University of Southern Queensland

Faculty of Engineering & Surveying

Analysis and Development of Iterative Fast Model Control Strategies for Systems with Constraints

A thesis submitted by

Sarang Ghude

M.Eng.Tech, B.Eng

in fulfilment of the requirements of

Doctor of Philosophy

2011

Abstract

In this research, new fast model control strategies are developed and analysed. To avoid confusion with the competing directions taken by Predictive Control they have been named 'Iterative Fast Model Control (IFMC) Strategies'. It has been shown that new IFMC strategies deliver near time optimal performance and have finite settling time. IFMC strategies that include various state constraints are also developed. The system responses and the Lyapunov stability have been analysed.

The possibility of extending IFMC strategies for systems up to an n^{th} order is supported by development of IFMC strategies for 6^{th} to 11^{th} order systems. Application of IFMC in a real life situation of Aircraft Lateral Control has been studied. Successful implementation of IFMC in a third order ball and beam experiment demonstrates its effectiveness practice. A performance comparison with other contemporary strategies showed that IFMC delivers performance with almost 45% improvement.

The purpose of any real time controller is to determine the drive that should be applied to the plant at each instant. In IFMC, this is performed with the aid of a fast model of the system that can run at a speed that may be a thousand or more times that of the system. Then a decision on plant input is made, based on the fast model behaviour. The input is constrained at both extremes, full positive and full negative.

Fast Model Control strategies were developed from the mid 1950s, initially termed "Predictive control" although subsequently this term has been taken up by a different research thread. There have been hardly any publications extending the original theme since 1990. It is possible that the slow and expensive computing resources of that time may have led to the methods being regarded as purely academic. However the more recent proliferation of high speed, yet small affordable microcontrollers may have renewed the relevance of fast model control strategies.

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Sarang Ghude
0050036313

Signature of Candidate

Date

ENDORSEMENT

Signature of Principal Supervisor

Date

Signature of Associate Supervisor

Date

Acknowledgments

I consider myself fortunate to have had an opportunity to work with Prof. John Billingsley. He boosted my interest in research during my masters then helped me to become a better researcher through my PhD candidature. His guidance and support was not limited to just research but was extended to many other areas. I have learned a lot from him and I hope I'll continue to learn in future also. I am grateful for his guidance, support and encouragement. "Thanks, John".

I sincerely thank, Assoc. Prof. Paul Wen for his guidance during my studies. I am thankful to Faculty of Engineering and Surveying of USQ, for offering me scholarship during my candidature. I thank, Mr. Mark Phythian, Assoc. Prof. David Buttsworth and Mr. Steven Goh for offering me teaching assistantship from time to time. I also thank Mr. Dean Believeu for his technical support and friendship.

Playing guitar and cricket have provided much needed relaxation, thus, helping indirectly in my research. My guitar self-learning effort might have lost its vigour if the invaluable guidance and encouragement from Dr. Rabi Misra was not there. I will always cherish this 'musical' friendship. "Thanks a lot, Rabi". I have learned a lot about playing cricket, at our cricket club that made me a better cricketer. A thanks would go to all my friends at the club, Thanks Uni-Bushchooks. I extend my thanks to all of my past and present friends at USQ and from home.

Last but most important, I express my deepest gratitude to my parents for their endless love, support and encouragement. This thesis is dedicated to my parents.

Sarang Ghude

Contents

List of Figures

Notations and Definitions

T	Plant time
t	Model time
t_+	Predicted model time, to have all model variables positive with full positive drive applied
t_-	Predicted model time, to have all model variables negative with full negative drive applied
dtp	Plant steplength
dtf	Model steplength
onside	A stage where variables have the sign, same as the input
offside	A stage where variables have the sign, opposite to the input
primary variable	First variable (integral) in the cascade
subsequent variable	Remaining variables (integrals) in the cascade

Chapter 1

Introduction

New Fast Model Control strategies are derived for the control of higher order systems in which the input is constrained. To avoid confusion with the competing directions taken by Predictive Control they have been named 'Iterative Fast Model Control (IFMC) Strategies'. Analysis of the new IFMC strategies is carried out in terms of time optimal performance, accuracy in reaching target and delivery of a stable performance. Practical implementation of IFMC strategy in a third order ball and beam experiment illustrate that these strategies work in practice.

A performance comparison with other contemporary strategies showed that IFMC delivers performance with almost 45% improvement. These strategies are applicable to various areas including (but not limited to) robotics, aviation, space systems, mechatronic systems etc. Newly developed IFMC strategies for higher order systems establish some ground work to control very high order systems that may not exist today but can exist in future due to rapidly developing technologies.

This research pays particular attention to cascaded-integrator systems, also known as a chain of integrators. In such systems, the output of one integrator becomes the input to the next, forming a chain or cascade. This continues until the last variable is reached. For example in a third order motion control problem, the acceleration integrates the input, this is again integrated to give he velocity and once more to give position. The order and complexity of such systems will increase with the involvement of more and more variables.

A particular property of such systems is that for any initial conditions, if a constant drive is applied then all the state variables will at some future time assume the same sign as the input. This will be referred to as the variable being 'onside'. Otherwise if of the opposite sign to the input the variable is termed 'offside'. These concepts play a significant role in Iterative Fast Model Control.

In nearly all practical control systems, a nonlinearity is present in the form of an input limit. Actuators will saturate for a full positive or a full negative drive. Therefore, these input constraints are central to the IFMC strategies. Sometimes there are desirable limits that should be imposed on different states of a system, for example limits on velocity or acceleration of a system in a motion control problem. Therefore, IFMC strategies are developed to accommodate these types of state constraints as well.

1.1 Background

It is important to draw a distinction between Iterative Fast Model Control and studies that have been given the name 'Model Predictive Control'. Even though both strategies use plant models that predict system behaviour, the similarity ends there. The method of choosing a plant input differs totally between these strategies.

In Model Predictive Control, a fast plant model predicts the future plant behaviour when presented with a projected input function. At each instant,a finite horizon open-loop optimal control is solved online, using previous plant states as initial conditions and predicted plant states. A series of control inputs is generated, out of which only the first input is given to the plant. The process is repeated at every instant.

In contrast, Iterative Fast Model Control employs its fast model to predict future plant behaviour for a full positive drive and a full negative drive. For each sense of drive, the model time is observed after which all model variables will be onside (will have same sign as that of input). The two times are denoted as t_+ for full positive drive and t_- for full negative drive. A simple comparison between t_+ and t_- determines the input to be applied to the plant. If t_+ is the greater then the plant input is made fully positive, if t_- is the greater then the plant input is fully negative.

The concept of control using fast model predictions was first coined by Coales and Noton at Cambridge University in 1956. They considered determining a bang-bang input signal to be applied to a plant to obtain time-optimal control, through the use of a fast model. John Billingsley then extended the Coales and Noton technique to higher order systems and simplified it to obtain sub optimal control in his doctoral work in 1968 at Cambridge.

In this line of research, more work was reported by various researchers up until 1990. Evolution of this theme of research is outlined in detail in the Literature Review. It seems that this research theme has been 'forgotten' since 1990 as there are hardly any published papers thereafter. One possible cause of this oversight could be the need for rapid computation, which was very slow and expensive at that time involving large computing devices. Today a modest embedded microcontroller can perform the calculations.

1.2 Motivation and Hypothesis

Microcontrollers and computers have now become 'very fast' and small in size. They have reasonably low cost and continue to increase in efficiency. With new fast model control strategies that can be embedded in such microcontrollers to deliver control performances that are near time optimal, accurate and stable, this could just be the key to unlock an exciting research area.

The initial simulations of fast model control strategies showed that their settling time performance is better than recent linear control based strategies, that are designed to control cascaded integrator systems with input constraints. Therefore, new strategies that are improvement on the previous fast model control strategies can be developed.

It can be hypothesized that new fast model control strategies can be developed to deliver near time optimal control performance when applied to a cascaded integrator system with input and state constraints.

In the area of Fast Model Control research as far as previous work is concerned, some strategies have been proposed and then proved mathematically by various authors. These strategies have been extended only to a fourth order system.

Detailed evolution of Fast Model Control strategies is given in the Literature Review.

Therefore, there is a scope to develop new fast model control strategies that are improvements on the previous ones and can deliver near time optimal performance. The strategies can be developed to accommodate input constraints as well as those of states. Also, they could be extended to higher order systems. It can be shown experimentally that these strategies actually work in practice. These are some of the novel aspects of this research.

1.3 Research Objectives

The overall research goal is to develop fast model control strategies (Iterative Fast Model Control strategies) to control systems of arbitrarily high order and to show that the new strategies can work in practice. Specific research objectives are outlined as follows

- Develop new Iterative Fast Model Control Strategies with input constraints to control higher order cascaded integrator systems where 'higher order' starts at 3rd and extends as high as possible.
- Develop methods to include programmed state constraints in Iterative Fast Model Control strategies.
- Analyse the performance of new strategies in terms of settling time, accuracy in reaching settling point, working of strategies and lyapunov stability.
- Implement new strategies in real life experiments to show and demonstrate their applicability in practice.

1.4 Structure of the Thesis

This thesis is divided into nine chapters where Chapters 3 to 8 cover main the research findings, Chapter 1, 2 and 9 are Introduction, Literature Review and Conclusions. At the end of each chapter there is a brief summary.

This first chapter introduces the research. It gives background information and motivation for the research. It also outlines research objectives and the structure of thesis.

In Chapter 2 the literature that has been reviewed, is outlined. In the Literature Review, contemporary control strategies are reviewed first of all. Then, control strategies that specifically considered cascaded integrator systems are outlined. Some attention is also given to the use of Lyapunov stability methods in stability analysis. Finally, the evolution of Fast Model Control strategy is outlined in detail.

In Chapter 3 a new Iterative Fast Model Control strategy is proposed for a third order cascaded integrator system. The performance of the new strategy against its predecessors is analysed in detail in terms of settling time optimality and settling point accuracy.

In Chapter 4 the mechanism of Iterative Fast Model Control is explained and a mathematical analysis of the primary variable curve shows that mean value of plant input lies inside the range of +1 and -1. The strategy is then extended to fourth, fifth and sixth order systems. The usefulness of 'slugging' in removing overshoots and making systems stable is discussed. ('Slugging' is a deliberate plant-model mismatch introduced to make the model somewhat pessimistic.)

In Chapter 5 the proposed IFMC strategy is modified to accommodate state constraints. The results are presented using examples of third, fourth and fifth order systems. It is shown that state constraints can be included independently or in combination.

In Chapter 6 the IFMC strategy is extended to systems of even higher order by introducing limits on some model variables. It shown that the strategy can be made to work for systems up to 11^{th} order. From the successful results of simulations of very high order systems, a strategy for an n^{th} order system is proposed.

In Chapter 7 a Lyapunov Stabiltiy Analysis of the IFMC strategy is carried out. Examples of second and third order systems are used. It is shown that the second and third order systems are asymptotically stable.

In Chapter 8, IFMC strategies are discussed with their application to real life experiments. First a fourth order system of a Boeing 747-400 aircraft lateral control is considered and simulation results indicate that IFMC strategy would give good results. The IFMC strategy is then implemented in an actual third order ball and beam experiment successfully. This showed that IFMC strategies actually work in practice.

In Chapter 9 Conclusions of the research are presented and suggestions of directions for future work are outlined.

Chapter 2

Literature Review

2.1 Introduction

Most control system designs are based on the mathematical models of the plant. But, when put in practice, control algorithms face non-linear dynamics such as input constraints and state constraints. In most of the real time control system the input given to the system cannot exceed a certain limit that is full drive either positive or negative. Various states of the system may also require limitations on their values.

Goodwin, Seron and Dona (2005) explain input and state constraints, with two examples of automobile control and chemical process control. The acceleration and deceleration control of an automobile are associated with the available throttle displacement and the braking actions which have the maximum and minimum limits, which are input constraints. State variables such as acceleration and deceleration will be constrained to certain limits to prevent wheels from losing traction.

In chemical process control, valves have maximum displacement (when fully open) and minimum displacement (when fully closed) which likewise imposes input constraints, and in addition the state variables will be constrained for operational reasons (Goodwin, Seron & Dona 2005).

Robotic manipulator control has restrictions on the performance such as maximum torque that can be applied by a motor etc stated Valle, Tadeo and Alvarez

(2002). The problem of constraints in control system has been studied over many years (Bernstein & Michel 1995).

2.2 Existing control Strategies

Various control methods have been developed, many of which have fundamental differences. Both linear and non-linear methods received attention. Some of the more significant methods are

Linear Quadratic Regulator (LQR)

Fuzzy Logic Control

Adaptive Control

Model Predictive Control

Goodwin, Seron and Dona (2005) has stated that many of existing control methods can be classified into four categories based on the approach they take to deal with the constraints. These do not cover all the possibilities or methods but in general the approaches can be classified as follows

Cautious Approach: In this approach performance demands are reduced to avoid the constraints. So methods that are designed for unconstrained systems can be used but this may result in considerable loss of the achievable performance.

Serendipitous Approach: In this approach no special attention is given to the constraints and the control methods are based on unconstrained design philosophies. Sometimes they may give good results but they may have more adverse effects on performance measures such as closed loop stability. Examples include LQR methods, Fuzzy Logic Control and Adaptive Control.

Evolutionary Approach: In this approach control methods are based on unconstrained design philosophies and further extended by adding modifications and embellishments to avoid or minimize negative effects of constraints, ensuring that the performance goals are met. Examples include various forms of anti-wind up control, High Gain - Low Gain control, Piecewise linear control, Switching control.

Tactical Approach: In this approach the constraints are incorporated in the system design right from the beginning. Examples will be Receding Horizon Control also known as Model Predictive Control and Fast Model Predictive Control

2.2.1 Control methods: Unconstrained design philosophies

Most of the extensively studied control methods have taken the serendipitous approach. Many of these methods claim to give improved performance. Among these methods most widely studied control methods are as follows

Linear Quadratic Regulator (LQR): Linear Quadratic Regulator is the base strategy for many other strategies including some that take tactical approach in dealing with systems with constraints. LQR is in fact a solution to a convex; least squares optimization problem that has some attractive properties such as that the optimal controller automatically ensures a stable closed loop system, achieves guaranteed levels of stability, robustness and is simple to compute. (Lublin & Athans 1996)

Fuzzy Logic Control: Fuzzy logic control by its very name indicates that it uses a logic that is not precise in nature which is similar to human interpretation of certain events. In fuzzy logic the object has a value between 0 and 1 representing its 'belongingness' to a certain set.

A good survey of Fuzzy logic control and its design methodology is presented in (Lee 1990). Fuzzy logic control has been used in many applications such as nuclear power plant (Ramaswamy, Edwards & Lee 1993), speed control of a DC servo motor (Lin 1994), inverted pendulum control and control of an automotive engine (Chen, Lei & Lei 1997),(Vachtsevanos, Farinwata & Pirovolou 1993). The applicability of fuzzy logic control to every control situation has been questioned and it has been stated that fuzzy logic control is not effective in all the cases. (Abramovitch 1994),(Chen & Kairys 1993)

2.2.2 Control methods with input/state constraints

Some control methods have been developed that take tactical approach by considering constraints right from the start that is at the design stage. Others are more general studies that have been modified.

Adaptive Control: Adaptive control is quite an ambiguous term. The term 'Adaptive' can be interpreted in many ways. Normally an adaptive control approach would include updating the system model according to the variations experienced and then compute the control signal for the new model.

Monopoli (1975) proposed modification in the model reference adaptive control law for control of systems with hard input saturations. Subsequently, various adaptive control methods have been developed to control the systems (including discrete systems) with constraints (Annaswamy & Karason 1995),(Karason & Annaswamy 1994), (Ohkawa & Yonezawa 1982), (Payne 1986), (Wang & Sun 1992), (Zhang & Evans 1987). Adaptive pole placement control of a system in presence of input constraints, has been implemented (Feng, Zhang & Palaniswamy 1991).

Predictive Control: The term 'Predictive Control' was first used by Chestnut and Wetmore (1959) when they simplified the strategy proposed by Coales and Noton (1956). These strategies used a fast model of the plant that predicted the system behaviour through straightforward simulation of plant dynamics. Then, based on the predictions plant input was determined which was bang-bang in nature.

Later on, the term predictive control was 'borrowed' to describe a number of other techniques culminating in Model Predictive Control. In Model Predictive Control, however, the decision on input is based on evaluation of a cost function. At each instant, past states, predicted states are used to solve a finite horizon open-loop optimal control problem online. A series of inputs is generated where only first input is used and other are discarded. Then the whole process is repeated at the next instant.

In terms of choice of plant input both themes of predictive control are quite distinct. The work on the original theme of predictive control continued until late 1980's (Mo & Billingsley 1990). It is possible that due to the slow and expensive computing of 1970's and 1980's, the original predictive control became oblivion afterwards.

On the other hand, Model Predictive Control has continually been studied for many years (Morari & Lee 1999) and has become so well known that the term 'Predictive Control' now means Model Predictive Control. Therefore, to avoid confusion here, original predictive control strategies are referred as Fast Model Control strategies. Next is the overview of contemporary model predictive techniques and the evolution of fast model control is discussed towards the end of this chapter.

The model based predictive control is also known as **receding horizon control**. It had advantage over many other strategies in use in late 1980s and 1990s due to its ability to handle constraints (Garcia, Prett & Morari 1989),(Maciejowski 2002). Several different algorithms have been developed under the banner of model predictive control (Byun & Kwon 1988),(Pike, Grimble, Johnston, Ordys & Shakoor 1996). Some of them are Model Algorithmic Control (MAC); Generalized Predictive Control (GPC); Dynamic Matrix Control (DMC); Extended Horizon Adaptive Control (EHAC); Extended Predictive Self Adaptive Control (EPSAC) and Internal Model Control (IMC) etc.

Out of these algorithms, Garcia, Prett and Morari (1989) have proposed use of **Internal Model Control (IMC)** algorithm to control a system with constraints. Clarke, Mohtadi and Tuffs (1987*a*, 1987*b*) proposed **Generalised Predictive Control or GPC**, which has been extended to control, systems with input constraints (Tsang & Carke 1988) with simulation results of an application to chemical process (Baili, Zehquiang & Zhuzhi 2006) and as a basic algorithm for systems with input and output constraints (Dion, Dugard & Tri 1987).

Another version of model predictive control known as **Direct Matrix Control or DMC** has been used to handle the multivariable constrained control problems and it had tremendous impact on the industry (Morari & Lee 1999) DMC algorithm was then extended to control the constrained nonlinear systems (Li & Biegler 1988). A survey of model predictive control techniques that considered various constraints was presented by Mayne et al. (2000). In this survey their focus was on stability and optimality.

2.3 Control of Cascaded Integrator Systems

Cascaded Integrator Systems are also known as Chain of Integrators or Multiple Integrators. These may be termed Double or Triple Integrators depending on the order of the system but this dissertation also considers chains of much greater order. These are the systems of the form $\frac{d^n x}{dt^n} = u$. A popular example of a cascaded integrator system is of motion control.

In a third order motion control, three variables that can be considered are acceleration, velocity and position. The state space equations for this cascaded system can be written as

$$\dot{a} = u$$
$$\dot{v} = a$$
$$\dot{x} = v$$

One of the key properties of such systems is that all that states of the system eventually take the sign of the input. The issue of controlling such systems, seems to have been tackled in different ways. On the lines of nonlinear control, fast model control strategies were developed until late 80's. Early 90's appears to have marked beginning of linear control based strategies.

On the linear control side, Schimtendorf and Barmish (1980) first defined the control problem with terms 'null controllability'. It meant that a system is null controllable when a control input would bring a system (all states) from any initial condition to origin (i.e. 0 or 'null') in finite time. When the input was limited to value in the set set Ω in R^m, a term Ω-null controllability was used. Later on with any constrained input the issue was addressed as ANCBC - Asymptotic Null-Controllability with Bounded Controls.

Schmitendorf and Barmish (Schmitendorf & Barmish 1980) considered single input linear chain of integrators where control values belonged to a pre-specified set Ω in R^m. It was shown that a simple bounded function of a linear feedback cannot globally stabilize chain of integrators of higher order. In a separate study (Sussmann & Yang 1991) also showed that for a cascaded integrator system of order $n \geq 3$, control laws based on simple linear functions cannot achieve global stability.

Later, Teel (1992) showed that it is possible to control chain of integrators by using bounded control strategies that are linear near origin. The proposed strategy first transformed the cascaded integrator system to a form where all the integrators received input at the same time. This form later was named as Feed-forward form. Then the input was designed with nested-type saturation functions.

Sussmann, Sontag and Yang in 1994 (Sussmann, Sontag & Yang 1994) suggested a new method in which the cascaded plant was first transformed to a form similar to one considered by Teel. Then a suitable feedback was constructed by compositions and linear combinations of saturated linear functions. Two types of feedbacks were suggested which further involved calculation of some constants using a separate procedure. Only mathematical results were presented that showed that, to implement feedback laws for cascaded integrator systems, real analytic functions can be used.

A technique of adding one integration was introduced by Tsinias (1989) and Byrnes and Isidori (1989). Mazenc and Praly (1994) made the technique of adding one integration more efficient. A lyapunov design was proposed to drive a state feedback law for systems that were of the transformed form of cascaded systems considered by Teel. This form of multiple integrators where all the integrators receive input at the same time, was named as a feedforward form. However it was also stated that this mathematical design may not be efficient in practice.

Lin (1995), extended Teel's approach to include state constraints along with the input constraints. The state constraints considered were the magnitude saturation limit of state measurements.

On a slightly different note, Jankovic, Sepulchre and Kokotovic (1996) proposed a recursive design procedure, that relied on the construction of a Lyapunov function, to stabilize a nonlinear cascaded system. This system was a combination of both forms of systems, cascaded form and the feedforward form whereas Teel's approach required transformation of cascaded form to a feedforward form.

Johnson and Kannan (2003) proposed an improvement in the transformation process of Teel's approach, focusing on pole placement and behavior of poles. The simulation results showed improvement in settling time.

Zanasi and Morselli (2003) proposed a nonlinear control law with input constraints to control a third order cascaded integrator system only. This control law was based on the use of switching surfaces and it allowed minimum time trajectory tracking for various types of signals.

Marchand (2003) first extended the Teel's approach by introducing state dependant saturation functions instead of standard saturation function and showed an improvement in the results. It increased the convergence speed and reduced the transitory excursions of the states.

Marchand (2005) mentioned an approach of linear anti-windup compensation where a linear feedback was designed first by ignoring input nonlinearities. Then to minimize its effects a compensation feedback was added. However with reference to (Megretski 1996) it was stated that this approach was too complex for stability and robustness analysis.

It was also pointed out that approaches taken by Lin and Saberi (1993), Megretski (1996) and Grognard, Sepulchre and Bastin (2002) that removed the drawback mentioned in (Sussmann & Yang 1991), would be very expensive. It was because these approaches required tuning of a Riccati equation by an online adaptable parameter ϵ at each step. (Marchand 2005)

Merchand extended the nonlinear control law proposed in (Sussmann et al. 1994) by relaxing the limit on parameter ϵ and redesigning the saturation function. Performance of new control law was compared against the existing strategies by Teel (1992), Sussmann, Sontag and Young (1994), Megretski (1996) and Lin (1998). The results showed that for a third order system new control law gives improved performance.

Based on the basic idea given by Teel, Zhou and Duan proposed a few new control laws (Zhou & Duan 2007), (Zhou & Duan 2008), (Zhou & Duan 2009). The first law proposed by Zhou and Duan (2007) was based on $2^n d$ control law of Teel. It used $\frac{n+1}{2}$ nested-type saturation functions for an $n^t h$ order system with some free parameters to improve the system performance. The saturation functions were calculated differently than Teel.It was shown through simulation results that this control law gives better performance than other existing laws mainly proposed by Teel and Merchand.

Zhou and Duan proposed two other control laws in (Zhou & Duan 2008). The control laws proposed therein, did not require the co-ordinate transformation which was essential for approaches taken earlier by researchers following Teel's line of work. It used certain pre-defined stable polynomials in the calculation of control law. The saturation functions were state dependant. In both cases, present input was calculated using past input as well. The proposed second law, showed better performance than previous laws of Teel (1992), Sussman, Sontag and Yang (1994) and Marchand (2005).

Zhou and Duan (2009) then generalized the results of Sussman, Sontag and Yang (1994) and Johnson and Kannan (2003). The system transformation was first carried out similar to Teels approach. Then the transformed co-ordinates were used to construct control laws similar to the laws proposed earlier (Zhou & Duan 2008). Simulation results showed improved settling performance than that of Teel and Merchand approaches.

Recently, Gayaka and Yao (Gayaka & Yao 2011) proposed a backstepping based controller design where Teel's approach was used as a framework. This scheme did not require co-ordinate transformation, instead it was based on actual tracking error dynamics. A set of inequalities when satisfied, would bring the states of tracking error dynamics to a region where controller would be unsaturated, independent of initial conditions. Simulation results showed better performance than the control laws proposed by Zhou and Duan (2007, 2008, 2009), Marchand (2005) and a law for feedforward system proposed by Kaliora and Astolfi (2004).

Therefore, it can be concluded that, at least in terms of settling time, the control law proposed by Gayaka and Yao is by far the most efficient control law for a third order cascaded integrator system, that is based on linear control theory.

When the simulation was carried out with the Iterative Fast Model Control Strategy, proposed in the current research, with same third order system and initial conditions used by Gayaka and Yao, the response showed a settling time improvement of about 45%. In fact the earlier 1968 and 1987 fast-model control strategies outperformed that of Gayaka and Yao.

Meanwhile Blaha, Schlegel and Mosna (2009) proposed a strategy for time optimal control of a cascaded integrator system with state constraints where all the integrators except the last one, were constrained. It is stated that time optimal control of chain of integrators with bounded input leads to bang-bang control.

This strategy was based on theory of Grobner bases and used some polynomials that defined trajectories from initial to final states as a function of predefined time intervals. It is claimed that similar results did not exist for time optimal control of constrained system.

It would be seen that the proposed 'Iterative Fast Model Control strategy with state constraints' gives much better performance even with such constrained system (Bláha, Schlegel & Mošna 2009) and not only that the strategy can be extended to higher order as much as n^{th} order.

2.4 Lyapunov Stability

Lyapunov stability theory is an established method for the stability analysis of nonlinear systems. Lyapunov proposed two methods of stability analysis. These methods are explained in several texts. These include (LaSalle & Soloman 1961),(Hahn 1963),(Willems 1970),(DeRusso, Roy, Close & Desrochers 1998),(Khalil 2002),(Spong, Hutchinson & Vidyasagar 2006).

The second method of Lyapunov has become popular in stability analysis. This method can be summarized as follows: If a function of the state can be found (usually called as a Lyapunov function) which is positive definite and for which the first derivative is always negative as time tends to infinity, and if this function is only zero when the state is at the origin, then the system is globally asymptotically stable (Hahn 1963), (LaSalle & Soloman 1961).

Kaufman (1966*a*) used Lyapunov's second method or direct method in time optimal bang-bang control of a second order plant. The idea was to drive the initial variables error and error rate to origin in minimum time. This represented the property of asymptotic stability in general. Then by designing a Lyapunov function in terms of error and error rate the asymptotic stability was proved.

Kaufman (1966*b*) carried out the stability analysis of a control strategy where a plant model was used to predict future trajectory and adjust, according to the changes in the plant so that the system reaches origin. The plant was a second order plant with bang-bang input. Two time optimal switching curves were calculated for the model with input at both extreme ends i.e. ± 1.

These curves then defined switching functions for various plants. One plant was an exact match of the model and a Lyapunov function in terms of plant parameters, error and error rate was used. The designed Lyapunov function included error and error rate because the strategy was designed to bring both of them to 0 as then system approaches the origin. It was shown that the system was asymptotically stable.

In the Linear Control of cascaded integrator systems, most of the researchers that followed Teel's approach and suggested improvements, used Lyapunov's direct method to prove the stability for respective control laws (Sussmann & Yang 1991), (Mazenc & Praly 1994), (Lin 1995),(Marchand 2003, Marchand 2005), (Zhou & Duan 2007, Zhou & Duan 2008, Zhou & Duan 2009) to name a few. In general the Lyapunov function V was designed in terms of square of the first integral so that the first derivative $\dot{V}$ consisted input function or the control law.

2.5 Evolution of Fast Model Control Strategies

The use of a fast model of a system to control the actual system was proposed by Ziebolz and Paynter (1953). The first paper that described a practical predictive automatic controller was published by Coales and Noton (1956). The strategy sought, time-optimal control of a second order system by applying full drive in one sense or the other, to both plant and model. The model drive is switched at a time that is iteratively corrected to bring the trajectory through the origin. Thus the time is predicted when the plant input must be switched.This strategy can be outlined as

Step 1: Assign current plant states to the states of the fast model.

Step 2: Run the fast model with input at one extreme (+1 or -1) with a single switch to other extreme.

Step 3: After input switch run the model until velocity is zero and then inspect the position.

Step 4: Depending on the error of position from the state origin, correct the switching time also if required reverse the initial drive direction. So that the model trajectory would pass through the origin.

This type of control was called 'bang-bang' control in USA, 'schwarz-weiss' (black-white) in Germany and on-off in UK Coales and Noton (Coales & Noton 1956). Coales and Noton (1956) demonstrated that ideal switching time can be predicted to achieve time-optimal control. This strategy was then greatly simplified by Chestnut and Wetmore (1959), to propose controllers for second order systems. These techniques were developed for time-optimal control of a plant. The time optimality proof for Chestnut Strategy is set out in (Billingsley 1968). This Chestnut strategy, for a second order plant $\ddot{x} = u$ where $|u| \leq 1$ can briefly be described as

Step 1: Set the model states corresponding to the plant states.

Step 2: Bring the model velocity to zero by applying a constant drive (+1 or -1).

Step 3: Inspect the model position.

Step 4: If position $x > 0$ then change the model drive to negative, if position $x < 0$ then change the model drive to positive.

Later, these two techniques were combined by Adey, Billingsley and Coales (1966) to give a new strategy for the time optimal control of third order plant, proof for the time optimality of this (Adey, Billingsley & Coales 1966) strategy is stated in (Billingsley 1968). The simplified strategy proposed by Chestnut and Wetmore (1959) was generalized for higher order plants by Gulko and Kogan (Gulko & Kogan 1963) and the proof of optimality for new strategies was given by Fuller (Fuller 1973).

Billingsley (1968) proposed a suboptimal fast model control strategy for a third order plant in which, first plant states were assigned to model states. Then the model was run ahead in time with input that was opposite to sign of first integral, until model's first integral attained same sign as that of input. At this point sign of second integral was checked, if same as input, then the model drive was reversed and model was run again until second integral attained same sign as input. Then the plant input drive was set to opposite sign of model position or last integral. And the process was repeated.

Dodds (1981) proposed a somewhat different strategy using a fast model of the plant. At first, model states were initialized with plant states. Then the model was run ahead in time with each sense of drive and the response of the 'position' that is the last integral of the model was observed. A comparison was carried out to determine which 'position' response had higher sign changes until no further change of sign in model could occur. The plant input was a drive corresponding to drive of 'position' response with higher sign changes.

Just as Dodds speculated in (Dodds 1981); the results of this strategy and Billingsley's 1968 strategy were found to be almost the same. Therefore for further analysis and comparison with new Iterative Fast Model Control Strategy, Billingsley's 1968 strategy is used. A performance comparison of both Dodds's 1981 and Billingsley's 1968 strategy is shown in Appendix B.

Billingsley (Billingsley 1987) proposed a new strategy in which switches used in the earlier fast model control strategies were removed. In this strategy the model was first set to the plant state then it was run simultaneously with full positive drive and full negative drive. Then, the input to the plant was decided from the time taken by the model acceleration and velocity to come onside where onside means that the input drive and model state have the same sign and by inspecting the corresponding 'position'.

This fast model control strategy was then extended to control multivariable systems by Mo and Billingsley (1990). Since then, this fast model predictive control strategy has not received much attention perhaps due to slow and expensive computing. It has been hard to gain access to some other research work in this theme, mainly by E. P. Ryan, N. C. Megson etc because it preceded the era of electronic publishing, and has limited access.

2.6 Conclusion

Several control strategies have been proposed and developed by a number of researchers across the world. Many of them are based on different philosophies. Some strategies, based on linear control, have been proposed to control cascaded integrator systems with input/state constraints. These strategies mostly considered a third order cascaded integrator system only. Two of them, tried to include state constraints. In general, the treatments were found to be mathematical rather than practical.

In the case of nonlinear control, fast model control strategies were developed from the mid 50's to late 80's to control cascaded integrator systems with input constraints. However, it seems that further development of such strategies was limited by the expensive and slow computing of that time. Therefore, for the control of higher order cascaded integrator systems with different constraints, there is a scope to develop control strategies that are more efficient and actually work in practice.

Chapter 3

Iterative Fast Model Control

3.1 Introduction

Recent and predicted advances in computing power imply that it has become possible to embed computationally intensive control algorithms in low-cost devices, giving ever improved performance. A new Iterative Fast Model Control Strategy has been proposed which can deliver near time optimal control of high-order systems with constrained inputs.

The new strategy is based on fast model control strategies that were once forgotten. It is important to show the evolution of the new strategy from its predecessors and to demonstrate its improvement. Therefore, earlier fast model control strategies from (Billingsley 1968) and (Billingsley 1987) are used for performance comparison. It is shown that these strategies give improved performance over the contemporary linear control based strategy (Gayaka & Yao 2011) at least in terms of settling time, results are shown in appendix A.

To determine the time optimal settling time performance of the new strategy, simulations and a practical experiment have been carried out. For comparison, the optimal settling times are determined using Pontryagin's principle. Also, analysis has been carried out to determine the accuracy with which all states reach the settling point, when subjected to the quantisation of discrete time control and numerical resolution.

3.2 Previous Fast Model Control Strategies

In earlier fast model control strategies, system responses were predicted with full positive drive and full negative drive, separately. The determining factor was the future time at which the the model was 'offside', meaning that, the input drive and the model position were of opposite signs.

3.2.1 1968 Fast Model Control Strategy

The strategy proposed by Billingsley (1968) for a third order plant $\dddot{x} = u$ can be summarized as

Step 1: Set the model states corresponding to the plant states.

Step 2: Set the model drive u_m to -sgn(a) where a is model acceleration.

Step 3: Run the model until $u_m * a >= 0$.

Step 4: At this point if $u_m * v_m > 0$ (where v_m is model velocity) then reset the model, reverse the model drive and continue,

Step 6: Until $u_m * v_m > 0$ then set the plant drive to $-sgn(x)$ where x is the position.

Step 7: Return to 1 for the next iteration.

The simulation of the strategy is shown in Fig. 3.1, the initial conditions are position $x = 10$, velocity $v = 0$ and acceleration $a = 0$. The steplength for model and the plant is same 0.01.

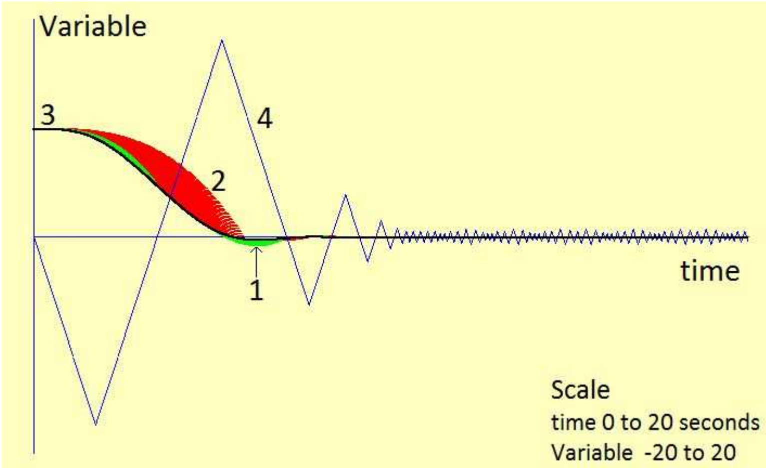


Figure 3.1: 1968 Fast Model Predictive Control Strategy Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

In this simulation, the green (lighter) and red (darker) traces are the predictions with full positive and full negative drive respectively. The curve-4 shows the acceleration. The values of acceleration are plotted with a coefficient of 10 for better visibility. The curve-3 (mostly through the traces) shows the actual path taken by the plant. The simulation shows slight overshoot near the arrowhead of 1, but the system is stable otherwise.

3.2.2 1987 Fast Model Control Strategy

The strategy proposed by Billingsley (1987) in 1987 can be outlined as follows and note that it does not use switches,

Step 1: Set the model states corresponding to the plant state.

Step 2: Run the model with full positive drive. Note the value of model time at which the model acceleration or velocity have negative sign. Record the model position at this instant.

Step 3: Run the model with full negative drive. Note the value of model time at which the model acceleration or velocity have positive sign also record the model position at this instant.

Step 4: Whichever value of offside time is greater apply drive opposite to the corresponding position. Restart the cycle.

The Simulation of the strategy with the same steplength for model and plant as 0.01 and the initial conditions as position $x = 10$, velocity $v = 0$ and acceleration $a = 0$ is shown in Fig. 3.2.

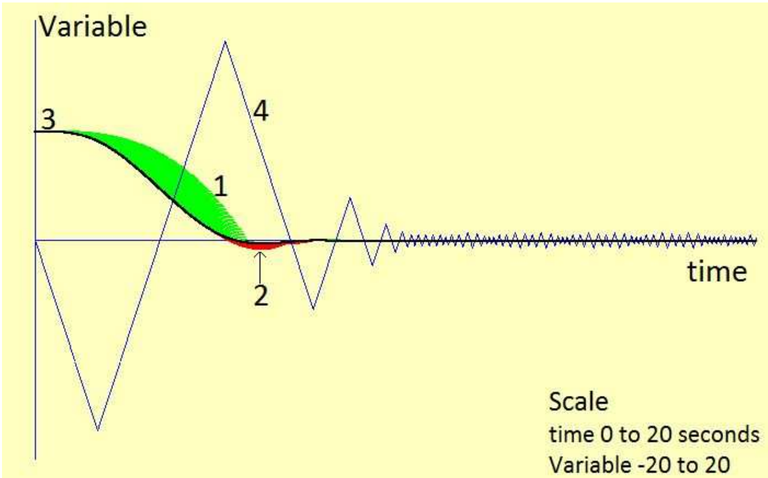


Figure 3.2: 1987 Fast Model Predictive Control Strategy Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

A small overshoot can be seen in Fig. 3.2 as well, near the arrowhead of 2. The simulation results show that the performance of both 1968 (Fig. 3.1) and 1987 (Fig. 3.2) fast model control strategies is similar.

3.3 New Iterative Fast Model Control Strategy

In the new Iterative Fast Model Control strategy, the model is considered to be offside if any of the states have a sign different from that of the input. An algorithm of the new strategy can be outlined as follows:

Step 1: Set the system model to the plant state

Step 2: Run the model with full positive drive. Note the time t_+ when all the model states come onside.

Step 3: Reset the system model to the plant state

Step 4: Run the model with full negative drive. Note the time t_- when all the model states come onside.

Step 5: Compare the times t_+ and t_- and whichever is greater apply the corresponding drive to the system. Go to Step 1

The simulation of the strategy is shown in Fig. 3.3, again the model/plant steplength and initial conditions are same as before, position $x = 10$, velocity $v = 0$ and acceleration $a = 0$.

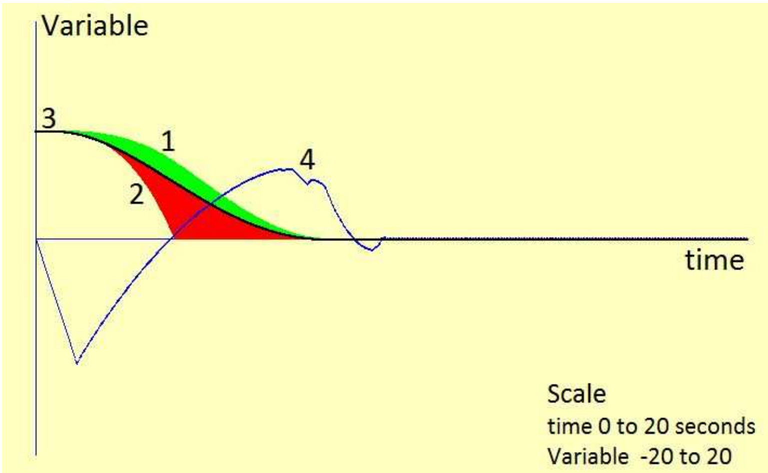


Figure 3.3: Iterative Fast Model Control Strategy Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

Fig. 3.3 shows that there is no overshoot in the system response. In this simulation, curve-1 and curve-2 indicate that the new Iterative Fast Model Control strategy is cautious and hence will deliver more stable and less oscillatory performance than its predecessors.

3.4 Time Optimality Test

Time optimal control is desirable for many control systems. Fast Model Control can be used to achieve near time optimal control performance. According to Pontryagin et al. (1963) if the input to an n^{th} order system is switched $n - 1$ times between full positive drive and full negative drive, then the time taken by the system to reach to a destination will be optimal.

To assess the degree of time optimality of the Iterative Fast Model Control, the system is taken backwards in time to a certain location noting the time t_bk and then the Iterative fast model control algorithm is used to bring it back to home noting the time of travel t_ret. The comparison of t_bk and t_ret would verify the time optimality of the algorithm.

A third order cascaded integrator system is considered to assess the performance of the Iterative Fast Model Control strategy. The initial values of the states of the system are considered as zero.

Then the system is taken backwards in time for 5 seconds. The input is switched two times between full positive drive and full negative drive to reach to a destination, then the values of the new destination states are noted. Taking these new values as initial conditions the system is then run with the new Iterative Fast Model Control Strategy to return to the initial starting position.

The system can be modelled as $\frac{d^3x}{dt^3} = u$ where $u = \pm 1$. As this is a third order system, the input is switched twice at $t_1 = 1$ and $t_2 = 3$ while going backwards in time. At the new destination, the initial conditions for the strategy are found as position $x = -12.337451$, velocity $v = 4.5551$ and acceleration $a = -1.010001$. The steplengths for model and plant are considered same, 0.01. The simulation of the system is shown in (Fig. 3.4).

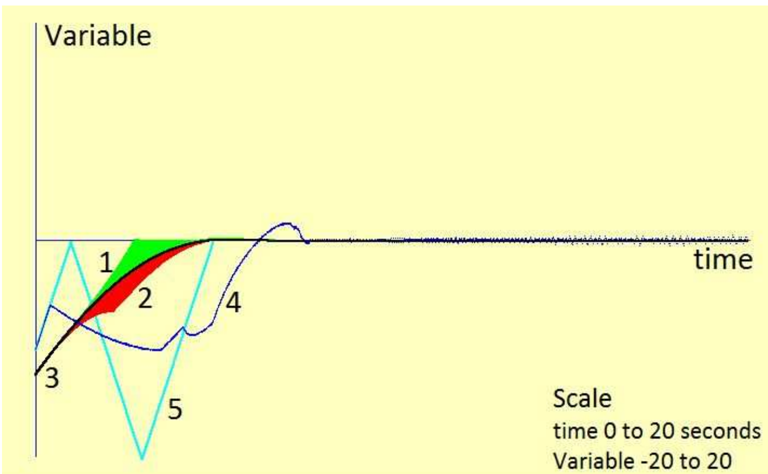


Figure 3.4: Time optimality test result of Iterative Fast Model Control with Third Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a), 5: 10*Backward journey of acceleration (a) from $x = 0$,$v = 0$ and $a = 0$

The curve-5 in Fig. 3.4 shows the backward journey of acceleration a and the curve-4 show the journey of acceleration a with Iterative Fast Model Control. The backward journey of position x is hidden behind the predictions with full positive and full negative drive. Curve-3 shows the path taken by position x to return to initial starting point.

From Fig. 3.4 it can be observed that the strategy takes approximately just over $(\frac{1}{3})^{rd}$ of 20 i.e. 7 seconds to return to starting point or home. The resulting ratio of travel-to-starting-point time t_{ret} against the optimal time of 5 sec i.e. t_{bk} appears to be around 1.4. A more vigorous time optimality test is carried out ahead.

As seen in the simulation, the acceleration enters a limit cycle or dithers, after system reaches initial starting point. Due to the quantization effects, the system will never reach to the exact starting point, hence resulting in dithering.

A less precise arrival at the origin can however be the decided by redetermining the error window around the starting point or home. The error for verifying arrival at the starting point can be calculated as follows,

$$error = x^2 + \dot{x}^2 + \ddot{x}^2 \tag{3.1}$$

and the condition such as $error \leq 0.0001$ will allow the system to settle down at certain states. This will leave a small window of error defined by the error condition where states will be very near to 0 but not at exact 0.

When the system or plant enters in this window it will be considered to have 'arrived' at home. The error can be calculated by any method but it is the 'error condition' that determines the arrival at home. It will be shown ahead that the error condition can be reduced up to a value of 10^{-10}.

One thing to note here is that this error is not calculating the deviation of each variable from its starting value that is 0 but it is defining a region around the starting point by adding squares of each variables. Therefore, even if the $error \leq 0.0001$, each individual variable may or may not be less than 0.0001 but it will be very near to 0. The goal of the strategy is to get the plant in this region in near optimal time.

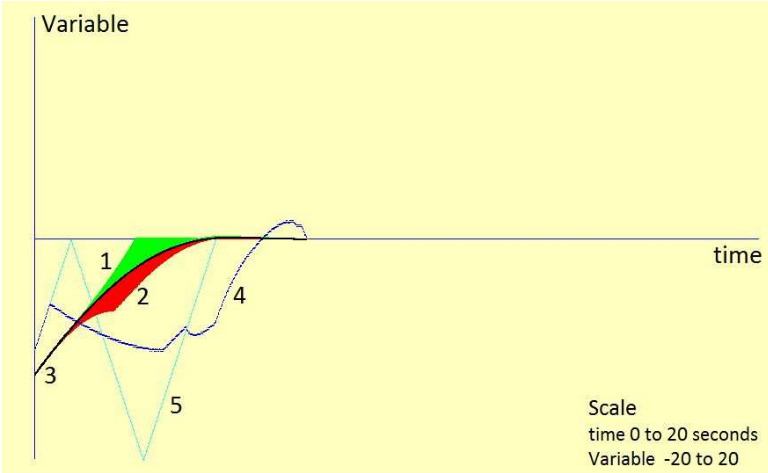
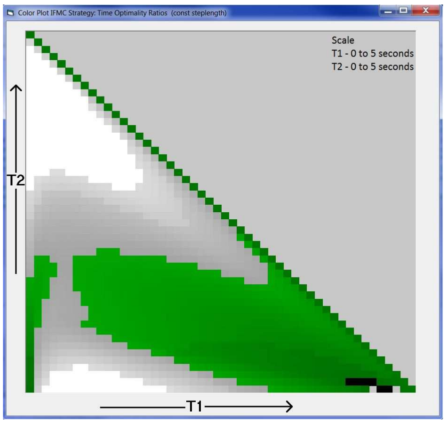


Figure 3.5: Iterative Fast Model Control with Third Order System, termination using error condition. Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a), 5: 10*Backward journey of acceleration (a) from $x = 0$,$v = 0$ and $a = 0$

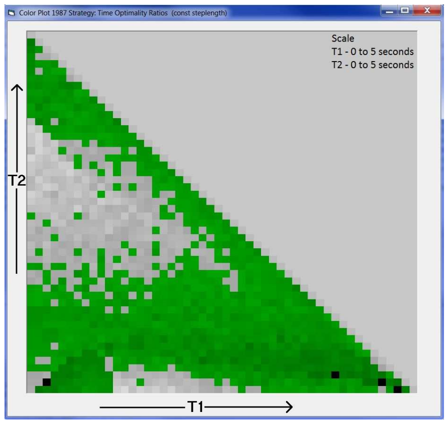
Fig. 3.5 shows the termination upon arrival at home. The final values of the individual variables are found as $x = -0.000246$, $v = 0.0026$ and $a = 3 \times 10^{-16}$. The time optimality ratio is found to be 1.5. The time optimality test is then performed on a greater scale to evaluate the performance of Iterative Fast Model Control. Its performance is then compared with the performance of previous fast model control strategies.

The test is carried out by considering all possible combinations of switching times between 0 and 5 seconds at intervals of 0.1 seconds to go backwards. For example ($t_1 = 1$ and $t_2 = 3$) or ($t_1 = 0.1$ and $t_2 = 1.2$) etc. The measure of time optimality is the ratio of, time taken to return to initial starting position and the backward travel time (5 seconds).

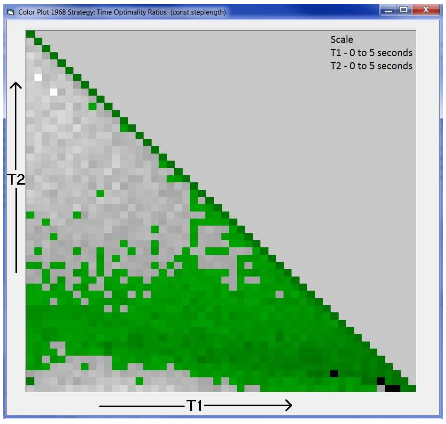
The system is considered as 'arrived to initial position' when the error condition is satisfied. The simulations for all three strategies (1968, 1987 and IFMC) are carried out. The resulting values are plotted as a colour point with switching times t_1 and t_2. t_1 values (0 to 5 step 0.1) are along x-axis and t_2 (0 to 5 step 0.1) along y-axis and each colour point inside the triangle represents corresponding time optimality ratio. In these simulations the error condition is $error \leq 0.0001$ and the model/plant steplengths are 0.01.



(a) Time Optimality Ratios for IFMC



(b) Time Optimality Ratios for 1987 Strategy



(c) Time Optimality Ratios for 1968 Strategy

Figure 3.6: Color Plots of Time Optimality Ratios

In the Fig. 3.6a, Fig. 3.6b and Fig. 3.6c inside the triangle the darkest (Black) points show the ratios below one. The darker (Green) points show the ratios between 1 and 1.5. The Gray points show the ratios between 1.5 and 2. The White points show the ratios over 2.

3.5 Simulation Results and Analysis

From Fig. 3.6b it can be seen that the most of the time optimality ratios for 1987 strategy are close to 1 (largest darker area), followed by 1968 strategy (Fig. 3.6c) whereas for IFMC strategy some of the time optimality ratios are over 2 (a couple of white areas) Fig. 3.6a. It is because in IFMC strategy all three variables are considered to be onside before making a decision on, which drive to apply whereas in 1987 and 1968 strategies only two variables are considered.

Thus, IFMC strategy gives more cautious control. Hence, in case of a fourth or higher order systems, IFMC strategy is likely to give more stable performance than 1987 strategy. The time optimal performance of 1968 strategy is inferior to 1987 strategy hence it is not considered for a fourth order system.

A comparison of Fig. 3.7 with Fig. 3.8 shows that for a fourth order system the new Iterative Fast Model Control strategy gives a more stable performance than the 1987 strategy in which switching times are in a converging geometric series.

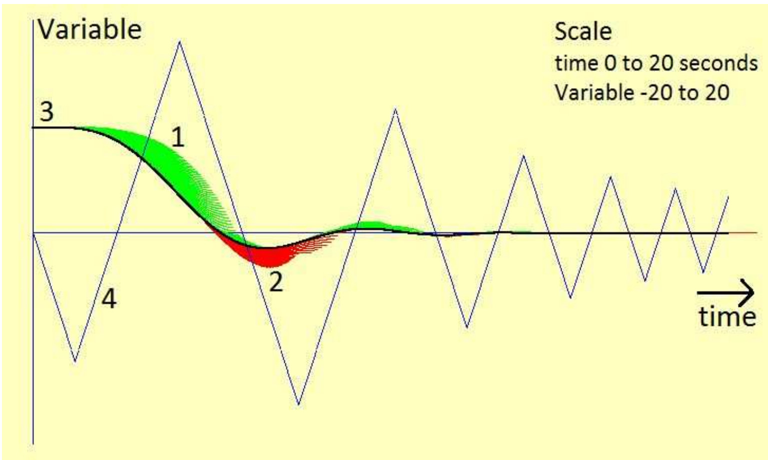


Figure 3.7: Fast Model Control Strategy 1987 - Fourth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Jerk (j)

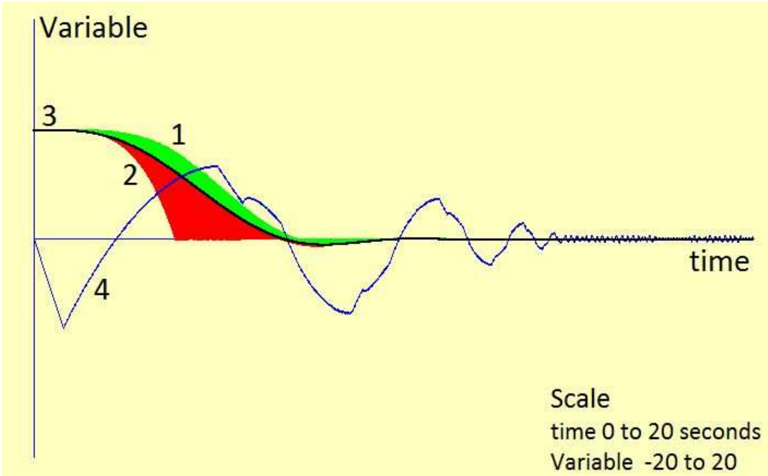


Figure 3.8: Iterative Fast Model Control - Fourth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Jerk (j)

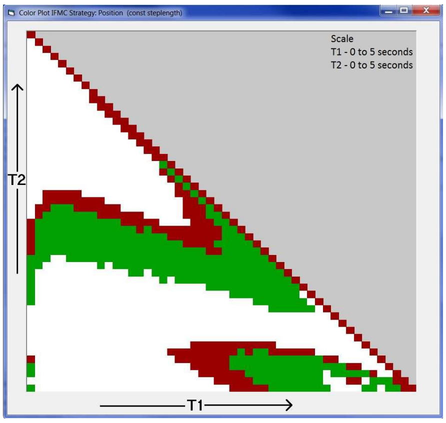
3.6 Accuracy in achieving target

Along with the time optimality of the strategy, it will be of interest to assess the performance of the strategy in terms of reaching the position of the starting point. As the position is derived from the velocity and the velocity is derived from the acceleration that receives the input, it is important to get the position nearest to the settling point's position (here-on target) first. The colour plots for position from all three strategies are shown in Fig. 3.9.

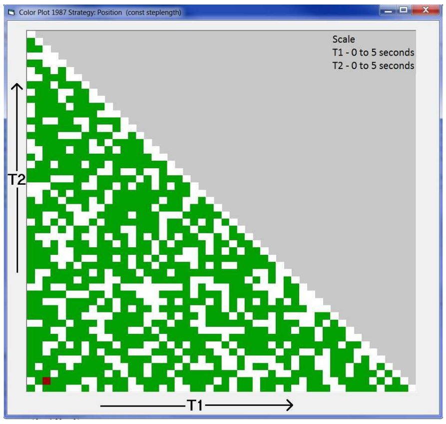
In these plots, the darker gray (red) points represent values between 0.001 and 0.0001, the lighter gray(green) points represent values between 0.0001 and 0.000001. The white points show the values that are less than 0.000001 or even smaller which can be considered as 0. It can be clearly seen that the IFMC strategy gets much closer to the target value than the other strategies.

The proximity to the target position is influenced by the steplength and the error condition. In this case, the target position is $x = 0$, and the two large white areas in the Fig. 3.9a show state values for position that are zero. It also shows that many values are above the limit.

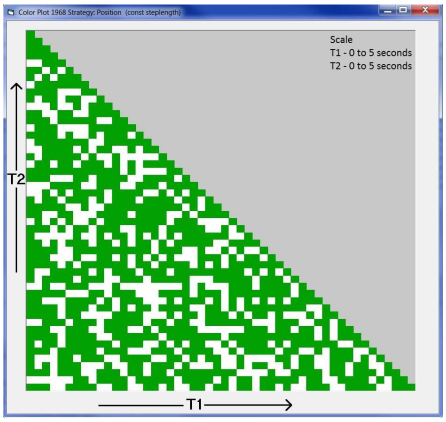
This issue is analogous to the hill-climb. If longer steps are taken for hill climbing then there is a good chance that the top of the hill will be missed. Thus, to reach



(a) Final values of position for IFMC Strategy



(b) Final values of position for 1987 Strategy



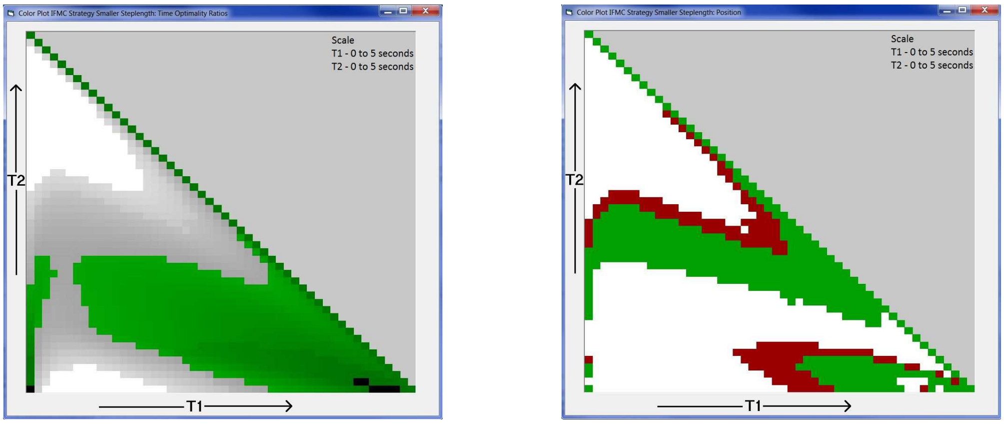
(c) Final values of position for 1968 Strategy

Figure 3.9: Final values of State (position) for different strategies

to the top, it becomes necessary to reduce the step-length as the distance to the top becomes smaller and smaller.

Similarly, if the steplength of the plant is reduced as the target gets closer and closer, it is possible to get near optimum proximity to the target. The exact target values will never be reached due to the quantization effects. Thus, the steplength for the plant is reduced by dividing 'the time taken by the model to come onside' by 1000 as the target is approached.

In simulation it did not show significant change in the time optimality ratios (Fig. 3.10a) than before (Fig. 3.6a). However, the final values for the variable position improved by many values going less than 0.0001. Fig. 3.10b shows that, by many darker gray (red) points in Fig. 3.9a have turned lighter gray (green) which indicates that more values of position are closer.



(a) Time Optimality Ratios (b) Final values of position

Figure 3.10: Color Plots IFMC Strategy with Smaller Steplength

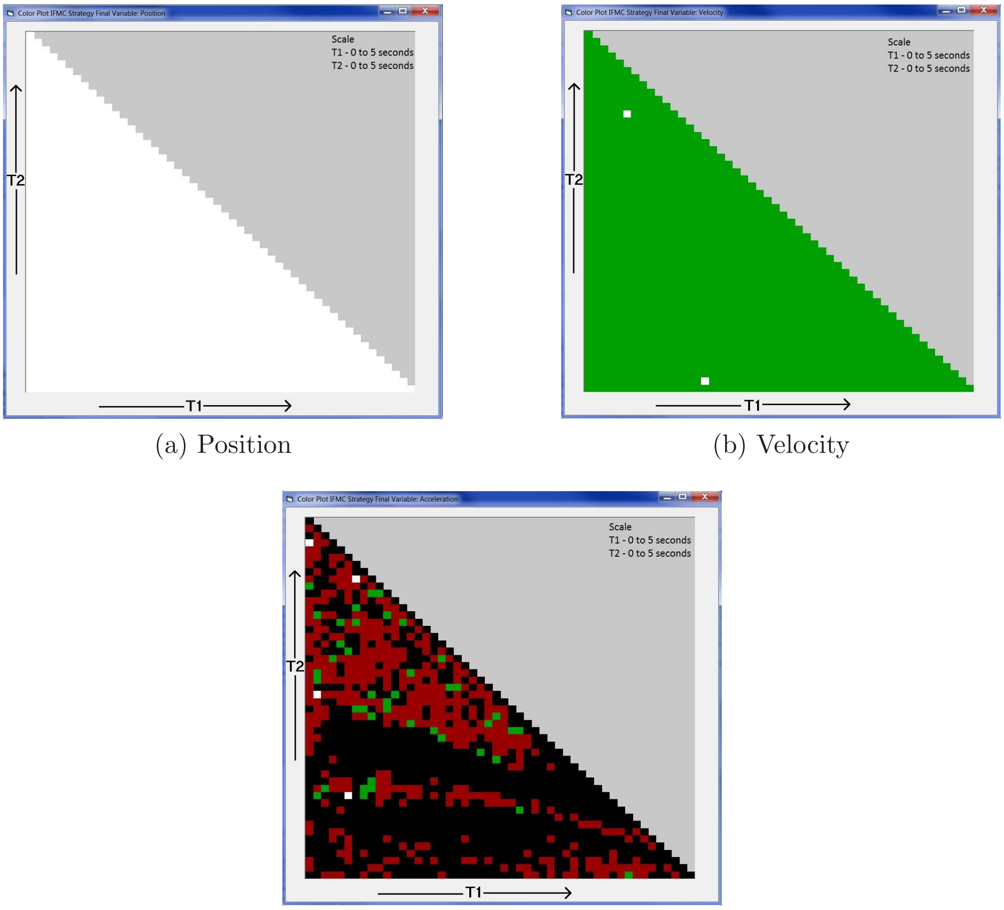
In the simulations so far, the $error \leq 0.0001$ is used as the terminating condition. For the conditions where $error \leq (0.00001 or less)$, it is found that the acceleration enters a limit cycle or dithers. Another option is to calculate the error in such a way that terminating conditions at a smaller level can be used. One way is to calculate error is shown in 3.2.

$$error = x^2 + \dot{x}^2 dt^2 + \ddot{x}^2 dt^4 \quad where \;\; dt = 0.01 \tag{3.2}$$

The idea of multiplying the squares of variables by $dt's$ is that if started from the origin, acceleration is checked after time dt then the velocity would be vdt^2 and the position would be xdt^3. Thus, it would be fair to confirm arrival at target if final values of state - position x are less than 10^{-6}.

Hence, to terminate upon arrival at starting point, the error needs to be compared with a very small number such as 10^{-10}. When this error is compared as $error \leq 0.0000000001$ in the IFMC strategy, the final values for all the states are found as in Fig. 3.11.

In these figures as well, the white points (region) indicate values less than 10^{-6}, the light gray (green) points indicate values between 10^{-4} and 10^{-3}, the darker gray (red) points indicate values between 10^{-3} and 10^{-2} and the black points indicate values grater than 10^{-2}.



(a) Position (b) Velocity

(c) Acceleration

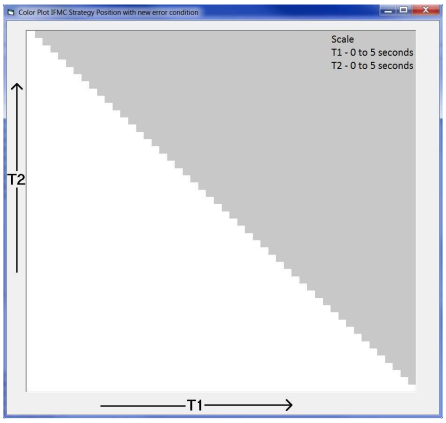
Figure 3.11: Color Plots of Final Variable Values

When (Fig. 3.11a)is compared with (Fig. 3.9a), it showed the significant improvement in the results with all the values being less than 10^{-6}.

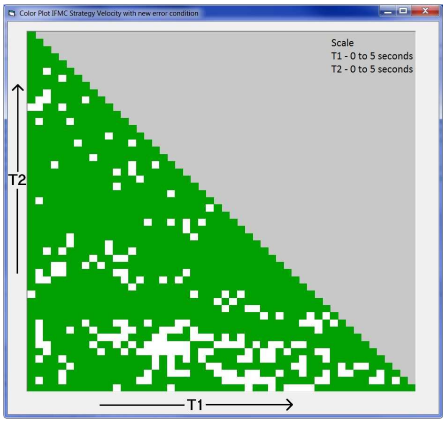
Fig. 3.11b shows that the most of the values for velocity are less than 10^{-4} but greater than 10^{-6}. For acceleration (Fig. 3.11c) shows that many values are over 10^{-2} and very few less than 10^{-6}. Thus, to improve the results in terms of final values of velocity and acceleration, the error is calculated as 3.3,

$$error = x^2 dt + \dot{x}^2 dt^4 + \ddot{x}^2 dt^4 \tag{3.3}$$

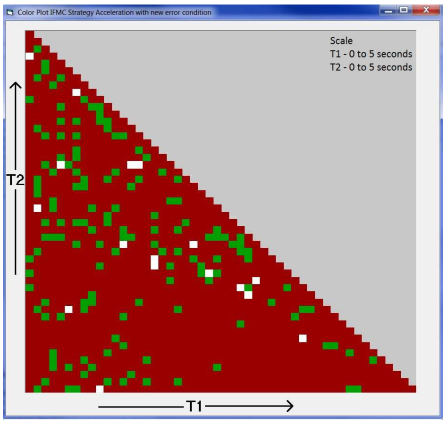
and terminating condition is used as $error \leq 10^{-10}$. Then, the simulation showed the results as in Fig. 3.12.



(a) Final values of position



(b) Final values of velocity



(c) Final values of acceleration

Figure 3.12: Color Plots of Final Variable Values with new error for 2008 Strategy

These results show that the results for final values of velocity and acceleration are much closer to zero.

Velocity in Fig. 3.12b shows that many more values are less than 10^{-6} (many white points in the plot) than before (Fig. 3.11b).

Acceleration in Fig. 3.12c shows that there are no values that are over 10^{-2}(no black points in the plot)and the number of values less than 10^{-4} has increased (few more green points) than before (Fig. 3.11c).

Several attempts are made to improve these results further but it is found that whenever the terms in the equation (3.3) are further multiplied by another value as low as dt the results for position deteriorate.

Thus, it can be stated that equation (3.3) with $error \leq 10^{-10}$ gives the sufficient condition for arrival at the starting point. The change in the error condition did not change the time optimal performance of the strategy very much, it remained similar to (Fig. 3.6a).

However, if the time optimality requirement is more important than the precision to arrive at the nearest target values then it is found that reducing model steplength than plant steplength initially that is before system approaches settling point and plant steplength becomes variable and reducing, improves the time optimality of the strategy.

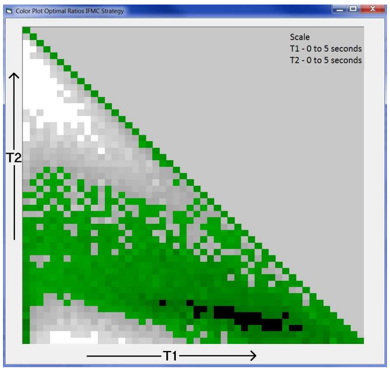
The simulation of IFMC strategy is carried out with model steplength as 0.002 and plant steplength as 0.2. The model/plant steplengths are further reduced as the target is approached using

$$\text{If} \quad dtp > error \;\; And \;\; dtp > 0.0001 \;\; Then \quad dtp = error : dtf = dtp$$

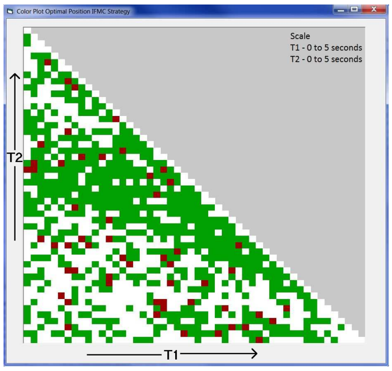
where dtf is the model steplength and dtp is the plant steplength. The error condition is used as in equation (3.2) with termination condition $error \leq 0.0001$. The simulation showed following performance Fig. 3.13.

Fig.3.13a when compared with Fig.3.6a showed that many time optimal ratios have gone close to one (the reduced white region that represented time optimal ratios over 2). Fig.3.13b shows that on many occasions the final position value is less than 10^{-6} (many white points) with most other values still less than 10^{-4} the gray region (green region).

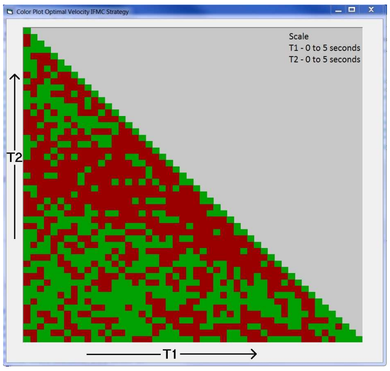
Fig.3.13c shows that final values of velocity are between 10^{-3} and 10^{-6} and finally Fig.3.13d shows that the values for acceleration are between 10^{-3} and 10^{-4}. In this case, the plant starts with higher steplengths but the steplength reduces as the plant gets closer to the target. Since the terminating condition used here is not very small the proximity to the target is compromised.



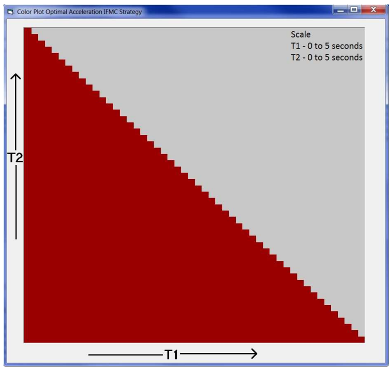
(a) Time Optimality Ratios



(b) Final values of Position



(c) Final values of velocity



(d) Final values of acceleration

Figure 3.13: Color Plots for time optimal performance of Iterative Fast Model Control Strategy

3.7 Important Points

A new All-onside predictive control strategy is proposed, where a model of the system is run ahead in time once with full positive drive and once with full negative drive until all the variables come onside, that is have the same sign as that of input. Then a decision on input u is made by comparing the times taken by models to come onside with respective drives. The u is the drive corresponding to the greater time.

Advantages of new IFMC Strategy over its predecessors

Some of the advantages of new Iterative Fast Model Predictive Control strategy over its predecessors are outlined below

1. The algorithm of the strategy is much simpler.

2. It is much more tolerant towards changes in model/plant steplengths
3. Since all the state variables are considered, it gives cautious control. Hence Iterative Fast Model Control is more likely to give stable performance when applied to fourth and higher order systems.

Effects of Steplengths

The model/plant steplengths are very important in reaching to the target. Initially, the steplengths can be higher but as the target is approached plant steplength in particular should be reduced to ensure the smallest possible error. Typically, if the plant step is reduced in proportion to the maximum time taken by the system variables to come onside,then it is possible to reduce the error from the target to a value which is less than 10^{-6}.

Ideally the model steplength is supposed to be higher than the plant steplength but it is found that in some cases, making initial plant step higher than the model steplength, can improve the time optimality of the system. Usually model/plant steplengths are taken in the proportions of 0.01 but individual choice can vary according to the need of the application.

Effects of Error Condition

The error condition used as a terminating condition to declare the arrival of system at the target point is equally important. The error is calculated using squares of each variable. Since, it is a third order cascaded integrator system, all variables are linked to each other. Thus, it becomes important to consider all of them to decide the arrival of system at the target.

The error needs to be very small to get the best proximity to the target position. To allow the comparison of error with a very small number such as 10^{-10}, squared variables are multiplied by various multiples of dt where dt is equal to 0.01 in the equation (3.3).

It is found that, using reducing plant steplength (as the plant approaches target) instead of dt that is 0.01 to calculate smallest possible value of error, does not cause the significant change in the final results. The terminating condition $error \leq (somenumber)$ needs to have a very small 'some number' such as 10^{-10} so that the error in reaching the target position is minimum.

3.8 Conclusion

New Iterative Fast Model Control Strategy has been examined for the control of cascaded integrator systems with input constraints. It is shown that the strategy gives near-time-optimal control performance for a third order cascaded integrator system with input limits. The response is smoother than that of the earlier methods, replacing the constant-ratio trajectory.

Without the deliberate model mismatch that was termed ‘slugging’, it is shown to be stable when applied to a four-integrator system. The effect of deliberate model inaccuracy is a matter for further investigation in the case of fourth and higher order systems.

It can be seen that the acceleration curve enters in a sliding mode mainly due to model times taking over each other as the system moves. The mathematical analysis of this sliding mode is carried out next, to find the continuous time equivalent of the average of the actual input.

Chapter 4

Sliding Model Analysis and Slugging

4.1 Introduction

A feature of Iterative Fast Model Control is that settling is completed in a finite time, unlike the exponential decay of a system with linear feedback.

Time optimal control requires the drive to take its extreme values, with a limited number of switches of sign. In this respect, Iterative Fast Model Control is suboptimal, since for much of the time it results in sliding motion. The input switches rapidly between extreme values to produce a mark-space ratio representing an effective drive that is contained between the limits.

For the simple second order system it is possible to perform an analysis in terms of this effective drive, but the extension to third order shows escalating complexity.

Applying the strategy to fourth and fifth order systems in its 'pure' form results in stability with a converging succession of overshoots. The performance is considerably improved by applying 'slugging'. This is a deliberate scaling of the initial conditions applied to the fast model.

It is found that although a sixth order system becomes unstable when there is no slugging, appropriate scaling of the initial conditions results in an acceptable response with finite settling time.

4.2 Analysis of a Second Order System Response

The sliding mode analysis of the system response shown in Fig. 4.1 has been carried out. The primary variable curve is of significance, because once sliding has commenced, its gradient gives a measure of the effective mean input.

Notations and Definitions:

x_p and v_p - plant variables where $\dot{x_p} = v_p$ and $\dot{v_p} = u$ i.e. $\frac{d^2x_p}{dT^2} = u$.

x_m and v_m - fast model variables that run ahead in time and predict system behaviour.

t - fast model time that indicates when system variables will be onside.

T - plant time, it is the real time.

t_+ - predicted model time after which the system variables will be onside, with full positive drive.

$x_m(t_+)$ and $v_m(t_+)$ - predicted values of model variables after full positive input for time t_+.

t_- - predicted model time after which the system variables will be onside, with full negative drive.

$x_m(t_-)$ and $v_m(t_-)$ - predicted values of model variables after full negative input for time t_-.

dtp - plant step-length, interval between two instants.

onside - a stage where variables have the sign same as the input.

offside -a stage where variables have the sign opposite to the input.

primary variable - first integral or variable of the cascade that receives input u.

subsequent variables - remaining integrals or variables of the cascade.

A second order cascaded integrator system can be defined as follows $\frac{d^2x}{dt^2} = u$ where $u = \pm 1$. Initial conditions for simulation are $x_p(T) = 20$ and $v_p(T) = 0$

where the plant time $T = 0$. System response is shown in Fig. 4.1 where Curve-4 (subdivided in curve-1 and curve-2) is the primary variable curve. Fig. 4.1 shows that after curve-1, a change in direction occurs and curve-2 starts in a sliding mode.

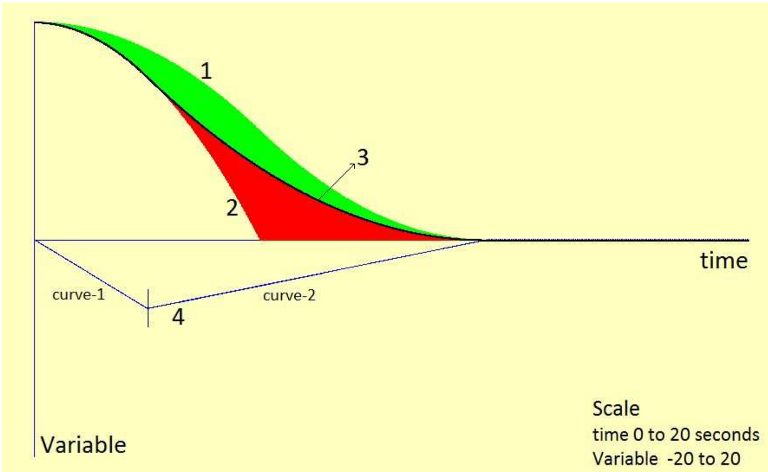


Figure 4.1: Iterative Fast Model Control of a Second Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 2*Velocity (v)

The working of Iterative Fast Model Control is explained step by step, starting with plant time $T = 0$. In the first instant when $T = 0$, a 'fast model' with same initial conditions is run ahead in time twice, once with full negative drive and once with full positive drive.

When the fast model is run with full negative drive, it continues to run until all model variables attain negative sign, at this point the model time t_- is noted. This fast model run predicts the plant variables values after time t_- for which the input will be full negative.

When the fast model is run with full positive drive, it continues to run until all model variables attain positive sign, at this point the model time t_+ is noted. This fast model run predicts the plant variables values after time t_+ for which the input will be full positive.

Now model times t_- and t_+ are compared. Since $t_- > t_+$ the input to the plant is -1. This completes one cycle of process.

In the next instant, after step time of dtp when plant time is $T + dtp$, the old plant values $x_p(T)$ and $v_p(T)$ are assigned as the new initial conditions to fast model of the system and process is repeated, giving new values of model times t_- and t_+.

Since, $t_- > t_+$ the system input remains -1 and process is repeated. As input is -1, at every new instant of the plant time the system gets closer to the its predicted value with negative drive. This decreases t_- at every new instant whereas t_+ keeps increasing.

Fig. 4.2, shows traces after a random time T when the process has been repeated for several such instants at an interval of $dtp = 0.01$ seconds.

Curve-1 and curve-4 in Fig. 4.2 show the predicted paths of variables $x_p(T)$ and $v_p(T)$ with full negative drive for time t_-. $x_m(t_-)$ and $v_m(t_-)$ are corresponding predicted model values of plant variables. This indicates a full negative input for time t_- will bring $x_p(T)$ to $x_m(t_-)$ and $v_p(T)$ to $v_m(t_-)$.

Curve-2 and curve-3 in Fig. 4.2 show the predicted paths of variables $x_p(T)$ and $v_p(T)$ with full positive drive for time t_+. $x_m(t_+)$ and $v_m(t_+)$ are corresponding predicted model values of plant variables. This indicates a full positive input for time t_+ will bring $x_p(T)$ to $x_m(t_+)$ and $v_p(T)$ to $v_m(t_+)$.

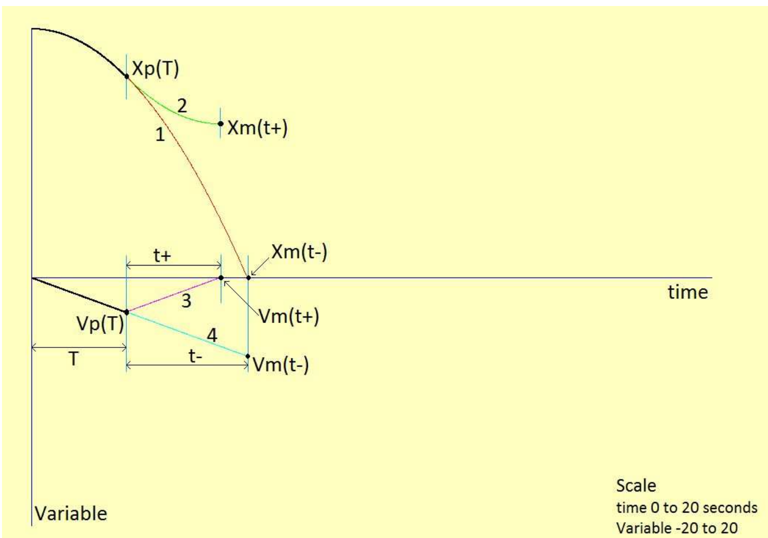


Figure 4.2: IFMC traces after time T when $t_- > t_+$ Curves:- 1: Prediction of position with full negative drive, 2: Prediction of position with full positive drive, 3: Prediction of velocity with full positive drive, 4: Prediction of velocity with full negative drive

As stated earlier, when the system starts working and moving towards its target (settling point), model time t_- decreases and t_+ increases, eventually there comes a time when the model time t_+ just exceeds of t_- such that t_+ and t_- appear to be equal, as shown in Fig. 4.3.

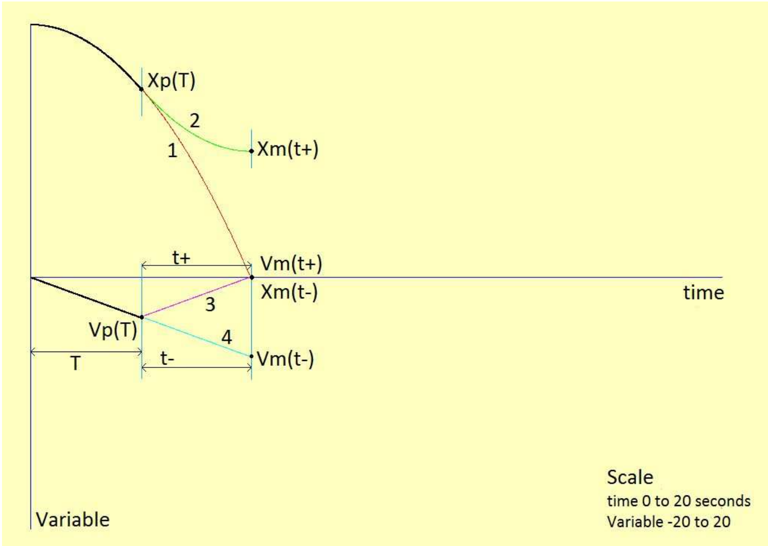


Figure 4.3: IFMC traces after time T when $t_-almost = t_+$ Curves:- 1: Prediction of position with full negative drive, 2: Prediction of position with full positive drive, 3: Prediction of velocity with full positive drive, 4: Prediction of velocity with full negative drive

At this stage, for a full positive drive input, variable $x_p(T)$ is already onside and a full positive input for time t_+ will bring $v_p(T)$ onside to a near zero value. For full negative drive variable $v_p(T)$ is already onside and a full negative drive input for time t_- will bring $x_p(T)$ onside to a near zero value. Recall that variable values due to quantization will not be exact zero but very near to zero.

At this instant, since t_+ is just greater than t_-, the input applied to the plant is changed to full positive drive. This continues until t_- becomes greater than t_+ and so and so forth.

Fig. 4.4, shows that when t_+ becomes just greater than t_- the plant input is switched from full negative drive to full positive drive. But, then t_- catches up quickly and input is switched back to full negative drive. It is evident from the Fig. 4.4, that the switching of input drives occur rapidly thereby generating the sliding mode.

It appears that t_+ and t_- are decreasing by almost same time. In the Fig. 4.4, the sliding mode is shown by the second part of curve-4 which is the primary variable curve, slope of which is the effective plant input. It is possible to find the average value of the effective plant input by the mathematical analysis of the sliding mode.

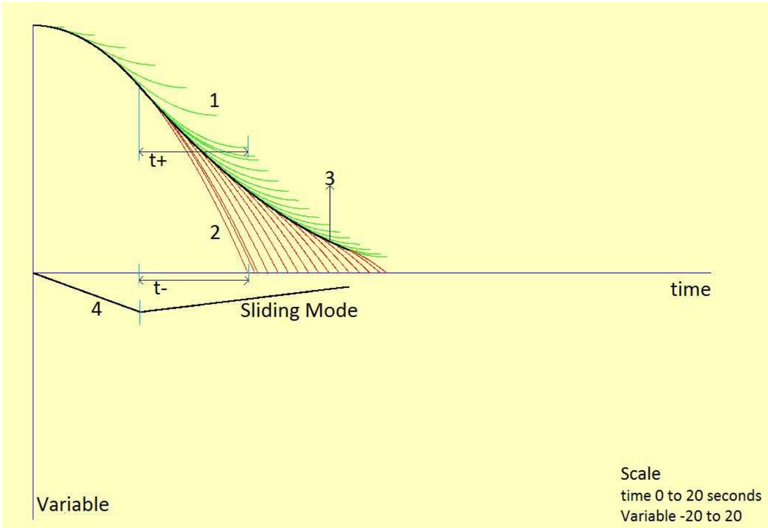


Figure 4.4: IFMC curves after times $t_+ = t_-$ Curves:- 1: Predictions of position with full positive drive, 2: Predictions of position with full negative drive, 3: Plant Position, 4: Plant Velocity

There are two approaches that can be taken here, the first approach is to calculate the rate of change of t_- and t_+ as a function of $x_p(T)$ and $v_p(T)$ the system variables and the input u. This will result in two different equations, that can then be equated and solved to find the average value of u.

The second approach is to find the equations for t_- and t_+ and then equate them to find an equation in terms of $x_p(T)$ and/or $v_p(T)$. Then find an equation for respective variables from the system differential equations. Equate these two equations and find the value of u. These approaches are shown below.

Approach-1

The system and the fast model variables are related as follows

$$\begin{aligned} x_m(0) &= x_p(T) \\ \dot{x_m}(0) &= \dot{x_p}(T) \\ i.e. \quad v_m(0) &= v_p(T) \end{aligned}$$

The system-model equations are

$$x_m(t) = x_m(0) + \dot{x_m}(0)t + \frac{1}{2}ut^2$$

$$i.e. \quad x_m(t) = x_p(T) + v_p(T)t + \frac{1}{2}ut^2 \tag{4.1}$$

$$\dot{x_m}(t) = \dot{x_m}(0) + ut$$

$$i.e. \quad v_m(t) = v_p(T) + ut \tag{4.2}$$

As stated earlier when t_+=t_-, from Fig. 4.3, if $u = +1$ is applied for time t_+ it will bring $v_p(T)$ onside to a near zero value as predicted by $v_m(t+)$. Therefore using equation 4.2

$$t_+ = -v_p(T) \tag{4.3}$$

Differentiating Equation 4.3,

$$\frac{dt_+}{dT} = -\frac{dv_p(T)}{dT} = -u \tag{4.4}$$

Similarly referring to Fig. 4.3, if $u = -1$ is applied for time t_- it will bring $x_p(T)$ onside to a near zero value as predicted by $x_m(t-)$. Therefore using equation 4.1,

$$x_p(T) + v_p(T)t - \frac{{t_-}^2}{2} = 0$$

$$i.e. \quad {t_-}^2 - 2v_p(T)t_- - 2x_p(T) = 0 \tag{4.5}$$

Solving equation 4.5 for t_- and for convenience referring $v_p(T)$ as v_p and $x_p(T)$ as x_p.

$$t_- = \frac{2v_p \pm \sqrt{4v_p^2 + 8x_p}}{2} = v_p + \sqrt{v_p^2 + 2x_p} \tag{4.6}$$

Differentiating equation 4.6

$$\frac{dt_-}{dT} = \frac{dv_p}{dT} + \frac{2v_pu + 2v_p}{2\sqrt{v_p^2 + 2x_p}}$$

$$\therefore \quad -u = u + \frac{v_p(u+1)}{\sqrt{v_p^2 + 2x_p}}$$

Simplifying above equation gives

$$(3v_p^2 + 8x_p)u^2 - (2v_p^2)u - v_p^2 = 0$$

Solving of above equation for u gives

$$u = \frac{1 \pm \sqrt{1 + (3 + \frac{8x_p}{v_p^2})}}{3 + \frac{8x_p}{v_p^2}} \tag{4.7}$$

When $t_{-}{=}t_{+}$ then from equations 4.3 and 4.5

$$v_p^2 + 2v_p^2 - 2x_p = 0$$
$$\therefore \; x_p = \frac{3}{2}v_p^2 \tag{4.8}$$

Substituting the value of x_f in equation 4.7 gives

$$u = \frac{1}{3} \quad or \quad u = -\frac{1}{5} \quad (Discard \;\; negative \;\; value) \tag{4.9}$$

Verification-1

From the system equation, it can be deduced that

$$\frac{d^2x_p}{dT^2} = \frac{dv_p}{dT} = u$$
$$\frac{dx_p}{dT} = v_p \tag{4.10}$$

Therefore using equations 4.10 and 4.9

$$v_p = uT = \frac{1}{3}T \quad that \;\; is \quad T = 3v_p$$

$$and \;\; x_p = \frac{1}{2}uT^2$$
$$\therefore \; x_p = \frac{1}{6}(3v_p)^2$$

which is equation 4.8; hence $u = \frac{1}{3}$ is verified.

Approach-2

Using equation 4.3 and 4.6, when t_{+}=t_{-}

$$-v_p = v_p + \sqrt{v_p^2 + 2x_p}$$
$$\therefore\ 4v_p^2 = v_p^2 + 2x_p$$
$$\therefore\ x_p = \frac{1}{2}\frac{v_p^2}{\frac{1}{3}} \tag{4.11}$$

Again, $v_p = uT$ gives $T = \frac{v_p}{u}$ then

Using $x_p = \frac{1}{2}uT^2$ the value of x_p is found as

$$x_p = \frac{1}{2}\frac{v_p^2}{u} \tag{4.12}$$

Comparing equation 4.11 and 4.12,

$$u = \frac{1}{3}$$

Verification-2

The values for plant variables $x_p(T)$,$v_p(T)$ and input u were stored in a MS-Excel compatible comma-separated values (csv) file, simultaneously during the simulation. The graph generated using these values showed the average value for u as 0.33 that is $\frac{1}{3}$.

The excel graph is shown in the (Fig. 4.5), the blue (lighter) lines forming a rectangle along x-axis indicate the switching input. The average value of u for the sliding mode was found by using the values between 1 and 9.74E-02 (arbitrary 0). The brown (darker) curve shows the response of the plant variable $x_p(T)$.

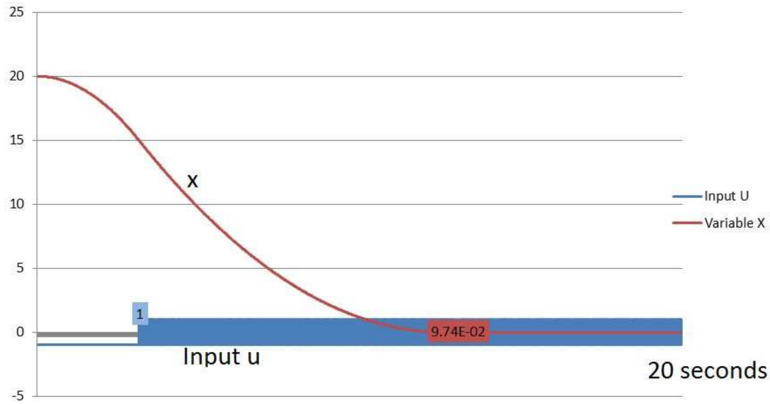


Figure 4.5: The graph of variable x and input u

4.3 Analysis of a Third Order System Response

A third order cascaded integrator system can be defined as $\frac{d^3x}{dt^3} = u$ where $u = \pm 1$. The response of the system is shown in Fig. 4.6.

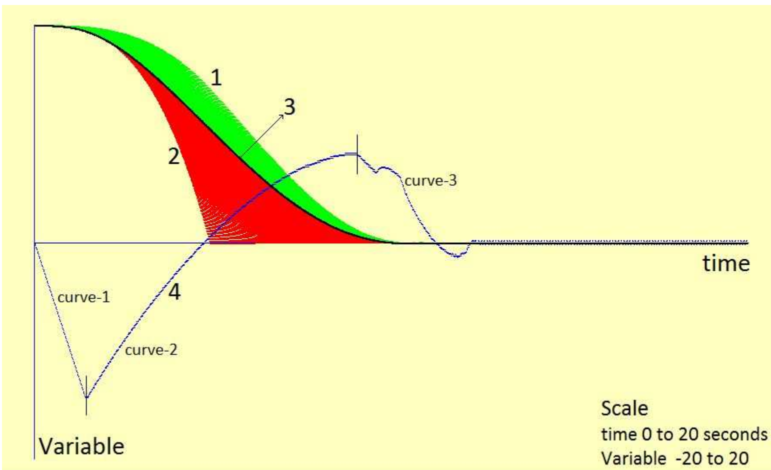


Figure 4.6: Iterative Fast Model Control of a Third Order System 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

In the analysis of the third order system, most of the notations used, are same as previous section, only a couple of new notations are added to accommodate the third variable. They are as follows

a_p - third plant variable where $\dot{x_p} = v_p$,$\dot{v_p} = a_p$ and $\dot{a_p} = u$ i.e. $\frac{d^3x_p}{dT^3} = u$.

a_m - third variable of the fast model

$a_m(t_-)$ - predicted values of a_m after full negative input for time t_-.

$a_m(t_+)$ - predicted values of a_m after full positive input for time t_+.

$a_p(T)$ - variable a_p after plant time T.

In Fig. 4.6, the third order system response shows that the switching of primary variable curve's direction occurs mainly twice, once from curve 1 to curve 2 and then from curve 2 to curve 3. After curve 3 switching occurs a few more times. Every time there is a change in direction of the primary variable curve, fast model is predicting arrival of a different variable to onside to a near zero value.

The initial conditions used here are $x_p(T) = 20$,$v_p(T) = 0$ and $a_p(T) = 0$. Thus similar to previous section initially system starts following the trajectory predicted by $x_m(t_-)$ until the model times t_- and t_+ are equal. Fig. 4.7 shows performance and predictions of different variables when t_+ is just ahead of t_- such that t_-=t_+. In this response the sliding mode starts with the curve 2.

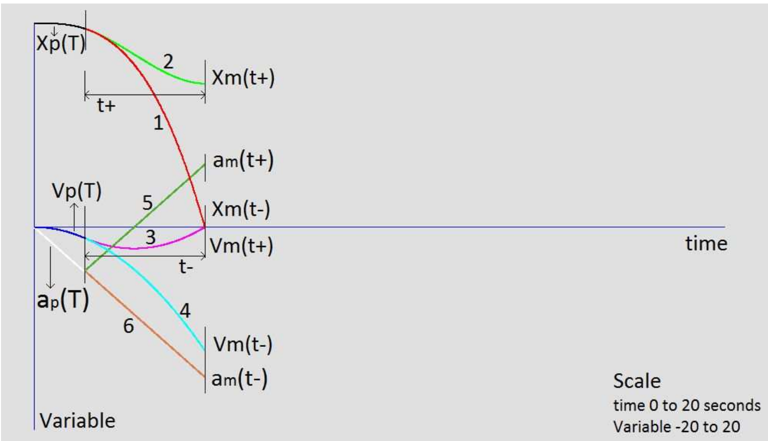


Figure 4.7: 3^{rd} order IFMC traces after time T when $t_-almost = t_+$ Curves:- 1: Prediction of position with full negative drive, 2: Prediction of position with full positive drive, 3: Prediction of velocity with full positive drive, 4: Prediction of velocity with full negative drive, 5: Prediction of acceleration with full positive drive, 6: Prediction of acceleration with full negative drive

For the sliding mode analysis, in the third order system, the system and the fast model variables are related as follows,

$$
\begin{aligned}
x_m(0) &= x_p(T) \\
\dot{x_m}(0) &= \dot{x_p}(T) \\
i.e.\ \ v_m(0) &= v_p(T) \\
\dot{v_m}(0) &= \dot{v_p}(T) \\
i.e.\ \ a_m(0) &= a_p(T) \\
=>\ \ \ddot{x_m}(0) &= \ddot{x_p}(T)
\end{aligned}
$$

The system-fast model equations are

$$
\begin{aligned}
x_m(t) &= x_m(0) + \ddot{x_m}(0)t + \dot{x_m}(0)\frac{t^2}{2} + \frac{ut^3}{6} \\
x_m(t) &= x_p(T) + a_p(T)t + v_p(T)\frac{t^2}{2} + \frac{ut^3}{6} && (4.13) \\
v_m(t) &= v_m(0) + \dot{v_m}(0)t + \frac{ut^2}{2} \\
v_m(t) &= v_p(T) + a_p(T)t + \frac{ut^2}{2} && (4.14) \\
\dot{v_m}(t) &= \dot{v_m}(0) + ut \\
a_m(t) &= a_p(T) + ut && (4.15)
\end{aligned}
$$

Fig: 4.7 shows the start of the sliding mode (curve-2) where t_+ is just ahead of t_-. If the full positive drive is applied for time t_+, variables $x_p(T)$ and $a_p(T)$ will be onside as indicated by $x_m(t_+)$ and $a_m(t_+)$ and variable $v_p(T)$ will arrive onside to a near zero value as indicated by $v_m(t_+)$.

Therefore, if $u = +1$ for time t_+ then $v_m(t_+) = 0$ then from equation 4.14

$$0 = v_p(T) + a_p(T)t_+ + \frac{u{t_+}^2}{2}$$

For convenience, using v_p for $v_p(T)$, a_p for $a_p(T)$ and x_p for $x_p(T)$ and Solving the above equation for t_+ gives

$$
\begin{aligned}
t_+ &= \frac{-2a_p \pm 2\sqrt{a_p^2 - 2v_p}}{2} \\
\therefore\ \ t_+ &= -a_p + \sqrt{a_p^2 - 2v_p} && (4.16)
\end{aligned}
$$

Differentiating above equation

$$\frac{dt_+}{dT} = -u + \frac{a_p u - a_p}{\sqrt{a_p^2 - 2v_p}} \tag{4.17}$$

From Fig. 4.7 when t_+ is just ahead of t_-, if the full negative drive is applied for time t_-, variables $v_p(T)$ and $a_p(T)$ will be onside as indicated by $v_m(t_-)$ and $a_m(t_-)$. However, variable $x_p(T)$ will arrive at $x_m(t_-)$ to a near zero value after time t_-.

Therefore, if $u = -1$ for time t_- then $x_m(t_-) = 0$ then from equation 4.13

$$\frac{t_-^3}{6} - v_p\frac{t_-^2}{2} - a_p t_- - x_p = 0 \tag{4.18}$$

Differentiating equation 4.18 with respect to plant time T gives

$$\frac{t_-^2}{2}\frac{dt_-}{dT} - v_p t_- \frac{dt_-}{dT} - \frac{a_p}{2}t_-^2 - a_p\frac{dt_-}{dT} - ut_- - v_p = 0$$

Rearranging and Solving for $\frac{dt_-}{dT}$ gives

$$\frac{dt_-}{dT} = \frac{v_p - t_- + a_p\frac{t_-^2}{2}}{\frac{t_-^2}{2} - v_p t_- - a_p} \tag{4.19}$$

At this instant as t_+ and t_- are equal, therefore substituting value of t_+ from equation 4.16 for t_- in equation gives

$$\frac{dt_-}{dT} = \frac{v_p - (-a_p + \sqrt{a_p^2 - 2v_p}) + a_p\frac{(-a_p+\sqrt{a_p^2-2v_p})^2}{2}}{\frac{(-a_p+\sqrt{a_p^2-2v_p})^2}{2} - v_p(-a_p + \sqrt{a_p^2 - 2v_p}) - a_p} \tag{4.20}$$

further simplifying gives

$$\frac{dt_-}{dT} = \frac{2v_p - 2(-a_p + \sqrt{a_p^2 - 2v_p}) + a_p(-a_p + \sqrt{a_p^2 - 2v_p})^2}{(-a_p + \sqrt{a_p^2 - 2v_p})^2 - 2v_p(-a_p + \sqrt{a_p^2 - 2v_p}) - 2a_p} \tag{4.21}$$

Comparing equation 4.17 and 4.21 and solving for u, will give the average value of u. For simplicity, the average value of u was found by using an excel table. The values for t_+, t_-, variables x_p,v_p,a_p and u were stored in an excel file simultaneously, as the simulation was carried out.

For the curve 2, the sliding mode begins when t_+ gets just ahead of t_-, and continues until beginning of curve 3. The end of curve 2 is marked by the peak value of a_p as the curve 3 starts in a different direction (descending). The average value of u for curve-2 using simulation data and MS-Excel is found to be 0.3122.

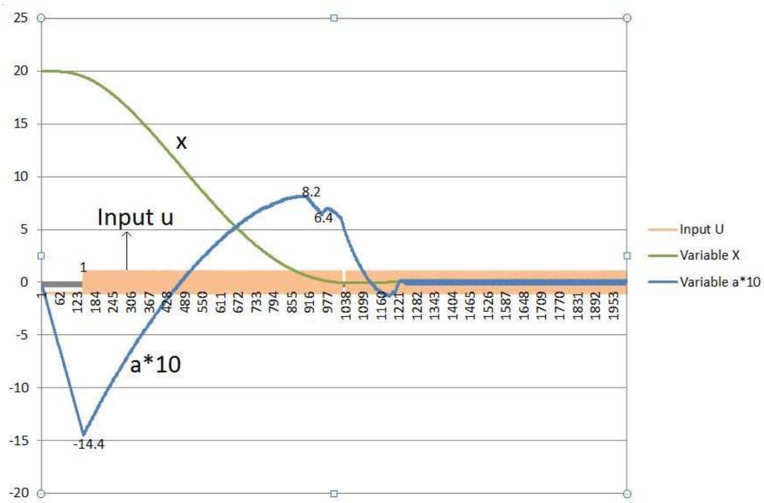


Figure 4.8: The graph of variable x,variable a and input u

In case of the curve 3, there is a change in direction of the curve at the beginning and at few more positions. As mentioned earlier, every time a change in direction occurs a different variable(s) will arrive onside to a near zero value. So, to find the average value of u, respective variables need to be considered.

Here for illustration, only fist part of curve 3 is considered that is from the beginning of curve 3 to the first change in the direction of the curve. Fig. 4.9, shows predictions of different variables at the beginning of the curve 3.

The $u = +1$ drive for time t_+ will bring $x_p(T)$ to a near zero value of $x_m(t_+)$ along with $v_p(T)$ to $v_m(t_+)$, $a_p(T)$ is already onside ($a_m(t_+)$) for a full positive drive. The drive $u = -1$ for time t_- will bring $a_p(T)$ and $x_p(T)$ onside to a near zero value to $a_m(t_-)$ and $x_m(t_-)$ respectively as $v_p(T)$ is already onside for a full negative drive.

Previously, at the beginning of curve-2, Fig. 4.7 predictions indicated $v_p(T)$ would come onside to a near zero value for a full positive input for time t_+ and $x_p(T)$ would come onside to a near zero value for a full negative drive for time t_-. This shows that whenever there is a change in the direction of the primary variable curve a different variable would come onside to a near zero value.

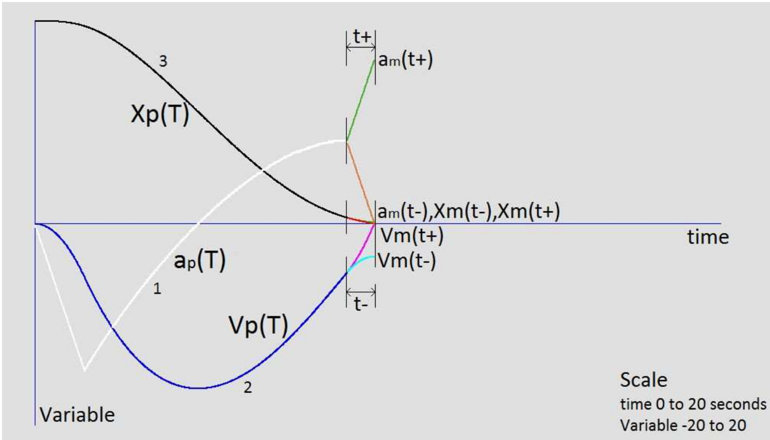


Figure 4.9: Simulation of Iterative Fast Model Control when t_-=t_+ second time

For the first part of curve 3, a full negative drive for t_- will bring $a_p(T)$ onside to a near zero value. Therefore, using $a_m(t_-) = 0$, $u = -1$ and equation 4.15

$$0 = a_p + ut_-$$
$$\therefore \; t_- = a_p$$

Differentiating

$$\frac{dt_-}{dT} = \frac{da_p}{dT} = u \tag{4.22}$$

As seen before for $u = +1$, v_p is offside, therefore using equation 4.17 and equation 4.22 the average value of u can be found out. The excel graph gives the average u as -0.2142

4.4 IFMC For Higher ($> 3^{rd}$) Order Systems

Iterative Fast Model Control strategy can be used to control higher order systems. In this section it is applied to a fourth, fifth and sixth order cascaded integrator systems. The initial simulations showed that the responses of such higher order systems tend to have overshoots, oscillations. In case of sixth order system the system showed unstable performance. The overshoots, oscillations and an unstable system can be improved using, 'Slugging'.

4.4.1 Slugging and its effects

'Slugging' means creating a mismatch between plant and the model whereby the plant variables that are assigned to the model variables are somewhat exaggerated by using a suitable 'slugging factor',thus, making the model 'pessimistic'. As the model's input has diminished authority to correct the response, the control action ultimately becomes more cautious and delivers better control of the system.

To apply slugging, when the plant variables are assigned to the respective model variables to be used as initial conditions, they are multiplied by a 'slugging factor'. Slugging can be applied in various ways depending on the system and its response.

Any number of variables can be multiplied by a 'slugging factor', usually excluding the last last integral or variable position x. The value of slugging factor could be same or different for all variables. It depends on the system. Here, slugging is applied in two different styles.

First, a slugging factor is applied to only the middle variables, that is variables excluding the first variable and the last variable. The slugging factor is the same for all remaining variables. This method is explained using fourth and fifth order systems.

Secondly, a variety of slugging factors are used for different variables. All variables except the last variable (position x) are multiplied by different 'slugging factors'. The 'slugging factors' do not necessarily have to be different for all the variables, they are applied to. Some of them could be same for two or more variables. In this method 'tuning' of the 'slugging factors' is required. This method is explained with a sixth order system.

4.4.2 IFMC for 4^{th} and 5^{th} order systems

Here, a fourth and a fifth order cascaded integrator systems are considered. A fourth order cascaded integrator system can be described by $\frac{d^4x}{dt^4} = u$ where $u = \pm 1$. The simulation of Iterative Fast Model Control strategy for this system is shown in the Fig. 4.10

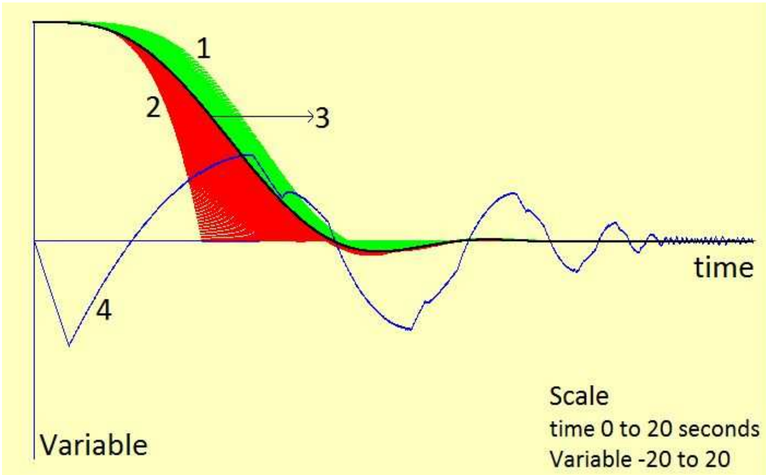


Figure 4.10: Simulation of Iterative Fast Model Control with a Fourth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Jerk (j)

A fifth order cascaded integrator system can be described by $\frac{d^5x}{dt^5} = u$ where $u = \pm 1$. The simulation of Iterative Fast Model Control strategy for this system is shown in the Fig. 4.11

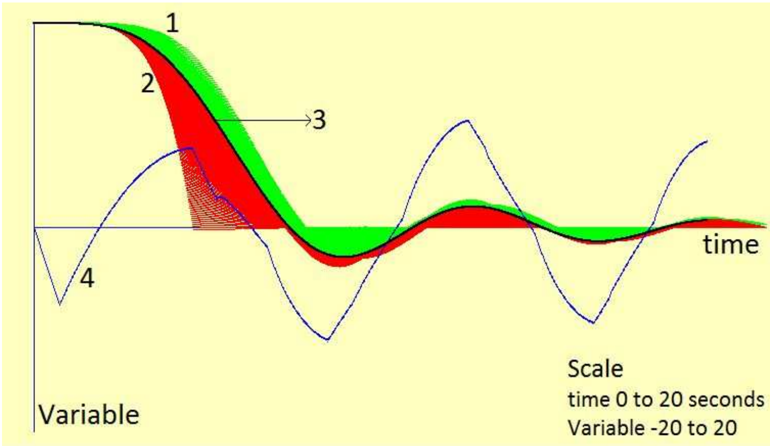


Figure 4.11: Simulation of Iterative Fast Model Control with a Fifth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 5: 10*Rate of jerk k

From the Fig. 4.10 and Fig. 4.11, it can be seen that the system responses contain overshoots, in case of the fourth order system Fig. 4.10, the strategy overcomes the overshoot but in case of the fifth order system Fig. 4.11, the overshoots eventually form oscillations. Slugging can be used to overcome this issue.

The slugging is applied using the first method outlined in previous section where all variables except first and last variable of the cascade are multiplied by a same 'slugging factor'.

In case of a fourth order system (Fig. 4.10) variables a and v are multiplied by a slugging factor of 1.15 every time they are assigned to respective model variables. It removed the overshoot Fig. 4.12.

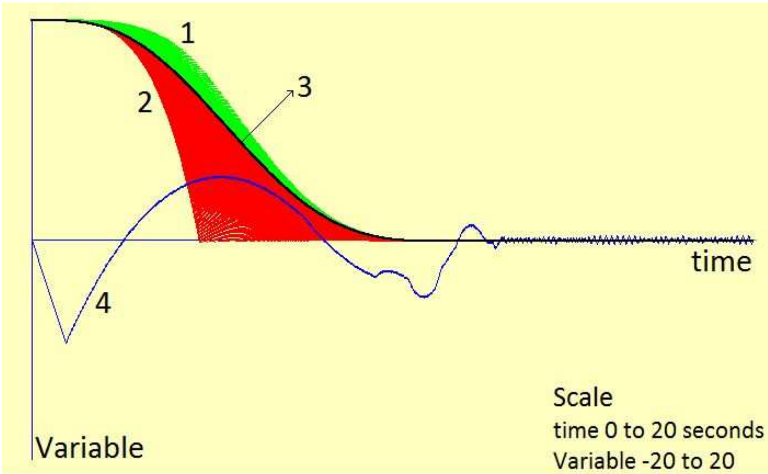


Figure 4.12: Simulation of Iterative Fast Model Control with Slugging for a Fourth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Jerk j

In case of a fifth order system, variables j, a and v are multiplied by a slugging factor of 1.26. The system response with slugging Fig. 4.13 showed significant improvement by removing all overshoots and settling the system quicker.

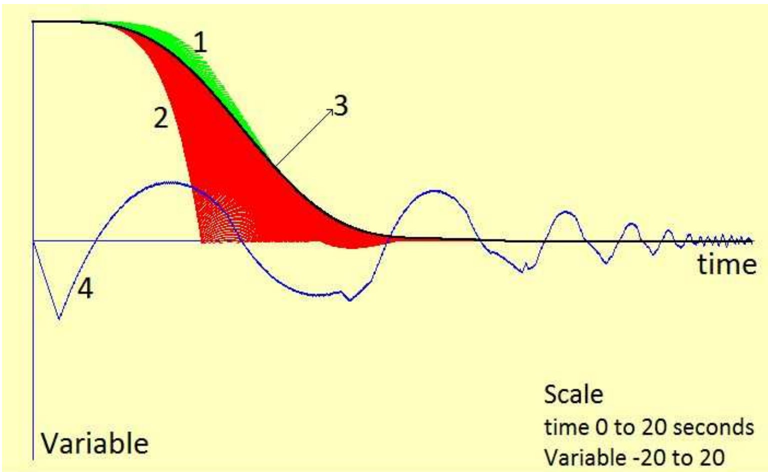


Figure 4.13: Simulation of Iterative Fast Model Control with Slugging for a Fifth Order System Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Rate of jerk

It has been observed that, for best results the slugging factor should be between 1 and 2. There is no specific way to calculate the slugging factor, here a trial and error method is used. First a value around 1.2 is tried then looking at the response and the error, this value is either increased or decreased.

The process is repeated until a desirable response is achieved. It has been further observed that the changes in the initial conditions of the variables do not have much effect on the value of the slugging factor. In this way of implementing slugging, a slugging factor of 2 or more may affect the stability of the system.

To see the effects of slugging on other variables a fifth order case is considered. In the case of a fifth order system without slugging (Fig. 4.11), performance of different variables is shown in Fig. 4.14. The coefficients of the variables in the figure indicate the factor by which the variable values are multiplied for clearer picture.

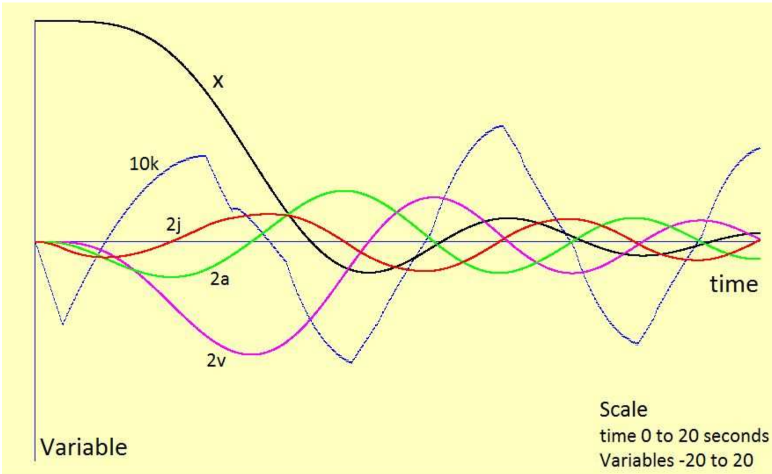


Figure 4.14: Individual variable curves of Fifth Order System without slugging

Fig. 4.15 shows variable responses when slugging is introduced in above mentioned fifth order system. It shows that variables are taking smaller values. The overshoots of variable x are removed and the curve of primary or first variable k is a lot smoother than before.

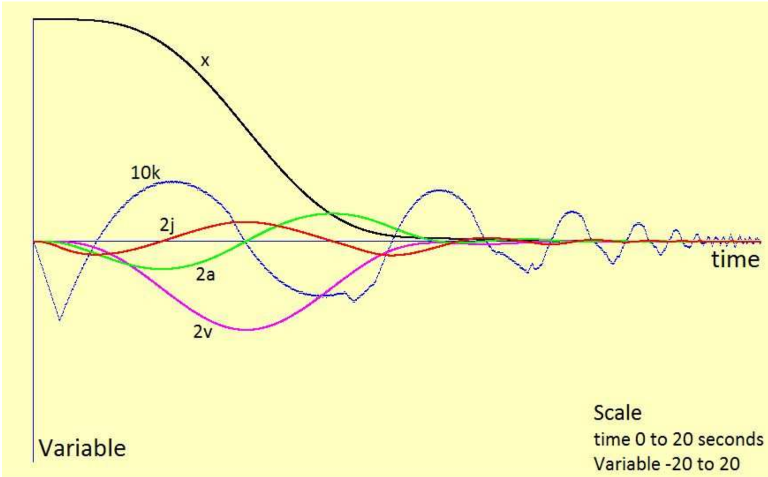


Figure 4.15: Individual variable curves of Fifth Order System with Slugging

4.4.3 IFMC for a 6^{th} order system

Two possible ways of using IFMC for a sixth order system have emerged. First is explained here with the focus on implementation of 'slugging' and demonstrating its usefulness. The second way is explained in Chapter 6 where IFMC is extended to even higher order systems.

Effects of slugging can be vividly seen by the application of the Iterative Fast Model Control to a sixth order system. Here, the second method of slugging is explained where all variables except last variable of a cascade are multiplied by variety of slugging factors.

The system under consideration is $\frac{d^6x}{dt^6} = u$ where $u = \pm 1$. Fig. 4.16 shows system response and Fig. 4.17 shows, performance of different variables. These variables from first to last are p, k, j, a, v and x.

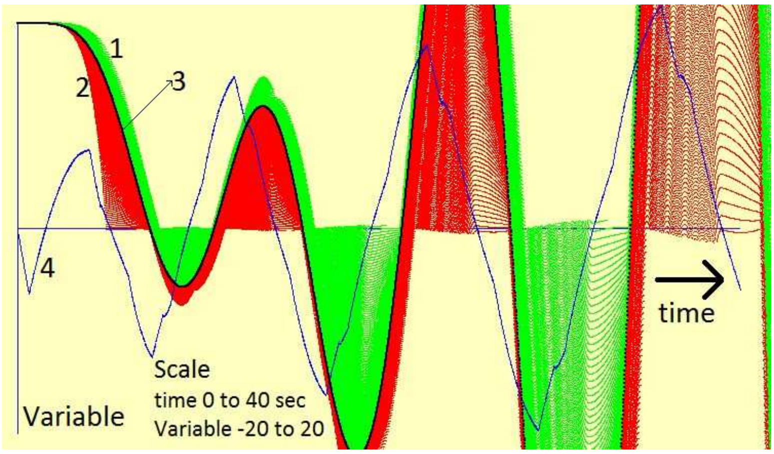


Figure 4.16: Simulation of Iterative Fast Model Control Strategy for a Sixth Order System without slugging

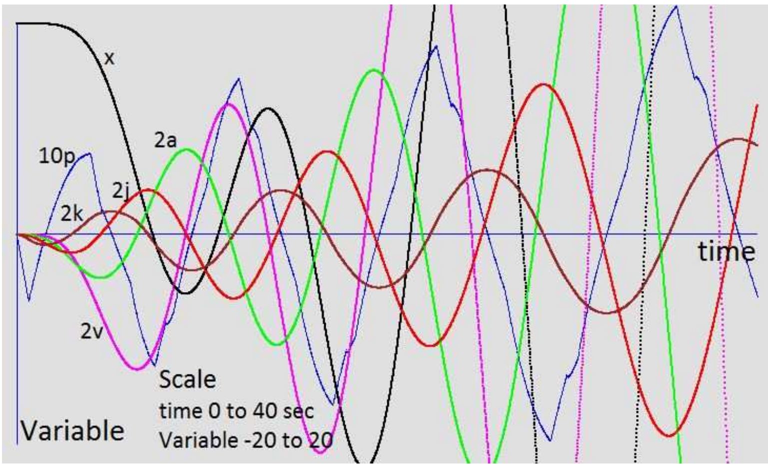


Figure 4.17: Individual variable curves of a Sixth Order System without slugging

From the Fig. 4.16 and Fig. 4.17 it can be seen that the system is quite unstable. Therefore, to stabilize the system 'slugging' is applied where the first five variables from p to v are multiplied by different slugging factors.

The slugging factor for p and v is 1.4, for k and a it is 2 and for j it is 2.5. The resulting system performance is shown in Fig. 4.18 and the performance of various variables is shown in Fig. 4.19.

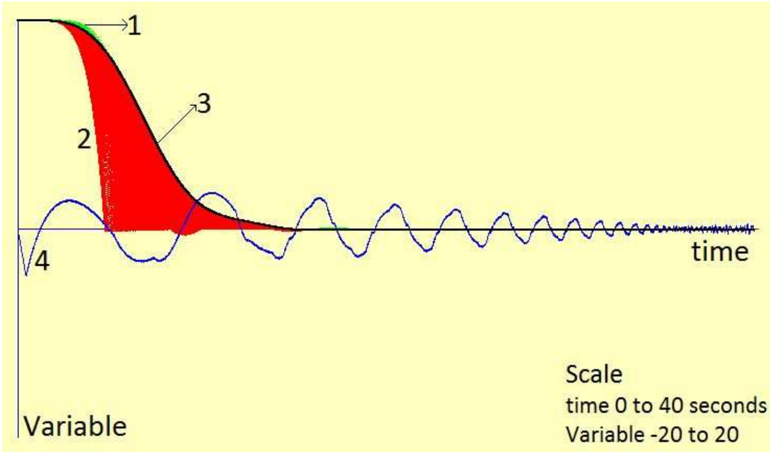


Figure 4.18: Simulation of Iterative Fast Model Control Strategy for a Sixth Order System with Slugging

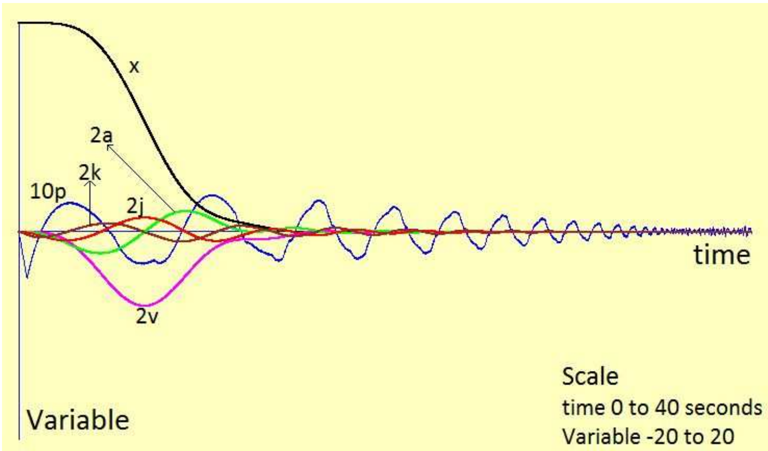


Figure 4.19: Individual variable curves of a Sixth Order System with slugging

Two different ways of applying slugging, explained in sections 4.4.2 and 4.4.3, are compared. The first way of applying same slugging factor to middle variables is relatively easy but it is found to be ineffective in case of a sixth order system. The second way of applying different slugging factors to different variables excluding position x is a little time consuming but effective in case of a Sixth order system.

As the order of the system and number of variables go even higher, implementing slugging using the second way, becomes difficult. The 'tuning' of slugging factors become time consuming.

The second way of implementing slugging was applied in cases of fourth and fifth order systems. However; it did not show any significant difference except the input or first variable curve appeared a bit smoother and the dithering at the end was bit less.

4.5 Important Points

From the discussion carried out in previous section, some of the important points can be summarized as follows,

- In Iterative Fast Model Control a decision on which input to apply at a particular instant, is dependant on the model times t_+ and t_-.
- A 'fast model'of the system is run ahead time once with full positive drive until all model variables have same positive sign and time t_+ is noted and once it is run with full negative drive, until all model variables have same negative sign and time t_- is noted.
- The system input is a drive corresponding to the greater model time between t_+ and t_- and it remains same until the other model time becomes just greater than current model time. Then the input is switched to opposite drive. This switching occurs many times.
- The switching of drives between full positive and full negative values result in sliding mode in the primary variable curve. The slope of the primary variable curve is the effective plant input.
- The mathematical analysis of the sliding mode of primary variable curve in the second order and third order systems showed that the average value of the input u, as indicated by sliding mode is usually less than +1 or -1.
- It has been shown that, the Iterative Fast Model Control can work with the systems up to sixth order. As the order of the system goes higher, overshoots resulting in oscillations occur which may result in the unstable system.
- Slugging can be used to remove overshoots and stabilize the system. In slugging, plant variables when used as initial conditions, are somewhat exaggerated by a slugging factor. Therefore, as the model's input has diminished authority to correct the response, a more cautious control action is generated.

- The value of the slugging factor should be between 1 and 2 and it is found by a trial and error method. The changes in the initial conditions for the model, or the changes in the step-length of the plant or model, do not have much effect on the value of the slugging factor.

- Two possible styles of implementing slugging are 1. Multiply all variables of the model excluding first and last variable by the same slugging factor, 2. Multiply all variables excluding only last variable by different slugging factors, it is not necessary that all slugging factors must be different, some could be same as well.

- Slugging helps to remove the overshoots and oscillations thereby resulting in the more stable system performance.

4.6 Conclusion

In Iterative Fast Model Control, the system response shows the sliding mode that occurs due to the frequent switching between full positive and full negative drive. In case of second order system the sliding mode analysis gave the average value of u as 0.33 which is verified by using a real-time data stored during the simulation.

In a third order system response, the primary variable curve showed a change in direction a few times. This occurs because every time there is a change, a different variable is predicted to come onside to a near zero value for corresponding drives.The mathematical analysis of third order system's primary variable curve showed that with higher order the equations become too complex.

The strategy is then extended to fourth and fifth order systems and 'slugging' is used to remove overshoots. The usefulness of slugging is further demonstrated by stable performance of a sixth order system which is unstable otherwise.

Chapter 5

Iterative Fast Model Control with State Constraints

5.1 Introduction

In addition to input constraints, there can be constraints associated with the states of the system that must be imposed by the controller. For example, in Aircraft lateral control, there will be constraints on the desirable bank-angle of the aircraft and its roll-rate, to avoid discomfort to passengers and to ensure smooth manoeuvring of the aircraft. Therefore, it is important to incorporate these constraints in the strategy. The Iterative Fast Model Control strategies that incorporate such constraints are developed.

The state constraints are incorporated in the strategy step by step. IFMC strategies are based on the use of a fast model that predicts the system behaviour. Therefore, for correct prediction of system behaviour, constraints on any system (plant) variable(s) must be emulated in the fast model as well.

A third order system example is used to demonstrate the IFMC strategy's performance with state constraints. The first integral, the variable of the system that receives the input, is termed here a primary variable while other variables are termed as subsequent variables.

A constraint on a primary variable is incorporated in a different way from constraints on subsequent variables. The system response showed that the system takes slightly longer to settle with all the constraints in place. Simulations of 4^{th} and 5^{th} order systems are carried out to demonstrate that many constraints can be implemented at the same time and smooth performance of IFMC with state constraints for higher order systems.

5.2 Developing IFMC with State Constraints for various systems

5.2.1 A third order system

A the third order system is reconsidered. Initial simulation of the proposed strategy is shown in Fig. 5.1. Initial conditions used are position $x = 10$, velocity $v = 0$ and acceleration $a = 0$.

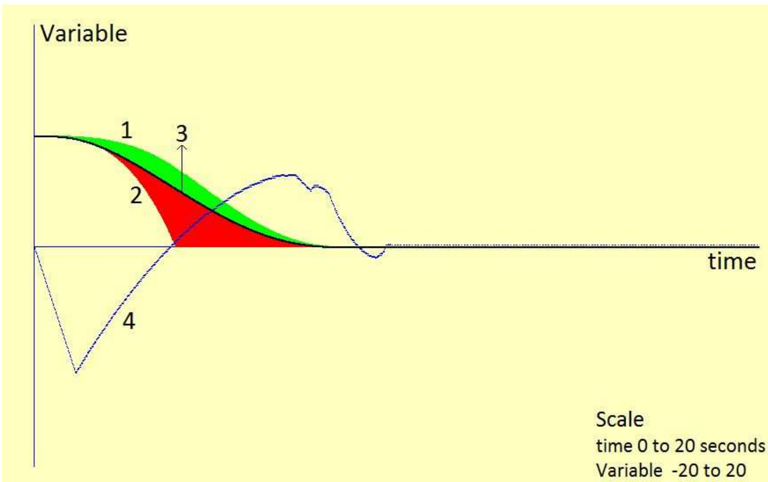


Figure 5.1: Individual variable performance of third Order system: Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*acceleration (a)

The performance of individual variables is shown in Fig. 5.2. The values taken by each variable during simulation are recorded in a separate comma-separated values (csv) file.

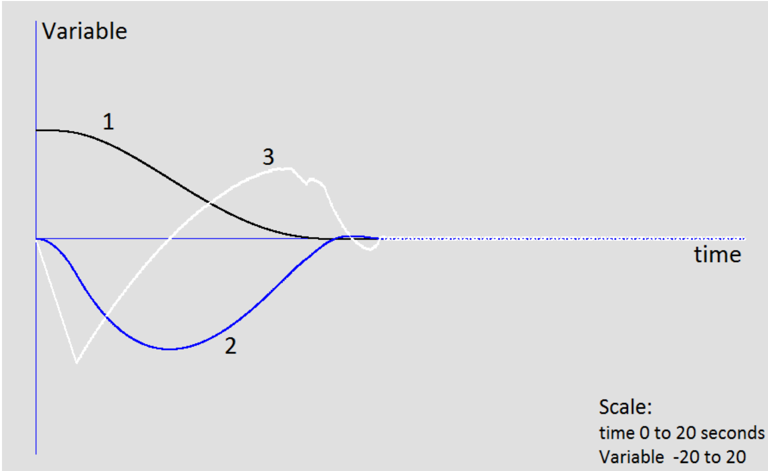


Figure 5.2: Individual variable performance of third Order system: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 10*Acceleration (a)

Fig. 5.2 and the data from the csv file shows that for acceleration curve (curve 3) peak values are at -1.15 in negative quadrant and at 0.65 in positive quadrant. Similarly, values of velocity (curve 2) change mainly in negative quadrant with a peak at -2.042 before returning to 0. The position (curve 1) gradually decreases from initial value of 10 before settling at 0.

It should be noted from Fig. 5.2 that although, it appears that the variables settle at 0, actual settling value is not exact zero. Due to the quantization effects of discrete simulation and drive application, the final states will perform small excursions around zero. More details about this issue can be found out in chapter 3 section 3.6. But, for now it is assumed that the system settles at zero. Suppose there is a limit on acceleration, say ± 0.5 and a limit of ± 1.2 on velocity.

Constraint on the primary variable

The acceleration receives the input u therefore it can be termed the 'primary' variable. The variable 'velocity' in some way dominates the system performance but it receives input u indirectly. Therefore it is called as a 'subsequent' variable. The method to incorporate a constraint on a primary variable is different from the method to incorporate a constraint on a subsequent variable.

The primary variable can be limited simply by setting the plant input to an opposing value if the plant primary variable exceeds its limit. However, this will have some influence on the behaviour of subsequent variables and also, for the correct prediction of system behaviour using the fast model, these limits must be accommodated in the fast model as well. Therefore, in the fast model, a simple limit is applied to this (primary) variable.

For the current system, a constraint is considered on acceleration which is, the 'primary variable'. To implement the constraint on the plant 'primary' variable, we limit primary variable of the model to that value. Then after a decision on input to the plant is made, check the plant primary variable for exceeding the limit. If it exceeds either positive or negative limit; set the plant input drive to the opposite sign of the exceeded limit.

Therefore in the present case when the model is run ahead in time with full positive drive until all variables achieve a positive sign, the acceleration value is limited to 0.35. When the model is run ahead in time with full negative drive until all the variables achieve a negative sign, the acceleration value is limited to -0.35. When a decision on u is made, plant acceleration is checked for limit violations. The simulation result with acceleration limit in place is shown in Fig. 5.3.

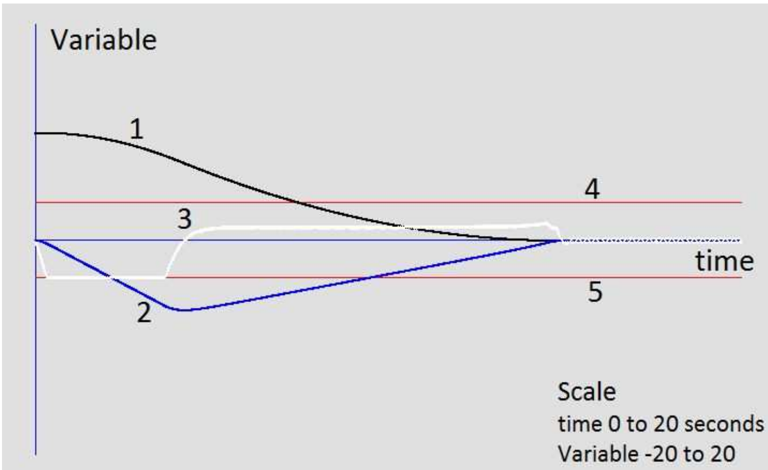


Figure 5.3: Performance of third Order system with constrained acceleration: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 10*Acceleration a, 4: 10*Upper acceleration limit (0.35), 5: 10*Lower acceleration limit (-0.35)

In the simulation Fig. 5.3, the position (curve 1) shows that the settling time has increased. Acceleration (curve 3) shows that the constraint is incorporated successfully and the acceleration remains within specified limits. The IFMC strategy to incorporate only primary variable constraint can be outlined as follows

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, limit the primary variable to its positive limit.
- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this process, limit the primary variable to its negative limit.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive as input. Otherwise select positive drive as input.
- Once the input drive is selected check the plant primary variable. If it is greater than its positive limit then set the plant input to negative drive. If it is less than its negative limit then set the plant input to positive drive.
- Repeat the process.

Constraint on the subsequent variable

A constraint of ± 1.2 on velocity is now considered. Since velocity is a subsequent variable the constraint limit is implemented using flags. The method to incorporate a constraint on a subsequent variable is as follows. When the model is run with full positive drive, the positive input strives to make all the variables positive. During this process, when model subsequent variable becomes less than the negative limit, flag-1 is marked. When the model is run with full negative drive, negative input strives to make all the variables negative. In this process, when model subsequent variable becomes greater than the positive limit, flag-2 is marked.

Then when a decision is made on the plant input drive by comparing model times (when all the variables come onside for respective drives). If the chosen input drive is full negative and flag-2 has occurred then the input is changed to full positive drive. If the chosen input drive is full positive and flag-1 has occurred then the input drive is changed to full negative drive.

In the current example, when the model is run with full positive drive, if the model velocity v becomes less than -1.2 then flag-1 is marked. When the model is run with full negative drive, if the model velocity v becomes greater than 1.2 then flag-2 is marked. Then, after the comparison of model times, input drive is decided. If it is a negative drive then flag-1 is checked if it is marked then the drive is changed to positive. If the decided input drive is positive then flag-2 is checked, if it is marked then the input drive is changed to negative.

The simulation result with a velocity constraint alone is shown in Fig. 5.4. Initial conditions are same as that of section 5.2.1. Constraint on acceleration is ignored for the moment. Simulation shows that the velocity remains within its specified limit but it also has an impact on acceleration. This demonstrates that a constraint on a subsequent variable can be implemented independently.

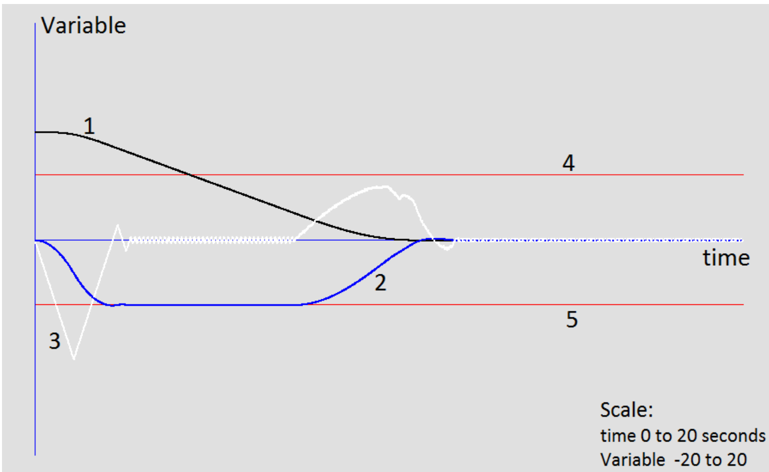


Figure 5.4: Performance of third Order system with constrained velocity: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 10*Acceleration (a), 4: 5*Upper velocity limit (1.2), 5: 5*Lower velocity limit (-1.2)

The strategy to incorporate only subsequent variable constraint can be outlined as follows

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, if the subsequent variable becomes less than its negative limit then mark it as flag-1.
- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this

process, if the subsequent variable becomes greater than its positive limit then mark it as flag-2.

- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive as input but if flag-1 has occurred then change the drive to positive. Otherwise select positive drive as input but if flag-2 has occurred then change the drive to negative.
- Repeat the process.

Example of the section 5.2.1 is carried forward by implementing constraints on both variables (primary and subsequent) together. Fig. 5.5 shows the result of the simulation. It shows that both variables remain within their specified limits.

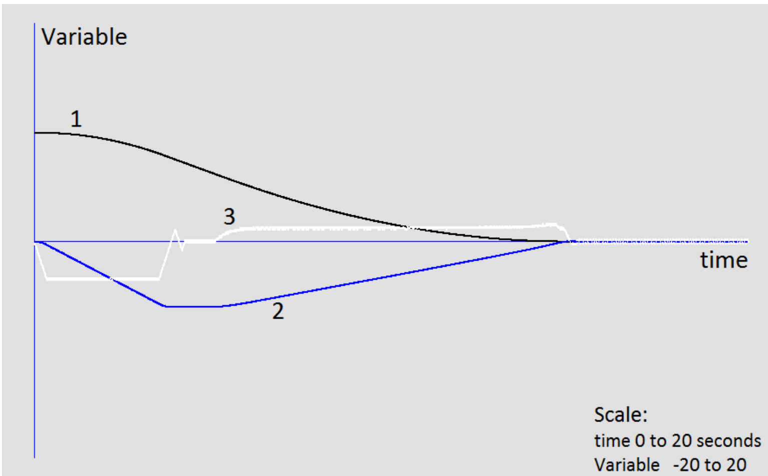


Figure 5.5: Performance of third Order system with constrained acceleration and velocity: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 10*Acceleration (a)

This demonstrates that constraints on primary variable and subsequent variable can be implemented together. A comparison of Fig. 5.3,Fig. 5.4 and Fig. 5.5 shows that the 'settling time' of the system increases as more and more constraints are implemented.

5.2.2 IFMC with state constraints for a third order system

The complete strategy of Iterative Fast Model Control with state constraints for a third order system can be outlined as follows. Here both constraints on primary and subsequent variables are included.

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, limit the primary variable to its positive limit and if the subsequent variable becomes less than its negative limit then mark it as flag-1.
- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this process, limit the primary variable to its negative limit and if the subsequent variable becomes greater than its positive limit then mark it as flag-2.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive as input but if flag-1 has occurred then change the drive to positive. Otherwise select positive drive as input but if flag-2 has occurred then change the drive to negative.
- Once the input drive is selected check the plant primary variable. If it is greater than its positive limit then set the plant input to negative drive. If it is less than its negative limit then set the plant input to positive drive.
- Repeat the process.

5.2.3 A fourth order system

A system response of Iterative Fast Model Control of a Fourth Order System with slugging is shown in Fig. 5.6. Initial conditions are $x = 10$, $v = 0$, $a = 0$ and $j = 0$. In setting the model initial conditions, acceleration a and velocity v are multiplied by a slugging factor of 1.15.

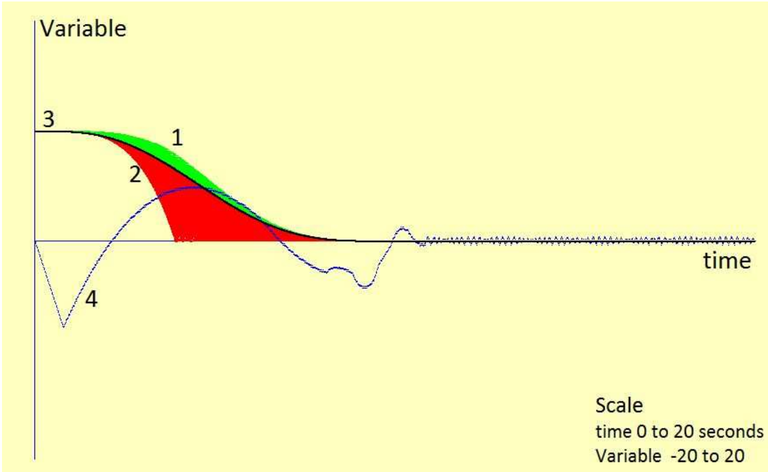


Figure 5.6: Performance of fourth Order system variables: Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Jerk (j)

In final simulation of IFMC with slugging for a fourth order system, performances of the individual variables are shown in Fig. 5.7.

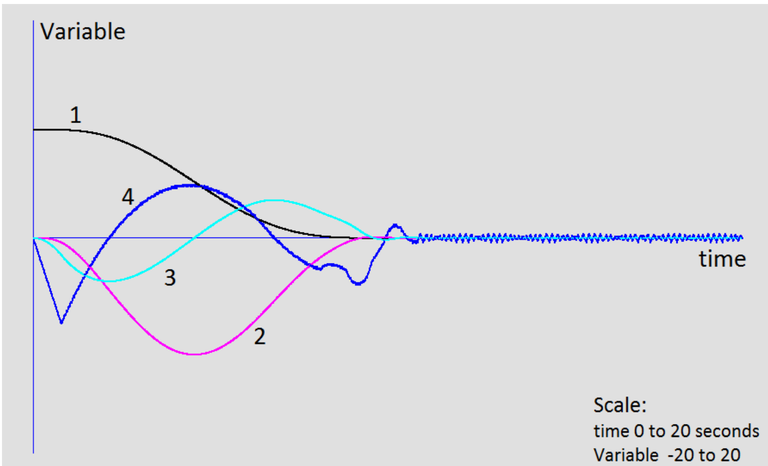


Figure 5.7: Performance of fourth Order system variables: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 5*Acceleration (a), 4: 10*Jerk (j)

The plant variable values were recorded in a separate csv file during simulation. Recorded data indicate that the value of variable v peaks at -2.15727 in the negative quadrant (curve 2). The value of variable a peaks at -0.8024 in negative quadrant and at 0.6970 in positive quadrant (curve 3). Finally the value of j peaks at -0.790 in the negative quadrant and at 0.490 in the positive quadrant (curve 4). A constraint of ± 0.45 is considered on the primary variable j, the jerk. Constraints of ± 0.5 and ± -1.4 are considered on subsequent variables a and v

respectively.

The constraint on primary variable j can be implemented using the same strategy as section 5.2.1. The value of j is limited to 0.45 during the model run with full positive drive and it is limited to -0.45 during the model run with full negative drive. Then, after a decision on input drive is made, if plant $j > 0.45$ then plant input is made full negative and if $j < -0.45$ then plant input is made full positive. The simulation result is shown in Fig. 5.8.

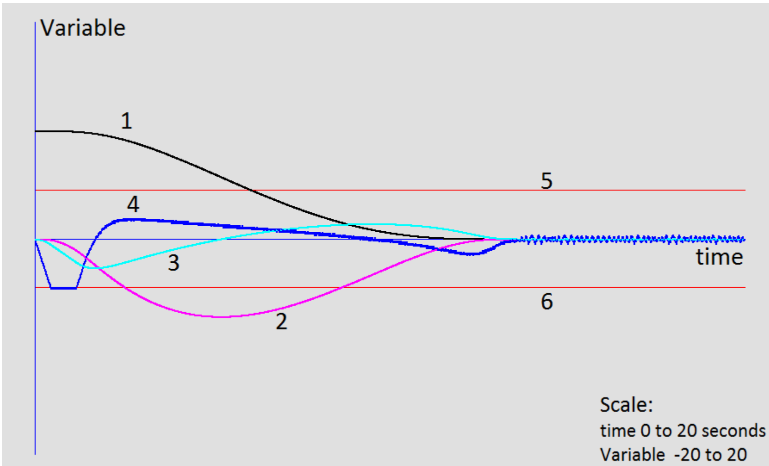


Figure 5.8: Performance of fourth Order system variables: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 5*Acceleration (a), 4: 10*Jerk (j), 5: 10*Upper limit of j (0.45), 6: 10*Lower limit of j (-0.45)

The strategy to put constraints on subsequent variables is same as in section 5.2.1. In this case there are two subsequent variables v and a. It is possible to include constraints on both variables at the same time but if the requirement is to put a constraint on only one of these variables then that works as well. Here constraints on both v and a are considered. As discussed before, the constraint on v is ± 1.2 and on a is ± 0.4.

When the model is run with full positive drive, this time both v and a are checked for crossing their respective negative limits, that is v is checked for crossing -1.2 and a for -0.4. If either should occur then flag-1 is marked. When the model is run with full negative drive, both v and a are checked for positive limit crossover. Variable v is checked for crossing $+1.2$ and a is checked for crossing 0.4. If either has occurred then flag-2 is marked. Then, after the comparison of model times t_+ and t_-, if the chosen input drive is negative and flag-1 has occurred then drive is changed to positive otherwise if input drive is positive and flag-2 has occurred then drive is changed to negative.

As shown before the constraints on primary variable and subsequent variables can be implemented together, the constraints on v and a are implemented along with constraint on j. To show the results clearly, the simulation results are shown in Fig. 5.9 and Fig. 5.10

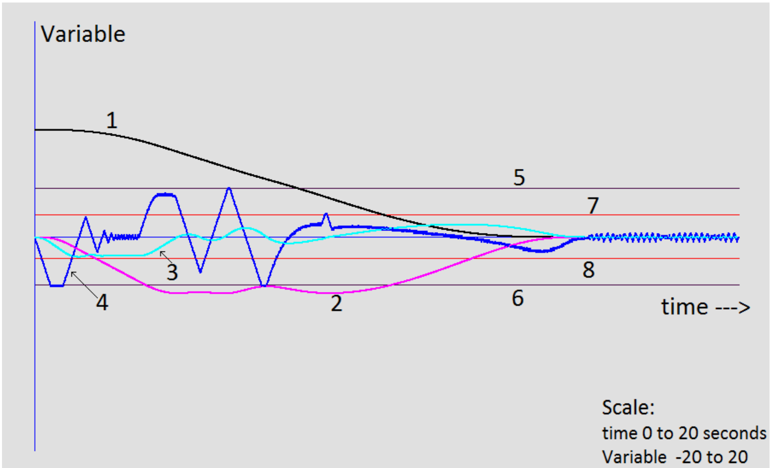


Figure 5.9: Performance of fourth Order system variables with constraints 1: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 5*Acceleration (a), 4: 10*Jerk (j), 5: 10*Upper limit of j (0.45), 6: 10*Lower limit of j (-0.45)

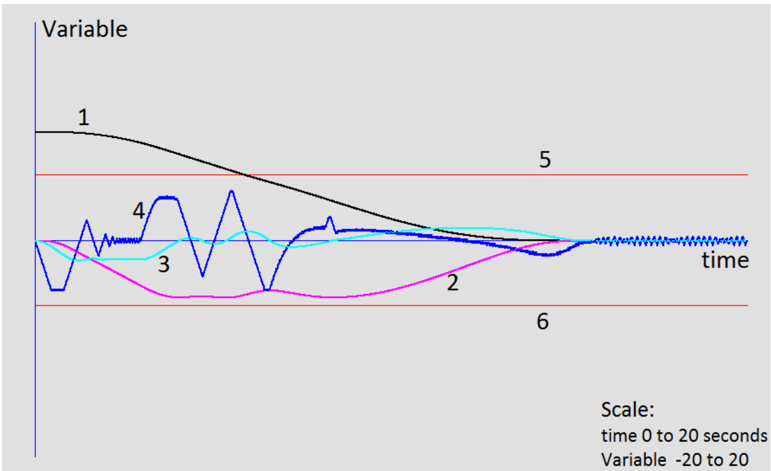


Figure 5.10: Performance of fourth Order system variables with constraints 2: Curves:- 1: Position x, 2: 5*Velocity v, 3: 5*Acceleration (a), 4: 10*Jerk (j), 5: 10*Upper limit of j (0.45), 6: 10*Lower limit of j (-0.45)

5.2.4 IFMC with state constraints for a fourth order system

The Iterative Fast Model Control strategy with constraints for a fourth order system can be outlined as follows. All the state constraints are considered.

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside, that is have the same positive sign. Note the model time as t_+. During this process, limit the primary variable to its positive limit and if either or both of the subsequent variables become less than their respective negative limits then mark it as flag-1.
- Run the model with full negative drive until all the variables come onside, that is have the same negative sign. Note the model time as t_-. During this process, limit the primary variable to its negative limit and if either of or both the subsequent variable become greater than their positive limit then mark it as flag-2.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive as input but if flag-1 has occurred then change the drive to positive. Otherwise select positive drive as input but if flag-2 has occurred then change the drive to negative.
- Once the input drive is selected check the plant primary variable. If it is greater than its positive limit then set the plant input to negative drive. If it is less than its negative limit then set the plant input to positive drive.
- Repeat the process.

5.2.5 A fifth order system

A fifth order system $\frac{d^5x}{dt^5} = u$ where $u = \pm 1$ is considered. The initial simulations of the system are carried out in section 4.4.2. Slugging was used to remove the overshoot. The final simulation of the fifth order system with slugging is shown in the Fig. 5.11 but with slightly different initial conditions than before. New initial conditions are primary variable $k = 0$, subsequent variables $j = 0$, $a = 0$, $v = 0$ and $x = 10$. The slugging factor of 1.26 is used with the variables j, a and v.

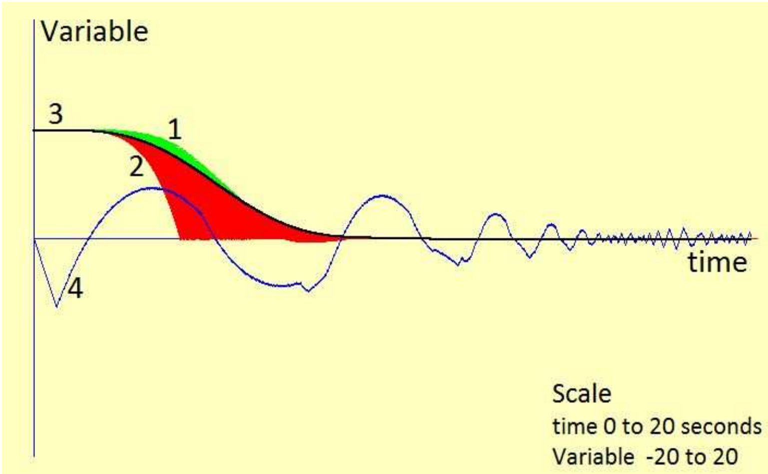


Figure 5.11: Performance of fifth Order system variables: Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 5: 10*Rate of Jerk (k)

Fig. 5.12 shows the performance of individual variables.

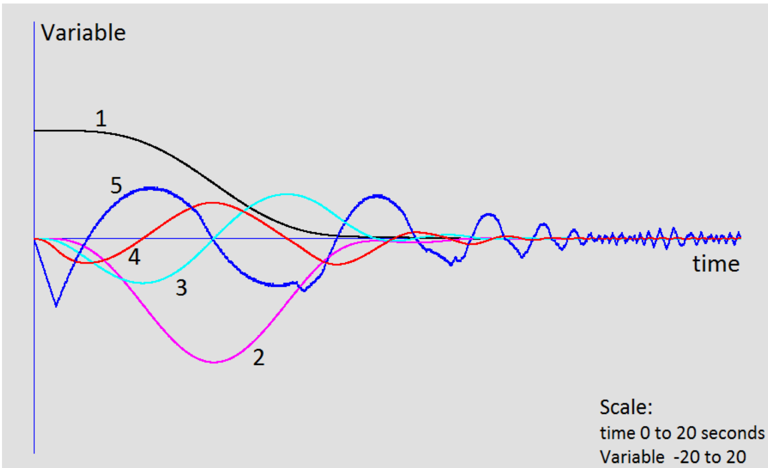


Figure 5.12: Performance of fifth Order system variables: Curves:- 1: Position x, 2: 5*Velocity (v), 3: 5*Acceleration (a), 4: 5*Jerk (j), 5: 10*Rate of Jerk (k)

To decide the limits on variables, values of variables were stored in a separate csv file. The negative and positive peak values for different variables are stated next to each variable with format (negative peak, positive peak). For primary variable k values are (-0.63,0.47), for subsequent variables j (-0.4804,0.6629), for a (-0.82733,0.819211), and for v there is only negative peak (-2.29411).

The limits on each state are assumed as $k(\pm 0.4)$, $j(\pm 0.35)$, $a(\pm 0.6)$, $v(\pm 1.7)$. The simulation of the fifth order system with these constraints using Iterative Fast Model Control is shown in two figures Fig. 5.13 and Fig. 5.14 for clarity.

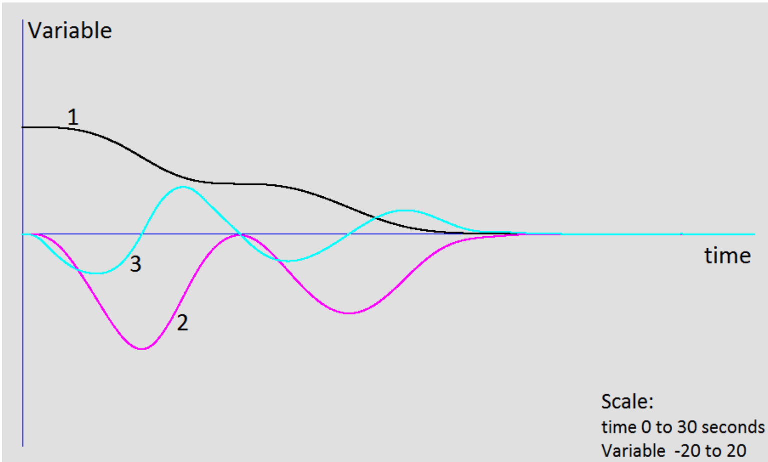


Figure 5.13: Performance of fifth Order system variables v and a with constraints and x: Curves:- 1: Position x, 2: 8*Velocity (v), 3: 8*Acceleration (a)

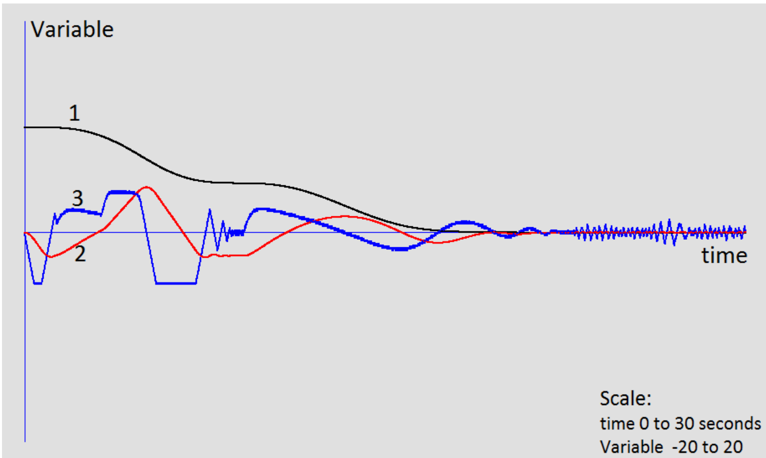


Figure 5.14: Performance of fifth Order system variables j and k with constraints and x: Curves:- 1: Position x, 2: $8 * j$, 3: $12 * k$

The strategy used here is similar to the strategy used for the fourth order system. The only difference being, when the subsequent variables are checked for limit crossovers during the respective cycle of positive and then negative drives, in fifth order all three variables are checked instead of just 2 variables as in the case of the fourth order system.

5.3 Generalised IFMC strategy with state constraints

The general Iterative Fast Model Control strategy with state constraints for systems of any order is outlined here. It is demonstrated that the constraints on variables can be put together all at once or independent of each other as per the requirement of the application.

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, limit the primary variable to its positive limit and if either of (or all of) the subsequent variables become less than their respective negative limits then mark it as flag-1.
- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this process, limit the primary variable to its negative limit and if either of (or all of) the subsequent variable become greater than their positive limit then mark it as flag-2.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive as input but if flag-1 has occurred then change the drive to positive. Otherwise select positive drive as input but if flag-2 has occurred then change the drive to negative.
- Once the input drive is selected check the plant primary variable. If it is greater than its positive limit then set the plant input to negative drive. If it is less than its negative limit then set the plant input to positive drive.
- Repeat the process.

5.4 Important Points

- A constraint on the first integral or the primary variable is implemented by limiting the model primary variable to its positive limit during the model run with full positive drive and to its negative limit for the model run with full negative drive. Then once a decision on input is made by comparing model times, the plant primary variable is checked for limit crossover. If it is more than positive limit then input drive is changed to full negative. If it is less than negative limit then input drive is changed to full positive.

- Constraints on remaining subsequent variables are implemented by flagging the negative limit crossover during positive drive model run and vice-versa and changing the selected input drive to its opposite, upon occurrence of respective flags.
- Here both positive and negative limits are considered however if only positive or only negative limit is required then it can be implemented with the same method. Implement the required limit only.
- The settling time of the system increases with the number of constraints.
- The constraints can be implemented independently or at the same time.
- The techniques to implement state constraints can be extended to even higher order systems.

5.5 Conclusion

The Iterative Fast Model Control strategies with state constraints have been developed. The simulations of 3^{rd}, 4^{th} and 5^{th} order systems demonstrated the usefulness of the strategies. A constraint on a primary variable or the first integral is implemented differently than other subsequent variables. The settling time increases by implementing state constraints. The state constraints can be applied to systems of even higher order. The constraints are applied on both positive and negative side but if required they can be implemented only on one side as well. The method is the same, just apply the required limit.

Chapter 6

Iterative Fast Model Control of Higher Order Systems

6.1 Introduction

In this chapter, Iterative Fast Model Control Strategy is extended to higher order systems. It is shown that, IFMC with limits on some of the system variables, can be applied to systems up to the 11^{th} order. The possibility of application of IFMC to a system with an order as high as n^{th} order is put forward and discussed.

It has been shown earlier that a sixth order system can be stabilized using slugging. An alternative method of limiting first three integrals or variables of the cascade to 0.01 for model runs with full positive drive and to -0.01 for model runs with full negative drive, is successfully used to stabilize the sixth order system. This method is then successfully implemented on 7^{th} to 10^{th} order systems. In case of 11^{th} order system, slugging is used as well.

It is found that the modified IFMC can be applied to systems with order less than 6 but it compromises the time optimal performance of the corresponding system. Therefore, it is possible to use the modified IFMC for systems with an n^{th} order, as long as $n > 3$ but it is more effective in cases of systems with order $n \geq 6$.

6.2 IFMC for a 6^{th} Order system

A sixth order system $\frac{d^6x}{dt^6} = u$ with $u = \pm 1$ shows unstable response with Iterative Fast Model Control but it has been shown earlier in section 4.4.3, that slugging stabilizes the system response. 'Slugging' is a deliberate model mismatch where plant variables are multiplied by a slugging factor before they are assigned to corresponding model variables.

In this case, all variables excluding last position variable, need to be multiplied by a slugging factor. This factor is different for each variable. The variables for a sixth order system from first to last are k,p,j,a,v and x. The initial conditions for variables are $x = 10$ and all other variables equal to 0. The slugging factors used are 1.4 for k and v, 2 for p and a and 2.5 for j. The system response is shown in Fig. 6.1.

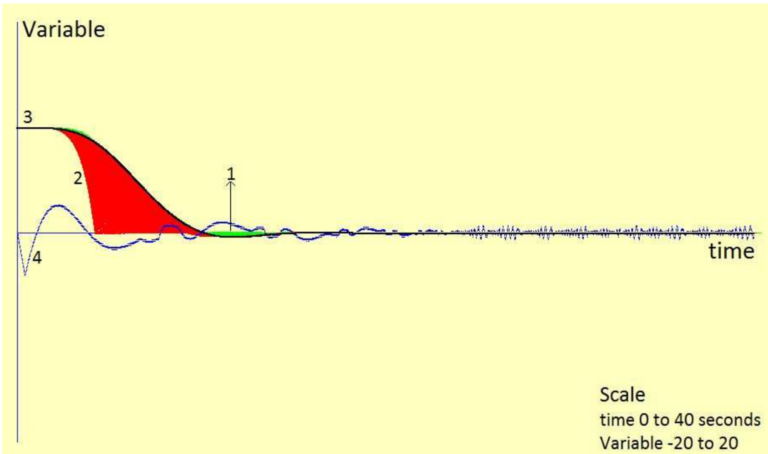


Figure 6.1: Sixth Order System Response with Slugging Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Primary variable (p)

The dithering towards the end of the curve-4 in Fig. 6.1 can be minimized if plant and model steplengths are calculated as $1/number$ where number is a multiple of 8. It is mainly because the numbers are usually seen as a set of bits or bytes by the computers and therefore it will give better rounding results.

In Fig 6.1, the curve 3 shows an overshoot just below region 1. The slugging factors can further be manipulated to remove the overshoot. The appropriate slugging factors are usually found by 'trial and error' and hence can be time consuming and repetitive.

Therefore, a new approach of using constrained model variables is taken to stabilize the sixth order system. This is a prelude to the strategy for even higher order systems. In the new approach, first three variables or first three integrals of the model are limited to 0.01 for full positive drive and to -0.01 for full negative drive and all the slugging factors are removed. Also, to reduce the dithering, model and plant steplengths are changed to 1/256 instead of 0.01.

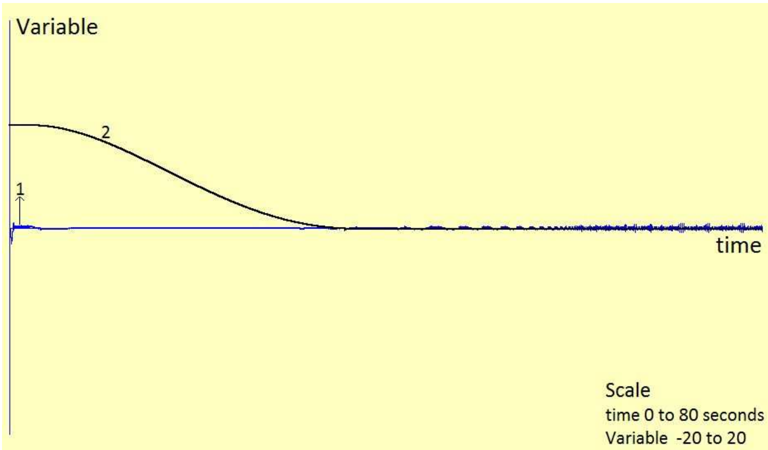


Figure 6.2: Sixth Order System Response with new approach Curves:- 1- 10*Primary variable (p), 2- Position x

Fig. 6.2 shows that the overshoot seen earlier in Fig 6.1 is now removed but the settling time is now increased. Also, the dithering shown before is now less. One thing to note here is that for better clarity the curve-1 is plotted by multiplying values of p by 10. Hence the values attained by p during dithering are far less than 0.0001. The performance of other variables with the new approach is shown in Fig. 6.3. The response values of second variable (k) of the cascade are found to be closer to zero (less than 10^{-2}), hence mostly along the 'time' axis, therefore second variable curve is not explicitly shown.

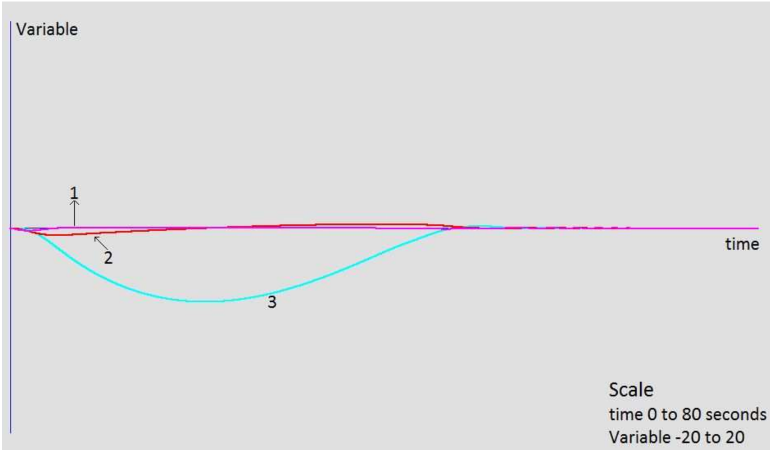


Figure 6.3: Sixth Order System Response with new approach Curves:- 1: 10*third variable (j), 2: 10*fourth variable (a), 3- 10*fifth variable (v)

Fig. 6.3 also shows that the variable j which is limited in the model remains within the limit in plant as well. All the variables settle without any noticeable dithering.

6.2.1 Why limit of ± 0.01

The system response with limits ± 0.01 on three of its respective model variables, has been found to be a lot smoother with reduced first integral dithering. It is found that the limit of ± 1 on model variables for respective drives result in oscillatory system response. A limit of ± 0.1 even though has a little quicker settling time than the limit of ± 0.01 but the system response has a few but small oscillations and higher first integral dithering.

The system responses for limits ± 1 and ± 0.1 are shown in Fig. 6.4 and Fig. 6.5 respectively. The plant and model steplengths are 1/256. A comparison of 6.2 with Fig. 6.4 and Fig. 6.5 shows that the limits of ± 0.01 deliver improved response.

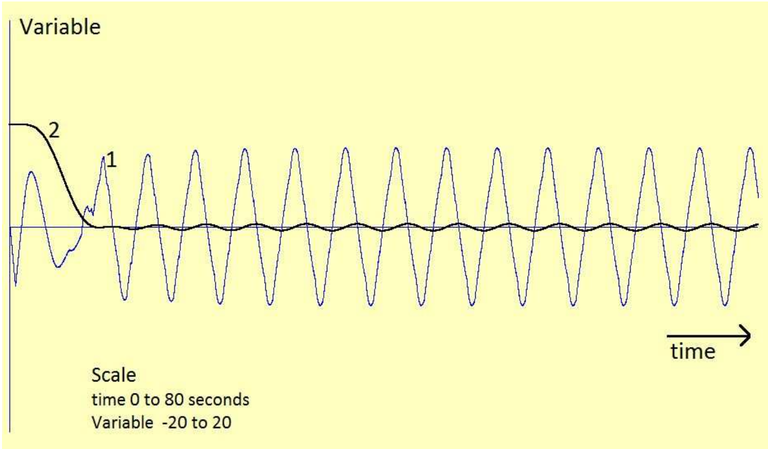


Figure 6.4: Sixth Order System Response with limit ±1 Curves:- 1: 10*Primary variable (p), 2: Position x

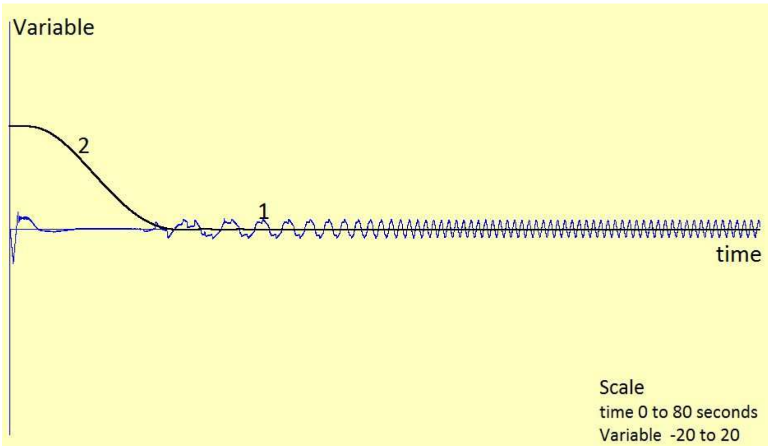


Figure 6.5: Sixth Order System Response with limit ±0.1 Curves:- 1: 10*Primary variable (p), 2- Position x

6.2.2 IFMC with a new approach

The Iterative Fast Model Control strategy with new approach can be outlined as follows,

- Set the model with the plant variables.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, limit the first three variables to 0.01.

- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this process, limit the first three variables to -0.01.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive. Otherwise select positive drive.
- Repeat the process.

This strategy is then extended to the higher order system with changes in the number of variables limited to either 0.01 or -0.01 for corresponding drives.

6.3 IFMC for a 7^{th} Order system

In case of 7^{th} order system, due to more variables, finding the appropriate 'slugging' factors turned out to be difficult. It was also observed that 'slugging' did not settle the system. Therefore, the new approach from the previous section, is extended to control the 7^{th} order system, $\frac{d^7x}{dt^7} = u$ where $u = \pm1$.

It was found that for a 7th order system a limit of ±0.1 on model variables for corresponding drives, made the system oscillatory. Therefore, first four integrals or variables are limited to 0.01 for full positive and to -0.01 for full negative drive respectively.The initial conditions are $x = 10$ and remaining variables equal to 0.

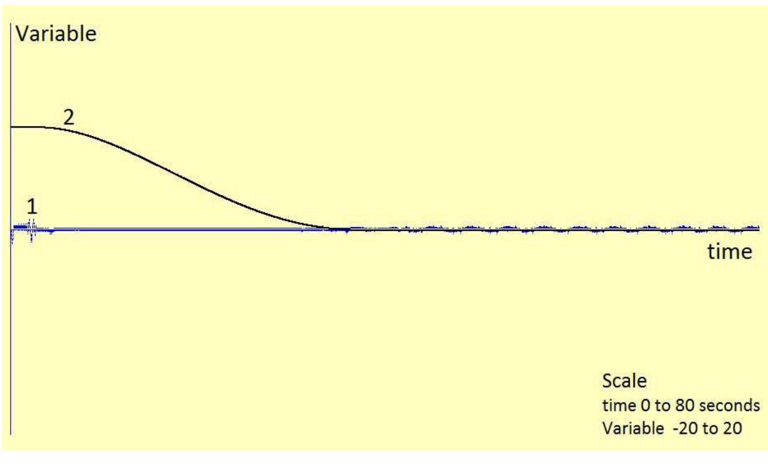


Figure 6.6: Seventh Order System Response with new approach Curves:- 1: 10*Primary Variable, 2: System response x

The plant and model steplengths of 1/256 increased the simulation plotting time. Therefore, in Fig. 6.6 plant and model steplengths are taken as 0.01. The limit of ±0.01 on model variables in 6^{th} and 7^{th} order systems, showed better system performance. Therefore in the next section onwards the limit of ±0.01 is used.

6.4 IFMC for 8^{th} to 11^{th} Order systems

For an 8^{th} order system, $\frac{d^8x}{dt^8} = u$ where $u = \pm 1$; the first five variables of the model are limited to ±0.01 for corresponding positive and negative drives. The plant and model steplengths are 0.01. The initial conditions are $x = 10$ and rest other equal to 0.

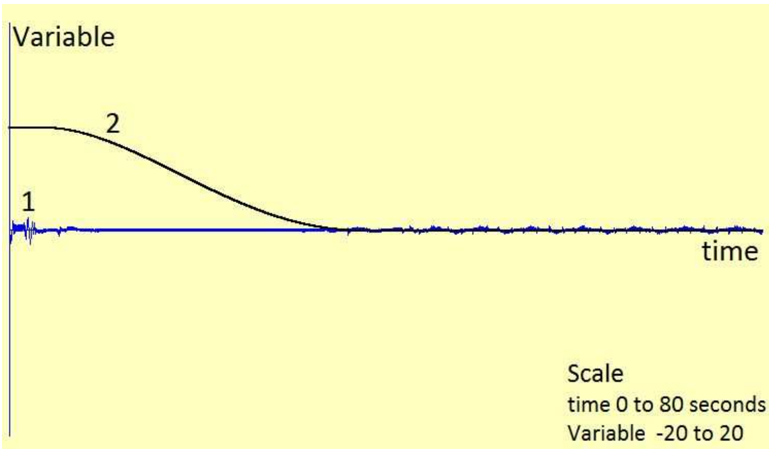


Figure 6.7: Eighth Order System Response with new approach Curves:- 1: 10*Primary Variable, 2: System response x

Fig. 6.7 shows that the system is stable with very low dithering. Next, the case of a 9^{th} order system $\frac{d^9x}{dt^9} = u$ where $u = \pm 1$ is considered. Here, first six variables of the model are limited to 0.01 for a positive and to −0.01 for a negative drive.

The plant, model steplengths are 0.01 with initial conditions $x = 10$ and rest other equal to 0. Fig. 6.8 shows the response of a 9^{th} order system. It shows that the system is stable with low dithering for input or first variable.

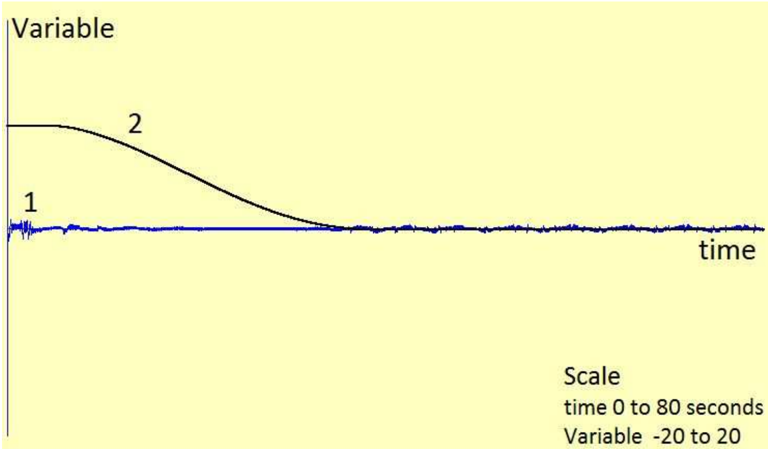


Figure 6.8: Ninth Order System Response with new approach Curves:- 1: 10*Primary Variable, 2: System response x

It has been shown by the responses of 7^{th}, 8^{th} and 9^{th} order systems, that the dithering for the input variable remains low. Therefore to conserve the computational and plotting time with higher order system, only system response (variable x) is plotted.

For the 10^{th} and 11^{th} order systems, the initial conditions are considered as $x = 10$ and remaining variables equal to 0. The plant and model steplengths are 0.01. For the 10^{th} order system first 7 variables, for the 11^{th} order system first 8 variables are limited to 0.01 for the full positive drive and to -0.01 for the full negative drive.

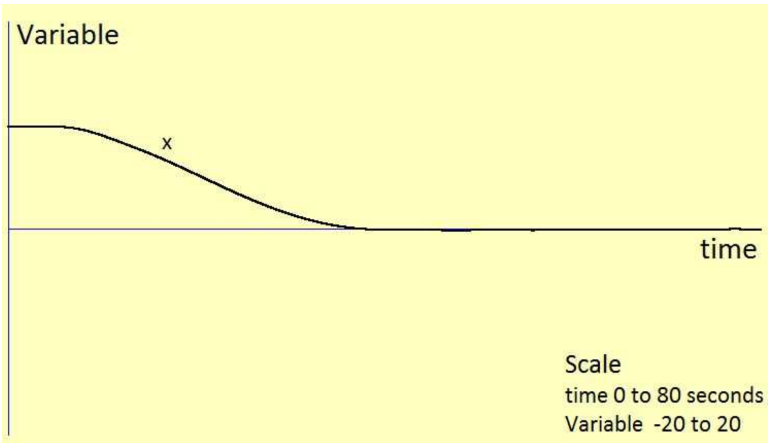


Figure 6.9: Tenth Order System Response with new approach Curves:- 1: 10*Primary Variable, 2: System response x

Fig. 6.9 shows the response of a 10^{th} order system and Fig. 6.10 shows the response of an 11^{th} order system.

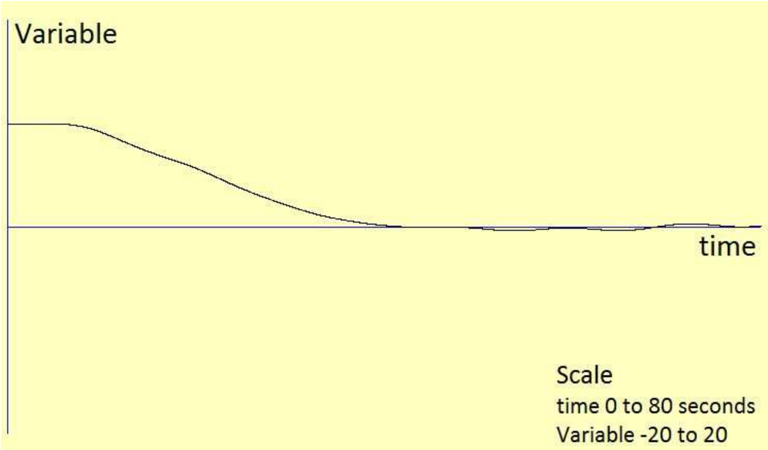


Figure 6.10: Eleventh Order System Response with new approach Curves:- 1: 10*Primary Variable, 2: System response x

The 11^{th} order system response is made bit smoother by using 'slugging'. Slugging is implemented by multiplying second to eight variables by a slugging factor of 1.09. The slugging factor is kept same for all the variables.

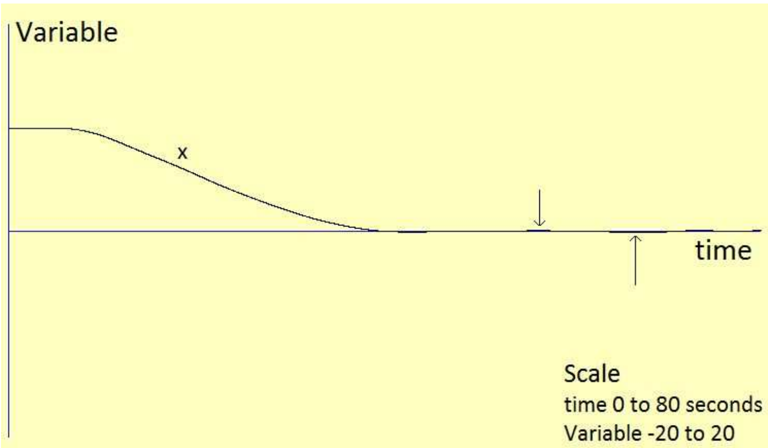


Figure 6.11: Eleventh Order System Response with new approach and slugging

The anomalies shown by arrows in Fig. 6.11 could possibly have appeared due to rounding of numbers or the screen resolution did not support certain value or better slugging factor was required. Here, the emphasis is given to test the applicability of the IFMC to higher order systems only. Therefore, fine tuning of corresponding system responses is not carried out. As the order of the system goes higher so do the calculations and numbers. The simulation time increases considerably as well.

6.5 IFMC for an n^{th} Order system

It has been observed from the simulations of systems from 6^{th} to 11^{th} order system that Iterative Fast Model Control Strategy outlined in section 6.2.2 follows a certain pattern. The number of model variables that need to be limited, increase by 1 with the increase in the order of the system. That is 3 for 6^{th} order, 4 for 7^{th} order, 5 for 8^{th} order, 6 for 9^{th} order, 7 for 10^{th} order and finally 8 for 11^{th} order, leaving last 3 variables all the time.

Since, the strategy has followed the same pattern for 6 different systems, it is possible that the strategy will continue to perform for systems with even higher order with similar pattern. Therefore for an n^{th} order system, first $n - 3$ model variables should be limited to 0.01 for a full positive drive and to -0.01 for a full negative drive.

If $n > 10$ and system response is not desired then the strategy can be altered by taking measures like slugging etc, as per the system requirements. The strategy for an n^{th} order system can be outlined as follows

- Set the model with the plant variables. Use slugging if required particularly when system order $n > 10$.
- Run the model with full positive drive until all the variables come onside that is have the same positive sign. Note the model time as t_+. During this process, limit the first $n - 3$ variables to 0.01.
- Run the model with full negative drive until all the variables come onside that is have the same negative sign. Note the model time as t_-. During this process, limit the first $n - 3$ variables to -0.01.
- Compare the times t_+ and t_-. If $t_- > t_+$, then select negative drive. Otherwise select positive drive.
- Repeat the process.

It is possible to use this strategy for an n^{th} order system as long as $n > 3$. In cases where $n = 4$ and $n = 5$, it was found that the settling time of the strategy is greater than the normal IFMC without limits. Therefore, the IFMC strategy discussed here, is more suitable for higher order systems where the order of the system is greater than or equal to 6 ($n \geq 6$).

6.6 Important Points

- The Iterative Fast Model Control strategy can be applied to an n^{th} order system by limiting first $n - 3$ variables to 0.01 for a full positive drive and to -0.01 for a full negative drive, where $n > 3$.
- In case of systems where order $n > 10$, slugging is used to make system response smoother.
- The strategy is mainly for systems with order $n \geq 6$. It works for systems with order $n = 4$ and $n = 5$ but the settling time is compromised.
- If the limit of ± 0.01 is increased then the strategy gives oscillatory performance for higher order systems.
- When the system settles, the input variable or the first variable response shows low dithering which can further be reduced if the plant and model steplengths are equal to $1/number$ where number is a multiple of 8. But, this will increase the computational and plotting time.
- The simulation time for the systems with order $n \geq 10$ increases substantially.
- The initial conditions are taken as $x = 10$ and other variables equal to 0 because x is the last variable of the cascade and other variables integrate to it. Therefore with higher order systems, for other variables initial conditions need to be appropriately small otherwise it increases the computational time and scale of the plots also may require adjustments for a complete and clear presentation of system response with every change in initial conditions.
- In the system responses where $n \geq 10$ certain anomalies could appear (shown by arrows in Fig. 6.11). The possible factors could be, rounding of numbers, incompatible screen resolution or not so perfect slugging factor. However, the system response remains stable otherwise.

6.7 Conclusion

An alternative method to stabilize an unstable 6^{th} order system is stated. This method is then extended to systems up to a 11^{th} order system and the possibility of extending the strategy further up to an n^{th} order is stated, where $n > 3$. In case of 4^{th} and 5^{th} order the settling time gets compromised therefore this strategy is recommended for higher order systems where order $n \geq 6$. The system responses for 7^{th}, 8^{th} and 9^{th} order systems are found to be smooth and stable. The computational time when system order is greater than 10, increases. The possible reasoning behind the anomalies that may appear in system responses where $n \geq 10$, is stated.

Chapter 7

Lyapunov Stability Analysis

7.1 Introduction

Simulation studies have shown that the Iterative Fast Model Control strategy can bring systems of cascaded integrators to the state origin for third, four and fifth order when the fast model is an exact replica of the plant. Now an attempt will be made to apply the techniques of Lyapunov to provide a proof of stability.

For the Lyapunov direct method, we must find a function that maps the state-space into a scalar, such that when the value is reduced to zero the state is confined to the origin. It must also be shown that for asymptotic stability, the function is always reducing while the plant is controlled.

Now recall that the strategy is based on the predicted times (t_+ and t_-) for each sense of drive until the plant states have all come 'onside'. When the model drive is the same as the plant drive, this parameter (t_+ or t_-) will decrease by one second per second of elapsed time. However the value of predicted time for the other sense of drive may increase, switching occurring when it becomes just greater than its counterpart such that both can be termed as equal to each other.

If the control is asymptotically stable, both parameters (t_+ and t_-) will tend to zero as the origin is approached. It is therefore tempting to use this predicted time as a Lyapunov function. However the task remains of showing that in the sliding mode, when the drive alternates between extremes, this 'greater predicted settling time' will actually reduce monotonically.

Algebraic analysis is relatively easy for low order systems, but the difficulty increases significantly with order.

For systems where it is necessary to apply 'slugging' to achieve stability, whereby multiplying factors are applied to the system states when setting the initial conditions of the model. In this case it can no longer be asserted that the 'time to onside' for the applied drive will reduce at the same rate as the elapsed time.

7.2 Lyapunov stability of IFMC for a second order system

A second order cascaded integrator system is defined as $\frac{d^2x}{dt^2} = u$ where $u = \pm 1$. The system response with Iterative Fast Model Control is shown in Fig. 7.1. The initial conditions are $x = 20$ and $v = 0$.

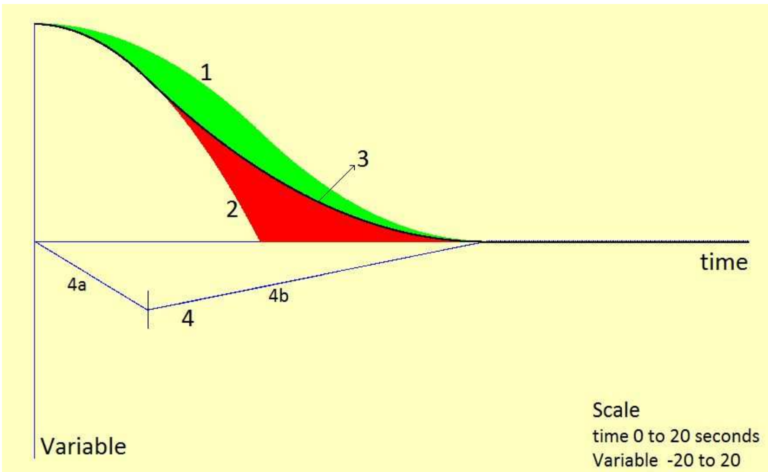


Figure 7.1: Second Order Response Reproduced for convenience

The mathematical analysis of this system response is carried out in section 4.2. It shows the relationship between model and plant variables and the system-model equations are derived. Final equations are reproduced here,

$$x_m(t) = x_p(T) + v_p(T)t + \frac{1}{2}ut^2 \tag{7.1}$$

$$v_m(t) = v_p(T) + ut \tag{7.2}$$

where T is the plant time, t is the model time. x_m and v_m are model variables and x_p and v_p are plant variables.

According to the Iterative Fast Model Control strategy, the model is run ahead in time, once with full positive drive, giving time t_+ when both model variables will have positive sign and once it is run with full negative drive, giving time t_- when both model variables will have negative sign. If $t_- > t_+$ then plant input will be full negative drive otherwise full positive drive will be applied to the system.

The decision of input is dependant on times t_+ and t_- and it is observed in section 4.2 that as the simulation progresses,

$$t_+ \ and \ t_- \rightarrow 0 \ as \ T \rightarrow \infty$$

Lyapunov's direct method is used where the Lyapunov function is defined in terms of t_+ and t_- and shown by equation 7.3

$$V(t(T)) = \frac{1}{2}\left(t_+^2 + t_-^2\right) \tag{7.3}$$

$$\dot{V(t)} = t_+ \cdot \dot{t_+} + t_- \cdot \dot{t_-} \tag{7.4}$$

The rationale of choosing t_+ and t_- for Lyapunov stability analysis is explained in the introduction of this chapter. In IFMC strategy at every instant of the plant time, future behaviour of the variables that is the behaviour of the system, is predicted. Once with full positive drive and once with full negative drive. These predictions after time T are shown in Fig. 7.2 which is reproduced here from section 4.2.

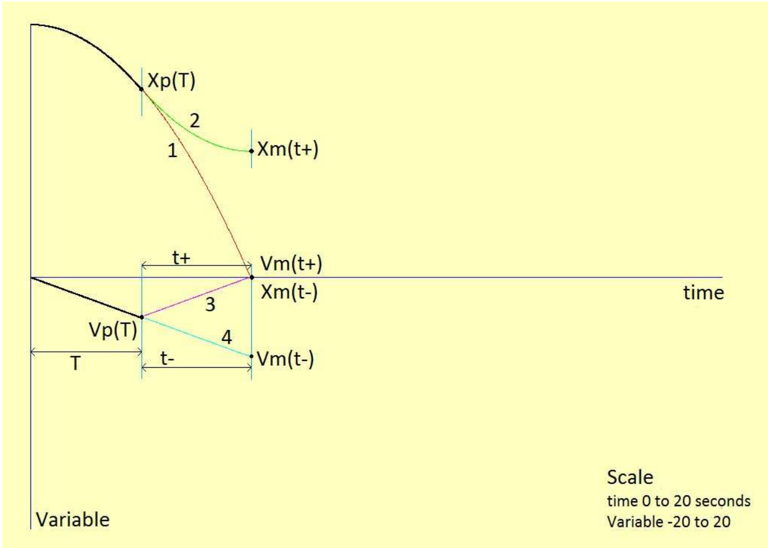


Figure 7.2: IFMC traces after time T when $t_{-}almost = t_{+}$

Fig. 7.2 shows that initially $t_{-} > t_{+}$ therefore input to the system is -1 until t_{+} is just ahead of t_{-} such that they appear equal, at this point the input switched to $+1$. Then the system enters in the sliding mode where t_{+} and t_{-} take over each other very quickly. The effective plant input is the slope of primary variable curve-4. Therefore the final Lyapunov function is defined in two parts

$$\dot{V(t)} = \dot{V_1(t)} + \dot{V_2(t)} \tag{7.5}$$

As the corresponding drive to t_{+} is full positive the change in t_{+} is $u = 1$ and as the corresponding drive to t_{-} is full negative change in t_{-} is $u = -1$ that is

$$\dot{t_{+}} = 1 \tag{7.6}$$

$$\dot{t_{-}} = -1 \tag{7.7}$$

The $\dot{V_1(t)}$, part of the equation 7.5, corresponds to curve 1. As the input to the system is -1, $\dot{t_{+}} = 0$ using equation 7.4,

$$\dot{V_1(t)} = 0 - t_{-} \tag{7.8}$$

To find the value of t_{-}, at the beginning predictions with full negative drive indicate that an input of -1 for time t_- will bring position x_p onside to a near zero value indicated by $x_m(t_-)$. Therefore using equation 7.1 and taking $x_m(t_-) = 0$ and solving for t_- gives

$$t_- = v_p(T) + \sqrt{v_p^2(T) + 2x_p(T)} \tag{7.9}$$

Using equation 7.3 and equation 7.9

$$\dot{V_1(t)} = -\left(v_p(T) + \sqrt{v_p^2(T) + 2x_p(T)}\right) \tag{7.10}$$

Fig. 7.2 shows that when t_+ is just greater than t_- such that they appear equal, the predictions with full positive drive indicate that it will bring plant velocity onside to a near zero positive value indicated by model velocity $v_m(t_+)$. Therefore using equation 7.2 and taking $v_m(t) = 0$ value for t_+ is found as

$$t_+ = -v_p(T) \tag{7.11}$$

At the same time Fig. 7.2 shows that predictions with full negative drive indicate that it will bring plant position onside to a value just below zero indicated by model position $x_m(t_-)$. The value of t_- is shown by equation 7.9.

For the $\dot{V_2(t)}$ part of the equation 7.5, using equations 7.4,7.6,7.11 and 7.9

$$\begin{aligned} \dot{V_2(t)} &= -v_p(T) - \left(v_p(T) + \sqrt{v_p^2(T) + 2x_p(T)}\right) \\ &= -\left(2v_p(T) + \sqrt{v_p^2(T) + 2x_p(T)}\right) \end{aligned} \tag{7.12}$$

Now combining equation 7.10 and 7.12 to give 7.5 show that $\dot{V(t)} < 0$ which shows that the system is asymptotically stable.

7.3 Lyapunov stability of IFMC for a third order system

A third order cascaded integrator system is defined as $\frac{d^3x}{dt^3} = u$ where $u = \pm 1$. The system response with the Iterative Fast Model Control strategy is shown in

Fig. 7.3. The initial conditions are $x = 20$, $v = 0$ and $a = 0$.

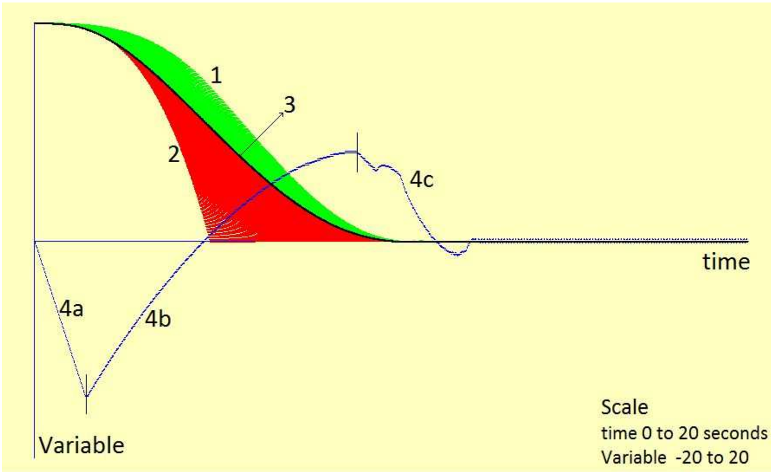


Figure 7.3: 3rd Order Response Reproduced for convenience Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: System Response x, 4(a,b,c): Primary Variable Curve

It is important to consider the primary variable curve because its slope gives the effective plant input. In Fig. 7.3, the primary variable curve is divided in 3 sections 4a, 4b and 4c. The switching points between 4a and 4b then 4b and 4c mark the change in the direction of the average input. During the 4c section of the curve the change in direction occurs on a few more occasions.

As discussed earlier in section 4.3 at every change in the direction of primary variable curve, a different variable comes onside. It is also shown that the equations become too complex to analyse the complete 4c section of the curve. Therefore, here curve 4c is considered up to the first direction change only, that is after switching from 4b to 4c until next direction change.

The mathematical analysis of this system response is carried out in section 4.3 where system-model equations are derived. Final equations are reproduced here,

$$x_m(t) = x_p(T) + a_p(T)t + v_p(T)\frac{t^2}{2} + \frac{ut^3}{6} \tag{7.13}$$

$$v_m(t) = v_p(T) + a_p(T)t + \frac{ut^2}{2} \tag{7.14}$$

$$a_m(t) = a_p(T) + ut \tag{7.15}$$

For the Lyapunov analysis, equation 7.3 and equation 7.4 are used here as well. Since it is a third order system, there will be a component of the Lyapunov function representing each section of the primary variable curve 4a,4b and 4c respectively. Hence the Lyapunov function will be

$$\dot{V(t)} = \dot{V_1(t)} + \dot{V_2(t)} + \dot{V_3(t)} \tag{7.16}$$

First $\dot{V_1(t)}$ part of the equation 7.3 is considered. It corresponds to the section 4a of the primary variable curve. During this part, the $t_- > t_+$ therefore the system input is -1. Hence the $\dot{V_1(t)}$ will be same as equation 7.3. To find the value of t_-, equation 7.13 gives

$$\frac{t_-^3}{6} - v_p\frac{t_-^2}{2} - a_p t_- - x_p = 0 \tag{7.17}$$

To find the value of t_-, First, equation 7.17 is differentiated with respect to plant time T. Same differentiation has been carried out in section 4.3 and the resulting equation 4.19 is reproduced here

$$\frac{dt_-}{dT} = \frac{v_p - t_- + a_p\frac{t_-^2}{2}}{\frac{t_-^2}{2} - v_p t_- - a_p} \tag{7.18}$$

In this region the plant drive is equal to -1 and the plant follows the predicted trajectory, therefore the value of $\frac{dt_-}{dT} = -1$. Substituting this value in equation 7.18 results in a quadratic equation 7.19

$$(a_p + 1)\frac{t_-^2}{2} - (1 + v_p)t_- + (v_p - a_p) = 0 \tag{7.19}$$

The roots of equation 7.19 give the value of t_- as

$$t_- = \frac{2(v_p + 1) \pm \sqrt{[2(v_p + 1)]^2 + 4(a_p + 1)(2a_p - 2v_p)}}{2(a_p + 1)}$$

$$\textit{that is}\;\; t_- = f(a_p, v_p) \tag{7.20}$$

Using equation 7.8 and 7.20 gives

$$\dot{V_1}(t) = -f(a_p, v_p) \tag{7.21}$$

Second $\dot{V_2}(t)$ part of the equation 7.3 corresponding to section 4b of the primary variable curve is considered. At the beginning of section 4b, t_+ gets just ahead of t_- such that they appear equal.

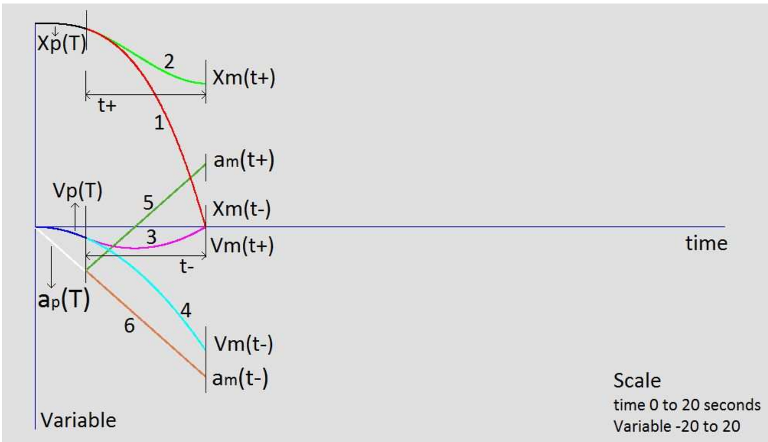


Figure 7.4: Simulation of Iterative Fast Model Control with Third Order System Variables

At this stage, Fig. 7.4 shows that x_p is already onside, a full positive drive for time t_+ will bring a_p onside as indicated by $a_m(t_+)$ and v_p will come onside but very close to zero indicated by $v_m(t_+)$. Therefore equation 7.14 gives

$$0 = v_p + a_p t_+ u \frac{t_+^2}{2}$$

Solving the above equation for t_+ gives

$$t_+ = \frac{-2a_p \pm 2\sqrt{a_p^2 - 2v_p}}{2}$$

which is nothing but

$$t_+ = -\left(\frac{2a_p \mp 2\sqrt{a_p^2 - 2v_p}}{2}\right)$$

$$\textit{that is } \; t_+ = -f_1(a_p, v_p) \tag{7.22}$$

At the same stage of beginning of curve 4b, a full negative drive for time t_- will keep a_p and v_p onside indicated by $a_m(t_-)$ and $v_m(t_-)$ respectively. It will bring x_p onside but very close to zero as indicated by $x_m(t_-)$. The value of t_- is given by 7.20.

Using equations 7.4,7.6,7.22 and 7.20, it can be shown that

$$\begin{aligned} \dot{V_2}(t) &= -f_1(a_p, v_p) - f_2(a_p, v_p) \\ &= -(f_1(a_p, v_p) + f_2(a_p, v_p)) \end{aligned} \tag{7.23}$$

The third part $\dot{V_3}(t)$ corresponds to first direction change in 4c after the switching from 4b to 4c. Fig. 7.5 is reproduced from section 4.3 for reference.

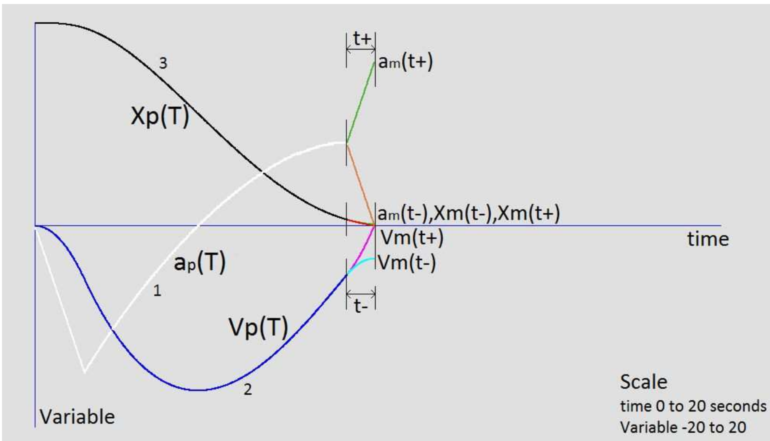


Figure 7.5: Simulation of Iterative Fast Model Control when t_-=t_+ second time

It shows that at the beginning of 4c, for a full positive drive x_p and a_p are already onside. A full positive drive for time t_+ will bring velocity onside to $v_m(t_+)$ which is almost zero. Therefore, the value of t_+ will be same as equation 7.22.

For a full negative drive, velocity v_p is already onside and if applied for a time t_-, it will bring x_p and a_p onside but close to zero as indicated by $x_m(t_-)$ and $a_m(t_-)$ respectively. Therefore, the value t_- can be found using equation 7.13 or equation 7.15. Here, equation 7.15 is considered, which gives

$$t_- = a_p \tag{7.24}$$

$$\textit{that is } t_- = f_3(a_p) \tag{7.25}$$

Using equations 7.4,7.6,7.22 and 7.24 it can be shown that

$$\begin{aligned} \dot{V_3}(t) &= -f_1(a_p, v_p) - f_3(a_p) \\ &= -(f_1(a_p, v_p) + f_3(a_p)) \end{aligned} \tag{7.26}$$

From equations ,, and it is concluded that $\dot{V(t)} < 0$ which shows the system is asymptotically stable.

However, this Lyapunov function represents more than 90% of the system response. For further change in directions in 4c, it was very difficult to keep track of which variables come onside because the system response is about to reach settling or equilibrium point. Also, the model times $(t_+ \ and \ t_-) \rightarrow 0$, as the plant time $T \rightarrow \infty$ and system converges.

This can be seen in Fig. 7.4 and Fig. 7.5. The remaining less than 10% part of the system response (curve-3) (corresponding to rest of 4c) converge to settling point or equilibrium point. This is shown in Fig. 7.3 (curve-3). Therefore, it can be assumed that the corresponding Lyapunov function components would have the 'negative' sign, indicating that it also converges.

7.4 Lyapunov stability of IFMC for Higher Order Systems

It has been shown that, the Iterative Fast Model Control strategy can be applied to higher order systems up to 11^{th} order and it is speculated that the strategy can be extended to an n^{th} order system.

As the order of the system increases, the number of variables also increase. It creates a complex situation where it becomes difficult to identify which variable is coming onside to a near zero value. A 5^{th} order system is chosen for stability analysis.

Response of a fifth order system, $\frac{d^5x}{dt^5} = u$ where $u = \pm 1$, with initial conditions $x = 20$, $v = 0$, $a = 0$, $j = 0$ and $k = 0$, over a 70 second period is shown in Fig. 7.6.

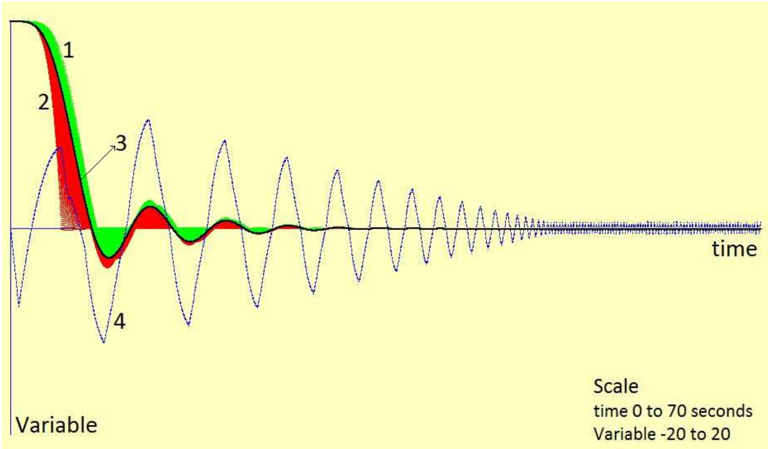


Figure 7.6: Response of a Fifth order system for 70 seconds

The system response in Fig. 7.6 shows initial overshoots but eventually settles down. In the Iterative Fast Model Control, if model time $t_+ > t_-$ then the plant input is +1. As the plant time increases, t_+ decreases at a rate of one second per second. At the same time t_- increases until it overtakes t_+.

In cases of the second and third order systems, the model time t_- increases steadily. But, in cases of higher order systems, there could be a change in the order of the variables to come onside. Then this trailing model time t_- may not increase steadily but may instead 'hop'.

Therefore, the Lyapunov reasoning that was applied in cases of the second and third order systems is unfortunately not valid for higher order cases.

7.5 Important Points

- In Iterative Fast Model Control, a system model is run ahead in time once with full positive drive and once with full negative drive until all the model variables attain same sign as that of corresponding input and time of this process is noted as t_+ and t_- respectively.
- At every instant the model is updated with new plant values and the process is repeated until the system settles down. Therefore, as the system approaches settling point, t_+ and t_- decrease.

- Lyapunov's direct method is used for analysis where a Lyapunov function is defined using t_+ and t_- as parameters. The responses of second and third order systems are considered up to the settling point.
- During Iterative Fast Model Control at every instant predictions with corresponding full positive and full negative drive indicate which variable will come onside last to a value that is near zero. At every direction change of the primary variable curve a different variable comes onside to the near zero value.
- Using system-model equations, values of t_+ and t_- can be found out to find the final Lyapunov function.
- In cases of second and third order system, it is shown that the Lyapunov function and the system response converge to the settling point that is origin. This shows the system stability.
- In case of higher order systems, observing which variable comes onside last is difficult. The model times may not necessarily be a continuous increments in time. Therefore the Lyapunov reasoning of second and third order system becomes invalid.

7.6 Conclusion

The model prediction times and the Lyapunov's direct method are used to establish the stability of the system. It is shown that, system responses of second and third order systems with Iterative Fast Model Control are stable. As the the order of the system increases, the number of variables also increase.

Thus keeping track of last variable to come onside becomes difficult. The increments in model times could be irregular which makes extension of Lyapunov arguments of second and third order case, to higher order cases, invalid.

Chapter 8

Iterative Fast Model Control: Applications

8.1 Introduction

The applicability of Iterative Fast Model Control is demonstrated using two examples. An Aircraft lateral control is considered whereby on approach for landing, an aircraft is aligned with the center of a runway by an autopilot system. Specifications of Boeing 747-400 and landing guidelines from Federal Aviation Authority, USA are used. The system is considered in principle only, to test IFMC strategies with state constraints. It is an example of a fourth order system. Simulation results have shown that IFMC gives effective performance.

A Ball and Beam experiment experiment has been conducted to demonstrate that Iterative Fast Model Control actually works in practice. The ball and beam system is a third order system, but the use of vision sensing for the ball position adds a small delay that increases the difficulty beyond that of the original experiment conducted by (Billingsley 1968). The aim of the experiment is to balance a table tennis ball at the center of the beam using Iterative Fast Model Control. The performance of the strategy is found to be satisfactory and it showed excellent recovery from external disturbances. The setup and process of experiment is stated through different algorithms that outline the methodology.

8.2 IFMC for Aircraft Lateral Control

The Aircraft Lateral Control issue is considered in broad terms, without specific details of any specific system. Some of the terms used in this section can be defined as

Bank angle - angle made by the wings of an aircraft with the horizontal axis.

Rollrate - rate of change of bank angle

Heading - the angle between the aircraft and the center of the runway.

Ailerons - the sections at the back of the aircraft wings that moved up and down to change the bank angle.

In a hypothetical situation, suppose the aircraft is at a distance 'x' from the center of the runway and its bank angle is 'ϕ' as shown in Fig. 8.1.

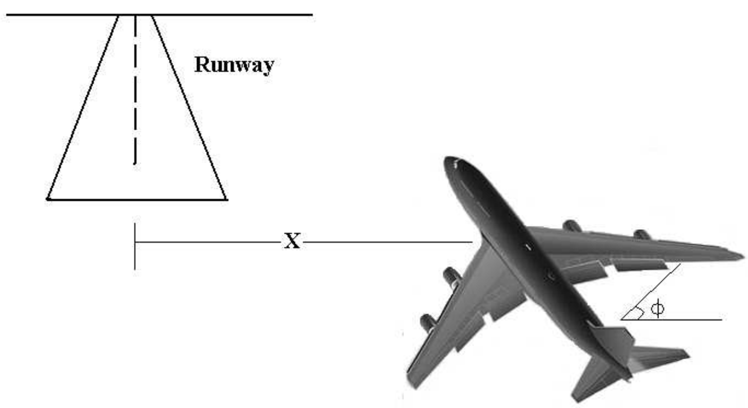


Figure 8.1: The landing of the Aircraft

Then, in order to direct the aircraft on the runway, several variables are involved. These variables can be stated as position x from the runway, bank angle of the aircraft ϕ, rate of change of bank angle roll rate $\dot{\phi}$, heading of the aircraft ψ. If the autopilot system is considered with finer details then the number of variables and consequently the order of the system will increase. For now, to simplify the formulation of the problem, only a fourth order system is considered. The system equations can be written as follows

$$\dot{\alpha} = u * k - c\alpha \tag{8.1}$$

$$\dot{\phi} = \alpha \tag{8.2}$$

$$\dot{\psi} = \frac{g}{v}\phi \tag{8.3}$$

$$\dot{x} = v\psi \tag{8.4}$$

The equation 8.1 gives the rate of change of rollrate. For simplicity it is considered that this change in the rollrate is caused by damped aileron angle. The damping factor is a term that involves a constant multiplied by rollrate. Aileron angle is driven by a motor, so can be considered as the integral of the input also there will be a rollrate feedback term which is a lag.

The rollrate gives the rate of change of bank angle equation 8.2. The bank angle multiplied by a ratio of gravity over velocity gives the rate of change of heading equation 8.3. Finally, heading multiplied by the velocity gives the change in the distance equation 8.4.

For the initial conditions let us consider that a Boeing 747-400 aircraft is heading towards a runway for landing, this process is called as approach. There are specific guidelines about approach speed and the distance from the runway (Branch 2007). (of Airport Safety & Standards 2011) and (Branch 2007) indicates that a Boeing 747 falls in the category D of the approach speed. Its approach speed is required to be 154 knots at a distance of 2.3 miles that is 3700 m from the center of the runway.

Therefore, let us consider when the control is handed over to the automatic system, the approach/landing velocity of our aircraft is 154 $knots$ that is 285 km/h (80 m/s). Assume that the aircraft is heading at an angle of 60° (1.05 rad) and is banked at an angle of 30° that is 0.5236 rad. Assuming the initial rollrate of $0°/s^2$, the initial conditions can be summarized as $x = 3700$, $\psi = -1.05$ since the aircraft is heading towards runway initial ψ will be negative, $\phi = 0.5236$, $rollrate = 0$ with the constant velocity of $v = 80\ m/s$.

The simulation of the aircraft lateral control along with the performance of different variables is shown in the Fig. 8.2.The variables are multiplied by respective coefficients for better visibility in Fig. 8.2.

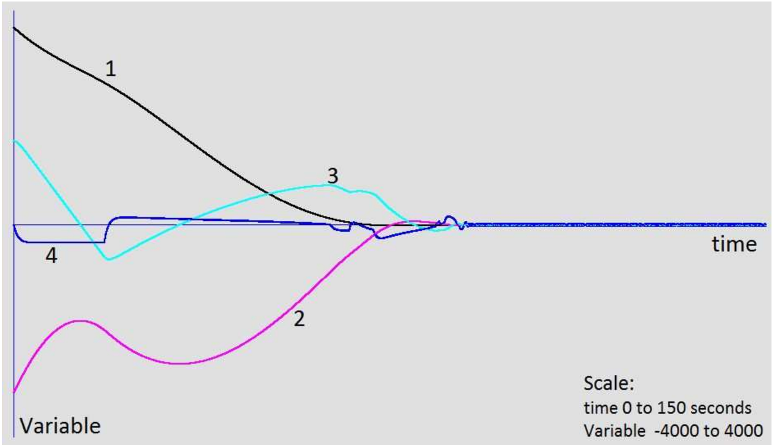


Figure 8.2: The Aircraft Rollangle (Bank-angle) Control using IFMC Strategy
Curves: 1: Distance x, 2: 3000* Heading (ψ), 3: 3000*Rollangle (ϕ) and 4: 8000*Rollrate (α)

In an aircraft lateral control several constraints are involved. If the bank angle is excessive then the apparent weight of the passengers is increased. Similarly rollrate can not be more than certain limit so that the bank angle does not change too quickly. A Boeing 747-400 Autopilot Flight Detector System (AFDS) (Boeing 2001) indicates that the bank limit varies between 15° to 25°.

It is assumed that the AFDS puts a bank angle limit of 16° that is 0.2792 rad. So that it would be easier to see the limitations and the bank angle curve performance. The Fig. 8.3 shows the response of bank angle with this limit in place and the distance x for reference. It shows that the bank angle phi stays within the specified limits.

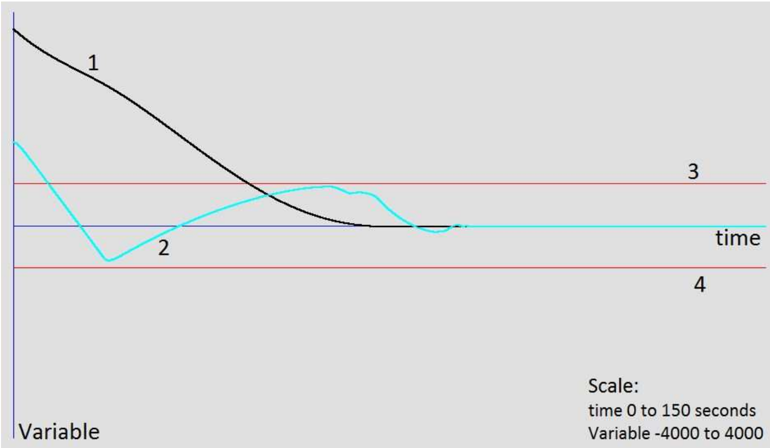


Figure 8.3: The performance of variable ϕ that is bank angle with limits Curves: 1: Distance x, 2: 3000*Bank angle (ϕ), 3: 3000*Upper limit of *phi* (0.2792), 4: 3000*Lower Limit of *phi* (-0.2792)

8.2.1 State Constraint - Wind-Gust

Suppose that after 7 seconds a strong wind-gust changes the bank angle of the aircraft to 43° that is 0.7505 *rad*. Fig. 8.4 below shows the simulation with the effect of wind-gust on various variables.

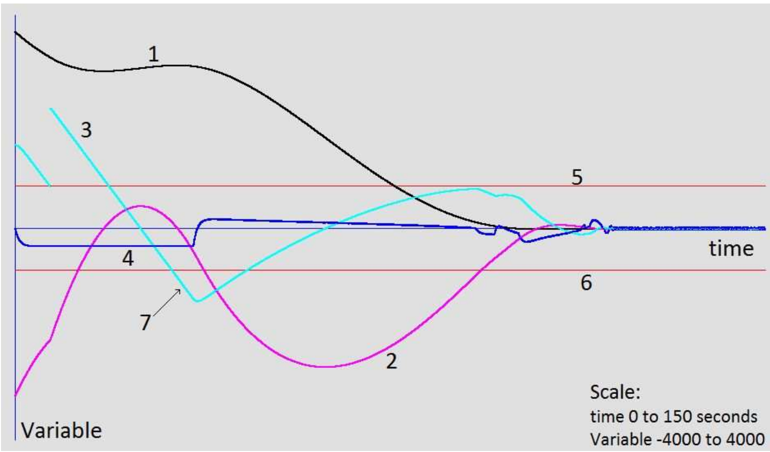


Figure 8.4: The performance of variables with wind-gust Curves:- 1: Distance x, 2: 3000*Heading (ψ), 3: 3000*Bank Angle (ϕ), 4: 8000*rollrate (α), 5: 3000*Upper limit of *phi* (0.2792), 6: 3000*Lower Limit of *phi* (-0.2792), 7: Bank Angle curve crosses the limit

When the performances of variables in Fig. 8.4 are compared with those of in Fig. 8.2 it can be seen that the course of flight for the aircraft changes (curve 1), the heading also changes (curve 2) but the bank angle crosses the lower limit of -0.2792 *rad* (pointed by 7).

This crossover of the bank angle can be prevented by incorporating this state constraint in to the Iterative Fast Model Control Strategy. Iterative Fast Model Control Strategy with State Constraints is introduced and discussed in chapter 5. Here it is presented in the context of aircraft bank angle control.

Now in the current example of aircraft bank angle control, the subsequent variable is the bank angle ϕ. Thus, according to the strategies in chapter 5, the limits on ϕ need to be implemented in the fast model with flags.

When the fast model is run ahead in time with full positive drive, the maximum time t_+ when all the variables become positive, is calculated. During this process the condition $\phi < -0.2792$ (lower limit) is flagged, to say Flag-1. When the maximum time t_- when all the variables become negative, is calculated, the condition $\phi > 0.2792$ (upper limit) is flagged to say Flag-2. Then when t_+ and t_- are compared to decide which input to give to the plant these flags are considered.

If $t_- > t_+$ then input drive will be full negative but if flag-2 has occurred then the drive will be changed to opposite that is full positive. If $t_+ > t_-$ then input drive will be full positive but if the flag-1 has occurred then the drive will be changed to opposite that is full negative.

The simulation of aircraft control with wind-gust using IFMC with constraints is shown in the Fig. 8.5 and the strategy is outlined at the end of this section.

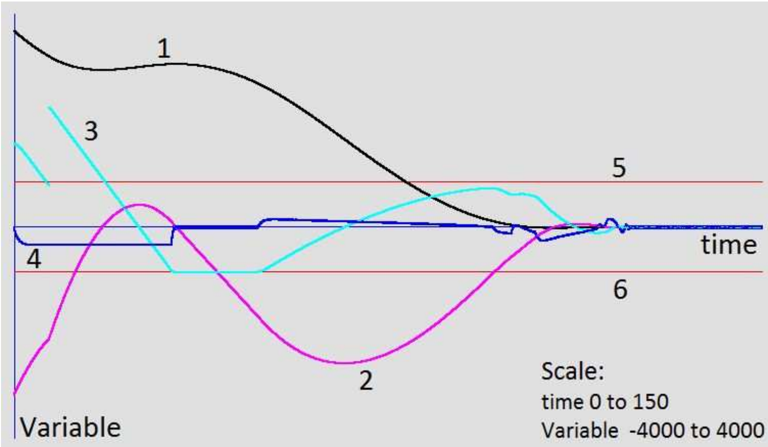


Figure 8.5: IFMC with bank angle limits incorporated Curves: 1: Distance x, 2: 3000*Heading (ψ), 3: 3000*Bank Angle (ϕ), 4: 8000*rollrate (α), 5: 3000*Upper limit of phi (0.2792), 6: 3000*Lower Limit of phi (-0.2792), 7: Check for overshoot

Upon scaling the Variable axis from (-4000 to 4000) to (-23 to 23), it is found that there is an overshoot in the distance curve. It is shown in the Fig. 8.6.

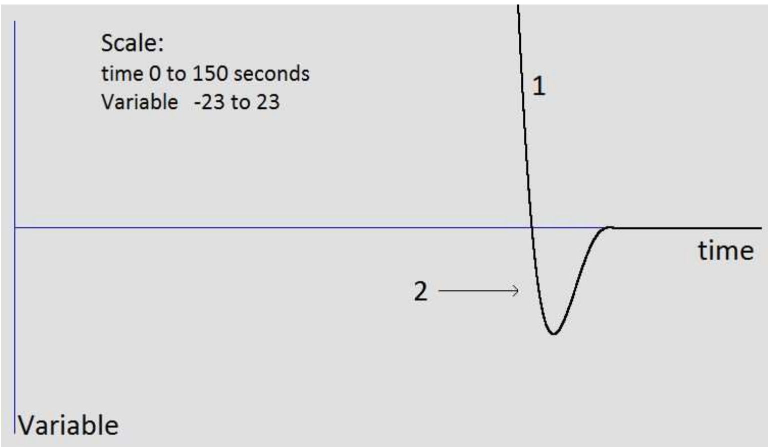


Figure 8.6: Scaled Variable axis with distance x. Curves:- 1: Distance x, 2: Overshoot

As shown in section 4.4.2, in IFMC for a fourth order system, slugging can be used to eliminate the overshoot. Therefore, a slugging factor of 1.2 is used. The variables, ϕ and ψ are multiplied by 1.2 as they are assigned to respective fast model variables. The improved response is shown in Fig. 8.7 and the performance of other variables with slugging is shown in Fig. 8.8.

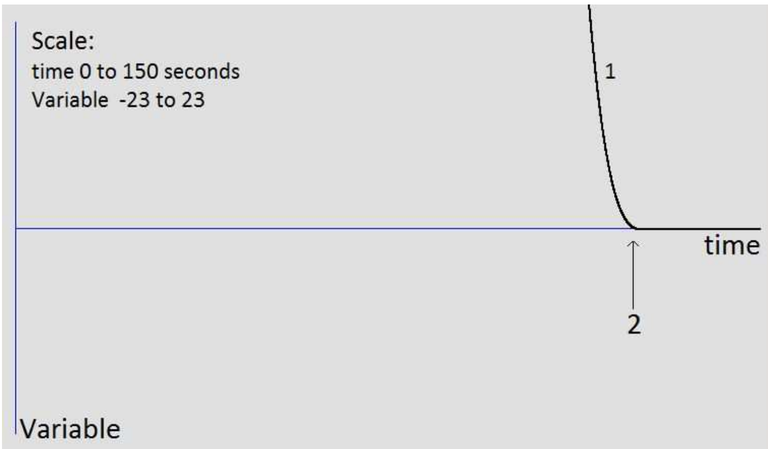


Figure 8.7: Distance x curve improvement with Slugging. Curves:- 1: Distance x, 2: No overshoot

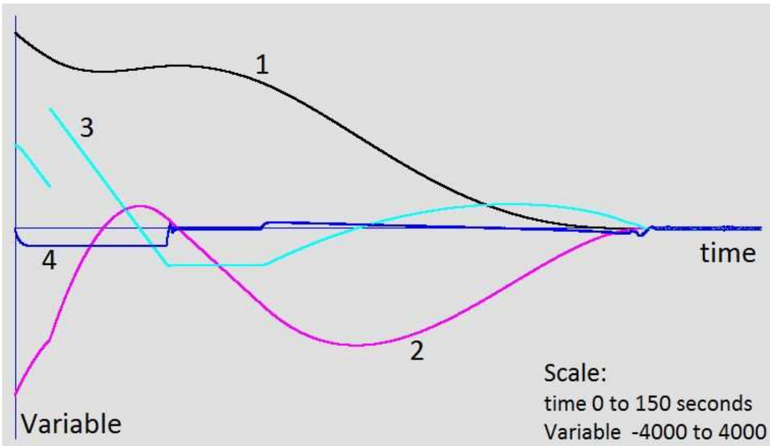


Figure 8.8: Performance of other variables with slugging. Curves:- 1: Distance x, 2: 3000*Heading (ψ), 3: 3000*Bank Angle (ϕ), 4: 8000*rollrate (α)

8.2.2 Disturbance - Turbulence

Now Suppose that the aircraft faces turbulence. The turbulence will most likely have its effect on rollrate. This effect was incorporated in the simulation by adding a random number to the rate of change of rollrate $\dot{\alpha}$ of the plant using rnd function of Visual Basic 6. The effect of turbulence is seen in the Fig. 8.9.

The simulations show that the rollrate curve is a little noisier. But, it does not have any detrimental effect on the performance of the entire system. There appears some dithering after the system settling points. The actual rollrate values are very small but in simulation plotting they are multiplied by a factor of 8000 for better visibility. Thus, this dithering is negligible.

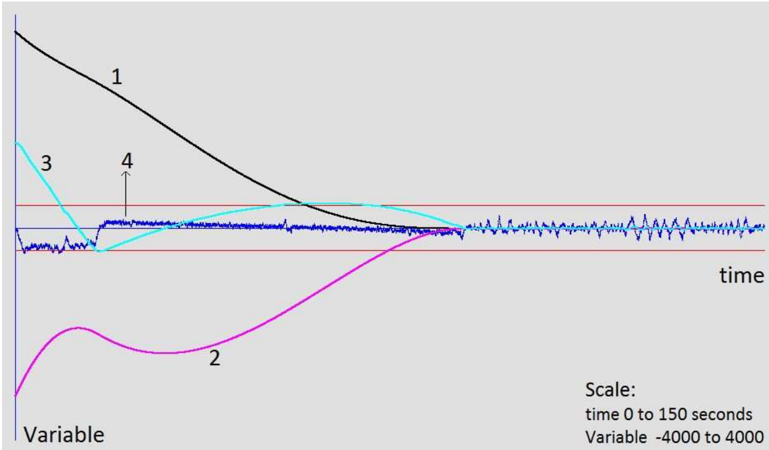


Figure 8.9: Iterative Fast Model Control in presence of disturbance/turbulence. Curves:- 1: Distance x, 2: 3000*Heading (ψ), 3: 3000*Bank Angle (ϕ), 4: 8000*rollrate (α)

8.2.3 IFMC with State Constraints for Aircraft Control

The IFMC strategy that can deal with different constraints associated with aligning an aircraft with the center of the runway can be outlined as follows

Set the model with aircraft variables x, ψ, ϕ, $rollrate$ to add slugging, multiply ψ and ϕ each by 1.2.

Run the model ahead in time with full positive drive until all the variables come onside that is have the positive sign note the time as t_+.

In this process flag the condition when $\phi < -0.2792$ as Flag-1.

Run the model with full negative drive until all the variables come onside that is have the negative sign note the time as t_-.

In this process flag the condition when $\phi > 0.2792$ as Flag-2.

Compare the times t_+ and t_-. If $t_- > t_+$ then apply full negative drive but if flag 1 has occurred then change the drive to full positive. Otherwise apply full positive drive but if flag 2 has occurred then change the drive to full negative.

Repeat the process

The change in ϕ was incorporated by assigning 0.733 to ϕ for 20 cycles at an interval of 0.01. The turbulence effect was incorporated by adding $0.5 * (Rnd - 0.05)$ to the rate of change of rollrate of the plant.

8.2.4 Important points

- The Iterative Fast Model Control strategy delivers near time optimal control performance. A plant model is run ahead in time with full positive and full negative drive. A decision on control input is made by looking at the time when all the model variables attains the sign same as that of corresponding drive. The drive corresponding to the greater time is then applied to the plant.
- The 'slugging' which is a deliberate plant model mismatch, can be used to remove overshoots in case of a fourth order system. The middle two variables of a fourth order cascaded system should be multiplied by a suitable slugging factor.
- To deal with the state constraints, flag the limits in the appropriate model during simulation and change the drive according to occurrence of these flags.

8.3 The Ball and Beam Experiment

The application of Iterative Fast Model Control strategy in practice is demonstrated using a ball and beam experiment. The aim of the experiment is to balance table-tennis ball on a beam using a stepper motor and Iterative Fast Model Control Strategy. After the control routine is executed, the ball should move from its initial position to the center of the beam and then strategy will try to maintain its position at the center. This will be achieved be tilting the beam to the right or left by connecting it to the shaft of the stepper motor. The input to the motor will be decided by the control algorithm.

8.3.1 Experiment Components and Setup

The components or apparatus used in the experiment are listed below

- A beam that rotates around the z-axis, mounted on a stand. The beam had a groove in the middle along its length so that the ball could rest on it and move freely to either end.
- A permanent magnet stepper motor.
- A circuit with a ULN2803A transistor chip and 4 LED's to indicate the input value.
- A Logitech web-camera.
- A Dell Optiplex GX-260 computer with Pentium-4 2.4 GHz processor and 512 MB Ram. The OS is Windows-XP and Microsoft Visual Basic 6.0 is used to interact with the beam.
- A data cable.
- A 20 V - 3 A power supply to give power to the stepper motor.
- A table-tennis ball.

The experiment is set up in the following way

- The beam is connected to the stepper motor in such a way that clockwise rotation of the shaft moves left side of the beam upwards and anticlockwise rotation of shaft moves right side of the beam upwards.
- The stepper motor is connected to the output pins of the transistor chip.
- The interface circuit is connected to the computer's printer port using a cable that is connected to the input pins of the transistor chip.
- The stepper motor is powered with 10 V using a 20V-3A power supply connected to the power input pins of the transistor chip.

- The web-camera is mounted about 1 m above the beam to track the position of the ball on the beam and pass it to the computer via usb cable. The image from the camera is refreshed every 33 ms.

The final system setup is shown in Fig. 8.10.

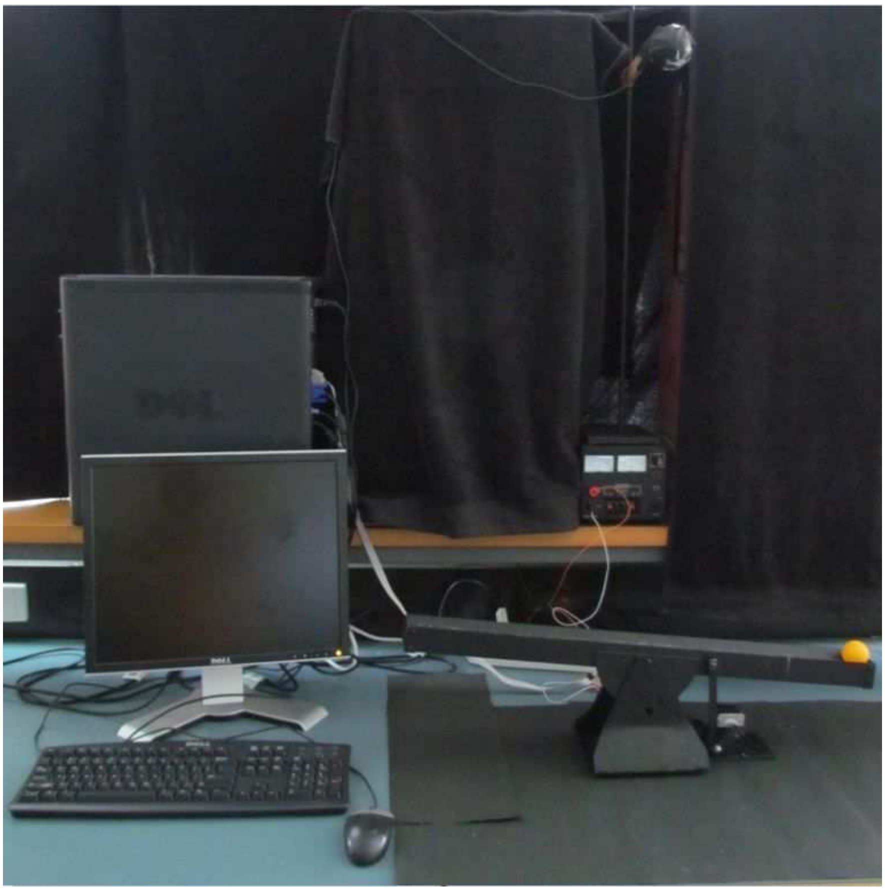


Figure 8.10: The experiment setup

8.3.2 System Identification and Modeling

In Iterative Fast Model Control, a model of the system is run ahead time. It predicts the system response, based on which the input to the system is decided. Therefore, it is very important to model the system as closely as possible. If the model of the system does not represent the system properties correctly then the predictions of the model will not be close to the system response, thereby resulting in the incorrect choice of input. This can give an inferior or undesired system performance.

The system variables are identified as position of the ball x, velocity of the ball v, acceleration of the ball a, tilt of the stepper motor $tilt$, rate of change in tilt $tiltrate$. At this stage only a third order cascaded integrator system is being considered where the integrated tilt gives the velocity of the ball and the integrated velocity gives the position of the ball. Therefore only x, v and $tilt$ are used in the model of the system.

The differential equations of the system are identified as follows

$$\dot{tilt} = Tc \cdot u \quad (8.5)$$

$$\dot{v} = Ka \cdot tilt \quad (8.6)$$

$$\dot{x} = v \quad (8.7)$$

Where Tc is the timer constant and Ka is the acceleration constant. The timer constant needs to be considered because the input to the actual stepper motor is given at every 33 ms. Therefore, Tc needs to be greater than 30.30. It shows that the model tilt will also receive, same input per second, as the actual motor.

The constant Ka is the acceleration constant. The stepper motor controls the tilt in finite increments, the steps synchronised with the frame capture. A count of these steps will give a measure of tilt, but the tilt datum has to be determined.

This is done by slowly tilting the beam in the sense that, it will cause the ball to move from its initial resting place at one of the ends of the track. As soon as it is seen to move, the tilting ceases and an estimate is made of the acceleration. The ball will now have traveled to the opposite end. Again the track is tilted slowly, now in the opposite direction and the estimate of acceleration is repeated.

There are now two values known for acceleration known for two opposing tilt values. By interpolation the value of the tilt count is now known for zero acceleration, while the ratio of the differences will give a parameter linking acceleration to tilt count that can be used in the fast model which is acceleration constant Ka.

The value of Ka is calculated in real time. It is constant for that set of system run. If the system is re-run the value of Ka will be different. But it will be constant for that particular set of system run.

Camera monitors ball position, simply by identifying the first bright pixel of the bright ball on the dark beam. This is then used to estimate the ball velocity, using a high-pass filter, and for calibration it is differentiated once again. The acceleration is also estimated in similar way. Approximation is illustrated using velocity equations.

$$xslow = xslow + velocity * dt$$
$$velocity = (x - xslow) * (somenumber)$$

8.3.3 Methodology

Microsoft Visual Basic 6.0 is used to implement the strategy on the ball and beam experiment. As soon as the project is run, a routine of $Form\_Load$ initializes the bit pattern for the stepper motor. It also initializes, a module named Squiz (to interact with camera) and the rate of refreshing the image which should be 0.033 in order to synchronize with camera image refreshing rate. The Graphical User Interface (GUI) designed for the experiment is shown in Fig. 8.11.

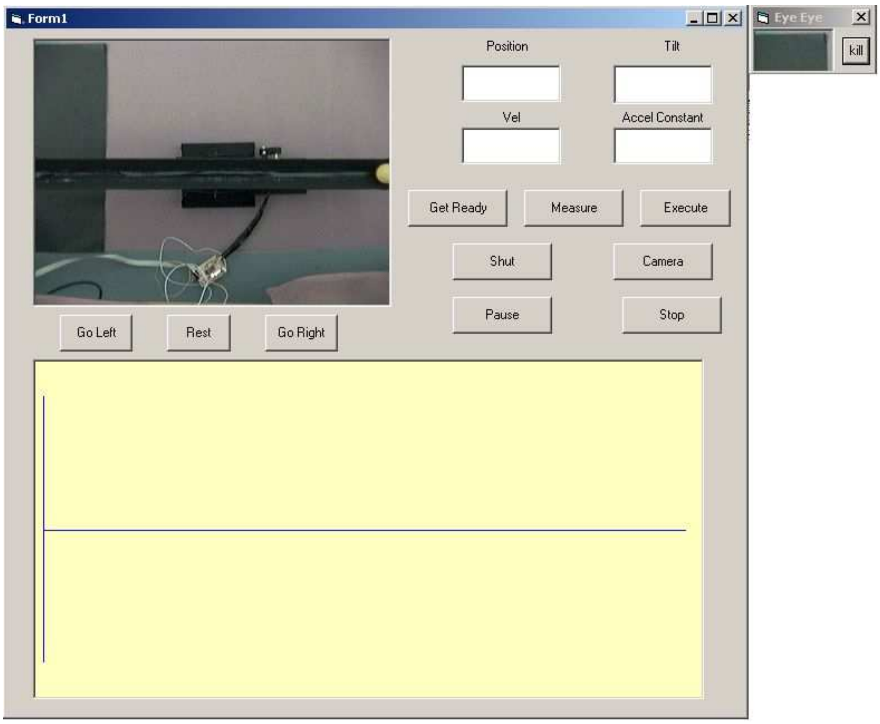


Figure 8.11: GUI - for the experiment

Fig. 8.11 shows, there are different modules. The caption on each button sums up its operation. Three buttons below the image of ball and beam are the manual controls. Go Left tilts the beam to left causing the ball to move to left and Go Right moves the ball to right by tilting the beam to right. The Rest pauses the operation.

The Camera button, when clicked it gives the image from the camera. It is used to adjust the beam position. It calls a subroutine called 'findball' which is used to track the position of the ball. Findball uses the image data and puts a horizontal red line in the middle of the captured image.

The ball moves along the red line, therefore the beam should be perfectly aligned with the red line horizontally such that when the ball is put on the beam the red line passes through center of the ball. Hence to achieve this some manual beam adjustment is required.

The 'Get Ready' is used to level the beam horizontally, before the strategy is applied to balance the ball. The 'measure' module executes the IFMC strategy. The 'Execute' module, executes Get Ready and measure together to avoid delays in executing the strategy after Get Ready. The Shut closes the application whereas 'pause' and 'stop' performs respective operations.

Tracking the ball

Squiz

The image of the ball and beam is captured using a separate module called Squiz. It interacts with the camera using a module called 'Eye Eye'. This module creates a small window that appears to the right of main Form1 window in Fig. 8.11.

Squiz identifies the ball as a bright object in the image. When the beam is aligned with the red line lengthwise, the ball should be placed on the beam such that the red line passes through the approximate center of this object. The x coordinate of this bright object's location gives the position of the left edge of the ball.

The values of position x on the beam range from 0 to 316 units (left end to right end). Therefore, the ball is supposed to be balanced in the middle at 158. So, the position in the middle should be zeroed by subtracting 158 from position x.

The position of the ball on the beam is tracked using a routine called 'findball'. The algorithm of findball can roughly be summed up as follows

- Use Squiz.SnapToClipboard command to capture the image and put it on the clipboard.
- Read the image data from the clipboard and draw a red line at the middle of the Pict which represents the picture on the main form.
- Identify the location of the bright object in the picture and assign the x-coordinate value of the left edge to position x.
- Draw a tiny white line of length 20 units (10 units above and below x) such that x is at the center of this line.
- Make the value of the position equal to zero at the middle by subtracting 158 from x.
- Calculate the velocity by using current ball position and a low pass filter as shown in section 8.3.2 .
- Calculate the acceleration using a method similar to velocity but using velocity instead of position.
- To increase or decrease the tilt, add input 'u' to the tilt. Output command ((tilt+offset) And 7) to the motor.
- Update time.

'findball' returns the position of the ball as well as velocity and acceleration every 33rd ms. Therefore, it is important to synchronize the input process with the refreshing of the image. So, a small process of giving the input to the motor is added in findball.

Get Ready

The Get Ready button when clicked calls the 'Calibrate' routine. This routine levels the beam horizontally and calculates the acceleration constant and tilt during the process of levelling. Its algorithm is outlined below.

- Initialize values of tilt, motor-offset and velocity as zero and locate the ball on the beam using findball.
- Tilt the beam to move the ball to opposite end. Stop tilting when the ball starts rolling in the right direction.
- Take a pause to let the ball roll to the end.
- Find the location of the using findball and tilt the beam until ball passes middle point towards the other end.
- Get the value of acceleration from findball and record the acceleration and tilt values as a1 and t1.
- Again Tilt the beam until the ball starts rolling with very low velocity.
- Get the value of acceleration from findball and record the acceleration and tilt values as a2 and t2.
- Calculate the acceleration constant ka by dividing a2-a1 by t2-t1.
- If $ka < 70$ then multiply ka by 1.25 until $ka > 85$
- Calculate the offset by (tilt And 7) and calculate the tilt by subtracting offset from the ratio of a2 and ka which multiplied by a factor of 1.25.
- Tilt the beam very slowly until the ball barely moves towards other end. This indicates that the beam is almost horizontal.

Implement IFMC

The algorithm of Iterative Fast Model Control strategy to control the position of the ball on a beam is outlined below

- Initialize the model tilt, velocity and position equal to zero. Assign model steplength as 0.01 and initialize t=0.
- Implement the strategy in a loop that continues to run until stop button is clicked.

- In the loop: use findball to get the plant values of tilt, velocity and acceleration. Assign these values to the model tilt, velocity and position respectively. Add the calibration error constant equal to 30 to plant position.
- Run the model with full positive drive until all the variables come onside that is have the positive sign. Note this time as t_+.
- Run the model with full negative drive until all the variables come onside that is have the negative sign. Note this time as t_-.
- Compare times t_- and t_+, whichever is greater assign a drive corresponding to it, to input u.
- Repeat the process

The input u is given to the motor at every 33rd ms when findball is executed to refresh the image.

8.3.4 Results

In the preparation of the experiment the beam is first tilted to put the ball at one end or the other. Therefore this tilt should be noticeable. The beam tilted with such tilt is shown in Fig. 8.12

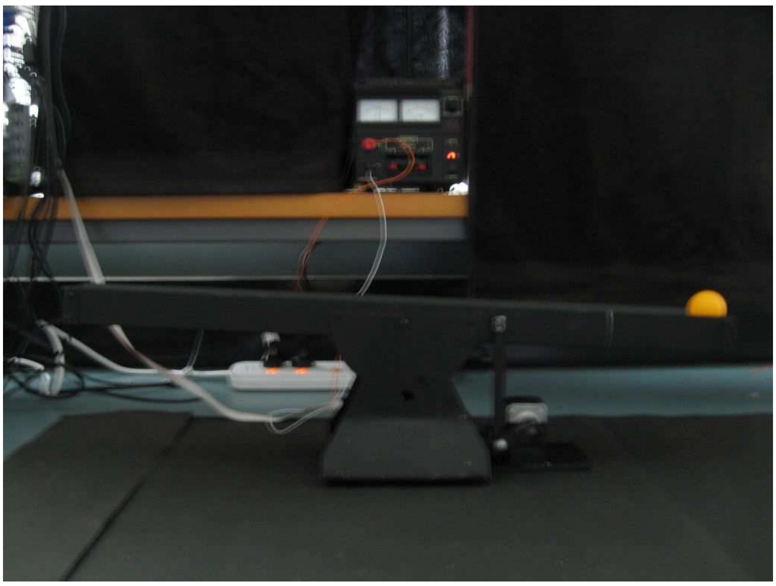


Figure 8.12: Ball and Beam Experiment: Initial Starting Position

The 'Get Ready' routine is executed to calculate the acceleration constant that relates the acceleration in pixels per second per second to the tilt angle measured in stepper motor counts and to determine the datum of the beam angle, when it is level. The path taken by the ball is traced by the program and plotted. It is shown in Fig. 8.13.

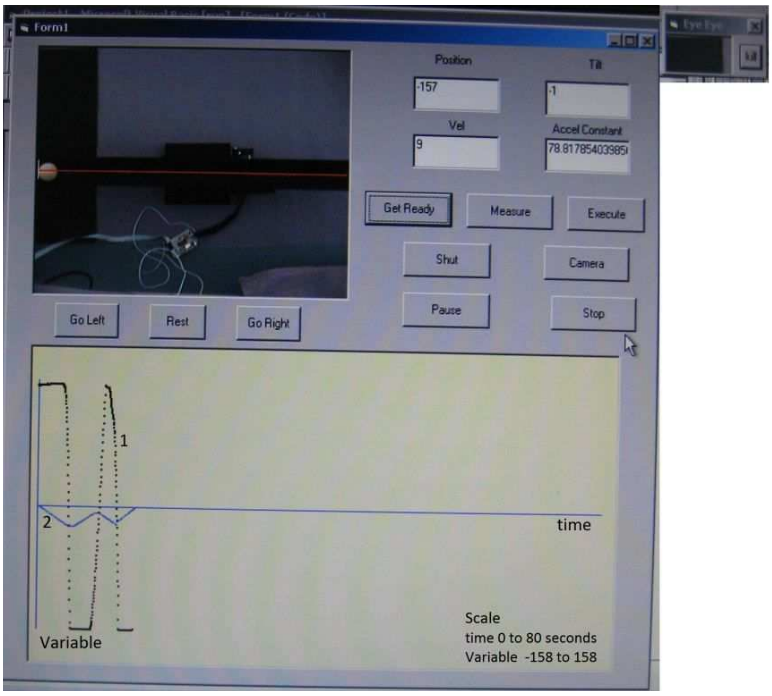


Figure 8.13: Ball and Beam Experiment: Beam leveling process - Get Ready Curves:- 1: Path taken by the ball 2: tilt multiplied by 2 for better visibility

Final beam level and ball position are shown in Fig. 8.14. It is found that getting the beam at a perfect horizontal level is difficult. It is because, in the last step of Get Ready algorithm, an input is given to the system such that there is just a little change in the ball position (usually to other side), which indicates the beam is horizontal.

But due to, limitations of hardware, a change in ball position is seen only when it starts moving to other side. Sometimes it moves faster, sometimes a little slower which makes perfect horizontal levelling difficult. Therefore, a reasonable horizontal level is considered as 'the level'.

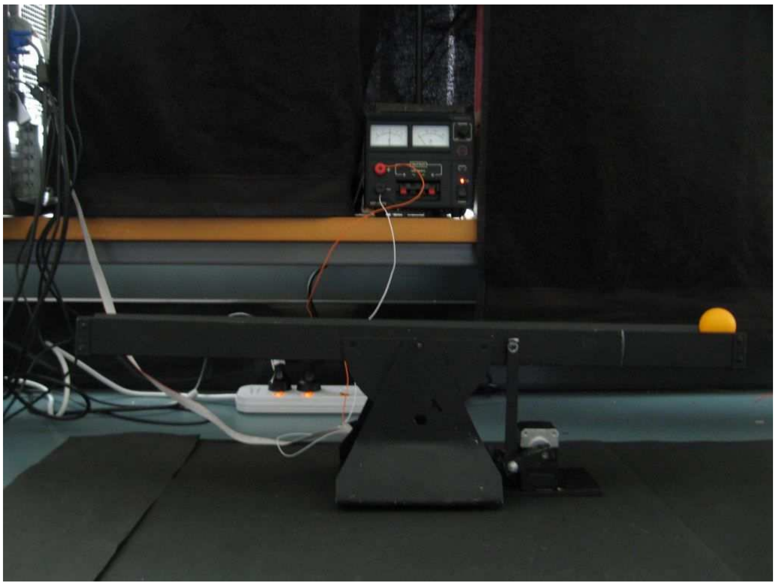


Figure 8.14: Ball and Beam Experiment: Beam leveled reasonably

Once the beam is levelled, Iterative Fast Model Control routine is executed by clicking on 'measure' button. Movement of the ball on the beam is shown by a graph in Fig. 8.15. The process is carried out as per the algorithm described in section 8.3.3. The model needs to represent the plant as closely as possible but for better accuracy model steplength is taken as 0.01 which is smaller than plant steplength of 0.033.

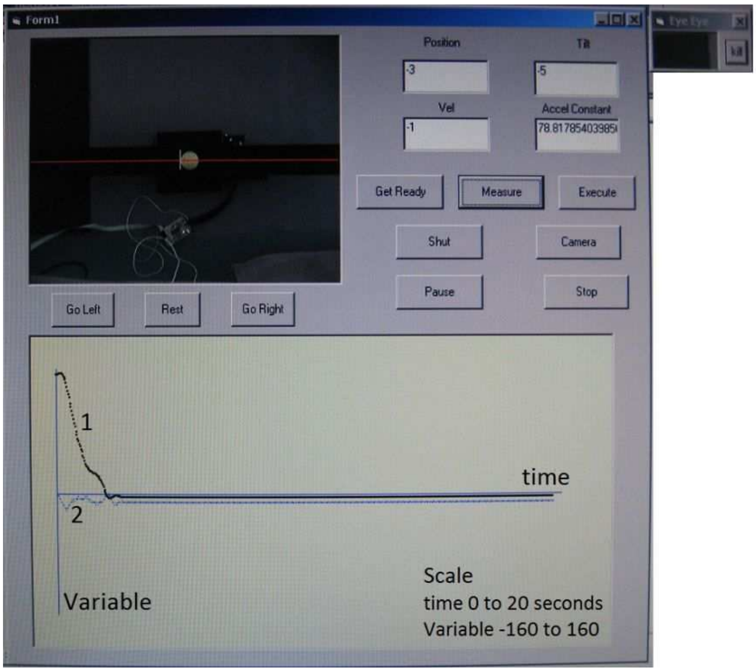


Figure 8.15: Ball and Beam Experiment: Ball balanced at the center of the beam Plot Curves:- 1: Path taken by the ball 2: 2*tilt

The final position of the ball on the actual beam is shown in Fig. 8.16.

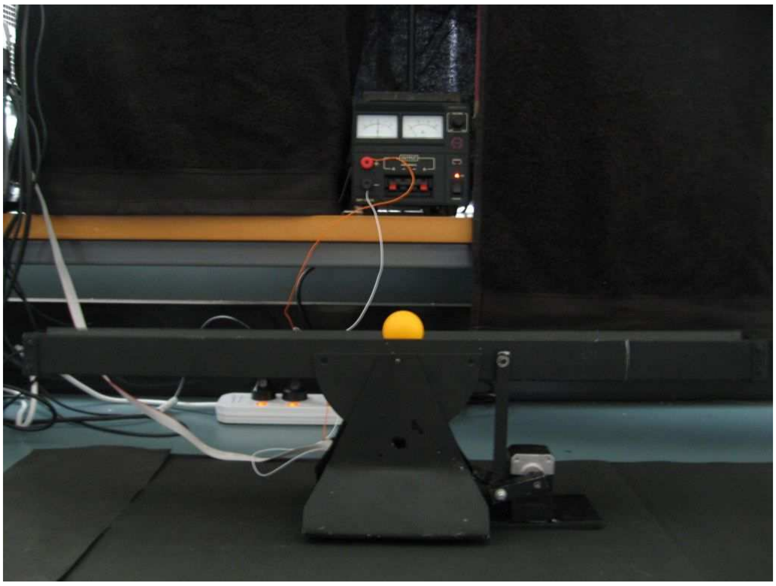


Figure 8.16: Ball and Beam Experiment: Ball balanced at the center of the beam

The disturbance rejection ability of the strategy is tested. A disturbance is added to the system when it is settled by pushing the ball to the left side of the beam and when it settled again, the ball is pushed to right.

On both occasions strategy recovered from the disturbance and settled. The results are shown in Fig. 8.17. After settling of system, first peak of curve-1 in the negative quadrant shows ball pushed to the left end and second peak in the positive quadrant shows ball pushed to some distance on right side of the beam.

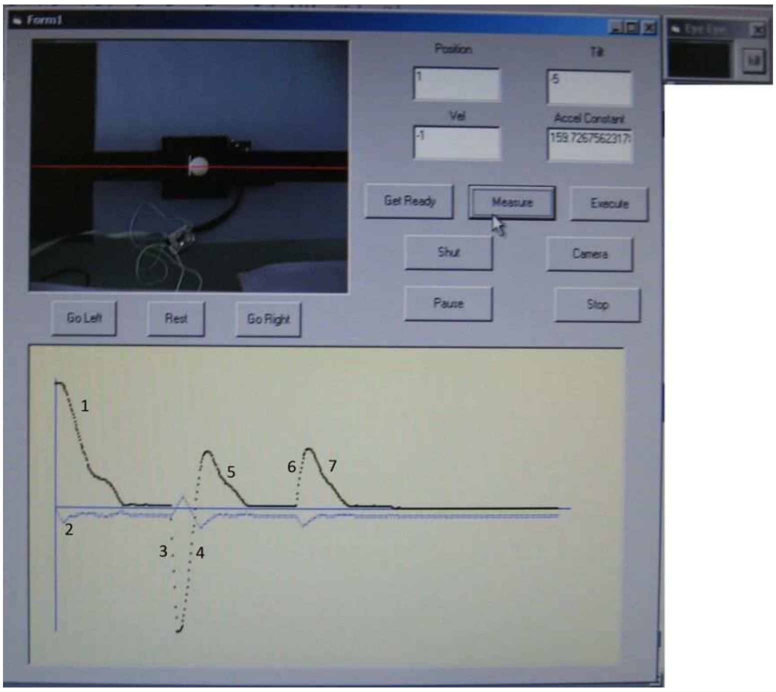


Figure 8.17: Ball and Beam Experiment: Recovery from External Disturbance. Curves:- 1- Path taken by the ball 2- tilt multiplied by 2, 3- Disturbance ball pushed to left, 4 and 5 - Recovery curves, 6- Disturbance ball pushed to right, 7- Recovery curve

8.3.5 Important Points

- It is important to represent the actual plant in the 'fast model' as closely as possible.
- At the beginning the beam should be tilted at some angle around 10°, to facilitate calculation of acceleration constant and tilt.
- The beam needs to be at a reasonable horizontal level before the implementation of the Iterative Fast Model Control strategy.
- A calibration constant needs to be added to the plant position when it is assigned to the model position, to compensate for the calibration error. This error is due to a difference between actual length of the beam and the perceived length of the beam by the camera.

- The model variable need to be of same data type as of corresponding plant variables.
- The value of acceleration constant should be at least 70. If it is less than 70 then it needs to be increased to a value greater than 85, for better performance.

8.4 Conclusions

Applicability of Iterative Fast Model Control to real life control problems is tested through two different examples. A fourth order example of an Aircraft Lateral Control is considered. Different constraints such as limited bank-angle, limited roll-rate are incorporated and it is found that Iterative Fast Model Control gives smooth maneuver. Final strategy of Aircraft Lateral Control with state constraints is outlined. The second example of a ball and beam experiment demonstrated that Iterative Fast Model Control works in practice and it can recover from external disturbances.

Chapter 9

Conclusions

New Iterative Fast Model Control Strategies are proposed and analysed. The proposed IFMC strategy for a third order system has been successfully implemented in a ball and beam experiment. Experimental results demonstrated that the IFMC strategies can work in practice.

All the objectives outlined initially (section 1.3) have been achieved. This chapter presents the conclusions of research. The conclusions are categorised in three sections, Section 9.1 New IFMC strategies, Section 9.2 Analysis of IFMC strategies and Section 9.3 Applications of IFMC. Section 9.4 outlines suggestions for future work.

9.1 New IFMC Strategies

9.1.1 IFMC for a third order system

A new Iterative Fast Model Control strategy is proposed to control cascaded integrator systems with input constraints. Such a constraint implies that the input cannot exceed its full drive limit in either sense.

The initial simulations are carried out with a third order cascaded integrator system. The performance of IFMC is compared with earlier Fast Model Control strategies proposed in 1968 and 1987.

A time optimality test has been carried out based on Pontryagin's principle. This revealed that the Iterative Fast Model Control Strategy gives near time optimal control performance. It is an improvement on its predecessors in the sense that it gives more cautious control and better accuracy in reaching origin for all states. Also, when extended to higher order systems Iterative Fast Model Control gives superior performance than its predecessors.

A very recent method has been proposed by Gayaka and Yao. Although it performs slightly better than all its predecessors, linear control theory based methods developed from the early 90's, a performance comparison shows that it is outperformed by all fast model control based strategies 1968, 1987 and Iterative Fast Model Control Strategy. They outperform the Gayaka and Yao strategy by 40 to 50%, in terms of settling time efficiency. Performance comparison results are shown in appendix A.

It can be concluded that the near time optimal performance of the new Iterative Fast Model Control strategy has been found to be superior.

9.1.2 IFMC for fourth and fifth order systems

Iterative Fast Model Control strategy is extended to fourth and fifth order cascaded integrator systems. It has been found that as the order of the system increases, the system response tends to have overshoots, or even an unstable response in the case of a sixth order system.

A technique of 'slugging' which is a deliberate mismatch of model and plant is used to improve the system response. It is shown that slugging removes overshoots and even make an unstable system stable. This also aids in improvement of settling time performance.

The appropriate 'slugging factor' can be determined in several ways. A pattern emerges in which first and last values can be unity. Intermediate values can be in some cases be given the same value between unity and 2, or a more complicated trial process can be used to give them individual values.

9.1.3 IFMC for higher (6^{th} to n^{th}) order systems

Iterative Fast Model Control strategies have been slightly modified to control very high order systems. Starting with 6^th order to 11^{th} order, simulations have shown that the system response is stable and has a finite settling time.

Two possible methods to control a sixth order cascaded integrator system have emerged. In one method, slugging is used to stabilize the system. It is discussed in section 4.4.3. In the second method a limit is put on the first three variables of the model. This is discussed in section 6.2. The second method, however, has been found to deliver an improved performance over the first method.

The method of constraining model variables has been extended successfully to control even higher order systems. Simulations have shown that systems up to 11^th order give stable converging performance. This supports the possibility of extension of strategy to a system of n^{th} order, where n is any positive integer.

9.1.4 IFMC with state constraints

Iterative Fast Model Control strategies that accommodate different state constraints are proposed in chapter 5. To deal with constraints imposed on different states, the state variables are classified as primary and subsequent variables. A 'primary' variable is a variable that receives the input directly, being the first integral of the cascade. All remaining variables are defined as subsequent variables.

Constraints on primary and subsequent variables can be implemented together or on an individual basis, depending on the requirement.

Inclusion of state constraints in the strategy slightly increases the settling time. A generalised method of Iterative Fast Model Control with state constraints for cascaded integrator systems of any order is proposed. An example of Aircraft lateral control is discussed in chapter 8 which demonstrates the usefulness of this control strategy in practice.

9.2 Analysis of IFMC Strategies

9.2.1 Settling time efficiency and settling point accuracy

IFMC is a terminating control where the system enters a limit cycle or dithers after reaching a neighbourhood of the settling point. The time taken by IFMC to reach this settling point is found to be near optimal.

A detailed analysis revealed that the settling time of the 1987 fast model control strategy is slightly shorter than IFMC. However; IFMC delivers more cautious control than the 1987 strategy with no overshoots in its response. Also, when applied to fourth and higher order systems, the performance of IFMC is found to be superior.

In the determination of settling point accuracy, emphasis is given to the proximity of the position to the settling point. An error is calculated which is a sum of weighted squares of the variables and a control terminating condition is used, that the error should be less than some small number.

'Some small number' determines the proximity of the position to the settling point, under the influence of plant and model step lengths. A termination error of 10^{-10} has been found to give the best positional accuracy when the model step length is 0.01 seconds.

The plant steplength can be varied to represent the decrease in the model prediction time as the settling point is approached. This results in the more rapid switching of the inputs, resulting in a better proximity of the system to the settling point.

9.2.2 Working of IFMC and Mathematical Analysis

In the Iterative Fast Model Control, a decision on the input is based on model times to 'come onside', t_+ and t_-. When the control is stable these model times will tend towards zero as time tends to infinity. The rapid plant input switching between +1 and -1 results in a sliding mode behaviour. The average value of the input during this sliding mode will appear to be a value between +1 and -1.

Calculus is applied to find the continuous time equivalent of this value which is then verified by the data obtained during simulations. Simple cases of the second and third order systems are used to outline the working of the Iterative Fast Model Control strategy. It is shown that the analysis becomes more complex as the order of the system increases.

9.2.3 Lyapunov stability analysis

The stability analysis of Iterative Fast Model Control is carried out using the direct method of Lyapunov. IFMC strategies are based on predicted model times t_+ and t_-. At every switching reversal these model times appear to overtake each other in terms of value. But, overall as the system approaches the origin, the value of these model times decrease with respect to plant time.

The plant input is selected as that of the greater of the model times. Without slugging, the simulation for this sense of input will be unchanged as plant time increases, meaning that the remaining time to come onside will decrease at an equal rate. It is therefore tempting to consider max(t+, t-) as a Lyapunov function to describe the motion. This approach can be justified in the cases of second and third order, showing that those systems are asymptotically stable.

Unfortunately it is not valid to apply this reasoning above to higher orders. The times t+ and t- are ruled by the final variable to come onside. As the system time advances, these can 'hop' in increments that are not continuous as inflections in the traces are passed.

9.3 Applications of IFMC

Applicability of Iterative Fast Model Control strategies in real life situations is explored by two examples.

9.3.1 Aircraft Lateral Control

A hypothetical example of Aircraft Lateral Control is considered where IFMC algorithm is used for automatic alignment of the aircraft with the centre of the runway. Specification of Boeing 747-400 and guidelines of FAA, USA are used to add 'reality'. Several state constraints are also considered to ensure passenger comfort during manoeuvre.

It shown that, the Iterative Fast Model Control strategy with state constraints can align the aircraft with the centre of the runway very smoothly, overcoming the situations of wind-gust and turbulence.

9.3.2 Ball and Beam Experiment

The usefulness of Iterative Fast Model Control Strategy in practice is demonstrated through successful implementation of the strategy in a Ball and Beam experiment. The goal of the experiment was to balance a table tennis ball at the centre of a beam.

The experimental results showed that the ball came to the centre of the beam with less than 2% error. A disturbance was added by pushing the ball to the left first and then again to the right. The system response showed excellent recovery from the disturbances which demonstrated the robustness of the Iterative Fast Model Control Strategy.

9.4 Suggestions for Future Work

1. It has been shown that IFMC strategies work with cascaded integrator systems. If a non-cascaded system has a cascaded-integrator component then the effects of IFMC of cascaded component on overall system can be tested.

2. Applications of Iterative Fast Model Control can further be explored by implementing IFMC strategies in different real time projects in the fields of robotics, space systems, industrial systems and such.

3. In this research, IFMC strategy is implemented in a ball and beam experiment using a desktop computer. In real applications, designers may be required to use embedded controllers. Therefore, it will be interesting to investigate the implementation of IFMC using standalone microcontrollers.

4. Despite modern computer speeds, the computing time for any but a slow 11th order system is excessive. It would be of interest to find ways of distributing the computation in a multi-core system, or in the parallel structure of a Graphics card.

5. In the current research, the case of SISO system has been considered. IFMC perhaps also can be applied to MIMO systems.

References

Abramovitch, D. (1994), Some crisp thoughts on fuzzy logic, *in* 'American Control Conference, 1994', Vol. 01, Baltimore, USA, pp. 168–172.

Adey, A. J., Billingsley, J. & Coales, J. F. (1966), Predictive control of higher order systems, *in* 'Proceedings of the 3rd IFAC World Congress on Automatic Control', Vol. 01, London, UK, pp. 270–272.

Annaswamy, A. M. & Karason, S. (1995), 'Discrete-time adaptive control in the presence of input constraints', *Automatica* **vol. 31**(no. 10), pp. 1421–1431.

Baili, S. U., Zehquiang, C. & Zhuzhi, Y. (2006), 'Multivariable decoupling predictive control with input constraints and its application on chemical process', *Chinese Journal of Chemical Engineering* **vol. 14**(no. 02), pp. 216–22.

Bernstein, D. S. & Michel, A. N. (1995), 'A chronological bibliography on saturating actuators', *International Journal of Robust Nonlinear Control* **vol. 5**(no. 5), pp. 375–380.

Billingsley, J. (1968), Predictive Control of Third and Higher Order Systems, PhD thesis, Department of Engineering, Cambridge University,UK.

Billingsley, J. (1987), Predictive control fast model, *in* M. G. Singh, ed., 'Systems and Control Encyclopedia: Theory, Technology, Applications', 1st edn, Vol. 06, Pregmon Press, Oxford, UK, pp. 3866–3877.

Bláha, L., Schlegel, M. & Mošna, J. (2009), Optimal control of chain of integrators with constraints, *in* 'Proceedings of the 17th International Conference on Process Control' 2009', Slovak University of Technology in Bratislava, Štrbské Pleso SK, pp. 51–56.

Boeing (2001), *Boeing 747 -441 Operations Manual*, d6-30151-425 edn, The Boeing Company, Seattle, Washington.

Branch, F. P. S. (2007), *Instrument Procedures Handbook*, Federal Aviation Administration, U.S. Department of Transportation.

Byrnes, C. I. & Isidori, A. (1989), 'New results and examples in nonlinear feedback stabilization', *Systems and Control Letters* **vol. 12**(no. 5), pp. 437–442.

Byun, D. G. & Kwon, W. H. (1988), Predictive control: A review and some new stability results, *in* T. J. McAvoy, Y. Arkun & E. Zafiriou, eds, 'Proceedings of the 1988 IFAC Workshop on Model Based Process Control', Pergamon Press, Oxford, UK, pp. 81–87.

Chen, G. & Kairys, G. (1993), When and when not to use fuzzy logic in industrial control, *in* 'Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE', Vol. 03, Toronto, Canada, pp. 2035–2040.

Chen, W. J., Lei, F. & Lei, K. K. (1997), Fuzzy logic controller for an inverted pendulum system, *in* 'Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conference on', Vol. 01, Beijing, China, pp. 185–189.

Chestnut, H. & Wetmore, V. (1959), Predictive control applied to a simple position control, *in* 'Engineering Lab. Report,59GL104', Vol. 128, General Electric Company, Schenectady, New York, pp. 227–232.

Clarke, D. W., Mohtadi, C. & Tuffs, P. S. (1987*a*), 'Generalised predictive control -i the basic algorithm', *Automatica* **vol. 23**(no. 02), pp. 137–148.

Clarke, D. W., Mohtadi, C. & Tuffs, P. S. (1987*b*), 'Generalised predictive control -ii extensions and interpretations', *Automatica* **vol. 23**(no. 02), pp. 149–160.

Coales, J. F. & Noton, A. R. M. (1956), 'An on-off servomechanism with predicted change-over', *Proceedings of the IEE - Part B: Radio and Electronic Engineering* **vol. 103**(no. 10), pp. 449–460.

DeRusso, P. M., Roy, R. J., Close, C. M. & Desrochers, A. A. (1998), *State Variables for Engineers*, 2nd edn, John Wiley and Sons Inc, New York, USA.

Dion, J. M., Dugard, L. & Tri, N. M. (1987), Multivariable - adaptive control with input output constraints, *in* 'Decision and Control, 1987. 26th IEEE Conference on', Vol. 26, Los Angeles, CA, USA, pp. 1233–1238.

Dodds, S. J. (1981), 'Bang-bang control law for single-input time-invariant plant', *Control Theory and Applications, IEE Proceedings D* **vol. 128**(no. 5), pp. 227–232.

Feng, G., Zhang, C. & Palaniswamy, M. (1991), Adaptive pole placement control subject to input amplitude constraints, *in* 'Decision and Control, 1991., Proceedings of the 30th IEEE Conference on', Vol. 03, Brighton, UK, pp. 2493–2498.

Fuller, A. T. (1973), 'Proof of the time optimality of a predictive control strategy for systems of higher order', *International Journal of Control* **vol. 18**(no. 06), pp. 1121–1127.

Garcia, C. E., Prett, D. M. & Morari, M. (1989), 'Model predictive control theory and practice -a survey', *Automatica* **vol. 25**(no. 03), pp. 335–348.

Gayaka, S. & Yao, B. (2011), Global stabilization of a chain of integrators with input saturation and disturbances, *in* 'American Control Conference (ACC), 2011', San Francisco, California, USA, pp. 3784 –3789.

Goodwin, G. C., Seron, M. M. & Dona, J. A. D. (2005), *Constrained Control and Estimation*, 1st edn, Springer-Verlag, London.

Grognard, F., Sepulchre, R. & Bastin, G. (2002), 'Improving the performance of low-gain designs for bounded control of linear systems', *Automatica* **vol. 38**(no. 10), pp. 1777–1782.

Gulko, F. B. & Kogan, B. Y. (1963), A method of optimal control prediction, *in* 'Proceedings of 2nd World IFAC Congress', Basel, Switzerland, pp. 63–68.

Hahn, W. (1963), *Theory and Application of Lyapunov's Direct Method*, Prentice Hall Inc., Englewood Cliffs, N. J., USA.

Jankovic, M., Sepulchre, R. & Kokotovic, P. V. (1996), 'Constructive lyapunov stabilization of nonlinear cascade systems', *Automatic Control, IEEE Transactions on* **vol. 41**(no. 12), pp. 1723–1735.

Johnson, E. N. & Kannan, S. K. (2003), Nested saturation with guaranteed real poles, *in* 'American Control Conference, 2003. Proceedings of the 2003', Vol. 1, Denver, Colorado, USA, pp. 497–502.

Kaliora, G. & Astolfi, A. (2004), 'Nonlinear control of feedforward systems with bounded signals', *Automatic Control, IEEE Transactions on* **vol. 49**(no. 11), pp. 1975–1990.

Karason, S. & Annaswamy, A. M. (1994), 'Adaptive control in the presence of input constraints', *IEEE Transactions on Automatic Control* **vol. 39**(no. 11), pp. 2325–2330.

Kaufman, H. & DeRusso, P. (1966*a*), 'Application of lyapunov's second method to the analysis of general second order bang-bang control systems', *Automatic Control, IEEE Transactions on* **vol. 11**(no. 1), pp. 124–126.

Kaufman, H. & DeRusso, P. (1966*b*), 'Stability analysis of predictive control systems', *Automatic Control, IEEE Transactions on* **vol. 11**(no. 3), pp. 455–464.

Khalil, H. (2002), *Nonlinear Systems*, 3rd edn, Prentice Hall Inc, Upper Saddle River, NJ, USA.

LaSalle, J. & Soloman, S. (1961), *Stability by Lyapunov's Direct Method*, Academic Press, New York, USA.

Lee, C. C. (1990), 'Fuzzy logic in control systems:fuzzy logic controller part 1 and 2', *IEEE Transactions on System Man and Cybernetics* **vol. 20**(no. 2), pp. 404–435.

Li, W. C. & Biegler, L. T. (1988), 'Process control strategies for constrained nonlinear systems', *Industrial and Engineering Chemistry Research* **vol. 27**(no. 08), pp. 1421–1433.

Lin, Y. C. (1994), The application of fuzzy logic control to speed control of a dc servo motor system, *in* 'American Control Conference,1994', Vol. 01, Baltimore, USA, pp. 590–594.

Lin, Z. (1995), Global stabilization and restricted tracking for linear systems subject to input and measurement saturation-a chain of integrators case, *in* 'American Control Conference, 1995. Proceedings of the', Vol. 4, Seattle, Washington, USA, pp. 2488–2492.

Lin, Z. (1998), 'Global control of linear systems with saturating actuators', *Automatica* **vol. 34**(no. 7), pp. 897–905.

Lin, Z. & Saberi, A. (1993), 'Semi-global exponential stabilization of linear systems subject to "input saturation" via linear feedbacks', *Systems and Control Letters* **vol. 21**(no. 03), pp. 225–239.

Lublin, L. & Athans, M. (1996), *The Control Handbook*, CRC Press Inc, Boca Raton, Florida, USA, chapter Linear Quadratic Regulator Control, p. 1548.

Maciejowski, J. M. (2002), *Predictive Control with Constraints*, Pearson Education Limited, Harlow, Essex, UK.

Marchand, N. (2003), Further results on global stabilization for multiple integrators with bounded controls, *in* 'Decision and Control, 2003. Proceedings. 42nd IEEE Conference on', Vol. 5, San Fransico, California, USA, pp. 4440–4444.

Marchand, N. (2005), 'Global stabilization of multiple integrators with bounded controls', *Automatica* **vol. 41**(no. 12), pp. 2147–2152.

Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. M. (2000), 'Constrained model predictive control: Stability and optimality', *Automatica* **36**(6), 789 – 814.

Mazenc, F. & Praly, L. (1994), Adding an integration and global asymptotic stabilization of feedforward systems, *in* 'Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on', Vol. 1, Buena Vista, Florida, USA, pp. 121–126.

Megretski, A. (1996), L2 bibo output feedback stabilization with saturated control, *in* 'Proceedings of the 13th IFAC World Congress', Vol. D, San Francisco, USA, pp. 435–440.

Mo, S. F. & Billingsley, J. (1990), 'Fast model predictive control of multivariable systems', *IEE Proceedings D, Control Theory and Applications* **vol. 137**(no. 6), 364–366.

Monopoli, R. V. (1975), Adaptive control for systems with hard saturation, *in* 'Decision and Control including the 14th Symposium on Adaptive Processes, 1975 IEEE Conference on', Vol. 14, Houstan, Texas, USA, pp. 841–843.

Morari, M. & Lee, J. H. (1999), 'Model predictive control past present and future', *Journal of Computers and Chemical Engineering* **vol. 23**(no. 4), pp. 667–682.

of Airport Safety, O. & Standards (2011), *Advisory Circular*, Federal Aviation Administration, U.S. Department of Transportation.

Ohkawa, F. & Yonezawa, Y. (1982), 'A discrete model reference adaptive control system for a plant with input amplitude constraints', *International Journal of Control* **vol. 36**(no. 05), pp. 747–753.

Payne, A. N. (1986), 'Adaptive one-step ahead control subject to an input amplitude constraint', *International Journal of Control* **vol. 43**(no. 04), pp. 1257–1269.

Pike, A. W., Grimble, M. J., Johnston, M. A., Ordys, A. W. & Shakoor, S. (1996), *The Control Handbook*, CRC Press Inc, Boca Raton, Florida, USA, chapter Predictive Control, p. 1548.

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V. & Mishchenko, E. F. (1963), *The Mathematical Theory of Optimal Processes*, John Wiley and Sons, New York, USA.

Ramaswamy, P., Edwards, R. M. & Lee, K. Y. (1993), Fuzzy logic control for nuclear power plant, *in* 'Neural Networks to Power Systems, 1993. ANNPS '93., Proceedings of the Second International Forum on Applications of', New York, USA, pp. 29–34.

Schmitendorf, W. E. & Barmish, B. R. (1980), 'Null controllability of linear systems with constrained controls', *SIAM Journal on Control and Optimization* **vol. 18**(no. 4), pp. 327–345.

Spong, M. W., Hutchinson, S. & Vidyasagar, M. (2006), *Robot Modeling and Control*, John Wiley & Sons, Inc, Hoboken, NJ, USA.

Sussmann, H. J., Sontag, E. D. & Yang, Y. (1994), 'A general result on the stabilization of linear systems using bounded controls', *Automatic Control, IEEE Transactions on* **vol. 39**(no. 12), pp. 2411–2425.

Sussmann, H. J. & Yang, Y. (1991), On the stabilizability of multiple integrators by means of bounded feedback controls, *in* 'Decision and Control, 1991., Proceedings of the 30th IEEE Conference on', Vol. 01, Brighton, UK, pp. 70–72.

Teel, A. R. (1992), 'Global stabilization and restricted tracking for multiple integrators with bounded controls', *Systems and Control Letters* **vol. 18**(no. 3), pp. 165–171.

Tsang, T. T. C. & Carke, D. W. (1988), 'Generalised predictive control with input constraints', *Control Theory and Applications, IEE Proceedings D* **vol. 135**(no. 06), pp. 451–460.

Tsinias, J. (1989), 'Sufficient lyapunov-like conditions for stabilization', *Mathematics of Control, Signals, and Systems (MCSS)* **vol. 2**(no. 4), pp. 343–357.

Vachtsevanos, G., Farinwata, S. S. & Pirovolou, D. K. (1993), 'Fuzzy logic control of an automotive engine', *IEEE Control Systems Magazine* **vol. 13**(no. 03), pp. 62–68.

Valle, F., Tadeo, F. & Alvarez, T. (2002), Predictive control of robotic manipulators, *in* 'Control Applications, 2002. Proceedings of the 2002 International Conference on', Vol. 01, Glasgow, Scotland, United Kingdom, pp. 203–208.

Wang, H. & Sun, J. (1992), Modified model reference adaptive control with saturated inputs, *in* 'Decision and Control, 1992., Proceedings of the 31st IEEE Conference on', Vol. 04, Tucson, Az, USA, pp. 3255–3256.

Willems, J. L. (1970), *Stability Theory of Dynamical Systems*, Thomas Nelson and Sons Ltd., UK.

Zanasi, R. & Morselli, R. (2003), 'Discrete minimum time tracking problem for a chain of three integrators with bounded input', *Automatica* **vol. 39**(no. 9), pp. 1643–1649.

Zhang, C. & Evans, R. J. (1987), 'Amplitude constrained adaptive control', *International Journal of Control* **vol. 46**(no. 01), pp. 54–64.

Zhou, B. & Duan, G. R. (2007), 'Global stabilisation of multiple integrators via saturated controls', *Control Theory Applications, IET* **vol. 1**(no. 6), pp. 1586–1593.

Zhou, B. & Duan, G. R. (2008), 'A novel nested non-linear feedback law for global stabilisation of linear systems with bounded controls', *International Journal of Control* **vol. 81**(no. 9), pp. 1352–1363.

Zhou, B. & Duan, G. R. (2009), 'Global stabilization of linear systems via bounded controls', *Systems and Control Letters* **vol. 58**(no. 1), pp. 54–61.

Ziebolz, H. & Paynter, H. M. (1953), Possibilities of a two-time computing system for control and simulation of dynamic systems, *in* 'Proceedings of the National Electronics Conference', Vol. 9, Chicago, USA, pp. 215–223.

Appendix A

Performance comparisons with recent strategies

The performances of fast model control strategies have shown an improvement of 41-45% over a recent control method, that was proposed by Gayaka and Yao (2011), to control a third order cascaded integrator systems with input constraints.

Gayaka and Yao (Gayaka & Yao 2011) proposed a control method that gives superior performance in terms of settling time than the other linear control based strategies in particular strategies proposed by Zhou and Duan in (2007), (2008) and (2009).

These strategies of Zhou and Duan had already shown improvement over other linear control strategies developed until 2007. This showed that the strategy proposed by Gayaka and Yao was the most efficient in terms of settling time.

Therefore performances of fast model control strategies developed in 1968, 1987 and new Iterative Fast Model Control strategy are compared with the results of Gayaka and Yao strategy. The system is a third order cascaded system with input constrained at full positive and full negative drive. The initial conditions used are exactly same as that of Gayaka and Yao, [-3,-3,3]. From the Fig. A.4, it can be seen that the settling time of the strategy of Gayaka and Yao is approximately 11 seconds.

Hence, to clearly show an improvement in the settling time with fast model control strategies, the time axis for the simulation plot is drawn from 0 to 10 seconds, for all 3 fast model control strategies.

The performance of 1968 Fast Model Control Strategy is shown in the Fig. A.1.

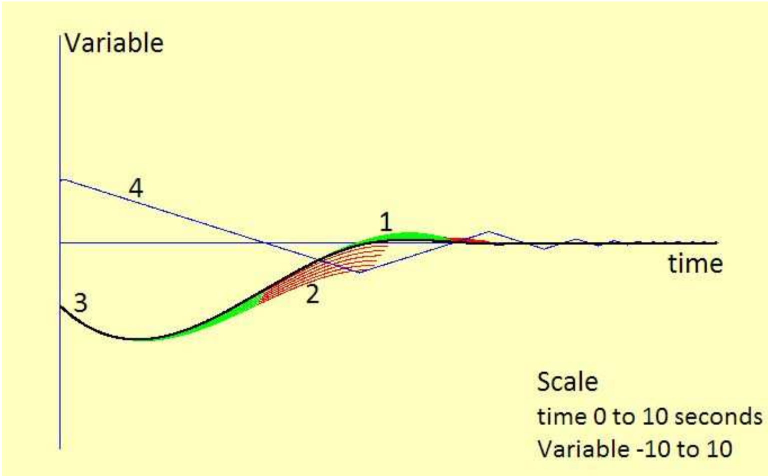


Figure A.1: Response of 1968 fast model control strategy with same initial conditions as (Gayaka & Yao 2011) Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: Acceleration a

The simulation results show that the settling time of 1968 fast model control strategy is approximately 6.5 seconds which is an improvement of nearly 41%. The performance of 1987 Fast Model Control Strategy is shown in the Fig. A.2.

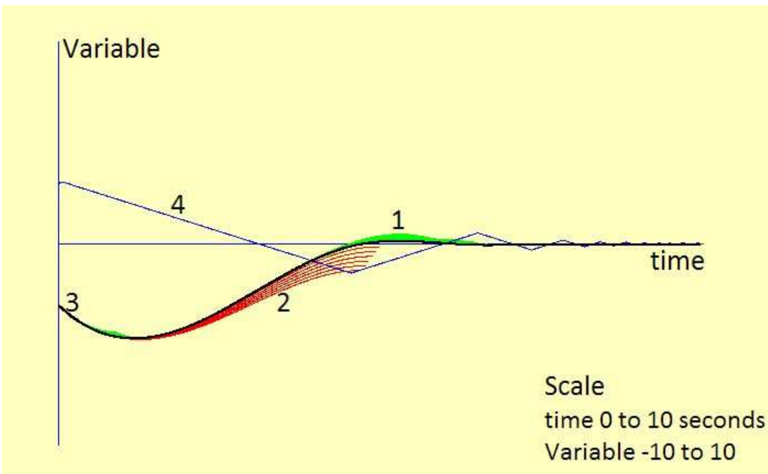


Figure A.2: Response of 1987 fast model control strategy with same initial conditions as (Gayaka & Yao 2011)Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: Acceleration a

The simulation results show that the settling time of 1987 fast model control strategy is nearly same as that of 1968 strategy. The performance of new Iterative Fast Model Control Strategy is shown in the Fig. A.3.

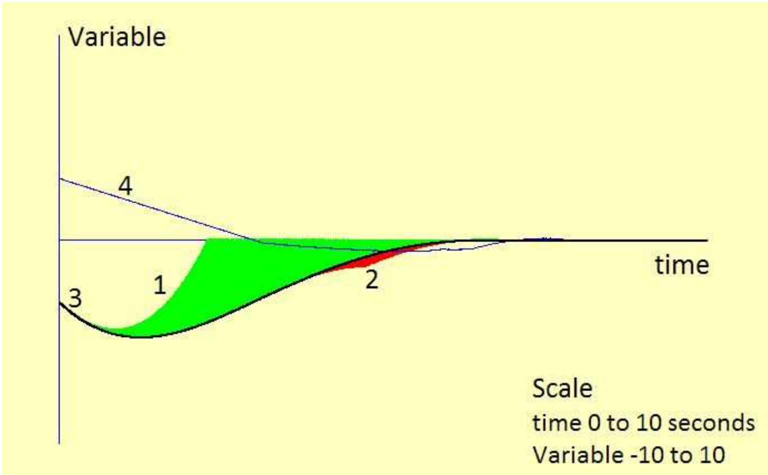


Figure A.3: Response of new Iterative fast model control strategy with same initial conditions as (Gayaka & Yao 2011)Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: Acceleration a

The simulation results show that the settling time of new Iterative Fast Model Control strategy is approximately 6.1 seconds which is an improvement of nearly 45%.

In the following figure, results from (Gayaka & Yao 2011) are reproduced for reference.

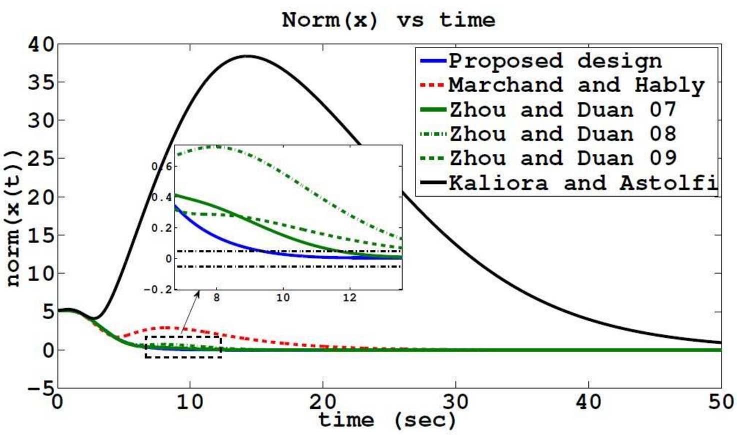


Figure A.4: Performance of Gayaka and Yao strategy reproduced from (Gayaka & Yao 2011, pp. 3789)

From Fig. A.4, it can be observed that Gayaka and Yao normalized the initial conditions, however the settling time appears to be around 11 seconds.

This clearly shows that with fast model control strategies, the settling time performance of a third order cascaded integrator system with input constraints improves significantly. The response of new Iterative Fast Model Control strategy has been found to be superior than all other strategies.

Appendix B

1981 & 1968 Fast Model Control Strategies Results

Billingsley (1968) proposed a strategy to control a third order cascaded integrator system with input constraints. This strategy used fast models of the plant and looked at the furthest time when final overshoot occurs before making a decision on plant input.

Dodds (1981) proposed a fast model control strategy for the same system with same input constraints. This strategy used fast models of the plant but looked at the number of sign changes occurring in the 'position' curve for each sense of drive until no further sign changes in the model could occur. Dodds speculated that results of this strategy would be similar to Billingsley's 1968 strategy.

The performances of both strategies are checked for a third order cascaded integrator system with constrained input at both extreme ends with initial conditions as position x=10, velocity v=0 and acceleration a=0.

The simulation results of the Billingsley's 1968 strategy are shown in Fig. B.1

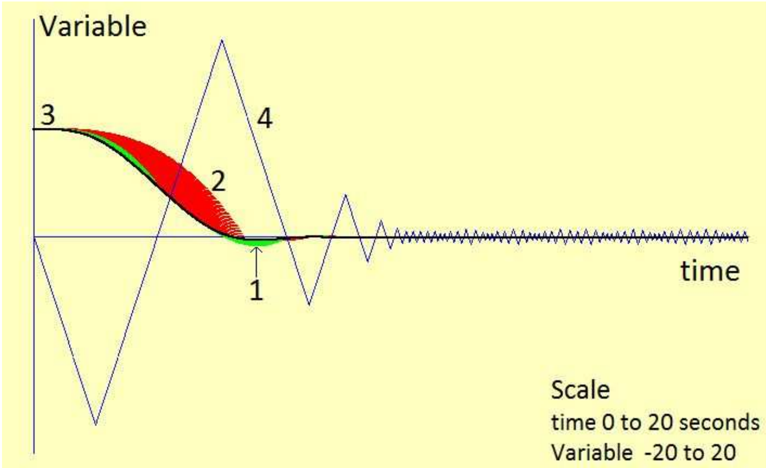


Figure B.1: 1968 Fast Model Predictive Control Strategy Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

The simulation results of the Dodds's 1981 strategy are shown in Fig. B.2

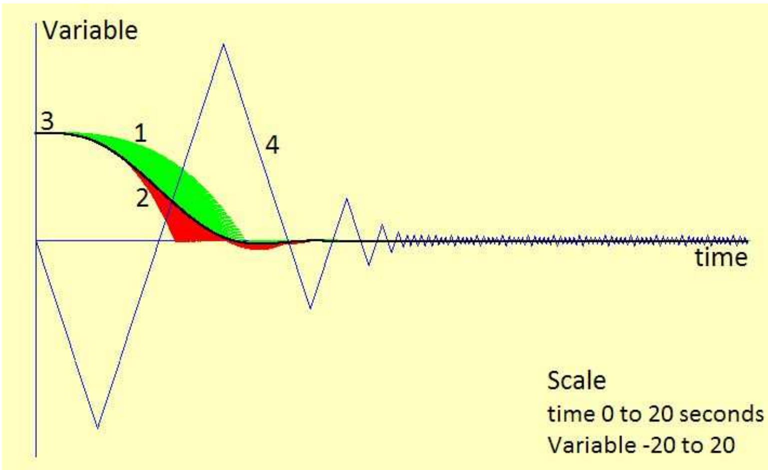


Figure B.2: Dodds's Fast Model Control Strategy Curves:- 1: Predictions with full positive drive, 2: Predictions with full negative drive, 3: Position x, 4: 10*Acceleration (a)

It can be seen from Fig. B.1 and Fig. B.2 that performances of both strategies are almost the same.