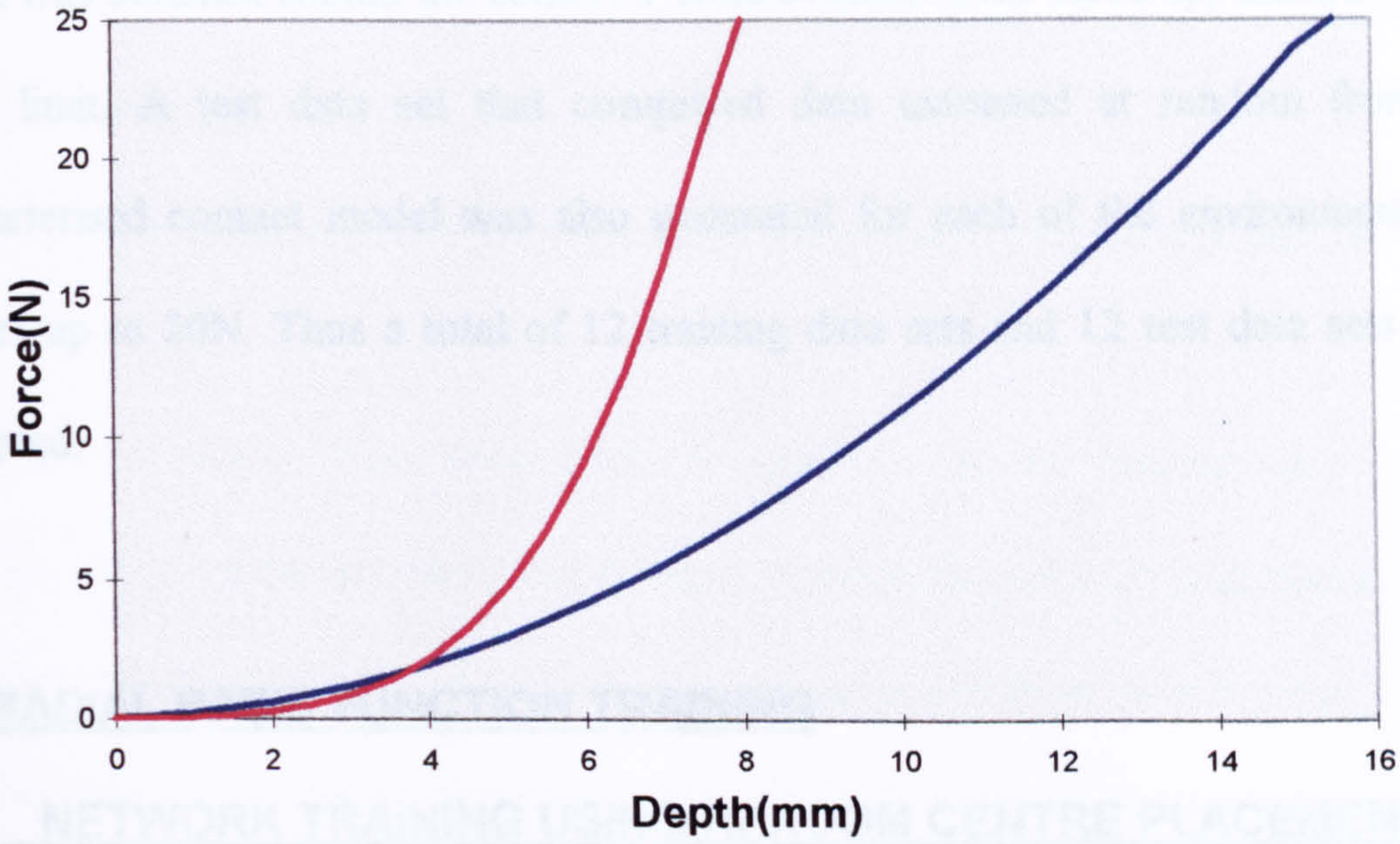


Figure 5.1 Simulated hardness profiles ■ Environment 1 ■ Environment 2

5.2.2 EXTRACTING NEURAL NETWORK TRAINING DATA FROM THE PARAMETERISED CONTACT MODEL

In order that an ANN could be trained, it was necessary to generate representative network training data. Additional data sets were required so that the ANN's performance could be tested and validated with unseen data that was not in the training data set. The RBF network architecture investigated is a feedforward network trained using supervised learning, and as such the training data set must comprise a target output vector for each input vector.

The ANN training data was extracted from the parameterised contact model using the novel data extraction procedure introduced in Chapter 4. Although the force application ranges upper limit was 15N, the training data set comprised data for contact up to 20N. The additional data was included to ensure that a bounded ANN

output was obtained should the control system overshoot the force application ranges upper limit. A test data set that comprised data extracted at random from the parameterised contact model was also generated for each of the environments for contact up to 20N. Thus a total of 12 training data sets and 12 test data sets were generated.

5.3 RADIAL BASIS FUNCTION TRAINING

5.3.1 NETWORK TRAINING USING RANDOM CENTRE PLACEMENT

A factor that can significantly influence an RBF networks estimating performance is the placement of the hidden layer node centres [Dacosta *et al.*, 1997]. Researchers have reported that the RBF centres should sample the network training data set input domain [Moody and Darken, 1989] but at present, there is no clear method of optimally placing the RBF centres. As such, the effect of placing the hidden layer node centres was investigated. Several RBF centre placement/optimisation techniques have been proposed, including random placement and k-means clustering, two techniques that were investigated during the development of the intelligent force control scheme's ANN.

Initially, an RBF network with six hidden layer nodes was used. The network topology thus comprised 3 input nodes, 6 hidden layer nodes, and a single output node. A bias node was included in the network. Node centres were randomly placed within the training data set input domain and an RBF network was trained using 2 iterations of the training data set extracted from environment 1. Two networks were retained:

- the network that yielded the lowest Mean Square Error (MSE) when tested with the training data set.
- the network that yielded the lowest MSE when tested with the test data set.

RBF centre vectors were then randomly placed over the training data set input domain for 10 centre iterations and for each iteration the network was tested with the original training data set and the test data set extracted from environment 1. Again, the two best networks were retained. The network training/selection procedure was then repeated for 500 centre randomisations. The training procedure was repeated with data extracted from other simulated environments so that the experiments repeatability could be assessed. Network performance was assessed by considering the Mean Square Error (MSE) between the network estimations and the target outputs (also known as the `loss function`). The loss function is expressed as:

$$mse = \frac{1}{N}\sum_{t=1}^{N}\left[y(t) - \hat{y}(t)\right]^2 \tag{5.2}$$

where y(t) is the target output, $\hat{y}(t)$ is the network estimation, and N is number of data vectors in the test data set.

Figure 5.2 shows the loss functions obtained for tests with networks that gave the best performance with the test data set extracted from environment 1. Results of tests with networks trained and tested with data extracted from environment 2 have been included for comparison.

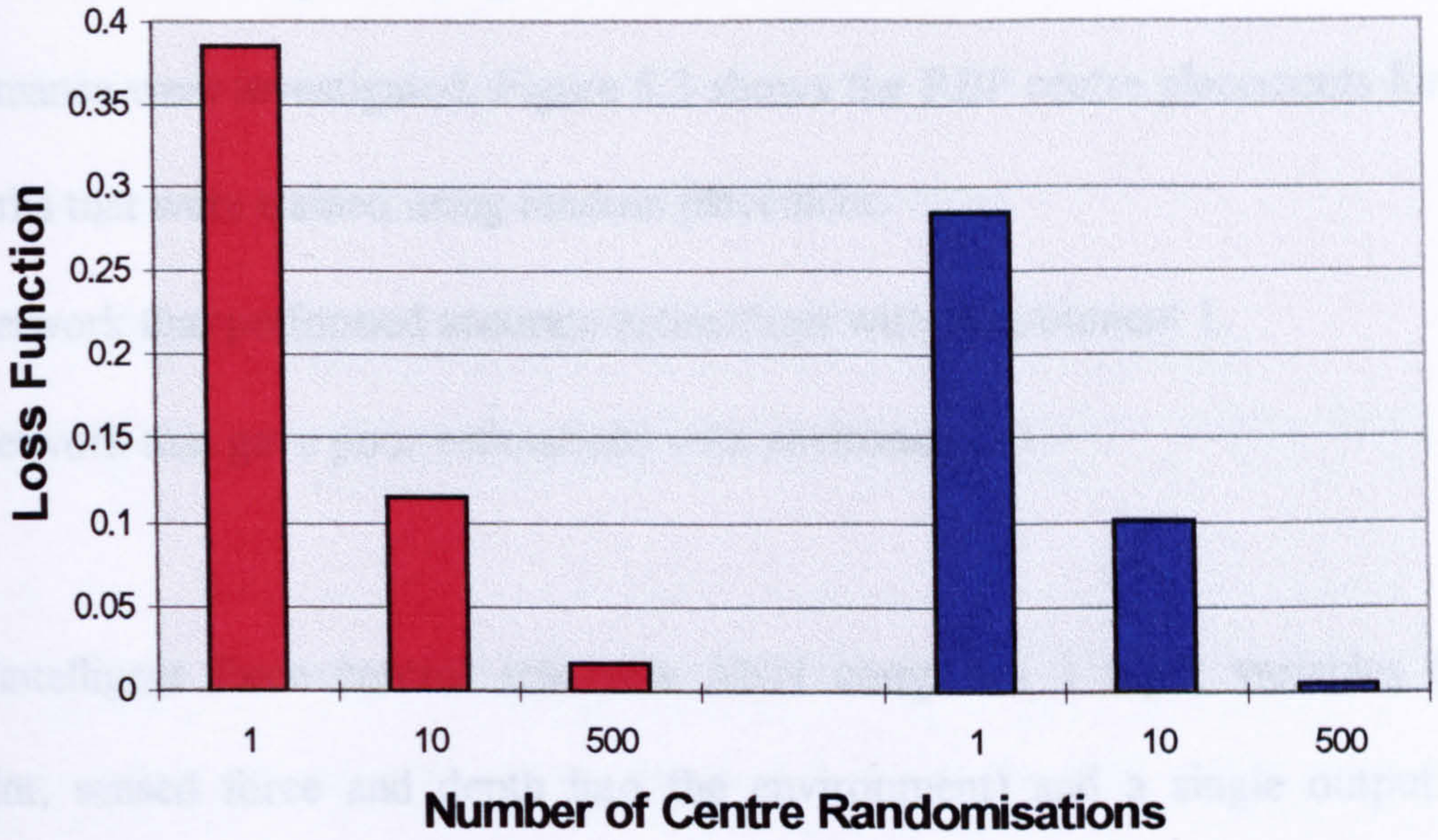


Figure 5.2 Loss functions for networks trained using differing numbers of centre randomisations. ■ Results for networks trained with data extracted from environment 1. ■ Results for networks trained with data extracted from environment 2.

Figure 5.2 shows that network performance was poor when a single random centre placement was used. However, performance was found to improve considerably (characterised by a decrease in the loss function) when a repeated random placement methodology was adopted with a large number of centre randomisations. A lower loss function was obtained for the same network topology with environment 2 which may be an indication that the input-output mapping for environment 2 is less complex than that of environment 1. Loss function results for tests with networks trained with data extracted from the remaining simulated environments were found to be in accordance with the results presented for environment 1. Thus the performance of networks trained to perform a 'single environment' input-output mapping was found to be dependent on the placement of the RBF centres.

Characteristics of the positioning of the RBF centres that may have affected network performance were investigated. Figure 5.3 shows the RBF centre placements for two networks that were trained using random placement:

- a network that performed accurate estimations with environment 1.
- a network that gave poor estimations with environment 1.

The intelligent force control scheme's ANN comprises 3 input variables (force setpoint, sensed force and depth into the environment) and a single output. The dimensions of each RBF centre vector are the same as the network training data set input domain and it is thus difficult to illustrate the centre positions in three dimensional space. As such, results are presented in two dimensions with the third dimension variable (force setpoint) kept constant. The network topology comprised 6 hidden layer nodes and thus 6 RBF centres.

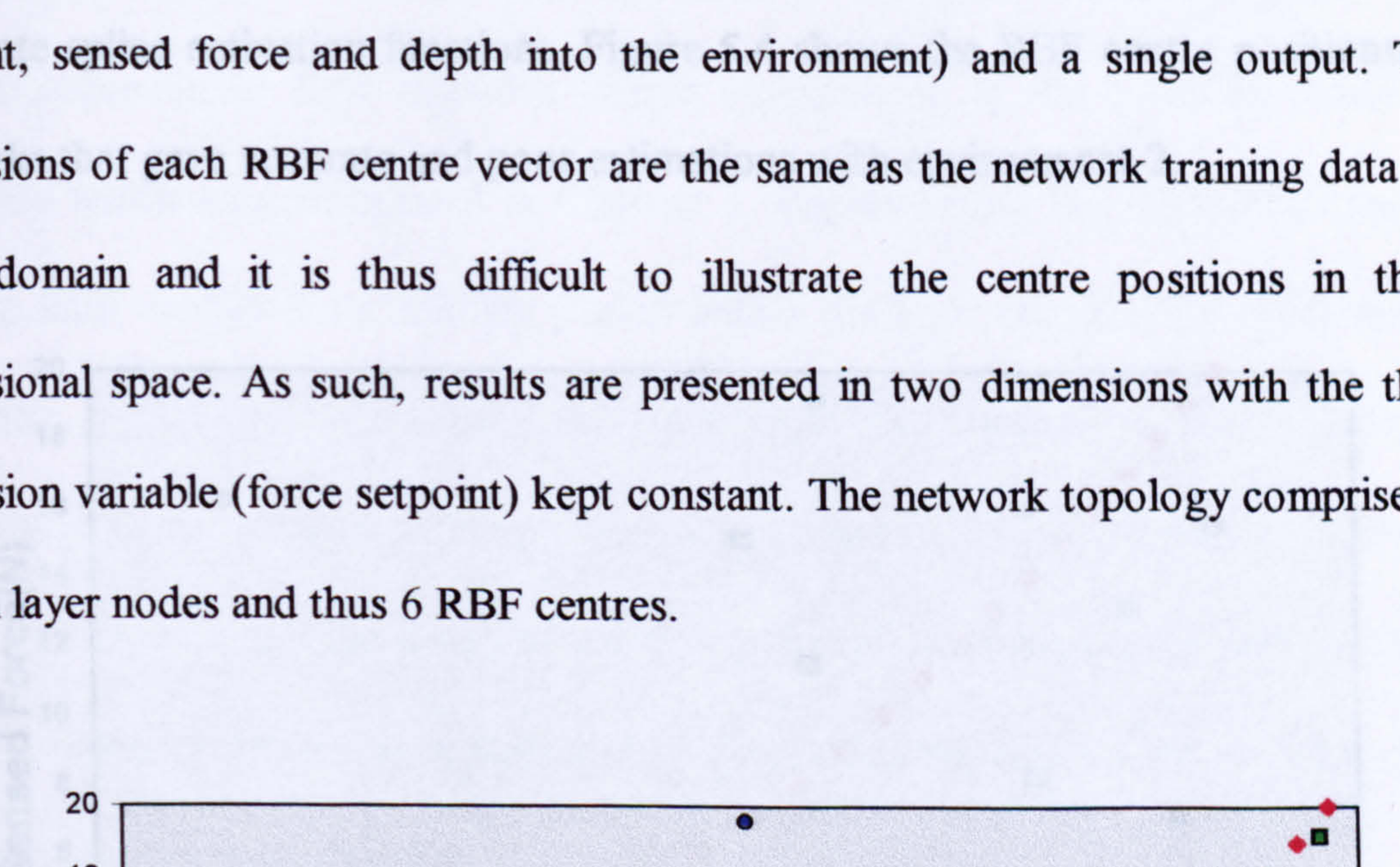


Figure 5.3 RBF centre placements for accurate and poor estimating networks. ◆Training data points. ■ Centre positions for networks that gave poor estimations. ● Centre positions for networks that gave accurate estimations.

Figure 5.3 illustrates that centre placement characteristics that may lead to a degradation in network performance are:

- centres placed in close proximity to other RBF centres.
- centres placed in close proximity to the training data points.
- centres placements that are not distributed about the complete training data range.

However, these characteristics may only be valid for RBF networks that comprise thin plate spline activation functions. Figure 5.4 shows the RBF centre positions for networks that gave accurate and poor estimations with environment 2.

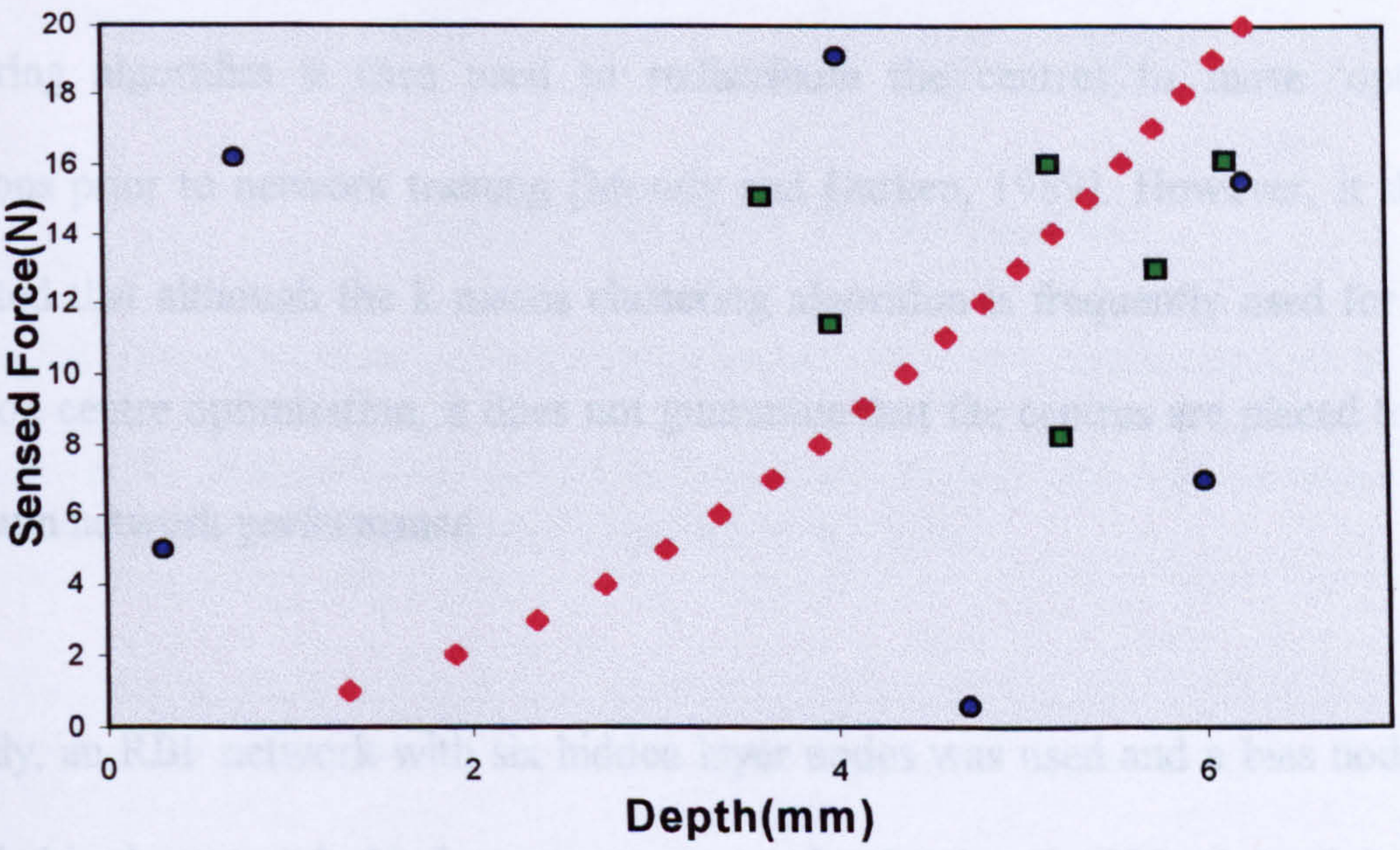


Figure 5.4 RBF centre placements for accurate and poor estimating networks. ◆Training data points. ● Centre positions for networks that gave accurate estimations. ■ Centre positions for networks that gave poor estimations.

The result illustrates that networks capable of performing accurate estimations with data extracted from environment 2 had the same centre placement characteristics as networks that performed well with environment 1. The RBF centres were not in close

proximity to the training data points or other RBF centres. Additionally, the centres were distributed about the complete training data range.

5.3.2 NETWORK TRAINING USING K-MEANS CLUSTERING

Although repeated random placement of the RBF centres was found to yield networks that were capable of accurately performing a single environment input-output mapping, the use of a centre optimisation algorithm was investigated. One popular approach to RBF network centre optimisation is the k-means clustering algorithm which was introduced in Chapter 3. Implementing the algorithm requires that the RBF centres are randomly placed within the input domain and the k-means clustering algorithm is then used to redistribute the centres to more 'optimal' positions prior to network training [Moody and Darken, 1989]. However, it should be noted that although the k-means clustering algorithm is frequently used for basis function centre optimisation, it does not guarantee that the centres are placed to give optimum network performance.

Initially, an RBF network with six hidden layer nodes was used and a bias node was included in the network. Node centres were randomly placed within the training data set input domain and an RBF network was trained using 2 iterations of the training data set extracted from environment 1. The network training procedure adopted during the random placement investigation was repeated, but the k-means clustering algorithm was used to redistribute the RBF centres prior to network training.

Figure 5.5 shows the loss functions for networks trained using k-means clustering with various numbers of centre randomisations. The results presented are for tests with networks that gave the best performance with the test data sets extracted from environments 1 and 2.

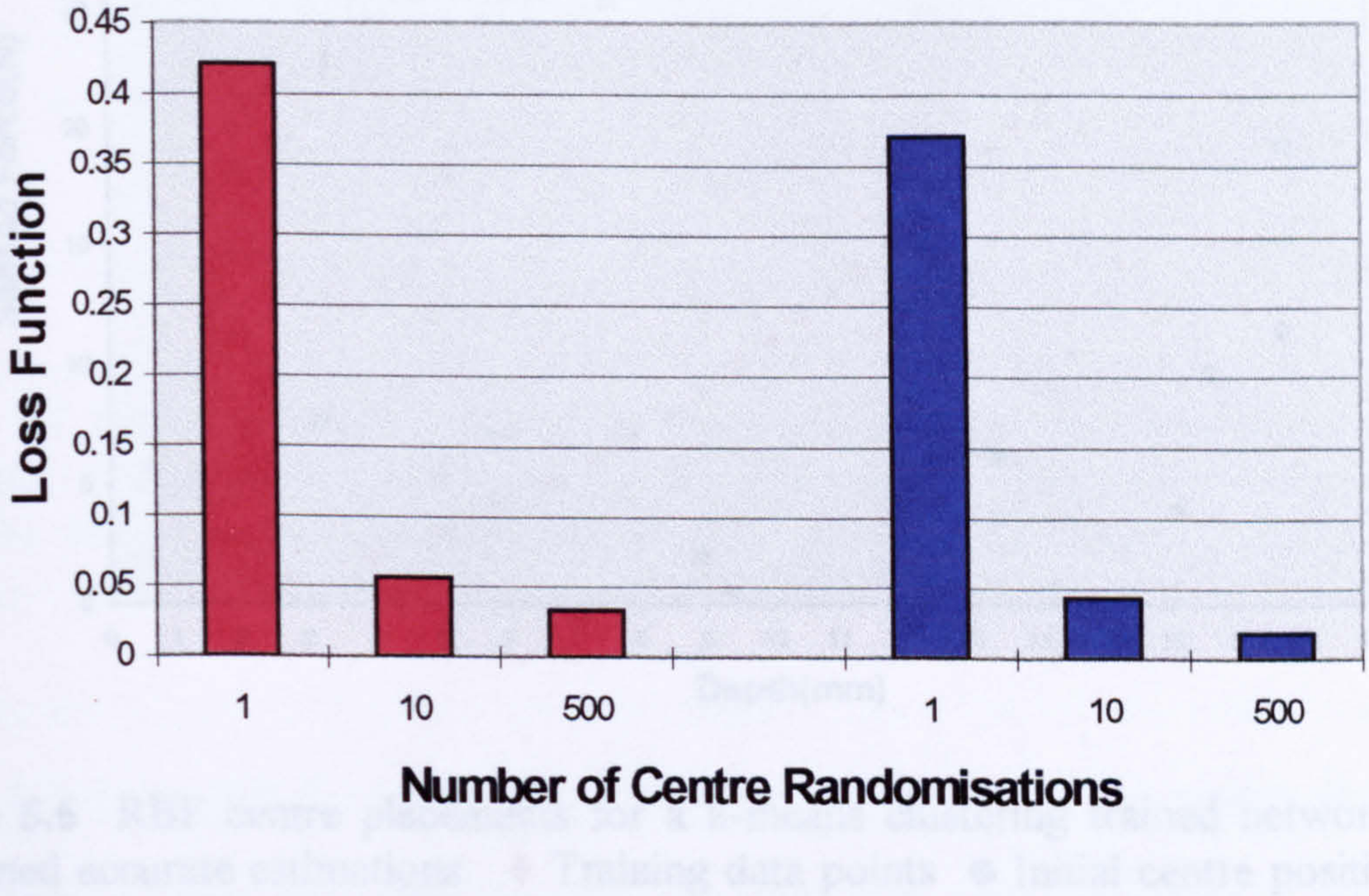


Figure 5.5 Loss functions for variation in number of centre randomisations using k-means clustering. ■ Results for networks trained with data extracted from environment 1. ■ Results for networks trained and tested with data extracted from environment 2. Networks comprised 6 hidden layer nodes.

Figure 5.5 shows that network performance was poor when the k-means clustering algorithm was used with a single centre randomisation. However, performance improved considerably (characterised by a decrease in the loss function) when the repeated random centre placement training procedure were used with several hundred centre randomisations. Thus the performance of the k-means clustering optimisation was found to be dependent on the initial centre placements.

Figure 5.6 shows the k-means clustering algorithm redistributing RBF centres which were initialised over twice the training data set input domain. The network comprised 6 hidden layer nodes and was trained using data extracted from environment 1.

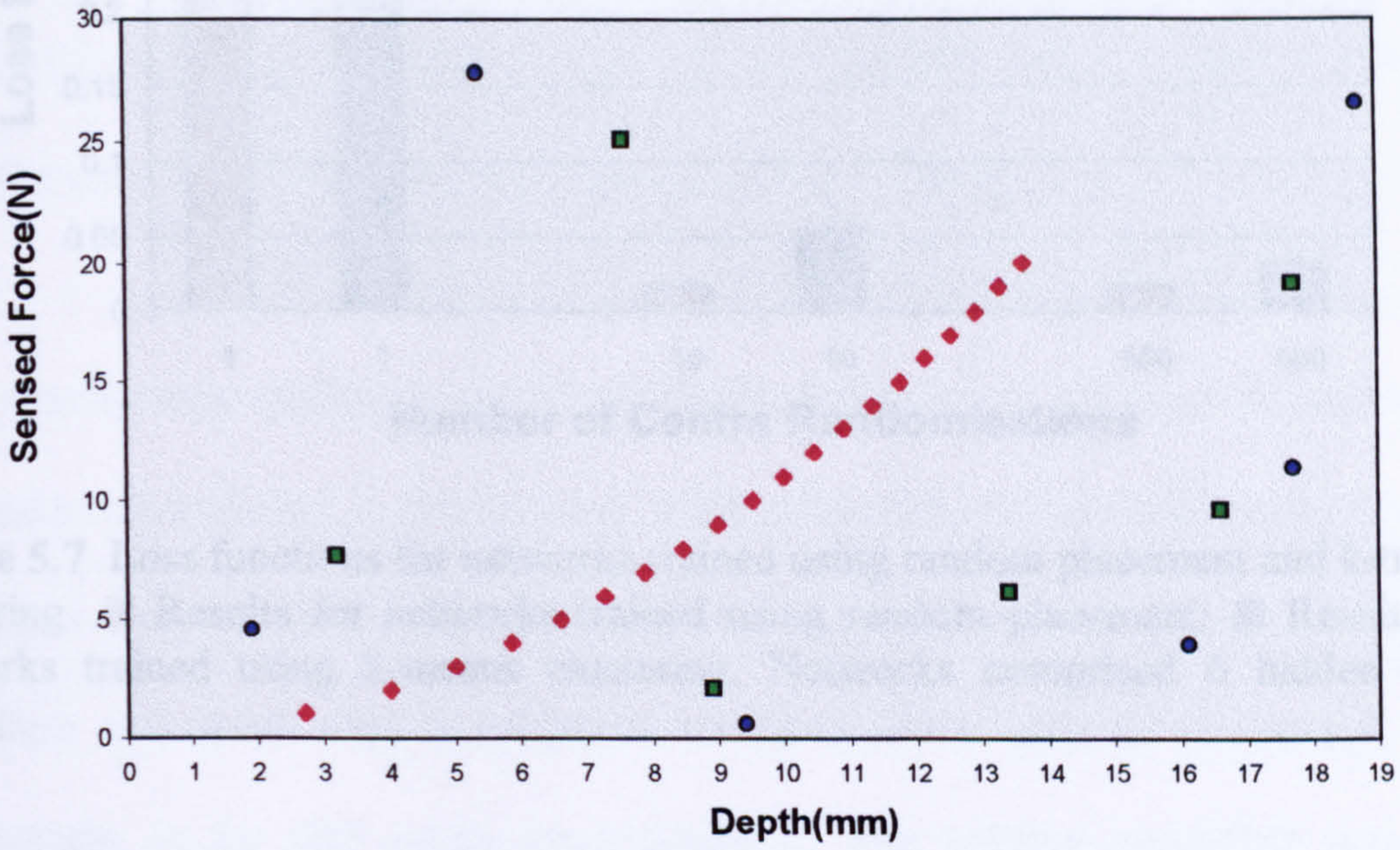


Figure 5.6 RBF centre placements for a k-means clustering trained network that performed accurate estimations. ◆ Training data points ● Initial centre positions ■ Centre positions after k-means clustering.

The result shows that the k-means clustering algorithm repositioned the centres to locations closer to the training data points. The characteristics of the redistributed centre positions were similar to the characteristics described for random placement trained networks with good estimating performance.

Figure 5.7 shows a comparison of the best random placement trained networks and the best k-means clustering trained networks. The loss function results presented are for network testing with the test data extracted from environment 1.

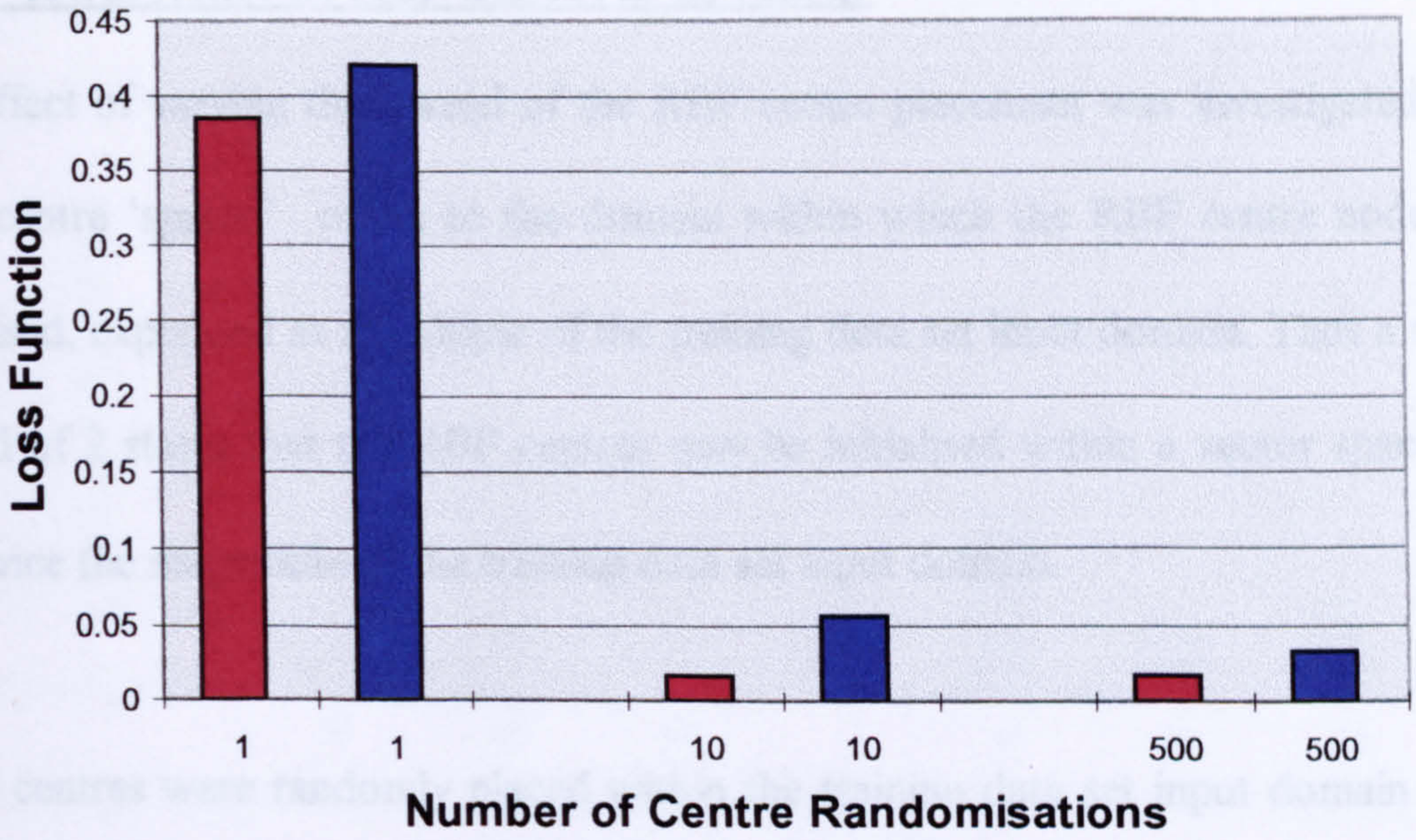


Figure 5.7 Loss functions for networks trained using random placement and k-means clustering ■ Results for networks trained using random placement. ■ Results for networks trained using k-means clustering. Networks comprised 6 hidden layer nodes.

Figure 5.7 shows that the performance of networks trained using k-means clustering was inferior to networks with the same topology trained using random placement. Results of tests with other simulated environments were found to be in accordance with the results presented for environment 1. Thus k-means clustering was found to be detrimental to network performance for single environment trained networks.

The deterioration in performance found with the use of the k-means clustering algorithm may be attributed to the clustering algorithm moving the centres towards the training data points. Centres placed in close proximity to the training data points was highlighted as a possible contributing factor that may lead to a deterioration in the performance of networks that comprise thin plate spline activation functions.

5.3.3 VARIATION IN RBF CENTRE SPREAD

The effect of varying the spread of the RBF centre placement was investigated. The RBF centre 'spread' refers to the domain within which the RBF centre nodes are initialised, expressed as a multiple of the training data set input domain. Thus a centre spread of 2 states that the RBF centres may be initialised within a vector space that has twice the magnitude of the training data set input domain.

Node centres were randomly placed within the training data set input domain and a bias node was included in the network. RBF networks with six hidden layer nodes were used throughout the investigation. The repeated random placement training procedure was used with 2 iterations of the training data set (extracted from environment 1) for 500 centre randomisations. The training procedure was then repeated with the RBF centres being placed within higher multiples of the input domain. Network performance was assessed by considering the loss function between the network estimation and the target output, for the complete test data set.

Figure 5.8 shows the loss functions for networks trained using centre spreads that were multiples of the training data set input domain. Loss function results for networks trained with data extracted from environment 2 have been included to illustrate the experiments repeatability.

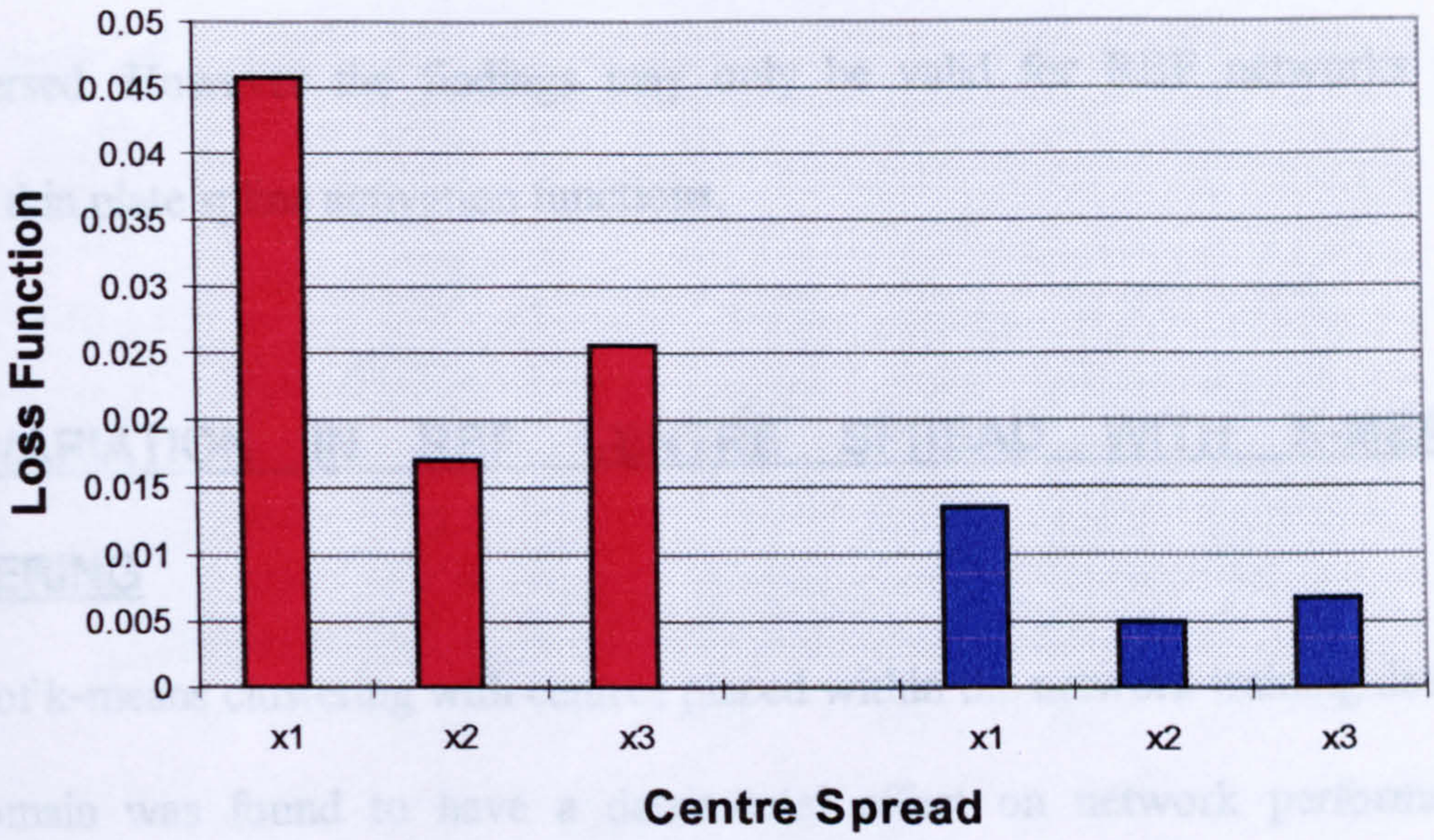


Figure 5.8 Loss functions for networks trained using random placement with various centre spreads. Centre spreads expressed as multiples of the training data set input domain. ■ Results for networks trained using data extracted from environment 1. ■ Results for networks trained using data extracted from environment 2.

Figure 5.8 shows network performance to be dependent on the centre spread. The performance of networks trained with centres placed within the input domain was inferior to the performance of networks that had centres randomised within wider domains. However, performance was found to deteriorate when the centres were placed beyond twice the input domain. Results of tests performed on networks trained with data extracted from other environments were found to be in accordance with the results presented for environment 1.

The result may again be explained by considering the aforementioned centre placement characteristics that were found to yield networks that performed well. Centre randomisations within wider domains reduces the probability of centres being placed in close proximity to the training data points and other RBF centres. The deterioration in network performance when centres were randomised within domains

higher than twice the training data set input domain suggests that the centres can be too dispersed. However the findings may only be valid for RBF networks that comprise thin plate spline activation functions.

5.3.4 VARIATION IN RBF CENTRE SPREAD WITH K-MEANS CLUSTERING

The use of k-means clustering with centres placed within the network training data set input domain was found to have a detrimental effect on network performance. However, increasing the RBF centre spread to twice the training data set input domain was found to improve the performance of random placement trained networks. As such, the effect of varying the spread of the RBF centre placement with k-means clustering optimisation was investigated.

A network topology that comprised 6 hidden layer nodes was used and a bias node was included in the network. RBF centres were randomly placed over the training data set input domain and the networks were trained using two epochs of the training data set for 500 centre randomisations. K-means clustering was used to redistribute the RBF centres prior to network training. The network was tested with the original training data set and a test data set and the two best performing networks were retained. The training procedure was then repeated with the RBF centres placed over higher multiples of the input domain.

Figure 5.9 shows the loss functions for networks trained using k-means clustering with centre placed within multiples of training data set input domain.

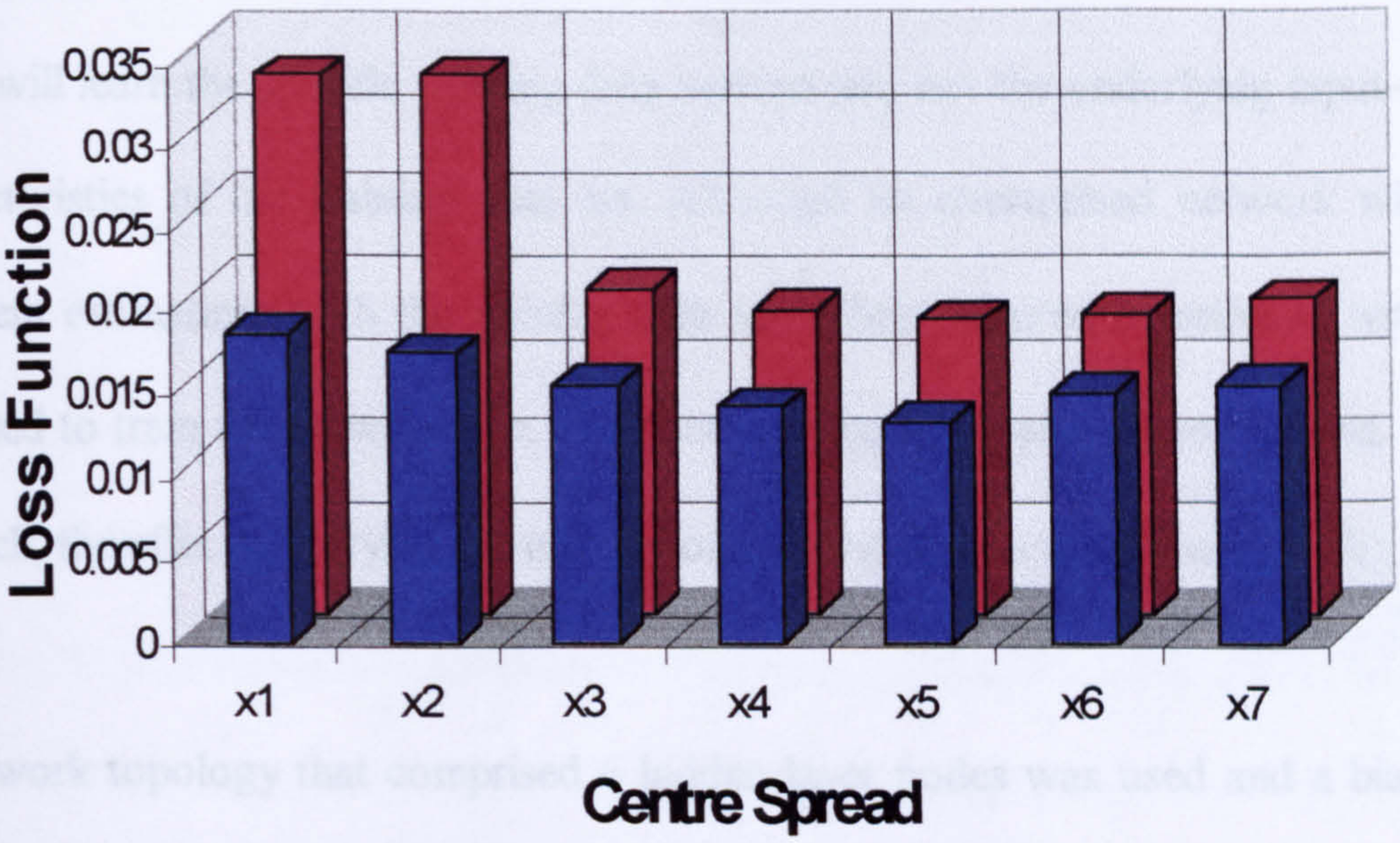


Figure 5.9 Loss functions for networks trained using k-means clustering with various centre spreads. Centre spreads expressed as multiples of the training data set input domain. ■ Results for networks trained using data extracted from environment 1. ■ Results for networks trained using data extracted from environment 2.

The results show network performance of k-means clustering trained networks to be dependent on the RBF centre spread. Performance was found to improve when the centres were placed within domains that were higher multiples of the input domain. However performance deteriorated when centre spreads greater than 5 times the training data set input domain were used. Results of tests with networks trained with data extracted from other environments were found to be in accordance with the results presented for environment 1.

5.3.5 VARIATION IN THE NUMBER OF TRAINING EPOCHS

Training a neural network requires that the training data set is presented to the network for at least one complete pass of the data set (an epoch). During the training phase, the ANN weights are adjusted in an attempt to map the input data to the

corresponding output data. However, if a neural network is trained for too long, the ANN will learn the specific training data vectors and not the underlying input-output characteristics of the training data set. Although an overtrained network will give excellent estimations with the training data set, when required to estimate with data not used to train the network (i.e. test data), estimations can be poor [Evans, 1994]. As such, the effect of varying the number of training epochs was investigated.

A network topology that comprised 6 hidden layer nodes was used and a bias node was included in the network. Node centres were randomly placed over twice the training data set input domain and an RBF network was trained using 1 iteration of the training data set. This was repeated for 500 centre iterations and for each iteration the network was tested with the original training data set and the test data set extracted from environment 1. Two networks were retained, the network that had the smallest Mean Squared Error (MSE) with the training data set and the network that had the smallest MSE with the test data set. The training procedure was then repeated for increased numbers of training epochs.

Figure 5.10 shows the loss functions obtained for tests with networks trained using different numbers of training epochs. The results presented are for tests with networks that gave the lowest loss functions with the test data set extracted from environment 1. These networks ensure that overtraining does not occur since they give accurate estimations with unseen data. Results for networks trained with data extracted from environment 2 are presented for comparison.

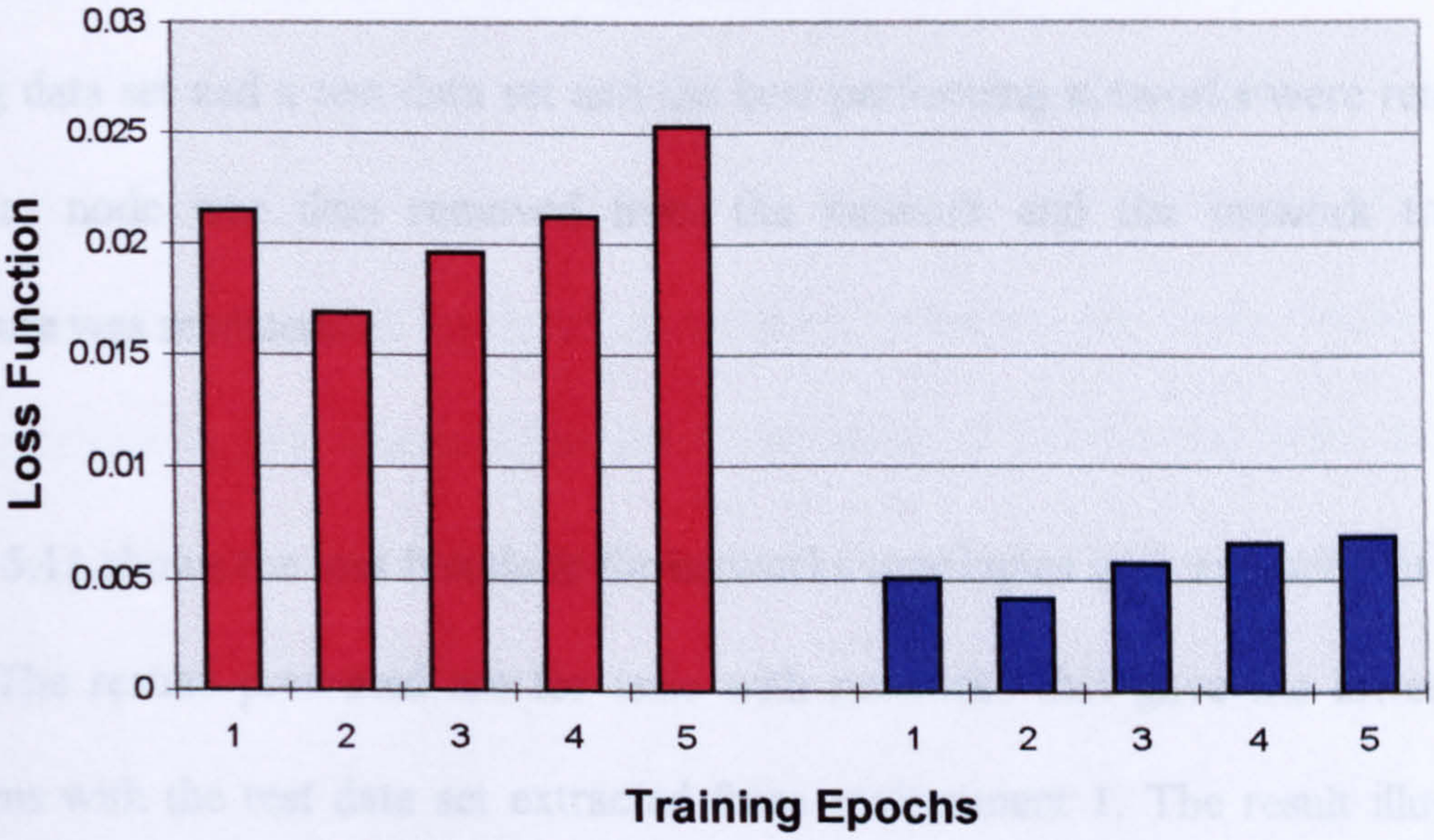


Figure 5.10 Loss functions for networks trained using random placement with varying numbers of training epochs. ■ Results for networks trained using data extracted from environment 1. ■ Results for networks trained using data extracted from environment 2. Networks comprised 6 hidden layer nodes.

Figure 5.10 illustrates that the performance of networks trained using a single pass of the training data set was found to be inferior to networks that were trained using 2 epochs of the training data set. However, performance was found to deteriorate when more than two training epochs were used to train the network, which is an indication that overtraining has occurred. Results with the other simulated environments were found to be in accordance with the results presented for environment 1.

5.3.6 THE EFFECT OF A BIAS NODE

The effect of including a bias node in the network topology was investigated. A network topology that comprised 6 hidden layer nodes was used and a bias node was included in the network. RBF centres were randomly placed over twice the training data set input domain and networks were trained using two epochs of the training

data set for 500 centre randomisations. The network was tested with the original training data set and a test data set and the best performing networks were retained. The bias node was then removed from the network and the network training procedure was repeated.

Figure 5.11 shows the loss functions for networks topologies with and without a bias node. The results presented are for tests with networks that gave the lowest loss functions with the test data set extracted from environment 1. The result illustrates that the inclusion of a bias node improved network performance. Results with the other simulated environments were found to be in accordance with results presented for environment 1.

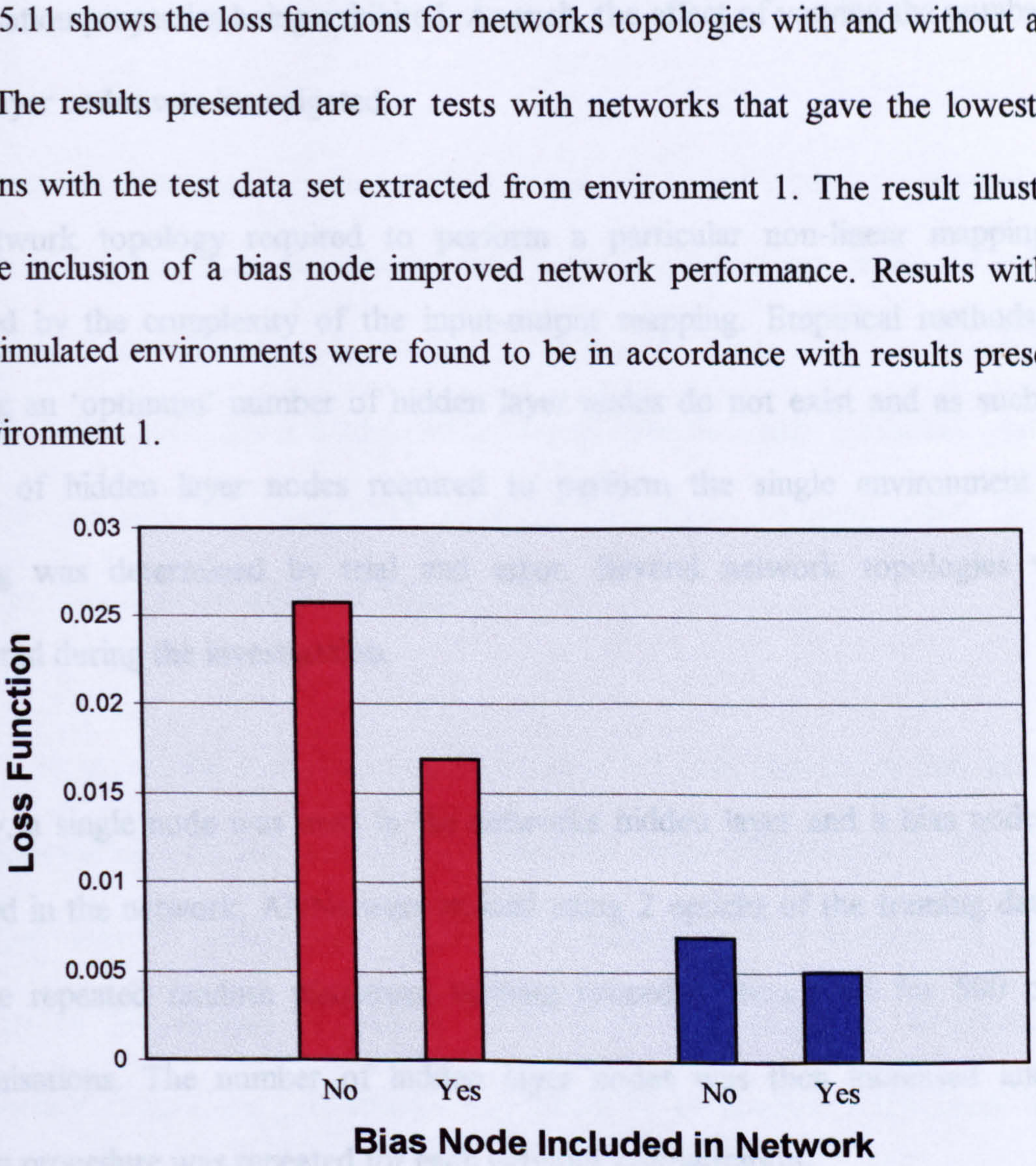


Figure 5.11 Loss functions for networks with and without bias nodes. ■ Results for networks trained using data extracted from environment 1. ■ Results for networks trained using data extracted from environment 2. Networks comprised 6 hidden layer nodes.

5.3.7 VARIATION IN THE NUMBER OF HIDDEN LAYER NODES

The number of hidden layer nodes included in a neural network can significantly affect the networks estimating performance. Too few hidden layer nodes limits the networks ability to model the required input-output mapping. Whereas, too many hidden layer nodes can result in over parameterisation of the network, which can result in poor generalisation properties being exhibited. As such, the effect of varying the number of hidden layer nodes was investigated.

The network topology required to perform a particular non-linear mapping is governed by the complexity of the input-output mapping. Empirical methods for selecting an 'optimum' number of hidden layer nodes do not exist and as such the number of hidden layer nodes required to perform the single environment I/O mapping was determined by trial and error. Several network topologies were considered during the investigation.

Initially, a single node was used in the networks hidden layer and a bias node was included in the network. ANNs were trained using 2 epochs of the training data set and the repeated random placement training procedure was used for 500 centre randomisations. The number of hidden layer nodes was then increased and the training procedure was repeated for each network configuration.

Network performance was assessed by considering the MSE between the network prediction and the target output and Akaikes Final Prediction Error (AFPE) [Akaike, 1974] was used as a additional performance measure. AFPE is expressed mathematically by equation 5.3:

$$AFPE = \left[\frac{N + N_w}{N - N_w}\right] LF \qquad (5.3)$$

where N_w is the number of adjustable model parameters (i.e. the network weights), N is the number of test data vectors, and LF is the loss function.

AFPE penalises over-parameterised models and is thus a useful measure of parsimony (i.e. network performance vs. network size). This is a significant factor for robotic applications, since minimising network size allows for faster real time processing. Although AFPE tests are a useful aid to model selection and validation, the test should be used with care since AFPE tests has been shown to overestimate the true parameter vector [Leontaritis and Billings, 1987].

Figure 5.12 shows the loss functions for various network topologies that gave the best performance with the test data set extracted from environment 1. The results show that network performance improved (characterised by a decrease in the loss function) as the number of hidden layer nodes was increased. However, the improvement in network performance was found to be minimal when more than 6 hidden layer nodes were used. Figure 5.13 shows AFPE for the same networks. The AFPE results show that the network topology that performed best relative to its size with data extracted from environment 1 comprised 6 hidden layer nodes. Figures 5.14 and 5.15 show loss function and AFPE results for networks trained with data extracted from environment 2 respectively.

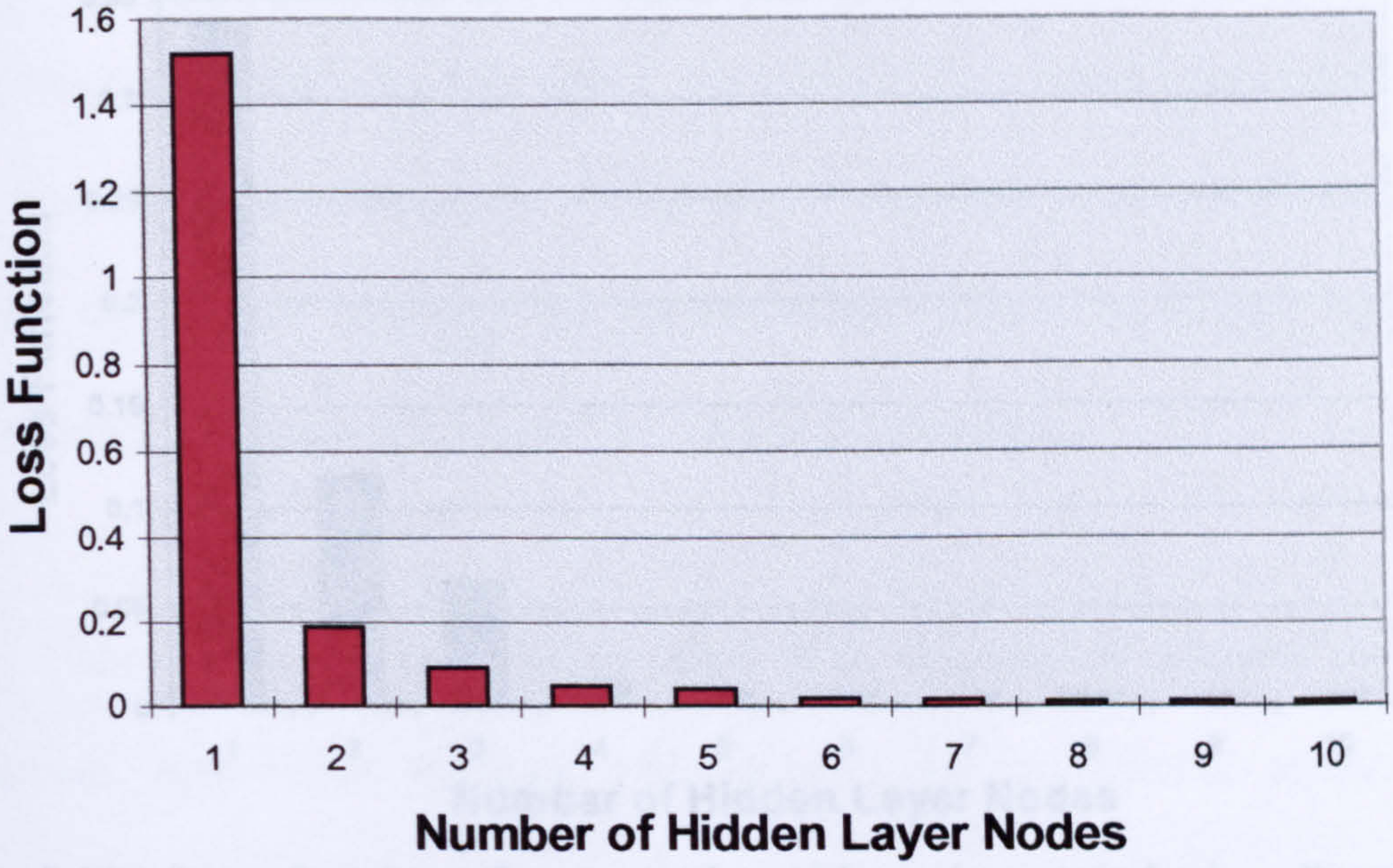


Figure 5.12 Loss functions for networks with various topologies. Results for networks trained using data extracted from environment 1. Networks trained using random placement.

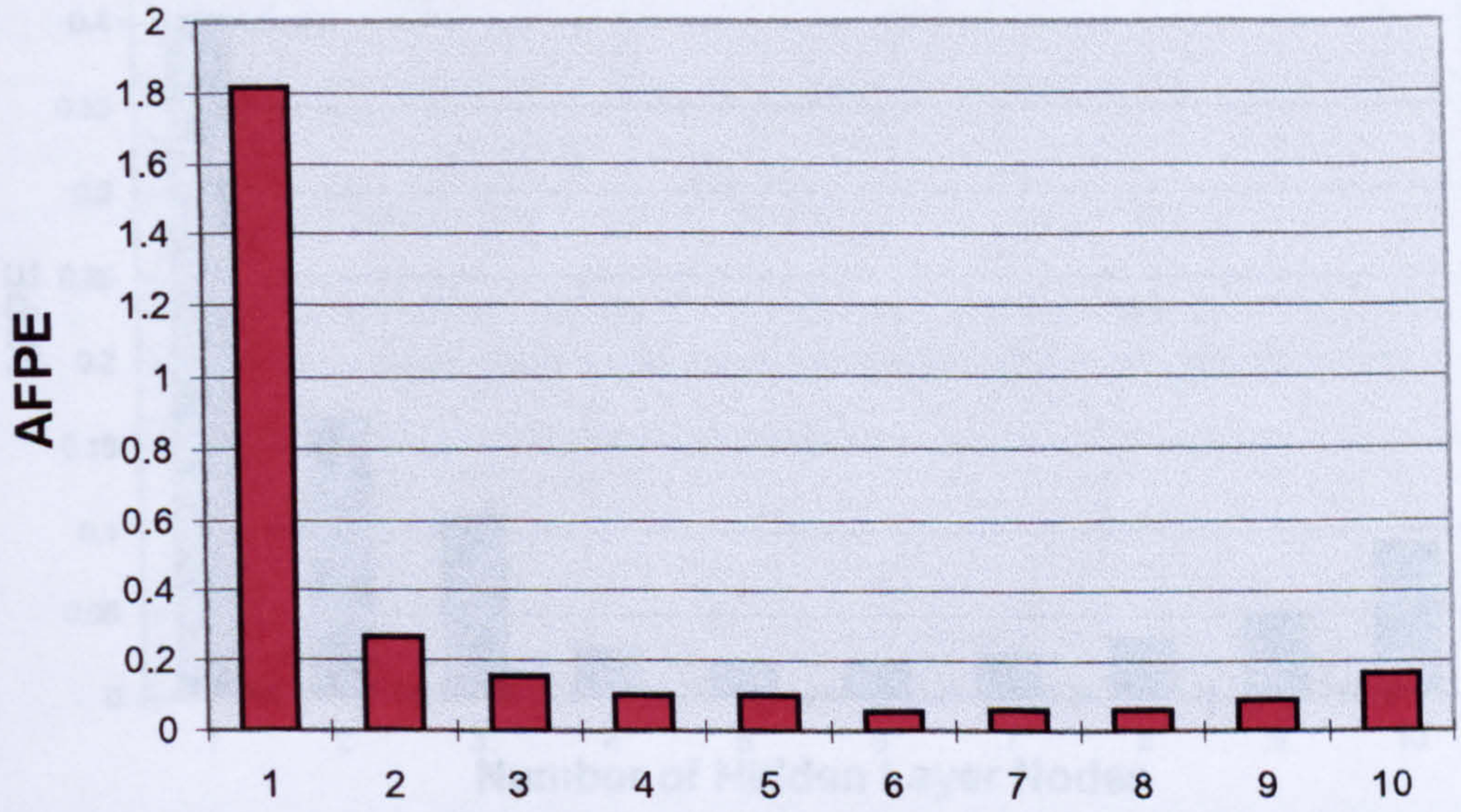


Figure 5.13 AFPE for networks with various network topologies. Results for networks trained using data extracted from environment 1. Networks trained using random placement. Number of test data vectors = 45.

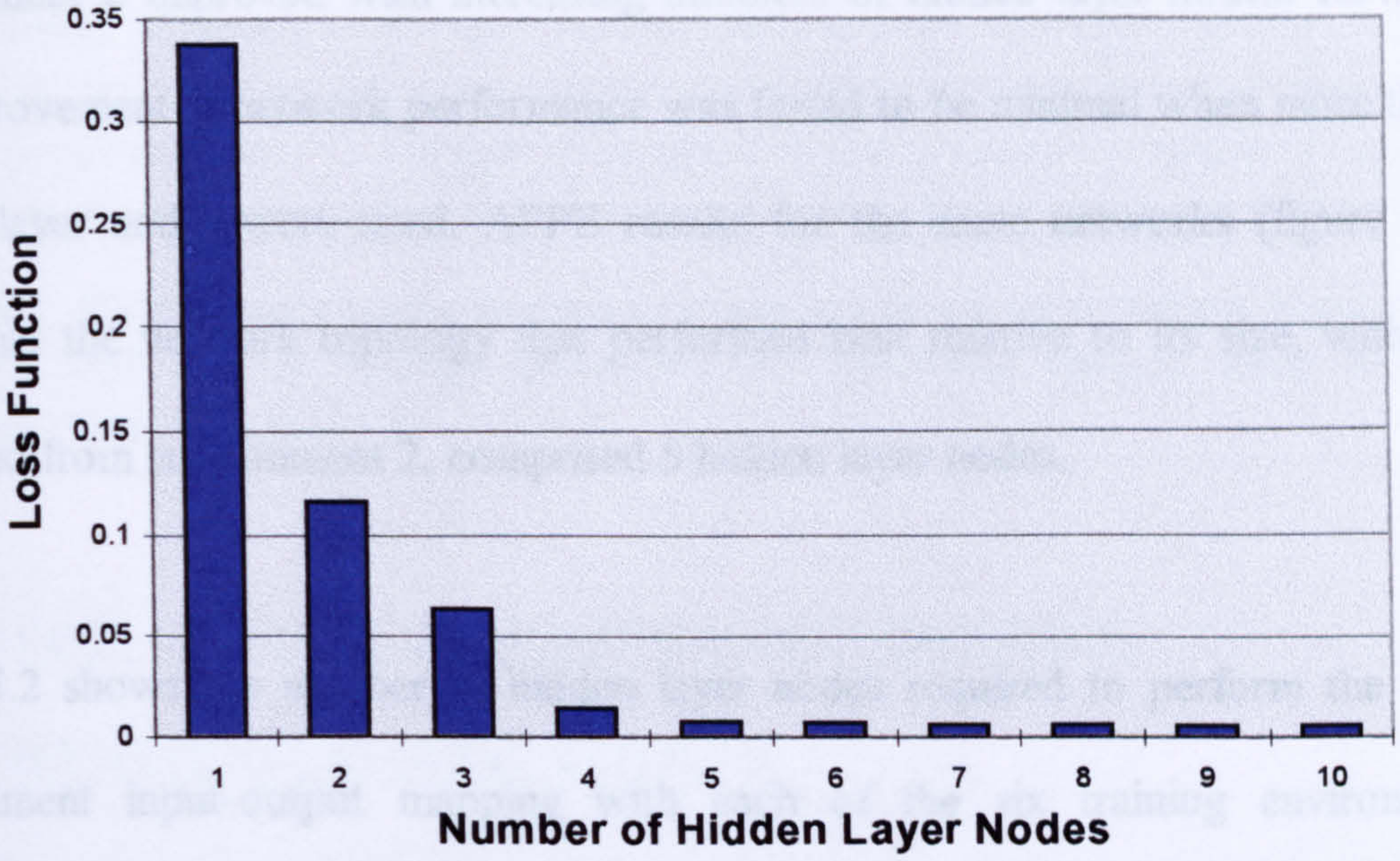


Figure 5.14 Loss functions for networks with various topologies. Results for networks trained using data extracted from environment 2. Networks trained using random placement.

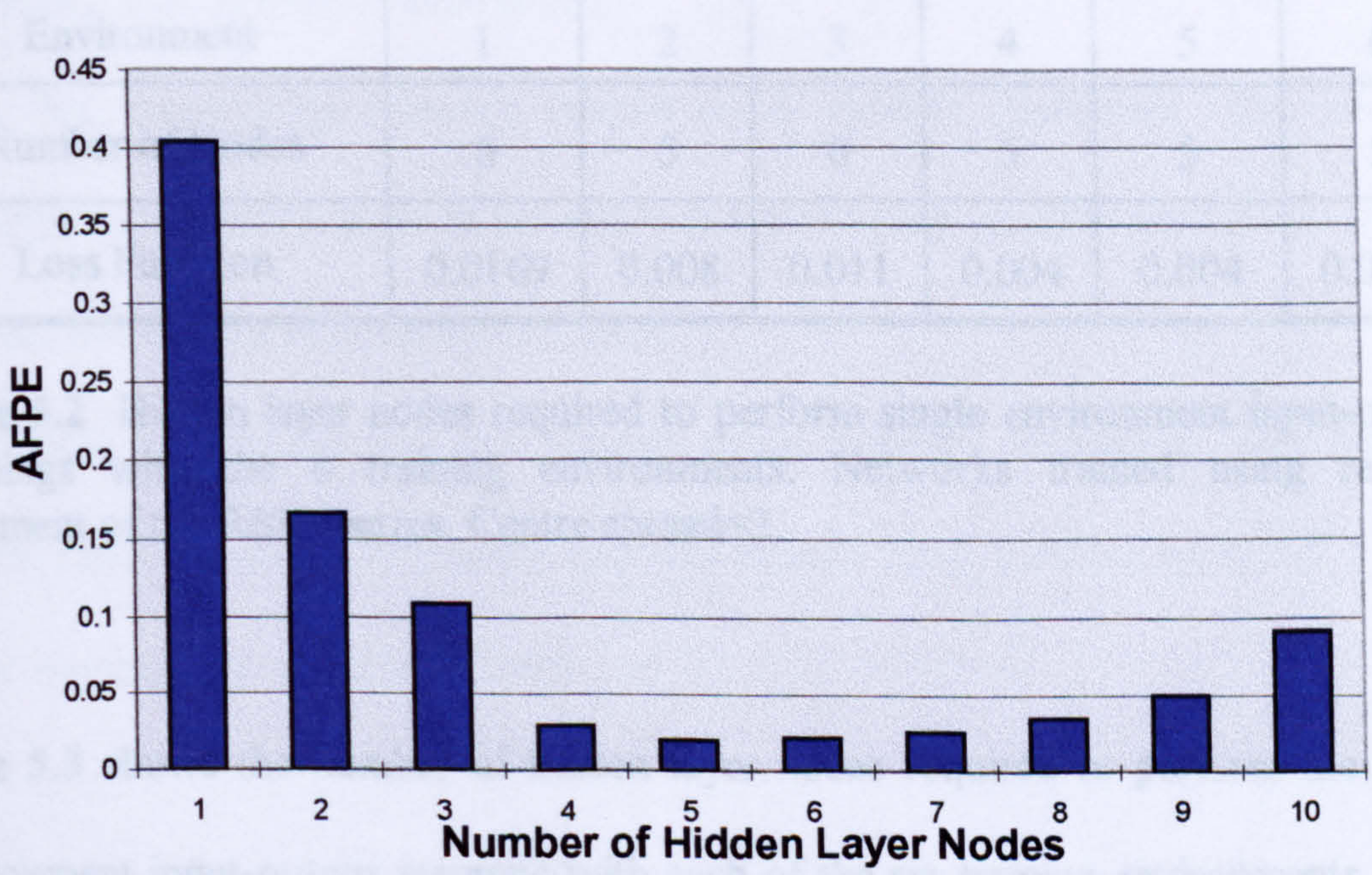


Figure 5.15 AFPE for networks with various network topologies. Results for networks trained using data extracted from environment 2. Networks trained using random placement. Number of test data vectors = 45.

Figure 5.14 shows that the performance of networks trained to respond to environment 2 improved with increasing numbers of hidden layer nodes. However, the improvement in network performance was found to be minimal when more than 5 hidden layer nodes were used. AFPE results for the same networks (figure 5.15) show that the network topology that performed best relative to its size, with data extracted from environment 2, comprised 5 hidden layer nodes.

Table 5.2 shows the number of hidden layer nodes required to perform the single environment input-output mapping with each of the six training environments investigated (based on the results of AFPE tests). The networks were trained using random placement of the RBF centres and a centre spread of 2.

Environment	1	2	3	4	5	6
Number of Nodes	6	5	6	5	5	5
Loss Function	0.0169	0.008	0.011	0.004	0.004	0.0059

Table 5.2 Hidden layer nodes required to perform single environment input-output mappings with the 6 training environments. Networks trained using random placement of the RBF centres. Centre spread=2.

Table 5.3 shows the number of hidden layer nodes required to perform the single environment input-output mapping with each of the six training environments for k-means clustering trained networks that were trained using with a centre spread of 5.

Environment	1	2	3	4	5	6
Number of Nodes	6	5	5	6	6	5
Loss Function	0.017	0.008	0.007	0.01	0.006	0.012

Table 5.3 Hidden layer nodes required to perform single environment input-output mappings with the 6 training environments. Networks trained using k-means clustering. Centre spread = 5.

5.4 EFFECT OF NOISE ON NETWORK PERFORMANCE

5.4.1 REPRESENTING NOISE EFFECTS

The effect of measurement noise on network performance was investigated. The effect on noise present on the force and depth measurements was investigated in isolation. Further testing with the noise signals acting simultaneously was then considered. Two networks were tested:

- the best performing network trained using random placement. The network comprised 6 hidden layer nodes and was trained with an RBF centre spread of 2 times the training data set input domain.
- the best performing network trained using k-means clustering. The network comprised 6 hidden layer nodes and was trained using a centre spread of 5 times the training data set input domain.

The effects of noise were simulated by adding white Gaussian noise to the measured variable. The signal to noise ratio (SNR) was calculated using equation 5.4:

$$SNR = 20 \log_{10} \frac{\sigma_s}{\sigma_n} \text{ dBs} \qquad (5.4)$$

where σ_s and σ_n are the standard deviation of the measured variable and the noise respectively.

The standard deviation of the noise signal was chosen so that a SNR of 20dBs was obtained.

5.4.2 EFFECT OF DEPTH MEASUREMENT ERROR ON NETWORK PERFORMANCE

Errors in the depth measurement were simulated by superimposing noise onto the depth measurements in the test data sets. Figure 5.16 shows loss functions obtained when the two best networks were tested with test data extracted from environment 1.

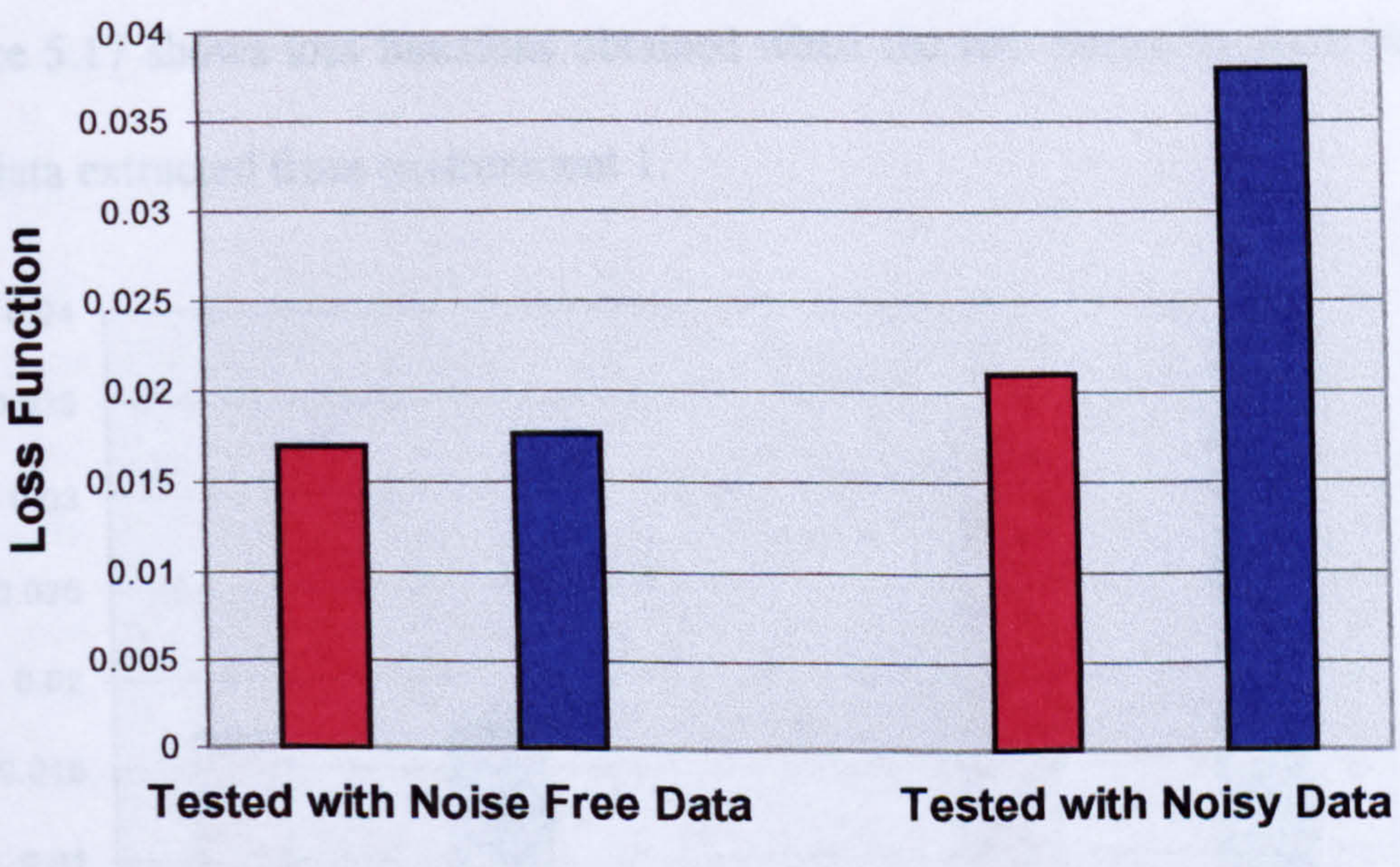


Figure 5.16 Loss functions for networks trained using random placement and k-means clustering. ■ Network trained using random placement. ■ Network trained using k-means clustering. Networks tested with noise free data and data that had noise added to the depth measurement.

The performance of the k-means clustering trained network can be seen to be inferior to the random placement trained network with noise free data. However, although estimations in the presence of noise added to the depth measurement were satisfactory, the network that was trained using random placement had a higher degree of noise immunity. Tests with networks trained with data extracted from other simulated environments were in accordance with the results presented for environment 1.

5.4.3 EFFECT OF FORCE SENSOR NOISE ON NETWORK PERFORMANCE

Errors in the force measurement and force sensor noise were simulated by superimposing the aforementioned noise onto the force measurements in the test data sets. Figure 5.17 shows loss functions obtained when the two networks were tested with test data extracted from environment 1.

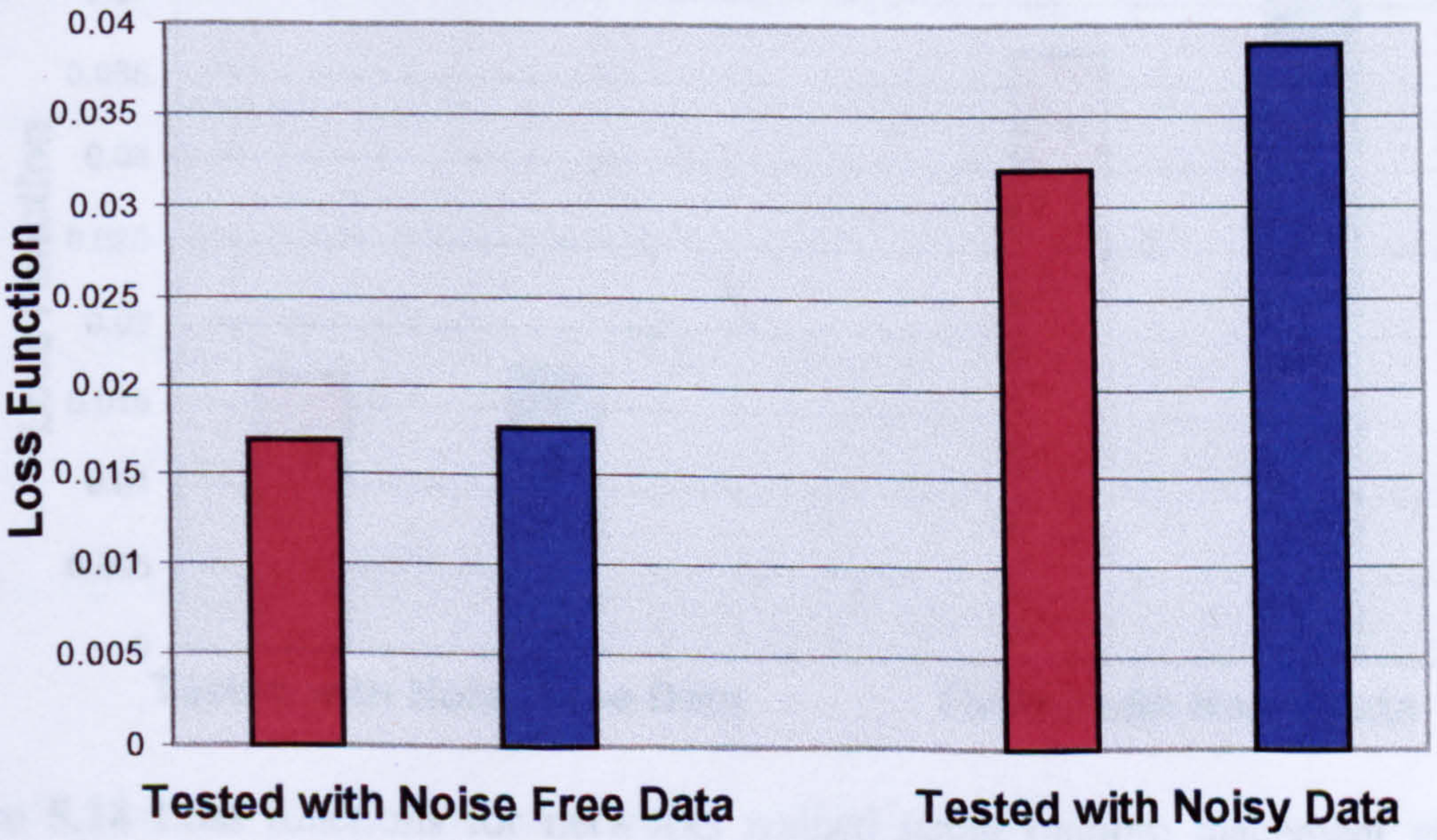


Figure 5.17 Loss Functions for networks trained using random placement and k-means clustering. ■ Network trained using random placement. ■ Network trained using k-means clustering. Networks tested with noise free data and data that had noise superimposed onto the force measurement.

Figure 5.17 shows that network performance remained satisfactory in the presence of force sensor noise. However, the network that was trained using random placement had a higher degree of noise immunity to noisy force measurements. Tests with networks trained using data extracted from the other simulated environments confirmed the repeatability of the result. Thus the random placement trained network was found to yield better estimations than the k-means clustering trained network with each of the noise sources acting in isolation.

5.4.4 COMBINED EFFECT OF FORCE SENSOR NOISE AND DEPTH MEASUREMENT ERROR ON NETWORK PERFORMANCE

Figure 5.18 shows the loss functions obtained when the two networks were tested with test data (extracted from environment 1) that had noise added to the force and depth measurements.

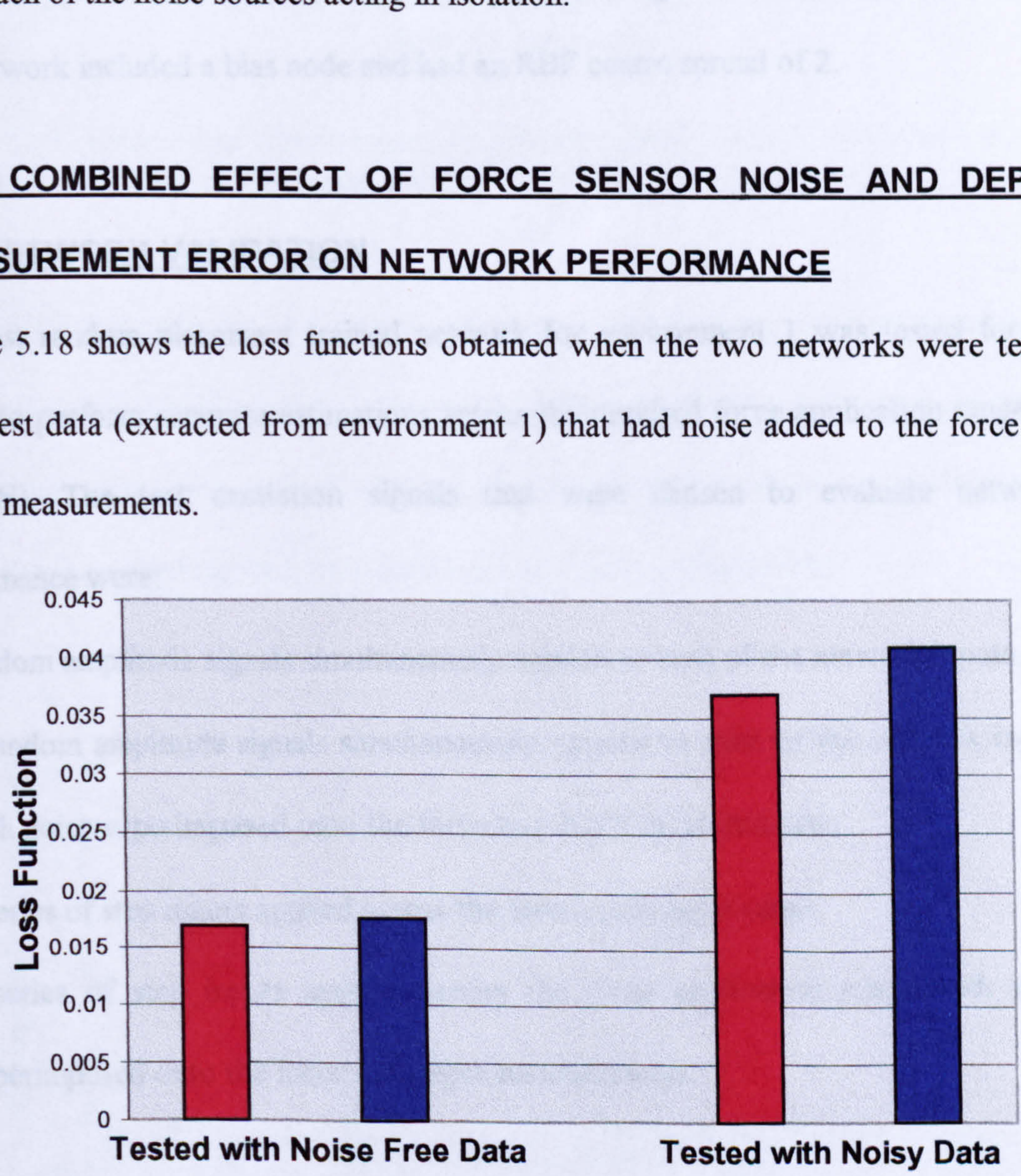


Figure 5.18 Loss functions for networks trained using random placement and k-means clustering. ■ Network trained using random placement. ■ Network trained using k-means clustering. Networks tested with noise free data and data that comprised noise added to the force and depth measurements.

Figure 5.18 shows that network estimations in the presence of noise added to the force and depth measurements were satisfactory. Additionally, the results of the noise tests show that the random placement trained network was found to yield better estimations than the k-means clustering trained network in the presence of noise. Tests with networks trained using data extracted from other environments confirmed the repeatability of the result. Thus the random placement trained network that comprised 6 hidden layer nodes was the best performing network for environment 1. The network included a bias node and had an RBF centre spread of 2.

5.5 NETWORK VALIDATION

The best random placement trained network for environment 1 was tested for its ability to perform accurate estimations across the required force application range (1 to 15N). The test excitation signals that were chosen to evaluate network performance were:

- random amplitude signals simultaneously applied to each of the network inputs.
- a random amplitude signals simultaneously applied to each of the network inputs with noise superimposed onto the force and depth measurements.
- a series of step inputs applied across the force application range.
- a series of step inputs applied across the force application range with noise superimposed onto the force and depth measurements.

Figure 5.19 shows the estimations obtained when the best random placement trained network was tested with a random amplitude signals applied to the network inputs.

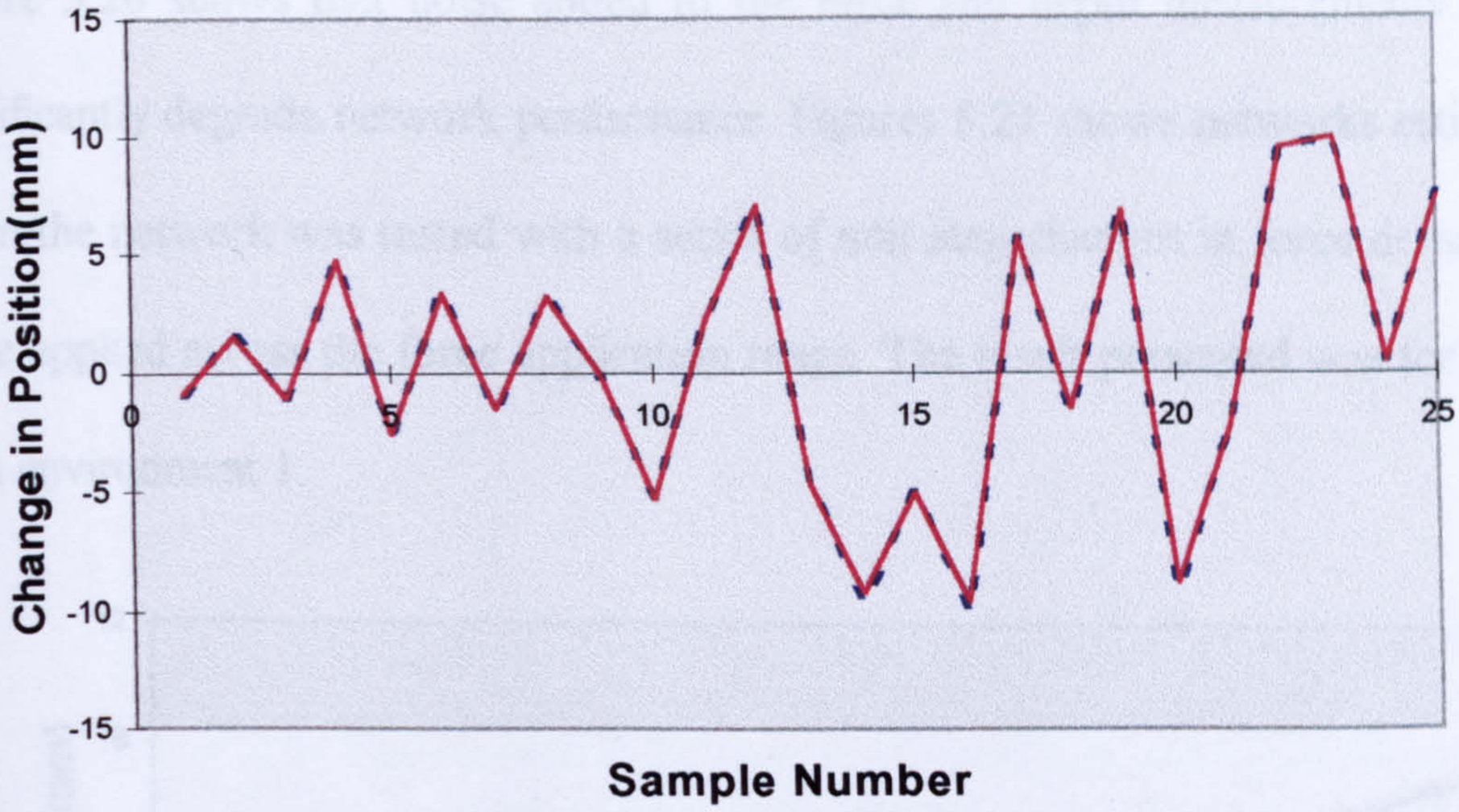


Figure 5.19 Estimations when random placement trained network was tested with a random amplitude signal. ■ Target output. ■ Network estimation.

Figure 5.19 illustrates that estimations were accurate when the network was tested with the random amplitude signals. Figure 5.20 shows estimations obtained when the best random placement trained network was tested with random amplitude signals that had noise superimposed onto the force and depth measurements.

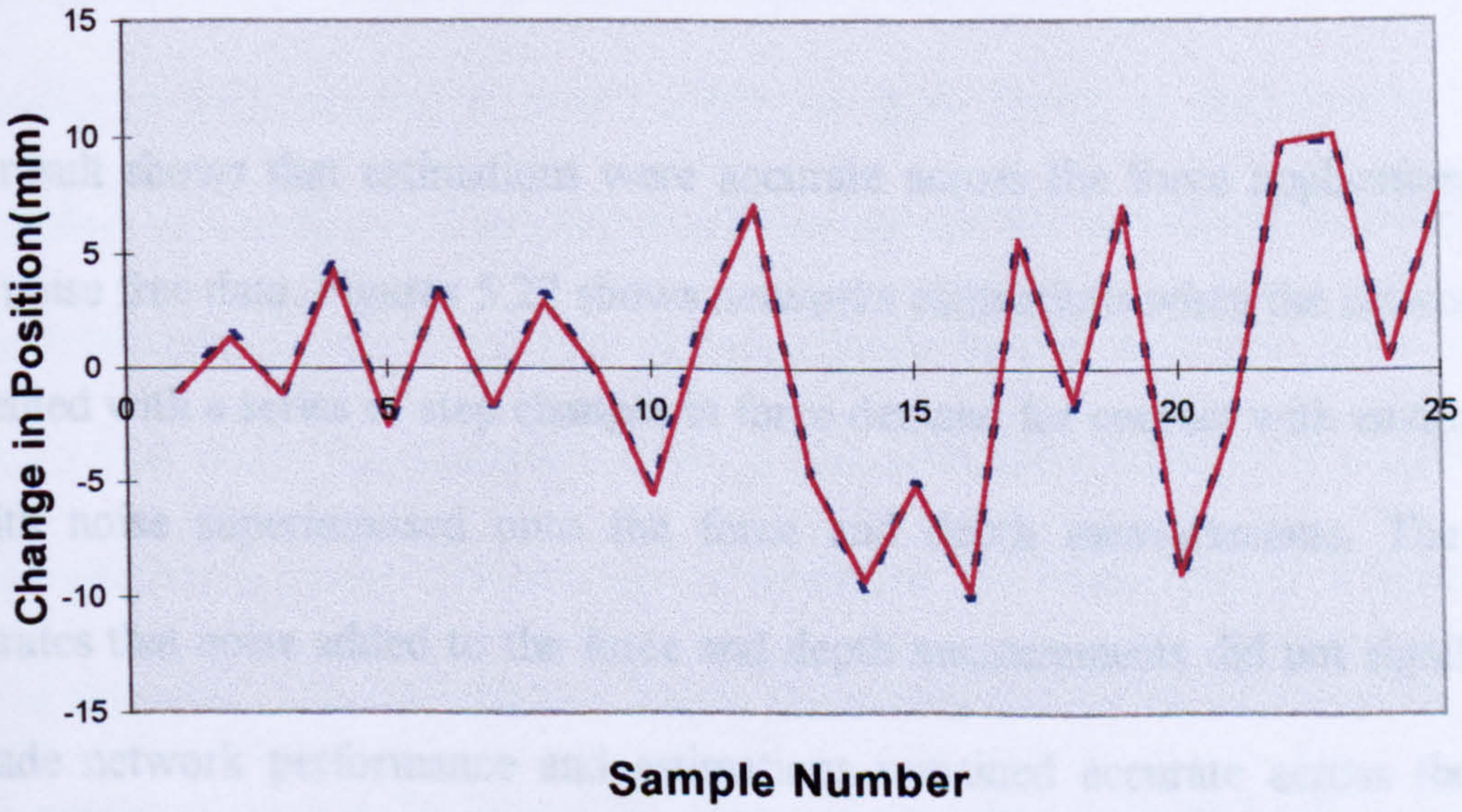


Figure 5.20 Estimations when random placement trained network was tested with a noisy random amplitude signal. ■ Target output. ■ Network estimation.

Figure 5.20 shows that noise added to the force and depth measurements did not significantly degrade network performance. Figures 5.21 shows networks estimations when the network was tested with a series of unit step changes in force demand that were applied across the force application range. The result presented was for contact with environment 1.

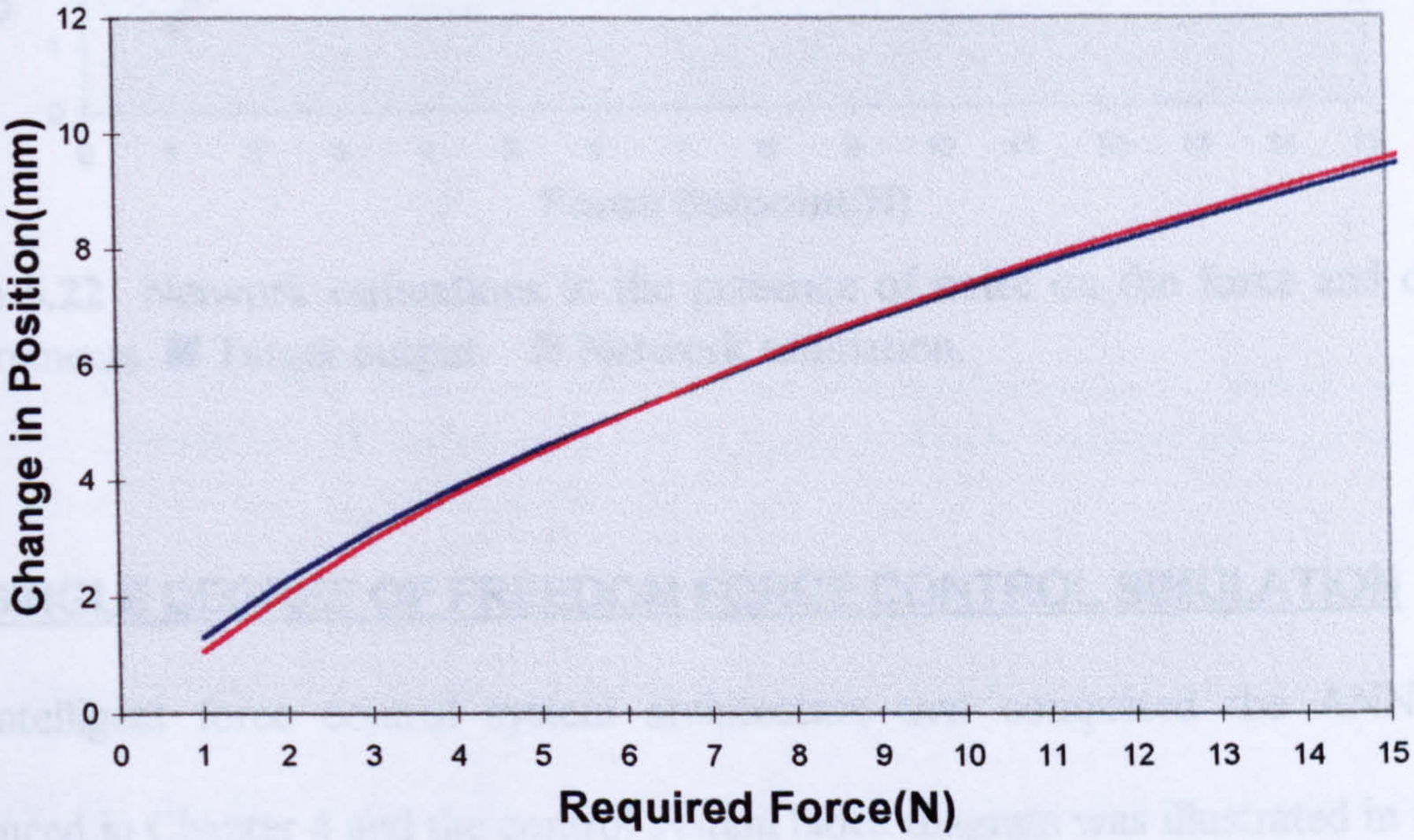


Figure 5.21 Noise free estimations across the force application range. ■ Target output. ■ Network estimation.

The result shows that estimations were accurate across the force application range with noise free data. Figures 5.22 shows networks estimations when the network was presented with a series of step changes in force demand for contact with environment 1 with noise superimposed onto the force and depth measurements. The result illustrates that noise added to the force and depth measurements did not significantly degrade network performance and estimations remained accurate across the force application range.

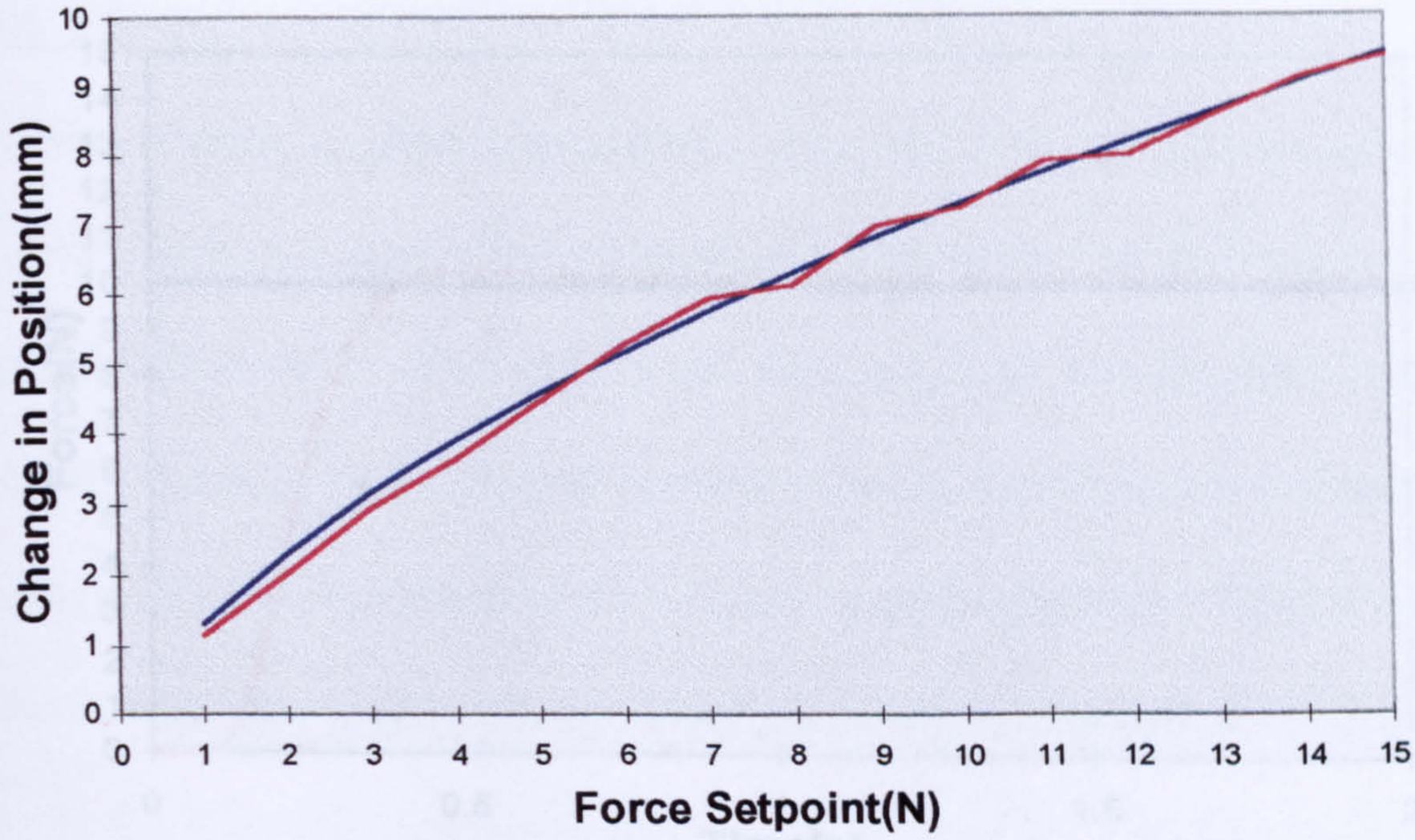


Figure 5.22 Network estimations in the presence of noise on the force and depth measurements. ■ Target output. ■ Network estimation.

5.6 SINGLE DEGREE OF FREEDOM FORCE CONTROL SIMULATION

The intelligent force control system architecture that comprised the ANN was introduced in Chapter 4 and the control system block diagram was illustrated in figure 4.8. The random placement trained network that comprised 6 hidden layer nodes was loaded into the single degree of freedom (DOF) manipulator ACSL simulation and the intelligent force control scheme's ability to apply forces to environment 1 was investigated by simulation. The control scheme applies forces without knowledge of the environments position, as such a requirement would limit the force control scheme's 'real world' applicability. When a non-zero force setpoint is commanded, the end-effector is moved towards the environment and when contact is sensed, the ANN is activated. The end-effector is then positioned within the environment by the ANN output. Figure 5.23 shows the transient response obtained when the force control scheme was commanded to apply a 10N force to environment 1.

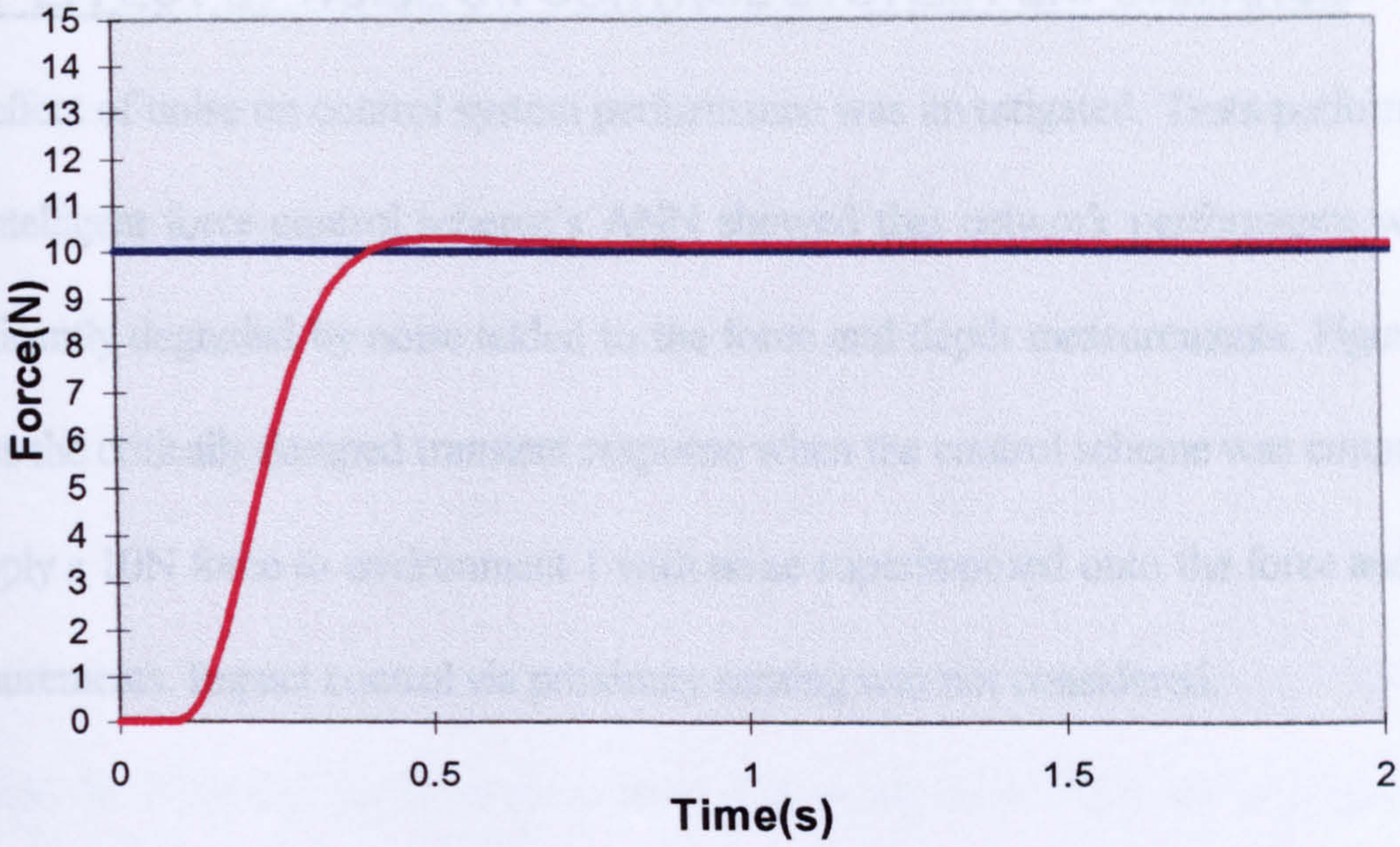


Figure 5.23 Transient response for contact with environment 1. Effects of noise were not considered. ■ Force setpoint ■ Sensed force. Impact with environment occurred at t=0.1 sec. K_p=160, K_v=0.02

The control system can be seen to recover from the initial impact (occurring at t=0.1 sec) and the response was stable. A settling time of 0.6 seconds was measured for the system to settle within 5% of its steady state value. A small steady state offset (the steady state contact force was 10.14N) was evident which can be attributed to non-ideal network estimations which resulted in non-ideal steady state positioning of the end-effector within the environment. This in turn resulted in a non-ideal steady state force being applied to the environment. Although the position control loop was critically damped, a slight overshoot was found with force application. As such, the system damping was increased by setting the tacho-feedback gain to 0.03. The increase in tacho-feedback gain critically damped the force response.

5.6.1 EFFECT OF NOISE ON CONTROL SYSTEM PERFORMANCE

The effect of noise on control system performance was investigated. Tests performed on the intelligent force control scheme's ANN showed that network performance was not significantly degraded by noise added to the force and depth measurements. Figure 5.24 shows the critically damped transient response when the control scheme was commanded to apply a 10N force to environment 1 with noise superimposed onto the force and depth measurements. Impact control via proximity sensing was not considered.

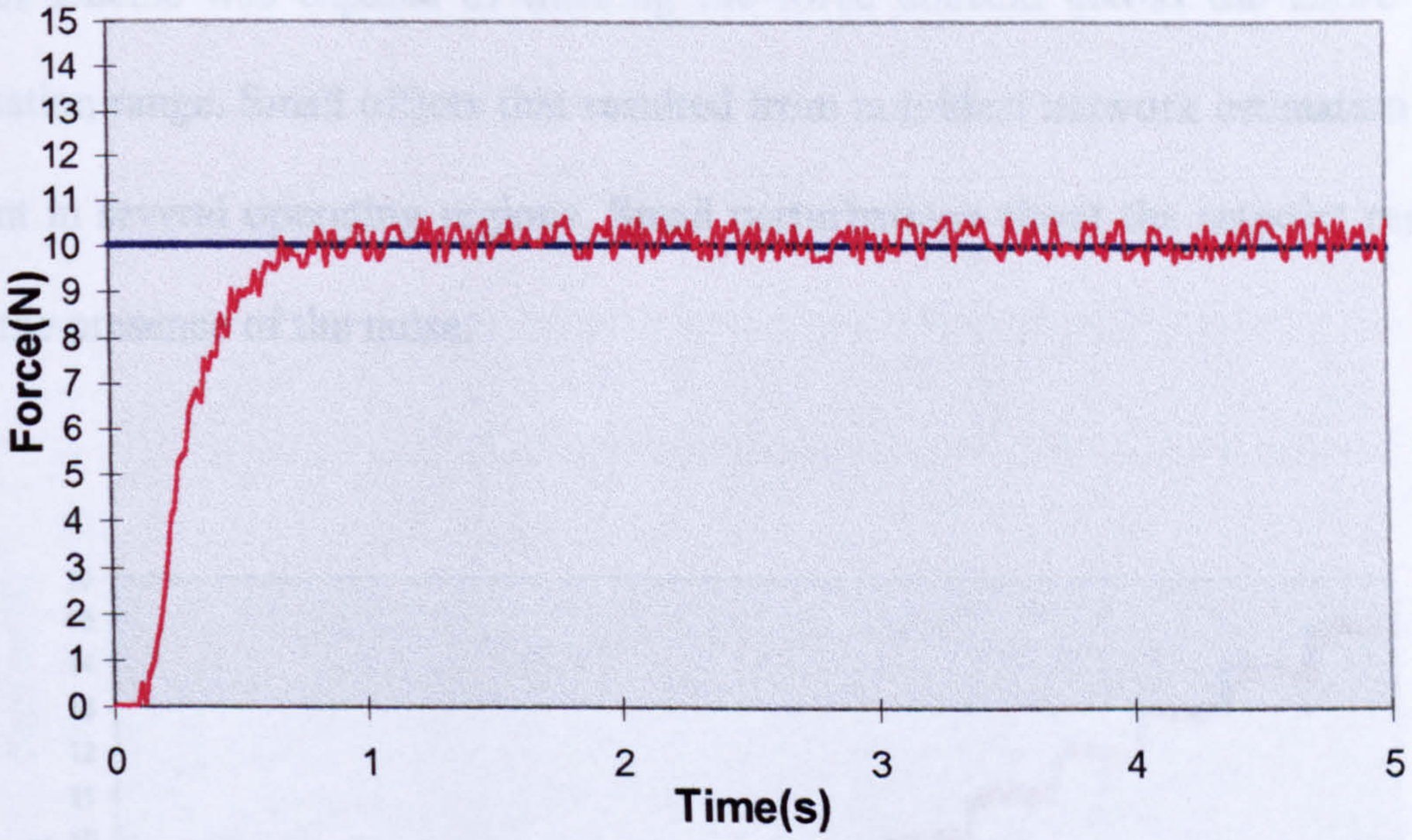


Figure 5.24 Transient response for contact with environment 1. Result generated with noise added to the force and depth measurements. ■ Force setpoint ■ Sensed force. Impact with environment occurred at t=0.1 sec. K_p=160, K_v=0.03

Figure 5.24 illustrates the intelligent force control scheme's ability to acquire and maintain a specified contact with its environment in the presence of noise. Perturbations about the setpoint resulted from the presence of the noise signal. Although masked by the noise, a slight offset was measured.

5.6.2 STEP TESTS

Step tests were performed so that the control schemes force application capability could be assessed across the entire force application range (1N to 15N). The control system was commanded to apply a 1N force to environment 1 and the force demand was increased in 1N steps at 5 second intervals.

Figure 5.25 shows the results of the step tests for contact with environment 1 with noise added to the force and depth measurements. The result shows that the intelligent force control scheme was capable of tracking the force demand across the entire force application range. Small offsets that resulted from non-ideal network estimation were evident in several operating regions. Small perturbations about the setpoint resulted from the presence of the noise.

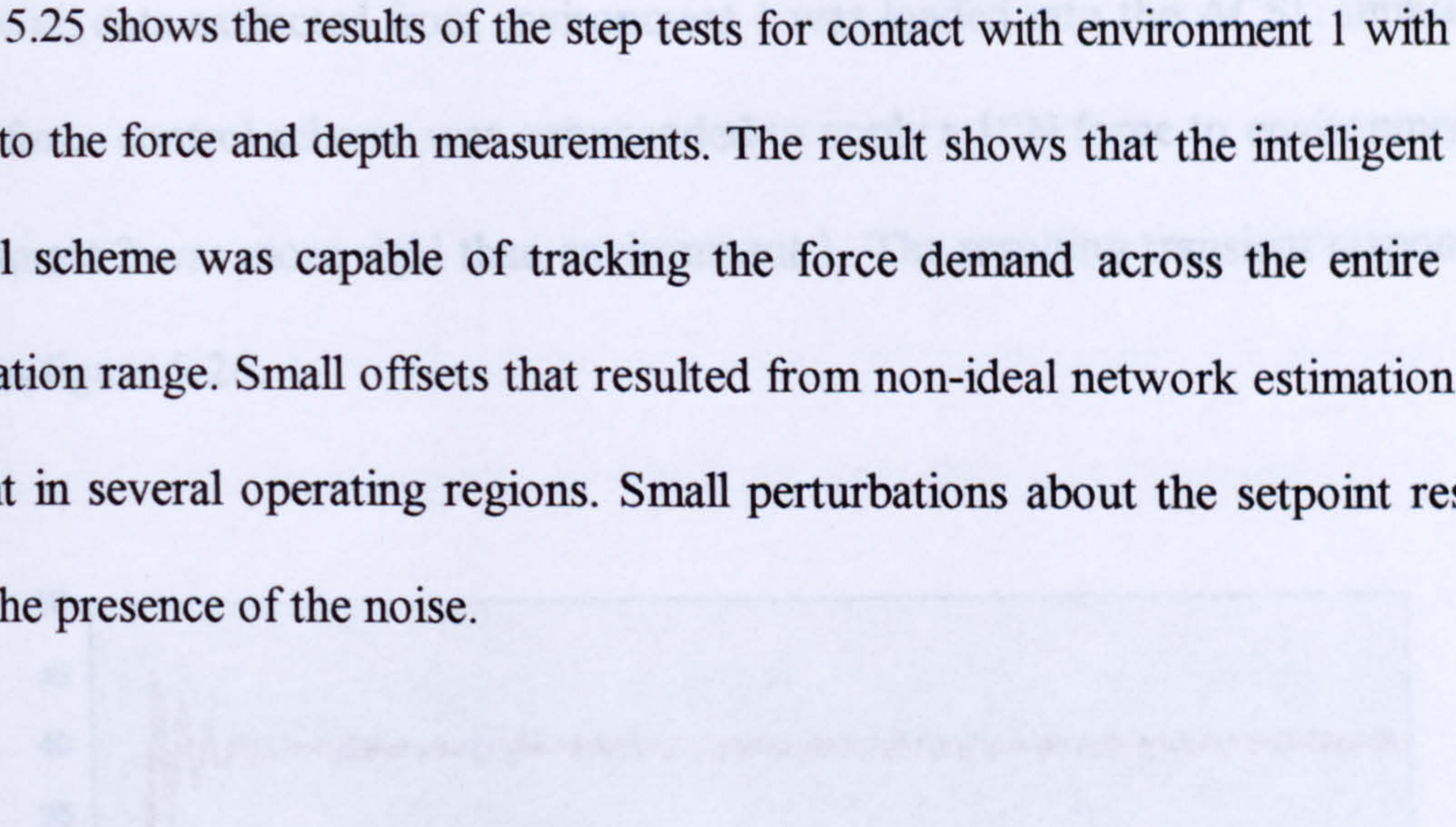


Figure 5.25 Transient response for step tests with environment 1. Result was generated with noise added to the force and depth measurements ■ Force setpoint ■ Sensed force. Impact with environment occurred at t=0.1 sec. K_p=160, K_v=0.03

5.6.3 CONTACT WITH ENVIRONMENTS WITH DIFFERING DEGREES OF RIGIDITY

The intelligent force control scheme was tested for its ability to apply forces to environments with differing degrees of rigidity. The control scheme must be capable of applying forces to a wide range of environments when the degree of environmental rigidity is not known *a priori*. The best random placement trained network that was trained with data extracted from environment 1 was loaded into the ACSL simulation and the force control scheme was commanded to apply a 10N force to environment 2. Environment 2 was more rigid than environment 1. The resulting transient response is shown in figure 5.26.

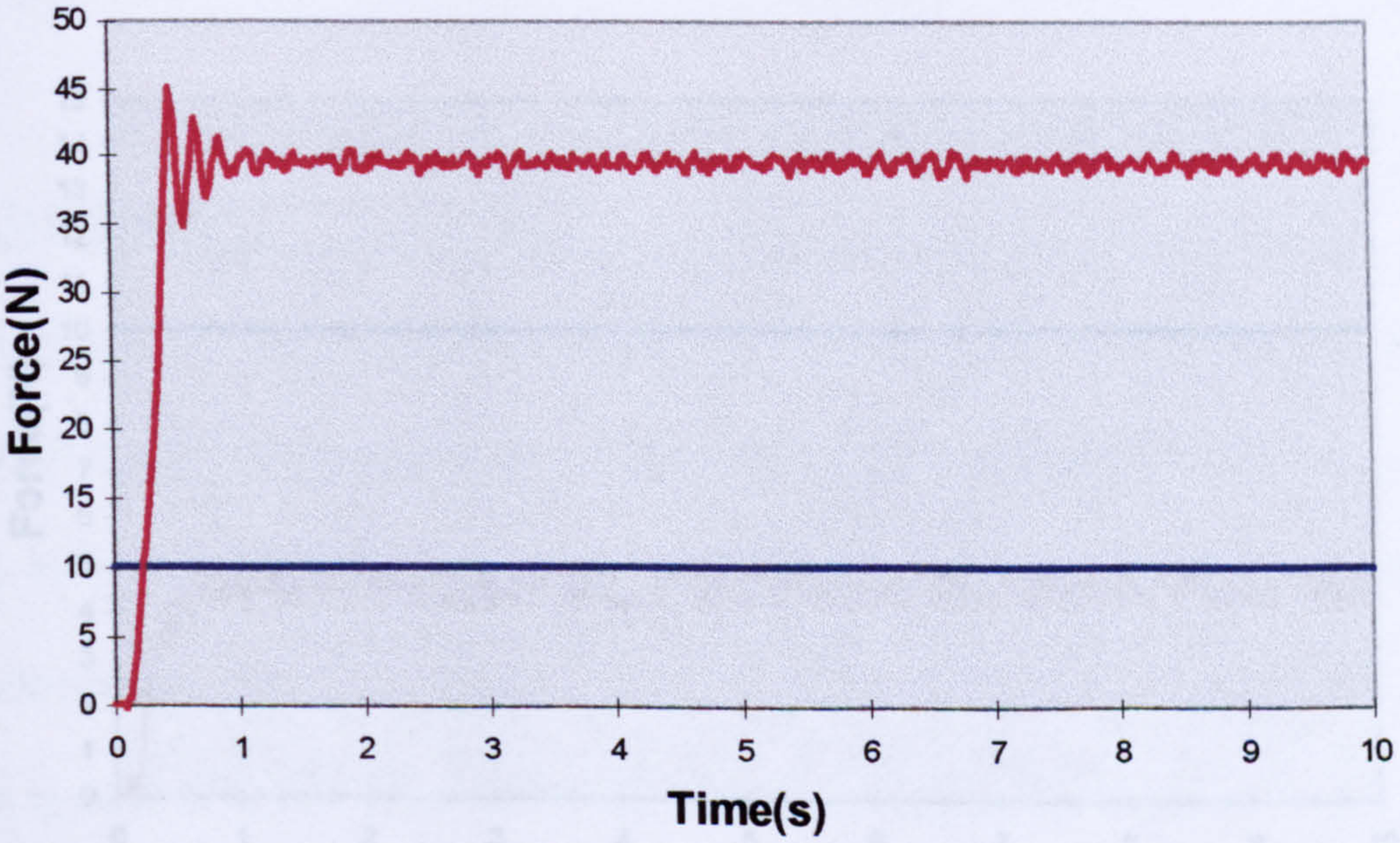


Figure 5.26 Transient response for contact with environment 2. Result was generated with noise added to the force and depth measurements. ■ Force setpoint. ■ Sensed force. Impact with environment occurred at t=0.1 sec. K_p=160, K_v=0.03.

Figure 5.26 shows that the transient response obtained was stable but oscillatory and a large steady state offset was evident. In this instance the applied force was greater

than the 10N force demand. Further network testing with environments that were more rigid than environment 2 showed that the offset increased as the environment became more rigid (results not presented). The large force error can be attributed to the fact that the control system's ANN does not have knowledge of contact with an environment as rigid as environment 2 and it is attempting to extrapolate on its knowledge of idealised reaction to environment 1.

Figure 5.27 shows the transient response when the control system was commanded to apply a force of 10 N to environment 10, an environment that was more compliant than environment 1. Contact model parameters for environment 10 are shown in Appendix A and the environment's hardness profile is shown in Appendix C.

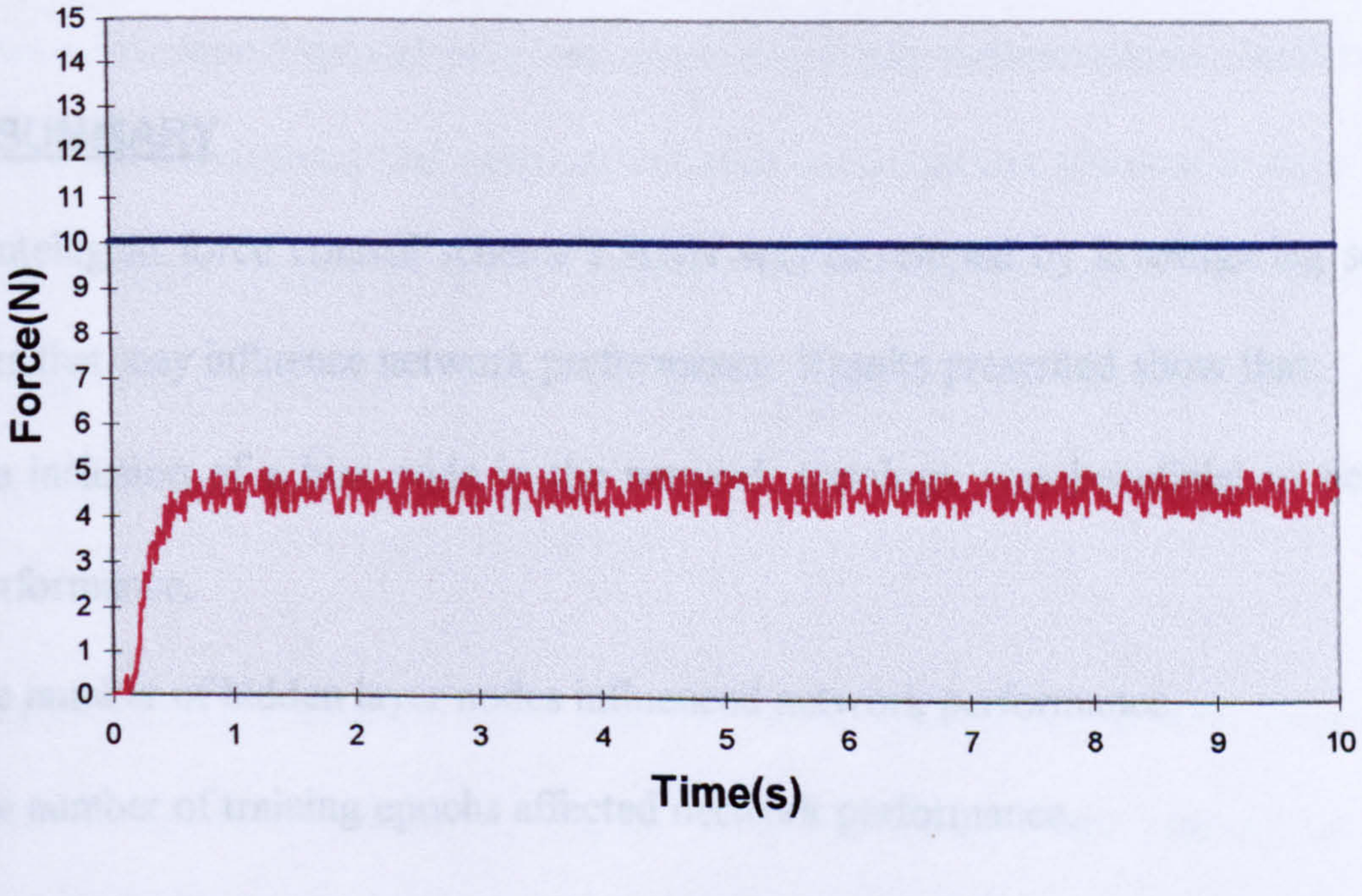


Figure 5.27 Transient response for contact with environment 10. Result was generated with noise was added to the force and depth measurements. ■ Force setpoint. ■ Sensed force. Impact with environment occurred at t=0.1 sec. K_p=160, K_v=0.03.

Figure 5.27 shows that the transient response for contact with environment 10 was stable, but a significant steady state offset was again evident. In this instance the steady state contact force was less than the force demand. Tests with environments that were more compliant than environment 10 showed that the offset increased as the contact environment became more compliant.

After several tests with a range of simulated non-rigid environments with differing degrees of rigidity, it was concluded that networks trained using data extracted from a single environment did not have the ability to perform accurate estimations with environments that have compliance modes that were different to the 'training' environment.

5.7 SUMMARY

The intelligent force control scheme's ANN was developed by investigating several factors that may influence network performance. Results presented show that:

- the inclusion of a bias node in the network topology was beneficial to network performance.
- the number of hidden layer nodes influenced network performance.
- the number of training epochs affected network performance.
- the initial centre placement influenced network performance. Networks trained using a single RBF centre placement gave poor estimations. As such a repeated random placement training methodology was adopted for several hundred centre randomisations.

- the spread of the RBF centre initialisations significantly affected network performance.
- the use of a k-means clustering algorithm to redistribute the RBF centres to more 'optimal' positions prior to network training was found to have a detrimental affect on network performance. However, performance was found to improve when k-means clustering was used with centre spreads that were higher multiples of the training data set input domain.

An investigation of the effect on network performance found that network performance was not significantly degraded by the presence of noise added to the force and depth measurements. Network testing with random amplitude signals and step tests showed that estimations were accurate with noise free data and with data that has noise superimposed onto the force and depth measurements. Additionally, results presented showed that networks trained using random placement were found to have a higher degree of noise immunity than networks trained using k-means clustering. The repeatability of the results was validated with several environments, each with differing degrees of rigidity.

Simulation results presented show that the ANN based force control system was capable of recovering from an impact with the environment without *a priori* specification of the contact environment position. A stable force response with slight offset was obtained when the control system made contact with the network 'training' environment. The offset was a result of non-ideal network estimations. When the environmental rigidity varied from that of the training environment, the transient

response was stable but significant steady state force errors were evident. The magnitude of the force errors were found to be directly proportional to the degree of variance in the contact environment rigidity from that of the training environment. Thus the single environment trained networks did not have the ability to perform accurate estimations with environments that had compliance modes that were different to the 'training' environment.

CHAPTER 6

CONTACT WITH A RANGE OF NON-RIGID ENVIRONMENTS

6.1 INTRODUCTION

The investigation into contact with a single non-rigid environment presented in Chapter 5 showed that a 'correctly trained' RBF network was capable of performing a single environment input-output mapping to a high degree of accuracy. However, networks trained with data extracted from a single non-rigid environment were found to be incapable of accurately responding to environments with differing degrees of rigidity.

This chapter describes an investigation that extends the work presented in Chapter 5 so that interaction with a range of non-rigid environments is possible. The investigation was centered around the training of RBF networks with a data set that comprised data extracted from several non-rigid environments. Once trained, the RBF network was tested for its ability to perform accurate estimations across a force application range of 1N to 15N, with a range of environments with differing degrees of rigidity. The effect of noise on network performance was investigated.

Two network training methodologies were investigated during the development of the intelligent force control schemes ANN, namely:

- repeated random placement of the RBF centres.
- k-means clustering with centre randomisations within higher multiples of the network training data set input domain.

Once trained, the ANN model was incorporated into the single degree of freedom manipulator simulation developed in Chapter 4 and the 'intelligent' force control systems ability to apply forces to the a range of non-rigid environments, each with differing compliance modes was investigated by simulation. The effect on network performance of noise added to the force and depth measurements was also investigated.

The network training objective was the development of a single RBF network that was capable of:

- accurately responding to environments represented in the network training data set.
- accurately responding to environments that were not represented in the network training data set.
- performing accurate estimations in the presence of noise added to the force and depth measurements.

6.2 NETWORK TRAINING STRATEGY FOR CONTACT WITH A RANGE OF ENVIRONMENTS

In an attempt to overcome the limitations of 'single environment' trained networks, a new network training strategy was required that allowed for contact with a range of

non-rigid environments. The network training strategy that was investigated was the use of a single RBF network to model 'idealised reaction' to a range of environments. The development of the 'multi-environment' network required networks to be trained with a data set that comprised data extracted from several non-rigid environments.

The network development phases are illustrated in figure 6.1, which shows the combining of the 'single environment' training data sets prior to network training and testing.

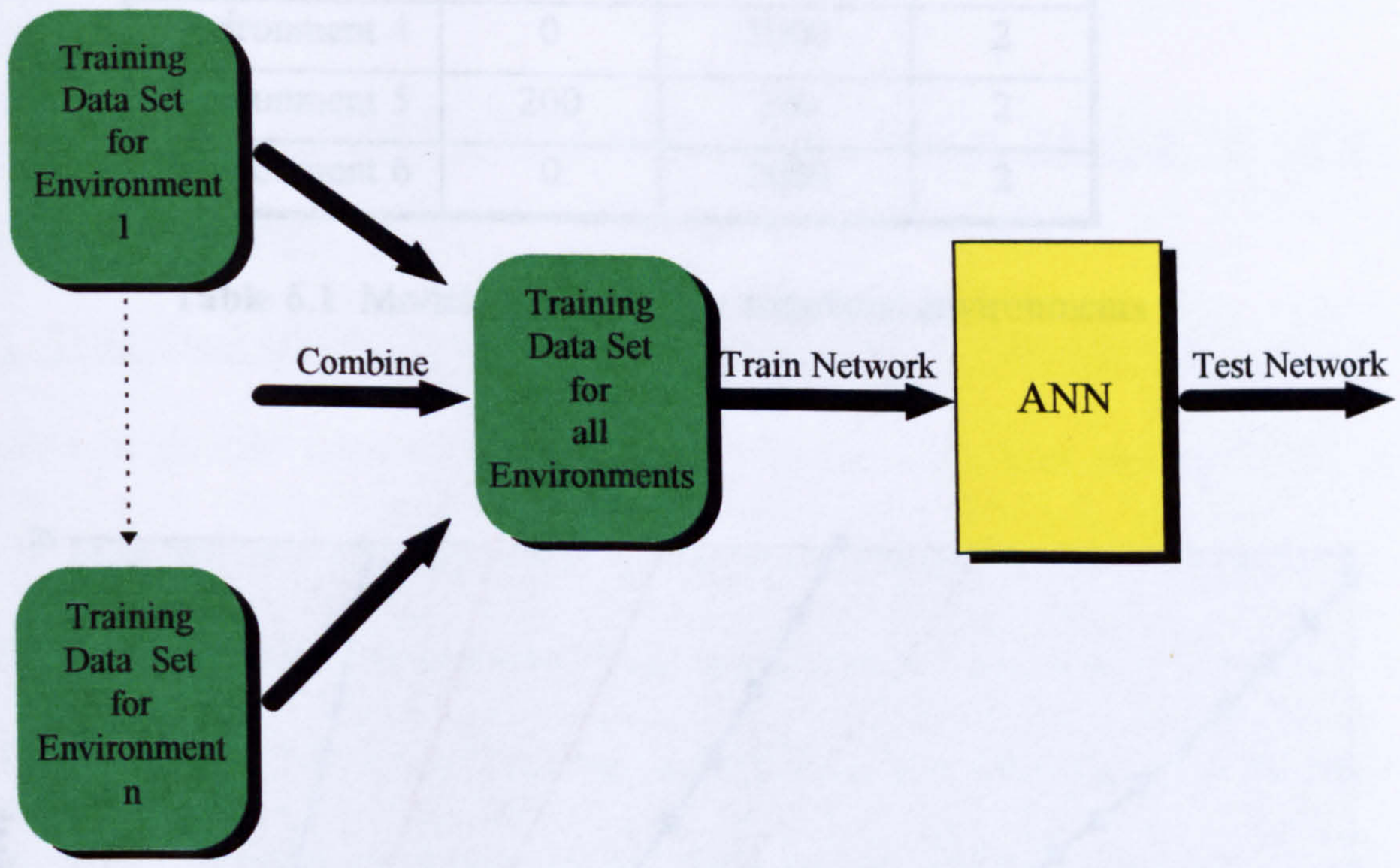


Figure 6.1 Development phases for multi-environment network training

6.3 NETWORK TRAINING DATA GENERATION

Contact with a fixed number of non-rigid environments was initially investigated. Six simulated environments, each with differing compliance modes, were chosen for the investigation. Training data sets for the six environments were generated using the

data extraction technique adopted during the single environment investigation presented in Chapter 5. The contact model that was adopted for the investigation was introduced in equation 5.1 and model parameters for the simulated environments are shown in table 6.1. Hardness profiles for the six simulated environments are shown in figure 6.2.

	k_1(N/m)	k_3(kN/m^2)	a
Environment 1	100	100	2
Environment 2	100	500	2
Environment 3	100	300	2
Environment 4	0	1000	2
Environment 5	200	200	2
Environment 6	0	2000	2

Table 6.1 Model parameters for simulated environments

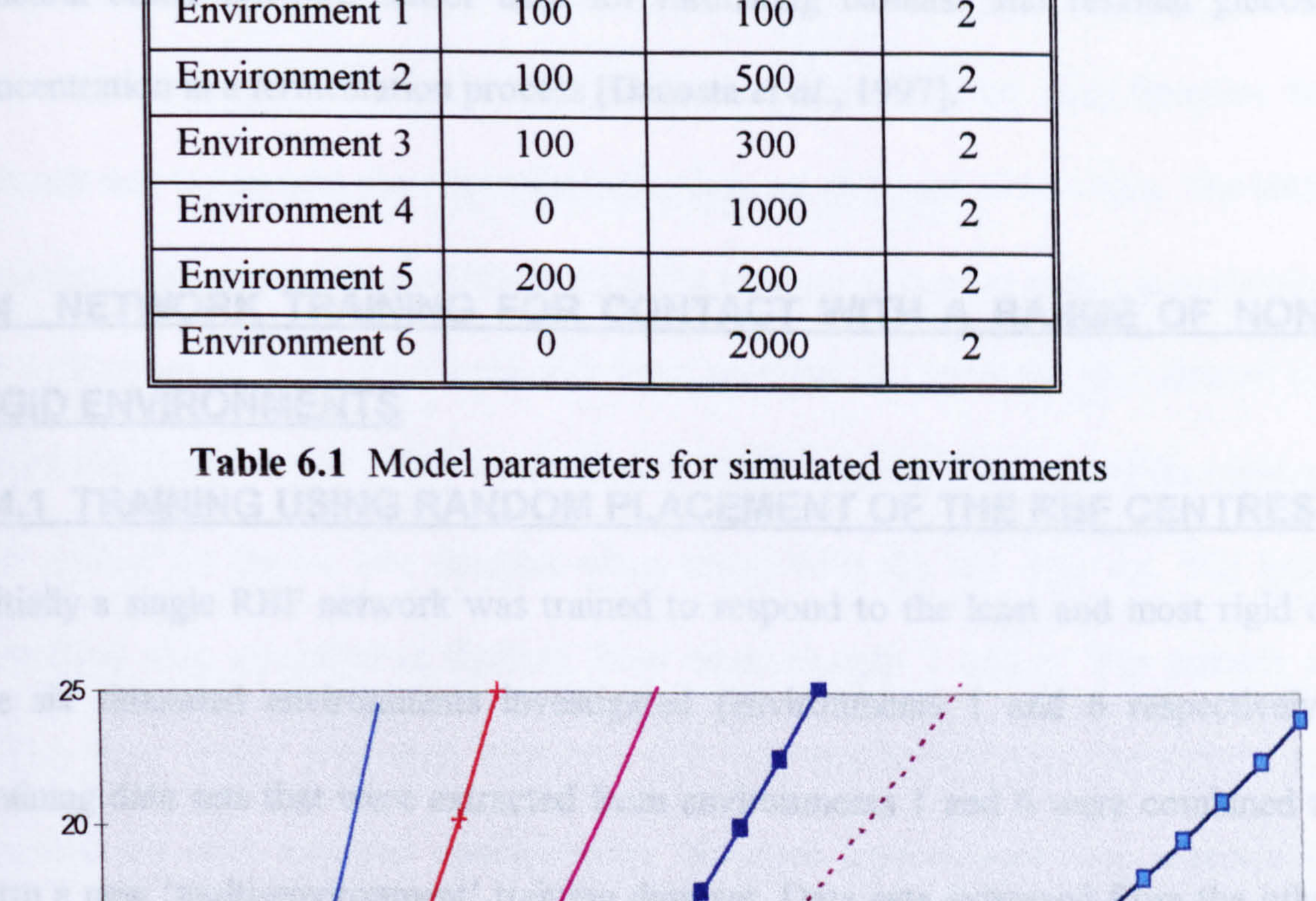


Figure 6.2 Hardness profiles for simulated environments

Network training data that was used to train 'multi-environment' networks was generated by combining 'single environment' data sets. The multi-environment training data set thus comprised knowledge of 'idealised reaction' to each environment 'represented' in the multi-environment training data set. The training of neural networks with combined data sets has been used for a range of applications, including a neural based character recognition system [Fincher and Wade, 1990] and a neural based software sensor used for estimating biomass and residual glucose concentration in a fermentation process [Dacosta *et al.*, 1997].

6.4 NETWORK TRAINING FOR CONTACT WITH A RANGE OF NON-RIGID ENVIRONMENTS

6.4.1 TRAINING USING RANDOM PLACEMENT OF THE RBF CENTRES

Initially a single RBF network was trained to respond to the least and most rigid of the six simulated environments investigated (environments 1 and 6 respectively). Training data sets that were extracted from environments 1 and 6 were combined to form a new 'multi-environment' training data set. Data sets extracted from the other simulated environments were used to test the networks ability to perform accurate estimations with environments that had intermediate compliance modes (i.e. compliance modes between those of environments 1 and 6).

Results from the single environment investigation presented in Chapter 5 showed that the number of hidden layer nodes used in an RBF network significantly influenced network performance. The number of hidden layer nodes required to perform a non-linear input/output mapping is governed by the complexity of the input/output

mapping and empirical methods for determining an 'optimum' number of hidden layer nodes do not exist. As such, a trial and error approach to finding a suitable network topology for multi-environment I/O mapping was adopted. Network training using random placement of the RBF centres was initially investigated.

A single node was used in the network's hidden layer and the network was trained using two epochs of the training data set. RBF centre vectors were randomly placed over twice the input domain for 500 centre iterations, and for each iteration the network was tested with the original training data set and a set of test data. The latter comprised data extracted at random from the two environments under investigation for contact up to 20N. Two networks were retained, the network that yielded the lowest loss function (i.e. MSE) when tested with the training data set and the network that yielded the lowest loss function when tested with the test data set. The latter comprised data extracted at random from environments 1 and 6. The number of hidden layer nodes was then increased and the training/selection procedure was repeated for each network configuration. Network performance was assessed by considering the MSE between the network estimation and the target output and Akaikes Final Prediction Error (AFPE) was used as a measure of network parsimony.

Figure 6.3 shows the loss functions for tests with networks that gave the best performance with the multi-environment test data set. Loss functions for tests with single environment trained networks, trained and tested using data extracted from environment 1, have been included for comparison.

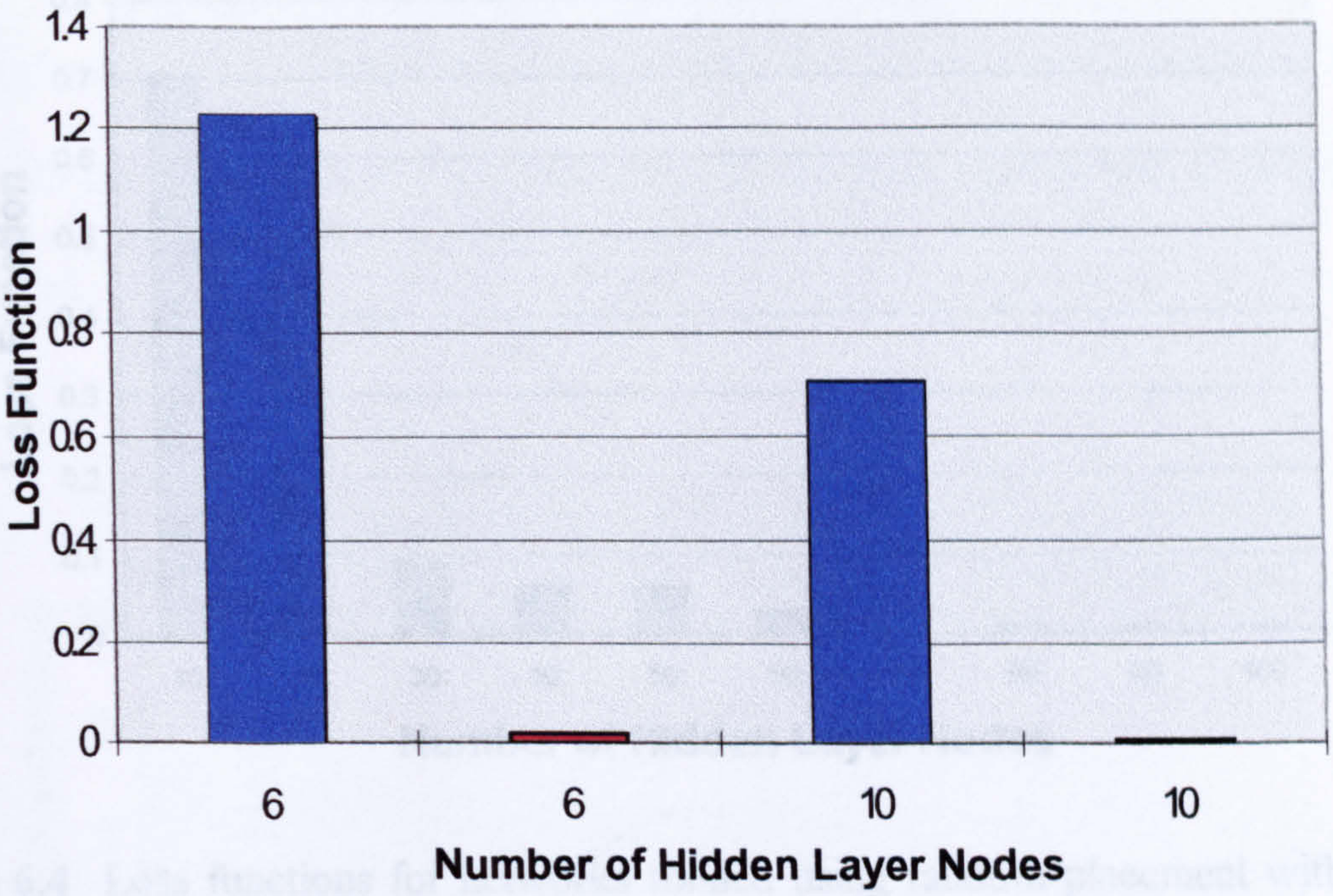


Figure 6.3 Comparison between single environment trained networks and multi-environment trained networks. ■ Results of tests with multi-environment trained network: Test data extracted from environments 1 and 6. ■ Results of tests with single environment trained network: Test data extracted from environment 1.

Figure 6.3 illustrates that network topologies that could accurately perform single environment I/O mappings were unable to perform multi-environment I/O mappings to the same degree of accuracy. However, the performance of the multi-environment trained network improved when the number of hidden layer nodes was increased and as such, the use of larger networks to model the multi-environment I/O mapping was investigated.

Figure 6.4 shows the results of tests with random placement trained networks, each comprised differing numbers of hidden layer nodes, that were trained with the 'multi-environment' data set. The test data set comprised data extracted at random from environments 1 and 6, across the force application range.

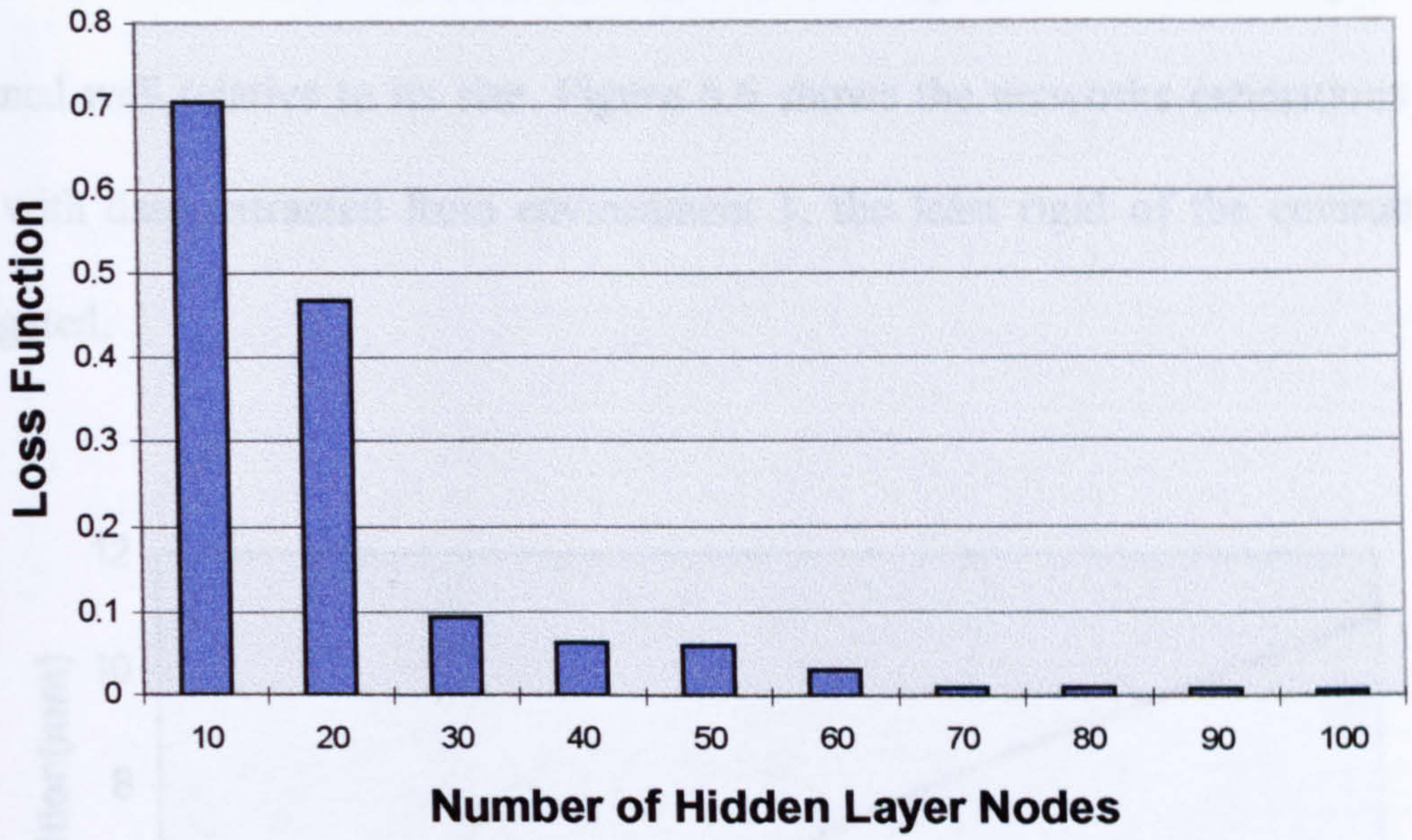


Figure 6.4 Loss functions for networks trained using random placement with data extracted from environments 1 and 6.

Figure 6.4 illustrates that network performance improved as the number of hidden layer nodes was increased. However, the increase in performance was found to be minimal for networks that comprised more than 70 hidden layer nodes. Akaikes Final Prediction Error (AFPE) was used as a measure of network parsimony. Figure 6.5 shows AFPE for the same networks.

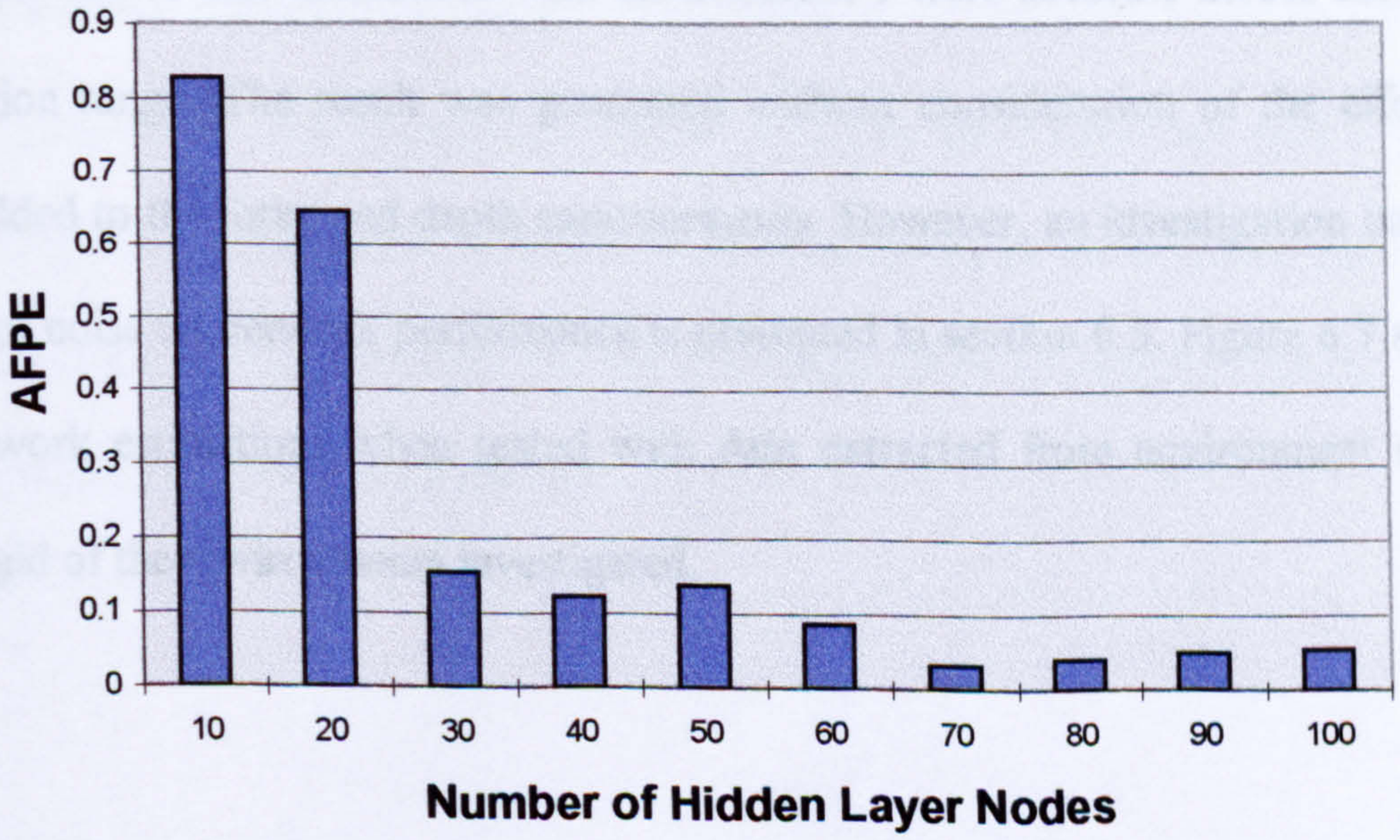


Figure 6.5 AFPE for networks trained using random placement.

The AFPE result shows that the network that comprised 70 hidden layer nodes performed well relative to its size. Figure 6.6 shows the networks estimations when tested with data extracted from environment 1, the least rigid of the environments investigated.

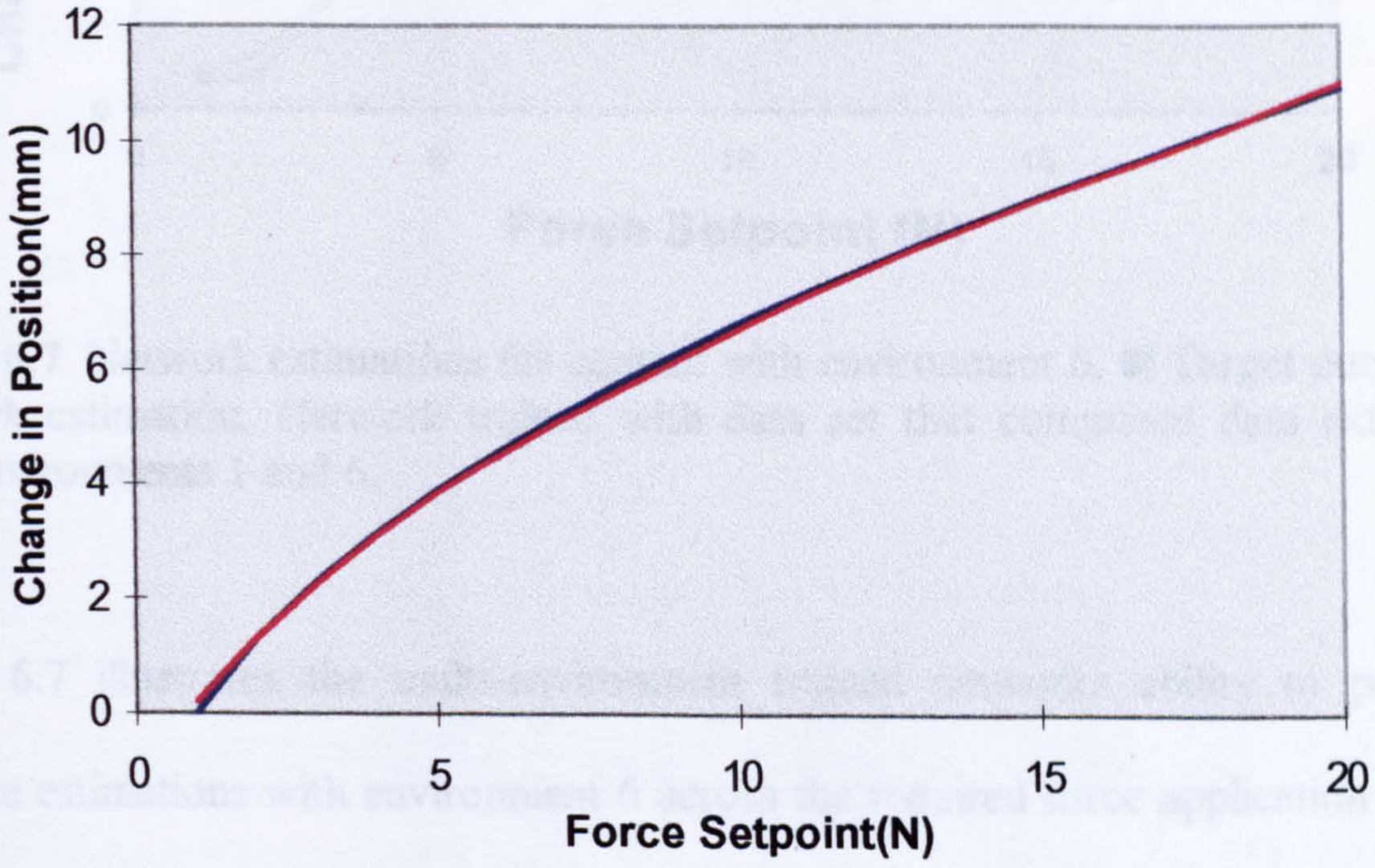


Figure 6.6 Network estimations for contact with environment 1. ■ Target Output ■ Network estimation. Network trained with data set that comprised data extracted from environments 1 and 6.

Figure 6.6 shows that estimations with environment 1 were accurate across the force application range. The result was generated without consideration of the effect of noise added to the force and depth measurements. However, an investigation into the effects of noise on network performance is presented in section 6.5. Figure 6.7 shows the network estimations when tested with data extracted from environment 6, the most rigid of the environments investigated.

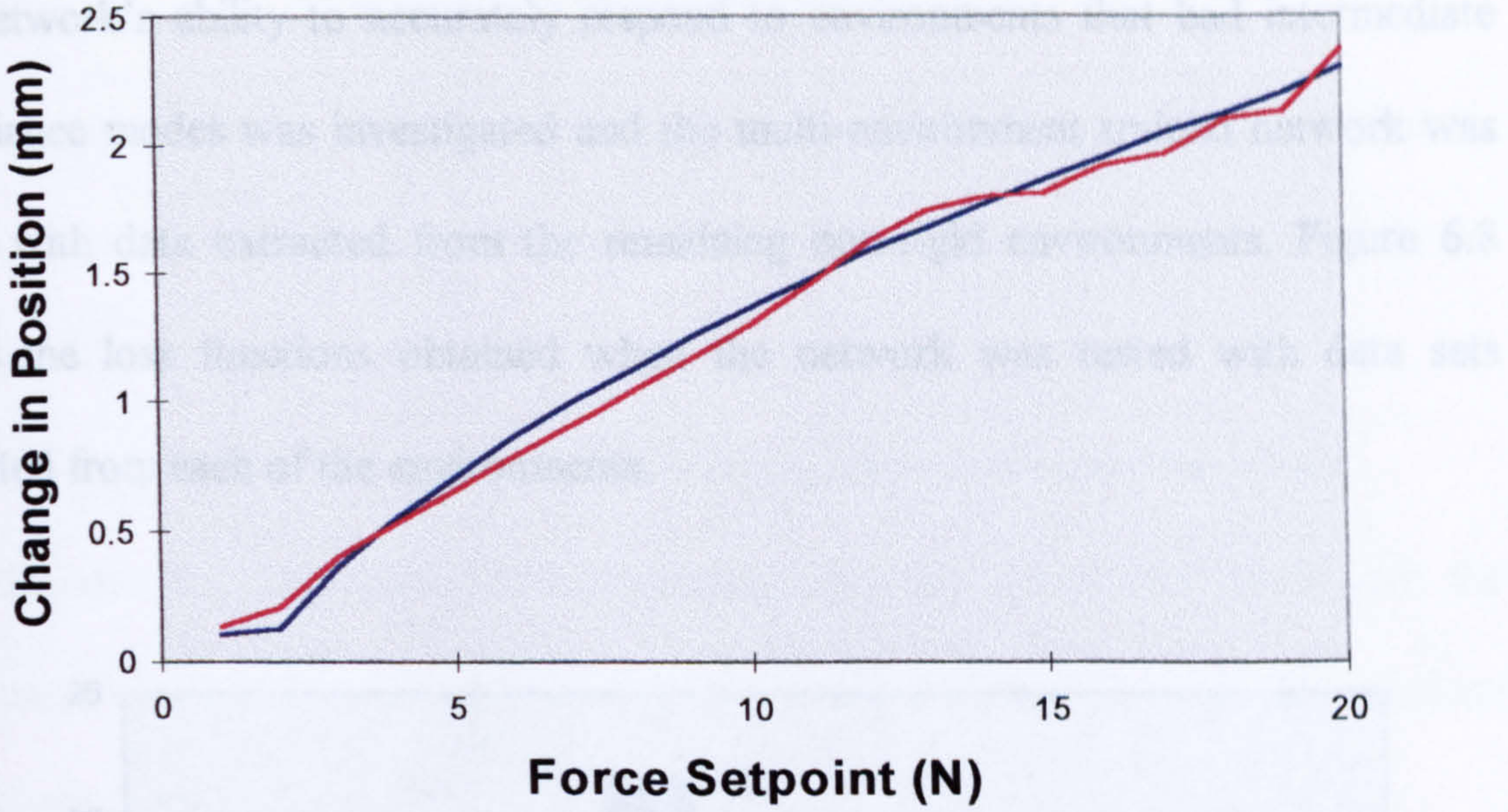


Figure 6.7 Network estimations for contact with environment 6. ■ Target output ■ Network estimation. Network trained with data set that comprised data extracted from environments 1 and 6.

Figure 6.7 illustrates the multi-environment trained networks ability to perform accurate estimations with environment 6 across the required force application range. Thus accurate estimations with both the most and least rigid environments investigated were realised with a single RBF network. The results are significant in that the network maintained the distinction between the two environments and the network training algorithm did not perform a 'best fit' through the two data sets (i.e. an averaging of the two data sets). Should the latter have occurred, loss function results of greater magnitude would have been obtained when the network was tested with each of the environments and the use of a single ANN to perform multiple input-output mappings would not be feasible.

The network's ability to accurately respond to environments that had intermediate compliance modes was investigated and the multi-environment trained network was tested with data extracted from the remaining non-rigid environments. Figure 6.8 shows the loss functions obtained when the network was tested with data sets extracted from each of the environments.

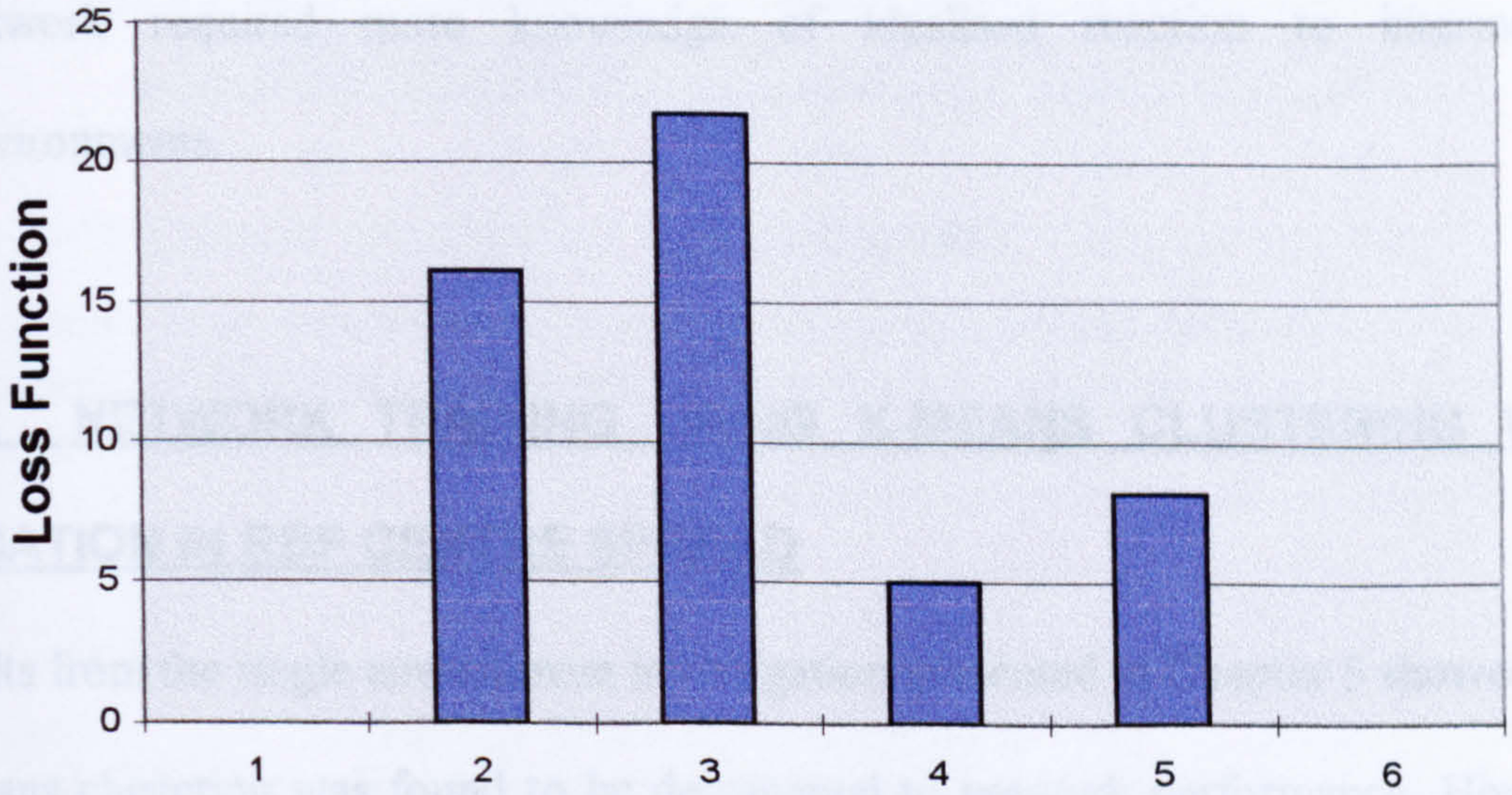


Figure 6.8 Loss functions for network testing with each of the environments. Network trained using random placement with data extracted from environments 1 and 6. Network comprised 70 hidden layer nodes.

The result illustrates that although estimations with environments represented in the multi-environment training data set were accurate, estimations with non-represented environments were poor. Thus networks that were trained using random placement with data extracted from the most and least rigid environments did not have the ability to perform accurate estimations with environments that had intermediate compliance modes.

Possible explanations for the poor performance are:

- the repeated random placement training/selection methodology produces networks that can only perform accurate estimations with environments represented in the network training data set.
- RBF networks are not capable of interpolating between compliance modes.
- RBF networks are capable of interpolating between compliance modes but the network required more knowledge of idealised reaction to intermediate environments.

6.4.2 NETWORK TRAINING USING K-MEANS CLUSTERING WITH VARIATION IN RBF CENTRE SPREAD

Results from the single environment investigation presented in Chapter 5 showed that k-means clustering was found to be detrimental to network performance. However, performance was found to improve significantly when the domain of the RBF centre initialisation was increased above the training data set input domain. Thus the effect of using k-means clustering with centre spreads that were higher multiples of the input domain was investigated.

The network training/selection methodology that was adopted for the previous investigation was again adopted, but k-means clustering of the RBF centres was performed prior to network training. Figure 6.9 shows loss functions for tests with k-means clustering trained networks that were trained using a range of centre spreads.

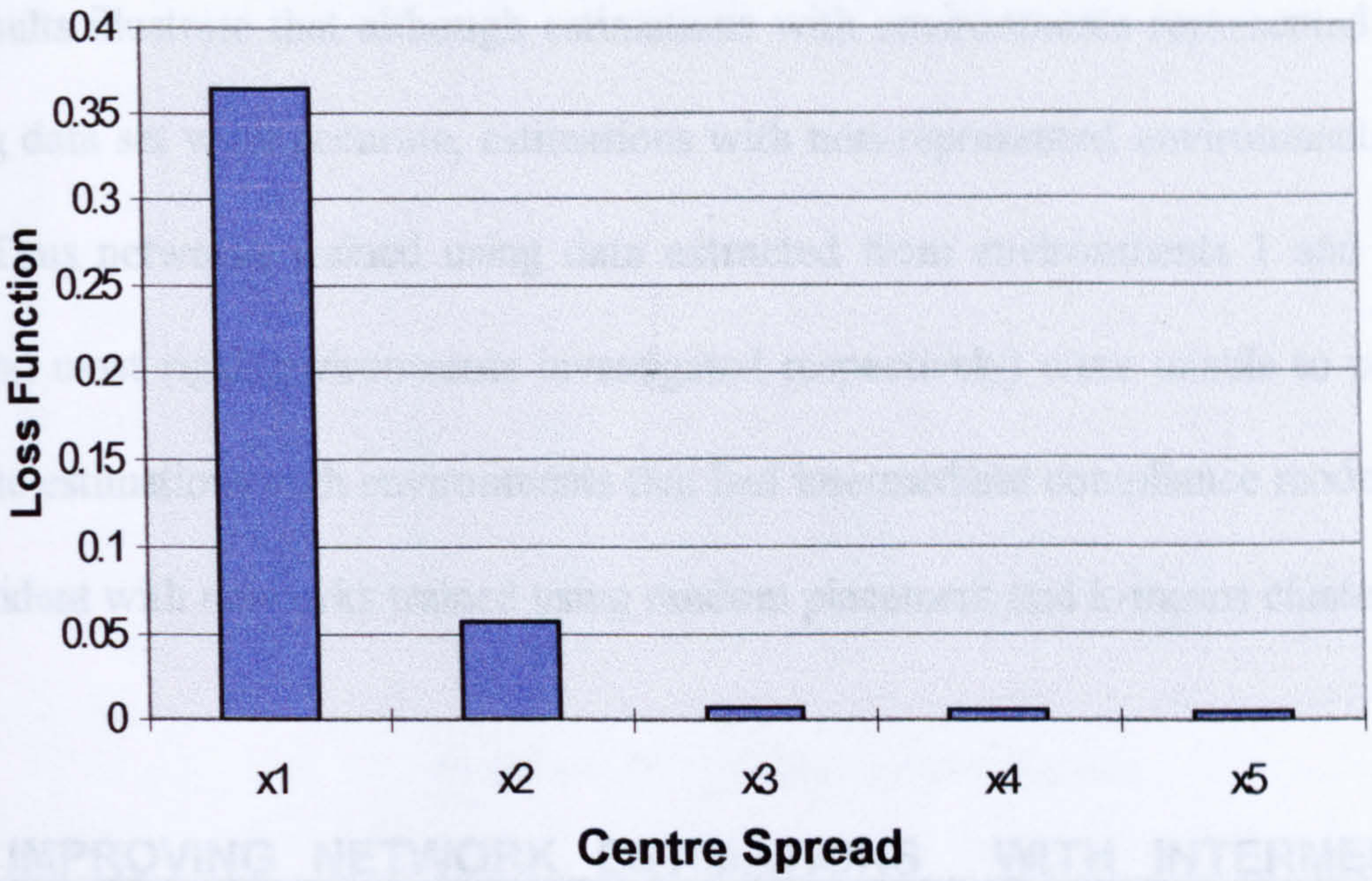


Figure 6.9 Loss functions for k-means clustering trained networks with various centre spreads. Networks tested with a data set that comprised data extracted from environments 1 and 6.

Figure 6.9 shows that network performance improved with increasing centre spread. However, the increase in performance above a centre spread of three times the input domain was minimal. Table 6.2 shows the loss functions obtained when k-means clustering trained network, that was trained using a centre spread of 5, was tested with data extracted from each of the simulated environments.

Test Environment	1	2	3	4	5	6
Loss Function	0.005	38	76.9	9.18	12.6	0.003

Table 6.2 Loss functions for k-means clustering trained network when tested with each of the environments. Network trained using a centre spread = 5.

The results illustrate that although estimations with environments represented in the training data set were accurate, estimations with non-represented environments were poor. Thus networks trained using data extracted from environments 1 and 6 (the least and most rigid environments investigated respectively) were unable to perform accurate estimations with environments that had intermediate compliance modes. This was evident with networks trained using random placement and k-means clustering.

6.4.3 IMPROVING NETWORK ESTIMATIONS WITH INTERMEDIATE ENVIRONMENTS

In an attempt to improve estimations with intermediate environments, knowledge of idealised reaction to an environment that had an intermediate compliance mode was incorporated into the multi-environment training data set. The single environment data set that was extracted from environment 3 was combined with the multi-environment training data set that already comprised data from environments 1 and 6. Environment 3 was chosen since it had a compliance mode that was approximately midway between the compliance modes of environments 1 and 6. Hardness profiles for environments 1,3, and 6 were illustrated in figure 6.2.

Networks were trained with the new multi-environment training data set using the repeated random placement training/selection procedure described in Chapter 4. Figure 6.10 shows the loss functions obtained when networks with various topologies were tested with a test data set that comprised data extracted at random from the three represented environments.

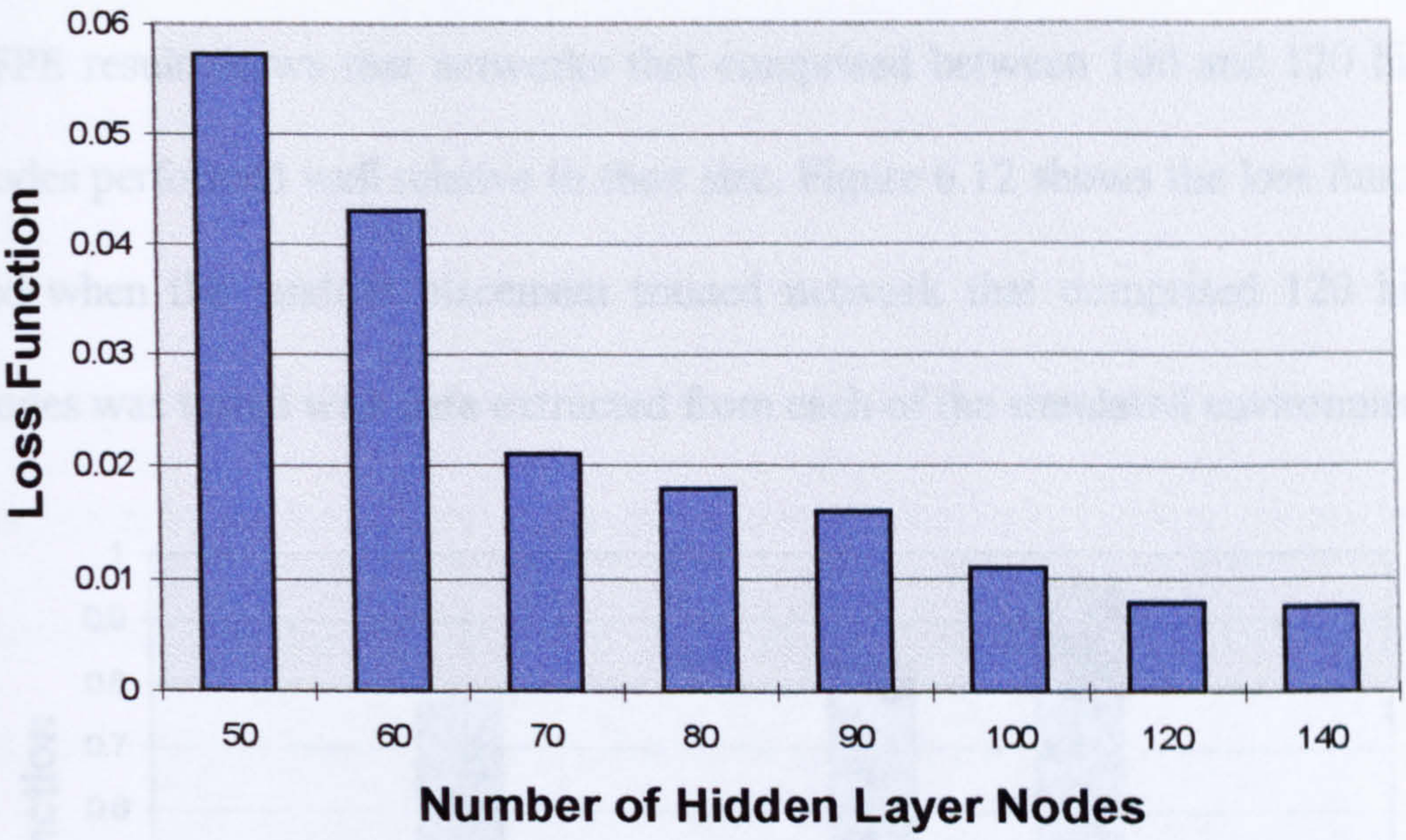


Figure 6.10 Loss functions for various network topologies trained/tested using data extracted from environments 1,3, and 6.

Figure 6.10 shows that network performance improved as the number of hidden layer nodes was increased but performance gains were minimal for network topologies that comprised more than 120 hidden layer nodes. Figure 6.11 shows AFPE results for the same networks.

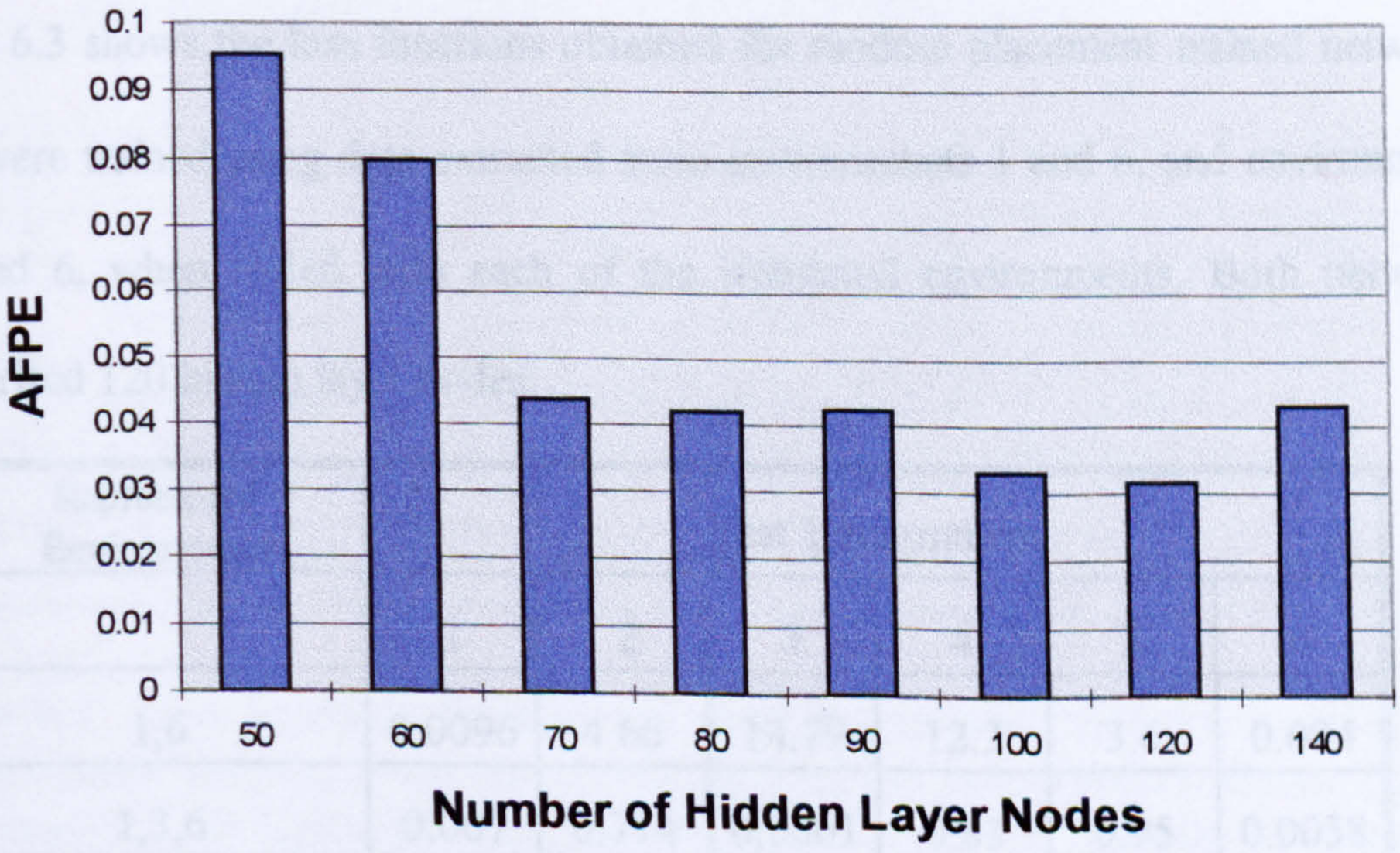


Figure 6.11 AFPE for various network topologies trained/tested using data extracted from environments 1,3, and 6.

The AFPE result shows that networks that comprised between 100 and 120 hidden layer nodes performed well relative to their size. Figure 6.12 shows the loss functions obtained when the random placement trained network that comprised 120 hidden layer nodes was tested with data extracted from each of the simulated environments.

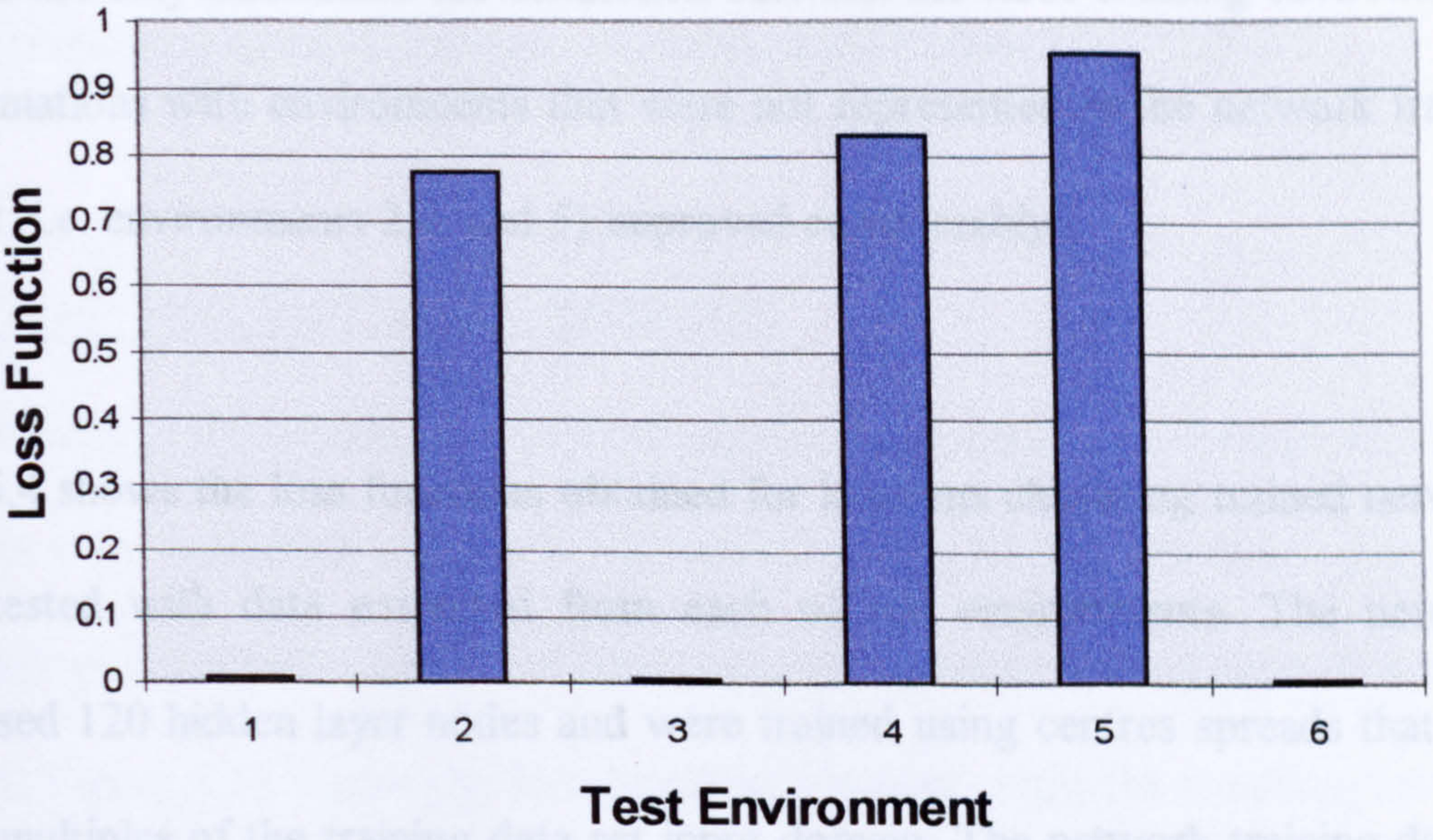


Figure 6.12 Loss functions for random placement trained network when tested with each of the environments. Network trained with data extracted from environments 1,3 and 6.

Table 6.3 shows the loss functions obtained for random placement trained networks that were trained using data extracted from environments 1 and 6, and environments 1,3,and 6, when tested with each of the simulated environments. Both networks comprised 120 hidden layer nodes.

Represented Environments	Test Environment					
	1	2	3	4	5	6
1,6	0.0096	4.66	14.79	12.3	3.6	0.004
1,3,6	0.007	0.774	0.0061	0.83	0.95	0.0058

Table 6.3 Loss functions for multi-environment trained networks when tested with each of the environments.

The results presented in figure 6.12 and table 6.3 show that the addition of data extracted from environment 3 significantly improved network estimations with environments that had intermediate compliance modes. Estimations with environments 1 and 6 remained accurate. The result is significant in that a single RBF network not only maintained the distinction between the three training environments, but estimations with environments that were not represented in the network training data set (i.e. environments 2,4, and 5) improved considerably.

Table 6.4 shows the loss functions obtained for k-means clustering trained networks when tested with data extracted from each of the environments. The networks comprised 120 hidden layer nodes and were trained using centres spreads that were higher multiples of the training data set input domain. The network training data set comprised data generated from environments 1,3, and 6.

Test Environment	Centre Spread				
	x1	x2	x3	x4	x5
Environment 1	0.57	0.122	0.039	0.032	0.004
Environment 2	0.65	0.778	2.67	2.11	4.48
Environment 3	0.08	0.044	0.02	0.021	0.004
Environment 4	0.039	0.56	3.09	2.38	6.27
Environment 5	0.126	0.456	2.31	3.44	3.98
Environment 6	0.32	0.071	0.025	0.0136	0.003

Table 6.4 Loss functions for k-means clustering trained networks, trained using various centre spreads, when tested with each of the environments. Centre spreads expressed as multiples of the training data set input domain. Represented environments: 1,3, and 6.

The results illustrate that estimations with environments represented in the network training data set generally improved with increasing centre spread but network performance with non-represented environments was found to degrade with increasing centre spread. Additionally, the performance of networks trained using k-means clustering were found to be inferior to similarly sized networks trained using random placement when tested with intermediate environments.

6.4.4 FURTHER NETWORK TRAINING

In an attempt to further improve estimations with intermediate environments, the six data sets were combined to form a new training data set and the training/selection methodologies adopted during the previous investigations were repeated with the data set. Figure 6.13 shows the loss functions obtained when the resulting networks were tested with a data set that comprised data extracted at random from each of the environments across the force application range.

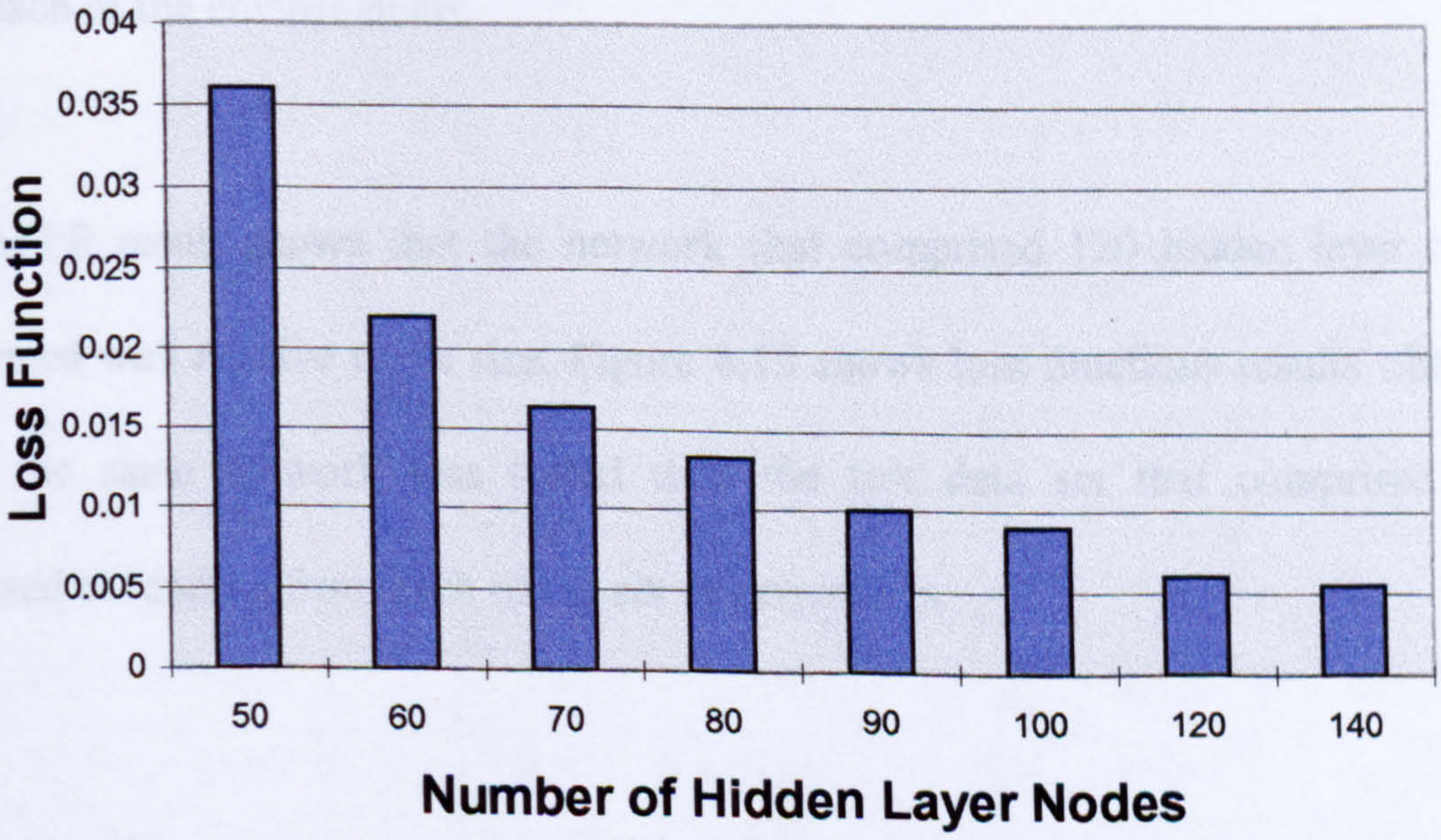


Figure 6.13 Loss function for random placement trained networks with various topologies that were trained and tested with data extracted from each of the environments.

Figure 6.13 shows that network performance improved with increasing numbers of hidden layer nodes but minimal performance gains were realised with network topologies that comprised more than 120 hidden layer nodes. Figure 6.14 shows AFPE results for the tests with the same networks.

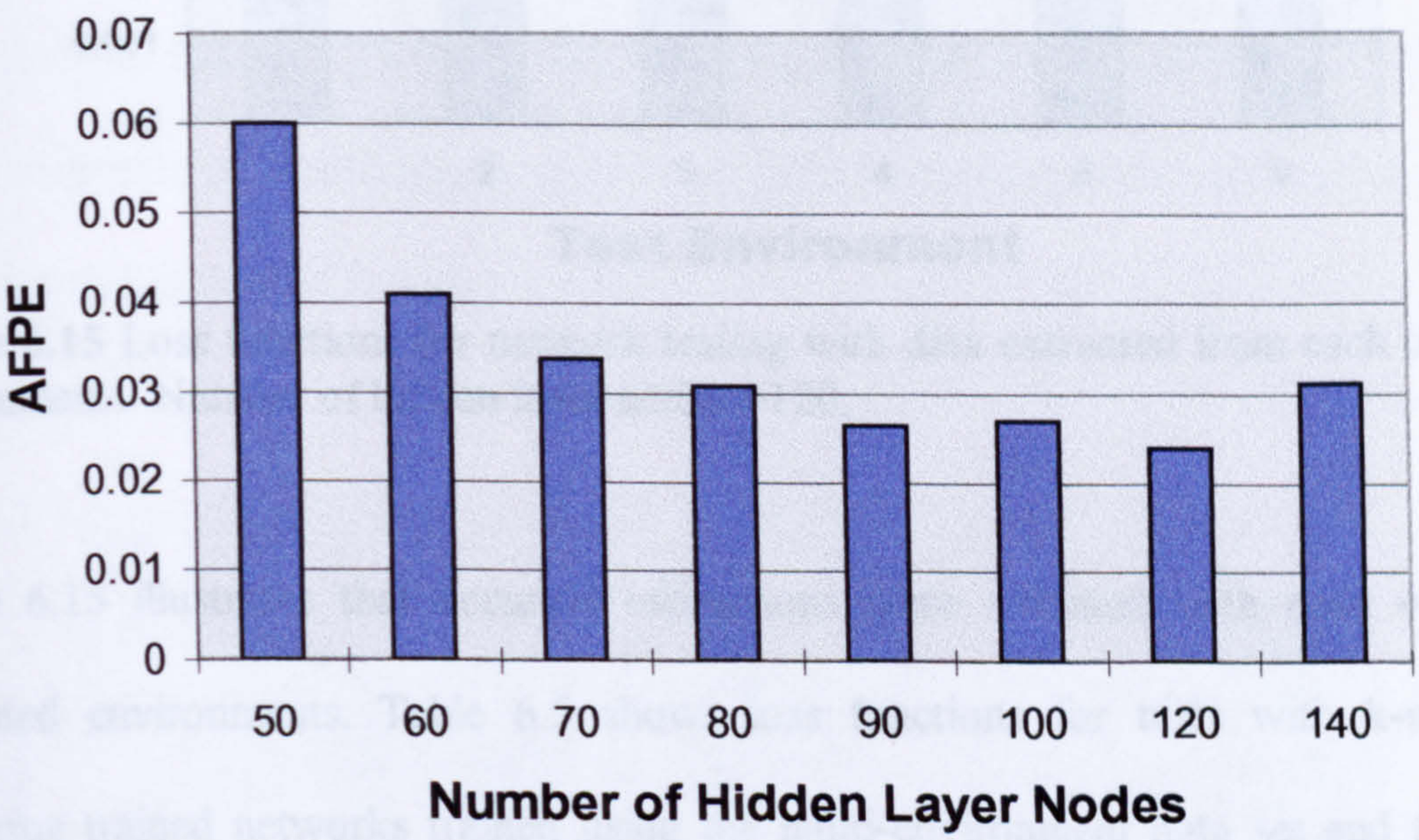


Figure 6.14 AFPE for random placement trained networks with various topologies trained and tested with a multi-environment data set that comprised data extracted from each of the environments.

The AFPE result shows that the network that comprised 120 hidden layer nodes performed well relative to its size. Figure 6.15 shows loss functions results obtained when the same network was tested with the test data set that comprised data extracted at random from each of the six environments.

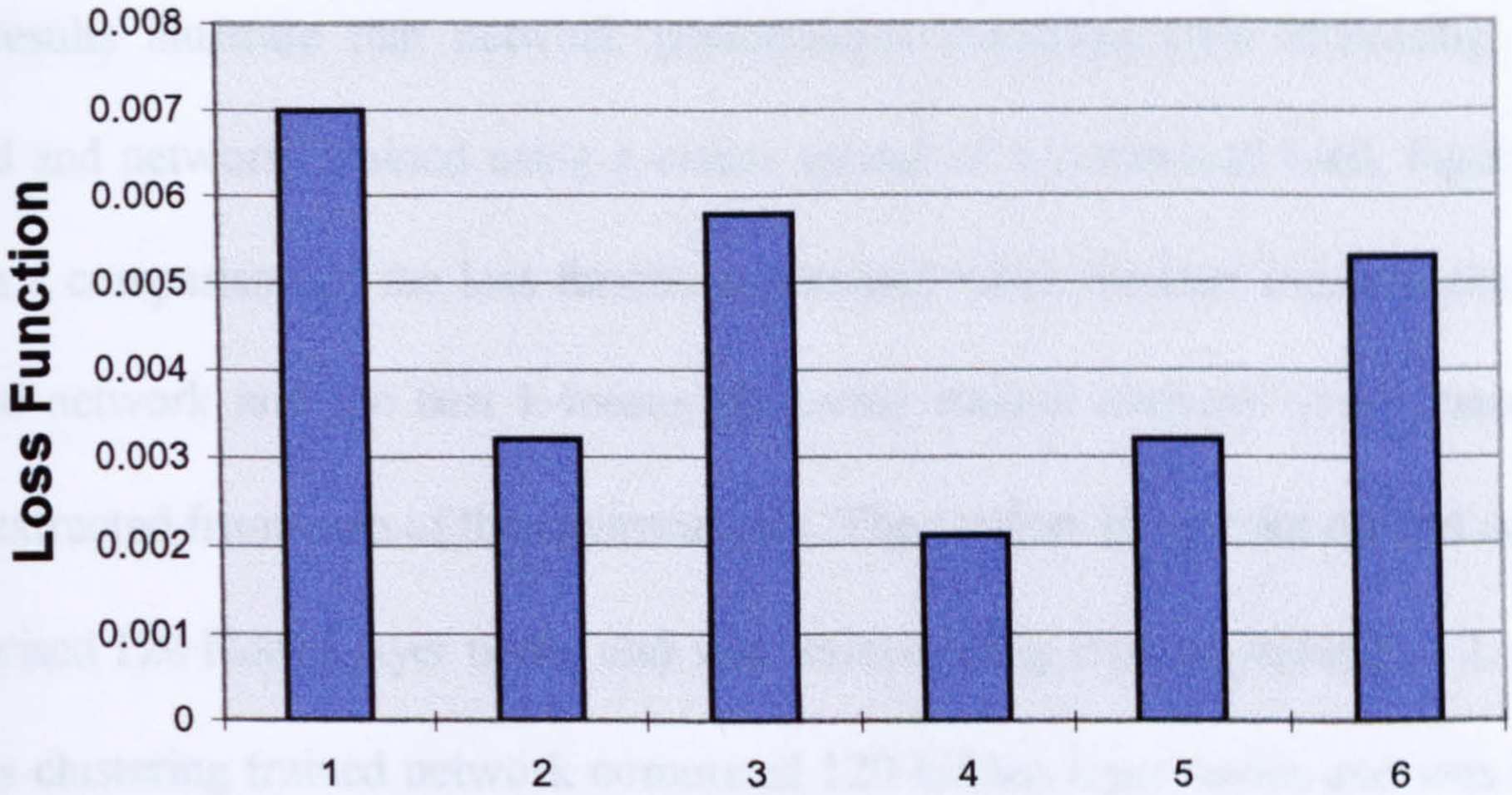


Figure 6.15 Loss functions for network testing with data extracted from each of the environments. Number of hidden layer nodes =120.

Figure 6.15 illustrates that accurate estimations were obtained with each of the simulated environments. Table 6.5 shows loss functions for tests with k-means clustering trained networks trained using the multi-environment data set and tested with data extracted from each of the six environments. The networks comprised 120 hidden layer nodes and were trained using centres spreads that were multiples of the training data set input domain.

Test Environment	Centre Spread				
	x1	x2	x3	x4	x5
Environment 1	0.33	0.066	0.012	0.0068	0.05
Environment 2	0.047	0.018	0.011	0.0065	0.0043
Environment 3	0.069	0.028	0.0098	0.0074	0.0064
Environment 4	0.1	0.013	0.016	0.0043	0.0023
Environment 5	0.16	0.058	0.021	0.0071	0.0031
Environment 6	0.25	0.025	0.01	0.055	0.006

Table 6.5 Loss functions for k-means clustering trained networks with various centre spreads. Centre spreads expressed as multiples of the training data set input domain.

The results illustrate that network performance improved with increasing centre spread and networks trained using a centre spread of 5 performed well. Figure 6.16 shows a comparison of the loss functions obtained when the best random placement trained network and the best k-means clustering trained network were tested with data extracted from each of the environments. The random placement trained network comprised 120 hidden layer nodes and was trained using a centre spread of 2. The k-means clustering trained network comprised 120 hidden layer nodes and was trained using a centre spread of 5.

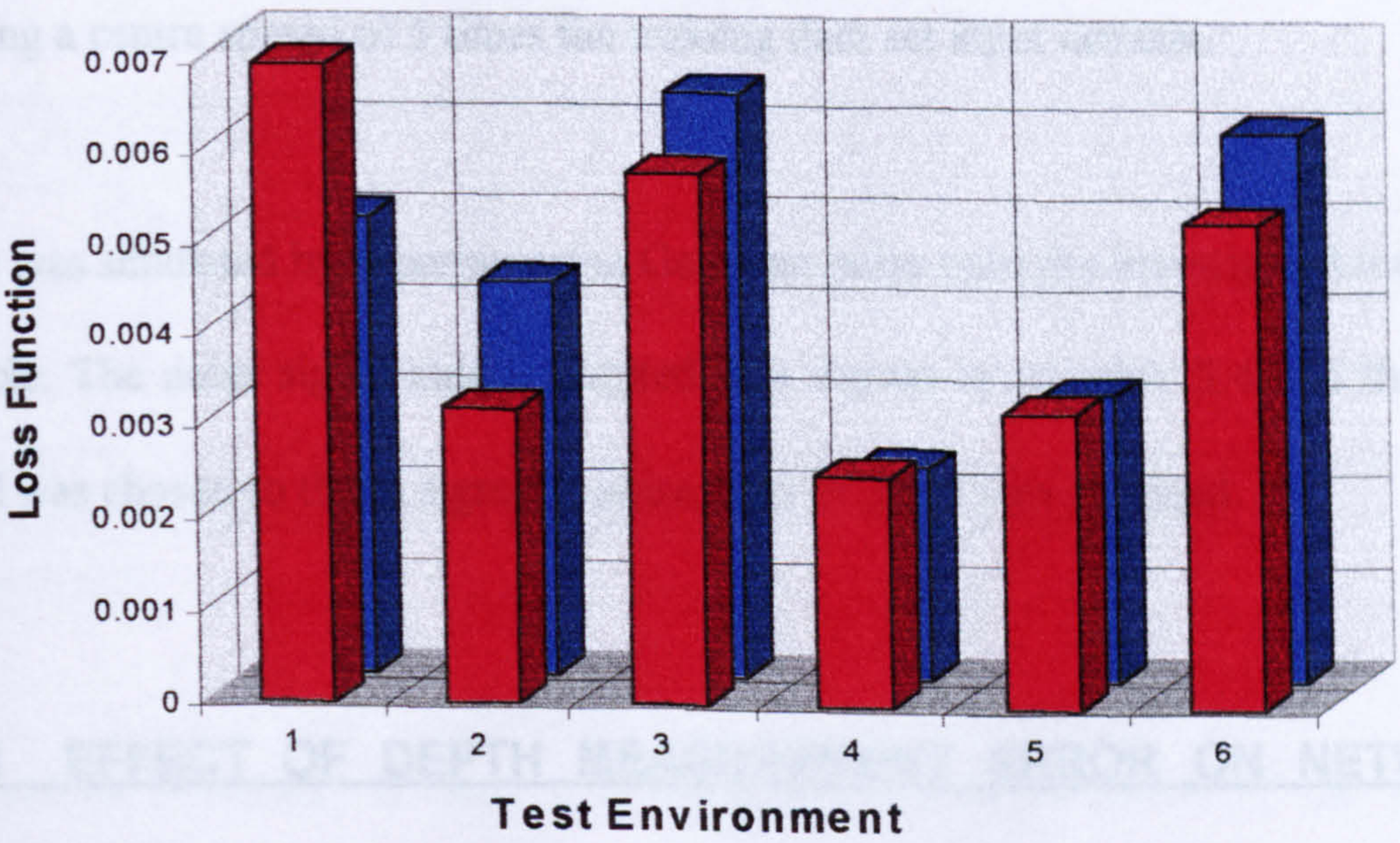


Figure 6.16 Loss functions for network testing with data extracted from each of the environments. ■ Random placement trained network. ■ K-means clustering trained network.

Figure 6.16 shows that the performance of the best k-means clustering trained network was similar to the performance of the best random placement trained and both networks performed well with each of the simulated environments.

6.5 EFFECT OF NOISE ON NETWORK PERFORMANCE

The effect on network performance of noise added to the force and depth measurements was investigated. Two networks were considered during the noise tests:

- the best performing multi-environment trained network that was trained using random placement. The network comprised 120 hidden layer nodes and was trained using an RBF centre spread of 2 times the training data set input domain.
- the best performing multi-environment trained network that was trained using k-means clustering. The network comprised 120 hidden layer nodes and was trained using a centre spread of 5 times the training data set input domain.

Noise was simulated by superimposing Gaussian noise onto the investigated measured variable. The noise signal model adopted was shown in equation 5.4 and the noise signal was chosen so that a signal to noise ratio of 20dB was obtained.

6.5.1 EFFECT OF DEPTH MEASUREMENT ERROR ON NETWORK PERFORMANCE

Errors in the depth measurement were simulated by superimposing noise onto the depth measurements in the test data sets. Table 6.6 shows the loss functions obtained when the best random placement trained network was tested with noise free data and data that had noise superimposed onto the depth measurements.

Test Environment	1	2	3	4	5	6
Noise Free	0.007	0.003	0.058	0.002	0.003	0.005
with Noise	0.021	0.007	0.016	0.006	0.004	0.004

Table 6.6 Loss functions for random placement trained network when tested with noise free data and data that had noise superimposed onto the depth measurement.

Although the presence of the noise on the depth measurement degraded performance, estimations remained satisfactory with data extracted from each of the environments. Table 6.7 shows the loss functions obtained for the same tests with the best k-means clustering trained network.

Test Environment	1	2	3	4	5	6
Noise Free	0.005	0.004	0.006	0.002	0.003	0.006
with Noise	0.038	0.006	0.016	0.016	0.005	0.007

Table 6.7 Loss functions for networks trained using k-means clustering when tested with noise free contact data and data that had noise superimposed onto the depth measurement.

The results presented in table 6.7 show that the performance of the k-means clustering trained network was degraded by the presence of noise on the depth measurement but estimations with each of the six environments were satisfactory. The performance of the k-means clustering trained network can be seen to be similar to the performance of the random placement trained network. Neither of the networks were particularly sensitive to noise added to the depth measurements.

6.5.2 EFFECT OF FORCE SENSOR NOISE ON NETWORK PERFORMANCE

Force measurement error/sensor noise was simulated by superimposing noise onto the force measurements in the test data sets. Table 6.8 shows a comparison of the loss functions obtained when the random placement trained network was tested with noise free data and data that had noise superimposed onto the force measurements.

Test Environment	1	2	3	4	5	6
Noise Free	0.007	0.003	0.005	0.002	0.003	0.005
with Noise	0.032	0.008	0.011	0.004	0.005	0.007

Table 6.8 Loss functions for random placement trained networks when tested with noise free data and data that had noise superimposed onto the force measurement.

The results presented in table 6.8 show that although network performance was degraded by the presence of noise added to the force measurements, estimations with each of the environments were satisfactory. Table 6.9 shows the loss functions obtained for the same tests with the best k-means clustering trained network.

Test Environment	1	2	3	4	5	6
Noise Free	0.005	0.004	0.006	0.002	0.003	0.006
with Noise	0.039	0.007	0.01	0.004	0.007	0.006

Table 6.9 Loss functions for networks trained using k-means clustering when tested with noise free contact data and data that had noise superimposed onto the force measurement.

The results illustrate that estimations were satisfactory in the presence of noise added to the force measurements. Additionally, the performance of the k-means clustering

trained network was again found to be similar to the performance of the random placement trained network. Neither network was sensitive to noise present on the force measurements.

6.5.3 COMBINED EFFECT OF FORCE SENSOR NOISE AND DEPTH MEASUREMENT ERROR ON NETWORK PERFORMANCE

The effect on network performance of the simultaneous action of noise added to the force and depth measurements was investigated. Table 6.10 shows the loss functions obtained when the random placement trained network was tested with data extracted from each of the environments. The results shown are for estimations with noise free data and data that had noise superimposed onto the force and depth measurements.

Test Environment	1	2	3	4	5	6
Noise Free	0.007	0.003	0.005	0.002	0.003	0.005
with Noise	0.037	0.011	0.017	0.006	0.005	0.006

Table 6.10 Loss functions for random placement trained networks when tested with noise free data and data that had noise superimposed onto the force and depth measurements.

The result illustrates that estimations were satisfactory with each of the environments in the presence of the added noise. Table 6.11 shows the loss functions obtained when the best k-means clustering trained network was tested with data extracted from each of the environments. The results presented are for estimations with noise free data and data that had noise superimposed onto the force and depth measurements.

Test Environment	1	2	3	4	5	6
Noise Free	0.0056	0.0041	0.0064	0.0023	0.0031	0.006
with Noise	0.069	0.0073	0.01	0.0046	0.0071	0.0062

Table 6.11 Loss functions for networks trained using k-means clustering when tested with data extracted from each of the environments. Network tested with noise free data and data that had noise superimposed onto the force and depth measurements.

The results presented in table 6.11 show that network performance was satisfactory in the presence of noise on the force and depth measurements and estimations with each of the environments were accurate. Thus the two networks tested were found to have similar estimating performance and degrees of noise immunity.

6.6 SINGLE DOF FORCE CONTROL SIMULATION

6.6.1 SIMULATED CONTACT WITH REPRESENTED ENVIRONMENTS

Contact with environments represented in the network training data set was investigated by simulation. The 120 hidden layer node random placement trained network was incorporated into the single degree of freedom manipulator ACSL simulation and the intelligent force control scheme's force application capability was investigated by simulation. Figures 6.17 to 6.19 show the simulated transient responses obtained when the intelligent force control scheme was commanded to apply a 10N force to environments 1, 3, and 6 respectively. Environments 1 and 6 were the least and most rigid of the simulated environments respectively. The results were generated without consideration of noise present on the force and depth measurements. The position of the contact surface was not specified to the control system prior to contact (i.e. the scheme autonomously applies forces when contact is sensed) and impact control via proximity sensing was not considered.

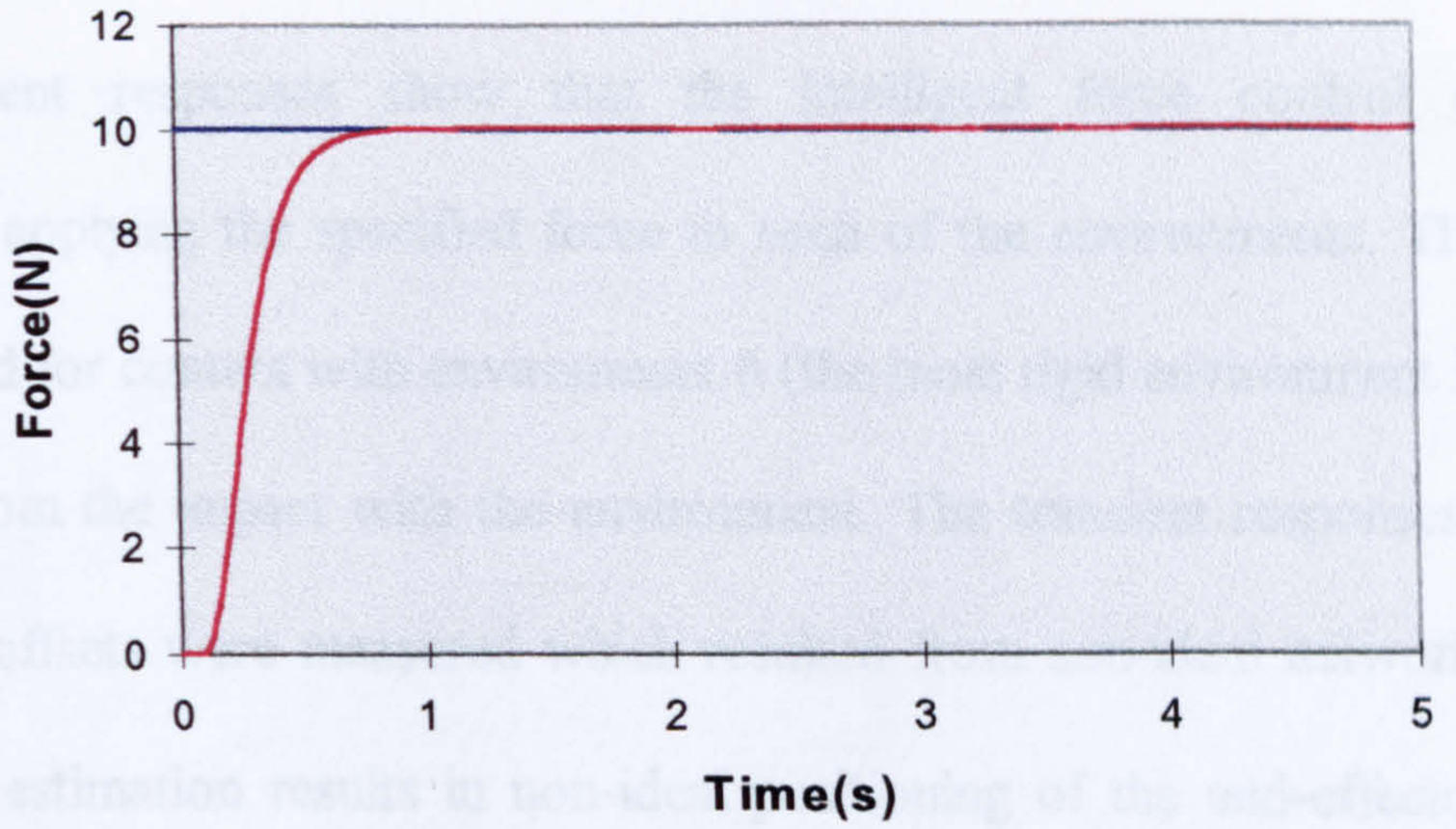


Figure 6.17 Transient response for contact with environment 1: Impact at t = 0.1 sec. ■ Force setpoint. ■ Applied force.

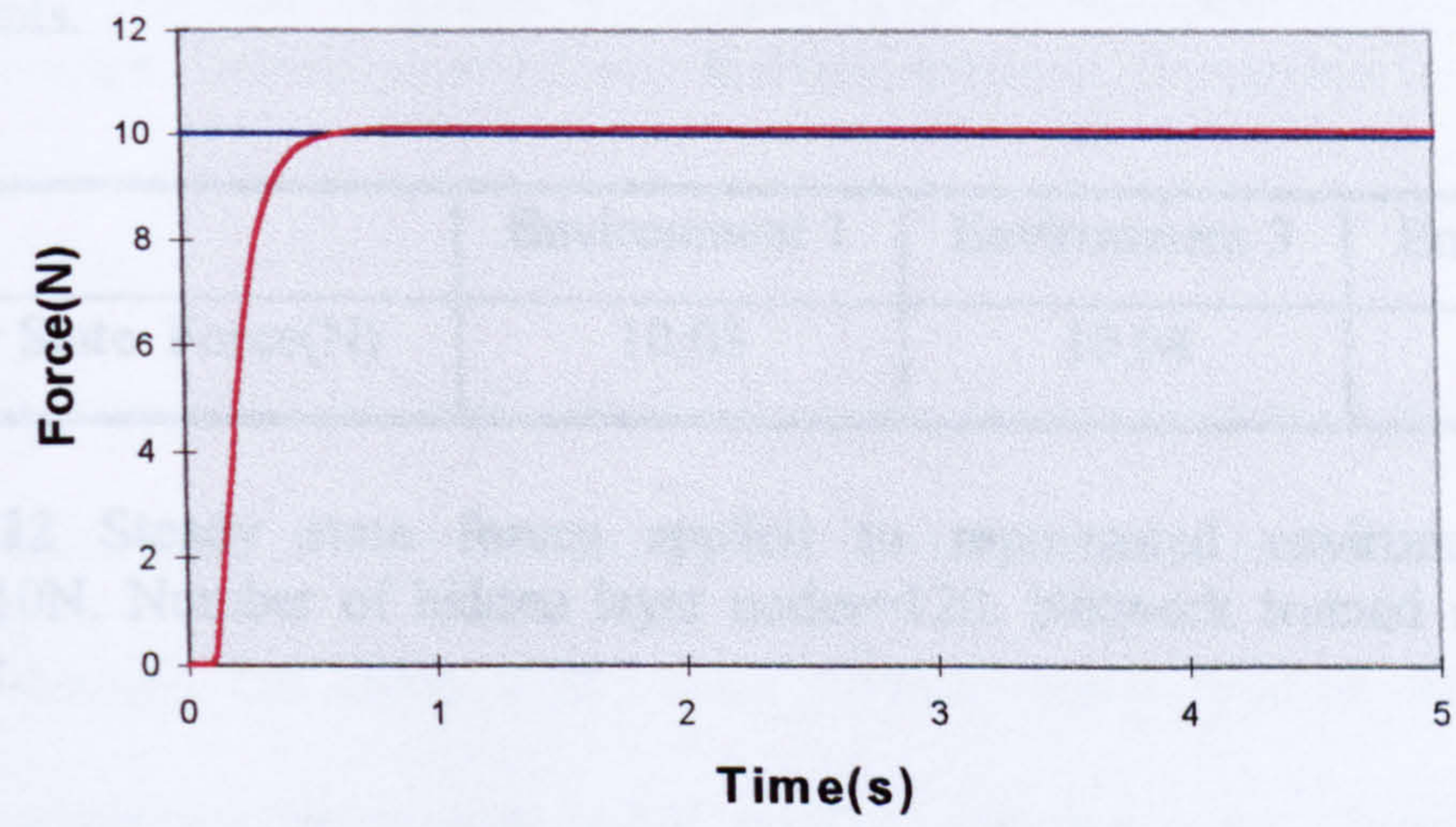


Figure 6.18 Transient response for contact with environment 3: Impact at t = 0.1 sec. ■ Force setpoint. ■ Applied force.

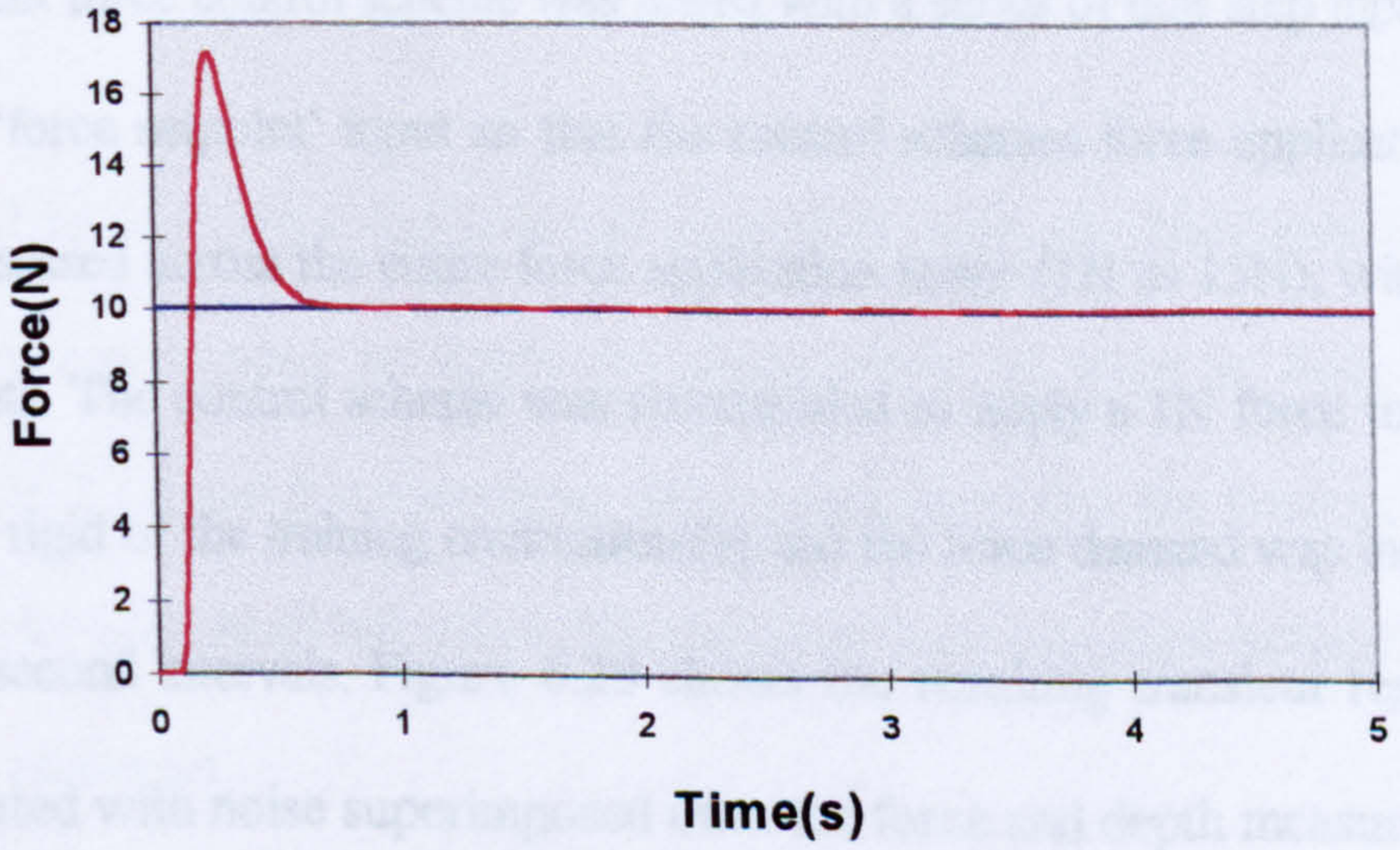


Figure 6.19 Transient response for contact with environment 6: Impact at t = 0.1 sec. ■ Force setpoint. ■ Applied force.

The transient responses show that the intelligent force control system was capable of applying the specified force to each of the environments. The overshoot experienced for contact with environment 6 (the most rigid environment investigated) resulted from the impact with the environment. The transient responses were stable but slight offsets were measured which resulted from non-ideal network estimation. Non-ideal estimation results in non-ideal positioning of the end-effector within the environment which in turn results in a non-ideal force being applied to the environment. Table 6.12 shows the steady state forces applied to each of the three environments.

	Environment 1	Environment 3	Environment 6
Steady State Force(N)	10.03	10.08	10.04

Table 6.12 Steady state forces applied to represented environments. Force setpoint=10N. Number of hidden layer nodes=120. Network trained using random placement.

6.6.2 NETWORK TESTING ACROSS THE FORCE APPLICATION RANGE

The intelligent force control scheme was tested with a series of unit step inputs applied to the ANNs 'force setpoint' input so that the control schemes force application capability could be assessed across the entire force application range (1N to 15N), with each of the environments. The control scheme was commanded to apply a 1N force to environment 1(the least rigid of the training environments) and the force demand was increased in 1N steps at 5 second intervals. Figure 6.20 shows the resulting transient response which was generated with noise superimposed onto the force and depth measurements.

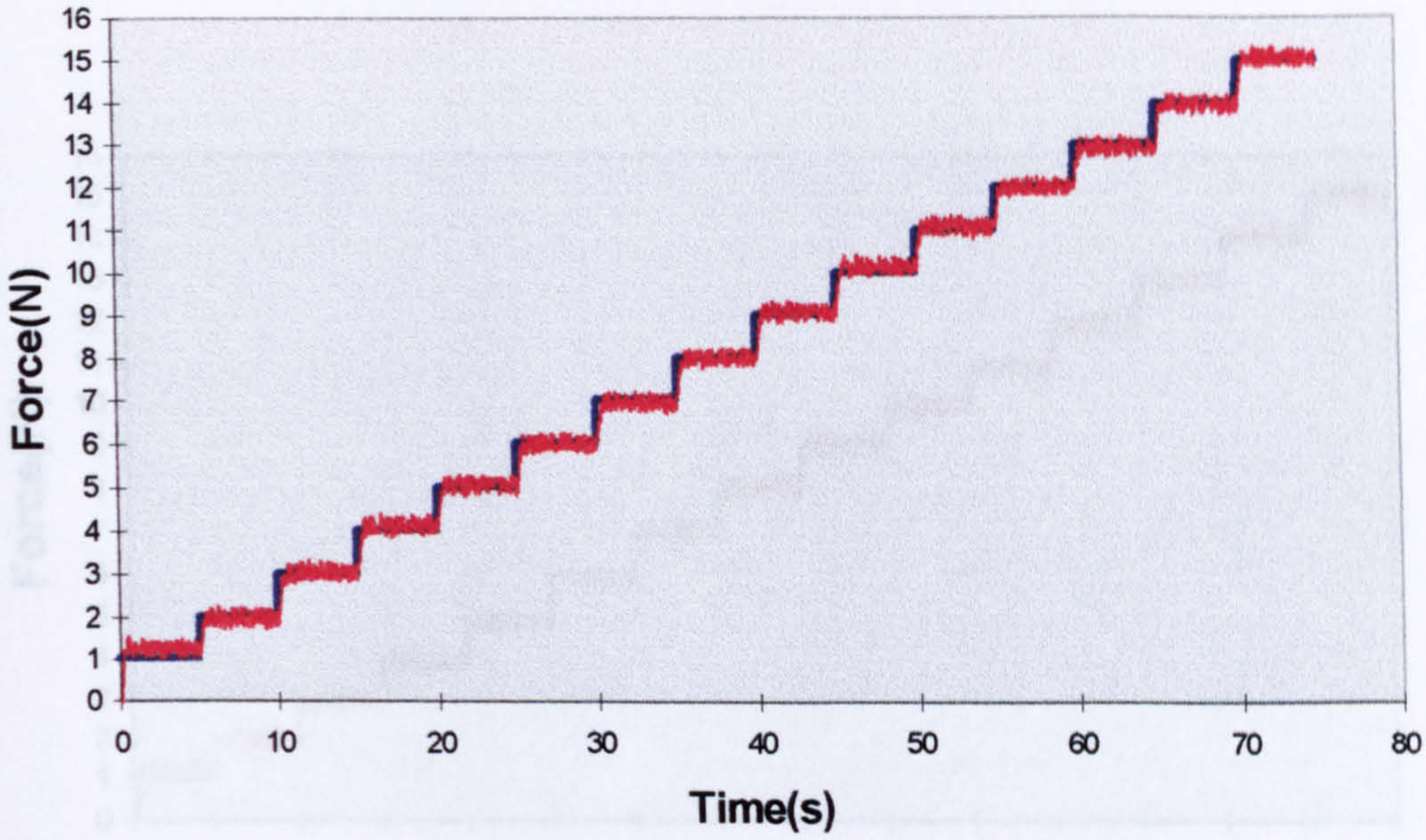


Figure 6.20 Transient response for contact with environment 1 across the force application range. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

Figure 6.20 shows that the intelligent force control system recovered from the impact with the environment that occurred at t = 0.1 seconds and the control system tracked the force demand but slight steady state offsets, which resulted from non-ideal network estimations, were measured at several operating points. Small fluctuations about the setpoint resulted from the presence of noise added to the force and depth measurements.

Figures 6.21 to 6.24 show the transient responses when the intelligent force control scheme was commanded to apply the same series of steps in force demand to environments 2,3,4,and 5 respectively. The results show that the force control system tracked the force demand but offsets that resulted from non-ideal network estimations were again measured.

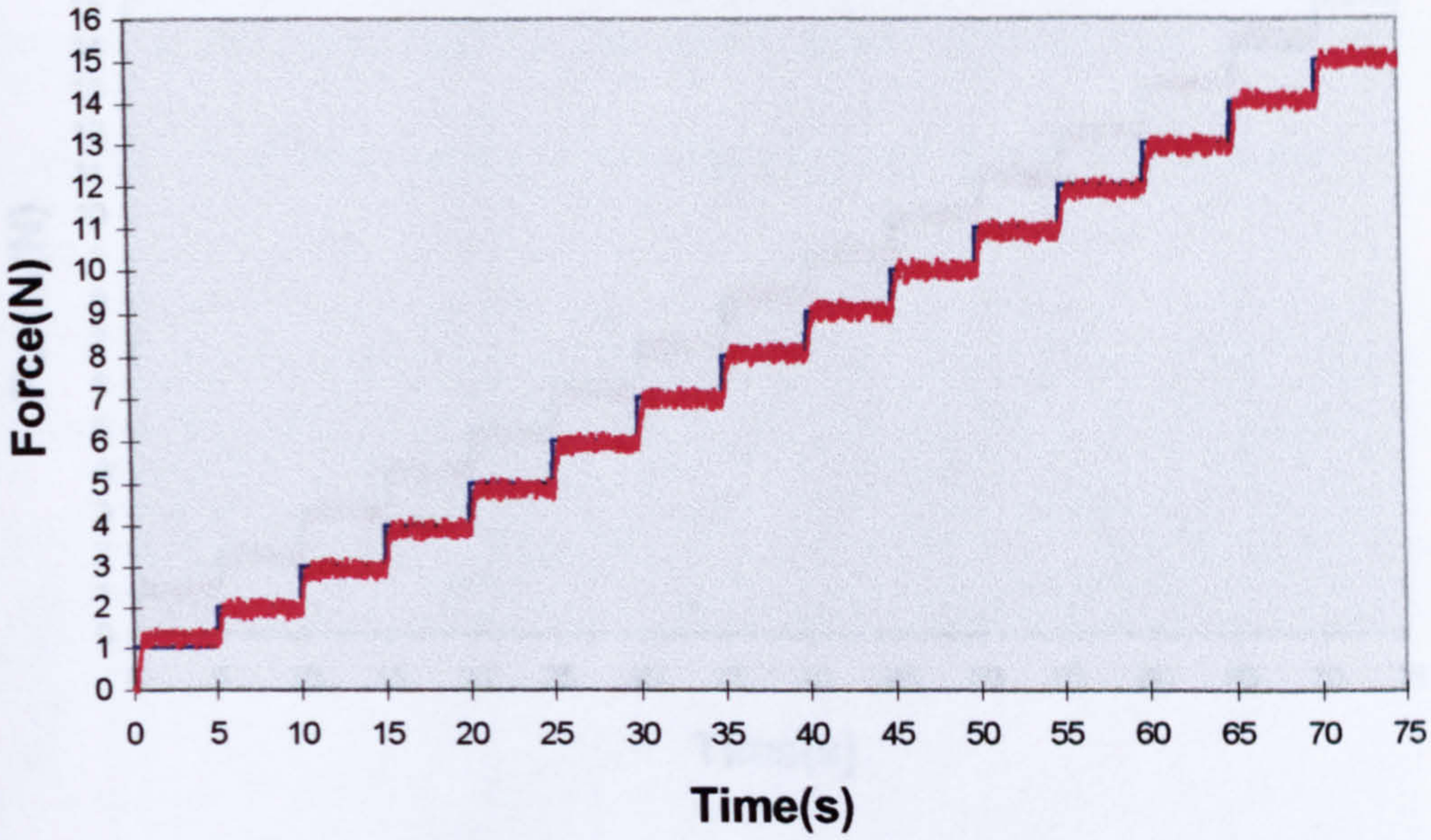


Figure 6.21 Step tests for contact with environment 2. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

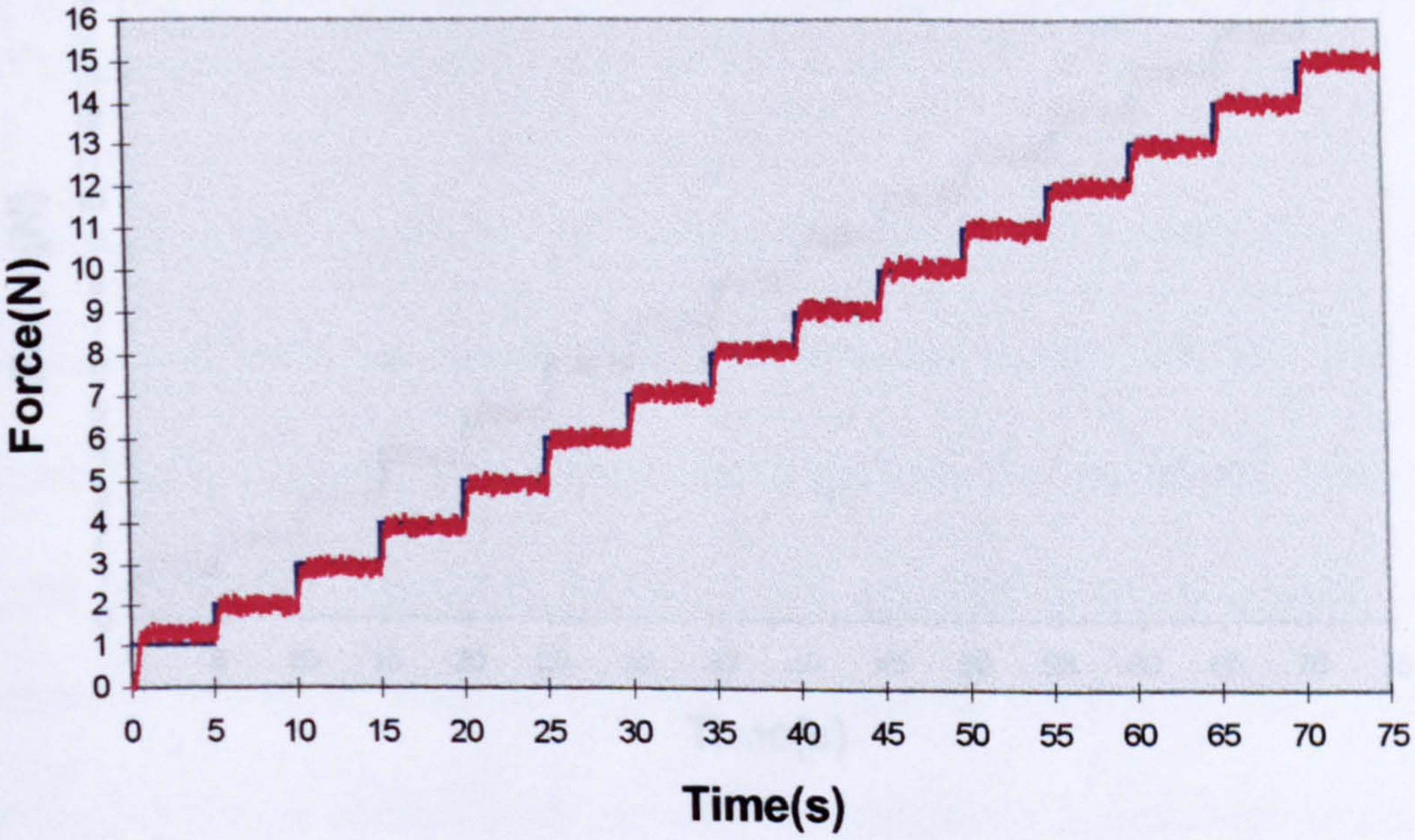


Figure 6.22 Step tests for contact with environment 3. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

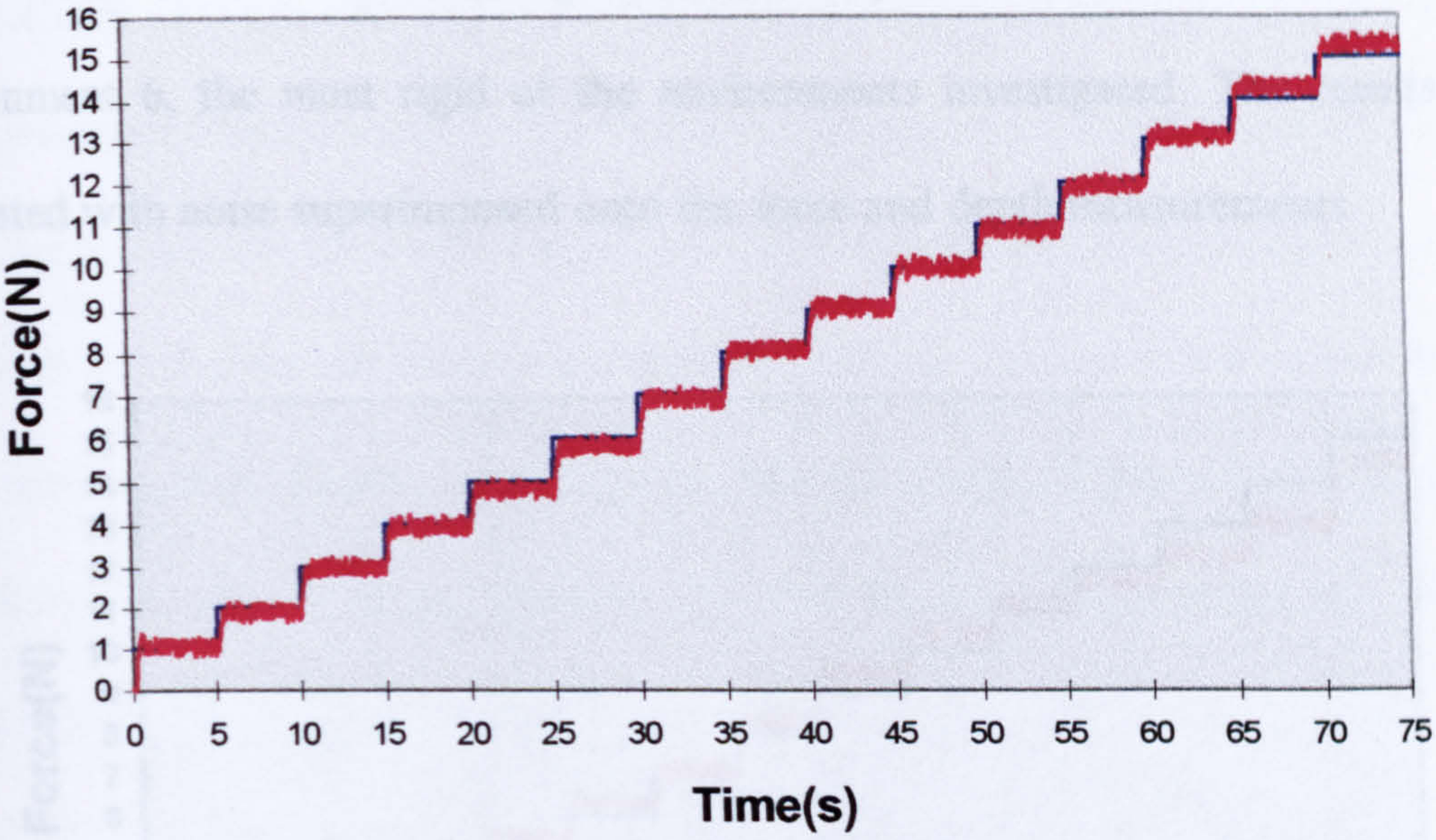


Figure 6.23 Step tests for contact with environment 4. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

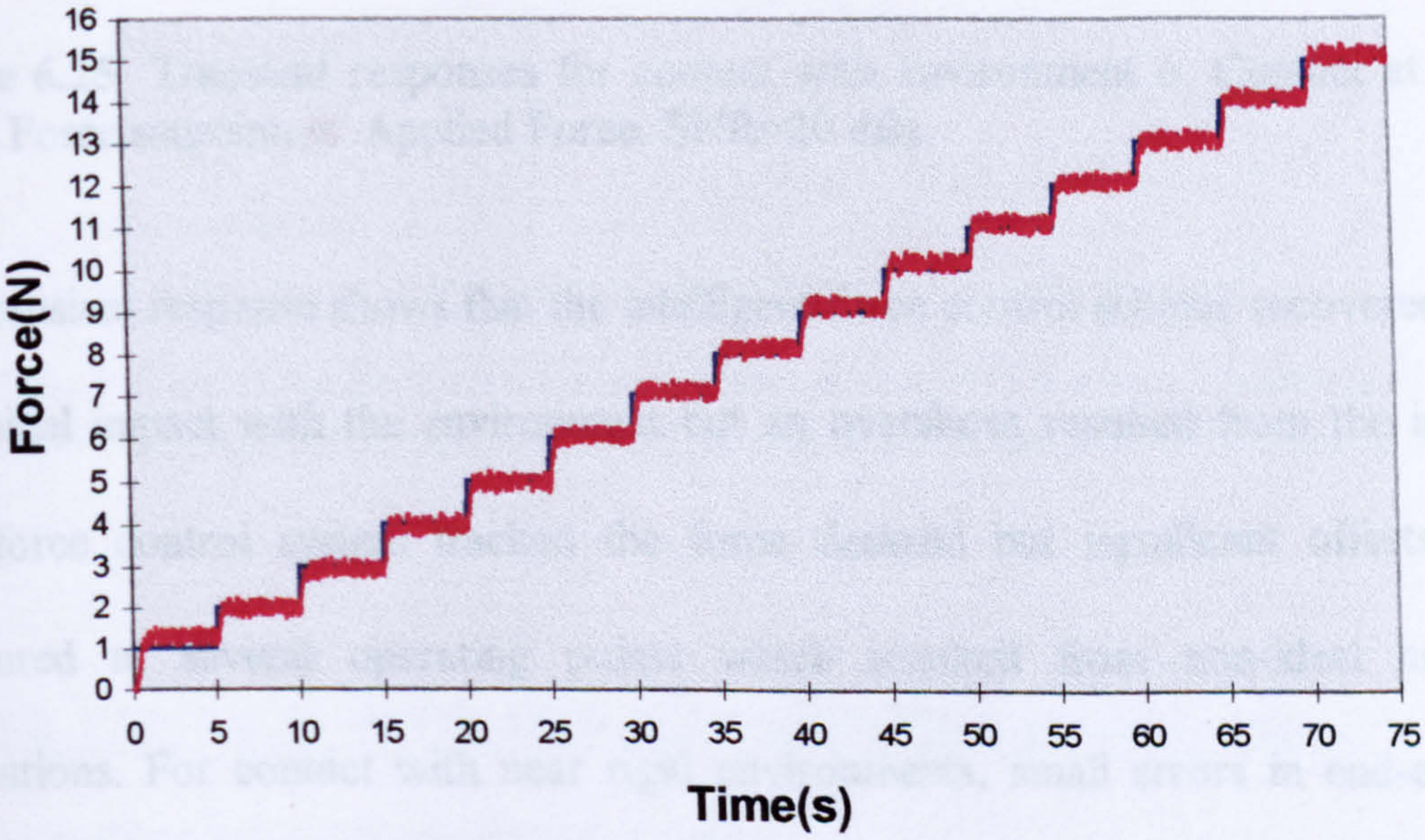


Figure 6.24 Step tests for contact with environment 5. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

Figure 6.25 shows the resulting transient response for the same tests with environment 6, the most rigid of the environments investigated. The results were generated with noise superimposed onto the force and depth measurements.

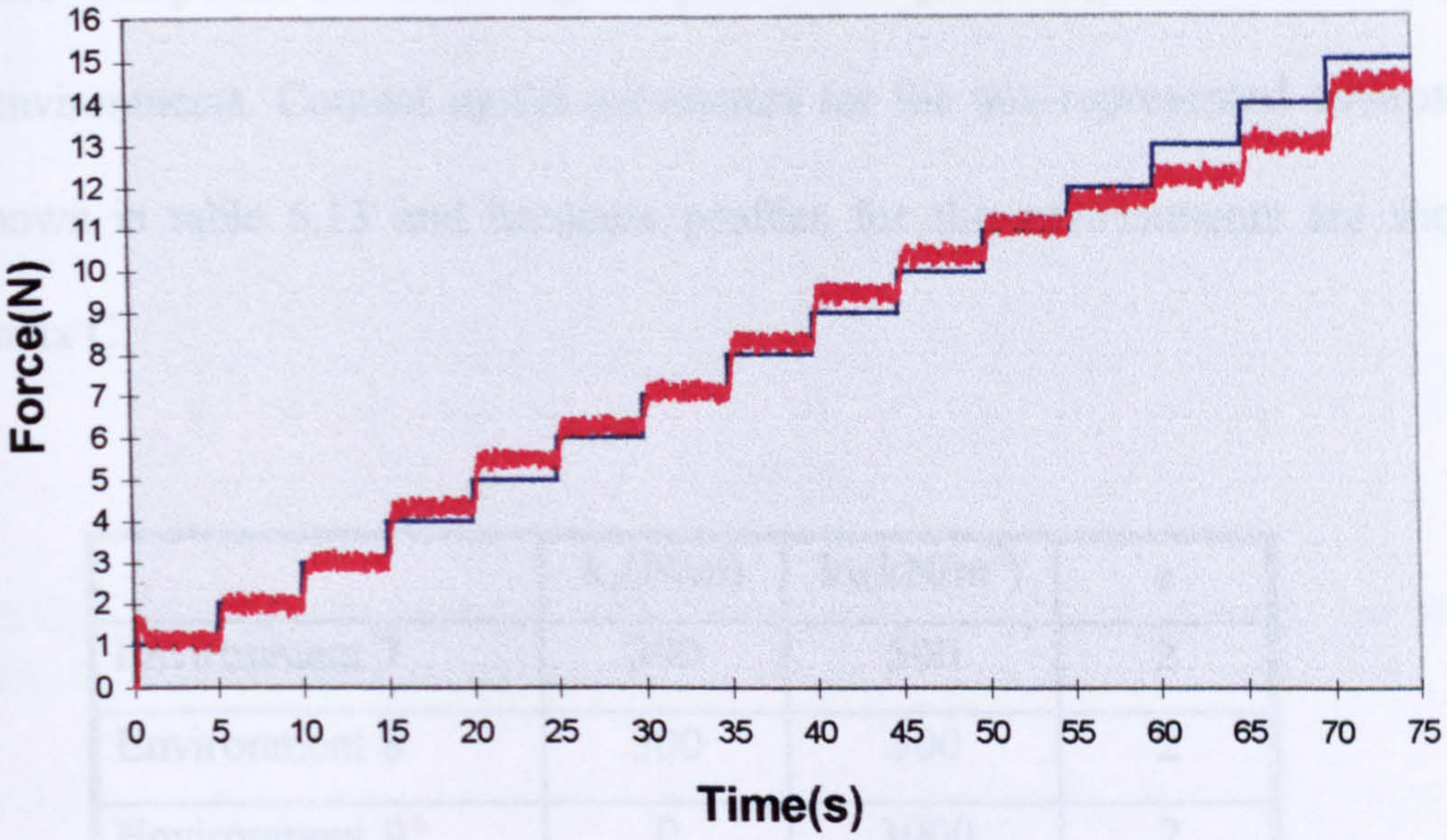


Figure 6.25 Transient responses for contact with environment 6. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied Force. SNR=20 dBs.

The transient response shows that the intelligent force control scheme recovered from the initial impact with the environment but an overshoot resulted from the impact. The force control system tracked the force demand but significant offsets were measured at several operating points which resulted from non-ideal network estimations. For contact with near rigid environments, small errors in end-effector positioning within the contact environment can result in significant force errors.

6.6.3 CONTACT WITH NON-REPRESENTED ENVIRONMENTS

Contact with environments that were not represented in the network training data set was investigated by simulation. The intelligent force control scheme's ANN does not have knowledge of idealised reaction to non-represented environments and must therefore extrapolate on knowledge acquired during training when estimating with such environments. Contact model parameters for the non-represented environments are shown in table 6.13 and hardness profiles for the environments are shown in Appendix C.

	k_1(N/m)	k_3(kN/m^2)	a
Environment 7	300	500	2
Environment 8	500	500	2
Environment 9*	0	3000	2
Environment 10*	0	50	2
Environment 11*	1000	0	2
Environment 12	2000	0	2

Table 6.13 Model parameters for non-represented environments. * indicates that the environment's data set comprises data that was outside the network training data set range.

6.6.3.1 CONTACT WITH NON-REPRESENTED ENVIRONMENTS WITH INTERMEDIATE COMPLIANCE MODES

Contact with non-represented environments that had intermediate compliance modes (i.e. compliance modes between those of environments 1 and 6) was investigated by simulation. Figures 6.26 and 6.27 show the transient responses obtained when the aforementioned step tests were performed with environments 7 and 8 respectively.

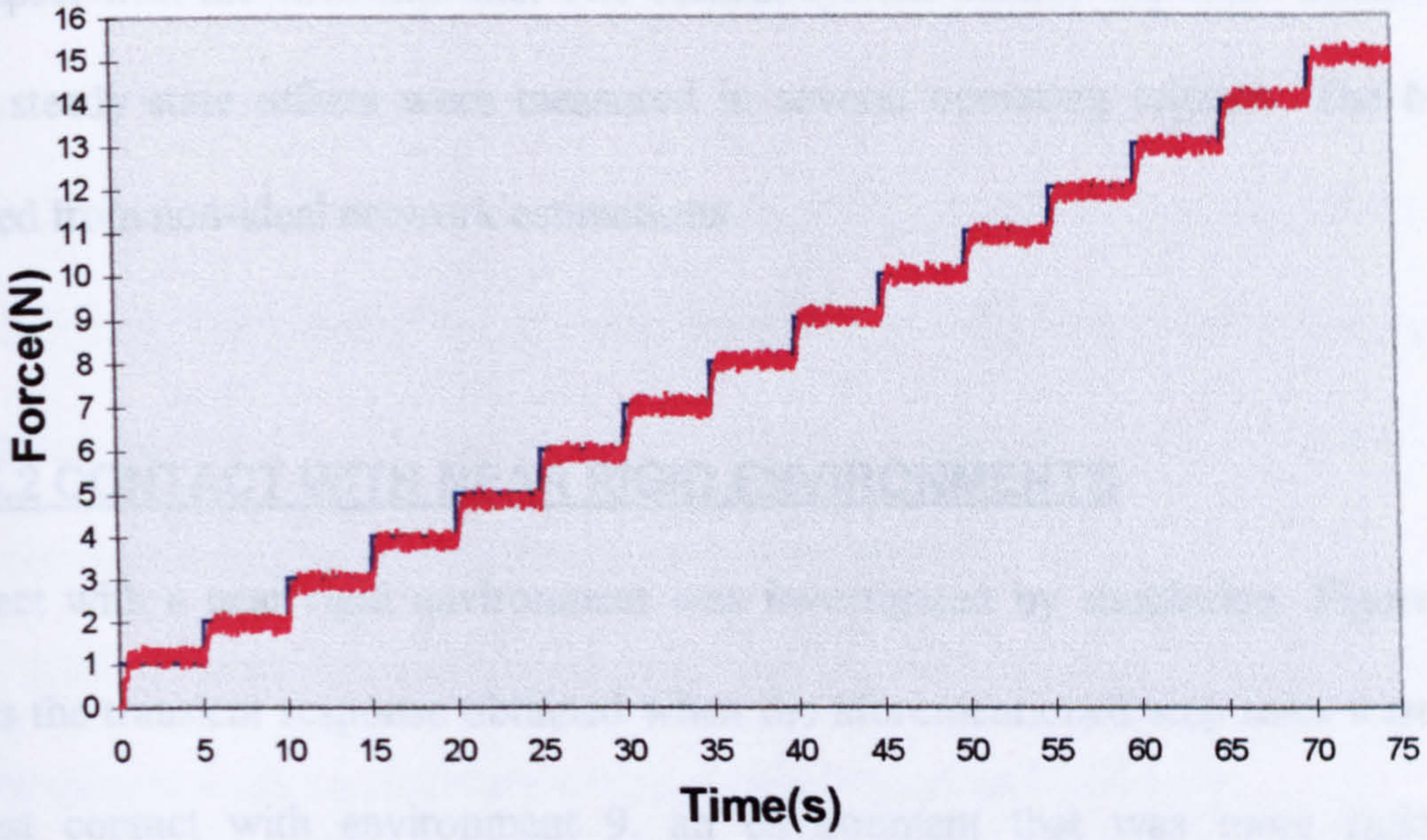


Figure 6.26 Step tests for contact with environment 7. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

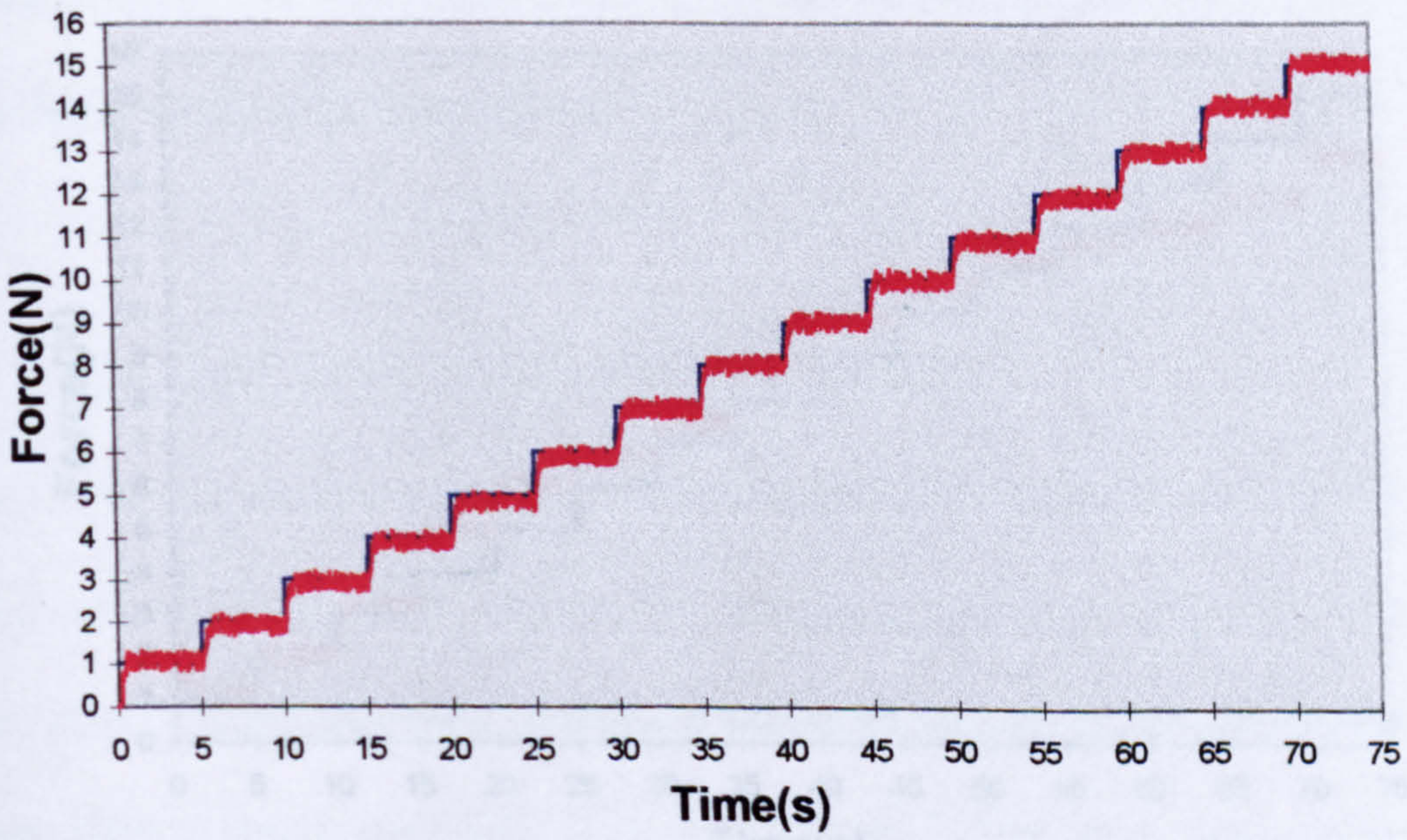


Figure 6.27 Step tests for contact with environment 8. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

The transient responses show that the intelligent force control system recovered from the impact with the environments. The control system tracked the force demand but slight steady state offsets were measured in several operating regions. The offsets resulted from non-ideal network estimations.

6.6.3.2 CONTACT WITH NEAR RIGID ENVIRONMENTS

Contact with a near rigid environment was investigated by simulation. Figure 6.28 shows the transient response obtained when the aforementioned step tests were used to test contact with environment 9, an environment that was more rigid than environment 6 (the most rigid environment represented in the network training data set).

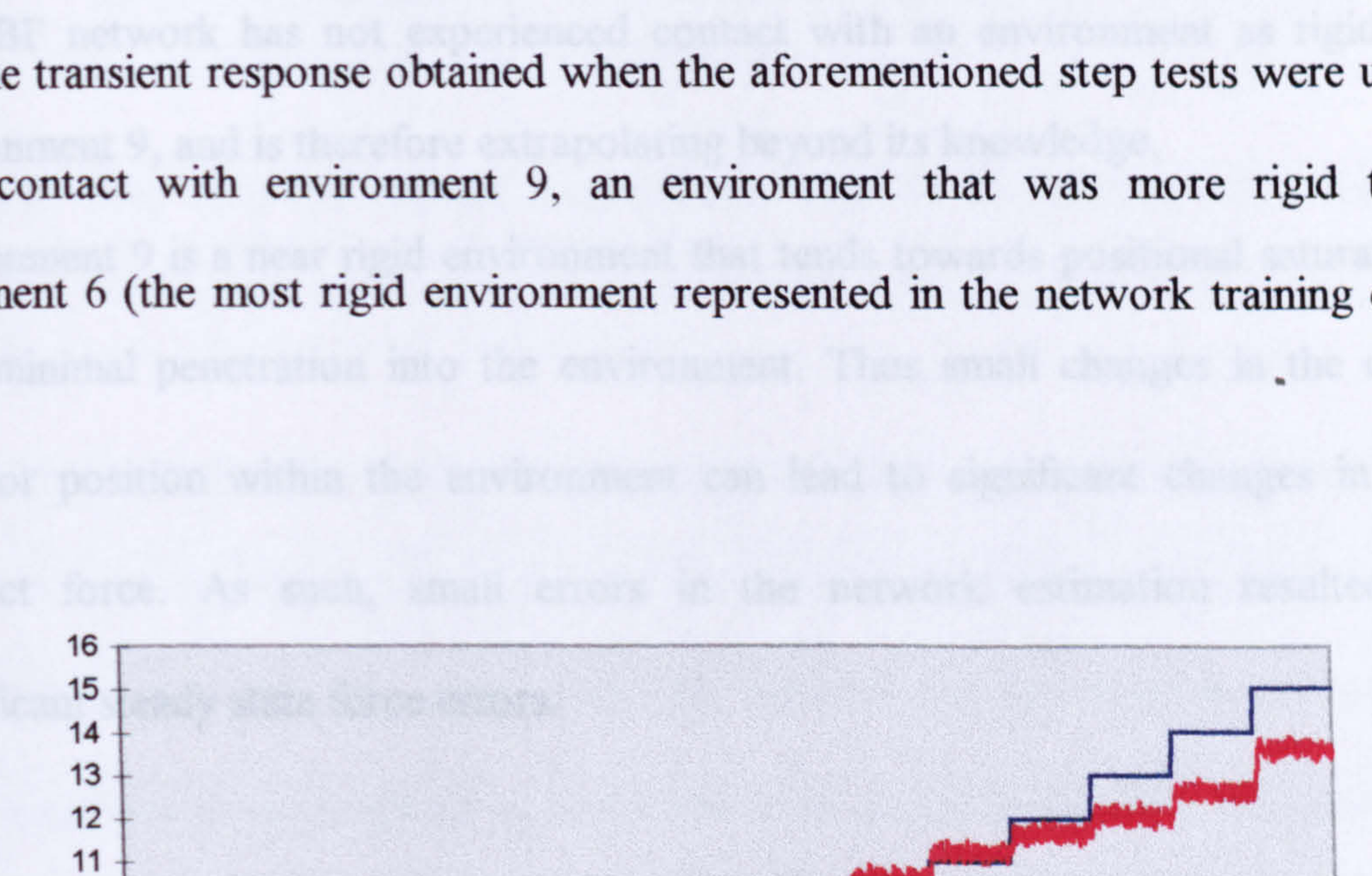


Figure 6.28 Step tests for contact with environment 9. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

The transient response presented in figure 6.28 shows that the force control system recovered from the initial impact with the environment but an overshoot resulted from the impact. Environment 9 was a near rigid environment and, as such, impact between the rigid end-effector and environment 9 resulted in contact that was greater than the force demand. The control system was incapable of achieving satisfactory contact with the environment for contact above 3N (although contact at 11N was acceptable). The poor results may be attributed to two factors, namely:

- the RBF network has not experienced contact with an environment as rigid as environment 9, and is therefore extrapolating beyond its knowledge.
- environment 9 is a near rigid environment that tends towards positional saturation with minimal penetration into the environment. Thus small changes in the end-effector position within the environment can lead to significant changes in the contact force. As such, small errors in the network estimation resulted in significant steady state force errors.

6.6.3.3 CONTACT WITH 'SOFT' ENVIRONMENTS

Contact with soft environments (i.e. very compliant environments) was investigated by simulation. Figure 6.29 shows the transient response obtained when the aforementioned step tests were performed with environment 10, an environment that was significantly more compliant than environment 1 (the least rigid environment represented in the network training data set).

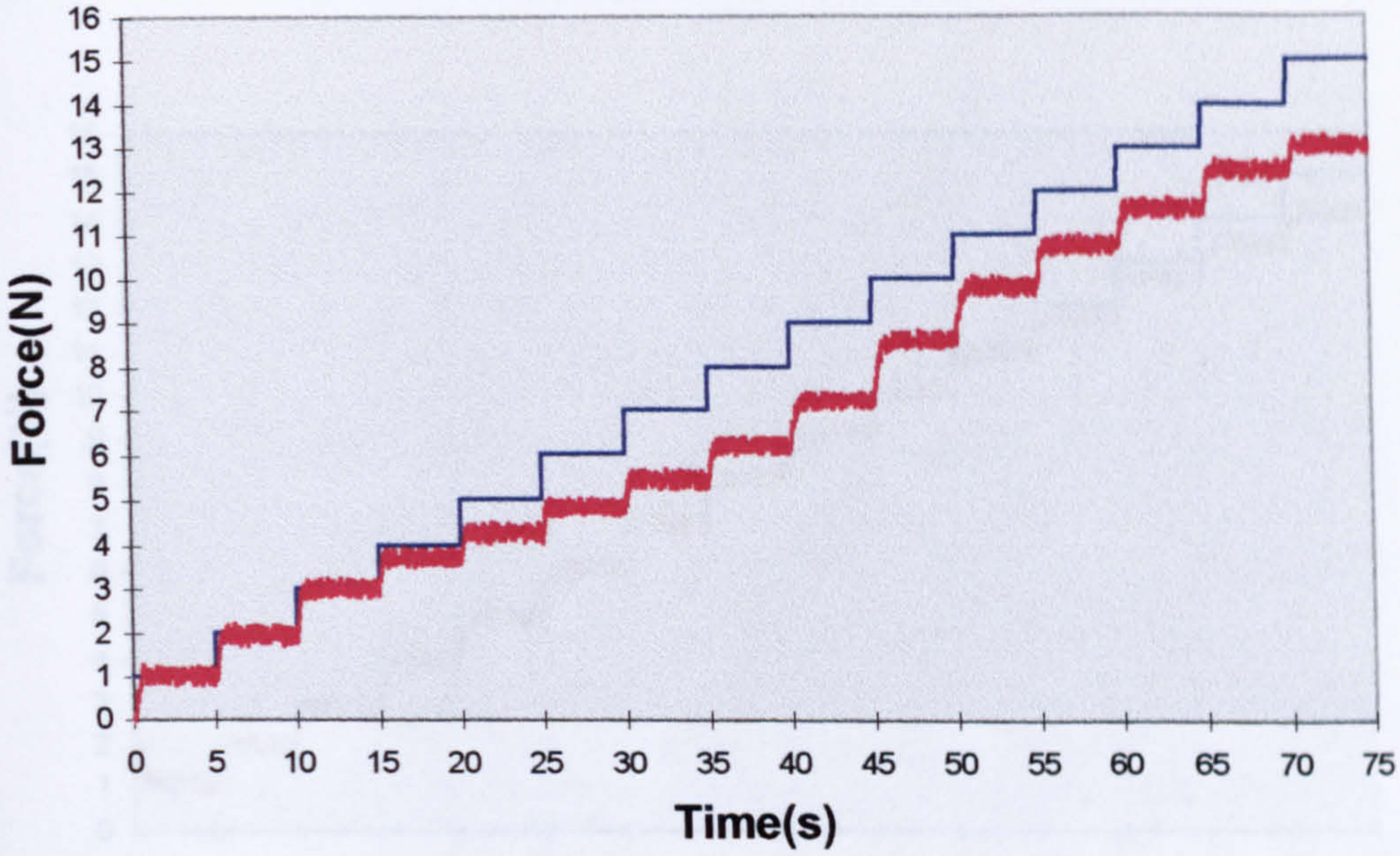


Figure 6.29 Step tests for contact with environment 10. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

Figure 6.29 shows that the intelligent force control scheme was capable of achieving satisfactory contact with environment 10 for contact up to 3N, but performance deteriorated considerably thereafter. The RBF network has not experienced contact with an environment as compliant as environment 10, and it is therefore extrapolating beyond its knowledge. The hardness profile for environment 10 (shown in Appendix C) shows significant deviation from the hardness profile for environment 1 (the least rigid of the training environments) for contact above 3N.

6.6.3.4 CONTACT WITH LINEAR ENVIRONMENTS

Contact with environments that had linear contact characteristics was investigated by simulation. Figure 6.30 shows the transient response obtained when the intelligent force control scheme experienced contact with a linear environment (environment 11).

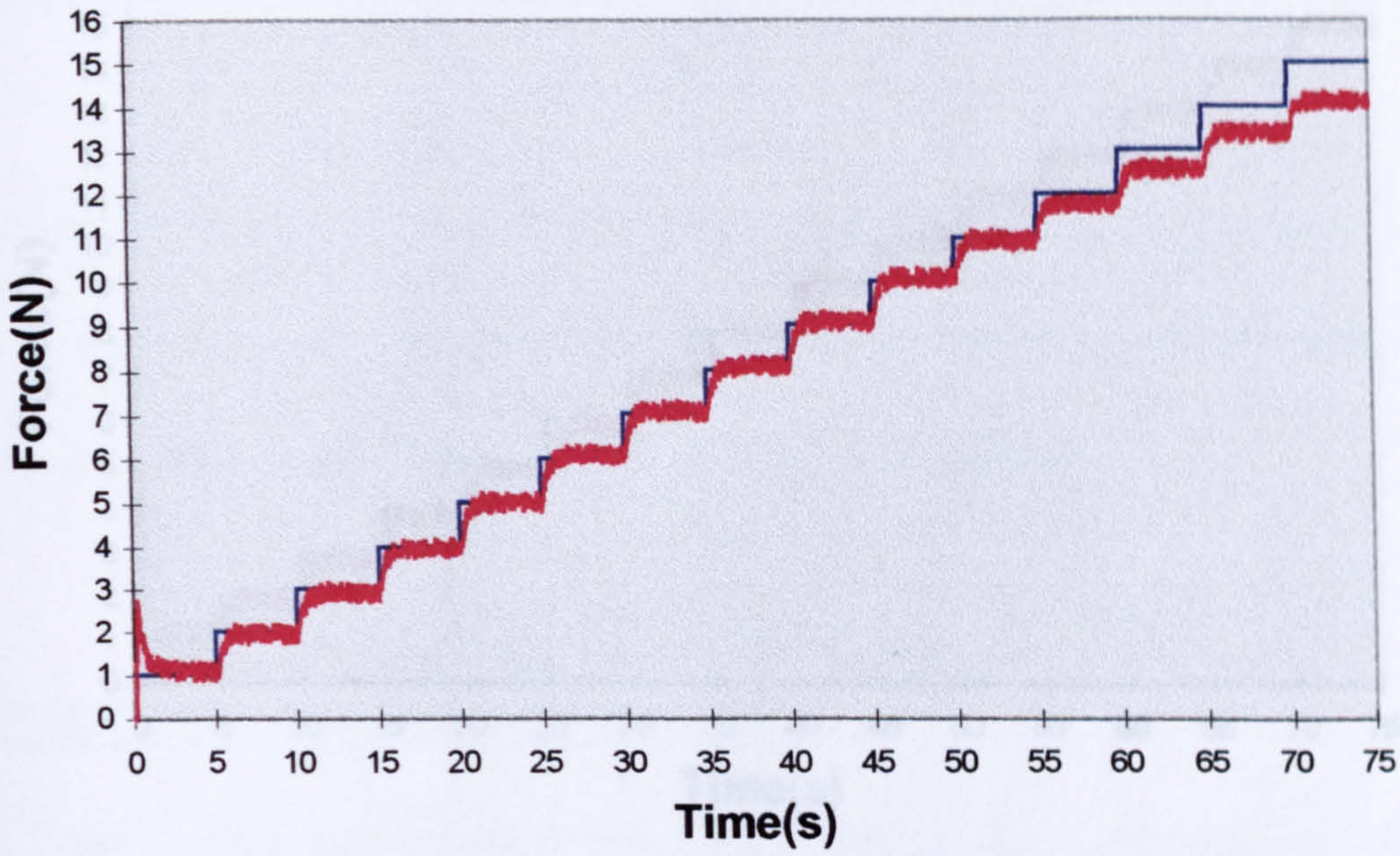


Figure 6.30 Step tests for contact with environment 11. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

Figure 6.30 shows that the intelligent force control system recovered from the initial impact with the environment but an overshoot resulted from the impact. The control system tracked the force demand for contact up to 10N, thereafter performance deteriorated. The hardness profile for environment 11 (see Appendix C) shows that contact with the environment up to 9N was within the training data range. However, the control scheme's ANN did not have knowledge of contact with the environment above 9N and was thus required to extrapolate on its knowledge. Further tests showed that the offset increased as the force demand moved the end-effector further into the environment and thus away from the network's training data range.

Figure 6.31 shows the transient response when the intelligent force control scheme experienced contact with environment 12, a linear environment that was bounded by the training data set.

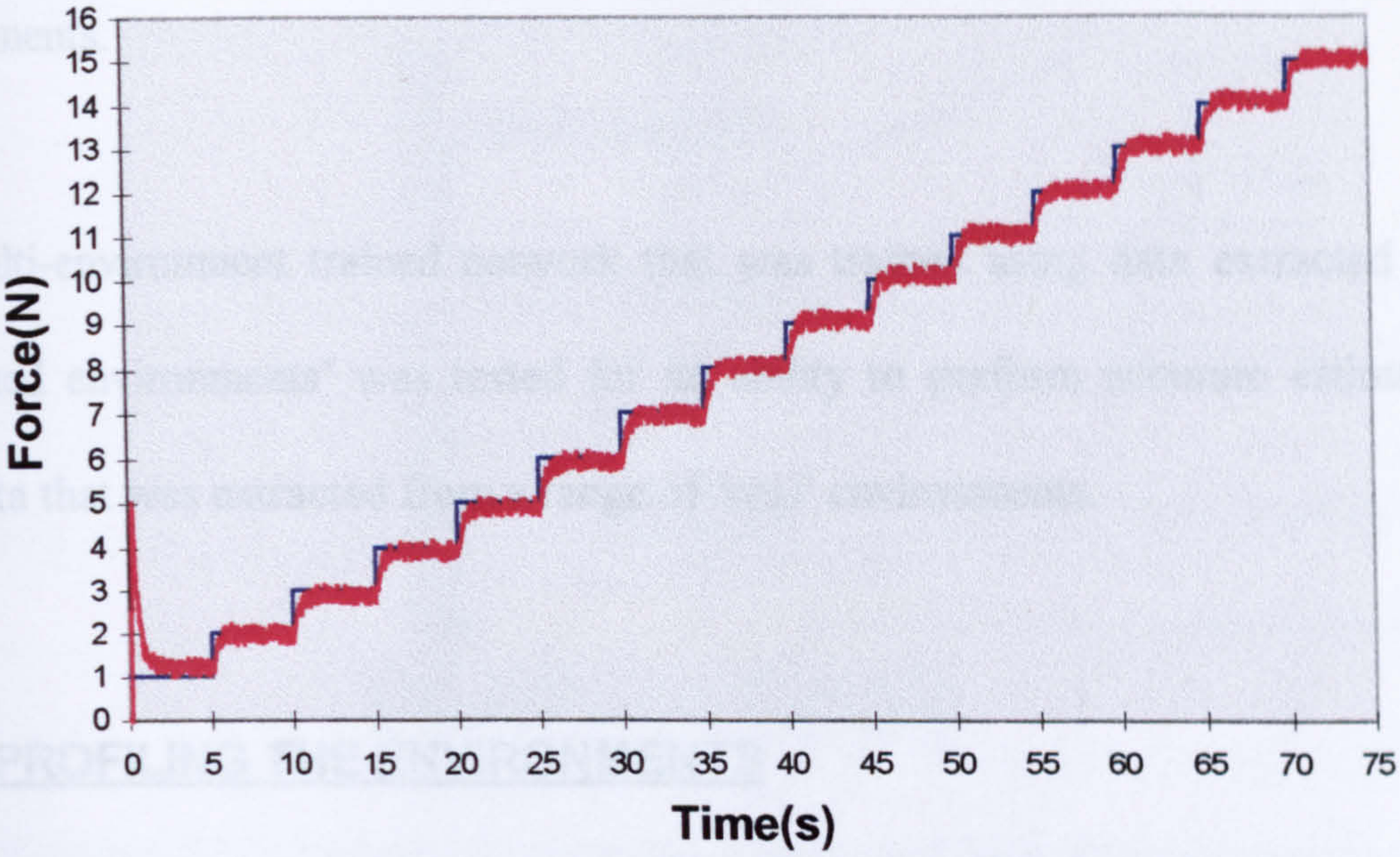


Figure 6.31 Step tests for contact with environment 12. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR=20 dBs.

The result shows that the intelligent force control system recovered from the initial impact with the environment but an overshoot resulted from the impact. The control system tracked the force demand across the force application range but offsets were measured at several operating points.

6.7 ANN ESTIMATION WITH DATA EXTRACTED FROM REAL ENVIRONMENTS

The simulated environments that were used to train and test the 'multi-environment' trained network were generated from a parameterised contact model, the form of which was not varied throughout the investigation. Although the contact model yielded environments that were similar to practically measured environments [Caldwell and Gosney, 1993], it is unlikely that all environments will adhere to the

model form, and degradation in network performance may result with such environments.

The multi-environment trained network that was trained using data extracted from 'simulated environments' was tested for its ability to perform accurate estimations with data that was extracted from a range of 'real' environments.

6.7.1 PROFILING THE ENVIRONMENTS

Several 'real' environments were considered during the investigation. A brief description of the physical characteristics of each of the environments is presented in table 6.15.

Name	Description
Environment 13	carpet tile: thickness = 6mm
Environment 14	plastic container
Environment 15	sponge: thickness = 18 mm
Environment 16	sponge: thickness = 18 mm (wet)
Environment 17	human hand (relaxed)
Environment 18	human hand (tense)

Table 6.15 Description of real environments investigated.

Environment 13 was a 6mm thick carpet tile and environment 14 was an empty plastic container. Environment 15 was an 18mm thick piece of sponge and environment 16 was the same piece of sponge that was soaked in water. Environments 17 and 18 were generated from contact with a human hand under relaxed and tensed conditions. Hardness profiles for each of the environments were measured using the apparatus illustrated in figure 6.32.

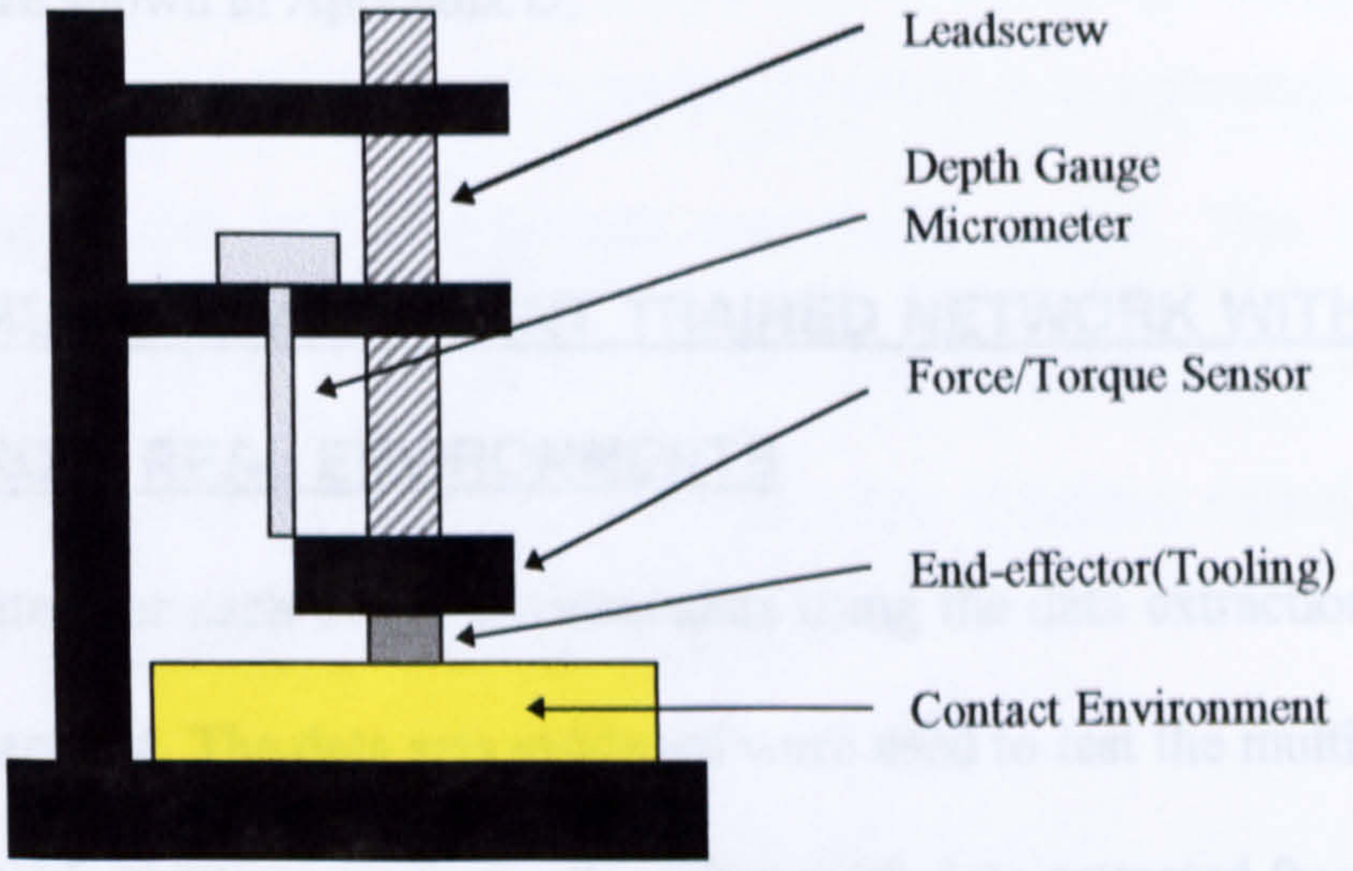


Figure 6.32 Apparatus for measuring contact environment parameters.

The measurement apparatus shown in figure 6.32 comprised a leadscrew mounted to a rigid support base. A force/torque sensor was rigidly coupled to the end of the leadscrew to provide contact force measurements and a depth gauge micrometer was used to provide measurements of the depth into the environment. Rotation of the leadscrew results in translation of end-effector which, during contact, results in variation of the contact force.

A hardness profile was generated for each of the environments investigated. This was achieved by bringing the end-effector into contact with the environment until a contact force of 1N was sensed. The end-effector was stopped and the depth into the environment was measured using the depth gauge micrometer. The sensed force and corresponding depth measurement were recorded. The end-effector was then moved further into the environment and the measurement procedure was repeated at 1N intervals, for contact up to 20N. Similar approaches to environmental 'profiling' have been reported and the shape of the profiles produced are similar to 'practically

measured' profiles reported by Caldwell and Gosney [1993]. Hardness profiles for each of the environments are shown in Appendix D.

6.7.2 TESTING THE MULTI-ENVIRONMENT TRAINED NETWORK WITH DATA EXTRACTED FROM REAL ENVIRONMENTS

A test data set was generated for each of the environments using the data extraction procedure described in Chapter 4. The data sets produced were used to test the multi-environment trained network's ability to perform estimations with data extracted from a range of 'real' environments. The network that was investigated was the same network used for the previous investigations. The random placement trained network comprised 120 hidden layer nodes and was trained using a centre spread of 2. Figure 6.33 shows the loss functions obtained when the multi-environment trained network was tested with data extracted from each of the 'real' environments.

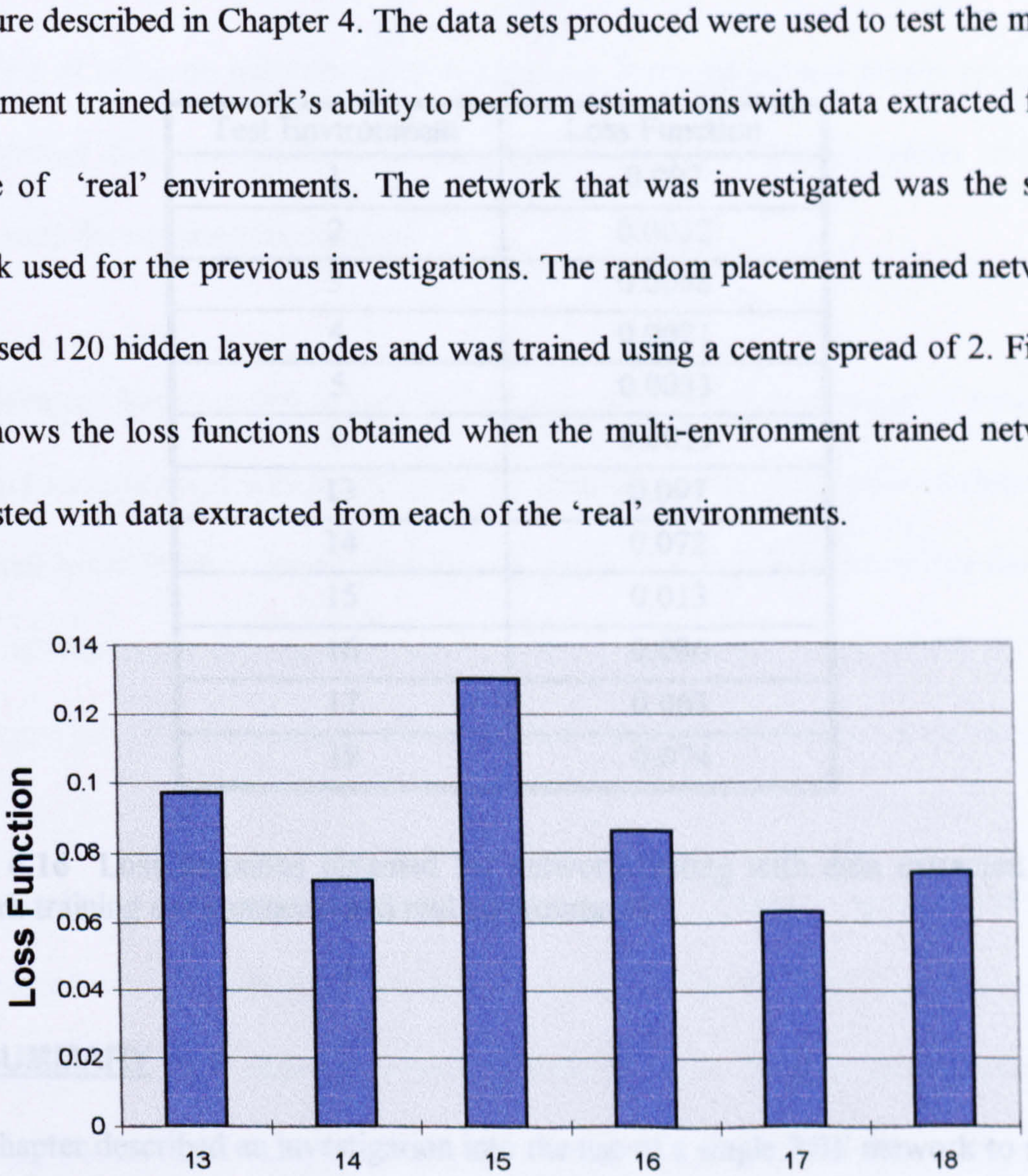


Figure 6.33 Loss functions for network estimations with each of the 'real' environments.

Figure 6.33 shows that estimations were satisfactory with each of the environments. However, estimations with the real environments were not as accurate as estimations with simulated environments that had intermediate compliance modes. This is illustrated by table 6.16 which shows a comparison of the loss functions obtained when the network was tested with the data extracted from the simulated training environments and data extracted from the real environments.

Test Environment	Loss Function
1	0.007
2	0.0032
3	0.0058
4	0.0021
5	0.0033
6	0.0053
13	0.097
14	0.072
15	0.013
16	0.086
17	0.063
18	0.074

Table 6.16 Loss functions obtained for network testing with data extracted from network training environments and real environments.

6.8 SUMMARY

This chapter described an investigation into the use of a single RBF network to model idealised reaction to a range of simulated non-rigid environments, each with differing degrees of rigidity. The feasibility of using a single RBF network to achieve this objective was demonstrated.

Data sets that comprised data extracted from several environments were combined to form a multi-environment training data set. Networks trained with the 'multi-environment' data set performed accurate estimations with environments that were represented in the network training data set. Accurate estimations were also achieved with a range of non-represented environments that had intermediate compliance modes.

The effect of noise on network performance was investigated and results presented demonstrated that noise added to the force and depth measurements did not significantly deteriorate performance.

The intelligent force control scheme's ability to apply forces to a range of simulated non-rigid environments was investigated by simulation. Results presented show that the neural based force control scheme was capable of applying forces to the majority of the environments investigated, across a force application range of 1N to 15N. However offsets were measured, which resulted from non-ideal network estimation.

The intelligent force control scheme's ability to apply forces to environments that were more/less rigid than the environments used to train the control system's ANN was investigated by simulation. Results presented show that the control scheme's force application capability deteriorated as the degree of environmental rigidity varied from that of the most/least rigid environments represented in the network training data set.

ANN estimation with data that was extracted from a range of 'real' environments was investigated. Results presented show that network estimations with each of the environments were satisfactory. However, estimations with the 'real' environments were not as accurate as estimations with simulated environments that had intermediate compliance modes.

CHAPTER 7

CONTACT WITH VARYING ENVIRONMENTS

7.1 INTRODUCTION

This chapter describes an investigation into the use of the intelligent force control scheme for applying forces to a surface that had varying rigidity. One practical application where a robot system may be required to apply forces to such a surface is profile tracking with simultaneous force application where a robot applies a specified force normal to the contact surface while simultaneously moving tangentially across the surface [Lange and Hirzinger, 1992]. The multi-environment trained network that was developed in Chapter 6 was tested for its ability to apply forces to a varying environment.

7.2 INTELLIGENT FORCE CONTROL WITH VARYING ENVIRONMENTS

Force application to environments with varying rigidity requires the force control system to maintain a specified contact with the environment when the degree of environmental rigidity changes. A practically useful force control scheme must be capable of sensing and reacting to changes in the degree of environmental rigidity in real time.

The intelligent force control scheme adjusts the position of the end-effector relative to the contact environment in a single degree of freedom such that a specified force is applied to the environment. The scheme may be employed to control the contact force for tasks that require profile tracking with simultaneous force application. This is illustrated in figure 7.1 which shows intelligent force control system controlled motion along the Z_{task} axis and position controlled motion along the X_{task} axis (which is orthogonal to the Z_{task} axis) to move the end-effector across the environment.

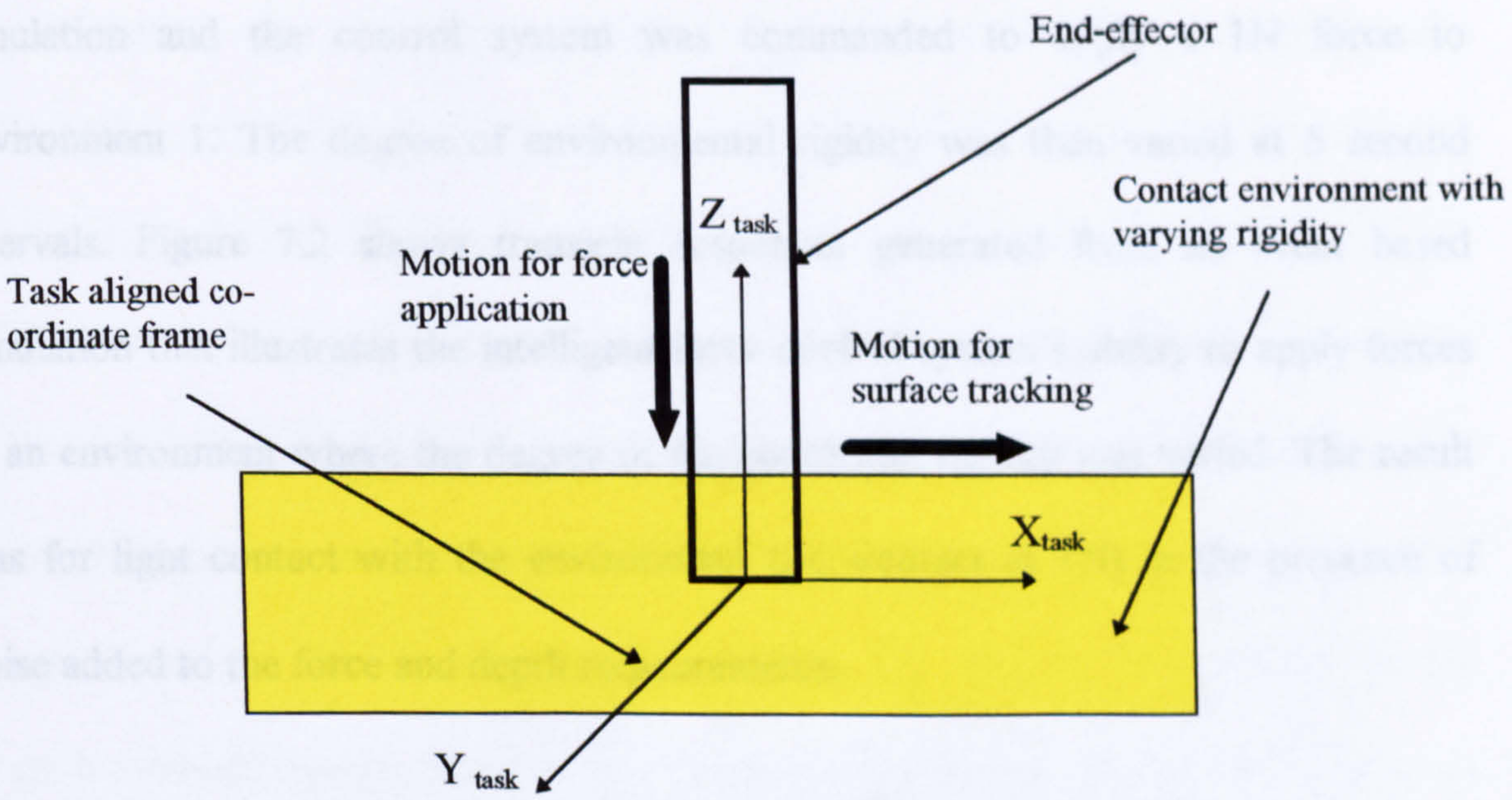


Figure 7.1 Illustration of profile tracking with simultaneous force application.

Methods for partitioning the task aligned coordinate frame into force controlled and position controlled axes have been proposed (Mason, 1981; McKerrow, 1991), however, methods for partitioning the task aligned coordinate frame were not the focus of this investigation and it was assumed that the force and position controlled axes could be determined. Additionally, the development of methods for specifying positional trajectories to traverse the surface was not within the scope of this work.

7.3 CONTACT WITH REPRESENTED ENVIRONMENTS

The multi-environment trained network that was developed in Chapter 6 was tested for its ability to acquire and maintain a specified contact with an surface that comprised several environments, each with differing degrees of rigidity. The network comprised 120 hidden layer nodes and was trained using a centre spread of 2.

Initially, contact with environments that were represented in the network training data set was investigated. The aforementioned network was loaded into the ACSL simulation and the control system was commanded to apply a 1N force to environment 1. The degree of environmental rigidity was then varied at 5 second intervals. Figure 7.2 shows transient responses generated from an event based simulation that illustrates the intelligent force control system's ability to apply forces to an environment where the degree of environmental rigidity was varied. The result was for light contact with the environment (i.e. contact at 1N) in the presence of noise added to the force and depth measurements.

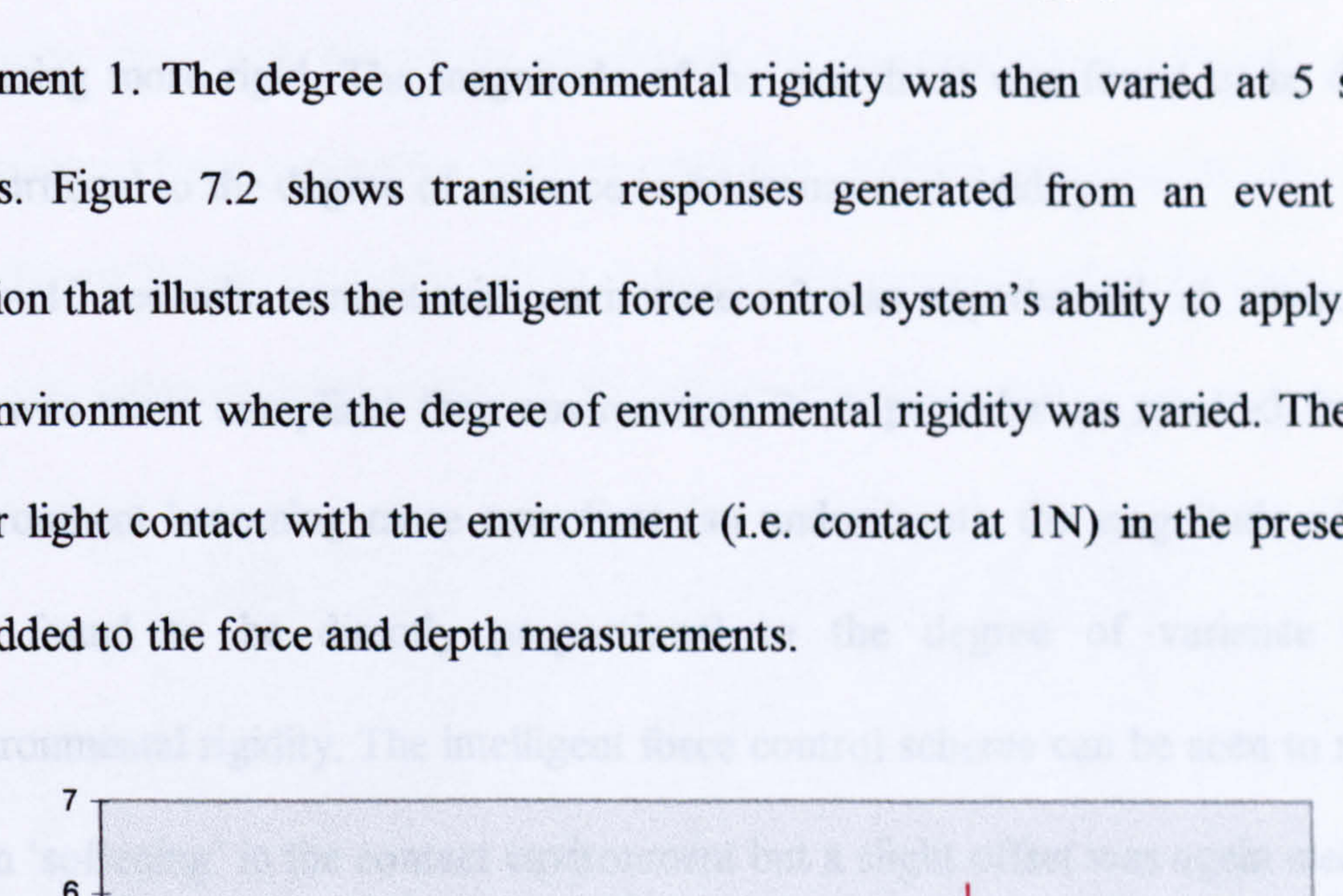


Figure 7.2 Transient response for event based simulation. Force setpoint=1N. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR = 20 dBs.

The event based simulation result shows the following contact sequence:

i. at t = 0.1 seconds, the end-effector impacts environment 1. The force control system recovered from the impact and the transient response was stable, however a slight offset was measured. Small perturbations about the setpoint resulted from noise superimposed onto the force and depth measurements.

ii. at t = 5 seconds, contact with environment 2 was experienced, an environment that was significantly more rigid than environment 1. The intelligent force control system can be seen to recover from 'hardening' in the environment but a perturbation (an overshoot) resulted from the environment instantaneously becoming more rigid. The magnitude of the overshoot was found to be directly proportional to the degree of variance in environmental rigidity.

iii. at t = 10 seconds, contact with environment 3 was experienced, an environment that was more compliant than environment 2. A perturbation resulted from the environment becoming more compliant (an undershoot), the magnitude of which was found to be directly proportional to the degree of variance in the environmental rigidity. The intelligent force control scheme can be seen to recover from 'softening' in the contact environment but a slight offset was again measured.

iv. at t = 15 seconds, contact with environment 4 was experienced, an environment that was more rigid than environment 3. Again, a perturbation (an overshoot) resulted from the environment becoming more rigid. The magnitude of the perturbation was significantly lower than the previous 'hardening' result since a lesser degree of environmental variance was experienced. The intelligent force control scheme can again be seen to recover from 'hardening' in the contact environment.

v. at t = 20 seconds, contact with environment 5 was experienced, an environment that was more compliant than environment 4. The intelligent force control scheme can be seen to recover from 'softening' in the contact environment.

vi. at t = 25 seconds, contact with environment 6 was experienced, an environment that was significantly more rigid than environment 5. The intelligent force control scheme can be seen to recover from significant 'hardening' in the environment. A large perturbation (an overshoot) resulted as the environment become significantly more rigid.

The results presented represents significant parameter variance in the contact environment. Minor parameter variance did not significantly disturb the control system. Figure 7.3 shows results of the same contact sequence when the force control scheme was commanded to apply a 7.5N force to environment 1.

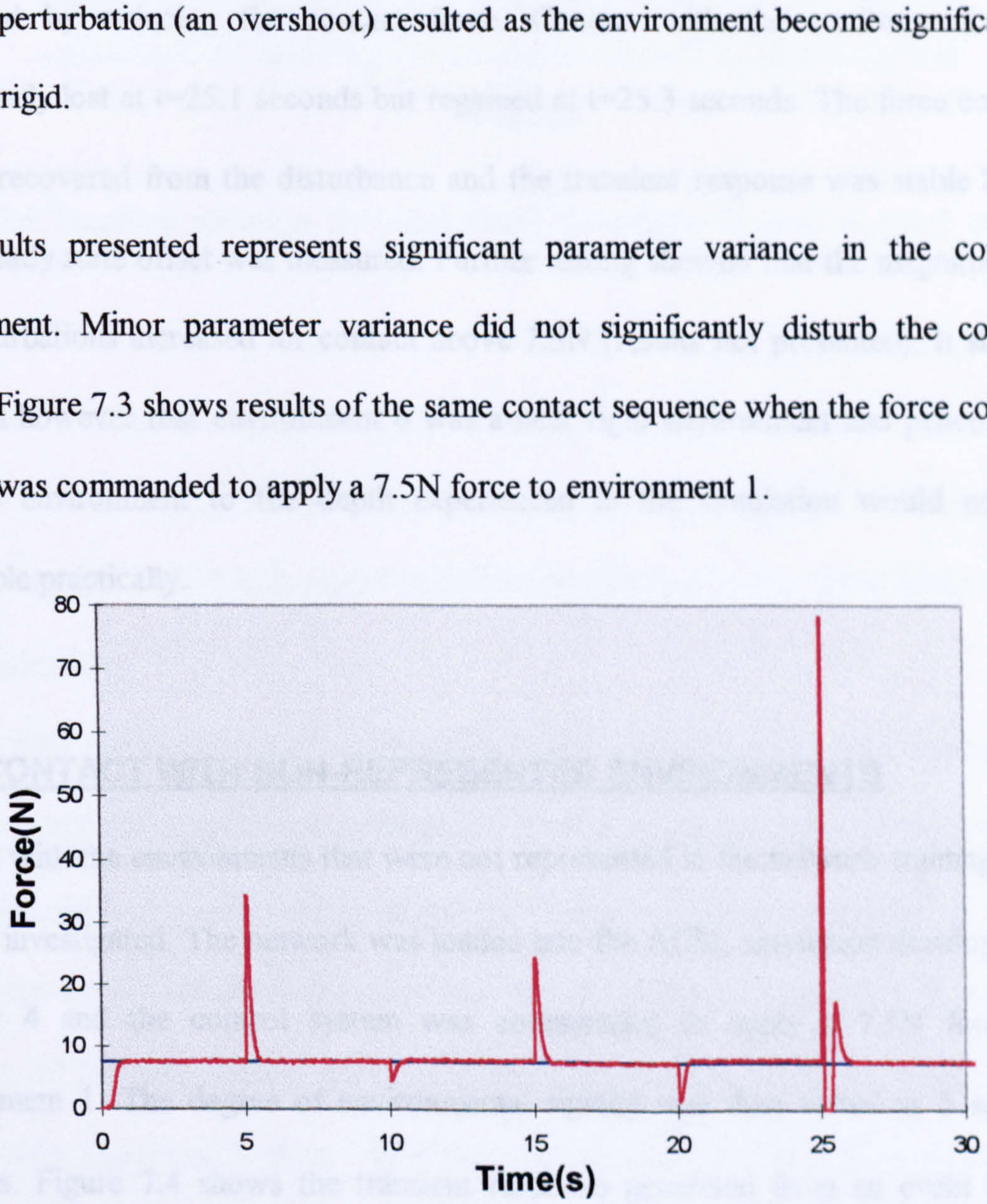


Figure 7.3 Transient response for event based simulation. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force. SNR = 20 dBs.

Figure 7.3 illustrates that the magnitude of the perturbations that resulted from variance in the environmental rigidity was greater than the result obtained for contact at 1N. The perturbation that occurred at t=25 seconds resulted from a significant change in environmental rigidity and the force applied to the environment reached 78.2N, far beyond the force application range. Although the intelligent force control scheme's ANN does not have knowledge of contact of this magnitude, the ANN has responded by reducing the contact force. Contact with the environment was momentarily lost at t=25.1 seconds but regained at t=25.3 seconds. The force control scheme recovered from the disturbance and the transient response was stable but a slight steady state offset was measured. Further testing showed that the magnitude of the perturbations increased for contact above 7.5N (results not presented). It should be noted however that environment 6 was a near rigid environment and penetration into the environment to the depth experienced in the simulation would not be achievable practically.

7.4 CONTACT WITH NON-REPRESENTED ENVIRONMENTS

Contact with the environments that were not represented in the network training data set was investigated. The network was loaded into the ACSL simulation developed in Chapter 4 and the control system was commanded to apply a 7.5N force to environment 1. The degree of environmental rigidity was then varied at 5 second intervals. Figure 7.4 shows the transient response generated from an event based simulation that illustrates the intelligent force control systems ability to apply forces to an environment that comprised the non-represented environments.

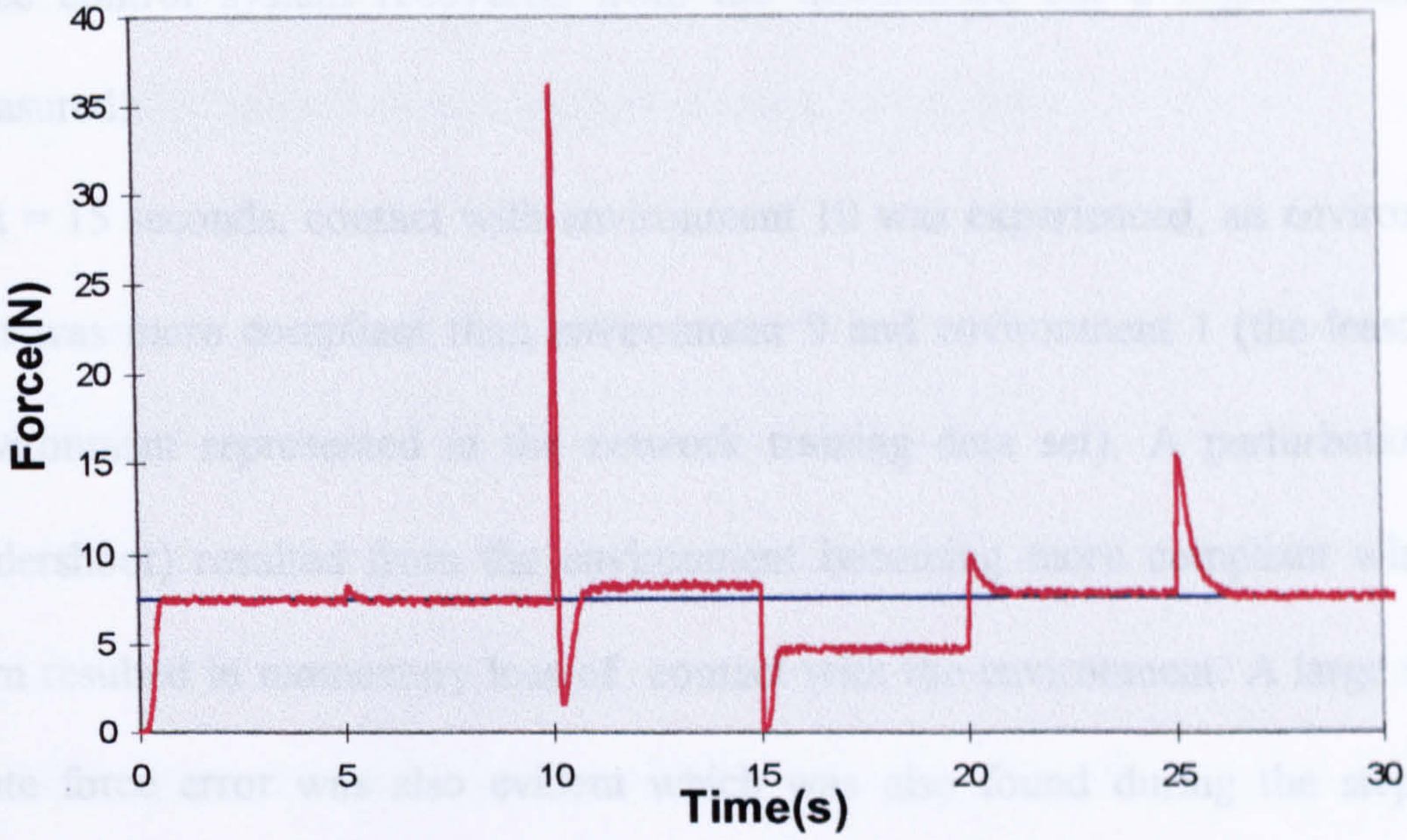


Figure 7.4 Transient response for event based simulation. Contact at t=0.1 sec. ■ Force setpoint. ■ Applied force.

The event based simulation result shows the following contact sequence:

i. at t = 0.1 seconds, the end effector impacts with environment 7. The system recovered from the initial impact and the response was stable but a slight offset was measured.

ii. at t = 5 seconds, contact with environment 8 was experienced, an environment that was slightly more rigid than environment 7. The intelligent force control scheme can be seen to recover from slight 'hardening' in the environment. A small perturbation (an overshoot) resulted from the environment becoming more rigid.

iii. at t = 10 seconds, contact with environment 9 was experienced, an environment that was significantly more rigid than environment 8 (and more rigid than environment 6, the most rigid of the network training environments). A large perturbation resulted from the significant change in environmental rigidity. The

force control system recovered from the disturbance but a slight offset was measured.

iv. at t = 15 seconds, contact with environment 10 was experienced, an environment that was more compliant than environment 9 and environment 1 (the least rigid environment represented in the network training data set). A perturbation (an undershoot) resulted from the environment becoming more compliant which in turn resulted in momentary loss of contact with the environment. A large steady state force error was also evident which was also found during the step tests performed on environment 10 (see figure 6.29).

v. at t = 20 seconds, contact with environment 11 was experienced, an environment that was more rigid than environment 10. The intelligent force control scheme can be seen to recover from 'hardening' in the contact environment.

vi. at t=25 seconds, contact with environment 12 was experienced, an environment that was significantly more rigid than environment 11.

The transient response results presented represent significant parameter variance in the contact environment. Minor degrees of parameter variance did not significantly disturb the control system.

7.5 SUMMARY

The use of the intelligent force control scheme's ability to apply forces to a varying environment was demonstrated. The scheme was tested with a range of simulated environments and satisfactory contact was made with environments that were

represented in the multi-environment training data set. However, slight offsets were measured, which resulted from non-ideal network estimations.

Simulation results presented show that perturbations resulted from variance in the degree of environmental rigidity, the magnitude of which were proportional to the degree of variance. Large 'swings' in environmental rigidity resulted in excessive contact or loss of contact with the environment. However, in most instances, the desired contact was re-established.

It should be noted however that results presented for contact with near rigid environments were not practically achievable, since such environments cannot be penetrated to the extent experienced in the simulation. The results were included to illustrate the effects of significant environmental variance.

CHAPTER 8

CONCLUSIONS AND FURTHER WORK

8.1 INTRODUCTION

The main requirement of a widely applicable force control scheme is an ability to apply forces to a range of environments without *a priori* knowledge of the contact environments mechanical properties. Humans perform contact tasks with a wide range of objects to a high degree of accuracy in diverse and challenging environments, and intelligence plays a major role in the human force control capability. However, attempts at endowing machines with levels of intelligence that allow for complete autonomy in real world environments have only been partially successful, and to date robots are not been successfully integrated into the majority of real world environments.

This study was centered around the development of an Artificial Neural Network (ANN) model that was capable of allowing interaction with a range of non-rigid environments. Once developed, the ANN was integrated into a simulation of a single DOF positional controlled mechanical manipulator, and the addition of the ANN endowed the manipulator with a force control capability. The force control scheme's ability to apply forces to a range of non-rigid environments was investigated by simulation.

8.2 SINGLE ENVIRONMENT INVESTIGATION

Initially, contact with a single non-rigid environment was investigated and several ANN training considerations that may influence network performance were considered. Radial Basis Function (RBF) networks that comprised thin plate spline activation functions were used throughout the investigation. One factors that was found to be significant was the placement of the RBF centres.

Two methods for placing the RBF centres were considered: random placement and k-means clustering used in conjunction with random placement. Both methods were found to yield networks capable of performing single environment input-output mappings to a high degree of accuracy. However, network performance was found to be dependent on the spread of the RBF centre placement, and the performance of random placement trained networks and k-means clustering trained networks were found to be optimum when the centres were placed over higher multiples of the network training data set input domain.

Once trained, the effect on network performance of noise added to the force and depth measurements was investigated. The effect of each of the noise components was investigated in isolation and then the simultaneous action of the noise signals was investigated. Results presented showed that network performance was not significantly degraded by the presence of the noise. A signal to noise ratio of 20dBs was used to test the networks.

The single environment trained networks were validated by applying random amplitude signals to the network inputs. Further validation was performed by applying a series of unit step changes in force demand across a force application range of 1N to 15N. Network estimations were found to be satisfactory with each of the tests with both noise free data and data that had noise superimposed onto the force and depth measurements.

The validated network was incorporated into a single degree of freedom mechanical manipulator simulation that was developed in the Advanced Continuous Control Language. The inclusion of the network endowed the manipulator with an autonomous force control capability. The quality of the force control was investigated by simulation and contact with the environment that was used to extract the network training data set was found to be accurate across the force application range. However, when presented with data extracted from environments that had degrees of rigidity that were different to the 'training' environment, estimations were poor. Steady state offsets were measured, the magnitude of which was found to be directly proportional to the degree of environmental variance from the rigidity of the training environment. Thus the intelligent force control scheme that comprised a single environment trained network had a very limited force control capability.

8.3 MULTI-ENVIRONMENT INVESTIGATION

A method for overcoming the limitations of the single environment trained network was investigated. A single RBF network was trained using a data set that comprised several 'single environment' data sets. Once trained with the multi-environment data

set, the RBF network was tested for its ability to perform accurate estimations with environments represented in the training data set. Results presented in Chapter 6 show that networks trained with a data set that only comprised data extracted from environments that had upper and lower limits of the compliance modes investigated did not have an ability to perform accurate estimations with environments that had intermediate compliance modes. The inclusion in the network training data set of data extracted from a several 'intermediate' environments was found to vastly improve network estimations with intermediate environments. Thus the network required knowledge of idealised reaction to a wide range of non-rigid environments.

The effect of noise on the performance of multi-environment trained networks was investigated. Results presented show that the networks were not sensitive to noise superimposed onto the force and depth measurements.

The multi-environment trained network was incorporated into the ACSL simulation and the intelligent force control scheme was tested for its ability to perform estimations with a range of environments. Initially contact with environments that were represented in the network training data set was investigated by simulation. Results presented show that contact with represented environments was accurate in the presence of noise superimposed onto the force and depth measurements. However, slight offsets were measured which resulted from non-ideal network estimations.

Contact with non-represented environments that had compliance modes within the bounds of the training data set environments were also found to be accurate. Several simulated environments were considered during the investigation.

8.4 ANN ESTIMATION WITH REAL ENVIRONMENTS

ANN estimation with data extracted from real environments was investigated. Hardness profiles for a range of real environments were measured. Data was generated from several environments (eg. Sponge, plastic, human hand, etc.) and the multi-environment trained network that was developed in Chapter 6 was tested for its ability to perform estimations with data extracted from the real environments. Although estimations with the environments were not as accurate as estimations with the simulated environments, estimations were satisfactory.

8.5 CONTACT WITH VARYING ENVIRONMENTS

The intelligent force control scheme's ability to apply forces to a varying environment was investigated. Results presented show that the scheme recovered from disturbances arising from variations in environmental rigidity. Perturbations were experienced which were found to be proportional to the degree of environmental variance. However, the control scheme was found to be capable of recovering from hardening and softening of the environment.

8.6 RECOMENDATIONS FOR FURTHER WORK

8.6.1 FURTHER ANN DEVELOPMENT

The ANNs developed during the investigation were trained with data that was extracted from a static contact model, the use of which can be justified by the fact that end-effector velocities during contact are significantly lower than for free motion. [Whitney, 1987]. However, some environments may have a significant viscous components which may deteriorate the performance of static data trained networks. Thus one possible area for further development is the use of dynamically trained networks for inclusion in the intelligent force control scheme.

Additional areas for further ANN development may be:

- the use of alternative activation functions in the RBF hidden layer nodes (e.g. Gaussian functions, multi-quadratic functions, etc.).
- consider other network architectures (i.e. multi-layer perceptron, recurrent network, etc.).
- consider alternative RBF centre placement/optimisation techniques. Several such techniques were highlighted in Chapter 3.

8.6.2 FURTHER CONTROL SYSTEM DEVELOPMENT

The intelligent force control scheme's force control capability was investigated by simulation and on-line application of the techniques should be investigated. However, the control scheme is based around sub millimeter positioning of an end-effector within the contact environment and practical implementation would require a high resolution positioning device. The technology is beyond the positioning capabilities of

the vast majority of present robot systems. Additionally accurate real-time depth measurement techniques are required. Application to a multi-axis robot system would also require partitioning of a task aligned coordinate frame into orthogonal position and force controlled axes.

Expansion of the control scheme to include interaction with rigid environments was beyond the scope of this investigation. However, the development would be essential if the control scheme was to be useful in a wide range of environments.

8.7 CONCLUDING REMARKS

The force control scheme presented in this dissertation has been defined as being 'intelligent', a term that is incorrectly used in many published works. The scheme should be regarded as an intelligent system since the control scheme's ANN has learned, via training, to respond to a wide range of environments of differing degrees of rigidity. Additionally, and more significantly, the control system's ANN has the ability to suitably respond to environments that it has not experienced, which requires the ANN to adapt to a new problem domains armed only with knowledge acquired during training.

REFERENCES

AKAIKE, H. (1974). A new look at the statistical model identification. *IEE Trans. Auto Cont.* **AC-19**(6), pp. 716-723.

AN, C., and HOLLERBACH, J. (1987). Kinematic stability issues in force control of manipulators, Proc. *IEEE Conf. on Robotics and Automation.* pp. 223-228.

ANDERSON, R.J. and SPONG, M.W. (1988). Hybrid impedance control of robotic manipulators. *IEEE J. Robotics and Automation* RA-4(5), pp. 549-555.

ANNASWAMY, A., and SETO, D. (1993). Object Manipulation using Compliant Fingerpads: Modelling and Control, *Trans. ASME*, Vol. 115, pp. 638-648.

BIERMAN,G., (1977). Factorization Methods for Discrete Sequential Estimation, New York: Academic.

BROOMEHEAD, D.S., and LOWE, D. (1988). Multivariable function interpolation and adaptive networks, *Complex Systems*, 2, pp. 231-255.

CALDWELL, D.G. and GOSNEY, C. (1993). Multi modal tactile sensing and feedback (tele-taction) for enhanced tele-manipulator control. *Proc. Int. Conf. on Intelligent Robots and Systems,* Yokohama, Japan, pp. 1487-1494.

CARELLI, R., KELLY, R., ORTEGA, R. (1990). Adaptive force control of robot manipulators, *Int. J. of Control.* **52**(1), pp. 37-54.

CHEN, S., BILLINGS, S.A., COWAN, C.F.N. AND GRANT, P.M. (1990). Practical identification of NARMAX models using radial basis functions. *Int. J. of Control.* **52**(6), pp. 1327-1350.

CHEN, S., and BILLINGS, S.A. (1992). Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. of Control.* **55**(5), pp. 1051-1070.

CHRISTODOULOU, C.,CLARKSON, T.G., TAYLOR, J.G. (1996) Speaker identification using pRAM neural networks, *Proc. Int. Conf. on Engineering Applications of Neural Networks*, pp.265-268.

COLBAUGH, R., SERAJI, H., GLASS, K. (1992). Impedance control for dextrous space manipulators. *Proc. Of 31st Conf. on Decision and Control*, USA, pp.1881-1886.

CYBENKO, G. (1989) Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*. **2**, pp. 303-314.

DACOSTA, P., KORDICH, C., WILLIAMS, D. AND GOMM, J.B. (1997). Estimation of inaccessible fermentation states with variable inoculum sizes. *Artificial Intelligence in Engineering, Elsevier.* **11**, pp. 383-392.

DOHERTY, S.K., GOMM, J.B. AND WILLIAMS, D. (1997). Experiment design considerations for non-linear system identification using neural networks. *Computers Chem. Engng.* **21**(3), pp. 327-346.

DOTE, Y., SUYITNO, A., STREFEZZA, M. (1990). Fuzzy learning grasping force controller for manipulator hand, Proc. 29th Conf. on Decision and Control, Hawaii, pp.2624-2647.

DUTTA, J.S. and SHEKHAR, S. (1988). Bond-rating: A non-conservative application of neural networks. *Proceedings of the IEEE Int. Conf. on Neural Networks*, pp. 142-150.

EIKENS, B. and KARIM, M.N. (1994). Real-time control of a waste water neutralization process using radial basis functions. *IFAC Advanced Control of Chemical Processes,* Japan. Pp. 113-118.

ELOSEGUI, P., DANIEL, R.W., SHARKEY, P.M. (1990). Joint servoing for robust manipulator force control, *Proc. Int. Symp. on Intelligent Control*, pp. 246-251.

EVANS, J.T. (1994). Investigation of a multi-layer perceptron network to model and control a non-linear system. *Ph.D. Thesis,* Liverpool John Moores University, UK.

FINCHER, D.W., and WADE,D.F., (1990). Multi-sensor data fusion using neural networks, *Proc. Decision and Control*, Hawaii, pp. 836-838.

FUKUDA, T., and KITAMURA, N. (1986). Adaptive force control of grippers with consideration of dynamics of objects. *RoManSy86,* Crackow, Poland, pp. 235-249.

FUKUDA, T., SHIBATA, T.,TOKITA, M., AND MITSUOKA, T. (1990). Adaptation and learning for robotic manipulator by neural network, *Proc. Decision and Control,* Hawaii, pp. 3283-3287.

GORINEVSKY, D., KAPITANOVSKY, A., GOLDENBERG, A. (1994). Design of radial basis function based controllers for autonomous parking of wheeled Vehicles, *Proc. American Control Conf.,* USA, pp. 221-225.

GUGLIELMO, K., and SADEGH, N. (1994). Implementing a hybrid learning force control scheme. *IEEE Control Systems,* February, pp. 72-79.

HOFLAND, A.G., MORRIS, A.J., MONTAGUE, G.A. (1992). Radial basis function networks applied to process control. In *Proc. of the American Control Conf.* pp. 480-484.

HOGAN, N. (1985). Impedance control: an approach to manipulation, Parts I, II, and III, *J. of Dynamics, Systems, Measurement and Control*, Vol. 107, pp. 1-23.

HOLLINGER, J.G., BERGSTROM, R.A., BAY, J.S. (1993). A fuzzy logic force controller for a stepper motor robot, *Proc. IEEE Int. Conf. on Robotics and Automation*, CA, USA, pp. 204-209.

HOPCROFT, J., KEARNEY, AND J., KRAFFT, D. (1991). A case study of flexible object manipulation. *Int. J. Robotic Research,* Vol. 10, No.1, pp. 41-50.

HORNICK, K., STINCHCOMBE, M., WHITE, H. (1989). Multistage feedforward networks are universal approximators. *Neural Networks*, 2, pp. 359-366.

HOWELL, A.J., and BUXTON, H. (1996). Improving generalisation in radial basis function networks for face recognition, EANN'96, London, UK., pp. 297-304.

HYDE, J.M., and CUTKOSKY, M.R. (1993). Contact transition control: an experimental study, *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, USA, pp. 363-368.

JANKOWSKI, K. and ELMARAGHY, H. (1992). Dynamic decoupling for hybrid control of rigid-/flexible-joint robots interacting with the environment, *IEEE Trans. on Robotics and Automation,* **8**(5), pp. 519-534.

KAUFMAN, L. and ROUSSEEUW, P.J. (1990). Finding groups in data - an introduction to cluster analysis, *John Wiley & Son,* New York.

KAZEROONI, H., SHERIDAN, T.B., HOUPT, P.K., (1986). Robust compliant motion for manipulators, part one: the fundamental concepts of compliant motion, *IEEE J. of Robotics and Automation,* Vol. RA-2, No.2, pp. 27-33.

KODOGIANNIS, V.S. (1994). Neural network techniques for modelling and learning control of an underwater robotic vehicle, *Ph.D. Thesis*, The University of Liverpool, U.K.

KOHONEN, T. (1990). The self-organising map, In *Proceedings IEEE.* **78**(9), 1464-1480.

KUO, L.E. and MELSHEIMER, S.S. (1994). Using genetic algorithms to estimate the optimum width parameter in radial basis function networks. In *Proc. American Control Conference, Maryland, USA*, pp. 175-179.

LANGE, F., and HIRZINGER, G. (1992). Iterative self-improvement of force feedback control in contour tracking, *Proc. Int. Conf. on Robotics and Automation*, Nice, France, pp. 1399-1404.

LASKY, T. and HSAI, T. (1991). On force-tracking impedance control of robot manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, CA, pp. 435-441.

LEONARD, J.A. and KRAMER, M.A. (1991). Radial basis function networks for classifying process faults. *IEEE Control Systems Magazine,* pp. 31-38.

LEONTARITIS, I.J. and BILLINGS, S.A. (1987). Model selection and validation methods for non-linear systems, *Int. J. Control.* **45**(1), pp. 311-341.

LEVIN, E. (1993). Hidden control neural architecture modeling of nonlinear time varying systems and its applications, *IEEE Trans. on Neural Networks*, **4**(1), pp. 109-116.

LEWIS, F.L., ABDALLAH, C.T., DAWSON, D.M. (1993). Control of manipulators, *Macmillan Publishing Co.*

LJUNG, L. and SODERSTROM, T. (1983). Theory and practice of recursive identification, *MIT Press,* Cambridge, Mass.

LU, Z., and GOLDENBERG, A.A. (1995). Robust impedance control and force regulation: theory and experiments, *Int. J. of Robotics Research*, Vol. 14, No. 3, pp. 225-254.

LUH, J.Y.S., (1983). Conventional controller design for industrial robots - a tutorial, *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. SMC-13, No. 3, pp. 298-316.

LUK, B.L., (1991). Robot force sensing using stochastic monitoring of the actuator current, *Ph.D. Thesis*, Portsmouth Polytechnic, UK.

MALLIARIS, M. and SALCHENBERGER, L. (1994). Neural networks for predicting options volatility, In *the World Congress on Neural Networks, San Diego, Int. Neural Network Society Annual Meeting*. II, pp. 290-295.

MANDAL, N. and PAYANDEH, S. (1993). Experimental evaluation of the importance of compliance for robotic impact control, Proc. IEEE Control Application Conf., pp. 2499-2503.

MAPLES, J.A., and BECKER, J.J. (1986). Experiments in force control of robotic manipulators. *IEEE Conf. On Robotics and Automation,* Philedelphia, Pen., USA, pp. 695-702.

MASON, M.T. (1981). Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 11, No. 6, pp. 418-432.

MAYEDA,H., AND IKEDA,N. (1993). Motion/force/ impedance control for robot tasks, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Yokohama, Japan, pp. 1522-1529.

M^{c}CULLOCH, W.S. AND PITTS, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophy.*, 5, pp. 115-133.

M^{c}KERROW, P.J. (1991). Introduction to Robotics, *Addison-Wesley Publishing Co.*

MINSKY M., PAPERT, S. (1969). Perceptrons, *MIT Press*, MA.

MOODY, J. and DARKEN, C.J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1, pp. 281-294.

NIU,S., and FISHER,D., (1991). MIMO system identification using augmented UD factorization, *Proc. American Control Conf.*, USA, pp. 699-703.

PATTERSON, D.W. (1996). Artificial neural networks: theory and applications, Prentice Hall.

PAUL, R.P., and SHIMANO, B. (1976). Compliance and control, *Proc. of Joint Automatic Control Conf.*, Lafayette, Indiana, USA, pp. 694-699.

PEI, H. (1992). Neural network robot control methods, *Int. Conf. on automation, Robotics, and Computer Vision*, pp. 871-876.

POTTMANN, M. and SEBORG, D.E. (1992). Identification of non-linear processes using reciprocal multiquadric functions. *J. Proc. Cont.* 2 (4), pp. 189-203.

PUJAS, A., DAUCHEZ, P., PIERROT, F., (1993). *Proc. Int. Conf. on Intelligent Robots and Systems*, Yokohama, Japan, pp. 1487-1494.

RAIBERT, M., and CRAIG , J. (1981). Hybrid position/force control of manipulators, *Trans. ASME*, Vol. 102, pp. 126-133.

RUMELHART, D.E., HILNTON, G.E. AND WILLIMAS, R.J. (1986). Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Micro Structure of Cognition*, MIT Press, pp. 318-362.

SERAJI, H., and COLBAUGH, R. (1993). Adaptive force-based impedance control, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Yokohama, Japan, pp. 1537-1544.

STEVEN, A. (1989). Hybrid force and position control in robotic surface processing. *Ph.D. Thesis,* Newcastle University, UK.

SUH, I, HONG, J., OH, S., AND KIM, K. (1991). Fuzzy rule based position/force control of industrial manipulators, *IEEE/RSJ Int. Workshop Intelligent Robots and Systems, IROS'91,* Osaka, Japan, pp. 1617-1622.

SURDHAR, J. S., WHITE, A.S., GILL, R., MISTRY, G. (1996). Fuzzy PD control applied to a 1DOF flexible link manipulator in a novel tip feedback sensor based configuration, *Int. Workshop on Advanced Robotics and Intelligent Machines,* University of Salford, Manchester, UK.

TAKAHASHI, Y. (1993). Adaptive predictive control of non-linear time-varying systems using neural network, *IEEE Int. Conf. Robotics and Automation,* pp. 1464-1468.

TAKEYASU, K., GOTO, T., INOYAMA, T. (1976). Precision insertion control and its application, *ASME J. of Engineering for Industry*, pp. 1313-1318.

TOKITA, M., MITOUKA, T., FUKUDA, T.,SHIBATA, T., ARAI, F. (1991). Position and force hybrid control of robotic manipulator by neural network, *Proc. IEEE Int. Conf. Robotics and Automation,* Nice, France, pp. 920-924.

TSAI, C., and ORIN, D.E. (1986). Modified hybrid control for an electro-hydraulic robot leg, *ASME Conf. on Computers in Engineering*, Chicago, IL, USA.

VENKATARAMAN S.T., GULATI, S., BARHEN, J., TOOMARIAN, N. (1992). Parameter learning and compliance control using neural networks, *Proc. 31st Conf. on Decision and Control*, Arizona, USA. pp. 321-325.

VOLPE, R., and KHOSLA, P. (1994). Analysis and experimental verification of a fourth order plant model for manipulator force control, *IEEE Robotics and Automation Magazine*, June Issue, pp. 4-13.

WADA, H., FUKUDA, T., KOSUGE, K., ARAI, F., AND WATANABE, K., (1993). Damping control with consideration of dynamics of environment, *IEEE/RSJ Int. Conf. Intelligent Robots and Systems,* Yokohama, Japan. pp. 1516-1521.

WARWICK, K., MASON, J.D. AND SUTANTO, E.L. (1995). Neural network basis function center selection using cluster analysis. In *Proc. of the American Control Conf.,* Washington, USA, pp. 3780-3781.

WEST, H., and ASADA, H. (1992). A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment, *IEEE Proc. Robotics and Automation*, pp. 251-259.

WHITNEY, D.E. (1977). Force feedback control of manipulator fine motions, *J. of Dynamics, Systems, Measurement, and Control*, pp. 91-97.

WHITNEY, D.E. (1987). Historical perspective and state of the art in robot force control. *Int. J. Robot. Res.* 6(1), pp. 3-14.

XU, Y., HOLLERBACH, J.M., MA, D. (1995). A non-linear PD controller for force and contact transition control, *IEEE Control Systems Magazine*, February Issue, pp. 15-21.

YABUTA, T., and YAMADA, T. (1990). Possibility of neural network controller for robot manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation,* Cincinnati, OH, USA, pp. 1683-91.

ZAMARRENO, J.M., and VEGA, P. (1996). Identification and predictive control of a melter unit from the sugar industry using recurrent neural nets, *Proc. Int. Conf. on Engineering Applications of Neural Networks,* pp. 119-126.

ZHOU, Y. (1991). Position/Force Adaptive Control of a robot with applications in construction, *Proc. American Control Conf.*, pp. 956-961.

APPENDIX A

CONTACT MODEL PARAMETERS

SIMULATED TRAINING ENVIRONMENTS

	k_1(N/m)	k_3(kN/m^2)
Environment 1	100	100
Environment 2	100	500
Environment 3	100	300
Environment 4	0	1000
Environment 5	200	200
Environment 6	0	2000

SIMULATED TEST ENVIRONMENTS

	k_1(N/m)	k_3(kN/m^2)
Environment 7	300	500
Environment 8	500	500
Environment 9	0	3000
Environment 10	0	50
Environment 11	1000	0
Environment 12	2000	0

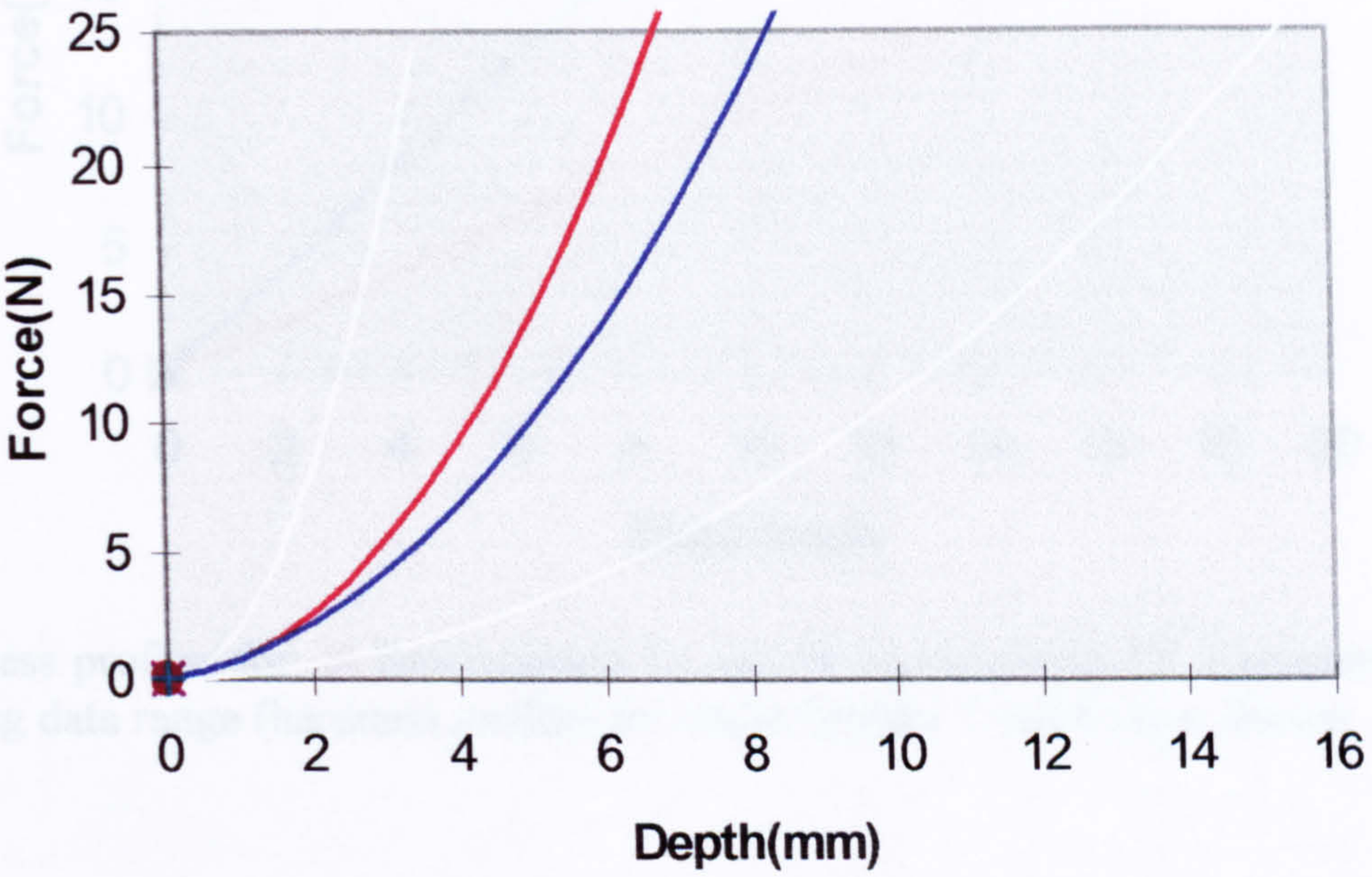
APPENDIX B

ACSL SIMULATION MODEL PARAMETERS

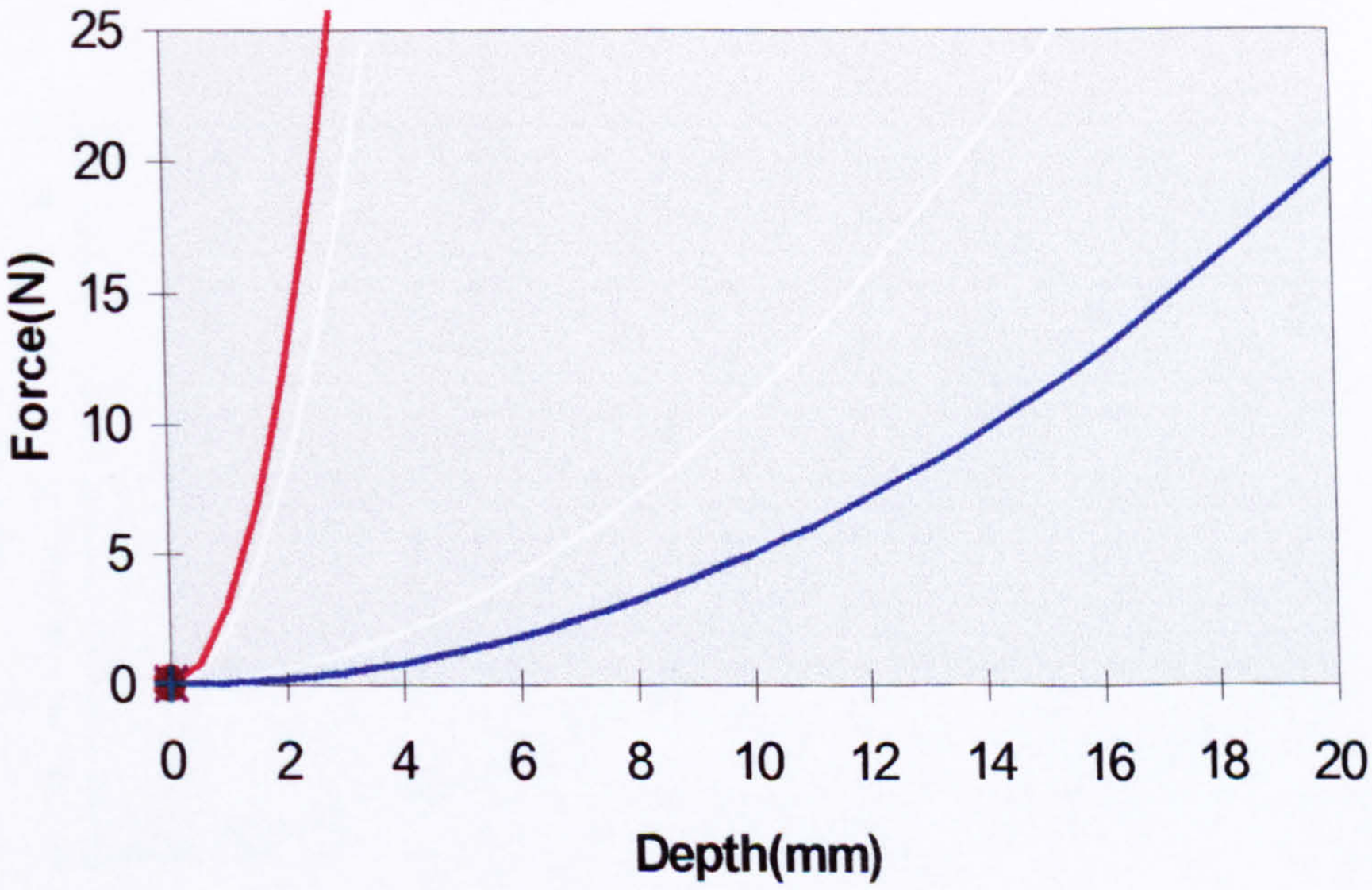
Effective system inertia	$J = 0.001 kgm^2$
Effective system friction	$B = 0.001$ Nm per rad sec^{-1}
Gear ratio	$n = 0.01$
Motor Armature Resistance	$R_a = 1$ ohm
Current amplifier gain	$K_a = 1$
Rotational motion to translational motion gain	$m = 0.1$

APPENDIX C

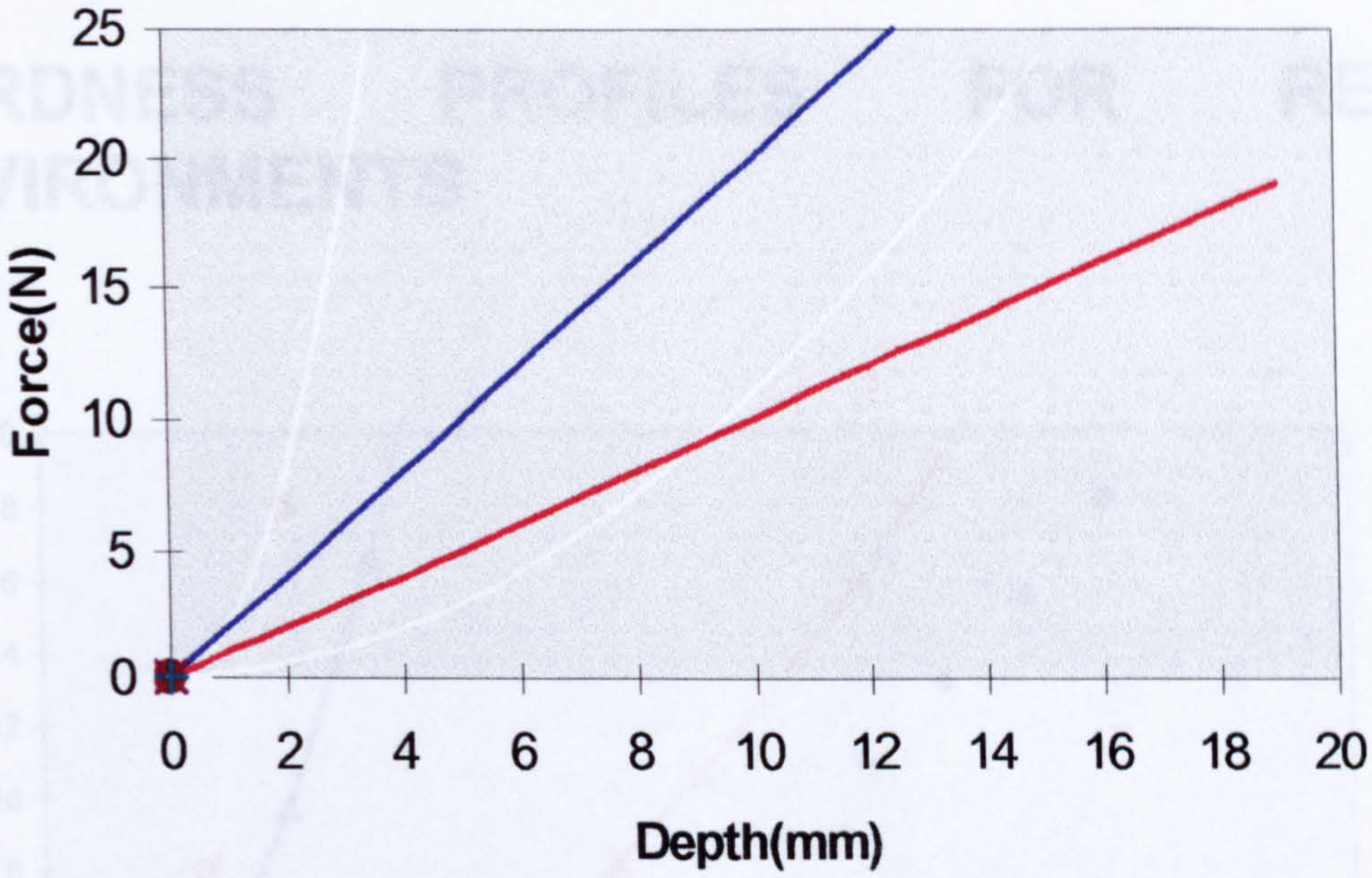
HARDNESS PROFILES FOR SIMULATED ENVIRONMENTS



Hardness profiles for: ■ Environment 7 and ■ Environment 8. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.



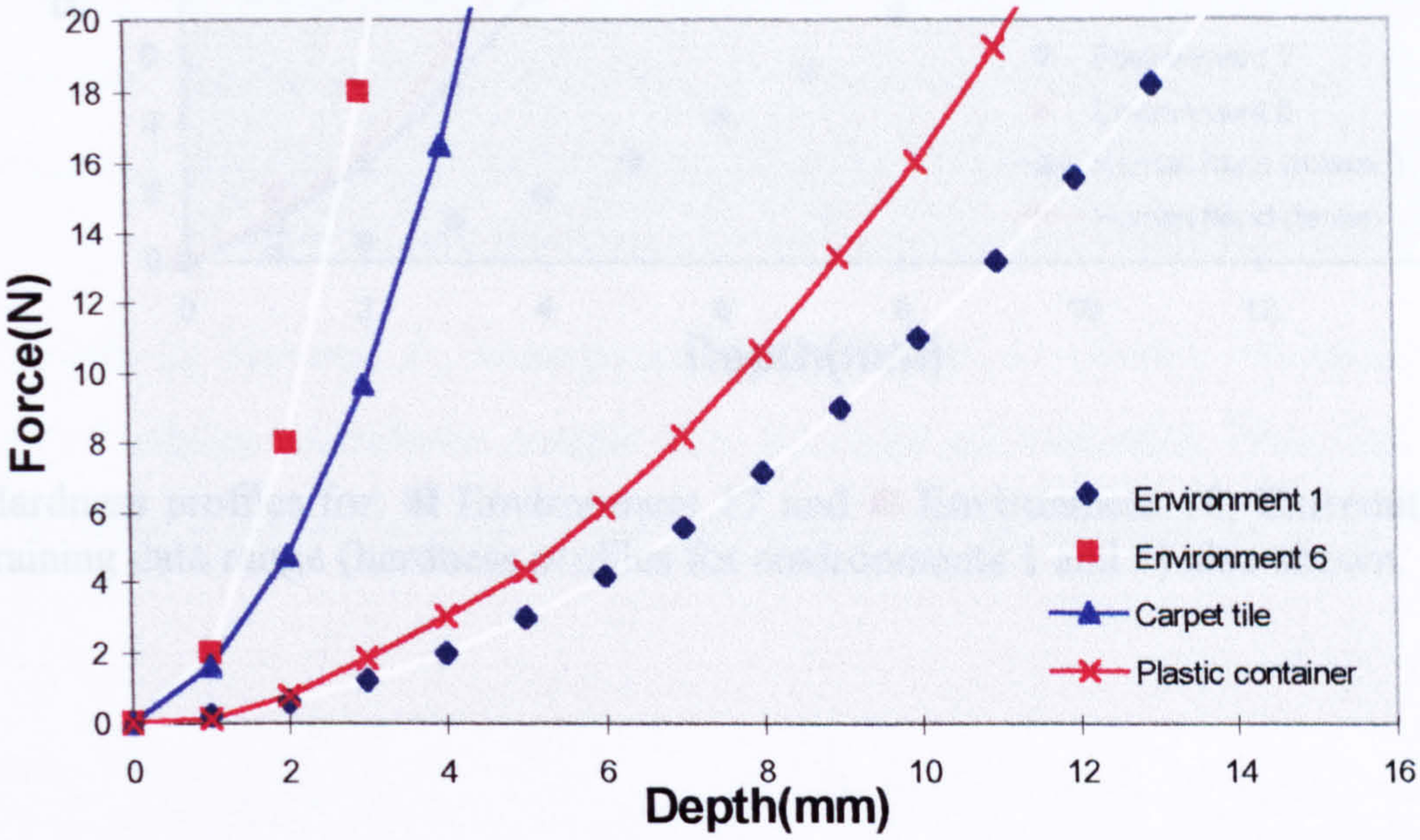
Hardness profiles for: ■ Environment 9 and ■ Environment 10. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.



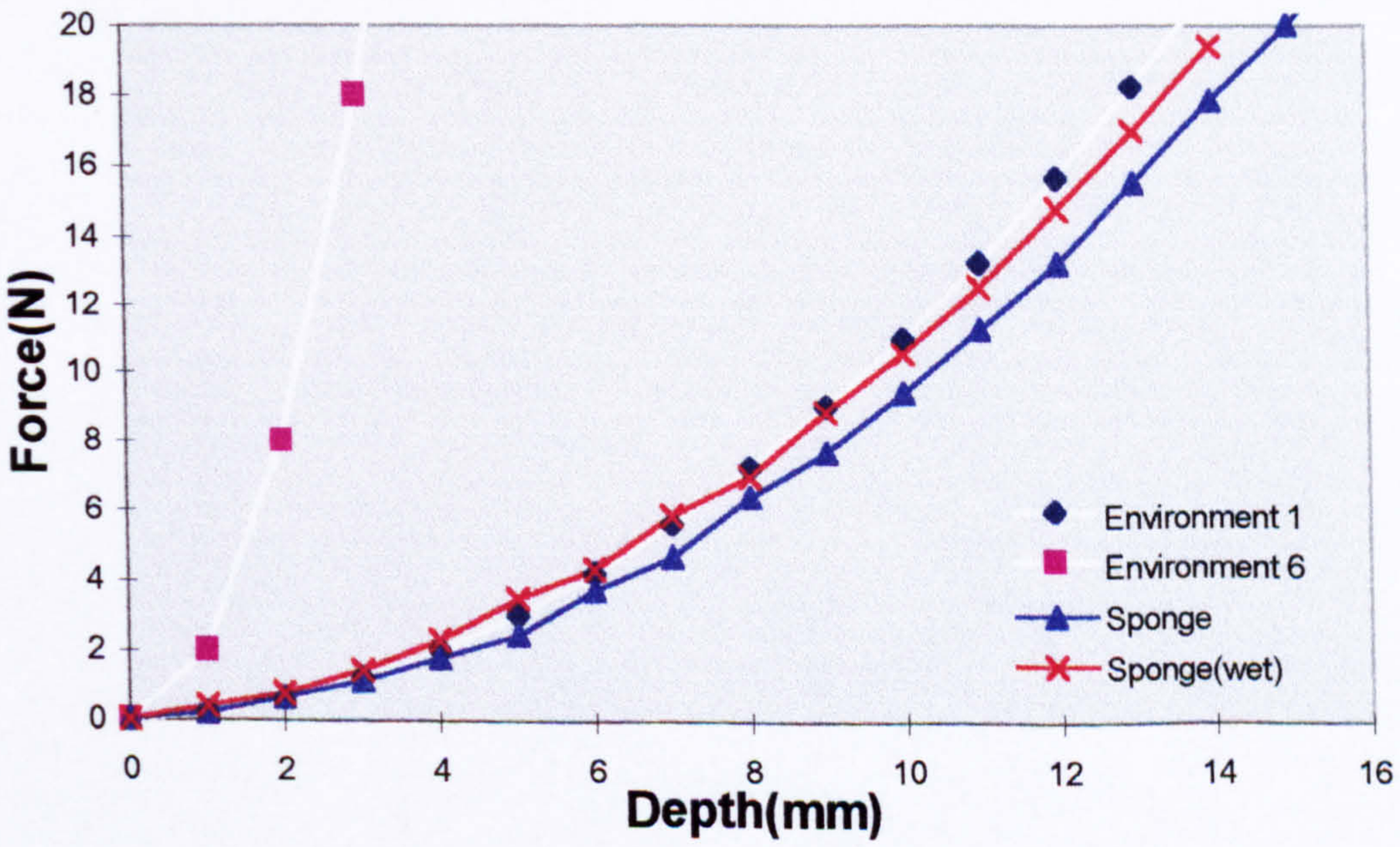
Hardness profiles for: ■ Environment 11 and ■ Environment 12. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.

APPENDIX D

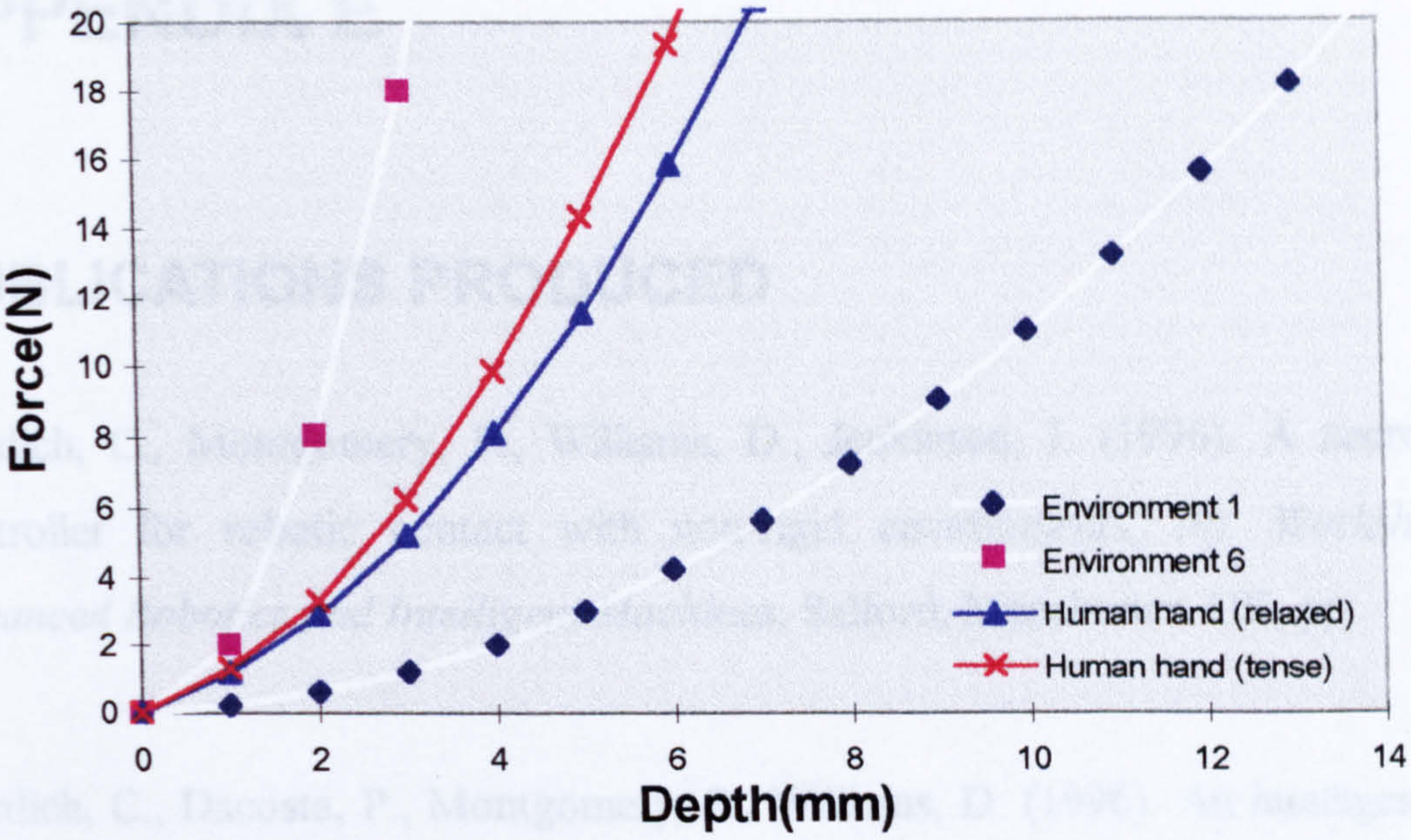
HARDNESS PROFILES FOR REAL ENVIRONMENTS



Hardness profiles for: ■ Environment 13 and ■ Environment 14. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.



Hardness profiles for: ■ Environment 15 and ■ Environment 16. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.



Hardness profiles for: ■ Environment 17 and ■ Environment 18. Extremities of training data range (hardness profiles for environments 1 and 6) also shown.

APPENDIX E

PUBLICATIONS PRODUCED

Kordich, C., Montgomery, P., Williams, D., Jenkinson, I. (1996). A neuro-force controller for robotic contact with non-rigid environments, *Int. Workshop on Advanced Robotics and Intelligent Machines*, Salford, Manchester, UK, pp.

Kordich, C., Dacosta, P., Montgomery, P., Williams, D. (1996). An intelligent force control scheme for robotic contact with non-rigid environments, *Proc. Int. Conf. on Engineering Applications of Neural Networks*, London, England, pp.197-200.