Automated Robust Control System Design for Variable Speed Drives

Nnamdi Okaeme, MEng (Hons)

Submitted to the University of Nottingham for the degree of Doctor of Philosophy,
June 2008.

Acknowledgements

I would like to thank Dr. Pericle Zanchetta for the consistently invaluable help and support he provided during the course of my doctorate programme. I would also like to thank Dr. Mark Sumner for his concerted efforts in providing essential support throughout the period of my research project. I would like to express my gratitude to Professor Pat Wheeler for acting as my internal examiner and Dr. Marco Liserre for serving as my external examiner. Special thanks is also due to my colleagues for the invaluable support provided during the research project. In particular, I would like to thank James Campbell, Wanchak Lenwari, David Saltiveri, Jesus Arellano-Padilla for their useful advise and healthy discussions regarding the set-up of my experimental system. Hearty thanks is due to Chinwe Agu, Fabio Carastro, Gareth Moore, Alan James Watson, Edward Christopher, Lee Empringham, Liliana Empringham, Qiang Gao, Yahan Hua, Saul Lopez-Arevallo, Andrew Trentin, Miliana Odavic, Chris Hill, Adam Curson and Ralph Feldman.

As an expression of my most sincere gratitude, I would like to thank the University of Nottingham for the opportunity provided to undertake my doctorate programme within their world class Power Electronics, Machines and Control ($PEMC$) Group by bestowing upon me the prestigious *Tower Innovation Scholarship* towards this cause. I vow that the investment made in me would reap positively unfathomable rewards. I would also like to thank my immediate family and close friends for their invaluable help, guidance and soothing conversation during the course of my study. To all those who have been involved in my personal and professional career development ever since my conception, I would like to say thank you. Lastly, and most importantly, I would like to thank my dearly beloved parents, my father, Mr. Azuka Raphael Okaeme and my mother, Mrs Maria Ngozi Okaeme. Without these two outstanding figures in my life, I can almost certainly say that I will not be in existence, talk less about achieving this feat in my career. I dedicate this work to my father, who now rests in serenity.

Abstract

Traditional PI controllers have been largely employed for the control of industrial variable speed drives due to the design ease and performance satisfaction they provide *but*, the problem is that these controllers do not always provide robust performance under variable loads. Existing solutions present themselves as complex control strategies that demand specialist expertise for their implementation. As a direct consequence, these factors have limited their adoption for the industrial control of drives. To counter this trend, the thesis proposes two techniques for robust control system design. The developed strategies employ particular *Evolutionary Algorithms*(EA), which enable their simple and automated implementation. More specifically, the EA employed and tested are the *Genetic Algorithms* (GA), *Bacterial Foraging* (BF) and the novel *Hybrid Bacterial Foraging*, which combines specific desirable features of the GA and BF.

The first technique, aptly termed *Robust Experimental Control Design*, employs the above mentioned EA in an automated trial-and-error approach that involves directly testing control parameters on the experimental drive system, while it operates under variable mechanical loads, evolving towards the best possible solutions to the control problem. The second strategy, *Robust Identification-based Control Design*, involves a GA system identification procedure employed in automatically defining an uncertainty model for the variable mechanical loads and, through the adoption of the Frequency Domain $\mathcal{H}_\infty$ Method in combination with the developed EA, robust controllers for drive systems are designed. The results that highlight the effectiveness of the robust control system design techniques are presented. Performance comparisons between the design techniques and amongst the employed EA are also shown. The developed techniques possess commercially viable qualities because they elude the need for skilled expertise in their implementation and are deployed in a simple and automated fashion.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electric motors use approximately 70% of the world's total electrical energy [1]. Within industrialised nations, the use of electrical motors demands similarly significant portions of the total electric energy produced. In the UK for example, 65% of the total electric energy used by industry powers drives, whilst 23% of the total electrical energy used by commerce is for electric motors [2]. Any industry that wishes to improve its profitability by reducing electricity consumption must achieve efficient use of electrical drives. Apart from the fact that electric drives significantly harness a large portion of total electric power generated, a second force behind the increasing attention being paid to its efficient use is that of the environmental effects from the emissions from generating power stations. In a typical fossil-fuel powered power station the generating efficiency is around 35%. But with the consideration of transmission, distribution, electrical and mechanical losses in a drive system, the overall effect is that 100 kWh of fuel input can result in only 11 kWh of useful output energy [2]. From these reasons, it is evident that there is an ever increasing demand for more efficient operations of electric drives. Amongst other issues, these demands can be met with improved control.

The electric drives are subject to very variable industrial load conditions during operation. Given the controllers are a significant determinant of the drives' operative efficiency, they must be designed appropriately to provide robust performance under these load conditions. The appropriate method should consider all possible load conditions during the controllers' design. There are many schemes reported that have achieved the aim. Most of these schemes are complex, demanding expertise and requiring significant user involvement. The appeal of a simple approach, requiring little or no user interaction in optimising simple linear controllers is strong and desirable for many industrial applications. The thesis is focused on describing a novel approach which meets these requirements. It is an automated approach which employs Evolutionary Algorithms for the design of robust controllers for electric drives.

1.1 Electric Drives and their Control

The predominance of electric drives is due to several factors particular to its structure: electric drives can be controlled easily and very high quality performance of the drives is achievable through electronic control. They can be easily adapted to function under any condition - explosive or radioactive environments, forced air ventilation or totally enclosed and even submerged in liquids [3]. Since electric drives do not require hazardous fuels and do not emit exhaust fumes, they hardly have any operative detrimental effect on their immediate environment. They can be brought into action very quickly on demand and can be made robust in dealing with variable load conditions. In comparison to other drives, the service requirements are not at all significant; this highlights the significant economic benefits it can provide. Electric drives can be designed to operate indefinitely in all four quadrants of the torque-speed plane without requiring a special reversing gear. The four quadrants of operations highlight the motoring and generating mode for the motors. They are available for any power (ranging from 10^{-6} to 10^{8} W) and cover a wide range of torque and speed demands [3].

The electric drive consists of the electric motor and the power converters. The power converters consist of a combination of power electronic devices that essentially provide

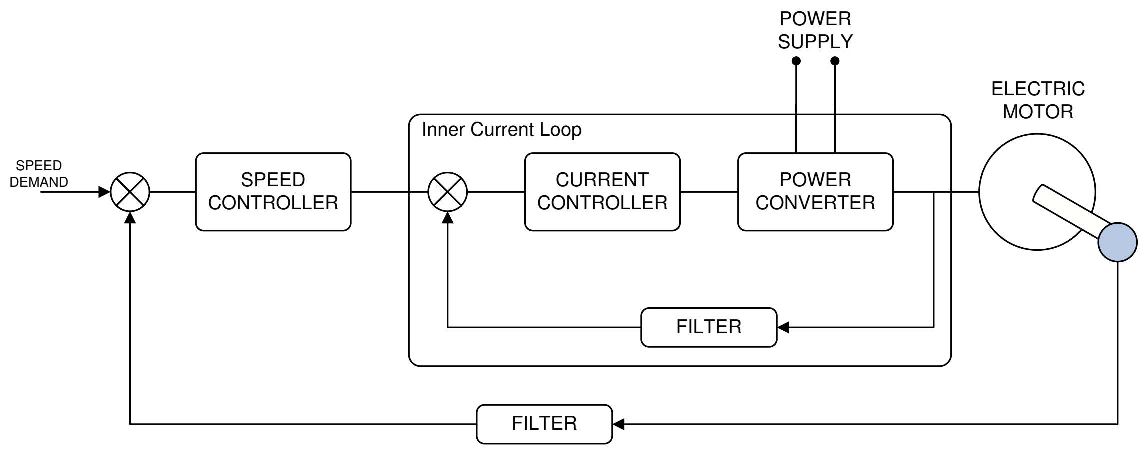


Figure 1.1: Overview of speed and current controlled variable speed electric drive

an average voltage (driving force) for the electric motor. Figure 1.1 represents a speed and torque/current controlled electric drive. The following sections will provide an overview on the controlled drive system, initially focusing on the electric motor and broadly highlighting the different types. Then, the power converters generally used within the drives will be briefly described and finally, a description of the controllers, required for the efficient operation of the drive, will be presented.

1.1.1 The Electric Motor

Electric motors convert electrical to mechanical energy. They play an important role as electromechanical energy converters in very significant manufacturing operations: transportation, material handling and most production processes. The ease of controlling these motors provides a significant property that can be taken advantage of in meeting ever-increasing user demands that include the need for flexibility and precision, which is often the result of advances in technology within industry. Also, the need for energy conservation in order to safe-guard the environment is a key driver for improved efficiency in the employment of variable speed electric drives [3], [4].

The electric motor, which simply consists of an arrangement of copper coils and steel laminations, is clearly a simple, but rather clever energy converter: the basic principle behind its operation exploits the force which is exerted on a current-carrying conductor placed in a magnetic field. In most cases, and regardless of the type, electric motors consists of a *stator* and a *rotor* or armature. Within the electric motors, the arrangement is such that there exist a strong magnetic field which interacts with many conductors, each carrying as much current as possible in order to produce as much force as is required to drive different applications [4]. To avoid going into too much detail on the very popular motors, a broad classification on the different types of electric motors will be provided. Under this classification, four types of electric motors can be identified: dc motors, ac induction motors, synchronous motors and special motors [5].

The DC motor is widely used due to its simplicity in construction and the ease with which it can be controlled. For these reasons, in the 1960s, the DC variable speed drives were the preferred choice for industrial applications [1]. Nowadays, although the AC machines are almost entirely the preferred choice, the DC machines are still valued for their wide speed and torque range and also their overall efficiency. There are three main types of DC motors: (1) series-wound including permanent-magnet and separately-excited motors (2) shunt-wound and (3) compound-wound. This classification is made according to the way in which the magnetic field in a DC motor is created [6]. The DC motor has a wide application range that includes machine tools, traction hoisting, and robotics, *etc.* The rapid advancements in power electronics and control strategies further enhanced the overall applications of conventional DC motors. The mechanical commutation process within the DC motor, which enables the flow of current in one direction within its current carrying conductors, pose some limitations to the functioning of the conventional DC machine. The results of this are a cap on the maximum possible ratings on the DC motors and also the development of design strategies that focus on ways to avoid the employment of commutators and its associated components in future DC machine designs. The commutators are prone to frequent failures, which could be responsible for the breakdown of the equipment. Apart from the inherent limitations on its maximum ratings, sparks, which is often the result of commutation within the DC commutator motor, limits its usage within hazardous environments [5].

"*The induction motor is the universal workhorse of modern industry*" [5], [6]. During the Late 1960s, the rapid advances in semiconductor technology for power and control applications enabled the AC drives contention with the DC Drives as the preferred choice for industrial applications [1]. Currently, the induction motor drive, which is a prominent member of the AC drives family, is the industry favourite. The squirrel-cage induction motor is particular noted for having certain desirable qualities of being economically affordable, reliable, and for these reasons it has become touted for most industrial applications; they have become widely available worldwide in mass scale production. Older models of induction motors were bulky and less efficient due to poor quality of previously available construction materials and lack of adequate analytical and design tools. The newer models are less bulky and, with the levels of technological advancement in this current age, significant energy conservation can be achieved with the high-efficiency class induction motors that often have an associated low cost [5].

Conventional synchronous motors are well-known industrial drives. The operation demands that they produce constant speed regardless of the load conditions they are operating under. Modern synchronous motors are brushless and they are widely used in low and medium rating industrial drives. For lower ratings, permanent magnet (PM) and reluctance motors are receiving wide attention. Because of high efficiency, simpler construction and control etc., the permanent magnet (PM) types are finding more favour than the reluctance motors [3], [5].

The special motor covers a wide spectrum of unique electric motors and moving devices. These motors are so called due to either the peculiar constructional features or their unusual applications. Some of these motors are the permanent magnet, stepping, switched reluctance, reluctance-permanent magnet and hysteresis motors [5].

A wide range of different types of electric motor can be found in many modern day applications. These could range from linear to rotary motions with the employed electric motors having ratings ranging from microwatt to megawatts. The basic purpose of an electric motor is to harness electric energy in doing mechanical work. The end product of such a conversion within the electric drives, the mechanical power is eventually used to drive almost anything; most devices that move are almost always powered by an electric motor. Such devices range from a cassette recorder or a computer disk or a fan to a vast rolling mill. The evidence of the wide usage and importance of the electric motors is highlighted by the statistics that about 65-75% of the world's total electrical energy is consumed by electric motors [1]. For these reasons, reliable and efficient electric motors form a significant focus and is, quite literally, a central driver to our modern way of life [2].

1.1.2 The Power Converters

The Converters are essential for the functioning of a drive system. They are generally required for the conversion of electric power to the form and value required by the supplied load. Given the impracticality of instantaneously stepping up and down the supply voltage to meet the immediate requirements of the connected load, the viable alternative is to alternately connect/disconnect the load from the supply by turning on/off switches which connect the load to the supply.

Power converters are made up of high-power fast-acting semiconductor devices, such as bipolar junction transistors (BJT), metal oxide semiconductor field effect transistors (MOSFET), insulated gate bipolar transistor (IGBT), silicon controlled rectifier or transistors (SCR), gate turn-off SCR (GTO) and MOS-controlled thyristor (MCT). These solid-state devices are configured in a certain circuit topology function as an on-off electronic switch to convert the fixed supply voltage into variable voltage and variable frequency supply. Significant advances in power semiconductor technology over the past two decades has enabled the development of compact, more efficient power electronic converter topologies with significantly improved reliability [3], [4].

The different converter types are broadly classified based on the types of input and output voltages of the converter. Given there are two voltage types,the DC (Direct Current) or AC (Alternating Current) voltages, the different classes of converters are: DC-to-DC Converters (Choppers), DC-to-AC Converters (Inverters), AC-to-DC

Converters (Rectifiers), AC-to-AC Converters (Cyclo-converters, Matrix converters). These converters are used, within the electric drive, according to the requirements of the electric motors and their applications.

1.1.3 Control of Electric Drives

Generally, the efficient operation of variable speed electric drives is largely governed by controllers - the more efficient the controller, the more effective the drive will be in carrying out its assigned function. Closed loop control of electric drives has not always been available and accepted for industrial applications. In the 1950s, the control of drives was normally open-loop and as a result motor speed was affected by changes of load. This meant that there was a limitation in the possible speed range because at low speeds, the motor would stall when the load was applied. But, by the late 1950s, the closed loop control of variable speed drives had gained wide industry acceptance, especially because of its proposed benefits of greater accuracy and faster response of variable speed drive systems, wider speed range of operation, ability to tune the drive system to suit the application, *etc* [7].

Within the modern day closed loop control system for variable speed drives, there exists an inner-current loop control and an outer-speed loop control [1]. Conventional controllers based on the Proportional, Integral and Derivative (PID) family are often the preferred choice for implementing the control of drives and they have served the industrial drives for decades. This is mainly because of their simple control structure, ease of design, low cost and because they sufficiently satisfy the requirements of many industrial applications [8], [9].

The actual tuning of a PID controller requires a good model (an abstraction of the real system) and certain design rules for its successful implementation. The tuning procedure can sometimes be a time-consuming and difficult task, above all, for complex plants. Some control engineers may take one to three days searching for a practically valid PID controller setting for certain industrial applications [8],

[9], [10]. Furthermore, it is also realised that many PID controllers for closed loop systems are poorly tuned. This could be due to the fact that the design is based upon approximations of the actual system dynamics - a model. A poorly tuned control system may waste energy and cause excessive and unnecessary wear of the plants within the closed loop control system. Although many improved PI/PID design methods are proposed, such as root locus, Ziegler-Nichols (Z-N) methods *etc*, they have shortcomings such as long testing time and limited control performance [8].

Despite the satisfactory performance of these controllers for some industrial applications, they fail to perform to the same standard under condition with significant mechanical load variation, nonlinearity, load disturbance; in other words, in such situations, these controllers are not robust. Given the ever-increasing demand for efficient performance of electric drives in industrial applications, measures must been taken to ensure their robustness and reliability. The field of control that deals with the design of such robust control systems that provide the most efficient performance under uncertain conditions is known as Robust control [11] and its principles have been adopted in fulfilling the project aims.

1.2 Robust Control Design

The problem of robust control, which involves designing control systems in the presence of plant uncertainties, is classical. The thesis focuses on providing easily implementable solutions for this problem. This is made possible with the incorporation of the Evolutionary Algorithms (a branch of Artificial Intelligence). The designed solution focuses on the design of robust digital controllers for variable speed drives.

In order to simply demonstrate the effectiveness of the proposed control design techniques, a Variable Speed Permanent Magnet DC motor drive has been chosen. The developed methodology can then be easily extended to other types of electrical drives. The problems addressed in the thesis focus on two approaches for robust control design - the experimental and identification-based approaches.

1.2.1 Robust Experimental Control Design

There have been very recent efforts successfully implemented for the design of robust controllers directly on the actual drive while it is operating under load conditions. Model-based techniques mainly rely upon the accuracy of the model used in the simulations. But, these models are only approximations of the real system and in most cases, the models are often simplified, as they neglect actual nonlinearities and uncertainties that exist in the real system. This could be due to the computing limitations of the employed processor or unavailability of modelling techniques to adequately characterise certain observed or unobserved system phenomena [12], [13], [14].

In the case of direct design, the effects of the actual and unknown *a priori* high-order phenomena, nonlinearities and uncertainty are fully accounted for in the design procedure and the resulting controller solution can be immediately harnessed on the actual system with known performance. These robust controller design techniques permit access to design tools that do not require skilled expertise for system modelling, or complex design methods for controller optimisation. On the other hand, in spite of the simplicity of the basic idea, direct optimisation strategies require dealing with several challenging problems initially before they can be successfully implemented. These problems have strongly limited the number of successful applications reported within this field [12], [13], [15], [16].

The attractiveness of the direct design procedure lies in the simplicity of the idea that underlies its implementation. Although there are problems that come along with the process, overcoming these obstacles would yield a method that is incredibly useful for the drive commissioning process, easily adaptable and capable of lending itself well to several commercial applications. The research work presented here will focus on overcoming the problems of implementing an experimental control design method and also setting up a direct design approach that employs evolutionary algorithms in an automated control-optimisation process for variable speed drives while they are subject to variable mechanical loads.

1.2.2 Robust Identification-based Control Design

Over the past twenty years, there has been significant drive for the development of new theory that can be employed in deriving solutions for robust control problems and the term robust control for this classical problem is only of recent vintage [11]. Currently, there is ever-increasing focus in formulating more methods for tackling such problems. The main issue related to this new area of research is model uncertainty. It is known as a fact that real physical systems can only be described by means of an uncertain model, and Robust Control theory has quantified the influence of this uncertainty in the stability and performance of control systems [11], [17], [18].

A number of theoretical approaches to the control of uncertain systems have been developed. Some of these methods include: frequency domain approach like the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ design methods, stochastic domain approach, game-theoretic or minimax approach, guaranteed-cost control, adaptive control methods, parameter estimation, Lyapunov-function theory, sliding-mode control (SMC), qualitative-feedback theory (QFT), Hurwitz-condition approach, norm-uncertainty approach, *etc.* These methods have many benefits and have been shown to be effective for different applications. But it is not all rosy: often with these robust control methods, the focus is on complex control structures which increases the difficulty of implementation. Also, given the specialist knowledge required for its development, it can demand significant user interaction and time consuming implementation.

The drives manufacturers are reluctant to incorporate control algorithms which require specialist skills for commissioning the control parameters, preferring to stay with traditional controllers such as Proportional plus Integral (PI), due to the ease of understanding and the ability to commission an adequate controller quickly. Hence, despite its advantages, these modern methods are often sacrificed in favour of traditional control methods [19]. Implementing a simple (automated) strategy with which to efficiently apply such theoretical robust control methods in conjunction with system identification techniques would prove invaluable for the control engineers regarding time to achieve control solutions and quality of these final solutions. This work will in fact develop an automated theoretical control design approach based on the $\mathcal{H}_\infty$ theory and a system identification process using evolutionary algorithms.

1.3 Project Objectives

In light of the discussion, the prime objectives of the research project will focus on developing simple but commercially attractive and viable methods for the purpose of designing robust control systems. The developed methods both employ particular evolutionary algorithms. The first technique considers the design of robust control systems that involves performing tests directly on the experimental system. The second approach is less direct and instead employs the robust control theory along with novel system identification methods in achieving its aims. The objectives of the thesis can be summarised as follows:

1. Develop particular Evolutionary Algorithms that can enable automation within the robust control system design procedure

2. Develop a novel approach for the automated direct design of robust controllers for variable speed electric drives that employs the developed evolutionary algorithms in the experimental design process.

3. Develop a novel approach for the automated design of robust controllers for variable speed drives that which employs the robust control theory, a novel system identification procedure and the evolutionary algorithms in the design process.

4. Provide a comparison of the developed evolutionary algorithms adopted to achieve the automation in the design procedure.

5. Provide a comparison of the different design techniques developed to achieve the design of robust control systems.

1.4 Thesis Outline

The thesis has been structured in the following manner:

Chapters *Two* and *Three* focus on providing a general framework for implementing the automated design tool, the evolutionary algorithms, and a detailed description of the particular evolutionary algorithms investigated during the research project.

Chapter 2: Describes the Evolutionary Algorithms and highlights the reasons why it is the preferred choice. It also gives a general overview of their implementation and highlights the different areas these algorithms have been successfully applied to.

Chapter 3: Describes the implementation of the particular algorithms deployed for the purpose of achieving the research objectives. The investigated algorithms are the Genetic Algorithms, Bacterial Foraging and Hybrid Bacterial Foraging.

Chapters *Four* and *Five* focus on the general description, hardware design and the conceptual implementation of the Experimental System

Chapter 4: Describes the general overview of the Experimental System used to test the effectiveness of the robust control design strategies developed during the research work. It also describes the experimental system's hardware design by initially characterising its specific components and finally illustrating how they come together.

Chapter 5: Describes an integral portion of the experimental system - the mechanical emulator system. The chapter explains the concept of the emulator system and illustrates how it is incorporated within the experimental system *.i.e.* the closed loop control system of the variable speed drive. The chapter also describes the nature of the controller to be optimised and provides results that show a working mechanical emulator system.

Chapters *Six* and *Seven* represent the culmination of the fervent efforts of the research work. The chapters describe the novel robust control system design procedures that result from applying the investigated evolutionary algorithms to the developed experimental system.

Chapter 6: Provides a detailed description of the Robust Experimental Control Design procedure. The novel design technique is achieved through the direct application of the investigated optimisation algorithms to the experimental system hardware. It also provides the results achieved by employing each of the different evolutionary algorithms during the direct design procedure.

Chapter 7: Provides an exhaustive description of the Robust Identification-based Design procedure. It describes the frequency domain approach and how it is employed in conjunction with the evolutionary algorithms and system identification methods to achieve a novel robust control design method. It also provides the results achieved through the adoption of the different evolutionary algorithms for the design procedure.

Chapter *Eight* provides insightful analysis and discussions on the evolutionary algorithms investigated and the novel robust control system design techniques employed

Chapter 8: The discussions within the chapter focus on: (1) The comparison of the evolutionary algorithms employed, which enable automation in the design procedure and (2) The comparison between the novel robust identification-based and experimental control design techniques. The chapter also concludes the thesis by providing recommendations on design strategies and identifies potential directions for future research.

Chapter 2

Evolutionary Algorithms

2.1 Introduction

In nature, achievement of an optimal state is necessary for continuous existence of objects, both living and non-living. This is evident in the efficient and optimal bond arrangements found within elements and their atoms that form the basic building blocks of life as we know it. Similar evidence is cited in different living species in their natural observation of the principle of *survival of the fittest*, together with biological evolution, in order to better adapt to their environment. In this case, a well-adapted species that dominates all other species in its surroundings can be suitably referred to as an optimal solution in its locale [20].

Recently, due to the complex nature of many emerging problems within the field of science and engineering, computing the global optimal solutions to such problems require more novel and advanced techniques. The use of classical gradient-based programming techniques may fail to solve such problems because they usually contain multiple local optima. In recent years, there has been a great deal of interest in some emerging artificial intelligence tools in the area of optimisation. These tools have the proposed benefits of successfully identifying the global optimal solutions to optimisation problems with multiple local optima. They are classed as Global Optimisation methods. As a result of their proposed benefits, global optimisation methods are increasingly being invoked to deal with emerging problems within the field of science and engineering [12], [13], [20].

Global Optimisation is a branch of applied mathematics and numerical analysis that deals with the optimisation of single or multiple criteria. These criteria are expressed mathematically as *objective functions*. The result of such a global optimisation process is the set of inputs for which these objective functions return the optimal values [13], [20], [21]. The algorithms (step-by-step logical instructions) implemented for the purpose of achieving such results are termed Optimisation Algorithms.

Optimisation algorithms are capable of and are largely employed in finding a global optimum solution to different optimisation problems. The different criteria that determine the optimum solution are specified within the objective functions. The optimisation algorithms search to find the set of inputs to the objective function that results in optimum outputs. These optimisation algorithms can be broadly grouped, according to their method of operation, into two categories: *deterministic* and *probabilistic* algorithms. Deterministic algorithms behave as they are defined; for any particular input, the same output is always produced. There are no alternate outcomes for the implementation of a deterministic algorithm. If no way to proceed exists, the algorithm terminates. These algorithms incorporate no procedures that involve the use random numbers in deciding their output. Examples of deterministic algorithms include State Space Search, Branch and Bound and Algebraic Geometry [20].

For many emerging science and engineering issues, deterministic algorithms are unfeasible. This is because the sample space for possible solutions to such problems is very large and there is uncertainty in the relations between the inputs and corresponding outputs. This suggests that a sample space cannot be partitioned in any wise fashion and as such, an exhaustive search of the sample space is necessary in order to determine which gives the best solutions. For this reason, probabilistic algorithms are employed in such cases. These algorithms employ a degree of randomness in their logical (step-by-step) sequence violating the constraints of the deterministic algorithms [20], [22]. Figure 2.1 shows the overall classification of probabilistic optimisation algorithms.

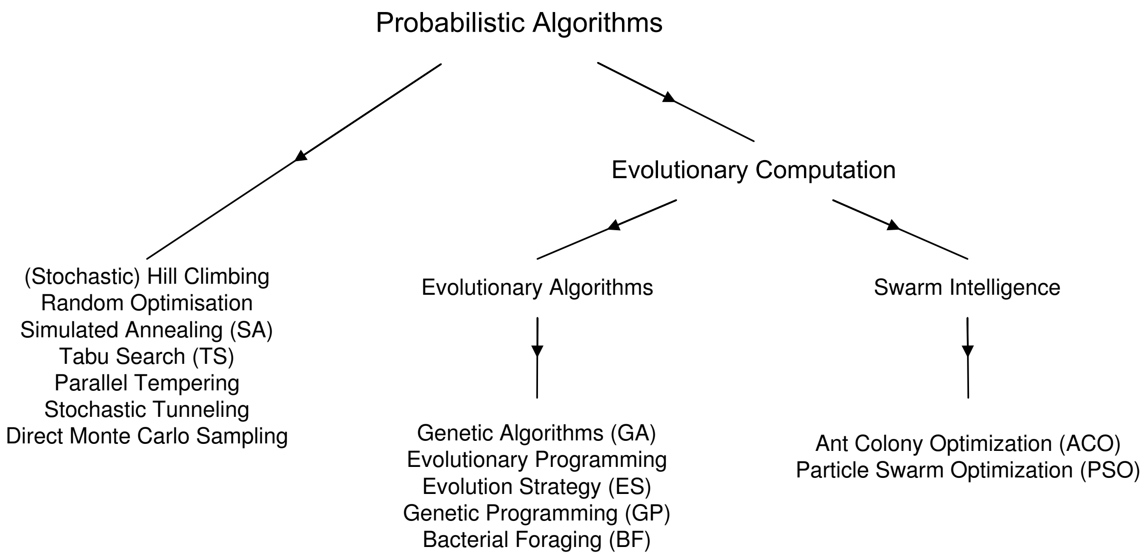


Figure 2.1: Classification of Probabilistic Optimisation Algorithm

In general, any task can be solved by a process that involves searching through a sample space of possible solutions and identifying which solution(s) yield the required output; such a systematic approach is classified as an optimisation process. For small spaces, classical exhaustive methods are usually preferred; for larger spaces, more efficient methods that may employ artificial intelligence must be employed. The methods of evolutionary computation are among such techniques; they are stochastic algorithms whose search methods model some natural phenomena: Darwin's genetic inheritance and Mendel's selection principle. Evolutionary Algorithms (EA) are the best known class of evolutionary computation methods [20], [23]. They are also an important class of probabilistic algorithms. They encompasses all such algorithms that are based on a set of multiple solution candidates (called population) which are iteratively refined using approaches based on biological theories of evolution. This field of optimisation is also a class of soft computing as well as a part of the area of artificial intelligence. Its roots are embedded deep within four landmark approaches: evolutionary programming (EP), evolution strategies (ES), genetic algorithms (GA) and genetic programming (GP). The genetic algorithm was popularised by Goldberg and the majority of control applications in the literature adopt this approach. *GP* is, perhaps, the next most popularly used method [14], [20], [23]. The following describes the more popular evolutionary computation techniques in some detail:

Genetic Algorithm: A *GA* performs a multi-directional search through a sample space of potential solutions. During the process, it maintains the population of potential solutions but randomly creates new solutions within this population. The creation of new solutions is achieved by the adoption of certain operators (mutation, crossover, selection, reproduction), which mimic the process of natural selection and evolution. This evolutionary approach has been adopted in achieving the aims of the research project and more detail is provided in section 3.2.

Evolution Strategy: Evolutionary strategies (*ES*) were developed in the mid 1960s by Bienert, Rechenberg, and Schwefel at the Technical University of Berlin, Germany as a method for evolving optimal shapes of minimal drag bodies in a wind tunnel using Darwin's evolution principle [24]. The earliest evolution strategies were based on a population consisting only of one individual. Through the years, the algorithm has been developed and it now has the ability to include several individuals. In evolutionary strategies, the representation used is a fixed-length real-valued vector. This bears similarity with the GAs in that each position in the vector corresponds to a feature of the individual [24].

Beginning with an initial parent generation in each iteration, a child generation is generated by randomly modifying the parent parameters. The main reproduction operator in evolution strategy is *Gaussian mutation*, in which a random value from a Gaussian distribution is added to each element of an individuals vector to create a new offspring. Another operator that could be used is *intermediate recombination*, in which the vectors of two parents are averaged together, element by element, to form a new offspring.

After the mutation step the objective function is evaluated for all children and the best of them are selected to form the new parent generation. With μ representing the number of parents and λ representing the number of children, strategies of selection and transmission can be distinguished in the following way [25], [26].

1. $(\mu + \lambda)$ - *plus* strategy: the next generation is selected from μ parents and λ offsprings.
2. (μ, λ) - *comma* strategy: the μ best parents out of the λ offsprings form the next parent generation, none of the parents survives.

Evolutionary Programming: This was developed by Fogel, Owens and Walsh in the mid 1960s. Their book, "Artificial Intelligence Through Simulated Evolution" formed an important marker for the widespread implementation of EP applications [27]. The representations used in evolutionary programming are typically tailored to the problem domain. One representation commonly used is the fixed-length real-valued vector. This is similar to the representation method for the GA and ES. The primary difference between evolutionary programming and other evolutionary computation approaches is that no exchange of material between individuals in the population is made. Thus, only mutation operators are used.

For real-valued vector representations, evolutionary programming is very similar to evolutionary strategies without recombination. A typical selection method would involve selecting each individual within the population as the N parents. For the reproduction phase, each parent is mutated to form N new individuals (offsprings). Finally, a random selection of N individuals from the new population that comprises parents and their offspring is implemented to form the new generation [23].

Genetic Programming: This is an interesting approach developed by Koza and it is becoming an increasingly popular technique. GP deals with the issue of how to automatically create a working computer program for a given problem from some initial problem statement. The GP achieves its aim of finding a global optimal solution by breeding populations of computer programs in terms evolution and the principle of natural selection, which is essentially based on the concept of survival of the fittest [28]. In *GP*, a computer program is often represented as a tree (a program tree) where the internal nodes correspond to a set of functions used in the program and the external nodes (terminals) indicate variables and constants used as the input to computer program (functions). The set of all functions and terminals is selected *a priori* in such a way that some of the composed trees yield a solution [23].

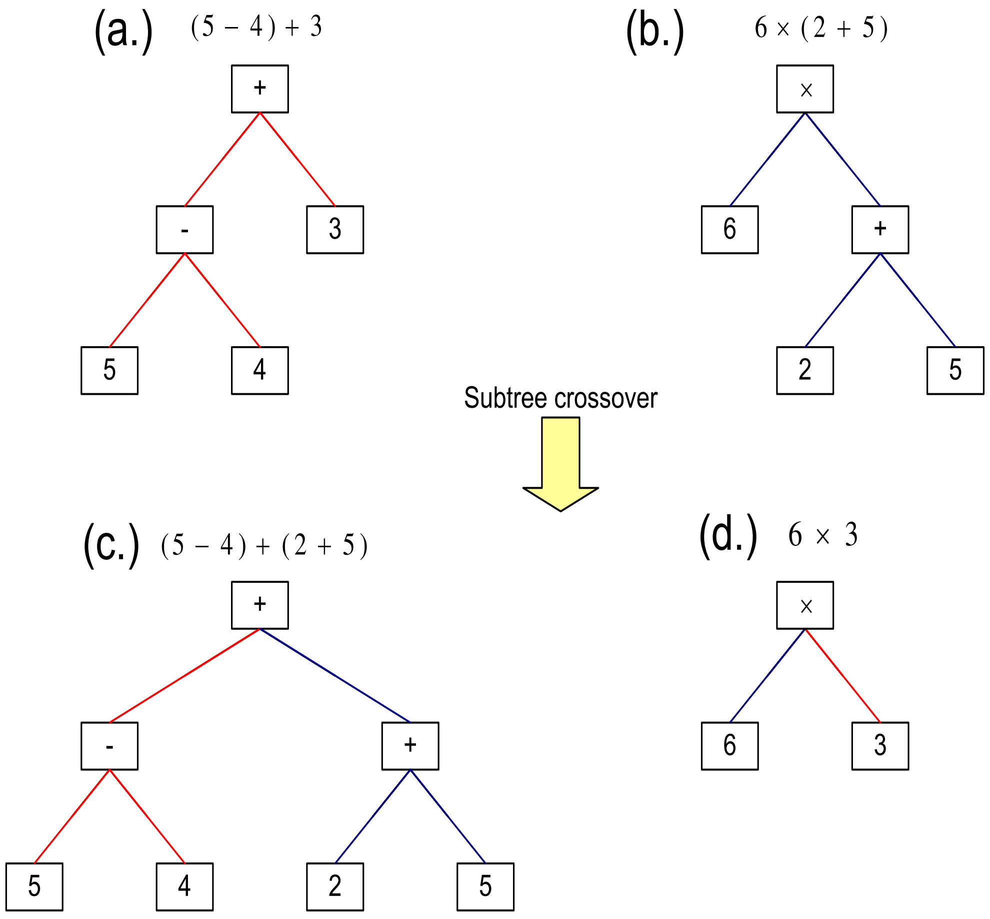


Figure 2.2: Subtree crossover of parents (a.) and (b.) to form offspring (c.) and (d.)

In a standard genetic program, the representation used is a variable-sized tree of functions and values. The objects that constitute the population are not fixed-length character strings that encode possible solutions to the optimisation problem but they are computer programs (functions) that, when executed, provide the candidate solutions to the problem. Genetic algorithms and genetic programming are similar in most other respects, except that the reproduction operators are tailored to a tree representation. The most commonly used operator, illustrated in figure 2.2 is *subtree crossover*, in which an entire subtree is swapped between two parents [14], [23].

Bacterial Foraging: Bacterial Foraging is a new evolutionary computational method proposed by Kevin Passino[29] in 2002. In this scheme, the foraging (methods for locating, handling and ingesting food) behaviour of E. coli bacteria present in our intestines is mimicked. The approach incorporates concepts of evolution and natural selection and as such it is suitably regarded as an evolutionary algorithm. Although it also incorporates concepts of Swarm Intelligence, the fundamental idea behind the algorithms development is based upon the natural selection of bacteria with good foraging habits and the evolution of better strategies using the good foragers as the parents. They undergo different stages such as chemotaxis, swarming, reproduction and elimination and dispersal. This evolutionary approach has also been adopted in achieving the aims of the research project and more detail is provided in section 3.3.

In the last decade, within the field of control and instrumentation engineering, evolutionary algorithms have been receiving increasing attention because of their potential to deal with problems not amenable to existing design techniques. Through the adoption of EA, there are more novel techniques being developed that are aimed at designing more efficient controllers for variable speed drives and a number of such EA design techniques have also been successfully implemented for such applications. In [12] and [16], automated design of controllers for variable speed drives have been investigated.

Although the conditions under which the method is investigated are not typical dynamic operative conditions for electronic drives, the results obtained show the ease and effectiveness of employing these optimisation algorithms. Also, the area of system identification by employing these algorithms is currently an active area of research [13]. In this chapter, a structured approach for the implementation of evolutionary algorithms will be presented and subsequently, its different applications will be discussed. But first of all, a highlight will be made as to why the evolutionary algorithms are the preferred choice.

2.2 Preference of Evolutionary Algorithms

The evolution of modern man through the ages and his inherited ability to adapt suitably to his environment is adequately perceived by most as an optimisation process. Researchers have adequately adapted this simple, but powerful, concept and applied it to real-world problems. The quote from [30] supports the widespread adoption of this concept for solving complex problems: "*it is quite natural, therefore to seek to describe evolution in terms of an algorithm that can be used to solve difficult optimisation problems. The classic techniques of gradient descent, deterministic hill-climbing and purely random search (with no heredity) have been generally unsatisfactory when applied to nonlinear optimisation problems, especially those with stochastic, temporal or chaotic components. But these are problems that nature has seemingly solved so well. Evolution provides inspiration for computing the solutions to problems that have previously appeared intractable*". The section highlights the few clear advantages on why Evolutionary Algorithms are the preferred alternative for computational algorithms in dealing with the emerging problems within the field of science and engineering.

In reality, most of the real-world problems do not have easily decipherable relations between the problem domain and its corresponding possible solutions [14]. This could be due to the noisy nature of the problem or general uncertainty within the problem domain that cannot be easily modelled through existing methods. In such a situation, a logical approach would be to perform an exhaustive (point-to-point) search of the solutions domain. Such a procedure would be inefficient and even more so if the solution domain is extremely large. This implies that traditional (gradient-based) methods would be ineffective in these extreme conditions. But, the EA approach are inherently suited to such extreme environments and, given that the nature of emerging problems within the science and engineering field are becoming characterised by such extreme conditions, EA are excellent alternatives to tackling these novel problems.

Generally, in the process of solving a problem, it is necessary to understand and characterise the behaviour of the problem. Likewise, in providing a control solution for a system, a vital initial step would be to model the system to be controlled. Having obtained a model, this would provide the platform for the next step, which is achieving the final control solution for the system. In deriving a model for the system, practitioners adopt either:

1. a simple linearised model of the system
2. a more complex (faithful) model of the system

The global optimum solution for the linearised model can be obtained using traditional methods. For the complex models, adopting a EA strategy are the more efficient alternatives in finding an optimal solution. Given the complex nature of these models, the global optimum may not be discovered but only an approximate optimal solution. There is strong support for the more superior quality possessed by approximate solutions for complex (faithful) models when compared with the global solutions attained using simple (less faithful) models [14], [30]. This highlights the usefulness of evolutionary algorithms, which are inherently suited in attaining near-optimum solutions for complex, real-world problems.

Critiques for the EA are often directed towards the number of parameters that need to be defined at initialisation. These parameters include the maximum number of generation, population size, probability of occurrence of selection and reproduction operators *etc.* The benefits that underlies this critique is, given the many parameter definitions required for initialisation, some wrongly defined parameters could still lead to a fairly successful EA search results. In other words, the many initialisation parameters enhances the robustness of the EA because "*even some obviously wrong parameter values do not prevent fairly good results*" [14].

Another very interesting application of the EA, solutions to which were not previously achievable with traditional methods, is temporal optimisation [14]. This process involves immediate identification of the system parameters based upon instantaneous inputs to the system. Such problems are regarded as being very difficult probably due to the instantaneous periodic focus and the demands of quality approximations of the behaviourial strategy for the system within a small time window. The EA are suited to such applications because they can easily adapt to changes in the problem domain. Most traditional approaches require a restart of the algorithm due to the difficulty in adapting to the changing instantaneous conditions [14], [30].

2.3 Implementation of Evolutionary Algorithms

Challenging optimisation problems, which elude acceptable solutions via conventional methods, arise regularly in control systems engineering. Evolutionary algorithms (EA) permit flexible representation of decision variables and performance evaluation and are robust to difficult search environments, leading to their widespread uptake in the control community.

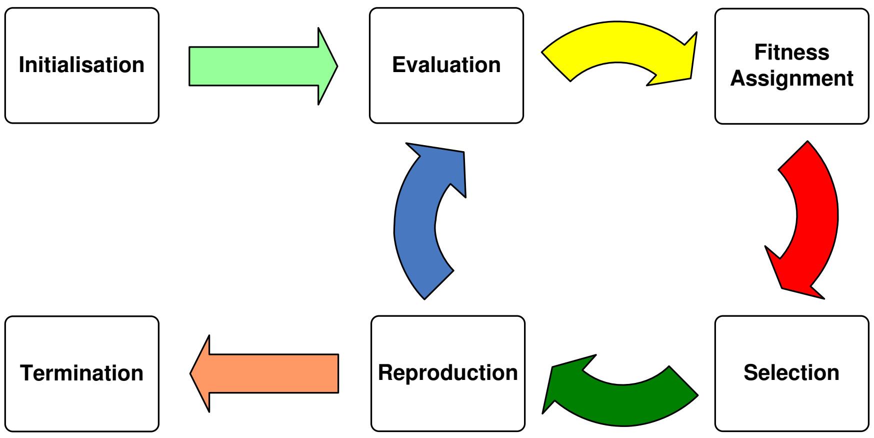


Figure 2.3: Structured Implementation of Evolutionary Algorithms

EA are global and parallel search and optimisation methods that are founded on biologically inspired mechanisms like mutation, crossover, natural selection and survival of the fittest. The solution candidates of a certain problems play the role of individuals. Their fitness is rated according to objective functions whose improvements form the aim of the optimisation and drive the evolution in specific directions. The advantage of evolutionary algorithms compared to other optimisation methods is that they make only few assumptions about the underlying fitness landscape and therefore perform consistently well in many different problem categories [20]. Generally, the implementation method for the EA can be summarised in the following steps: Initialisation, Evaluation, Fitness assignment, Selection and Reproduction. The structure is represented in figure 2.3.

2.3.1 Initialisation

Population initialisation is a crucial task in evolutionary algorithms (EA) because it can affect the convergence speed and also the quality of the final solution. Although this fact is known, reports within this research field are very few. Certain approaches have been considered for improving the initialisation process that is a necessary step for the EA implementation. Two of the more significant approaches will be described as follows:

Quasi-random method: This involves generating numbers with a "perfect uniform distribution". Each successive point generated via this method is equally spaced from the previous point. As a consequence of these equally spaced points, individuals within the initial population would be evenly distributed within the sample space. Traditionally, numbers generated 'randomly' for the initial population tend to mimic random numbers. They are not truly random (or independent) as they are developed algorithmically. Hence, they are termed *pseudo-random* numbers. These numbers tend not to be evenly spread around the sample space and form clusters in some portions of the sample space [31].

The fact that the *pseudo-random* number distribution does not encompass the sample space of solutions suggest it is less likely to discover the optimum solution using this initialisation method. It has been reported that employing quasi-random method of initialisation can improve the quality of final solutions without significantly increasing the speed of convergence of the algorithm to the best possible solution [31]. For higher dimensional problems, their advantage pales into insignificance because it becomes more difficult to generate quasi-random sequences [32], [33].

Opposition-based learning method: This is a novel initialisation approach that employs opposition-based learning to generate an initial population. Generally, evolutionary optimisation methods are initialised with some solutions (initial population) and the aim is to improve performance towards some eventual optimal solutions. The process of searching terminates when certain user-specific criteria are met. In the absence of *a-priori* information about the solution, a random guess is useful in defining the start point for the optimisation process. The computation time required in discovering the best solution could depend on the distance of the initial guess from optimal solution. We can improve our chances of finding the optimal solution and reduce the time to find the optimal solution by starting with a closer (fitter) solution. This is achieved by checking the opposite guess of the initial guess simultaneously. By doing this, the closer one to the solution (say the guess or opposite guess) can be chosen as the initial solution [33], [34]

Definition - Let x be a real number in an interval $[a,b]$ $(x \in [a,b])$; the opposite number, $\bar{x}$ is defined by (2.1).

$$\bar{x} = a + b - x$$

Conducted experiments over a comprehensive set of benchmark functions, reported in [32], demonstrate that replacing the random initialisation with the opposition-based population initialisation can accelerate convergence speed for these cases. In summary, the basic idea behind the opposition-based learning is considering the estimate and opposite estimate (guess and opposite guess) at the same time in order to achieve a better approximation for current candidate solution. Unlike quasi-random number generation, the calculation of opposite candidates is not difficult or time consuming. Furthermore, there are no dimensionality limitations. Hence, the idea is applicable to a wide range of optimisation methods [33], [34], [35].

Although these proposed approaches have potential benefits, due to the ease of implementation and the well documented successes of the *pseudo-random* method of initialisation, this initialisation approach has been adopted within the EA employed in achieving the research project objectives. In future projects, it would be useful to investigate these other approaches and their benefits within particular applications.

2.3.2 Evaluation

The *evaluation* stage of the implementation of the optimisation algorithms is of utmost importance. This is because, by the evaluation of individual solutions using the objective function(s), the respective objective values are assigned to each individual (potential solution) within the population (sample space of possible solution). These objective values quantify the quality of possible solutions for the particular optimisation problem being investigated. It is a very vital function for the algorithms implementation as it is used in defining the best solution.

In the evaluation of the objective function, it is necessary to define this function and then calculate the corresponding objective value for the respective inputs. Most technical literature define the objective function with single indices, as integral time errors in systems' step responses (ITAE) or variants of this indices [15]. In recent literature, often underlined is the fact that control problems are, by their very nature, multi-objective, and also a single time-response based index may not be sufficient to define the desired specification of the controller design. In addition, the objective function can include any measurable, observable, or calculable behaviour or characteristics of the problem under consideration. Such characteristics of the problem may include, for example, the rise time, steady state error, overshoot etc. Hence, the choice of integral time errors seems not to fully capitalise on the full capabilities of evolutionary algorithms in some cases [12],[13].

Given that the objective function can take into account several aspects of the design problem (in the time and frequency domain), the objective functions can be either combined by incorporating different indices to yield a unique fitness for a particular problem or considered separately in a multiobjective optimisation problem [12],[13]. Certain Optimal controller designs have been achieved using a frequency domain approach - $\mathcal{H}_\infty$, as the objective function. In this case, the objective function is the sum of the squared robust stability and disturbance attenuation performance indices [14].

Calculating the objective function could be achieved through analysis and by adopting different formulae. It could also be achieved through computer simulations or by direct experimentation on the system to be controlled. The analytical formulae are generally obtained from linear models of the process and the controller, neglecting all the nonlinear effects for example hysteresis and voltage drops across brushes in electric drives. In such cases, using simple linear controllers, the integral-time-based performance indices (e.g., IAE or ITAE) can be computed through closed formulae of controller parameters. More frequently, the nonlinear effects assumed to be negligible are in fact not. In such cases, obtaining an accurate simulation model of the controlled process that accounts for the nonlinearities is vital. With such a simulation model, a direct approach to computing the fitness of the controller is to carry out closed loop simulations [12].

Recently, some authors have devised special techniques to implement evolutionary design strategies on-line. The obvious advantage of direct experimentation is the accuracy of the results obtained. All model-based approaches largely rely on the accuracy of the model used in the simulations or application of analytical formulae. The controller designed using direct experimentation already considers all possible nonlinearities and higher order disturbances during the design process; hence the optimal control from such a design procedure is ready for use directly after the experimental process is completed [12], [13], [14].

2.3.3 Fitness assignment

The fitness assignment process assigns a positive real number to an individual (possible candidate solution). The positive real number emphasises how fit the individual is relative to the other individuals within the population. This fitness assignment process is essential as it differentiates significantly candidates with very similar objective values. There are several fitness assignment strategies that have been described in evolutionary algorithms related literature. Some of the more common strategies will be described in the following sections [20], [36].

2.3.3.1 Rank-Based Fitness Assignment

Rank-based Fitness Assignment is also known as *Pareto ranking.* It assigns fitness values to individuals based on their rank within the population. The individuals are ranked according to the magnitude of their objective values. The purpose of the designed optimisation algorithms in this research project is to minimise the objective value *.i.e.* the best (optimal) solution should have the minimum possible objective value. Considering a pair of possible solutions for an optimisation task, the first possible solution, $individual_1$, is said to be *dominated* if the objective value it produces, after substitution in the objective function, is less optimal than that produced by the second possible solution, $individual_2$. Hence, the second possible solution is said to be *nondominated.* Also, if the objective value produced by the first possible solution is the same as that produced by the second possible solution, then $individual_1$ is said to *cover* $individual_2$ or vice versa.

Generally, the rank of an individual within the population depends on the number of individuals dominating the individual. While this method of ordering is a good typical approach capable of directing the search into the direction of the pareto optimal solution and delivering a broad scan of the solution space, it neglects the fact that the user of the optimisation is not often interested in the whole sample space of possible solutions but only has interest in selected regions [20].

2.3.3.2 Weighted Sum Fitness Assignment

Weighted Sum Fitness Assignment is reported as the most primitive form of fitness assignment. It simply involves assessing the fitness of an individual by summing the weighted objective values (assuming a multi-objective evolutionary algorithm). The weightings of the different fitness values are defined entirely arbitrarily depending only on the user's preferred specifications [20].

2.3.3.3 Niche Size Fitness Assignment

In the search for a global solution, the underlying idea is that most of the solutions domain is crowded by sub-optimal individuals (solutions) and around the global optimal regions, there are only few individuals. Niche size fitness assignment attributes a fitness value to each individual which reflects how crowded the niche of the individual is. The more crowded the niche of an individual is, the less fit the individual. Niche means the area in a given distance around the individual. The crowdedness then describes the count of other individuals inside that niche.

It is usually not wanted that niches are crowded significantly with all the possible solutions, since that would mean the optimisation algorithm converges fast. Instead, the preferred choice is many niches with few individuals inside, which means that many different solution patches are investigated and a broader scan of the sample space of solutions can be obtained. The niche fitness does not contain any information about the prevalence/dominance of the any selected solution over other solutions, it solely concentrates on assigning fitness values to individuals based on their niche [20], [37].

2.3.3.4 SPEA Fitness Assignment

The *Strength Pareto Evolutionary Algorithm* is a recently developed means for fitness assignment. With the optimisation of multiple objective evolutionary algorithms, the various objectives to be optimised can be conflicting: meeting the optimal solution, with a chosen set of parameters, of one objective function can result in a degraded solution for a another objective function. A *pareto-optimal* solution for a multiobjective optimisation problem is a set of solutions, consisting of certain parameters, with which the corresponding objective functions cannot be improved in any dimension without degradation in another. SPEA uses a mixture of new and established techniques in order to find multiple optimal solutions in parallel.

The fitness assignment strategy adopted in this approach initially requires a classification of possible solutions (within the sample space) on the basis of whether they are *dominated*, *nondominated*, *covered* - Pareto-ranking. Within any one generation, the nondominated individuals are stored in a Pareto set. This is updated after each generation passes in such a way that only a particular number of nondominated solutions are stored in the Pareto set. The fitness assignment for each individual solution within the Pareto set is based upon how many other individual solutions dominate and cover the individual solution under consideration. SPEA Fitness Assignment is not concerned with the individuals in the population within any one generation - a unique feature of this approach. The fitness value assigned to an individual solution is called its strength. The more the individuals that dominate and cover a solution, the less fit the individual for reproduction.

2.3.3.5 NSGA Fitness Assignment

Generally, pareto-based approaches use the objective functions to distinguish between the non-dominated and dominated solutions in the current population. The fitness assignment of an individual is based on the information from both the non-dominated and dominated sets. A ranking procedure entirely based on the non-dominated set to decide the fitness of the individuals was first suggested by Srinivas and Deb [36]. Based upon these ideas, the Non-dominated Sorting Genetic Algorithm (NSGA) was developed. NSGA Fitness Assignment is based on several layers of classifications of the individuals.

Before selection is performed, the population is ranked on the basis of non-domination: all non-dominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Sharing is achieved dividing the original fitness value of an individual solution by a quantity proportional to the number of individuals within a given distance around it. Then this group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the population are classified [20].

2.3.4 Selection

Selection essentially involves choosing a specified number of individuals from a list of candidate solutions for the mating pool, where reproduction will occur. The ultimate aim is to form a new generation of individuals that are potentially better solutions than their parents. The requirement for selection is that the fitter individuals have greater chances of survival than their weaker counterparts. The process of selection mimics nature and, in particular, the principle of natural selection, albeit in a simplistic manner, given the fitter individuals are more likely to be the determinants for the new generation. The process of selection may be implemented in a deterministic or a randomised manner, according to the specification of the user, which would normally depend on the application.

There are generally two classes of selection algorithms; those that occur *with replacement* and those that occur *without replacement*. In a selection algorithm *without replacement*, each individual from the input list is taken into consideration for reproduction at most once and therefore also will occur in the mating pool one time at most. The list of selected individuals , returned by algorithms *with replacement*, can contain the same individual multiple times. There are several types of selection method discussed in related literature [29], [38]. Some of the more common and important types of selection - Roulette wheel, Truncation, Rank, Tournament, Elitist - will be discussed in more detail in the following sections.

2.3.4.1 Roulette Wheel Selection

The Roulette wheel selection method is one of the oldest selection methods. It is also known as stochastic sampling. In this method, an individual's chance of being selected is proportional to its fitness relative to the other individuals [23], [39]. A roulette wheel is shown in the figure 2.4. Each portion represents an individual's fitness relative to the other individuals.

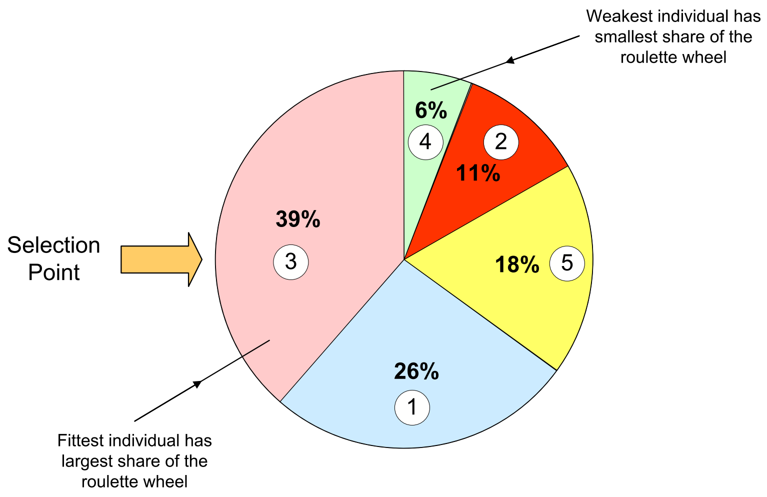

Weakest individual has smallest share of the roulette wheel
6%
4
2
11%
39%
Selection Point
3
18%
5
26%
1
Fittest individual has largest share of the roulette wheel

Figure 2.4: Roulette Wheel Selection

On each turn of the roulette wheel, an individual is selected at the selection point. It is evident from the roulette wheel that the fittest individuals (individuals having the largest portions of the wheel) are most likely to be selected on each turn. Hence, the fittest individuals will most likely form the members of the mating pool which is required in forming the new generation of solutions. The number of turns of the roulette wheel equals the number of individuals required in the mating pool.

2.3.4.2 Truncation Selection

Truncation selection does not attempt to emulate natural selection techniques. It is an artificial selection method in which the individuals are first ordered by their fitness and only a proportion of the fittest individuals selected are used in the mating pool for reproduction. The proportion of the fittest individuals selected is defined arbitrarily. This proportion indicates the proportion of the population that serve as parents and it is reported to range between 50 - 100% of the total population. Individuals that do not fall within the proportion do not reproduce. This method of selection promotes premature convergence and does not encourage diversity in generations of individual solutions [20], [40].

2.3.4.3 Rank Selection

With this method of selection, at each generation, the individuals in the population are sorted according to their fitness and each individual is assigned a rank in the sorted population. This method of selection is more useful than the Roulette wheel method in avoiding premature convergence in a situation where the fitness value of one individual is relatively much greater than the rest [40].

2.3.4.4 Tournament Selection

The Tournament selection is a two stage process. The first stage involves the selection of a group of individuals for the population. The group must consist of at least two individuals. The next stage involves selecting the fittest individual from the group. Once this is implemented, the unselected members can be discarded from further selection (*without replacement*) or they can be reinserted into the population for further selection (*with replacement*). This process is repeated until the mating pool is filled with the required number [20].

2.3.4.5 Elitist Selection

The Elitist selection strategy guarantees that the best individual generated will not be lost from one generation to the next as a consequence of sampling effects or the application of genetic operators. It simply involves directly passing the best solution of a previous generation to the current generation. This ensures that the algorithm always converges at the global optimum value, if it is found within successive generations.

There is the downside that there might be convergence to a local optimum value as a result of the elitist approach. Often, with the implementation of Evolutionary algorithms, the elitist selection strategy is combined with other selection strategies [20], [24], [41].

2.3.5 Reproduction

Reproduction is the process that produces a new generation of solutions from a current generation. It is the necessary step required to combine the selected individuals within the mating pool to form the offspring of solutions. The methods to achieve reproduction during the implementation of evolutionary algorithms can be broadly grouped into four reproduction operators - Creation, Duplication, Mutation and Crossover.

These operators can be implemented in a deterministic or a randomised way [20], [24]. The aim of the combined applications of these operators during reproduction is to ensure that the reproduced population are better than the current population; this is facilitated through inheritance of the best characteristics of the current generation by the reproduced generation.

2.3.5.1 Creation

The creation operator is used to generate a new solution that has no relation with existing solutions. This process is to generate initial solutions. These solutions can be generated in a random manner or, through experience, in a manner specified by the user. This is known as seeding the initial population, which can help increase the rate of convergence of the algorithm [42].

2.3.5.2 Duplication

The duplication operator is used to create an exact copy of an existing solution. This operator could be useful to increase the numbers of particular individuals with desired fitness values and, in turn, decrease the unwanted individuals in order to keep the total number of individuals constant within the population. The selective preference could enable convergence to the global optimum or conversely, encourage premature convergence to local optimum values.

2.3.5.3 Mutation

The mutation operator is used to generate a new individual by modification of an existing individual. Figure 2.5 illustrates the process of mutation on an individual that is binary encoded. During the mutation operation, first, the mutation point is selected. In the example in figure 2.5, the mutation point is on bit three of the individual. Second, the actual mutation is performed by merely changing the bit at the mutation point to its complement bit.

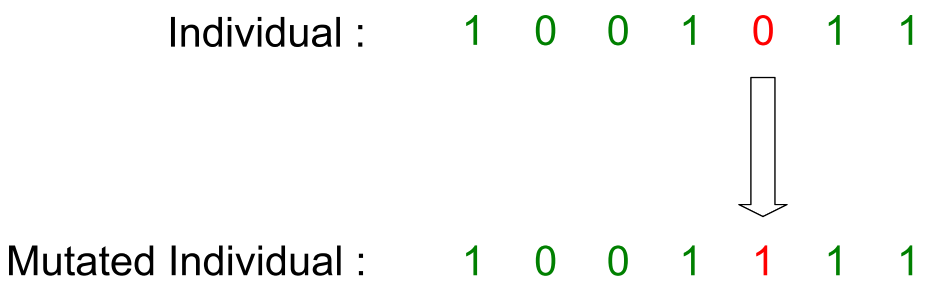

Individual : 1 0 0 1 0 1 1
Mutated Individual : 1 0 0 1 1 1 1

Figure 2.5: Illustration of the mutation

The process can be implemented in a stochastic manner by randomly selecting individuals and, in the same manner, picking the different positions within the selected individual to serve as the mutation point. It can also be achieved in a deterministic fashion, by specifically selecting particular individuals and ,in the same fashion, subsequently choosing positions on the individual as the mutation point.

2.3.5.4 Crossover

The crossover operator is used to create two new individuals by the combination of the features of two existing individuals. The process of crossover involves two stages: the first stage involves the selection of the crossover site. In the example illustrated in figure 2.6, which illustrates the principle of the crossover operator by considering binary encoded individuals, the crossover site is between bits three and four. The second stage involves swapping the bits that are to the right of the crossover site. The crossover operator can be applied in a random or deterministic manner.

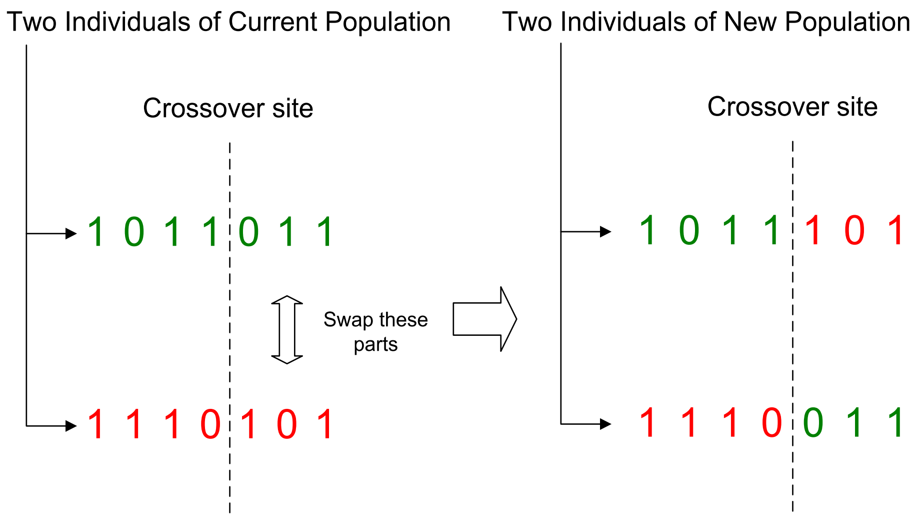


Figure 2.6: Illustration of the crossover

The four reproduction operators - Creation, Duplication, Mutation and Crossover are used in conjunction with one another to reproduce whole populations of individuals. The combined use of the reproduction operators ensures that the population experiences both divergence and convergence in varying magnitudes to ensure that, through generations, the algorithm eventually converges at the global optimum solution.

2.3.6 Termination

Evaluation, Fitness Assignment, Selection and Reproduction is repeated until the Termination criterion is reached. Classical termination criteria for the EA include

- the maximum number of generations
- the maximum number of function evaluations
- the maximum computation time
- the absolute global optimum value
- the convergence of the individuals within the population

A combination of these classical termination criteria are often used to bring the EA to a stop. At the point of termination, the EA should have arrived at the global optimum solution to the optimisation problem.

2.4 Applications of Evolutionary Algorithms

Applications in science and engineering requiring evolutionary algorithms are numerous. In order to create an awareness of the relevance of of Evolutionary Algorithms, its application in three key industrial sectors will be summarised. The considered sectors are Power (Energy and Electric Power), Transport (Aerospace and Automotive) and Communications (Telecommunications).

2.4.1 Energy and Electric Power

In the sector of energy and electric power, the focus is on energy generation. Some of the methods employed to achieve energy generation harness wind, water, fuel cells, solar energy, *etc.* The sector is also concerned with effectively distributing the generated energy to the consumers. As a result, there is a lot of work undertaken in the design of electrical distribution systems and control and power electronics systems to ensure safe, effective and efficient operation of the production facilities [43]. To highlight the effectiveness of the EA within this sector, its applications in the key areas within this sector will be summarised

2.4.1.1 Control Systems

Generally, in the design of control systems, three stages are employed: (1) Identification of the system (2) Designing the controllers for the modelled system (3) Evaluating the Robustness and Performance of the designed system [12], [13], [14], [15].

2.4.1.1.1 System Identification: In the absence of a model of a physical system, due to its variation with time and operating conditions, a means to obtain a mathematical model that considers both the structure and parameters of such a physical process will be beneficial. System Identification procedures provide a suitable method to achieve such a model. Given that many physical systems possess a time-varying nature and are dependent on operating conditions, dynamic system identification procedures are often necessary. For these reasons, System Identification that incorporates EA is currently an active area of research [13], [14].

Research and development of linear system identification has been studied for more than three decades. However, identification of nonlinear systems is an emerging topic of interest. Nonlinear characteristics such as saturation, dead-zone, etc., are embedded in the very fabric of many real systems. In order to analyse and control such systems, identification of nonlinear characteristics is necessary. Although nonlinear system identification is more challenging, given the usefulness of such identified models in designing more effective control strategies and the emergence of tools to achieve the more demanding identification process, it has been receiving increasing attention [13].

The process of system identification of linear and nonlinear systems generally involves two intrinsic problems: the *selection* of a suitable model structure and the *estimation* of model parameters. The very nature of the EA, which enables parallel evaluation of large solution space, makes the EA easily adaptable to identifying linear and nonlinear models for systems especially where there is no *a priori* knowledge of the possible structure and parameters for the model to be designed. A large number of practitioners have reported successful linear system identification and there is an ever-increasing number of successful identification of nonlinear systems [12], [13].

2.4.1.1.2 Controller Design: The applications for Evolutionary Algorithms in control system design can be classified in two main groups: The *offline* and *online* methods:

Offline method: This approach is adopted for the design of controllers using models of the systems to be controlled. In designing the controllers, the complexity of the process model could require equally complex controllers that demand expertise in designing their structure and parameters. Although the complexity of models can be sacrificed for simple linear models, the resulting control solutions might also have imposed quality limitations on its performance [15]. The application of evolutionary algorithms to offline controller design methods can absolve the need for expertise in designing the corresponding (complex) controllers required by the complex systems. This could potentially improve the quality of the achieved system dynamics. EA have been widely and successfully applied to off-line design applications.

Online method: In spite of the simplicity of the basic idea, the on-line application of evolutionary algorithms for optimisation requires dealing with several challenging problems, which have strongly limited, to date, the number of successful applications in this field. Online EA design techniques allow us to obtain automatic design tools that do not require skilled expertise for system modelling because it harnesses trial-and-error controller optimisation directly on the actual process to be controlled. The benefit of such an approach throws more certainty on the quality of the final solution obtained. Designing and developing a system capable of properly running an evolutionary search procedure in real-time, by directly commanding the physical hardware, can be extremely complex. First of all, safety mechanisms to avoid poor-performing or even unstable solutions, which can cause permanent damage to the hardware, must be developed. The algorithm must also be designed to avoid interferences between testing of subsequent solutions. Furthermore, to obtain significant and practically useful results, thousands of experiments are necessary to obtain controllers with high fitness, and mechanisms for improving the speed of convergence become indispensable to obtain results of practical interest in reasonable search times [12].

The EA have been employed in both online and offline optimisation of the structures and parameters of controllers for closed loop systems. Some research have focused only on the optimisation of the parameters of a pre-defined controller structure. More recently, the optimisation has been performed simultaneously on the structure and parameters of controllers. It has also been demonstrated that evolutionary optimisers can be used to derive superior controller structures in aerospace applications in less time (in terms of function evaluations) than other methods such as Linear Quadratic Regulators (LQR) and Powell's gain set design [13], [14].

2.4.1.1.3 Robust Stability Analysis: Evolutionary algorithms have been utilised in the context of efficient robust control system design. With all physical systems, there are bound to be parameters that vary with operating conditions. Depending on the complexity and the significance of these variations, the time-invariant models used in the design of control systems will need to model these parameter variations as a first step in order to design truly robust control systems [13]. Evolutionary algorithms have also been incorporated in procedures that tune both the structure and parameters of controllers while considering the parameter variations that exist within the nominal model [12], [44]. These parametric variations are often not discrete but continuous and of infinite combinations. Thus, in order to model all the possible parametric variations and to simultaneously optimise the controllers accordingly, applications involving $\mathcal{H}_\infty$ robust control theory, which help in defining models that suitably bound these variations, have been employed in combination with evolutionary algorithms in recent reports [19].

2.4.1.2 Power Electronics

The design of electronic circuits is often time consuming and can be a mundane process due to the need to repeat a number of design steps. Normally, during the electronic circuit design process, the circuit topologies and its relevant components are selected logically but, on the other hand, design decisions such as switching frequency, duty cycle, heat sink size, *etc* are made based on experience and intuition. This could provide room for error in the design process especially if electronic circuits for entirely new applications are being developed. For this reason, it would be of great benefit to the design engineer to have a computer-aided design tool that would remove uncertainty, through the dependance on intuition and establish more certain approaches in the electronic circuit design procedure [45].

For a typical power electronic design process that employs the EA, the parameters to be optimised may include the switching frequency, inductor parameters (number of primary and secondary windings, air-gap size), number of capacitors, diodes and their configuration (parallel or series). Through simulations, the quality of the randomly generated design is evaluated against the given specification for the circuit; this may include the losses the circuit generates at certain load conditions, minimum deviation from the specified output voltage for the expected input voltage range and load conditions, the output voltage ripple being within specified limits in the presence of certain voltage input and load conditions, *etc* [46]. Successful applications of the EA in the design of electronic circuits have been reported in several academic journals and its application in this field is increasingly being taken up amongst power electronics engineers[45], [46].

2.4.1.3 Power Networks

The cost and reliability of power distribution systems are becoming as important as those of their associated power generation and transmission systems. The planning of these large-scale and complicated distribution systems depends on computers and mathematical optimisation tools. The goal of the modern day power distribution system planning focuses on meeting the ever-increasing and dynamic load demands, within operational constraints, economically, reliably and safely by making optimised decisions based upon several factors such as sub-station location, size, reliability, voltage level of the distribution network *etc* [47].

Currently, an important issue in the field of power distribution is the need to optimise networks that serve commercial and residential areas. Generally, these networks are designed based upon two criteria: (1) minimal cost and (2) reliability of the supply (sub-stations and/or feeders). The solutions to the optimisation problem would be the parameters that enable the planner obtain optimal locations and sizes of the sub-stations and supplies to these stations (feeders) that achieve the best values for the two design criteria. The EA have been extensively adopted within this field of power networks in optimising models for these power distribution networks. The very complicated nature of the design: simultaneously and objectively evaluating the network-design criteria, which can be seen as complements of each other, presents a problem that the EA are capable of dealing with effectively; the nature of the problem makes it virtually impossible to be tackled by widely practised *ad-hoc* methods.

2.4.2 Aerospace and Automotive

In the field of aeronautical design, optimisation has a very significant role. To highlight the significance, it is established achieving optimal designs for structures, aerodynamic shape and flight trajectories can bring about significant savings in costs, fuel consumption, reduction in green house emissions, *etc* [48]. In most cases, achieving these optimal solutions involve solving complex nonlinear partial differential equations that are not amenable to traditional gradient-based methods. The EA have been adopted in dealing with such complex problems because these problems require a sensible random and exhaustive search technique; these requirements are in line with the inherent nature of the EA.

The EA have been used in achieving practical optimal designs of the wing-shape for a super-sonic aircraft [48], [49]. The four major parameters that were considered for the design were aerodynamic drag at supersonic cruising speeds, drag at subsonic speeds, aerodynamic load (bending force on the wing) and the twisting moment of the wing. These parameters are mutually exclusive and attaining an optimal wing design required tradeoffs between the parameters. A multiple-objective EA provided the means to objectively assess the separate contributions of each parameter and to provide the necessary trade-offs required to meet the design objectives. This is another instance that has encouraged the wider the uptake of unconventional EA approaches over traditional approaches for engineering design.

In the automotive industry, due to tighter restrictions on both air pollution and energy consumption, there is a significant drive towards developing Hybrid Electric Vehicles (HEV). However, the design of these vehicles are difficult due to the large number of inter-related design parameters and conflicting design objectives. Traditional optimisation techniques have proven to be ineffective at dealing with hybrid electric vehicle design due to the rather complicated nature of HEV systems. As a result, the EA are being increasingly adopted to deal with such design problems as the stochastic search techniques are particularly suited to dealing with such complex engineering design [50].

2.4.3 Telecommunications

Telecommunications are a significant symbol of the modern age of society. Over the last decade, the growing demand for data communications has influenced the rapid development of network infrastructures, cellular networks and internet services. Currently, new technologies such as cellular mobile radio systems, optical fibres and high speed networks have are in wide-spread demand and on a global scale. These new technologies permit the fast data communications and provide myriads of avenues for new services and applications. As a result of the current situation, there is interest stirring up in technology and telecommunication problems such as antennae design, efficient allocation of base stations, frequency assignment to mobile phones and structural design problems relating to routing information through the network. These problems and may others found in the field of Telecommunications can be expressed as optimisation problems [51].

The size of existing telecommunication infrastructure is constantly on the increase and the result is that the typical optimisation problems are becoming increasing more complex and are posing significant challenges to existing algorithms. The effect of this has been a search for an alternative algorithm that can tackle these newly found complexities. The EA are now being increasingly adopted by Telecom engineers in dealing with these optimisation problems. The EA have already been successfully applied to a number and a diverse range of telecommunication applications that include hardware design, data transmission and network design.

The EA have been successfully applied to optimisation of the hardware infrastructure involved in antennae design, The optimisation problem with the antenna design focuses on keeping the structure as simple and low-cost as possible and at the same time achieving the particular electrical requirement. The robustness and the versatility of the employed EA make the method especially beneficial in instances that have large search spaces and have proven better performances than other algorithms [52]. In the field of data transmission, applications that have required the EA optimisation include aspects related to the direct communication of data between two components such as cellular phones and base stations. In the area of network design, the EA have been applied to minimising the costs of setting up networks and optimising the network's reliability and connectivity, which focuses on the network's functionality despite failures in some of its links or nodes. They have also been applied to optimising placement and configuration of the antennae systems within such networks [51].

2.5 Conclusion

The chapter has focused on the description, implementation and the applications of the Evolutionary Algorithms. In achieving this aim, initially, a classification of optimisation algorithm was presented and the importance of Evolutionary Computing, the field to which Evolutionary Algorithms belong to, was highlighted. Next, The preference for the Evolutionary Algorithms over other algorithms within the artificial intelligence was highlighted. Finally, the general implementation and application of these algorithms were discussed. The general implementation of the Evolutionary Algorithms can be summarised in six phases. These are the phases of Initialisation, Evaluation, Fitness Assignment, Selection, Reproduction and Termination. The second to fifth phase of the Evolutionary Algorithms are repeated until the criteria for termination are met.

The Applications of Evolutionary Algorithms within three important sectors of industry - Power, Transport and Communication. The description of the applications of the *EA* to the Power industry, includes applications in the field of control system design, which can be categorised into the off-line and online methods, system identification for process models and robust stability analysis of control systems. Within the field of control systems design, there are already well established strategies for off-line applications but the field of the online applications have been very limited to date due to the very challenging problems that need to be dealt with for its implementation. The research project deals with these difficulties in order to achieve its objectives.

Although the many benefits of the Evolutionary Algorithms have been highlighted in this chapter, they have their limitations. It has been suggested that many of the limitations of existent evolutionary algorithms, such as premature convergence, stagnation, loss of diversity, lack of reliability and efficiency, are derived from the fundamental convergent evolution model and the oversimplified "survival of the fittest" Darwinian evolution model. Within this model, the higher the fitness the population achieves, the more the search capability is lost. This is also the case for many other conventional search techniques [20].

Chapter 3

Investigated Evolutionary Algorithms

3.1 Introduction

The usefulness of the evolutionary algorithms has been highlighted in chapter 2. Some of the different applications of these algorithms are also discussed. This chapter focuses on the three evolutionary algorithms employed for the robust automated control design of this research project. The algorithms are employed to optimise the closed loop speed controller for the experimental described in chapter 4. In achieving the optimum control solution for the experimental system, the algorithms ensure that the speed response of the closed loop experimental system follows the desired speed as closely as is possible; this serves as an important criteria for quantifying the quality of the closed loop system's response.

The first *EA* considered is The Genetic Algorithm (*GA*). The *GA* is currently amongst the most popular evolutionary algorithms and it has become a very attractive tool for optimisation processes. The simplicity and elegance of its underlying concept and the requirements for the *GA* implementation are some of the reasons for its popularity. For these reasons, the GA was deemed a suitable approach in fulfilling the aims of the research project.

The Bacteria Foraging (BF) Optimisation Algorithm is relatively new on the scene of Evolutionary Algorithms. The reason it has been employed in this research work is that recent publications have demonstrated that in some cases it has comparable performance with the GA [68]. Also, some other publications have suggested it is more effective than the GA under certain criteria [69], [70]. Given the overall positive remarks on the BF, it was deemed suitable to investigate the effectiveness of the emerging evolutionary algorithm and do some comparison with its more popular counterpart.

The Hybrid Bacteria Foraging (HBF) algorithm is a novel optimisation approach developed during the course of the research project. Although similar algorithms have been implemented in [71] and [72], the novelty of the developed HBF algorithm is due to the uniqueness of its application in order to meet the project demands. Its development stemmed out of curiosity as to the effectiveness of the combination of particularly desirable features of two remarkable search and optimisation techniques. The details of its implementation and the manner in which it combines the principles of both GA and BF will be discussed

3.2 Genetic Algorithm

The Genetic Algorithms (GA) are a subset of evolutionary computing, which is a rapidly growing part of Artificial Intelligence (AI). The GA is a random search and optimisation method that is inspired by Charles Darwin's evolution theory and Gregor Mendel's principles of *natural selection* and *survival of the fittest*. The beginnings of genetic algorithms (GA) can be traced back to the early 1950s when several biologists used computers for simulations of biological systems. However the work done in late 1960s and early 1970s at the University of Michigan, under the direction of John Holland, led to GA as they are known today [14], [15]. John Holland is credited as the Father of the Genetic Algorithms, since he was the first to introduce them in the 1960s. Further work for the algorithm's development was carried out by Holland and his students and colleagues at the University of Michigan.

The motivation for the design of the Genetic Algorithms came from a desire to study the natural adaptation within nature and to implement its underlying processes within a computer system. In 1975, John Holland published a book titled *"Adaptation in natural and artificial systems"*. This presented a theoretical framework for the computer implementation of evolution theories. For the Genetic Algorithms, its implementation can be summarised in four steps - initialisation, selection, reproduction, termination. These are highlighted in figure 3.1.

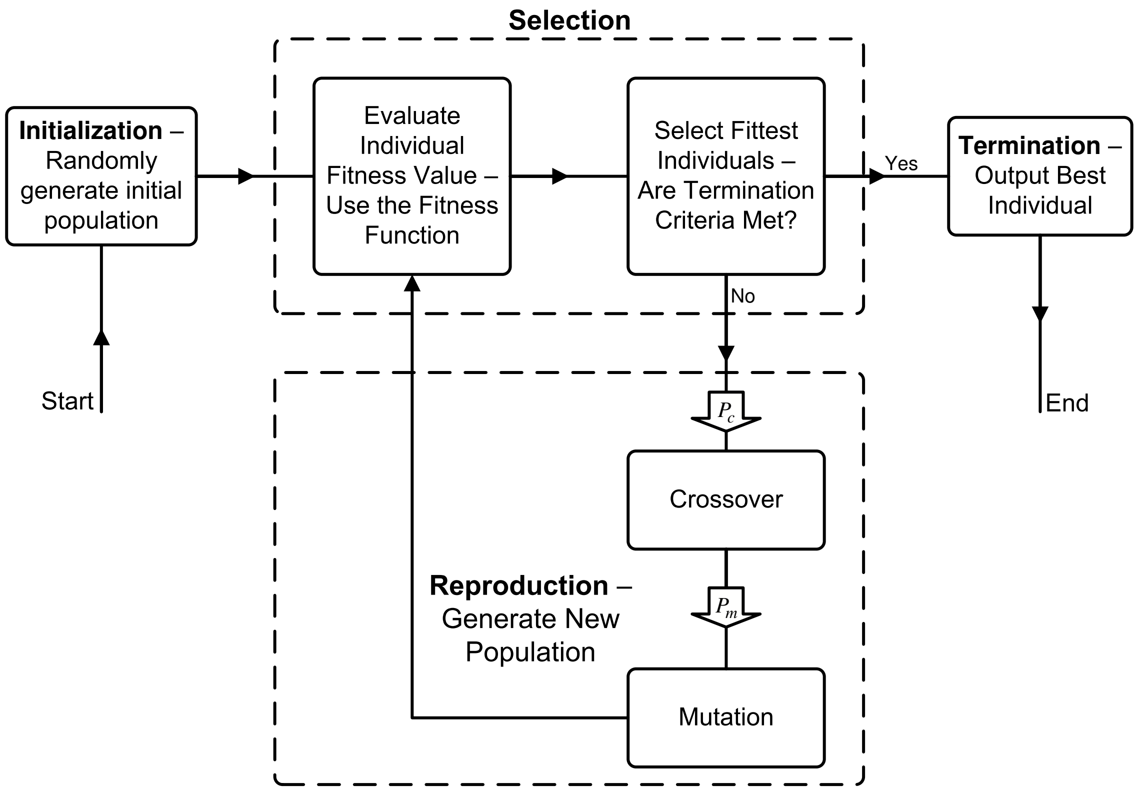

Selection
Initialization – Randomly generate initial population
Evaluate Individual Fitness Value – Use the Fitness Function
Select Fittest Individuals – Are Termination Criteria Met?
Yes
Termination – Output Best Individual
No
Start
End
P_c
Crossover
Reproduction – Generate New Population
P_m
Mutation

Figure 3.1: Flow Chart of The Genetic Algorithm

Genetic Algorithm (GA) is a stochastic global search method that is inspired by the theories of evolution and natural selection. It operates on a population of potential solutions, termed individuals, applying the principle of evolution, simulated by means of mathematical operations that mimic the process of selection, crossover and mutation. A fitness function measures the fitness of an individual to survive in a population of individuals. The genetic algorithm will seek the solution that maximises or minimises the fitness function, generating at each step a new generation of solutions using the operations of mutation and crossover and selecting the best individuals for the population at the following step. The results of this evolutionary manipulation are new solutions, termed offspring. The new sets of solutions produced, representing a new generation, are then interbred by the same means as the parent solutions. The process stops when a termination criterion has been reached. The output of the GA optimisation should, in theory, converge at the best possible solution. Unlike other optimisation algorithm, because the GA performs a search of the solution space in parallel and not by point to point, it is far less likely to converge at local optimum but more likely to reach the global optimum solution [65].

3.2.1 Initialisation

At the start of the GA, a specified number of individuals and the maximum number of generations are chosen; and using these specified variables, the first generation of individuals is generated randomly. Also defined at the start of the GA are the range of values the individuals can take, the number of offspring to produce from one generation to the next,the crossover and mutation rate for the reproduction function and finally the termination criteria. Each individual represents a possible solution to the robust controller optimisation problem. An individual is represented as a string of numbers and encoded within each individual are the parameters and the structure for the digital controller to be designed.

3.2.2 Selection

The selection procedure of the GA is responsible for choosing a specified number of the 'fittest' individuals from the current population of individuals for reproduction, in order to form the new generation of solution. The selection process is dependant on the fitness of each individual. In order to quantify the 'fitness'(quality of the response) of each individual(possible solution), a user-defined fitness function is employed. Each individual is tested and evaluated using the fitness function and the output of the function - the fitness value quantifies the fitness of each individual. The fitness function employed for the optimisation procedure is defined as the Integration of the Absolute Error (IAE) between the demand speed, ω^* and the real speed of the system, ω. The fitness value evaluation is represented in figure 3.2.

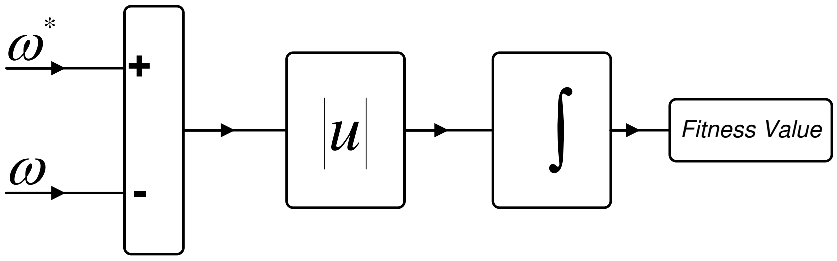


Figure 3.2: Fitness function of Genetic Algorithms

Having quantified the fitness of each individual, the selection of individuals to place in the mating pool is performed. There are a number of ways to implement the selection. Some of the more common selection methods are the elitism, roulette-wheel selection, stochastic universal sampling and Tournament selection. These have been described in section 2.3.4. A combination of elitism and the stochastic universal sampling is used to implement the selection function for the GA harnessed in the research project.

The elitist selection ensures that at least one copy of the fittest individual(s) of the current generation is passed directly onto the new generation. The main advantage of this approach is it prevents the best solutions from being lost through generations, thus increasing the probability of convergence to a global optimum solution. On the flip side, there is also a risk of convergence to a local optimum. The stochastic universal sampling is used to select the remaining members of the new generation. It provides zero bias and minimal spread within the selected population. The individuals are mapped onto a line segment such that each segment represents each individual's fitness value relative to the other individuals. Then equally spaced pointers are placed over the line. The number of pointers placed represents the number of individuals to be selected. The first pointer is selected randomly. Its position must lie within the length of the interval between the first two pointers.

3.2.3 Reproduction

As mentioned in the previous section, the individuals chosen in the selection phase of the *GA* are used to form the individuals of the new generation. By means of the elitist approach, copies of the best individual(s) from the current generation are passed on directly to the new generation. The remaining part of the new generation is formed by using a probability guided simulation of *crossover* and *mutation* on the individuals selected using the Stochastic Universal Sampling method. With the opposing and collective diverging and converging effects of mutation and crossover respectively, there is the increased chance of the *GA* arriving at a global optimum solution at termination.

3.2.3.1 Crossover

The *crossover* function is used within the *GA* to create two individuals for the new generation from the combination of features of two individuals of the current population. It is essentially the genetic operator that simulates sexual reproduction. The combination is done by merely swapping parts of the individuals between each other. Its occurrence depends upon the crossover rate value P_c, which simply determines how often, within any generation, the crossover function is carried out on pairs of individuals. The value of P_c is often chosen to be in the range 0.5 - 1.0 [73]. Crossover can be implemented easily by two methods. These methods are the 'single-point' and the 'multiple-point' crossover. The implementation always involves two individuals within the population being considered. For the purpose of illustration, the individuals considered are members of a real number decimal population. The implementation of binary crossover has been presented in figure 2.6; but given that real number decimal coding is best for engineering applications, it has been employed for the research project. The two methods for its implementation are illustrated in figure 3.3.

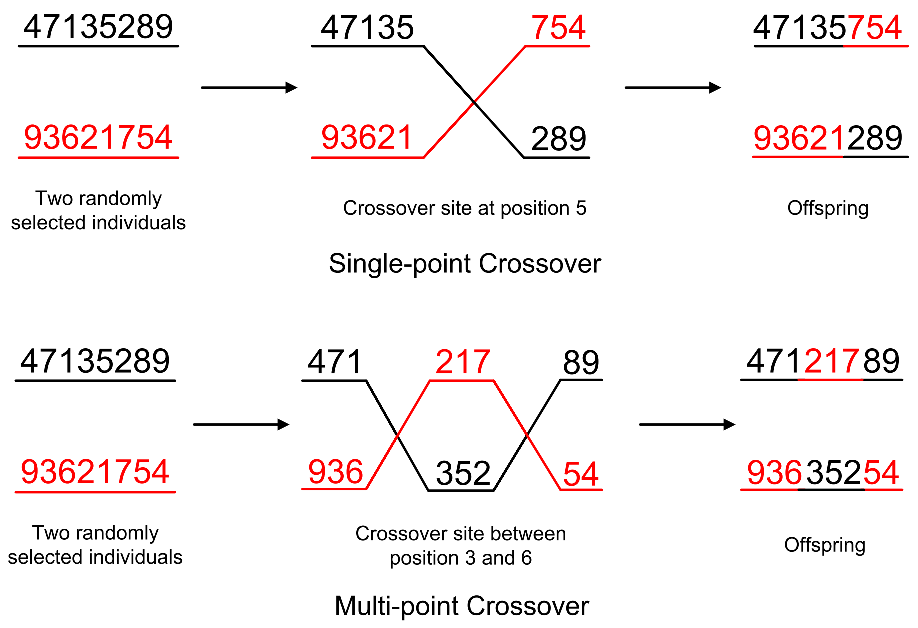


Figure 3.3: Illustration of the crossover function

The single-point crossover has been adopted for the research project. During the simulation of the single-point crossover, the two individuals selected randomly for reproduction are paired off. A *crossover site* is selected and all the digits of one individual to the right of the crossover site are exchange with those of the other. For the multi-point crossover, initially, two *crossover points* are selected and the area between these two points serve as the *crossover site*. The portion of first individual within the site is swapped with the corresponding portion of the second individual. The resulting individuals, termed offspring, form the individuals of the population of the new generation. The crossover function is necessary to ensure convergence of the *GA* to an optimal solution.

3.2.3.2 Mutation

In order to prevent the premature convergence of individuals to a local optimum and widen the search capability of the algorithm, the *mutation* function is incorporated. The mutation function enables the *GA* to search the sample space of solutions more effectively through its random application on the individual. Naturally, its effect could be of no consequence to the search procedure or it could have a potentially negative effect by creating individuals which are really bad solutions to the optimisation problem; it could also *potentially* discover really good solutions. In all cases, the mutation operator improves the wider search capabilities of the optimisation algorithm.

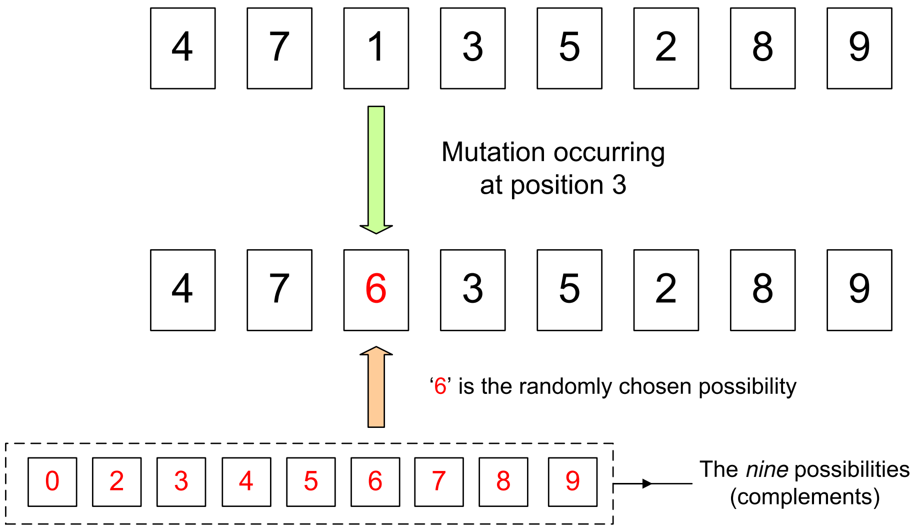

4 7 1 3 5 2 8 9
Mutation occurring at position 3
4 7 6 3 5 2 8 9
'6' is the randomly chosen possibility
0 2 3 4 5 6 7 8 9
The *nine* possibilities (complements)

Figure 3.4: Illustration of the mutation function

The application of mutation involves randomly choosing a position to perform the operation and then changing the figure in that position to any one of its complementary values. For the decimal population, any figure (in any position) would have *nine* complementary values. An illustration of this operator is shown in figure 3.4. An illustration of binary mutation has also been illustrated in figure 2.5; but the real number decimal coded approach is preferred and adopted for the research project because they are the best for engineering applications.

It must be highlighted that mutations can occur randomly to every position of each parameter within an individual. Figure 3.5 shows the mutation of a typical individual that encodes both the controller's structure and relevant parameters: k represents the 'gain' of the controller; A, B, D and E represent the 'zeros' of the controller; C, F and G represent the controller's 'poles' while F_0 and F_1 represent the flags that define the controller structure. More details on the nature of the digital controller optimised is presented in section 5.4. In figure 3.5, the PI controller is mutated into a Fourth

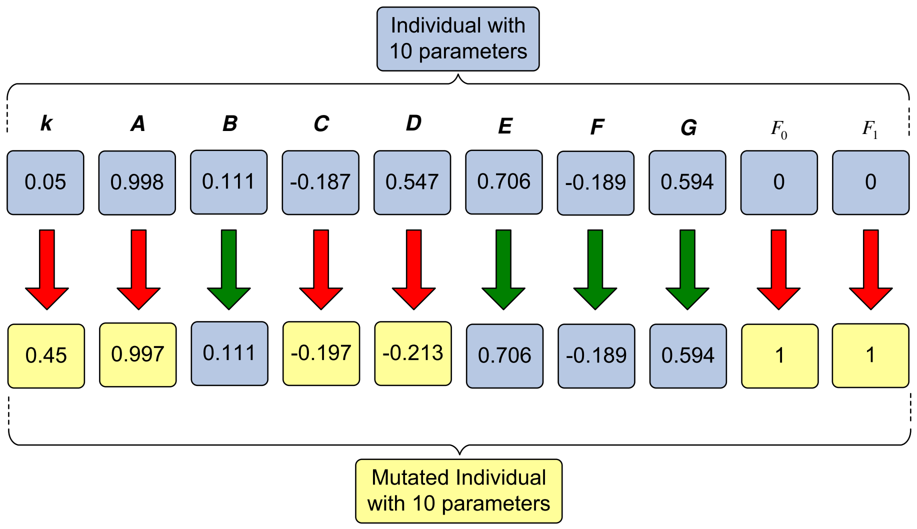


Figure 3.5: Illustration of the mutation on typical individual

order controller. Randomly altering the individuals within a population enables the *GA* to effectively search through the sample space, thus increasing its chances of converging at a global optimum solution. The mutation operation does not occur as frequently as the crossover function and it is regulated generally by using a small mutation probability, P_m. The mutation operator is carried out on a single individual and the value of P_m is often chosen to in the range 0.005 - 0.05 [73].

3.2.4 Termination

This is the point at which the *GA* outputs the optimum controller structure and parameters having reached the termination criteria specified within the algorithm. Some of the possible criteria for termination includes

- A maximum computation time specified at the start has been reached
- A total number of generations has been achieved.
- The objective value below a certain fixed value
- A certain number of iterations have been performed

A combination of the possible termination criteria are often used to bring the GA to a stop. In the case of an effective search, at the point of termination, the GA should arrive in an area around the global optimum solution to the optimisation problem being considered. With regards to the research project, the global optimum solution will be the best possible structure and parameter for the digital controller for a variable speed drive that ensures robust performance, in terms of speed dynamics, in the presence of variable mechanical load.

3.3 Bacterial Foraging

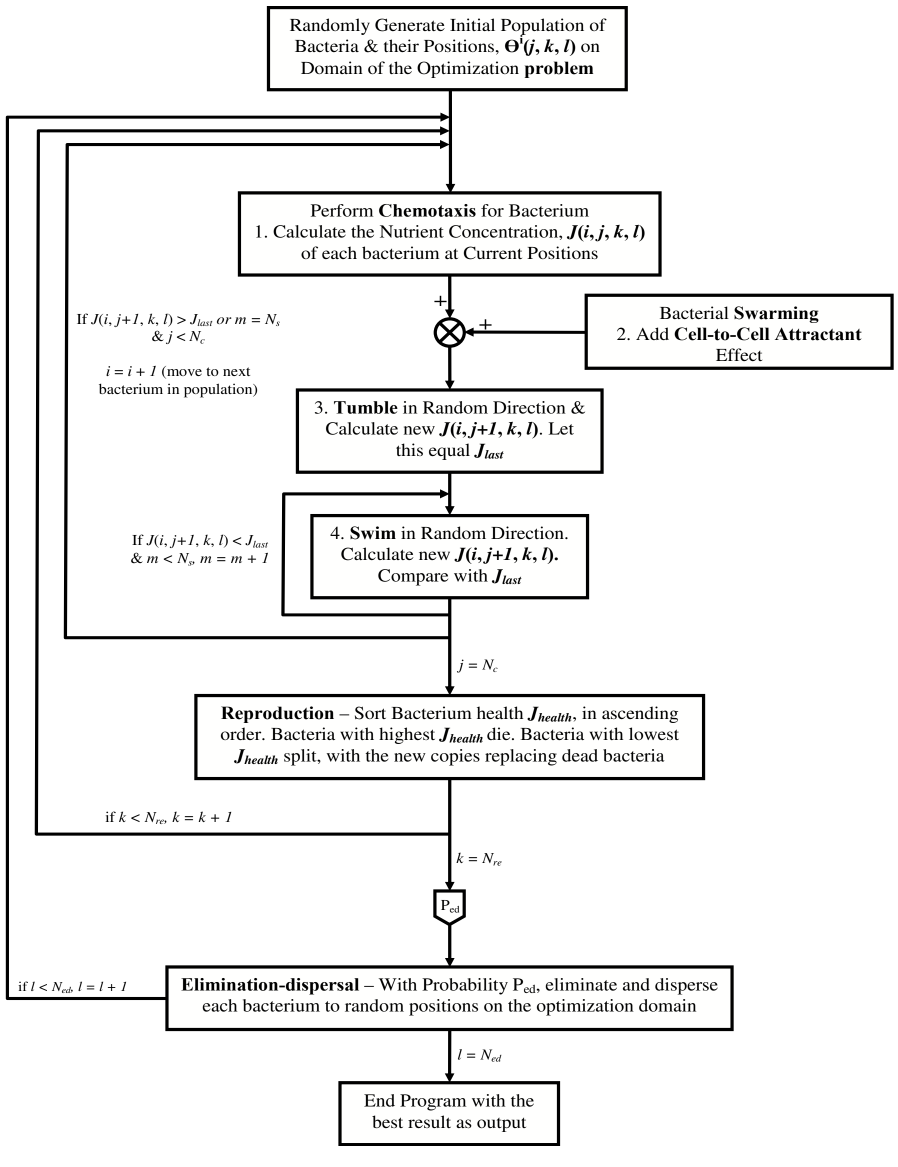
From the observation of nature, it is evident that living organisms with good foraging habits are most likely to propagate their genes through generations. Such organisms are healthier and are able to successfully adapt to their environment. As a result, they are more likely to produce offspring, thus spreading their genes through generations. This simple observation is clearly stated within Darwin's principle of natural selection. The very tenets of this elegant principle is increasingly being adopted by scientists, within algorithms, as tools in achieving optimal solutions for many optimisation problems. Bacterial Foraging (BF) is one of such algorithms. It was formally introduced in 2002 by Kevin M. Passino in [29]. The BF is a stochastic search and optimisation technique based on the foraging habits of *Escherichia coli*, more commonly known as *E. coli*, a bacterium commonly found in the gut of human beings.

Extensive studies have been carried out on the foraging habits of *E. coli* [29], [68]. This has enabled the successful implementation of a computer model of their foraging habit as an optimisation algorithm. A fundamental part of the BF is the movement of the bacterium. Each one possesses six rigid whip-like structures, called flagella, to enable motion. These flagella make between 100-200 rotations per second and the manner in which they are made to rotate guides the types of motion exhibited by the bacterium in response to the stimuli it encounters within different mediums. The motion (or taxis) of *E. coli* is generally triggered in response to different chemicals.

For this reason, the bacterial motion is termed *chemotaxis*, which literally means motion triggered by chemical stimulants. There are many other types of taxis that are used to describe bacterial motion and some of these include: aerotaxis (attracted to oxygen), phototaxis (light), thermotaxis (temperature), magnetotaxis (magnetic lines of flux) *etc*. It has also been noted that certain bacteria can change shape and number of flagella to ensure efficient foraging within different media [71]. The chemotactic motion exhibited alternately by the *E. coli*, whilst searching for better forage, are of two distinct types: (1) tumble and (2) runs (swims). More details on the types of chemotactic motion are provided in section 3.3.2. The chemotactic motion of *E. coli* is modelled within the BF algorithm according to the possible mediums the bacteria encounters and its response within such mediums. This is summarised as follows:

1. Neutral substance medium: Bacterium tumbles and runs alternately
2. Noxious substance medium: Bacterium tumbles more than it swims as it attempts to get out of the noxious substance (climb down the noxious substance gradient). It essentially seeks more favourable conditions
3. Nutrient substance medium: Bacterium swims more than it tumbles while it searches for even more favourable nutrient mediums (up the nutritious substance gradient)

The BF optimisation algorithm is regarded as a social foraging algorithm because it requires collective search and foraging of a number of bacteria within a colony. For any social foraging technique to be effective, it is necessary that each bacterium is able to communicate and harness this communication capability with other bacteria to fully capitalise on the collective searching and foraging of the colony. For the bacteria colony of *E. coli*, communication is ensured by the secretion of chemicals which either alert other bacteria on whether suitable forage or unsuitable conditions have been encountered. The chemicals are termed *attractants* when its overall effect congregates bacteria to suitable foraging spots. In the case where it alerts other bacteria to move away from unfavourable conditions, they are termed *repellents*. This useful means of communication within the bacteria colony is incorporated within the Bacterial Foraging Algorithm by implementing the Swarming function, described in section 3.3.3. The BF optimisation algorithm has been developed, within MATLAB/Simulink, particularly to meet the demands of the research project. The implementation of the BF optimisation algorithm is summarised in Figure 3.6. The highlighted stages - initialisation, chemotaxis, swarming, reproduction and elimination/dispersal are described in the following sections.



Randomly Generate Initial Population of Bacteria & their Positions, $\Theta^i(j, k, l)$ on Domain of the Optimization problem
Perform Chemotaxis for Bacterium
1. Calculate the Nutrient Concentration, $J(i, j, k, l)$ of each bacterium at Current Positions
+
+
Bacterial Swarming
2. Add Cell-to-Cell Attractant Effect
If $J(i, j+1, k, l) > J_{last}$ or $m = N_s$ & $j < N_c$
$i = i + 1$ (move to next bacterium in population)
3. Tumble in Random Direction & Calculate new $J(i, j+1, k, l)$. Let this equal J_{last}
If $J(i, j+1, k, l) < J_{last}$ & $m < N_s$, $m = m + 1$
4. Swim in Random Direction. Calculate new $J(i, j+1, k, l)$. Compare with J_{last}
$j = N_c$
Reproduction – Sort Bacterium health J_{health}, in ascending order. Bacteria with highest J_{health} die. Bacteria with lowest J_{health} split, with the new copies replacing dead bacteria
if $k < N_{re}$, $k = k + 1$
$k = N_{re}$
P_{ed}
if $l < N_{ed}$, $l = l + 1$
Elimination-dispersal – With Probability P_{ed}, eliminate and disperse each bacterium to random positions on the optimization domain
$l = N_{ed}$
End Program with the best result as output

Figure 3.6: Flow Chart of the Bacterial Foraging Optimisation Algorithm

3.3.1 Initialisation

At the start of the *BF* algorithm, all the parameters required for its implementation are specified. These include the number of bacteria within the population, the positions of each bacterium within the sample space, the number of chemotactic steps taken during each bacterium lifetime, the number of reproduction and elimination/dispersal events that would occur during the algorithm's implementation.

Having defined these initial parameters, the bacterial population is randomly distributed to different positions within the search space. The different bacterial positions represent potential solutions to the optimisation problem. The nutrient concentration function evaluates the nutrient concentration in the each of the different bacterial position. The aim of the *BF* is to seek out the best solution that optimises the nutrient concentration function (The solution that optimises the nutrient concentration function represents the best structure and parameters for the robust digital controller). After the initialisation phase, each bacterium searches for for the best solution within the sample space by performing *chemotaxis*.

3.3.2 Chemotaxis

E. coli has the proclivity to convene at nutrient-rich areas by an activity called chemotaxis. The word 'chemotaxis' simply means movement in response to chemical stimulus. They achieve chemotaxis in two different ways: Each bacterium can either 'run', which is movement in a specified direction, or it can 'tumble', which is a movement in a random direction. The bacteria always alternates between these two modes of chemotaxis all its life while searching for nutrients. Figure 3.7 illustrates the two different modes of chemotaxis. In order to search for the positions with best nutrient concentrations, each bacterium takes a specified number of chemotactic steps. Each chemotactic step is described by swim intervals during which the bacteria moves in a straight line interspersed with tumbles, when the bacteria has random reorientation.

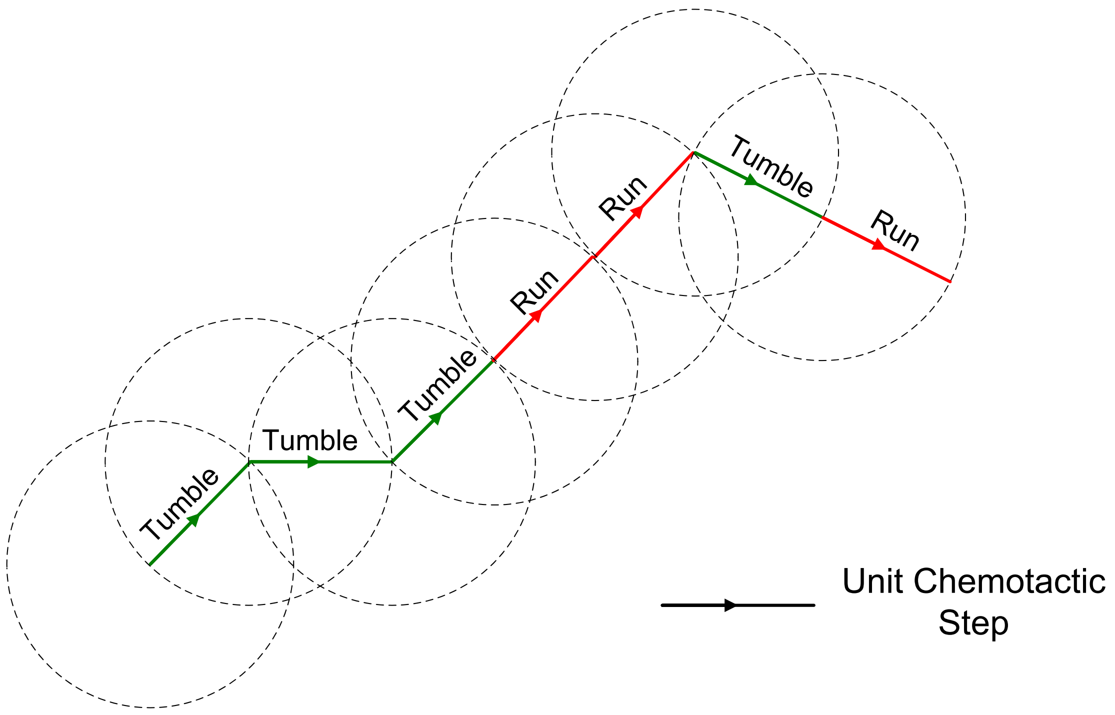


Figure 3.7: Modes of Chemotaxis - *Tumble* and *Run*

For the search procedure, bacteria need to have some direction. This is necessary to indicate whether the search procedure implemented by the algorithm is progressive. In order to incorporate such an indicator, the BF optimisation algorithm adopts a nutrient concentration function. The nutrient concentration function quantifies nutrient concentration at each bacteria position and outputs a corresponding nutrient concentration value. It has the same implementation as the objective function described in section 3.2.2. At the initialisation of the algorithm or at the beginning of each chemotactic step, tumble or run/swim within the chemotactic loop, each bacterium has, what is referred to as, its 'initial' position. The nutrient concentration function evaluates each 'initial' bacterium position to output a corresponding initial nutrient concentration value. The 'initial' nutrient concentration value serves as a reference to compare the subsequent nutrient concentration values that will be obtained during the algorithm's execution. By carrying out such a comparison, the algorithm can decide which is a good direction to progress the search in. Thus, the nutrient concentration function is a means to indicate a progressive search procedure.

The subsequent nutrient concentration values are determined each time a bacteria tumbles (moves in random directions) or run/swim (moves in a specified direction) from its initial position. In the chemotactic process, the tumble must always occur before the run/swim. After a tumble, which is a move in a random direction to a 'new' position, the nutrient concentration is evaluated. The 'new' nutrient concentration value is compared with the 'initial' nutrient concentration value. The outcome of the comparison determines whether a swim will follow subsequently. If the 'new' is better than the 'initial' nutrient concentration value, a run/swim will follow in the direction randomly chosen by the tumble. On the contrary, if the 'new' is worse than the 'initial' nutrient concentration value, there will be no run/swim. The chemotactic process for this bacterium ends by storing the position (and its corresponding nutrient concentration) achieved. In the case of the research project, 'better' means 'smaller', given the objective is to minimise the nutrient concentration function.

If the conditions for a run/swim to occur are satisfied, the 'new' nutrient concentration value (obtained as a result of the tumble) now represents the 'initial' nutrient concentration value and thus, serves as a reference point for the first run/swim that subsequently follows. After the first swim (move in the direction specified by the tumble) occurs, a new bacterium position is obtained. The 'new' nutrient concentration is quantified. In order for a second swim to then occur, the same conditions for the first swim to occur must be met. if it is not satisfied, the run/swim ends by storing the position and nutrient concentration of the last successful swim (the first swim in this case).

After a tumble or a run/swim, the bacterial position, represented by θ, is given by (3.1). Within the equation, i represents the counter for each bacteria within the population; j represents the number of chemotactic steps that each bacteria has undertaken during its lifetime; k represents the counter for the reproduction steps while l is the indicator for the elimination and dispersal events that is occurring.

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j) \tag{3.1}$$

where $\theta^i(j,k,l)$ represents the position of the i^{th} bacteria at the j^{th} chemotactic step, the k^{th} reproductive step and the l^{th} elimination and dispersal step. $C(i)$ is the size of the chemotactic step taken in a random direction by the i^{th} bacteria. At the end of each chemotactic step, the nutrient concentration is calculated.

3.3.3 Swarming

When cells of the *E. coli* are randomly distributed in a solution that has varying concentrations of nutrients and noxious substances randomly distributed within it, each bacterium would secrete attractants to signal other cells if it finds that it is swimming in areas with good nutrient concentration. This facilitates the convergence of cells of bacteria to form groups around areas in the solution with high nutrient concentration. This enhances the effectiveness of the search and foraging procedure.

Also, when cells of bacteria experience noxious substances, each cell would secrete repellents to divert the search and foraging process away from the areas with noxious substances. This causes divergence of the bacteria cells, ensuring that they spread out to other areas, thus improving the effectiveness of the search procedure. This behaviour of the foraging of bacteria termed swarming has been modelled within the BF optimisation algorithm. The mathematical expression for swarming can be represented as in (3.2)

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}^{i}(\theta, \theta^{i}(j,k,l)) \\ &= \sum_{i=1}^{S}\left[-d_{attract}exp\left(-w_{attract}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2\right)\right] \\ &+ \sum_{i=1}^{S}\left[-h_{repellent}exp\left(-w_{repellent}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2\right)\right] \end{aligned} \qquad (3.2)$$

$J_{cc}(\theta, P(j,k,l))$ is the cell-to-cell attraction/repulsion function value that is to be added to the nutrient concentration function, which is to be optimised. S is the total number of bacteria, p is the number of parameters to be optimised which are present in each bacterium, $d_{attract}$ and $w_{attract}$ represents the quantification of the depth and width of the attractant released by each bacterium and $h_{repellent}$ and $w_{repllent}$ represent the quantification of the depth and width of the repellant secreted by each bacterium.

3.3.4 Reproduction

Reproduction occurs when every cell in the bacteria population has moved the specified number of chemotactic steps. The reproduction occurring is not sexual reproduction, as in humans, but conjugation, which merely involves the transmission of genetic material from a donor to a recipient cell. It is the next phase after chemotaxis. Reproduction is necessary to cause the all the bacteria within the population to converge at the bacteria population with the best nutrient concentration value.

Reproduction in BF optimisation algorithm within the bacteria population is achieved in two stages. The first stage essentially involves assigning a fitness value to each bacterium within the population. The fitness value determines which bacteria is fit enough to reproduce. In some research publications, the fitness value of each bacteria within the population has been evaluated by calculating the sum of its nutrient concentration values obtained over total number of chemotactic steps [71], [74].

For this research project, it is derived by considering only a single value, which is the best nutrient concentration value that corresponds to the best position occupied by the bacteria during chemotaxis. The benefits of this definition of the fitness value is it ensures the best positions are not lost but are propagated through the bacterial population by means of the process of reproduction.

Having achieved the fitness assignment, the bacteria are then ranked according to their fitness values. The bacterium with smaller nutrient concentration values are ranked higher than those with larger values. In order to reproduce, the one-half of the population of bacteria, ranked the least, is killed off. The remaining half which has the better ranks is replicated - each surviving bacterium splits into two copies of itself. The resulting population produced still has the same number of bacteria within the population but, for each bacteria that is not 'killed-off' there exist an identical copy. The new population then serves as initial position for the next chemotactic process or the next elimination-dispersal event.

3.3.5 Elimination and Dispersal

The environment of a population of bacteria could be subject to changes. These changes could have a gradual impact such as variations in concentration of nutrients in the area the bacteria population reside due to their foraging habits; the more they feed within their environment, the less forage will be left to feed on in future. It could also be sudden, possibly as a result of torrid environmental conditions appearing such as weather conditions, diminished nutrient concentration *etc.* This could result in death of some members of the bacteria population or general irregularities within the climate of the bacteria environment. The events that bring about sudden or gradual changes within the foraging environment of bacteria have been modelled in a simple form within the BF optimisation algorithm; they are termed elimination and dispersal events.

Elimination: The process simply involves randomly killing off some of the poorly performing bacteria within the population. The need for this is simply to provide room for new members of the bacterium population that potentially are situated in areas with higher nutrient concentration. The obvious effect of elimination is a reduction in the total bacterial population. In order to counter the reduction, a complimentary process termed Dispersal occurs.

Dispersal: The eliminated bacteria are randomly replaced by new ones, which are probably situated in different (and possibly better) locations than the previously existing members of the population. These new locations might be better because they may be situated closer to spaces with better nutrient concentration. This enhances the global nature of the search procedure by dispersing some parts of the population into other parts of the sample space.

The elimination and dispersal events, which occur after a specified number of reproductive steps, help ensure the BF optimisation algorithm is truly global in its search procedure. The elimination and dispersal events could have a negative influence on the foraging of the bacteria by destroying bacteria in certain positions, which might be close to nutrient-rich areas and transcribing portions of the bacterial population to positions with poorer nutrient concentration. It could also have the positive effect of introducing the bacterial population to environments with higher concentrations of solutions or moving them away from noxious substances, thus enhancing the progress of the search and foraging procedure.

3.3.6 Termination

At this juncture, the BF outputs the positions of the bacterium that correspond to an optimal design of both structure and parameters for the robust controller. The termination criteria that may be employed in bringing the algorithm to a halt include:

- The maximum computation time for the algorithm has been reached
- Convergence to a fixed low nutrient concentration value is reached
- A total number of elimination and dispersal events have occurred.
- A certain number of iterations have been performed

As with the GA, a combination of possible termination criteria are used to bring the algorithm to a stop. The termination criteria used within the research project are convergence to a fixed low fitness function value and reaching the total number of elimination and dispersal events. After termination, depending on the success of the search procedure that has occurred, the BF may have converged at the best possible solution.

3.4 Hybrid Bacterial Foraging

The Hybrid Bacteria Foraging (HBF) is an example of a hybrid evolutionary algorithm. Its name has been chosen for two reasons:

1. it is formed from the combination of the GA and the BF
2. it is largely similar in implementation to the BF.

Similar hybrid evolutionary algorithms have been developed and used in [42], [72], [71] and [70] but the novelty of the developed algorithm lies especially in the uniqueness of the application it has been specifically developed for. The circumstance that led to the development of the HBF stemmed from the process of investigating ways to improve the effectiveness of the simple BF optimisation algorithm (described in section 3.3.4) as an approach for the design of robust controllers for the electronic drive.

In the formation of the HBF optimisation algorithm, the aim was to combine specific desirable functions of the BF and the GA into one algorithm. The BF optimisation algorithm is known for its 'excellent local search' capabilities but it does have obvious limitations in its global search approach. This presents a scenario, which is the converse for the GA: it has excellent global search capabilities but is rather limited in its local search procedure. Merging the two algorithms, through selective combination of certain favourable functions of the BF and GA could potentially yield an algorithm that has excellent local and global search capabilities. Every other process within the HBF is exactly the same as those of the BF optimisation algorithm apart from the Chemotactic and Reproduction process. The modifications that are implemented through the combination with GA are the reasons for the differences. These constituent processes were modified because they largely determine the effectiveness of the HBF and sensible modifications to such processes can bring about significant improvements in the performance of any algorithm.

3.4.1 Hybrid Chemotaxis

The Hybrid chemotactic process includes the process of tumble and run/swim. It also includes the GA reproductive process which is the modification that aims to improve the normal chemotactic process of the BF. After every bacterium has performed chemotaxis, the new bacteria position and the corresponding nutrient concentration values achieved are modified using the reproductive process adopted from the GA. The idea is to create a new set of bacteria positions from the initial set, which is derived from the tumbles (and swims). The new set obtained through the GA modifications will then be used in next chemotactic step.

In deriving the new set of bacteria positions, first, the initial set of bacteria positions are ranked. The ranking is based on their nutrient concentration values. The smaller values are ranked higher and vice versa. After the ranking process, a certain (user-specified) number of bacteria positions, which are the most highly ranked are passed directly into the new set of bacteria positions. The remaining members of the new set are formed from the initial set by randomly simulating crossover to produce new bacteria positions and then carrying out the mutation function on randomly selected bacteria positions. These GA reproduction operators are described in section 3.2.3. The new set of bacteria positions are used as the initial positions for the next chemotactic step.

3.4.2 Hybrid Reproduction

A similar modification described for the Hybrid chemotaxis has been implemented for the reproduction phase of the BF to yield the hybrid reproduction of the HBF. The reproduction phase described in section 3.3.4 involves initially assigning fitness values to each member of the bacteria population, ranking each member of the population according to its respective fitness value, killing-off of the bottom-half of the ranked population and finally duplicating each member of the top half of the population. The *GA* modification alters this reproduction process by using the operators of crossover and mutation to produce a new population, rather than merely duplicating the top-half of the ranked bacterial population.

The bacterial population is ranked according to their fitness values of each bacterium. The fitness value is derived by considering only a single value, which corresponds to the best (minimum) nutrient concentration value the bacterium experienced all through the chemotactic process. The lower the fitness value of the bacterium, the better its rank and vice versa. Having achieved a ranked bacterial population, a number of highly ranking bacterium are passed unaltered to the new population of bacteria. The remaining members of the new population are obtained by applying the functions of crossover and mutation randomly on the remaining bacteria within the ranked population. The modified approach to reproduction enables a better chance of convergence of the bacterial population to the positions that correspond to the best structure and parameters for the controller being designed. At the same time, the mutation function enables the algorithm to search wider areas within the sample space thus enhancing the global nature of the search procedure.

3.5 Conclusion

The chapter presents a detailed description of the structure and implementation of the particular evolutionary algorithms employed during the research project. The algorithms investigated are the popular Genetic Algorithms, the emerging Bacterial Foraging and the novel Hybrid Bacterial Foraging Algorithms. The stochastic and global nature of these algorithms ensure that during the search procedure, they are likely to converge at the best possible solution. A unique advantage to employing the Genetic Algorithms in tackling optimisation problems is it has good global search capabilities. It spends more of its time searching widely across the sample space but it is less inclined to performing localised searches. This could be a disadvantage especially when the best possible solutions can be found within a localised area.

The Bacterial Foraging Optimisation Algorithm, on the other hand, is more focused in searching locally. Although it does search globally, by its nature, it does so less frequently and through the aid of elimination and dispersal events. The Hybrid Bacterial Foraging Algorithm combines the specific benefits of both the Genetic Algorithms and the Bacterial Foraging Algorithm, implementing both local and global search of the sample space of possible solutions. In general, the nature of these algorithms ensures that once the process begins, there is no need for any user interaction, implying reduced amounts of human time allocated to control design. This makes it very viable commercially. Also, given the requirements for their implementation, these algorithms can be adapted for any optimisation problem and can lend itself to myriads of applications in control and engineering in general.

Chapter 4

Experimental System Description

4.1 Introduction

In order to test the efficiency and effectiveness of the Evolutionary Algorithms, whose general implementation is described in chapter 2, in designing robust control systems, the system prototype employed was a variable speed drive that operates under variable mechanical loads. In this chapter, the ultimate aim is to provide a concise and detailed description of the Experimental system that has been employed for the testing of the Evolutionary Algorithms during the research project. To achieve this objective, it is necessary to:

1. Describe, in a generic fashion, the variable speed drive system employed for the research work. This involves characterising the electric motor used, the nature of the converters employed and the closed loop control systems employed for the motor drive control.

2. Provide details on how the mechanical loads, which are a necessary condition for the optimisation, have been incorporated within the experimental system.

3. Describe how the experimental system has been specifically implemented to

meet the requirements of the research project. This involves describing each component of the experimental system and demonstrating how they come together to yield the overall system.

4.2 Variable Speed Drive System

The detailed schematic of the system employed is shown in figure 4.1. The experimental system mainly comprises a variable speed drive system that controls a variable mechanical load.

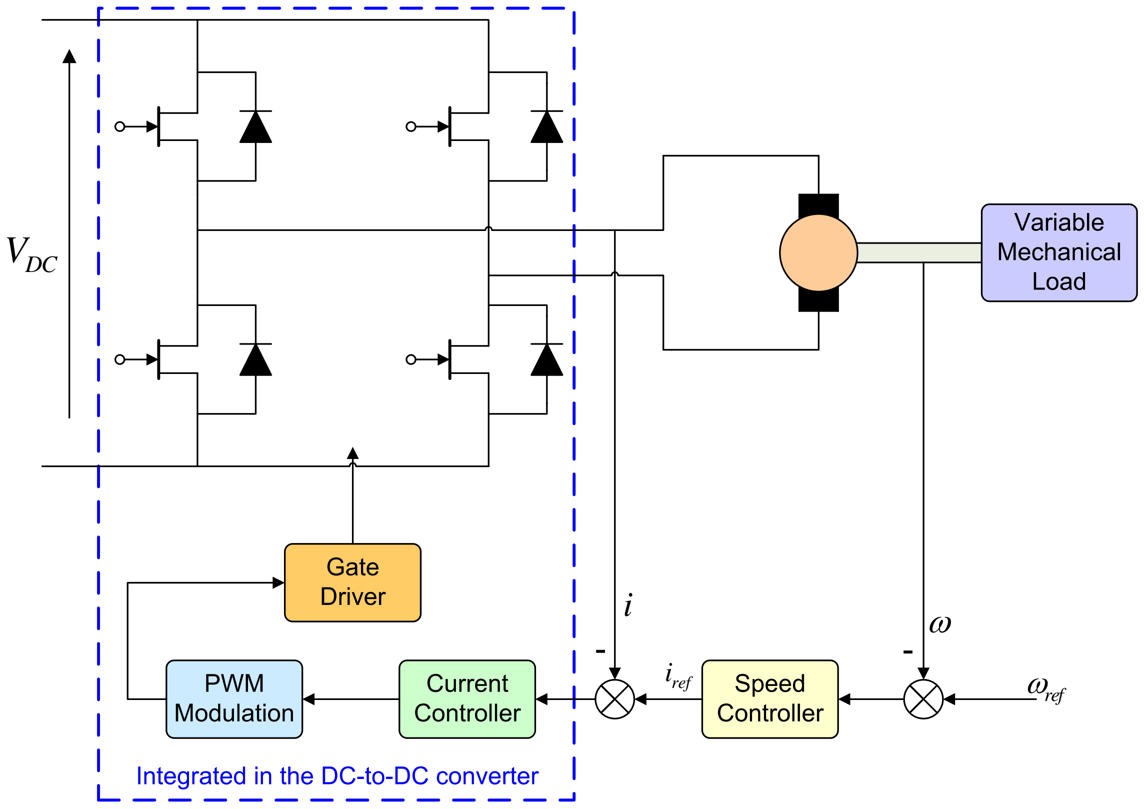


Figure 4.1: Experimental System

The variable speed drive system employed essentially consists of a cascade control system that has an inner (current), outer (speed) control loop, a power conversion stage and a permanent-magnet DC motor. The system's power conversion stage employs a DC-to-DC converter based on a Junction Field Effect Transistor ($JFET$) H-bridge configuration with a Pulse Width Modulation (PWM) scheme which switches at $20kHz$ and has a nominal current of $3A$. Integrated within the DC-to-DC converter is an analogue PID current controller that is employed in the implementation of the inner (current) loop.

The speed control of the electric motor is performed by an outer speed loop implemented as a digital control system. The electric motor employed is a brushed Permanent-Magnet DC servomotor. The selected drive was the preferred choice because of the ease with which it could be controlled and this facilitated the implementation of the experimental system and the overall optimisation procedure. The speed controller employed in the outer control loop forms the subject of the optimisation procedure and its nature is fully characterised in section 5.4.

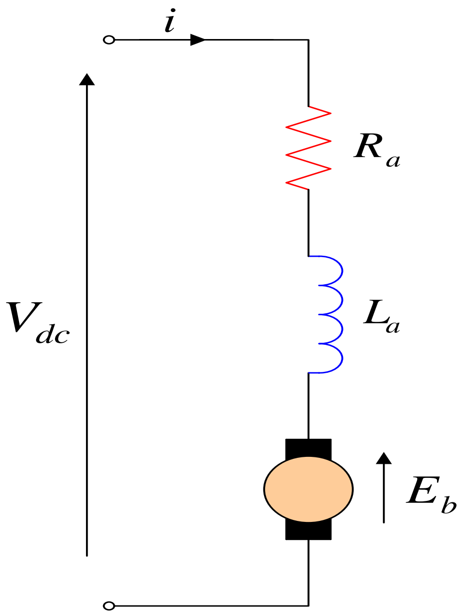


Figure 4.2: DC motor equivalent circuit

The equivalent circuit that describes the electrical dynamics of the DC motor can be represented in figure 4.2. The corresponding equation for the DC motor operation is given in (4.1). V_{dc} represents the DC voltage across the terminals of the armature of motor, L_a represents the inductance of the armature windings, R_a represents the armature resistance, i represents the current flowing in the armature windings and E_b denotes the back EMF generated as a result of the motor's operation.

$$V_{dc} = L_a \frac{di}{dt} + iR_a + E_b \tag{4.1}$$

At the start of operation, a voltage, V_{dc}, is applied across the armature of the DC motor. The current, i, that would flow initially through the armature windings will be large, since it is only armature winding resistance, R_a and the impedance of its inductance, L_a, that will be experienced. Such large currents could damage the motor by affecting its brushes, commutator or windings. To reduce this high start-up current, in large DC motors, starting resistors are placed in series with the armature. The magnetic field produced by the current, i, will interact with the field produced by the permanent magnet, resulting in torque production that would cause rotation of the armature. As soon as the armature begins to rotate, following *Faraday's laws of electromagnetic induction*, an Electromagnetic Force (EMF) is induced to stop the motion of the armature through the magnetic field. The direction of the EMF is opposite to that of the applied Voltage, V_{dc}, hence it is termed the Back EMF. The effect of the Back EMF is that it reduces the overall current in the armature windings as the speed of the armature/motor increases.

4.3 Variable Mechanical Load

The variable mechanical load forms the necessary conditions under which the speed controller of the experimental system has been optimised. In order to deal with the impracticality of having the different mechanical loads physically present for testing the developed control algorithms, it was necessary to emulate the different mechanical load dynamics. To help visualise the incorporation of such an emulator system into the experimental system, it would be useful to represent the experimental system in figure 4.1 as the simplified schematic in figure 4.3.

The Programmable Load Emulator System has been developed to accurately reproduce the dynamics of these different mechanical loads. Adopting such a system has

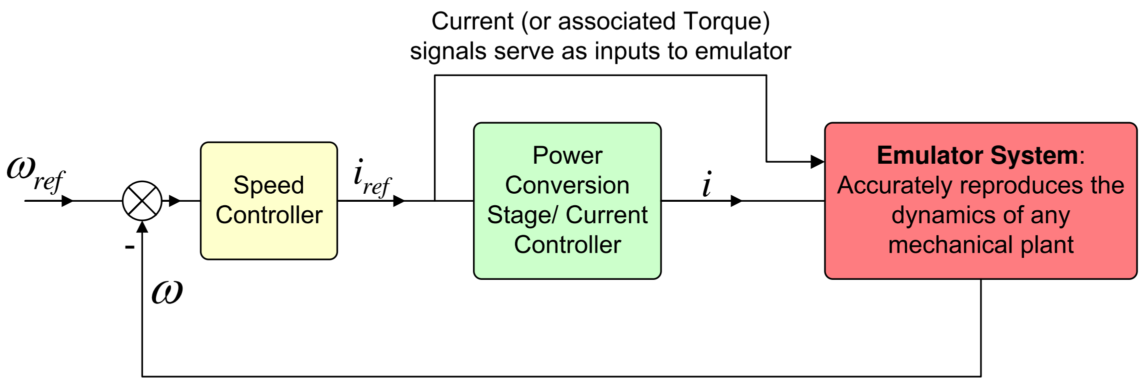

Current (or associated Torque) signals serve as inputs to emulator
ω_{ref}
-
ω
Speed Controller
i_{ref}
Power Conversion Stage/ Current Controller
i
Emulator System: Accurately reproduces the dynamics of any mechanical plant

Figure 4.3: Incorporation of Emulation System

provided the flexibility in harnessing different load conditions, which in turn provides the means to exhaustively test the control algorithms as tools for robust control system design. The concept behind its implementation and the results that verify its effectiveness are presented in chapter 5. In the following section, the implementation of the Experimental System will be described by fully characterising its constituents components.

4.4 Experimental System Implementation

In order to meet the requirements of the experimental system prototype, it was necessary to design a system that is able to experimentally emulate any mechanical load and also to implement a variable speed drive system that controls these mechanical loads. This has been achieved by suitably controlling twin motors coupled on the same shaft, with the control system of one motor serving as the variable speed drive system and the control system of the other motor providing the means to dynamically emulate different mechanical loads.

The layout of the hardware design for the experimental system is shown in figure 4.4; it shows the different parts and highlights how they come together to yield the overall experimental system. The subsequent sections will give a description of the different parts highlighted in figure 4.4.

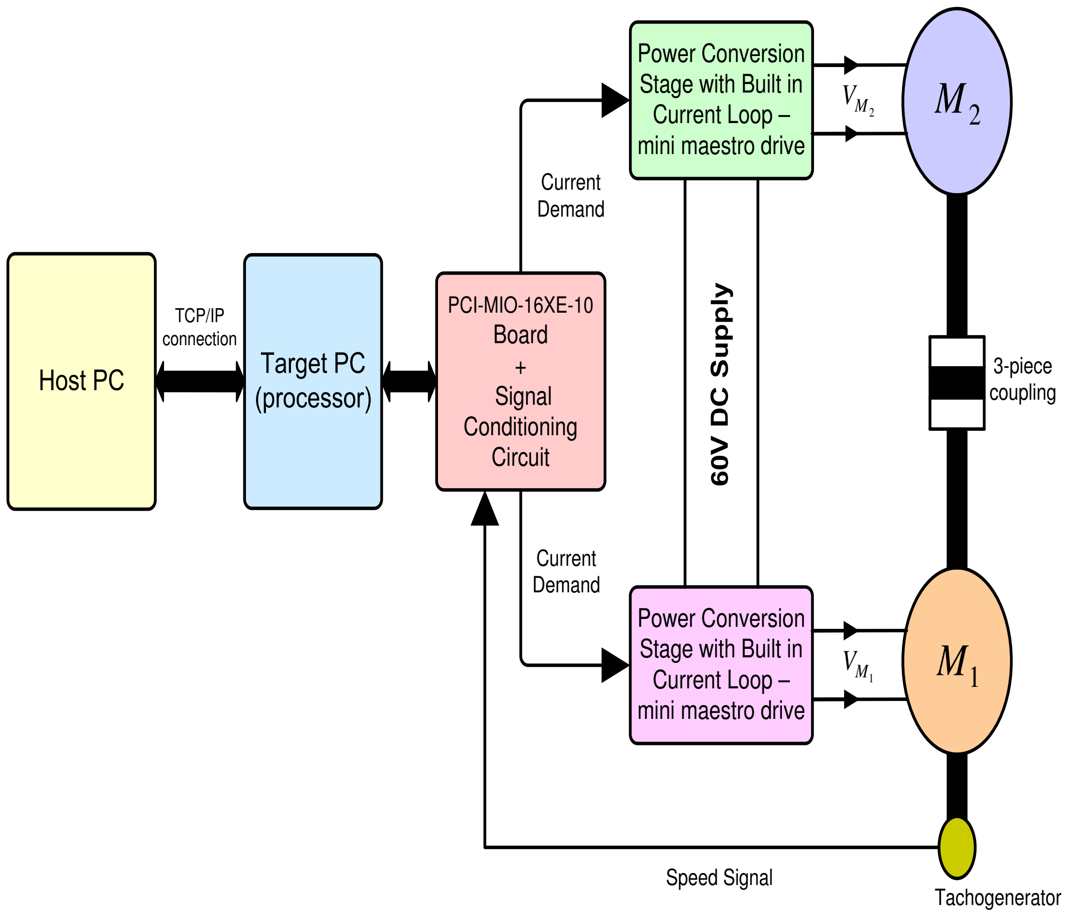

Host PC
TCP/IP connection
Target PC (processor)
PCI-MIO-16XE-10 Board + Signal Conditioning Circuit
Current Demand
Power Conversion Stage with Built in Current Loop – mini maestro drive
V_{M_2}
M_2
60V DC Supply
3-piece coupling
Current Demand
Power Conversion Stage with Built in Current Loop – mini maestro drive
V_{M_1}
M_1
Speed Signal
Tachogenerator

Figure 4.4: Overall layout of experimental rig.

4.4.1 The Motors

The motors used on the test rig are the **DCM 2B 30/03 A2** Control Technique brushed permanent-magnet Matador DC servomotors. The parameters of the motor at rated conditions are given in table 4.1.

The two motors are coupled together along their shafts. The coupling used are produced by the Ruland Manufacturing company. The particular coupling used for the servo application is the oldham coupling. It is a three piece coupling comprising two aluminium hubs and a centre piece. This centre piece, which can be metallic or made of plastic, serves as the torque transmitting material. For the purpose of the research project, it has been selected as a plastic material. This is because it provides electrical

Stall Torque	0.32 Nm
Stall Current	4.6A
Maximum Peak Current	23 A
Rotor Inductance	1.34 mH
Rotor Resistance	0.85 Ω
Volt Constant	7.3 V/krpm
Rotor Inertia	0.0000324 kgm^2

Table 4.1: DC motor parameters at rated conditions

isolation and it can act as as a mechanical fuse - when the plastic centre piece breaks, it breaks cleanly and prevents the transmission of power, thus preventing possible damage to more expensive machine components.

The unique design of the coupling enables it to operate with zero backlash, which means no misalignment between the two motors [53]. This implies that the coupling can bring about an excellent link between the two motors, M_1 and M_2 in figure 4.4.. This feature is particularly important in the research project for the emulation strategy adopted.

Information about the speed of the motors is obtained from the tacho-generator signal. The signal is used in monitoring the state of experimental system. Table 4.2, which summarises the specifications of the tachogenerator, has been provided by Control Techniques, the manufacturers of the *DC* drives.

4.4.2 The Power Conversion Stage

The power conversion stage employed in the experimental setup is the Control Techniques DCD 60×10/20 mini maestro. It consists of a DC-to-DC converter and an analogue current controller based on a MOSFET H-bridge configuration switching at 20kHz with nominal current of 3A. The permanent magnet brushed DC servo-motors are particularly suited to these power converter interfaces in delivering powerful and

voltage constant	0.01 V/rpm
peak to peak ripple	1.6 %
rms ripple	0.7 %
linearity error	0.1 %
voltage tolerance	±5.0 %
voltage variation	-0.02 % per °C
nr. of comm. segments	25
nr. of poles	4

Table 4.2: Tachogenerator specification

accurate motor control.

The converters can be powered from DC supply with output voltage of between 24V to 72V. It can also be powered from a rectifier with DC-link voltage of between 20V to 80V. The maximum allowable ripple on this voltage is 2V peak-to-peak. A DC supply of 60V has been selected for the purpose of this research work and for a simplified configuration since the aim of this research is focused on control system design.

The mini maestro units provide the options for analogue speed and current control. Only the analogue current control provided is used for the research project. The built-in current controller uses a proportional plus integral and derivative (PID) controller. The 20kHz switching frequency of PWM within the current loop, which employs Junction Field Effect Transistors (JFET), ensures fairly silent operation of the drives. The analogue speed controller is bypassed and instead, a digital controller is used, whose software implementation allows the GA routine to interact with it. The schematic of the converter is shown in Appendix A.3

4.4.3 The xPC target System

The xPC target system is a high performance environment that consists of a host and target PC connected together. It enables the connection of Simulink and State flow models to physical systems and the execution of these model applications in real-time on a PC-compatible software. The host-target connection is achieved using either a RS-232 or TCP/IP protocol (direct, LAN or ethernet). The TCP/IP cable connection is preferred because it is the faster of the two options. It provides data rates of up to 100Mbit/sec over any distance [54]. In the following sections, the functions of the Host and the Target PC will be described.

4.4.3.1 The Host PC

The host PC is used for control and monitoring of the downloaded target PC applications. It provides the environment for the execution of real-time applications using the xPC target toolbox of MATLAB-Simulink. MATLAB is a high-level technical computing language. It provides a suitable environment for the development of algorithms, visualising and analysing data. Simulink is a platform for multi-domain simulation and model-based design of dynamic systems.

It is a requirement that the host-PC runs MATLAB, Simulink, Real-Time Workshop, xPC target and a C-compiler as the development environment for the running of real-time applications. Applications are designed separately on MATLAB, using its command-line interface - the m-files, and Simulink. The applications implemented within Simulink are compiled into C-code and downloaded onto the target PC, via the communication link between the two, where it is processed. The command-line interface of MATLAB is then used in passing commands to the target PC to control the real-time functioning of the downloaded application.

The optimisation algorithms, the digital controller and the closed loop programmable emulation system are implemented on the Host PC. The optimisation algorithms are programmed in the m-files of MATLAB. Both the digital controller, which is implemented in the difference equations format, and the closed loop programmable emulation system are designed in Simulink. By using the C compiler on the Host PC, the Simulink diagrams can be converted to C-code, which aiss data processing.

4.4.3.2 The Target PC

The target PC processes the information downloaded from the host PC and is capable of implementing three particular functions. These are:

1. the target application control
2. the parameter tuning
3. the signal (data) acquisition function

The classification of the functions of the target PC characterises the commands directed from the host PC in a similar fashion. The target application control function is essentially used for downloading the target applications onto the target PC from the host PC, start and stop execution, change the stop and sample time and detect CPU overloads.

The parameter tuning function enables changing the model parameters on the target PC. This is achieved by using the command-line interface to alter the model parameters, either before or during execution of the application, after downloading it to the target PC. The xPC target kernel running on the target-PC enables the signal logging in order to acquire signals throughout the execution of an application. Data is stored in real time in Random Access Memory (RAM) or in the file system of the target system during the signal acquisition mode. On completion of the execution phase, the signals obtained can be analysed and deductions can be made accordingly.

4.4.4 The Interface Board

The interface board used in the experimental rig is the National Instruments PCI-MIO-16XE-10. This is the heart of the experimental system. It is an I/O board with 16 single or 8 differential analog input (A/D) channels (16-bit) with a maximum sample rate of 100 kHz, 2 analogue output (D/A) channels (16-bit), 8 digital input and output lines and two counter/timers (24-bit) with a maximum source clock rate of 20MHz. The I/O board is housed within the target PC.

By means the PCI-MIO-16XE-10 board, the digital signals created within the target PC are converted to analogue signals and, after conditioning, the generated analogue signals are transferred to the power conversion stage. These analogue conditioned signals serve as the reference of the current loop within the converters for the motors, M_1 and M_2. The conditioned analogue tacho-generator signal, which encodes the speed of the motors, are converted to digital signals by the PCI-MIO-16XE-10 board. These digital signals are processed on the target PC and transferred to the host PC where they are stored and analysed.

4.4.5 The Signal Conditioning Circuit

The purpose of conditioning is to ensure that all the transmitted signals reach their destinations with minimal distortions and the amplitudes of the input signals to the circuit devices are within the allowable range. It is necessary to condition the tacho-generator signal. This conditioned signal serves as the input to the PCI-MIO-16XE-10 board. It is also necessary to condition the output voltage signals from the PCI-MIO-16XE-10 board. The conditioned output PCI-MIO-16XE-10 signals serve as the reference for the analogue current control loops, which is situated in the converters on-board control. Figure 4.5 is the circuit used to condition the tacho-generator signal. It comprises the Potential Divider, Buffer, Active Low-pass Filter and Voltage Limiter

4.4.5.1 Potential Divider

The potential divider is to step-down the tacho-generator signal. It also ensures that the inputs to the buffer are kept within the specified allowable range.

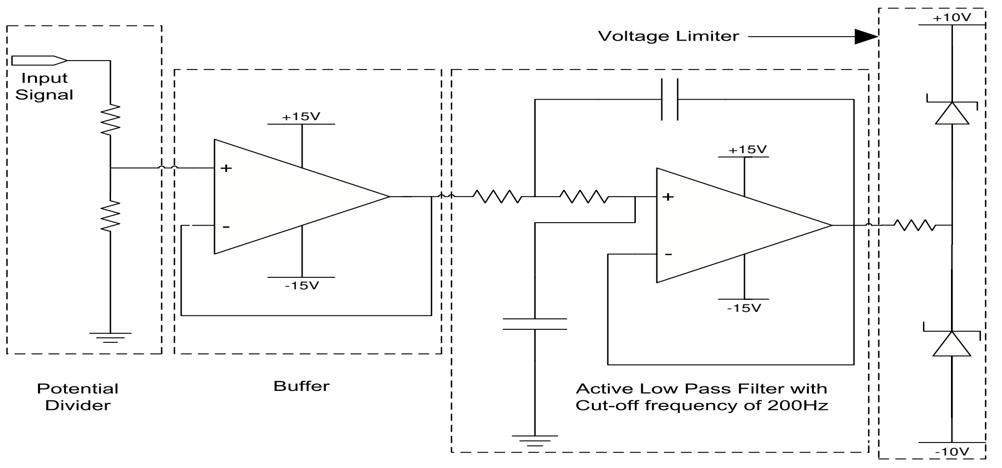


Figure 4.5: Signal Conditioning Circuit

4.4.5.2 Buffer

The buffer is simply an operational amplifier with unity gain. It is also known as a voltage-follower circuit. It is used to ensure the stepped down tacho-generator signal is transmitted with minimal distortions from the input point (the output of the potential divider circuit) to the input point (the Low-pass filter).

4.4.5.3 Active Low-pass filter

The second order low-pass filter is used to filter out noise. The Pulse Width Modulator (PWM) of the DC-to-DC converter, switching at 20kHz was a significant noise source. The cut-off frequency of the filter is approximately 200Hz. Shielded cables were also used wherever possible to help reduce effects of noise on the signal quality.

4.4.5.4 Voltage Limiter

The voltage limiting circuit is achieved simply by using zener diodes rated at 10V. The rated values of these diodes are chosen to coincide with the range of allowable input signals to the PCI-MIO-16XE-10 board and the mini maestro drive for the motors.

Figure 4.5 shows the arrangement of these zener diodes that ensures that signals with amplitudes greater than 10V are clamped at 10V. The circuit used to condition the output PCI-MIO-16XE-10 signals consists of the buffer (the signals are buffered first) and the voltage limiting circuit. The potential divider is excluded because the maximum allowable signal from the output of the PCI-MIO-16XE-10 board has amplitude of 10V. This has the same magnitude as the maximum allowable input to the drives of the motors. Both the drive and the PCI-MIO-16XE-10 board have low-pass filter incorporated to adequately remove noise.

4.5 Conclusion

In this chapter , the experimental system, which is employed in testing the effectiveness of the developed robust control system design techniques, has been described. The system essentially comprises a variable speed drive that is subjected to variable mechanical load conditions. In fully characterising the developed prototype system, it was necessary to:

1. Provide detailed insight of variable speed drive employed. This has been achieved by providing a description of the drive's inner current and the outer speed control systems and the DC Permanent-magnet motor that is employed within the electric drive system.
2. Elucidate the manner by which the variable mechanical loads are implemented and also highlight how the programmable load emulator system, which is used in accurately reproducing the different mechanical load dynamics, is incorporated into the experimental system prototype.

Given the impracticality of having different mechanical loads within the labs to test the developed algorithms, it was necessary to develop a programmable system capable of emulating these different loads and which was also incorporated within the experimental system. As a result, the hardware design of the Experimental System was arranged in the manner described in this chapter to ensure adequate representation of the variable speed drive prototype and also include the means by which to subject the drive to accurately reproduced dynamics of different mechanical loads within the experimental system. The concept for the implementation of the programmable mechanical emulator system and how it is fully incorporated within the experimental system is described in chapter 5.

Chapter 5

Mechanical Emulator System

5.1 Introduction

There have been considerable research activities in the field of nonlinear, adaptive and robust control methods for electrical servo drives. For the purpose of validating the control system effectiveness under different operative conditions, linear, nonlinear and time-varying mechanical loads are usually required. It is not a practical option having these different loads present within the laboratory; however dynamically emulating these mechanical loads using a dynamometer is a viable alternative [55]. For this reason, an emulation strategy has been employed to successfully develop a viable test bed to investigate the performance of evolutionary algorithms for automated control system design.

Torque controlled load dynamometers are commonly used in engine test beds or the testing of electrical machines. For such applications, the drives are usually tested under steady state or slowly changing conditions. For applications that require testing with loads having faster dynamics, simulated load emulation under open-loop conditions has been adopted *i.e.* the emulated load is not part of a closed loop speed or position control system [55], [56], [57]. In order to test variable speed drives, such tests might be insufficient because dynamic or non-linear effects in some loads can be dominant and these dynamometers do not meet the requirements for investigating the closed-loop dynamic behaviour of the drives and the motion-control techniques. Therefore, it is essential that dynamometers, for the testing of drives, should be able to produce dynamic load effects [58], [59].

More recently, dynamometers that can imitate practical load dynamics satisfactorily by using the inverse mechanical-dynamics (IMD) principle have been reported in [56], [57]. Using the IMD principle, the shaft speed (or position) is measured and used to derive the torque for the dynamometer to follow the desired dynamics. The method is effective only in the continuous systems where the sampling effects are not considered. In the digital systems, the sampling effects introduce noise, which can cause system instability under certain conditions. Stabilising filters are often introduced to counteract the problem but the introduction of such filters changes the open-loop pole-zero structure and this in turn causes the closed loop characteristics of the emulation strategy to differ from that of the desired load dynamics. Although exact dynamic matching of the emulated load and the desired load was not obtained, the target of achieving an acceptable time-response matching for an open-loop emulation was reached [55], [60].

In this chapter, the dynamometer control strategy employed during the course of the project is described. The results obtained, using the experimental emulation strategy, are provided and the simulation results are shown alongside for comparison. The dynamometer control strategy enables the accurate dynamic emulation of linear and nonlinear load models inside the speed-control loops. The strategy, which uses forward dynamics, preserves the open-loop pole-zero structure of the desired load dynamics and is suitable for discrete-time implementations. A similar emulation strategy has been employed in [55], [58], [60], [61]. The dynamometer control strategy provides the means with which mechanical load parameters (such as inertia and friction) can either be preprogrammed or varied with speed or position. In addition to drives/engine testing, another useful application of the mechanical load emulation is to provide off-site testing of drive converters which handle real industrial applications.

5.2 Dynamic Emulation of Mechanical Loads

The purpose of emulation is to accurately reproduce the dynamic response of a mechanical load for a given input signal. For the research project, this is achieved experimentally for both a stiff and flexible shaft mechanical load. In order to explain the concept of emulation, it would be useful to initially imagine two electric motors referred to as systems A and B in figure 5.1. System A is the subject of the emulation and System B would be used to reproduce the exact dynamic response of System A to an input Electric Torque, T_e. System A is chosen to have Inertia, J_M which is greater than the inertia, J_L of system B, Both systems are characterised by the same friction, B.

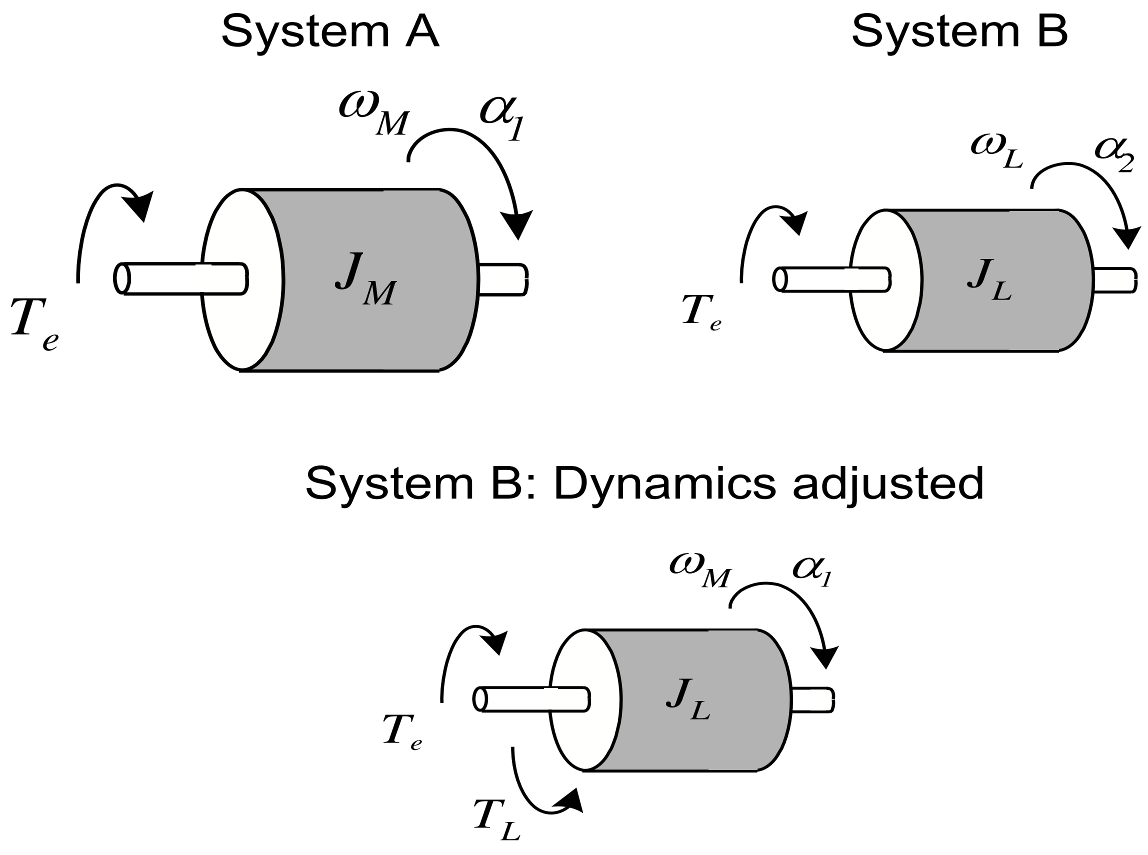


Figure 5.1: Illustration of Emulation

Considering that both systems start from rest, for the same electrical torque input, T_e to system A and system B, the acceleration, α_1 produced on system A would be smaller than the acceleration, α_2 of system B. In order to make system B move with the same acceleration, α_1 as system A for the same torque input, an opposing torque, T_L to the electric torque, T_e must also be applied to system B; the magnitude of T_L must be such that the resultant torque, T_e - T_L must now produce an acceleration on system B, which is the same as that produced by T_e on system A. The task with the emulation strategy has to deal with obtaining the required dynamics of the opposing torque, T_L. The solution to the problem is achieved by employing the described emulation strategy.

The mechanical dynamics of an electrical machine can be given by

$$T_e = J\frac{d\omega}{dt} + B\omega \tag{5.1}$$

where T_e is the electrical driving torque, J is the moment of inertia, B is the viscous friction coefficient and ω is the angular speed. The open-loop transfer function (OLTF) relating speed and electrical torque is:

$$\frac{\omega}{T_e} = \frac{1}{Js + B} \tag{5.2}$$

In this chapter, the objective is to control the load machine such that the relation between the shaft angular speed, ω and the electrical driving machine torque, T_e will be equivalent to the mechanical load model to be emulated, G_{em} i.e.

$$\frac{\omega}{T_e} = G_{em} \tag{5.3}$$

For the cases considered during the research, the desired relations to successfully emulate the stiff-shaft and flexible-shaft mechanical load models are given in equations (5.11) and (5.21), respectively.

5.3 Implementation of the Emulation Strategy

The emulation system employs a Permanent Magnet DC motor, which is exactly the same as the drive motor; the two motors are coupled on a common shaft. The drive machine and its DC-to-DC converters provide the required test system for research into the development of control algorithms. The load motor is controlled such that its speed (or position) response of the coupled system (of M_1 and M_2) to a given drive torque is equivalent to that of any desired linear or nonlinear mechanical load.

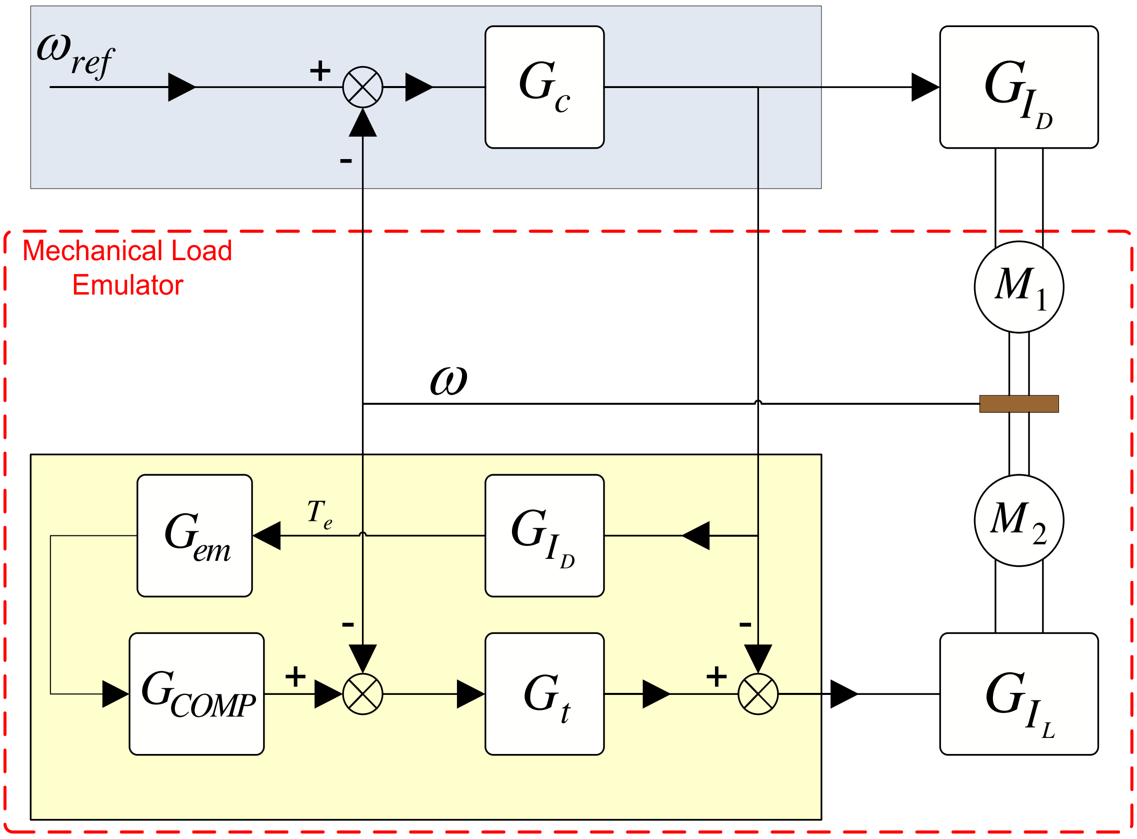


Figure 5.2: Mechanical Load Emulator System

Figure 5.2 highlights the mechanical load emulation system incorporated within the total closed loop speed control system. G_C is the driving machine's speed controller (the optimisation algorithms under test are applied to optimise this controller and it is fully characterised in section sec:EDCC), G_{I_D} and G_{I_L} are the current loops of the driving machine and dynamometer, respectively, G_{em} is the emulated load dynamics,

G_{COMP} is the inverse of the speed-tracking loop (this is described in section 5.3.3), G_t is the speed-tracking loop controller, T_e and T_L are the driving machine and dynamometer electrical torque demands, respectively.

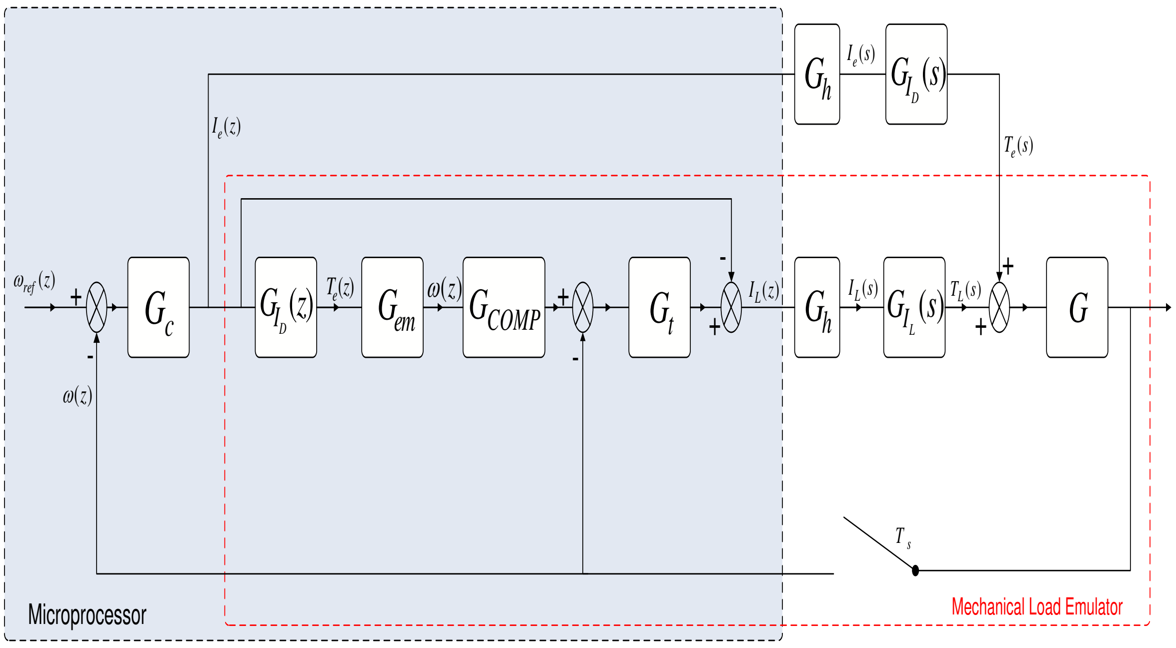


Figure 5.3: Total closed loop speed-control with Load Emulation

Figure 5.3 represents the simulation model of the complete closed-loop control with load emulation implemented within MATLAB/Simulink. G is the real total dynamics of the driving and load machines, ω_{ref} is the ideal speed demand, ω is the real shaft speed, G_h is the D/A converter and T_s is the sampling time used in the A/D converter.

In the emulation strategy highlighted in figure 5.2, the electrical torque input, T_e is initially applied to the mechanical load, G_{em}. The speed dynamics obtained is compensated by G_{COMP} for the effects of the speed-tracking loop; the compensated speed dynamics then serve as the reference for the this loop. The real shaft speed, ω is compared with the compensated speed and the error is fed through a controller G_t to derive the electrical torque, T_L, for the load machine. The structure of the programmable load emulator system is such that there are two significant control loops: The first focuses on controlling the speed response of the drive motor, M_1

while the second control loop controls that of the load motor, M_2. The following sections will discuss in detail the control of the drive and load motor.

5.3.1 Control of the Drive Motor

For successful emulation, the controller, G_C of the drive motor, M_1 is required in deriving the electrical torque, T_e. The electrical torque, T_e, (or its associated current, I_e) serves as the input to the mechanical load emulation system. To obtain the required torque input, T_e, the controller, G_C is employed in controlling the speed of the emulated load, G_{em} such that it follows the speed demand, ω_{ref}. The result is that the desired electric torque signal, T_e is produced by the controller, G_C in the process. T_e is then harnessed as the electric torque input for the drive motor, M_1 and within the mechanical load emulator system, where it is specifically employed in deriving the dynamics of the compensating load torque, T_L that serves as the input to the load motor, M_2. Naturally, if the total dynamics of the coupled system (of M_1 and M_2), G equalled that of the mechanical load, G_{em}, for the same input torque, T_e, both systems will move with the exact same speed dynamics, ω and no compensating load torque, T_L will be required; in this case, there would be no need for emulation. The emulation strategy is especially required where the total dynamics of the coupled system, G and the mechanical system, G_{em} differ; in this case, for the same initial conditions, the drive torque, T_e, will cause different speed dynamics on the systems. Hence, in order to achieve a successful emulation strategy, it is necessary to derive the compensating load torque, T_L and the control of the load motor, M_2 enables the derivation of such a torque compensation; the process is subsequently described.

5.3.2 Control of the Load Motor

The control for the load motor is used in deriving a load torque, T_L, which acts to compensate the derived torque input, T_e to the emulation system. In order to derive the load torque, T_L, it is necessary to

1. Obtain the total torque dynamics,(T_e+T_L) required to force the coupled system, G to move with the same speed dynamics of ω as the mechanical load, G_{em}.
2. Obtain the difference between the derived total torque dynamics, (T_e+T_L) and electric torque input, T_e; the result of this gives the required load torque, T_L.

To achieve the first objective, a closed loop control system that has the coupled system, G, as its plant is designed. The designed control system, known as the *speed-tracking loop*, is shown in figure 5.4. The details of its design are described in section 5.3.3. The torque signal produced by its controller, G_t equals the sum of the electric drive and load torque, (T_e+T_L) which should normally be used to drive the coupled system, G at the desired speed dynamics, ω. In order to obtain the load torque, T_L, the difference between the sum of the torque dynamics, (T_e+T_L) and the derived torque input, T_e, obtained via the control of the drive motor, is calculated and this results in the load torque, T_L. The arrangement required to achieve this result can be observed in the dynamometer control in figure 5.2.

5.3.3 Modelling the Speed-tracking Loop

Within the mechanical load emulator, in order for the emulated load, G_{em} to have its dynamic properties truly represented (in the open and closed loop), the dynamics of the speed tracking loop must be cancelled out entirely by the compensation term, G_{COMP}. Figure 5.4 shows the open-loop system for the emulated load. In order to ensure accurate emulation of the mechanical load model, G_{COMP} must be inverse of the closed loop transfer function (CLTF) of the speed tracking loop, *i.e.*

$$G_{COMP} = \frac{1 + G_t G_{I_L} G}{G_t G_{I_L} G} \tag{5.4}$$

From figure 5.4, a good model of the compensation term, G_{COMP}, which cancels out the dynamics of the speed tracking loop, reduces the closed-loop control of the

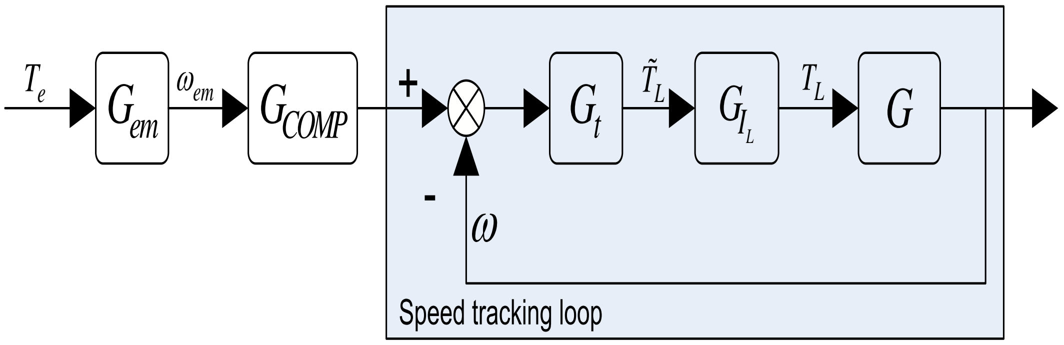


Figure 5.4: Open loop load emulation system

emulated load in figure 5.3 to the desired system in figure 5.5, which is simply a cascaded closed-loop system that comprises an inner current loop, G_{I_D} and an outer speed loop in which a speed controller, G_C acts on an emulated mechanical load, G_{em}.

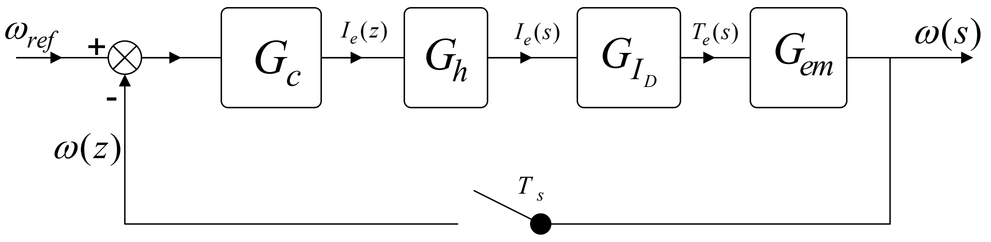


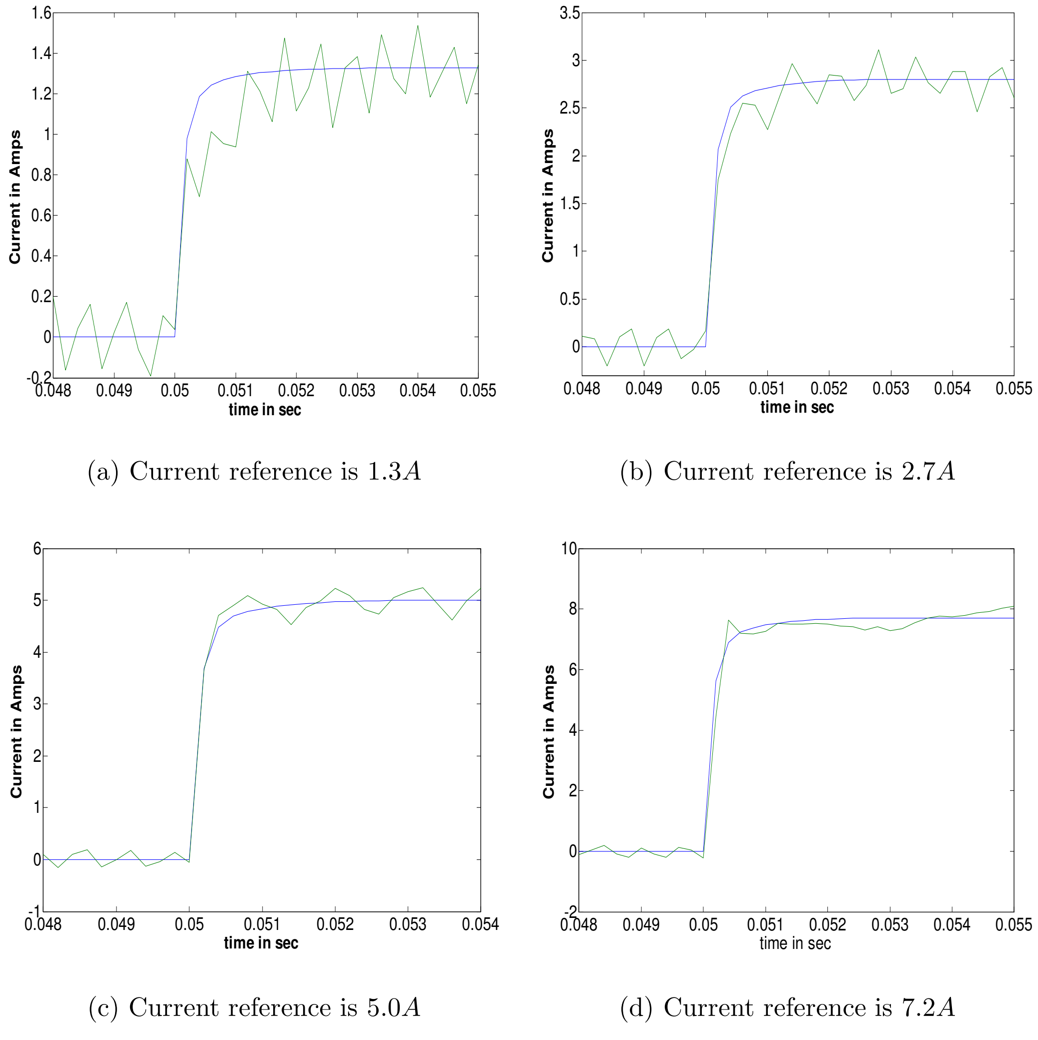
Figure 5.5: Reduced Simulation System

In order to model the compensation term, G_{COMP} it is necessary to initially model the speed tracking loop; this would involve modelling the inner current loop, G_{I_L} and the outer speed loop. The inverse of the speed tracking loop model will equal the required compensation term.

5.3.3.1 Modelling the Current Loop

The modelling of the current loop involved identifying (hand-tuning) the current (PID) controller parameters to ensure there was minimal error between the experimental and simulation responses. The inductance and resistance of the motor windings were provided in data sheets from the manufacturer and as such, these values were kept fixed during the modelling process. The details of the identified parameters are in table 5.1.

The results that show the successful modelling of the current loop are shown in figure 5.6. They show the responses produced by the experimental and simulation system for current references of $1.3A$, $2.7A$, $5.0A$ and $7.2A$: the experimental system response is highlighted in green while the corresponding simulation results are shown in blue.



(a) Current reference is $1.3A$

(b) Current reference is $2.7A$

(c) Current reference is $5.0A$

(d) Current reference is $7.2A$

Figure 5.6: Identification of Load Machine Current loop

The $2kHz$ ripple, which is ever present on the current waveforms shown in figure 5.6 can be attributed to the clamping of the rotor of the DC motor using a rotor bar, as a means to isolate the effects of the mechanical speed loop, during the current loop identification process. The clamping of the rotor, although adequate for the modelling process, was not perfect. As a result, slight oscillations of the rotor occurred despite the clamping measures taken. The observed $2kHz$ current ripples are the ever-present external disturbance that exist at the output of the current control loop due to the imperfect clamping of the rotor during the modelling process.

5.3.3.2 Modelling the Mechanical Speed Loop

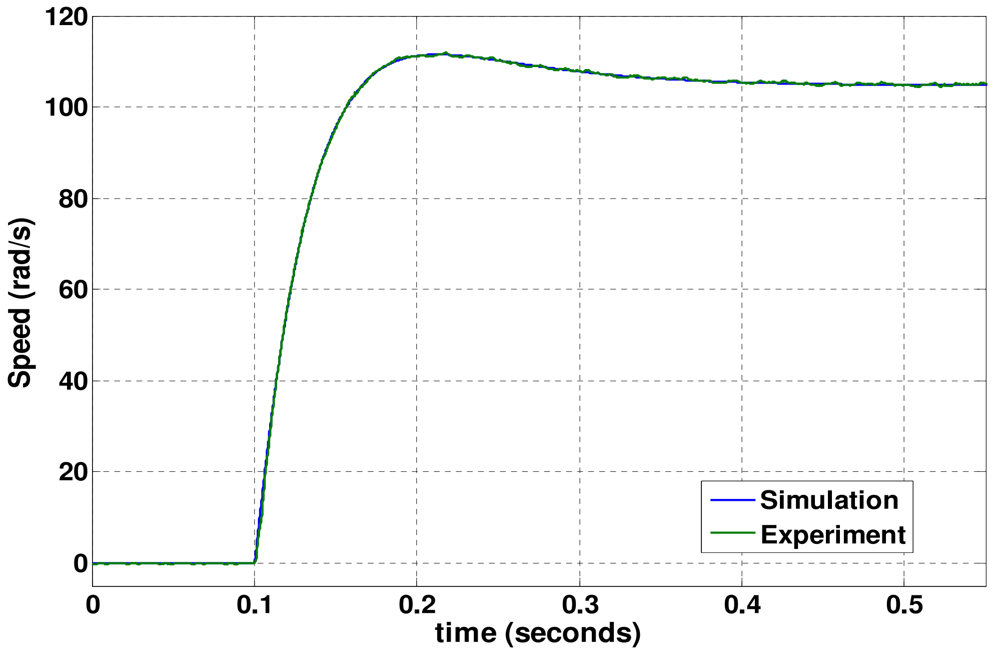
The total Inertia, J and Friction B for the coupled system comprising the drive and load machine, G have been identified by employing a similar hand tuning approach as in section 5.3.3.1. There was no need to identify the speed controller, G_t as this was implemented digitally and used on both the experimental system, in obtaining data, and the simulation system, in the tuning process for identifying the mechanical load parameters.

Figure 5.7 shows the results of the comparison between the plant's speed response data at two different speed references and the corresponding simulation data obtained by adopting the identified lumped mechanical model for G. The results highlight the quality of the model obtained for the parameters of the mechanical plant. Table 5.1 summarises the results of the system identification process; it shows the identified parameters and their respective values.

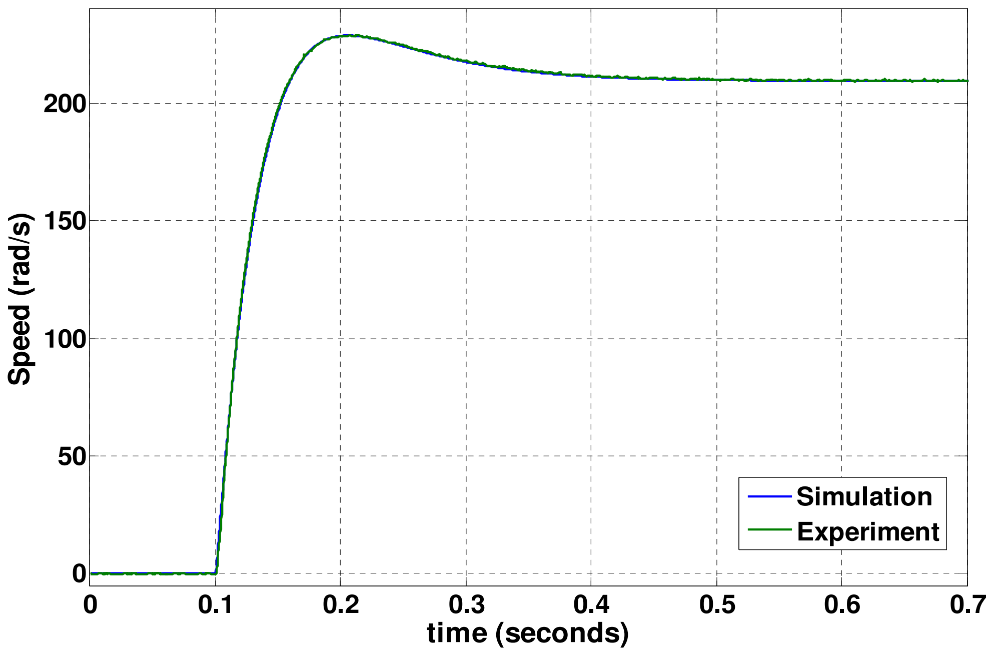
The identified speed-tracking loop digital model is a proper discrete function with the denominator being two orders (z^2) greater than the numerator. This implies that the G_{COMP} term, which is the inverse of the speed-tracking loop model, is an improper

Controller Parameters				*Mechanical Model*	
k	$a(rad/s)$	$b(rad/s)$	$c(rad/s)$	$J(kgm^2)$	$B(Nms)$
6	1000	1230	2150	0.00011	0.00035

Table 5.1: Identified System Parameters



(a) Speed response (105 rad/s) using identified mechanical parameters.



(b) Speed response (210 rad/s) using identified mechanical parameters

Figure 5.7: Speed response using identified mechanical parameters

discrete function. In order to successfully implement its transfer function block within Simulink, a factor of, z^2 is added to the denominator of the G_{COMP} term. The result is both the numerator and denominator become equal. The extra delays introduced in the denominator are compensated for by introducing a factor z to the model of the emulated load, G_{em} and the model of the drive motor current control loop, G_{I_D}. As a result, there would be two advance terms which will cancel the two delays introduced in the compensation term, G_{COMP}.

5.4 Electronic Drive Controller Characteristics

The very nature of the optimisation process demanded that the controller, G_C to be optimised was implemented in the digital $z - domain$ through a microprocessor. The reason is the 'nature' of the optimisation process is such that both the structure and parameters of the digital controller are simultaneously and regularly updated in a bid to discover a best possible robust controller for the closed loop system under consideration. The digital controller implementation provides a suitable option for regularly making updates on both the structure and parameters; it forms the focus of the automated optimisation process, which is made possible through the application of the evolutionary algorithms. The design of the digital controller is implemented within MATLAB/Simulink; its inherent characteristics, which enable the simultaneous optimisation of both its structure and parameters will form the subject of the subsequent sections. The description will focus on its structure, parameter settings and the anti-windup scheme adopted.

5.4.1 Controller Structure

The structure of the digital controller distinctly incorporates a Proportional plus Integral (PI), a real pole and zero, and a complex pole and zero pair. The structures made possible for selection by virtue of the controller design are the Proportional plus

Integral (*PI*), PI plus Lead/Lag (*PI + real zero/pole*), Third order (*PI + complex zeros/poles*) and a Fourth order controller (*PI + real zero/pole + complex zeros/poles*) structure. The different controller structures are selected based on the values of the flags, F_0 and F_1. These flags are encoded within the individuals (possible solutions). The combination of the flags and the resulting controller structure selected is summarised in table 5.2.

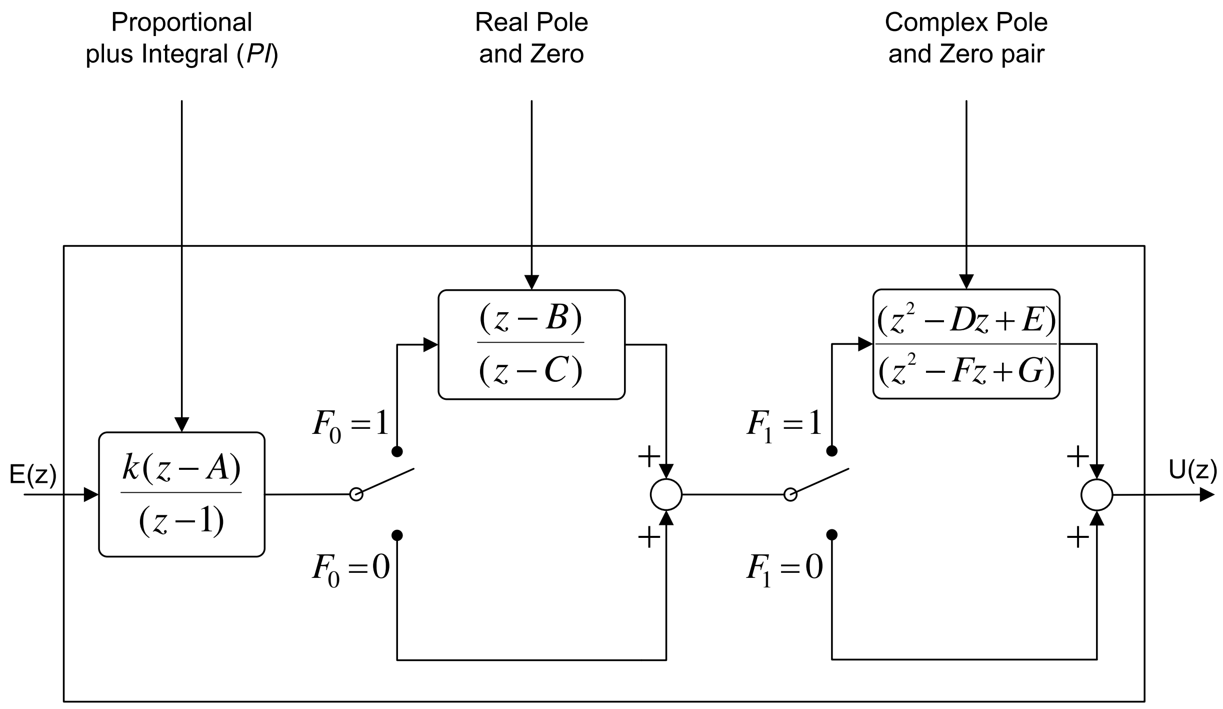
All the controller structures include the PI; this ensures that for any randomly chosen controller, there is no steady state error between the step demand and output speed of the control system. Although more options can be made regarding controller structure, the need to ensure zero steady state error makes the incorporation of the PI necessary and thus reduces the options on controller structures evaluated. Figure 5.8(a) gives the basic layout of the digital controller employed: it shows the possible structures within the digital controller, which depend on the different flag settings. The simulink schematic of the digital controller is shown in Appendix A.1

5.4.2 Controller Anti-windup Scheme

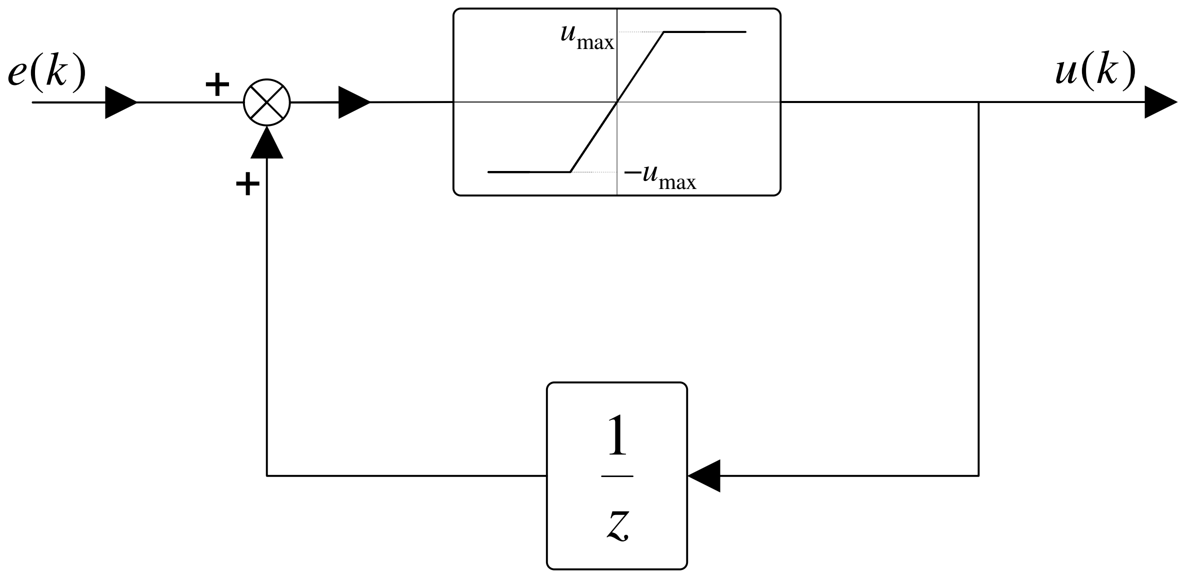
A necessary part of the controller structure is the integrator anti-windup mechanism. Every system requires a drive (actuator) signal to force it to do as demanded. Within a feedback control system, the controller provides the actuation signal to force the system to act accordingly. There exists a maximum actuation signal for every real system and, when the signal reaches this maximum, the controller, in effect, becomes

F_1	F_0	Structure	Parameters
0	0	*PI*	k, A
0	1	*PI + real zero/pole*	k, A, B, C
1	0	*PI + complex zeros/poles*	k, A, D, E, F, G
1	1	*PI + real zero/pole + complex zeros/poles*	k, A, B, C, D, E, F, G

Table 5.2: Controller structure and corresponding parameters



(a) layout of digital controller



(b) integrator anti-windup scheme

Figure 5.8: Controller Implementation

disconnected from the system it attempts to control. This is because the output produced by the controller (after it reaches the maximum) has no further effect on the actuation of the system it is supposed to control. The effect of this becomes especially significant in control design with the incorporation of integrators in controllers of feedback systems.

The controller, with an integrator, always attempts to cancel out errors eventually when the system reaches steady state. To achieve this, the error (the difference between the demand and output of the control system) is fed into the controller and its integrator sums up the error over time and outputs an actuating signal proportional sum of the error over time, to affect the motion of the system under consideration. If the actuation signal of the system reaches its saturation (maximum) limit, any further signal produced by the controller is of no consequence on the system's response.

There is the possibility of errors existing when the system's actuation input saturates. Due to the presence of these errors, the integrator continues production of actuating signals proportional to the existing errors. If the actuation input for the system to be controlled remains in saturation for long periods, the integrator would eventually output very high values because the error is not being reduced as the system fails to respond according, given it is in saturation. As a result these high controller outputs, the integrator is said to (*wind-up*). Once, the actuation signal reduces from the maximum, the controller becomes reconnected to the system; but it could take an awfully long period for the integrator to readjust its output or (*wind-down*) to give that required output to achieve the desired system response. From the description, to ensure accurate and efficient control, it is necessary to stop the integrator from integrating to very high error values as avoiding this improves the responsiveness of the system by reducing the wind-down time for the integrator. The approach taken is to limit the output of the integrator to a maximum value, which corresponds to the maximum actuation signal for the system to be controlled. This ensures that it never needs to wind-down from unnecessarily high values. The incorporation of the anti-windup scheme within the digital controller is shown in figure 5.8(b). It represents the integrator's action in a difference equation format and it shows the actuation limit placed within the loop, which bounds the output of the integrator. Its value is set to be the maximum and minimum actuation signals for the mechanical load system which the controller acts on. To place it context, for the research project, the actuation limits corresponds to the current/torque inputs to electric drive system.

5.4.3 Controller Parameters

The parameters to be optimised are the gain, poles and zeros of the digital controllers and the flags that define the structure of the digital controller. There is the need to specify the limits for each of the parameters in order to guide the optimisation algorithms in their search procedure. The limits are defined such that the position of each pole and zero lie within the unit circle. This is a necessary precaution to ensure that each of the randomly selected poles or zeros have a stable response. Table 5.3 shows the limits of each pole and zero parameters optimised by the algorithms. The limits also ensure that the values for each flag is actually implemented as a binary number. The number of parameters to be optimised for any one digital controller selected randomly would vary from a minimum of four, for a PI controller (and its flags), to a maximum of ten parameters for a fourth order controller.

5.5 Mechanical Load

The mechanical load emulator has been employed to emulate particular load dynamics that represent typical industrial mechanical loads. These emulated loads serve as the operating conditions for controller optimisation and as such the design process will

Structure	*PI*		*Pole & Zero*		*Complex Pole & Zero pair*			
Parameters	k	A	B	C	D	E	F	G
Range	[0, 2]	[-1, 1]	[-1, 1]	[-1, 1]	[-2, 2]	[-1, 1]	[-2, 2]	[-1, 1]

Table 5.3: Range of Controller Parameters

lend itself well under actual industrial conditions. The two mechanical loads considered for the purpose of investigating the effectiveness of the optimisation algorithms in designing robust control systems for industrial electric drives are: (1) Stiff-shaft mechanical load and (2) Flexible-shaft mechanical load. The mechanical loads investigated in the research project are interesting because they are quite commonly found in different industrial applications [60]. These mechanical loads are synonymous with the loads experienced by industrial drives in operating lifts, hoists, fans, *etc.* They can also be used to represent the load dynamics experienced by car tyres under different load conditions [62]. The section provides a detailed description of the mechanical loads that are investigated during the research project.

5.5.1 Stiff-shaft mechanical load

The mechanical load system simply comprises a drive motor and a load motor coupled together by means of a stiff shaft. The drive motor has its characteristic inertia, J_M and friction, B_M. The load motor, which is connected to the drive motor, has inertia, J_L and friction B_L. The mechanical load system can be represented as in figure 5.9. The transfer function of the stiff-shaft mechanical load, given as (5.11), is derived by initially considering the equations of the two motors separately and then combining the individual equations to yield the overall transfer function. A useful assumption made in the derivation of the transfer function of the linear mechanical load is (5.8).

The assumption is justified because the motors are connected using a stiff shaft hence there is negligible backlash. As a result, the speed of the drive motor can be assumed to be exactly equal to the load motor speed. The experimental implementation of the mechanical load within the lab is achieved using the programmable load emulator described in chapter 5.3. The bandwidth limitations of the programmable load emulator and the maximum load torque of the drive motor permit only a certain range of mechanical loads to be emulated [58], [60]: the total inertia, J_{em} of the mechanical load is chosen to vary between the minimum possible value of J and a maximum value of $5J$. A similar variation pattern is selected for the total friction of the system, B_{em}

with a minimum value of B and a maximum value of $5B$. The minimum values of the total inertia, J_{em} and friction, B_{em} correspond to the total inertia, $0.00013kgm^2$ and total friction, $0.00052Nms$ of the mechanical drive train used for the emulation.

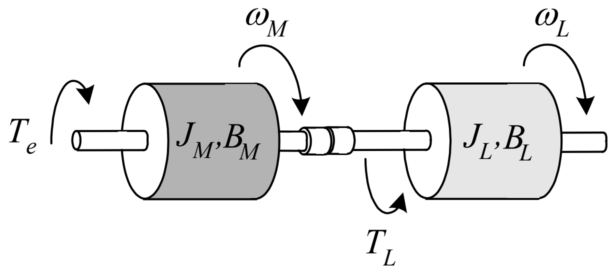


Figure 5.9: Stiff-shaft mechanical load

$$\begin{aligned}
T_L(s) &= J_L s\omega_L(s) + B_L\omega_L(s) && (5.5)\\
T_e(s) &= J_M s\omega_M(s) + B_M\omega_M(s) + T_L(s) && (5.6)\\
T_e(s) &= J_M s\omega_M(s) + B_M\omega_M(s) + J_L s\omega_L(s) + B_L\omega_L(s) && (5.7)\\
\omega_M(s) &= \omega_L(s) && (5.8)\\
T_e(s) &= (J_M + J_L)s\omega_L(s) + (B_M + B_L)\omega_L(s) && (5.9)\\
J_{em} &= J_M + J_L\\
B_{em} &= B_M + B_L && (5.10)\\
G_{em}(s) &= \frac{\omega_L(s)}{T_e(s)} = \frac{1}{J_{em}s + B_{em}} && (5.11)
\end{aligned}$$

5.5.2 Flexible-shaft mechanical load

The flexible-shaft mechanical load simply comprises two motors, the drive and the load motor connected together by a spring or flexible shaft. Within the mechanical load model, the friction of the two motors have been neglected. The drive motor has its inertia represented as $J_{M_{em}}$ and the load motor inertia is symbolised by $J_{L_{em}}$. the flexible transmission between the two motors has its characteristic damping coefficient, D_{em} and a spring constant, K_{em}. As a result of the flexible transmission between the drive and the load motor, there is a *backlash* present between the motors. The flexible-shaft mechanical load can be represented as in figure 5.10.

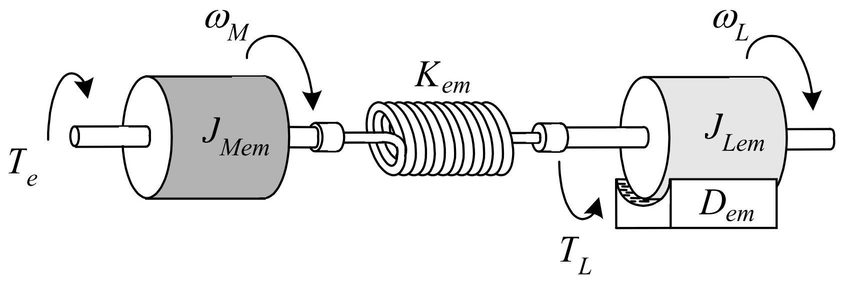
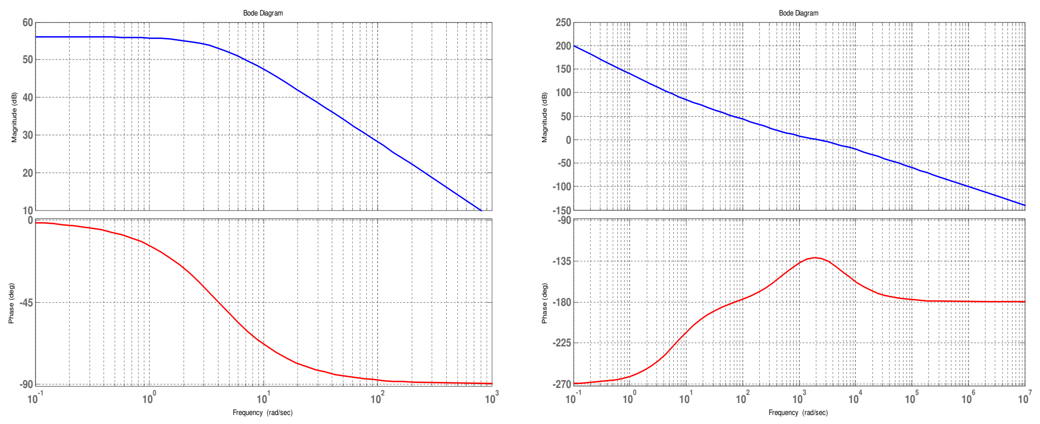


Figure 5.10: Flexible-shaft mechanical load [60]

$$
\begin{aligned}
J_{M_{em}} s\omega_M &= T_e(s) - T_L(s) && (5.12)\\
T_L(s) &= J_{L_{em}} s\omega_L && (5.13)\\
T_L(s) &= D_{em}(\omega_L - \omega_M) - K_{em}\left(\frac{\omega_L}{s} - \frac{\omega_M}{s}\right) && (5.14)\\
\frac{\omega_L(s)}{\omega_M(s)} &= \frac{\frac{D_{em}s}{K_{em}} + 1}{\frac{J_{L_{em}}}{K_{em}}s^2 + \frac{D_{em}}{K_{em}}s + 1} && (5.15)\\
T_e(s) &= J_{M_{em}} s\omega_M + J_{L_{em}} s\omega_L && (5.16)\\
T_e(s) &= J_{M_{em}} s\omega_M + J_{L_{em}}\left(\frac{s\omega_L}{s\omega_M} \times s\omega_M\right) && (5.17)\\
T_e(s) &= J_{M_{em}} s\omega_M + J_{L_{em}}\left(\frac{D_{em}s + K_{em}}{J_{L_{em}}s^2 + D_{em}s + K_{em}} \times s\omega_M\right) && (5.18)\\
\frac{\omega_M(s)}{T_e(s)} &= \frac{\frac{J_{L_{em}}}{K_{em}}s^2 + \frac{D_{em}}{K_{em}}s + 1}{s(J_{M_{em}} + J_{L_{em}})\left(\frac{J_{M_{em}}J_{L_{em}}}{(J_{M_{em}} + J_{L_{em}})K_{em}}s^2 + \frac{D_{em}}{K_{em}}s + 1\right)} && (5.19)\\
G_{em}(s) &= \frac{\omega_L(s)}{T_e(s)} = \frac{\omega_L(s)}{\omega_M(s)} \times \frac{\omega_M(s)}{T_e(s)} && (5.20)\\
G_{em}(s) &= \frac{\frac{D_{em}}{K_{em}}s + 1}{s(J_{M_{em}} + J_{L_{em}})\left(\frac{J_{M_{em}}J_{L_{em}}}{(J_{M_{em}} + J_{L_{em}})K_{em}}s^2 + \frac{D_{em}}{K_{em}}s + 1\right)} && (5.21)
\end{aligned}
$$

In a similar fashion as the stiff-shaft mechanical load and due to the bandwidth limitations of the emulator, its inertia, $J_{L_{em}}$ has been chosen to vary between J_L

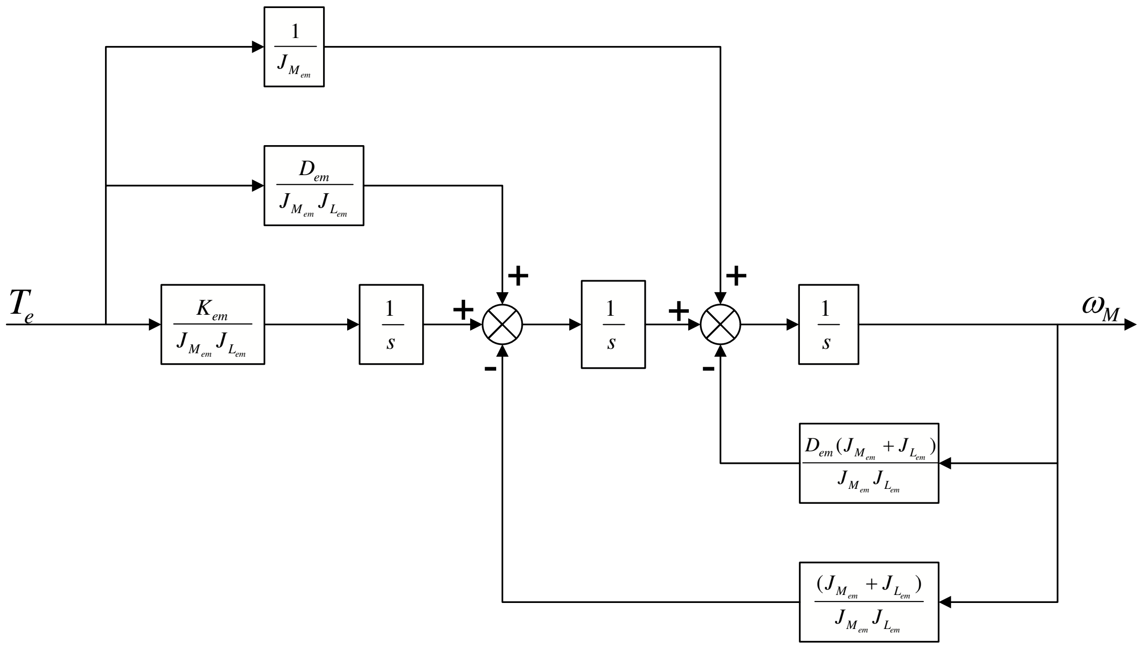
and $5J_L$. J_L has the chosen value of $0.00007kgm^2$. The damping coefficient of the spring, D_{em} has been selected to vary between D and $5D$ and its minimum value, D has been selected as $0.1Nm/rad/s$. The spring constant K_{em} has been chosen as $2.0Nm/rad$ and the speed *backlash* between the motors has been selected as $5rad/s$. These parameters of the flexible shaft mechanical system were also selected for the purpose of visualising the effectiveness of the automated optimisation procedure when compared with the hand-tuned design and through experimentation the selected value was deemed suitable to demonstrate this.



(a) Bode Plot for Stiff Shaft Load (b) Bode Plot for Flexible Shaft Load

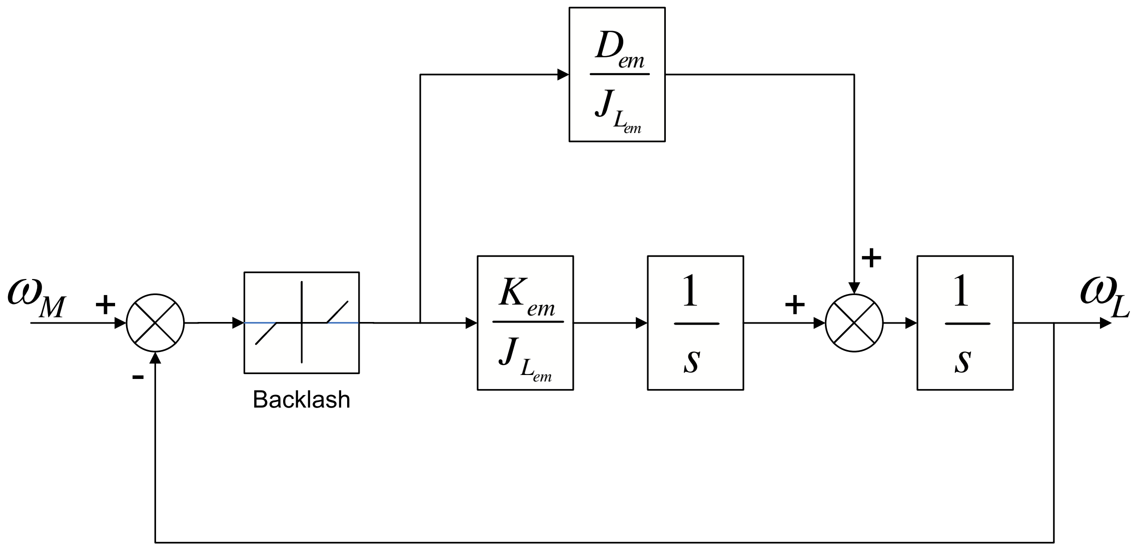
Figure 5.11: Bode Plots for Average Models of Mechanical Load

The transfer function of the nonlinear load with flexible shaft has been derived by considering, separately, the transfer functions of the different parts of the mechanical load system and then combining them to yield the its overall transfer function. The steps to the derivation have been shown between equations (5.12) to (5.21). The Schematic for the difference equation (5.21) is in Appendix A.2. It can be represented in continuous domain as in figure 5.12. Its implementation combines equations (5.15) and (5.19) and in between them exist a flexible shaft connection, which causes a backlash, in terms of speed, between the motors. The bode plots for the average models of the stiff and flexible shaft mechanical loads are shown in figure 5.11. Within the stiff shaft load, J_{em} and B_{em} have values of $3J$ and $3B$ respectively while in the flexible shaft load model, $J_{L_{em}}$ and D_{em} have values of $3J_L$ and $3D$ respectively.



T_e
$\frac{1}{J_{M_{em}}}$
$\frac{D_{em}}{J_{M_{em}} J_{L_{em}}}$
$\frac{K_{em}}{J_{M_{em}} J_{L_{em}}}$
$\frac{1}{s}$
$\frac{D_{em}(J_{M_{em}} + J_{L_{em}})}{J_{M_{em}} J_{L_{em}}}$
$\frac{(J_{M_{em}} + J_{L_{em}})}{J_{M_{em}} J_{L_{em}}}$
ω_M

(a) Transfer function Schematic between Electrical Torque Input and Drive Motor Speed



(b) Transfer function Schematic between Drive and Load Motor Speed

Figure 5.12: Schematic of Flexible Shaft Mechanical Load

5.6 Emulation of Mechanical Loads

The dynamometer control strategy has been used to emulate both a stiff and flexible shaft mechanical load. The drive controller, G_C used in the emulation strategy is a PI controller designed via the root-locus method to give the best possible system response. External load disturbances are considered in both cases within the control of the mechanical load. The disturbance torque has magnitude of 0.15 Nm, which is a third of the rated-torque of each DC motor employed in the emulation system.

The speed demand profiles used in demonstrating the emulation strategy last for $5s$: From $0s$ to $0.1s$, a speed demand of $0rpm$ is requested; from $0.1s$ to $2s$, there is a speed demand of $0rpm$ to $1000rpm$; finally, from $2s$ to $5s$, a demand speed of $1000rpm$ to $2000rpm$ is requested and a disturbance torque is introduced at $3.5s$.

5.6.1 Emulation of the Stiff shaft Mechanical Load

The dynamics of the Stiff shaft mechanical load described in section 5.5.1 have been accurately experimentally reproduced using the programmable load emulator. In order to emulate the stiff-shaft mechanical load, its transfer function, described in equation (5.11), is implemented as a difference equation, using the zero-order hold approximation. The difference equation incorporates an advance term which cancels one of the two delays introduced within the transfer function for the compensation term, G_{COMP}. In accurately reproducing the dynamics of the stiff shaft mechanical load, three different combinations of total inertia, J_{em} and total friction, B_{em} have been considered to demonstrate the accuracy of the strategy for load emulation harnessed:

- **Case 1:** $J_{em} = J = 0.00013\ kgm^2$; $B_{em} = 5B = 0.00260\ Nms$
- **Case 2:** $J_{em} = 3J = 0.00039\ kgm^2$; $B_{em} = 3B = 0.00156\ Nms$
- **Case 3:** $J_{em} = 5J = 0.00065\ kgm^2$; $B_{em} = B = 0.00052\ Nms$

The plots shown in the figures 5.13, 5.14 and 5.15 validate the emulation strategy. They show the emulation achieved using the experimental set-up and compare the results with the response obtained using the reduced simulation model in figure 5.5. The plots show the emulation of the stiff shaft mechanical load at the three different combinations of the total inertia and friction values and it also includes the load disturbance applied at the output of the controller at time equals 3.5 *seconds*.

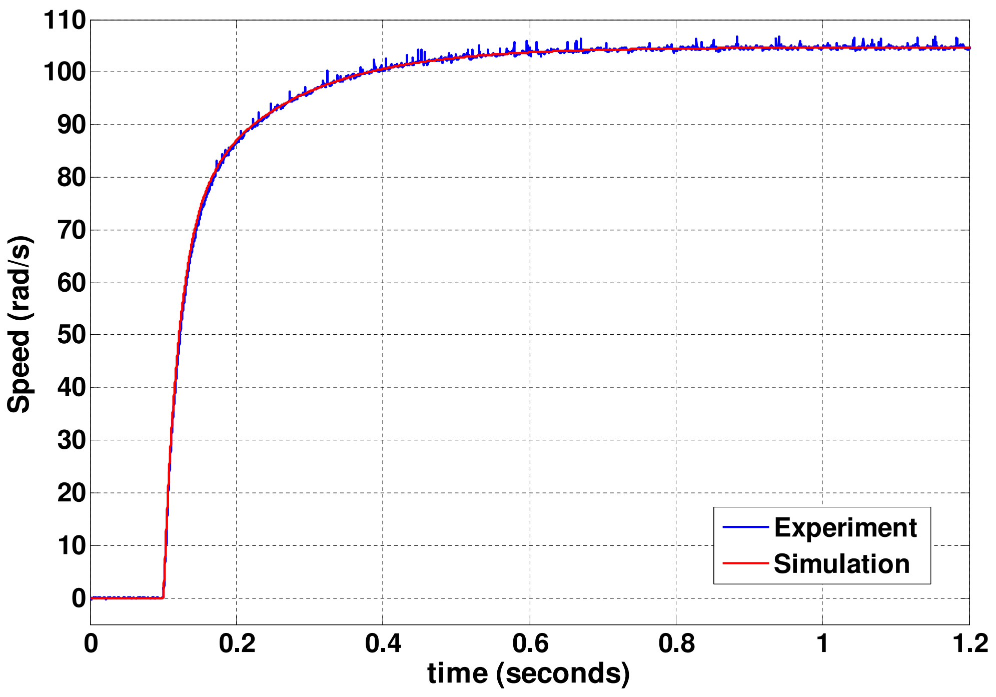
5.6.2 Emulation of Flexible Shaft Mechanical Load

The dynamics of the flexible shaft mechanical load described in section 5.5.1 have also been accurately reproduced using the dynamometer control strategy described. For the purpose of digitally implementing the flexible shaft load to be emulated, its transfer function, given in equation (5.21), is discretised using the zero-order hold approximation and implemented as a difference equation and in the form of a Simulink schematic. The difference equation incorporates an advance term which cancels one of the two delays introduced within the transfer function for the compensation term, G_{COMP}.

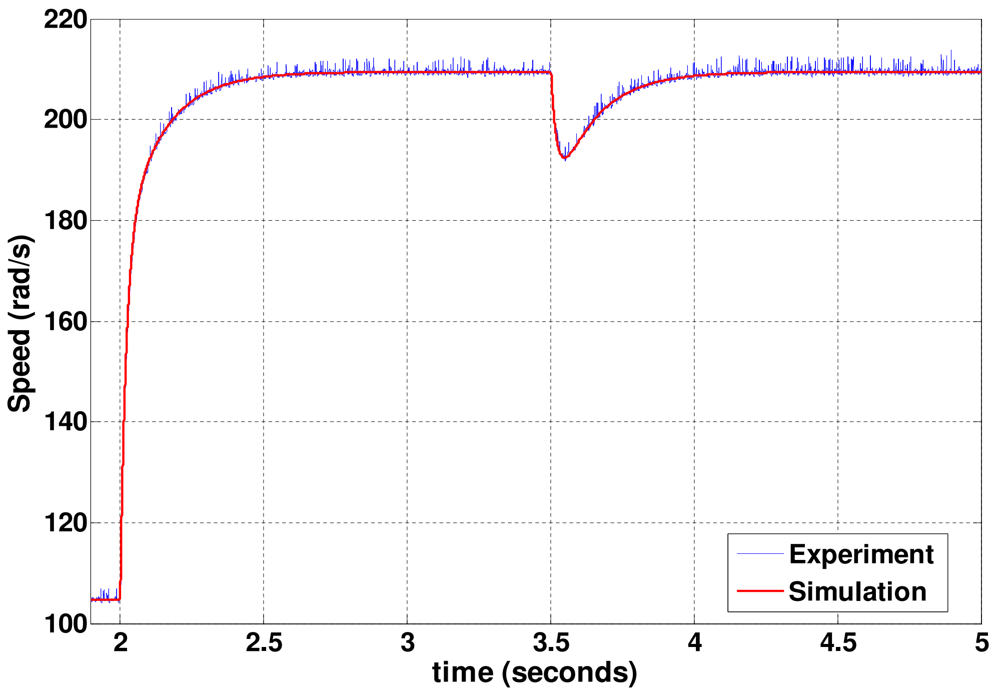
In order to demonstrate the validity of the emulation strategy adopted, three combination of load motor moment of inertia, $J_{L_{em}}$ and damping coefficient, D_{em} have been tested. In each of the following cases, the drive motor inertia, $J_{M_{em}}$ is 0.00007 kgm^2, the spring constant, K_{em} is 2.0 Nm/rad and there is a constant *backlash* of 5 rad/s. The cases considered are:

- **Case 1:** $J_{L_{em}} = J_L = 0.00014\ kgm^2$, $D_{em} = 5D = 0.5\ Nm/rad/s$
- **Case 2:** $J_{L_{em}} = 3J_L = 0.00042\ kgm^2$, $D_{em} = 3D = 0.3\ Nm/rad/s$
- **Case 3:** $J_{L_{em}} = 5J_L = 0.00070\ kgm^2$, $D_{em} = D = 0.1\ Nm/rad/s$

Figures 5.16, 5.17 and 5.18 show the comparison between the results of the reduced simulation model and those obtained from the experimental set-up . This further

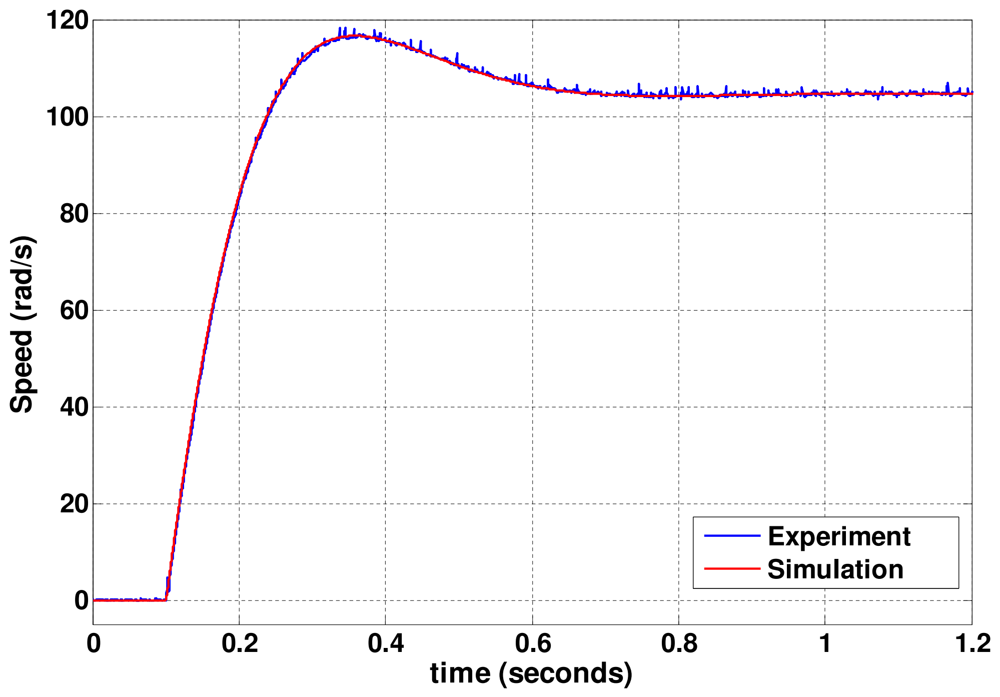


(a) Speed response to step demand of 0 to 1000 rpm

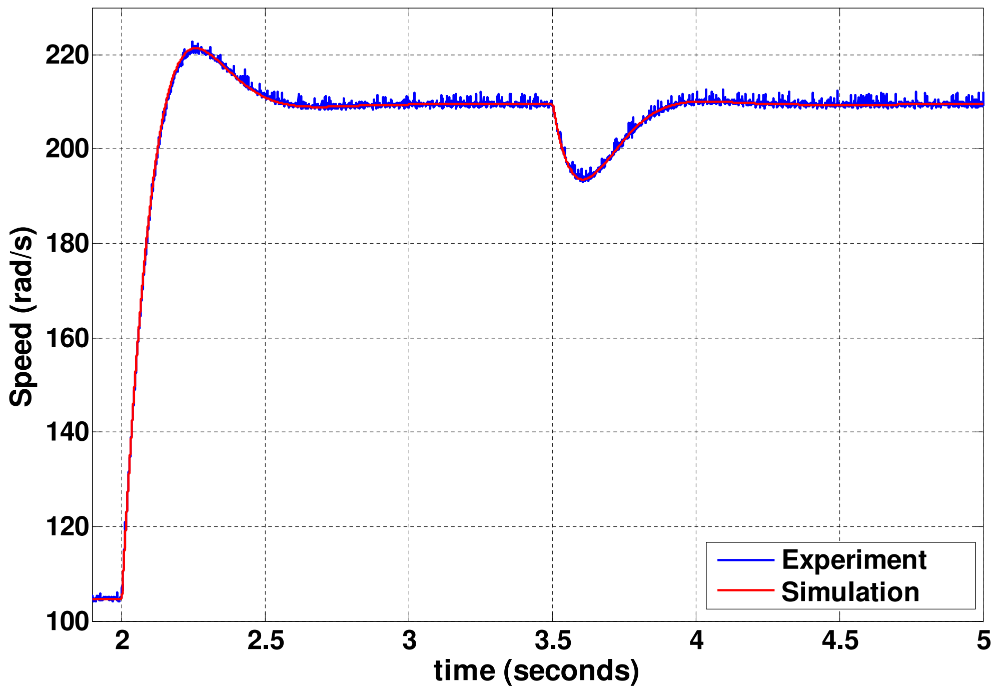


(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.5s$

Figure 5.13: Stiff shaft load emulation - Inertia of J and Friction of $5B$

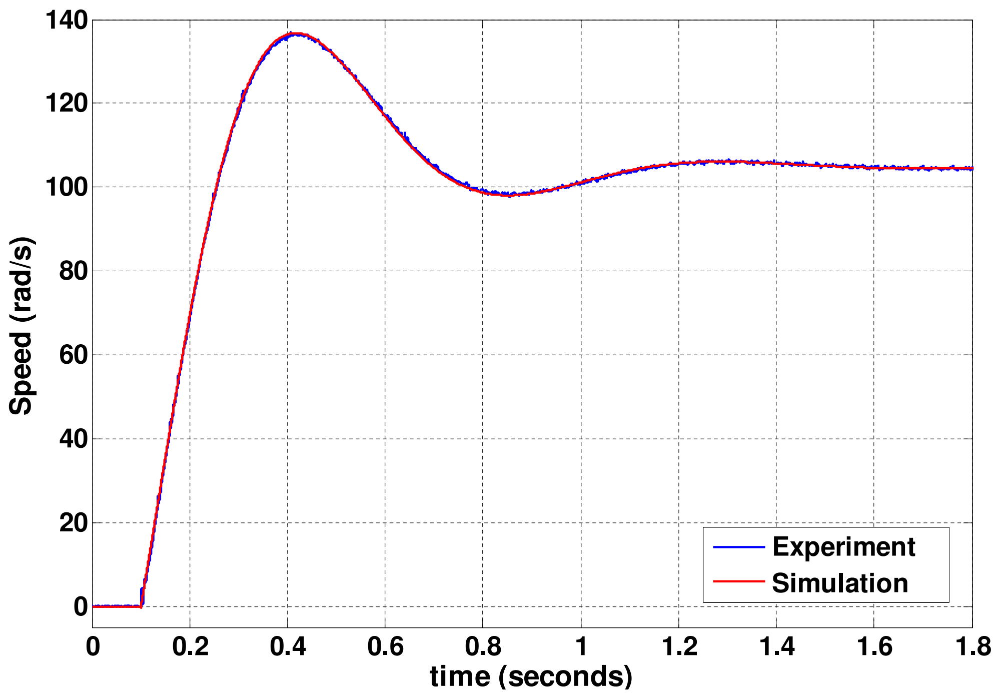


(a) Speed response to step demand of 0 to 1000 rpm

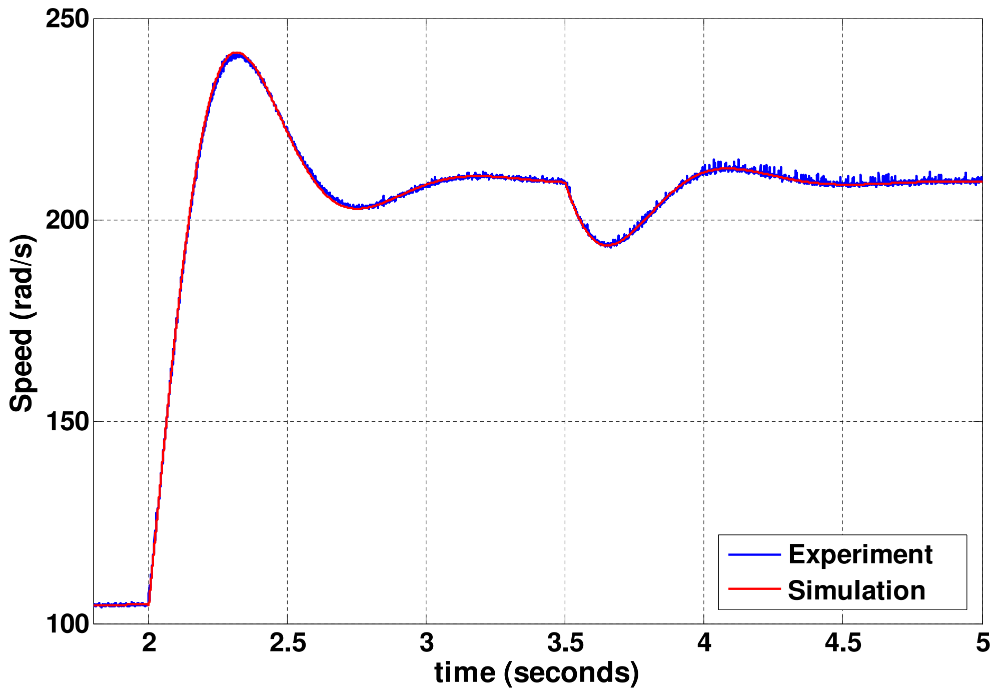


(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.5s$

Figure 5.14: Stiff-shaft load emulation - Inertia of $3J$ and Friction of $3B$



(a) Speed response to step demand of 0 to 1000 rpm



(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.5s$

Figure 5.15: Stiff-shaft load emulation - Inertia of $5J$ and Friction of B

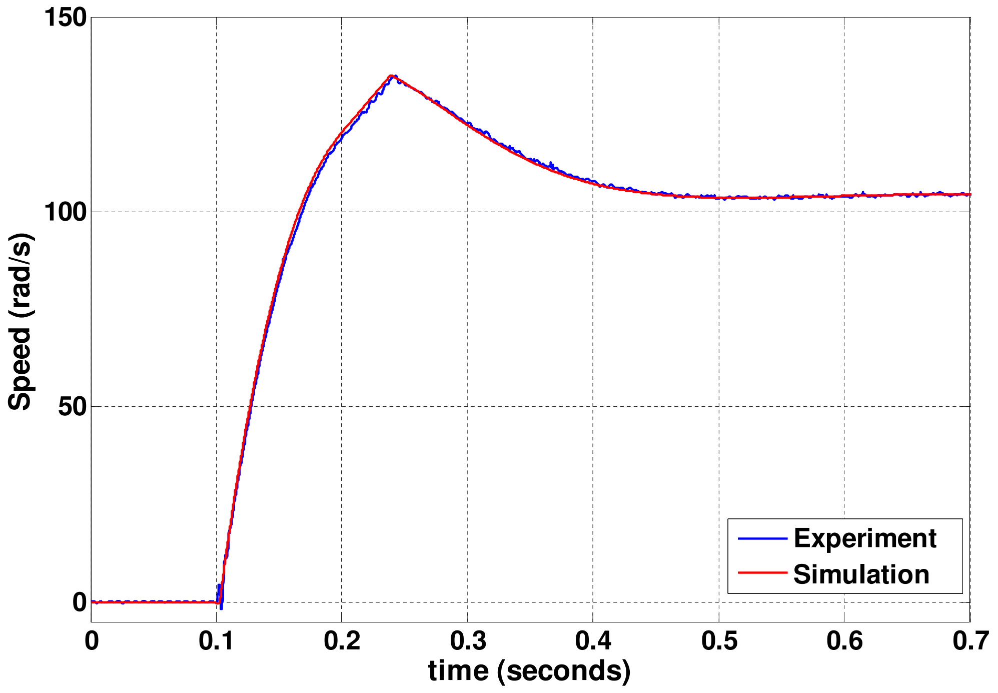
validates the emulation employed for the research project. Load disturbance are also added to the outputs of the closed loop control system at time equals 3.5 *seconds*.

5.7 Limitations of the Emulation System

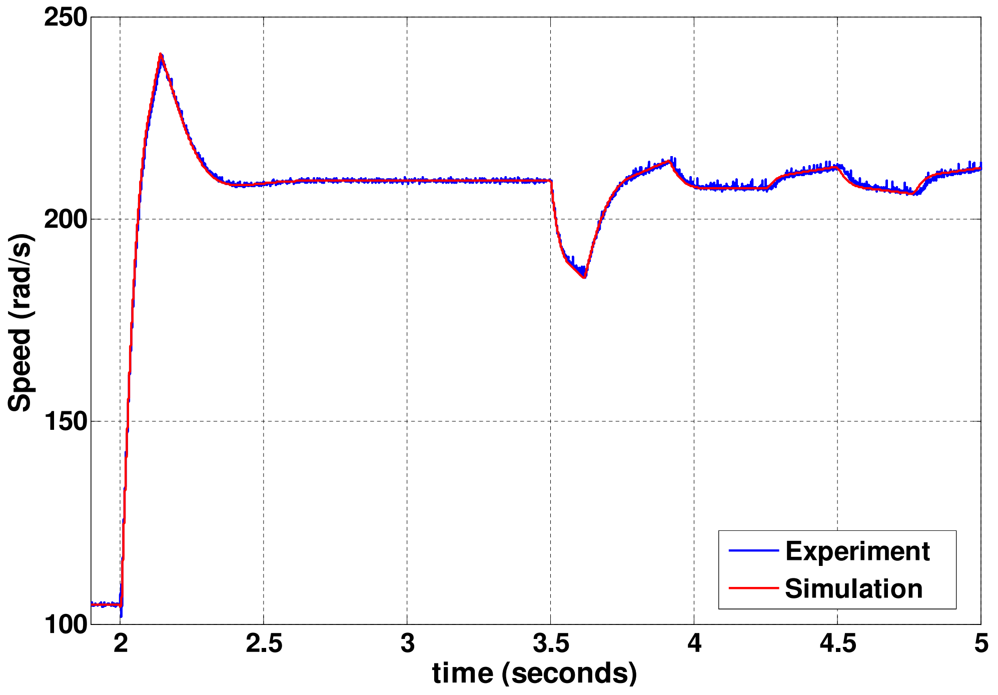
There are system limitations that restrict the characteristics of mechanical loads that can be emulated using the dynamometer. The dynamometer torque response is one of those factors [59], [60]. Another factor lies in the bandwidth of the active low-pass filter used to reduce the noise from the speed signal produced by the tachogenerator. For the experimental rig considered in this chapter, the torque bandwidth is fairly modest and lies within 1000 - 1200 Hz, limiting the emulated mechanical dynamics to frequencies up to 200 - 220 Hz. This estimate is based on the practical recommendation that, for the speed tracking loop, the bandwidth of the outer (speed) loop should be five times less than the inner (current) loop [63]. The bandwidth of the active low-pass filter designed is approximately 200 Hz and it coincides with the resulting boundaries imposed by the limits of the torque dynamics.

During the emulation of any mechanical load, it is necessary that the output of the speed tracking loop controller, G_t never goes into saturation. The compensation would be invalid if this occurs. This is because the exact torque dynamics, required to force the dynamic behaviour of the load machine to match with that of the mechanical plant model being emulated, would be impossible to obtain. This is equivalent to clipping off certain parts of the required controller dynamics. A point to note is that the controller can have any structure; this includes a Proportional-plus-Differential (PD), Proportional-plus-Integral (PI), *lead* or Proportional(P) controller. There is also no need for an integrator to maintain zero error for the tracking loop since the tracking loop itself is not subject to any disturbance input [58], [59], [61].

The minimum inertia that can be emulated is made equal to the total inertia of the coupled system, which includes the inertias of the drive and load motors [58], [59], [61]. Emulating systems with inertias lower than this value could result in saturating the

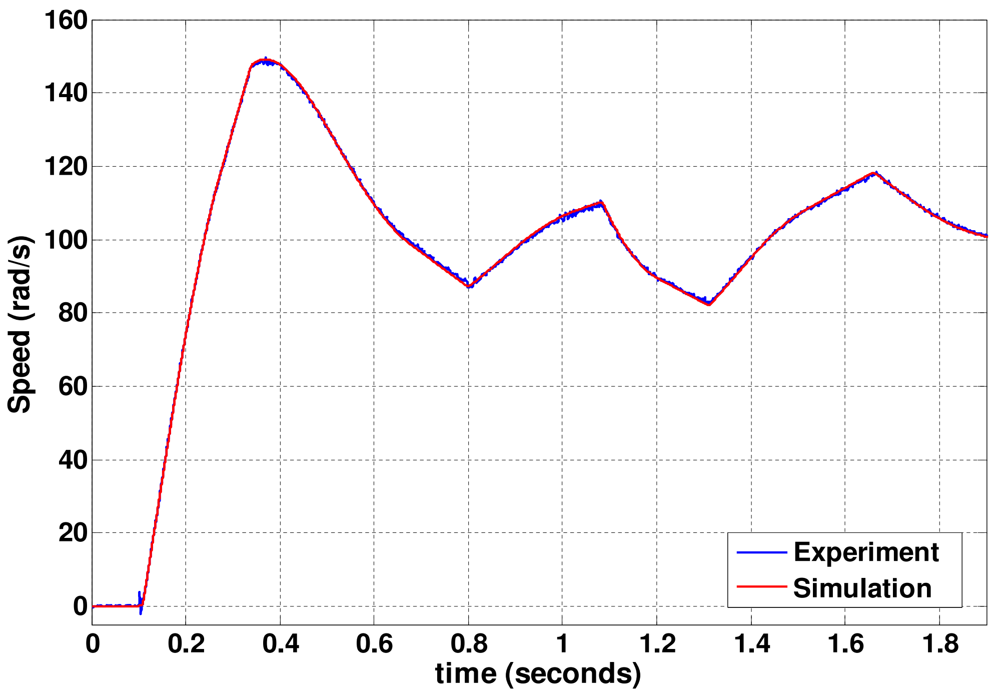


(a) Speed response to step demand of 0 to 1000 rpm

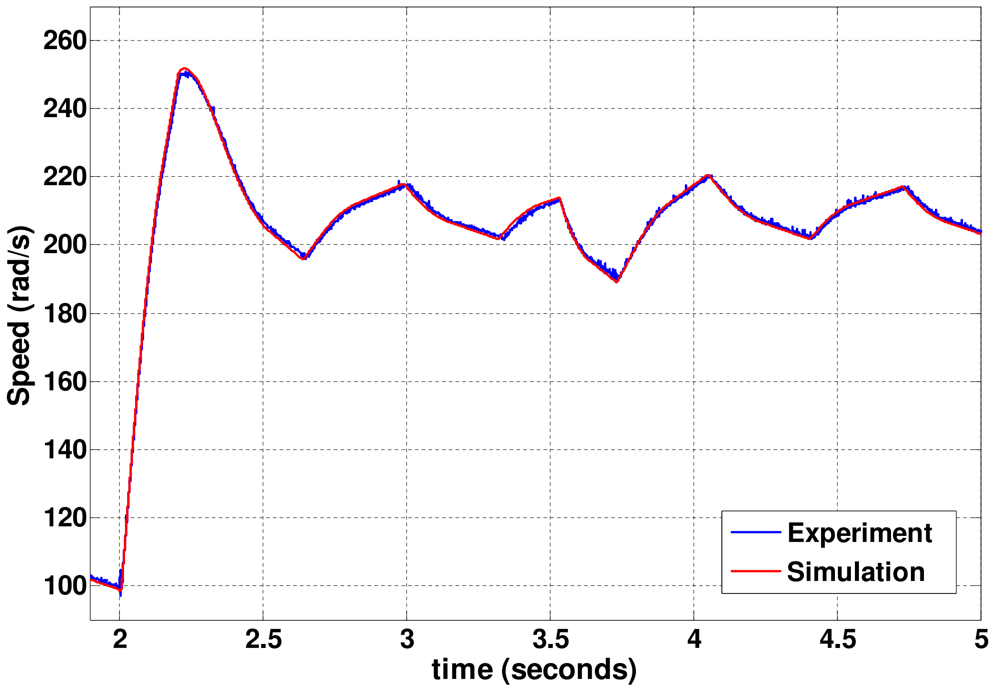


(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.5s$

Figure 5.16: Flexible shaft load emulation - Inertia of J_L and Damping of $5D$

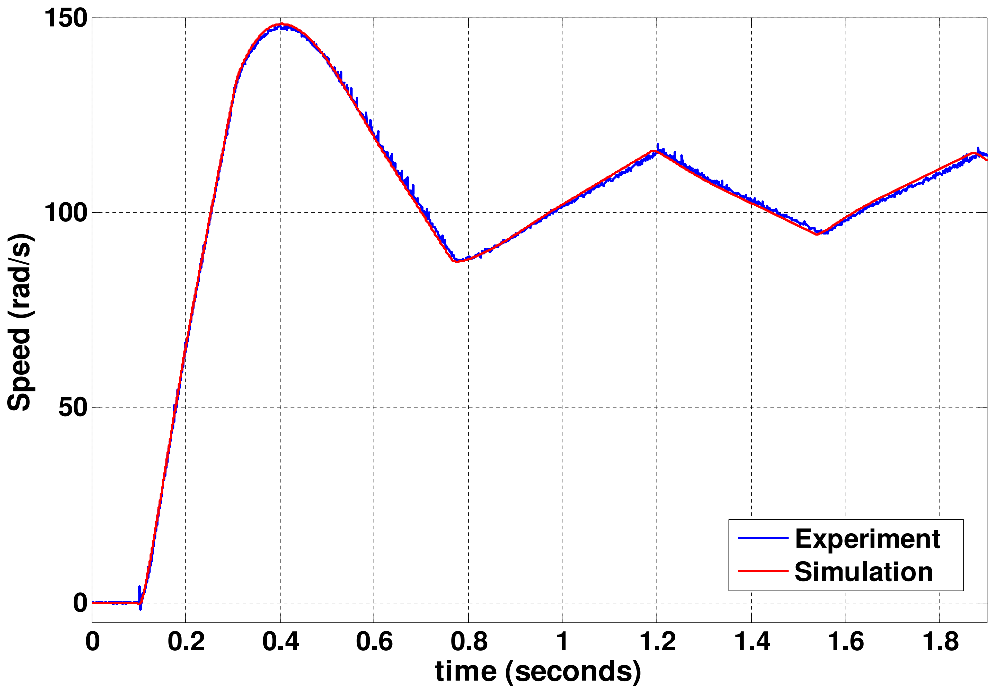


(a) Speed response to step demand of 0 to 1000 rpm

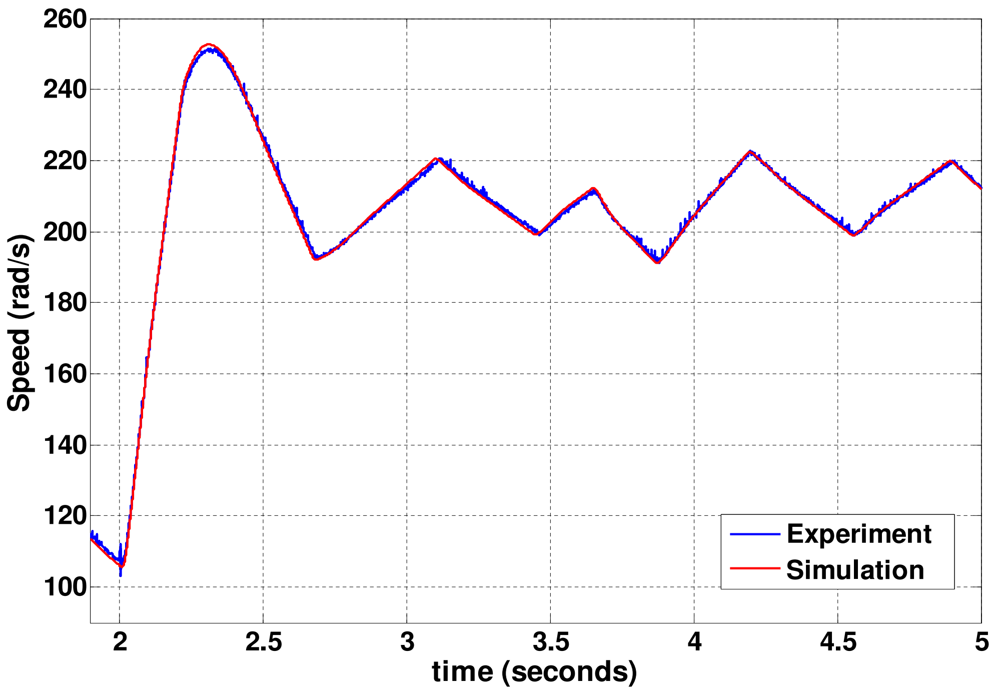


(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.5s$

Figure 5.17: Flexible shaft load emulation - Inertia of $3J_L$ and Damping of $3D$



(a) Speed response to step demand of 0 to 1000 rpm



(b) Speed response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $3.65s$

Figure 5.18: Flexible shaft load emulation - Inertia of $5J_L$ and Damping of D

speed controller of the tracking loop and this would invalidate the emulation. Hence, this saturation must be avoided. For this reason, the mechanical loads emulated in the research project have parameter values that are always greater than the total dynamics of the coupled (drive and load motor) system used in the emulation system.

The benefits of having the digital controller and the mechanical emulator system on the same platform within the experimental set-up provides certain benefits. Some of these benefits can be highlighted by considering the sampling time of the system. The sampling time used within the experimental system is the same for both the digital controllers and the mechanical emulator system. In the absence of such an integrated platform, there may have been issues in matching the different sample times for proper synchronisation of the two systems. Another benefit of the compact platform can be highlighted in the distance that exists between the digital controllers and the mechanical emulator system. As a result of the close proximity of the two systems, there was no need for signal maintenance/boosting electronic circuits such as current mirrors or configured amplifiers. The absence of same platform implementation for the two systems may not necessarily limit the effectiveness of the employed emulation strategy but it probably would have made its implementation more involving

5.8 Conclusion

This chapter has described a dynamometer control strategy for the dynamic emulation of mechanical loads. The emulation strategy has been employed within the research work for the particular emulation of the dynamics of a Stiff and Flexible-shaft mechanical load. The strategy ensures that the (open and closed loop) dynamics of the mechanical load models are preserved during the emulation. The emulation strategy is based on a speed-tracking control with implicit feed-forward of its inverse dynamics that provides compensation for the closed-loop tracking control dynamics [55], [60].

The emulation strategy implemented provides a platform for the testing of different motor drive control strategies. The emulation system requires the driving machine torque reference signal as an input variable. This is not a restriction as the aim is to provide a test-bed for motion control strategies of drives. If a torque reference signal is not available, then an electrical torque observer or estimator is required. The experimental emulation facility allows the emulation of mechanical load models with different plant-parameter values. The bandwidth of the emulation strategy is limited by the inner current control loop of the load machine and the bandwidth of the noise filter for the tachogenerator.

The described approach for emulation within the chapter incorporates the model of the dynamometer current loop in its compensation term. In the cases considered during the research project, the model of the dynamometer current control loop can be justifiably neglected; simply assuming that it has a unity gain, would produce similar results with the emulation strategy; the point being that there is not a serious requirement to model accurately the current loop especially given it has much faster dynamics than the speed loop.

In summary, the programmable load emulator system incorporates delays and accurately modelled compensation terms which cancel out the dynamics of the speed tracking loop. The cancellation ensures the mechanical load dynamics imposed by the emulator truly represents the desired load in both steady state and transient conditions. Having eliminated the speed tracking loop dynamics, the emulator imposes the desired dynamics of the digital mechanical load on the motor drive controller. By adopting this approach, it is possible to investigate the performance of different robust, adaptive and nonlinear optimisation algorithms on a true experimental system under different mechanical load conditions.

Chapter 6

Robust Experimental Control Design

6.1 Introduction

The chapter describes the *Robust Experimental Control design* method adopted to achieve the objective of an automated design of robust speed control systems for electronic drives. This method is one of the solutions that result from the application of the automated design tool, *EA*, directly onto the experimental system. The other method proposed in this work, based on a more theoretical approach, is described in more detail in chapter 7. The experimental control design method is one that enables the direct optimisation of the controller for the drive while it is subjected to *accurate dynamics of modelled operating (load) conditions.* Although it is often tempting to term this method an online optimisation process, there exists a difference between the two approaches. *Online optimisation* means that the evolutionary algorithms perform the robust design while the drive is running during normal work situation (in a factory for example) without interrupting the production chain, or whatever the drive has been placed to do. Although there exists differences between the two, similar issues are involved in setting up the test beds for the implementation of both optimisation approaches.

There have been recent developments regarding the techniques devised for the design of robust control systems online/experimentally. Some of these techniques are reported in [12], [16], [41] and [65]. The experimental control design method is implemented through direct trial-and-error experimentation on the actual system. The benefits of the approach employed over traditional model-based techniques include:

1. The high fidelity in the quality of the final controller solution given the manner with which the method is implemented. With regards to model-based approaches, the fidelity of the solution largely depends on the quality of the model. Often these models may fail to account for non-linearities, noise *etc* within the system, thus placing limitations on the quality of the final solution achieved via this approach.
2. The non-requirement for detailed analysis of the system to be optimised. Under the model-based approach, modelling the system to be controlled involves carrying a fairly detailed analysis to predict the behaviour of the system in response to different inputs. The modelling process often requires some level of expertise and it can also be incredibly time-consuming
3. The robustness in its application, *i.e.* it can be applied in a very similar manner to many different systems without the need to entirely re-develop the approach. This is a follow-on from the second point. The model-based approach demands defining a a model for each of the different system types it is applied to.

Although the underlying concept is simple, the design and development of a system, capable of automatically designing robust control systems through direct experimentation on the actual system can be difficult. The difficulty stems from a number of reasons, which include:

1. A means to prematurely stop the testing of badly performing controller solutions. The premature halting process is achieved through the incorporation of

specifically designed software logic circuits. The process serves as the safety mechanism that prevents damage to the hardware.

2. A means to avoid interference between the testing of subsequent solutions. Such a mechanism ensures that the outcome of the tests of any one solution is not influenced by previously tested solutions. The method adopted within the research project simply allocates a time period within which a controller solution is tested and evaluated. The pre-allocated testing time prevents overlapping between tests and their outcomes.

3. Often many experiments need to be performed and their outcome evaluated, given the trial-and-error nature of the optimisation process. For this reason, mechanisms that bring about convergence to an optimal solution within relatively short periods are necessary for the successful application of the method. The methods adopted to ensure convergence will be discussed.

This chapter deals with the description of an experimental process for the automated design of robust digital controllers for variable speed DC drives. To this point, there are only a few documented researches that have adopted such an automated experimental robust control system design method [12], [16]. In a similar fashion to the same general motivations of the mentioned literature, the process attempts to fully exploit the potentials of evolutionary design by letting the search algorithm explore alternative structures (controllers of different orders), and optimise the associated parameters describing a linear anti-windup controller for an industrial drive. Also during the search and optimisation procedure, the variable speed drives are subject to certain operating conditions such as the emulated mechanical load dynamics of typical industrial loads. The operating conditions are a necessary part of the optimisation process that ensure the controllers designed are robust and as such will perform efficiently under the given operating condition.

The description of the experimental procedure involves highlighting the necessary additions to the experimental system described in chapter 4 and then summarising the entire experimental process. The automated experimental optimisation process is repetitive and involves several trial-and-error events which are somewhat similar. To describe the adopted optimisation process, it would be sufficient to describe the implementation of one trial-and-error event that occurs. Finally, experimental results of the automated optimisation will be presented and analysed. Also, the simulations results will be provided for the purpose of comparison.

6.2 Experimental Optimisation System

The detailed conceptual layout of the automated experimental optimisation system can be suitably represented in the diagram of figure 6.1. The particular aim of the research focuses on developing a unique and novel optimisation approach for designing robust 'speed' controllers for the electric drives. The actual physical hardware for the experimental system is shown in figure 4.4; it consists of two identical 1.2kW permanent magnet DC servomotors coupled along the same shaft. One serves as the Driving motor and the other is the load motor with the possibility of emulating different kinds of mechanical load types. Both motors, for their power conversion stage, employ a Control Technique DCD60*10/20, DC-to-DC converter and an analogue current controller based on a MOSFET H-Bridge configuration switching at 20kHz with nominal current of 3A. Both the control strategy and the mechanical load emulation are programmed using xPC target toolbox in MATLAB/Simulink and interfaced with the motors using the National Instrument PCI-MIO-16XE-10 high resolution I/O board.

Figure 6.1 highlights the application of the evolutionary algorithms to the digital speed controller, G_C and also the incorporation of the logic protection circuit used in prematurely stopping the testing of bad controller parameters within the experimental system. Given the Evolutionary Algorithms enables automatic optimisation of the controller and the load emulator enables the accurate reproduction of the variable industrial load conditions, under which the controller is optimised, the experimental optimisation set-up adequately incorporates the necessary systems to meet one of the

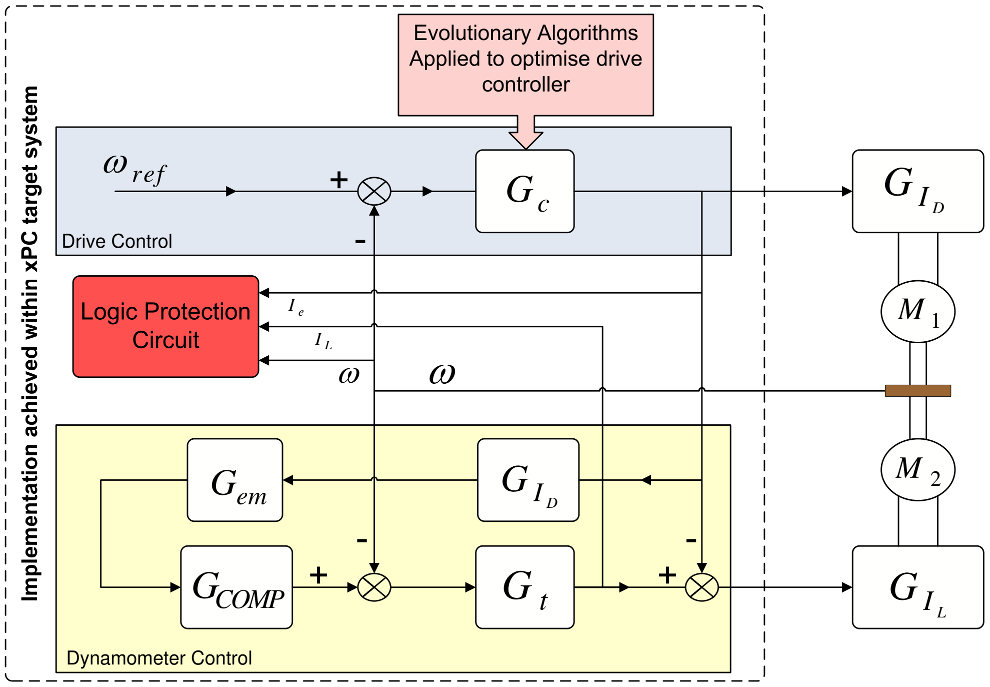


Figure 6.1: Experimental system

main aims of the projects.

The function of the logic protection scheme implemented within MATLAB/Simulink is ultimately to stop the premature testing of 'bad' controllers. These controllers produce responses that do not meet the specification of the desired output response. It must be emphasised that the logic circuit does not stop the testing of all bad controllers but only those that quite significantly detract from the desired response or produce a response that could potentially be damaging to the physical hardware. The implementation of the logic protection scheme can be can be summarised in the following manner:

1. The logic protection scheme uses three signals of the experimental rig as its inputs: the speed signal, ω, produced by the tachogenerator, the current demand signal, I_e produced by the digital speed controller and the current demand signal produced by the speed-tracking controller, I_L.

2. The logic circuit is simply a combination of logic conditions, defined around

these three inputs, which predict bad controller responses.

3. The circuit prematurely stops the further testing of controllers that meet the criteria for obviously bad controller responses and permits only the testing of those controllers that do not trigger the preset conditions

Apart from the benefits of damage prevention to the physical hardware and reduction in the stress and fatigue suffered by the system during experimentation, thus prolonging its lifespan, the incorporation of the logic protection scheme brings about a peripheral benefit of time reduction for the optimisation process achieved through the incorporation of the safety mechanism. The schematic that represents the functioning of the protection scheme is shown in figure 6.2 and its actual implementation within Simulink is in Appendix A.4.

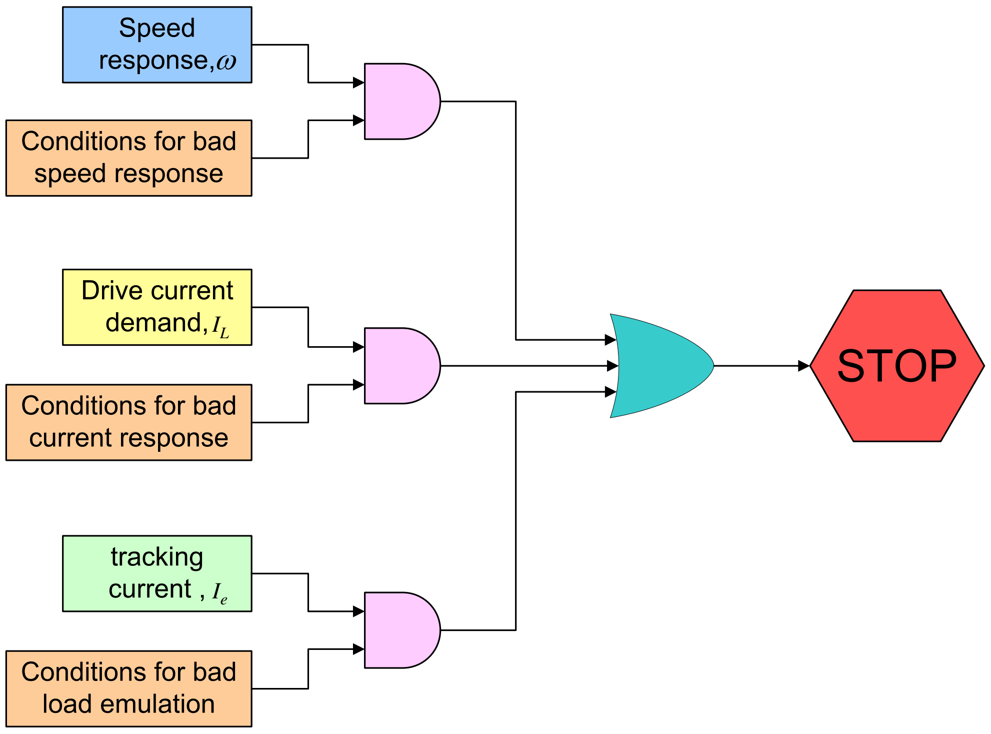

Speed response, ω
Conditions for bad speed response
Drive current demand, I_L
Conditions for bad current response
tracking current , I_e
Conditions for bad load emulation
STOP

Figure 6.2: Logic Protection Scheme

For every controller parameter directly tested on the experimental system, regarding the functioning of the logic protection circuit in relation to speed responses, the resulting response of the mechanical load, ω is compared against a set of conditions that define the bad speed responses. These conditions check that:

1. the speed response follows a zero speed demand accurately and within error limits defined by the output of the tachogenerator.
2. the speed response does not exceed certain specified limits for minimum and maximum speed.

In a similar fashion, the current demand, I_e produced by the drive controller, G_c is compared against similar sets of conditions, which check that:

1. when a zero-speed demand is requested, an approximate zero-average current is requested by the controller. The amplitudes of the voltage signal that represent this zero-average current must be within set limits defined by the error of the analogue-to-digital converter
2. the speed response, ω has a transient response that lasts for a specified time period. This time period is specified using the response obtained when a PI controller is employed for the same system.
3. the current demand by the controller is within the nominal value when the transient response of the system has subsided.

The final input to the logic protection circuit is the current demand, I_L of the speed-tracking controller, G_t. This is essentially used to ensure that:

1. the emulation strategy is not invalidated by any controller randomly selected by the EA during the optimisation process.
2. if I_L reaches saturation at any point during testing, the evaluation process is stopped given this implies such a controller invalidates the emulation process.

If any one of the inputs to the logic protection circuit, derived from the controllers under test, meet their respective criteria for bad responses, further testing is stopped and the evolutionary algorithm moves along to test the next solution.

6.3 Automated Experimental Optimisation

The experimental optimisation process implemented provides the means to automatically tune simultaneously both the structure and parameters of the digital controllers while it is being subject to varying mechanical loads. The benefits of the approach is there is no need to have any knowledge of the model of the mechanical load for which the robust controllers are to be designed because the optimisation is implemented directly on the system that includes the mechanical load. The unique feature of this approach lies in its evaluation stage, which is critical for directly designing robust control systems.

The general structure for the implementation of the different algorithms are described in chapter 3. Each of the different algorithms adopt a similar evaluation stage. The evaluation stage deserves particular mention because the manner by which it is implemented in this chapter, defines the experimental method for robust control system design. The purpose of the evaluation stage is to quantify the quality of each randomly selected controller. These values are used by the evolutionary algorithms to determine which controller gives the best performance. The evaluation stage tests that each randomly selected controller performs well to the selected combinations of parameter variations.

The objective function is adopted by the GA to for the purpose of evaluation while the BF and HBF adopts the nutrient concentration function for evaluation of individual solutions. The evaluation process is similar for the different algorithms and it is implemented in three stages: assigning of controller its structure and parameters, testing the defined controller and evaluating controller performance

6.3.1 Assignment of Controller Structure and Parameters

Although this phase of the overall evaluation process takes the least amount of time, its successful implementation is crucial to the accuracy of the results from the evaluation stage. If controller parameters are wrongly assigned, the the optimisation would be driven in the wrong directions from the very beginning, resulting in bad control solutions. Each individual within the population of solutions encodes data that correspond to the structure and parameter of the digital controller under consideration. For the evaluation process to begin, the controller must be assigned with its structure and parameters by the values encoded by the individual under evaluation. The assignment is done via the MATLAB m-file code. Within the coding sequence, approximately $25ms$ is set aside for the successful implementation of the controller definition stage. The controller definition process largely depends on the speed of the computer's processor; this implies that for more powerful processors, the process can be implemented in a shorter time. The newly defined controller is then downloaded onto the Simulink model for the digital controller.

6.3.2 Testing the Defined Controller

The digital controller, whose controller structure and parameter have been assigned, is tested experimentally, using the setup in figure 6.1. The controller should ideally be subjected to all the possible combinations of parameter variations for the stiff and flexible shaft mechanical load. There are an infinite number of possible combinations; hence, the controller is tested separately on nine combinations of parameter variations for the these mechanical loads. These combinations, in each case, are suitable to cover the range of parameter variations for these loads.

During testing, each controller is subjected to the same operating conditions that include speed transients and load disturbance, as well as variations of the mechanical load parameters. The profile of the speed transients and load disturbances can be simply categorised in four phases: During *Phase one*, between the time period of 0 *ms* to 100 *ms*, a speed of 0 *rad/s* is requested. This period is useful in immediately weaning out obviously bad controllers from further testing; some bad controllers might tend to produce accelerated responses when there is, in fact, no speed demanded. *Phase two* lasts between the periods of 100 *ms* and 600 *ms*. During this period, the experimental system is expected to rotate at approximately 105 *rad/s*. A speed demand of 210 *rad/s* is requested in *Phase three*, between the periods of 600 *ms* and 1050 *ms*. Finally, in *Phase four*, coupled with the speed demand requested in Phase three, a torque disturbance, approximately 0.15 *Nm*, is also imposed on the output of the controller, between the periods of 1050 *ms* and 1500 *ms*. The real-time simulation is performed within a time period of 1700 *ms*. This stage of the evaluation process takes the longest and is implemented in real-time using the xPC target system; it has no dependence on the power of the computer's processor. The data obtained from the real-time simulation phase are stored and used in the next stage, which involves quantifying the quality of the performance of the different controllers.

6.3.3 Quantifying the Quality of Controllers' Performance

The speed response produced by the tested controller is used to obtain the numeric value that represents its performance. The performance value is achieved using the Integration of the Absolute Error (IAE) (section 3.2.2) between the desired speed and the actual speed response produced by the controller. Also incorporated within this stage is a penalty factor on speed responses that have an overshoot of more than 8%. Although this is not explicitly shown within the code, the conditional statements, and the signals they act upon evaluate whether or not the controllers' speed responses meet certain criteria. If they fail to meet the criteria, they penalised by the addition of a really large value (in this case '5000') to their objective value. Quantifying the controllers performance is implemented in a window of approximately 25 *microseconds* during the programme run; this period allocated also depends on the power of the processor being harnessed. Defining the periods of the different stages is necessary to ensure that the testing of a previous controller solution does not affect the evaluation of current solution.

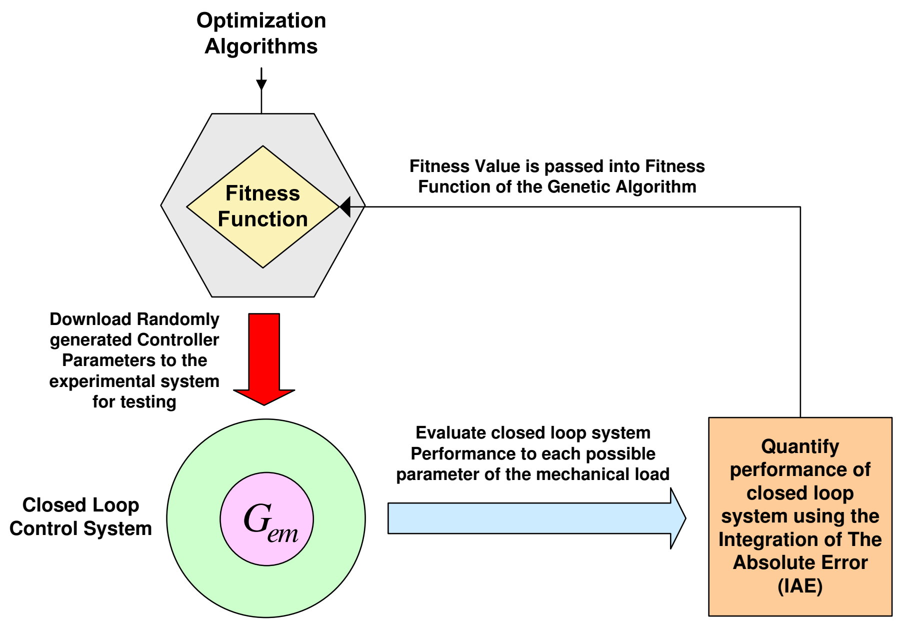

Optimization Algorithms
Fitness Function
Fitness Value is passed into Fitness Function of the Genetic Algorithm
Download Randomly generated Controller Parameters to the experimental system for testing
Closed Loop Control System
G_{em}
Evaluate closed loop system Performance to each possible parameter of the mechanical load
Quantify performance of closed loop system using the Integration of The Absolute Error (IAE)

Figure 6.3: Evaluation and Fitness Assignment During Optimisation process

It is also necessary to incorporate protection circuits that prematurely stop the testing of badly performing individuals. The effects of such protection schemes are threefold; first, it reduces the overall time for the experimental design of robust digital controller by prematurely halting further testing of badly performing individuals as soon as they are detected; second, it reduces the stress and fatigue experienced by the machine during the experiments and third, it helps in preventing damage to the machine. Such protection are also useful in the simulation particularly for the purpose of reducing time for the overall optimisation process. Figure 6.3 summarises the evaluation stage for the experimental process for the design of robust digital controllers.

6.4 Results

The results obtained by employing the experimental robust control design method highlight the effectiveness of the proposed approach. The similarities between the results obtained by the different evolutionary algorithms used in combination with the proposed method show that the designed algorithms are effective in their search procedures. These similarities also provide more confidence in the truly global nature of the obtained solutions; this suggests that the corresponding speed responses they provide are the most efficient under the considered operating conditions. In the following sections, the general speed profile of the results obtained will be explained and the results of the controlled speed responses of the stiff and flexible shaft loads in the presence of speed transients and load torque disturbance will be characterised.

6.4.1 Stiff Shaft Load Response

By employing the experimental control design, robust control systems for the stiff shaft load have been designed using each of the three evolutionary algorithms. In the subsequent paragraphs, a comparison between the speed responses produced by the separately designed robust control systems will be provided. At this juncture, it will be useful to explain the general profile of the speed response observed across the results provided.

The general profile of the speed responses obtained from the designed robust control system can be explained in the following manner: The torque produced by the controller, which is derived from the difference between the demand speed and the actual speed of the mechanical load, forces the load to move until it reaches the desired speed. On reaching the desired speed, there is no more torque produced, as there is no more speed error. The result is the load moves with a constant velocity equal to the desired speed. The resulting mechanical load speed response produced can be reasonably characterised according to its rise time, settling time, overshoot, disturbance rejection time and steady state output.

Rise time: Observed across the speed responses is the variations in the rise times as a result changing the inertia of the mechanical loads. The responses of the GA designed fourth order controller in figures 6.6, 6.7 and 6.8 have respective rise times of $71ms$, $73ms$ and $85ms$. The speed responses of the fourth order controller designed by the BF is show in figures 6.12, 6.13 and 6.14; they have respective rise times of

74.2ms, 72.2ms and 85.2ms. Similar responses have been obtained using the HBF. The responses of the fourth order controller it designed are shown in figures 6.18, 6.19 and 6.20 and these have respective rise times of 80.2ms, 71ms and 83.8ms.

Settling time: For the observed responses in figures figures 6.6, 6.7 and 6.8 achieved by employing the GA, the respective settling times are 130ms, 100ms and 135ms. The controller designed by the BF has produced speed responses in figures 6.12, 6.13 and 6.14 with settling times of 114ms, 116ms and 200ms respectively.The HBF algorithms has produced similar speed responses in figures 6.18, 6.19 and 6.20 with the respective settling times of 159ms, 93ms and 171ms.

Overshoot: The GA fourth order controller produces a maximum overshoot of 1.23%, which is found in its response in figure 6.8. The BF controlled response in figure 6.14 produces a maximum overshoot of 3.8%, which is slightly higher than that of the GA. The HBF response has produces a maximum overshoot of similar magnitude as its counterparts of 1.4% in figure 6.20

Disturbance Rejection: All the responses shown for each of the different evolutionary algorithms reject disturbance quickly. The GA controller speed response reject disturbance in less than 180ms, while the BF achieves total rejection of disturbance in less than 150. The HBF is the fastest with disturbance rejection time of less than 110ms.

Steady State: The effects of the total friction is observed in the steady state motion of the stiff-shaft load. The larger the friction in the system, the greater the torque demand required to achieve the demand speed because of the extra torque needed to overcome friction. A natural outcome of doing work against friction is noise (and heat) produced. The noise produced as a result of the greater friction is observed in the steady state as slight oscillations around the steady state speed of the control systems. The larger the load friction, the larger the observed steady state noise oscillations. For the GA controller responses in figures 6.6, which has the largest value of friction, the steady state response is its noisiest with maximum oscillation around the steady state value of 3%. In a similar manner, the BF optimisation algorithm produces its noisiest signal in figure 6.12, with maximum steady-state oscillations of 1.1%. The HBF control system response also follow the same trends producing its noisiest response in figure 6.18 for the system with the largest friction value; its maximum steady-state oscillation is 1.5%.

6.4.2 Flexible Shaft Load Response

Although the speed responses obtained with the flexible-shaft mechanical load, is largely due to the same reasons as those for the stiff-shaft load closed loop system, the effects of the spring and backlash within the system demands more reasons to clarify the observed speed responses. The aim of the designed control systems is ultimately to keep the speed of the load motor moving at a desired value. Like the stiff-shaft load, the controller produces a torque derived from the error between the desired speed and the actual speed of the load that acts on the mechanical load to force it to rotate according to the demand speed.

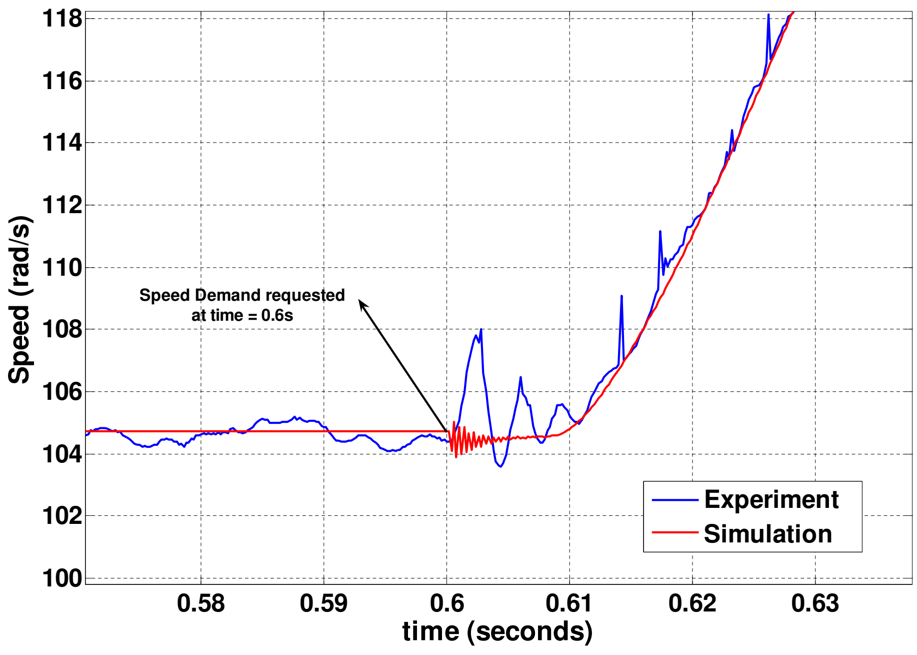


Figure 6.4: Time-lag oscillations due to Backlash

Due to the presence of the spring and backlash, there exists a slight complication. The issue is that the controller cannot directly influence the speed response of the load. Indirect control is achieved by first causing the motion of the drive motor and then the effect is transmitted to the load motor via the flexible shaft that connects the two motors. As a result of backlash, there exists a time lag that exist between the application of the torque on the drive motor and the load motor responding accordingly. Figure 6.4 confirms the existence of the time lag both in the experimental and its corresponding simulation results. During the lag phase, some speed oscillations are observed on the speed response. These oscillations are due to the imperfect coupling between the two permanent magnet DC motors employed within the experimental system. Although these oscillations are captured within the simulation model to an extent, they could not be perfectly modelled.

This time-lag period is largely dependent on the magnitude of the backlash present. The damping, D_{em} and stiffness, K_{em} of the spring generally affect the speed oscillations that can be experienced as torque is transmitted across it; the larger these values, the more they represent a stiff-shaft and the less likely there would be oscillations observed in the spring, especially during motion in steady state. The effect (of larger damping values) can be observed in the speed responses in figures 6.9, 6.10, 6.15, 6.16, 6.21 and 6.22

In order to explain the steady-state oscillations observed in 6.11, 6.17 and 6.23 after the external disturbance (load) torque is applied, figure 6.5 will be considered. It is worth mentioning that a possible reason for the slight discrepancy observed between the experimental and simulation results in these figures could be due to the fact that errors from the model derived at a lower frequency have become accentuated due to the higher operating bandwidth of the automatically designed control system. These modelling errors may have stemmed from the imperfect coupling between the motors, the simple model for the current loop assumed or the trial-and-error approach employed in tuning the DC motor parameters. The same explanations can be applied to the similar results obtained in chapter 7. The variation of the torque input, T_e, drive motor speed, ω_m, load motor speed, ω_L and the speed difference, ω_m - ω_L,

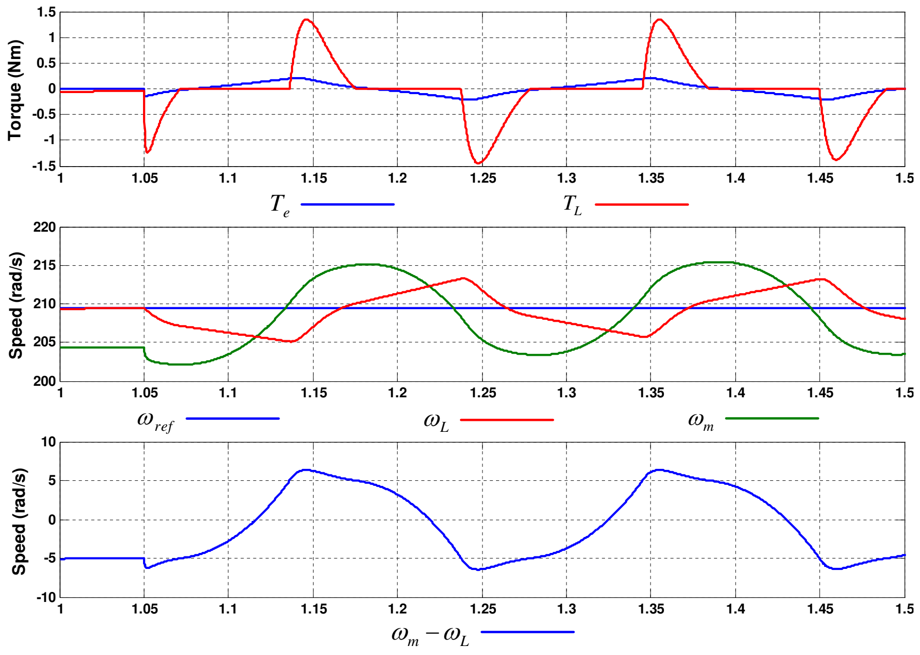


Figure 6.5: Close-up of control system variables

which serves as the input to the spring that causes backlash are shown in figure 6.5. The observed motion can be considered in two phases - 'before' the load disturbance application and 'after' the application of load disturbance.

Before the application of the load disturbance, which can be observed between the period of $1s$ and $1.05s$ in figure 6.5, the system is in steady-state with the load motor speed, ω_L, following the desired reference. The drive motor speed, ω_L, is also in steady state and the difference between the load and drive motor speed is just under $5rad/s$. The backlash caused by the spring within the flexible shaft load acts just like a switch: (1) it is turned 'on' when its input, ω_m - ω_L, exceeds $5rad/s$ which causes the spring to produce a load torque, T_L that acts on the load motor (2) it is turned 'off' when its input goes below $5rad/s$ and the result is there is no load torque produced by the spring. In summary, the backlash acts as a switch only allowing transmission of torque through the spring once the backlash limits have been exceeded. In steady state, the input to the spring, ω_m - ω_L, lie within its backlash limits; as a result, there is no load torque and the load motor maintains its steady-state speed.

After the load disturbance is applied in figure 6.5 at $1.05s$, the drive torque input, T_e, suddenly decreases causing the drive motor speed, ω_m, to decrease in a similar manner. As a result of the sudden decrease in the drive motor speed, the input to the spring, ω_m - ω_L, becomes greater than $5rad/s$ and the spring backlash is turned 'on'. A few milliseconds after $1.05s$, the transmission of load torque, T_L to the load motor begins suddenly, decreasing the load motor speed, ω_L. The designed control system senses the deviation of the load motor speed, ω_L, from the desired speed reference value, ω_{ref} and it attempts to correct the error by increasing the torque input, T_e to the flexible shaft mechanical load. The increased torque input, T_e, initially causes a decreased rate of reduction of ω_m up until $1.07s$; thereafter, ω_m begins to increase. The overall effect of the change in T_e between $1.05s$ and $1.07s$ causes the input to the spring, ω_m - ω_L, to eventually fall back to within the backlash limits and reduces T_L to zero by $1.07s$.

Although there is no more torque transmission to the load motor, the load motor speed continues to decrease between the period of $1.07s$ and $1.13s$ as a result of the damping present within the spring of the flexible shaft load. Also, given the control system still senses an error between the speed reference and the load motor speed, there is the continued increase in the torque input, T_e, to the flexible shaft mechanical plant. The result is a continued rise in the drive motor speed, ω_m. As the load motor, ω_L, decreases and the drive motor speed, ω_m, increases, the input to the spring, ω_m - ω_L, also increases. When this difference becomes greater than $5rad/s$ at about $1.13s$, the backlash is switched on and the load torque, T_L, generated between $1.13s$ and $1.17s$ causes a decrease in the spring's input, ω_m - ω_L by increasing the load motor speed, ω_L and restoring it back to the desired speed, ω_{ref}. On reaching the desired speed, given there is no direct control on the load motor speed, rather than staying at the desired speed, it keeps on increasing at a rate governed by the damping effect in the spring. The increase in load motor speed continues until the backlash limits are exceeded; it is only once they are exceeded that the required load torque can be produced to force the load motor speed back to the desired speed reference, hence the oscillatory motions observed on the waveform.

The flexible shaft load speed responses will be characterised according to its rise time, settling time and steady state output ripple. The speed transients produced are in response to step inputs of 0 - 1000 rpm between times of $0.1s$ and $0.6s$ and step inputs from 1000 - 2000 rpm between time periods of $0.6s$ and $1.5s$. An external load disturbance of $0.15Nm$ is applied after $1.05s$ to the control system.

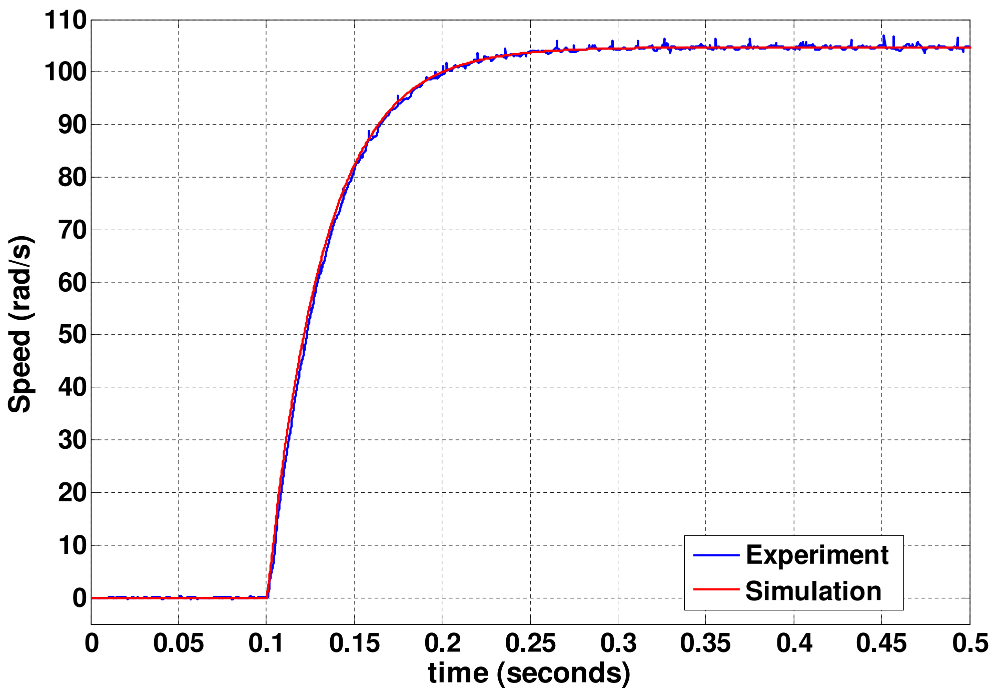
Rise time: The results of the GA optimisation has produced the following responses in figures 6.9, 6.10 and 6.11. with rise times of $120.4ms$, $102.4ms$ and $98ms$ respectively. The speed responses of the BF optimised control system in figures 6.15, 6.16 and 6.17 have the rise times of $105.2ms$, $95.6ms$ and $96.2ms$ respectively. Also the rise times of the speed responses for the HBF optimised control system in figures 6.21, 6.22 and 6.23 are $120.2ms$, $103.8ms$ and $100ms$ respectively.

Settling time: The settling times achieved by the speed responses of the GA optimised control system in figures 6.9, 6.10 and 6.11 are $200ms$, $190ms$ and $190ms$. For the BF optimised speed responses in figures 6.15, 6.16 and 6.17, the settling times achieved are $180ms$, $165ms$ and $155ms$. The settling times of the speed responses in figures 6.21, 6.22 and 6.23 obtained using the HBF optimised controller are $200ms$, $180ms$ and $133ms$.

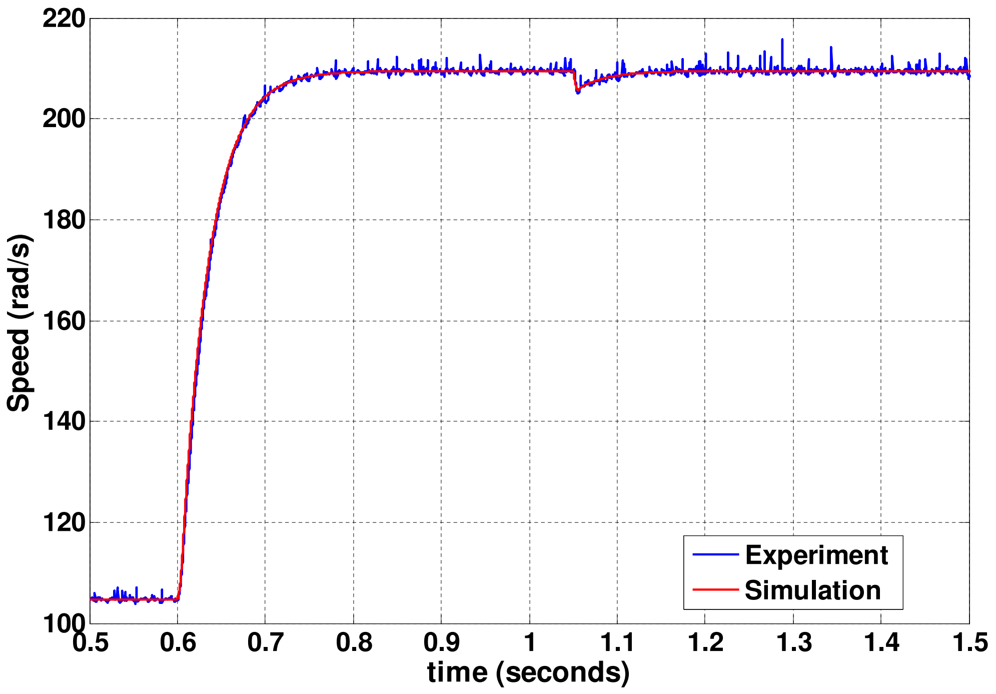
Steady-state output: On application of the torque disturbance, for the speed responses to the mechanical load with inertia of $5J_L$ and damping of D, the resulting torque oscillations have slightly different magnitudes . The GA control system produce torque oscillations with a magnitude of 6 rad/s around the final speed. The BF optimised system has torque oscillations of magnitude 5.65 rad/s. Finally, the speed response of the HBF control system produces these torque oscillations with a magnitude of 5 rad/s.

6.5 Conclusion

This chapter has described the robust experimental robust control method for the automated design of robust digital control systems. The proposed approach has been

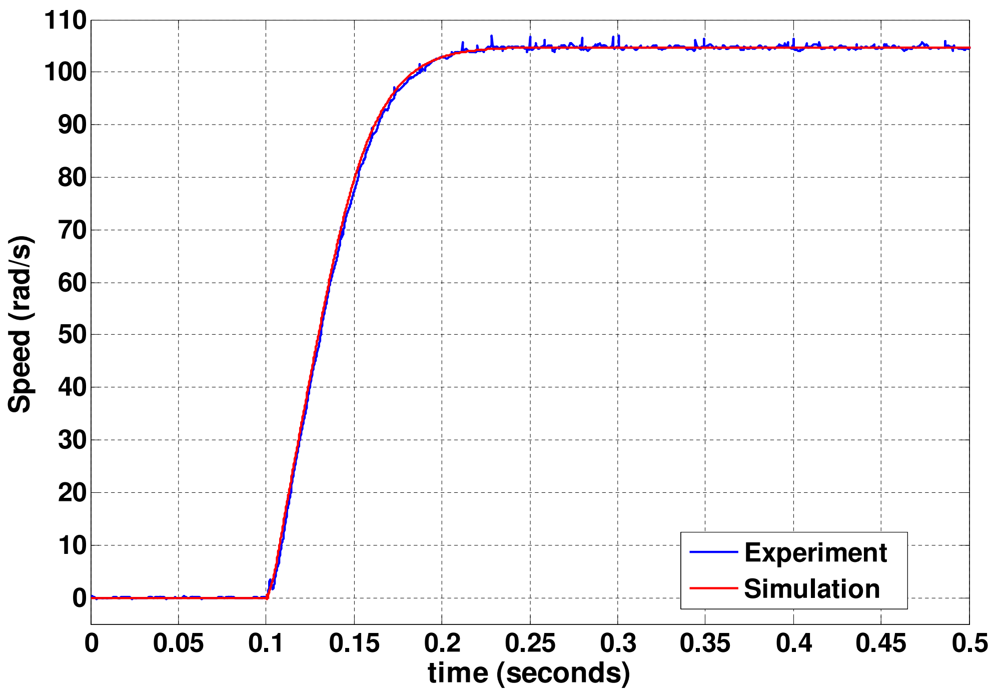


(a) Controller response to step demand of 0 to 1000 rpm

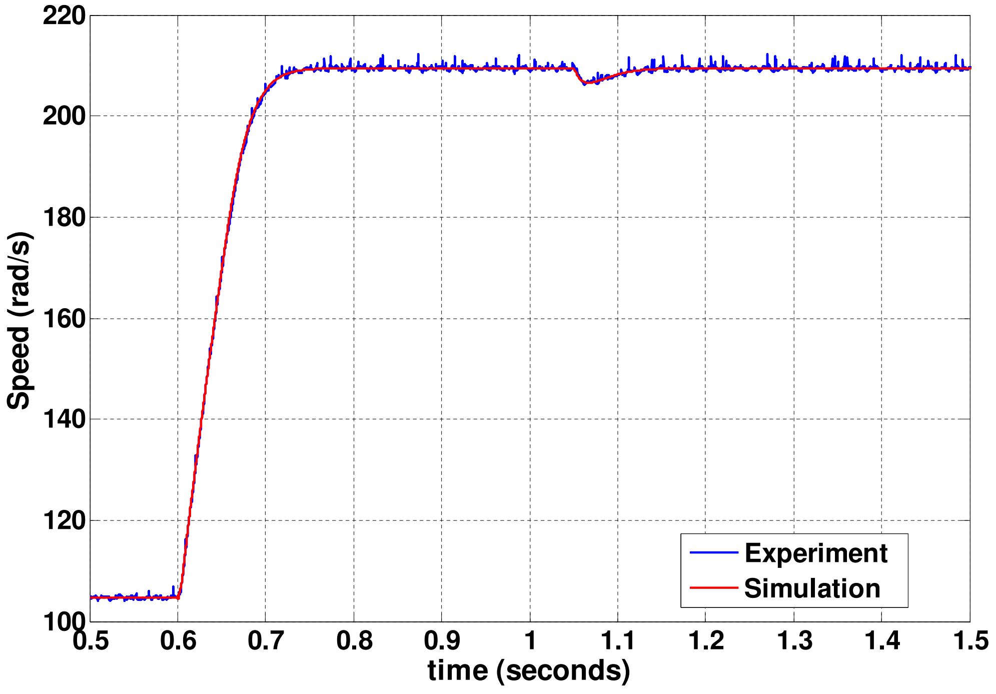


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance applied of $0.15Nm$ at $1.05s$

Figure 6.6: GA Fourth order Controller Response for Inertia J and Friction $5B$

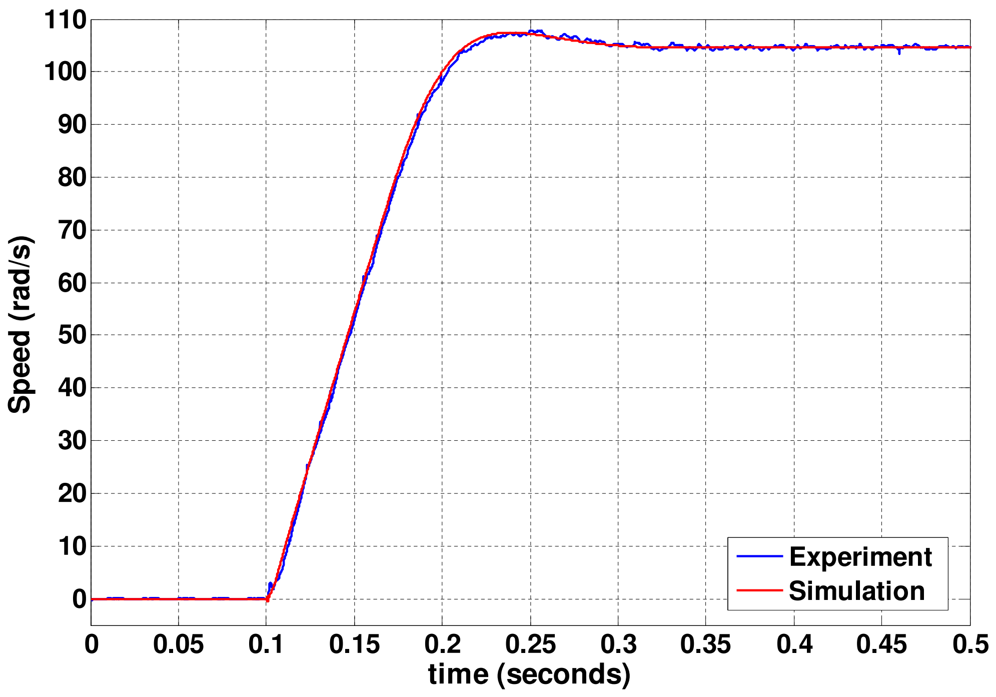


(a) Controller response to step demand of 0 to 1000 rpm

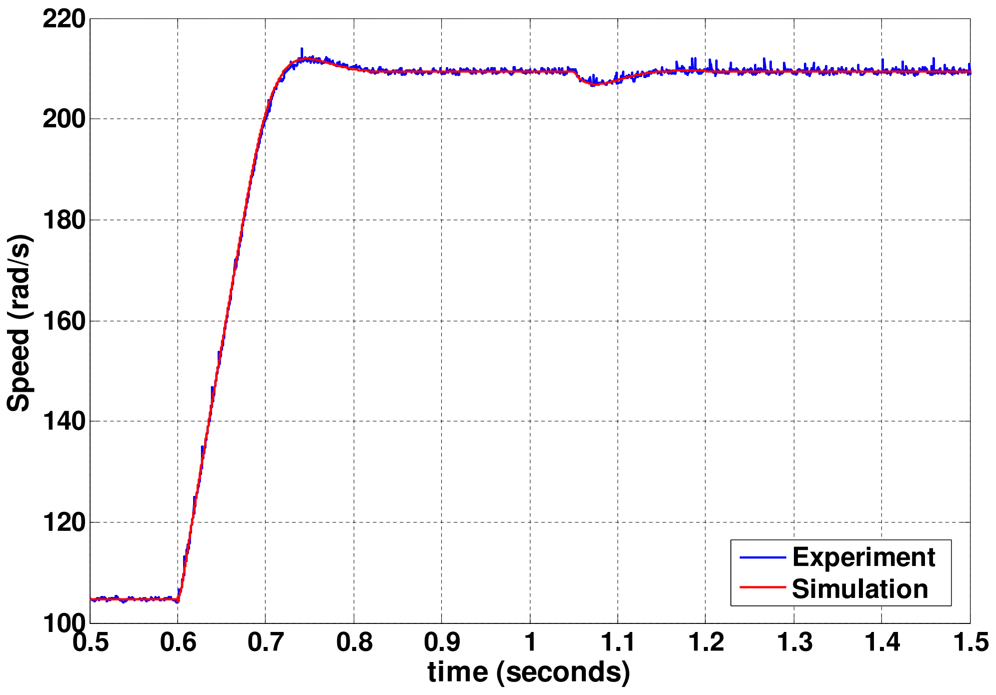


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.7: GA Fourth order Controller Response for Inertia $3J$ and Friction $3B$

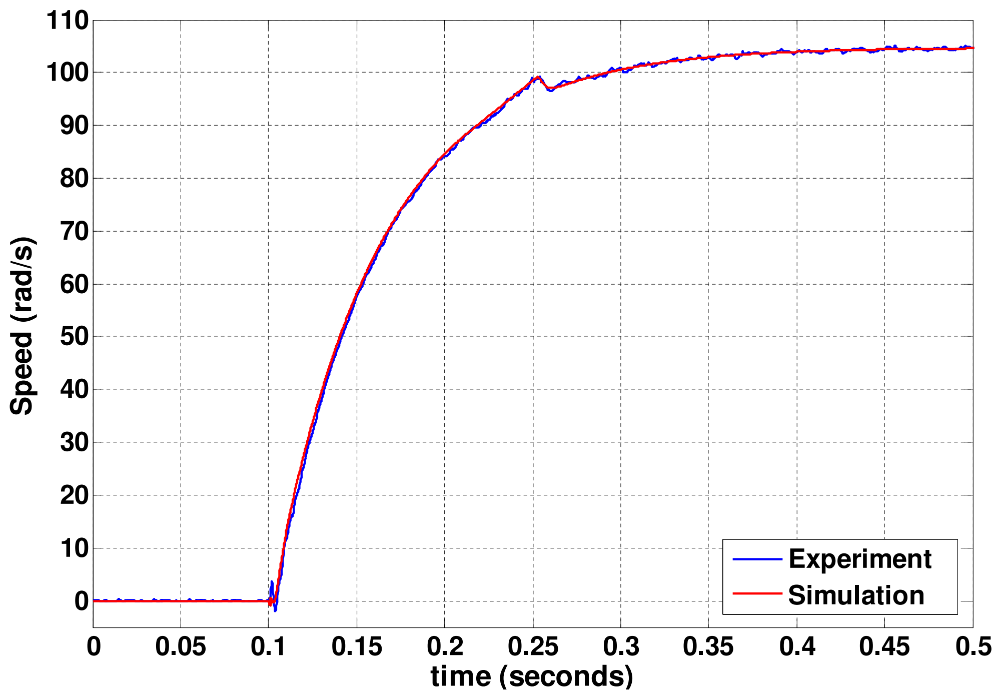


(a) Controller response to step demand of 0 to 1000 rpm

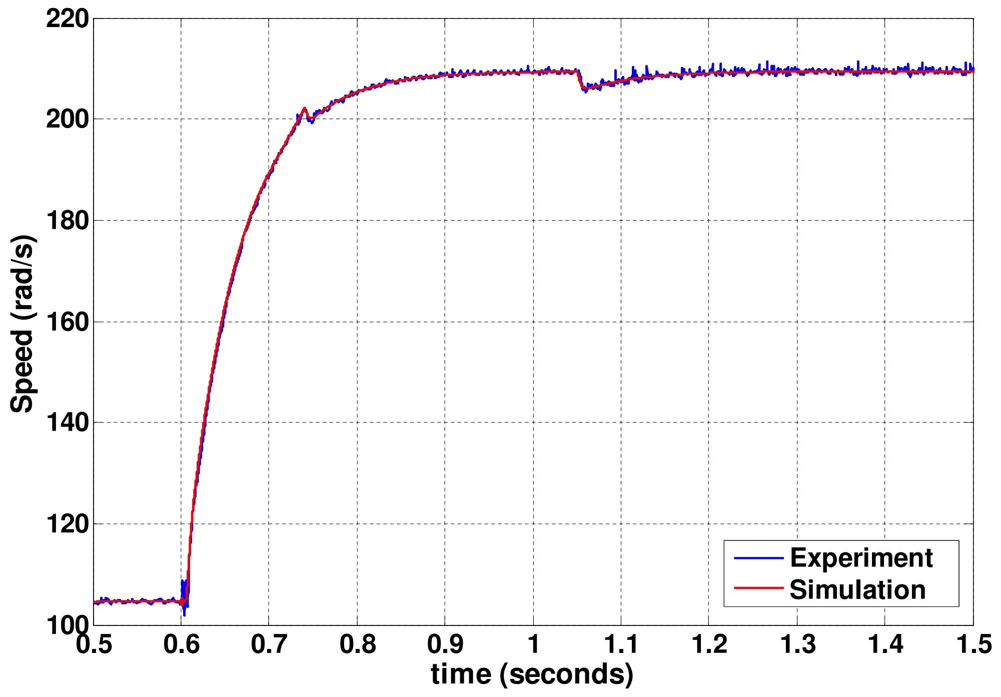


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.8: GA Fourth order Controller Response for Inertia $5J$ and Friction B

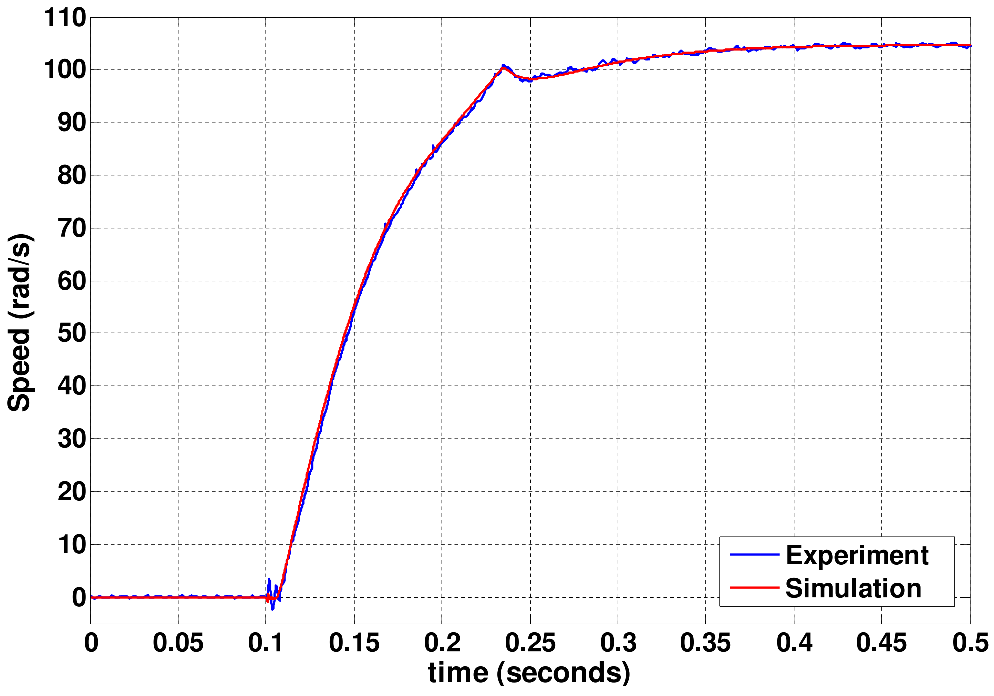


(a) Controller response to step demand of 0 to 1000 rpm

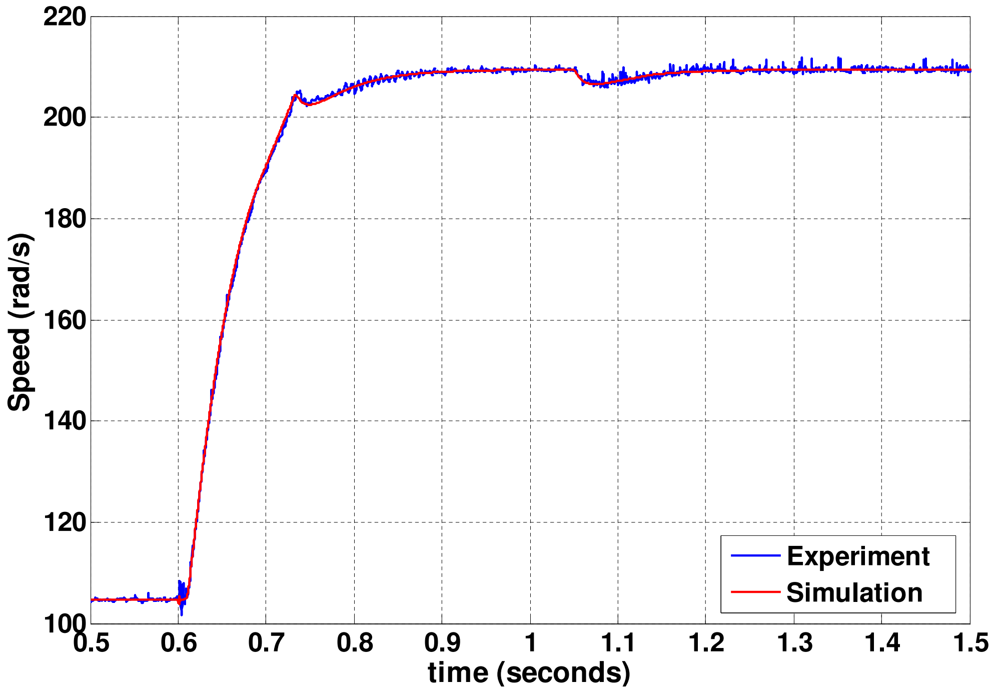


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.9: GA Fourth order Controller Response for Inertia J_L and Damping $5D$

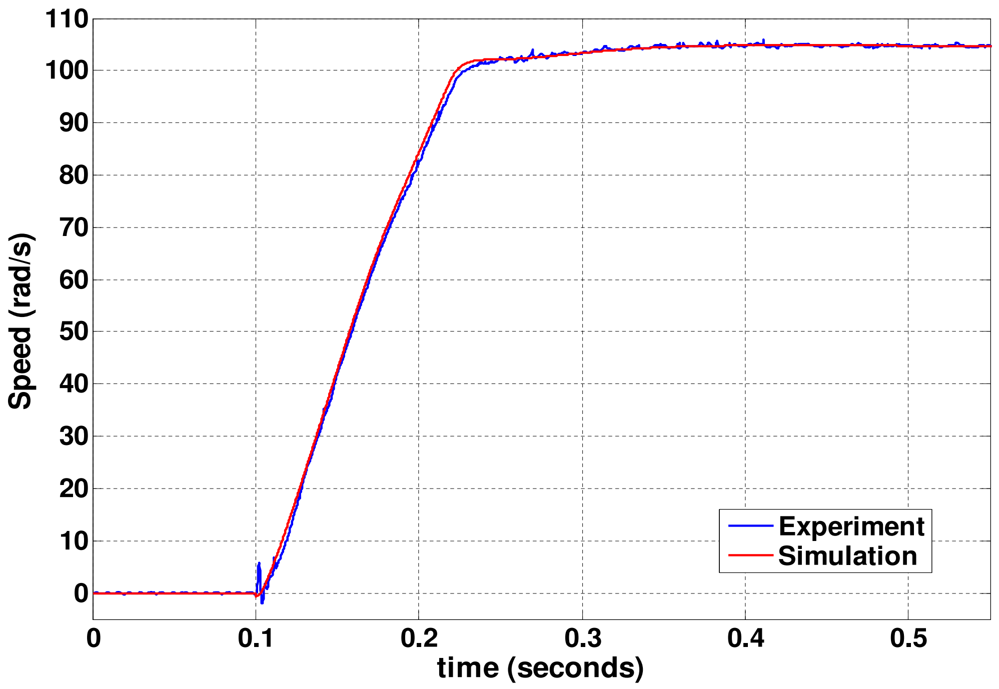


(a) Controller response to step demand of 0 to 1000 *rpm*

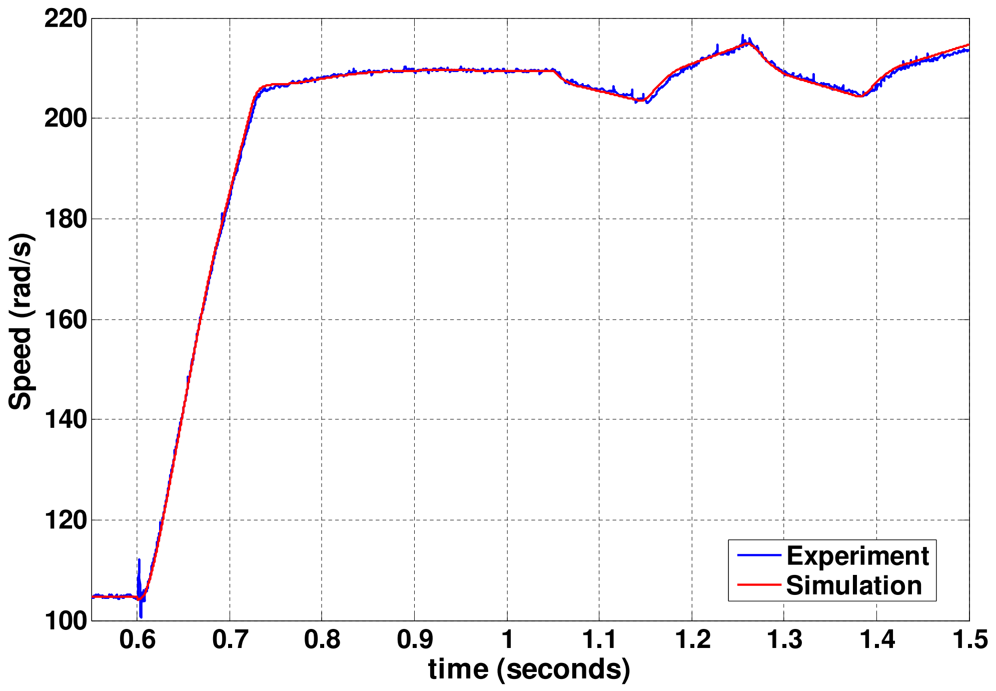


(b) Controller response to step demand of 1000 to 2000 *rpm* with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.10: *GA* Fourth order Controller Response for Inertia $3J_L$ and Damping $3D$

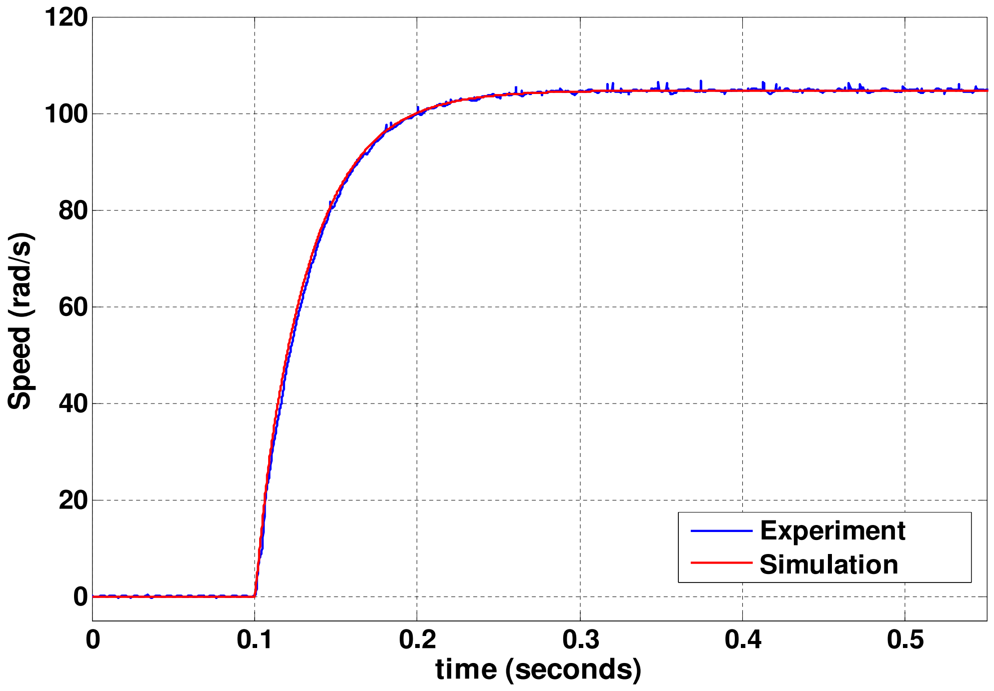


(a) Controller response to step demand of 0 to 1000 rpm

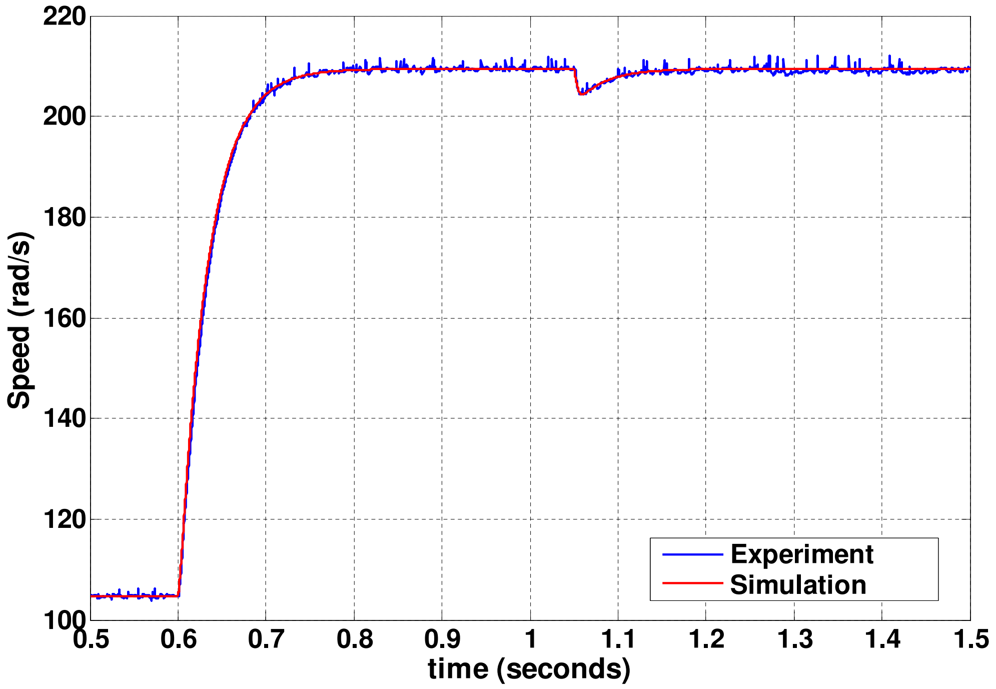


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.11: GA Fourth order Controller Response for Inertia $5J_L$ and Damping D

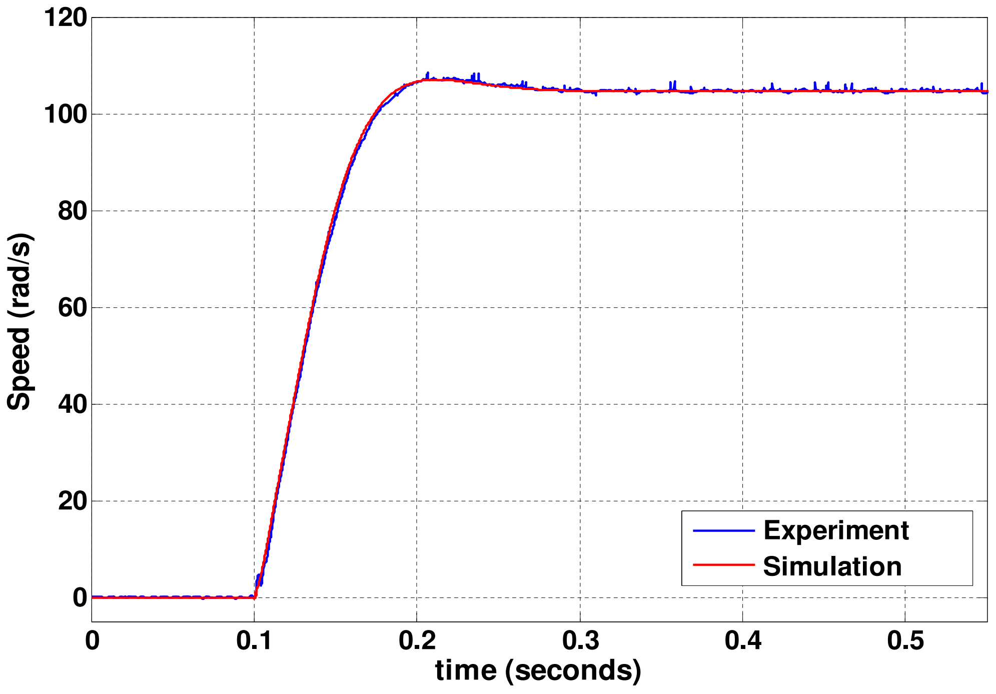


(a) Controller response to step demand of 0 to 1000 rpm

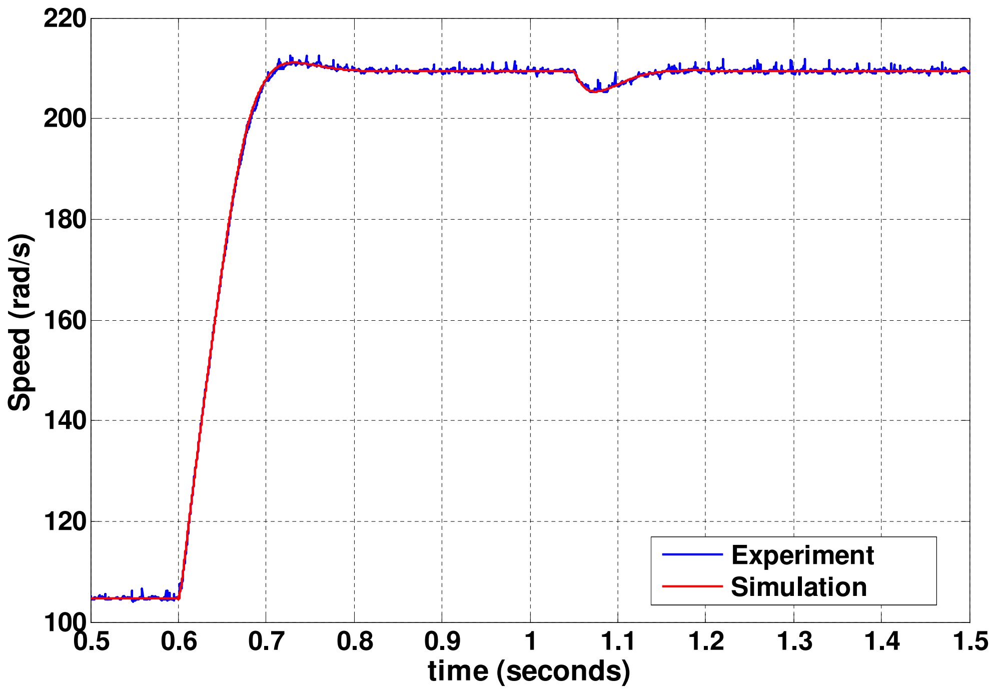


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.12: BF fourth order Controller Response for Inertia J and Friction $5B$

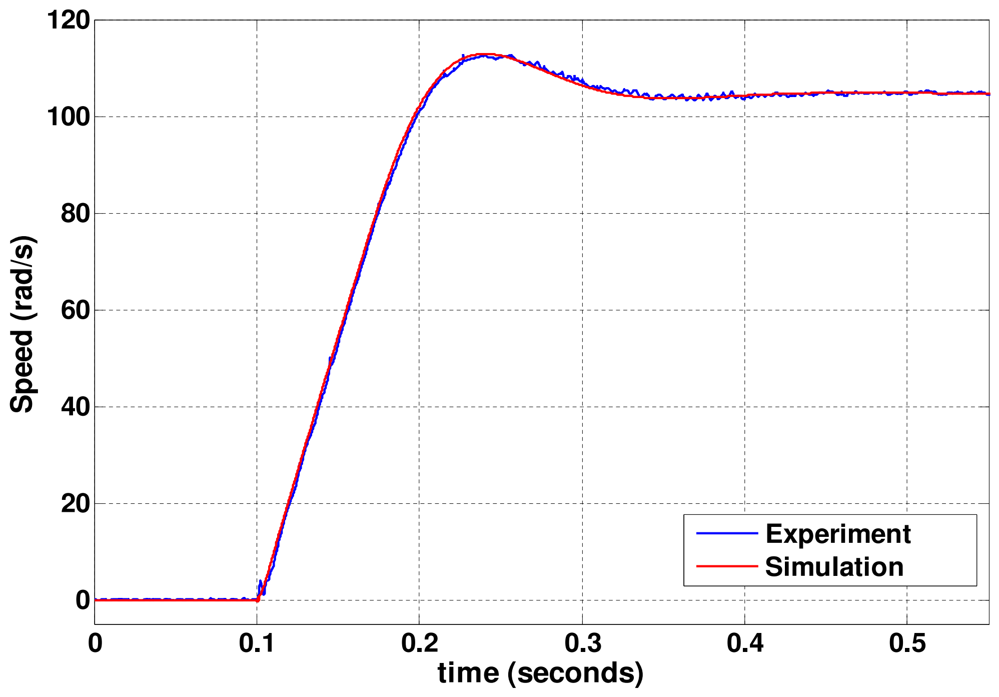


(a) Controller response to step demand of 0 to 1000 *rpm*

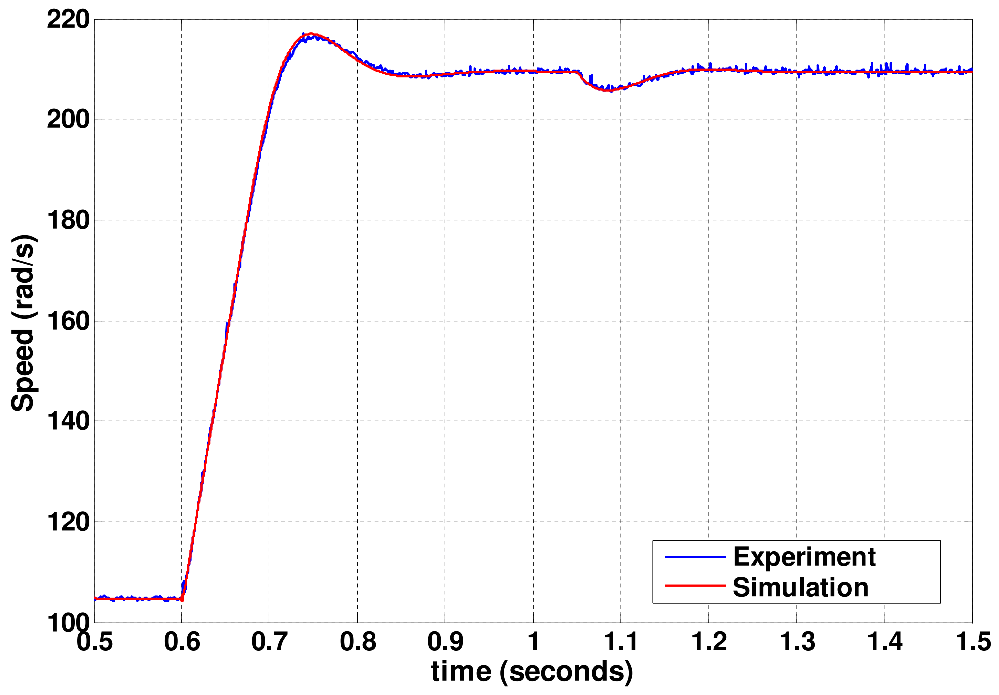


(b) Controller response to step demand of 1000 to 2000 *rpm* with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.13: BF fourth order Controller Response for Inertia $3J$ and Friction $3B$

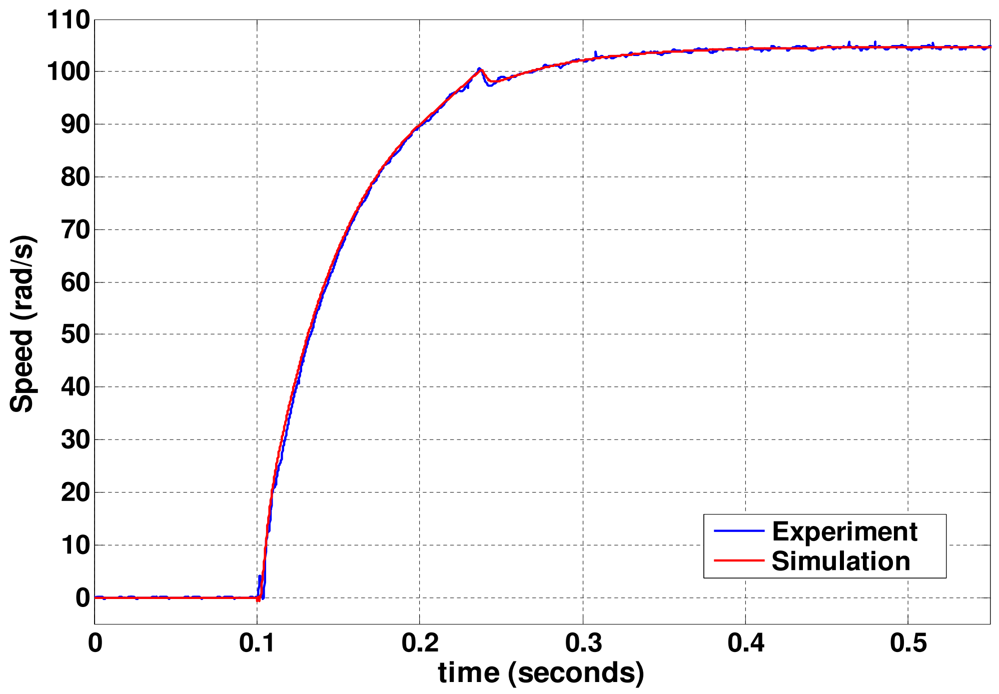


(a) Controller response to step demand of 0 to 1000 rpm

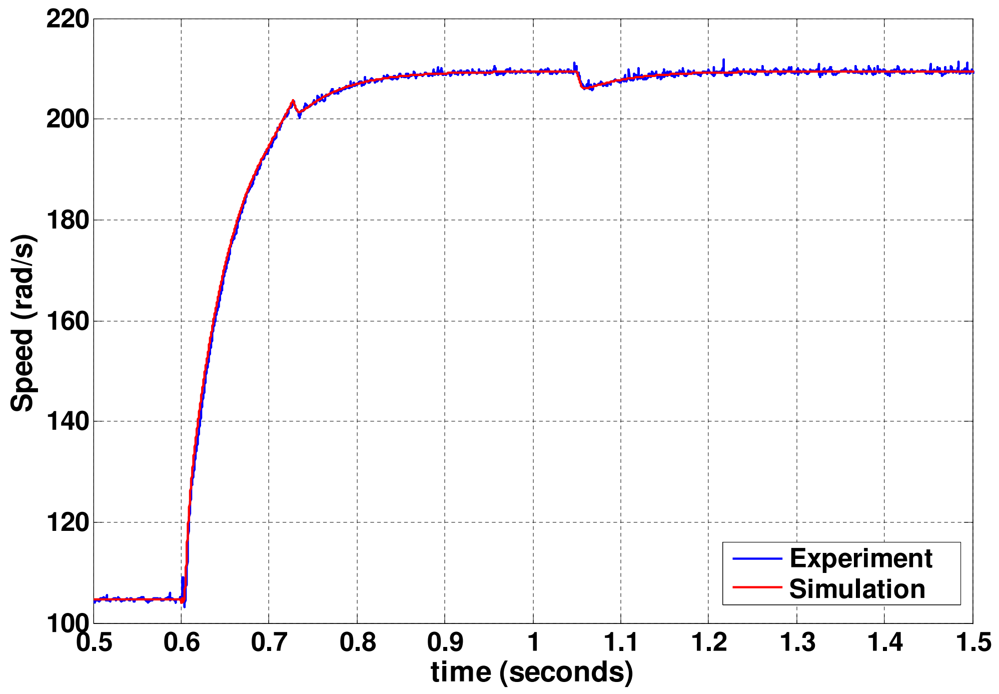


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.14: BF fourth order Controller Response for Inertia $5J$ and Friction B

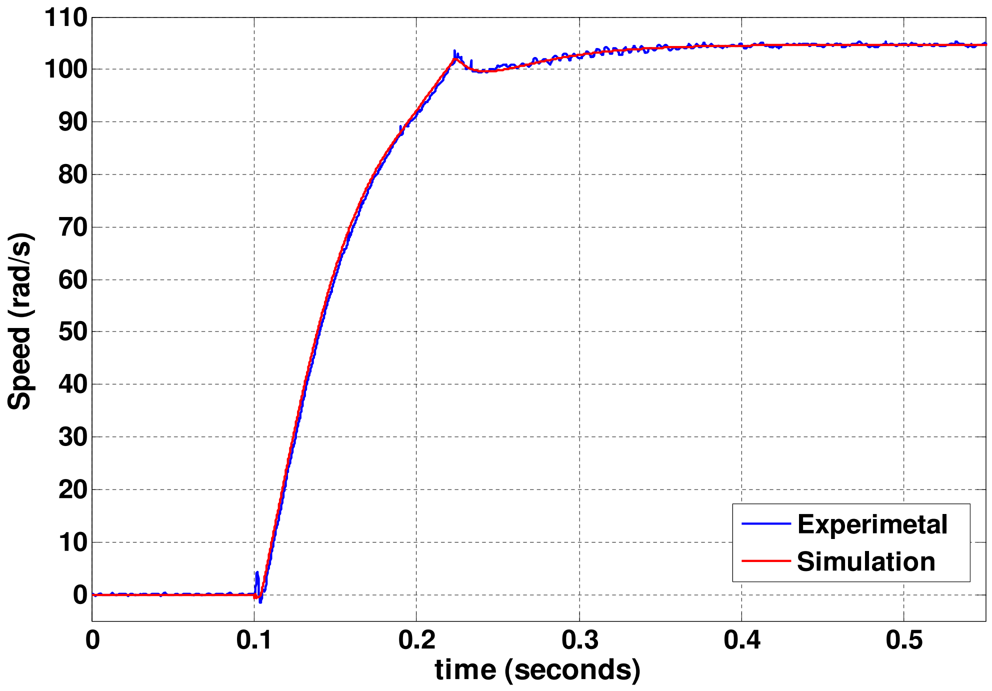


(a) Controller response to step demand of 0 to 1000 rpm

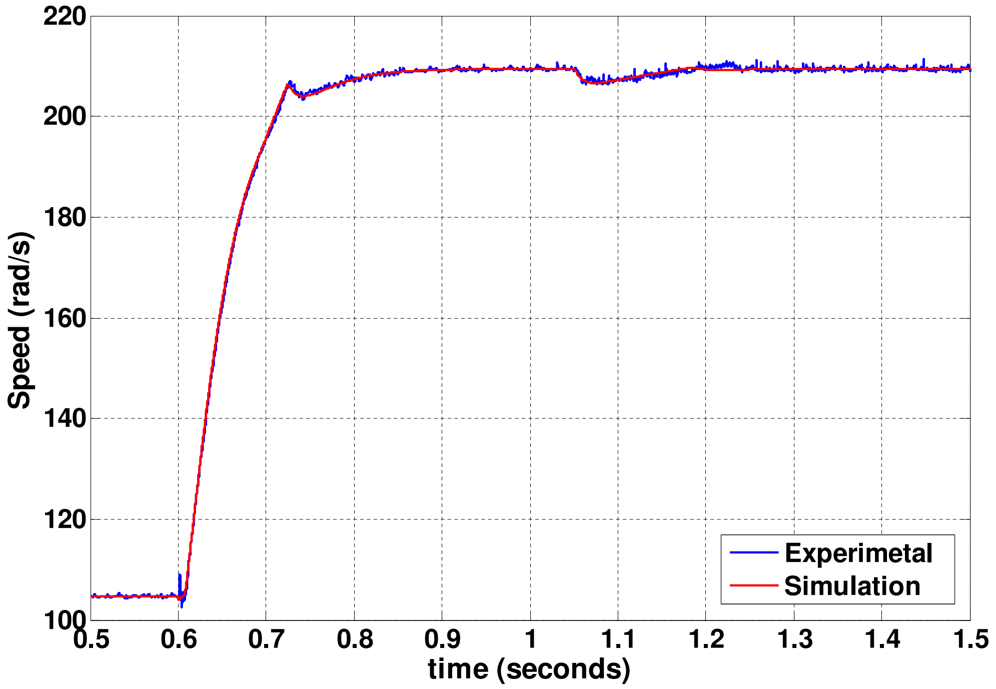


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.15: BF fourth order Controller Response for Inertia J_L and Damping $5D$

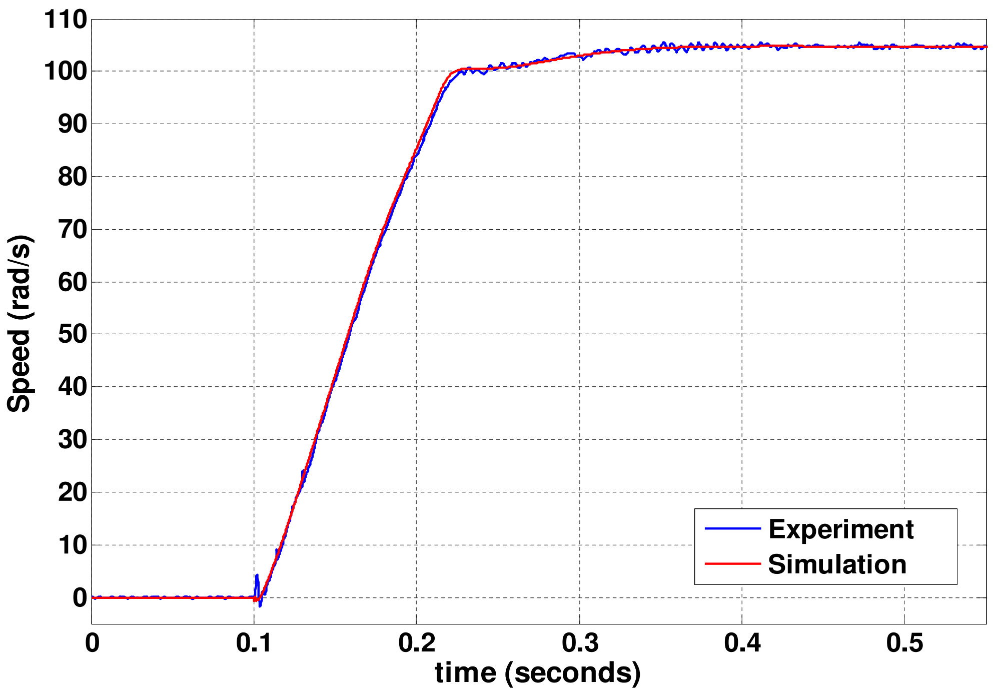


(a) Controller response to step demand of 0 to 1000 rpm

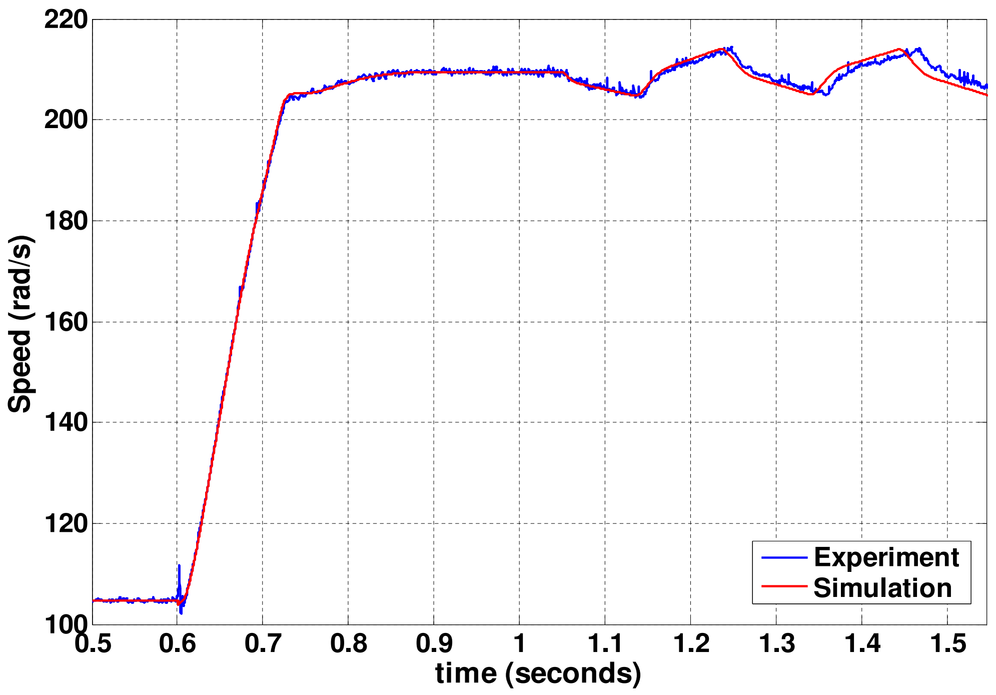


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.16: BF fourth order Controller Response for Inertia $3J_L$ and Damping $3D$

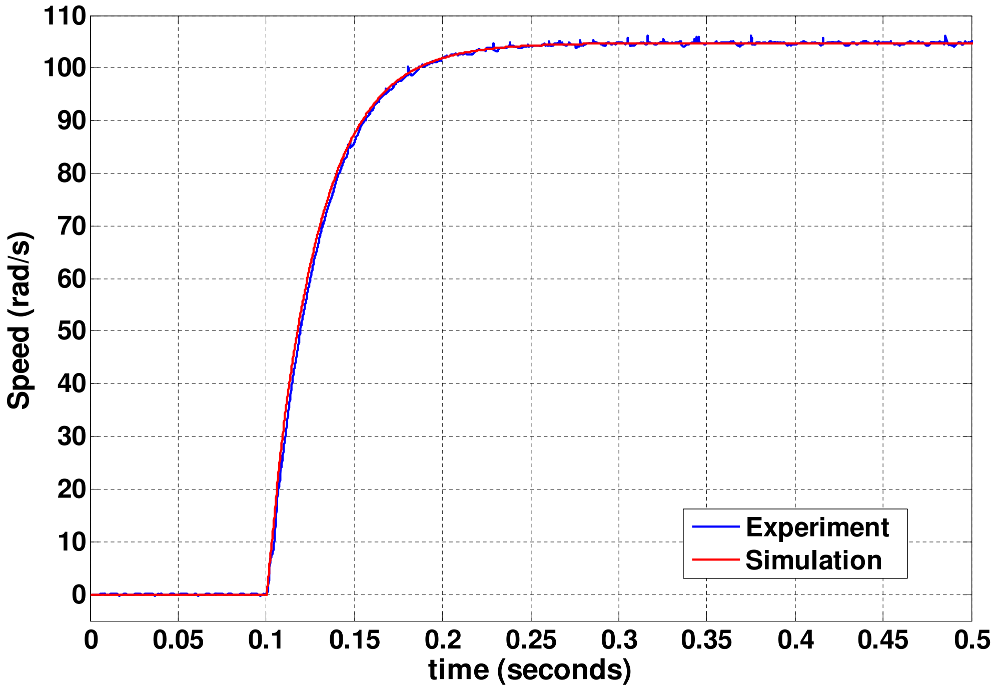


(a) Controller response to step demand of 0 to 1000 rpm

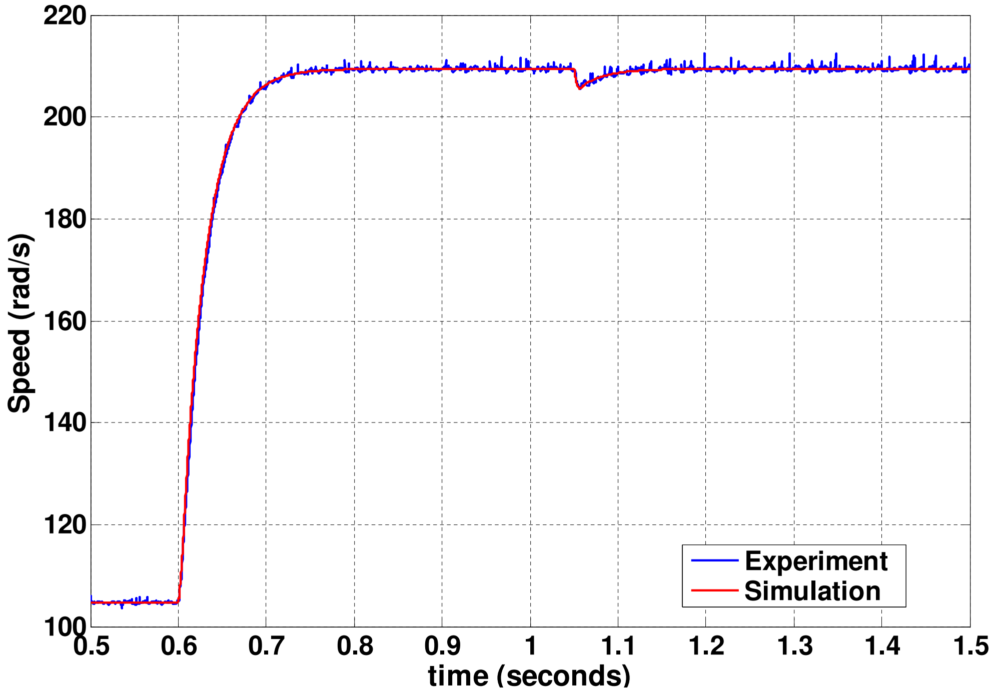


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.17: BF fourth order Controller Response for Inertia $5J_L$ and Damping D

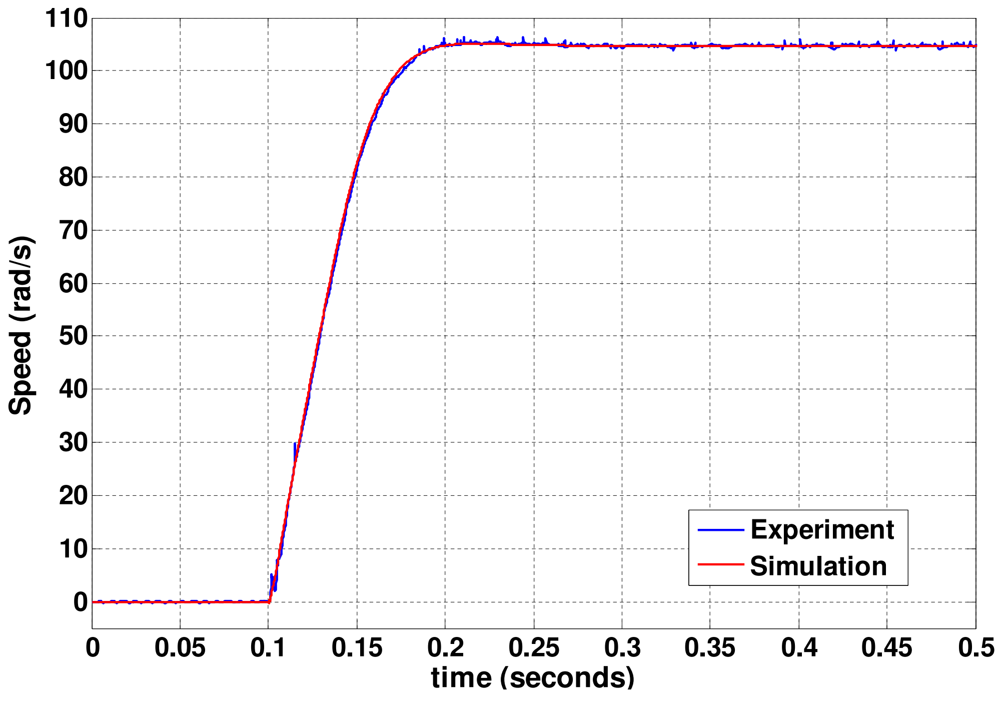


(a) Controller response to step demand of 0 to 1000 *rpm*

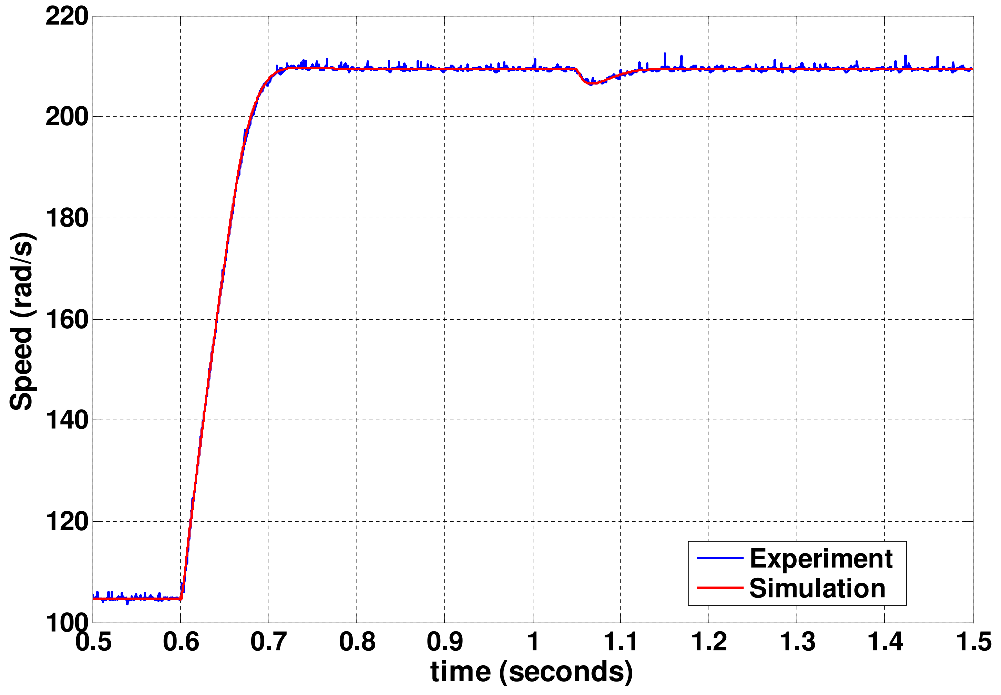


(b) Controller response to step demand of 1000 to 2000 *rpm* with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.18: HBF fourth order Controller Response for Inertia J and Friction $5B$

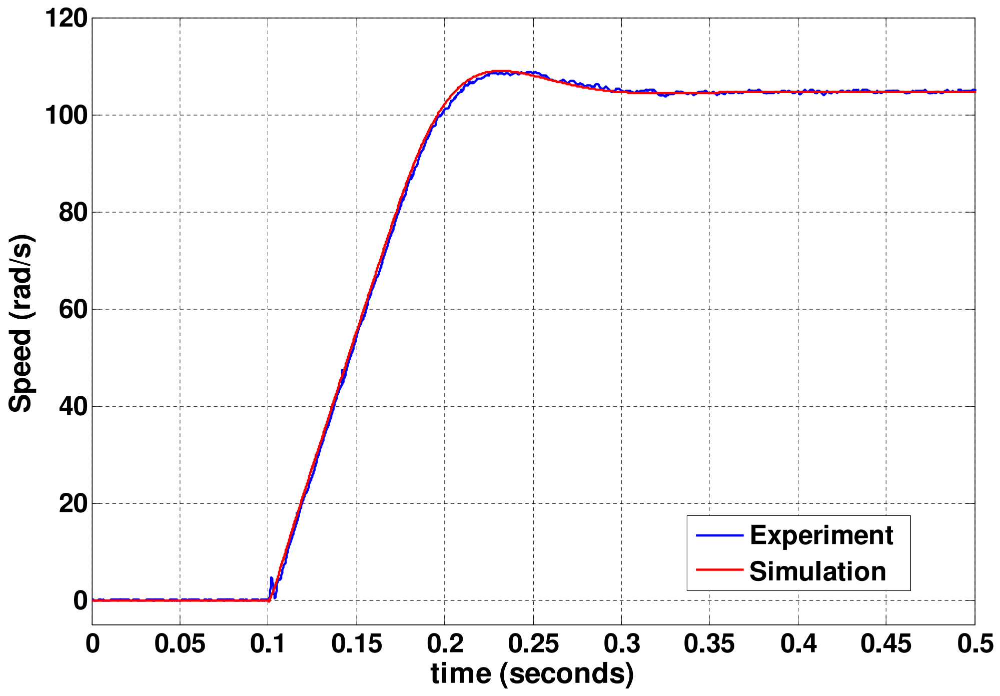


(a) Controller response to step demand of 0 to 1000 rpm

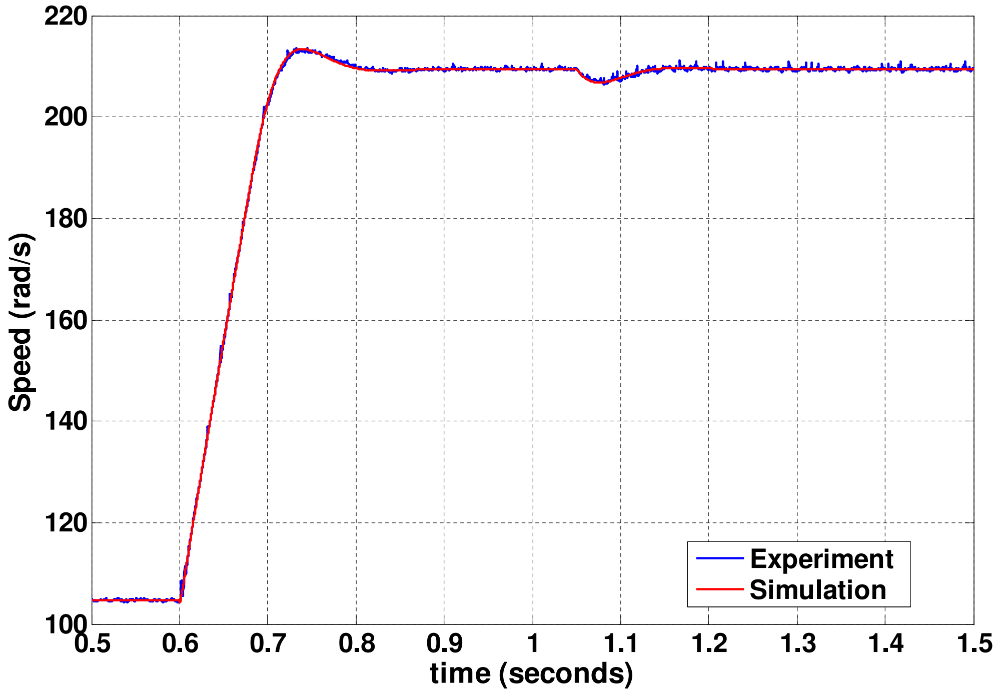


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.19: HBF fourth order Controller Response for Inertia $3J$ and Friction $3B$

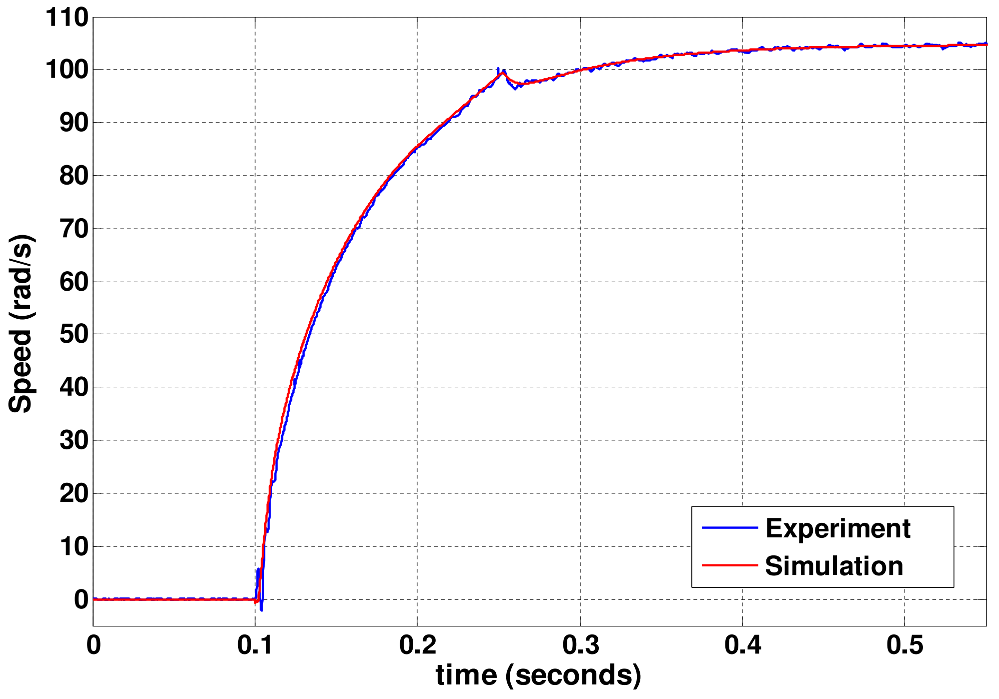


(a) Controller response to step demand of 0 to 1000 rpm

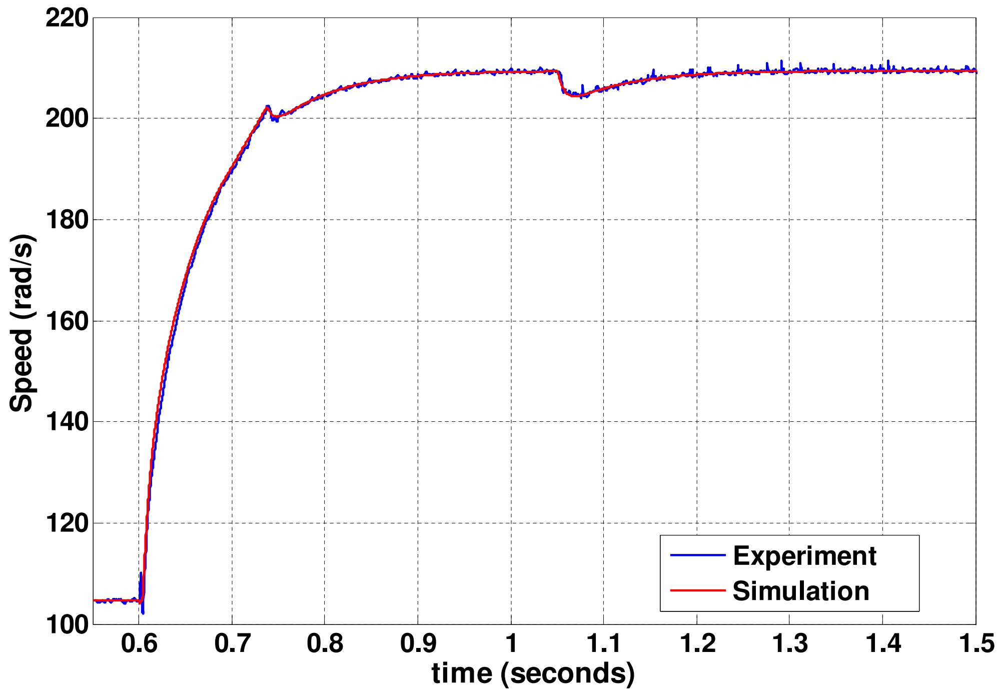


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.20: HBF fourth order Controller Response for Inertia $5J$ and Friction B

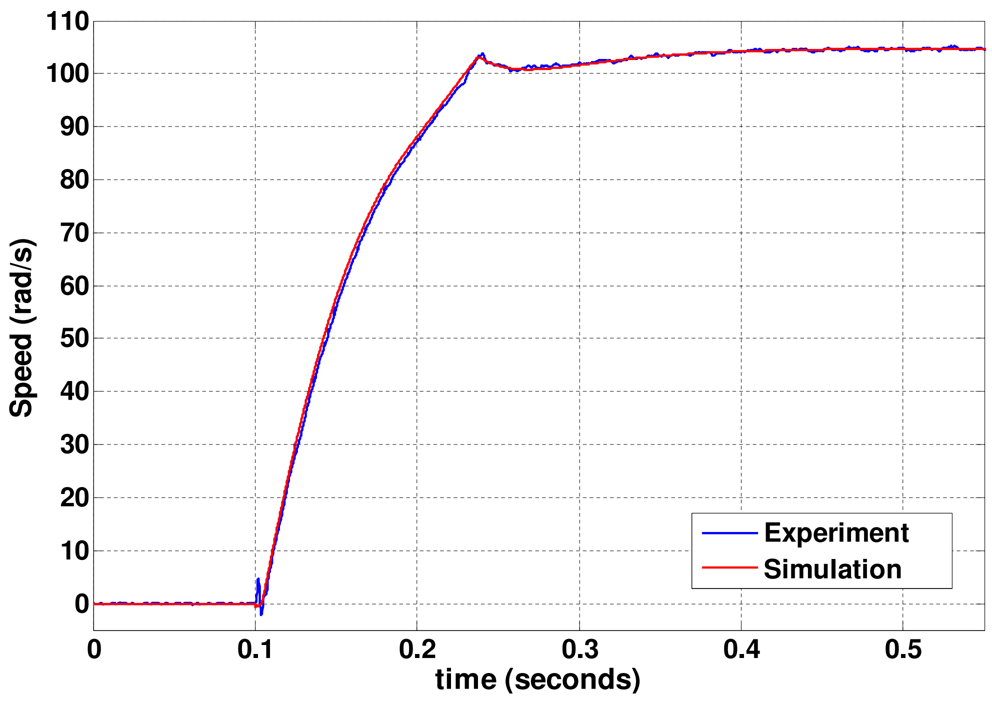


(a) Controller response to step demand of 0 to 1000 *rpm*

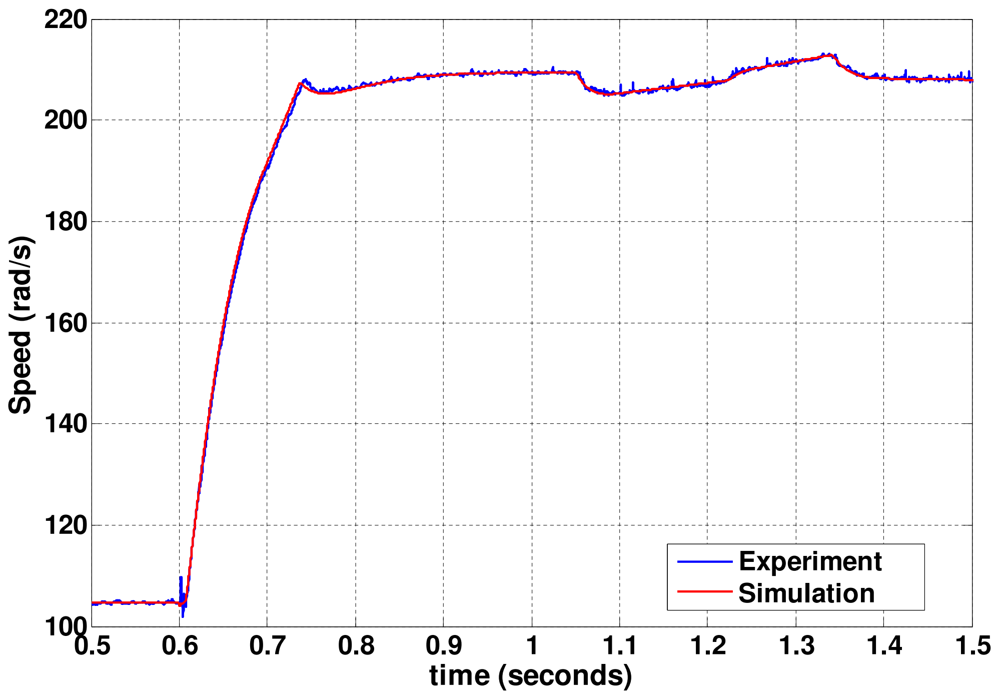


(b) Controller response to step demand of 1000 to 2000 *rpm* with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.21: HBF fourth order Controller Response for Inertia J_L and Damping $5D$

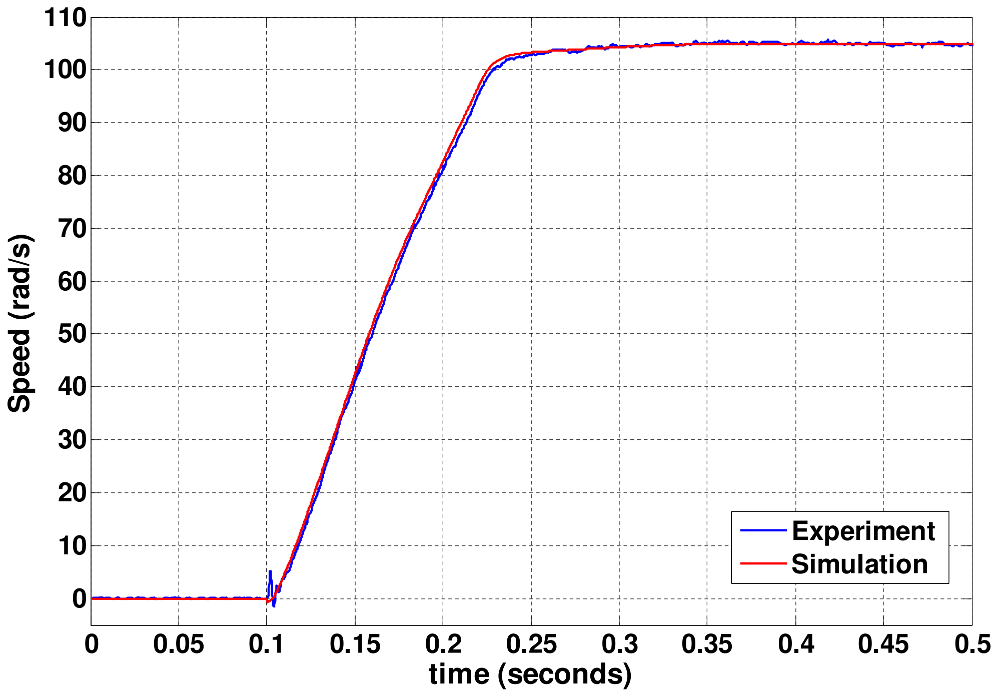


(a) Controller response to step demand of 0 to 1000 rpm

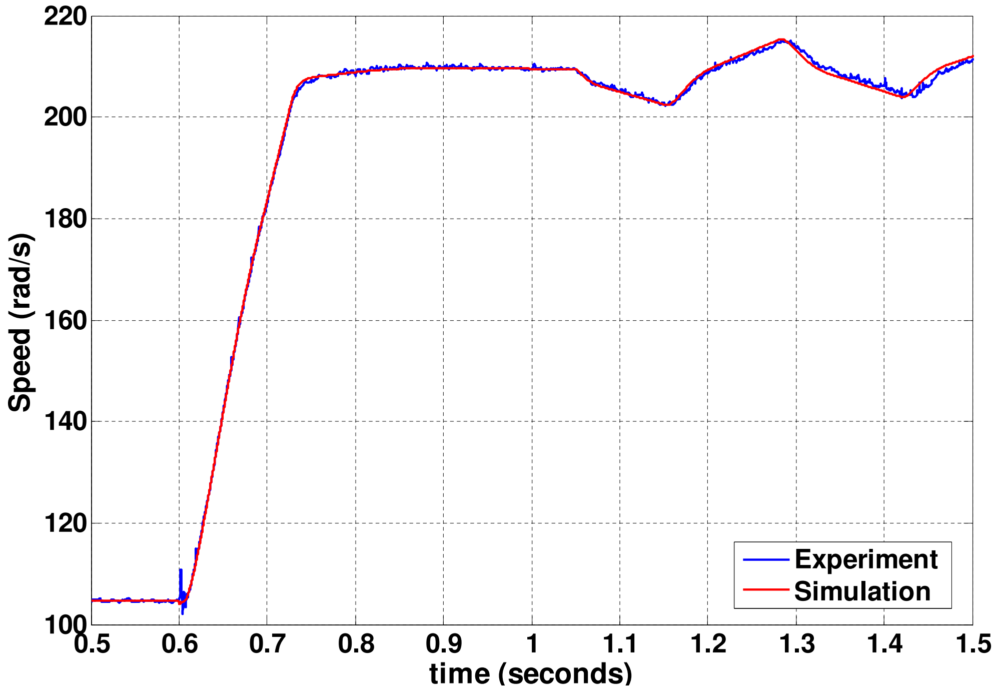


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.22: HBF fourth order Controller Response for Inertia $3J_L$ and Damping $3D$



(a) Controller response to step demand of 0 to 1000 rpm



(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 6.23: HBF fourth order Controller Response for Inertia $5J_L$ and Damping D

applied specifically to the optimisation of robust controllers for variable speed drives. The different components of the experimental setup are described and the schematics of the simulation and experimental systems that result from the combination of the different parts are shown. The experimental system comprises the digital controller, the evolutionary algorithms which optimises the controller while it is subject to the variable mechanical load. The variable mechanical load is a necessary part of the experimental system. It replicates the typical industrial operating load conditions for the drive. Hence, the digital controller designed under these operating conditions will enable the drive to perform robustly under industrial load conditions

The experimental approach to the optimisation process involves a series of processes - initialisation, evaluation, fitness assignment, selection, reproduction and termination. A crucial step is the evaluation process, which involves quantifying the quality of the possible solutions.The experiments require a series of real-time simulations of the experimental system for calculating the performance value of the individuals. Given the random nature of the process, it is necessary to incorporate protection circuits within the experimental set-up to stop the testing of badly performing individuals. These are implemented in both hardware and software.

The results of the automated optimisation process verify the effectiveness of the design approach. it also highlights the good agreement between the simulation model and the experimental system. It is also noted that the different Evolutionary Algorithms produce similar optimised speed responses. But, it would be interesting to compare the different algorithms in order to highlight the uniqueness of each. A detailed comparison of the algorithms employed is discussed in chapter 8.

The automated experimental optimisation process described in this chapter highlights important benefits of a simple and automated approach to the design of the Robust digital controllers for variable speed drives. It also does not require any detail on the model of the mechanical load in order to design robust digital controllers for such a system. The inherent disadvantages in the approach lies in the fact that it imposes stress and causes fatigue to the experimental system, thus shortening its life span.

Also, the time taken for the optimisation can be an issue. In order to address the problems, a theoretical approach to the design of Robust digital controllers has been investigated and it will be discussed in chapter 7.

Chapter 7

Robust Identification-Based Control Design

7.1 Introduction

Self-commissioning - the automatic design of current, flux and speed control parameters - is now available on most commercial electronic drives. Advanced commissioning schemes perform some simple tests on the machine to determine flux and torque estimators for Vector Control and Direct Torque Control. However the controllers employed are usually limited to Proportional plus Integral (plus derivative sometimes). This keeps the automated design process simple, but also means that the controllers can be "fine tuned" by commissioning engineers if required, without too much detailed knowledge of advanced control structures. The disadvantage of employing these PID controllers within the closed loop system is highlighted in the responses produced when the parameters of system it controls vary significantly during normal operation. Some drive applications require precise speed or position control, which standard self commissioning schemes cannot achieve. Machine tools for example require fast response, minimal overshoot, fast settling in the presence of variable inertia, large disturbance torques etc. For these applications, more sophisticated drive controllers are required, preferably self-commissioned with the drive undergoing its normal operating cycle so that the extremes of load behaviour can be seen. [12], [16], [41] and [65] proposed the use of Genetic Algorithms whereby the drive was allowed to operate in situ, and the GA gradually optimised the controller parameters (and order) to match predefined performance specifications. The advantage of this method was that the commissioning of a high performance speed/position controllers could be fully automated using routines run on a PC. Once commissioning was complete, the PC could be removed. This approach forms the subject of chapter 6. The main drawback was that the self-commissioning period was long, as many controller solutions had to be experimentally evaluated. Also, it was possible for "bad controllers" to be tested, adding mechanical stress to the commissioning process.

In this chapter, the focus will be on a new approach that does not involve direct experimentation on the physical hardware during the design procedure. The approach harnesses only mathematical/simulation models of the system under consideration. The benefits of the system identification-based robust control design approach is immediately highlighted by the fact that the physical hardware does not experience stress during the design procedure and hence, the issues surrounding the testing of bad controllers become insignificant. The design method incorporates the application of the investigated evolutionary algorithms together with the Robust Control theory and a system identification procedure. This chapter will describe the robust control theory, the system identification methods and demonstrate how it has been applied within the theoretical approach for robust control design. Robust control theory considers the design of controllers that fare well across a range of models. Given the necessary consideration of a range of models, robust control is inherently about model uncertainty, particularly focusing on the implications of model uncertainty for control design. The theory originated in the 1980s in the control theory branch of the engineering and applied mathematics literature, and it is now perhaps the dominant tool for Robust control theoretical design [15], [38]. System identification approaches provide a means to characterise system mechanisms by analysis of measured input and output data from the system. The analysis essentially involves fitting the measured data to a mathematical representation of the system [30]. The adoption of the

Evolutionary Algorithms in conjunction with these application, ensures the design method is automated in its procedure for synthesising robust control systems.

7.2 Robust Control

Robust control refers to the control of plants with unknown dynamics while they are subjected to unknown disturbances [75]. A basic desiderata for the practical implementation of robust control is that the system remain stable in the presence of perturbations; ensuring stability and good performance in the worst case scenario would ensure robust performance of control systems in the face of all possible perturbations [18]. Clearly, the main issue in achieving robust control is uncertainty and how the system can deal with this problem. Robust control theory attempts to provide systematic techniques and methods to solve control problems where uncertainty is a dominant issue [76]. The following sections will provide a description of uncertainty and highlight the methods used, in general, to design control systems in the face of uncertainty. They will focus on the particular method, frequency domain uncertainty modelling techniques, harnessed to achieve the aim of this work.

7.2.1 Uncertainty

The first stage when designing a closed loop control system create a model of the actual plant. Once this has been obtained, the control would be designed around it. This can be achieved using conventional approaches like the root-locus design or more commonly, the trial-and-error approach. This would involve randomly, but sensibly, evaluating different parameters for the controller until a suitable set of controller parameters, which give good closed loop system responses are obtained. The controller obtained through either approach is transcribed to the real system to verify the quality of the controller designed, using the plant model,on the actual plant.

The reality of models is that they are, at best, an excellent 'approximation' of the real system. The fact that it is an 'approximation' implies that there are bound to be discrepancies between the real and modelled plant. These discrepancies, termed modelling errors, can manifest itself as subtle parameter variations around the modelled value at frequencies comparable to the system dynamics that are very sensitive to operating conditions such as temperature, elevation from ground etc. It could also be in the form of un-modelled dynamics at high frequencies. Often, these discrepancies can be justifiably ignored, but in other cases, failure to model these dynamics accurately could result in significantly detrimental performance by controllers designed on the modelled plant.

Apart from plant modelling errors, there are system variables that are excluded when representing the actual closed system of the plant as a computer simulation model. These parameters can include noise from the system measurement devices such as transducers and they could also include external disturbances which are often very variable depending on the conditions within which the system is operating. The difficulty with modelling such random quantities, which are often negligible in magnitude, leads to the justifiable conclusion of excluding them within system models. In some cases, this is a viable option resulting in good designs of controllers. In other cases, the effects are dramatic giving rise to poorly designed closed loop control systems. These various un-modelled parameters, often neglected in theoretical designs are termed uncertainties.

There are two types of uncertainties that affect control systems:

1. Plant Uncertainty which arises due to errors in modelling, changes in parameters, inexact and incomplete data, modelling approximations
2. Signal Uncertainty which arises due to exogenous signals such as disturbances, sensor noise, etc [44], [77], [78].

The various sources of model uncertainty can be broadly grouped in three categories:

7.2.1.1 Parametric uncertainty

Parametric uncertainty arises from the fact that, within a linear model, there are always parameters that are approximately known and, in some cases, completely wrongly estimated. These parameters may also vary due to nonlinearities or a change in operating conditions. Parametric uncertainty can sometimes be referred to as *structured* uncertainty [44], [78].

7.2.1.2 Neglected and un-modelled dynamics uncertainty

Neglected and un-modelled dynamics uncertainty, which are errors within a model, result from missing dynamics arising from intentionally ignored high frequency behaviour. These missing dynamics could also be the result of the inability to model certain dynamics as a result of poor understanding of the physical process on which the model is based. This form of uncertainty is present in any model of a real system [44].

7.2.1.3 Lumped uncertainty

Lumped uncertainty simply combines the previous two categories of uncertainties in one lumped model. It represents several forms of parametric and/or un-modelled dynamics uncertainty combined into a single lump. Lumped model uncertainty is often referred to as *unstructured* uncertainty [44], [78].

There are three main concerns for control system engineers in meeting their objectives: observability, controllability and stability. Within a designed control system, observability is classified as the ability to observe all of the parameters or state variables of the system. Controllability defines the ability of the system to be controlled by forcing its state variables to change to any desired state. Stability is described as the bounded response produced by the system in response to a similarly bounded input to the system. Any successful control system will have to possess these three abilities but in the face of uncertainty, control design engineers will find it a task including all these properties within their designed systems especially given the limited system information often provided for the modelling process [75].

7.2.2 Robust Control Techniques

One method to deal with uncertainty in the past was stochastic control. This branch of control aims at reducing the bounds of uncertainty due to randomness within the system to be controlled. In stochastic control, uncertainties of a somewhat random nature in the system are modelled as probability distributions. These distributions are combined to produce a stochastic model for the design of the optimal control law. The justification for the combination is based on the fact that all random processes within nature are sum total of other independent contributing factors [75].

Robust control methods seek to bound the uncertainty rather than express it in the form of a distribution. One method of bounding the uncertainty is adopting a frequency domain technique (section 7.3). Given a bound on the uncertainty, the control can deliver results that meet the control system requirements in all cases. Therefore robust control theory can be described as a worst-case analysis method rather than a typical case method. Often, in order to satisfy certain robust performance requirements, some performance must be sacrificed [18]. There are currently a variety of techniques that have been developed for robust control. Brief descriptions of some methods are provided, highlighting their basic concepts.

7.2.2.1 Frequency Domain $\mathcal{H}_2$ and $\mathcal{H}_\infty$ techniques

The Frequency Domain techniques provide a viable approach to tackling the problems of robustness and performance in the design of robust control systems. In order to adopt a quantitative method with the application of the Frequency Domain Techniques, system norms are adopted. These system norms are single numbers, which give measures of the overall magnitude of the systems being investigated. The system norms that lend themselves particularly well in the design of optimal control systems are the *Hardy Space Norms* [44]. These norms are called after the British pure mathematician G. H. Hardy (1877 1947).

Hardy Space Norms are employed for characterising *stable systems*. The relevant system norms used in the frequency domain technique for robust control system design are the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. The overall aim of the Hardy space inspired Frequency Domain techniques is to achieve a system that is robust in the face of plant model uncertainty [44]. Further details on the frequency domain techniques applied in achieving the project's aims are provided in section 7.3.

7.2.2.2 Adaptive Control

An adaptive control system harnesses observers in tracking each significant state variable in the system. The system can adjust each observer to account for time varying parameters of the system. In an adaptive system, there is always a dual role of the control system: The output is controlled to eventually equal the desired input while, simultaneously, the system continues to adapt and improve its performances according to changes in the system parameters [79].

7.2.2.3 Parameter Estimation

Parameter estimation technique establishes boundaries in the frequency domain that cannot be crossed to maintain stability. These boundaries are evaluated by using uncertainty vectors. This technique is graphical and has some similarities to the root locus design method. The technique proposes certain benefits on providing clues to the user on how to change the system to make it more insensitive to uncertainties.

7.2.2.4 Sliding-Mode Control

This is another approach to robust control. It is a are part of a class of Variable Structure Control Systems (VSCS). These systems are characterised by sets of feedback control laws and decision rules, which are essentially switching functions. Particular feedback control structures are switched to or adopted depending upon the system behaviour being exhibited that needs to be controlled. The clear advantage of such an arrangement is its ability to combine several different structures within a single system. Also, its closed-loop responses become insensitive to certain classes of uncertainty, given the control laws are defined in a region-specific manner. The issues with its adoption are the need to understand the possible behaviours that can be exhibited by the system, appropriately defining the different control laws for these different regions of behaviour and implementing the decision laws to enable correct switching in between the feedback control structures [60].

7.3 Frequency Domain $\mathcal{H}_\infty$ Method

Classical frequency domain control design methods deal with the issue of robustness using gain and phase margin settings [80]. The presence of interactions in cross-coupled multivariable systems make these methods unreliable indicators of robustness in control systems. The incorporation of the Hardy space norms make the traditional frequency domain techniques readily applicable to problems involving multivariable systems with cross-coupling between channels. As a result, the inclusion of the Hardy Space norms can be viewed as a concept that has evolved the application of classical frequency domain methods.

The choice of implementing Robust Control System Designs in this research project via the Hardy space inspired frequency domain techniques is guided by the fact that these techniques are of recent vintage within the field of Control Engineering, hence there is probably a lot of exploration to be undertaken in order to exploit its full potentials. It has also been highlighted that the Hardy Space inspired frequency domain techniques enable dealing with the issues of robustness and performance far more directly when compared with other optimisation techniques [81].

The frequency domain $\mathcal{H}_\infty$ technique has been selected in order to implement the design of robust control systems for the mechanical plants investigated within the research projects. The preference for the $\mathcal{H}_\infty$ over the $\mathcal{H}_2$ method is due to several reasons. One of the reasons stems from the description of the norms: the $\mathcal{H}_2$- and $\mathcal{H}_\infty$-norms are expressed in (7.1) and (7.2) respectively. The $\mathcal{H}_2$ norm measures the 'average' gain of the system $f(s)$ taken over all frequencies while the $\mathcal{H}_\infty$ norm, which is perhaps a more fundamental norm for systems, characterises the 'maximum' gain of the system over the frequency domain [82].

$$\|f(s)\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} |f(j\omega)|^2 \, d\omega\right)^{\frac{1}{2}} \tag{7.1}$$

$$\|f(s)\|_\infty = \underset{\omega}{max}\,|f(j\omega)| = \lim_{p\to\infty}\left(\int_{-\infty}^{\infty} |f(j\omega)|^p \, d\omega\right)^{\frac{1}{p}} \tag{7.2}$$

A highlighted problem with the approach that employs $\mathcal{H}_2$ norm is it is only able to deal with the system, $f(s)$, when it has specific input signals (*.e.g.* impulse inputs); it becomes difficult to implement the $\mathcal{H}_2$ approach with systems that have a range of input signals (*.e.g.* step and ramp inputs). On the other hand, the $\mathcal{H}_\infty$ approach can be conveniently applied to the system, $f(s)$, with a range of input signals [83].

The frequency domain $\mathcal{H}_\infty$ approach, in comparison to the $\mathcal{H}_2$ method, can also be conveniently applied in representing both structured and unstructured uncertainty within models. It does not seem to have much competition (when compared with other norms) in terms of quantifying unmodelled dynamics uncertainty [44]. Prominent authors within the field of robust control theory, Owen and Zames made the following observation in [44], [84]: "*The design of feedback controllers in the presence of non-parametric and unstructured uncertainty... is the raison d'etre for $\mathcal{H}_\infty$ feedback optimisation. If the disturbances and plant models are clearly parameterised then $\mathcal{H}_\infty$ methods seem to offer no clear advantages over more conventional state-space and* *parametric methods*".

For these reasons, the frequency domain ($\mathcal{H}_\infty$) approach has been employed in the research project for the design of robust control systems. Its application is implemented in two stages; the first stage involves defining the uncertainty model for the plant to be controlled - this is at the heart of the problems concerned with designing robust control systems. The second stage involves designing the control system around the plant uncertainty model obtained. This stage employs the evolutionary algorithms in a random search procedure to achieve an optimised design (section 7.4).

7.3.1 Frequency Domain Uncertainty Models

In order to ensure that the control systems that are designed based on the proposed method are truly robust to all possible parameter variations, it is necessary to have a plant model that considers the uncertainty. A means to achieve this is to obtain a model that bounds the different uncertainties that can be experienced in reality. The model derived thus is termed an *'uncertainty model'*. The uncertainty model, in essence, is a somewhat more accurate mathematical representation of the actual plant, given it considers the possible system variations. The closed loop control system is then designed around the uncertainty model. As an illustration, the actual uncertainty within a model can be represented in figure 7.1

The aim of selecting a suitable uncertainty model is to bound the actual uncertainty using a disc-like approximation. In order to bound the uncertainty *i.e.* define the uncertainty model of the plant, it is necessary to first categorise the general types of uncertainties that exist within models. Once a suitable uncertainty classification is achieved, using certain approaches, appropriate models for these can be defined.

There are a number of models, described using the $\mathcal{H}_\infty$ frequency-domain paradigm, that are used in robust control design to consider *unstructured* uncertainties within models. This subsequent sections describes the different models for plant uncertainties

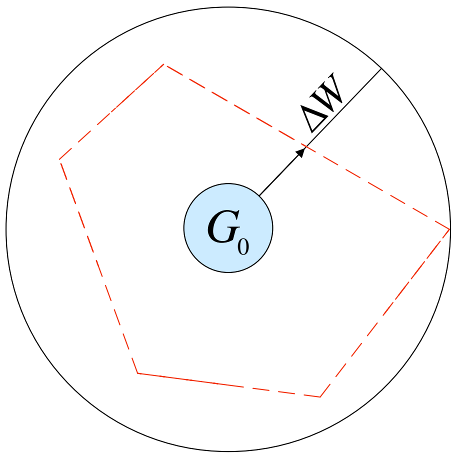


Figure 7.1: Disc-like approximation (black-line) of actual uncertainty (red-line)

and eventually discusses the uncertainty model chosen and the underlying reasons for the selection. Generally, the uncertainty models considered are the multiplicative, additive and the two feedback perturbation models. These models can also be used to consider *structured* uncertainty.

7.3.1.1 Multiplicative uncertainty model

The multiplicative uncertainty/perturbation model is the most commonly used in robust control design when considering unstructured uncertainty [44].

$$G_{em}(s) = G_0(s)(1 + \Delta(s)W_m(s)) \tag{7.3}$$

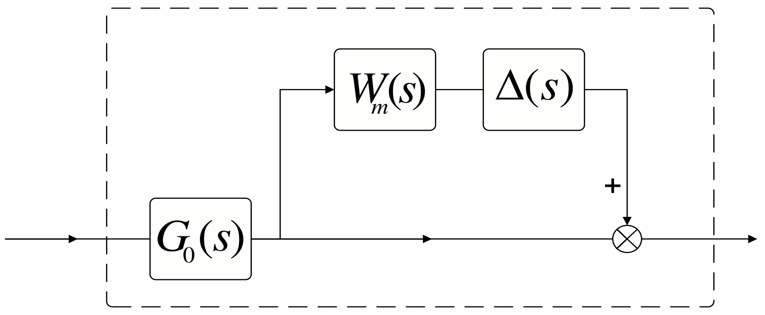


Figure 7.2: Multiplicative uncertainty model

It can effectively capture a wide variety of model uncertainty including *structured* as well as neglected and unmodelled dynamics uncertainty [44], [78]. The multiplicative perturbation is defined mathematically in equation (7.3). $G_0(s)$ represents the nominal transfer function, $\Delta(s)$ is any stable transfer function which at each frequency has a magnitude that is not larger than one. $W_m(s)$ is the Weighting function that represents the upper bound of the multiplicative uncertainty. It is a weight that is introduced in order to normalise the perturbations to be less than one in magnitude at each frequency. Its model is represented in figure 7.2 [44], [78].

7.3.1.2 Additive uncertainty model

Implementing the additive uncertainty model implies that real plant $G(s)$ would have a transfer function represented as in equation 7.4.

$$G_{em}(s) = G_0(s) + \Delta(s)W_a(s) \tag{7.4}$$

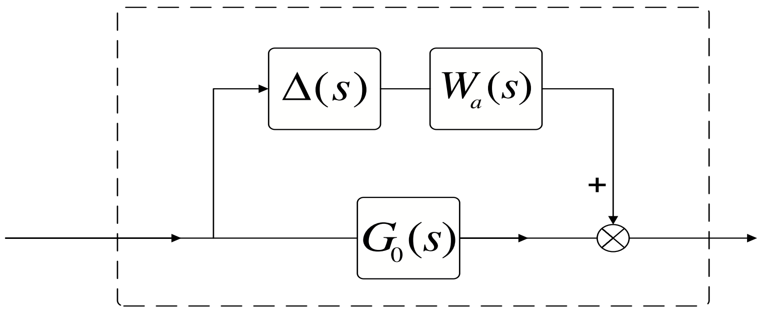


Figure 7.3: Additive uncertainty model

$W_a(s)$ is the Weighting function that represents the upper bound of the additive uncertainty. The additive perturbation model can be represented as in figure 7.3 [78].

7.3.1.3 Feedback uncertainty model

These models are especially useful in accounting for the different types of uncertainty. They can be implemented in two different ways. The first model is shown in (7.5) and the corresponding schematic is in figure 7.4.

$$G_{em}(s) = \frac{G_0(s)}{1+\Delta(s)W_f(s)} \tag{7.5}$$

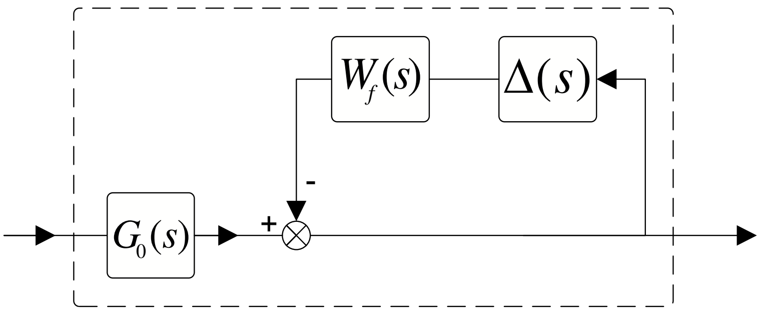


Figure 7.4: Feedback uncertainty model(one)

A second possible model is shown in (7.6). Its corresponding schematic is in figures 7.5 $W_f(s)$ is the Weighting function that, in both cases, represents the upper bound of the feedback uncertainty [44], [77], [78].

$$G_{em}(s) = \frac{G_0(s)}{1+\Delta(s)W_f(s)G_0(s)} \tag{7.6}$$

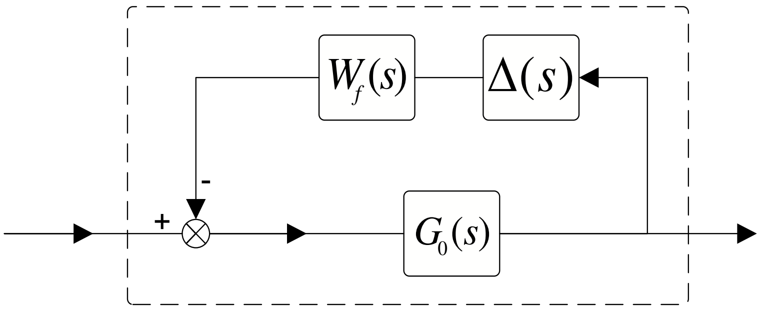


Figure 7.5: Feedback uncertainty model(two)

The uncertainty model, described in (7.6), has been selected in order to model the uncertainty that exists within the plant case studies for which control systems is to be designed in this project. The plants considered are the stiff and flexible Shaft mechanical loads, which have been described in chapter 5. The subsection 7.4.2 will focus on describing the parameters of the feedback uncertainty models adopted for each of the mechanical loads.

7.3.2 Frequency Domain $\mathcal{H}_\infty$ Control Design

The objective of the frequency domain $\mathcal{H}_\infty$ control technique is to synthesise controllers that are capable of achieving robust performance in a closed loop system that has uncertainty within its plant. The $\mathcal{H}_\infty$ control method can be seen as worst case optimisation because it amounts to minimising the effects on the output of the worst input (disturbance, noise, *etc*) on the closed loop system [81]. The procedure for employing the $\mathcal{H}_\infty$ method can be summarised in two steps:

1. specify the control task as a 'mathematical' optimisation problem
2. solve the specified optimisation problem

7.3.2.1 Defining the Mathematical Optimisation Problem

The aim of robust control is to design systems that are stable, especially in the face of perturbations. A feature of stable systems is they have defined maximum limits on their outputs: for an input of any frequency, the stable system would eventually settle at a particular output value. A useful indicator of how well a system defines boundaries on possible input signals is provided by the '$\mathcal{H}_\infty$ norm'; the norm indicates the system's peak gain value and the frequency at which it occurs.

The frequency domain $\mathcal{H}_\infty$ control method essentially deals with minimising the $\mathcal{H}_\infty$ norm (*.i.e.* peak value) of certain closed loop frequency response functions. These frequency response functions define the relationship between the input signals (disturbance, noise, *etc*), whose effects are to be controlled, and the output of the closed loop system. For the purpose of achieving solutions for robust performance (see 7.4.6), the frequency response functions to be minimised often consists of functions that complement one another. This means that a minimum for one function will give a maximum value for the other existing function. As a result, the optimisation problem for the $\mathcal{H}_\infty$ control method is not a straightforward task. In obtaining solutions for nominal performance (see 7.4.4), the mathematical optimisation task is more direct. The frequency response functions present within the mathematical optimisation problem will depend on the uncertainty model chosen to represent the plant perturbations.

7.3.2.2 Solving the Optimisation Problem

The solution of the defined mathematical optimisation problem is the controller that minimises the peak value of closed loop frequency response functions that exist within the optimisation problem. One of the largest disadvantages of the $\mathcal{H}_\infty$ control method is the inability to fix the controller structure *a priori*. Generally, the $\mathcal{H}_\infty$ method provides controller solutions with high order. These solutions are often obtained by trial-and-error (root-locus) approaches, which makes it difficult to discover and implement. In such cases, having obtained the high order control solution, there is the need to apply order reduction methods to achieve final solutions with reduced order [80], [81].

In order to avoid the difficulty of finding and implementing high order controller solutions, certain design methods have adopted the approach of fixing the structure of the controller *a priori*, preferably to one of low-order (PID) controller, and then obtaining the relevant parameters for the chosen controller structure through a similar trial-and-error procedure [80].

In both these approaches, the quality of the final control solution is sacrificed. In one case, the compromise is due to the difficulty in implementing the high order solution, which theoretically has the potential of producing better closed loop control system responses. In the other case, the compromise stems from restricting the possible control structures that can be optimised by initially selecting a low order solution. Also, the trial-and-error approach by which the control parameters are selected under both methods makes it difficult to make an exhaustive search of all the possible solutions.

The approach proposed in this research project substitutes the traditional, trial-and-error search procedure with a simple and automated search procedure that employs particular evolutionary algorithms. The benefits are highlighted in the exhaustive search capabilities that the evolutionary algorithms provide [13], [37]. With regards to the optimisation of the controller, the structure of the possible control solution is not selected *a priori* or restricted to a low-order structure. Due to the exhaustive search capabilities of the EA, there is the possibility of testing different order controllers (up to fourth-order) and subsequently selecting the best performing controller for the closed loop system from multitudes of possible solutions.

7.4 Robust Control System Design

Generally, in designing a robust control system, there are three steps involved:

1. identification of the nominal/average model of the plant.
2. defining the uncertainty model of the plant
3. designing the closed loop system around the uncertainty model

The design of the closed loop system involves an automated trial-and-error approach that requires the particular EA. During the design process, each randomly selected control system is evaluated against certain criteria in to determine if it meets the overall objective of robust performance. The evaluation criteria are

- *Nominal stability*: stability with no model uncertainty
- *Nominal performance*: performance with no model uncertainty
- *Robust stability*: stability with model uncertainty
- *Robust Performance*: stability and performance with model uncertainty

7.4.1 Nominal Model Identification

Defining the parameters of the nominal model is a typical system identification problem. A system identification problem can be formulated as an optimisation task where the objective is to find a model and a set of parameters that minimise the prediction error between the plant outputs i.e, the measured data, and the model output. Existing system identification models approaches are highly analytical and based on mathematical derivation of the system's model.

The are also other approaches, which are experimentally based on fitting a model to recorded data by assigning numerical values to its parameters. Evolutionary computing can be used in conjunction with these different approaches for the purpose of matching recorded experimental data with analytically derived general models. Employing evolutionary computing approaches seems very promising as it requires little knowledge about the problem [85], [86], [87].

The identification process involves defining the nominal system parameters for the mechanical loads representing the plant of the system for which controllers are to be optimised. The two mechanical loads considered are the stiff and flexible shaft load. The GA is used to identify the nominal/average total inertia, J_{em} and total friction, B_{em} of the stiff shaft mechanical load and the total load inertia, $J_{L_{em}}$ and damping, D_{em} of the flexible shaft mechanical load. The parameters are identified as a function of their drive shaft speed. The two stage process for the identification of the mechanical load system parameters involves:

1. Experimental Data Capture
2. Off-line System Identification using GA

The identification process can also be used to derive non-linear models that predict the behaviour of the system parameters over a speed range. For the targets of this research project, only the average model identification process was adopted in order to simplify the nominal model with which to perform further analysis.

7.4.1.1 Experimental Data Capture

For the first stage, the drive control is designed using the approximated rated values of the nominal system parameters for the Stiff and Flexible shaft mechanical load. Sub optimal performance is achieved but this does not matter at the moment. The response of the real system is captured for a series of simple speed and torque transients, which can be executed in less than one minute.

7.4.1.2 Off-line System Identification using GA

The basic idea is to use a GA routine off-line to identify the motor mechanical parameters in the simulation model, by recursively running the simulation and trying to minimise the error between the real measured speed responses and the simulated one under the same experimental conditions. In the simulation model the control parameters are kept the same as those implemented in the DSP of the experimental set-up, while the motor mechanical parameters are changed by the GA at every new simulation in order to match the simulation speed responses to the real measured ones that serve as the reference. The summary of the system identification process is presented in figure 7.6

All the information on the mechanical parameters is coded into two element strings, known as individuals that represent the inertia and friction values to be identified. At the initialisation of the Genetic Algorithm, a specified number of individuals and generations are chosen, and the first population of individuals is generated randomly. Each individual is tested on the Simulink model and the speed closed loop dynamics is obtained. The quality of the response obtained is quantified using the Fitness function. In this case, the fitness function is simply defined as the Integration of The Absolute Error (IAE) between the experimental data and the response obtained from the model. This implies that smaller the fitness value, the better the match between the individual and the real mechanical parameters. At the end of the trials of each individual within one generation, the GA ranks the individuals based upon

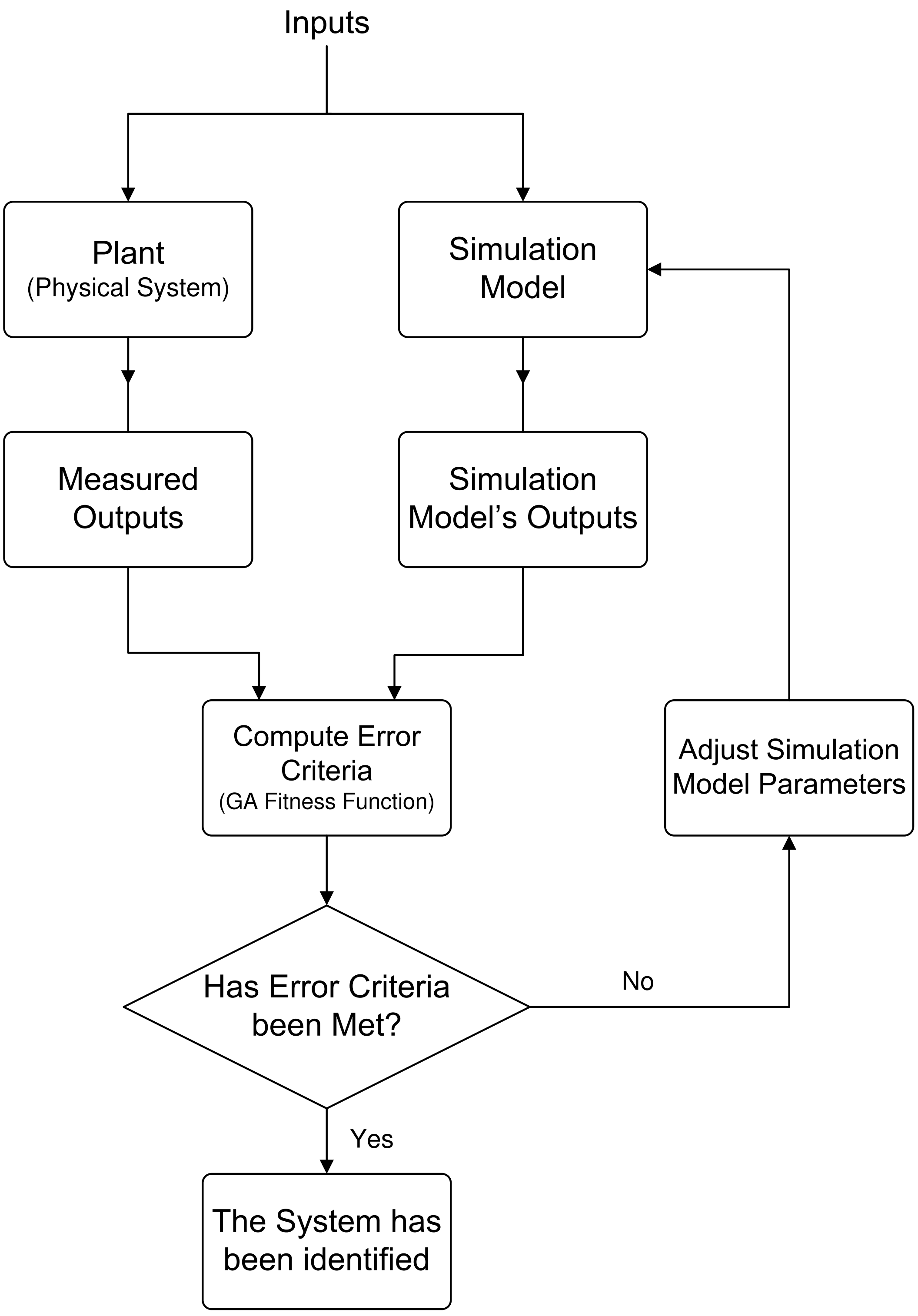
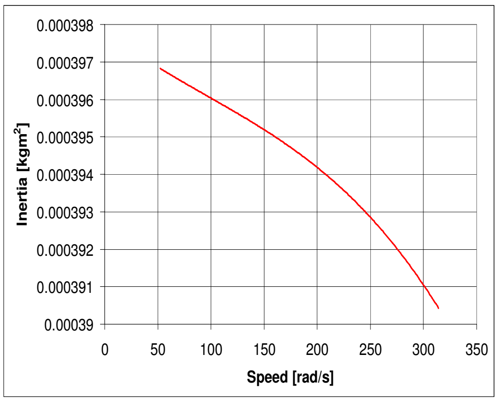
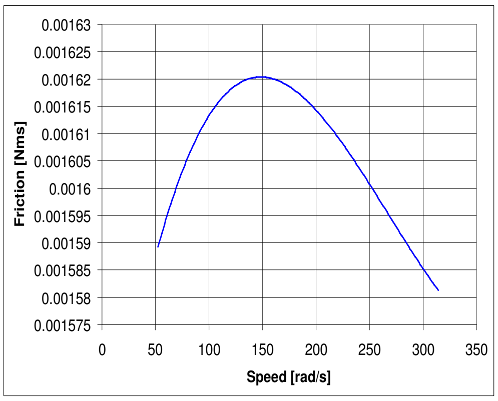

Inputs
Plant
(Physical System)
Simulation Model
Measured Outputs
Simulation Model's Outputs
Compute Error Criteria
(GA Fitness Function)
Adjust Simulation Model Parameters
Has Error Criteria been Met?
No
Yes
The System has been identified

Figure 7.6: System identification process: used in the nominal model identification for the mechanical loads

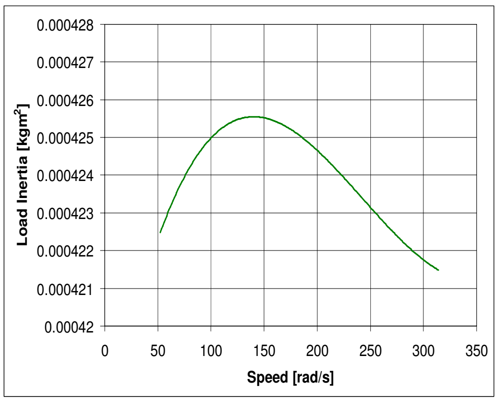
their corresponding fitness values. The individual with the lower fitness values rank higher and vice versa. A number (defined by the user) of high ranking individuals are then passed onto the next generation (elitist selection). The remaining members of the new generation are formed by performing evolutionary processes of mutation and crossover on the remaining lower ranking individuals.



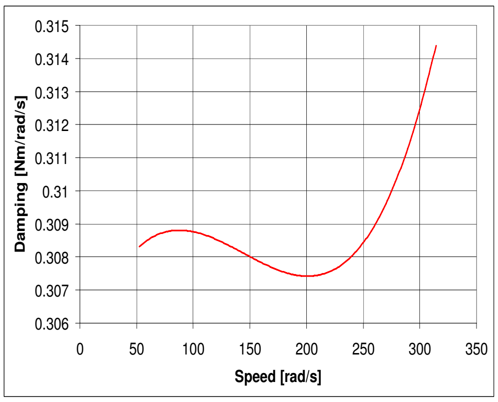
(a) Identified stiff-shaft total inertia



(b) Identified stiff-shaft total friction



(c) Identified flexible-shaft load inertia



(d) Identified flexible-shaft damping

Figure 7.7: Identification of Load Machine Current loop

Stiff Shaft mechanical model

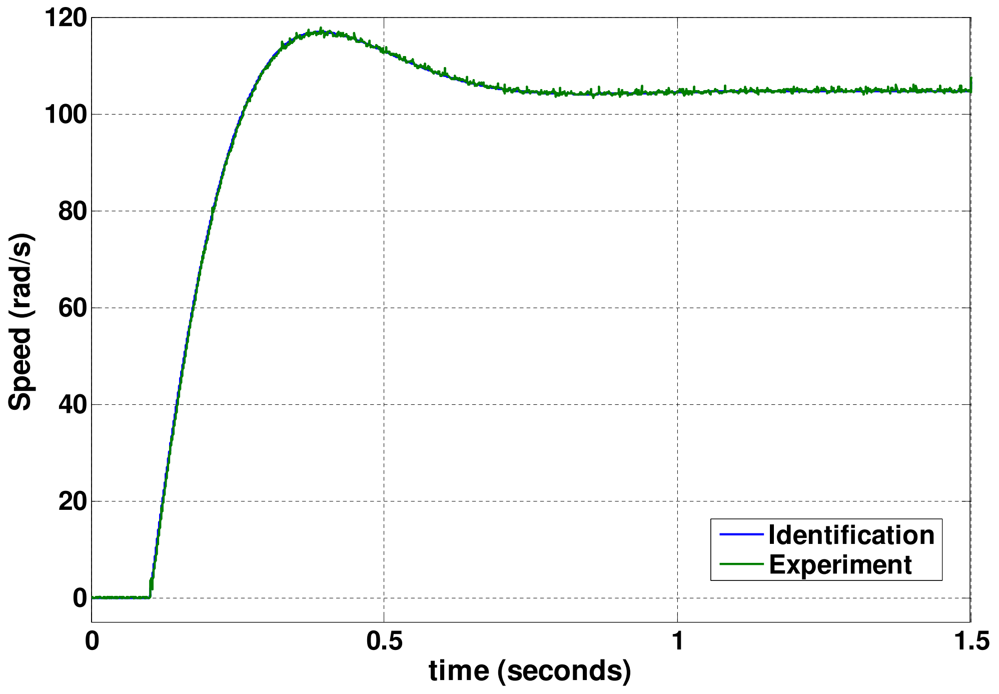
The nominal system parameters have been identified using the described identification process: $\bar{J}_{em}$ is equal to 0.000394 Kgm^2 and $\bar{B}_{em}$ is equal to 0.0016 Nms. Its transfer function is given in equation (7.7). Figures 7.7(a) and 7.7(b) show the results of the parameters of total inertia and friction identified for the load within the considered speed range of 0 - 3000 rpm. Figure 7.8 shows the match between the experimental and identified model for the stiff shaft mechanical load. In figure 7.8(a), the match between the identified model and the experimental system for a speed transient of $0rpm$ to a $1000rpm$ is shown. To further demonstrate the accuracy of the identification, the match between the experimental system and its model in their speed transients produced as a result of a step demand from $1000rpm$ to $2000rpm$ in the presence of an external load torque disturbance of $0.15Nm$, introduced after $2.75s$, is shown in figure 7.8(b).

$$G_0(s) = \frac{1}{\bar{J}_{em}s + \bar{B}_{em}} = \frac{1}{3Js + 3B} \tag{7.7}$$

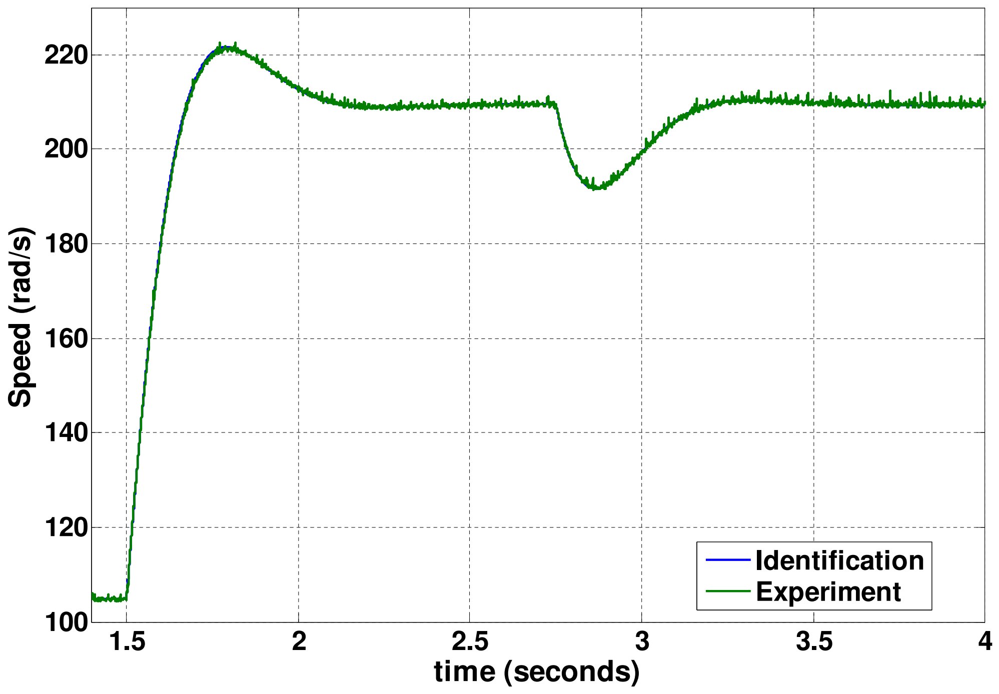
Flexible Shaft mechanical model

The nominal system parameters have also been identified using the described identification process: $\bar{J}_{L_{em}}$ is equal to 0.00042 Kgm^2 and $\bar{D}_{em}$ is equal to 0.3 $Nm/rad/s$. Its transfer function is given in equation (7.8). Figures 7.7(c) and 7.7(d) show the results of the parameters of total load inertia and damping identified for the load within the considered speed range of 0 - 3000 rpm. Figure 7.9 shows the match between the experimental system and the identified model for the flexible shaft mechanical load. The speed transient produced by a step input of $0rpm$ to a $1000rpm$ by the experimental system and the identified model is shown in figure 7.9(a). The excellent agreement that further highlights the effectiveness of the identification process is shown in figure 7.9(b); in the figure, the speed transients produced by a step input from $1000rpm$ to $2000rpm$ in the presence of load disturbance torque of $0.15Nm$, introduced after $3.5s$, has been obtained for the experimental system and its model.

$$\begin{aligned} G_0(s) &= \frac{\frac{\bar{D}_{em}}{K_{em}}s + 1}{(J_{M_{em}} + \bar{J}_{L_{em}})s\left(\frac{J_{M_{em}}\bar{J}_{L_{em}}}{(J_{M_{em}} + \bar{J}_{L_{em}})K_{em}}s^2 + \frac{\bar{D}_{em}}{K_{em}}s + 1\right)} \\ &= \frac{\frac{3D}{K_{em}}s + 1}{(J_{M_{em}} + 3J_L)s\left(\frac{J_{M_{em}} \times 3J_L}{(J_{M_{em}} + 3J_L)K_{em}}s^2 + \frac{3D}{K_{em}}s + 1\right)} \end{aligned} \tag{7.8}$$



(a) Identification of stiff shaft nominal model: Speed transient to a step input 0 - 1000rpm



(b) Identification of stiff shaft nominal model: Speed transient to a step input of 1000 - 2000rpm and external disturbance of $0.15Nm$ introduced after $2.75s$

Figure 7.8: Identification of stiff shaft nominal load

7.4.2 Mechanical load Uncertainty Model

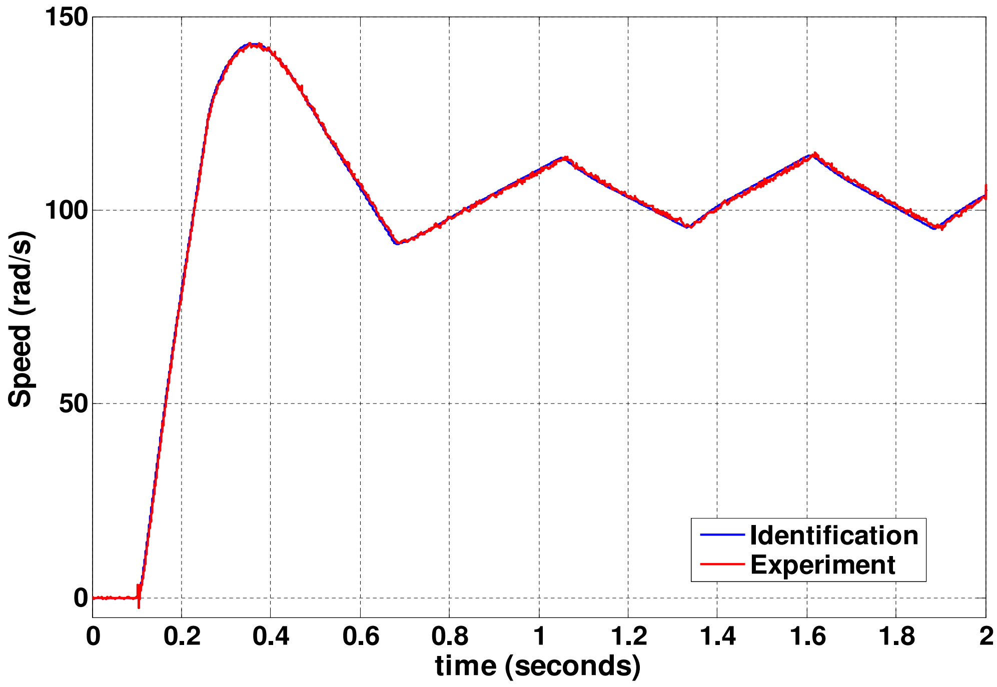
The feedback uncertainty model shown in figure 7.5 has been selected in order to account for the perturbations within the mechanical loads. It is necessary that the feedback uncertainty models for the stiff and flexible shaft mechanical loads investigated during the research project are defined in order to implement a robust control system for these mechanical plants. The first step to this effect is to define and identify a benchmark nominal model $G_0(s)$. This is represented as the centre point of the approximate uncertainty region in figure 7.1. The nominal models, $G_0(s)$ for the mechanical loads have been defined in section 7.4.1. As a result of having identified the nominal model, the next step would be to define the robust weighting function, $W_f(s)$, which is represented as the radius of the disc-like approximation of uncertainty in figure 7.1.

$$W_f(s) \geq \left| \frac{1}{G_{em}(s)} - \frac{1}{G_0(s)} \right| \tag{7.9}$$

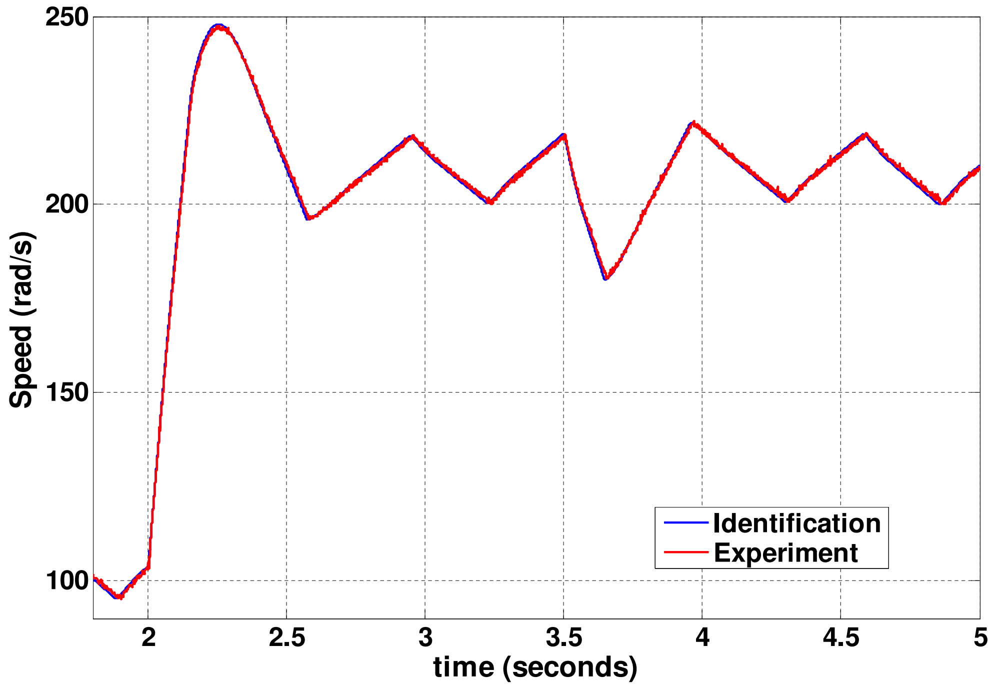
$$\begin{aligned} W_f(s) &= 0.0178(0.2494s + 1) \quad (7.10)\\ W_f(s) &= 0.0026(0.0013s + 1) \quad (7.11) \end{aligned}$$

The weighting function, $W_f(s)$ bounds the actual uncertainty of the plant. It essentially defines the upper bounds of the uncertainty within feedback perturbation model. The weighting function can be obtained by the equation 7.9, which is derived from the expression of the feedback uncertainty in (7.5). Equation 7.9 implies that in order to obtain the expression for the weighting function, $W_f(s)$, it is necessary to derive the:

1. inverse of the transfer functions of every plant within the uncertainty model
2. inverse of the transfer function of the identified plant nominal model

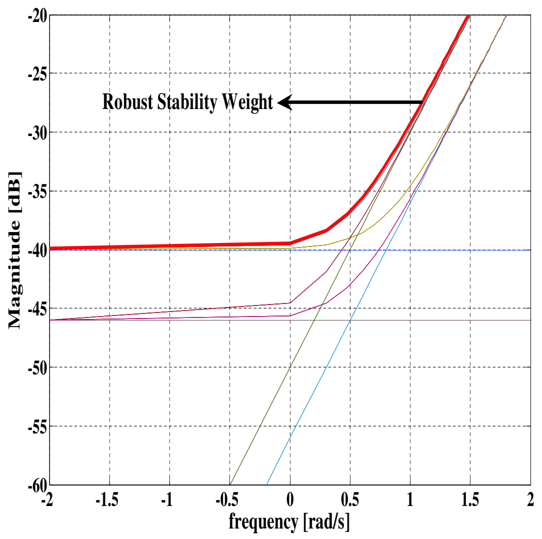


(a) Identification of flexible shaft nominal model: Speed transient to a step input 0 - 1000rpm

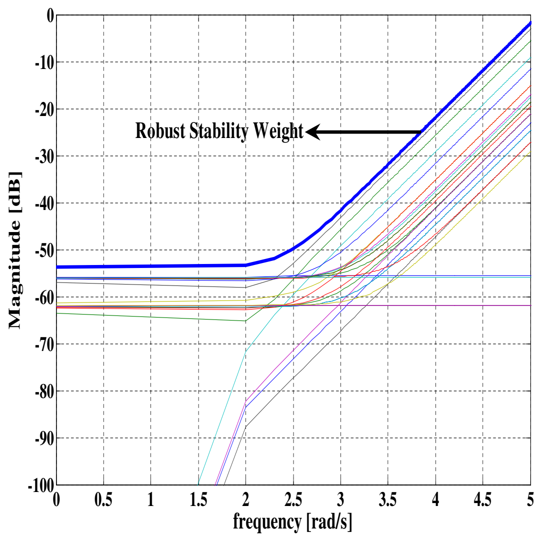


(b) Identification of flexible shaft nominal model: Speed transient to a step input of 1000 - 2000rpm and external disturbance of 0.15Nm introduced after 3.5s

Figure 7.9: Identification of flexible shaft nominal load



(a) Robust weighting for stiff-shaft load



(b) Robust weighting for flexible-shaft load

Figure 7.10: Robust weighting functions for Mechanical load

The weighting function, $W_f(s)$ is defined as the minimum magnitude plot that completely envelopes the largest magnitude plot of the difference between the two derived quantities. The equation that expresses the weighting function, $W_f(s)$ is in (7.9). The weighting functions calculated for the stiff and flexible shaft load are given in (7.10) and (7.11). The corresponding magnitude plots for the stiff and flexible mechanical plants are shown in figures 7.10(a) and 7.10(b). It can be observed that the magnitudes of all the possible transfer functions, which is defined by (7.9), are plotted on the same axes as the highlighted weighting functions, $W_f(s)$.

7.4.3 Nominal Stability

This is the first condition that must be satisfied along the design process for the implementation of a robust control system. The stability analysis focuses on the control system with the nominal model with no uncertainty model analysis performed at this stage. The generic closed-loop system represented by figure 7.11 is internally stable if all internal signals decay to zero after the external signals w, d_1 d_2 and n have vanished.

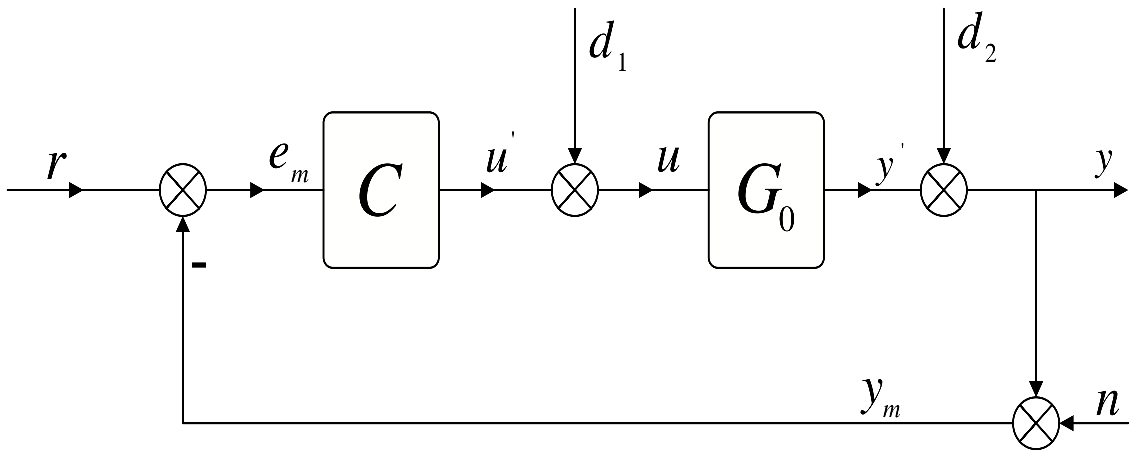


Figure 7.11: Generic closed-loop control system [78]

The standard feedback loop is internally stable, if and only if [78]:

- The poles of the closed-loop system are located in the left-half plane
- There is no right-hand plane pole-zero cancellation in forming CG_0

For any control system randomly selected during the optimisation process, to evaluate the internal stability of the nominal model, it is necessary to initially identify the

C	Controller	G_0	nominal model
y	controlled output	d_1	(plant) input external signal
y_m	measured controlled output	d_2	(plant) output disturbance
u	actuating signal	y'	plant output
e	tracking error ($e = r$ - y)	u'	controller output
e_m	measured tracking error ($e_m = r$ - y_m)	u	control signal
r	reference input	n	sensor noise

Table 7.1: Generic closed-loop control system parameters

nominal system parameters. Having identified this, its nominal stability is evaluated based on the conditions for internal stability for the standard feedback loop.

7.4.4 Nominal Performance

Having verified the nominal internal stability of the control system under consideration, its nominal performance is now evaluated. The control system is said to have nominal performance if it satisfies certain conditions. For an internally stable closed loop system, the nominal performance condition is given by (7.12), where W_t represents the weighting on the complementary sensitivity function, T defined in (7.13), describes the relationship between the controlled output y and the difference between the reference input r and the sensor noise, n.

The nominal performance condition, defined in (7.12), simply states that there are specifications that the designed system must adhere to. These specifications are defined by the weight, W_t. If W_t multiplies T, which is defined by (7.13), the resulting function will have a magnitude lower than 'one' at all frequencies only if T has been been designed such that it meets the performance specification. If this condition is not verified, the designer will understand that the control system does not adhere to the design constraints with the nominal model. Ideally, the value of T at every frequency should be 'zero': this implies that the controlled output would be unaffected by the noise inputs to the systems [88], [89].

The nominal performance condition can also be given by equation (7.14), where W_s represents the weighting on the sensitivity function, S. The sensitivity function, defined in (7.15), describes the effects of the output disturbance on the controlled output, y. The sensitivity function also describes the effects of variations in the external inputs, r and d_2 on the tracking error, e. To achieve a good disturbance rejection with the designed control system, the value of the sensitivity function at every frequency must be as small as possible; ideally, it should be 'zero', which implies its complement, T must be 'one' [88], [89]. This implies there exist conflict in achieving the design constraint for the Sensitivity, S and its Complement, T.

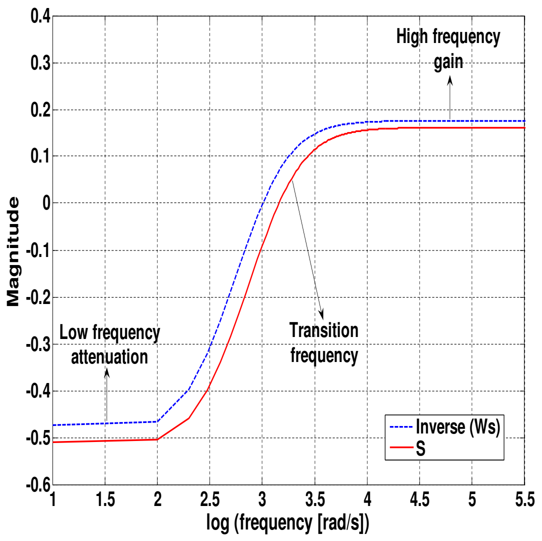
The demands of the sensitivity function, S and its complement, T are conflicting and as such, to ensure good robust control design, there must be some compromise defining the sensitivity function, S, which specifies the system's sensitivity reduction and disturbance response and its complement, T, which defines the plant uncertainty, the response and measurement noise of the system. Typically, disturbance rejection and sensitivity reduction are desired over a lower frequency range, while plant modelling errors and measurement noise generally occur at higher frequencies. For this reason, it is possible to achieve a compromise by defining a suitable performance criteria, in terms of weights, on the Sensitivity, S function and its complement, T to facilitate good control design [88].

$$\|W_tT\|_\infty < 1 \tag{7.12}$$
$$T = \frac{y}{r-n} = \frac{CG_0}{1+CG_0} \tag{7.13}$$
$$\|W_sS\|_\infty < 1 \tag{7.14}$$
$$S = \frac{e}{r-d_2} = \frac{y}{d_2} = \frac{1}{1+CG_0} \tag{7.15}$$

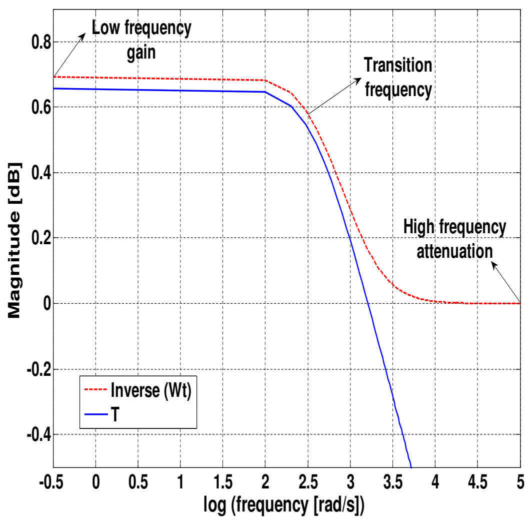
The weighting on the sensitivity function, W_s specifies the performance in relation to disturbance rejection (in terms of magnitude at different frequencies) required the of the control system. W_s can be defined considering three design goals that concern the steady state error, the stability margin and the transition frequency, which is the frequency at which disturbances are amplified instead of attenuated for the sensitivity function (but represents the bandwidth, frequency of attenuation, for the weight, W_s). A general formula which represents W_s is given in (7.16) and its bode plot can be represented as in figure 7.12(a). It dictates that the maximum steady state error can be limited to M_S, the high frequency disturbance amplification can be limited to A_S and the transition frequency, ω_t can be limited to ω_S [78].

$$W_s = \frac{s+\omega_S A_S}{A_S(s+\omega_S M_S)} \tag{7.16}$$
$$W_t = \frac{s+\omega_T A_T}{M_T(s+\omega_T M_T)} = \frac{s+360}{1.5(s+1800)} \tag{7.17}$$

In a similar vein, the weighting, W_t is the performance specification on the Complementary sensitivity function, T but in relation to noise rejection. The design approach adopted within the research project employed this performance specification. The general form for the weighting is given as (7.17) and its bode plot for the formula is in figure 7.12(b).



(a) Typical Performance Weight on S



(b) Typical Performance Weight on T

Figure 7.12: Robust weighting functions for Mechanical load

Three design goals are defined within (7.17): (1) The maximum limit of the noise signal amplitude at low frequencies can be limited to M_T; this value was chosen as 1.5 based on general design specification in [44]. (2) The high-frequency noise attenuation can be limited to A_T; this was selected as 0.1 based upon general design specification in [44], [78]. (3) The transition frequency (bandwidth) of the complementary sensitivity function, T, which is the frequency at which noise is attenuated, can be limited to ω_T. Its value was selected as approximately a 1200 rad/s, which is the bandwidth of the active-low pass filter designed for the signal conditioning of the tachogenerator signals within the experimental system in section 4.4.5. Within the weighting function W_t, the transition frequency, ω_T represents the frequency at which noise signals are amplified rather than attenuated. Given the random distribution of solutions, redefining the boundaries for the sensitivity or complementary sensitivity function does not guarantee an increase or decrease in the number of viable possible solutions to the control optimisation problem.

7.4.5 Robust Stability

A necessary criterion for a robust control system is that it is stable in the face of perturbation. This implies that the controller designed must have a stable closed loop system response for all the possible plant perturbations determined by the foreseen perturbations to the system. In the theoretical design, these different plant variations need to be considered within a single suitable uncertainty plant model.

$$\|W_f G_0 S\|_\infty < 1 \tag{7.18}$$

A feedback uncertainty model has been selected to define the plant uncertainty in section 7.4.2. The robust stability condition for the feedback uncertainty model is stated in equation (7.18). This equation is somewhat similar to the nominal performance condition in (7.12) given it defines a weight, $W_f G_0$ on the sensitivity function. This weight specifies the performance criteria on the sensitivity function across the perturbations within the plant uncertainty model. Each control system, randomly selected during the automated optimisation process, is evaluated based on this condition. Table 7.2 shows the robust stability conditions for each of the other perturbation models defined in section 7.3.1.

7.4.6 Robust Performance

The next phase in the design deals with the evaluation for robust performance. The criteria for robust performance combines the conditions of nominal performance and robust stability for the plant perturbation model. Given the conditions for robust stability considers the Sensitivity function, S and those for the nominal performance considers its complement T, the resulting control system, designed to respect the combination of these two conditions, will provide suitable trade-offs for the ideally conflicting system performance indicators.

Within the algorithm, only those controllers that meet the conditions for robust performance are further tested to evaluate their actual dynamic response under certain load conditions. The nominal performance condition for an internally stable closed-loop system with no uncertainty in its nominal model, G_0 is given in equation (7.12). If the nominal model, G_0 is perturbed with feedback uncertainty as in equation (7.19), the nominal performance condition can be represented as (7.22). The robust performance condition is then a combination of the nominal performance and the robust stability conditions of the perturbed plant.

$$G_0 \rightarrow \frac{G_0}{1+\Delta W_f G_0} \tag{7.19}$$

$$\left\|W_t\tilde{T}\right\|_\infty < 1 \tag{7.20}$$

$$\begin{aligned}
\tilde{T} &= \frac{\dfrac{G_0 \times C}{(1+\Delta W_f G_0)}}{1+\dfrac{G_0 \times C}{(1+\Delta W_f G_0)}} \\
&= \frac{G_0 C}{1+G_0 C+\Delta W_f G_0} \\
&= \frac{\dfrac{G_0 C}{(1+G_0 C)}}{1+\dfrac{\Delta W_f G_0}{(1+G_0 C)}}
\end{aligned}$$

$$\tilde{T} = \frac{T}{1+\Delta W_f G_0 S} \tag{7.21}$$

$$\left\|W_t\tilde{T}\right\|_\infty = \left\|\frac{W_t T}{1+\Delta W_f G_0 S}\right\|_\infty < 1 \tag{7.22}$$

A sufficient condition for robust performance of the perturbed system is given in (7.23).

$$\|W_f G_0 S\|_\infty < 1 \quad \textit{and} \quad \left\|\frac{W_t T}{1+\Delta W_f G_0 S}\right\|_\infty < 1 \tag{7.23}$$

From equation (7.23), *a necessary and sufficient condition* for robust performance for the perturbed plant with feedback uncertainty is given in equation (7.24)

$$\left\| |W_t T| + |W_f G_0 S| \right\|_\infty \ < \ 1 \tag{7.24}$$

Table 7.2 shows the conditions for nominal performance, robust stability and robust performance for all the uncertainty models described in section 7.3.1.

7.5 Automated Theoretical Optimisation

The automated identification-based approach to designing robust control systems builds on some of the advantages of the experimental optimisation process described in chapter 6. During its implementation, the objective functions, adopted by the investigated evolutionary algorithms, ensure that each randomly selected controller (structure and parameters) satisfies the conditions for robust performance. In a similar fashion to the experimental approach, each of the three investigated algorithms adopt a similar controller solution evaluation stage. The unique feature of the robust identification-based control design approach lies in the evaluation stage (fitness function), which is critical for theoretically determining robust performance.

The purpose of the evaluation stage is to quantify the quality of each randomly selected controller. These values are used by the evolutionary algorithms to determine which controller gives the best performance. The evaluation checks that each tested controller satisfies the four conditions necessary for robust control system design:

Perturbation	*NPC*	*RSC*	*RPC*
$(G_0 + W_a\Delta)$	$\|W_s S\|_\infty < 1$	$\|W_a CS\|_\infty < 1$	$\| \|W_s S\| + \|W_a CS\| \|_\infty < 1$
$G_0\,(1 + \Delta W_m)$	$\|W_s S\|_\infty < 1$	$\|W_m T\|_\infty < 1$	$\| \|W_s S\| + \|W_m T\| \|_\infty < 1$
$G_0/\,(1 + \Delta W_f)$	$\|W_t T\|_\infty < 1$	$\|W_f S\|_\infty < 1$	$\| \|W_t T\| + \|W_f S\| \|_\infty < 1$
$G_0/\,(1 + \Delta W_f G_0)$	$\|W_t T\|_\infty < 1$	$\|W_f G_0 S\|_\infty < 1$	$\| \|W_t T\| + \|W_f G_0 S\| \|_\infty < 1$

Table 7.2: Necessary and Sufficient Conditions for Robust Performance: NPC - Nominal Performance Condition, RSC - Robust Stability Condition, RPC - Robust Performance Condition

the nominal stability, nominal performance, robust stability and robust performance. The commented MATLAB m-file code for the implementation of the fitness function of the Robust identification-based control design method is in the attached CD-ROM Appendix.

Nominal stability: Incorporated within the objective functions of each investigated evolutionary algorithm is a verification strategy that tests if each randomly selected controller will achieve nominal stability with the mechanical loads being considered. This is achieved, within the digital z-domain, by ensuring:

1. Poles of the closed-loop system lie within the unit circle: the magnitude of each closed-loop pole and zero must be less than or equal to 'one'.
2. There is no pole-zero cancellation outside the unit circle in forming CG_0.

This is achieved by ensuring that all the poles and zeros of the controller, C lie within the unit circle. This, in turn, can be simply done by bounding the magnitudes of each pole and zero to 'one'. Only those controllers that satisfy the nominal stability condition are further tested against the nominal performance criteria.

Nominal performance: The nominal performance condition is specified in equation (7.17). Those randomly selected controllers, which do meet the user-specified demands for nominal performance, are excluded from further testing.

Robust stability: The feedback perturbation model has been adopted for modelling the uncertainty within the mechanical loads. Each randomly selected controller is evaluated on its robust stability based on the condition given in the equation (7.18). As in the previous evaluation stage, only those controllers that satisfy the condition for robust stability are further tested.

Robust performance: This is, in a sense, the ultimate test for a truly robust control system. The condition for robust performance combines both the conditions for nominal performance and robust stability. This is represented in equation (7.24).

The controllers that meet the robust performance are then subject to further testing on the quality of their performance.

Evaluation of robust control systems: For those controllers that perform robustly, its performance to the nominal model for the stiff shaft and flexible shaft mechanical load (with backlash), for different speed transients and in the presence of load disturbance, is evaluated using the simulation model. The quality of the performance is quantified by evaluating the Integration of the Absolute Error (IAE) between the reference speed and the speed response produced by the robust closed-loop control system. There is a penalty factor incorporated for those controlled responses that exceed an 8 % overshoot. The controllers are then ranked according to their fitness values - the controllers with lower fitness values are ranked higher and vice versa. The next stage, after suitable ranking the individual solutions according to their fitness values is reproduction, in a bid to generate better solutions

The process of the generation of possible solutions and their subsequent evaluation using the described fitness function is repeated during the implementation of the investigated optimisation algorithms until a termination criterion, as described in chapter 2, has been reached. At this point, the algorithms should have converged to the best solution, which is a closed-loop system that provides robust performance in the presence of system uncertainties and also has minimum overshoot on it speed responses under the different load conditions. Figure 7.13 summarises the evaluation process of the identification-based approach to robust control system design.

An important advantage of the method proposed in this chapter over its experimental counterpart is that given the optimisation is performed on a simulation model of the actual system, the actual plant experiences little or no stress during the process, thus prolonging its functioning lifespan. Another important advantage is the length of time required for the process. It can be much reduced when compared with the experimental process for a similar optimisation problem described in section 6.3. The reason is the time duration largely depends on the power of the computer processor used in implementing the optimisation process. in this case, the time taken to determine and

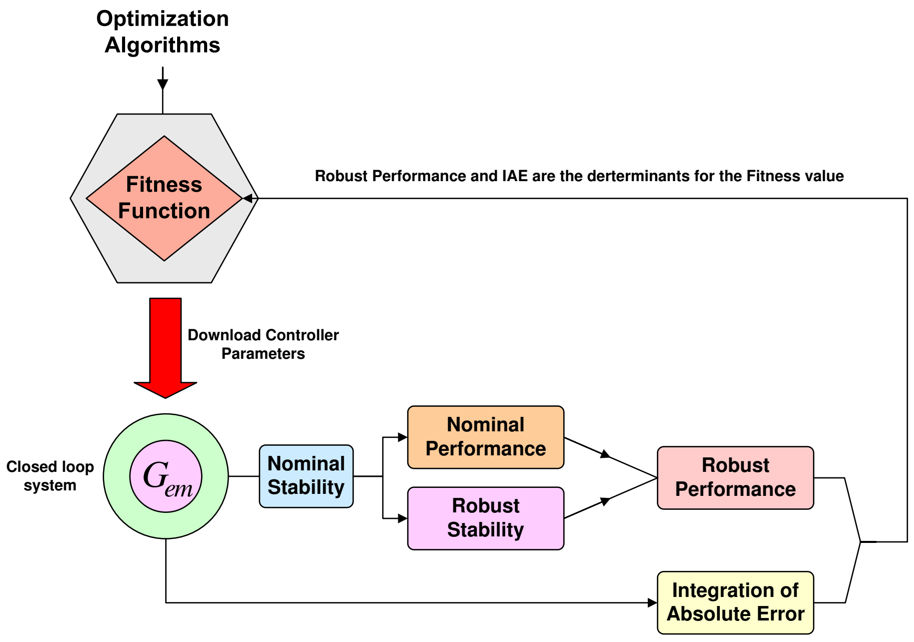


Figure 7.13: Summary of theoretical optimisation evaluation process

evaluate the robust performance of any one randomly selected controller is approximately 300 ms. A possible disadvantage is that the automated identification-based approach process requires that the mechanical plant for which the control system is to be designed must be modelled. However, once a structure of the system model is selected, its parameters are automatically identified by the GA procedure, reducing greatly the modelling efforts and enormously increasing the accuracy. Once a suitable model has been automatically designed, the robust control system design process is relatively straightforward; this is largely due to the use of the evolutionary algorithms in the design procedure.

7.6 Results

The information provided by the results obtained in this section, through the adoption of the evolutionary algorithms and the identification-based control design method, can be summarised as follows: the similarities between these sets of results and those in section 6.4, obtained via the robust experimental design method, provides validation for both these approaches. It also demonstrates that the identification-based method is an effective option for the design of robust control systems. Taking another perspective, the similarities between the results obtained, through the combined adoption of the identification-based control design method and the evolutionary algorithm, is a testament to the effectiveness of the algorithms employed. Also, given the similarities in the results obtained by the different algorithms, there is more confidence that the obtained solutions are truly global and the corresponding speed responses they provide are the best possible. Given the undeniable similarities between the results obtained, it will be useful and sufficient to explain the general speed response profile, obtained through the adoption of the proposed approach, for the stiff and flexible shaft mechanical loads. The dynamic responses obtained will then be characterised.

7.6.1 Stiff-shaft load speed response

Robust control systems for the stiff shaft load have been designed using each of the three evolutionary algorithms. In the subsequent paragraphs, a comparison between the speed responses produced by the robust control systems designed using each of the three algorithms will be provided. To achieve this, the resulting mechanical load speed response produced will be characterised according to its rise time, settling time, overshoot, disturbance rejection time and steady state output.

Rise time: During the mechanical load's transition from initial speed to final speed, its response is largely governed by its inertia: the larger the inertia, the slower the initial acceleration (rate of speed change). This feature is observed across the speed responses for stiff-shaft load with different inertias. The responses of the GA designed (fourth order) controller in figures 7.14, 7.15 and 7.16 have respective rise times of 80.4 ms, 74.4 ms and 86.2 ms. The BF results shown in figures 7.20, 7.21 and 7.22 with inertias J, $3J$ and $5J$ have rise times of 104 ms, 99 ms and 90 ms respectively. The speed responses obtained via the HBF optimisation algorithm in figures 7.26, 7.27 and 7.28 have rise times of 86 ms, 80ms and 87 ms respectively.

Settling time: The settling time is a criteria used to evaluate how quickly the different responses settle to the final value. The speed responses, obtained using the GA designed fourth order controller, in figures 7.14, 7.15 and 7.16 achieve a settling time within two percent around the final value of 148 ms, 112 ms and 115 ms respectively. The BF optimised responses in figures 7.20, 7.21 and 7.22 have settling times of 191 ms, 147 ms and 122 ms respectively. Also, the HBF speed responses in figures 7.26, 7.27 and 7.28 produce similar settling times as the other two algorithms of 153 ms, 126 ms and 119 ms.

Overshoot: During the control design procedure, the aim was to keep the maximum overshoot to about eight-percent of the final value. The outcome of the restriction has been successfully enforced, given the results obtained from the optimisation. The maximum overshoot obtained with each algorithm occurred when the load had its largest inertia of $5J$; this is due to the output of the controllers being in saturation for longer as it produces the required torque demand to drive the heavier mechanical load to the desired speed. The maximum overshoot obtained with the GA designed controller is 2.7%. With the BF optimised control system, there exist an overshoot of 1.7%. Finally for the controller, designed via the HBF algorithm, there exists a maximum overshoot of 1.68%.

Disturbance Rejection: This is a measure of how quickly the designed control systems reject the load torque disturbance of 0.15 Nm introduced imposed on the system after 1050 ms. The robust control systems reject disturbance in similar time periods, largely appearing to be independent of the different mechanical loads they are controlling. The GA robust control system rejects disturbance in 210 ms, the BF optimised control system rejects disturbance in 150 ms while the HBF algorithm also achieves disturbance rejection in 150 ms.

Steady State: For the GA controller responses in figures 7.14, which has the largest value of friction, the steady state response is the noisiest with maximum oscillation around the steady state value of 1.27%. In a similar manner, the BF optimisation algorithm produces the noisiest signal in figure 7.15, with maximum steady-state oscillations of 1.83%. The HBF control system response also follow the same trends producing the noisiest response in figure 7.16 for the system with the largest friction value; its maximum steady-state oscillation is 1.55%.

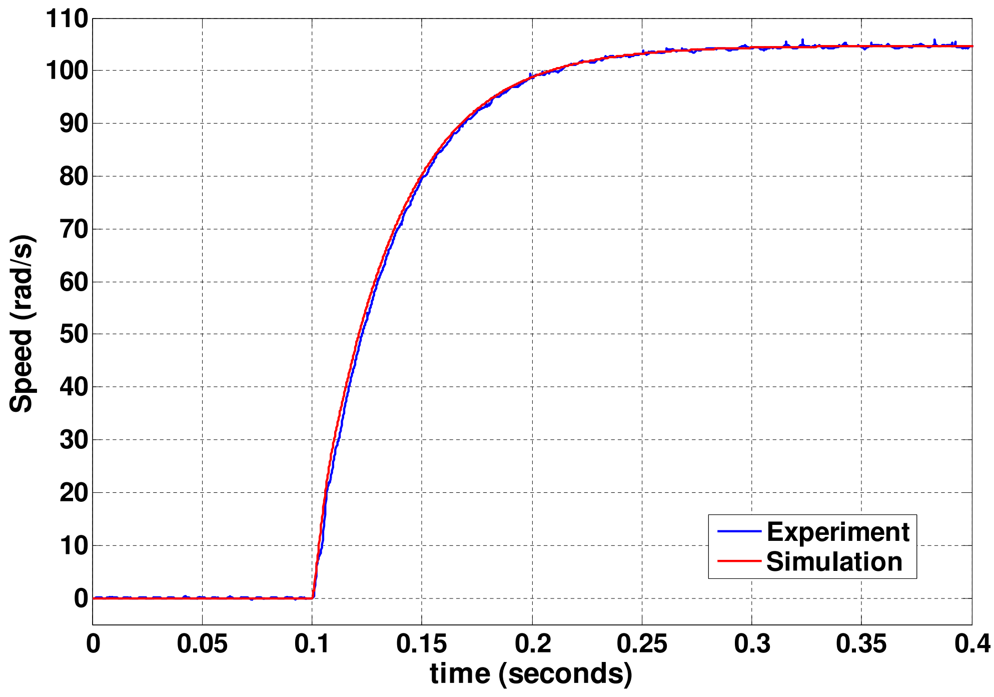
7.6.2 Flexible-shaft load speed response

The resulting mechanical load speed responses produced can be suitably characterised according to its rise time, settling time and steady state output ripple. The speed transients produced are in response to step inputs of 0 - 1000 rpm between times of 0.1s and 0.6s and step inputs from 1000 - 2000 rpm between time periods of 0.6s and 1.5s. An external load disturbance of 0.15Nm is applied after 1.05s.

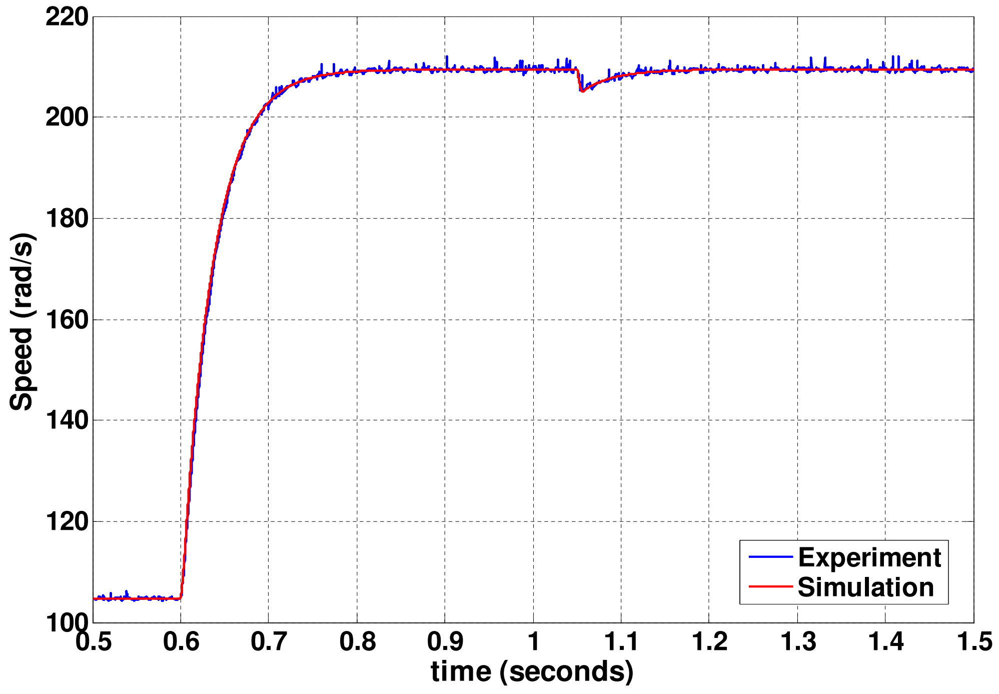
Rise time: The results of the GA optimisation has produced the following responses in figure 7.17, 7.18 and 7.19 with rise times of 122 ms, 106 ms and 101 ms respectively. The speed responses of the BF optimised control system in figure 7.23, 7.24 and 7.25 have the rise times of 123 ms, 107 ms and 101 ms respectively. Also the rise times of the speed responses for the HBF optimised control system in figures 7.29, 7.30 and 7.31 are 116 ms, 102.4 ms and 98.4 ms respectively.

Settling time: The 2% settling times achieved by the speed responses of the GA optimised control system in figure 7.17, 7.18 and 7.19 are 258 ms, 232 ms and 185 ms. For the BF optimised speed responses in figure 7.23, 7.24 and 7.25, the settling times achieved are 261 ms, 238 ms and 148 ms. The settling times of the speed responses obtained using the HBF optimised controller are 234, 231 and 204 ms.

Steady-state output: On the application of torque disturbance, for the the mechanical load with inertia of $5J_L$ and damping of D, the resulting torque oscillations have different magnitudes . The GA control system produce torque oscillations with a magnitude of 5.9 rad/s around the final speed. The BF optimised system has torque oscillations of magnitude 5.76 rad/s. Finally, the speed response of the HBF control system produces these torque oscillations with a magnitude of 4.6 rad/s.

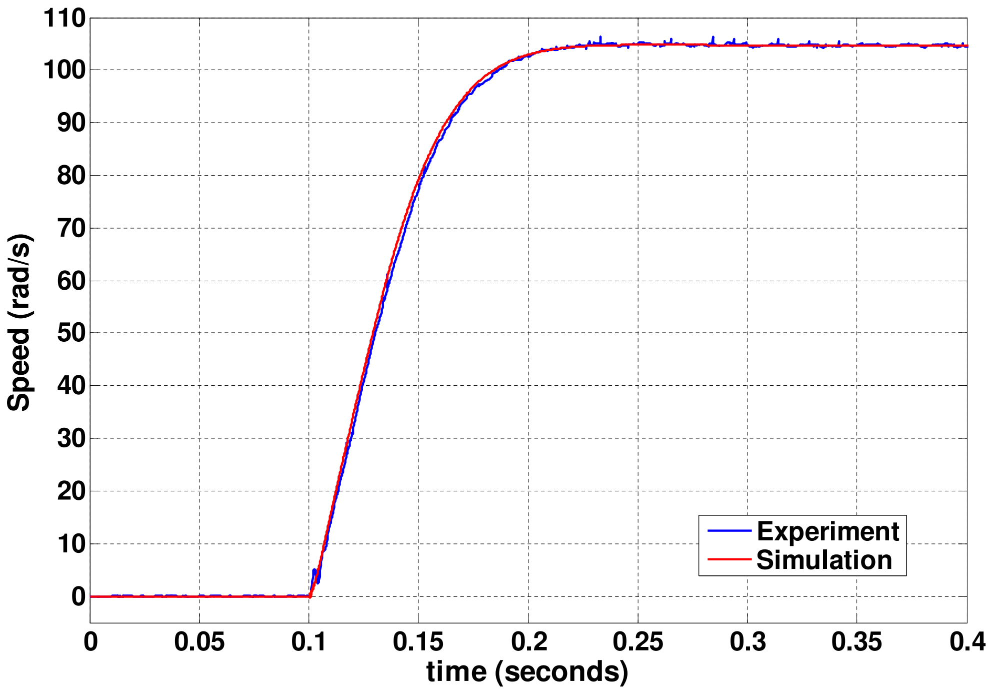


(a) Controller response to step demand of 0 to 1000 rpm

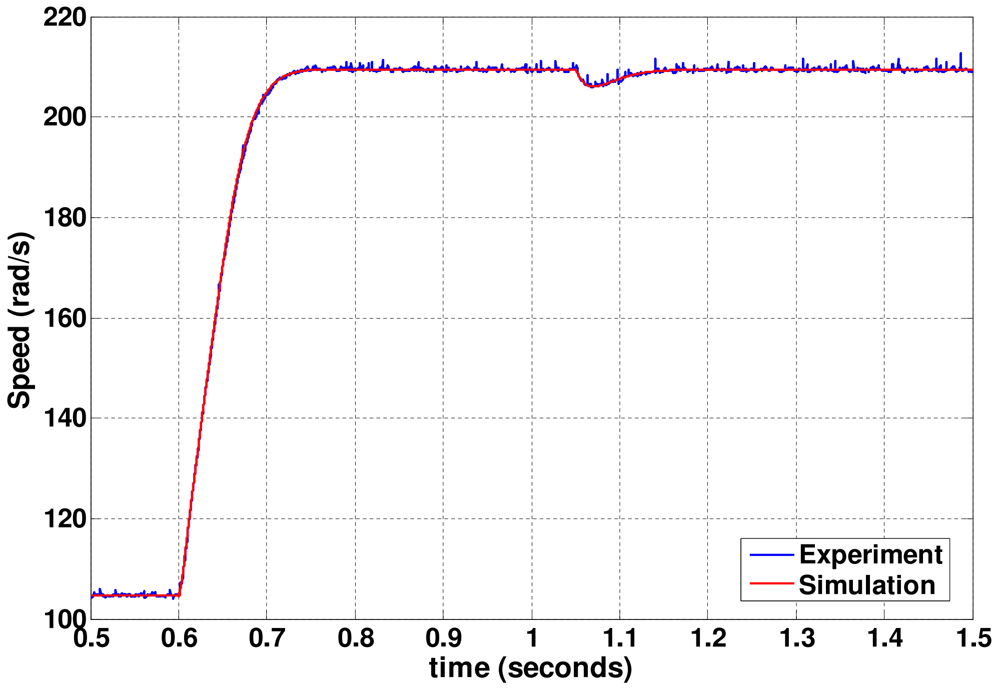


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.14: GA Fourth order Controller Response for Inertia J and Friction $5B$

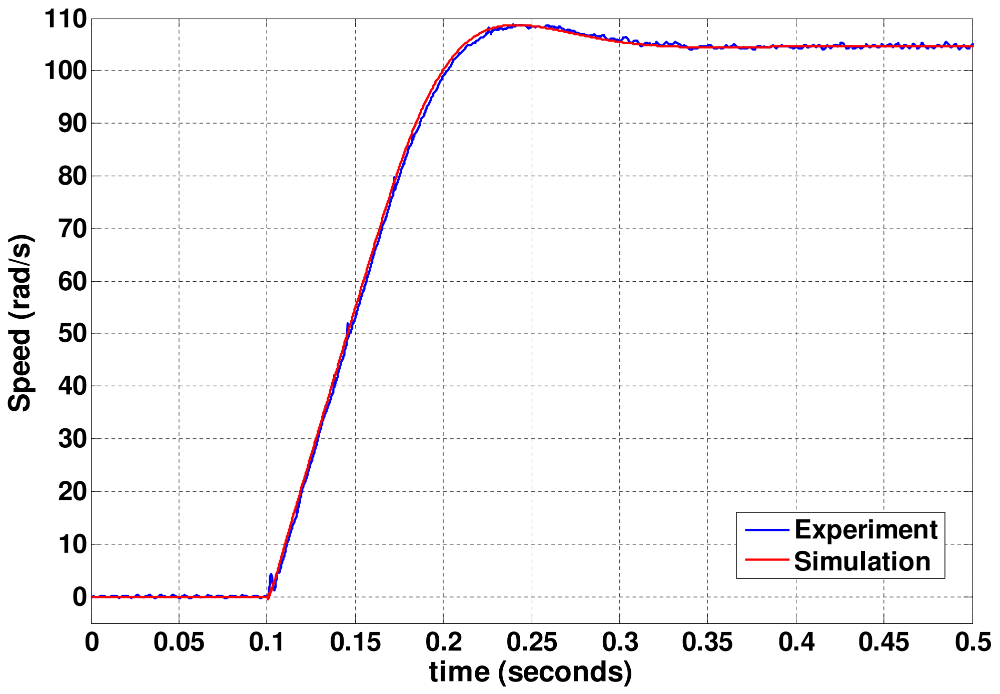


(a) Controller response to step demand of 0 to 1000 rpm

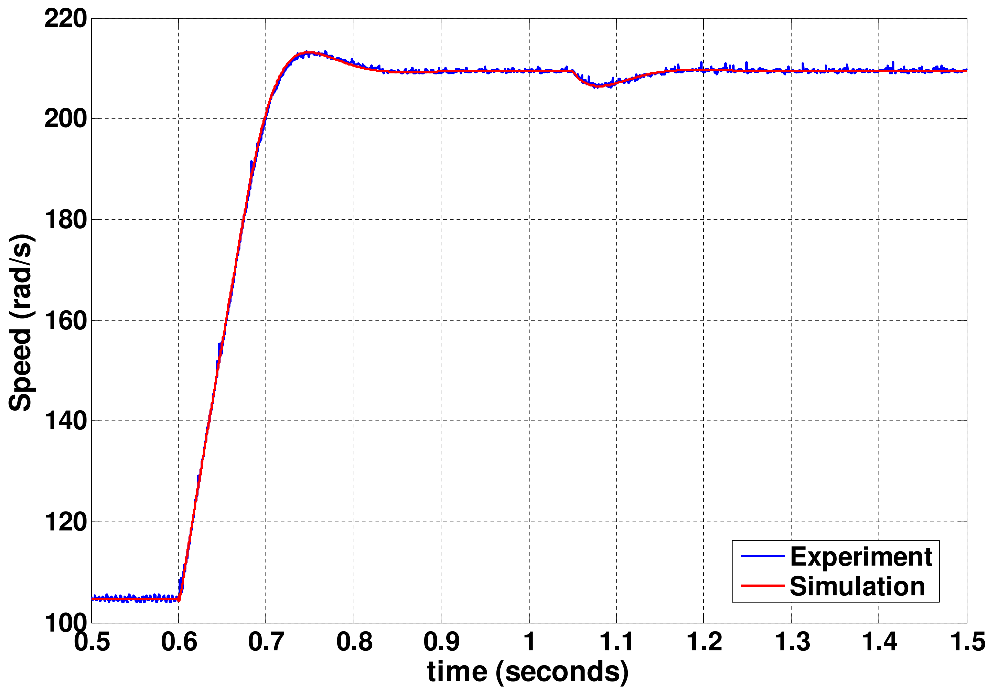


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.15: GA Fourth order Controller Response for Inertia $3J$ and Friction $3B$

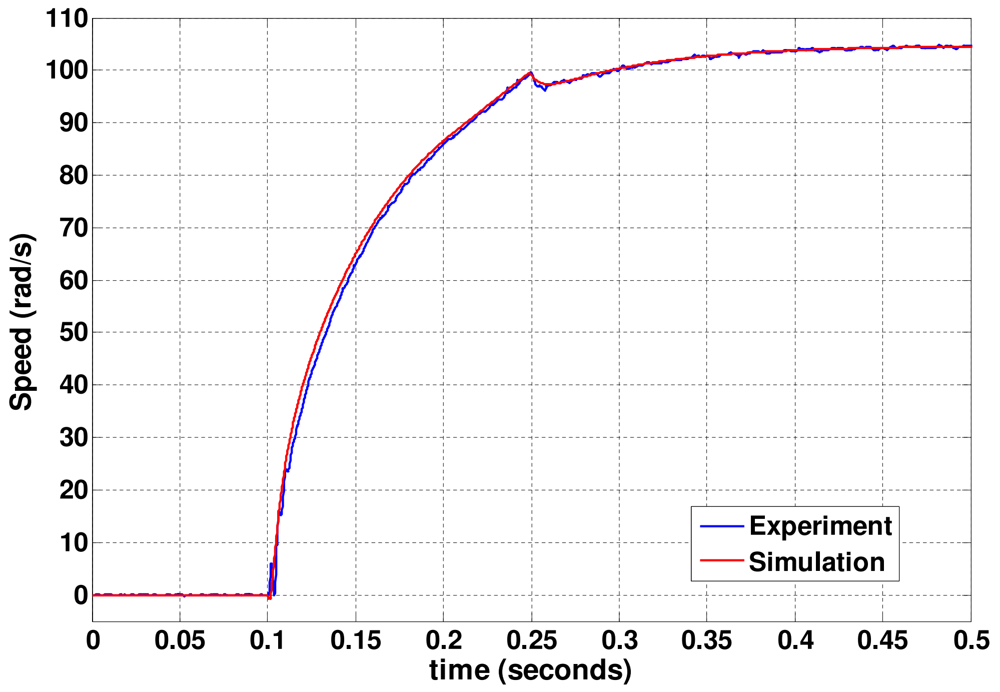


(a) Controller response to step demand of 0 to 1000 rpm

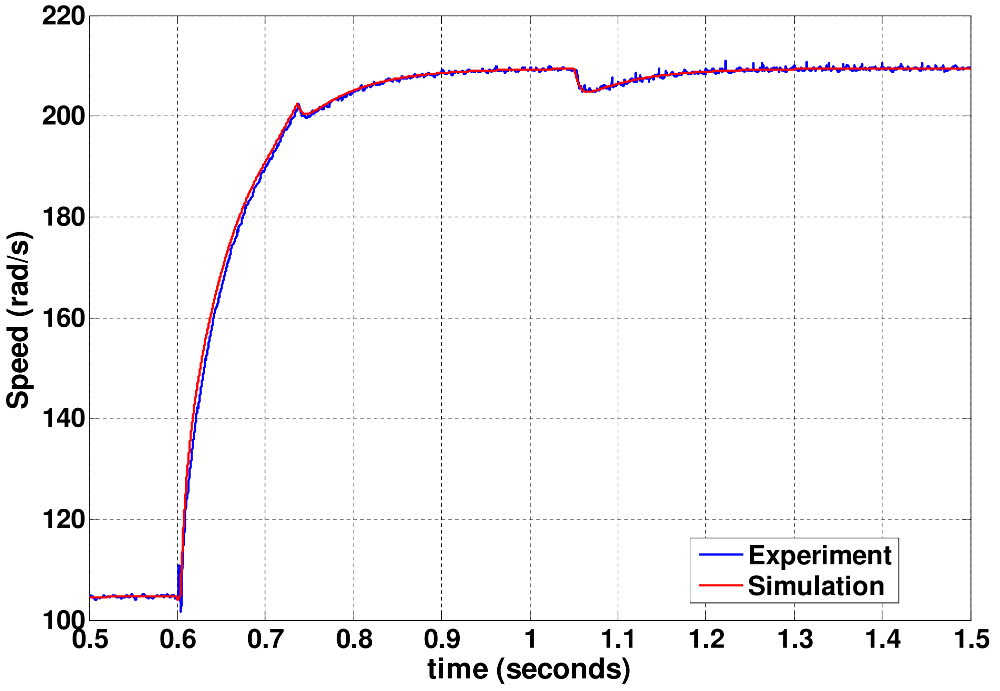


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.16: GA Fourth order Controller Response for Inertia $5J$ and Friction B

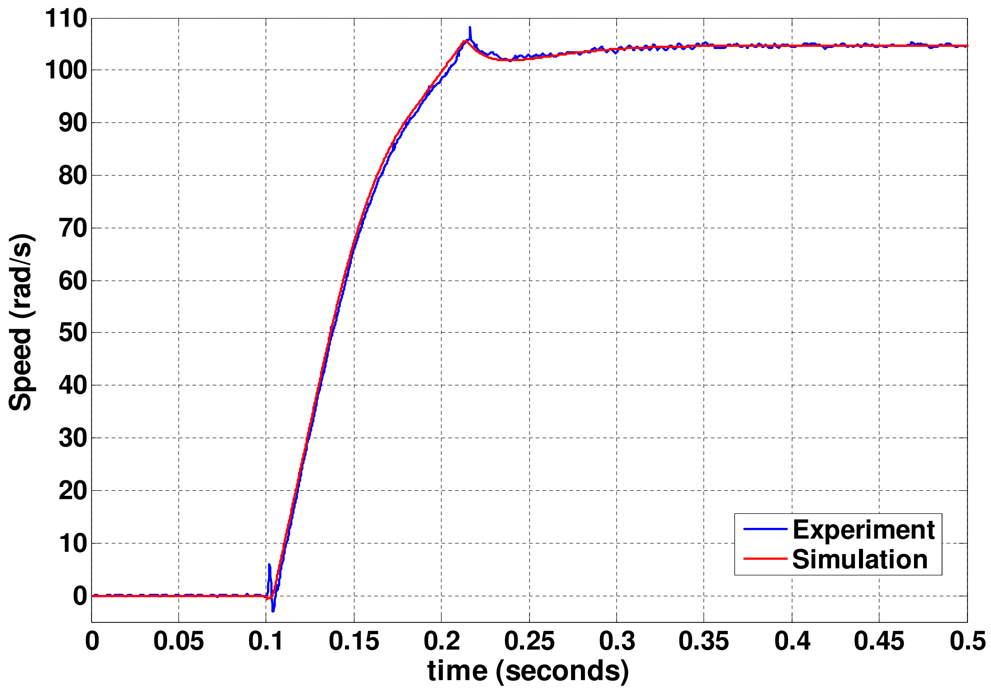


(a) Controller response to step demand of 0 to 1000 rpm

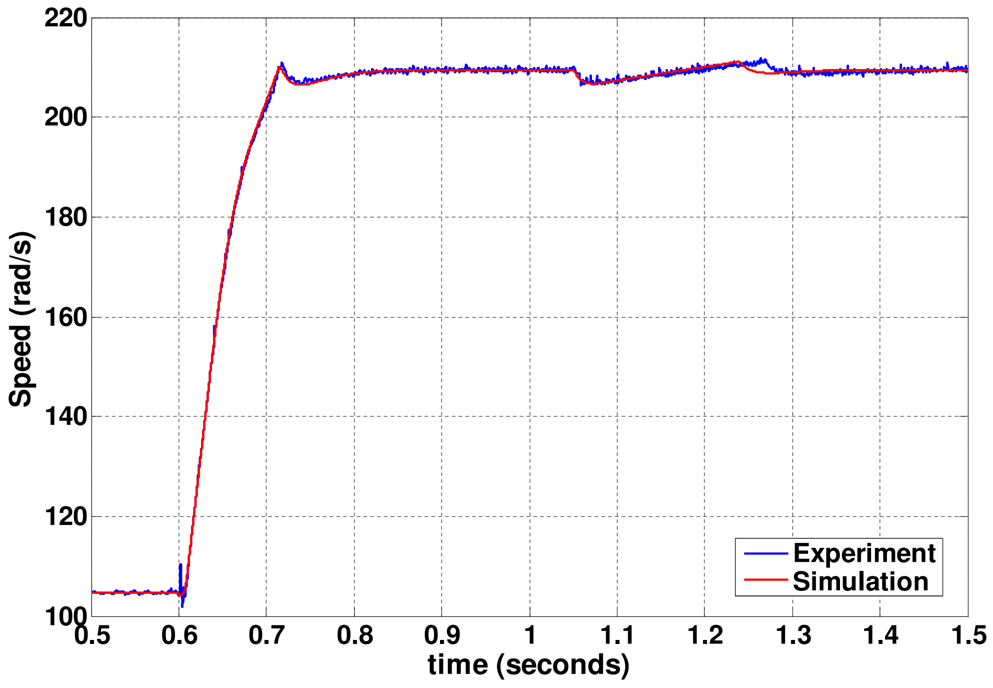


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.17: GA Fourth order Controller Response for Inertia J and Damping $5D$

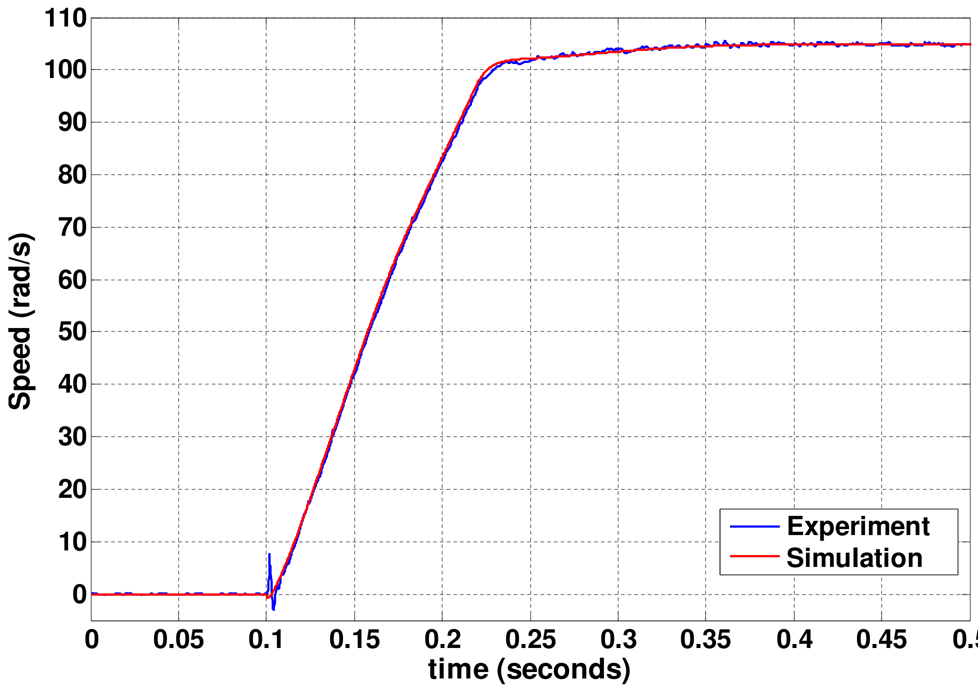


(a) Controller response to step demand of 0 to 1000 rpm

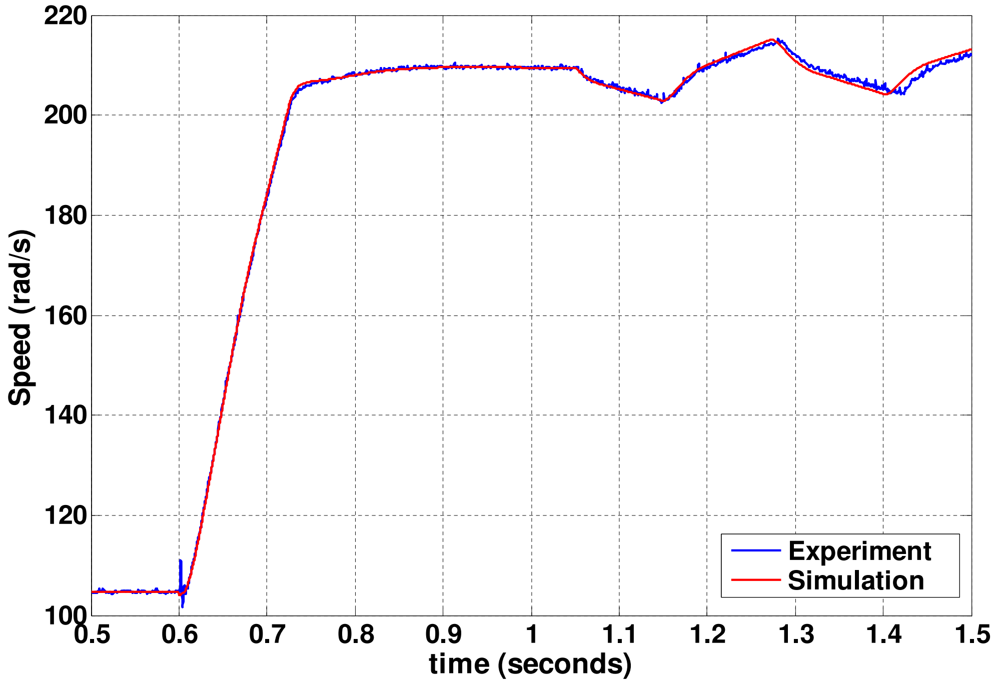


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.18: GA Fourth order Controller Response for Inertia $3J$ and Damping $3D$

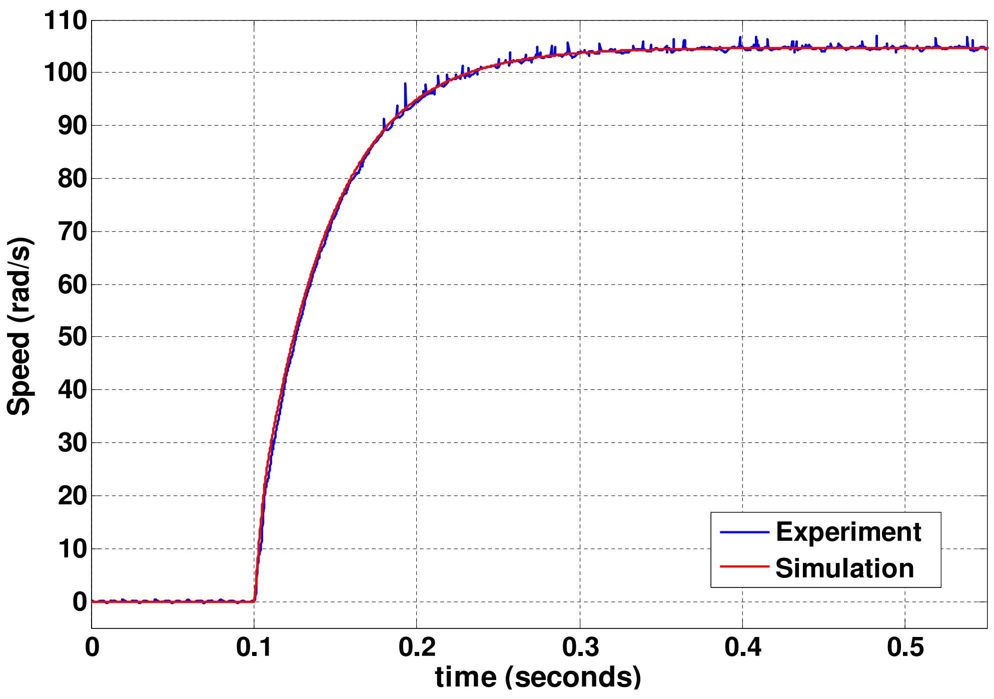


(a) Controller response to step demand of 0 to 1000 rpm

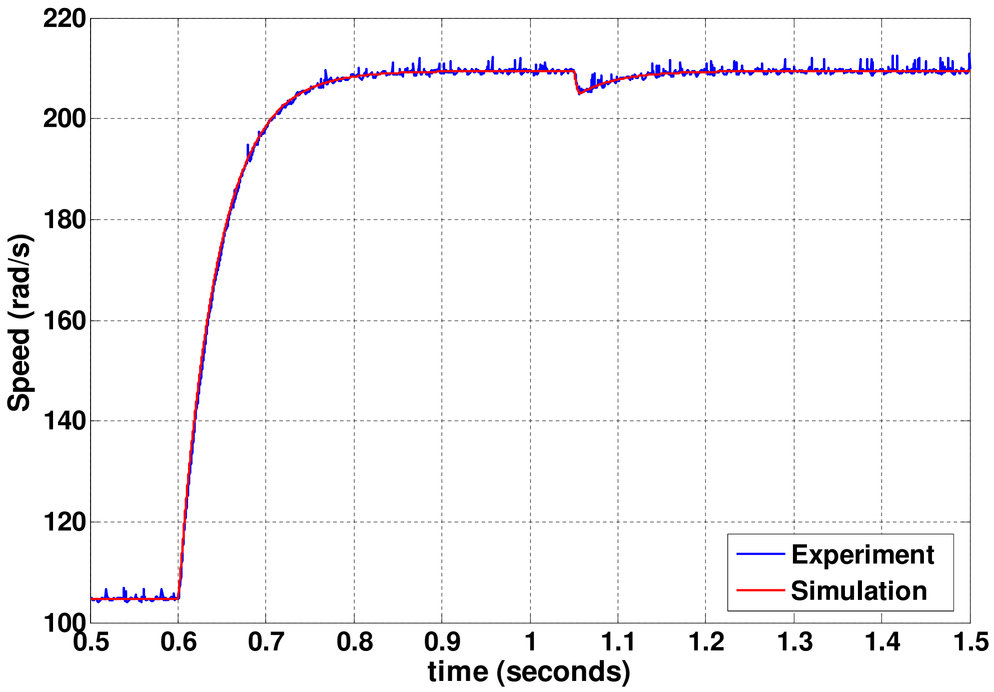


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.19: GA Fourth order Controller Response for Inertia $5J$ and Damping D

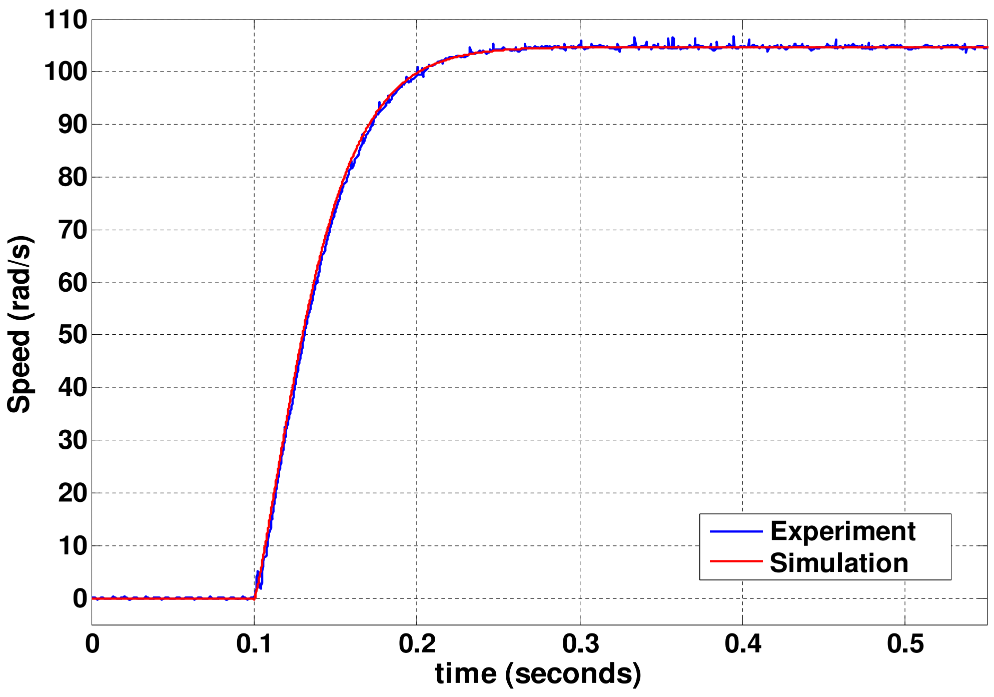


(a) Controller response to step demand of 0 to 1000 rpm

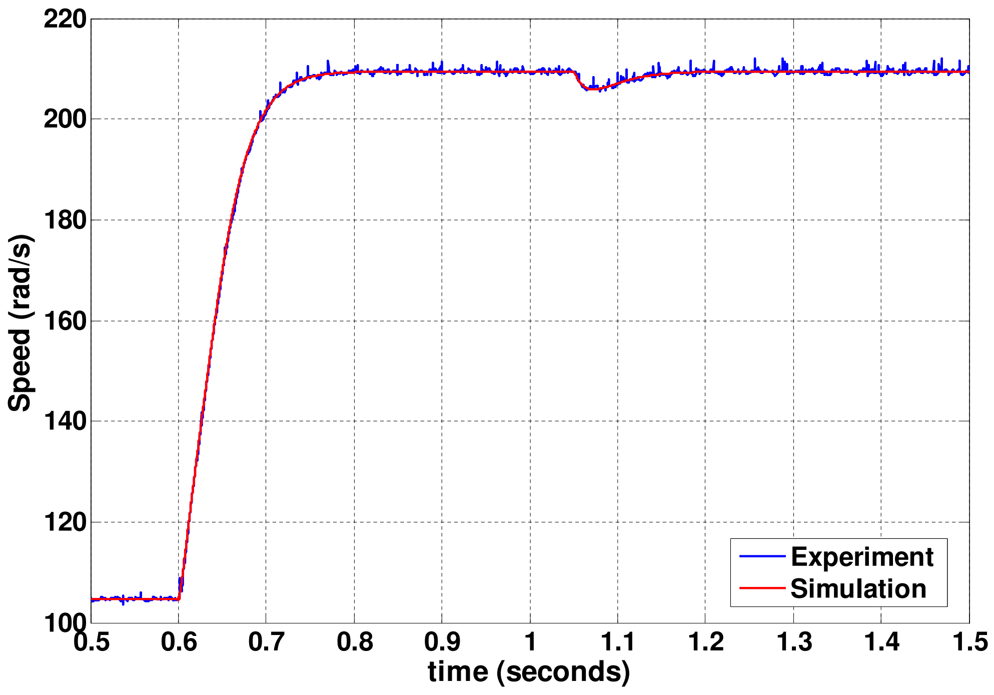


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.20: BF Fourth order Controller Response for Inertia J and Friction $5B$

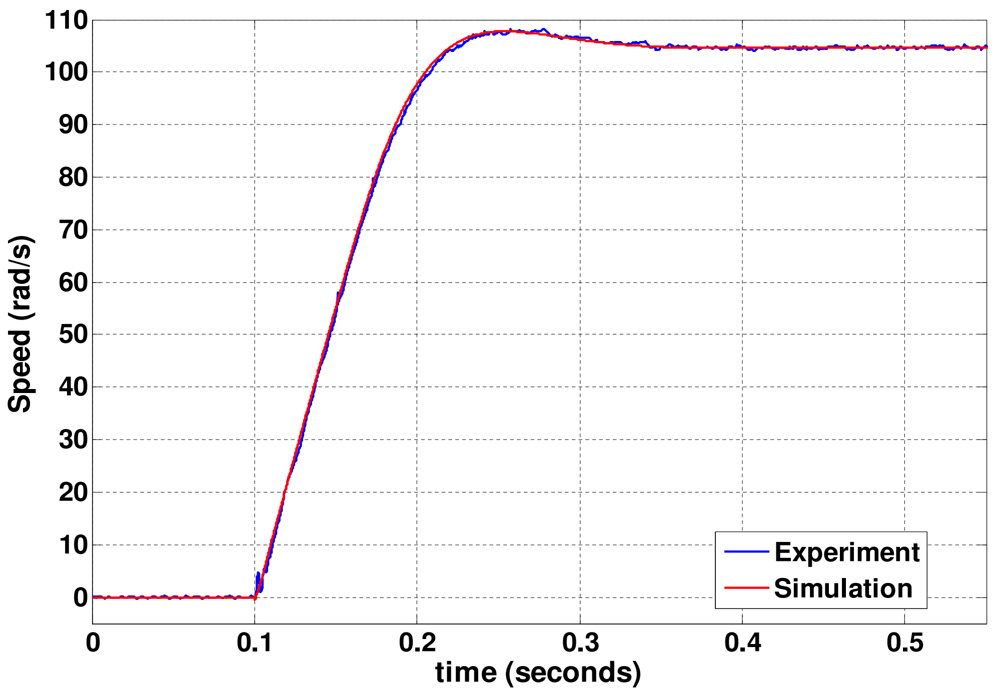


(a) Controller response to step demand of 0 to 1000 rpm

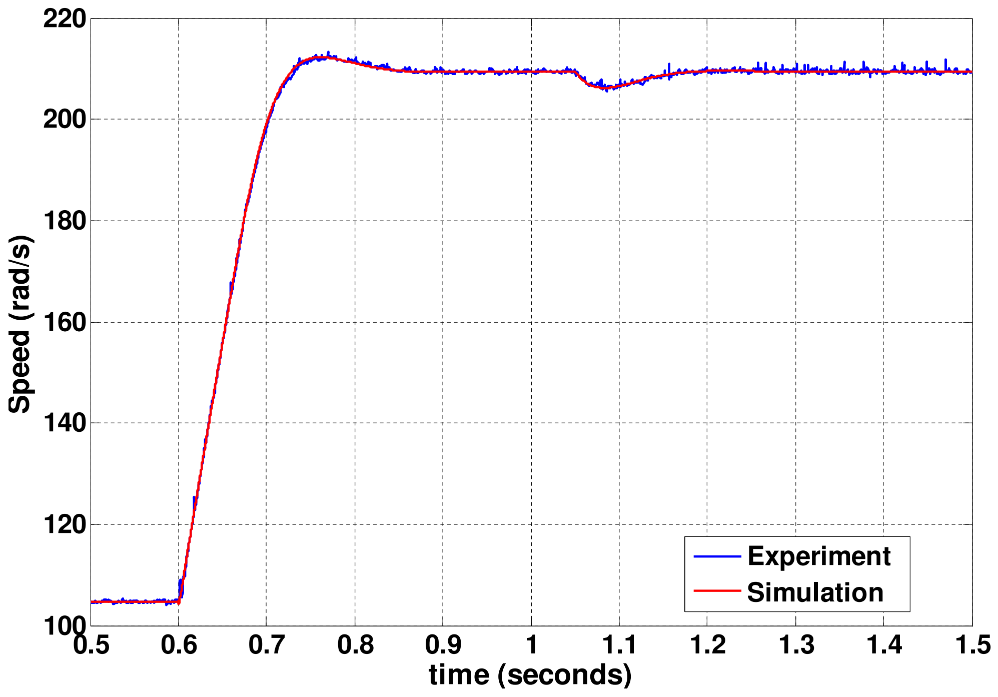


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.21: BF Fourth order Controller Response for Inertia $3J$ and Friction $3B$

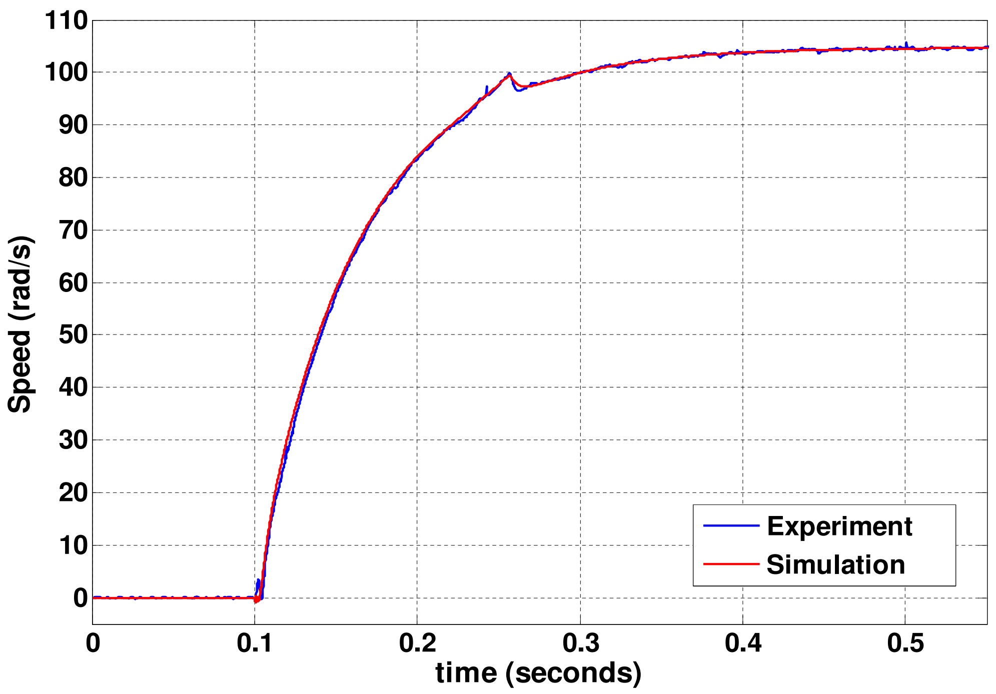


(a) Controller response to step demand of 0 to 1000 *rpm*

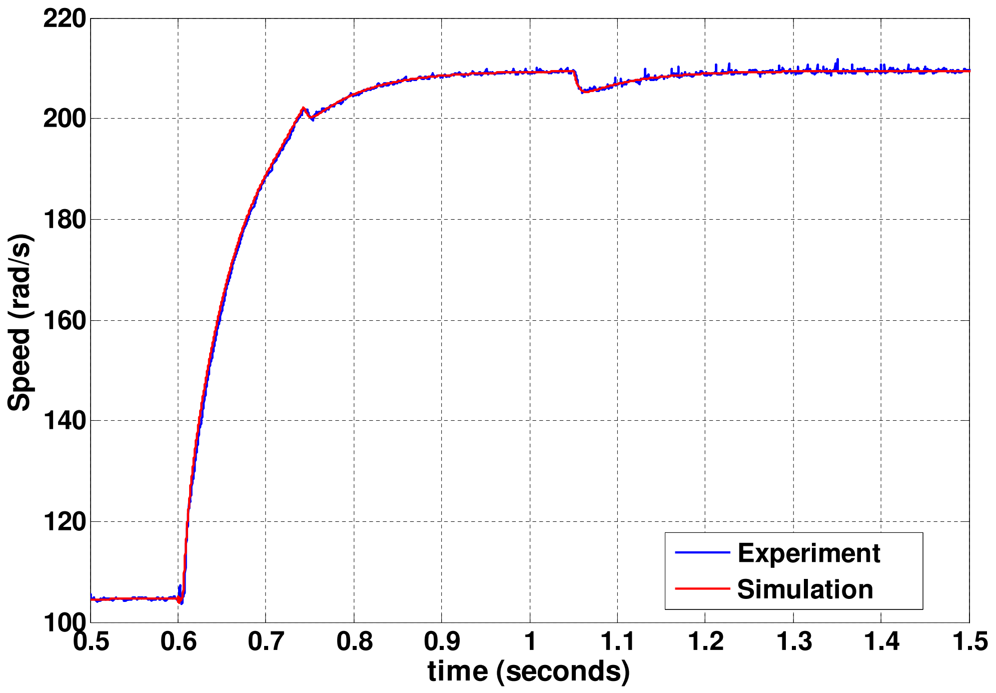


(b) Controller response to step demand of 1000 to 2000 *rpm* with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.22: BF Fourth order Controller Response for Inertia $5J$ and Friction B

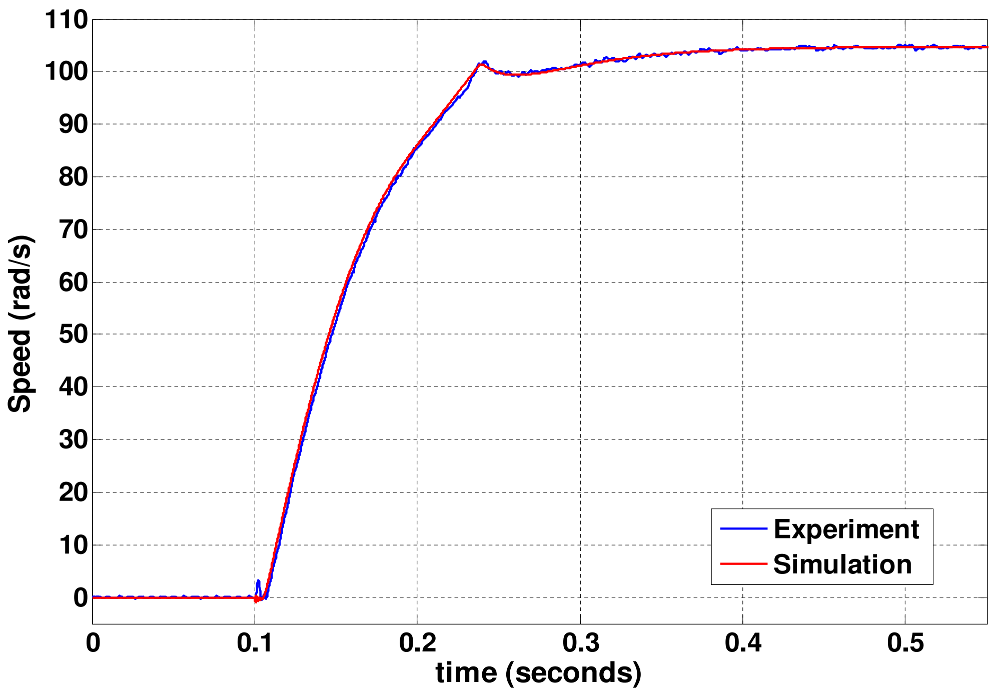


(a) Controller response to step demand of 0 to 1000 rpm

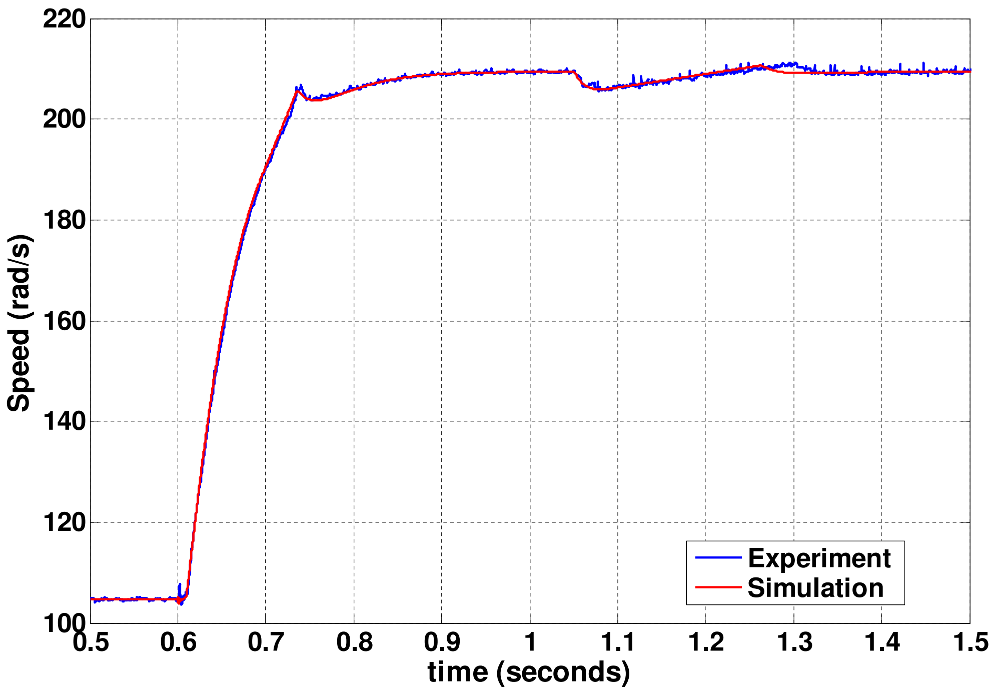


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.23: BF Fourth order Controller Response for Inertia J and Damping $5D$

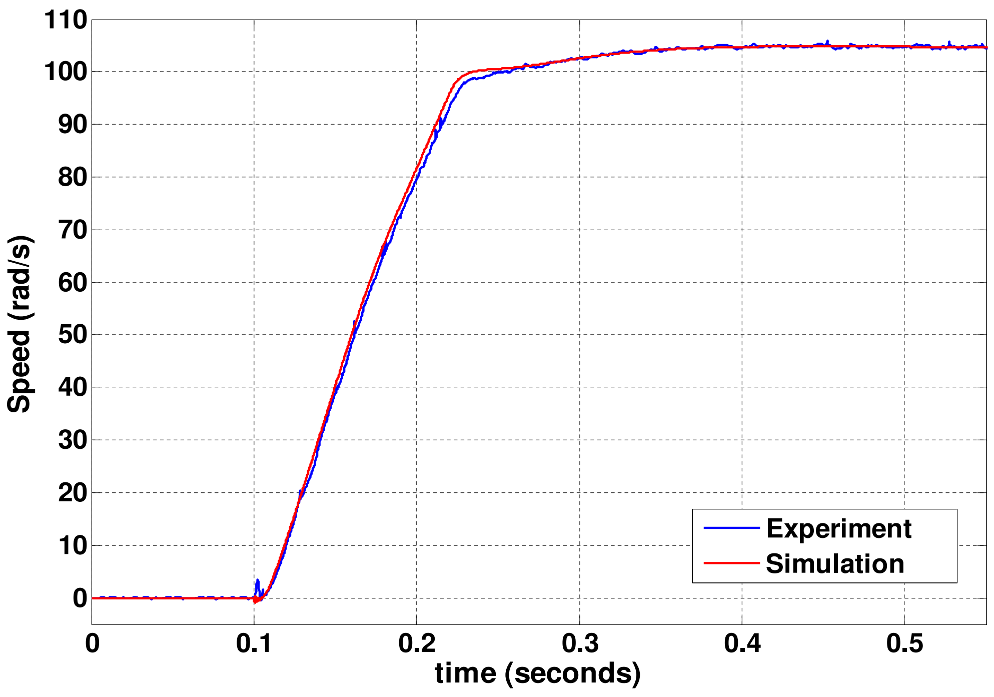


(a) Controller response to step demand of 0 to 1000 rpm

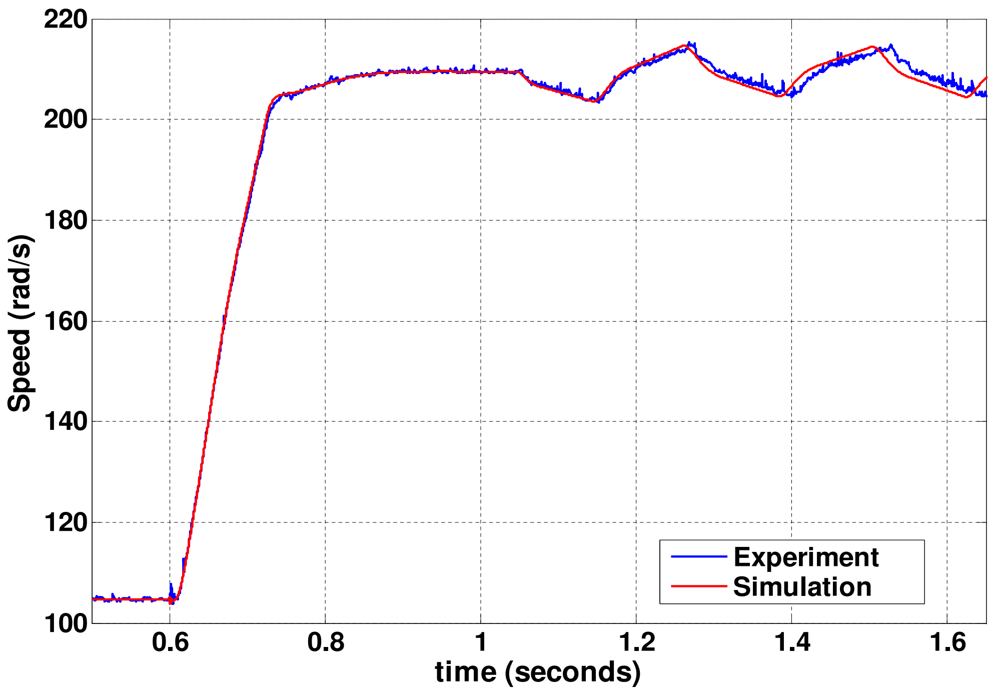


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.24: BF Fourth order Controller Response for Inertia $3J$ and Damping $3D$

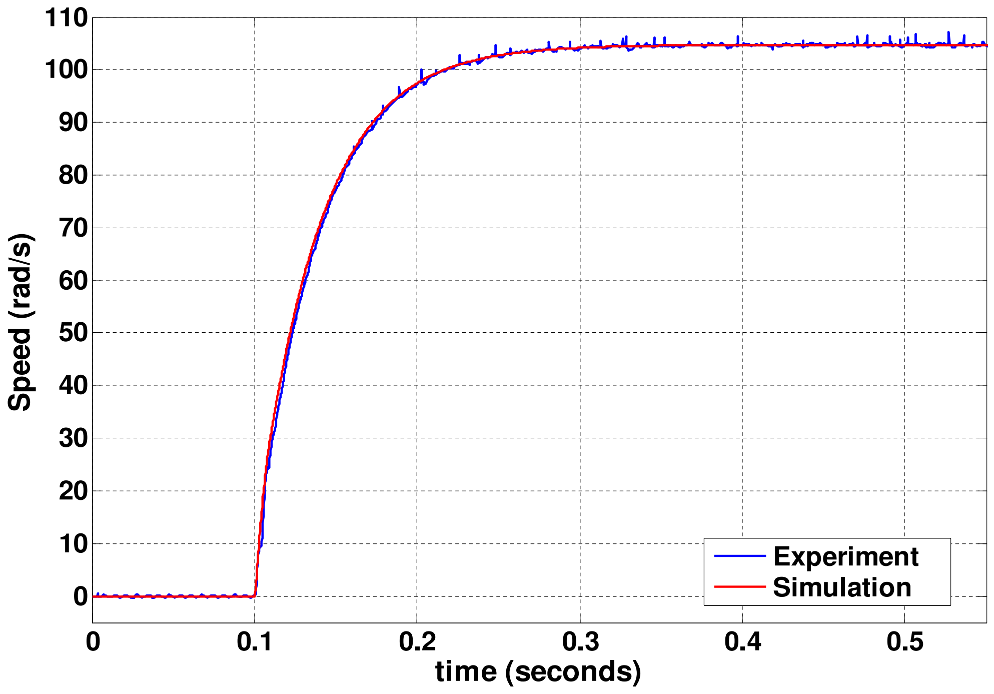


(a) Controller response to step demand of 0 to 1000 rpm

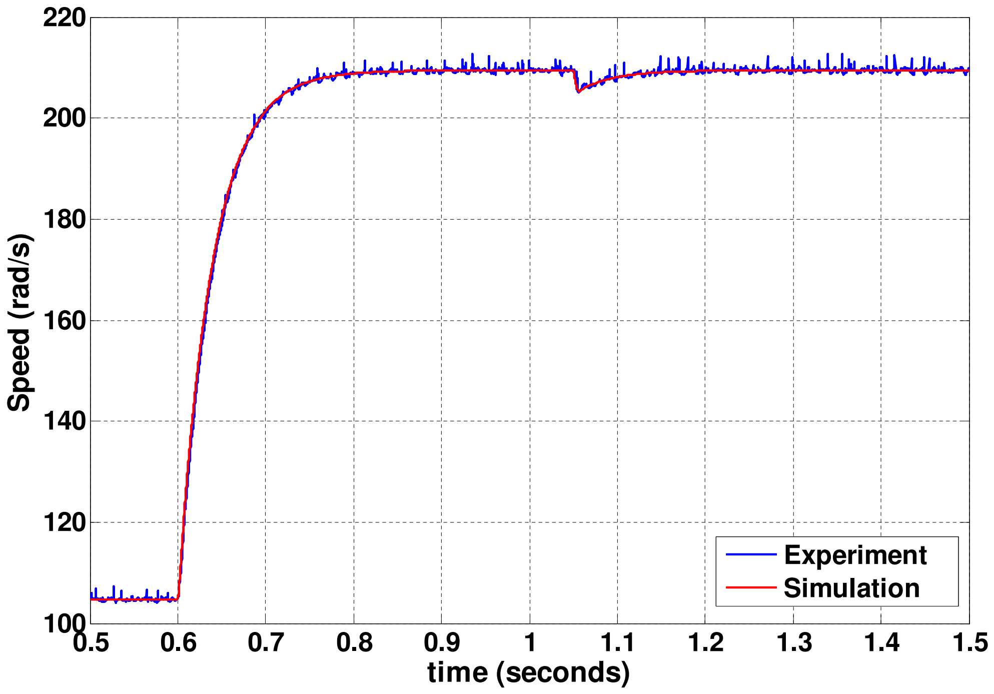


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.25: BF Fourth order Controller Response for Inertia $5J$ and Damping D

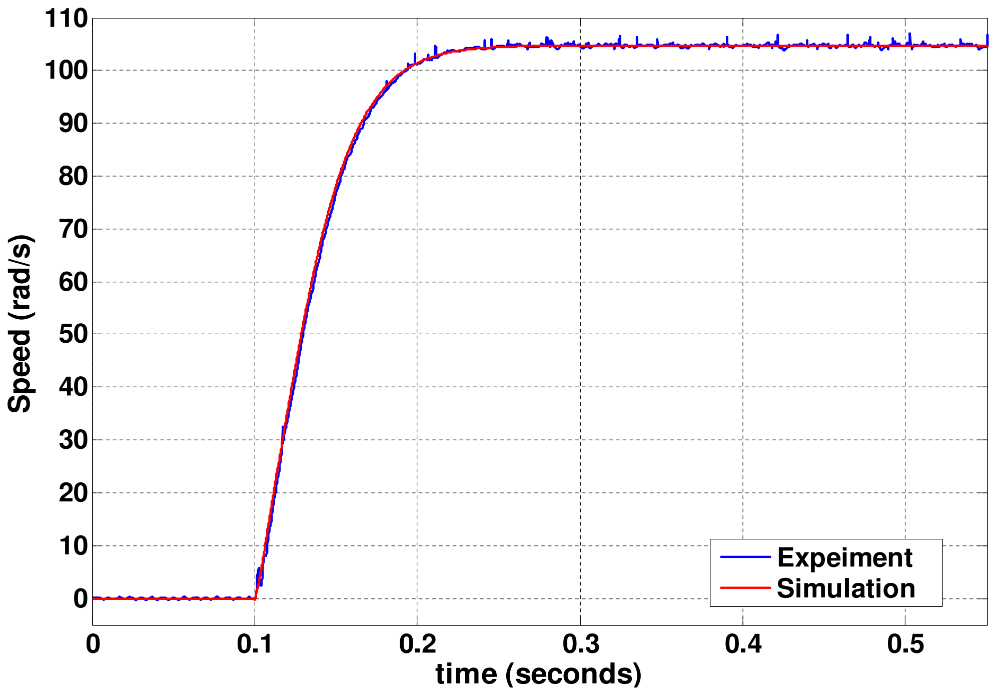


(a) Controller response to step demand of 0 to 1000 rpm

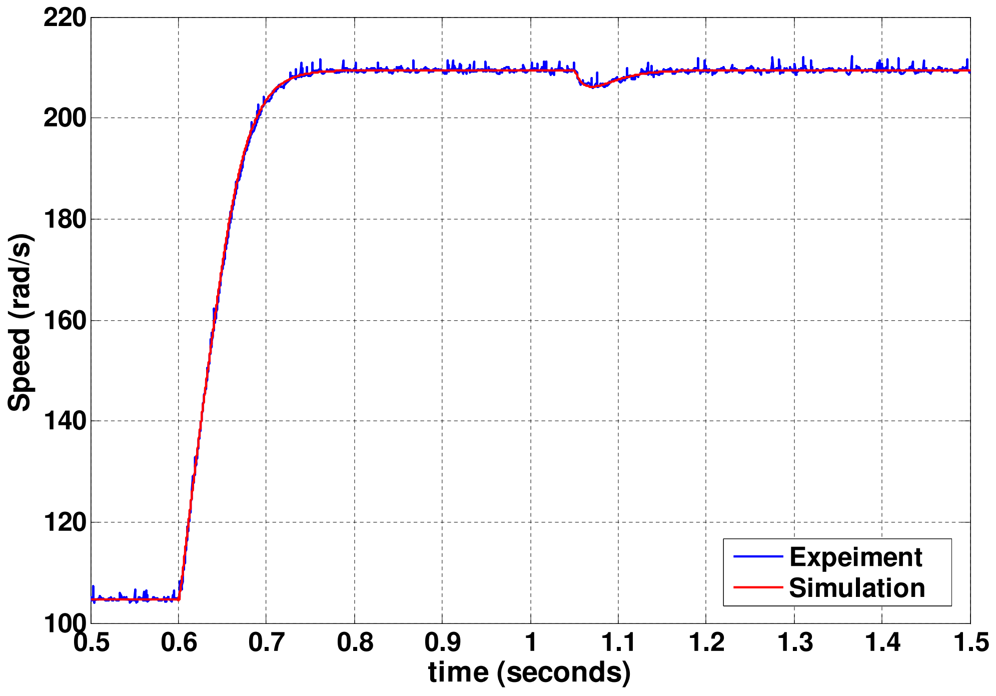


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.26: HBF Fourth order Controller Response for Inertia J and Friction $5B$

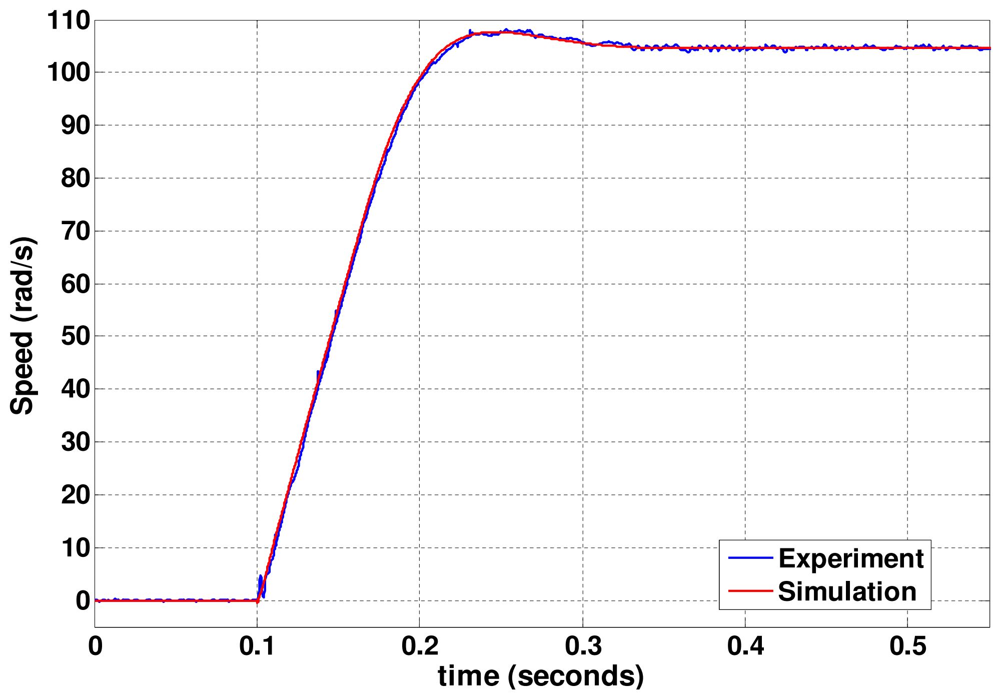


(a) Controller response to step demand of 0 to 1000 rpm

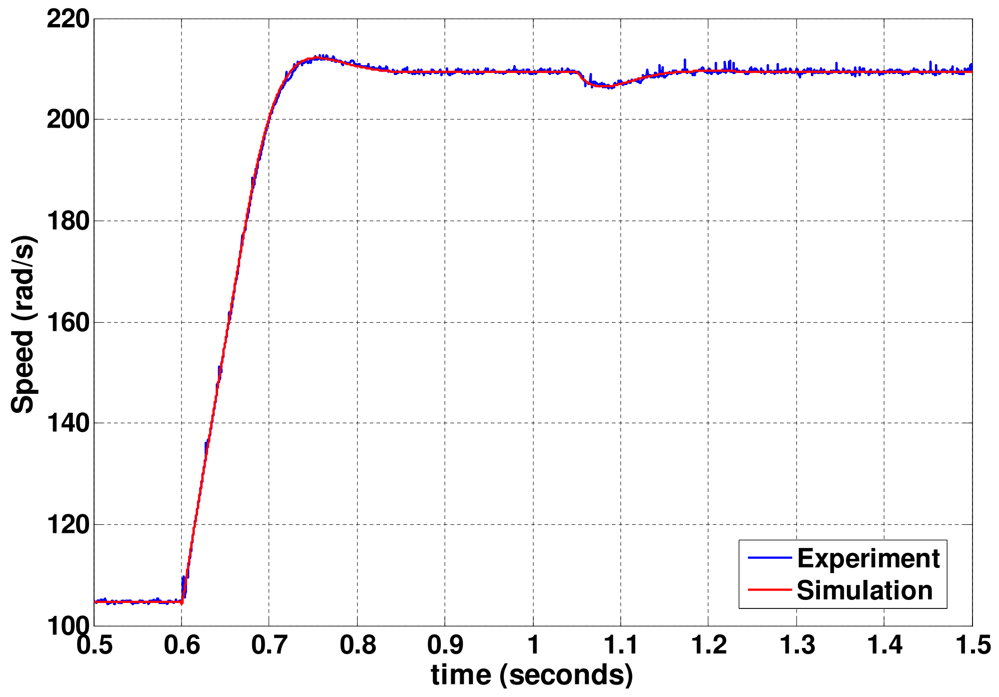


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.27: HBF Fourth order Controller Response for Inertia $3J$ and Friction $3B$

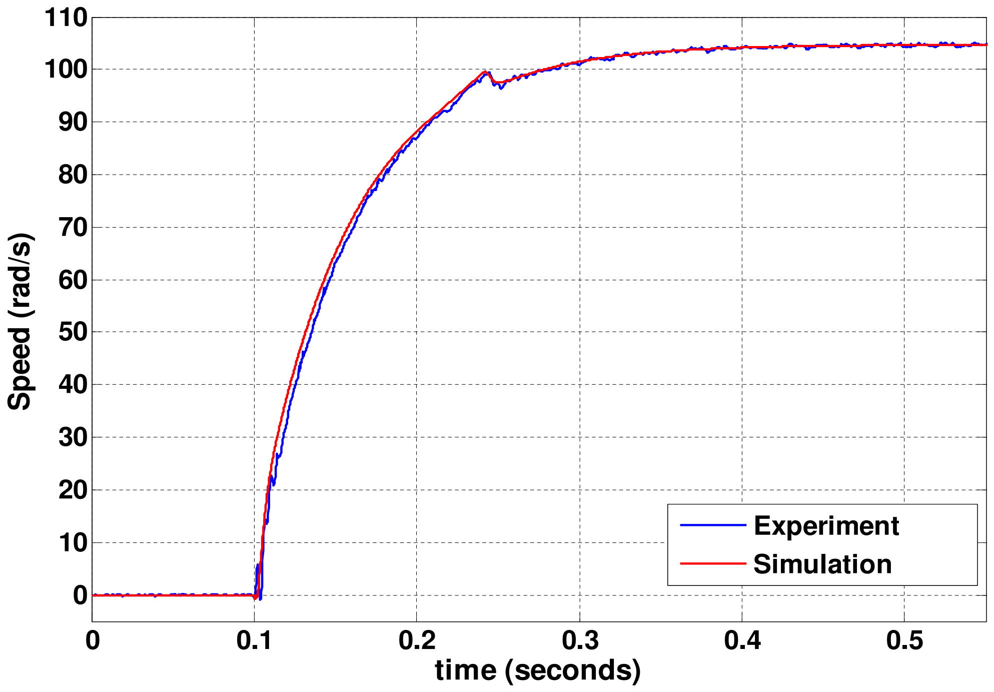


(a) Controller response to step demand of 0 to 1000 rpm

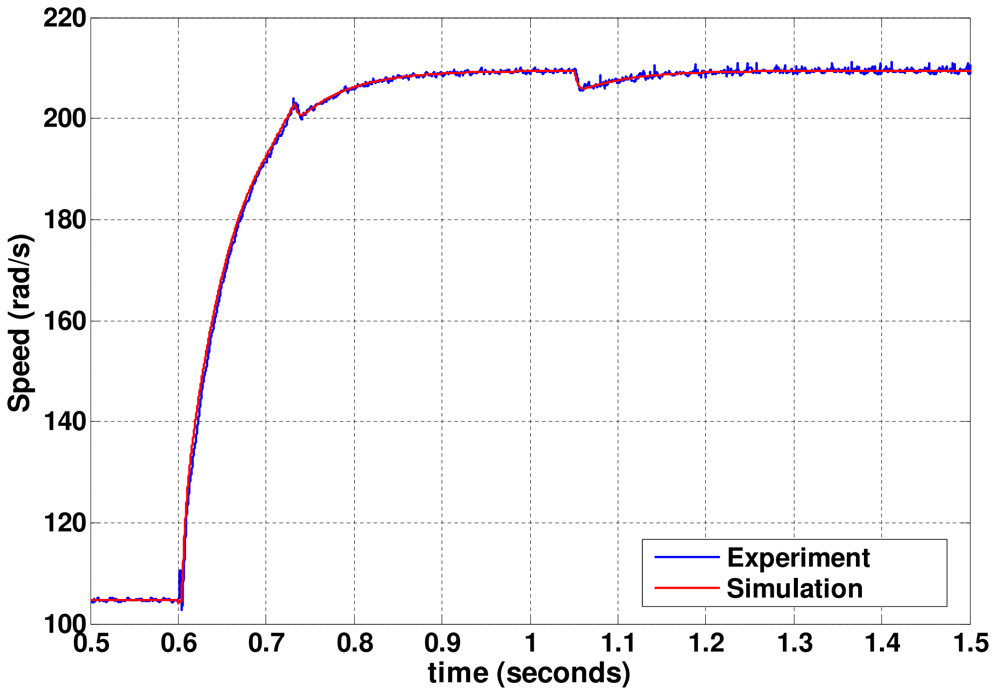


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.28: HBF Fourth order Controller Response for Inertia $5J$ and Friction B

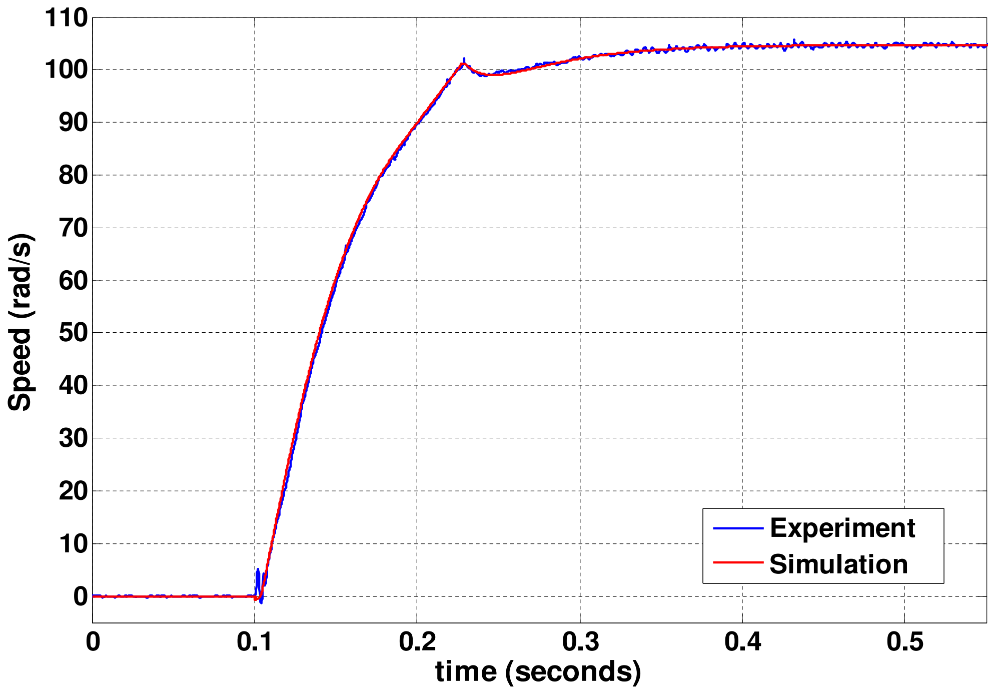


(a) Controller response to step demand of 0 to 1000 rpm

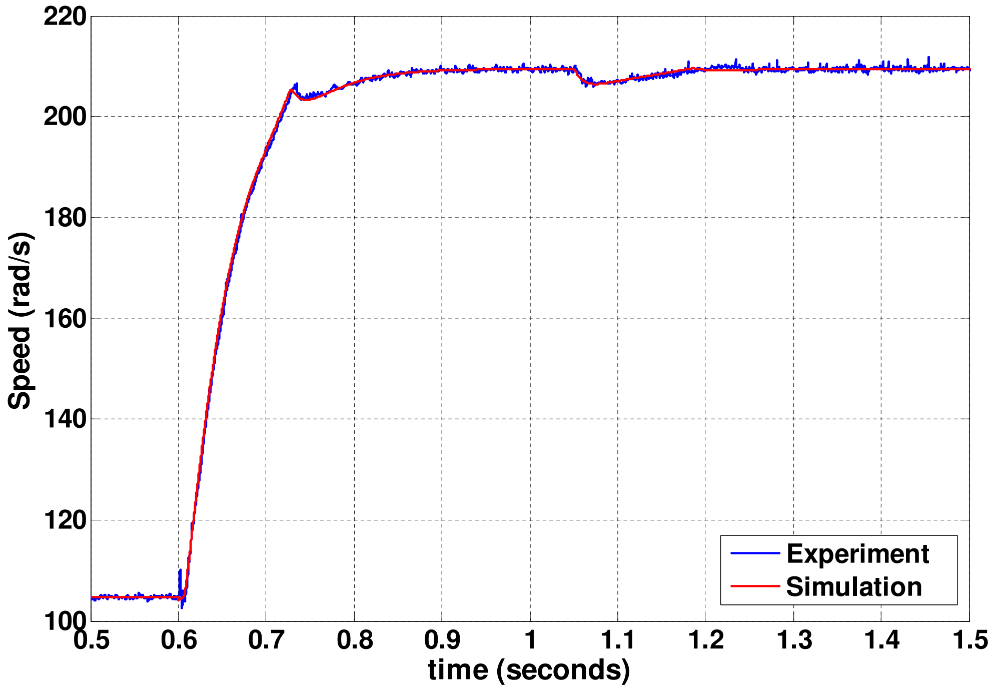


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.29: HBF Fourth order Controller Response for Inertia J and Damping $5D$

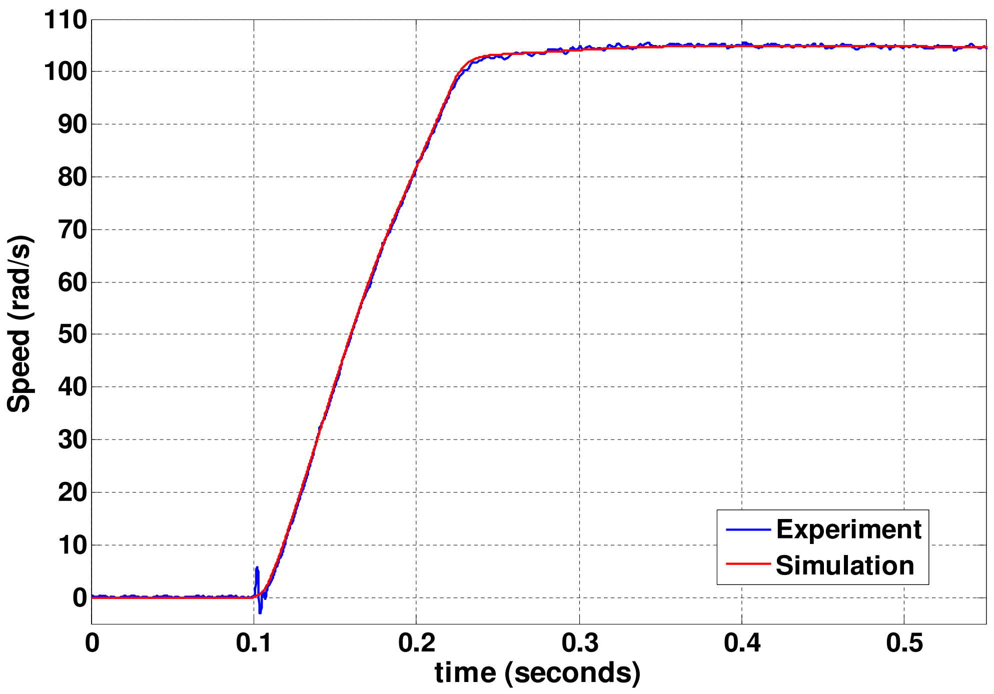


(a) Controller response to step demand of 0 to 1000 rpm

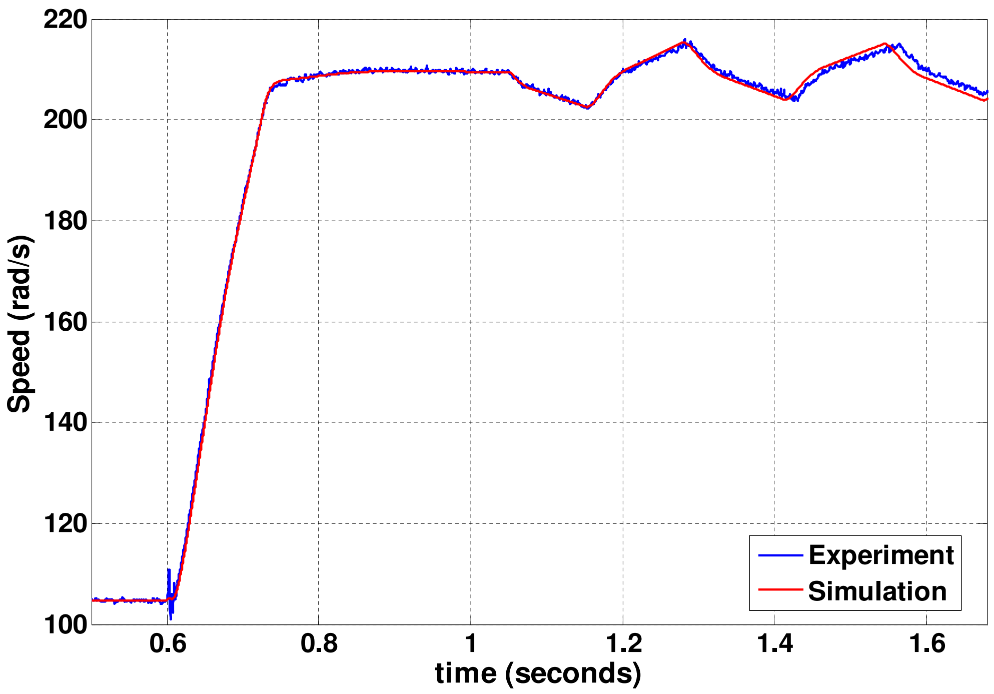


(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.30: HBF Fourth order Controller Response for Inertia $3J$, Damping $3D$



(a) Controller response to step demand of 0 to 1000 rpm



(b) Controller response to step demand of 1000 to 2000 rpm with load disturbance of $0.15Nm$ applied at $1.05s$

Figure 7.31: HBF Fourth order Controller Response for Inertia $5J$ and Damping D

7.7 Conclusion

The chapter has described the automated identification-based approach to the design of robust control systems. The approach subsequently involves the automatic design through heuristic optimisation strategies with constraints to provide robust performance in the face of largely variable mechanical loads and also deliver speed responses with minimum possible overshoot. In order to ensure that the closed-loop system meets the criteria for robust performance, the system is evaluated against different criteria. These performance conditions are the nominal stability, nominal performance and robust stability. A control system that satisfies these three criteria are said to have robust performance.

In evaluating the robust stability, the uncertainty model for the plant is defined. This involves defining the the robust weighting that models the bounds for the uncertainty of the actual plant. Combining both conditions of nominal performance and robust stability give rise to the robust performance conditions. Controllers that satisfy the conditions for robust performance are subject to further testing, which involves quantifying the quality of its response to certain operating conditions. A penalty factor is also added on to the derived performance value for those controlled speed response that have overshoots greater than 8%.

The chapter also highlights the important benefits of the investigated identification-based approach; as well as being simple, given its automated nature, it also introduces much less stress to the robust experimental system when it is applied for a similar optimisation problem when compared to the experimental approach. With regards to the durations of the optimisations, the times can be much reduced when compared to the experimental optimisation process, which largely depends on the speed of the computer processor. The slight drawback lies in the need to model the uncertainty within the mechanical load for which the control system is to be designed. This requires some skill and, when compared with the experimental approach, it is not as easy to adopt.

Chapter 8

Discussions and Conclusion

8.1 Introduction

Chapters 6 and 7 have focused on the implementation of two novel automated methods for robust control system design, which both employ evolutionary algorithms and an optimisation process. These chapters have demonstrated the effectiveness and validity of the algorithms employed in the design procedure given the quality of the obtained experimental results and the good agreement with the simulations. Also highlighted within the chapters are the benefits of employing each of the investigated approaches for a robust control design problem.

The results provided in sections 6.4 and 7.6 are very similar. Hence, in order to provide a useful comparison that gives more insight into the differences in performance of the investigated evolutionary algorithms and the adopted robust design methods, an extended analysis on the obtained results will be provided within the final chapter. The more-in-depth analysis of the results will facilitate effectively comparing the different evolutionary algorithms employed. It will also aid in highlighting the advantages and disadvantages between the two different robust control design methods adopted for the research project. To summarise, the chapter will provide:

1. a comparison between each of the different algorithms - GA, BF and HBF - employed during the research project by adopting two performance indices: efficiency and effectiveness of the evolutionary algorithms.
2. a comparison between the two control design methods - the experimental and identification based approach - employed to achieve the aims of the research by harnessing the performance indices of efficiency and effectiveness.
3. recommendations on the best combination of robust control system design method and evolutionary algorithms to employ depending upon the complexity of the system that needs to be controlled.
4. suggestions for future directions of the research work in order to further facilitate the realisation of the benefits that can be reaped from the ardent investments of time and effort that have been made in the research project

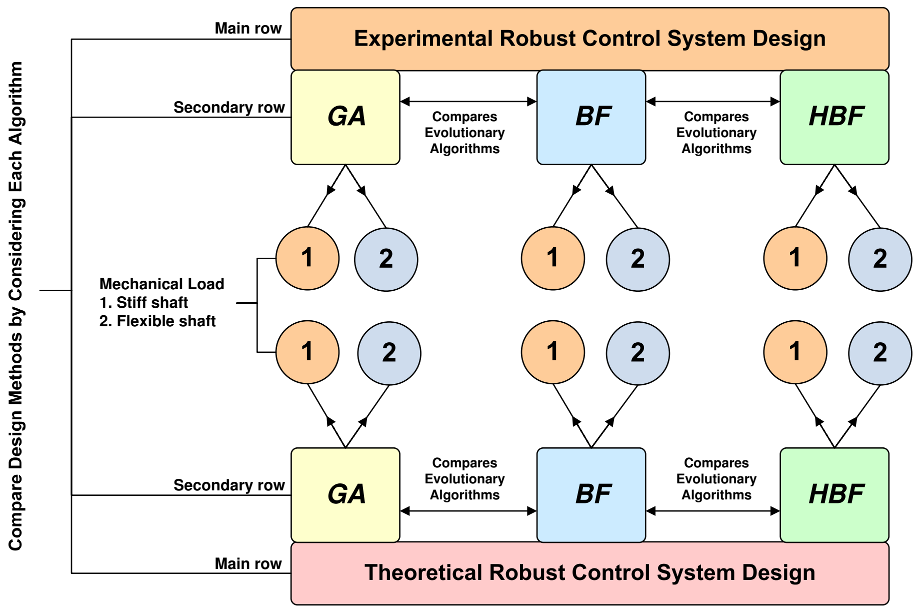
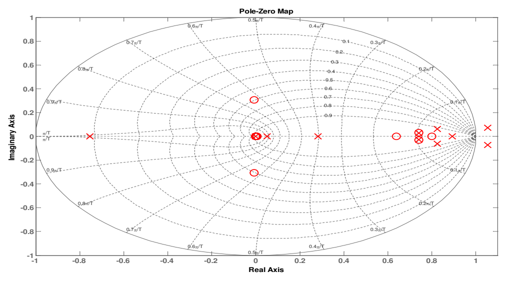
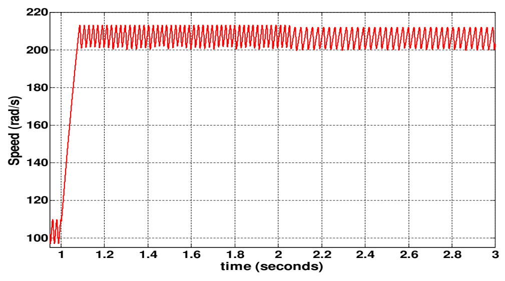

Compare Design Methods by Considering Each Algorithm
Main row
Experimental Robust Control System Design
Secondary row
GA
Compares Evolutionary Algorithms
BF
Compares Evolutionary Algorithms
HBF
Mechanical Load
1. Stiff shaft
2. Flexible shaft
1
2
Secondary row
GA
BF
HBF
Main row
Theoretical Robust Control System Design

Figure 8.1: Classification of Results

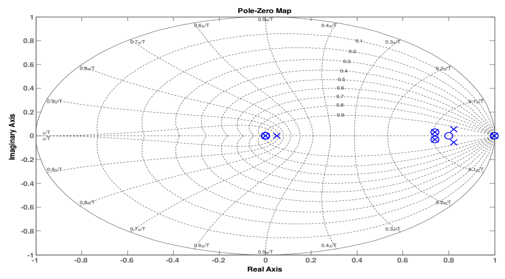
Figure 8.1 summarises the classification of the results obtained for the research project and also presents a format for comparing the results obtained using the different robust control system design methods and evolutionary algorithms.



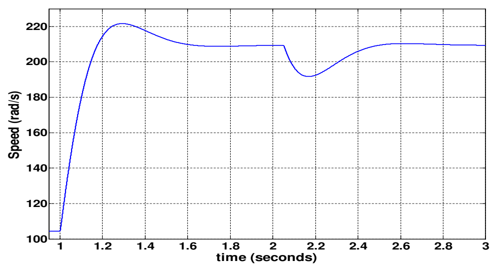
(a) First Stage: Root Locus Plots



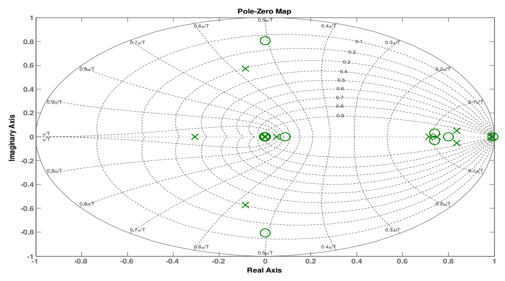
(b) First Stage: Step Response



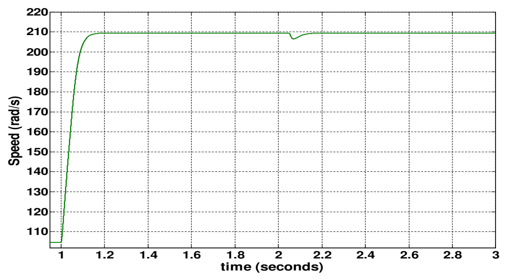
(c) Second Stage: Root Locus Plots



(d) Second Stage: Step Response



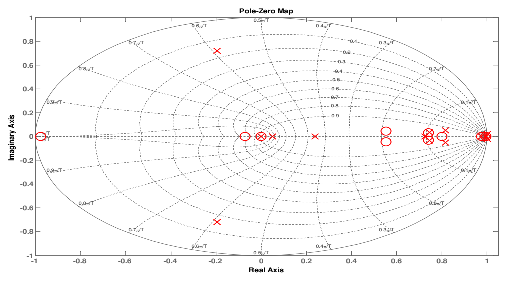
(e) Final Stage: Root Locus Plots



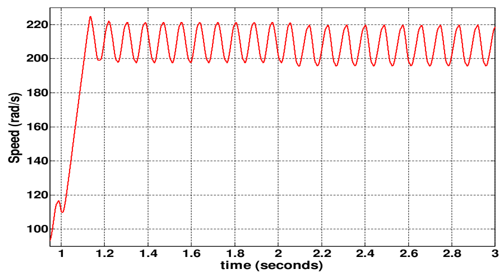
(f) Final Stage: Step Response

Figure 8.2: Three stages during EA optimisation for Stiff Shaft Plant

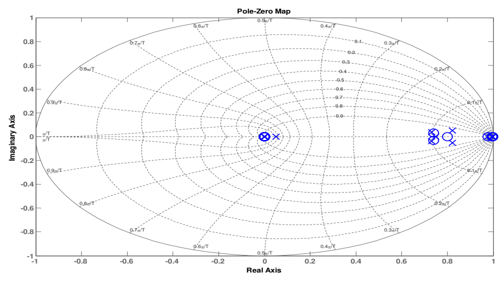
Figures 8.2 and 8.3 present a pictorial reminders of the typical results achieved during the optimisation process for the control system design; they show the root-locus and step response plots of closed-loop systems for the stiff and flexible shaft mechanical loads evolving to the best possible solutions through three distinct phases of the optimisation process. Figure 8.2(a) shows a solution from the initial stages of the closed system optimisation process for the stiff shaft mechanical plant. The solution obtained clearly has some issues with stability; this can be observed in the root-locus plot, with some of its closed loop poles existing outside the boundaries set by the unit circle and also in its time domain response to a step speed demand of $1000rpm$ and $2000rpm$ and a disturbance torque of $0.15Nm$ at $2.05s$ in figure 8.2(b). Figures 8.2(c) and 8.2(d) highlight a solution that has been achieved further into the EA optimisation while figures 8.2(e) and 8.2(f) show the final solution of the EA optimisation process.



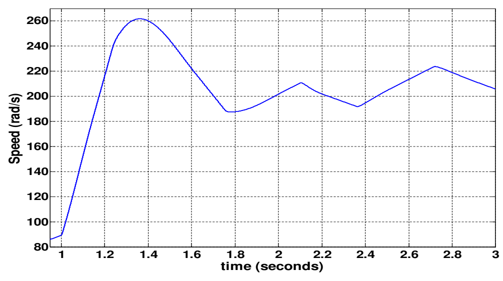
(a) First Stage: Root Locus Plots



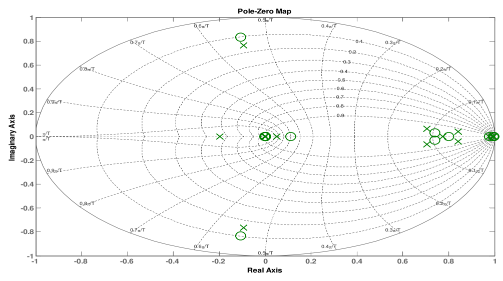
(b) First Stage: Step Response



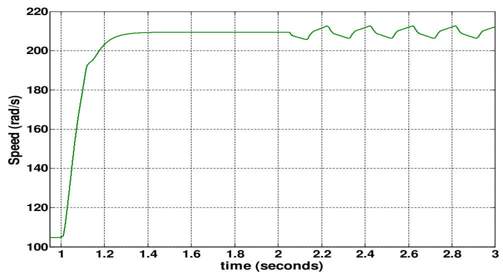
(c) Second Stage: Root Locus Plots



(d) Second Stage: Step Response



(e) Final Stage: Root Locus Plots



(f) Final Stage: Step Response

Figure 8.3: Three Stages during EA optimisation for Flexible Shaft Plant

In a similar fashion, figures 8.3(a) and 8.3(b) show a solution from the initial stage; its response is clearly oscillatory and this is highlighted in the root-locus plots that has some of its closed loop poles just outside the unit circle. Figures 8.3(c) and 8.3(d) show an improved response which occurs further into the optimisation process and finally, figures 8.3(e) and 8.3(f) show the best possible solution arrived at by the end of the EA optimisation process.

8.2 Performance Index

The comparison of the investigated evolutionary algorithms and the different control design methods is based on two performance indicators: efficiency and effectiveness. These indices are considered based on the suggestions provided in [90], which illustrates a comparison of Genetic Algorithms other Global Optimisation Algorithms and its use in Calibration Problems. The performance indices employed are described as follows:

1. **Efficiency:** considers the period taken by each algorithm to reach a global minimum. The period is measured by the number of function evaluations required by each algorithm to arrive at its best objective value.
2. **Effectiveness:** considers the minimum objective value achieved by each optimisation algorithm after its termination criteria has been met.

In order to compare the effectiveness of the different algorithms employed and the developed control design methods, it was necessary to redefine a new objective function termed the Performance Comparison Function (PCF). The function is used to provide a common platform to compare the solutions obtained by adopting the different robust control design techniques and evolutionary algorithms. The index considers the rise time, (2%) settling time and overshoot of each controlled response produced separately by the nine combinations of parameter variations for the mechanical loads considered during the automated experimental optimisation process described in section 6.3. The PCF also considers the Integration of the Absolute Error (IAE) between their respective responses and the speed demand. The performance index is given as:

$$PCF\ value = \sum_{i=1}^{9} (rise\ time + settling\ time + overshoot + IAE) \tag{8.1}$$

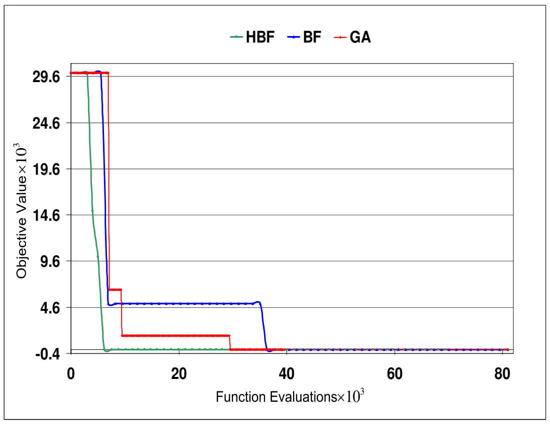
8.3 Comparison of Evolutionary Algorithms

Within this section, the evolutionary algorithms employed will be compared based on the performance indices described in the previous section. The aim of the comparison is to highlight the algorithm provides best performance under each of the two robust control design methods.

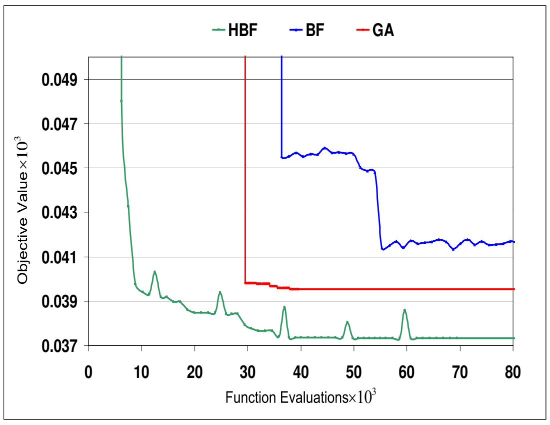
8.3.1 Comparison under Identification-based Approach

Figures 8.4 and 8.5 show the progress made by each optimisation algorithm in finding an optimum solution for the Identification-based design of robust controllers for the stiff and flexible shaft mechanical loads.

Stiff Shaft Mechanical Load: Figure 8.4 shows the progressive descent of the objective function in the investigated algorithms up to their eventual final output. Their profiles are drawn out on the same axis to provide a lucid comparison on the effectiveness and efficiency of the employed algorithms. For the BF, its profile shows that up until 6900 function evaluations, only bad solutions are experienced. There on after, it discovers better solutions and first reaches 10% of its minimum objective value that corresponds to a PCF value of 41.34 after approximately 36400 function evaluations. The reason for its very distinct transition can be attributed to its focus on a local search which dominates its search procedure and only occasionally adopting



(a) Optimisation profile for EA



(b) Close-up of Convergence of EA

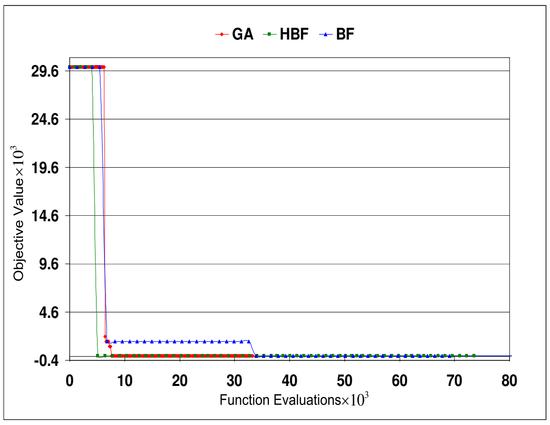
Figure 8.4: EA Stiff Shaft Controller Optimisation

a global search through the elimination and dispersal events. The GA profile tells a somewhat similar story: up until the 7130 function evaluations, only bad solutions are discovered. Thereto, significantly better solutions are discovered until it first arrives (10%) around its optimal solution that has a corresponding PCF value of 39.54; this occurs after 29500 function evaluations have been performed.

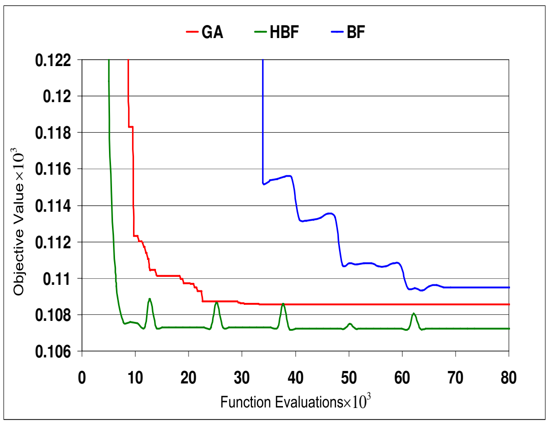
The best (most efficient and effective) of the three algorithms, based upon the performance criteria employed, is the HBF; it converges fastest, after 7800 function evaluations, to 10% of its optimal solution that has a corresponding PCF value of 37.33. This is also the smallest PCF value attained. The algorithms demonstrates improvements over its parents, the BF and GA; it incorporates the best features of each parent algorithm and the result is an improvement in its global search ability.

	Stiff Shaft		**Flexible Shaft**	
	Evaluations	PCF Value	Evaluations	PCF Value
BF	36400	41.34	34000	109.31
GA	29500	39.54	10000	108.57
HBF	7800	37.33	6400	107.23

Table 8.1: EA Performance Summary under Theoretical Design Method



(a) Optimisation profile for EA



(b) Close-up of Convergence of EA

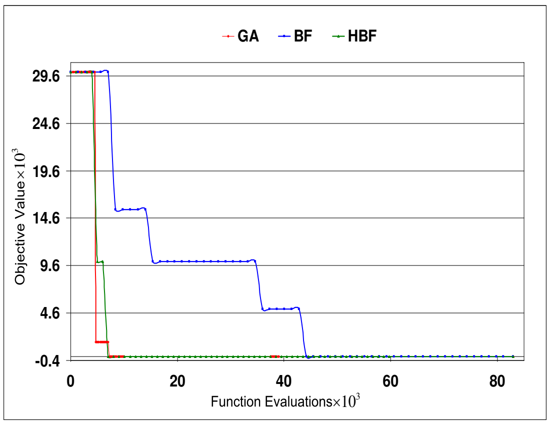
Figure 8.5: EA Flexible Shaft Controller Optimisation

Flexible Shaft Mechanical Load: The PCF profiles created by the algorithms in their respective searches for an optimal solution is shown in figure 8.5. In its optimisation procedure, the BF first reaches 10% of its final solution after 34000 function evaluations. The optimal solution attained by the BF has a corresponding PCF value of 109.31. The best solution attained by the GA has a corresponding PCF value of 108.57; After approximately 10000 function evaluation, the algorithm first discovers a value, which is 10% of its best. As in the case of the robust controller optimisation for the stiff shaft load, the HBF is the best optimisation algorithm in terms of the performance criteria employed: it obtains a best solution with PCF value of 107.23 and it first arrives at 10% of its best objective value after 6400 function evaluations. The summarised details of the optimisation results obtained by the different algorithms employed under the Identification-based control design method for the mechanical loads is shown in Table 8.1.

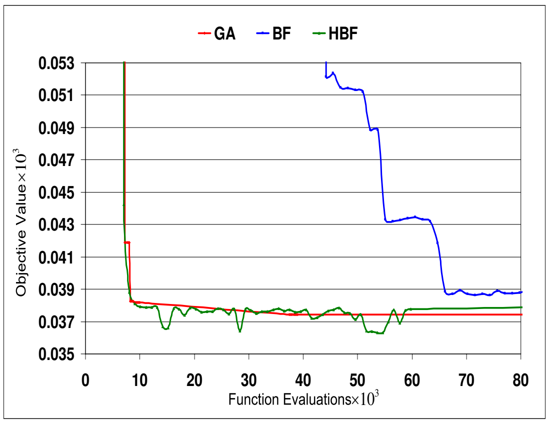
8.3.2 Comparison under Experimental Control Design

Figures 8.6 and 8.7 show the progress profiles achieved by the different algorithms employed during the automated experimental robust control design of the controllers for the stiff and flexible shaft loads.

Stiff Shaft Mechanical Load: From figure 8.6(b), it can be seen that the BF optimisation algorithm first converges to 10% of its best value after about 50000 function evaluations. The best controller solutions discovered by the BF have a corresponding PCF value of 38.63. The GA give a better performance in its search for a best solution: it reaches 10% of its best solution after 8000 function evaluations and with a corresponding PCF value of 37.43, the GA performs better than the BF in its optimisation process. The HBF outperforms the both the BF and GA firstly, by reaching 10% of its best value, only after 7000 function evaluations and secondly, the minimum value it attains has a corresponding PCF value of 36.31.



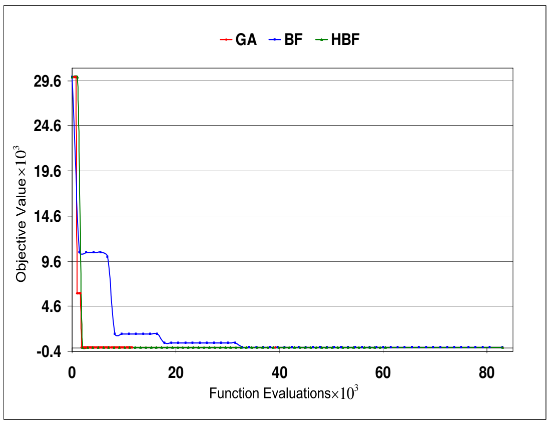
(a) Optimisation profile for EA



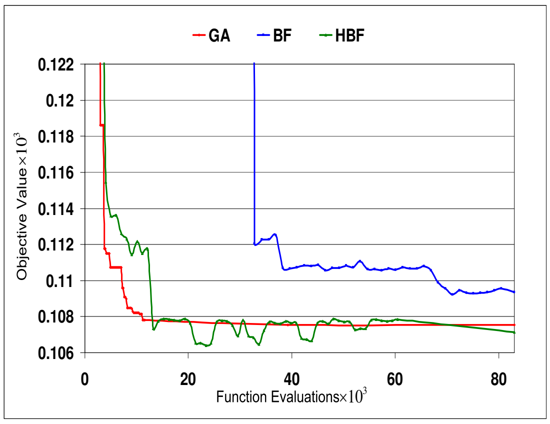
(b) Close-up of Convergence of EA

Figure 8.6: EA Experimental Stiff Shaft Controller Optimisation

Flexible Shaft Mechanical Load: Figure 8.7(b) highlights the latter stages of the optimisation for each of the employed algorithms. The BF first reaches about 10% of its best solution after about 32000 function evaluations. Its best solution has a corresponding PCF value of 109.23. The GA gives a better overall performance than the BF: it reaches 10% of its final value only after about 5000 function evaluations. and its best solution has a lower PCF value of 107.53. Again the HBF, which combines the excellent local search capability of the BF with the more global search ability of the GA is the best performer: it reaches 10% of its best objective value after about 4000 function evaluations and its best solution has a corresponding PCF value of 106.39. The summarised results obtained by the different algorithms under the automated experimental control design method for the mechanical loads is shown in Table 8.2.



(a) Optimisation profile for EA



(b) Close-up of Convergence of EA

Figure 8.7: EA Experimental Flexible Shaft Controller Optimisation

The performances of the GA relative to the BF obtained is supported by the results provided in [68] and [70], which highlight the more global search ability of the GA over the BF from the quicker convergence it achieves around its final solution during its search procedure. The performance of the HBF display improvements over its parent algorithms. These improvements were 'somewhat' expected given the performance comparisons, obtained in [70], [71] and [72], between other Hybrid Evolutionary Algorithms and their parent algorithms. It must be highlighted that the

	Stiff Shaft		**Flexible Shaft**	
	Evaluations	PCF Value	Evaluations	PCF Value
BF	50000	38.63	32000	109.23
GA	8000	37.43	5000	107.53
HBF	7000	36.31	4000	106.39

Table 8.2: EA Performance Summary under Experimental Design Method

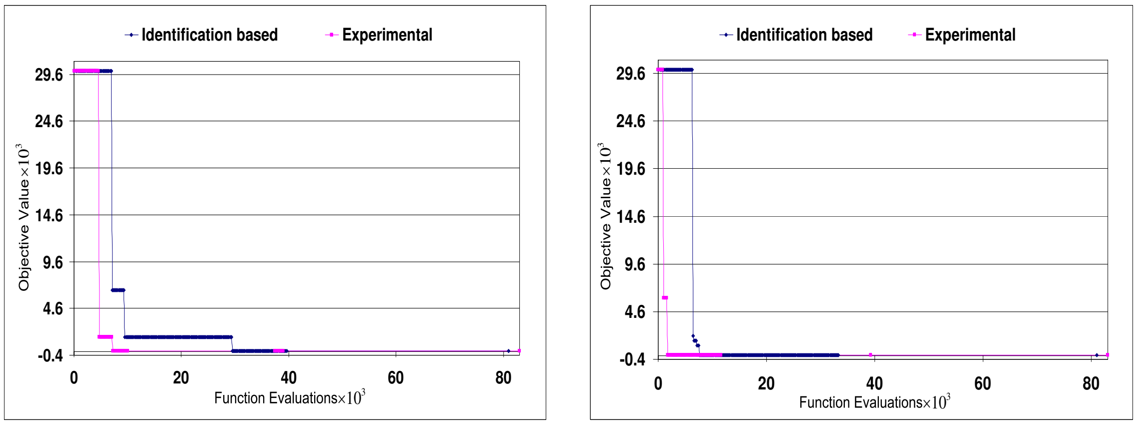
comparisons were implemented for optimisation tasks, which are different from that considered within this research project.

8.4 Comparison of Control Design Methods

The efficiency and the effectiveness of the design methods employed with each of the different optimisation algorithms has be compared. The summary of the comparison will be presented in the following sections.

8.4.1 Comparison under *GA*

Figure 8.8 shows the profiles of the *GA* employed under the identification-based and experimental design methods juxtaposed on the same axis. It can be observed that the experimental approach performs better than its identification-based counterpart in terms of the speed in finding a best controller solution for each mechanical load considered for the research project.



(a) *GA* optimisation - Stiff Shaft (b) *GA* optimisation - Flexible Shaft

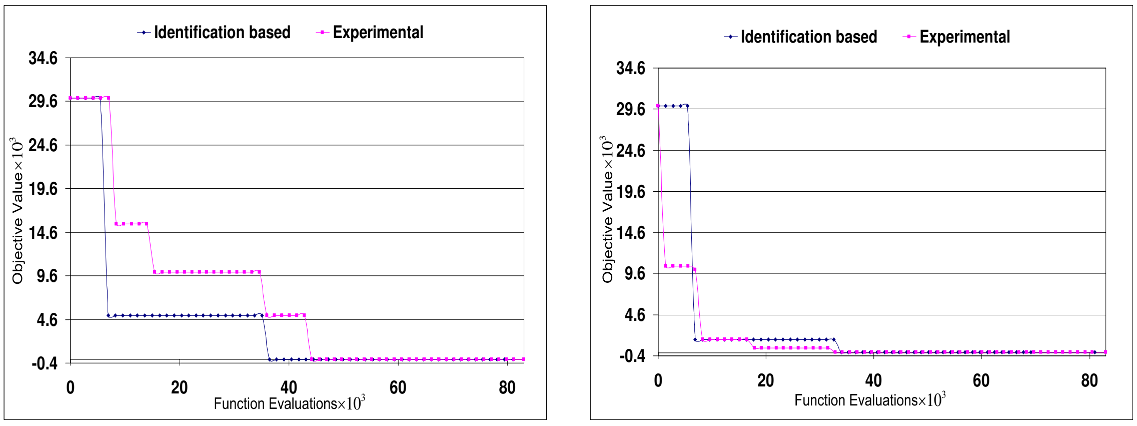
Figure 8.8: *GA* optimisation Profile

The experimental approach can be said to be more efficient because it takes fewer function evaluations in achieving an optimal solution. The experimental approach takes approximately 8000 and 5000 function evaluations which is better than the corresponding 29500 and 10000 function evaluations required by the identification-based approach for the stiff and flexible shaft loads respectively.

The experimental design method is also more effective, slightly achieving a better quality controller solutions than its counterpart for both mechanical loads: for the stiff and flexible shaft loads respectively, the experimental approach achieves controlled responses with PCF values of 37.43 and 107.53 while the identification-based design achieves a PCF value of 39.54 and 108.57.

8.4.2 Comparison under BF

Figure 8.9 presents a side by side comparison of the progressive search made by the identification-based and experimental approach, each employing the BF optimisation algorithm during the design procedure.



(a) BF optimisation - Stiff Shaft (b) BF optimisation - Flexible Shaft

Figure 8.9: BF optimisation Profile

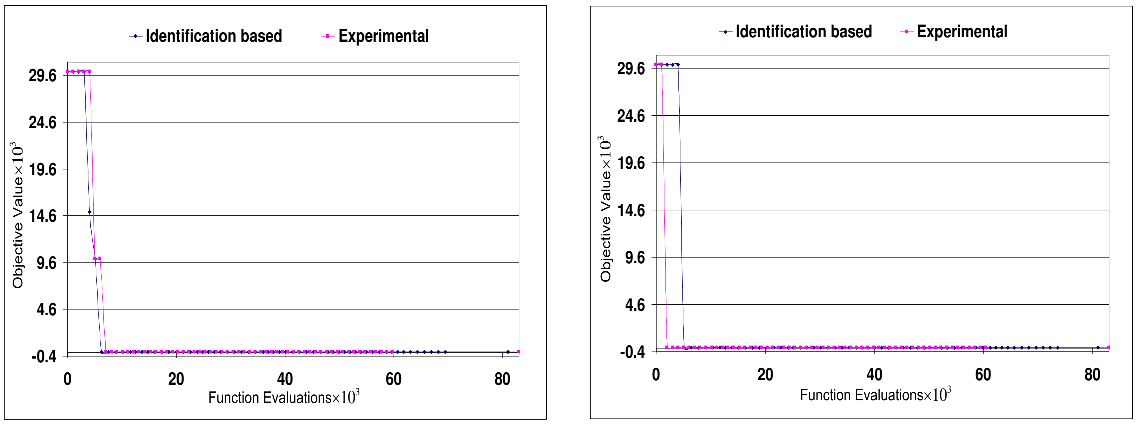
For both the stiff shaft and flexible shaft load cases, the experimental design method is more effective because it achieves better quality controller solutions with a PCF

value of 38.63 and 109.23 compared with the PCF value of 41.34 and 109.31 obtained by the identification-based approach respectively.

In terms of efficiency, although for the stiff shaft load, the identification-based approach is more efficient given it requires 36400 function evaluations to arrive around its best solution compared to the 50000 function evaluations required by the experimental approach, the latter is slightly more efficient in optimising controllers for the flexible shaft load, given it requires 32000 function evaluations compared to the 34000 function evaluations required by its alternative.

8.4.3 Comparison under HBF

In figure 8.10, the PCF profiles of the progressive search of the HBF employed under the identification-based and experimental approach are presented on the same axis.



(a) HBF optimisation - Stiff Shaft (b) HBF optimisation - Flexible Shaft

Figure 8.10: HBF optimisation Profile

In employing the HBF, the experimental approach is more efficient in achieving a good end result to the optimisation task. It requires 7000 and 4000 function evaluations for the stiff and flexible shaft loads respectively. The theoretical approach is less efficient given it arrives around its best objective value after 7800 and 6400 function evaluations during the controller optimisation for the stiff and flexible shaft load respectively.

Regarding the effectiveness of the two approaches, The experimental approach is marginally more effective given the better quality of the controller solutions it discovers during its search procedure. For the stiff and flexible shaft loads, with the experimental approach, the HBF achieves minimum PCF values of 36.31 and 106.39 while the identification-based approach attains minimum values of 37.33 and 107.23. Also, a point to note is, in employing the HBF for the optimisation problem, the difference between the results obtained using the identification-based and the experimental design method is very much reduced.

8.5 Recommendations on Design Strategy

Based upon the results and logically exhaustive discussions provided in section 8.3, certain recommendations will be made in this section on the design strategies analysed in this research project. Also recommendations on a design strategy will be proposed within a wider context; this would depend on the complexity of the plant for which a control system is to be designed. It would also depend on time constraints placed on finding suitable robust controller solutions to design problems.

For this research project, in designing robust controllers for the case of a stiff shaft mechanical load, an *Experimental Approach* that employs the HBF optimisation algorithm would be the preferred design strategy given it is the most efficient in its search procedure and it is most effective, in terms of the quality of the solution it achieves. In a similar vein, for the design of robust controllers for the flexible shaft mechanical load case, an *Experimental Approach* that employs the HBF turns out to be the preferred choice for very much the same reasons highlighted earlier.

A possible reason for the success of the experimental approach over the theoretical approach could be that, apart from the fact that the real system prototype is directly used, the experimental approach is optimised specifically for distinct combinations of mechanical parameter values. For this reason, it is highly optimised for these values and, given a similar objective function is employed within the Performance Comparison Function, it has a better value representing its quality. With the identification-based approach, the controllers are optimised for a continuum of combinations of parameter values that may include other combinations slightly larger than the maximum that is distinctly defined in the experimental design approach. Hence, with the identification-based approach, what we have is a controller that is robust to a range of parameters values that is slightly larger than the range considered in the experimental design technique. In so doing, sacrifices are made in performance for the particular combination of parameters used in the Performance Comparison Function (PCF).

It must be highlighted that the small price paid in employing the identification-based approach, in terms of performance, is greatly rewarded by the fact that the risks of wear and damage to the experimental system is entirely avoided during the control optimisation procedure. Regarding the Evolutionary Algorithms, in all the considered cases, the HBF is the overall best optimisation algorithm under the considered performance criteria.

In a wider context, given the experience obtained during the research project, if the system for which a controller is to be designed is somewhat complex, it could be advisable to employ a combination of the direct experimental design method and the identification-based approach in optimising robust controllers. This would involve possibly testing controllers first with on a good simulation model of the system under the theoretical design method and only those that are potentially good solutions will be further tested on the rig, which should be kitted with all the necessary protection circuits, during the automated design procedure with the preferred algorithm being the *Hybrid Bacterial Foraging* optimisation algorithm. The obvious benefits of such an approach over the experimental design method could be a reduction in the time required for the optimisation and minimised risk of wear and damage to the experimental system.

8.6 Conclusion and Further Work

The objectives of the research project highlighted in section 1.3, have successfully been achieved. The milestones achieved during the project can be summarised as follows:

1. Three different Evolutionary Algorithms, one of which is an original contribution have been specifically developed in software within MATLAB/Simulink. They are the Genetic Algorithms (GA), Bacterial Foraging (BF) and Hybrid Bacterial Foraging (HBF) Algorithm to enable the implementation of robust control design techniques.

2. Two Automated Robust Control Design Methods have been successfully developed. The First method is a direct design approach implemented on the actual experimental set-up. It is known as the Automated Experimental Robust Control Design Method. The second avoids the direct design method but instead employs a more theoretical approach to robust control system design, which adopts a novel strategy by combining a GA system identification process and a frequency domain $\mathcal{H}_\infty$ control design method. It is aptly termed the Automated Robust Identification-Based Control Design Method.

3. A detailed comparison between the different algorithms employed has been provided based on the results obtained and the way in which they are employed during the separate search procedures implemented by the different algorithms. The result of the comparison presents the HBF as the preferred candidate for robust control design

4. In a similar manner, a detailed comparison of the Design methods employed has been provided. Although the results highlights that the Experimental approach in combination with the HBF is the overall best, in the end, using HBF and the Experimental or the Identification-based approach produce robust control systems with almost equivalent performance, with the advantage that the

identification-based approach does not impose risks of damage and wear to the experimental system.

The Automated design methods developed in the research project has the potential to improve the overall efficiency of variable speed drives which are widely employed in industries worldwide. Given the age of Green Energy, these design methods could aid reductions in green-house emissions and improve energy conversion within generation systems. It also has certain unique potentials for commercialisation, particularly in the process of commissioning variable speed drives. To make the highlighted potential even more practical in the near future, there are a couple of suggestions for further work. The first is directed at improving the performance, in terms of efficiency and effectiveness of the adopted evolutionary algorithms by employing different initialisation methods for population generation. Given the validation obtained from the research work on the strength of the developed methods, a second suggestion would be to apply these algorithms online on industrial processes; its efficiency can then be compared with already established process control systems. By vividly highlighting its potentials in a more practical setting, they could come to fruition more quickly.

References

[1] P.C. Sen, C. S. Namuduri, and P. K. Nandam, "Evolution of control techniques for industrial drives," *IEEE International Conference on Power Electronics, Drives and Energy Systems*, vol. 2, No. 1, pp. 869 – 875, 1996.

[2] D. S. Henderson, "Variable speed electric drives - characteristics and applications," Tech. Rep., Napier University, UK, 1996.

[3] W. Leonhard, *Control of electrical drives*, Springer-Verlag, Berlin, 2001.

[4] A. Hughes, *Electric motors and drives*, Newnes, London, 2006.

[5] M. A. Rahman, "Modern electric motors in electronic world," *IEEE Conference on Industrial Electronics, Control and Instrumentation*, pp. 644 – 648, 1993.

[6] L. Warnes, *Electrical and Electronics Engineering: Principles and Practice*, PALGRAVE, 1998, New York, USA.

[7] P. Shilston, "Development of industrial drives in the 1950s," *Engineering Science and Education Journal*, vol. 101, No. 2, pp. 153 – 158, 2001.

[8] Q. Bi, Wen-Jian Cai, Qing-Guo Wang, Eng-Lock Lee, Yong Sun, and Ke-Dian Liu, "Advanced controller auto-tuning and its application in HVAC systems," *Control Engineering Practice*, vol. 8, No. 2000, pp. 633 – 644, 2000.

[9] K. I. Krakow and S. Lin, "Pi control of fan speed to maintain constant discharge pressure," *ASHRAE Transactions*, vol. 101, No. 2, pp. 398 – 407, 1995.

[10] M. J. Pinnella, E. Wechselberger, D. C. Hittle, and C. O. Pederson, "Self-tuning digital integral control," *ASHRAE Transactions*, vol. PO-86-05, No. 2, pp. 202 – 210, 1986.

[11] H. Dorato, "A historical review of robust control," *IEEE Control Systems Magazine*, vol. 7, No. 2, pp. 44 – 47, 1987.

[12] F. Cupertino, E. Mininno, and D. Nasoand B. Turchianoand L. Salvatore, "On-line genetic design of anti-windup unstructured controllers for electric drives with variable load," *IEEE Transactions on Evolutionary Computation*, vol. 8, No. 4, pp. 347 – 364, 2004.

[13] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, No. 4, pp. 1223 – 1241, 2002.

[14] Z. Michalewicz and M. Michalewicz, "Evolutionary computation techniques and their applications," *IEEE International Conference on Intelligent Processing Systems*, vol. 1, No. 1, pp. 14 – 25, 1997.

[15] Z. Michalewicz, K. Deb, M. Schmidt, and T. Stidsen, "Evolutionary algorithms for engineering applications," Tech. Rep., University of North Carolina, 1999.

[16] Wander G. da Silva, Paul P. Acarnley, and John W. Finch, "Application of genetic algorithms to the online tuning of electric drive speed controllers," *IEEE Transactions on Industrial Electronics*, vol. 47, No. 1, pp. 217 – 219, 2000.

[17] R. Sarker, K. Liang, and C. Newton, "A new multiobjective evolutionary algorithm," *European Journal of Operational Research*, vol. 140, No. 4, pp. 12 – 23, 2001.

[18] Noah Williams, "Robust control," Tech. Rep., Princeton University, 2006.

[19] Nnamdi Okaeme, Pericle Zanchetta, and Mark Sumner, "Robust control design through experimental load identification for variable speed drives," *IEEE Industrial Applications Conference, IAS*, vol. 1, No. 1, pp. 1257 – 1264, 2007.

[20] Thomas Weise, *Global Optimization Algorithms Theory and Application*, Thomas Weise, 2007-11-27 edition, 2007.

[21] K. M. Passino, *Biomimicry for optimisation, control and automisation*, Spring-Verlag, London, 2005.

[22] A. J. Chipperfield and P. J. Fleming, *Parallel and Distributed Computing Handbook*, McGraw-Hill, New York, 1995.

[23] Talib S. Hussain, "An introduction to evolutionary computation," Tech. Rep., Department of Computing and Information science, Queen's University, Kingston, 1997.

[24] Mehrdad Dianati, Insop Song, and Mark Treiber, "An introduction to genetic algorithms and evolution strategies," Tech. Rep., University of Waterloo, 2002.

[25] A.E. Eiben and M. Schoenauer, "Evolutionary computing," Tech. Rep., Free University Amsterdam The Netherlands, 2002.

[26] Chang-Yu Hung, *Material Cutting Plan Generation Using Multi-Expert and Evolutionary Approaches*, Ph.D. thesis, Virginia Polytechnic Institute and State University, 2000.

[27] Mark Burgin and Cristian S. Calude, "Complexity of algorithms and computations," *Theoretical Computer Science*, vol. 383, No. 2-3, pp. 111 – 114, 2007.

[28] Matthew Walker, "Introduction to genetic programming," Tech. Rep., Massey University, 2001.

[29] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimisation and control," *IEEE Control Systems Magazine*, vol. 22, No. 3, pp. 52 – 67, 2002.

[30] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, Oxford University Press, New York, 1997.

[31] H. Maaranen, K. Miettinen, and M. M. Makela, "Quasi-random initial population for genetic algorithms," *Computers and Mathematics with Applications*, vol. 47, No. 2004, pp. 1885 – 1895, 2004.

[32] Shahryar Rahnamayan, Hamid R. Tizhoosh, and Magdy A. Salama, "A novel population initialization method for accelerating evolutionary algorithms," *Computers and Mathematics with Applications*, vol. 53, No. 2007, pp. 1605 – 1614, 2007.

[33] Hamid R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," *IEEE International Conference on Computational Intelligence for Modelling, Control and Automation*, vol. 1, No. 1, pp. 695 – 701, 2005.

[34] Farhang Sahba, Hamid R. Tizhoosh, and Magdy A. Salama, "Application of opposition-based reinforcement learning in image segmentation," *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, vol. 1, No. 1, pp. 246 – 251, 2007.

[35] Lin Han and Xingshi He, "A novel opposition-based particle swarm optimization for noisy problems," *IEEE International Conference on Natural Computation*, vol. 1, No. 1, pp. 1 – 7, 2007.

[36] Daniel Kunkle, "A summary and comparison of moea algorithms," Tech. Rep., Northeastern University, 2005.

[37] Carlos M. Fonseca and Peter J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," Tech. Rep., University of Sheffield, 1993.

[38] David B. Fogel, "What is evolutionary computation?," *IEEE Spectrum*, vol. 1, No. 1, pp. 26 – 32, 2000.

[39] Ajith Abraham, "Evolutionary computation," Tech. Rep., Oklahoma State University, 2005.

[40] Hartmut Pohlheim, "Genetic and evolutionary algorithm toolbox for use with matlab documentation," Tech. Rep., Technical University Llmenau, 2005.

[41] Nnamdi Okaeme, Pericle Zanchetta, and Mark Sumner, "Automated online design of robust position and speed digital controllers for variable speed drives,"

IEEE Industrial Electronics Conference, IECON, vol. 1, No. 1, pp. 4719 – 4724, 2006.

[42] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: Methodologies, architectures, and reviews," *Springer-Verlag Berlin Heidelberg: Studies in Computational Intelligence*, vol. 75, No. 2007, pp. 1 – 17, 2007.

[43] Sloan Centre, "Electrical engineering overview," Tech. Rep., Sloan Career Cornerstone Centre, 2007.

[44] Sigurd Skogestad and Ian Postlethwaite, *Multivariable Feedback Control*, John Wiley and Sons, New York, 1996.

[45] J. Fivas and W. A. Cronje, "Application of evolutionary computation in power electronics," *IEEE Africon*, vol. 43, No. 4, pp. 729 – 734, 2002.

[46] Z. Zhang, H. S. Chung, W. Lo, S. Y. Hui, and A. K. Wu, "Implementation of a decoupled optimisation for design of switching regulators using genetic algorithms," *IEEE Transaction on Power Electronics*, vol. 16, No. 6, pp. 752 – 763, 2001.

[47] E. Diaz-Dorado, J. Cidras, and E. Miguez, "Application of evolutionary algorithms for the planning of urban distribution networks of medium voltage," *IEEE Transaction on Power Electronics*, vol. 16, No. 6, pp. 879 – 884, 2002.

[48] A. Marczyk, "Genetic algorithms and evolutionary computation," Tech. Rep., The TalkOrigins Archive, 2004.

[49] S. Obayashi, D. Sasaki, Y. Takeguchi, and N. Hirose, "Multiobjective evolutionary computation for supersonic wing-shape optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, No. 2, pp. 182 – 187, 2000.

[50] I. Kroo, "Aeronautical applications of evolutionary design," Tech. Rep., Stanford University, 2004.

[51] E. Alba and J. Francisco Chicano, "Evolutionary algorithms in telecommunications," *IEEE Melecon*, pp. 795 – 798, 2006.

[52] P. D. Karamalis, N. D. Skentos, and A. G. Kanatas, “Selecting array configurations for mimo systems: An evolutionary computation approach,” *IEEE Transactions on Wireless Communications*, vol. 3, No. 6, pp. 1994 – 1998, 2004.

[53] Ruland Manufacturing Company, “What to look for in a servo coupling,” Tech. Rep., Ruland, 1990.

[54] The MathWorks, “xpc target system datasheet,” Tech. Rep., University of Berkeley, 2004.

[55] J. Arellano-Padilla, G.M. Asher, and M. Sumner, “Control of an ac dynamometer for dynamic emulation of mechanical loads with stiff and flexible shafts,” *IEEE Transactions on Industrial Electronics*, vol. 53, No. 4, pp. 1250 – 1260, 2006.

[56] E.R. Collins and Y. Huang, “A programmable dynamometer for testing rotating machinery using a three-phase induction machine,” *IEEE Transactions on Industrial Electronics*, vol. 9, No. 3, pp. 521 – 527, 1994.

[57] P. Sandholdt, E. Ritchie, J.K. Pedersen, and R.E. Betz, “A dynamometer performing dynamical emulation of loads with non-linear friction,” *IEEE International Symposium on Industrial Electronics*, vol. 53, No. 4, pp. 873 – 878, 1996.

[58] Z.H. Akpolat, G.M. Asher, and J.C. Clare, “Experimental dynamometer emulation of nonlinear mechanical loads,” *IEEE Transactions on Industry Applications*, vol. 35, No. 6, pp. 1367 – 1373, 1999.

[59] Z.H. Akpolat, G.M. Asher, and J.C. Clare, “Dynamic emulation of mechanical loads using a vector-controlled induction motorgenerator set,” *IEEE Transactions on Industry Applications*, vol. 46, No. 2, pp. 370 – 379, 1999.

[60] Jesus Arellano-Padilla, *Fuzzy-Sliding Mode Control Approach to the Robust Position Control of Servo Drive Systems*, Ph.D. thesis, University of Nottingham, 2003.

[61] Z.H. Akpolat, G.M. Asher, and J.C. Clare, “A practical approach to the design of robust speed controllers for machine drives,” *IEEE Transactions on Industrial Electronics*, vol. 47, No. 2, pp. 315 – 324, 2000.

[62] M. Croft, G. Walker, and G. Hovland, "A new efficiency model for a regenerative dynamometer," *Australian University Power Engineering Conference*, vol. 1, No. 1, pp. 1 – 6, 2004.

[63] Mi-Ching Tsai, "Application of H_∞ control to improve the current and speed loops of switched reluctance motor drives," *Journal of Dynamic Systems, Measurement and Control*, vol. 123, No. 2, pp. 363 – 369, 2006.

[64] Wander G. da Silva, Paul P. Acamley, and John W. Finch, "On-line optimisation of a fuzzy drive controller using genetic algorithm," *IEEE International Symposium on Industrial Electronics*, vol. 2, No. 1, pp. 1441 – 1446, 2004.

[65] Nnamdi Okaeme, Pericle Zanchetta, and Mark Sumner, "Automated online design of robust speed digital controllers for variable speed drives," *IEEE Industrial Applications Conference, IAS*, vol. 1, No. 1, pp. 658 – 663, 2006.

[66] C. Benachaiba, S. Dib, and O. Abdelkhlek, "Genetic algorithm-based self-learning fuzzy pi controller for shunt active filter," *International Journal of Applied Engineering Research*, vol. 1, No. 1, pp. 203 – 216, 2006.

[67] Sidhartha Panda and N. P. Padhy, "Comparison of particle swarm optimizationand genetic algorithm for tcsc-basedcontroller design," *International Journal of Computer Science and Engineering*, vol. 1, No. 1, pp. 41 – 49, 2006.

[68] S. Mishra, "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Transaction on Evolutionary Computation*, vol. 9, No. 1, pp. 61 – 73, 2005.

[69] S. Mishra and C. N. Bhende, "Bacterial foraging technique-based optimized active power filter for load compensation," *International Journal of Computer Science and Engineering*, vol. 22, No. 1, pp. 457 – 465, 2007.

[70] Tai-Chen Chen, Pei-Wei Tsai, Shu-Chuan Chu, and Jeng-Shyang Pan, "A novel optimization approach: Bacterial-ga foraging," *Innovative computing, innovation and control conference*, pp. 391 – 394, 2007.

[71] Dong Hwa Kim, Ajith Abraham, and Jae Hoon Cho, “A hybrid genetic algorithm and bacterial foraging approach for global optimization,” *IEEE Transaction on Evolutionary Computation*, vol. 177, No. 2007, pp. 3918 – 3937, 2007.

[72] Dong Hwa Kim and Jae Hoon Cho, “A biologically inspired intelligent pid controller tuning for avr systems,” *International Journal of Control, Automation, and Systems*, vol. 4, No. 5, pp. 624 – 636, 2006.

[73] M. Srinivas and L. M. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms,” *IEEE Transactions on Systems, Man and Cybernetic*, vol. 24, No. 4, pp. 656 – 667, 1994.

[74] W. J. Tang, Q. H. Wu, and J. R. Saunders, “Bacterial foraging algorithm for dynamic environments,” *IEEE Congress on Evolutionary Computation*, vol. 1, No. 1, pp. 1324 – 1330, 2006.

[75] Leo Rollins, “Robust control theory,” Tech. Rep., Carnegie Mellon University, 1999.

[76] P. P. Khargonekar, “Control of uncertain systems using non-linear feedback,” Tech. Rep., University of Minnesota, 1989.

[77] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Company, London, 1990.

[78] J. Ackermann, P. Blue, T. Bunte, and L. Guvenc, *Robust Control: The Parameter Space Approach*, Springer-Verlag London Limited, London, 2002.

[79] P. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice Hall, Inc, London, 1996.

[80] M. Jamshidi, L. dos Santos Coelho, R. A. Krohling, and P. J. Fleming, *Robust Control Systems with Genetic Algorithms*, CRC press LLC, Florida, 2003.

[81] H. Kwakernaak, “Robust control and h-infinity optimisation - tutorial paper,” *Automatica*, vol. 29, No. 2, pp. 255 – 273, 1993.

[82] S. Toffner-Clausen, P. Anderson, and J. Stoustrup, "Robust control," Tech. Rep., Aalborg University, 2001.

[83] H. T. Toivonen, "Robust control methods," Tech. Rep., Abo Akademi University, 1998.

[84] J.G. Owen and G. Zames, "Robust $\langle_\infty$ disturbance minimization by duality," *IEEE Systems and Control Letters*, vol. 19, No. 4, pp. 255 – 263, 1992.

[85] T. Kumon, T. Suzuki, and M. Iwasakiand M. Matsuzakiand N. Matsuiand S. Okuma, "System identification using a genetic algorithm and its application to internal adaptive model control," *Electrical Engineering in Japan*, vol. 142, No. 4, pp. 45 – 55, 2003.

[86] E. Vladu, "Using Genetic Algorithms in System Identification," Tech. Rep., Department of Electrical Engineering and Information Technology, University of Oradea, 1992.

[87] T. Johnson and P. Husbands, "System identification using genetic algorithms," Tech. Rep., University of Sussex, Brighton, 1992.

[88] W. S. Levine, *The Control Handbook*, Prentice Hall, Inc, New York, 1996.

[89] M. T. Tham, "Introduction to robust control," Tech. Rep., University of Newcastle Upon Tyne, 2002.

[90] D. P. Solomatine, "Genetic and other global optimization algorithms - comparison and use in calibration problems," *International Conference on Hydroinformatics*, pp. 1021 – 1028, 1998.

Appendix A

Schematics

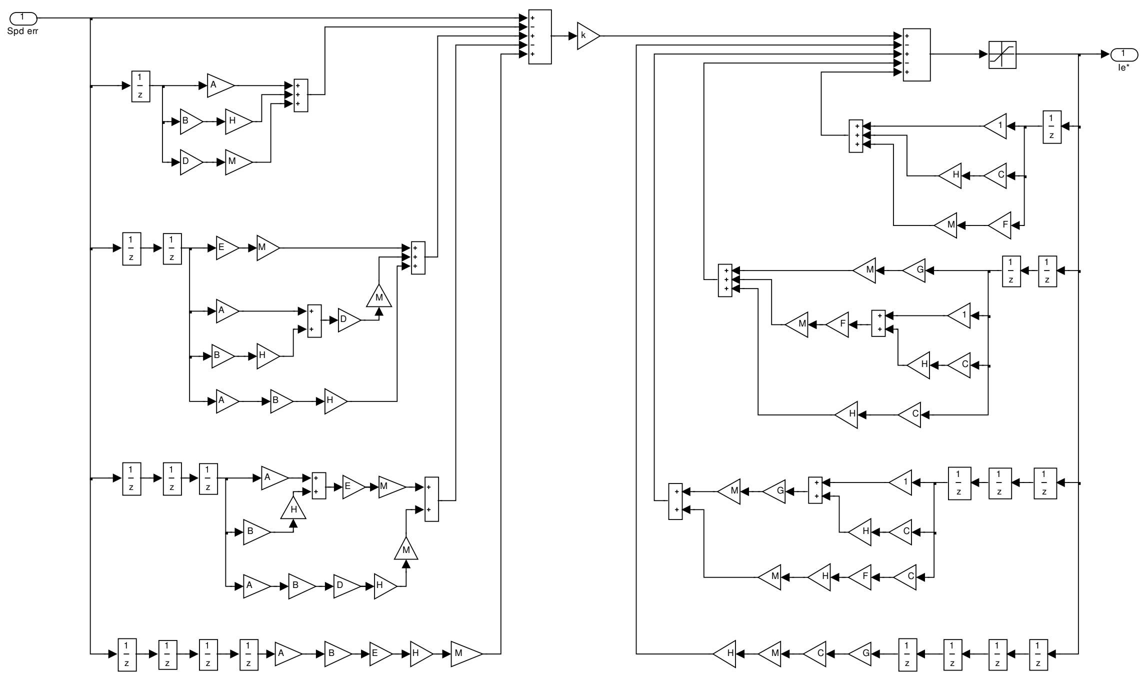
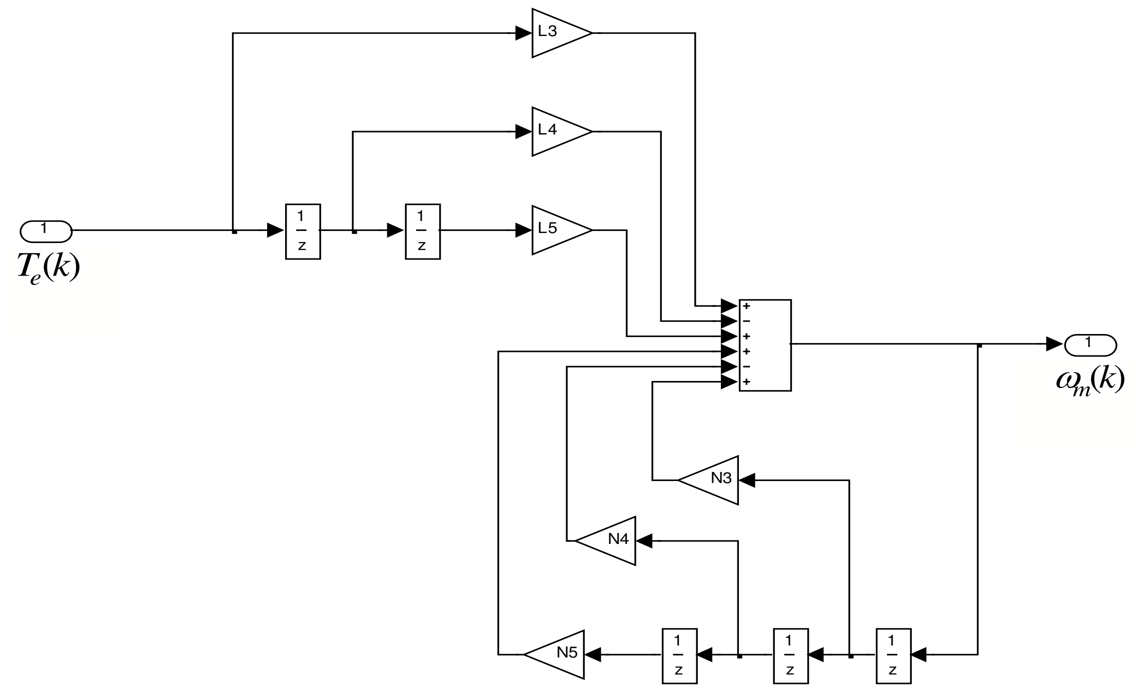
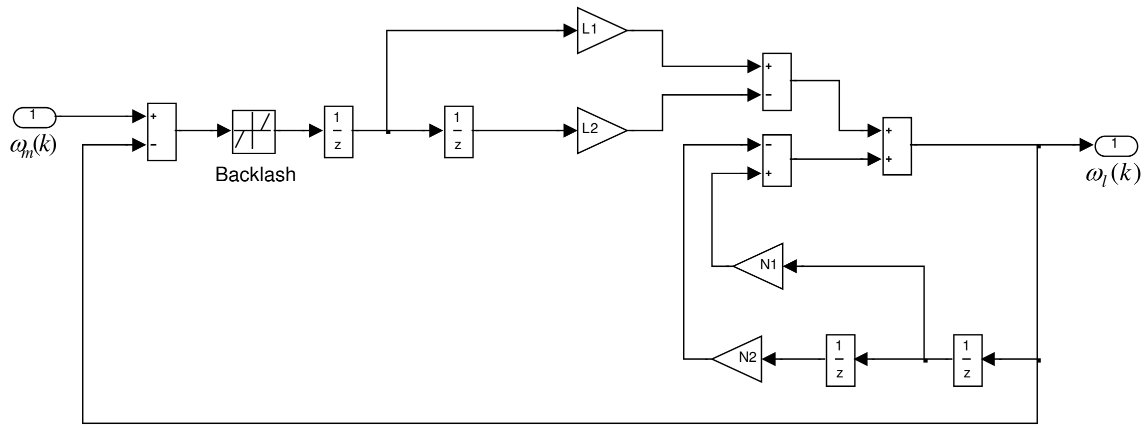


Figure A.1: Digital Controller Structure



(a) Transfer Function between Torque input and Drive Motor Speed



(b) Transfer Function between Drive and Load Motor Speed

Figure A.2: Flexible Shaft Mechanical Load Transfer Function

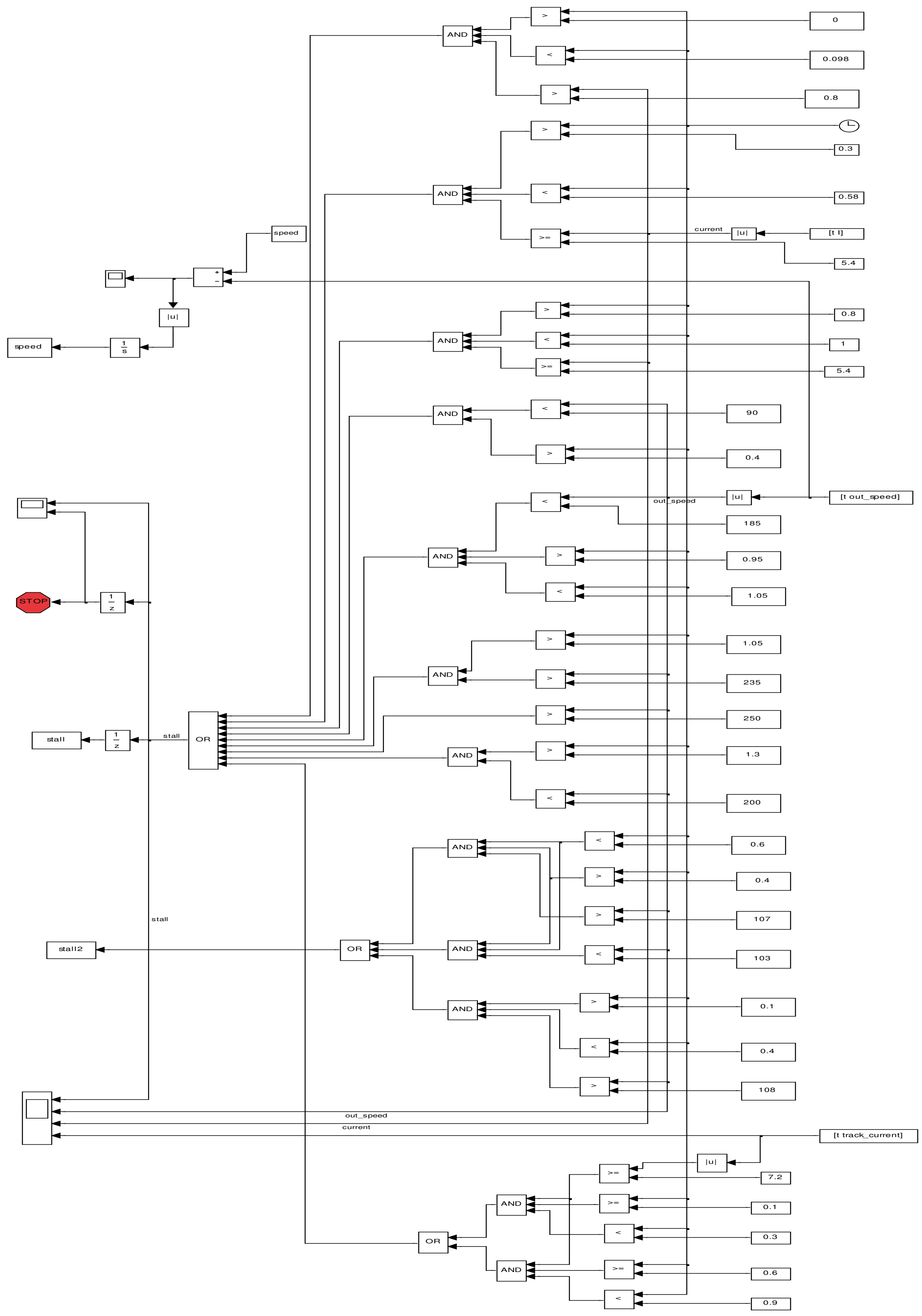


Figure A.4: Logic Protection Circuit

Appendix B

Robust Performance Conditions

B.1 Nominal Stability

For the unperturbed closed loop control system in figure 7.11, the closed loop transfer function is given by

$$\frac{CG_0}{1+CG_0} = \frac{L}{1+L}, \qquad where \qquad L = CG_0 \tag{B.1}$$

Using the Nyquist Stability Criterion, the internal stability of the nominal feedback system is determined by considering the Nyquist plot of its characteristic equation, (1+L). If the nominal feedback system is internally stable, two conditions will hold true [78]:

1. The Nyquist plot of (1+L) does not pass through the critical point of (-1,0)
2. The number of counterclockwise encirclements equals the number of poles of G_0 in the real axis of the s-plane, Re(s) ≥ 0 plus the number of poles of C in Re(s) ≥ 0

B.2 Robust Stability

This is simply the internal stability for each of the plants that exist within the uncertainty (perturbed) model. Using the Nyquist Stability Criterion, if the system has a nominal closed loop model that is internally stable, the system's robust stability is certain only if the perturbations introduced by the other plants within the uncertainty is less than one.

In the following sections, the robust stability of each uncertainty model will be derived by focusing on the modified characteristic equation in order to identify the perturbation introduced on the Nyquist plot of the internally stable nominal closed loop system

B.2.1 Additive Uncertainty

The perturbation model, G is represented as

$$G = G_0 + \Delta W_a \tag{B.2}$$

The perturbed transfer function for L thus becomes

$$GC = (G_0 + \Delta W_a)C \tag{B.3}$$

The key equation that considers the modified characteristic equation of (B.1) is

$$\begin{aligned} &= 1 + (G_0 + \Delta W_a)C \\ &= (1 + CG_0)\left(1 + \frac{\Delta W_a C}{1 + CG_0}\right) \\ &= (1 + L)(1 + \Delta W_a CS) \qquad \text{(B.4)} \\ \textit{where} \quad & L = CG_0 \ \textit{and} \ S = \frac{1}{1 + CG_0} \end{aligned}$$

From (B.4), the robust stability criteria of the additive uncertainty model is

$$\|W_a CS\|_\infty < 1 \tag{B.5}$$

B.2.2 Multiplicative Uncertainty

Perturbation model,G is represented as

$$G = (1 + \Delta W_m)\, G_0 \tag{B.6}$$

The perturbed transfer function for L thus becomes

$$\begin{aligned} GC &= ((1 + \Delta W_m)\, G_0)C \\ &= (1 + \Delta W_m)\, L \end{aligned} \tag{B.7}$$

The key equation that considers the modified characteristic equation of (B.1) is

$$\begin{aligned} &= 1 + (1 + \Delta W_m)\, L \\ &= 1 + L + \Delta W_m L \quad = \quad (1 + L)\left(1 + \frac{\Delta W_m L}{(1 + L)}\right) \\ &= (1 + L)(1 + \Delta W_m T) \\ where \qquad & T = \frac{L}{1 + L} \end{aligned} \tag{B.8}$$

From (B.8), the robust stability criteria of the additive uncertainty model is

$$\|W_m T\|_\infty < 1 \tag{B.9}$$

B.2.3 Feedback Uncertainty

First Model: The perturbation model, G is represented as

$$G = \frac{G_0}{1 + \Delta W_f} \tag{B.10}$$

The perturbed transfer function for L thus becomes

$$\begin{aligned} GC &= \frac{G_0}{1 + \Delta W_f} C \\ &= \frac{L}{1 + \Delta W_f} \end{aligned} \tag{B.11}$$

The key equation that considers the modified characteristic equation of (B.1) is

$$\begin{aligned} &= 1 + \frac{L}{1 + \Delta W_f} \\ &= \frac{1 + L + \Delta W_f}{1 + \Delta W_f} \quad = \quad \frac{(1 + L)\left(1 + \frac{\Delta W_f}{1 + L}\right)}{1 + \Delta W_f} \\ &= (1 + L)\left(\frac{1 + \Delta W_f S}{1 + \Delta W_f}\right) \end{aligned} \tag{B.12}$$

From (B.12), the robust stability condition for the considered feedback uncertainty is

$$\|W_f S\|_\infty < 1 \tag{B.13}$$

Second Model: The perturbation model, G is represented as

$$G = \frac{G_0}{1 + \Delta W_f G_0} \tag{B.14}$$

The perturbed transfer function for L thus becomes

$$\begin{aligned} GC &= \frac{G_0}{1 + \Delta W_f G_0} C \\ &= \frac{L}{1 + \Delta W_f G_0} \end{aligned} \tag{B.15}$$

The key equation that considers the modified characteristic equation of (B.1) is

$$\begin{aligned} &= 1 + \frac{L}{1 + \Delta W_f G_0} \\ &= \frac{1 + L + \Delta W_f G_0}{1 + \Delta W_f G_0} \quad = \quad \frac{(1 + L)\left(1 + \frac{\Delta W_f G_0}{1 + L}\right)}{1 + \Delta W_f G_0} \\ &= (1 + L)\left(\frac{1 + \Delta W_f G_0 S}{1 + \Delta W_f G_0}\right) \end{aligned} \tag{B.16}$$

From (B.16), the robust stability condition for the considered feedback uncertainty is

$$\|W_f G_0 S\|_\infty < 1 \tag{B.17}$$

B.3 Nominal Performance

The nominal performance of the unperturbed system in figure 7.11 is given as (B.18) or (B.19)

$$\begin{aligned} \|W_s S\|_\infty &< 1 \qquad (B.18) \\ \|W_t T\|_\infty &< 1 \qquad (B.19) \end{aligned}$$

$$where\ \ S = \frac{1}{1+CG_0} \qquad and \qquad T = \frac{CG_0}{1+CG_0}$$

For the Uncertainty models considered, their respective sufficient conditions for nominal performance will be derived by separately considering (B.18) and (B.19).

B.3.1 Additive Uncertainty

Nominal Performance according to (B.18)**:** Substituting (B.2) into the expression for S, gives the perturbed sensitivity function as (B.20)

$$\begin{aligned} &= \frac{1}{1+CG_0+\Delta W_a C} \\ &= \frac{1}{(1+CG_0)\left(1+\dfrac{\Delta W_a C}{1+CG_0}\right)} \\ &= \frac{S}{1+\Delta W_a CS} \qquad (B.20) \end{aligned}$$

Substituting (B.20) into (B.18), the nominal performance condition is

$$\left\|\frac{W_s S}{1+\Delta W_a CS}\right\|_\infty < 1 \qquad (B.21)$$

Nominal Performance according to (B.19)**:** Substituting (B.2) into the expression for T, gives the perturbed sensitivity function as

$$\begin{aligned}
&= \frac{C(G_0+\Delta W_a)}{1+C(G_0+\Delta W_a)} \\
&= \frac{CG_0\left(1+\dfrac{\Delta W_a}{G_0}\right)}{(1+CG_0)\left(1+\dfrac{\Delta W_a C}{1+CG_0}\right)} \\
&= \frac{T\left(1+\dfrac{\Delta W_a}{G_0}\right)}{1+\Delta W_a CS} \qquad \text{(B.22)}
\end{aligned}$$

Substituting (B.22) into (B.19), the nominal performance condition is

$$\left\| \frac{W_t T\left(1+\dfrac{\Delta W_a}{G_0}\right)}{(1+\Delta W_a CS)} \right\|_\infty < 1 \qquad \text{(B.23)}$$

B.3.2 Multiplicative Uncertainty

Nominal Performance according to (B.18)**:** Substituting (B.6) into the expression for S, gives the perturbed sensitivity function as

$$\begin{aligned}
&= \frac{1}{1+C\left(1+\Delta W_m\right)G_0} \\
&= \frac{1}{(1+CG_0)\left(1+\Delta W_m\left(\dfrac{CG_0}{1+CG_0}\right)\right)} \\
&= \frac{S}{1+\Delta W_m T} \qquad \text{(B.24)}
\end{aligned}$$

Substituting (B.24) into (B.18), the nominal performance condition is

$$\left\| \frac{W_s S}{1+\Delta W_m T} \right\|_\infty < 1 \qquad \text{(B.25)}$$

Nominal Performance according to (B.19)**:** Substituting (B.6) into the expression for T, gives the perturbed complementary sensitivity function as

$$\begin{aligned}
&= \frac{(1+\Delta W_m)CG_0}{1+(1+\Delta W_m)CG_0} \\
&= \frac{CG_0(1+\Delta W_m)}{1+CG_0+\Delta W_m CG_0} \\
&= \frac{CG_0(1+\Delta W_m)}{(1+CG_0)\left(1+\dfrac{\Delta W_m CG_0}{1+CG_0}\right)} \\
&= \frac{T(1+\Delta W_m)}{(1+\Delta W_m T)} \qquad \text{(B.26)}
\end{aligned}$$

Substituting (B.26) into (B.19), the nominal performance condition is

$$\left\| \frac{W_t T(1+\Delta W_m)}{(1+\Delta W_m T)} \right\|_\infty < 1 \qquad \text{(B.27)}$$

B.3.3 Feedback Uncertainty

Nominal Performance according to (B.18)**:** Considering the **First model**, defined by (B.10), and substituting the expression into that for S, gives the perturbed sensitivity function as

$$\begin{aligned}
&= \frac{1}{1+\dfrac{CG_0}{1+\Delta W_f}} \\
&= \frac{1+\Delta W_f}{1+CG_0+\Delta W_f} \\
&= \frac{1+\Delta W_f}{(1+CG_0)\left(1+\dfrac{\Delta W_f}{1+CG_0}\right)} \\
&= \frac{S(1+\Delta W_f)}{1+\Delta W_f S} \qquad \text{(B.28)}
\end{aligned}$$

Substituting (B.28) into (B.18), the nominal performance condition is

$$\left\| \frac{W_s S(1+\Delta W_f)}{1+\Delta W_f S} \right\|_\infty < 1 \qquad \text{(B.29)}$$

Nominal Performance according to (B.19)**:** Considering the **First model**, defined by (B.10), and substituting the expression into that for T, gives the perturbed complementary sensitivity function as

$$\begin{aligned} &= \frac{CG_0}{1+\Delta W_f + CG_0} \\ &= \frac{CG_0}{(1+CG_0)\left(1+\dfrac{\Delta W_f}{1+CG_0}\right)} \\ &= \frac{T}{1+\Delta W_f S} \end{aligned} \tag{B.30}$$

Substituting (B.30) into (B.19), the nominal performance condition is

$$\left\| \frac{W_t T}{1+\Delta W_f S} \right\|_\infty < 1 \tag{B.31}$$

Nominal Performance according to (B.18)**:** Considering the **Second model**, defined by (B.14), and substituting the expression into that for S, gives the perturbed sensitivity function as

$$\begin{aligned} &= \frac{1}{1+\dfrac{CG_0}{1+\Delta W_f G_0}} \\ &= \frac{1+\Delta W_f G_0}{1+CG_0+\Delta W_f G_0} \\ &= \frac{1+\Delta W_f G_0}{(1+CG_0)\left(1+\dfrac{\Delta W_f G_0}{1+CG_0}\right)} \\ &= \frac{S(1+\Delta W_f G_0)}{(1+\Delta W_f G_0 S)} \end{aligned} \tag{B.32}$$

Substituting (B.32) into (B.18), the nominal performance condition is

$$\left\| \frac{W_s S(1+\Delta W_f G_0)}{(1+\Delta W_f G_0 S)} \right\|_\infty < 1 \tag{B.33}$$

Nominal Performance according to (B.19)**:** Considering the **Second model**, defined by (B.14), and substituting the expression into that for T, gives the perturbed complementary sensitivity function as

$$\begin{aligned}
&= \frac{CG_0}{1+\Delta W_f G_0 + CG_0} \\
&= \frac{CG_0}{(1+CG_0)\left(1+\dfrac{\Delta W_f G_0}{1+CG_0}\right)} \\
&= \frac{T}{1+\Delta W_f G_0 S}
\end{aligned} \tag{B.34}$$

Substituting (B.34) into (B.19), the nominal performance condition is

$$\left\| \frac{W_t T}{1+\Delta W_f G_0 S} \right\|_\infty < 1 \tag{B.35}$$

B.4 Robust Performance

For each uncertainty model, the robust performance condition the nominal performance and robust stability conditions. In the following sections, *sufficient conditions* for robust performance for the considered uncertainty models will be summarised

B.4.1 Additive Uncertainty

For this uncertainty model, the robust performance conditions can be expressed as (B.36) or (B.37)

$$\|W_a CS\|_\infty < 1 \;\; \mathit{and} \;\; \left\| \frac{W_s S}{1+\Delta W_a CS} \right\|_\infty < 1 \tag{B.36}$$

$$\|W_a CS\|_\infty < 1 \;\; \mathit{and} \;\; \left\| \frac{W_t T \left(1+\dfrac{\Delta W_a}{G_0}\right)}{(1+\Delta W_a CS)} \right\|_\infty < 1 \tag{B.37}$$

B.4.2 Multiplicative Uncertainty

The robust performance conditions can be expressed as (B.38) or (B.39)

$$\|W_m T\|_\infty < 1 \;\; and \;\; \left\| \frac{W_s S}{1 + \Delta W_m T} \right\|_\infty < 1 \qquad \text{(B.38)}$$

$$\|W_m T\|_\infty < 1 \;\; and \;\; \left\| \frac{W_t T(1 + \Delta W_m)}{(1 + \Delta W_m T)} \right\|_\infty < 1 \qquad \text{(B.39)}$$

B.4.3 Feedback Uncertainty

First model: The robust performance conditions can be expressed as (B.40) or (B.41)

$$\|W_f S\|_\infty < 1 \;\; and \;\; \left\| \frac{W_s S(1 + \Delta W_f)}{1 + \Delta W_f S} \right\|_\infty < 1 \qquad \text{(B.40)}$$

$$\|W_f S\|_\infty < 1 \;\; and \;\; \left\| \frac{W_t T}{1 + \Delta W_f S} \right\|_\infty < 1 \qquad \text{(B.41)}$$

Second model: The robust performance conditions can be expressed as (B.42) or (B.43)

$$\|W_f G_0 S\|_\infty < 1 \;\; and \;\; \left\| \frac{W_s S(1 + \Delta W_f G_0)}{(1 + \Delta W_f G_0 S)} \right\|_\infty < 1 \qquad \text{(B.42)}$$

$$\|W_f G_0 S\|_\infty < 1 \;\; and \;\; \left\| \frac{W_t T}{1 + \Delta W_f G_0 S} \right\|_\infty < 1 \qquad \text{(B.43)}$$

Appendix C

Published Papers

The research project has resulted in the publication of the following papers:

- N. Okaeme, P. Zanchetta and M. Sumner, ”Robust Control Design through Experimental Load Identification for Variable Speed Drives,” *IEEE Industry Applications Conference, IAS*, pages 1257 - 1264, Sept. 2007. Submitted for publication on the **IEEE Transaction on Industry Applications**
- N. Okaeme, P. Zanchetta and M. Sumner, ”Robust Control Design through Experimental Load Identification for Variable Speed Drives,” *IEEE Industry Applications Conference, IAS*, pages 1257 - 1264, Sept. 2007.
- N. Okaeme, P. Zanchetta and M. Sumner, ”Automated Online Design of Robust Position and Speed Digital Controllers For Variable Speed Drives,” *IEEE Industry Electronics Conference, IECON*, pages 4719 - 4724 , Nov. 2006.
- N. Okaeme, P. Zanchetta and M. Sumner, ”Automated Online Design of Robust Speed Digital Controllers For Variable Speed Drives,” *IEEE Industry Applications Conference, IAS*, pages 658 - 663, Oct. 2006.