ADVANCED ASSISTIVE CONTROL STRATEGIES FOR SMART HOSPITAL BEDS

By

Huy Hoang Nguyen

A thesis submitted in partial fulfilment of the requirements for the Degree of Doctor of Philosophy

University of Technology Sydney

Faculty of Engineering and Information Technology

2016

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I, Huy Hoang Nguyen, certify that the work in the thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgements

First of all, I would like to express my sincere thanks to the Vietnamese Ministry of Education and Training (MOET) and the University of Technology Sydney (UTS) for offering me a University of Technology Sydney – Vietnam International Education Development scholarship (UTS-VIED) without which I would not have been able to accomplish this doctoral study. I am also grateful to Hanoi University of Science and Technology for allowing me to join the Doctor of Philosophy course.

My heartfelt gratitude is extended to my principal supervisor Professor Hung Nguyen for his excellent scholarly guidance and continuous support of my PhD study and related research. Thanks to his immense knowledge and useful advice, I found my doctoral studies challenging but highly inspiring and enjoyable. I was not only motivated by his academic knowledge but also his patience, and I believe the combination of these qualities contributed significantly to the quality of my research.

Besides my principal supervisor, I would like to extend my sincere gratitude to A.Prof Steven Su and Dr Nghia Nguyen (my co-supervisors) and Prof. Hans von Holst. Their insightful comments and encouragement were much appreciated.

I am also grateful to all my colleagues and the staff in the Centre for Health Technologies and the Royal Institute of Technology, Stockholm who showed me their consideration and understanding and provided me with full support over the duration of my doctoral studies.

Finally, I am most grateful to my parents, my sister Lan, my brother in law Dai and my nephew for their great love and spiritual support over the years. The completion of my thesis owes a great deal to their constant care, faith and encouragement.

Table of contents

CHAPTER 1 . INTRODUCTION ... 1

1.1 PROBLEM STATEMENT ... 1

1.2 AIMS OF THESIS ... 5

1.3 CONTRIBUTIONS ... 6

1.4 STRUCTURE OF THESIS ... 7

1.5 PUBLICATIONS RELATED TO THESIS ... 8

CHAPTER 2 . LITERATURE REVIEW ... 9

2.1 HOSPITAL BEDS ... 9

2.1.1 Definition, structure and classification ... 9

2.1.2 Advantages and disadvantages ... 11

2.2 SMART HOSPITAL BEDS ... 13

2.2.1 Definition and current research ... 13

2.2.2 Advantages and disadvantages ... 16

2.3 ADVANCED CONTROL FRAMEWORK ... 17

2.3.1 Operator detection approaches ... 17

2.3.2 Operator-following strategy ... 20

2.3.3 Advanced low-level control ... 21

2.4 DISCUSSION ... 31

CHAPTER 3 . OPERATOR DETECTION METHODS FOR THE TRACKING FUNCTION OF SMART HOSPITAL BEDS ... 34

3.1 INTRODUCTION ... 34

3.2 SMART HOSPITAL BED STRUCTURE ... 35

3.2.1 Bed frame ... 35

3.2.2 Sensor system ... 36

3.2.3 Control devices ... 37

3.2.4 Power supply system ... 38

3.2.5 Motors and actuators ... 39

3.2.6 Wheels ... 39

3.2.7 Air mattress ... 40

3.3 OPERATOR DETECTION METHODS ... 41

3.3.1 Feature extraction ... 41

3.3.2 Gaussian distribution method ... 44

3.3.3 Neural network classification method ... 45

3.4 OPERATOR-FOLLOWING STRATEGY ... 50

3.5 EXPERIMENTAL RESULTS ... 52

3.5.1 Operator detection performance ... 53

3.5.2 Operator following performance ... 57

3.6 DISCUSSION ... 61

CHAPTER 4 . AN OPTIMAL MULTIVARIABLE PID CONTROL STRATEGY FOR SMART HOSPITAL BEDS ... 63

4.1 INTRODUCTION ... 63

4.2 MULTIVARIABLE DYNAMIC MODEL ... 64

4.2.1 Mathematical model ... 64

4.2.2 Parametric identification process ... 68

4.3 OPTIMAL MULTIVARIABLE PID CONTROL STRATEGY ... 70

4.3.1 Stage 1: Triangular Diagonal Dominance decoupling ... 71

4.3.2 Stage 2: Optimal PID control ... 72

4.4 RESULTS 74
4.4.1 OMPID controller design 74
4.4.2 Experimental results 79
4.5 DISCUSSION 87
CHAPTER 5 . INTELLIGENT MULTIVARIABLE CONTROL STRATEGIES 89
5.1 INTRODUCTION 89
5.2 ADVANCED OPERATOR-FOLLOWING CONTROL STRATEGY 90
5.2.1 Neural network based operator detection 92
5.2.2 Operator-following strategy 92
5.2.3 Neural multivariable low-level control 93
5.3 OPTIMAL MULTIVARIABLE NEURAL NETWORK CONTROL 94
5.3.1 Stage 1: Two-phase diagonal decoupling technique 94
5.3.2 Stage 2: Optimal neural network control design 95
5.4 RESULTS 101
5.4.1 OMNN controller design 101
5.4.2 Experimental results 109
5.5 DISCUSSION 117
CHAPTER 6 . CONCLUSION AND FUTURE RESEARCH DIRECTIONS ... 119
6.1 CONCLUSION 119
6.2 FUTURE RESEARCH DIRECTIONS 122
APPENDIX A. ARITIFICIAL NEURAL NETWORKS 124
APPENDIX B. AUTONOMOUS NAVIGATION ROBOT - TURTLEBOT 132
REFERENCES 137

List of figures

Figure 2.1: Mechanical hospital bed (Audrey 2016) 10

Figure 2.2: Semi – electric hospital bed (Drivemedical 2015) 11

Figure 2.3: Electric hospital bed (Stryker 2013a) 11

Figure 2.4: Zoom Motorised Drive System (Stryker 2013b) 14

Figure 2.5: IntelliDrive Powered Transport System (Hill-Rom 2014) 14

Figure 2.6: I-Drive Power^{TM} System (Linet 2013b) 15

Figure 2.7: The bed wheelchair system (Panasonic 2009) 16

Figure 3.1: The smart hospital bed system 36

Figure 3.2: The sensor system: the encoder (Encoder 2016), ... 37

Figure 3.3: The central control system: the KTA 198 board (Oceancontrols 2009), the motor controller (RoboteQ 2015), the Phidget 8/8/8 board (Phidgets 2012) and the laptop 38

Figure 3.4: The DC motor (NPC 2011) and actuators (Linak 2016) 39

Figure 3.5: The wheel system 40

Figure 3.6: The air mattress 40

Figure 3.7: Laser data processing procedure: 42

Figure 3.8: Feature determination 43

Figure 3.9: Neural network structure 45

Figure 3.10: The hospital bed system and the target for tracking 52

Figure 3.11: Distribution of the two data sets 53

Figure 3.12: The Boundary B_{WG} with θ^{W}_{G} = 0.2, 0.15, 0.1, 0.05 55

Figure 3.13: The Boundary B_{GH} with θ^{G}_{H} = 0.2, 0.15, 0.1, 0.05 55

Figure 3.14: Demonstration of the operator-following task 59

Figure 3.15: Trajectory of the hospital bed and operator 60

Figure 3.16: Linear velocity of the hospital bed and operator 60

Figure 3.17: Angular velocity of the hospital bed and operator 61

Figure 4.1: The smart hospital bed system 65

Figure 4.2: Step response of the smart hospital bed dynamics 70

Figure 4.3: Optimal Multivariable PID control scheme........ 71

Figure 4.4: Optimal PID control method for a velocity subsystem 72

Figure 4.5: Decoupled models with TDD decoupling technique........ 76

Figure 4.6: Root Locus plots for the linear velocity subsystem........ 77

Figure 4.7: Closed-loop step responses of controllers with initial and optimal parameters 78

Figure 4.8: Root Locus plots for the angular velocity subsystem........ 78

Figure 4.9: Closed-loop step responses of controllers with initial and optimal parameters 79

Figure 4.10: The smart hospital bed travels on the granite surface 80

Figure 4.11: The open loop and closed-loop response of the hospital bed system with $v_d = 1\,m/s;\ \omega_d = 1\,rad/s$ 80

Figure 4.12: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $vv_d = 0.4\,m/s;\ \omega_d = 0.4\,rad/s$........ 81

Figure 4.13: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 0.6\,m/s;\ \omega_d = 0.6\,rad/s$ 82

Figure 4.14: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 0.8\,m/s;\ \omega_d = 0.8\,rad/s$ 82

Figure 4.15: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 1\,m/s;\ \omega_d = 1\,rad/s$ 83

Figure 4.16: The smart hospital bed travels on various environmental conditions: . 84

Figure 4.17: System outputs of the smart hospital bed being controlled by OMPID control with $v_d = 1\, m/s;\ \omega_d = 1\, rad/s$ in the case of travelling on the carpet surface 85

Figure 5.1: Advanced operator-following control strategy 91

Figure 5.2: Optimal multivariable neural network control structure for a smart hospital bed 94

Figure 5.3: Optimal neural network control structure 96

Figure 5.4: Neural network control structure 97

Figure 5.5: Opened-loop of the decoupled transfer matrices $P_{DD}(s)$, $P_{1DD}(s)$ and $P_{2DD}(s)$ using the diagonal decoupling technique 103

Figure 5.6: System outputs of three representational sub-systems $P_{DD}(1,1)$, $P_{1DD}(1,1)$ and $P_{2DD}(1,1)$ these being controlled by $ONNCv$. 106

Figure 5.7: System outputs of three representational sub-systems $P_{DD}(2,2)$, $P_{1DD}(2,2)$ and $P_{2DD}(2,2)$ these being controlled by $ONNC\omega$. 109

Figure 5.8: Comparison between OMNN control approach and OMIPD control approach 110

Figure 5.9: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.4\, m/s;\ \omega_d = 0.4\, rad/s$ 111

Figure 5.10: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.6\, m/s;\ \omega_d = 0.6\, rad/s$ 111

Figure 5.11: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.8\, m/s;\ \omega_d = 0.8\, rad/s$ 112

Figure 5.12: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 1\, m/s;\ \omega_d = 1\, rad/s$ 112

Figure 5.13: Output responses of the smart hospital bed being controlled by OMNNC with $v_d = 1\, m/s;\ \omega_d = 1\, rad/s$ in the case of travelling on the carpet surface 114

Figure 5.14: Trajectory of the hospital bed and operator 116

Figure 5.15: Linear velocity of the hospital bed and operator 116

Figure 5.16: Angular velocity of the hospital bed and operator 117

List of tables

Table 3.1: Experimental results of the Gaussian Distribution Method 56

Table 3.2: Mean values of training, validation and testing results utilising four algorithms 56

Table 3.3: Best performance of NN based operator detection using four algorithms 57

Table 3.4: Comparisons of mean values and best performances of testing results for two proposed classification algorithms 57

Table 4.1: Output performance of the linear velocity subsystem (on the granite surface) 83

Table 4.2: Output performance of the angular velocity subsystem (on the granite surface) 84

Table 4.3: Output performance of the linear velocity subsystem (on the carpet surface) 85

Table 4.4: Output performance of the angular velocity subsystem (on the carpet surface) 86

Table 4.5: Output performance of the linear velocity subsystem (on the cement surface) 86

Table 4.6: Output performance of the angular velocity subsystem (on the cement surface) 86

Table 5.1: Results of NN-LM based operator detection algorithm 92

Table 5.2: Output performance of the linear velocity subsystem (on the granite surface) 113

Table 5.3: Output performance of the angular velocity subsystem (on the granite surface) 113

Table 5.4: Output performance of the linear velocity subsystem (on the carpet surface) 114

Table 5.5: Output performance of the angular velocity subsystem (on the carpet surface) 115

Table 5.6: Output performance of the linear velocity subsystem (on the cement surface) 115

Table 5.7: Output performance of the angular velocity subsystem (on the cement surface) 115

List of Abbreviations

2D	:	2-dimensional
ANN	:	Artificial neural network
DC	:	Direct current
DSP	:	Digital signal processing
DV	:	Disparity vectors
FN	:	False negative
FP	:	False positive
G	:	Girth
GaRBF	:	Gaussian radial basis function
GDM	:	Gaussian distribution method
GDX	:	Gradient descent with momentum
H	:	Height
HJB	:	Hamilton-Jacob-Bellman
HSV	:	Hue, saturation and value
IAV	:	Inscribe Angle Variance
IMM	:	Interacting multiple model
LM	:	Levenberg Marquardt
LRF	:	Laser range finder
MLP	:	Multilayer perceptron
NN	:	Neural network
OFV	:	Optical flow vectors
OMPID	:	Optimal multivariable proportional integral derivative
OMNNC	:	Optimal multivariable neural network controller
ONN	:	Optimal neural network
ONNCv	:	Optimal neural network controller for linear velocity subsystem
ONNCω	:	Optimal neural network controller for angular velocity subsystem
OPID	:	Optimal proportional integral derivative
P	:	Proportional
PD	:	Proportional derivative
PID	:	Proportional integral derivative
RBF	:	Radial basis function

RANSAC	:	Random sample consensus
RP	:	Resilient backpropagation
SCG	:	Scaled conjugate gradient
Sen	:	Sensitivity
Spec	:	Specificity
TDD	:	Triangular Diagonal Dominance
TP	:	True positive
TN	:	True negative
UTS	:	University of Technology Sydney
W	:	Width

Abstract

The intention of employing hospital beds is to bring comfort to the hospitalised people and support the clinical staff in patient-care activities. Over many years, hospital beds have been upgraded from simple beds created by crude stretchers to smart beds equipped with various advanced functionalities such as motorised wheels and an intelligent steering system. With these features, patient transportation becomes much easier and safer than before. In addition, the number of manual handling tasks and injuries of nurses in relation to the transfer of patients are significantly reduced.

However, a drawback of the current smart hospital beds is that their steering systems are controlled by human users. In recent years, the increasing number of people being hospitalised (especially obese patients) has led to work overload problems for nurses. Stress, exhaustion and distraction of staff are some of the effects of the nursing work overload problem. These are also major factors causing unpredictable risks for both patient and medical staff during transportation within hospitals.

An integration of an operator-following function not only allows the smart hospital bed to deal with the problem of human control but also creates an innovative solution in terms of patient transport. To achieve this, a 3-stage operator-following control strategy is required for the bed system. In the first stage, the operator is identified by the utilisation of a target detection algorithm. In the second stage, based on the information obtained relating to the operator, operator-following controllers generate desired velocities for the smart hospital bed. In the final stage, a low-level controller drives the bed system to track the desired velocities.

In relation to operator detection, two approaches, consisting of the Gaussian Distribution Method (GDM) and the Artificial Neural Network (ANN), are investigated and developed for the smart hospital bed. Technically, the GDM classifier is based on a threshold condition of a Mahalabonis distance between a testing point and the training data set. On the other hand, the ANN classifier is based on a construction of a neural network model through a training procedure. The experimental results show that the operator detection performance of the Neural Network - Levenberg Marquardt (NN-LM) method is better than that of the GDM method (92.41% sensitivity and 89.8% specificity vs 91% sensitivity and 88.5% specificity).

In terms of the low-level control, two approaches including an Optimal Multivariable Proportional Integral Derivative (OMPID) control method and an Optimal Multivariable Neural Network (OMNN) control approach are introduced and designed for the smart hospital bed. In theory, the OMPID control algorithm is a combination of a Triangular Diagonal Dominance (TDD) decoupling technique and an Optimal Proportional Integral Derivative (OPID) control design. Meanwhile, the OMNN control strategy combines a two-phase diagonal decoupling technique and an Optimal Neural Network (ONN) control design. Real-time implementation indicates that the OMNN controller drives the smart hospital bed to track the desired velocities with higher accuracy, smaller overshoot, shorter rise time and settling time than the OMPID controller.

From the results obtained from the operator detection algorithms and the low-level control algorithms, an advanced operator-following control strategy is developed for the smart hospital bed. This is a combination of the neural network based operator detection method, the Proportional Integral Derivative (PID) based operator-following strategy and the intelligent multivariable low-level control approach. The Levenberg-Marquardt (LM) learning algorithm is chosen to train the neural network classifier. Two PID controllers are utilised to minimise the distance and angle error between the operator and the bed system. The Optimal Multivariable Neural Network control strategy takes the responsibility of stabilising the overall system under the effect of uncertainties. The experimental results serve to show that the operator-following performance of the smart hospital bed using the advanced operator-following control strategy is more stable and efficient than that of the bed system without the proposed approach.

CHAPTER 1 . INTRODUCTION

1.1 PROBLEM STATEMENT

Hospital beds are medical devices that are principally developed for hospitalised people. Compared to regular beds, hospital beds have many advantages such as the capabilities of changing the patient's posture, adjusting the bed height to assist the patient to get in or out of bed, and reducing the risk of falling out of bed through a side rail system. Over the years, hospital beds have been upgraded with various functionalities to provide comfort for patients and convenience for hospital employees. Despite great improvements to the different functionalities of hospital beds, they still pose several problems, especially in relation to patient transport.

One of these problems is that a patient has to be switched between many different beds of various sizes on their journey from an ambulance to a recovery room. This is inconvenient and time consuming for nurses particularly when they encounter special patients such as obese patients and critical neurosurgical patients. For obese patients, the caregivers always need to exert more energy and cooperate with at least one other staff member to complete the patient transferral task. For critical neurosurgical patients, the hospital employees must be extremely careful to ensure patient safety.

Another problem is that the clinical staff have to perform many manual handling operations to complete the patient transport. Many studies show that patient handling tasks are recognised as one of the major causes of musculoskeletal injuries among the nursing workforce (Byrns et al. 2004; Nelson & Baptiste 2004; Pompeii, Lipscomb & Dement 2008; Retsas & Pinikahana 2000). According to a study of Pompeii *et al*, one-third of all musculoskeletal injuries resulted from patient handling activities (2009).

In order to deal with these drawbacks, a smart hospital bed is considered to be a promising solution. Unlike the conventional hospital bed, motorised wheels and an intelligent steering system are integrated into this kind of bed. With these features, the smart hospital bed is an efficient and safe way for the hospital employees to transport patients. In addition, it not only minimises bed switching but also reduces manual handling tasks and nursing injuries.

In the literature, various smart hospital beds have been proposed; however, all the steering systems of the current smart beds are controlled by human users. A recent article indicates that human mistakes are major contributing factors in terms of the potential and unexpected risks of patient transport within hospitals (Chao et al. 2015). Patient transportation will not be guaranteed if hospital staff experience work overloads.

Owning an operator-following ability supports the smart hospital bed to deal with the problem of human control. By tracking hospital staff, the smart hospital bed has the capability to travel to a destination without the steering of the user. Due to this, unexpected risks in patient transport can be substantially decreased. Moreover, this function enables the smart bed to collaborate with an autonomous navigation robot to perform the patient transport. The robot functions to guide the bed to the destination. It can be seen that this collaboration not only assists the caregivers to fully concentrate on patient-care activities but also creates an innovative solution with respect to patient transport. To the best of our ability, we have not found other articles which focus on this innovation.

To enable the operator-following function, a 3-stage control strategy is required for the smart hospital bed. In the first stage, an operator detection algorithm is implemented to allocate the operator. Then, in the second stage, the obtained information of the target is adopted as the control input for an operator-following controller in the second stage. Depending on each situation, this controller is appropriately designed to meet the requirements of the following behaviour such as tracking with a fixed distance, moving side-by-side or simultaneously following the operator and avoiding obstacles. In the last stage, there is a low-level controller playing the role of stabilising the overall system.

The operator-following performance of the smart hospital bed heavily depends on the operator detection method and the low-level controller. The operator detection enables the smart bed to classify the operator in a complex space consisting of the target and non-target objects. The more accurate the detection method is, the more efficient the tracking performance is. The low-level controller guarantees the stability of the overall system under uncertainties. The lack of this controller leads to the instability and inefficiency of the bed system since various uncertain factors impact the bed system such as internal system uncertainties, the surface friction or the weight of the patient.

In terms of the operator detection, a vision system is a popular approach for developing operator detection algorithms such as in the works of Hyukseong *et al* (2005), Zhichao *et al* (2007), Perng *et al* (2012), Motokucho *et al* (2014a), Li *et al* (2015). Although this approach is successful in detecting the operator, the vision system usually suffers from huge computational complexity and its performance can be easily affected by the settings of the camera and uncertainties (Chen Tun et al. 2011). On the other hand, many studies show that the laser scanner can meet the successful performance of following an operator due to its advantage of high accuracy and reliability (Chen Tun et al. 2011; Kmiotek & Ruichek 2008; Lee et al. 2006; Mozos, Kurazume & Hasegawa 2010).

With respect to the low-level control design, technically, there are three major problems that should be considered. The first problem is the dynamic modelling of the smart hospital bed. In the literature, one of two popular ways for constructing the dynamic model is derived from the utilisation of the Euler-Lagrange formulation. In fact, this approach is appropriate for simulation purposes since it is difficult to precisely calculate the parameters of the Euler-Lagrange dynamic equation. The alternative approach is the employment of techniques for approximating and identifying the dynamic model. Nevertheless, this approach frequently utilises complicated computational algorithms therefore it is limited in terms of real-time application.

A coupling effect is another problem which should be considered when treating the dynamics of the smart hospital bed as a multivariable system. Technically, the coupling effect is the crossing interaction between the system's inputs and outputs. It is one of the factors leading to instability of the overall system.

Diagonal decoupling and triangular decoupling techniques are effective solutions for coping with the coupling effect problem. Technically, both decoupling techniques transform a multivariable system into independent scalar systems. However, the resulting system compensated by the diagonal decoupler has a diagonal structure whereas that compensated by the triangular decoupler has a triangular structure. Moreover, the decoupling approach completely eliminates the coupling effect problem while the triangular approach reduces it in such a way that off-diagonal elements of the compensated system do not influence the stability of the overall system. Recently, the application of these techniques on the smart hospital bed has been lacking.

The last problem is the choice of an appropriate control method for dealing with system uncertainties. In the literature, several advanced control methods have been utilised for controlling wheeled mobile systems including the Lyapunov-based control methods, adaptive control methods and the sliding mode control method. To achieve robust performance, the Lyapunov-based control and adaptive control methods require heavy and complex computation. On the other hand, the sliding mode control suffers two main disadvantages consisting of the chattering phenomenon (Slotine & Sastry 1983) and "the difficulty involved in the calculation of what is known as the equivalent control" (Ertugrul & Kaynak 2000). Moreover, few experimental results have been reported to confirm the effectiveness of these control approaches on the wheeled mobile systems such as electrical power wheelchairs and autonomous mobile robots.

Recently, neural networks have offered various efficient solutions for coping with control problems with uncertainties since they have the advantages of learning by experience and mapping nonlinear functions. In most neural network control schemes, the intelligent controller is directly connected to the system. However, the smart hospital bed is an unknown system therefore it faces the problem of an unknown plant Jacobian. In theory, the definition of the plant Jacobian is the partial derivative of a system's outputs in relation to its inputs. The calculation of the plant Jacobian is required when utilising back propagation to train the neural network controller. In the literature, findings related to the solution of the plant Jacobian problem on the smart hospital bed are absent.

1.2 AIMS OF THESIS

In accordance with the problem statement above, this doctoral study focuses on developing advanced assistive control strategies for supporting the smart hospital bed during the transportation of a patient. Hence, three particular aims are investigated and given as follows.

The first purpose is about the development of an operator detection algorithm with a high accuracy. A feature extraction is firstly employed to remove redundant information from environmental data and determine features of target and non-target objects in the remaining data. Subsequently, the operator is identified by a classification method. In this study, two classification approaches consisting of the GDM and the ANN classification methods are investigated and implemented.

The second purpose is to construct a hospital bed dynamic model and design a multivariable control strategy for dealing with the coupling effect problem and tracking the desired velocities. Regarding the bed as a linear multivariable system with uncertainties, its approximated dynamic model is obtained via a parametric identification process. A TDD decoupler is calculated to reduce crossing effects between the system's input and output variables and to simplify a multivariable system into two independent scalar systems. Following this, an Optimal PID control method is applied for each independent scalar system.

The last purpose is to develop intelligent multivariable control strategies for supporting the bed performance during its operation. One of proposed control approaches is an advanced operator-following control strategy including 3 design stages. The aim of the first design stage is to obtain information related to the operator whereas that of the second design stage is to control the operator-following behaviour. The last stage is derived from the development of an optimal neural network low-level controller which is also the alternative intelligent control strategy.

For the advanced low-level controller, this is a combination of a two-phase diagonal decoupling technique and an optimal neural network control method.

1.3 CONTRIBUTIONS

In terms of smart hospital bed development, this thesis offers the following key contributions:

(i) The proposal and implementation of a neural network based operator detection algorithm for the smart hospital bed. Experimental results confirm that this method is applied successfully on the hospital bed and its accuracy is high.

(ii) The investigation and reconstruction of a hospital bed dynamic model based on a parametric identification process. The obtained dynamic model plays a crucial role for designing advanced control strategies for the bed system.

(iii) A multivariable control strategy is proposed and developed for smart hospital beds. This approach is a combination of a TDD decoupling technique and an OPID control method. Experimental results show that the proposed strategy has been successfully implemented in the smart hospital bed system. The coupling effect is effectively reduced to an acceptable level.

(iv) An optimal multivariable neural network controller is designed for the smart hospital bed under the effect of uncertainties. The design combines a two-phase diagonal decoupler and optimal neural network controllers. Real-time experiments demonstrate that the advanced controller not only significantly reduces the crossing interaction between the system's inputs and outputs but also guarantees the robustness of the overall system.

(v) An advanced operator-following control strategy is proposed and implemented to enhance the operator-following performance of the smart hospital bed. In this approach, a neural network based operator detection method is firstly adopted to detect the operator from the obtained environmental data. Accordingly, information related to the operator is utilised as control inputs for two PID controllers which play a role of minimising tracking errors of the operator-following performance. Finally, the intelligent multivariable control algorithm is implemented to let the system track desirable velocities. Experimental results confirm that the overall system is stable in the presence of uncertainties.

1.4 STRUCTURE OF THESIS

This thesis contains six chapters, an appendix and references. Each chapter of the thesis is organised as follows:

- Chapter 2 contains the literature review related to the topic. It commences with a background for conventional hospital beds. Then a current research review of smart hospital beds related to the patient transport problem is presented. Following this, the chapter introduces an advanced control framework for the smart hospital bed. At the end of the chapter, a discussion about the smart hospital beds and the advanced control framework serves to provide the motivation for this study.
- Chapter 3 describes operator detection algorithms for the tracking purpose of a smart hospital bed. Section 3.2 describes the system structure of the smart hospital bed which is employed throughout this thesis. After this, two target detection algorithms consisting of the Gaussian Distribution Method and the Neural Network classification method are provided. This chapter also introduces an operator-following control strategy for the smart hospital bed. Experimental results and a discussion conclude the chapter.
- Chapter 4 presents an Optimal Multivariable PID control strategy for smart hospital beds. A multivariable dynamic model of the bed system is firstly approximated via a parameter verification procedure. Then the multivariable control algorithm with two stages: stage 1 – a TDD decoupling technique and stage 2 – an Optimal PID control design – are described. Experimental results are displayed in the next section. The chapter concludes with a discussion about the proposed multivariable control strategy.

- Chapter 5 details intelligent multivariable control strategies for smart hospital beds. In section 5.2, an advanced operator-following control algorithm is presented. Subsequently, an advanced low-level control method combining a two-phase diagonal decoupling technique and an optimal neural network control design is introduced in the next part of the chapter. The end of the chapter focuses on experimental results and discusses the proposed approaches.
- Chapter 6 offers the general conclusion for this doctoral study. In addition, the final chapter introduces future research approaches for the field of operator detection and some potential control methods based on novel decoupling techniques.

1.5 PUBLICATIONS RELATED TO THESIS

The following conference papers have been presented during the doctoral research:

- Huy Hoang, N., Tuan Nghia, N., Clout, R., Gibson, A. & Nguyen, H.T. 2013, 'Development of an assistive patient mobile system for hospital environments', *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp. 2491-4.
- Huy Hoang, N., Tuan Nghia, N., Clout, R. & Nguyen, H.T. 2014, 'An advanced control strategy of an electrical - Powered hospital bed', *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pp. 1190-3.
- Huy Hoang, N., Tuan Nghia, N., Clout, R. & Nguyen, H.T. 2015, 'A novel target following solution for the electric powered hospital bed', *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 2572-5.

CHAPTER 2 . LITERATURE REVIEW

2.1 HOSPITAL BEDS

2.1.1 Definition, structure and classification

A hospital bed is essentially designed for people who are hospitalised during medical treatment. Over the years, the hospital bed has been transformed from a simple bed created by crude stretchers to a multifunctional bed equipped with various features e.g. wheels, side rails and adjustable body parts. In general, the current hospital bed has some common features consisting of a sturdy flexible frame, rails on the side and wheels. These are distinctive features of the hospital bed as opposed to the regular bed.

The flexible frame feature allows the hospital bed to fully adjust its body parts such as the head, middle and foot in a wide range of ways. In addition it also assists the hospital staff in patient-care procedures. Usually the bed frame is divided into several sections which can be moved in a number of different ways. Some auxiliary sections are probably added into the bed to provide precise adjustments for special bed positions including Trendelenburg, reversed Trendelenburg or chair positions. In the Trendelenburg position, the body of the patient is supine and the head is tilted down, allowing the patient’s feet and legs to remain above the level of the heart (Bridges & Jarquin-Valdivia 2005). Meanwhile, the reverse Trendelenburg position is the opposite position of the Trendelenburg position.

Another typical component of the hospital bed is the side rail system which is designed to protect the patient from falling accidents. Depending on each particular procedure, the guardrails are flexible and can be moved up or down. For instance, the side rails are raised in the night time to serve as protection for the patient or are lowered to assist the clinical staff when transferring the patient between the bed and the other medical equipment. There are various types of side rails to serve different purposes. Some are simply to prevent the patient from rolling out whereas others have equipment that can aid the patient without physically confining the patient to the bed (Wikipedia 2010).

Another feature is the wheel system. Being equipped with a set of wheels enables the bed to easily travel within the hospital environment and makes the patient transportation between two wards simpler. In order to ensure patient safety, the medical staff can use the brake system to stop the bed while transporting the patient or to lock the wheel when performing the patient transfer. Sometimes the auxiliary driving wheel is placed under the bed to meet the requirement of high performance. This wheel allows the bed to easily turn around its own axis and steer in narrow spaces.

Beside the most essential components, the hospital bed has some other equipment including an over bed table, mattress and various kinds of linen or pads which are used on the bed. The over bed table is utilised for mealtimes and as work surfaces for both hospitalised people and the hospital staff. Technically, it is located in the middle side of the bed, pulled over the bed and adjusted to various heights according to patient demand. The mattress is designed to withstand frequent soiling without trapping in moisture that can lead to bacterial growth (Etolen 2003).

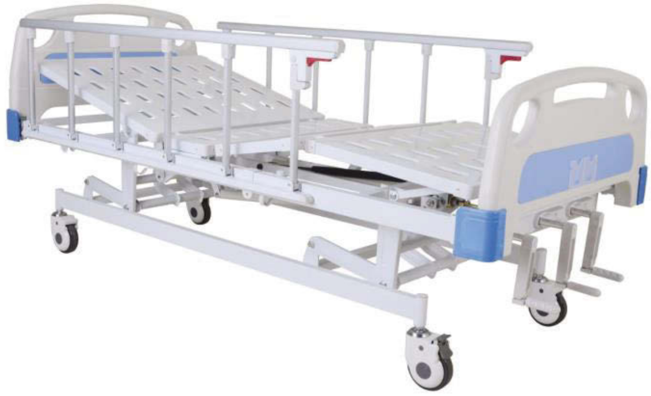


Figure 2.1: Mechanical hospital bed (Audrey 2016)

Nowadays, there are three common types of hospital bed including the mechanical bed, semi-electric bed and full-electric bed. If these beds are compared, the flexible frame and the height of the mechanical bed are adjusted by turning a manual crank at the foot of the bed while those of the electric bed are adjusted by a control system with various actuators. The semi-electric bed utilises two motors, one to raise the head and the other to raise the foot, however there is still the requirement of a manual crank to change the height of the bed (Wikipedia 2010). Demonstrations of the hospital beds are shown in Figure 2.1 to 2.3.

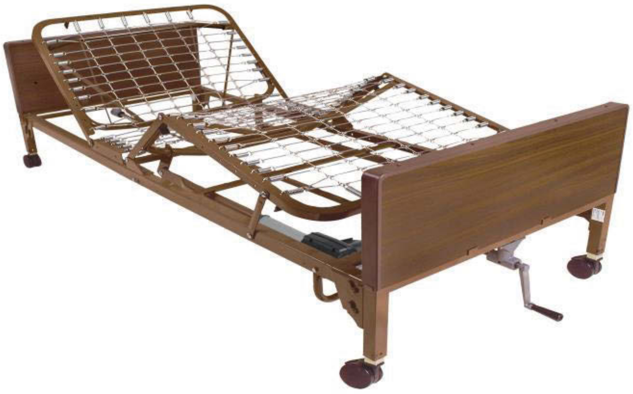


Figure 2.2: Semi – electric hospital bed (Drivemedical 2015)

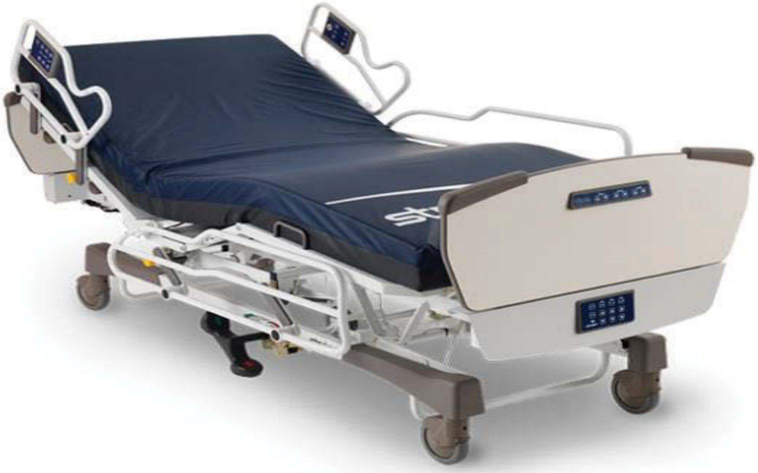


Figure 2.3: Electric hospital bed (Stryker 2013a)

2.1.2 Advantages and disadvantages

Featuring many outstanding qualities in comparison to regular beds, the hospital bed aims to maximise comfort for hospitalised people and provide convenience for the hospital staff during the medical treatment. Generally, the hospital bed offers some important advantages which are listed as follows:

- The hospital bed assists the patient to change position. Spending long periods of time in bed can result in the patient facing the problem of pressure ulcers or bedsores. In order to deal with this problem, the bed enables the hospitalised patients to change the bed position. This adjustment allows the patients to reduce the pressure between the bed surface and particular parts of their body. Moreover, it improves the patients' blood circulation during periods of time in bed.

- With the side rail system, the hospital bed protects the patient from falling accidents and makes them feel secure while lying on the bed. For some special patients such as patients with dementia or other cognitive impairments, guardrails are important to reduce the potentiality of injuries due to falls. In addition, the bed rail system is useful for some patients who seek to re-position themselves.
- Easy adjustment of the bed height and raising the patient higher or lower simplifies the problem of patients getting in or out of bed. Through such means, patients can easily sit up and get out of bed. For some special patients with hip, knee and back problems, the hospital bed can be transformed into a chair position which can assist the patients to comfortably get into a standing posture.
- The medical bed supports the hospital staff in various procedures when they face the problem of work overload and patient handling tasks which can lead to musculoskeletal injuries. In the case of transporting patients between two wards, the auxiliary driving wheel makes it easy for the hospital staff to steer the bed. In terms of patient care, the hospital bed can lift up the patient to a position which is suitable for the caregivers to perform their work without straining their bodies.

Despite having many advantages, there are still some drawbacks of the hospital bed. In particular, this study is concerned with the drawbacks of the bed in relation to patient transportation. Generally, a patient is moved through many different beds of various sizes and functionalities on their journey from the ambulance to the recovery ward, and hospital employees have to perform a number of manual handling operations to complete the transport. This is unsafe and time consuming, especially in relation to obese patients and critical surgery patients. In addition, patient handling activities such as lifting heavy patients and pushing the hospital bed have been identified as major causes of nursing injuries.

For the manual and semi-electric hospital beds, the proposed problem of special patient transport is a big issue since the caregiver must not only exert a high level of care but also expend more energy to complete the task. For the electric hospital beds, despite many sections of the hospital bed being motorised, its wheel system is not motorised for transportation. The only improvement of its steering system is the integration of the auxiliary fifth wheel which enables the staff to easily steer the bed to the destination. However, in the case of serving special patients, the caregiver still has to cooperate with at least one other staff member and expend more energy to move the bed.

This section presents several fundamentals about the current hospital bed including a definition, typical features, advantages and classification. It can be seen that the hospital bed is indispensable for patients and nurses during medical treatment. Concerning the patient transport problem, this section analyses the disadvantages of the current hospital bed when transporting the patient within the hospital environment. The current hospital bed should accordingly be upgraded to overcome its drawbacks. The smart hospital bed, a next generation in terms of the hospital bed, is a promising solution for the problems of the current hospital bed.

2.2 SMART HOSPITAL BEDS

2.2.1 Definition and current research

A smart hospital bed is the next generation in terms of the conventional hospital bed. It is equipped with advanced technologies such as a sensory system, vision system, central processing unit, wireless connection and motorised wheels. This section focuses on the current research into the smart hospital bed which is related to the improvement of patient transport.

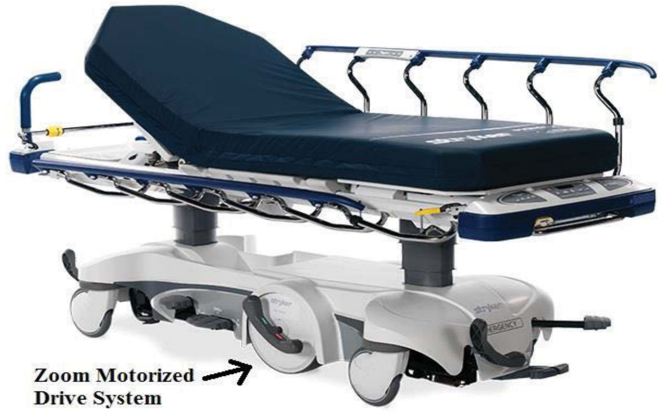


Figure 2.4: Zoom Motorised Drive System (Stryker 2013b)

To solve the problem of transporting patients, the smart hospital bed is equipped with an intelligent driving system. Various driving systems have been proposed in the literature. The Zoom Motorised Drive System of Stryker is one of these systems. This system is designed to virtually eliminate the strenuous pushing and pulling that occurs during the transportation of heavy patients (Stryker 2013c). Compared with the normal auxiliary driving wheel, the Stryker's Zoom system reduces physical intensity by 42% (Stryker 2013d) and the mean L4/L5 spinal disc compression force by 42% on incline transition (Stryker 2013e). Demonstration of the Zoom Motorised Drive System is shown in Figure 2.4.

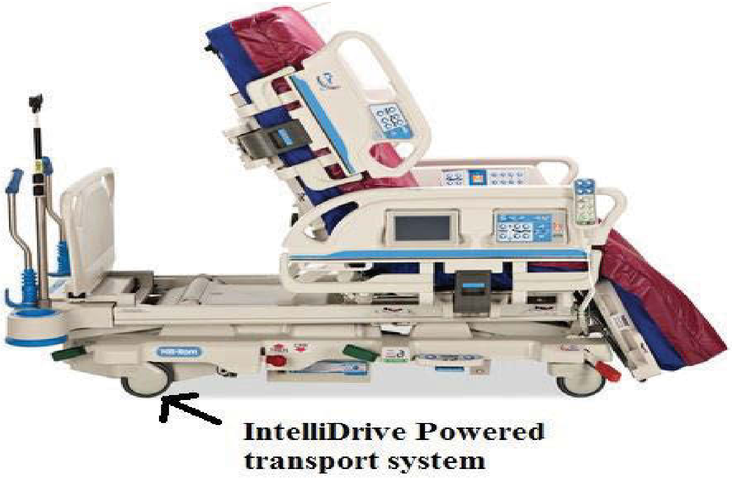


Figure 2.5: IntelliDrive Powered Transport System (Hill-Rom 2014)

Another driving system is known as IntelliDrive Powered Transport, which is proposed by Hill-Rom. Similar to the Stryker's Zoom system, the Hill-Rom's IntelliDrive system not only enables the safe and easy transport of patients throughout the hospital; it also reduces the number of hospital staff needed for patient transport (Hill-Rom 2015). Besides this, it has some particular advantages, including the capabilities of precise turning, manoeuvring, stopping and navigating ramps with automatic deceleration on the decline and acceleration on the incline. Figure 2.5 shows a demonstration of the IntelliDrive Powered Transport system.

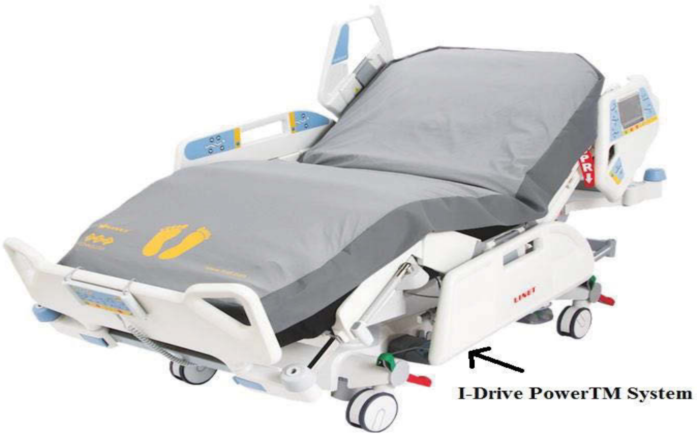


Figure 2.6: I-Drive Power^{TM} System (Linet 2013b)

Aiming to maximise safety during patient transport, Linet developed a new technology in terms of motorised in-bed patient transport called I-Drive Power^{TM} (Linet 2013a). Unlike the two drive systems above, the I-Drive Power^{TM} can provide excellent performance in relation to transporting patients even under extreme load conditions when driving or turning. In addition, the intelligent accelerator optimises the drive speed in various drive situations such as during long travel periods or in narrow spaces. In order to avoid the risk of self-activation, a Safety Sense^{TM} system is designed to work as a power on/off button and driving handle. A demonstration of the I-Drive Power^{TM} system is illustrated in Figure 2.6.

Another solution is presented by Panasonic. This manufacturer developed a 2-in-1 system for patient transport, called a bed wheelchair. In this system, a bed part is detached from the main bed, then it is actively transformed into a wheelchair. While in wheelchair mode, the bed has the ability of detecting and avoiding obstacles. Moreover, the wheelchair is equipped with a manual electronic control system; hence the patient is able to easily transport themselves between two wards without the need for assistance from hospital staff. This design not only supports the patient (especially elderly or disabled patients), it gives them the freedom to get in and out of bed on their own and it decreases the workload of the caregiver (Hornyak 2009). However, it is not suitable for obese patients over 200 kilos. Figure 2.7 shows the Panasonic bed wheelchair system.

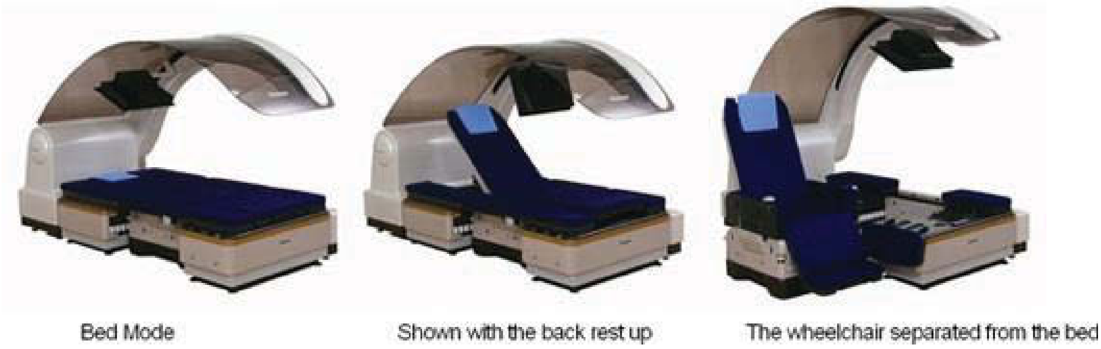

Bed Mode
Shown with the back rest up
The wheelchair separated from the bed

Figure 2.7: The bed wheelchair system (Panasonic 2009)

2.2.2 Advantages and disadvantages

Being equipped with motorised wheels and an intelligent driving system, the smart hospital bed efficiently supports nurses in transporting a patient. The clinical staff can easily steer the bed without the need for expanding large amounts of energy or collaborating with other staff. Therefore, the bed switch process is minimised. In addition, the manual handling tasks and nursing injuries during patient transportation are reduced significantly.

However, it can be seen that the current intelligent steering systems of the smart hospital beds are controlled by human users. In fact, human control is a major contributing factor to the potential and unexpected risks of patient transport within hospitals (Chao et al. 2015). Accordingly, the performance of patient transport will not be guaranteed if the hospital staff encounter work overload conditions.

A smart hospital bed with an operator-following function is an innovative solution to solve the human control problem in patient transportation. With this function, the smart bed transports a patient to a destination through tracking and following a nurse or an autonomous robot. These operators take the responsibility of navigating for the bed. Hence, unpredictable accidents caused by the hospital employee can be substantially reduced. Additionally, this function enables the staff to completely focus on patient-care activities when the bed follows an autonomous robot.

2.3 ADVANCED CONTROL FRAMEWORK

In fact, there are various dynamic factors impacting on the bed's performance such as internal system uncertainties, environmental conditions (hard platform, glass, high friction way), external expected pulling and pushing forces or even the obvious case of a patient lying on the bed (especially with heavy patients, over 200kilos). These factors lead to the instability and inefficiency of the overall system. Therefore it is necessary to develop an advanced control framework for the smart hospital bed.

In terms of operator-following control design, the advanced control framework for the smart hospital bed has three main stages containing operator detection, operator-following strategy and advanced low-level control. In the first stage, an operator detection algorithm is implemented to allocate the operator. Accordingly, obtained information of the target is adopted as control inputs for an operator-following controller in the second stage. Depending on each situation, this controller is appropriately designed to meet the requirements of the following behaviour. In the last stage, there is an advanced low-level controller playing the role of stabilising the overall system.

In the literature, very little research has been done on the advanced control framework for the smart hospital bed. This review therefore focuses on research related to wheeled mobile robots and electric powered wheelchairs since their mechanical systems are similar to that of the smart bed which is comprised of two driving wheels.

2.3.1 Operator detection approaches

2.3.1.1 Vision system

A vision system is one of the popular approaches to develop operator detection algorithms. In a work of Hyukseong *et al* (2005), the researchers proposed an efficient object detection method by using two independently moving cameras. Technically, the colour histogram analysis was employed to identify the position of the operator in the input image. Then the distance between the operator and the robot was derived from two moving cameras-based distance estimation procedure. Similar to Hyukseong *et al*, Perng *et al* (2012) utilised colour features for operator recognition and a combination of two vision systems for locating the operator position. Instead of using the colour histogram analysis, Perng *et al* 's approach was based on a HSV colour space map. However, these solutions require the operator's colour features to be different from the colour of the environment.

Still using two vision systems, Zhichao *et al* (2007) presented a novel approach to identify the operator without the need for the requirement related to the colour conditions of operator and environment. In this study, the researchers detected the operator by using a combination of the Lucas-Kanade method, Random Sample Consensus (RANSAC) algorithm and Viola-Jones approach to detect an upper part of the operator.

In another study, researchers from Chitose Institute of Science and Technology proposed an effective object detection algorithm based on features of the lower part of the operator (Motokucho & Oda 2014b). The operator's features were determined by employing the Optical Flow Vectors (OFV) and Disparity Vectors (DV) of stereo cameras. The OFV and DV were calculated by a block matching algorithm from right/left images and two consecutive ones respectively.

2.3.1.2 Laser range finder

An alternative approach employed for developing operator detection algorithms is the laser range finder (LRF). In the work of Kmiotek *et al* (2008), the researchers presented a target detection algorithm called the bounding box method. In the first stage, the laser data was separated into sets of segments by a segmentation procedure. The obtained segments represented the target and non-target objects. For each segment, the researchers constructed a bounding box and extracted its diagonal feature. For any given segment if the value of the diagonal feature of the bounding box was smaller than a threshold, the segment would be regarded as the target; otherwise it would be labelled as a non-target object.

Also utilising the laser scanner to detect the target, Chen Tun *et al* (2011) proposed a target detection algorithm which is called the modified Inscribe Angle Variance. This is a modification of the IAV method introduced by Xavier *et al* (2005). In their study, the researchers used the geometrical properties of the target. In the first stage, a curve shape was constructed to cover the laser raw data representing a testing object. Then the mean and standard deviation of the angle feature of the obtained curve was calculated. Finally, the testing object was the target if the mean and standard deviation value satisfied the threshold conditions. The effectiveness of the modified IAV algorithm was evaluated by experiments in relation to human detection of mobile robots in an indoor environment.

Unlike Kmiotek *et al* and Chen Tun *et al*, Lee *et al* (2006) proposed a target detection method via the motion feature of the target (human). Firstly, a data clustering procedure was implemented to obtain the clusters which are related to human legs. Based on the geometric characteristic of human legs, the researchers created new human objects from the remaining clusters. Then, an extended Kalman filter was utilised to extract the motion features of the human objects. The walking models corresponding to the human objects were derived from their motion features. It can be seen that Lee *et al* 's method is effective in tracking multiple moving targets.

Another target detection method was proposed by Mozos *et al* (2010). In their study, the researchers utilised multiple layers of laser range finders to acquire data from various parts of the target such as the head, upper body or a leg. In order to effectively detect the body parts of the target, the Adaboost algorithm was utilised to train the classifiers. Afterwards, the position of the target was derived from a probabilistic combination of the outputs of the proposed classifiers. Experimental results confirmed the effectiveness of the proposed method in detecting people within indoor environments.

2.3.2 Operator-following strategy

Operator-following strategies are divided into various types such as moving toward the operator, following the operator at a fixed distance, moving side-by-side with the operator or simultaneously tracking the operator in a crowded environment and avoiding obstacles. Each following strategy is implemented by a corresponding control algorithm generating appropriate linear and angular velocities for the bed system.

The Proportional Integral Derivative is the most popular method which is utilised to control the operator-following behaviours since it is simple and effective. Normally, two controllers are required to control the linear and angular velocity variables of the mobile system. In the literature, various combining strategies have been proposed to implement the following behaviours such as P – P (Chung et al. 2012; Luo et al. 2009; Yoo & Woojin 2011; Yoonchang & Woojin 2011), PD – PD (Cai & Matsumaru 2013; Li, Song, et al. 2015; Sakai, Hiroi & Ito 2014), P – PID (Eui-Jung et al. 2014) and PID – PID (Yang et al. 2013) controllers.

Fuzzy logic is another approach which is employed to design the operator-following control strategies. In a work by Shaker *et al* (2008), the researchers aimed to develop a conventional fuzzy controller for providing decisions to achieve the smoothness and safety objectives of the operator-following problem. Experimental results showed that the proposed method met the goal of moving smoothly towards the operator without exhibiting any noticeable oscillations. However, since the fuzzy system is designed with only one output of acceleration/deceleration, the vehicle system is restricted to moving forward and backward.

In an effort to enable the vehicle system to move with various speeds and directions, a fuzzy model introduced by a research group from National Chiao Tung University was designed with two outputs containing linear velocity and angular velocity variables (Bing-Fei & Cheng-Lung 2013). In their study, the researchers aimed to design the fuzzy controller so that their vehicle system always followed the operator at a desirable distance and angle. To achieve this goal, distance error and angle error variables were utilised as fuzzy model inputs. Based on a fuzzy rule table, the proposed controller generated the control signs of linear velocity and angular velocity for the vehicle system.

2.3.3 Advanced low-level control

A low-level control algorithm plays a role in guaranteeing the stability of the overall system under the influence of uncertain factors. In terms of the current operator-following design, a virtual spring dynamic model-based control algorithm is a popular approach for dealing with the dynamic problems of a wheeled mobile robot and electric powered wheelchair (Do & Lin 2015; Morioka, Joo-Ho & Hashimoto 2002; Motokucho & Oda 2014a). Technically, this dynamic model is described by several system parameters such as the mass of the system, the moment of inertia of the vehicle system, viscous coefficients of translation and rotation. However, this control approach may be impractical for the smart hospital bed. Due to parametric variations caused by the bed mechanism and the operating environment, it is difficult to obtain the exact dynamic model.

Expanding the research area related to advanced control strategies for dynamic problems, several methods including nonlinear control, sliding mode control and intelligent control have been proposed for wheeled mobile robots and electric powered wheelchairs.

2.3.3.1 Nonlinear control

Lyapunov-based control approach

One of the popular nonlinear control methods is the Lyapunov-based control algorithm. In the research of Yulin *et al* (1998) the researchers utilised the Lyapunov theory to design a robust control algorithm for a differentially-steered wheeled mobile robot. A procedure was implemented to construct the full mathematical dynamic model consisting of the external loads and tyre model; then the simplified dynamic model was derived from several assumptions of wheel-ground contact. Based on model analysis with uncertainties, an exponentially stable tracking control algorithm was developed by applying Lyapunov theory. In spite of simulations confirming the robustness of the proposed method against uncertainties, this method is not suitable for the smart bed system. Due to various factors impacting the bed, this simplified dynamic model is not adequate for the hospital bed model.

Also utilising the Lyapunov design technique, Aguiar *et al* (2000) proposed a control algorithm for dealing with the dynamics problem of a wheeled mobile robot. A simple dynamic model is adopted for developing the nonlinear controller. In an effort to analyse the convergence of the proposed control approach, the researchers used the Lyapunov's stability theory combined with the Lasalle's invariance principle. Based on this analysis, the nonlinear adaptive control rule generating a global convergence of trajectories of the closed-loop system within the existence of parametric modelling uncertainty was obtained. Nevertheless, there is a lack of experimental results to verify the efficiency of this method in real-time implementation.

With the purpose of solving the parametric uncertainty problem of nonholonomic wheeled mobile robots, a research group of the University of Tehran developed a dynamic tracking control method based on Lyapunov functions and model reference adaptive control (Gholipour & Yazdanpanah 2003). In the first step, Euler Lagrange formulation was utilised for describing the dynamic model of the nonholonomic mobile systems. In the next step, nonlinear control rules were designed to deal with the kinematic stabilisation. Finally, a model reference adaptive controller using the output variables of the nonlinear controller as its inputs was developed to stabilise the dynamics part with uncertain parameters. During the control design, Gholipour *et al* applied the Lyapunov theory for guaranteeing the stability of state variables and the overall system. However, in order to confirm the quality and efficiency of the proposed control approach, only simulations were conducted.

A similar endeavour to apply the Lyapunov-based control method on a wheeled autonomous robot is articulated by Wang *et al* (2009). In their study, the Lagrange formulation was also employed to construct the dynamic model of the autonomous mobile system. In the design stage, two motion controllers were derived from the use of a Lyapunov's direct method and a computed torque method.

Choosing appropriate output variables leads the mobile system with the obtained control laws to converge to a desired trajectory with asymptotic stability. The feasibility and effectiveness of the method have been demonstrated by experimental results.

Considering a wheelchair with its displaced mass centre, the Lyapunov's method is also utilised in an effort to design control for this kind of mobile system. To do this, Andaluz *et al* (2015) firstly proposed the kinematic and dynamic models of the electric-powered wheelchair whose centre of gravity is not placed at the wheel's axis centre of the system. Then, two corresponding controllers were developed to deal with the kinematic and dynamic stabilisation of the robot. For each cascade subsystem, the appropriate Lyapunov function was employed to analyse the stability and robustness of the proposed control laws. Finally, real-time experiments were executed to show the proposed controller performance.

Back-stepping control strategy

Another advanced method which can be applied to the smart hospital bed is back-stepping. This method is utilised in a study of Fukao *et al* (2000) to develop the adaptive tracking control strategy for a nonholonomic mobile robot with unknown parameters. The researchers proposed that an adaptive dynamic controller could be developed by the back-stepping strategy if there was the existence of an adaptive controller for the kinematic model with unknown parameters. In order to verify this idea, Fukao *et al* firstly indicated an adaptive tracking controller for the kinematic model based on the work of Kanayama *et al* (Kanayama et al. 1990). Then the theorem was applied to the dynamic model for developing the torque controller. The limitation of this design is that there are a lot of assumptions. In addition, the effectiveness of the control algorithm is verified by only simulation results.

Like Fukao *et al,* Wenjie *et al* (2000) attempted to apply the back-stepping technique on wheeled mobile robots. The researchers suggested a three-stage control design to cope with the robot dynamics with unknown inertia parameters. In the first stage, a suitable tracking error was introduced to support the development of the control law. Subsequently, the researchers ignored the system dynamic part and regarded the control signal as a virtual force. In the final stage, the kinematic based control law was derived from a well-known back-stepping strategy. The efficiency of the obtained controller was confirmed by simulation results.

Unlike previous back-stepping controllers for wheeled mobile systems, Chwa (2010) proposed a novel back-stepping control technique based on feedback linearisation. With the purpose of developing a simpler and modular control structure, the author described the control structure in the form of a cascaded kinematic and dynamic linearisation. Following this, the kinematic part was used to design the pseudo commands corresponding to the forward linear velocity and the heading direction angle variables while the dynamic controller took responsibility for generating actual torque forces to guarantee the actual outputs of the system to track their reference inputs. Despite various simulations which were implemented to validate the proposed control scheme, computation of this method proved to be too heavy and complicated.

The unknown mass and inertia parameter problem of the robot dynamic model can be also be handled by the back-stepping based nonlinear control strategy. In the work of Aneesh (2012), the author constructed a dynamic robot model consisting of the kinematic model and the torque model derived from Lagrange's formulation. Afterward, the Kanayama transformation was adopted to convert error dynamics into mobile coordinates. As a result, the tracking problem changed to stabilisation. In the control design stage, Aneesh developed a nonlinear back-stepping controller for kinematics and a model reference adaptive controller for the dynamics of the system. To achieve the kinematic stabilisation, both of the obtained controllers were shown to meet the Lyapunov stability criterion. Through numerical simulations, the robust performance of the proposed control approach was verified.

The back-stepping method was also presented in the work of the research group from China University of Petroleum (Huichao, Shurong & Haiyang 2012). Unlike the control approaches mentioned above, the robot model in this study was divided into three parts consisting of a conventional kinematic model, an Euler-Lagrange based dynamic model and a DC motor model. Based on the obtained mobile robot model, three tracking controllers corresponding to the three partial models were developed via the back-stepping method and the Lyapunov function. To validate effectiveness of the proposed tracking controller, the researchers performed several simulations.

Another study attempting to apply the back-stepping method for the wheeled mobile robot with kinematic and dynamic uncertainties was presented by Li *et al* (2014). First of all, the robot dynamic model was calculated via the Lagrange method with several assumptions. Due to the difference between parametric and non-parametric uncertainties, two controllers were subsequently designed to enhance the operational efficiency of the system. In particular, an adaptive kinematic controller was utilised to evaluate the parametric uncertainties whereas the purpose of an adaptive robust back-stepping dynamic controller was to compensate for the non-parametric uncertainties of the robot dynamic system. Finally, the researchers applied Lyapunov's theory for analysing the stability of the robot via the uniformly ultimately bounded closed-loop control system.

By combining back-stepping techniques with low-pass filters, Chang *et al* (2015) introduced a novel tracking control algorithm for mobile robots . This method not only dealt with the problem of the partial loss of actuator effectiveness for the robot systems but also overcame the problem of "explosions of complexities" caused by repeated differentiations of virtual control rules of the conventional back-stepping techniques. Through simulation results, the proposed control method was verified to effectively preserve the desired control objectives under uncertainties and actuator faults.

From the control point of view, several arguments indicate that the nonlinear control strategies such the Lyapunov based control and back-stepping technique are impractical for the smart hospital bed. The first reason concerns the dynamic modelling procedure via Lagrange's function. Since there are various uncertain factors impacting on the bed system, it is difficult and inefficient to obtain the dynamic hospital bed model from the Euler-Lagrange's formulation. Another reason is that these methods are confined to a number of assumptions and their computation is too heavy and complicated. The final reason is that there is a lack of real-time experiments to verify the effectiveness of these control strategies.

2.3.3.2 Sliding mode control

Owning advantages of fast response and robustness for dealing with system uncertainties and external disturbances, Sliding Mode Control (SMC) is known as an efficient method to design dynamic control strategies.

An early work endeavouring to utilise SMC on a wheeled mobile system was introduced by Jung-Min and Jong-Hwan (1999). In this study, the researchers proposed a novel SMC law for the trajectory tracking problem based on the utilisation of an Euler-Lagrange's dynamic equation. A computed torque method was used to feedback-linearise the system dynamics. Despite the simulation and experimental results confirming the robustness of the proposed control scheme and the efficiency of the accurate tracking ability, the researchers only coped with the problem of external disturbances. Parameter variations were not considered in this study.

In an effort to deal with the bounded uncertainty problem in the robot dynamics, a quasi-SMC technique formed the basis for a discrete-time SMC law presented by Corradini and Orlando (2002), this assured the implementation of the controller. Utilising this design method, the robust asymptotic vanishing of tracking errors was theoretically verified. Performing real-time experiments on the LABMATE vehicle validated the effectiveness of the proposed control law.

Forming a conservative upper bound for nonholonomic system dynamics, Zuozhi *et al* (2005) proposed a SMC algorithm to ensure that the system could meet any desirable trajectory in the presence of structured and unstructured uncertainties. Based on the Lagrange's dynamic equation and Lyapunov's theory, the stability analysis illustrated that trajectories not only have the ability of reaching the sliding surface in finite time but are confined to sliding surfaces in the subsequent time. Despite the simulations successfully demonstrating the proposed algorithm, the chattering phenomena still happened near sliding surfaces.

In order to improve the sliding mode technique for wheeled mobile robot control, a novel approach was presented by Wang *et al*_(2010). With the purpose of solving the speed jump problem existing in the traditional sliding mode method, the researchers utilised a back-stepping technique based on a neural dynamics model. Firstly, this type of dynamics model was derived from the biological membrane model of a nervous system proposed by Hodgkin and Husley (1952). Then, an improved back-stepping based sliding mode controller was developed based on the obtained dynamics model. However, only the simulation results verified the effectiveness of the proposed approach.

Another work attempting to utilise a back-stepping sliding mode control method on a wheeled mobile robot was introduced by a research group from MES College of Engineering, Kerala (Vishnu Prasad, Pottakulath & Ajmal 2014). In the study, the dynamic model was firstly derived from the Euler-Lagrange approach. Afterwards, the back-stepping sliding mode controller was developed to search for an acceptable control law so that the tracking error would asymptotically move to zero as the time approached infinity with an initial error. During the control design, the Lyapunov functions were appropriately chosen to guarantee the stability of the overall system. Various simulations were executed to illustrate the robustness and performance of the proposed control strategy.

2.3.3.3 Intelligent control

Neural networks have been widely utilised to develop dynamic control algorithms for the wheeled mobile system since they have many advantages consisting of experience based learning, the capability to generalise or the capability to map nonlinear functions. An early work attempting to utilise this intelligent control technique on a mobile robot system was presented by Fierro and Lewis (1998). In their study, the researchers employed a multilayer feed-forward neural network to design a neural control strategy. The unknown dynamics of the nonholonomic mobile robot was approximated by utilising a multilayer feed-forward neural network online. Then, a back-stepping based control law was derived from the use of the obtained dynamic model. In the control design, Lyapunov's theory was adopted to prove the stability of the mobile system. Despite simulation results verifying the performance of the proposed control strategy the neural training algorithm and control scheme are complicated.

To make the control structure simpler and to enhance computational efficiency, Tiemin *et al* (2001) introduced a new controller based on a single layer neural network. Lagrange formulation was firstly utilised to introduce the robot dynamics. Then, the training algorithm was obtained from the Lyapunov's theory therefore the analysis of the system stability was no longer necessary. Although the researchers pointed out that this control approach can be implemented in real-time, there was a lack of experimental results to validate the effectiveness of the proposed control structure.

Boquete *et al* (1999) presented a novel neural control algorithm for a motorised wheelchair. In their study, the proposed control structure based on an inverse control strategy consists of a PID controller playing the role of a dynamic controller and a neural controller taking the responsibility of kinematic controller. A recurrent RBF network model was designed for the neural controller. To guarantee the convergence of the neural control law, the Lyapunov' stability conditions were studied. During the network training, a gradient descent algorithm was applied to update weights. In order to deal with the problem of Jacobian calculation, the researchers approximated the wheelchair model with the use of a recursive equation. Therefore, the Jacabian matrix could be calculated from the obtained wheelchair model.

Extending the previous approach, Boquete *et al* proposed two solutions for dealing with the problem of the Jacobian calculation. In the first approach, a Kalman filter was employed for identifying online the wheelchair dynamics (Boquete et al. 2002). The wheelchair model was approximated through the study of step input signals and their corresponding output responses in the state space. By ignoring the noise system, the Jacobian calculation could be obtained from the model provided by the Kalman filter.

In the alternative approach, Boquete *et al* designed an inverse control strategy with the use of two neural networks (Boquete et al. 2005). In their control structure, the first NN was serially connected to the wheelchair and acted as a neural controller whereas the second NN was connected in a parallel sense to the wheelchair and acted as a neural identifier. During the online mapping, the wheelchair model was identified by the second NN. As a result, the problem of Jacobian calculation for the neural controller could be solved. Unlike the first approach, the assumption of the wheelchair dynamics was no longer necessary. Performing real-time experiments verified the effectiveness of both improved approaches.

Another advanced control strategy based on neural network control was introduced by Nguyen *et al* (2007). Regarding the electric power wheelchair as a multivariable system with uncertainties, the researchers firstly constructed the approximate dynamics model via a parameter verification procedure and then proposed a novel neural multivariable control strategy with two stages. In stage 1, a decoupling technique with a triangular diagonal dominance property was utilised to simplify the multivariable control problem into a series of single variable control problems. Then in stage 2, a neural network controller was developed for each independent scalar control system. Simulations and experiments served to demonstrate the successful performance of the proposed approach.

With the purpose of utilising neural networks for developing the dynamic control of nonholonomic mobile robots, Bugeja *et al* (2009) introduced two novel dual adaptive neural control schemes. In this study, the nonlinear robot dynamic functions were approximated by ANNs consisting of a GaRBF and a MLP network. In each neural network structure, the researchers randomly evaluated the unknown network parameters in real time and neglected preliminary offline neural network training. As opposed to other adaptive control strategies, the proposed control laws in this research did not depend on the heuristic certainty equivalence property but accounted for the uncertainty in the estimates. Various simulations were executed to illustrate the effectiveness of the two proposed control schemes.

Utilising the reinforcement learning algorithm, Luy (2012) proposed a new method to design a tracking control scheme for a wheeled mobile robot. In his study, Luy firstly altered Lagrange's formulation to derive a strict feedback tracking system in the state space. Then, the researcher utilised the obtained feedback tracking system to develop a neural network tracking control scheme with an actor critic structure. With the use of a reinforcement learning technique and Lyapunov's theory, a new turning law was designed so that the proposed neural network was able to learn online a tracking-HJB equation and assure the stability of the closed-loop system in real-time. Finally, the efficiency of the neural controller was validated by the results of the simulation for the mobile robot system.

Li *et al* (2015) introduced a neural adaptive control approach for wheeled mobile robots. This is a hybrid control strategy which is a combination of a RBF neural network, PD control and a robust term. Due to a property of learning fast, the researchers used the RBF network to approximate uncertain mobile robot dynamics while the robust term was added to decrease the approximation error caused by external disturbances. By updating the weight of the RBF network, the control strategy not only assured the desired performance of the velocity tracking but also minimised the posture tracking error for the wheeled mobile robot. However, only simulation results confirmed the effectiveness of the Li's method.

Also using the advantage of the RBF neural network, Raeisi *et al* (2015) presented another tracking control algorithm for wheeled mobile robots within the presence of uncertainties. In their study, the RBF neural network was adopted for compensating for the nonlinearities of the system model. The design of an adaptive output feedback tracking controller with only position measurement was derived from a work by Arteaga *et al* (2004). Unlike the previous control strategies, this control approach can be effectively employed for various wheeled mobile robot systems since its output equations can be appropriately chosen. Nevertheless, there is a lack of real-time implementations to demonstrate the effects of this control method.

Due to its universal function approximation ability, the fuzzy logic system is also used for designing the advanced control algorithms of the wheeled mobile systems. In the work of Das *et al* (2006), the fuzzy logic method was utilised to estimate nonlinear wheelchair functions including unknown dynamics. Subsequently, an adaptive control law was derived from the estimated dynamic function and Lyapunov's stability theory. During the control design, only the position information of the wheelchair was acquired to decrease the cost. Despite simulation and experimental results verifying the efficiency of the control approach, the expert knowledge required for forming fuzzy rules is an important requirement of the design.

2.4 DISCUSSION

Considering the improvements of the smart hospital bed, the operator-following function is a safe, efficient and innovative solution for dealing with the problem of transporting a patient. It not only minimises bed switching but also reduces manual handling tasks and nursing injuries. Without human control in its steering system, unexpected risks caused by humans in the patient transport can be substantially decreased. Moreover this solution enables hospital staff to completely focus on patient-care activities when the bed follows an autonomous navigation robot. Research related to the operator-following function of the smart hospital bed has been absent in the literature.

To enable the operator-following ability, an operator detection algorithm must be developed for the smart bed system. Popular approaches consisting of the vision system and the laser range finder have been employed for designing operator detection algorithms. Technically, the vision system usually suffers from huge computational complexity and its performance can be easily affected by the settings of the camera and uncertainties (Chen Tun et al. 2011). Moreover, the vision based algorithms are fragile against the lighting condition (Hiroi, Matsunaka & Ito 2012). On the other hand, the laser range finer not only deals with the lighting problem of the vision system but also has many other advantages such as high accuracy of measurement, low computational cost, high resolution and reliability (Hiroi, Matsunaka & Ito 2012; Yoo & Woojin 2011). As a result, the laser scanner is utilised by many researchers to implement the following task.

In the section 2.3.1, various LRF based operator detection algorithms were reported including the bounding box, IAV, Kalman filter and Adaboost methods. Despite these methods being successfully implemented on the mobile system for detecting moving objects, they still have several drawbacks. Chung *et al* (2012) show that the disadvantage of the bounding box and IAV algorithms is their low accuracy. The Kalman filter approach has the drawback of the computational cost (Hiroi, Matsunaka & Ito 2012). Meanwhile, the Adaboost method is sensitive to noisy data and outliers.

Operator-following strategy is also involved when developing the operator-following function for the smart hospital bed. The operator-following strategy is divided into various types consisting of moving toward the operator, following the operator with a fixed distance, moving side-by-side with the operator or simultaneously tracking the operator in a crowded environment and avoiding obstacles. Each following strategy is implemented by a corresponding control algorithm which generates appropriate linear and angular velocities for the bed system. Generally, most operator-following behaviours can be successfully achieved by utilising the PID method since it is simple and effective.

Due to variation of system uncertainties, an advanced low-level control algorithm is an indispensable part of the operator-following control algorithm for the smart hospital bed. Since no research related to low-level control has been done on the smart hospital bed, this chapter has reviewed various advanced dynamic control strategies with respect to wheeled mobile robot and wheelchair systems. The popular advanced low-level control strategies consist of nonlinear control methods, sliding mode control method and intelligent control strategies. In general, most of these approaches may not be suitable for the smart hospital bed.

The first reason lies in the construction of the dynamic model. Most researchers utilise two common ways for calculating the system dynamics. One of these is the utilisation of the Euler-Lagrange formulation. Technically, the parameters of this dynamic formulation are difficult to obtain to an exact degree. Due to this, control solutions based on this dynamic equation are only suitable for simulation purposes. The alternative method is the employment of model approximation and identification techniques to reach the dynamics. Nevertheless, these techniques are only proper for particular control structures. Moreover, these techniques require complicated computational algorithms for their implementation, therefore confining them to real-time application.

The second reason is that most advanced control approaches do not treat the wheeled mobile system as a multivariable system. The coupling effect of the multivariable control problem is neglected in most reported studies. Accordingly, these control approaches may not be optimal when they are applied to the smart hospital bed.

One of few works regarding a wheeled mobile system as a multivariable system was presented by Nguyen *et al* (2007). In their study, the researchers utilised a TDD decoupling technique to reduce coupling effects and simplify a multivariable system into a series of independent scalar systems. Moreover, Nguyen *et al* used an effective experimental procedure to approximate the dynamic model of the wheeled mobile system. However, the obtained dynamic model was not optimal since the researchers restricted the range of voltage inputs when implementing the parameter verification procedure.

From the discussion above, this thesis will focus on developing advanced assistive control strategies to support the smart hospital bed during patient transportation. Firstly, two operator detection algorithms which have not been published in relevant literature are investigated and developed for the smart hospital bed. Then, two multivariable low-level control strategies are proposed and designed for the smart hospital bed to deal with the coupling effect problem and to drive actual velocities to track desired velocities. Both methods treat the smart hospital bed as a multivariable system with uncertainties. Finally, an advanced operator-following control strategy is implemented to enhance the bed performance during its operation. This is a novel control strategy with respect to the operator-following solution.

CHAPTER 3 . OPERATOR DETECTION METHODS FOR THE TRACKING FUNCTION OF SMART HOSPITAL BEDS

3.1 INTRODUCTION

The operator-following function is considered to be an effective and innovative solution for smart hospital beds in terms of patient transportation. This solution allows the smart hospital bed to travel to the destination by following a clinical staff member or an autonomous navigation robot. As a result, unpredictable accidents caused by the hospital employee can be substantially reduced.

To achieve this goal, an operator detection method is required for the smart hospital bed since it allows the bed to classify the operator in the space including the target and various non-target objects. In the literature, many studies show that an operator can be successfully detected by a laser scanner due to its advantage of high accuracy and reliability. Accordingly, two operator detection algorithms based on a laser ranger finder are investigated and developed in this chapter.

The first algorithm, a Gaussian Distribution Method (GDM), is one of the efficient classification techniques dealing with the one-class problem: the target class and non-target class. In this method, a training data set of the target class is modelled as a Gaussian distribution. Then the classifier is based on a threshold condition of a Mahalabonis distance between a testing point and the training data set. If the Mahalabonis distance is bigger than the threshold value, the testing object is classified as the non-target. Otherwise, the testing object is the target.

The second algorithm, a Neural Network (NN) classification method, is a classification technique based on an Artificial Neural Network (ANN) theory. The ANN is known as a computational model constructed by a network of highly interconnecting processing elements operating in parallel (Kumar & Kumar 2013). It has many advantages such as the capability of adaptive learning, self-organisation and real time operation. Through a training process, a neural network has the ability of classifying any data with arbitrary accuracy.

The chapter is organised as follows. Firstly, a system structure of the smart hospital bed utilised throughout this study is briefly introduced in section 3.2. In the next section, two proposed operator detection methods consisting of the GDM and the NN classification method are described. Section 3.4 provides an operator-following strategy for the smart hospital bed. Then, section 3.5 presents several experimental results to measure the effectiveness of the proposed methods. Finally, discussion can be found in the last part of this chapter.

3.2 SMART HOSPITAL BED STRUCTURE

To validate the proposed algorithms in this doctoral research, a smart hospital bed developed by the UTS Centre for Health and Technologies through a collaboration with the Royal Institute of Technology, Stockholm is employed. Technically, the bed consists of a bed frame, a sensor system, a central control system, a power supply system, motors and actuators, wheels and an air mattress. The appearance of the bed is shown in Figure 3.1.

3.2.1 Bed frame

The bed frame of the smart hospital bed is divided into three main parts including an upper frame, support lifts and a lower frame. The specifications of each component are described as follows.

- Upper frame: Similar to many other hospital beds, the bed surface includes 4 sections which are the head, bottom, thigh and calf parts. Six linear actuators mounted under the bed surface are responsible for adjusting the upper frame to various positions such as lying face up in a horizontal position, a seated position, leg elevation position, back elevation position, knee brake position, back and leg elevation position and a back elevation with knee brake position.
- Support lifts: In order to support a patient to easily get in or out of bed, three lifting columns are integrated into the smart hospital bed to raise or lower the bed

height. Each of them has an electric actuator with a chain drive inside for performing its enlarging and abridging movements. Combining the motion of these lifts also places the patient in several special positions, for instance: the Trendelenburg position, the reverse-Trendelenburg position, the left or right tilt position.

- Lower frame: This component is designed as a rectangular base to protect the central control system, wheel system, direct current (DC) motors and the power supply system mounted inside.

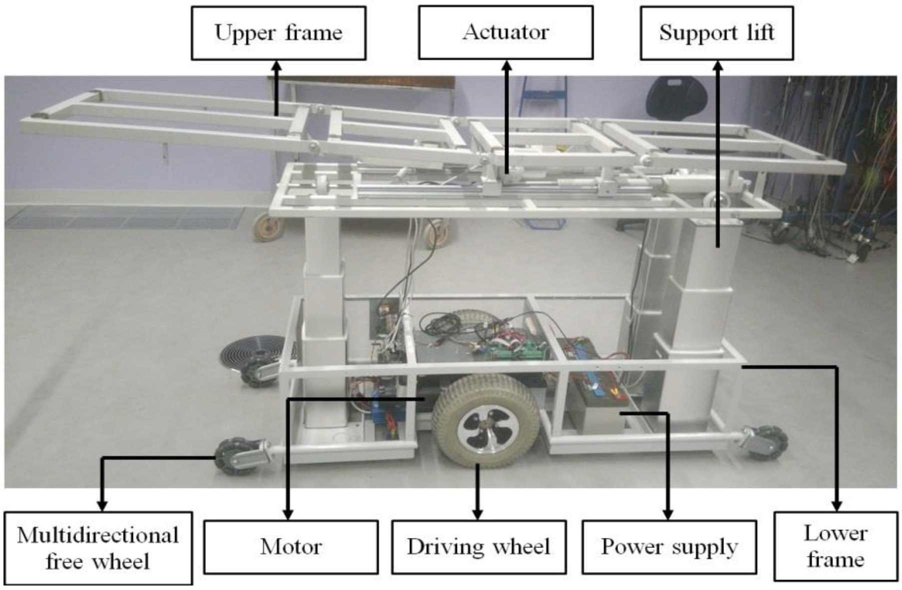


Figure 3.1: The smart hospital bed system

3.2.2 Sensor system

The smart bed has a measuring sensor system including two incremental optical encoders, a digital type dual-axis inclinometer LCA326T and a laser range finder URG-04LX as shown in Figure 3.2.

- *Incremental optical encoders:* the incremental optical encoders are attached to both driving wheels. Each of these encoders generates two square waves in incremental quadrature mode which provide the instantaneous velocity information of the driving wheels (Chao et al. 2015). Through a converting procedure, these velocities are transformed into the linear velocity and angular velocity of the smart hospital bed.

- *Inclinometer:* being mounted under the bed surface, the digital type dual-axis inclinometer LCA326T with low-cost plays a role in terms of measuring an angle of inclination of the bed surface with two dimensions. Based on the measuring information of this sensor, the bed surface is precisely adjusted to the desired position. Technically, this sensor has a maximum measuring range of $[-90^o,90^o]$, a 0.01^o resolution, and it consumes low power.
- *Laser range finder:* the laser URG-04LX is utilised as the main sensor to provide the distance information of obstacles for the smart hospital bed due to its advantages such as high accuracy and resolution, and low power consumption. The laser is located in the middle front section of the hospital bed. Technically, this sensor can operate up to 10Hz (scanning rate), with a 240^o field-of-view, a 0.36^o resolution, an accuracy of 1% of the measurement, and a maximum radius of 4m. These specifications of the sensor are very appropriate for the smart bed system since the laser sensor is able to give all the information required about the obstacles in front of the bed. This information supports the bed when travelling within the hospital, for instance: following a target, obstacle avoidance and local path planning.

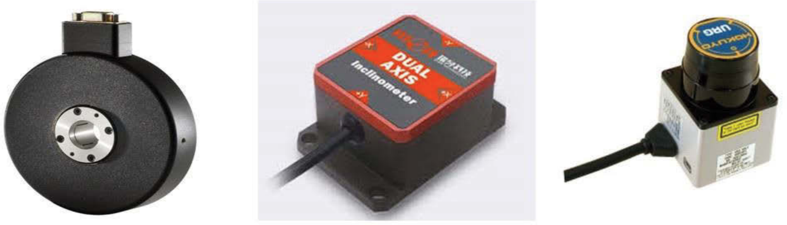


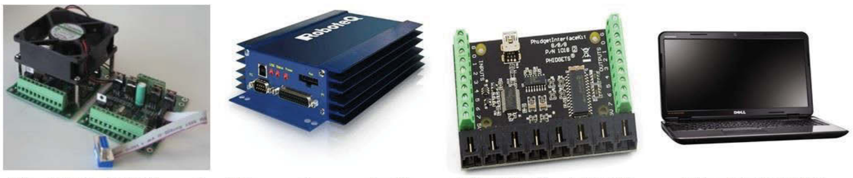
Figure 3.2: The sensor system: the encoder (Encoder 2016), the inclinometer (Rion 2011) and the LRF (Hokuyo 2009)

3.2.3 Control devices

The control devices consist of a KTA-198 board, a motor controller, a Phidget 8/8/8 board and a central processing unit as displayed in Figure 3.3. Details of each part are described below:

- *KTA-198 board*: this device is used to control actuators attached to the bed frame. As each controller board can control two actuators, 4 boards are required to control all actuators attached to the smart hospital bed. The KTA-198 controllers acquire the position data of actuators and transmit them to the central processing unit via the RS232 connection. Depending on the desirable bed position, the KTA-198 boards receive commands from the central processing unit to adjust the position of the actuators.

- *Motor controller*: RoboteQ HDC2450 is utilised as a wheel motor controller for the bed system. The controller is responsible for receiving feedback data from encoders and transferring this information to the central processing unit via the serial interface, USB interface or Wifi connection. Besides this, it also receives motor commands from the central processing unit and sends control signals to the DC motors.
- *Phidget 8/8/8*: in order to sense the environment, the smart bed is equipped with a Phidget sensor interface board to obtain parameters from various sensors such as temperature, humidity, pressure. This is a USB based controller with 8 analog inputs, 8 digital inputs and 8 digital outputs. Besides this, it communicates to the central processing unit via a USB port.
- *Central processing unit*: there are various options to design the central processor for instance: a micro controller board (Keil MCB2300), a mini-computer (MAC mini) or a laptop. In this study, the laptop with model Dell Inspiron 5110 and LabWindows 2010 software with the Real-time module are utilised to implement all proposed algorithms.



The KTA 198 board   The motor controller   The Phidget 8/8/8 board   The Dell 5110 laptop

Figure 3.3: The central control system: the KTA 198 board (Oceancontrols 2009), the motor controller (RoboteQ 2015), the Phidget 8/8/8 board (Phidgets 2012) and the laptop

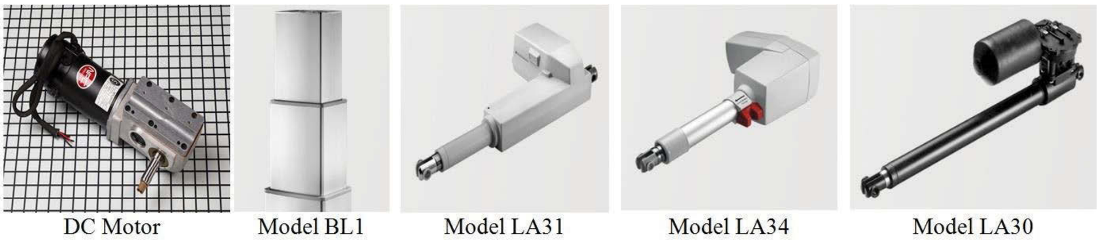
3.2.4 Power supply system

The power supply system of the bed includes two 12 volt batteries in series and a separate 12 volt battery. The two 12 volt batteries in series supply the power to

DC motors and linear actuators. On the other hand, the separate 12 volt battery provides the power for the other electronics including the central control system, sensor system and air mattress. A power switch system and an emergency stop button are located on one side of the bed.

3.2.5 Motors and actuators

The smart hospital bed is equipped with two DC motors distributed by NPC Robotics and nine linear actuators developed by Linak. Each DC motor is attached to each driving wheel and powered by 24 volt and 20 amp duty. Among the actuators, three of them (models BL1) are mounted inside the support lifts while the others (three models LA31, one model LA34 and two models LA30) are attached under the bed surface. Movements of the linear actuators are combined together to adjust the smart hospital bed to various positions. The two 12 volt batteries in series are also utilised to provide power to the linear actuators. Figure 3.4 illustrates the motors and actuators employed in the bed system.



DC Motor Model BL1 Model LA31 Model LA34 Model LA30

Figure 3.4: The DC motor (NPC 2011) and actuators (Linak 2016)

3.2.6 Wheels

The bed system has two driving wheels, each of which has a diameter of 320 mm placed at the middle side of the lower frame and this is attached to its own DC motor. With the purpose of preventing the smart hospital bed from falling over, multidirectional free wheels are added. Each of the multidirectional wheels consists of 16 free-turning rollers which are positioned around the wheel periphery. With this type of passive wheel, the bed has the capacity to move in any direction while the driving wheels are rotating. Technically, the smart bed will go in a straight line if both the driving wheels are driven in the same direction and speed. In the case of both steering wheels moving with the same velocity in different directions, the bed will rotate about the central point of the axis. Otherwise, depending on the speed of the rotation and its direction, the bed can turn left or right with various linear and angular velocities. The wheel system employed in the bed is shown in Figure 3.5.

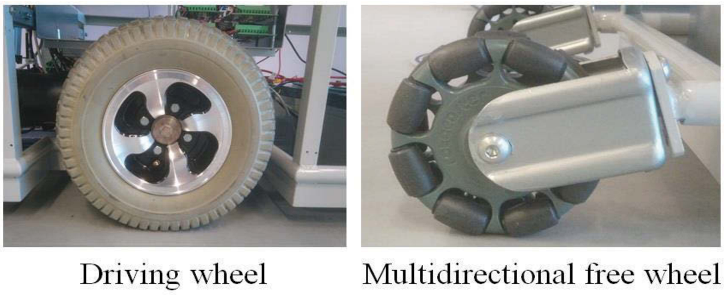


Figure 3.5: The wheel system

3.2.7 Air mattress

The hospital bed is equipped with an Alternating Pressure Therapy/Low Air Loss Mattress system developed by Lumex. The mattress system is not only efficient for preventing and treating stages 1 – 4 of pressure ulcers but also convenient for long-term bedridden patients. In order to provide the best therapy and comfort for each type of patient, the hospital bed staff can adjust the pressure and alternating cycle time of the mattress. Providing 6 options of mattress pressure and an optional static mode enables the mattress to meet all the requirements of the patient. Figure 3.6 presents the air mattress integrated into the bed system.

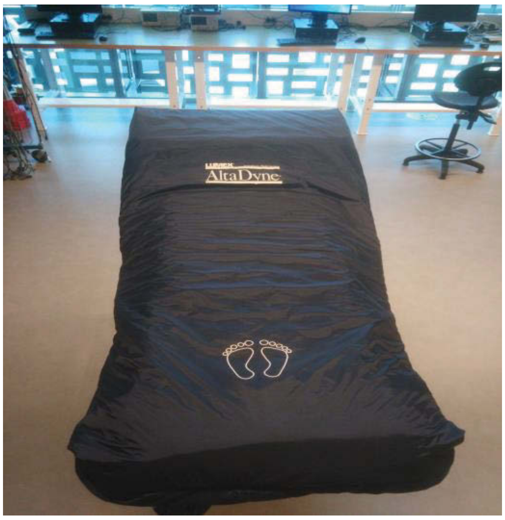


Figure 3.6: The air mattress

3.3 OPERATOR DETECTION METHODS

3.3.1 Feature extraction

This stage aims to determine features of the objects including the target and non-target objects in front of the smart hospital bed. In order to achieve this goal, the environmental data obtained from the laser range finder are represented as a set of points before being divided into different clusters by a threshold value. Finally, three features are derived from each remaining cluster.

3.3.1.1 Data acquisition

As discussed in Section 3.2, the laser range finder is mounted at the front of the hospital bed. It provides the environmental information around the hospital bed. These scanning data are represented as a set of points in the polar coordinate system (d_i^*, β_i^*) whose origin is the location of the sensor. Setting the laser as a reference point, points of the range scan are converted into the Cartesian coordinate system as follows:

$$\begin{cases} x_{Pi} = d_i^* sin\beta_i^* \\ y_{Pi} = d_i^* cos\beta_i^* \end{cases} \tag{3.1}$$

where (x_{Pi}, y_{Pi}) is the position of the points of the laser scan in the Cartesian coordinate system.

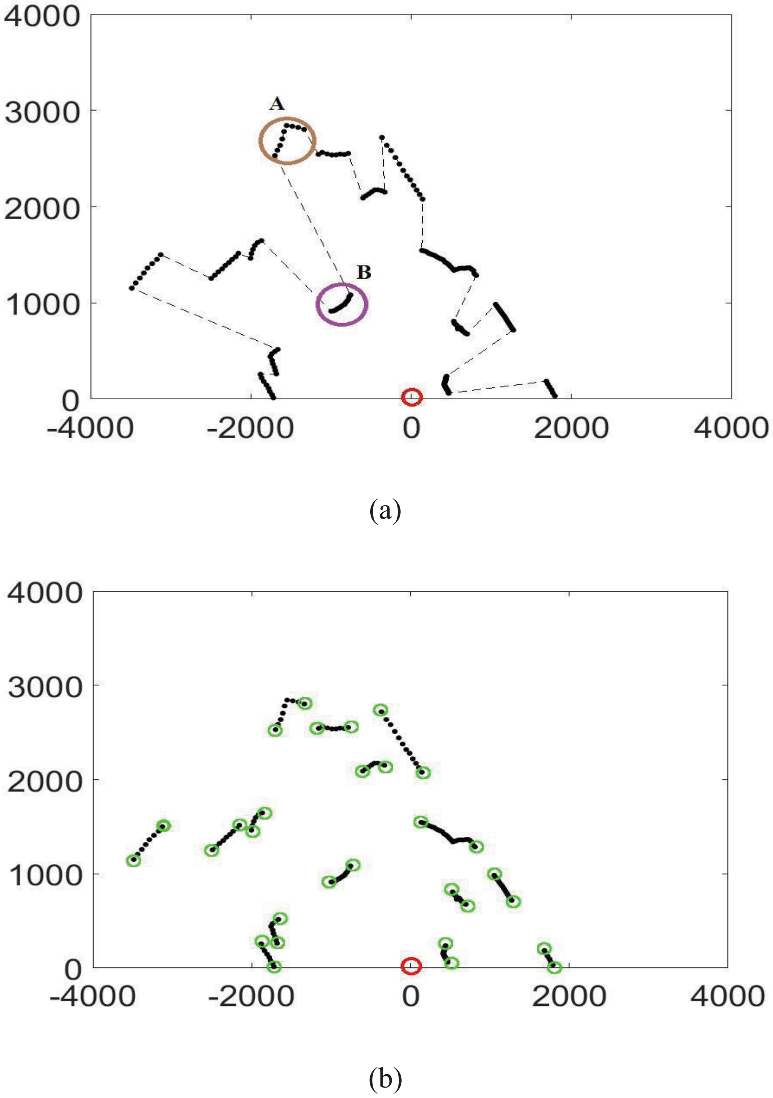
3.3.1.2 Clustering

The purpose of the clustering phase is to separate the set of points into a set of point clusters that are related to different objects detected by the laser range finder. In order to do this, a threshold condition is employed and defined as follow:

$$D(P_i, P_{i+1}) < D_{threshold} \tag{3.2}$$

where $\mathrm{D}(P_i, P_{i+1})$ is the distance between two consecutive scan points, $D_{threshold}$ is the threshold distance parameter. This condition means that if the distance between two consecutive scan points is smaller than a specific threshold, these points are in the same cluster otherwise each point belongs in a different cluster.

Procedures 3.3.1.1 and 3.3.1.2 are demonstrated in Figure 3.7 (a, b) with the angular resolution and scanning angle of the laser scanner set at 1.05^o and 180^o, respectively. It can be seen in Figure 3.7.a that points in a purple circle (cluster B) represent the same object since the distance between two consecutive scanning points is smaller than a threshold distance value. Paralleling the purple circle, points in a brown circle (cluster A) depict the same object. Figure 3.7.b shows the laser scanning data after clustering with an employment of $D_{threshold}$ chosen at 36mm.



(a)

(b)

Figure 3.7: Laser data processing procedure: (a) Data acquisition, (b) Clustering

3.3.1.3 Feature determination

After clustering, three features of each remaining cluster are shown in Figure 3.8 and determined below:

- G is the girth of the cluster, which is defined by the following equation:

$$G = \sum_{i=1}^{n} \overline{P_i P_{i+1}} \tag{3.3}$$

- W is the distance between the start and end point of the cluster and defined by the following equation:

$$W = \overline{P_1 P_n} \tag{3.4}$$

- H is the biggest distance between one point of the cluster to the line of the start and end point of the cluster:

$$H = \max\left(\frac{\left|a.x_{P_i} + b.y_{P_i} + c\right|}{\sqrt{a^2 + b^2}}\right) \tag{3.5}$$

where *a, b, c* are parameters of the line equation through the start and end points of the cluster.

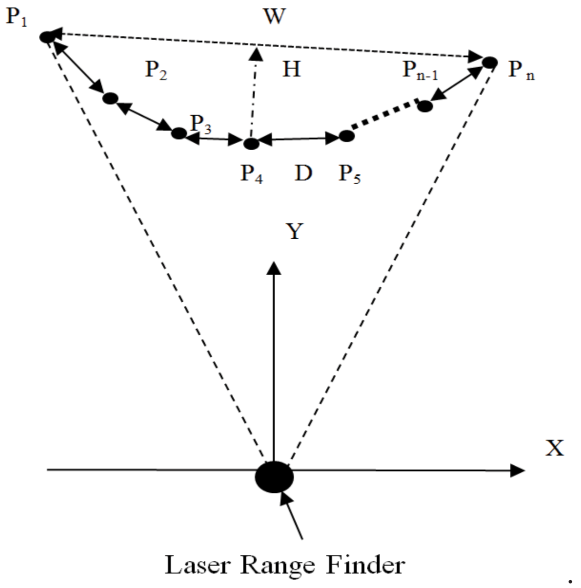


Figure 3.8: Feature determination

3.3.2 Gaussian distribution method

The GDM is a one-class classification method based on the Mahalabonis distance condition to identify the target. In order to do this, the target class is firstly modelled as a Gaussian distribution. Then a new point is identified as a target or a non-target by applying the Mahalabonis distance as follows:

$$F(\chi) = (\chi - \mu^*)^T \Sigma^{-1} (\chi - \mu^*) \tag{3.6}$$

The classifier is defined as:

$$h(\chi) = \begin{cases} target & if \quad F(\chi) \leq \theta \\ outlier & if \quad F(\chi) > \theta \end{cases} \tag{3.7}$$

The mean μ^* and the covariance matrix $\sum$ are the sample estimation of data χ. The threshold θ is set according to the target error.

Aiming to investigate the correlation between three features G, W, H of the operator, m samples of the operator are collected. From these, two respective normal data sets $\chi_i^{W_i G_i}$ and $\chi_j^{G_j H_j} (i, j = 1 \div m)$ of features (W_i, G_i) and (G_j, H_j) are set up. Utilising equation (3.7) in 2D space, the Mahalabonis distances are given by:

$$F\begin{pmatrix} W_i \\ G_i \end{pmatrix} = (d_{G_i}, d_{W_i})(\begin{matrix} \sigma_{GG} & \sigma_{GW} \\ \sigma_{WG} & \sigma_{WW} \end{matrix})(d_{G_i}, d_{W_i})^T \tag{3.8}$$

$$F\begin{pmatrix} G_i \\ H_i \end{pmatrix} = (d_{H_i}, d_{G_i})(\begin{matrix} \sigma_{HH} & \sigma_{HG} \\ \sigma_{GH} & \sigma_{GG} \end{matrix})(d_{H_i}, d_{G_i})^T \tag{3.9}$$

$$d_{G_i} = G_i - G_{mean} \; ; \; d_{W_i} = W_i - W_{mean} \text{ and } d_{H_i} = H_i - H_{mean} \tag{3.10}$$

$$G_{mean} = \frac{\sum_{i=1}^{m} G_i}{m} \; ; W_{mean} = \frac{\sum_{i=1}^{m} W_i}{m} \text{ and } H_{mean} = \frac{\sum_{i=1}^{m} H_i}{m} \tag{3.11}$$

The classifiers are defined as:

$$h\begin{pmatrix} W_i \\ G_i \end{pmatrix} = \begin{cases} target & if \quad F\begin{pmatrix} W_i \\ G_i \end{pmatrix} \leq \theta \overset{W}{G} \\ outlier & if \quad F\begin{pmatrix} W_i \\ G_i \end{pmatrix} > \theta \overset{W}{G} \end{cases} \tag{3.12}$$

$$h\begin{pmatrix} G_i \\ H_i \end{pmatrix} = \begin{cases} target & if \quad F\begin{pmatrix} G_i \\ H_i \end{pmatrix} \leq \theta_H^G \\ outlier & if \quad F\begin{pmatrix} G_i \\ H_i \end{pmatrix} > \theta_H^G \end{cases} \tag{3.13}$$

From the equations (3.12) and (3.13), the object is identified as a target if and only if $F\begin{pmatrix} W \\ G \end{pmatrix} \leq \theta_G^W$ and $F\begin{pmatrix} G \\ H \end{pmatrix} \leq \theta_H^G$.

3.3.3 Neural network classification method

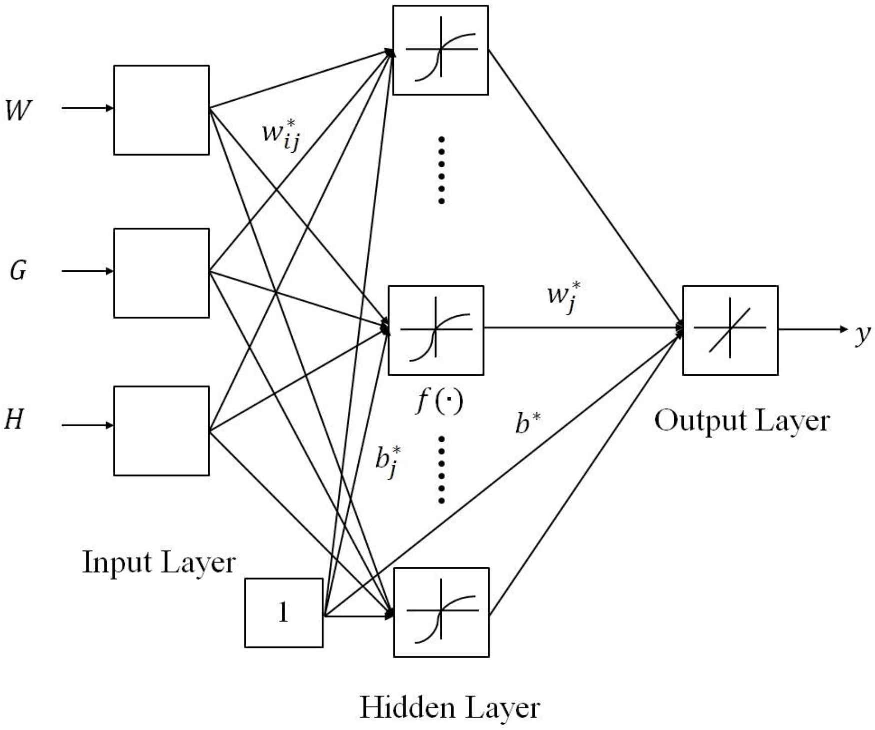


Figure 3.9: Neural network structure

The second operator detection method is based on the neural network classification. Values of three features (W, G, H) derived from the feature extraction procedure are normalised and then utilised as the inputs of the neural network classifier. This neural network has a multilayer feed-forward neural network structure with one input layer, one hidden layer and one output layer, which is shown in Figure 3.9. Details of the artificial neural network are described in Appendix A.

3.3.3.1 *Neural network structure*

The structure of the neural network has 3 input nodes which are 3 feature parameters of each object obtained from the feature extraction procedure. The output layer has one node which indicates the state of the target or non-target. The target represents the operator while the non-target denotes obstacles. It can be seen from Figure 3.9, the output of the proposed NN is given as follows:

$$Y = \sum_1^k w_j^* f\left(\sum_{i=1}^3 w_{ij}^* X_i - b_j^*\right) - b^* \tag{3.14}$$

where:

- X_i and Y are inputs and an output of the neural network, respectively.
- k represents the number of the hidden nodes
- w_j^* is the weight between jth node of the hidden layer and the output node
- w_{ij}^* is the weight between the jth hidden nodes and ith node of the input layer
- b_j^* and b^* are the biases for jth node of the hidden layer and the output node, respectively
- $f(.)$ is the activation function utilised in the hidden layer and the output layer, shown as follows:

$$f(.) = tansig(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{3.15}$$

3.3.3.2 *Initialisation*

In an effort to make a comparison between performances of different neural networks, the same initialisation is applied for all networks. In this study, the weight initialisation method introduced by Nguyen & Widrow (1990) is employed. In theory, this algorithm chooses values in order to distribute the active region of each neuron in the layer evenly across the layer's input space. It generates initial weight and bias values for a layer so that the active regions of the layer's neurons will be distributed at roughly over the input space (Mondal & Mandal 2014). Compared with the random weights and biases initialisation, the weight initialisation method of Nguyen & Widrow reduces the number of wasted neurons and increases the speed of the training works.

3.3.3.3 Training algorithms

The training procedure is implemented to find optimal parameters for the neural network model. In this study, the Levenberg-Marquardt (LM) algorithm is utilised since it is a fast and effective training algorithm. In theory, the Levenberg-Marquardt method increases the convergence speed of the error back propagation learning process by locating the local minimum of the training process based on the estimation of the second directional derivative of the error function.

For the application of operator detection in this study, the overall data set including the target data part and non-target data part is divided into three sets, a training set, validation set, and a testing set. The validation set is utilised as an early stopping strategy to guarantee that the ANN does not over-train whereas the testing set is employed for testing the generalisation of the neural network.

Besides the LM algorithm, three other training algorithms are employed for comparing the performances with the proposed approach. These training methods consist of: Scaled Conjugate Gradient, Resilient back-propagation and the Gradient descent with momentum and adaptive learning rate. These are called SCG, RP and GDX, respectively.

(a) Levenberg-Marquardt

Similar to the quasi-Newton methods, the LM algorithm is designed to approach the second-order training speed without the requirement of the Hessian matrix calculation (Hosseini, Amini & Saradjian 2003). When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as:

$$H = J^T J \tag{3.16}$$

and the gradient can be computed as

$$g = J^T e \tag{3.17}$$

where J is the Jacobian matrix that contains the first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The

Jacobian matrix can be computed through a standard back-propagation technique that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1}^{*} = x_{k}^{*} - [J^{T}J + \mu I]^{-1}J^{T}e \qquad (3.18)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm.

(b) Scaled Conjugate Gradient (SCG)

The scaled conjugate gradient (SCG) algorithm is based on conjugate directions, but this algorithm does not perform a line search at each iteration. The SCG algorithm is a variation of a standard conjugate gradient algorithm (Depenau 1995). The major idea of the conjugate gradient algorithm is that it, up to second order, produces non-interfering directions of search. This means that minimisation in one direction d_t followed by minimisation in another direction d_{t+1} implies that the error has been minimised over the whole subspace spanned by d_t and d_{t+1}. The search directions are given by:

$$d_{t+1} = -E'(w_{t+1}) + \beta_t d_t \qquad (3.19)$$

where E is overall error, w_t is a vector containing all weight values at time step t and is

$$\beta_t = \frac{\left|E'(w_{t+1})\right|^2 - E'(w_{t+1})^T E'(w_t)}{\left|E'(w_t)\right|^2} \qquad (3.20)$$

In the standard conjugate gradient algorithms the step size ϵ_t is found by a line search which can be very time consuming because this involves several calculations of the error and the first derivative. The step size is estimated by a scaling mechanism thus avoiding a time consuming line search. The step size is given by:

$$\epsilon_t = \frac{-d_t^T E'(w_t)}{d_t^T s_t + \lambda_t |d_t|^2} \tag{3.21}$$

where s_t is

$$s_t = \frac{E'(w_t + \sigma_t d_t) - E'(w_t)}{\sigma_t} \quad , \qquad 0 < \sigma_t \leq 1 \tag{3.22}$$

ϵ_t is the step size that minimises the second order approximation to the error function. s_t is the one sided difference approximation of $E''(w_t)d_t$. λ_t is a scaling parameter whose function is similar to the scaling parameter found in the Levenberg-Marquardt method.

(c) Resilient back-propagation (RP)

The Resilient Back-Propagation algorithm (RP) is a learning heuristic for supervised learning in feed-forward artificial neural networks (Cavdar & Aydin 2015). This method aims to eliminate the harmful effects of the magnitudes of the partial derivatives. In this method, only the sign of the derivative is utilised to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update (Kishore, Sunand & Rajesh babu 2012). The weight update is given by:

$$\Delta w_t = \begin{cases} -\Delta_t \; if \; \frac{\partial E}{\partial w_t} > 0 \\ \Delta_t \; if \; \frac{\partial E}{\partial w_t} < 0 \\ 0 \; else \end{cases} \tag{3.23}$$

$$w_{t+1} = w_t + \Delta w_t$$

where E is overall error; Δ_t is a vector containing all individual update values of weights, which solely determines the size of the weight update. The Δ_t is defined below:

$$\Delta_t = \begin{cases} \eta^+ * \Delta_{t-1} \; if \; \frac{\partial E}{\partial w_{t-1}} \frac{\partial E}{\partial w_t} > 0 \\ \eta^- * \Delta_{t-1} \; if \; \frac{\partial E}{\partial w_{t-1}} \frac{\partial E}{\partial w_t} < 0 \\ \Delta_{t-1} \; else \end{cases} \tag{3.24}$$

with $0 < \eta^- < 1 < \eta^+$

It can be seen from the equations 3.23 and 3.24, the adaption rule works as follows: every time the partial derivative of the corresponding weight w_t changes its sign between two consecutive iterations, which indicate that the last update was too big and the algorithm has jumped over a local minimum, the update-value Δ_t is decreased by the factor η^-. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions of the error surface (Anastasiadis, Magoulas & Vrahatis 2005).

(d) Gradient descent with the momentum and adaptive learning rate (GDX)

The Gradient descent with momentum and adaptive learning rate is an improvement on the standard gradient descent algorithm. The weight update is given by:

$$\Delta w_t = \rho \Delta w_{t-1} + \zeta \frac{\partial E}{\partial w_t} \tag{3.25}$$

$$w_{t+1} = w_t + \Delta w_t \tag{3.36}$$

where ζ denotes the learning rate; E is overall error. $0 < \rho < 1$ is a momentum parameter which must be determined by trial and error. Momentum simply adds a fraction ρ of the previous weight update to the current one. When the gradient keeps pointing in the same direction, this will increase the size of the steps taken towards the minimum. When the gradient keeps changing the direction, the momentum will smooth out the variations (Singh & Gupta 2010).

3.4 OPERATOR-FOLLOWING STRATEGY

In the literature, various operator-following behaviours have been reported such as following at a fixed distance, moving toward the target, following side by side with the target or combining target-following and obstacle avoidance.

Depending on each situation, an appropriate control algorithm will be designed for the system. In this study, the idea of the operator-following function is keeping a setting at a constant distance between the operator and the smart hospital bed. Meanwhile, the direction of the smart bed constantly faces the operator.

To achieve this goal, the smart hospital bed needs to be able to control its linear velocity and angular velocity. The distance between the hospital bed and the operator is controlled by the linear velocity in order to maintain the desired distance. The angle between the heading direction of the smart hospital bed and the direction of the operator in relation to the hospital bed local coordinate is controlled by the angular velocity of the smart hospital bed.

Two typical PID controllers are employed to control the bed's linear velocity and angular velocity. The information of the operator obtained from the operator detection algorithm is utilised for calculating the inputs of the two PID controllers. The following equations describe the PID controller that is applied to the smart hospital bed.

$$v = K_{pv}e_d + K_{iv}\int e_d + K_{dv}\dot{e_d} \tag{3.27}$$

$$\omega = K_{p\omega}e_\varphi + K_{i\omega}\int e_\varphi + K_{d\omega}\dot{e_\varphi} \tag{3.28}$$

with

$$e_d = D_a - D_d \tag{3.29}$$

$$e_\varphi = \varphi_a - \frac{\pi}{2} \tag{3.30}$$

$$tg\left(\varphi_a - \frac{\pi}{2}\right) = \frac{tg\left(\alpha - \frac{\pi}{2}\right)cos\left(\alpha - \frac{\pi}{2}\right)D^*}{O_RP + cos\left(\alpha - \frac{\pi}{2}\right)D^*} \tag{3.31}$$

$$D_a = \frac{O_RP + cos\left(\alpha - \frac{\pi}{2}\right)D^*}{\cos\left(\varphi_a - \frac{\pi}{2}\right)} \tag{3.32}$$

where K_{pv}, K_{iv} and K_{dv} are parameters of the linear velocity controller; $K_{p\omega}$, $K_{i\omega}$ and $K_{d\omega}$ are parameters of the angular velocity controller; D_a , D_d denote the actual distance between the hospital bed and the target and the desired tracking distance, respectively. P is the position of the laser range finder. D^* is the distance between the operator and the laser scanner. (O,X,Y), (O_R, X_x, Y_y) and (P, x_p, y_p) are the global coordinate system, the local coordinate system of the hospital bed and the local coordinate system of the laser scanner, respectively. a represents the length of the smart hospital bed. α and φ_a are defined in Figure 3.10.

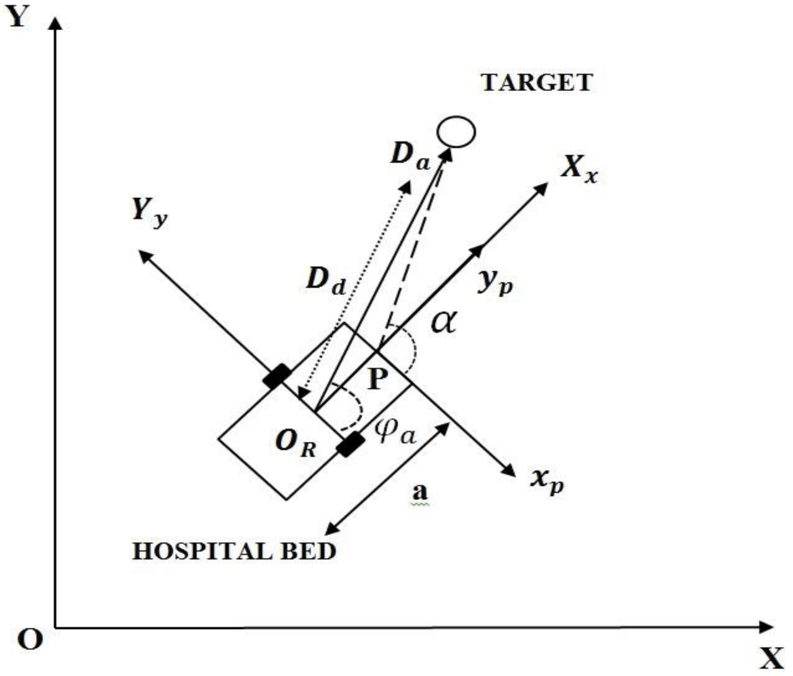


Figure 3.10: The hospital bed system and the target for tracking

3.5 EXPERIMENTAL RESULTS

In this section, experiments have been divided into two major parts. Section 3.5.1 contains several experiments which are implemented to estimate the performance of the two proposed operator detection algorithms. A comparison between the accuracy rate of the two proposed approaches is also executed in this part. By comparing the results, the best algorithm is employed for the experiment in the next section. Section 3.5.2 shows the experimental results of the smart hospital bed when it follows an operator.

In relation to the operator following function of the smart hospital bed, the operator can be a hospital staff member or an autonomous navigation robot. In recent years, the autonomous navigation robot has been widely used within the hospital environment to support clinical staff in transportation tasks. In the future, this kind of robot can replace hospital staff to collaborate with the smart bed to transport the patient. Accordingly, in this study, an autonomous navigation robot called Turtlebot is utilised to implement the experiments related to the operator-following task. Details of this robot are shown in Appendix B.

3.5.1 Operator detection performance

The quality of two proposed operator detection algorithms is tested in this section. Two data sets were collected from the laser range finder. The first data set ℕ included 200 samples of the Turtlebot. On the other hand, the second data set ℚ contained 750 samples of the Turtlebot and 1500 samples of the non-Turtlebot objects. The non-Turtlebot objects consisted of a trash can, a recycling bin, a carton box, human legs, a flower pot and a warning column. The second data set ℚ was divided into three parts which are a training set, validation set, and testing set and these randomly selected with proportions of 35%, 35%, and 30% out of the overall data, respectively. In total, there were 2450 samples in the two data sets. The distribution of the two data sets is presented in Figure 3.11.

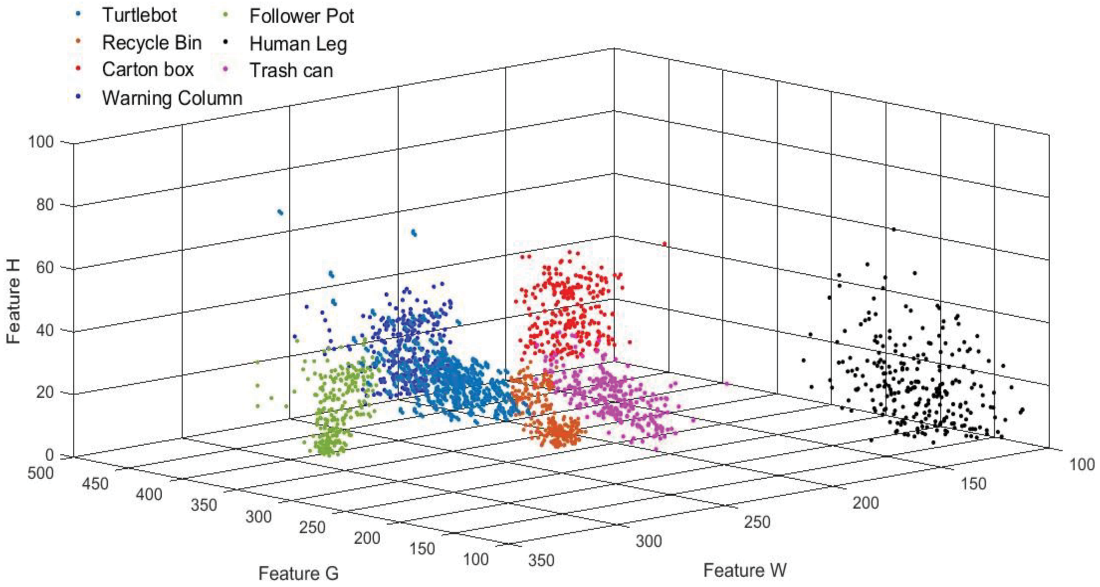


Figure 3.11: Distribution of the two data sets

To measure the performance of the classification results, sensitivity (or true positive rate) and specificity (or true negative rate) were used and given as follows:

$$Sensitivity = \frac{TP}{TP+FN} \tag{3.33}$$

$$Specificity = \frac{TN}{TN+FP} \tag{3.34}$$

where TP (True Positive) is the number of target events which are correctly classified as target; FN (False Negative) is the number of target events which are wrongly classified as non-target; TN (True Negative) is the number of non-target events which are correctly classified as non-target; FP (False Positive) is the number of non-target events which are wrongly classified as target (Nguyen, Steven & Nguyen 2014).

3.5.1.1 Gaussian distribution method results

This experiment was implemented to estimate the efficiency of the Gaussian Distribution Method in terms of detecting an operator. To do this, the first data set $\mathbb{N}$ was utilised as a training data set for developing two classifiers $h\begin{pmatrix} W_i \\ G_i \end{pmatrix}$ and $h\begin{pmatrix} G_i \\ H_i \end{pmatrix}$.

In terms of geometry, conditions 3.12 and 3.13 can be explained as follows:

- In (W, G)coordinate system, a boundary B_{WG} is created to cover the training data set (W, G).
- In (G, H)coordinate system, a boundary B_{GH} is created to cover the training data set (G, H).

Considering a testing object with 3 features (W^*, G^*, H^*), if the position of the point (W^*, G^*) is in the boundary B_{WG} and the position of the point (G^*, H^*) is in the boundary B_{GH} , the testing object is the target.

To illustrate our proposed method in geometry, a *DDtools* toolbox was utilised to build the boundaries B_{WG} and B_{GH}. Figures 3.12 and 3.13 show the construction of the boundaries B_{WG} and B_{GH} with the different threshold values, respectively. Changing the threshold θ from 0.2 to 0.05 with a step of 0.05, four respective contours (1, 2, 3, 4) were obtained.

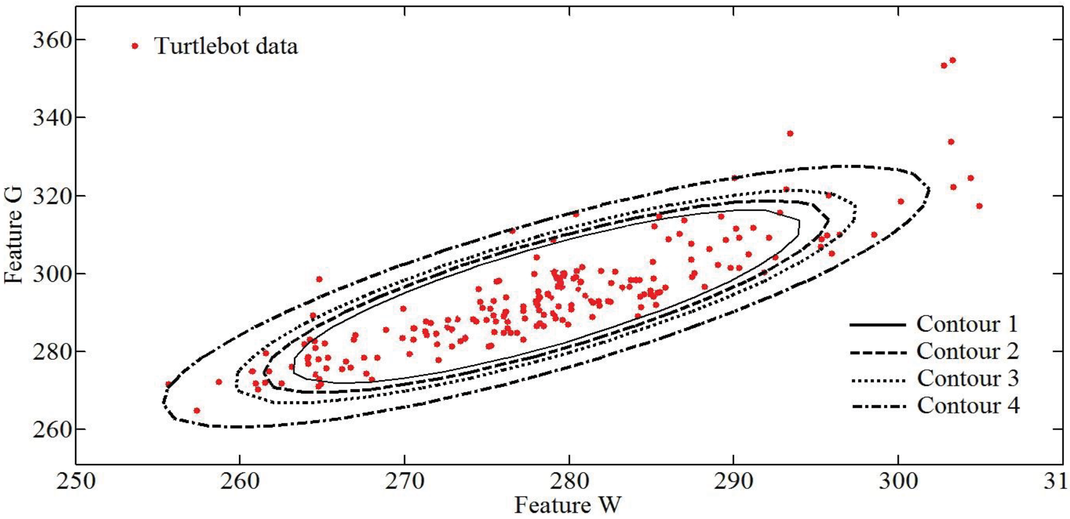


Figure 3.12: The Boundary B_{WG} with θ_{G}^{W} = 0.2, 0.15, 0.1, 0.05

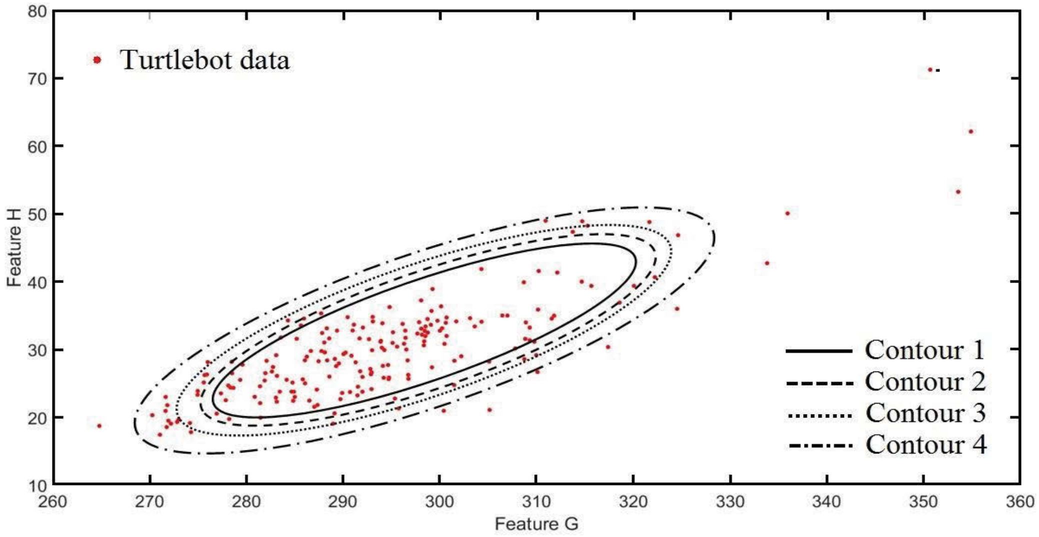


Figure 3.13: The Boundary B_{GH} with θ_{H}^{G} = 0.2, 0.15, 0.1, 0.05

It can be seen from the Figures 3.12 and 3.13 that the performance of the classifiers is dependent on the appropriate value of the thresholds θ_{G}^{W} and θ_{H}^{G}. For testing the efficiency of the proposed method, the testing set of the second data set $\mathbb{Q}$ was employed. After extensive experiments, both thresholds θ_{G}^{W} and θ_{H}^{G} were chosen at 0.06. Table 3.1 depicts the average testing result and best performance of the proposed learning methods with respect to sensitivity and specificity over 30 runs.

GDM	Sensitivity	Specificity
Average testing result	89.2%	81.4%
Best performance	91%	88.5%

Table 3.1: Experimental results of the Gaussian Distribution Method

3.5.1.2 Neural network classification method results

The aim of this experiment is to estimate performance of the neural network classification with various learning algorithms. To do this, the second data set $\mathbb{Q}$ was employed for the training and testing process. Training parameters including the maximum epoch and minimum gradient were chosen delicately. In particular, the maximum epoch was 1000. In addition, the minimum gradient was 0.0001 and a goal was set for 0.001.

For a comparison of the performances, four training algorithms, Levenberg-Marquardt (LM), scaled conjugate gradient (SCG), resilient back propagation (RP) and gradient descent with momentum (GDX), were utilised to train the neural network model. Table 3.2 presents the average (mean) training, validation and testing results of the proposed learning methods with respect to sensitivity and specificity over 30 runs. In addition, the best performance of neural network based operator detection using four algorithms is reported in Table 3.3. It can be seen from tables 3.2 and 3.3, the NN-LM achieves the best performance when comparing with three other learning algorithms.

Method	**Training**		**Validation**		**Testing**	
	Sens (%)	Spec (%)	Sens (%)	Spec (%)	Sens (%)	Spec (%)
NN-LM	91.24	90.58	91.15	86.37	91.65	88.56
NN-SCG	89.84	87.68	88.45	87.76	88.55	86.4
NN-RP	88.45	85.34	87.28	86.5	87.75	85.6
NN-GDX	89.76	88.24	89.6	87.46	88.25	85.24

Table 3.2: Mean values of training, validation and testing results utilising four algorithms

Method	**Hidden nodes**	**Training**		**Validation**		**Testing**	
		Sens (%)	Spec (%)	Sens (%)	Spec (%)	Sens (%)	Spec (%)
NN-LM	5	93.45	92.18	92.43	88.64	92.41	89.8
NN-SCG	4	91.83	88.56	92.16	90	91.13	88.81
NN-RP	6	91.3	90.26	86.13	89.49	90.58	90.26
NN-GDX	5	91.63	89.12	90.03	89.52	89.38	86.85

Table 3.3: Best performance of NN based operator detection using four algorithms

3.5.1.3 Comparison of two proposed classification methods

Table 3.4 shows the comparison between the performance of the GDM classifier and the neural network classifier. Compared to the performance of the neural network classification method using Levenberg-Marquardt, the GDM obtained worse testing results. The best performance of 91 % sensitivity and 88.5% specificity of GDM are lower than that of the neural network classification method (92.41% and 89.8%).

Method	**Mean values of testing results**		**Best performance of testing results**	
	Sens (%)	Spec (%)	Sens (%)	Spec (%)
NN-LM	91.65	88.56	92.41	89.8
GDM	89.2	81.4	91%	88.5

Table 3.4: Comparisons of mean values and best performances of testing results for two proposed classification algorithms

3.5.2 Operator following performance

The experiment in this section was implemented to estimate the performance of the smart hospital bed when it followed a target without an employment of a low-level controller. In order to do this, the neural network model with the best performance obtained from the previous experiment was utilised as the operator identifier for the smart hospital bed. In addition, two PID controllers were utilised as two path-following controllers to minimise errors of distance and angle while the bed tracked the operator. The parameters of the two PIDD controllers were experimentally chosen as $K_{pv} = 1.6, K_{iv} = 1.8, K_{dv} = 0.15, K_{p\omega} = 2.5, K_{i\omega} = 3, K_{d\omega} = 0.2$. Figure 3.14 illustrates the target-following task of the smart hospital bed.

In Figure 3.14:

- Points O, P denote starting positions of the smart hospital bed and the operator, respectively.
- The circle represents the operator while the rectangle depicts the smart hospital bed.
- From P to C the operator moves with $v = 0.4(m/s)$ and $\omega = 0\ (rad/s)$
- From C to F velocities including $v = 0.5(m/s)$ and $\omega = 0.125\ (rad/s)$ are introduced to the operator.
- From F to K the operator travels with $v = 0.4(m/s)$ and $\omega = 0\ (rad/s)$.
- From K to N velocities consisting of $v = 0.5(m/s)$ and $\omega = 0.125\ (rad/s)$ are applied on the operator.
- From N to P the operator moves with a linear velocity of 0.4 (m/s) and keeps the angular velocity at zero.
- Additionally, path NC is an uphill path; meanwhile path FK is a downhill path.
- Images I, II, III, IV illustrate the operator-following task of the smart hospital bed at 4 positions including (O,P); (D,E); (G,H) and (L,M), respectively.
- During the operator-following, the bed system carries an 85 kg person.

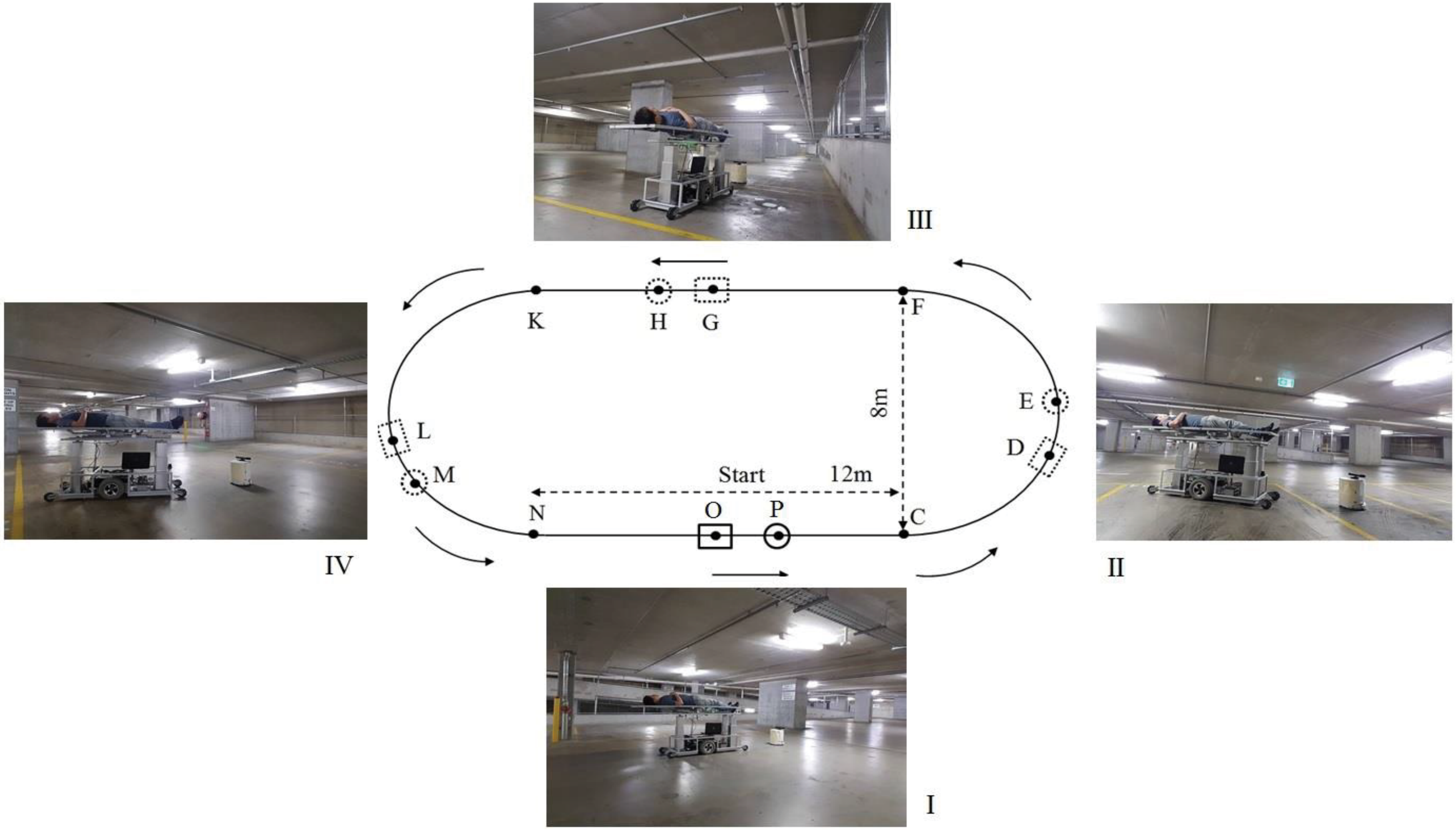


Figure 3.14: Demonstration of the operator-following task

Experimental results of the operator-following task are shown in Figure 3.15, 3.16 and 3.17.

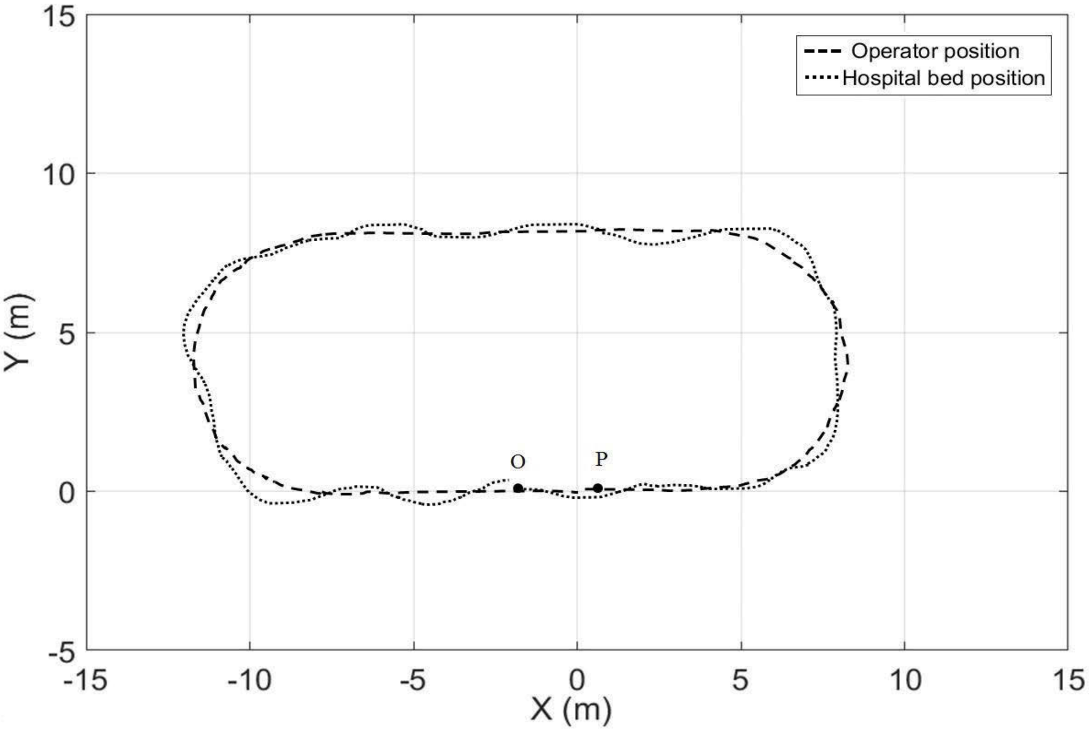


Figure 3.15: Trajectory of the hospital bed and operator

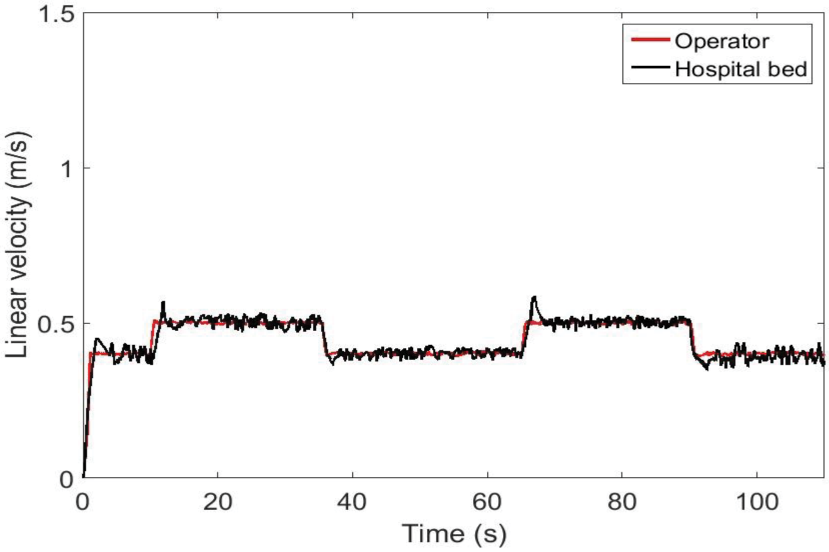


Figure 3.16: Linear velocity of the hospital bed and operator

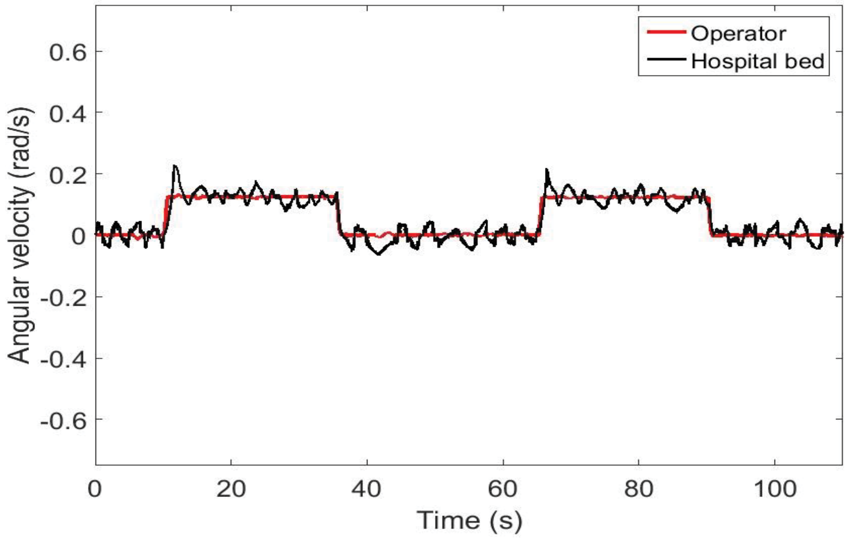


Figure 3.17: Angular velocity of the hospital bed and operator

3.6 DISCUSSION

The Gaussian Distribution Method is a simple one-class classification, based on calculating Mahalabonis distances between testing points and a training target data set to classify target and non-target objects. In table 3.1, the average testing results of the GDM based operator detection approach are 89.2 % and 81.4 % for sensitivity and specificity, respectively. The deviation between the average testing result and the best performance of the sensitivity is 1.8% whereas that of the specificity is 7.1%. This means that the GDM based operator detection algorithm is reasonably efficient but could still be improved.

In terms of the average testing result, it can be seen from Table 3.2 that the NN-LM learning algorithm provides the best sensitivity and specificity (91.65% and 88.56%) compared with the other three algorithms containing SCG, RP and GDX whose sensitivity and specificity are (88.55% and 86.4%), (87.75% and 85.6%) and (88.25% and 85.24%), respectively. On the other hand, results reported in Table 3.3 show that using the LM algorithm with the 5 hidden nodes produces the best classification in which the training results of 93.45% sensitivity and 92.18% specificity are gained and the testing set leads to 92.41% sensitivity and 89.8% specificity.

In a comparison of the performance of the GDM classifier with that of the NN-LM classifier, it can be observed from table 3.4 that the average testing results of the NN-LM classifier are more than those of the GDM classifier with amounts of 2.45% sensitivity and 7.16% specificity. In addition, the best performance of the NN-LM based operator detection method is also more than that of the GDM based operator detection method with amounts of 1.41% sensitivity and 1.3% specificity. Moreover, the deviation between the best performance and mean value of the specificity of testing results in the GDM classifier is larger than that of the NN-LM classifier. Therefore, the neural network classification method provides the better performance than that of the GDM classification approach

It can be observed from the results of section 3.5.2 that the performance of the smart hospital bed during the operator-following is acceptable but could still be improved. One way to enhance the operator-following performance of the bed system is to apply a low-level controller to its operator-following control strategy. The low-level controller has an advantage in terms of dealing with uncertain factors of the bed system such as internal system uncertainties, environmental conditions, and frictions between surface and tyre. Consequently, the operator-following control strategy using the low-level controller provides the robustness of the overall system.

CHAPTER 4 . AN OPTIMAL MULTIVARIABLE PID CONTROL STRATEGY FOR SMART HOSPITAL BEDS

4.1 INTRODUCTION

The experimental results presented in the previous chapter (section 3.5.2) have demonstrated that the operator-following performance of the smart hospital bed can be affected by uncertain factors. In fact, there are various uncertain factors impacting the bed system such as internal system uncertainties, coupling effects, the friction between the bed tyre and the surface and the variation of the centre of gravity. These factors lead to the instability and inefficiency of the overall bed system during its operation. As a result, it is necessary to develop a low-level control for dealing with system uncertainties.

Treating the smart hospital bed as a multivariable system with uncertainties, its dynamic model is required for designing a low-level controller. Due to the variation of uncertain dynamics, it is difficult to calculate an exact hospital bed dynamic model. Therefore, with an assumption of bounded uncertainties, three models consisting of a lower bounded model, a nominal model and an upper bounded model have been employed for representing the dynamics of the smart hospital bed. Parameters of the three dynamic models can be derived from a parametric identification process.

In the control design stage of the multivariable system, there have been various control methods such as a centralised control method, decentralised control method and decoupling control method. Due to the advantage of reducing a multivariable system to a series of independent scalar systems, the decoupling technique is integrated into the proposed control approach. Thanks to this technique, the design procedure and its computational complexity are simplified.

Owning the ease-to-use and real-time property, Proportional-Integral-Derivative control has been widely utilised to develop tracking controllers. In this chapter, the major work focuses on designing an Optimal Multivariable PID Control strategy for the smart hospital bed system. This control algorithm is based on a Triangular Diagonal Dominance decoupling technique introduced by Nguyen *et al* (1979). The proposed control strategy contains two design stages. First, a decoupler with a TDD property is constructed to reduce the multivariable control problem of the smart hospital bed into a series of independent scalar control problems. Secondly, an Optimal PID controller is designed for each independent scalar system. Since the proposed approach combines the TDD decoupling approach and PID control design, it is called the Optimal Multivariable PID Control (OMPIDC) approach.

The chapter is organised as follows. In section 4.2, a dynamic multivariable model of the smart hospital bed system is obtained, including the presence of transportation lag. Section 4.3 describes the Optimal Multivariable PID Control approach in detail. In the next section, various experiments are provided to estimate the performance of the proposed control approach. Finally, a discussion can be found in the last part of this chapter.

4.2 MULTIVARIABLE DYNAMIC MODEL

As mentioned in section 3.2, the smart hospital bed contains two steering wheels and four free multidimensional wheels. Each driving wheel is placed at the middle side of the lower frame of the bed so that both of them are in the same axis. Meanwhile, the four free wheels are mounted in the four corners of the hospital bed base. Motion and orientation of the steering wheels are achieved by two DC motors directly connected to them.

4.2.1 Mathematical model

The smart hospital bed system is shown in Figure 4.1.

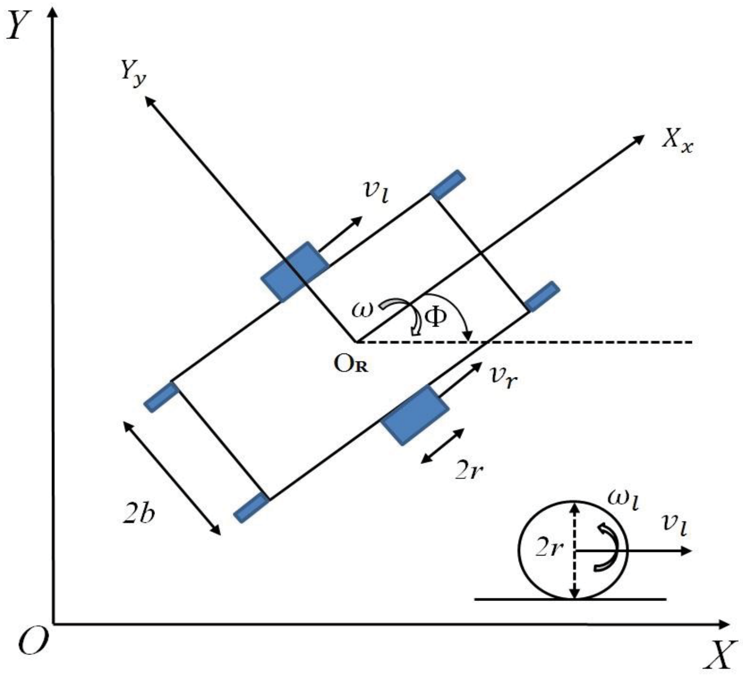


Figure 4.1: The smart hospital bed system

In Figure 4.1, r and $2b$ denote the radius of the driving wheels and the distance between these wheels, respectively. Point C represents the centre of mass of the hospital bed. Since the smart hospital bed has a symmetrical structure, the position of the point C is also the position of point O_R depicting the origin of the local coordinate frame of the bed. (O, X, Y) is the global coordinate frame where (O_R, X_x, Y_y) is the local coordinate frame attached to the hospital bed. Φ indicates the angle between the heading direction of the bed system and the OX axis.

According to the study of Das *et al* (2006), the full dynamic equation of the smart hospital bed dynamics can be described as follows:

$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \overline{\varrho_{dr}} \\ \overline{\varrho_{dl}} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ b & -b \end{bmatrix} \begin{bmatrix} \varrho_r \\ \varrho_l \end{bmatrix} \tag{4.1}$$

where m, I denote the bed mass and the bed inertia, respectively. ϱ_r is the input torque to the right steering wheel whereas ϱ_l is the input torque to the left steering wheel. $[\overline{\varrho_{dr}} \quad \overline{\varrho_{dl}}]^T$ is the bounded uncertainties vector. $[\dot{v} \quad \dot{\omega}]^T$ is the first derivative of the velocities of the bed system including the linear velocity v and the angular velocity ω.

Ignoring the inductance of the motors, the equation governing the right and left DC motors of the bed system can be given as:

$$\begin{bmatrix} \varrho_{sr} \\ \varrho_{sl} \end{bmatrix} = \begin{bmatrix} K_{Tr} i_r \\ K_{Tl} i_l \end{bmatrix} \tag{4.2}$$

where $[\varrho_{sr} \quad \varrho_{sl}]^T$ is the torque vector in which $\varrho_{sr}$is the torque generated by the right motor and ϱ_{sl} is the torque generated by the left motor. K_{Tr} and K_{Tl} represent torque constants of the right motor and the left motor whereas i_r and i_l indicate the current flowing in the coils of the right and left motors, respectively.

$$\begin{bmatrix} u_r \\ u_l \end{bmatrix} = \begin{bmatrix} R_{ar} i_r + K_{br} \dot{\theta}_{rm} \\ R_{al} i_l + K_{bl} \dot{\theta}_{lm} \end{bmatrix} \tag{4.3}$$

where R_{ar} and R_{al} denote resistances of the right and left motors, respectively. Counter electromotive force coefficients of the right and left motors are represented by $K_{br}$and K_{bl}. $\dot{\theta}_{rm}$ and $\dot{\theta}_{lm}$ are the respective right and left motors' velocities.

The relationship between the motors' velocities and the driving wheels' velocities is given by:

$$\begin{bmatrix} \dot{\omega}_r \\ \dot{\omega}_l \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \dot{\theta}_{rm} \\ \dot{\theta}_{lm} \end{bmatrix} \tag{4.4}$$

Additionally, the relationship between the torques of the motors and the torques of the driving wheels are calculated as:

$$\begin{bmatrix} \varrho_r \\ \varrho_l \end{bmatrix} = N \begin{bmatrix} \varrho_{sr} \\ \varrho_{sl} \end{bmatrix} \tag{4.5}$$

The linear and angular velocity of the hospital bed system can be obtained from the velocities of the steering wheels, shown as follows:

$$v = \frac{1}{2} r(\omega_r + \omega_l) \tag{4.6}$$

$$\omega = \frac{r}{2b}(\omega_r - \omega_l) \tag{4.7}$$

Utilising equations 4.1 to 4.7, the dynamic equation of the hospital bed system, consisting of motor dynamics can be given as:

$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}\begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} + \begin{pmatrix} Q_{dr} \\ Q_{dl} \end{pmatrix} = \frac{NK_T}{R_a}\frac{1}{2b}\begin{bmatrix} 1 & 1 \\ 2r & -2r \end{bmatrix}\begin{pmatrix} u_r \\ u_l \end{pmatrix} - \frac{N^2K_TK_b}{R_a}\frac{1}{2b}\begin{bmatrix} 1 & 1 \\ 2r & -2r \end{bmatrix}\begin{pmatrix} v \\ \omega \end{pmatrix} \quad (4.8)$$

where $K_{Tr} = K_{Tl} = K_T$; $K_{br} = K_{bl} = K_b$; $R_{ar} = R_{al} = R_a$

Due to variation of dynamic uncertainties, it is difficult to calculate precisely the parameters of the equation (4.8). As a result, without loss of generality, the dynamic equation (4.8) of the smart hospital bed system is expressed as the following equation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} G_{11}(s) + \Delta G_{11}(s) & G_{12}(s) + \Delta G_{12}(s) \\ G_{21}(s) + \Delta G_{21}(s) & G_{22}(s) + \Delta G_{22}(s) \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.9)$$

where u_1 and u_2 are reference inputs. $G_{ij}(s)$ and $\Delta G_{ij}(s)$ denote the element of system transfer matrix $G(s)$ and system uncertainty, respectively.

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} + \begin{bmatrix} \Delta G_{11}(s) & \Delta G_{12}(s) \\ \Delta G_{21}(s) & \Delta G_{22}(s) \end{bmatrix} \quad (4.10)$$

with
$$G_{ij}(s) = \frac{K_{ij}}{(1+sT_{ij})}e^{-\tau_{ij}s} \qquad i,j = 1,2. \quad (4.11)$$

Assuming that the system uncertainty $\Delta G_{ij}(s)$ is bounded, the hospital bed dynamic model can be represented by three dynamic models consisting of a lower bounded model, a nominal model and an upper bounded model, shown as follows:

$$G_{low}(s) = \begin{bmatrix} \frac{K_{11}^{low}}{(1+sT_{11}^{low})}e^{-\tau_{11}^{low}s} & \frac{K_{12}^{low}}{(1+sT_{12}^{low})}e^{-\tau_{12}^{low}s} \\ \frac{K_{21}^{low}}{(1+sT_{21}^{low})}e^{-\tau_{21}^{low}s} & \frac{K_{22}^{low}}{(1+sT_{22}^{low})}e^{-\tau_{22}^{low}s} \end{bmatrix} \quad (4.12)$$

$$G_{nom}(s) = \begin{bmatrix} \frac{K_{11}^{nom}}{(1+sT_{11}^{nom})}e^{-\tau_{11}^{nom}s} & \frac{K_{12}^{nom}}{(1+sT_{12}^{nom})}e^{-\tau_{12}^{nom}s} \\ \frac{K_{21}^{nom}}{(1+sT_{21}^{nom})}e^{-\tau_{21}^{nom}s} & \frac{K_{22}^{nom}}{(1+sT_{22}^{nom})}e^{-\tau_{22}^{nom}s} \end{bmatrix} \quad (4.13)$$

$$G_{up}(s) = \begin{bmatrix} \frac{K_{11}^{up}}{(1+sT_{11}^{up})}e^{-\tau_{11}^{up}s} & \frac{K_{12}^{up}}{(1+sT_{12}^{up})}e^{-\tau_{12}^{up}s} \\ \frac{K_{21}^{up}}{(1+sT_{21}^{up})}e^{-\tau_{21}^{up}s} & \frac{K_{22}^{up}}{(1+sT_{22}^{up})}e^{-\tau_{22}^{up}s} \end{bmatrix} \quad (4.14)$$

4.2.2 Parametric identification process

In this section, parameters of the hospital bed dynamic models in equation 4.12 to 4.14 are derived from a parametric identification process. The idea of this method is first to measure the responses of the hospital bed system with respect to various step inputs and then to identify the varying ranges of the system parameters.

Applying this parametric identification process on the smart hospital bed, the varying ranges of the hospital bed system parameters were estimated as follows:

$$K_{11} = [0.93; 1.17]; \quad T_{11} = [0.24; 1.43]; \quad \tau_{11} = [0.07; 0.35] \tag{4.15}$$

$$K_{12} = [0.01; 0.03]; \quad T_{12} = [0.07; 0.36]; \quad \tau_{12} = [0.07; 0.25] \tag{4.16}$$

$$K_{21} = [0.048; 0.08]; \; T_{21} = [0.06; 0.2]; \quad \tau_{21} = [0.04; 0.35] \tag{4.17}$$

$$K_{22} = [0.28; 1.09]; \quad T_{22} = [0.3; 1.29]; \quad \tau_{22} = [0.07; 0.3] \tag{4.18}$$

To obtain the parameters of the nominal dynamic model, the following equations were employed:

$$K_{nom} = \frac{1}{M}\sum_{i=1}^{M} K_i \tag{4.19}$$

$$T_{nom} = \frac{1}{M}\sum_{i=1}^{M} T_i \tag{4.20}$$

$$\tau_{nom} = \frac{1}{M}\sum_{i=1}^{M} \tau_i \tag{4.21}$$

where M is the number of experiments.

According to the boundary values of the model parameters in (4.15 – 4.18), the approximate dynamic model of the hospital bed system can be represented by three models including a lower bounded model, a nominal model and an upper bounded model, and illustrated as follows:

$$G_{lower}(s) = \begin{bmatrix} \frac{0.93}{(1+0.24s)} e^{-0.07s} & \frac{0.015}{(1+0.07s)} e^{-0.07s} \\ \frac{0.048}{(1+0.06s)} e^{-0.04s} & \frac{0.28}{(1+0.3s)} e^{-0.07s} \end{bmatrix} \tag{4.22}$$

$$G_{nom}(s) = \begin{bmatrix} \frac{1}{(1+0.8s)} e^{-0.1s} & \frac{0.025}{(1+0.1s)} e^{-0.2s} \\ \frac{0.06}{(1+0.1s)} e^{-0.1s} & \frac{0.51}{(1+0.65s)} e^{-0.1s} \end{bmatrix} \tag{4.23}$$

$$G_{upper}(s) = \begin{bmatrix} \frac{1.17}{(1+1.43s)} e^{-0.35s} & \frac{0.03}{(1+0.36s)} e^{-0.25s} \\ \frac{0.08}{(1+0.2s)} e^{-0.35s} & \frac{1.09}{(1+1.29s)} e^{-0.3s} \end{bmatrix} \tag{4.24}$$

In the equation (4.11), the time delay part $e^{-\tau_{ij}s}$ can be replaced by the following first order form:

$$e^{-\tau_{ij}s} \approx \frac{1}{1+\tau_{ij}s} \tag{4.25}$$

This approximation is utilised for simplifying the control design process in the state space. As a result, three dynamic models of the smart hospital bed are rewritten as below:

$$G_{lower}(s) = \begin{bmatrix} \frac{0.93}{(1+0.24s)(1+0.07s)} & \frac{0.015}{(1+0.07s)(1+0.07s)} \\ \frac{0.048}{(1+0.06s)(1+0.04s)} & \frac{0.28}{(1+0.3s)(1+0.07s)} \end{bmatrix} \tag{4.26}$$

$$G_{nom}(s) = \begin{bmatrix} \frac{1}{(1+0.8s)(1+0.1s)} & \frac{0.025}{(1+0.1s)(1+0.2s)} \\ \frac{0.06}{(1+0.1s)(1+0.1s)} & \frac{0.51}{(1+0.65s)(1+0.1s)} \end{bmatrix} \tag{4.27}$$

$$G_{upper}(s) = \begin{bmatrix} \frac{1.17}{(1+1.43s)(1+0.35s)} & \frac{0.03}{(1+0.36s)(1+0.25s)} \\ \frac{0.08}{(1+0.2s)(1+0.35s)} & \frac{1.09}{(1+1.29s)(1+0.3s)} \end{bmatrix} \tag{4.28}$$

The step response of the smart hospital bed dynamics is shown in Figure 4.2.

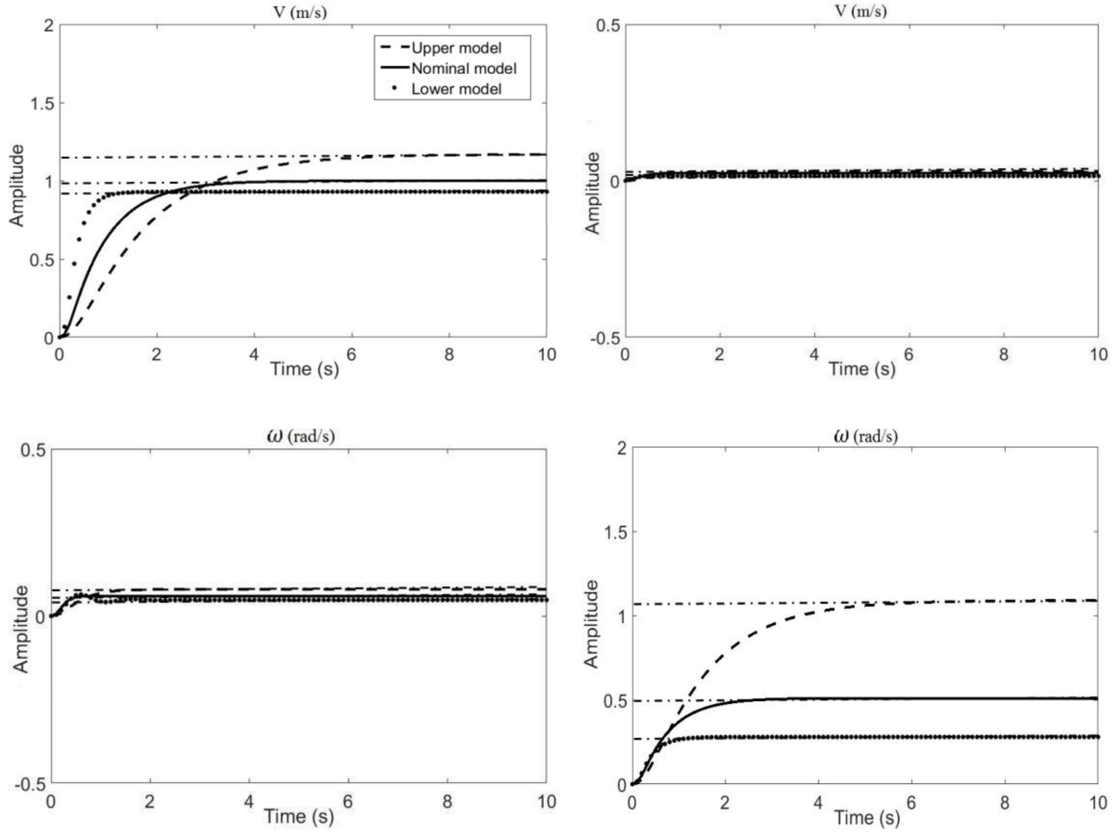


Figure 4.2: Step response of the smart hospital bed dynamics

4.3 OPTIMAL MULTIVARIABLE PID CONTROL STRATEGY

In theory, the aim of the Nguyen *et al* 's decoupling technique is to determine a stable and proper pre-compensator so that the compensated system has a special property which is called the Triangular Diagonal Dominance (TDD) (1979). With this characteristic, only diagonal elements of the transfer function matrix are taken into consideration for the closed-loop control design. In other words, the TDD decoupling technique offers an effective solution to transform a multivariable control problem into a collection of independent scalar control problems.

In this section, an Optimal Multivariable PID Control strategy with two design stages is proposed for the smart hospital bed. In the first stage, a pre-compensator with TDD property is constructed to decrease the crossing interaction between system's inputs and outputs to an acceptable low level. Through the obtained pre-compensator, the multivariable control design problem is simplified into two independent scalar control design problems. In the second stage, an Optimal PID controller is developed for each independent scalar control problem. The structure of the Optimal Multivariable PID Control is illustrated in Figure 4.3.

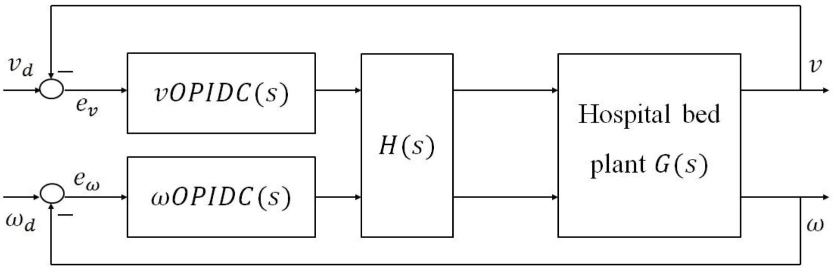


Figure 4.3: Optimal Multivariable PID control scheme

4.3.1 Stage 1: Triangular Diagonal Dominance decoupling

Consider a smart hospital bed whose transfer function matrix is given by a proper square $2x2$ plant $G(s)$. In order to utilise a decoupling technique at this stage, the nominal model of the smart hospital bed is adopted. The aim of this design stage is to seek a pre-compensator $H(s)$ so that the compensated system is in TDD form. According to Nguyen *et al* (1979), a pre-compensator matrix $H(s)$ can always be constructed such that:

$$P_{TDD}(s) = G(s).H(s) \tag{4.29}$$

where $P_{TDD}(s)$ is triangular

Apply the following steps to construct the pre-compensator $H(s)$:

Step 1: Move the lowest degree element in the first row to the $(1-1)$ position.

Step 2: Subtract a multiple of the first column from the second column to make sure that $\delta(G_{12}) < \delta(G_{11})$

In order to utilise the diagonal elements of the $P_{TDD}(s)$ for the closed-loop control design stage, the compensated plant $P_{TDD}(s)$ must satisfy all conditions of the TDD property which are depicted as follows:

- The $G(s)$ is stable
- $G(s)$ is in minimum phase

- $\delta[|G(s)|_+] = \delta[G_+(s)]$
- The off-diagonal part of the pre-compensated matrix $P_{TDD}(s)$ is stable
- Any unstable pole in the off-diagonal part of $P_{TDD}(s)$ occurs in only one off-diagonal entry and this also occurs in the same row or same column, with at least the same multiplicity

After checking Nguyen *et al* ‘s TDD conditions, if the $P_{TDD}(s)$is not TDD, the two following steps are utilised:

Step 1: Calculate $T(s) = diag[\frac{l_i(s)}{m_i(s)}]$ such that $T(s)G(s)$ is proper and after cancellation, proper.

Step 2: Calculate $U(s)$ such that $T(s)G(s)U(s)$ is triangular, because $T(s)$ is diagonal and $G(s)U(s)$ is triangular.

If $P_{TDD}(s)$ is TDD, the off-diagonal elements do not affect the stability of the system. In other words, the multivariable system is simplified into two independent scalar systems.

4.3.2 Stage 2: Optimal PID control

Through the TDD decoupling technique, the two obtained velocity subsystems contain a linear velocity subsystem and an angular velocity subsystem. For each subsystem, an independent PID controller is developed to let a velocity output track a desired reference input. In this study, a combination of a Root Locus technique and a Least Square optimisation algorithm is utilised to obtain optimal PID gains for the proposed controllers. A general control scheme for velocity control loops is shown in Figure 4.4.

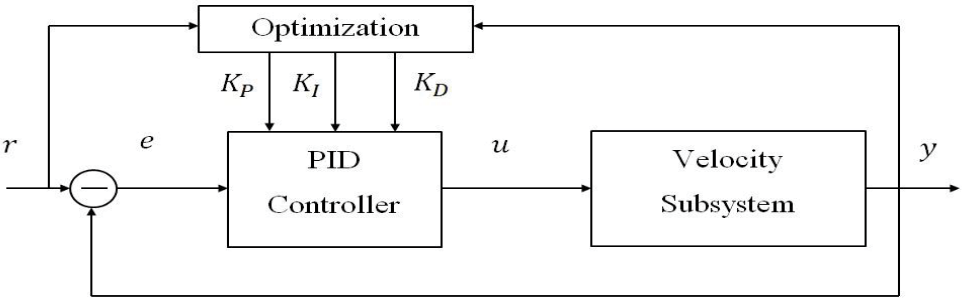


Figure 4.4: Optimal PID control method for a velocity subsystem

A mathematical description of the PID controller is given as follows:

$$\mathrm{G_{PID}(s) = K_P + K_I \frac{1}{s} + K_D s} \tag{4.30}$$

In the equation 4.30, the proportional term $\mathrm{K_P}$ provides an overall control action proportional to the error signal through constant gain factor. The integral term $\mathrm{K_I}$ decreases steady state errors through low-frequency compensation whereas the derivative term $\mathrm{K_D}$ enhances transient response through high-frequency compensation (Campo 2012).

4.3.2.1 Initialising PID parameters

As the optimisation algorithm needs initial parameters for the first iteration, a tuning method is implemented to obtain parameters for a PID controller. In theory, Root Locus technique based PID tuning is a design method in which the compensator transfer function is designed such that the response gets the desired shape. The compensator transfer function is designed with the help of Root Locus plots. First, the desired parameters are added in the Sisotool and accordingly the region of stability is highlighted in the step response plot. By adding poles and zeros in the root locus plot the desired shape of loop is achieved within the highlighted region (Bhowate & Deogade 2015).

4.3.2.2 Least Square Optimisation

One way to optimise the PID controller is to minimise the error between the output y and input r for all time steps from $\mathrm{t_0}$ (initial instant of time) to $\mathrm{t_f}$ (final instant of time) with the variables which are the parameters of the PID controller (Velasco et al. 2008). This can be done by utilising the Least Square Optimisation algorithm:

$$\mathrm{min_x \|f(x)\|_2^2 = min_x(f_1(x)^2 + f_2(x)^2 + \cdots + f_n(x)^2\,)} \tag{4.31}$$

where,

$$\mathrm{f(x)} = \begin{bmatrix} \mathrm{f_1(x)} \\ \mathrm{f_2(x)} \\ \vdots \\ \mathrm{f_n(x)} \end{bmatrix} = \begin{bmatrix} \mathrm{r_1(x) - y_1(x)} \\ \mathrm{r_2(x) - y_2(x)} \\ \vdots \\ \mathrm{r_n(x) - y_n(x)} \end{bmatrix} \tag{4.32}$$

with n = number of time steps from t_0 to t_f.

4.4 RESULTS

Firstly, the OMPID controller was designed for the smart hospital bed which was described in the previous chapter. After obtaining the controller, several real-time experiments were implemented for evaluating the controller's performance.

4.4.1 OMPID controller design

From the section 4.2.2, the obtained nominal model of the smart hospital bed was utilised to design the OMPID controller.

4.4.1.1 Stage 1: TDD decoupler

Utilising the TDD decoupling technique (Nguyen & Anderson 1979), the following steps were made in order to construct a desirable compensator H(s):

$$H(s) = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \tag{4.33}$$

Step1: In the first row, the element (1,1) already has the lowest degree ($\delta = 2$)

Step 2: Subtract a multiple of the first column from the second to ensure $\delta(g_{12}) < \delta(g_{11})$.

$$\frac{0.025}{(1+0.1s)(1+0.2s)} = a\left(\frac{1}{(1+0.8s)(1+0.1s)}\right) + b$$

where:

$$a = \left(\frac{0.025}{1}\right)\left(\frac{(1+0.8s)}{(1+0.1s)}\right)\left(\frac{(1+0.1s)}{(1+0.2s)}\right) \text{ and } b = 0$$

Thus, the pre-compensator $H(s)$ was given as

$$H(s) = \begin{bmatrix} 1 & -a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{-0.025(1+0.8s)}{1+0.2s} \\ 0 & 1 \end{bmatrix} \tag{4.34}$$

According to the equation 4.29, the decoupled transfer function matrix corresponding to the nominal model of the smart hospital bed was obtained as follows:

$$P_{TDD}(s) = \begin{bmatrix} \frac{1}{0.08s^2+0.9s+1} & 0 \\ \frac{0.06}{0.01s^2+0.2s+1} & \frac{0.009s^2+0.15s+0.5085}{0.0013s^4+0.0345s^3+0.31s^2+1.05s+1} \end{bmatrix} \tag{4.35}$$

After checking the Nguyen *et al* conditions for the TDD property, the obtained transfer function matrix $P_{TDD}(s)$ is a TDD.

Next, the TDD property of the decoupled models corresponding to the upper-bounded model and the lower-bounded model was checked. To do this, their transfer functions $P_1(s)$ and $P_2(s)$ were given as follows:

$$P_1(s) = G_{lower}(s)H(s) \tag{4.36}$$

$$P_2(s) = G_{upper}(s)H(s) \tag{4.37}$$

$$P_1(s) = \begin{bmatrix} \frac{0.93}{0.0168s^2+0.31s+1} & \frac{-0.0006s^2-0.0134s-0.008}{0.0002s^4+0.0089s^3+0.1145s^2+0.58s+1} \\ \frac{0.048}{0.0024s^2+0.1s+1} & \frac{0.0013s^3+0.032s^2+0.183s+0.2788}{0.000007s^5+0.0006s^4+0.015s^3+0.1544s^2+0.67s+1} \end{bmatrix} \tag{4.38}$$

$$P_2(s) = \begin{bmatrix} \frac{1.17}{0.5005s^2+1.78s+1} & \frac{0.001s^3+0.0088s^2+0.0394s+0.0008}{0.009s^5+0.138s^4+0.8s^3+2.1543s^2+2.59s+1} \\ \frac{0.08}{0.07s^2+0.55s+1} & \frac{0.015s^3+0.193s^2+0.812s+1.088}{0.005s^5+0.091s^4+0.59s^3+1.759s^2+2.34s+1} \end{bmatrix} \tag{4.39}$$

It can be seen from the two equations 4.38 and 4.39, the off-diagonal elements (1,2) of both matrices $P_1(s)$ and $P_2(s)$ have very small values of 0.008 and 0.0008, respectively. In addition, their off-diagonal elements (2,1) are stable and proper. Therefore, the decoupled transfer function matrices of the upper bounded model and the lower bounded model can be approximated as two TDD matrices. This means that the smart hospital bed system under the influence of uncertainties can be successfully decoupled by utilising TDD decoupling techniques. Figure 4.5 shows the open-loop response of the decoupled hospital bed system.

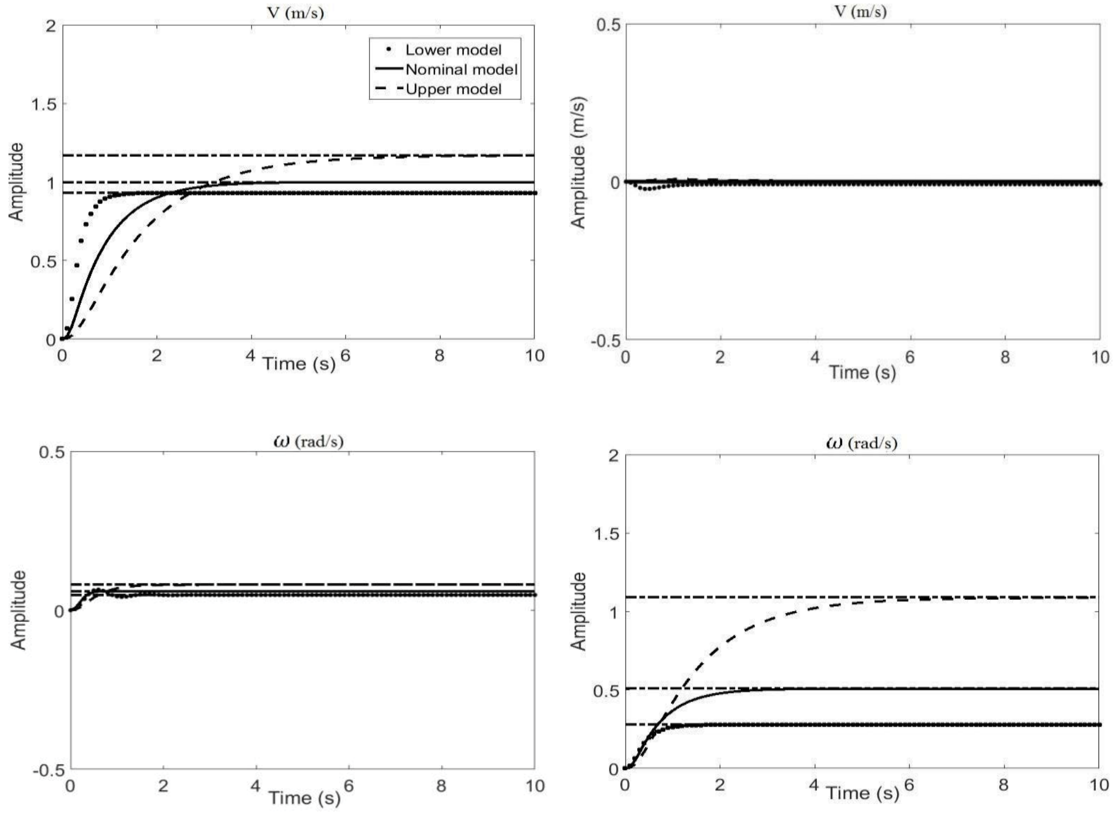


Figure 4.5: Decoupled models with TDD decoupling technique

In the transfer function matrices $P_{TDD}(s)$, $P_1(s)$ and $P_2(s)$, the elements (2,2) are still complicated for the closed-loop control design stage. In order to simplify the decoupled systems $P_{TDD}(s)$, $P_1(s)$ and $P_2(s)$ into the approximated form, a cancelling pole-zero pairs procedure with suitable tolerance was applied for their elements (2,2). The obtained results are shown as follows:

$$P_{TDD}(s) = \begin{bmatrix} \frac{1}{0.08s^2+0.9s+0.1} & 0 \\ \frac{0.06}{0.01s^2+0.2s+1} & \frac{0.047s+0.52}{0.006s^3+0.14s^2+0.85s+1} \end{bmatrix} \tag{4.40}$$

$$P_1(s) = \begin{bmatrix} \frac{0.93}{0.0168s^2+0.31s+1} & \frac{-0.0006s^2-0.0134s-0.008}{0.0002s^4+0.0089s^3+0.1145s^2+0.58s+1} \\ \frac{0.048}{0.0024s^2+0.1s+1} & \frac{0.01s+0.2788}{0.00084s^3+0.0358s^2+0.41s+1} \end{bmatrix} \tag{4.41}$$

$$P_2(s) = \begin{bmatrix} \frac{1.17}{0.5005s^2+1.78s+1} & \frac{0.001s^3+0.0088s^2+0.0394s+0.0008}{0.009s^5+0.138s^4+0.8s^3+2.1543s^2+2.59s+1} \\ \frac{0.08}{0.07s^2+0.55s+1} & \frac{4.35s+1.1}{0.0774s^3+0.705s^2+1.79s+1} \end{bmatrix} \tag{4.42}$$

4.4.1.2 *Stage 2: Optimal PID controllers*

After being decoupled by the pre-compensator $H(s)$, only diagonal elements of the compensated system were necessary for the closed-loop control design procedure. With the purpose of developing two OPID controllers for two subsystems $P_{TDD}(1,1)$ and $P_{TDD}(2,2)$, the nominal decoupled model of the smart hospital bed was adopted. For the smart hospital bed described in the previous chapter, the desired performance of the two subsystems was chosen as follows:

- Settling time $T_{set} \leq 4(s)$
- Rise time $T_{rise} \leq 2.5(s)$
- System overshoot $M_p \leq 10\%$
- System error converges to zero or acceptable tolerance

(a) Linear velocity subsystem

First, the Root Locus technique was utilised to obtain initial PID parameters. Figure 4.6 illustrates the Root Locus plots for the linear velocity subsystem.

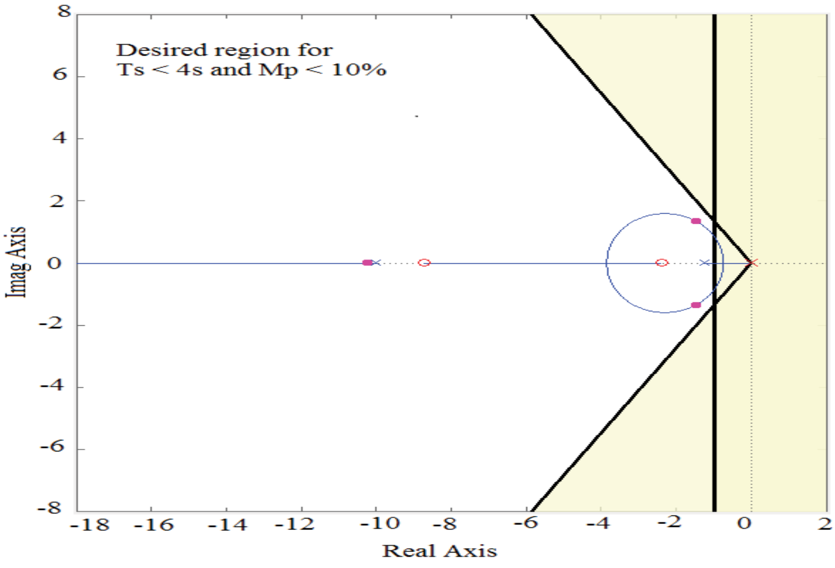


Figure 4.6: Root Locus plots for the linear velocity subsystem

It can be seen from Figure 4.6 that the white region is the desired area so that the design requirements of settling time and system overshoot are satisfactory. After extensive simulations, the initial PID parameters were chosen as $K_{Pv}^{init} = 1.7, K_{Iv}^{init} = 2.2, K_{Dv}^{init} = 0.52$. Applying the Least square optimisation, the optimal PID parameters were obtained as $K_{Pv}^{opt} = 1.5, K_{Iv}^{opt} = 1.6, K_{Dv}^{opt} = 0.1$. Figure 4.7 depicts step responses of controllers of initial and optimal parameters.

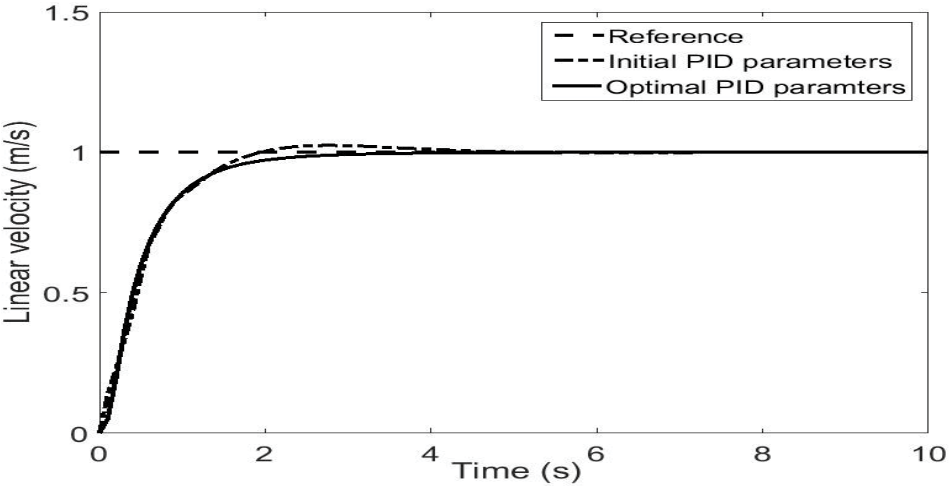


Figure 4.7: Closed-loop step responses of controllers with initial and optimal parameters for the linear velocity subsystem

(b) Angular velocity subsystem

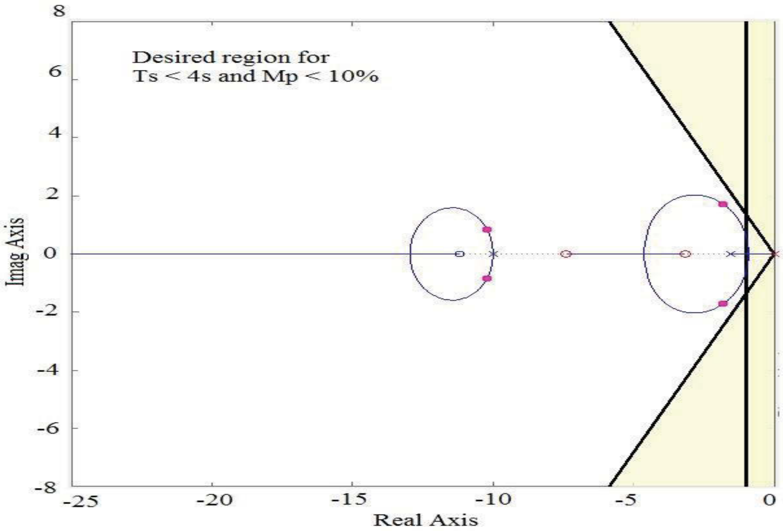


Figure 4.8: Root Locus plots for the angular velocity subsystem

Paralleling the controller design of the linear velocity subsystem, the Root Locus technique was also employed to obtain the initial PID parameters. Figure 4.8 illustrates the Root Locus plots for the angular velocity subsystem. The initial PID parameters were chosen as $K_{P\omega}^{init} = 2.05, K_{I\omega}^{init} = 3.1, K_{D\omega}^{init} = 0.8$. Through the Least square optimisation, the optimal PID parameters were derived: $K_{P\omega}^{opt} = 3.35, K_{I\omega}^{opt} =$

$3.8, K_{D\omega}^{opt} = 0.32$. A comparison between the step responses of controllers with initial and optimal PID parameters is shown in Figure 4.9.

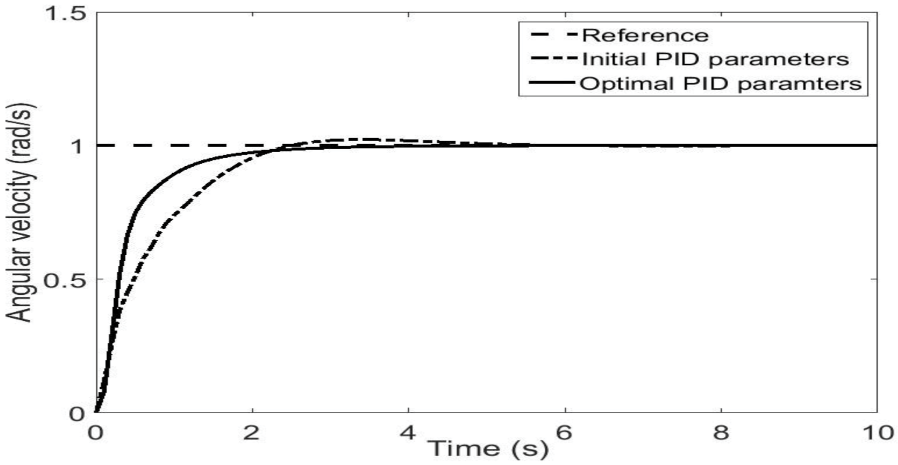


Figure 4.9: Closed-loop step responses of controllers with initial and optimal parameters for the angular velocity subsystem

4.4.2 Experimental results

In order to test the quality of the proposed control approach, real-time experiments were implemented on the smart hospital bed. The control algorithms were programmed in Labwindows 2010 with the Real-time module. The sampling time T_s was chosen as 0.1(s). There are three experiments which have been carried out in this section.

4.4.2.1 Experiment 1

The aim of this experiment is to compare the closed-loop performance of the controlled system with the open loop performance of the bed at the same reference step input. For the open loop, in the first case, the referent values of the linear velocity ($v_d = 1\ (m/s)$) and the angular velocity ($\omega_d = 0(rad/s)$) were directly introduced to the smart hospital bed. In the second case, the new values of the linear velocity ($v_d = 0\ (m/s)$) and angular velocity ($\omega_d = 1(rad/s)$) were applied on the bed system. The corresponding outputs were measured for 10 seconds. For the closed-loop, the desired values of the linear and angular velocities in the first case of the open loop were utilised as inputs of the OMPID controller. Outputs of this controller were introduced to the smart hospital bed. In the next step, this procedure was repeated with the controller's inputs which were the referent velocities in the second case of the open loop. System outputs were also measured for 10 seconds. In all cases, the smart hospital bed carried a 75kg person and moved on a granite surface, as demonstrated in Figure 4.10. Figure 4.11 illustrates the performance of the open-loop and closed-loop.

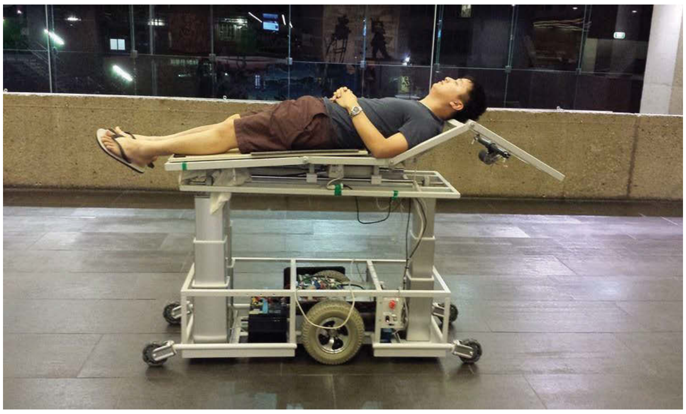


Figure 4.10: The smart hospital bed travels on the granite surface

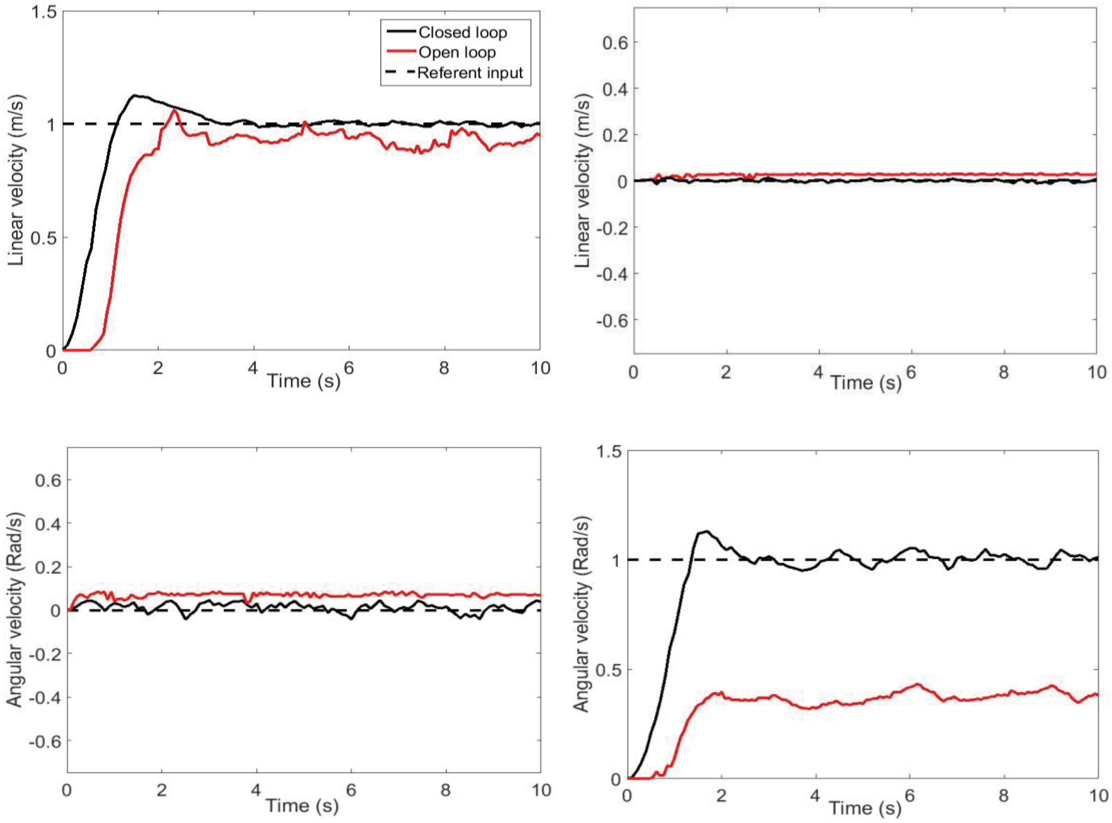


Figure 4.11: The open loop and closed-loop response of the hospital bed system with $v_d = 1\ (m/s)$; $\omega_d = 1\ (rad/s)$

4.4.2.2 Experiment 2

The second experiment was implemented to show the performance of the controlled system at various referent inputs when the smart hospital bed loaded with a 75kg person travelled on the granite surface. The referent value of linear velocity was firstly introduced while simultaneously keeping angular velocity reference at zero. Secondly, linear velocity reference was kept at zero while introducing a referent value of angular velocity. System outputs were then measured for both cases. Figures 4.12, 4.13, 4.14 and 4.15 present the actual system outputs with $\mathrm{v_d} = 0.4\ (\mathrm{m/s});\ \omega_\mathrm{d} = 0.4\ (\mathrm{rad/s})$, $\mathrm{v_d} = 0.6\ (\mathrm{m/s});\ \omega_\mathrm{d} = 0.6\ (\mathrm{rad/s})$, $\mathrm{v_d} = 0.8\ (\mathrm{m/s});\ \omega_\mathrm{d} = 0.8\ (\mathrm{rad/s})$ and $\mathrm{v_d} = 1\ (\mathrm{m/s});\ \omega_\mathrm{d} = 1\ (\mathrm{rad/s})$, respectively. Element (1,1) represents the response linear velocity loop whereas element (2,2) represents the response of the angular velocity loop. The coupling effects in the smart hospital bed dynamics are described as the responses of elements (1,2) and (2,1). Corresponding to each case of the desired velocities, the simulation performance of the proposed controller was plotted to compare it with the real performance of the OMPID controller.

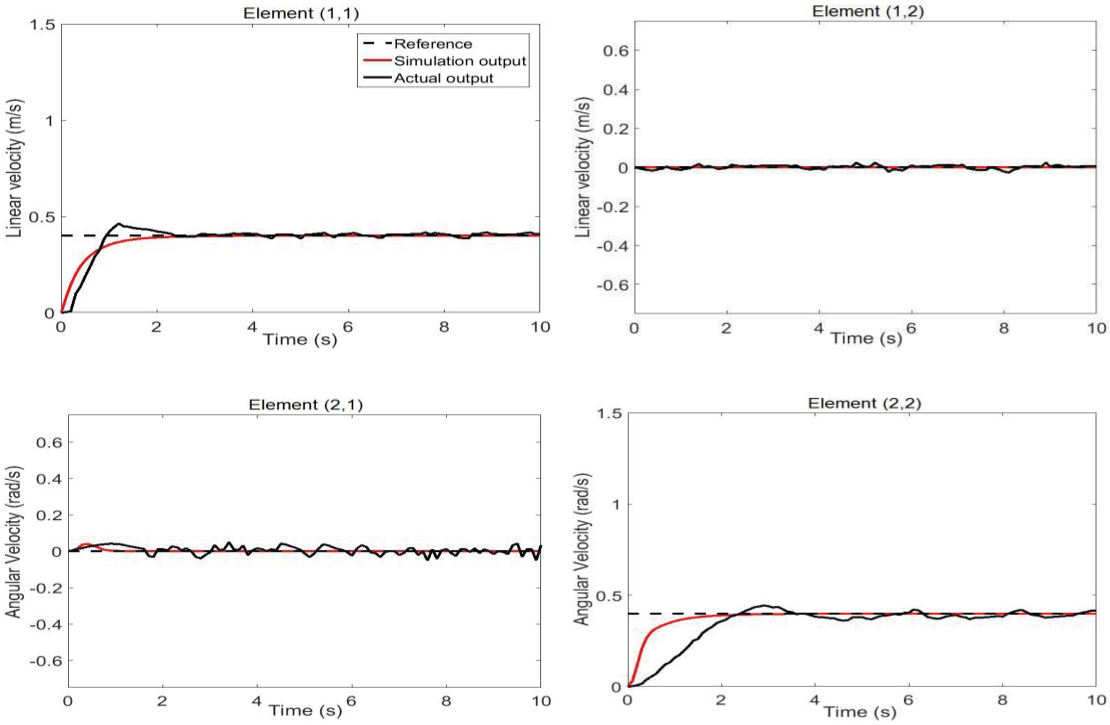


Figure 4.12: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $\boldsymbol{v_d = 0.4\ (m/s);\ \omega_d = 0.4\ (rad/s)}$

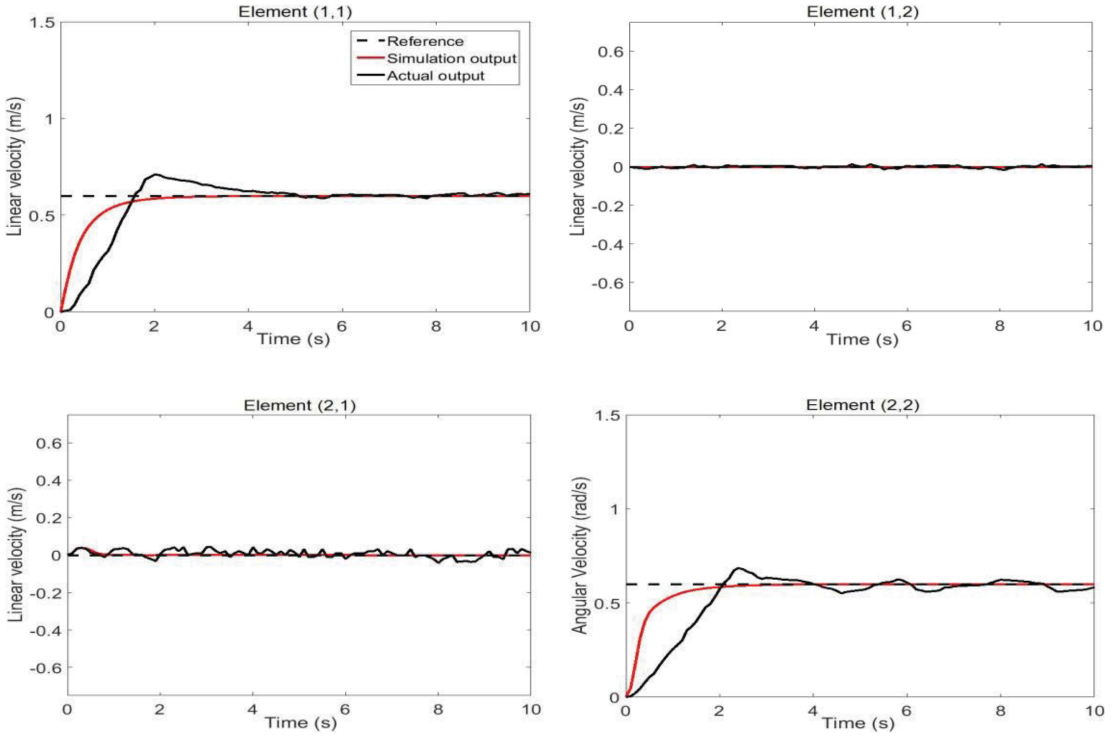


Figure 4.13: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 0.6\ (m/s);\ \omega_d = 0.6\ (rad/s)$

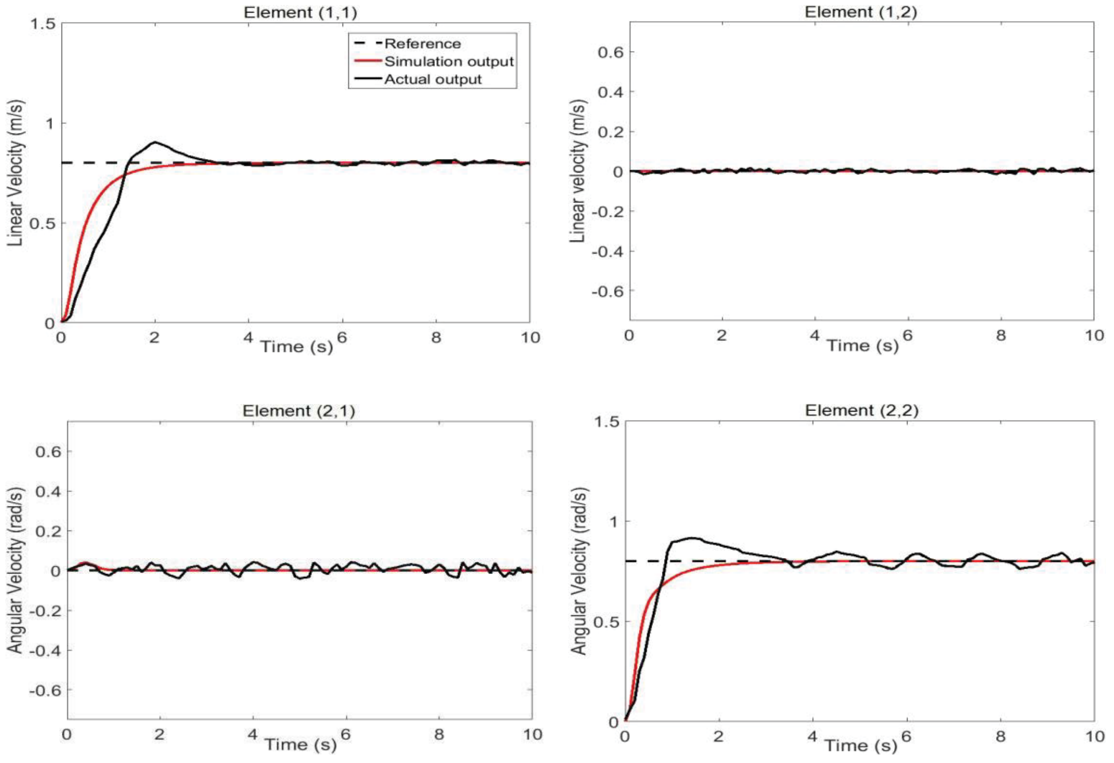


Figure 4.14: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 0.8\ (m/s);\ \omega_d = 0.8\ (rad/s)$

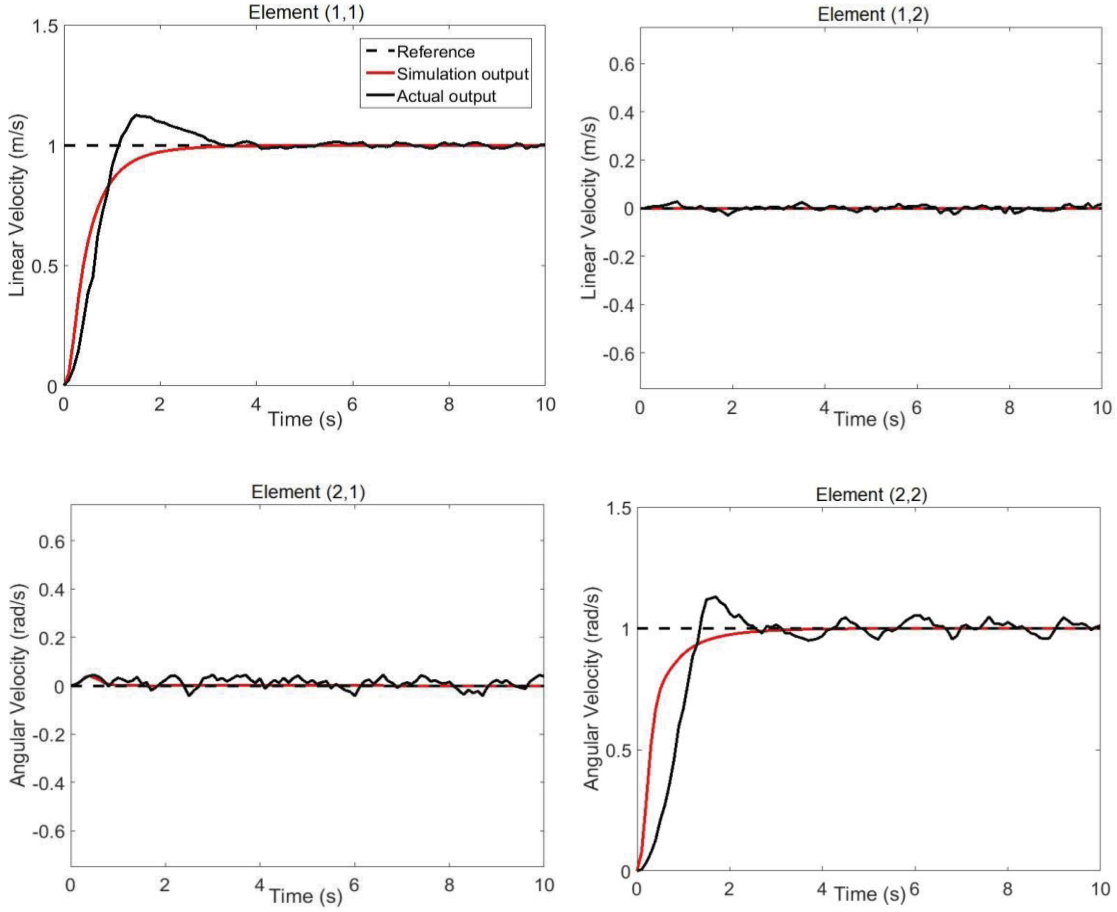


Figure 4.15: The closed-loop response of the hospital bed dynamics being controlled by OMPID control with $v_d = 1\ (m/s);\ \omega_d = 1\ (rad/s)$

Tables 4.1 and 4.2 present the output performance of the two velocity closed-loops when the bed system travels on the granite surface.

$v_d(m/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.6	2.1	15
0.6	1.6	3.3	14
0.8	1.7	2.7	12
1	1.8	2.8	13

Table 4.1: Output performance of the linear velocity subsystem (on the granite surface)

$\omega_d(rad/s)$	$T_{rise}\,(s)$	$T_{set}(s)$	$M_p(\%)$
0.4	2.1	3.4	11
0.6	2	3	17
0.8	1.4	2.6	15
1	1.5	2.3	13

Table 4.2: Output performance of the angular velocity subsystem (on the granite surface)

4.4.2.3 Experiment 3

The final experiment was conducted for estimating the bed performance on various surfaces while carrying an 85kg person. Two different environmental conditions including a carpet surface and a cement surface were chosen to implement this experiment. For each environment, the experimental procedure in experiment 2 was repeated to acquire the system outputs. Figure 4.16 illustrates the smart hospital bed travelling on the two different surfaces. Figure 4.17 demonstrates system outputs of the hospital bed when it travelled on the carpet surface in the cases of $v_d = 1\,(m/s);\ \omega_d = 0\,(rad/s)$ and $v_d = 0\,(m/s);\ \omega_d = 1\,(rad/s)$. After extensive experiments, tables 4.3 – 4.6 summarise the actual output performances of the linear velocity closed-loop and the angular velocity closed-loop when the bed system moved on different surfaces with various referent velocity inputs.

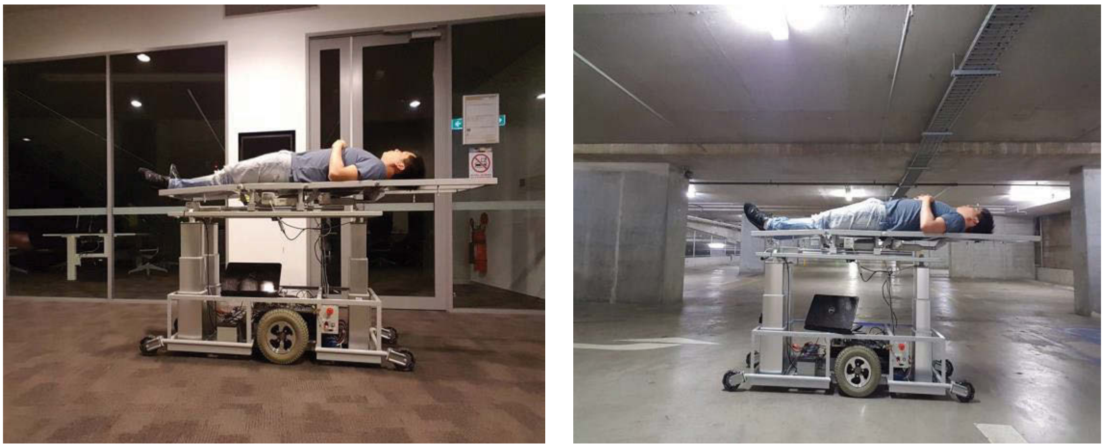


Figure 4.16: The smart hospital bed travels on various environmental conditions: left image: carpet surface, right image: cement surface

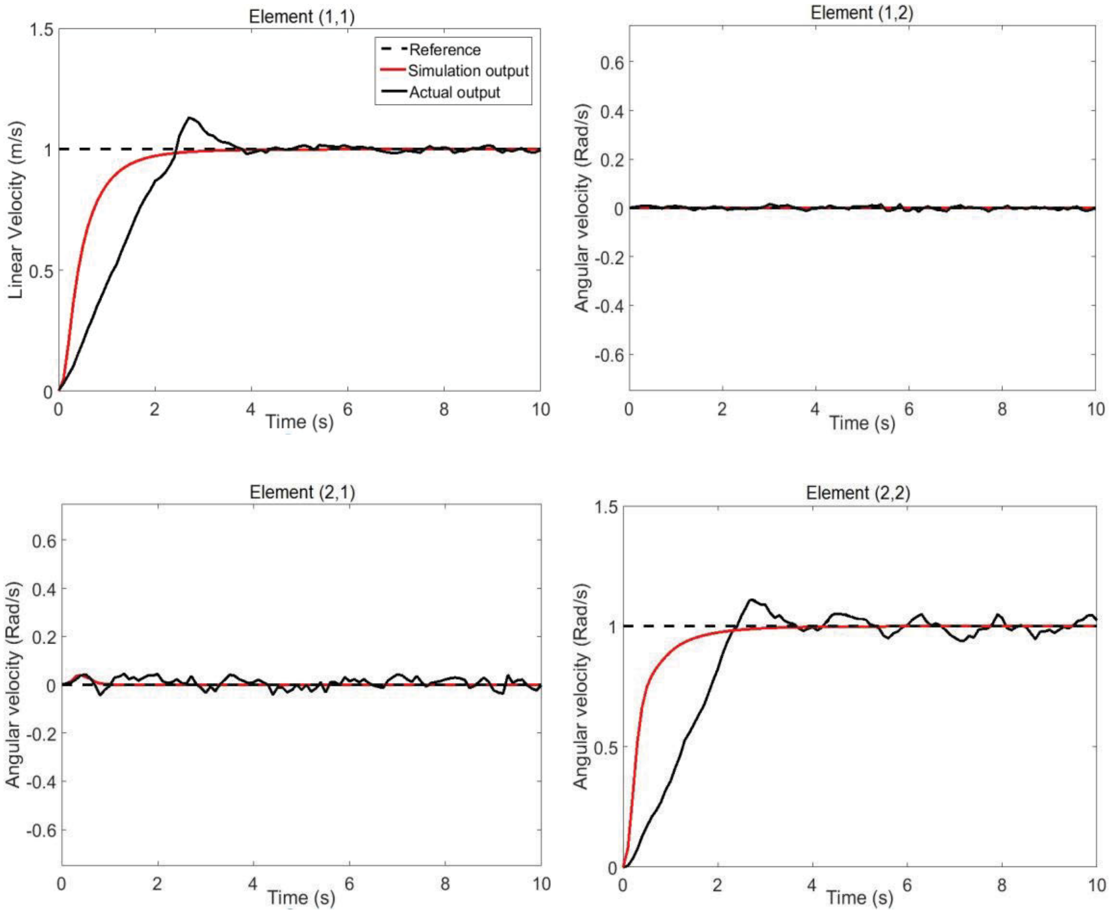


Figure 4.17: System outputs of the smart hospital bed being controlled by OMPID control with $\boldsymbol{v_d = 1\ (m/s);\ \omega_d = 1\ (rad/s)}$ in the case of travelling on the carpet surface

The hospital bed travelling on the carpet surface

$v_d(m/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.9	2.8	12
0.6	2.2	3.0	11
0.8	2.3	3.1	11
1	2.3	3.3	13

Table 4.3: Output performance of the linear velocity subsystem (on the carpet surface)

$\omega_d(rad/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	2.5	4.2	13
0.6	2.8	4.3	14
0.8	2.7	4.4	16
1	2.5	4.4	13

Table 4.4: Output performance of the angular velocity subsystem (on the carpet surface)

The hospital bed travelling on the cement surface

$v_d(m/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.6	2.3	13
0.6	1.6	2.9	17
0.8	1.7	2.8	16
1	1.8	2.8	18

Table 4.5: Output performance of the linear velocity subsystem (on the cement surface)

$\omega_d(rad/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	2.4	3.2	14
0.6	1.8	2.9	18
0.8	1.3	2.7	17
1	1.4	2.5	19

Table 4.6: Output performance of the angular velocity subsystem (on the cement surface)

4.5 DISCUSSION

In the first stage of the OMPID control design, the TDD decoupling technique is utilised to deal with the crossing effect between sub-systems in the smart hospital bed model. Based on the nominal dynamic model of the hospital bed system, the pre-compensator $H(s)$ is obtained. The graphs in Figure 4.5 show that the interaction between the system's inputs and outputs of the hospital bed is effectively reduced by the decoupler $H(s)$. Due to this, the multivariable problem of the smart hospital bed is simplified into two independent scalar problems.

For each independent scalar system, the Optimal PID control methodology is applied to develop an OPID controller. In order to obtain PID controllers, an exact model of the hospital bed system is required therefore its nominal model is adopted for the design stage. The initial parameters of the OPID are firstly obtained by the Root Locus technique then optimised by the Least Square optimisation algorithm to achieve the optimal performance which is verified through the simulation study.

The real-time experiments are conducted for estimating the effectiveness of the proposed OMPID approach. In the first experiment, the proposed control algorithm effectively reduces the interconnection between the control variables of the hospital bed system. The comparison results between the open-loop and closed-loop responses of the bed system show that the OMPID controller is successfully implemented on the smart hospital bed system since the actual velocities are almost centred around the set-point. However, large overshoot and oscillations of fairly significant amplitude can be seen in the output responses.

In the second experiment, the quality of the proposed controller is tested at various referent velocity inputs in both the simulation and real-time implementation. It can be seen from Figures 4.12 to 4.15 that the OMPID control approach achieves good performance for reducing the coupling effect. The obtained linear velocity controller enables the system error of the linear velocity subsystem to converge to zero in all cases. Conversely, the system error of the angular velocity subsystem fluctuates with significant amplitude around the set-point. From the results reported in the two tables 4.1 and 4.2, both the linear velocity controller and angular velocity controller still face the problem of large overshoot. One reason to explain this problem is that the development of the OMPID controller is based on the nominal model of the smart hospital bed. Since the smart hospital bed is a multivariable system with uncertainties, system parametric variations affect the tracking performance of the controlled system. This explanation is confirmed through the output responses of the bed system in the section 4.4.2.3.

In the final experiment, the smart hospital bed loaded with an 85kg person travelled in two different environmental conditions. The graphs in Figure 4.17 show that the coupling effect is decreased to acceptable levels and the performance of the controlled system is stable. However, the results summarised in tables 4.3 to 4.6 indicate that the overshoot is still more than expected. In the case of moving on the carpet, large overshoot, long rise time and settling time can be seen in output responses. For a more robust performance of the smart hospital bed, we will discuss a neural network control strategy in the next chapter.

CHAPTER 5 . INTELLIGENT MULTIVARIABLE CONTROL STRATEGIES

5.1 INTRODUCTION

In the previous chapter, the dynamic model of the smart hospital bed was obtained by approximating it as a linear dynamic model with uncertainties. The derived model shows that the hospital bed is a coupled multivariable system with two inputs and two outputs. An Optimal Multivariable PID Control approach was proposed and developed for the smart hospital bed. This approach is a combination of a TDD decoupling technique and an Optimal PID control design. Despite experimental results pointing out that the coupling effect problem is successfully reduced by the utilisation of the OMPID control, a large overshoot can be seen in system outputs. Moreover, system uncertainties affect the overall system when the hospital bed travels in different environmental conditions.

In this chapter, an advanced operator-following control strategy is introduced and developed for the smart hospital bed. The design integrates a neural network based operator detection algorithm, a PID based operator-following strategy and an intelligent multivariable low-level control method. First, the advanced operator-detection enables the smart hospital bed to identify the operator with a high accuracy. Then, two PID controllers are designed for generating referent velocities. Finally, the advanced low-level controller is developed to provide robust performance for the smart hospital bed under the effect of system uncertainties.

The proposed intelligent multivariable low-level control strategy is derived from a combination of a two-phase diagonal decoupling technique and an optimal neural network control design. In this combination, the decoupling technique is an extension of the Triangular Diagonal Dominance decoupling technique introduced in the previous chapter. Through the proposed diagonal decoupler, the hospital bed multivariable system is simplified into two independent scalar systems with uncertainties. Based on the advantages of the neural network, including the ability to map nonlinear functions and robustness in the presence of noise, an optimal neural network controller is developed for each independent scalar subsystem to deal with the parametric uncertainty problem of the hospital bed system.

The chapter is organised as follows. Firstly, the advanced operator-following control strategy for the smart hospital bed is presented in section 5.2. Then, section 5.3 describes an optimal multivariable neural network control algorithm in detail. In the next section, the controller design and various experiments are provided to estimate the performance of the proposed control approaches. Finally a discussion can be found in the last part of this chapter.

5.2 ADVANCED OPERATOR-FOLLOWING CONTROL STRATEGY

With the aim of supporting the smart hospital bed during patient transportation, an advanced operator-following control strategy is proposed. This strategy enables the smart hospital bed to follow an operator (such as a clinical staff member or an autonomous navigation robot) at a desired distance. In addition, the direction of the smart bed constantly faces the operator. The structure of the proposed control approach contains a neural network based operator detection algorithm, an operator-following strategy and an advanced low-level control method, as shown in Figure 5.1.

Recently, a laser range finder (LRF) capable of radially measuring the straight-line distance in one plane has been miniaturised so it can be mounted on the hospital bed easily. The directivity of the LRF is very strong and its accuracy and resolution are also high compared with the ultrasonic sensor and the infrared ray sensor. Good use has been made of these characteristics of the LRF, and many researchers apply LRF to operator detection and tracking operations. Despite various algorithms having been proposed to detect the target by using a LRF, none of the studies have been related to the neural network classification to identify the operator. Due to the neural network being equipped with so many desirable properties including experience based learning, the ability to generalise and the ability to map nonlinear functions, a neural network classifier is integrated into the proposed advanced control strategy.

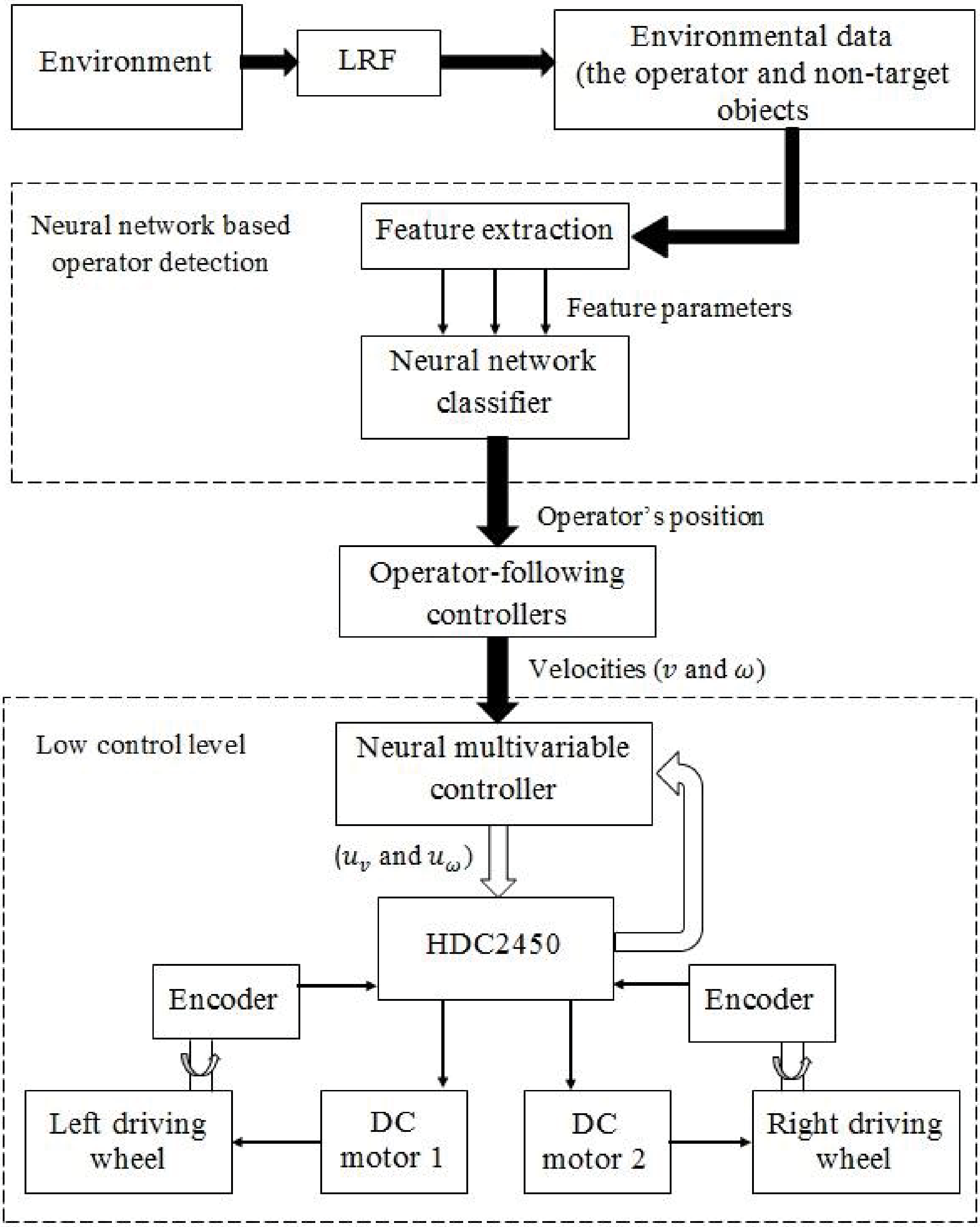

Environment
LRF
Environmental data (the operator and non-target objects
Neural network based operator detection
Feature extraction
Feature parameters
Neural network classifier
Operator's position
Operator-following controllers
Velocities (v and ω)
Low control level
Neural multivariable controller
(u_v and u_ω)
HDC2450
Encoder
Encoder
Left driving wheel
DC motor 1
DC motor 2
Right driving wheel

Figure 5.1: Advanced operator-following control strategy

5.2.1 Neural network based operator detection

In order to enable the neural network classifier to detect the operator, a feature extraction procedure is first implemented to remove redundant information from the laser data and then determine the feature parameters of the target and non-target objects which are represented by the remaining data. These feature parameters are utilised as inputs for the neural network classifier. The details of the proposed feature extraction procedure are described in section 3.3.

To apply the proposed neural network classifier in real-time implementation, it is first trained to obtain the optimal network structure and its optimal weights. In Chapter 3, four different training algorithms were introduced and implemented on the smart hospital bed. Experimental results in section 3.5.1 showed that the Levenberg-Marquardt training algorithm achieved the best performance when detecting the operator. As a result, the LM algorithm is chosen for the training process when developing the neural network classifier for the advanced operator-following control strategy. After extensive training, the optimal neural network structure and its optimal weights are derived from the best performance of the training process. In our study, the average result and best performance of the NN-LM based operator detection algorithm are presented in table 5.1 as follows.

NN-LM	**Training**		**Validation**		**Testing**	
	Sens (%)	Spec (%)	Sens (%)	Spec (%)	Sens (%)	Spec (%)
Average result	91.24	90.58	91.15	86.37	91.65	88.56
Best performance	93.45	92.18	92.43	88.64	92.41	89.8

Table 5.1: Results of NN-LM based operator detection algorithm

5.2.2 Operator-following strategy

Through the neural network based classifier, the operator is identified. Its position is employed for calculating the distance and angle errors between the smart bed and the operator. As discussed in section 3.4, our aim is to guarantee that the smart hospital bed follows the operator with a fixed desired distance and the direction of the smart bed constantly faces the operator. This means that the distance and angle errors converge to zero during the bed's operation. To achieve this, two PID controllers are employed and are shown as follows:

$$v(t) = K_{pv}e_d(t) + K_{iv} \int e_d(t) + K_{dv} \frac{de_d(t)}{dt} \tag{5.1}$$

$$\omega(t) = K_{p\omega}e_\varphi(t) + K_{i\omega} \int e_\varphi(t) + K_{d\omega} \frac{de_\varphi(t)}{dt} \tag{5.2}$$

where e_d and e_φ denote the distance error and angle error between the hospital bed and the operator, respectively; K_{pv}, K_{iv} and K_{dv} are gains of the linear velocity controller and $K_{p\omega}$, $K_{i\omega}$ and $K_{d\omega}$ are gains of the angular velocity controller. The details of the proposed operator-following control strategy are presented in Chapter 3 (section 3.4). After obtaining the linear velocity and angular velocity values from the two PID controllers, these velocities are utilised as the reference inputs of the low-level controller.

5.2.3 Neural multivariable low-level control

For other wheeled mobile systems such as wheeled mobile robots, or powered electrical wheelchairs, referent velocities obtained from the operator-following controller are often introduced directly to the mobile system since the researchers assume that dynamic factors do not significantly affect the mobile system. However, the smart hospital bed system is a highly nonlinear system therefore the assumption of neglecting system dynamics is impractical. In fact, there are various dynamic factors impacting on the system such as internal system uncertainties, environmental conditions (hard platform, glass, high friction way) and even the obvious case of a patient lying on the bed (especially with heavy patients, over 200 kilos). These factors affect the performance of the smart bed during its operation. Therefore, in order to enhance the hospital bed performance while following the operator, the reference velocities are introduced to a neural multivariable low-level controller. This advanced low-level controller can cope well with the uncertain dynamic parameters of the hospital bed system. The details of the intelligent multivariable low-level control algorithm are shown in the next section.

5.3 OPTIMAL MULTIVARIABLE NEURAL NETWORK CONTROL

The optimal multivariable neural network control approach is a combination of a two-phase diagonal decoupling technique and an optimal neural network control design. In the first design stage, the pre-compensators are utilised to fully eliminate the crossing effects between the control inputs and outputs of the multivariable system. After decoupling, the obtained model of the hospital bed has a diagonal structure. In addition, the hospital bed system is decomposed into two independent single variable systems consisting of the linear velocity (v) subsystem and angular velocity (ω) subsystem. Based on the work of Nguyen T. N. (2009), an optimal neural network controller is developed for each independent scalar subsystem. The optimal multivariable neural network control structure with two design stages is presented in Figure 5.2.

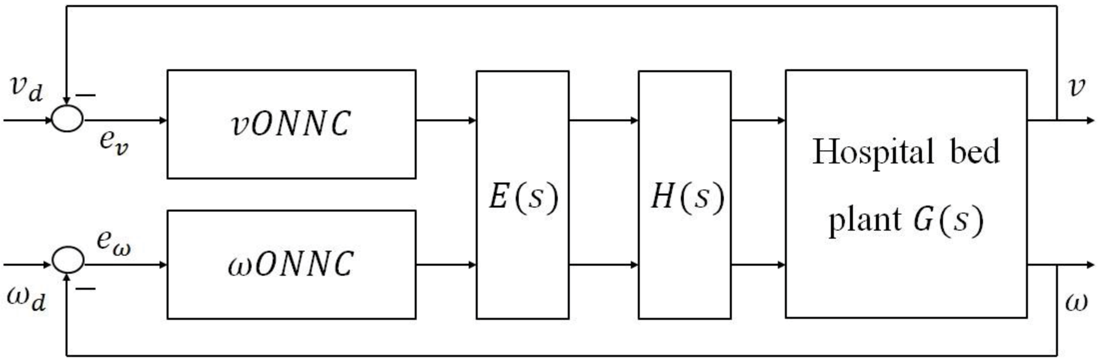


Figure 5.2: Optimal multivariable neural network control structure for a smart hospital bed

5.3.1 Stage 1: Two-phase diagonal decoupling technique

In theory, this technique aims to construct pre-compensators so that the compensated system has a diagonal structure. Firstly, the triangularisation phase takes responsibility for seeking the first pre-compensator so that the compensated system has a triangular structure with the TDD property. Then, the diagonalisation phase plays the role of searching for the second pre-compensator so that the compensated system has a diagonal structure. Details of the two-phase diagonal decoupling technique are described as follows:

Triangularisation phase

Consider a smart hospital bed whose transfer function matrix is given by a proper square $2x2$ plant $G(s)$. The aim of this phase is to find the first pre-compensator $H(s)$ so that the compensated plant $P_{TDD}(s) = G(s).H(s)$ has a triangular structure. This triangular structure is obtained with the use of the TDD decoupling technique in chapter 4.

Diagonalisation phase

In this phase, a theorem introduced by Nguyen *et al* (2014) is utilised to search the second compensator so that the compensated plant is diagonal.

Given the row-normalised matrix A(s) of the triangular matrix $P_{TDD}(s)$ as follows:

$$A(s) = \tilde{P}_{TDD}^{-1}(s)P_{TDD}(s) \tag{5.3}$$

where $\tilde{P}_{TDD}(s) = diag[t_{11}(s), t_{22}(s)]$

if the row-normalised matrix A(s) of the triangular matrix is stable then there exists a pre-compensator E(s) such that

$$P_{DD}(s) = G(s).H(s).E(s) \tag{5.4}$$

with $P_{DD}(s)$is proper and diagonal and E(s) is found as

$$E(s) = \begin{cases} A^{-1}(s), \ \ if\ A(s)\ is\ stable\ and\ proper \\ A^{-1}(s)X^{-1}(s), \ \ if\ A(s)\ stable\ and\ not\ proper \end{cases} \tag{5.5}$$

where $X^{-1}(s)$ is a diagonal polynomial matrix chosen so that $A^{-1}(s)X^{-1}(s)$ is stable and proper.

5.3.2 Stage 2: Optimal neural network control design

Through the two-phase diagonal decoupling technique, the smart hospital bed system is simplified to two independent scalar systems including a linear velocity subsystem and an angular velocity subsystem. For each independent scalar system, the optimal neural network control algorithm is applied to let the output track the desired reference input. The optimal neural network control scheme for each velocity subsystem is presented in Figure 5.3.

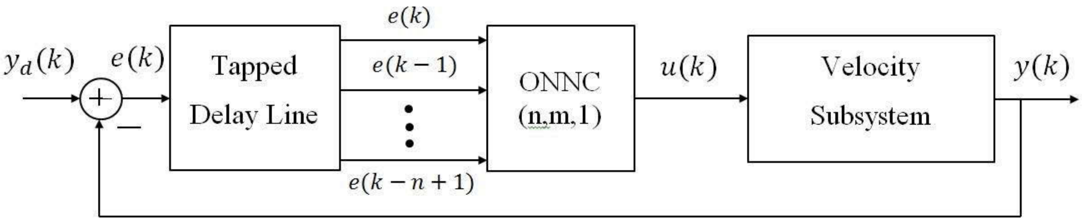


Figure 5.3: Optimal neural network control structure

One way to design the neural network controller is through utilisation of the state space model of the velocity subsystem. After being decoupled, the velocity subsystem remains nonlinear. Therefore, its state space model is expressed as follows:

$$\begin{cases} x(k+1) = (A + \Delta A)x(k) + (B + \Delta B)u(k) \\ y(k) = Cx(k) \\ x(0) = x_0 \end{cases} \tag{5.6}$$

where:

- $x = [x_1, \dots, x_n]^T$ denotes the system state.
- u represents the control input.
- A, B, C denotes the system matrix, the input matrix and the output matrix, respectively.
- ΔA and ΔB are the uncertainty matrices.

Assume that ΔA and ΔB are bounded, the discrete dynamic state space model (5.6) of the velocity subsystem can be represented by three state space models including a nominal state space model, an upper bounded state space model and a lower bounded state space model, shown as follows:

$$\begin{cases} x(k+1) = A_{nom}x(k) + B_{nom}u(k) \\ y(k) = Cx(k), x(0) = x_0 \end{cases} \tag{5.7}$$

$$\begin{cases} x(k+1) = A_{low}x(k) + B_{low}u(k) \\ y(k) = Cx(k), x(0) = x_0 \end{cases} \tag{5.8}$$

$$\begin{cases} x(k+1) = A_{up}x(k) + B_{up}u(k) \\ y(k) = Cx(k), x(0) = x_0 \end{cases} \tag{5.9}$$

In the proposed control scheme, an optimal neural network controller is a feed-forward multiplayer neural network with a structure including one input layer, one hidden layer and one output layer. The structure of the optimal neural network controller is presented in Figure 5.4.

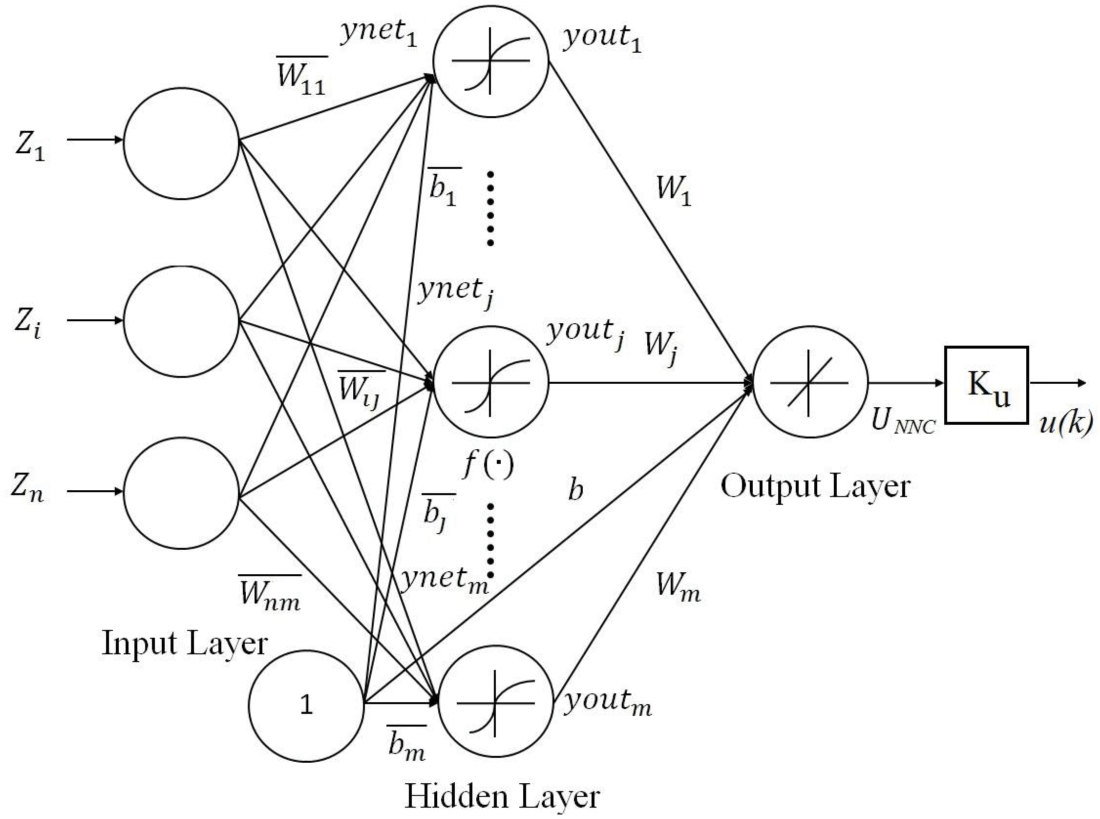


Figure 5.4: Neural network control structure

Computation of output control

It can be seen from Figure 5.4, the output of the neural network controller at time (k) is computed as follows:

$$ynet_j(k) = \sum_1^n \overline{W_{ji}}(k)\, Z_i(k) + \overline{b_j}(k) \tag{5.10}$$

$$yout_j(k) = f(ynet_j(k)) \tag{5.11}$$

$$U_{NNC}(k) = \sum_1^m yout_j(k)W_j(k) + b(k) \tag{5.12}$$

where:

- Z_i is $i-th$ node of the input layer
- $\overline{W_{ij}}$ is the weight between $i-th$ node of input layer and $j-th$ node of a hidden layer.
- $\overline{b_j}$ is the bias of $j-th$ node of the hidden layer.
- W_j is the weight between $j-th$ node of the hidden layer and the output node.
- b is the bias of the output node.
- n, m represent the number of the input and hidden nodes, respectively.
- k denotes the discrete time index.
- $f(.)$ is the activation function of hidden layer, as shown in the equation below:

$$f(z) = \frac{1- e^{-z}}{1+ e^{-z}} \tag{5.13}$$

Weight updating rules

The system error is given by:

$$e(k+1) = y_d(k+1) - y(k+1) \tag{5.14}$$

The control law is designed to enable the system error $e(k+1)$ to converge to zero, or at least to an acceptable tolerance with a desired finite time of T_0. To do this, firstly, the cost function of the neural network controller is defined as follows:

$$E = \frac{1}{2}(e(k+1))^2 = \frac{1}{2}\big(y_d(k+1) - y(k+1)\big)^2 \tag{5.15}$$

Subsequently, in order to minimise the cost function *E*, it is necessary to change the weights of the neural network controller to the direction of a negative gradient. Utilising the Gradient Descent algorithm, the resulting functions are:

$$\Delta W_j(k) = -\gamma \frac{\partial E}{\partial W_j(k)} = -\gamma \frac{\partial E}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial U_{NNC}(k)} \frac{\partial U_{NNC}(k)}{\partial W_j(k)} \tag{5.16}$$

$$\Delta b(k) = -\gamma \frac{\partial E}{\partial b(k)} = -\gamma \frac{\partial E}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial U_{NNC}(k)} \frac{\partial U_{NNC}(k)}{\partial b(k)} \tag{5.17}$$

$$\Delta \overline{W_{ij}}(k) = -\varepsilon \frac{\partial E}{\partial \overline{W_{ij}}(k)} = -\varepsilon \frac{\partial E}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial U_{NNC}(k)} \frac{\partial U_{NNC}(k)}{\partial \overline{W_{ij}}(k)} \tag{5.18}$$

$$\Delta \overline{b_j}(k) = -\varepsilon \frac{\partial E}{\partial \overline{b_j}(k)} = -\varepsilon \frac{\partial E}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial U_{NNC}(k)} \frac{\partial U_{NNC}(k)}{\partial \overline{b_j}(k)} \tag{5.19}$$

where γ and ε are learning rates of the output layer and hidden layer weights, respectively.

For a given neural network control structure, the last term of the equations (5.16) to (5.19) can be easily calculated by the following equations:

$$\frac{\partial U_{NNC}(k)}{\partial W_j(k)} = \left(\sum_{j=1}^{m} W_j\, f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) + b\right) f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) \tag{5.20}$$

$$\frac{\partial U_{NNC}(k)}{\partial b(k)} = \sum_{j=1}^{m} W_j\, f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) + b \tag{5.21}$$

$$\frac{\partial U_{NNC}(k)}{\partial \overline{W_{ij}}(k)} = \left(\sum_{j=1}^{m} W_j\, f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) + b\right) f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) W_j Z_i \tag{5.22}$$

$$\frac{\partial U_{NNC}(k)}{\partial \bar{b}_j(k)} = \left(\sum_{j=1}^{m} W_j\, f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) + b\right) f\left(\sum_{i=1}^{n} Z_i \overline{W_{ij}} + \bar{b}_j\right) W_j \tag{5.23}$$

The third term in equations 5.16 – 5.19 is designated as the Jacobian of the velocity subsystem, shown as follows:

$$J(k) = \frac{\partial y(k+1)}{\partial u(k)} \tag{5.24}$$

Utilising the state space equation 5.4 and applying the following chain, it is then easy to calculate the plant Jacobian directly from the velocity subsystem.

$$J(k) = \frac{\partial y(k+1)}{\partial u(k)} = \frac{\partial y(k+1)}{\partial x(k+1)} \frac{\partial x(k+1)}{\partial u(k)} = C(B + \Delta B) \tag{5.25}$$

The second and the fourth terms in equations 5.16 – 5.19 can be directly calculated as follows:

$$\frac{\partial E}{\partial y(k+1)} = -e(k+1) \tag{5.26}$$

$$\frac{\partial u(k)}{\partial U_{NNC}(k)} = K_u \tag{5.27}$$

Finally, the updating rules for the neural network controller are:

$$W_j(k+1) = W_j(k) + \Delta W_j(k) \tag{5.28}$$

$$b(k+1) = b(k) + \Delta b(k) \tag{5.29}$$

$$\overline{W_{ij}}(k) = \overline{W_{ij}}(k) + \Delta \overline{W_{ij}}(k) \tag{5.30}$$

$$\overline{b_J}(k+1) = \overline{b_J}(k) + \Delta\overline{b_J}(k) \tag{5.31}$$

In these equations, the terms $\Delta W_j(k)$, $\Delta b(k)$, $\Delta\overline{W_{\iota J}}(k)$, $\Delta\overline{b_J}(k)$ are derived from equations 5.16 – 5.27.

Training algorithm

- **Step 1: Initialisation**
  - All weights of the intelligent controller consisting of $\overline{W_{\iota J}}$, W_j, $\overline{b_J}$, b are initialised to very small values.
  - The learning rates ε, γ are chosen at small positive values.
  - Set the iteration number $k = 0$.
  - Set *E(0)* so that it is greater than the maximum tolerable error E_{max} this being set to a relatively small value.
  - Set the maximum iteration k_{max}
  - Set the maximum tolerable error E_{max}
- **Step 2: Forward propagation**
  - Find error value $e(k+1)$ between the referent input $y_d(k+1)$ and the output response $y(k+1)$ of the velocity subsystem.
  - Set inputs for the neural network by passing the system error through the tapped delay line block.
  - Calculate the control signal using the following equation:

    $$u(k) = K_u U_{NNC}(k) = K_u(\textstyle\sum_1^m yout_j(k) W_j(k) + b(k))$$

    with:

    $$yout_j(k) = f(ynet_j(k))$$

    $$ynet_j(k) = \textstyle\sum_1^n \overline{W_{J\iota}}(k)\, Z_i(k) + \overline{b_J}(k)$$

  - Apply the obtained control signal to the velocity subsystem.
- **Step 3: Backward propagation**
  - Utilise equation 5.15 to find the cost function value E.
  - Update new weights of the neural network for the next iteration using the equations from 5.16 to 5.31.
- **Step 4: Stopping criteria**

- The training process will stop and go to step 5 if $\begin{bmatrix} l & \geq & l_{max} \\ E & \leq & E_{max} \end{bmatrix}$
- Otherwise, $l = l + 1$ and go to step 2.

- **Step 5: Finish training**

- Stop training and save the optimal weights of the neural network at the last iteration.

5.4 RESULTS

To validate the proposed intelligent control approaches, the neural multivariable controller was firstly designed for the smart hospital bed. Accordingly, various experiments were implemented in real-time to estimate the efficiency of the obtained intelligent controller and the advanced operator-following control strategy.

5.4.1 OMNN controller design

5.4.1.1 Stage 1: Two-phase diagonal decoupler

In order to design pre-compensators so that the compensated model has a diagonal structure, the nominal model of the smart hospital bed was utilised. According to the methodology introduced in section 5.3, two phases consisting of a triangularisation phase and diagonalisation phase were required in the diagonal decoupling design.

Triangularisation phase

Chapter 4 provided full details of the TDD decoupling procedure for the hospital bed system. The obtained TDD decoupler was given by:

$$H(s) = \begin{bmatrix} 1 & \frac{-0.025(1+0.8s)}{1+0.2s} \\ 0 & 1 \end{bmatrix} \tag{5.28}$$

The compensated plant P(s) was calculated as follows:

$$P_{TDD}(s) = G_{nom}(s)H(s) \tag{5.29}$$

$$P(s) = \begin{bmatrix} \frac{1}{0.08s^2+0.9s+0.1} & 0 \\ \frac{0.06}{0.01s^2+0.2s+1} & \frac{0.047s+0.52}{0.006s^3+0.14s^2+0.85s+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{(1+0.8s)(1+0.1s)} & 0 \\ \frac{0.06}{(1+0.1s)^2} & \frac{7.25(s+11.17)}{(s+1.54)(s+10)^2} \end{bmatrix} \tag{5.30}$$

Diagonalisation phase

$A(s) = \tilde{P}^{-1}(s)P(s) =$

$$\begin{bmatrix} \frac{1}{(1+0.8s)(1+0.1s)} & 0 \\ 0 & \frac{7.25(s+11.17)}{(s+1.54)(s+10)^2} \end{bmatrix}^{-1} \times \begin{bmatrix} \frac{1}{(1+0.8s)(1+0.1s)} & 0 \\ \frac{0.06}{(1+0.1s)^2} & \frac{7.25(s+11.17)}{(s+1.54)(s+10)^2} \end{bmatrix} \tag{5.31}$$

$$A(s) = \begin{bmatrix} 1 & 0 \\ \frac{0.827(s+1.54)}{s+11.17} & 1 \end{bmatrix} \tag{5.32}$$

Clearly, $A(s)$ is stable and proper, therefore the pre-compensator E(s) was obtained as follows:

$$E(s) = A(s)^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{0.827(s+1.54)}{s+11.17} & 1 \end{bmatrix} \tag{5.33}$$

As a result, the nominal compensated model for the smart hospital bed was depicted as follows:

$$P_{DD}(s) = P_{TDD}(s).E(s) \tag{5.34}$$

$$P_{DD}(s) = \begin{bmatrix} \frac{1}{(1+0.8s)(1+0.1s)} & 0 \\ 0 & \frac{7.25(s+11.17)}{(s+1.54)(s+10)^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{0.08s^2+0.9s+0.1} & 0 \\ 0 & \frac{0.047s+0.52}{0.006s^3+0.14s^2+0.85s+1} \end{bmatrix} \tag{5.35}$$

Therefore, by using the two pre-compensators H(s) and E(s), the compensated model has a diagonal structure.

Next, the DD property of the decoupled upper-bounded model and decoupled lower-bounded model was checked. To do this, their transfer functions $P_{1DD}(s)$ and $P_{2DD}(s)$ were presented as follows:

$$P_{1DD}(s) = P_1(s)E(s) \tag{5.36}$$

$$P_{2DD}(s) \; = \; P_2(s)E(s) \tag{5.37}$$

$P_{1DD}(s) =$

$$\begin{bmatrix} \dfrac{0.936}{0.0168s^2+0.311s+1} & \dfrac{-0.0006s^2-0.0134s-0.008}{0.0002s^4+0.0089s^3+0.1145s^2+0.58s+1} \\ \dfrac{0.0002s^4+0.0036s^3+0.045s^2-0.0015s-0.005}{0.023s^6+0.0072s^5+0.264s^4+0.86+1.45s^2+2.36s+1} & \dfrac{0.01s+0.2788}{0.00084s^3+0.0358s^2+0.41s+1} \end{bmatrix} \tag{5.38}$$

$P_{2DD}(s) =$

$$\begin{bmatrix} \dfrac{1.168}{0.5006s+1.785s+1} & \dfrac{0.0009s^3+0.0088s^2+0.0181s+0.0008}{0.009s^5+0.138s^4+0.8s^3+2.1543s^2+2.59s+1} \\ \dfrac{0.0062s^4+0.047s^3+0.038s^2+0.024+0.0009}{0.047s^6+0.0058s^5+0.157s^4+0.72+2.56s^2+1.42s+1} & \dfrac{4.35s+1.1}{0.0774s^3+0.705s^2+1.79s+1} \end{bmatrix} \tag{5.39}$$

The step response of the decoupled system $P_{DD}(s)$, $P_{1DD}(s)$ and $P_{2DD}(s)$ were presented in Figures 5.5.

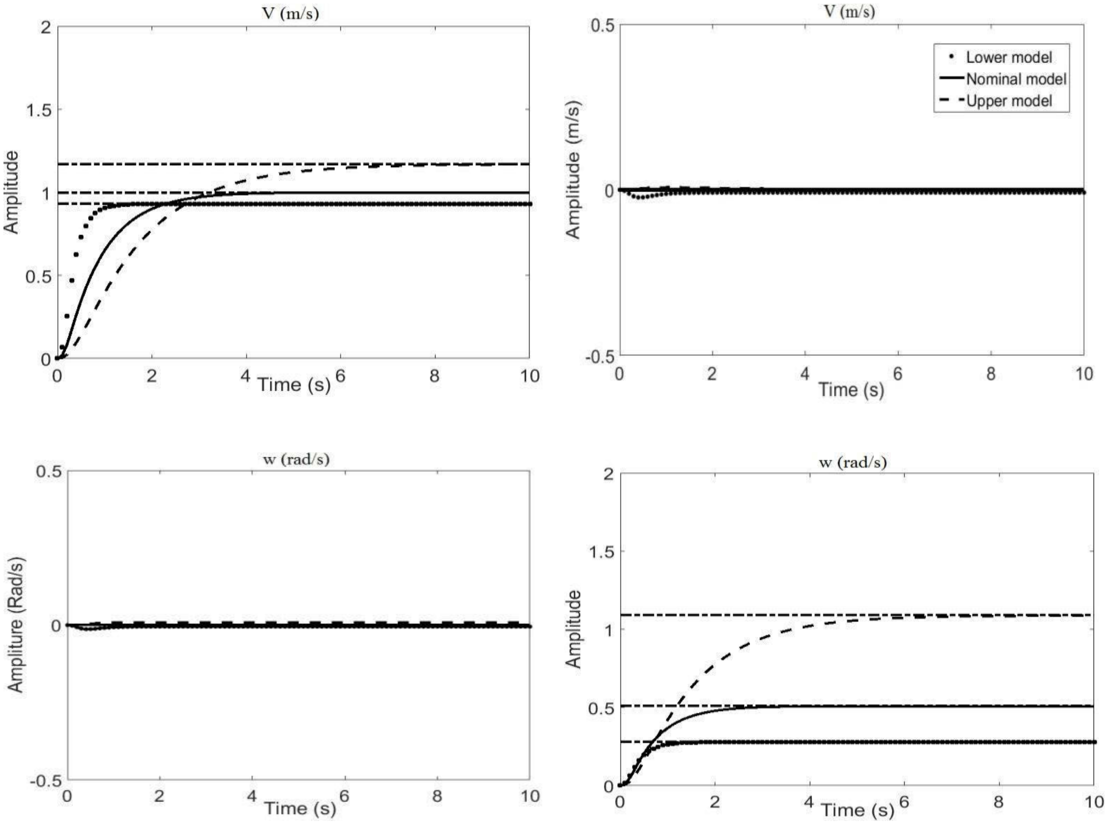


Figure 5.5: Opened-loop of the decoupled transfer matrices $P_{DD}(s)$, $P_{1DD}(s)$ and $P_{2DD}(s)$ using the diagonal decoupling technique

It can be seen from the two equations 5.38 and 5.39 that the off-diagonal elements of both transfer functions have very small values of 0.005 and 0.0009. Moreover, the off-diagonal transfer functions are stable and appropriate. As a result, the two pre-compensators H(s) and E(s) reduce the multivariable control problems into two independent scalar problems.

5.4.1.2 Stage 2: Optimal neural network controllers

Two neural controllers being independently designed for the linear velocity closed-loop and the angular velocity closed-loop are denoted as $ONNC_v$ and $ONNC_\omega$, respectively. The desired performance of the two subsystems was chosen as follows:

- Settling time $T_{set} \leq 4(s)$
- Rise time $T_{rise} \leq 2.5(s)$
- System overshoot $M_p \leq 5\%$
- System error converges to zero or acceptable tolerance

$ONNC_v$ for the linear velocity subsystem

The element (1,1) of the decoupled hospital bed dynamic model represented the transfer function of the linear velocity subsystem. Since the dynamic model of the smart hospital bed was expressed by three models including a lower bounded model, a nominal model and an upper bounded model, the decoupled dynamic model was also represented by three corresponding decoupled models $P_{DD}(s)$, $P_{1DD}(s)$ and $P_{2DD}(s)$. Therefore, three transfer functions for the linear velocity subsystem were given by:

$$P_{DD}(1,1) = \frac{1}{0.08s^2+0.9s+1} \tag{5.40}$$

$$P_{1DD}(1,1) = \frac{0.936}{0.0168s^2+0.311s+1} \tag{5.41}$$

$$P_{2DD}(1,1) = \frac{1.168}{0.5006s^2+1.785s+1} \tag{5.42}$$

In order to design the neural network controller for the linear velocity subsystem, its transfer functions were transformed into the discrete time dynamic state space with a sampling time Ts = 0.1 (s). Equations 5.43, 5.44 and 5.45 represented the nominal transfer function, lower bounded function and upper bounded function of the linear velocity subsystem in a discrete state space form, respectively.

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & -0.3247 \\ 1 & 1.25 \end{bmatrix} x(k) + \begin{bmatrix} 0.03029 \\ 0.004399 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) \end{cases} \quad (5.43)$$

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & -0.7007 \\ 1 & 1.684 \end{bmatrix} x(k) + \begin{bmatrix} 0.009238 \\ 0.00104 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) \end{cases} \quad (5.44)$$

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & -0.158 \\ 1 & 0.8989 \end{bmatrix} x(k) + \begin{bmatrix} 0.08528 \\ 0.01572 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) \end{cases} \quad (5.45)$$

According to the equations 5.24 and 5.25, the plant Jacobian calculations for the three equivalent models were obtained as follows:

$$J_v(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.0043$$

$$J_{1v}(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.00104 \quad (5.46)$$

$$J_{2v}(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.01572$$

After doing extensive simulations and experiments, the optimal neural network structure for the $ONNC_v$ of the linear velocity subsystem was (3,5,1) which corresponded to three input nodes, five hidden nodes and one output node. The training procedure described in section 5.3.2 was utilised for training the proposed neural network controller. During the training process, the learning rates γ and ε were chosen as relatively small. These optimal values were chosen as $\gamma = 0.01$ and ε = 0.005. The constant K_u was chosen as 2.

In order to improve the generalisation property of the neural network controller, three decoupled models of the linear velocity subsystem were alternatively utilised in the training procedure. Additionally, an extensive number of training epochs with various values of the input reference were conducted. The training epoch consisted of 100 iterations, equivalent to 10 seconds with sampling time Ts = 0.1s.

After the training procedure, optimal weights were obtained as follows for the neural network controller:

$$\overline{vW_{ji}} = \begin{bmatrix} 0.0472 & 0.1320 & 0.0436 \\ 0.1028 & -0.0213 & 0.1236 \\ 0.0303 & -0.0204 & 0.0589 \\ 0.2311 & 0.0236 & 0.0407 \\ 0.1151 & 0.0308 & -0.0714 \end{bmatrix} \quad \overline{vb_j} = \begin{bmatrix} 0.0614 \\ -0.3004 \\ -0.1404 \\ 0.5003 \end{bmatrix} \tag{5.47}$$

$$vW_j = [0.0546 \quad 0.0238 \quad 0.1412 \quad -0.0357] \quad vb = [-0.1486]$$

where $\overline{vW_{ij}}$, vW_j, $v\overline{b}_j$, vb are weights and biases for the hidden-output and input-hidden layers of the $ONNC_v$.

In order to estimate the performance of the obtained neural network controller $ONNC_v$, the step reference input was introduced. Figure 5.6 shows the output responses of three linear velocity closed-loop systems equivalent to the nominal model, lower bounded model and upper bounded model.

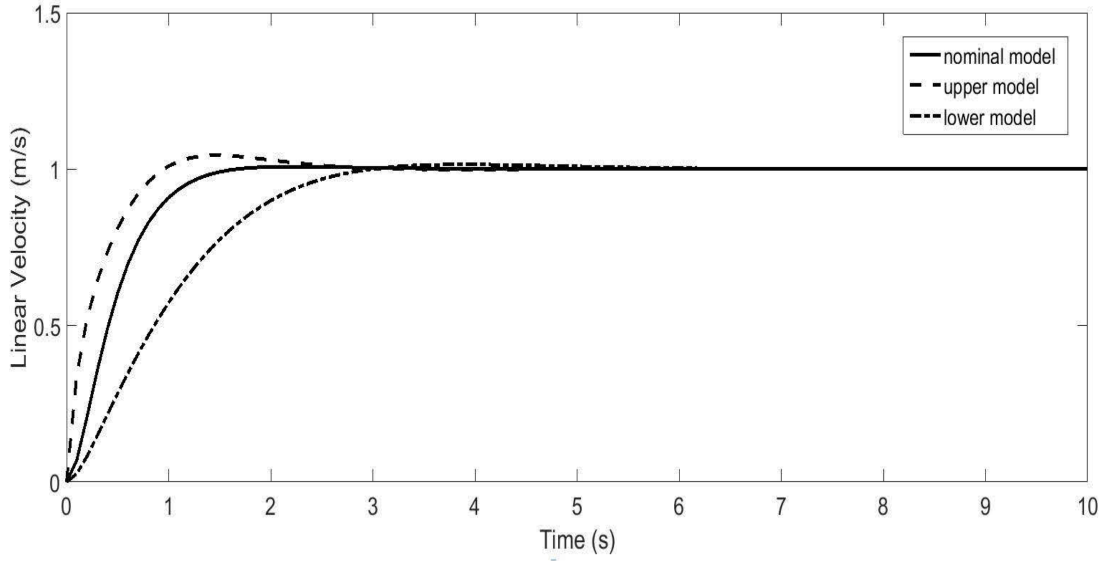


Figure 5.6: System outputs of three representational sub-systems P_{DD}(1,1), P_{1DD}(1,1) and P_{2DD}(1,1) these being controlled by $ONNC_v$

***$ONNC_\omega$* for the angular velocity subsystem**

The transfer function of the angular velocity subsystem is expressed by the element (2,2) of the decoupled hospital bed dynamic model. Similar to the linear velocity subsystem, the three transfer functions representing the angular velocity subsystems were as follows:

$$P_{DD}(2,2) = \frac{0.047s+0.52}{0.006s^3+0.14s^2+0.85s+1} \tag{5.48}$$

$$P_{1DD}(2,2) = \frac{0.01s+0.2788}{0.00084s^3+0.0358s^2+0.41s+1} \tag{5.49}$$

$$P_{2DD}(2,2) = \frac{4.35s+1.1}{0.0774s^3+0.705s^2+1.79s+1} \tag{5.50}$$

Also utilising the discrete state space model with a sampling time of Ts = 0.1 (s) to design the neural network controller, three corresponding discrete state space models of the angular velocity subsystem were shown as follows:

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 & 0.116 \\ 1 & 0 & -0.7661 \\ 0 & 1 & 1.593 \end{bmatrix} x(k) + \begin{bmatrix} -0.006048 \\ 0.0099 \\ 0.002691 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(k) \end{cases} \tag{5.51}$$

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 & 0.0141 \\ 1 & 0 & -0.2502 \\ 0 & 1 & 1.038 \end{bmatrix} x(k) + \begin{bmatrix} -0.001326 \\ 0.01974 \\ 0.001674 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(k) \end{cases} \tag{5.52}$$

$$\begin{cases} x(k+1) = \begin{bmatrix} 0 & 0 & 0.4022 \\ 1 & 0 & -1.659 \\ 0 & 1 & 2.248 \end{bmatrix} x(k) + \begin{bmatrix} -0.1525 \\ 0.04847 \\ 0.02101 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x(k) \end{cases} \tag{5.53}$$

In accordance with the equations 5.24 and 5.25, the plant Jacobian for the three corresponding models was calculated as follows:

$$J_\omega(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.00261$$

$$J_{1\omega}(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.001674 \tag{5.54}$$

$$J_{2\omega}(k) = \frac{\partial y(k+1)}{\partial u(k)} = 0.02101$$

The training procedure introduced in section 5.3.2 was also utilised to train the neural network controller for the angular velocity subsystem. After extensive simulations and experiments, the neural network structure (3,5,1) was employed for the angular velocity controller $ONNC_{\omega}$. The training parameters were chosen as follows:

- The learning rates γ and ε were chosen as 0.01 and 0.015, respectively.
- The constant K_u was chosen as 4.
- Training epoch included 100 iterations.

With the aim of improving the generalisation property of the neural network controller, three decoupled models of the angular velocity subsystem were utilised during the training process. In addition, similar to the training work of the linear velocity controller, an extensive number of training epochs with various values of the input reference were conducted in the training work of the angular velocity controller.

After the training procedure, optimal weights for the neural network controller were obtained as follows:

$$\overline{\omega W_{ji}} = \begin{bmatrix} 0.0621 & -0.2031 & -0.0301 \\ 0.0257 & 0.0153 & 0.2607 \\ -0.0034 & 0.0012 & 0.0418 \\ 0.1107 & 0.0387 & 0.0095 \\ 0.2165 & -0.1274 & 0.1807 \end{bmatrix} \quad \omega\overline{b_j} = \begin{bmatrix} -0.1214 \\ 0.0624 \\ 0.0412 \\ 0.0062 \\ 0.0027 \end{bmatrix} \qquad (5.55)$$

$$\omega W_j = [-0.0754 \quad -0.0427 \quad 0.0368 \quad -0.1258 \quad 0.0576] \quad \omega b = [-0.1206]$$

where $\omega\overline{W_{ij}}$, ωW_j, $\omega\overline{b_j}$, ωb are weights and biases for the hidden-output and input-hidden layers of the $ONNC_{\omega}$.

For testing the performance of the obtained neural network controller $ONNC_{\omega}$, the step reference input was introduced. Figure 5.7 presents the system outputs of three angular velocity closed-loop systems corresponding to the nominal model, lower bounded model and upper bounded model.

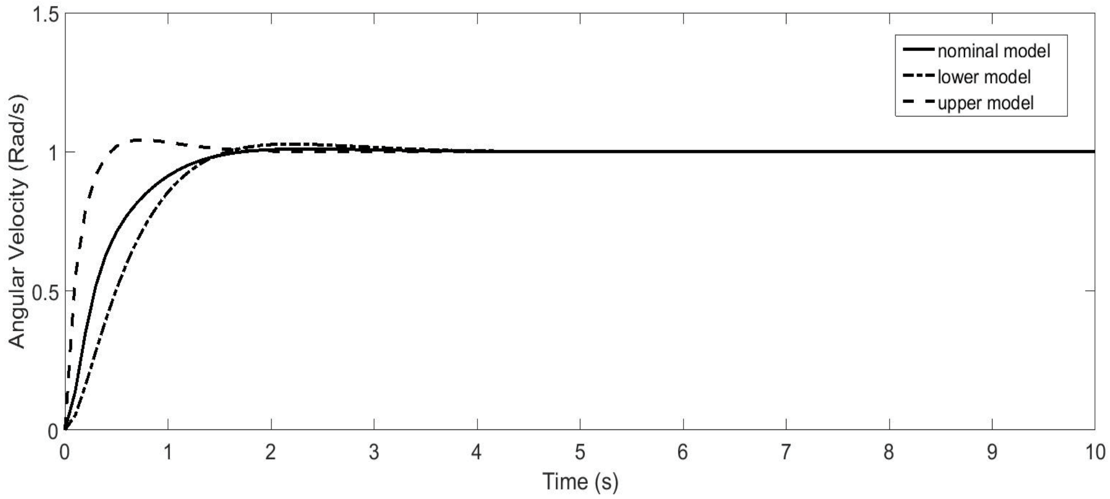


Figure 5.7: System outputs of three representational sub-systems P_{DD}(2,2), P$_{1DD}$(2,2) and P$_{2DD}$(2,2) these being controlled by $ONNC_{\omega}$

5.4.2 Experimental results

In order to verify the proposed control approaches, four real-time experiments were implemented in this section. The first experiment involved comparing results between the closed-loop responses of the system controlled by the intelligent multivariable low-level controller and those controlled by the OMPID controller. Like the experiments 2 and 3 of the previous chapter, experiments 2 and 3 of this chapter were conducted for testing the quality of the proposed low-level controller when the smart hospital bed carried a person and travelled in different environmental conditions at various velocities. In the final experiment, the operator-following task was implemented to estimate the performance of the advanced operator-following control strategy.

5.4.2.1 Experiment 1

In this experiment, the smart hospital bed loaded with a 75kg person moved on a granite surface. In the first case, the linear velocity reference is set at 1 (m/s) while simultaneously keeping the desired value of the angular velocity at 0 (rad/s). In the second case, the desired value of the linear velocity is kept at 0 (m/s) while setting the angular velocity reference at 1 (rad/s). The closed-loop step responses of the hospital bed system were observed and measured in both cases. Figure 5.8 presents a comparison of the bed performances controlled by the two proposed control approaches.

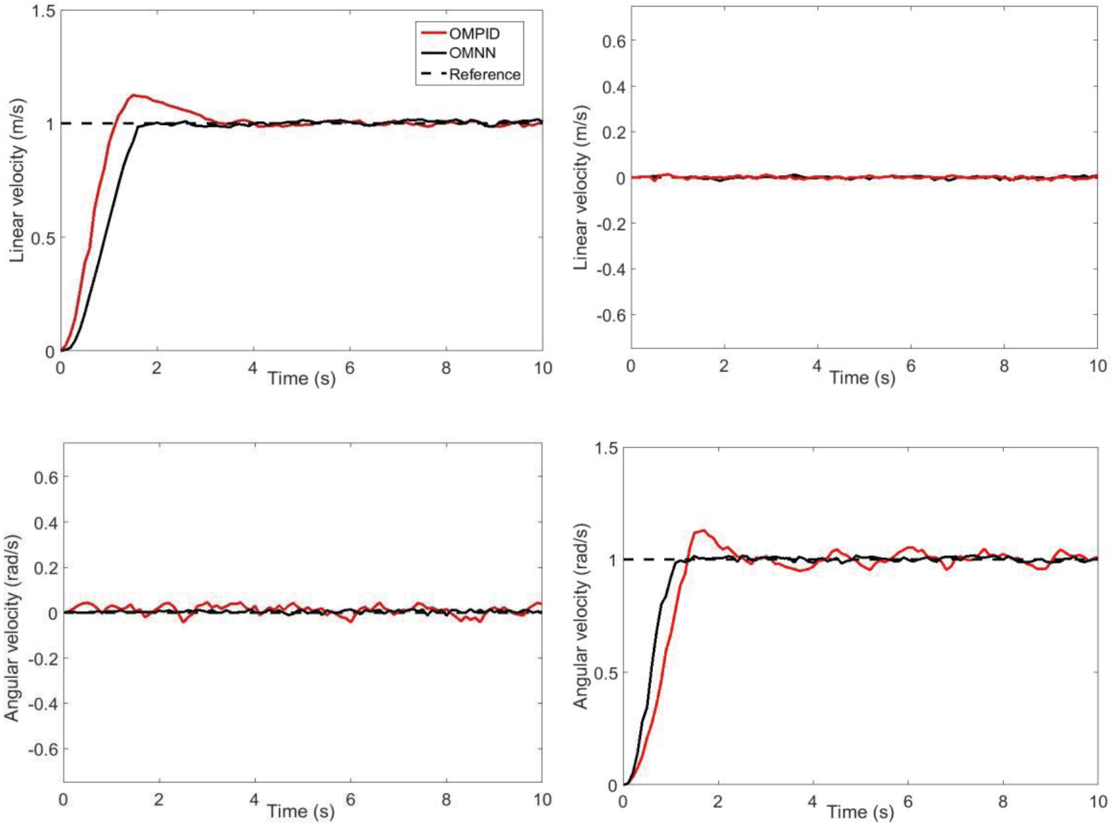


Figure 5.8: Comparison between OMNN control approach and OMIPD control approach

5.4.2.2 Experiment 2

In this experiment, the environmental condition was kept as in experiment 1. By introducing various step input values of the linear velocity and angular velocity, the system outputs of the smart hospital bed were measured and plotted. Figures 5.9, 5.10, 5.11 and 512 show the system responses with $v_d = 0.4\ (m/s)$; $\omega_d = 0.4\ (rad/s), v_d = 0.6\ (m/s)$; $\omega_d = 0.6\ (rad/s)$, $v_d = 0.8\ (m/s)$; $\omega_d = 0.8\ (rad/s)$, $v_d = 1\ (m/s)$; $\omega_d = 1\ (rad/s)$, respectively. In each figure, the output of the linear velocity subsystem is represented by element (1,1) whereas the output of the angular velocity subsystem is represented by element (2,2). Elements (1,2) and (2,1) depict the coupling effects in the hospital bed dynamics.

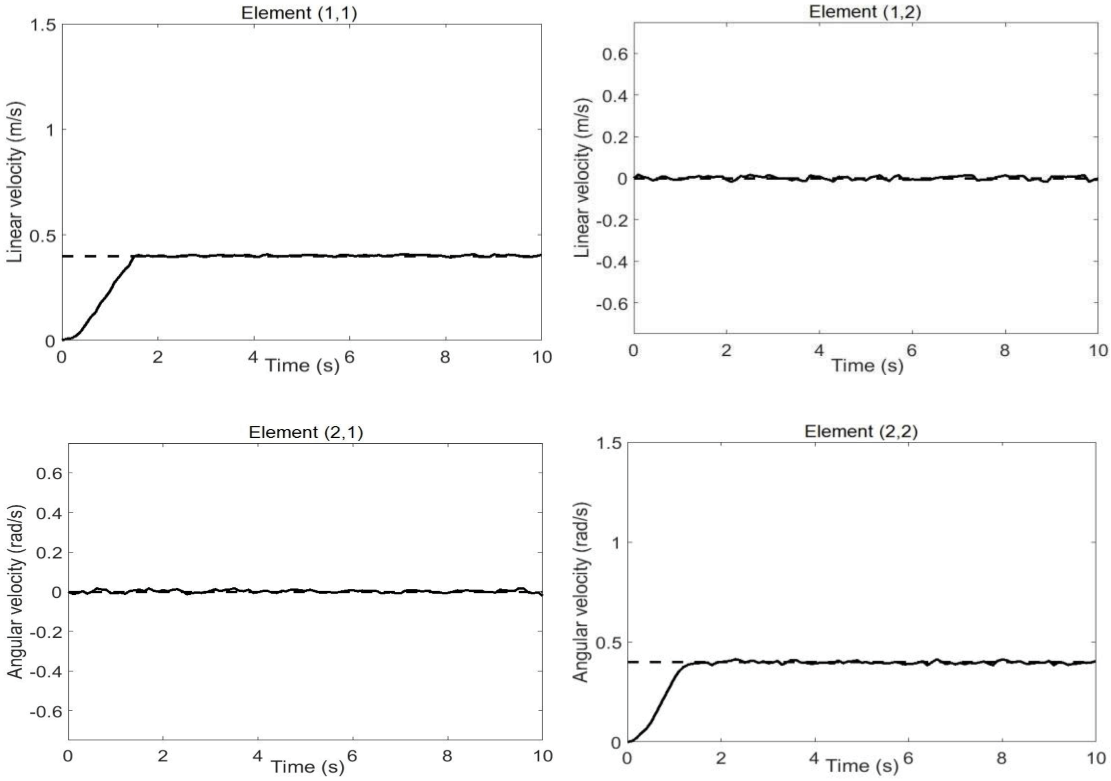


Figure 5.9: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.4\ (m/s);\ \omega_d = 0.4\ (rad/s)$

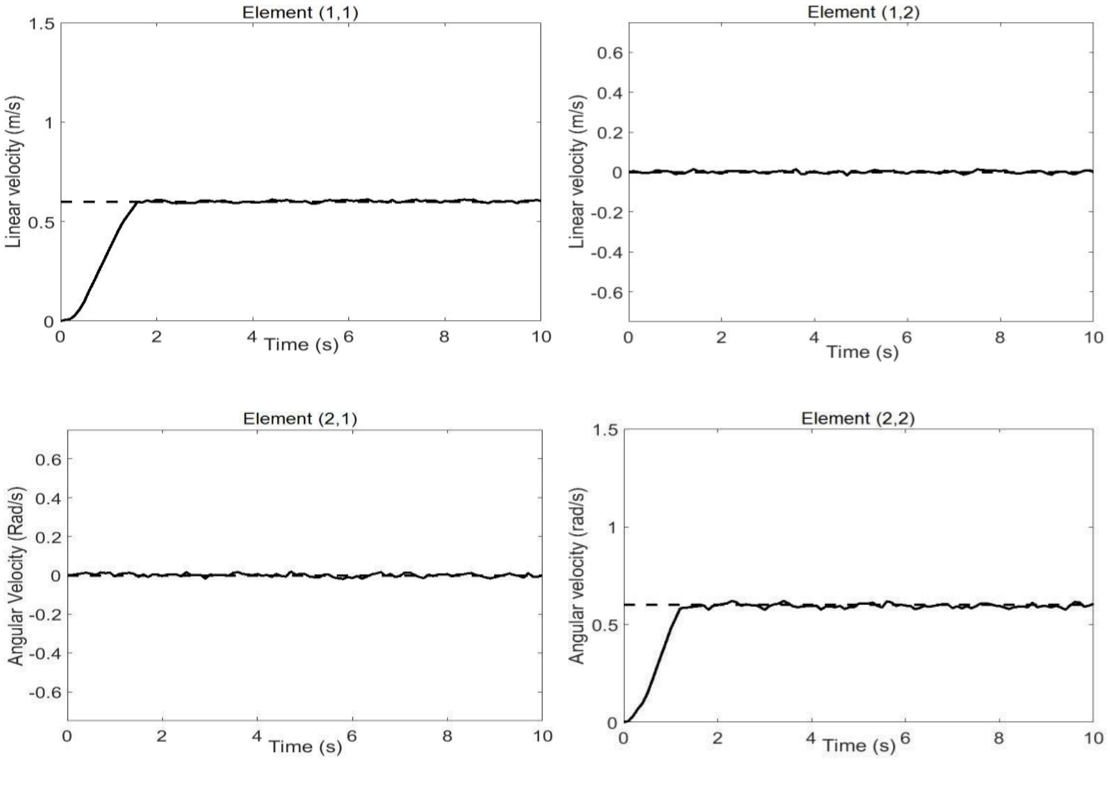


Figure 5.10: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.6\ (m/s);\ \omega_d = 0.6\ (rad/s)$

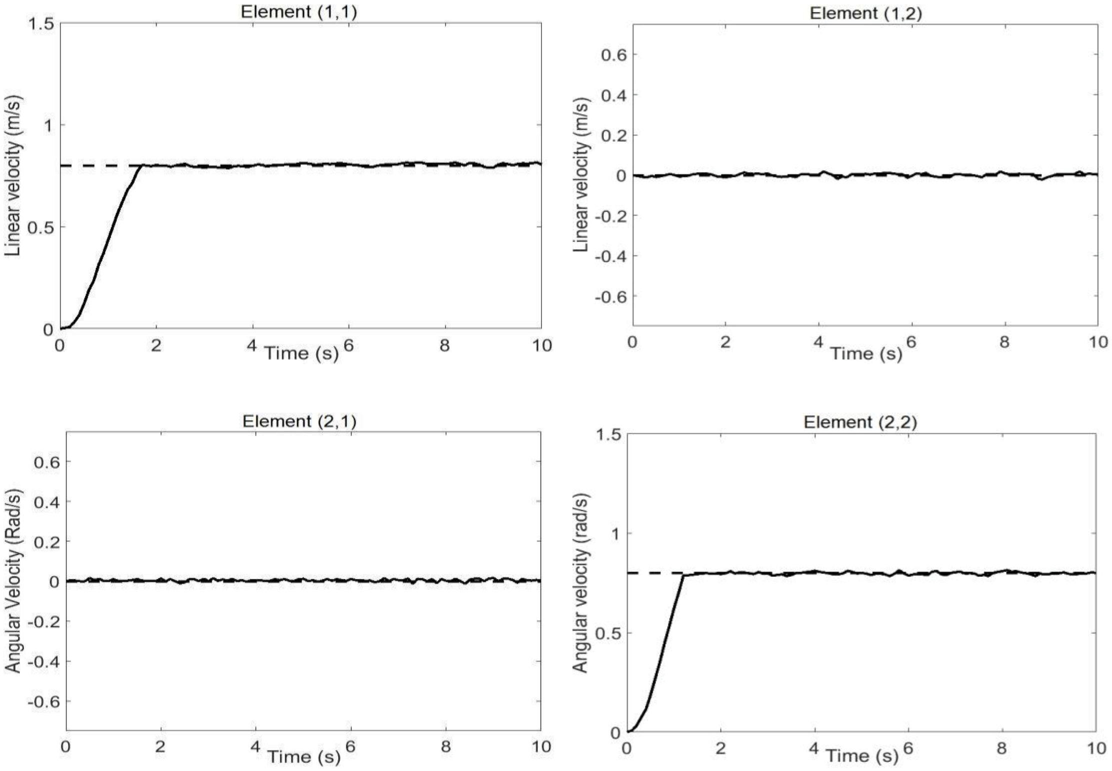


Figure 5.11: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 0.8\ (m/s);\ \omega_d = 0.8\ (rad/s)$

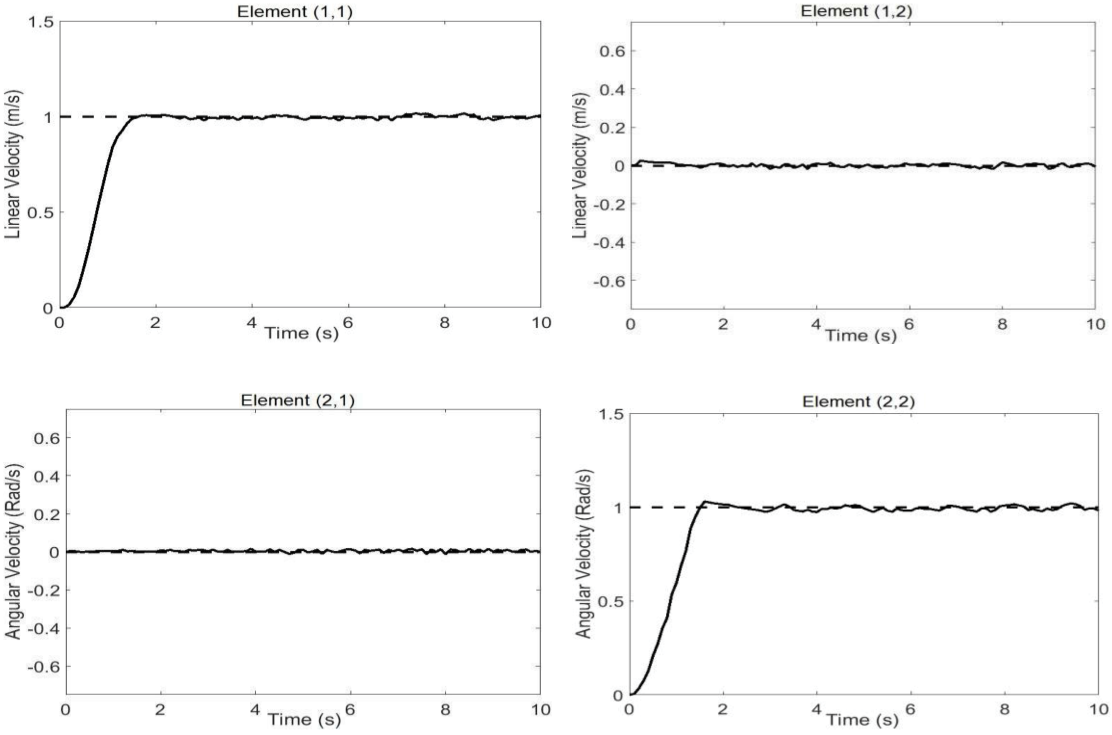


Figure 5.12: The closed-loop response of the hospital bed dynamics being controlled by OMNNC with $v_d = 1\ (m/s);\ \omega_d = 1\ (rad/s)$

Tables 5.2 and 5.3 were drawn up for analysing the output performance of the two velocities loop when the bed travelled on the granite surface, shown as follows:

$v_d(m/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.6	1.8	1
0.6	1.7	1.8	2
0.8	1.8	2.0	2
1	1.9	2.1	2

Table 5.2: Output performance of the linear velocity subsystem (on the granite surface)

$\omega_d(rad/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.5	1.7	3
0.6	1.4	1.6	2
0.8	1.4	1.7	1
1	1.5	1.7	2

Table 5.3: Output performance of the angular velocity subsystem (on the granite surface)

5.4.2.3 Experiment 3

In the third experiment, the smart hospital bed loaded with an 85 kg person travelled on two different surfaces, a carpet surface and a cement surface. In each situation, various referent velocity inputs were introduced. Figure 5.13 shows the performance of the smart hospital bed when it moved on the carpet surface with two cases: $v_d = 1\ (m/s);\ \omega_d = 0\ (rad/s)$ and $v_d = 0\ (m/s);\ \omega_d = 1\ (rad/s)$**.** After extensive experiments, four summary tables 5.4 to 5.7 related to output responses of the linear velocity and angular velocity loop were obtained.

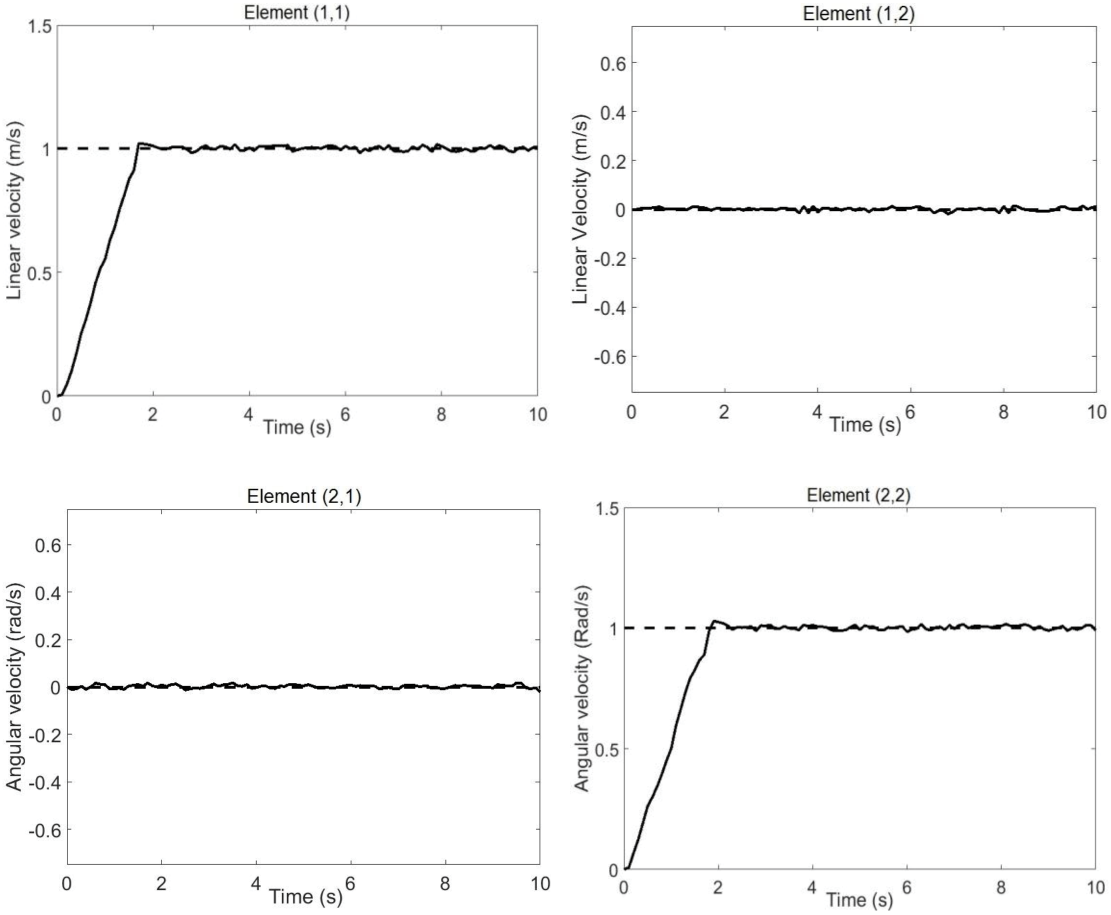


Figure 5.13: Output responses of the smart hospital bed being controlled by OMNNC with $v_d = 1\ (m/s)$; $\omega_d = 1\ (rad/s)$ in the case of travelling on the carpet surface

The hospital bed travelling on the carpet surfaces

$v_d(m/s)$	$T_{rise}\ (s)$	$T_{set}(s)$	$M_p(\%)$
0.4	1.7	1.9	2
0.6	1.8	2.0	3
0.8	1.9	2.1	2
1	1.8	2.2	2

Table 5.4: Output performance of the linear velocity subsystem (on the carpet surface)

$\omega_d (rad/s)$	$T_{rise}\ (s)$	$T_{set} (s)$	$M_p (\%)$
0.4	1.7	1.9	3
0.6	1.8	2.0	2
0.8	1.8	2.0	2
1	1.9	2.1	3

Table 5.5: Output performance of the angular velocity subsystem (on the carpet surface)

The hospital bed travelling on the cement surfaces

$v_d (m/s)$	$T_{rise}\ (s)$	$T_{set} (s)$	$M_p (\%)$
0.4	1.5	1.7	4
0.6	1.6	1.9	3
0.8	1.8	2.0	2
1	1.8	2.1	3

Table 5.6: Output performance of the linear velocity subsystem (on the cement surface)

$\omega_d (rad/s)$	$T_{rise}\ (s)$	$T_{set} (s)$	$M_p (\%)$
0.4	1.5	1.7	3
0.6	1.6	1.8	2
0.8	1.7	1.9	4
1	1.7	2.0	3

Table 5.7: Output performance of the angular velocity subsystem (on the cement surface)

5.4.2.4 Experiment 4

The final experiment was conducted for validating the effectiveness of the advanced operator-following control strategy. Details of the experiment were derived from the experiment 3.5.2. The autonomous navigation robot (called Turtlebot) was still chosen as the operator. Experimental results of the operator-following task are shown below:

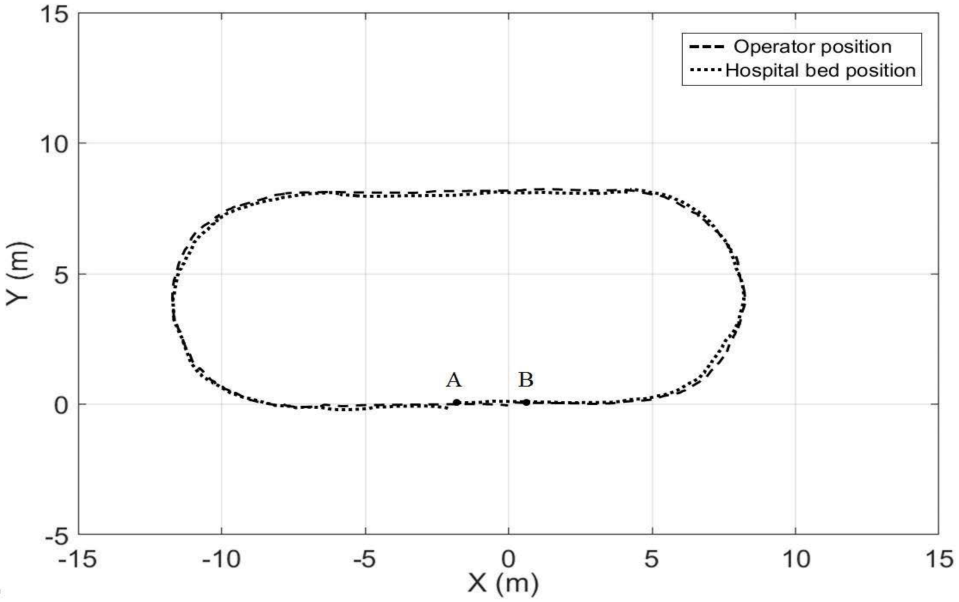


Figure 5.14: Trajectory of the hospital bed and operator

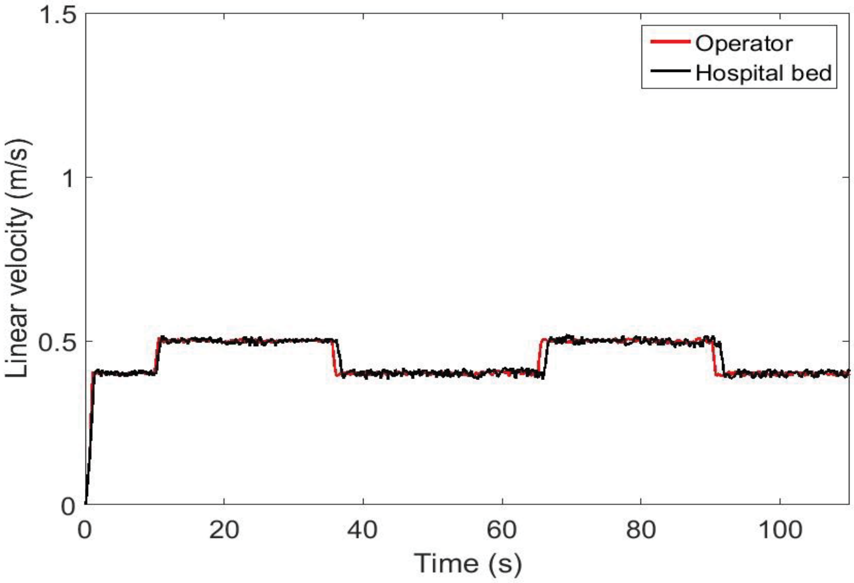


Figure 5.15: Linear velocity of the hospital bed and operator

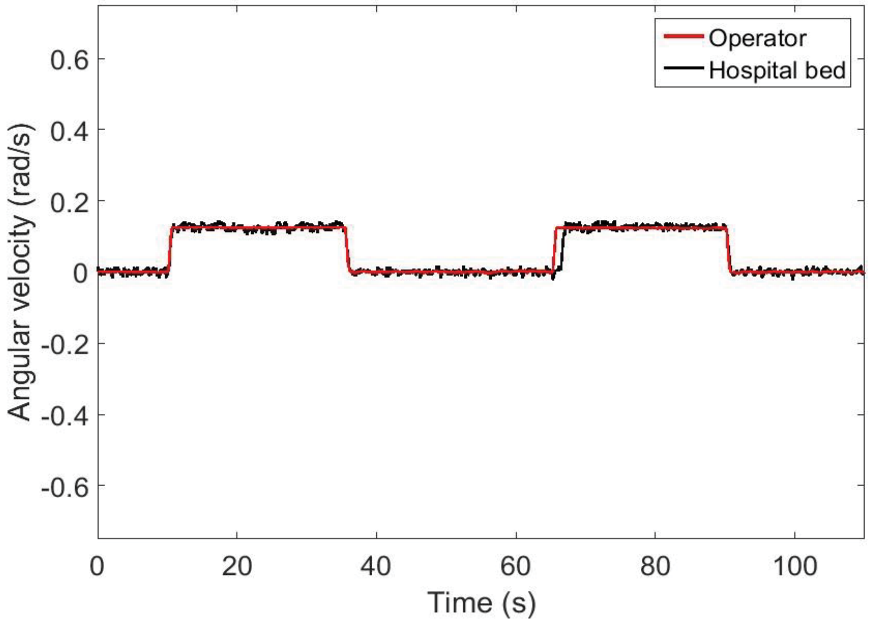


Figure 5.16: Angular velocity of the hospital bed and operator

5.5 DISCUSSION

For the smart hospital bed system, the proposed advanced operator-following control strategy provides an effective solution for enhancing the hospital bed performance when it follows an operator. In the proposed approach, an intelligent multivariable low-level controller is developed for guaranteeing the stability of the overall system under the effects of the parameter uncertainties.

In the first design stage of the proposed low-level control approach, the diagonal decoupling technique significantly reduces the multivariable hospital bed dynamics. Through the pre-compensators, two independent velocity subsystems are obtained. For each velocity subsystem, there are three corresponding decoupled models consisting of the nominal model, the lower bounded model and the upper bounded model. These models are utilised alternatively in the training process for obtaining the optimal neural network controllers in the second design stage of the proposed control method. Figures 5.6 and 5.7 show that the two neural network controllers $ONNC_v$and $ONNC_\omega$, can guarantee the desired performance of the two sub-systems even in the presence of system uncertainties.

With the aim of validating the proposed intelligent low-level control method, the first real-time experiment is conducted by introducing step inputs to the hospital bed system. Output performance of the system controlled by the advanced low-level controller is compared with those of the OMPID control approach. The experimental results show that the proposed control approach not only reduces the coupling effects between control variables in the hospital bed system but also drives the actual velocities to track the desired velocities better than the OMPID control method.

In the presence of system uncertainties, the effectiveness of the intelligent multivariable low-level control is also confirmed through the results of the experiments 2 and 3. The experiment 2 shows the hospital bed performance at various velocity step inputs in the case of travelling on the granite surface and being loaded with a 75kg person. On the other hand, the experiment 3 is conducted for estimating the quality of the proposed low-level controller when the smart hospital bed carries an 85kg person and moves in two different environmental conditions consisting of the carpet surface and the cement surface. It can be seen from Figures 5.9 to 5.13 that the actual outputs of both velocities subsystems can track the reference velocities with small error. In addition, results reported in the six tables 5.2 to 5.7 indicate that the performance of the neural network multivariable controller is satisfactory under the design conditions including the short rise time and settling time and small overshoot.

The final experiment presents the operator-following performance of the smart hospital bed when applying the advanced operator-following control strategy. Compared with the experimental results in section 3.5.2, it can be observed that the hospital bed performance is improved.

CHAPTER 6 . CONCLUSION AND FUTURE RESEARCH DIRECTIONS

6.1 CONCLUSION

The objective of this thesis was to develop advanced assistive control strategies for the smart hospital bed. The proposed approaches were divided into three major aspects including the operator detection algorithm, the advanced low-level control method and the advanced operator-following control strategy. These three aspects are crucial for developing the operator-following function of the smart hospital bed.

Through the literature review, the research direction on the development of the operator-following solution for the smart hospital bed is practical and innovative. Compared with current patient transport solutions, the operator-following solution offers more benefits such as a reduction of the physical demands of nursing work and the risks associated with patient transportation. Another advantage is that following an autonomous navigation robot enables the hospital staff to completely concentrate on patient-care activities. In terms of developing advanced functions for the smart hospital bed, the work done in this doctoral study is unique.

Chapter 3 was about the investigation and development of the operator detection algorithm for the smart hospital bed. This work was necessary for the bed system since it allowed the bed to identify the target which needs to be followed. The main contribution of this chapter was the design of two operator detection algorithms based on a single laser range finder. It enabled the smart bed to classify the operator in a complex space including the operator and various non-target objects. The algorithms consisted of the Gaussian distribution method and the neural network based operator detection method.

Experimental results showed that the proposed methods successfully detected the operator with high accuracy. The average testing results of the GDM based operator detection algorithm were 89.2% for sensitivity and 81.4% for specificity whereas the best performance of this algorithm was 91% for sensitivity and 88.56% for specificity. Comparing the performance of the GDM classifier with that of the NN-LM classifier confirmed that the NN-LM classification approach detected the operator more accurately than the GDM classification approach. There were a sensitivity of 91.65% and a specificity of 88.56% in the mean testing results of the NN-LM classifier. In addition, the best performance of the NN-LM classifier was a 92.41% true positive rate and a 89.8% true negative rate.

Chapter 3 also involved the operator-following performance of the smart hospital bed. This was conducted from a combination of the NN-LM based operator detection method and a PID based operator-following control strategy. Real-time implementation of the smart hospital bed showed that the bed system successfully detected and followed the target. The trajectory of the wheeled mobile system was acceptable when compared with that of the operator.

Chapter 4 described the work for solving the dynamics problem of the hospital bed system. It commenced with a construction of an approximate multivariable dynamic model of the smart hospital bed. The obtained model was important for developing multivariable low-level control algorithms. Based on this model, an Optimal Multivariable PID control strategy was developed for the bed system. Technically, the OMPID approach combined the Triangular Diagonal Dominance decoupling technique and the Optimal PID control method. The TDD decoupling technique provided a very effective solution for reducing the multivariable control problem into a series of scalar control problems. Thanks to this technique, the design procedure and its computational complexity were simplified. After being decoupled, an OPID controller was developed for each independent scalar subsystem.

Real-time implementation of the smart hospital bed system revealed that the Optimal Multivariable PID controller effectively reduced the coupling effect between the control variable’s inputs and outputs. The real-time results also showed that the OMPID controller successfully drove the actual velocities to track the reference velocities in various environmental conditions. The mean rise time, settling time and overshoot of the linear velocity subsystem were around 1.8s, 2.8s and 14% whereas those of the angular velocity subsystem were about 2s, 3.3s and 15%, respectively.

Chapter 5 introduced the proposal and development of intelligent control algorithms for the smart hospital bed. One of the proposed control approaches was the optimal multivariable neural network control algorithm. The OMNN control method was developed for guaranteeing the stability of the overall bed system under the effect of uncertainties. This algorithm was a combination of a two-phase diagonal decoupling technique and an optimal neural network control design. Similar to the OMPID control approach, the OMNN control algorithm also required two design stages. To simplify the control structure, the first design stage focused on searching pre-compensators so that the resulting compensated system had a diagonal structure. After the diagonalisation, the hospital bed system was decomposed into two scalar sub-systems. In the second design stage, the optimal neural network control method was employed to develop independent controllers for two corresponding velocity subsystems. Unlike the OMPID method, the ONNC approach utilised three dynamic models consisting of the lower bounded model, the nominal model and the upper bounded model for developing the advanced controller. During the training procedure, these three dynamic models were alternatively employed for calculating the optimal weights of the neural network controllers. Accordingly, the obtained intelligent controller can deal with the variation of the dynamic parameters of the hospital bed system model.

The effectiveness of the intelligent multivariable low-level control strategy was confirmed via the real-time smart hospital bed experiments. The experimental results showed that the OMNN controller significantly decreased the decoupled effect problem. In various environmental conditions, the results also demonstrated that the OMNN controller guaranteed the stability of the overall bed system and satisfied its performance. The average rise time, settling time and overshoot were around 1.7s, 1.9s and 2% for both velocity subsystems, respectively. Comparing the output responses of the system controlled by the OMNN controller with those controlled by the OMPID controller, the OMNN controller provided a better performance than the OMPID controller.

The alternative control approach was the advanced operator-following control strategy which was derived from a mixture of a neural network based operator detection algorithm, PID controllers and an intelligent multivariable low-level control method. In this method, the neural network based operator detection algorithm played the role of detecting the operator within the real environment. The Levenberg-Marquardt training algorithm was utilised to design the neural network classifier. The PID controllers took the responsibility of guaranteeing the desired distance and orientation between the smart hospital bed and the operator whereas the intelligent multivariable low-level controller enabled the bed to track its desired performance under the effect of uncertainties. Experimental results revealed that the operator-following performance of the smart hospital bed was enhanced when utilising the advanced operator-following control strategy.

6.2 FUTURE RESEARCH DIRECTIONS

Besides the proposed methodology described throughout this thesis, some further research directions should be concerned with improving the system performance in terms of transporting a patient. The first work is to integrate navigation and obstacle avoidance algorithms into the autonomous steering system of the bed. These will allow the bed to adapt to various situations during patient transport. Moreover, this autonomous steering system can be combined with the user intention for the purpose of developing a semi-autonomous mechanism which will enable the user to share a control task with a smart bed system. In clinical environments, this solution is valuable and convenient for the smart hospital bed since it allows the bed system to flexibly collaborate with the hospital staff, the autonomous navigation robot or a person with a disability in order to perform patient transportation. Moreover, it is also effective in several special cases such as avoiding high-speed obstacles which is in the unviewable field of the laser scanner or in lost tracking the operator.

The second future research direction is to improve the hardware system for detecting the operator and obstacles. Despite the laser range finder’s ability to provide environmental information with high accuracy, resolution and reliability of measurement, it can only scan the environment in a single plane with a 240^o field of view. Owing to this, the laser scanner cannot detect objects placed behind the laser or below or above its scanning plane. A way to deal with this drawback is through the utilisation of a sensor fusion system. Various sensor fusion systems can be integrated into the smart hospital bed such as a combination of multiple cameras and laser scanners or a mixture of a sphere camera and a Light Detection And Range (LiDAR) device. These sensor fusion systems enable the smart hospital bed to observe the operating environment with a 360^o field of view.

In this doctoral study, the intelligent multivariable low-level control strategy is a combination of a diagonal decoupling technique and an optimal neural network control method. In recent times, deep learning has been attracting a great deal of attention from worldwide researchers. In theory, the number of hidden layers and neurons in the deep learning network are more than those of the conventional neural network. In addition, the deep learning network provides a powerful set of techniques for the learning process of a network. As a result, many large and complex problems which could not be figured out by the conventional neural network are able to be solved by deep learning methods. Despite deep learning being utilised in various applications consisting of pattern recognition and classification problems, research in its application to the field of automatic control has been largely lacking. Therefore, a significant contribution for future research could be the application of the deep learning framework on the high-level control and low-level control strategies of the smart hospital bed system.

APPENDIX A. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are computational models inspired by the functioning of the human brain. They consist of simple but highly interconnected computing devices, each of which imitates the biological neurons (Alexakis et al. 2003). In theory, neural network models are created by single neurons whose operation is modelled by mathematical equations. These individual neurons are connected together as a network. Each neural network has its learning laws according to which it is capable of adjusting parameters of the neurons.

The neuron

The neuron is a program that learns concepts, i.e. it can learn to respond with *True* (1) or *False* (0) for inputs we present to it, by repeatedly "studying" examples presented to it. The structure of a single neuron is very simple. There are two inputs, a bias and an output. A simple schematic diagram is shown in figure A.1.

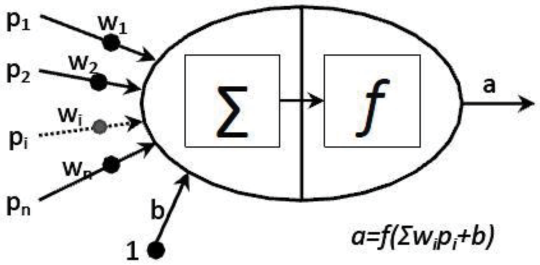


Figure A.1: An artificial neuron with one input and bias

The scalars inputs $\mathrm{p_i}$ are transmitted through connections that multiply their strength by the scalar weight w_i to form the product $w_i p_i$, again a scalar. All the weighted inputs $w_i p_i$ are added and to $\sum w_i p_i$ we also add the scalar bias b. The result is the argument of the transfer function f, which produces the output a. The bias is much like a weight, except that is has a constant input of 1. Each of the inputs and the bias are connected to the main neuron by a weight. A weight is generally a real number between 0 and 1. When the input number is fed into the neuron, it is multiplied by the corresponding weight. After this, the weights are all summed up and fed through a hard-limiter. Basically, a hard-limiter is a function that defines the threshold values for ‘firing’ the neuron. For example, the limiter could be:

$$f(x) = \left\{ \begin{matrix} x \le 0 \to 0 \\ x \ge 1 \to 1 \end{matrix} \right\} \tag{A.1}$$

Actually, both $w_i$and $b_i$are adjustable scalar parameters of the neuron and such parameters can be adjusted so that the network exhibits some desired or interesting behaviours. The way a neuron learns to distinguish patterns is through modifying its weights, the concept of a learning rule must be introduced. In the neuron, the most common form of learning is by adjusting the weights by the difference between the desired output and the actual output. Mathematically, this can be written:

$$\Delta w_i = x_i \delta \tag{A.2}$$

where δ = *(desired output) - (actual output)*

Transfer function

The transfer function also called activation function is a monotonically increasing, continuous, differentiable function, applied to the weight input of a neuron to produce the final output. In theory, there are different types of transfer functions which are utilised for designing the neural network. Several common activation functions consisting of Log-Sigmoid, Tan-Sigmoid and Linear transfer functions are shown as follows:

- Log-sigmoid transfer function

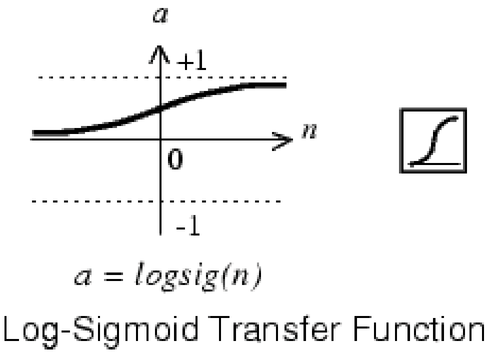


Figure A.2: Log-sigmoid transfer function

Math expression: $a = \frac{1}{1+ e^{-n}}$ (A.3)

The sigmoid transfer function is not only natural from the statistic point of view but is also good squashing functions for bounded activation (Duch & Jankowski 1999). Its output ranges over all values between 0 and 1 and makes a transition from values near 0 to values near 1. The sigmoid function is commonly used in back-propagation networks since it is easy to calculate its derivatives.

- Tan-sigmoid transfer function

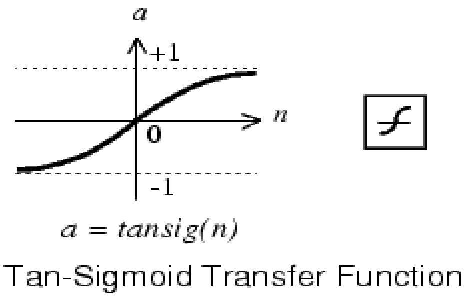


Figure A.3: Tan-sigmoid transfer function

Math expression:$a = \frac{1- e^{-n}}{1+ e^{-n}}$ (A.4)

The tan-sigmoid transfer function also called hyperbolic tangent transfer function is related to a bipolar sigmoid which has an output in the range of -1 to +1. Technically, this function is a good trade-off for neural networks, where speed is more important than the exact shape of the transfer function (Dorofki et al. 2012).

- Linear (purelin) transfer function

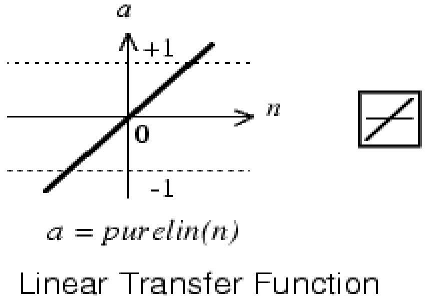


Figure A.4: Linear transfer function

Math expression: $a = f(n) = n$ (A.5)

In fact, the relationship between input and output of most real models is nonlinear. However, in some cases, real models operated within nominal parameters have input/output behaviour that is close enough to linear. Purelin transfer function is utilised to represent the input/output relationship in this situation.

Single-layer feed-forward network

A layered neural network is a network of neurons organised in the form of layers. The right image of Figure A.5 shows a neuron with a single R-element input vector. In this neural network structure, the individual element inputs $p1, p2, \dots, pR$ are multiplied by weights w11, w12, …, w1R and the weighted values are fed to the summing junction. Their sum is simply Wp, the dot product of the (single row) matrix W and the vector p. The net input to the transfer function f is n, the sum of the bias b and the product Wp. This sum is passed to the transfer function f to get the neuron's output a, which in this case is a scalar. Note that if there were more than one neuron, the network output would be a vector (Demuth H. 2001).

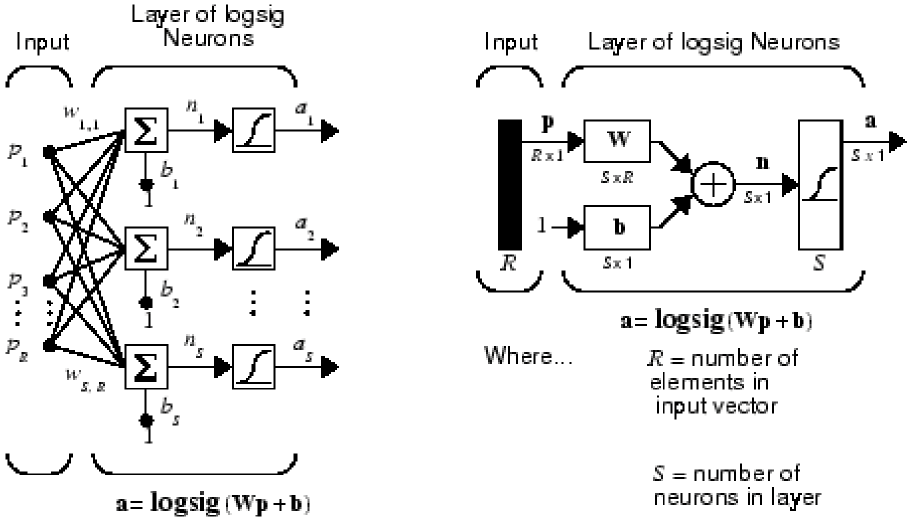


Figure A.5: Single-layer network of S *logsig* neurons

The input vector elements enter the network through the weight matrix *W*.

$$W = \begin{Bmatrix} w_{1,1}w_{1,2} \dots \dots w_{1,R} \\ w_{2,1}w_{2,2} \dots \dots w_{2,R} \\ \vdots \\ w_{S,1}w_{S,2} \dots \dots w_{S,R} \end{Bmatrix} \tag{A.6}$$

The row indices on the elements of matrix *W* indicate the destination neuron of the weight and the column indices indicate which source is the input for that weight.In other words, this network is strictly of a feed forward type. The designation “single-layer” refers to the output layer of computation nodes. The input layer of source nodes does not count, because no computation is performed there.

A one-layer network with *R* input elements and *S* neurons are shown in figure A.5. In this network each element of the input vector *p* is connected to each neuron input through the weight matrix W_p. The *ith* neuron has a summer that gathers its weighted inputs and bias to form its own scalar output *n(i)*. The various *n(i)* taken together form an *S*-element net input vector *n*. The sum, *n*, is the argument of the transfer function *f*. Finally, the neuron layer outputs form a column vector *a*. It is common for the number of inputs to a layer to be different from the number of neurons. A layer is not constrained to have the number of its inputs equal to the number of its neurons.

Multi-layer feed-forward network

Multi-layer network may distinguish itself by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden neurons is to intervene between the external input and the network output. By adding one or more hidden layers, the network is enabled to extract higher-order statistics and is particularly valuable when the size of the input layer is large.

Each neuron in the hidden layer is connected to a local set of source nodes that lie in its immediate neighbourhood. Likewise, each neuron in the output layer is connected to local set variations of the source signal.

A network can have several layers. Each layer has a weight matrix W, a bias vector *b*, and an output vector *a*. To distinguish between the weight matrices, output vectors and so on, for each of these layers, we will append the number of the layer to the names for each of these variables. For instance, the weight matrix and output vector for the first layer are denoted as *W1* and *A1*, for the second layer these variables are designated as *W2*, *A2* and so on. The network shown above has *R1* inputs, *S1* neurons in the first layer, *S2* neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input 1 is fed to the biases for each neuron.

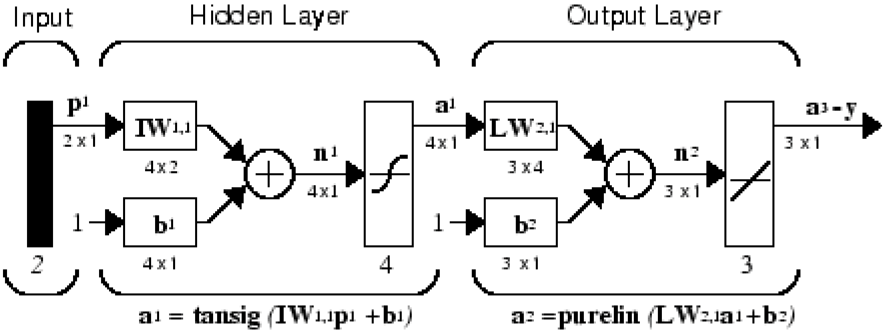


Figure A.6: Two-layer *tansig/purelin* network

The outputs of each intermediate layer are the inputs to the following layer. Thus layer 2 can be analysed as a one-layer network with *S1* inputs, S2 neurons, and an *S1xS2* weight matrix *W2*. The inputs to layer 2 is *a1*, the output is *a2*. All the vectors and matrices of layer 2 can be treated as a single layer network on its own. This approach can be taken with any layer of the network. The layers of a multi-layer network play different roles. A layer that produces the network output is called an output layer. All other layers are called hidden layers (Demuth H. 2001). Multi-layer networks are quite powerful. For instance, a network of three layers is used extensively in back-propagation neural network. Normally, no more than three layers are required in neurons like feed-forward networks, because a three-layer network can generate complex decision regions.

Backpropagation

Backpropagation was created by generalising the Widrow-Hoff learning rule to multi-layer networks and nonlinear differentiable transfer function created backpropagation (Mishra & Sawarkar 2012). Input vectors and the corresponding output vectors are used to train the network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard backpropagation is a gradient decent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term backpropagation refers to the manner in which the gradient is computed for nonlinear multi-layer networks. There are numbers of variations on the basic algorithm which are based on other standard optimisation techniques, such as conjugate gradient and Newton methods.

The backpropagation neural network is a feed-forward network that usually has hidden layers (Hasnain S.K.U 2001). The activation function for this type of network is generally the sigmoid function. Since the activation function for these nodes is sigmoid function above, the output from each node is given by:

$$\sigma_i^k = F(a_i^k) \tag{A.7}$$

where a_i is the total input to node *i*, which is given by:

$$\sigma_i^k = \sum_{j=1}^{n} w_{ij} a_j^K + \theta_i \tag{A.8}$$

Weight w_{ij} is the weight of the connection form node *j* to node *i*. Now, as for perception, we will minimise the error in the network by using the gradient descent algorithm to adjust the weights. So the change in the weight from node *j* to node *i* is given by:

$$\Delta_K W_{ij} = -\alpha \frac{\partial E^K}{\partial W_{ij}} \tag{A.9}$$

where E^K is the mean square error for the K_{th} pattern. The error for a hidden node *i* is calculated from the errors of the nodes in the next layer to which node *i* is connected. This is how the error of the network is back propagated.

So, putting them all together, the change for the weight w_{ij}, where node *i* is in a hidden layer, is given by:

$$\begin{aligned}\Delta_k w_{ij} &= \alpha \delta_i^k \sigma_j^k \\ &= \alpha \left(F'\left(a_k^k\right) \sum_{n=1}^{N_{p+1}} \delta_n^k w_{ni} \right) \sigma_j^k \\ &= \alpha \left(\sigma_i^k \left(1 - \sigma_i^k\right) \sum_{n=1}^{N_{p+1}} \delta_n^k w_{ni} \right) \sigma_j^k \end{aligned} \tag{A.10}$$

The changes in the weights of the network, which allow the network to learn, are now totally defined. This generalised delta rule for backpropagation neural networks defines how the weights between the outputs layer and the hidden layer change, and how the weights between other layers change also. This network is called backpropagation because the errors in the network are fed backward, or back propagated, through the network.

Generalisation is perhaps the most useful feature of a backpropagation network. Since the network uses supervised training, a set of input patterns can be organised into groups and fed to the network. The network will “observe” the patterns in each group, and will learn to identify the characteristics that separate the groups. Often, these characteristics are such that a trained network will be able to identify the correct groups, even if the patterns are noisy. The network learns to ignore the irrelevant data in the input patterns.

APPENDIX B. AUTONOMOUS NAVIGATION ROBOT - TURTLEBOT

Turtlebot bot is a low-cost robot with open source software. It was created by Willow Garage by Melonee Wise and Tully Foote in November 2010 (Ackerman 2013). The structure of the Turtlebot is divided into four parts and as shown in Figure B.1.

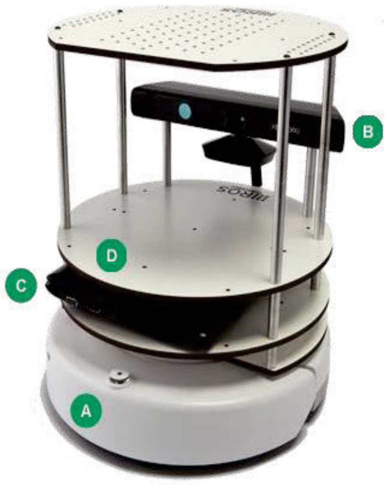


Figure B.1: The Turtlebot

Four parts of the Turtlebot include:

- Part A: iRobot create, 3000 mAh Ni-MH Battery Pack, 300 degrees/second Signgle Axis Gyro, 12V 1.5Amp Power Board Supply (for powering the Kinect)
- Part B: 3D sensor of Microsoft (Kinect XBOX 360)
- Part C: Central process unit – ThinkPad X130e with the processor of AMD FUSION E300, a graphic card of AMD Radeon HD 6320 500MHz turbo up to 600MHz and an internal hard drive of 320Gb.
- Part D: Turtlebot frame

Part A – iRobot Create

In fact, the Turtlebot is a variant of the iRobot Create. This iRobot Create was manufactured by iRobot and introduced in 2007. It is explicitly designed for robotics developments and improves the experience beyond simply hacking the Roomba robot. The upside and bottom side of the iRobot Create are depicted in Figure B.2 and B.3.

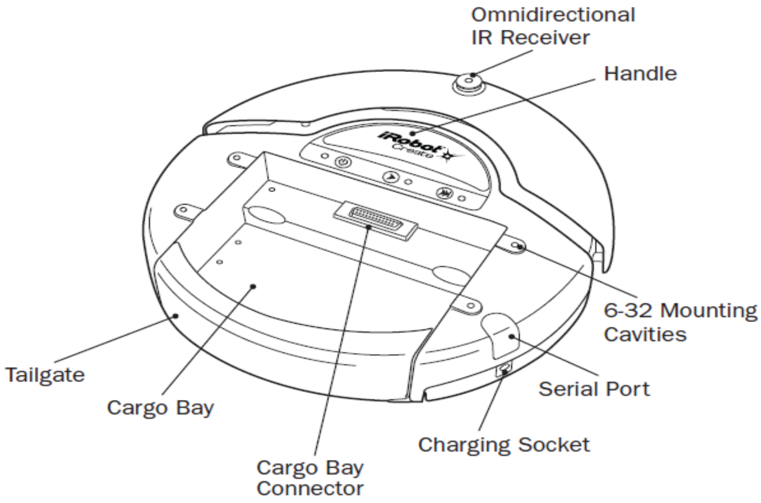


Figure B.2: Upside of the iRobot Create

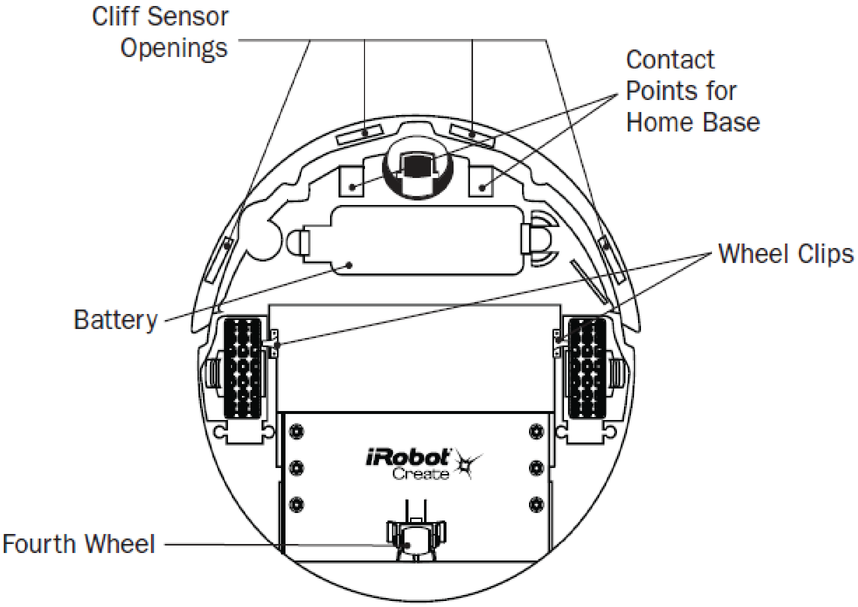


Figure B.3: Bottom side of the iRobot Create

The iRobot Create robot has several differences from the Roomba robot. Instead of keeping the vacuum cleaner part of the Roomba, the developer replaces it by a 25 pin cargo bay connector. This device includes the same signals as the serial port on the side, as well as including several digital inputs and outputs and motor drive outputs. The cargo bay connector provides four digital inputs, an analog input, three digital outputs, three low-side driver outputs(useful for driving motor), a charging indicator, a power toggle, serial TX and RX, a 5V reference, battery ground and battery voltage. Figure B.4 shows details of the 25 pin cargo bay connector.

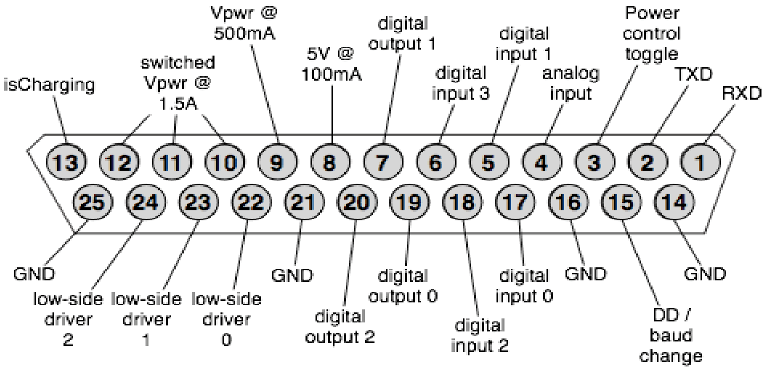


Figure B.4: The 25 pin cargo bay connector

iRobot Create has many types of sensors which are divided into two groups, external sensors and internal sensors.

External Sensors	Caster wheel drop	Right bumper	Front right cliff
	Left wheel drop	Wall	Front left cliff
	Right wheel drop	Left cliff	Play and select buttons
	Left bumper	Right cliff	Ommidirectional IR receiver
Internal Sensors	Charging state	Battery temperature	Left wheel overcurrent
	Battery current	Battery Voltage	Right wheel overcurrent
	Battery charge	Right wheel encoder	Left wheel encoder

Table B.1: Sensor system of the iRobot Create

Part B – Kinect XBOX 360

The Kinect sensor manufactured by Microsoft is a flat black box sitting on a small platform. Differently from the original Kinect sensor, the Kinect sensor used in the Turtlebot is modified. The adapter is replaced by a powered board connecting to the 25 pin cargo bay connector. Technically, the Kinect sensor consists of three main parts: color VGA camera, depth sensor and multi-array microphone.

- Color VGA camera: This video camera aids in facial recognition and other detection features by detecting three color components: red, green and blue. Microsoft calls this an "RGB camera" referring to the color components it detects. The name of this RGB camera is MT9M112 (1/4-Inch, 1.3 Megapixel CMOS Image Sensor) and from Micron.
- Depth sensor: two sensors including an infrared projector and a monochrome CMOS sensor work together to acquire the environmental information in 3D regardless of the lighting conditions. The infrared projector and the monochrome sensor called MT9V112 (1/6-Inch SOC VGA CMOS Digital Image sensor) and MT9M001(1/2-Inch Megapixel CMOS Digital Image Sensor), respectively.
- Multi-array microphone: this is an array of four microphones that can isolate the voices of the players from the noise in the room. This allows the player to be a few feet away from the microphone and still use voice controls.

Part C – Central processing unit

The Turtlebot uses a computer laptop as its central processing unit. Robot Operating System (ROS) is the major operating system using for the Turtlebot. ROS provides libraries and tools to help software developers create robot applications. It also supports hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more.

Part D – Turtlebot frame

Combining stainless sticks, iRobot Create and plates with 1 inch spacing hole pattern creates the Turtlebot whose shape is a four-layer cylinder. The bottom layer is the iRobot Create. The two next layers are designed for holding the computer laptop and the Kinect sensor. The top layer is used for the carrying application within indoor environments.

REFERENCES

Ackerman, E. 2013, *Interview: TurtleBot Inventors Tell Us Everything About the Robot*, <http://spectrum.ieee.org/automaton/robotics/diy/interview-turtlebot-inventors-tell-us-everything-about-the-robot>.

Aguiar, A.P., Atassi, A.N. & Pascoal, A.M. 2000, 'Regulation of a nonholonomic dynamic wheeled mobile robot with parametric modeling uncertainty using Lyapunov functions', *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, pp. 2995-3000 vol.3.

Alexakis, C., Nyongesa, H.O., Saatchi, R., Harris, N.D., Davies, C., Emery, C., Ireland, R.H. & Heller, S.R. 2003, 'Feature extraction and classification of electrocardiogram (ECG) signals related to hypoglycaemia', *Computers in Cardiology, 2003*, pp. 537-40.

Anastasiadis, A.D., Magoulas, G.D. & Vrahatis, M.N. 2005, 'New globally convergent training scheme based on the resilient propagation algorithm', *Neurocomputing*, vol. 64, pp. 253-70.

Andaluz, V.H., Canseco, P., Varela, J., Ortiz, J.S., Pérez, M.G., Morales, V., Robertí, F. & Carelli, R. 2015, 'Modeling and Control of a Wheelchair Considering Center of Mass Lateral Displacements', in H. Liu, N. Kubota, X. Zhu, R. Dillmann & D. Zhou (eds), *Intelligent Robotics and Applications: 8th International Conference, ICIRA 2015, Portsmouth, UK, August 24-27, 2015, Proceedings, Part III*, Springer International Publishing, Cham, pp. 254-70.

Aneesh, D. 2012, 'Tracking Controller of mobile robot', *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, pp. 343-9.

Arteaga, M.A. & Kelly, R. 2004, 'Robot control without velocity measurements: new theory and experimental results', *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 297-308.

Audrey 2016, *Hospital Bed 2 Crank And Three Cranks Manual Hospital Bed*, <http://audreybraun.com/hospital-bed-2-crank/attachment/560/>.

Bhowate, A. & Deogade, S. 2015, 'Comparison of PID tuning techniques for closed loop controller of DC-DC boost converter', *International Journal of Advances in Engineering & Technology*, vol. 8, no. 1, pp. 2064-73.

Bing-Fei, W. & Cheng-Lung, J. 2013, 'Simultaneous Localization and Human Following for a Wheelchair Robot using a Camera with Depth Information', *Advances in information Sciences and Service Sciences (AISS)* vol. 5, no. 10, pp. 984 - 93.

Boquete, L., Barea, R., García, R., Mazo, M. & Sotelo, M.A. 2005, 'Control of a Robotic Wheelchair Using Recurrent Networks', *Autonomous Robots*, vol. 18, no. 1, pp. 5-20.

Boquete, L., Martín, P., Mazo, M., García, R., Barea, R., Rodríguez, F.J. & Fernández, I. 2002, 'Hardware implementation of a new neurocontrol wheelchair-guidance system', *Neurocomputing*, vol. 47, no. 1–4, pp. 145-60.

Bridges, N. & Jarquin-Valdivia, A.A. 2005, 'Use of the Trendelenburg Position as the Resuscitation Position: To T or Not to T?', *American Journal of Critical Care* vol. 14, no. 5, pp. 364-8.

Bugeja, M.K., Fabri, S.G. & Camilleri, L. 2009, 'Dual Adaptive Dynamic Control of Mobile Robots Using Neural Networks', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 129-41.

Byrns, G., Reeder, G., Jin, G. & Pachis, K. 2004, 'Risk factors for work-related low back pain in registered nurses, and potential obstacles in using mechanical lifting devices', *J Occup Environ Hygiene*, vol. 1, pp. 11-21.

Cai, J. & Matsumaru, T. 2013, 'Robot human-following limited speed control', *RO-MAN, 2013 IEEE*, pp. 81-6.

Campo, A.B. 2012, 'PID Control Design', in V.N. Katsikis (ed.), *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1*, vol. 1.

Cavdar, S. & Aydin, A. 2015, 'An Empirical Analysis for the Prediction of a Financial Crisis in Turkey through the Use of Forecast Error Measures', *Journal of Risk and Financial Management*, vol. 8, no. 3, p. 337.

Chang, Y.H., Wu, C.I. & Yang, C.Y. 2015, 'Adaptive output-feedback fault-tolerant tracking control for mobile robots under partial loss of actuator effectiveness', *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 6306-11.

Chao, W., Savkin, A.V., Clout, R. & Nguyen, H.T. 2015, 'An Intelligent Robotic Hospital Bed for Safe Transportation of Critical Neurosurgery Patients Along Crowded Hospital Corridors', *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 23, no. 5, pp. 744-54.

Chen Tun, C., Jiun-Yi, L., Ming-Fang, C. & Li Chen, F. 2011, 'Multi-robot cooperation based human tracking system using Laser Range Finder', *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 532-7.

Chung, W., Kim, H., Yoo, Y., Moon, C.B. & Park, J. 2012, 'The Detection and Following of Human Legs Through Inductive Approaches for a Mobile Robot With a Single Laser Range Finder', *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3156-66.

Chwa, D. 2010, 'Tracking Control of Differential-Drive Wheeled Mobile Robots Using a Backstepping-Like Feedback Linearization', *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 6, pp. 1285-95.

Corradini, M.L. & Orlando, G. 2002, 'Control of mobile robots with uncertainties in the dynamical model: a discrete time sliding mode approach with experimental results', *Control Engineering Practice*, vol. 10, no. 1, pp. 23-34.

Das, T. & Kar, I.N. 2006, 'Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots', *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 501-10.

Demuth H., B.M. 2001, 'Neural network Toolbox for use with Matlab', The Mathworks, Inc.

Depenau, J. 1995, 'Automated Design of Neural Network Architecture for Classification'.

Di, L. & Jinhua, Y. 2014, 'Adaptive robust control of wheeled mobile robot with uncertainties', *Advanced Motion Control (AMC),2014 IEEE 13th International Workshop on*, pp. 518-23.

Do, M.Q. & Lin, C.H. 2015, 'Embedded human following mobile robot with an RGB-D camera ', paper presented to the *14th IAPR International conference on machine vision application*, Miraikan, Tokyo, Japan.

Dorofki, M., Elshafie, A.H., Jaafar, O., Karim, O.A. & Mastura, S. 2012, 'Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data ', paper presented to the *IACSIT* Singapore.

Drivemedical 2015, *Semi electric bed* <http://www.drivemedical.com/index.php/semi-electric-bed-single-crank-792.html>.

Duch, W. & Jankowski, N. 1999, 'Survey of Neural Transfer Functions', *Neural Computing Surveys* vol. 2, pp. 163-212.

Encoder, P.C. 2016, *Model 775*, <http://encoder.com/products/incremental-thru-bore-motor-mount-encoders/model-775/>.

Ertugrul, M. & Kaynak, O. 2000, 'Neuro sliding mode control of robotic manipulators', *Mechatronics*, vol. 10, no. 1–2, pp. 239-63.

Etolen, N. 2003, *What Are the Different Types of Hospital Bed Equipment?*, <http://www.wisegeek.com/what-are-the-different-types-of-hospital-bed-equipment.htm>.

Eui-Jung, J., Jae Hoon, L., Byung-Ju, Y., Jooyoung, P., Yuta, S. & Si-Tae, N. 2014, 'Development of a Laser-Range-Finder-Based Human Tracking and Control Algorithm for a Marathoner Service Robot', *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1963-76.

Fierro, R. & Lewis, F.L. 1998, 'Control of a nonholonomic mobile robot using neural networks', *Neural Networks, IEEE Transactions on*, vol. 9, no. 4, pp. 589-600.

Fukao, T., Nakagawa, H. & Adachi, N. 2000, 'Adaptive tracking control of a nonholonomic mobile robot', *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 5, pp. 609-15.

Gholipour, A. & Yazdanpanah, M.J. 2003, 'Dynamic tracking control of nonholonomic mobile robot with model reference adaptation for uncertain parameters', *European Control Conference (ECC), 2003*, pp. 3118-22.

Hasnain S.K.U, A.S.M. 2001, 'Artificial Neural Networks in Cardiology - ECG Wave Analysis and Diagnosis Using Backpropagation Neural Networks', *IEEE Engineering in Medicine and Biology Society Conference*, Melbourne, Australia.

Hill-Rom 2014, *New Progressa® Bed System from Hill-Rom Helps Treat and Prevent Complications for ICU Patients, Eases Caregiving for Critical Care Staff*, <http://ir.hill-rom.com/releasedetail.cfm?ReleaseID=832975>.

Hill-Rom 2015, *Compella™ Bariatric bed*, <http://www.hill-rom.com.au/globalassets/en/beds/compella/hill-rom-compella-brochure-5en126306-01.pdf>.

Hiroi, Y., Matsunaka, S. & Ito, A. 2012, 'A mobile robot system with semi-autonomous navigation using simple and robust person following behavior', *Journal of Man, Machine and Technology (JMMT)*, vol. 1, no. 1, pp. 44-62.

Hodgkin, A.L. & Huxley, A.F. 1952, 'A quantitative description of membrane current and its application to conduction and excitation in nerve', *The Journal of Physiology*, vol. 117, pp. 500-54.

Hokuyo 2009, *URG-04LX*, <https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html>.

Hornyak, T. 2009, *Panasonic's Robotic Bed transforms into wheelchair*, <http://www.cnet.com/au/news/panasonics-robotic-bed-transforms-into-wheelchair/>.

Hosseini, A.E., Amini, J. & Saradjian, M.R. 2003, 'Back propagation neural network for classification of irs 1d satellite images', *High Resolution Mapping from Space*, Hanover, Germany

Huichao, M., Shurong, L. & Haiyang, C. 2012, 'Robust backstepping tracking control for mobile robots', *Control Conference (CCC), 2012 31st Chinese*, pp. 4842-6.

Hyukseong, K., Youngrock, Y., Jae Byung, P. & Kak, A.C. 2005, 'Person Tracking with a Mobile Robot using Two Uncalibrated Independently Moving Cameras', *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2877-83.

Kanayama, Y., Kimura, Y., Miyazaki, F. & Noguchi, T. 1990, 'A stable tracking control method for an autonomous mobile robot', *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 384-9 vol.1.

Kishore, G., Sunand, G. & Rajesh babu, N. 2012, 'Face Recognition Using Eigen faces By ANN with Resilient Back propagation Algorithm', *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, vol. 2, no. 11, pp. 223-8.

Kmiotek, P. & Ruichek, Y. 2008, 'Object oriented bounding box based representation using laser range finder ensory data', *Vehicular Electronics and Safety, 2008. ICVES 2008. IEEE International Conference on*, pp. 180-4.

Kumar, S.S. & Kumar, K.A. 2013, 'Neural Network in Medical and Healthcare', *International Journal of Innovative Research and Development*, vol. 2, no. 8, pp. 241-4.

L, B., R, G., R, B. & M, M. 1999, 'Neural Control of the Movements of a Wheelchair', *Journal of Intelligent and Robotic Systems*, vol. 25, no. 3, pp. 213-26.

Lee, J.H., Tsubouchi, T., Yamamoto, K. & Egawa, S. 2006, 'People Tracking Using a Robot in Motion with Laser Range Finder', *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2936-42.

Li, W., Song, P., Tan, J.T.C., Zhu, C. & Duan, F. 2015, 'Verification the feasibility of SIGVerse for human-robot interaction simulation through following task', *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1189-94.

Li, Z., Wang, Y., Song, X. & Liu, Z. 2015, 'Neural adaptive tracking control for wheeled mobile robots', *Fluid Power and Mechatronics (FPM), 2015 International Conference on*, pp. 610-7.

Linak 2016, *Linak products*, <http://www.linak.com.au/products/>.

Linet 2013a, *i-Drive Power*, <http://linet-prelive.linet.com/en-GB/news/news-and-press-releases/2013/i-drive-power>.

Linet 2013b, *Multicare*, <http://www.linetamericas.com/multicare2/#features>.

Luo, R.C., Chang, N.W., Lin, S.C. & Wu, S.C. 2009, 'Human tracking and following using sensor fusion approach for mobile assistive companion robot', *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, pp. 2235-40.

Luy, N.T. 2012, 'Reinforcement learning-based tracking control for wheeled mobile robot', *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 462-7.

Mishra, S. & Sawarkar, S. 2012, 'Video Compression Using Spiht and Neural Network', *International Journal of Computational Engineering Research* vol. 2, no. 7, pp. 98-103.

Mondal, S.C. & Mandal, P. 2014, 'Application of artificial neural network for modelling surface roughness in centerless grinding operation', paper presented to the *5th International & 26th All India Manufacturing Technology, Design and Research Conference (AIMTDR 2014)* Guwahati, Assam, India.

Morioka, K., Joo-Ho, L. & Hashimoto, H. 2002, 'Physical agent for human following in intelligent sensor network', *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, pp. 1234-9 vol.2.

Motokucho, T. & Oda, N. 2014a, 'Vision-based human-following control using optical flow field for power assisted wheelchair', *Advanced Motion Control (AMC),2014 IEEE 13th International Workshop on*, pp. 266-71.

Motokucho, T. & Oda, N. 2014b, 'Vision-based human-following control using optical flow field for power assisted wheelchair', *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*, pp. 266-71.

Mozos, O.M., Kurazume, R. & Hasegawa, T. 2010, 'Multi-Part People Detection Using 2D Range Data', *International Journal of Social Robotics*, vol. 2, no. 1, pp. 31-40.

Nelson, A. & Baptiste, A. 2004, 'Evidence-Based Practices for Safe Patient Handling and Movement', *Online Journal of Issues in Nursing*, vol. 9, no. 3.

Nguyen, D. & Widrow, B. 1990, 'Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights', *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pp. 21-6 vol.3.

Nguyen, L.L., Steven, S. & Nguyen, H.T. 2014, 'Neural Network Based Diagonal Decoupling Control of Powered Wheelchair Systems', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 2, pp. 371-8.

Nguyen, N.T., Nguyen, H.T. & Su, S. 2007, 'Advanced robust tracking control of a powered wheelchair system', *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pp. 4767-70.

Nguyen, T.H. & Anderson, B.D.O. 1979, 'Triangularization Technique of the Design of Multivariable Control Systems', *IEEE Transactions on Automatic Control*, vol. 24, no. 3, pp. 455-60.

Nguyen, T.N. 2009, 'Advanced robust multivariable control strategies for powered wheelchair systems', University of Technology, Sydney.

NPC, R. 2011, *NPC-B82HT*, <http://hobby.npcrobotics.com/store/npc-b82ht/>.

Oceancontrols 2009, *Ocean Controls KTM-198 Modbus Dual Bidirectional DC Motor Speed Controller*, <https://oceancontrols.com.au/datasheet/ocean/ktm198_modbus_manual.pdf>.

Panasonic 2009, *Panasonic Develops Bed-Shaped Robot Transformable into Wheelchair to Support People with Limited Mobility to Lead Independent Life*

<http://news.panasonic.com/global/press/data/en090918-2/en090918-2.html>.

Perng, J.W. & Chuang, C.K. 2012, 'Vision-Based Human Following and Obstacle Avoidance for an Autonomous Robot in Intersections', *Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on*, pp. 496-9.

Phidgets 2012, *Phidget interface kit 8/8/8*, <http://www.phidgets.com/products.php?product_id=1018>.

Pompeii, L., Lipscomb, H. & Dement, J. 2008, 'Surveillance of musculoskeletal injuries and disorders in a diverse cohort of workers at a tertiary care medical center.', *Am J Ind Med*, vol. 51, pp. 344-56.

Pompeii, L.A., Lipscomb, H.J., Schoenfisch, A.L. & Dement, J.M. 2009, ' Musculoskeletal injuries resulting from patient handling tasks among hospital workers', *Am. J. Ind. Med*, vol. 52, no. 7, pp. 571-8.

Raeisi, Y., Shojaei, K. & Chatraei, A. 2015, 'Output feedback trajectory tracking control of a car-like drive wheeled mobile robot using RBF neural network', *Power Electronics, Drives Systems & Technologies Conference (PEDSTC), 2015 6th*, pp. 363-8.

Retsas, A. & Pinikahana, J. 2000, 'Manual handling activities and injuries among nurses: an Australian hospital study', *J Adv Nursing*, vol. 31, pp. 875-3.

Rion, E. 2011, *Standard inclinometers*, <http://www.rion-electronic.com/inclinometer.php>.

RoboteQ 2015, *HDC2450*, <http://www.roboteq.com/index.php/roboteq-products-and-services/brushed-dc-motor-controllers/hdc2450-detail>.

Sakai, K., Hiroi, Y. & Ito, A. 2014, 'Teaching a robot where objects are: Specification of object location using human following and human orientation estimation', *2014 World Automation Congress (WAC)*, pp. 490-5.

Shaker, S., Saade, J.J. & Asmar, D. 2008, 'Fuzzy Inference-Based person following robot', *International journal of systems applications, engineering and development*, vol. 2, no. 1, pp. 29-34.

Singh, S.K. & Gupta, A.K. 2010, 'Application of support vector regression in predicting thickness strains in hydro-mechanical deep drawing and comparison with ANN and FEM', *CIRP Journal of Manufacturing Science and Technology*, vol. 3, no. 1, pp. 66-72.

Slotine, J.J. & Sastry, S.S. 1983, 'Tracking control of non-linear systems using sliding surfaces with application to robot manipulators', *American Control Conference, 1983*, pp. 132-5.

Stryker 2013a, *Electric and manual medical surgical bed* <http://patientcare.stryker.com/en/products/beds/medical-surgical-beds/electric-and-manual-beds>.

Stryker 2013b, *Prime Series Stretchers*, <https://patienthandling.stryker.com/en/products/transport/prime/prime-series-stretchers#>.

Stryker 2013c, *Zoom Motorized Drive Enhances Efficiency of Patient Transport*, <https://patienthandling.stryker.com/~/media/patienthandling/doc/case%20studies/wmu%20zoom%20efficiency%20study_a_0113.ashx>.

Stryker 2013d, *Zoom Motorized Drive Reduces Perceived Exertion when Transporting Up an Incline*, <https://patienthandling.stryker.com/~/media/patienthandling/doc/case%20studies/wmu%20zoom%20incline%20study_a_0113.ashx>.

Stryker 2013e, *Zoom Motorized Drive Reduces Spinal Loading while Transporting on Carpet*,

<https://patienthandling.stryker.com/~/media/patienthandling/doc/case%20studies/wmu%20zoom%20carpet%20study_a_0113.ashx>.

Tiemin, H. & Yang, S.X. 2001, 'An efficient neural controller for a nonholonomic mobile robot', *Computational Intelligence in Robotics and Automation, 2001. Proceedings 2001 IEEE International Symposium on*, pp. 461-6.

Velasco, F.J., Revestido, E., Lopez, E., Moyano, E. & Casado, M.H. 2008, 'Optimization Techniques for Tuning Heading Controllers of an Autonomous Inscale Fast-ferry Model', paper presented to the *8th WSEAS International Conference on Simulation, Modelling and Optimization*, Santander, Cantabria, Spain.

Vishnu Prasad, S.S., Pottakulath, V. & Ajmal, M.S. 2014, 'Development of backstepping sliding mode tracking control for Wheeled Mobile Robot', *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, pp. 1013-8.

Wang, Y.-T., Chen, Y.-C. & Lin, M.-C. 2009, 'Dynamic Object Tracking Control for a Non-Holonomic Wheeled Autonomous Robot', *Tamkang Journal of Science and Engineering* vol. 12, no. 3, pp. 339-50.

Wang, Z.P., Yang, W.R. & Ding, G.X. 2010, 'Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots Based on Neural Dynamic Model', *2010 Second WRI Global Congress on Intelligent Systems*, vol. 2, pp. 270-3.

Wikipedia 2010, *Hospital bed* <https://en.wikipedia.org/wiki/Hospital_bed>.

Xavier, J., Pacheco, M., Castro, D., Ruano, A. & Nunes, U. 2005, 'Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver', *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3930-5.

Yang, Y., Feng, G., Wang, S., Guo, X. & Wang, G. 2013, 'Using bearing-sensitive infrared sensor arrays in Motion localization for human-following robots', *Control Conference (ASCC), 2013 9th Asian*, pp. 1-5.

Yoo, Y. & Woojin, C. 2011, 'Detection and following of human legs using the SVDD (Support Vector Data Description) scheme for a mobile robot with a single Laser Range Finder', *Electrical, Control and Computer Engineering (INECCE), 2011 International Conference on*, pp. 97-102.

Yoonchang, S. & Woojin, C. 2011, 'Human tracking of a mobile robot with an onboard LRF (Laser Range Finder) using human walking motion analysis', *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pp. 366-70.

Yulin, Z., Daehie, H., Chung, J.H. & Velinsky, S.A. 1998, 'Dynamic model based robust tracking control of a differentially steered wheeled mobile robot', *American Control Conference, 1998. Proceedings of the 1998*, vol. 2, pp. 850-5 vol.2.

Zhichao, C. & Birchfield, S.T. 2007, 'Person following with a mobile robot using binocular feature-based tracking', *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 815-20.

Zuozhi, S., Dongbin, Z., Jianqiang, Y. & Xinchun, L. 2005, 'Robust motion control for nonholonomic constrained mechanical systems: sliding mode approach', *American Control Conference, 2005. Proceedings of the 2005*, pp. 2883-8 vol. 4.