Proton Beam Steering Control System for High Precision Radiotherapy at iThemba LABS: An Investigation on Actuator Saturation Constraints

M. S. Tśoeu.

Thesis presented in fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Cape Town.

Supervisor:

Prof. J. Tapson
Dept. of Electrical and Electronics Engineering
University of Cape Town

Co-Supervisors:

Mr. L. Serutla
Electronics and Information Technology Group
iThemba LABS (Laboratory for Accelerator Based Sciences)

Mr. J. Nieto-Camero
Medical Radiation Group
iThemba LABS (Laboratory for Accelerator Based Sciences)

May 29, 2008

DECLARATION

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

Signature:.....

Date:.....

ABSTRACT

Control systems are an integral part of a wide variety of larger systems, ranging from research, medical, production, mineral extraction, to financial and economic systems. While the purpose of the control systems in these diverse applications is different some of the problems arising in them are common. Practical systems are all subject to constraints of some sort. These can be in the form of physical limitations, the system configuration and the internal states of the system in question. The constraints vary in their complexity and hence they are sometimes referred as *hard* or *soft* constraints. A lot of research effort has been devoted towards dealing with constraints in control systems in order to obtain optimal results. An observation shows that there are two main trends in the methods employed in this effort. These are, **Anti-windup** schemes and the widely deployed **Model Predictive Control (MPC)** and its descendant methods. This thesis aims at studying some of the techniques used to deal with constraints with special application to the Proton beam steering control at iThemba LABS. The steering of charged particles occurring in research plants is one of the interests of control systems. In this work an investigation of the algorithm for the control of the proton beam steering system in the radiotherapy treatment facility at iThemba LABS is conducted. This algorithm is intended to autonomously maintain the beam centered with reference to the axis of the beamline, and keep the beam front parallel to the central axis of the beamline as stated by van Tubbergh and De Kock, 2006. Furthermore, the algorithm is responsible for monitoring the distribution of the proton beam, in a plane normal to the beam travel direction. Moreover, scaleability, reconfigurability and fault tolerance are to be ensured due to the nature of the steering system. The result of the work is a robust and fail-safe algorithm, that autonomously keeps the proton beam position centered and continuously monitors the distribution of the beam in real-time. This algorithm is implemented on a LINUX Operating System running on an ETX computer.

ACKNOWLEDGEMENTS

Having reached the end of the work, gratitude is in order for my academic supervisor Prof. J. Tapson, and my industrial supervisors, Mr. L. Serutla and Mr. J. Nieto-Camero. Furthermore, I thank the management of iThemba LABS for giving me the opportunity to do research work at the facility, and my colleagues and friends who helped me both professionally and morally. Special thanks to Mphakiseng Masuabi, *I'll always treasure all the support you gave me.* Last but, not least I thank my mother, my sisters and brothers, and my entire family, for all the support they gave me throughout, *I wouldn't have done it without you guys* 'KEA LEBOHA 'ME LE LONA BANA BESO'.

Contents

List of Figures

List of Abbreviations and Notation

- BSCU - Beam Steering Control Unit
- DC - (Direct Current) at ZERO frequency
- DMC - Dynamic Matrix Control
- EG - Error Governor
- FC - Faraday Cup
- IMC - Internal Model Control
- IMP - Internal Model Principle
- LQR - Linear Quadratic Regulator
- MIMO - Multi-Input Multi-Output
- MPC - Model Predictive Control
- MWIC - Multiwire Ionization Chamber
- PBSCS - Proton Beam Steering Control System
- PI - Proportional + Integral
- PSU - Power Supplier
- RG - Reference Governor
- RHOC - Receding Horizon Optimal Control

- SABUS - South African Bus Standard
- SISO - Single Input Single Output
- SS - Supervisory System
- TSB - Therapy Safety Bus

Chapter 1

INTRODUCTION

The treatment of cancer using radiotherapy techniques is a critical procedure. This is due to the fact that excessive patient exposures to radiation can be detrimental. Furthermore, uncontrolled radiation spills are also dangerous to people working around the area of the radiation spill. It is therefore important that every section of a radiation beamline be well designed to minimize the risk, and optimize the results of the radiotherapy treatment.

This treatment is the only hope for some of the cancer patients. A fraction of these cancer patients can only be cured using proton radiotherapy. Advancements in techniques, technologies, accuracy and safety in the treatment environment improve the treatment success, and the patient confidence in radiotherapy treatment.

One of the key factors in making radiotherapy successful and minizing the risk associated with it, is through the deployment of effective, accurate and fail-safe control systems to control and monitor the large equipment used in the radiotherapy treatment. It is the aim of this thesis to contribute in the efforts dedicated towards improving the proton beam steering section of the control systems, by investigating methods of dealing with actuator saturation constraints. At iThemba LABS, the proton beam steering system is aimed at keeping the proton beam centered relative to the cross-sectional axes of the

beamline (X,Y) and parallel to the axis along the beamline (Z-axis). This is illustrated schematically in Figure 1.1 with a slight abuse of axes labeling order.

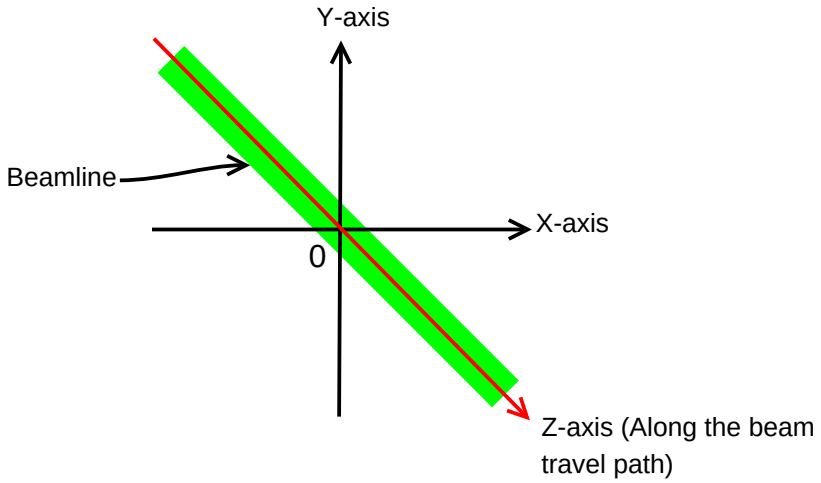


Figure 1.1: Illustration of beamline axes, showing the cross-sectional axes of the beamline used as reference for beam centering, and the axis along the beam line to which the beam should be parallel

A number of systems have been implemented to realize this task, such as [30], [16], and others. *The aim of this thesis is to improve beam steering by investigating the methods of dealing with constraints in the proton beam steering control at iThemba LABS.*

Several methods have been used to deal with constraints in control systems. Some of these can be grouped as **anti-windup** methods that deal with input constraints to avoid integrator windup due to saturation. Other popularly used methods are the **Model Predictive Control (MPC)** schemes, which rely on the prediction of future evolution of the system outputs, and prediction of the disturbance and past system inputs, to come up with optimal inputs to the system. Some schemes deriving from the latter method are the L_2 Anti-windup techniques [6], the Reference Governor (RG) [5] and Error Governor (EG) [26] anti-windup methods (which decouple the stabilization and tuning efforts of the closed loop from the part that deals with the constraints). Some of these methods will be simulated and tested on the steering control system to see which one of them yields better results.

Including **Chapter 1: Introduction**, this thesis is divided into ten chapters. Chapter

2 gives a brief overview on **iThemba LABS**, the aims of the facility, and the current status of beam steering control technology at iThemba LABS. Thereafter, Chapter 3 is on **Particle Beam Steering Control Applications and Methods** and it takes a look at prior work that has been done in dealing with control constraints, and reviews literature on technologies and practices involved in the beam steering control. **Chapter 4: Proton Beam Steering Control System Analysis** focuses on the structure of the proton beam steering control problem. Analysis and decomposition of the system structure is performed to simplify the control into smaller control tasks. Chapter 5 is the **Proton Beam Steering Control System Synthesis and Design**, and it focuses on the design of the controllers that are used for simulation and experimentation for the beam steering task. **Chapter 6: Proton Beam Steering System Windup Simulation** follows, and it focuses on the simulation, and experimentation with the designed controllers and anti-windup techniques.

The next chapter is **Chapter 7: Control Implementation**. This chapter reviews methods and literature on Real-Time computer implementation of control systems. Experimental testing of the steering control system is not done in this work, therefore Chapter 8, which is on **Absence of Experimental Results** is included to explain why experimental testing is not feasible. Chapter 9 is on the **Results and Discussion** and it deals with the results obtained from the preceding chapters and correlates them to related theory and discusses findings. The final chapter is the **Conclusion** (Chapter 10), which summarizes the findings and results of the work done in this thesis and draws conclusions on the investigated problem.

Chapter 2

iTHEMBA LABS and PROTON THERAPY

2.1 Location and Purpose

iThemba LABS (Laboratories for Accelerator Based Sciences) is an institution for accelerator based sciences. It is aimed at providing research into particle based sciences, for use in particle radiotherapy treatment of cancer, research in the areas of particle beams, and production of radio-isotopes for use in radio-medicine and research. This institution is administered by the National Research Foundation (NRF) [57]. It has two sites, one within the Gauteng Province and another in the Western Cape Province. Aerial photographs of the sites are shown in Figure 2.1 below.

(a) Gauteng site

(b) Western Cape site

Figure 2.1: iThemba LABS sites

At the iThemba LABS. site in the Western Cape, a new project is ongoing. This project is aimed at expanding the facility and upgrading the equipment and technologies used in the radiotherapy treatment using protons. It is within this project that several subprojects have been defined. One of these projects is the Proton Beam Steering Control system which is the subject of the work done in this thesis.

2.2 Treatment Facility

The diagram in Figure 2.2, shows the plan of the treatment facility at iThemba LABS. The sections of the plan at which the steering control system is deployed, are the proton therapy treatment vaults marked 5 and 7.

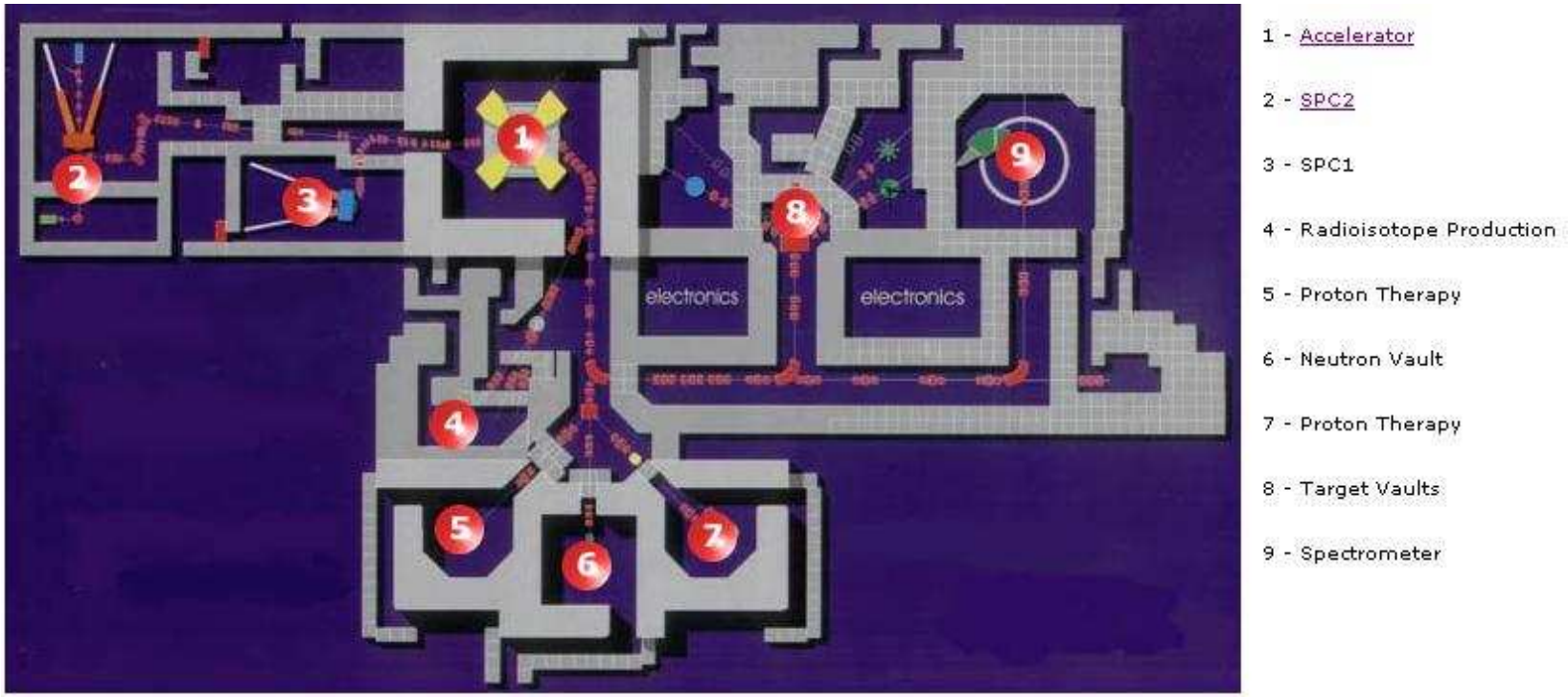


Figure 2.2: The treatment facility plan at iThemba LABS. This shows the Proton therapy treatment vaults where the beam steering system is situated (labeled 5 and 7)

2.3 The Second Beamline

Shown in Figure 2.3 is an informative diagram of the proton beamline section of the treatment facility. In the diagram the encircled portion indicates the Beam Steering Control System. This diagram is aimed at showing how the entire system's components and subsystems are interconnected and does not show the accurate physical positions of the components relative to each other.

2.4 The current control system

In the already existing facility at iThemba LABS, a distributed computer control system running on OS/2 operating system is implemented [30]. This control system has a number of subsystems, among which is the beam steering control system. This steering control system has been in operation since 1990.

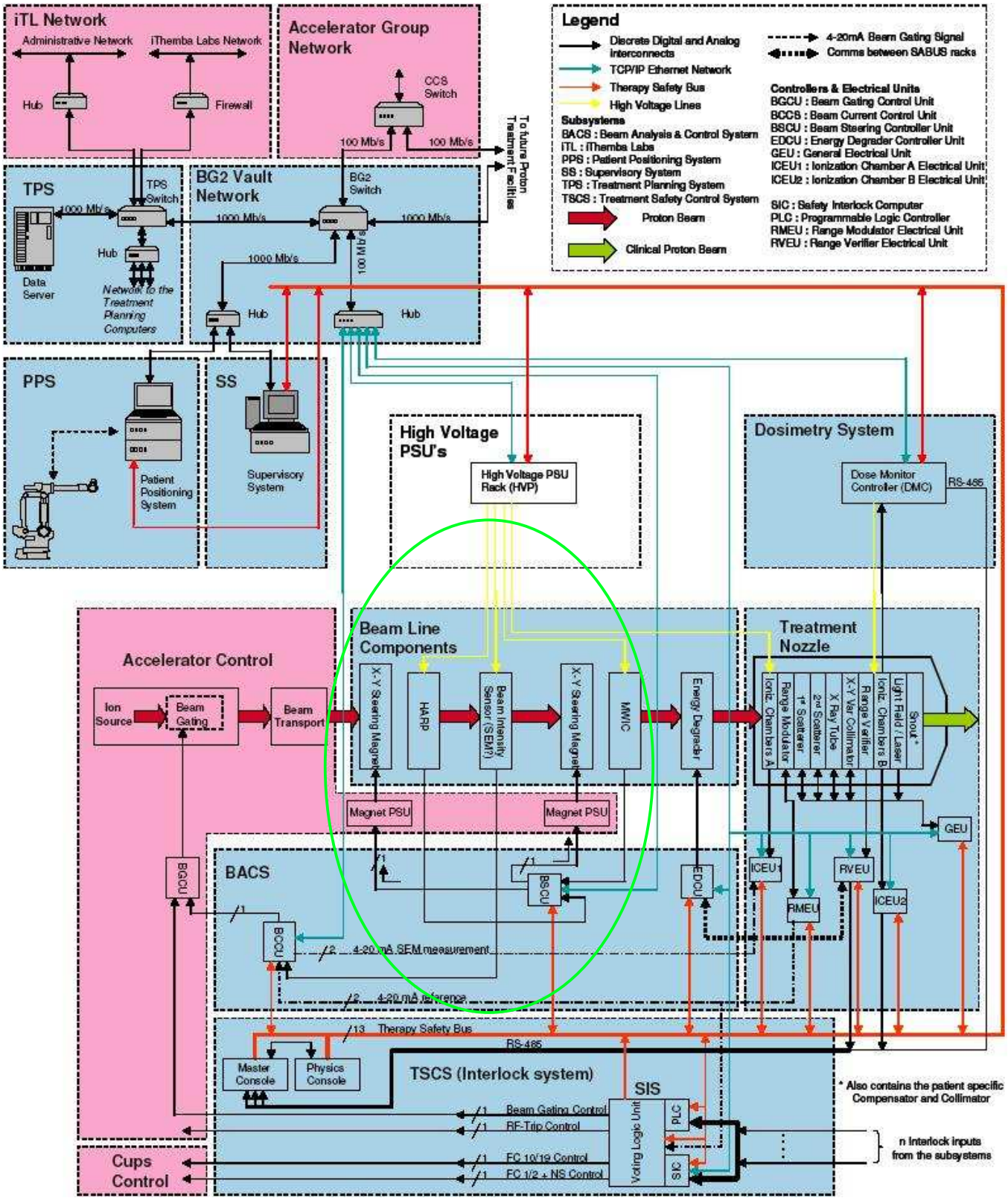


Figure 2.3: Radiotherapy Treatment Control System (TCS). This diagram shows the interconnection of the components of the system and the location of the Proton beam steering system (encircled) within the TCS

Chapter 3

Particle Beam Steering Control Applications and Methods

Due to a huge number of developments of industrial plants, the implementations of similar systems to the proton beam steering control system have been done before. Furthermore, due to the critical nature of such control systems, emphasis is made on fault tolerance and fail-safety [30], robustness, scalability and intelligence [7].

3.1 Current Control System at iThemba LABS.

According to Ivan [30], since 1990, iThemba LABS has been operating a homogeneous distributed Personal Computer (PC) based control system running on the OS/2 operation system. This control system is partitioned into four types of nodes, namely; the graphics nodes, the console nodes, the instrumentation nodes and the database nodes. The instrumentation nodes are responsible for the manipulation of the beamline components, and the graphics nodes are responsible for the display of the beam properties, such as its profile, symmetry and many other features. The database nodes in the system (two

in this case) are used to hold configuration data. This data is loaded from the nodes to memory resident databases in the subsystems of the control system. The reason behind the memory resident databases is speed improvement. Furthermore, the reason for duplicate database nodes is to ensure fault tolerance. The variables of the system are stored in a shared table and accessed through console nodes.

In this distributed control system, an Ethernet network is used to connect the different computing nodes, and a CSMA/CD 10BASE5 network is used. The network traffic is kept low to ensure efficiency. Functions of the control system are distributed over multiple computing nodes. For the beam steering control, the control task is split into two computing components that run on two different computing nodes [29]. The first component is charged with the computation of the beam's horizontal and vertical centers from the data acquired from the beam position sensors. The second component gets the beam center information from the first component and generates control signals to offset any errors that may be there in the beam position. A test experiment performed on this control system [51] showed that the control system can achieve 0.3% beam symmetry, and 4.0% on beam flatness. These performance measures are defined later in Section 3.3. Moreover, it was also evident from display of the relative beam position that the control system has an oscillatory behavior.

3.2 Other Implemented Steering Control Systems

Apart from the control system currently in operation at iThemba LABS, there are numerous other control systems that have been implemented to achieve similar tasks at other facilities. Experiments performed in [39] showed that steering accuracies of $\pm 0.015mm$ could be attained. In the same experiments and from work done in [16] it was concluded that accuracies of $\pm 0.1mm$ on the beam centroid can be achieved provided the shape of the beam profile does not change considerably over time.

A lead-lag compensation scheme was implemented in real-time at Brookhaven National Laboratories in 1987 [34]. The Los Alamos National Laboratories also implemented a steering system [47], and accuracies of $2mm$ to $1mm$ were attained, from uncontrolled deviations of about $4.5mm$. In this system, among several algorithms tested was the Least Squares steering algorithm used over multiple dipole magnets and Beam Position Monitors (BPM). The algorithm has an advantage of relieving saturation constraints on dipole magnets by distributing the steering tasks optimally over the available sets of steerers. Another successful application of beam steering done was at Los Alamos in 1998 [41]. This application used proportional-only control and achieved accuracies in the order of $\pm 1mm$.

It has also been indicated in [18] that the control algorithm to perform the steering task should have a way of adapting to changes in the beamline parameters, which may be caused by instabilities in some of the beamline components, see [45] and [11]. Furthermore, some of the beam conditioning devices are capable of producing low energy modes in the beam. These modes will respond differently to the action of the steering magnets and hence have the potential of creating unauthentic measurements. It is therefore necessary for the control algorithm to be able to deal with such disturbances.

The work done in [7] on the application of intelligent system concepts to beam alignment has gone a step further in advancing the recommendations made in [18] by using fuzzy techniques. It was also indicated in [7] that the control algorithms should have some intelligence in them. This is to reduce the dependence of the algorithm's performance on the configuration changes, and to enable the control algorithms to solve more complex tasks.

3.3 Measurement and Sensors

A control system that was implemented in [27] for beam position monitoring at the Electron Stretcher Accelerator (ELSA), like the steering control used at iThemba LABS, used measurements from the beam position sensors to compute the position of the beam, and calculate the error. Then, gain control was used to effect actuation on the beam position.

Common sensors are used in a large number of particle steering control systems and among these sensors are ionization chambers [15], beam position monitors, quadrant chambers and HARP sensors [54]. At iThemba LABS, an algorithm used to compute the relative beam position from the sensors used is the same as the one used in [43, 8, 48, 34]. The algorithm divides a sensor into halves, in the vertical and horizontal directions. To calculate the relative position of the beam in one direction, the difference in the beam intensities measured in the halves in that direction is divided by the total beam intensity. This measurement algorithm is non-linear, due to the relatively circular shape of the beam [51]; this has been confirmed in [8]. This non-linearity introduces error in the control. However, due to the small changes in beam position that are dealt with by the steering system, the algorithm does the task well, and it is not computationally demanding, which is an added advantage. The algorithm is given in the equation below.

$$BeamPosition[mm] = \kappa \frac{L - R}{L + R} + offsets \tag{3.3-1}$$

or

$$RelativeBeamPosition[\%] = (\frac{L - R}{L + R} + offsets) * 100\% \tag{3.3-2}$$

where L is the beam intensity on the left half of the Beam Position Sensor, R is the beam

intensity on the right half of the sensor and κ is the conversion constant to convert from relative position to millimeters. The definition of relative beam position is similar to that of beam symmetry and they are used interchangeably. Another measured feature of the beam is the beam flatness measured as,

$$Flatness = \frac{|D_{mean} - D_{max}|}{D_{mean}} \times 100\% \tag{3.3-3}$$

where D_{mean} is the beam intensity mean measured by the two halves of the sensor, and D_{max} is the maximum of the two.

3.4 Control Limitations

The understanding of limitations present in a system is important in controlling it [4]. The physical limitations that are present in real-world systems are of different forms, and they dictate the design of control systems [22]. The discussion below lists some of the limitations that are present in control systems.

1. Sensor Limitations:

   The limitations in sensors are in the form of *measurement noise*, which limits the bandwidth of the system, the *resolution and linearity* which affect the accuracy with which measurements are done and *sensor bandwith* which limits the applicability of a sensor in measurements of high frequency signals.

2. Actuator Limitations:

   Actuators used are physical systems which have limitations both in amplitude and *slew-rate*. Both of these limitations have a direct impact on the bandwidth of closed loop systems. A closed loop bandwidth which is too high compared to the open loop bandwidth effects too much control action, often resulting in actuator saturation. Furthermore, *sticky behavior* in some of the industrial actuators results in integrator

windup in the controller [22].

3. Disturbances

   These are unexpected inputs to the system that are often at low frequencies. They can occur at any point of the closed loop system, and influence the lower bound on the closed loop bandwidth.

4. Structural Limitations:

   Some industrial systems have substantial dead-times. This dictates the types of control that can be used, and has an implication on the disturbance rejection and bandwidth of the closed loop system.

3.5 Actuator Limitations and Anti-Windup Schemes

In many control applications, arising in the wide variety of fields where control systems are used, there are both physical and safety limits over the amount of actuation that a controller can give to the controlled system. In most cases control engineers address these actuator saturation problems by simple *min-max* logic, also called limiter logic [13]. This approach works well for preventing the actuator from saturating. However, in controllers that have integration action, this creates another problem. The integrators continue accumulating even when the control action has been limited. This gives rise to several problems.

1. Response overshoot - the response of the closed loop overshoots the required set-point.

2. Slow response - It takes some time before the response follows changes in the set-point.

3. Input vector direction - in Multi-Input Multi-Output control systems, the different

constraint levels on the control vector elements can drastically change the direction of the control vector and hence result in unexpected output behavior.

Solutions under the anti-windup scheme all aim at one goal. This goal is to leave the behavior of the closed loop unchanged if the control input is not saturating. Furthermore, for large control actions, the anti-windup solution should change or augment the closed loop system in a way that prevents saturation from occurring, and ensure that the closed loop performance assumes a graceful degradation when the anti-windup scheme is activated. The approaches under this method can be viewed as state-space and transfer function approaches. The latter are based on optimizing a cost function subject to avoiding saturation in order to achieve the goal mentioned. The former, are based on the restructuring of the designed controller in order to turn off controller modes that windup, when saturation occurs.

3.5.1 Controller Re-Configuration for Anti-Windup

One solution to the anti-windup problem has been outlined in [22] and other texts, and it relies on the controller re-structuring. This method re-configures the controller in a manner in which the integrators and slow controller dynamics will be turned off when the actuator saturates. Furthermore, the configuration preserves the transfer characteristics of the controller, for any controller state that does not cause saturation. An illustration with a controller C in block diagram is given Figure 3.1.

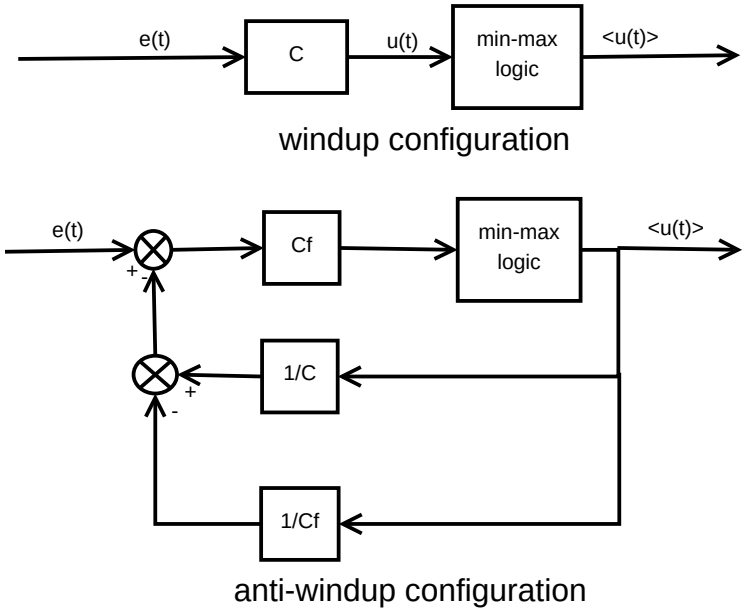


Figure 3.1: Rearrangement of controller C into components Cf, $\frac{1}{Cf}$ and $\frac{1}{C}$ to achieve anti-windup. In the diagram the error *e(t)*, unlimited control action *u(t)* and limited control action $< u(t) >$ are shown

In the rearrangement in Figure 3.1, the block *Cf* is chosen to be a factor of the controller, in this case the proportional gain. *Cf* is the limitting value of the controller as the frequency approaches zero. *u(t)* is the unconstrained plant input and $\langle u(t) \rangle$ is the constrained input. Analysis of the rearranged controller shows that it still retains its transfer characteristics. However, when the controller output *u(t)* saturates, the integral action is turned off. A simple proof that the controller retains its integrity for non-saturation control inputs is shown below.

Proof:

Given an arbitrary plant and arbitrary stable controller with a transfer function whose numerator and denominator polynomial degrees are equal such as,

$$C = \frac{K_p(s - z_1)(s - z_2)\dots(s - z_n)}{s(s - p_1)(s - p_2)\dots(s - p_{n-1})} \tag{3.5-1}$$

taking $C_f = K_p$ the proportional gain, and computing the remaining two feedback blocks as in the block diagram 3.1. It is then clear that combining the feedback arranged

controller with the foraward gain K_p in Figure 3.1 gives,

$$\frac{K_p}{1+\frac{s(s-p_1)(s-p_2)...(s-p_{n-1})-(s-z_1)(s-z_2)...(s-z_n)}{(s-z_1)(s-z_2)...(s-z_n)}} \tag{3.5-2}$$

which then simplifies to

$$\frac{K_p(s-z_1)(s-z_2)\dots(s-z_n)}{s(s-p_1)(s-p_2)\dots(s-p_{n-1})} \tag{3.5-3}$$

the original controller.

The underlying controller can be designed using different control approaches in any time-domain. The discussion below briefly outlines the approaches that may be taken and their merits and disadvantages. The only requirement for ease of reconfiguration is that, preferably, the controller transfer function should have the same numerator and denominator degree. The control engineer can always achieve this by carefully including or leaving out some non-dominant modes of the controller.

Continuous-Time Control Methods

Based on the nature of the controlled plant model, several control schemes are available in continuous-time and discrete time domains. The choice of the time domain depends on the implementation of the control system and the accuracies achievable when design is done in that time domain [22]. The most popular of the control schemes used in industry is Proportional-Integral-Derivative (PID) control. This is mainly due to the performance it offers. One group of control methods is that of the Smith Predictor method and Root Locus with Pade Approximation, which are used to control systems with dead-time. Other than that, methods like the *minor loop control* and *feedforward control* are also available [9]. After a control method is selected, which will address the necessary performance needs of the problem in hand, the task of controller tuning remains. This is the process of designing the parameters of a controller, to achieve the desired closed loop

response and disturbance rejection.

Iterative and variance control schemes have also been investigated in [40], for automatic beam centering. These approaches gave good results, however the convergence times of the controls is a concern, with iterative control being the faster of the two. Furthermore, work has been done in PID control to derive accurate numerical tuning algorithms that can be used for online PID controller tuning. The work includes the approaches that were derived by Ziegler and Nichols in the 1940's [58], which are still widely deployed in industry today. Following that, more work was on PID tuning using frequency domain methods in [31] in 1997. Parameterized approaches are also present [22], that determine the best controller out of a set of acceptable ones. However, in employing these methods, one needs to keep an eye on internal stability.

The Ziegler and Nichols tuning methods were also tested on a first order plus dead time plant model to evaluate their performance in tuning controllers for plants with different controllability levels [3]. It was shown that these tuning methods do not produce best results under all levels of plant controllability.

Internal Model Control: This control method is discussed aside from the rest of the methods mentioned above due to its special application in this work. Internal Model Control (IMC) is a robust control method that aims at achieving best set-point tracking and output disturbance rejection. This is a model based control approach whose effectiveness is greatly determined by the accuracy of the nominal plant model used.

IMC shows that if the plant model is accurate, then good set-point tracking can be achieved by using a controller that is the inverse of the plant model [49]. This is in the absence of disturbances. In his book with Goodwin *et al* [22], Morari pointed out that if the generating function of the disturbance at the output of the closed loop is known, then the disturbance can be eliminated by including the inverse of that function within the designed controller. IMC forms the foundation of the popular Smith Predictor control

and Model Predictive Control (MPC) which are both model based control methodologies.

The block diagram in Figure 3.2 is used for IMC analysis. This model is the one used for Smith Predictor control with addition of the dead-time term. It is also used for the IMC control algorithm of MPC with a slight modification to compute the error between the constrained and unconstrained control actions, see Figure 3.4 in the MPC section later on.

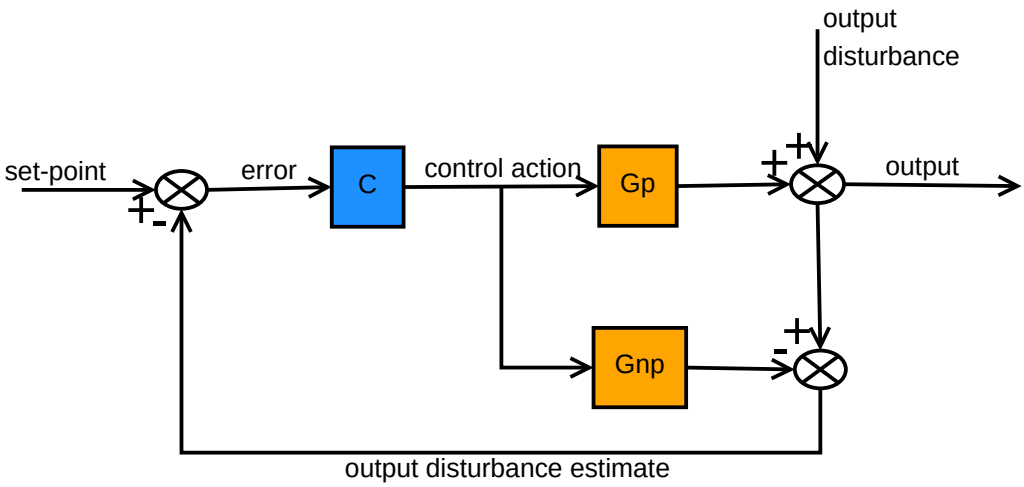


Figure 3.2: Block diagram diagram configuration used for IMC. In the block diagram, G_p is the actual plant, G_{np} is the nominal plant model and C is the controller.

Sampled Data Control Methods

Digital computers, which are an integral part of modern control systems, deal with data in the form of binary bits. This being the case some of the principles of classical control can not be applied directly in analyzing systems to be implemented on digital computers [22]. The digital control theory has well developed techniques for analyzing and designing controllers to be implemented on digital computers.

Digital control systems bear a closed resemblance to their corresponding analogue control systems, as the sampling interval used to convert the analogue data to digital format approaches zero [36] (see also [10]). A number of transformations are available for analyzing digital control systems such as the z-Transform for general analysis, the δ-

Transform, which is more appropriate when the sampling interval approaches zero [22] and the modified z-Transform for systems with a dead-time that is not an integral multiple of the sampling interval, and for analysis of intersample behavior of digital systems [10]. These transforms are used to transform difference equations resulting in the modeling of digital systems into algebraic equations that are easy to analyse.

There are several options for the design and analysis of digital control systems. The design can be carried out in the continuous time domain and the resulting controller converted to digital form by well defined transformation techniques like the bilinear transformation [22]. However, in this approach, the robustness of the final controller depends on the sampling rate, for high sampling rates this yields good controllers.

Alternatively, the design or synthesis and analysis of the control systems can be carried out in the discrete time domain. Many of the analysis and design techniques for analogue control are applicable to sampled data systems. Furthermore, there are well defined control types for digital control systems. These are reviewed in the text that follows.

Root Locus Control: The root locus control scheme is similar to proportional control in that a proportional gain is used to control the plant. Root locus plots are used to determine the stability of the system. This method does not eliminate steady state errors unless high gain is used. In discrete time control systems the gain that can be used is limited by the sampling rate used.

Minimum Prototype Design: This design method ensures that the error is zero at the sampling points. Furthermore, it achieves this in a minimum number of sampling periods. However, this method has been noted for poor intersample behavior [22].

Minimum-Time Dead-Beat Control: Improvement is made upon the minimum prototype control by imposing further conditions that, the response due to step disturbances and inputs must reach steady-state at the same sampling intervals and that the closed loop should settle in n sampling intervals where n is the order of the characteristic function

of the plant. This improves the intersample behavior from the previous design method.

The two design approaches discussed above work on the principle of placing all the closed loop poles at the origin to ensure finite settling time. They achieve this through Pole Placement techniques. A general form of pole placement allows for the closed loop poles to be placed at some compromise position between 0 and 1, in order to relax the requirements on the control effort. This has an effect of slowing down the system.

Dahlin Control: This control method relaxes the conditions that are imposed by the minimum prototype control scheme. It allows the response of the closed loop system to behave like a lag with dead time [10].

Repetitive Control: In this scheme, periodic inputs are exploited in order to ensure that the complementary sensitivity of the closed loop is 1 at the input frequencies. This is achieved by incorporating the discrete time model of the input signal in the denominator of the controller [22].

3.5.2 Reference Governor (RG) and Input Saturation

Model Predictive Control, discussed in detail in Subsection 3.6.1, has been applied on a wide variety of industrial applications to address the problem of constraints. This method relies on the prediction of future evolution of the system outputs based on the plant model, hence it has also been known as Model Based Predictive Control (MBPC). However, this control scheme has heavy computational requirements and has been mostly suited to slow systems. The Reference Governor (RG) has been investigated in [5], which is a memoryless nonlinear module whose task is to manipulate the reference input to the control system in order to observe the constraints. More work was done in [20] to extend the method to discrete time systems.

In his dissertation, Bemporad [5] indicated that the governor has the effect of slowing

down the closed loop, hence a fast enough primal controller has to be designed to remain robust in the presence of the Reference Governor. This method of handling constraints is less computationally demanding than Model Predictive Control approach and it separates the constraint serving from the plant control. However, it has been stated in [6], that this method does not perform well under large exogenous disturbances on the closed loop. Furthermore, due to the computation intensity of this scheme, it is recommended for unstable plants.

3.5.3 Error Governor and Input Saturation

An alternative to the RG discussed earlier is the Error Governor (EG) which responsible for manipulating the error fed to the controller [26]. Like the RG, this approach separates the plant control from the constraints and is also less computationally demanding.
Moreover, unlike the RG method, EG is more suited for stable plants controlled by *neutrally stable* controllers [26]. The structure given in Figure 3.3 below is used for the EG problem formulation.

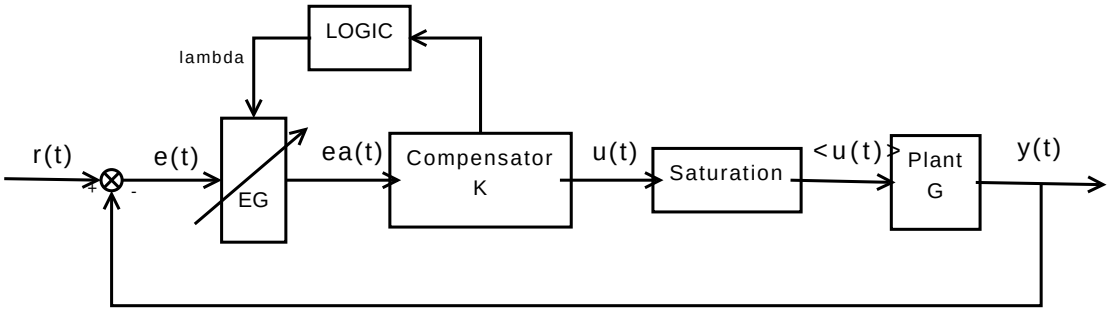


Figure 3.3: The EG problem block diagram is a full diagram including the reference input *r(t)*, the augmented error *ea(t)*, the compensator with transfer function K, the plant with transfer function G and the plant output *y(t)*. Also shown, is the LOGIC block for determining the λ value, between 0 and 1, to be used to augment the error signal *e(t)* to give *ea(t)*

In this problem the aim is to augment the error signal with a nonlinear constant λ in the range 0 to 1, in order to prevent saturation behavior from occurring. The mathematical formulation of the problem as can be found in [26], and it is as follows.

Given a stable linear system *G(s)* controlled by a neutrally stable compensator *K(s)* represented by the following models,

G(s):

$$\dot{x}(t) = Ax(t) + B\langle u(t)\rangle \tag{3.5-4}$$

$$y(t) = Cx(t) \tag{3.5-5}$$

K(s):

$$\dot{x}_k(t) = A_k x_k(t) + B_k e(t) \tag{3.5-6}$$

$$u(t) = C_k x_k(t) \tag{3.5-7}$$

The EG augments the error resulting in the augmented compensator.

$K_a(s)$:

$$\dot{x}_c(t) = A_c x_c(t) + B_c \lambda(t) e(t) \tag{3.5-8}$$

$$u(t) = C_c x_c(t) \tag{3.5-9}$$

The choice of a $\lambda(t)$ at time t that prevents saturation from occurring is done following the algorithm outlined in [26] as follows;

define a scalar valued finite function

$$g(x_0) : g(x_0) = \| \, u(x_0, t) \, \|_\infty \tag{3.5-10}$$

where $x_0 = x_c(0)$ and $u(x_0, t) = C_c e^{A_c t} x_c(0)$ and

$$\dot{x}_c(t) = A_c x_c(t) \tag{3.5-11}$$

$$u(t) = C_c x_c(t) \tag{3.5-12}$$

also define a set (symmetric about the origin)

$$B_{A,C} = \{x : g(x) \leq |u_{max-min}(t)|\} \tag{3.5-13}$$

With the function and the set defined, the problem is to determine a scalar $\lambda(t)$ between 0 and 1 such that saturation never occurs.

The procedure for selecting the appropriate value of λ at time t is as follows.

- if the states $x_c(t)$ are interior points in the set $B_{A,C}$ then choose $\lambda(t) = 1$.
- if the states $x_c(t)$ are elements of the set $BdB_{A,C}$ (on the boundary of $B_{A,C}$) then choose the largest $\lambda(t)$ that satisfies,

$$\lim_{\varepsilon \to 0} sup \frac{g(x_c(t) + \varepsilon[\dot{x}_c(t)]) - g(x_c(t))}{\varepsilon} \leq 0 \tag{3.5-14}$$

or if $g(x_c)$ is differentiable, $\lambda(t)$ is chosen such that

$$Dg(x_c(t))\dot{x}_c \leq 0 \tag{3.5-15}$$

where $Dg(x_c(t))$ is the Jaccobian matrix of $g(x(t))$.

- if the states $x_c(t)$ are not elements of $B_{A,C}$ then $\lambda(t)$ is chosen to minimize (3.5-14). However the problem may lead to $\| u(x_0,t) \|_\infty > |u_{min-max}|$. Then a value of $\lambda(t)$ that drives the states into $B_{A,C}$ is chosen.

3.6 Receding Horizon Optimal Control Schemes

3.6.1 Model Predictive Control and Input Saturation

Model Predictive Control is an algorithm based control scheme that is aimed at solving a receding horizon optimization problem over finite intervals in order to come up with an optimal control action. This control scheme takes into consideration the possibilities of future limitations on the inputs, outputs and states [22]. This approach has the advantage of not seeking a globally optimal solution, instead the solution is required to be optimal over the interval of interest.

The formulation of the MPC control problem uses past inputs, outputs, predictions of future inputs, outputs and disturbances in order to come up with an optimal controller to address the control requirements and constraints handling. A traditional formulation of the MPC problem can be found in [38, 13], and is stated below.

$$min_{\triangle u(k)\ldots\triangle u(k+m-1)} \sum_{l=1}^{p} \|y_p(k+l|k) - r(k+l)\|^2_{\Gamma_l} + \| \triangle u(k+l-1)\|^2_{B_l}$$

where the predicted output values $y_p(k+l)$ are given by

$$y_p(k+l|k) = \sum_{i=1}^{l} H_i \triangle u(k+l-i) + \sum_{i=l+1}^{n-1} H_i \triangle u(k+l-i) + H_n u(k+l-n) + d_p(k+l|k) \quad (3.6\text{-}1)$$

and the predicted disturbance $d_p(k+l|k)$ is given by

$$d_p(k+l|k) = d_p(k|k) = y_m(k) - \sum_{i=1}^{n-1} H_i \triangle u(k-i) + H_n n(k+l-n) \quad (3.6\text{-}2)$$

and

$$\sum_{l=1}^{p} C_{yl}^{j} y_p(k+l|k) + C_{ul}^{j} u(k+l-1) + c^j \leq 0 \; j = 1, n_c \quad (3.6\text{-}3)$$

where

$y_m(k)$ is the measured output at instant k

$\triangle u(k+l)$ is the back difference $u(k+l) - u(k+l-1)$

H_i are model step response matrix coefficients and i = 1,n

n is the model truncation order.

n_c is the number of constraints.

p is the horizon length (generally p$\gg$n)

m is the number of manipulated variable moves in the future.

$\|x\|_Q^2 = x^T Q x$

Γ_l and B_l are weighting matrices.

$C_{yl}^{j}, C_{ul}^{j}, c^j$ are constant matrices.

This formulation seeks to minimize the 2-norm of the error between the process output and the setpoint. The initial formulation of this algorithm is for Dynamic Matrix Control (DMC) [13], and results in a quadratic program. The results of this algorithm can not be analyzed for stability and robustness in the presence of constraints, and the tuning is not straightforward. An alternative algorithm is used, to minimize the 1-norm of

the error between the predicted constrained output and the ideal unconstrained output. This algorithm results in a linear programming problem and is an improvement on the traditional one which is not easy to analyze even for constrained situations.

The ∞-norm of the error between the plant output and the setpoint is also used in which case a controller for the unconstrained case is designed using Internal Model Control (IMC). Thereafter the minimization is applied to the error between the predicted constrained and the ideal unconstrained output. The block diagram in Figure 3.4 is used for analysis of the IMC algorithm of MPC.

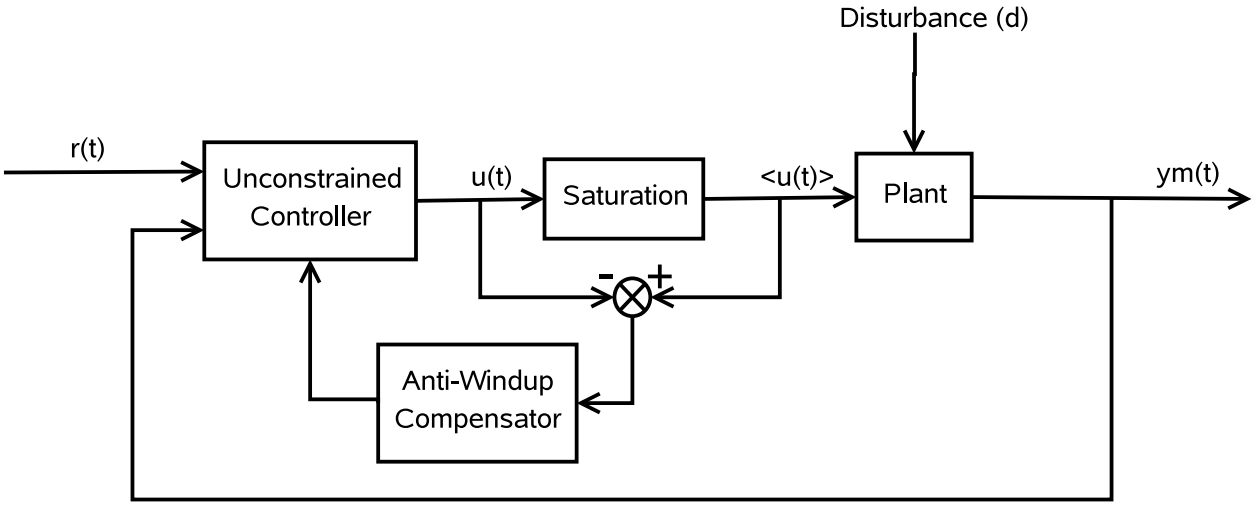


Figure 3.4: The general block diagram used for the IMC algorithm of MPC

The cost function for minimizing the predicted constrained output and the ideal unconstrained process output is given below as stated in [13].

$$min_{\triangle u(k)\ldots\triangle u(k+m-1)} max_{l=1,p;i=1,r} w_{li}|y_p(k+l|k) - r_p(k+l)| \qquad (3.6\text{-}5)$$

3.7 Other Receding Horizon Optimal Control Strategies

Other Receding Horizon Optimal Control (RHOC) approaches to the input saturation problem deriving from MPC have also been suggested. Among these methods is the L_2 anti-windup method proposed in [6]. This method transforms the anti-windup problem into a state feedback problem. The full details on formulation of the method are in [6]. This method only adds an extra degree of anti-windup freedom on top of the MPC strategy to deal with input limitations. As a result this method is not attended here in order to limit the scope of this thesis.

3.8 Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator has also been used to control plants with input saturation constraints and worked well. This method uses the quadratic matrices Q and R within the Ackerman's equation to penalize the variables in the cost function. Setting the quadratic matrix values of the control input large with reference to the other variable quadratics helps avoid saturation [22]. Below is a statement of the Ackerman's LQR cost function

$$J(y,u) = \int_{t0}^{t} y^T Q y + u^T R u dt \tag{3.8-1}$$

3.9 Safety and Fail-Safety

Due to the nature of the proton beam steering control problem, safety conditions are put in place to avoid hazardous conditions due to over-steering and other problems arising in

other parts of the beamline. The safety in the entire treatment control system is mainly handled by safety *Interlock signals* [54]. The steering control system has an interface to a set of signals on the Therapy Safety Bus (TSB), which is a collection of safety signals, to interrupt when a problem is detected within the proton beam steering unit. These are shown in Figure 3.5 below and described in subsection that follows outlined below:

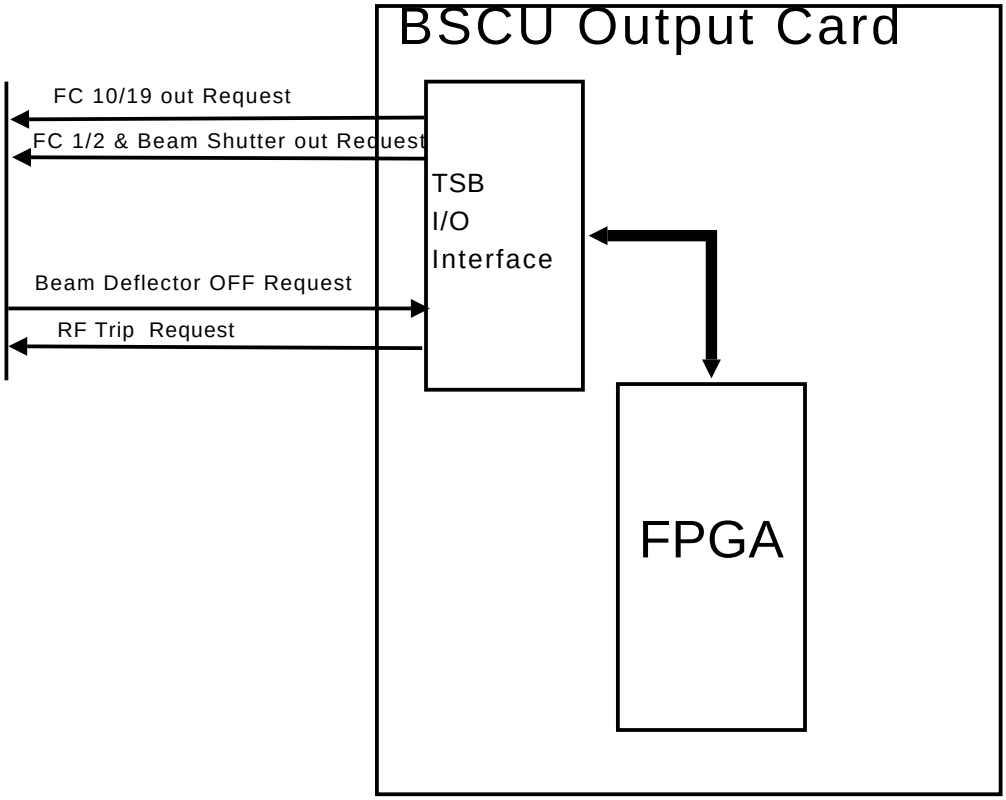


Figure 3.5: The Beam Steering Control Unit (BSCU) output card showing the interface of the BSCU to the TSB

3.9.1 The Therapy Safety Bus Interface

The BSCU is able to set the signals indicated in Figure 3.5 above to GO or NOGO. If there is a fault within the unit the BSCU sets the correct signals in order to interrupt the beam. All of the signals are discrete with a '1' indicating a GO and '0' indicating a NOGO status. The descriptions of the interlock signals are given below:

1. Faraday Cup 10/19 Out request: - Faraday Cups (named after Michael Faraday)

are conductive metal cups constructed to capture charged particles and give out electrical current, and are used in the measurement of charged particle beams currents. This signal indicates when Faraday Cups 10 and 19 are completely out of the low energy beam. For fail safety the NOGO (0) state of this signal (which may be due to a fault) means the Faraday cups are inserted into the beamline.

2. Faraday Cup 1/2 and Beam Shutter Out request: - This signal indicates when Faraday Cups 1 and 2 and the Beam Shutter are completely out of the high energy beam. The GO and NOGO states of this line are also assigned with a safety perspective. This contact and the previous one are triggered when a patent's movements are sensed [54].

3. RF Trip OFF request: - This signal requests the RF Trip OFF. The contact requests that RF trip be on when it has a NOGO (or fault) state, for fail-safety.

4. Beam Deflector OFF request: - The signal requests the Beam Deflector OFF. A NOGO (or fault) state on this contact activates the Beam Deflector.

The diagram in Figure 3.6 below shows the logic dependence of the BSCU activation on the respective *Interlock signals*, and its capability to modify the status of the contacts.

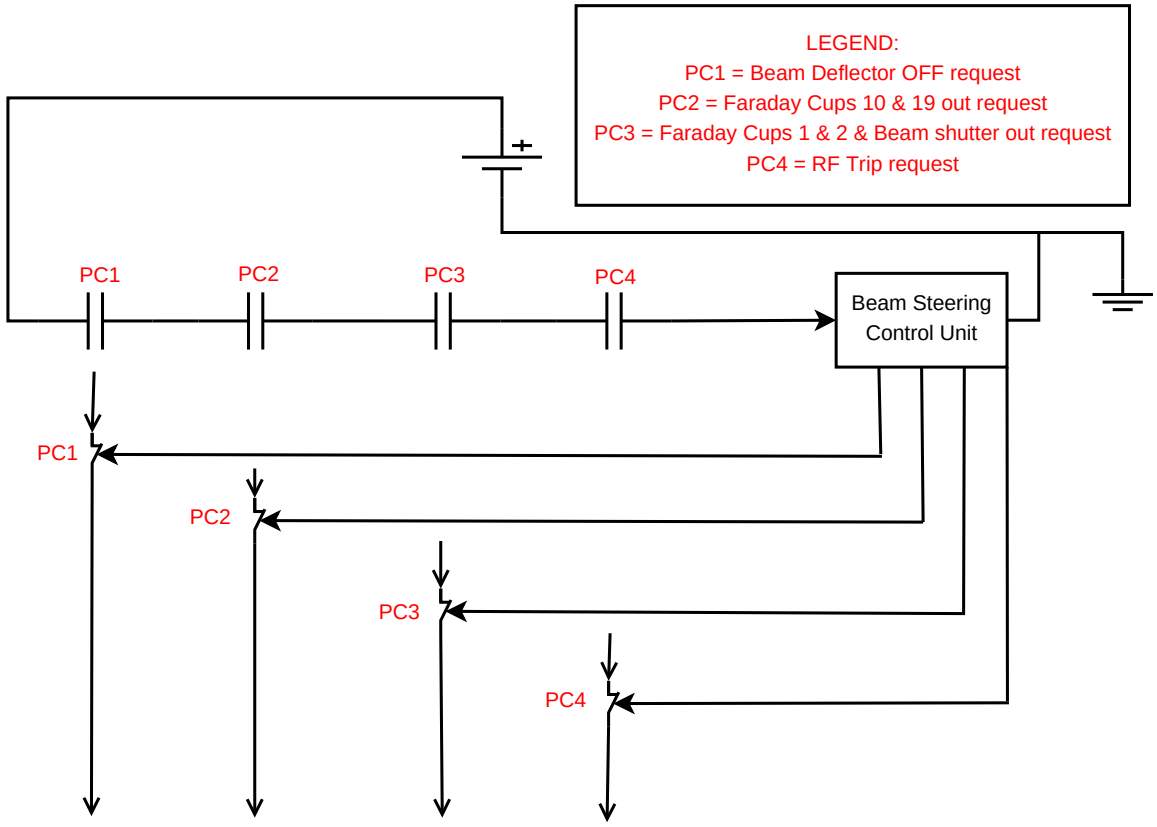


Figure 3.6: Ladder logic diagram showing the importance of the TSB lines in the functionality of the BSCU. At any given time the TSB signals should be on GO state in order for the Beam Steering Control Unit to be enabled. They are placed on the HOT side of the BSCU power supply for fail-safety. Furthermore, the diagram also show the ability of the BSCU to manipulate the interlock signals' status using relay switches

3.9.2 Fault Detection

In order for the controller to assume a fail-safe behavior, it has to have a way of detecting faults within the system hardware and software. The severity of the fault has to be evaluated and appropriate permissive signals triggered. Then the controller has to notify the supervisory system on the type of fault that occurred. The fault finding task is performed independent of other control tasks by running a self diagnostic routine to check all the important aspects of the steering control unit. The potential fault or hazard areas are:

1. Communications: Here the steering system checks to make sure that there is

communication between itself and other modules. These are, the sensors, the steering magnets, the TSB, the Ethernet connection to the SS, the RS-485 and RS-232 channels as well as the SABUS interface.

2. Position Error: This is a routine that is coupled with the beam steering controller, to ensure that the beam position error is within safety error limits determined by iThemba LABS. If the error is too large appropriate interlocks are triggered and a status message is sent to the Supervisory System (SS).

3. Software Deadlocks: These are the problems that may arise due to race conditions within the parallel data sharing components of the BSCU software. They can be detected if the software fails to respond to changes within a specified time.

Chapter 4

Proton Beam Steering System Analysis

4.1 System Definition

The proton beam steering system comprises two sets of steering dipole magnet pairs. In each dipole pair one dipole steers the beam in the vertical (Y) and the other in the horizontal (X) directions as shown in Figure 4.1, where the dipole pair sets are marked *Steerer1* and *Steerer2*. The magnet pairs are positioned at some distance from each other on the beamline, and each pair of dipole magnets is powered from a set of bipolar power suppliers (actuators) [32]. Each Steerer has a sensor(s) for its control. The system becomes a coupled MIMO system whose control is complex due to the dependence of the distinct sets of magnets on each other's effects on the beam. The MIMO analysis of the

system results in a 4×4 transfer function matrix of the form given in (4.1-1).

$$G_{full} = \left[\begin{array}{cc|cc} g_{x11} & g_{x12} & 0 & 0 \\ g_{x21} & g_{x22} & 0 & 0 \\ \hline 0 & 0 & g_{y33} & g_{y34} \\ 0 & 0 & g_{y43} & g_{y44} \end{array}\right] \tag{4.1-1}$$

The structure in (4.1-1) is block-diagonal, and it can be decomposed into two 2×2 control tasks.From the physical model of the system shown in Figure 4.1, rearrangements were made to come up with the structure in (4.1-1). Each 2×2 control task is a cascaded steering task in one steering direction for Steerer1 and Steerer2, hence the use of subscripts x and y on the transfer functions to indicate that they are in the respective direction.

From this point on, the control of the steering system block for one steering direction will be considered in detail, and the directional subscripts are omitted.

4.2 MIMO Structure Analysis

4.2.1 General Analysis

In general, in order to achieve the two main goals of the beam steering control system, it should be possible to measure outputs that can ascertain beam centeredness, and parallelism to the beam line in each beam steering axis (x and y). The configuration that allows these requirements to be met is shown in Figure 4.1. In this alignment, the two sets of steering magnet pairs are placed one after the other down the beamline, and the two sensors are placed at different locations on the beamline after the steering magnet sets. In this arrangement, the beam is parallel if the difference between the two sensor readings is zero. Furthermore the beam is centered if it is parallel, and passes at the

center of either one of the sensors. Otherwise, if it is not parallel to the beamline axis then it is only centered at the point of measurement. However, the first sensor is chosen to measure the centeredness for reasons that will be discussed later.

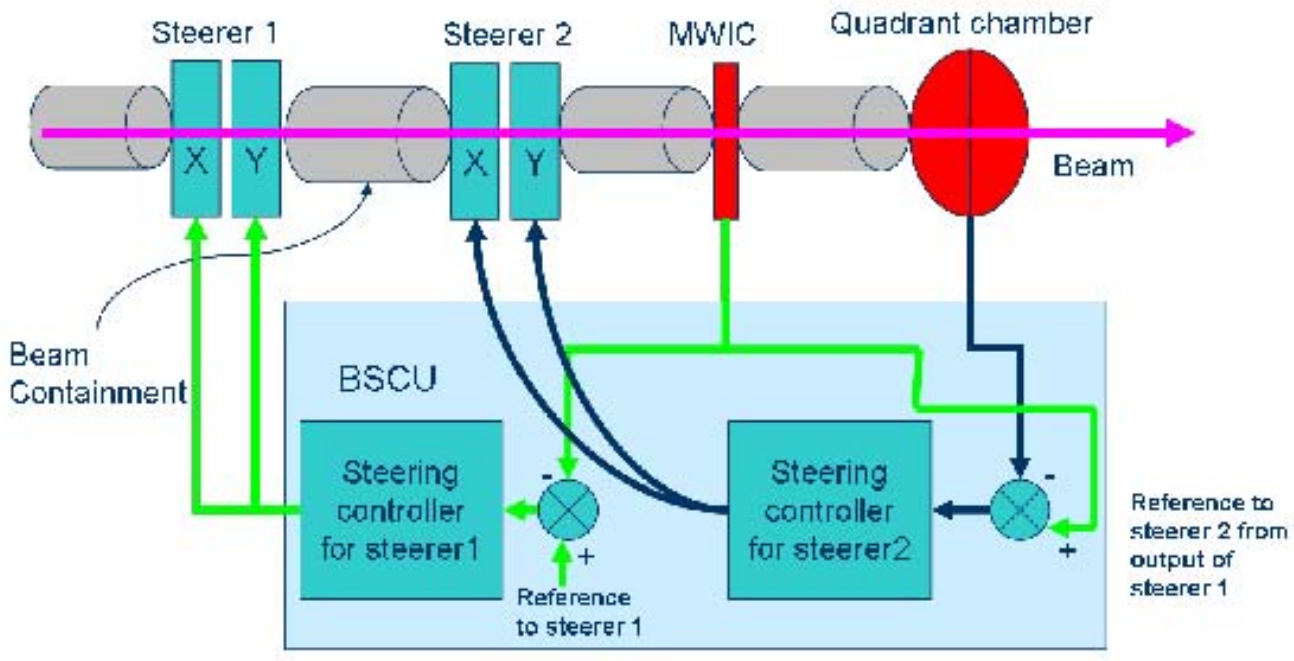


Figure 4.1: Physical model of the beam steering system showing the relative positions of the system components for fully coupled system structure (MWIC = Multiwire Ionization Chamber)

It should be noted that a configuration with a magnet set between the sensors is not the best way to go about this problem, since it makes it difficult to establish if the beam is parallel. In fact, it really complicates the problem.

This can be shown to result in two fully coupled MIMO system with a structures of the form (4.2-1) in each steering direction.

$$G_{full-1D} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \tag{4.2-1}$$

In this transfer function matrix, a transfer function g_{ij} (on the i^{th} row and j^{th} column) is a transfer function from the j^{th} input to the i^{th} output. The geometrical analysis of this system shows that the DC gains $g_{21dc} > \{g_{11dc}$ and $g_{22dc}\} > g_{12dc}$, and all of them

depend on the distance between the magnet sets and the sensors. The typical open loop step response showing interaction between the Steerers is shown in Figure 4.2.

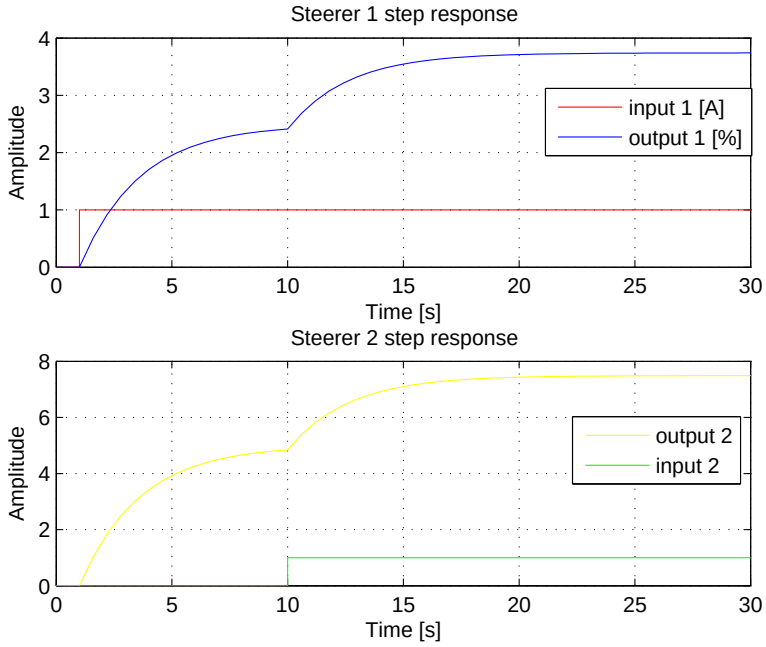
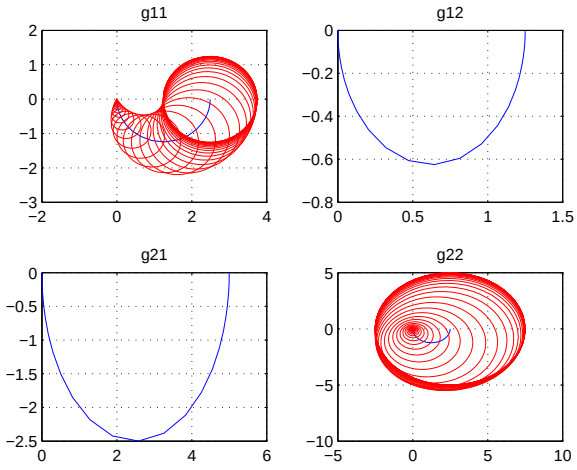


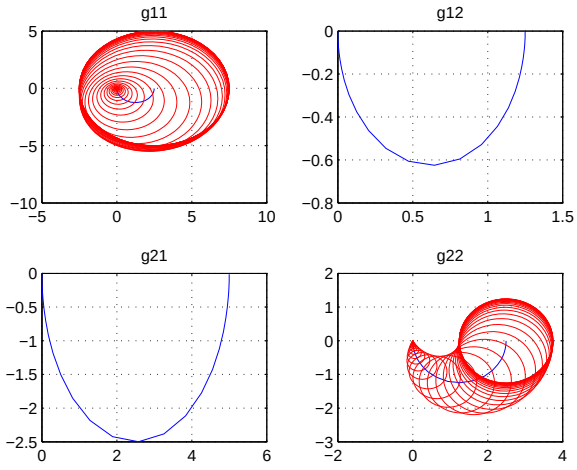
Figure 4.2: Unit step response showing full interaction for the steering task in one steering direction. Each step input applied at different times from the other has an effect on both outputs

Diagonal Dominance analysis of this model shows that it is neither row nor column dominant. This is shown in the illustrative Nyquist-Gershgorin plot in Figure 4.3 using equal spacing between all components, which yields models of relative gains in (4.2-2),

$$G_{full-1D} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \frac{2.49}{2.6s+1} & \frac{1.25}{2.6s+1} \\ \frac{5}{2.6s+1} & \frac{2.49}{2.6s+1} \end{bmatrix} \tag{4.2-2}$$



(a) Gershgorin circles encircle the origin of g_{22} Nyquist plot: therefore row dominance fails



(b) Gershgorin circles encircle the origin of g_{11} Nyquist plot: therefore column dominance fails

Figure 4.3: Direct Nyquist Array and Gershgorin circles to show that the full steering system structure in a given steering direction is not diagonally dominant. This is shown by the Gershgorin circles that encircle the origin of the Nyquist plots

4.2.2 Physical Decoupling

However, by physically moving the sensor for Steerer1 (MWIC) closer to the dipole magnets of Steerer2, the dominance of the off-diagonal transfer function g_{12} diminishes to zero. Placing the sensor for the first pair of steering magnets (MWIC) within or very close to the second pair of steering magnets as shown in the physical model in Figure 4.4, the steering task can be physically decoupled into a lower triangular structure, in which the second steering stage takes the output of the first stage as a reference. The structure is shown in (4.2-3).

$$G_{tr} = \begin{bmatrix} g_{11} & 0 \\ g_{21} & g_{22} \end{bmatrix} \tag{4.2-3}$$

where, g_{11} is the transfer function from the input of Steerer1 to the its output, g_{22} is the transfer function from the input of Steerer2 to its output and g_{21} is the transfer function of the input of Steerer1 to the output of Steerer2. Since a manipulation on the input of Steerer2 is not seen by Steerer1 the corresponding transfer function is zero.

In this scenario, the first steering task ensures the centering of the beam, while the second stage maintains the beam parallel to the beamline axis, by following the output of Steerer1. The triangular structure can then be decoupled by multiplying it by a precompensator $\mathbf{K}$ of the form (4.2-4), to eliminate g_{21}. Alternatively, SISO controllers can be designed for the main diagonal transfer functions. However, due to the dominance of the off-diagonal transfer function, this alternative is more likely to yield undesirable results.

$$K = \begin{bmatrix} 1 & 0 \\ k_{21} & 1 \end{bmatrix} \tag{4.2-4}$$

where, $k_{21} = -\frac{g_{21}}{g_{11}}$.

The procedure discussed up to here completely decouples the Beam Steering Control System into a diagonal system given as (4.2-5)

$$G_{full} = \left[\begin{array}{cc|cc} g_{x11} & 0 & 0 & 0 \\ 0 & g_{x22} & 0 & 0 \\ \hline 0 & 0 & g_{y33} & 0 \\ 0 & 0 & 0 & g_{y44} \end{array}\right] \tag{4.2-5}$$

Another approach is to control the triangular structure IMC with SISO controllers designed for each of the diagonal transfer functions g_{11} and g_{22}.

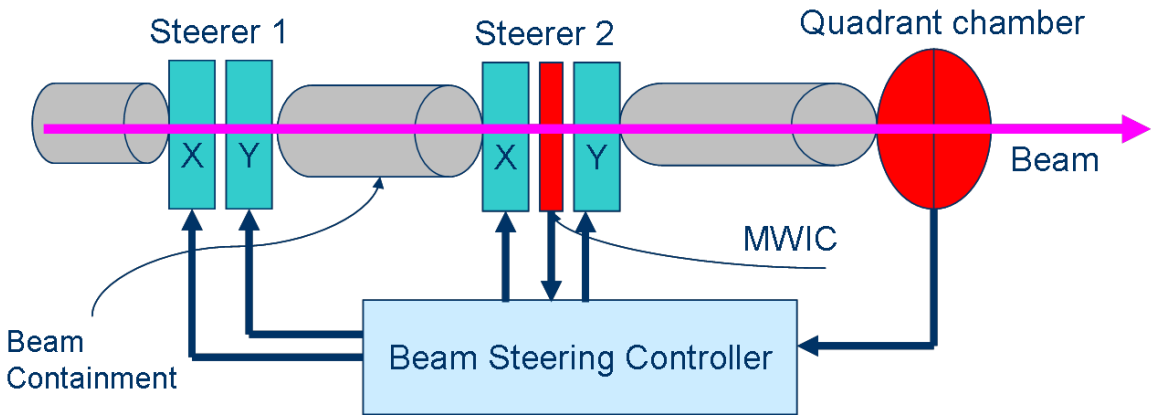


Figure 4.4: Physical model of the proton beam steering system with MWIC moved close to Steerer2 to eliminate g_{12}

With the steering system components set as shown in Figure 4.4, the two steering control systems can be designed in the same manner, and then cascaded by feeding the beam position as read from the MWIC as the reference to the second steering feedback loop. The block diagram in Figure 4.5 below shows the physical cascading of the two feedback loops.

From this point, the design of the steering control is done for one steering dipole and PSU set and it is duplicated for the other at implementation. The manipulated inputs of the steering system are the Voltage (V) settings to the power suppliers and the controlled output is the proton beam position in cartesian coordinates.

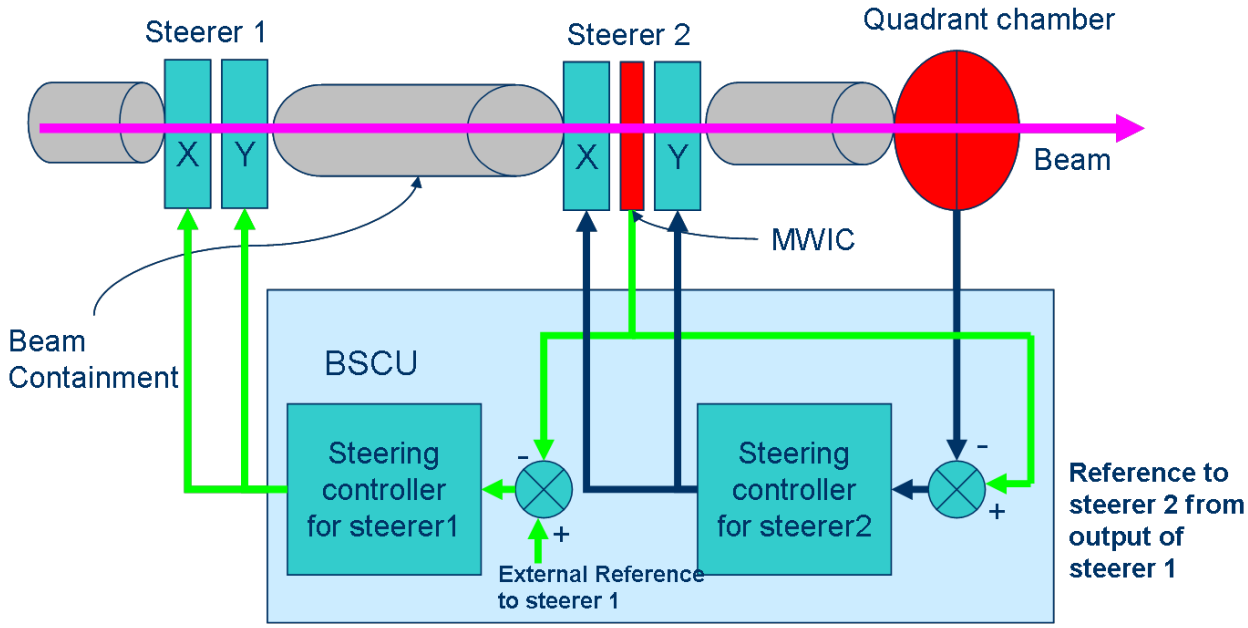


Figure 4.5: Physical model of BSCU showing cascaded steering SISO loops: Steerer2 takes reference from output of Steerer1

In the light of the information above, the basic building block of the steering system is a combination of a power supplier and a dipole magnet, the input of which is voltage(V) and output is relative beam position (%). This system is depicted in the block diagram in Figure 4.6 below.

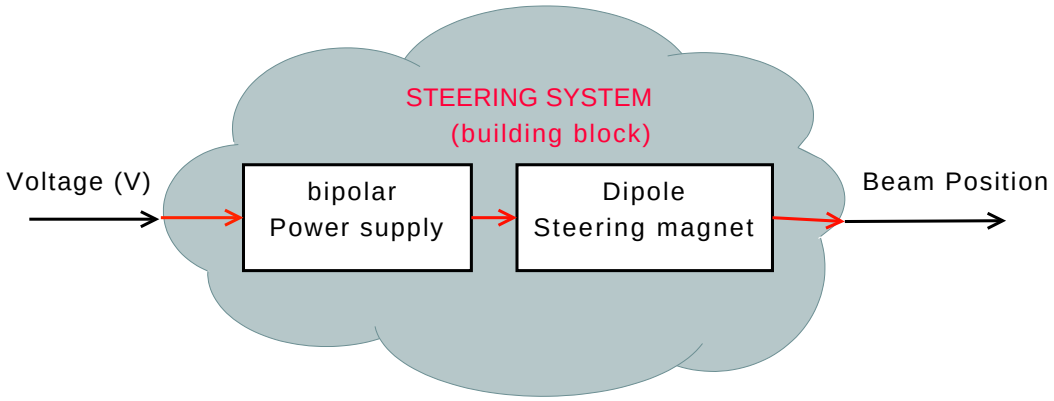


Figure 4.6: Block diagram showing the definition of the steering system's basic building block

4.3 System Identification

4.3.1 Input/Output Ranges

The controlled output voltage is in the range of $\pm 5V$ [32], which is mapped to a range of $\pm 12.45\%$ in relative beam position.

4.3.2 Sensors

The sensors used for the measurement of the beam position are Multiwire Ionization Chambers (MWIC) [15] and Beam Position Monitors; see Section 3.3. The MWIC sensor works on the principle of ionization of gases by charged particles, thus inducing current in equally spaced wires placed in the gas chamber. The output current of these sensors is low and has to be amplified, and filtered to get rid of measurement noise.

4.3.3 Modeling from Physics Principles

A simplified model of the basic building block can be represented by a series resistor-inductor (RL) circuit as shown in Figure 4.7. Since the relation ship between current and magnetic field is proportional as given by the equation (4.3- 1) [56], the model for the system is a first order model with no dead-time. This is shown in the short derivation of the form of the model.

$$\begin{aligned} I[Amps] &= \frac{1}{0.4\pi} B_{\perp}[G] G[cm] \\ &= \alpha B_{\perp}[G] \end{aligned} \tag{4.3- 1}$$

where,

$$\alpha = \frac{1}{0.4\pi}[G]G[cm]$$

and the current I is proportional to the required normal magnetic field $B_\perp$, and the magnet pole gap G.

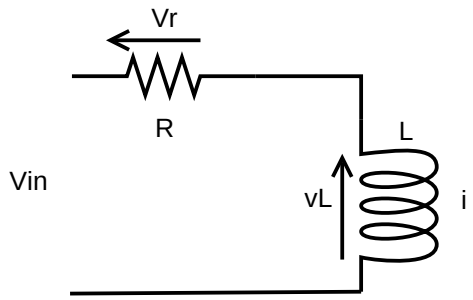


Figure 4.7: A simplified model of the of the steering system basic building block with series resistor R and inductor L. The output is represented as coil current i which will be replaced by scaled magnetic field $\alpha B_\perp$ in the derivations due to their proportional relationship.

The transfer characteristics of the circuit above is of the form,

$$\frac{I(s)}{V_{in}(s)} = \frac{\frac{1}{R}}{\frac{L}{R}s + 1} \tag{4.3- 2}$$

replacing the current *I(s)* by a constant multiple of the normal magnetic field $\alpha B_\perp$ yields,

$$\frac{B_\perp(s)}{V_{in}(s)} = \frac{\frac{\alpha}{R}}{\frac{L}{R}s + 1} \tag{4.3- 3}$$

4.3.4 Step Perturbation

In experiments conducted in [50, 51], step tests were used to come up with a model for the steering system. The plot shown in Figure 4.9 is a sample outcome of the first set experiments to establish the time constant. The PSU and dipole magnet are shown in Figure 4.8. Below is the derived model of the steering system basic building block.

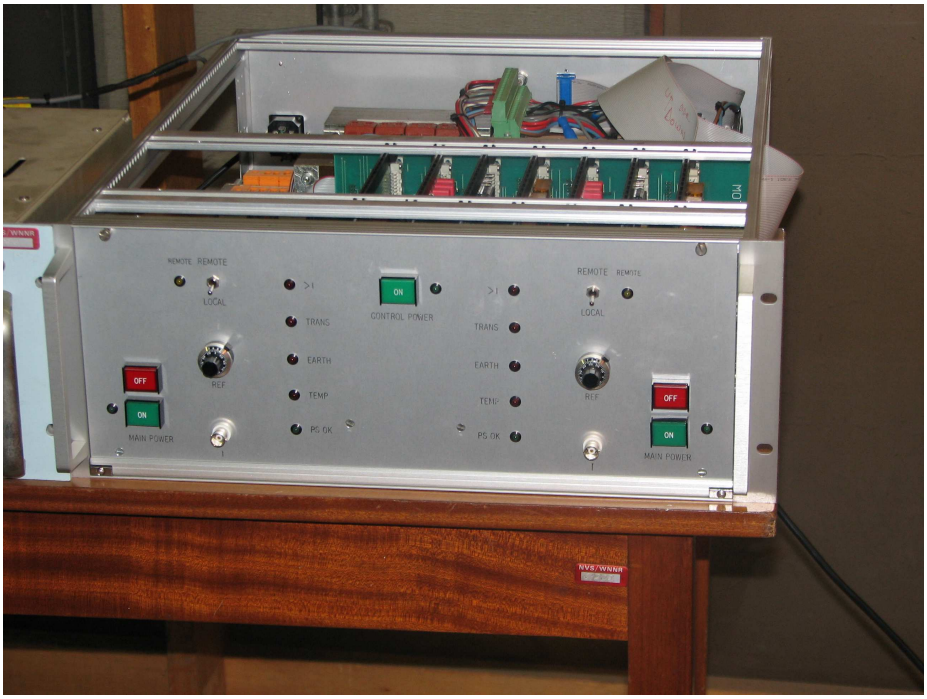
$$G(s) = \frac{2.49[\%/V]}{2.6s + 1} \tag{4.3-1}$$

digitized to be,

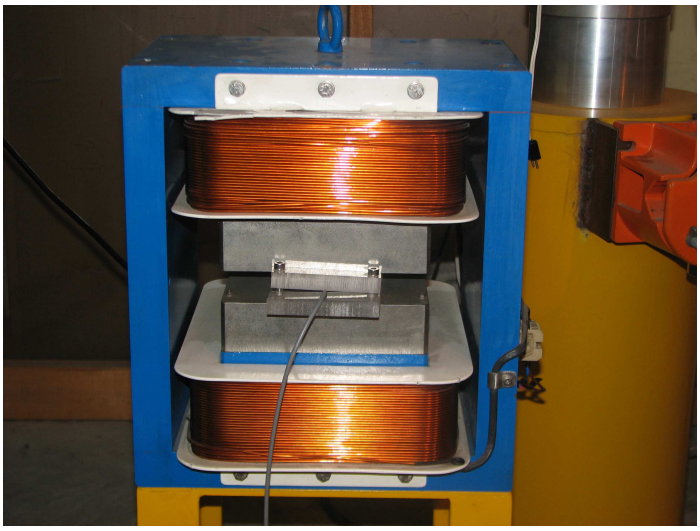
$$G(z) = \frac{0.009559\%/V}{z - 0.9962} \tag{4.3-2}$$

at a sampling interval of 0.01 seconds.

The sampling rate is chosen to be $\simeq 10$ times faster than the targeted $\simeq \times 5$ to 10 closed loop speed up. This means that a sampling interval of 0.025 is a reasonable choice, so 0.01 seconds is used. However, for PI control 0.005 seconds is used.



(a) Bipolar Power Supply



(b) Dipole Steering Magnet

Figure 4.8: Pictures of the bipolar power supply and dipole magnet used at iThemba LABS

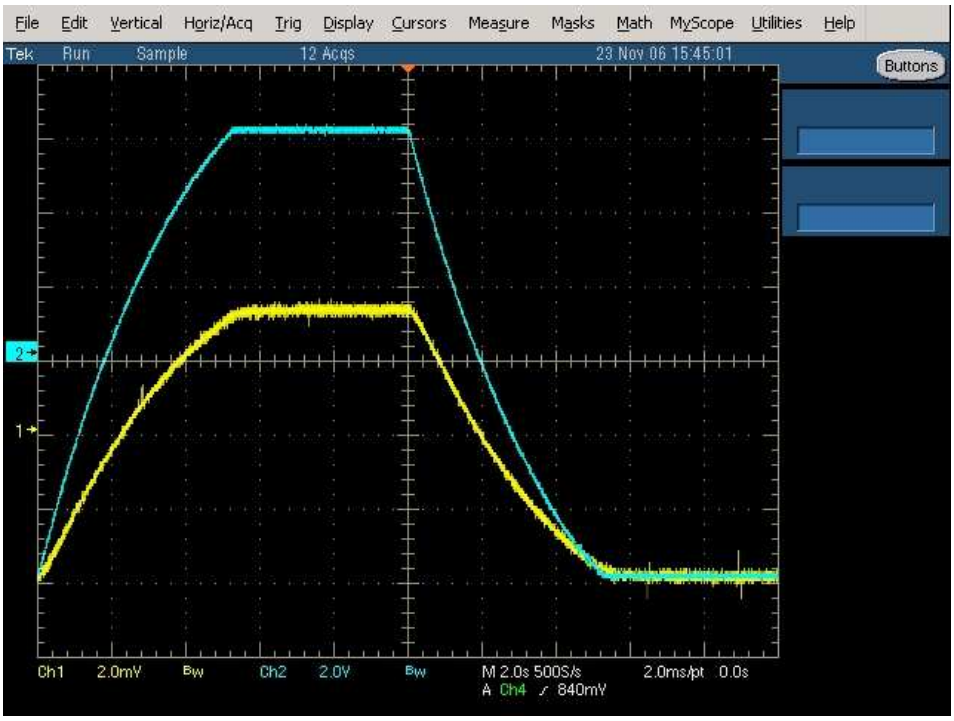


Figure 4.9: Step response of the steering system's basic block with magnetic field measured. This is obtained from an oscilloscope with the vertical axis as voltage (Volts for the power supply output (blue), and mV for the gaussmeter output (yellow)) and time in seconds on the horizontal axis. This is used to determine the time constants of the block. Modeling from Physics suggests that there is no deadtime so the step input is not plotted here

The steering system building block is a first order system, with a time constant of 2.6 seconds and a DC gain of $2.49\%/V$. There were a number of models resulting from the experiments but the one above was chosen as it is the one which is more difficult to control [2]. The time constants for other different magnets and power supply is not expected to be considerably different from that of the set that were used for System Identification. However the DC gain of this model is dependent on the distance between the steering dipole magnet and the beam position sensor increasing with increasing distance. Depending on the robustness of the closed loop to changes in these two parameters, they may be controllable by the same controller, or a different controller may need to be tuned for each steering set [9].

4.4 Stability and Relative Stability

The two models given in the previous subsection can be seen to be stable as one has a pole at $s = -0.38$ and the other at $z = 0.9623$. The second model also shows that it does not have any ringing.

The bode plots shown in Figure 4.11 and 4.10 show the relative stability margins of the continuous and discrete time models respectively. It should be noted that accuracy of the discrete time model and its stability depend on the sampling interval used to convert it from the continuous time model. As the sampling interval approaches zero the model becomes more accurate, but the bandwidth requirements also increase. Otherwise increasing the gain of the model at the same sampling interval results in ringing and ultimately instability. This is illustrated by the root locus plot of the discrete time model given in Figure 4.12.

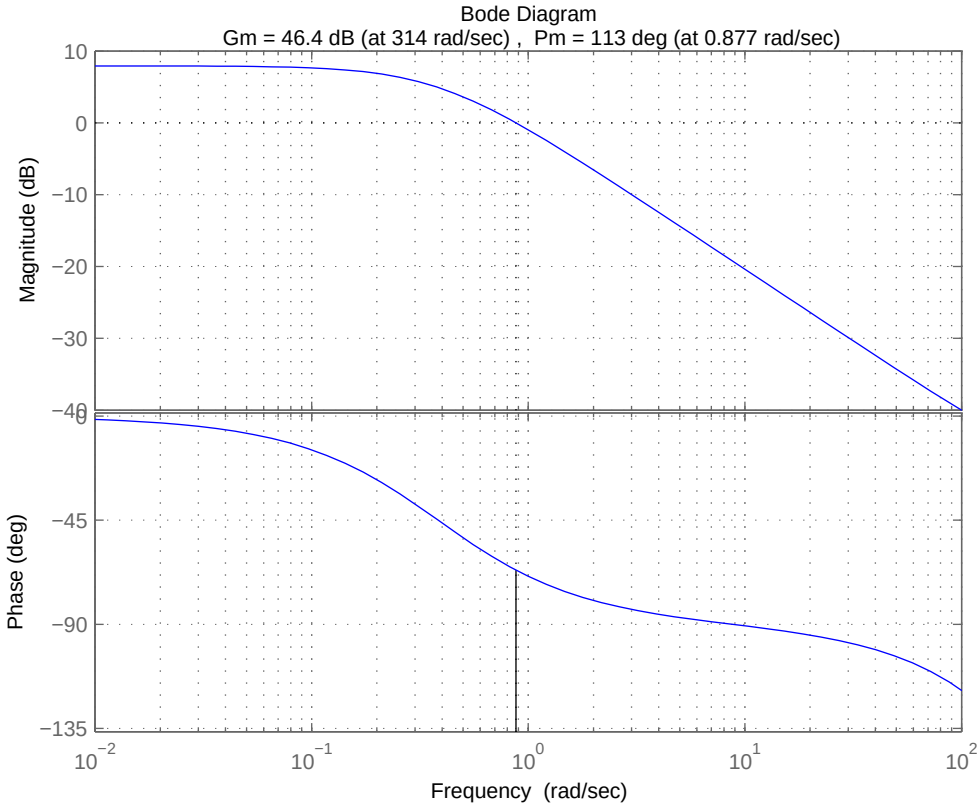


Figure 4.10: Bode plot of the discrete model of the steering system basic block, showing Gain and Phase margins Gm and Pm respectively at the top of the plot

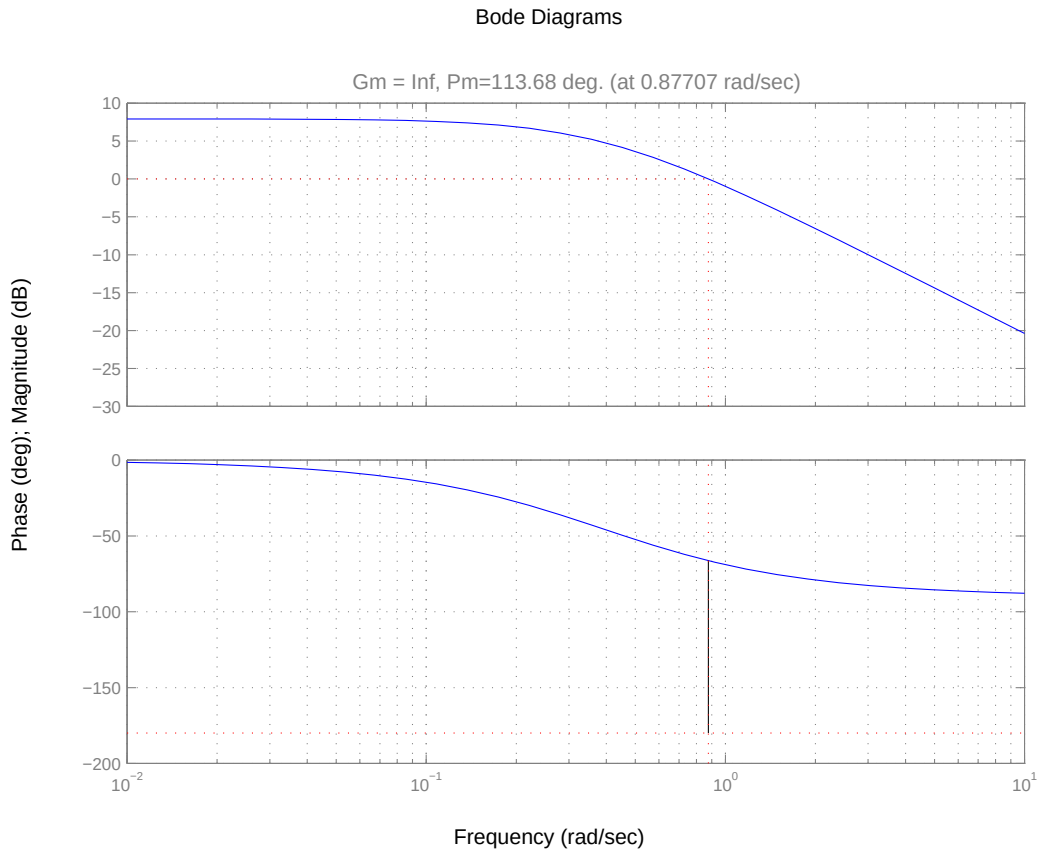


Figure 4.11: Bode plots of the continuous time model of the steering system basic block, showing Gain and Phase margins Gm and Pm respectively at the top of the plot

The cascaded (lower triangular) steering control system for one steering direction is however not diagonally dominant as shown in the Direct Nyquist array with Gershgorin circles in Figure 4.13. The Gershgorin circles encircle the origin of the diagonal element, showing a strong contribution by the off-diagonal transfer function [12, 35]. It should be noted however that the transfer functions used are approximates. These approximates are arrived at by the fact that, the distance between each dipole magnet and its corresponding sensor are approximately the same, and the distance from the first set of dipole magnets to the last sensor is about twice the distances mentioned earlier for the individual pairs, see 4.5. Therefore, while all the time constants are approximately the same, the gain of the off-diagonal transfer function is about twice that of the diagonal terms, and this is easily shown geometrically. The approximated transfer function matrix is,

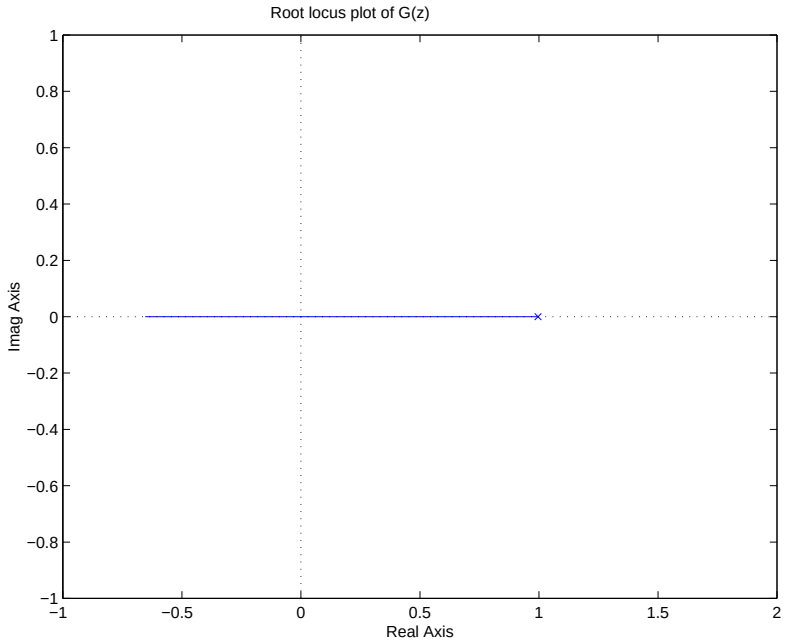


Figure 4.12: Root locus of the discrete time model showing a ringing pole due to low sampling rate compared to closed loop bandwidth or root-locus gain. This is shown by the pole drifting into the left half of the z-plane

$$G_{tr} = \begin{bmatrix} g_{11} & 0 \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \frac{2.49}{2.6s+1} & 0 \\ \frac{5}{2.6s+1} & \frac{2.49}{2.6s+1} \end{bmatrix} \tag{4.4-1}$$

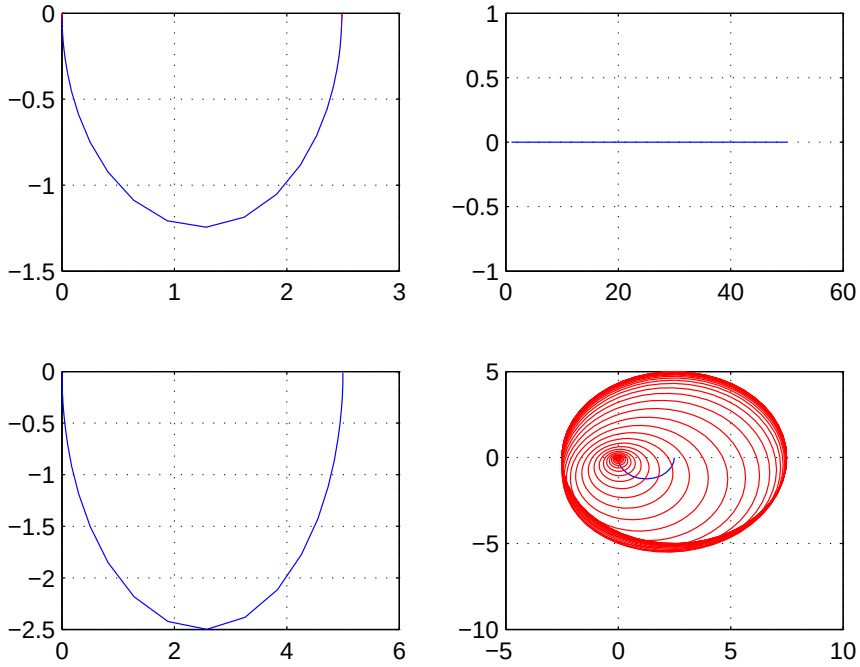
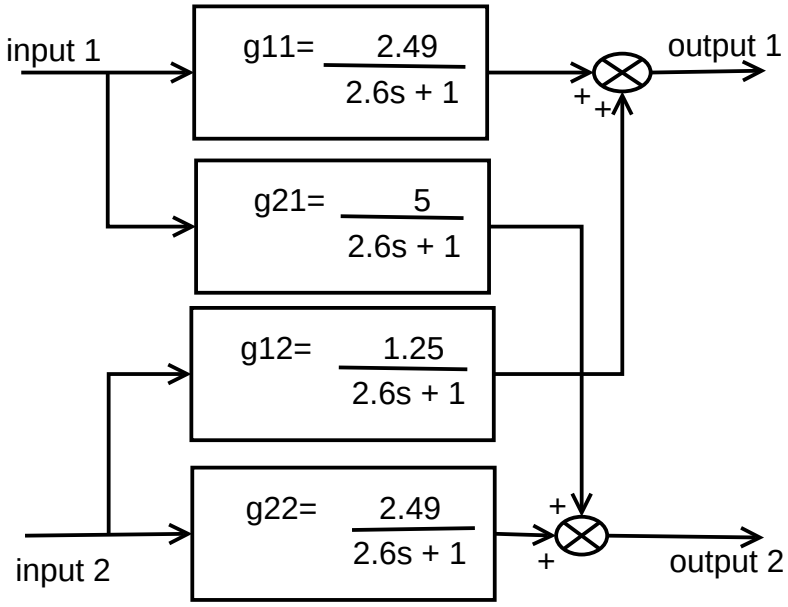


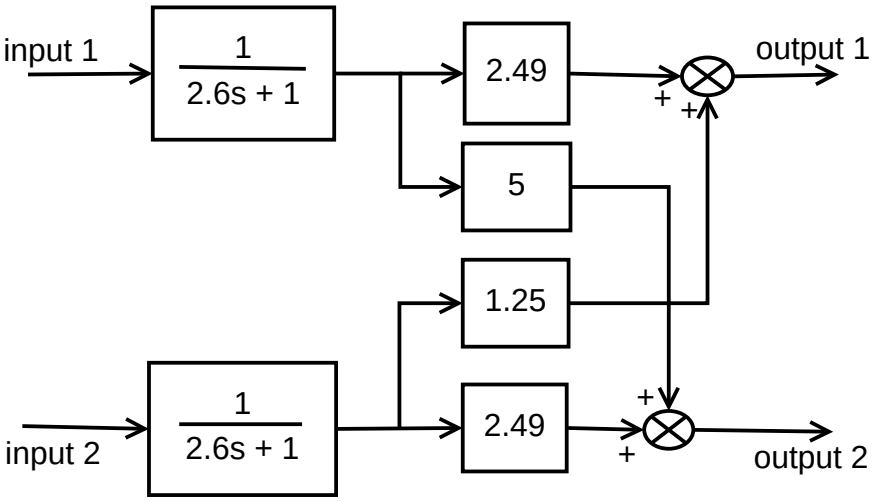
Figure 4.13: Direct Nyquist Array and Gershgorin Circles showing elimination of g_{12}. However Row Dominance still fails due to g_{21}, as shown by Gershgorin circles encircling the origin of the Nyquist plot. This is the case with Column dominance which is not shown here

The condition shown in Figure 4.13 shows that something has to be done to improve the diagonal dominance of the cascaded steering control system.

It can be concluded that the *full* steering system and *lower triangular* structures are stable. This is seen because the x and y steering subsystem have only two poles each at $s = -0.38$. This is illustrated in the block diagram in Figure 4.14 [12].



(a)



(b)

Figure 4.14: This figure shows the system block diagram of the steering system in one steering direction in (a) and how it is rearranged to extract the poles of the MIMO system in (b).

Chapter 5

Proton Beam Steering Control Synthesis and Design

5.1 Introduction

As it has been mentioned in the preceding chapter, the entire proton beam steering control system can be treated as a combination of four decoupled SISO steering tasks represented by the steering system building block. The four SISO control tasks are for the X and Y steering of the first set of steering magnets (*Steerer1*) and the X and Y of the second steering magnet set (*Steerer2*). The transfer functions of the control system are of the same form, and control design continues for one model and is implemented as is, for other models or customized to control the slightly different model. It should be noted that the models can only differ considerably in their DC gains.

5.2 Unconstrained Proportional + Integral Control

Proportional-only control can not eliminate steady state error in the control loop, therefore, a single integrator is added to ensure zero steady state error, and make the controller a PI type. The PI controller is tuned by Affine parameterization as follows [22].

5.2.1 PI Tuning

First the steering system model is inverted to give,

$$G^{-1} = \frac{2.6s + 1}{2.49} \tag{5.2-1}$$

it is then multiplied by a function F to make it proper, and called Q. It turns out that the selected F should be the desired closed loop transfer function hence,

$$Q = F * G^{-1} = \frac{2.6s + 1}{2.49} \times \frac{1}{0.025s + 1} \tag{5.2- 2}$$

This Q is then substituted in the inversion equation for the closed loop, to yield the PI controller C,

$$C = \frac{Q}{1 - GQ} \tag{5.2-3}$$

$$\therefore C = \frac{41.77(2.6s + 1)}{2.6s} \tag{5.2-4}$$

A non-dominant pole at $s = -24.44$ is then added to this controller in order to prevent direct feedthrough of the controller input to the control action [26].

The equation for the PI controller used is given below.

$$C = \frac{41.77(2.6s + 1)}{2.6s(s + 24.44)} \tag{5.2-5}$$

5.2.2 Internal Stability Check

The requirement of internal stability is met when all the transfer functions within the closed loop are stable. In the case of the full structure of the steering system in one direction, there are six transfer functions occurring within the closed loops. They are; the sensitivity of the first loop S_1, its noise transfer function R_1 and its complimentary sensitivity T_1 and the same goes for the second loop S_2, R_2, T_2.

An alternative way for checking internal stability for Affine tuned controllers is to check if Q is stable and proper. If the condition is met then the closed loop is internally stable [22]. For the steering system the condition is met as seen in (5.2- 2).

The plot in Figure 5.1 below shows the closed loop response of the steering system with PI control and unlimited control action. It can be seen that the output tracks the set-point asymptotically. Furthermore, the closed loop is sufficiently fast. A simulation with the digitized PI controller shows similar results at a sampling interval of 0.005 seconds. The results show a 100% settling time of 0.25 seconds.

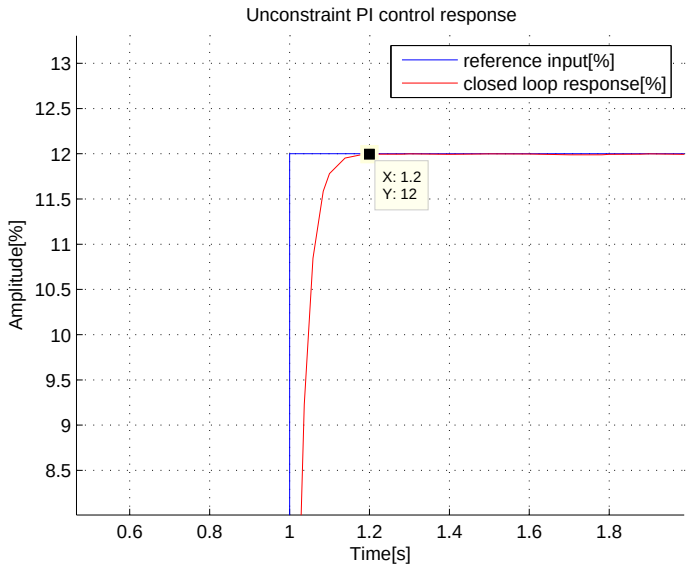


Figure 5.1: Continuous time closed loop step response of steering control system's basic block under unconstrained PI control. The output settles in 0.25s (100%)

5.3 Unconstrained Model Predictive Control

An unconstrained controller for the steering system is synthesized in MATLAB. The controller in (5.3-1) below is the output of the synthesis at 0.01 second sampling intervals.

$$unconstrained\_mpcController = \frac{-0.0935z^3 + 0.09322z^2 + 4.835 \times 10^{-18}z + 4.833 \times 10^{-18}}{z^3 - 2.46z^2 + 2.07z - 0.507} \tag{5.3-1}$$

Simulations results of the controller are given in Figures 5.2 and 5.3 below are the simulations for unit step and step of 12% relative beam position.

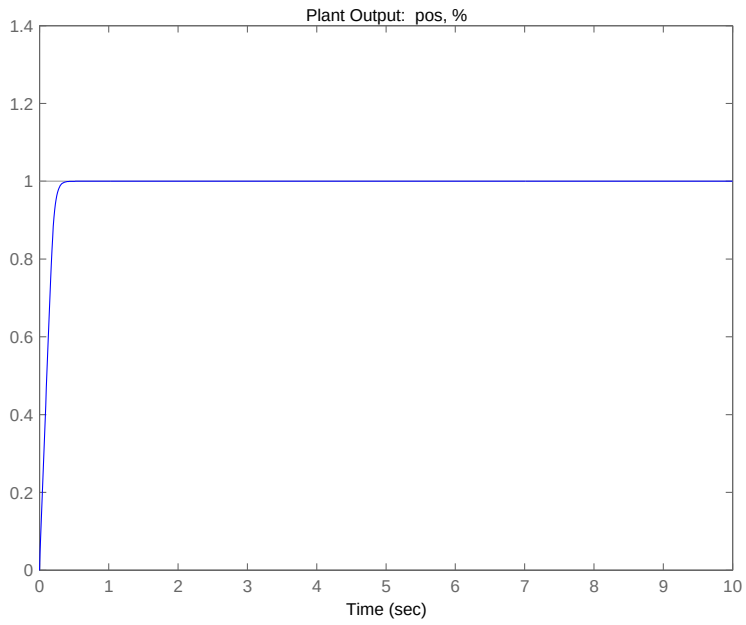


Figure 5.2: Unit step response of steering system's basic block under unconstrained MPC. The response settles in 0.25s (100%)

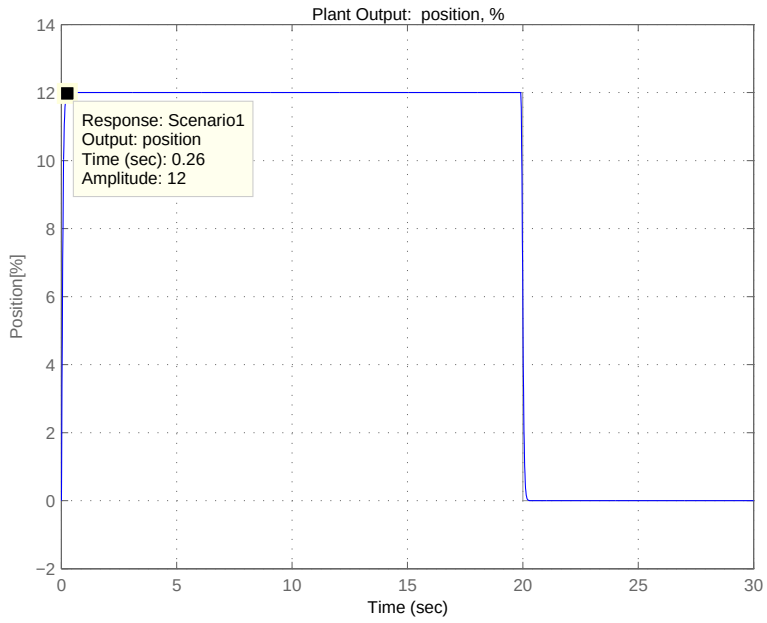
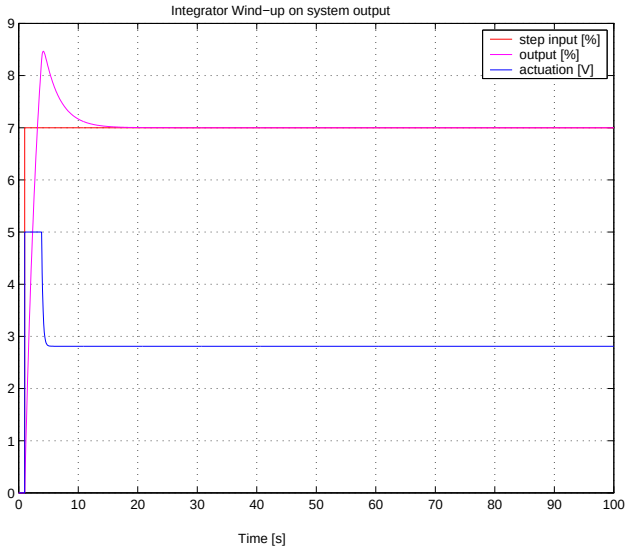


Figure 5.3: The plot shows that, Model Predictive Control of the unconstrained steering system gives a 100% settling time of 0.26s

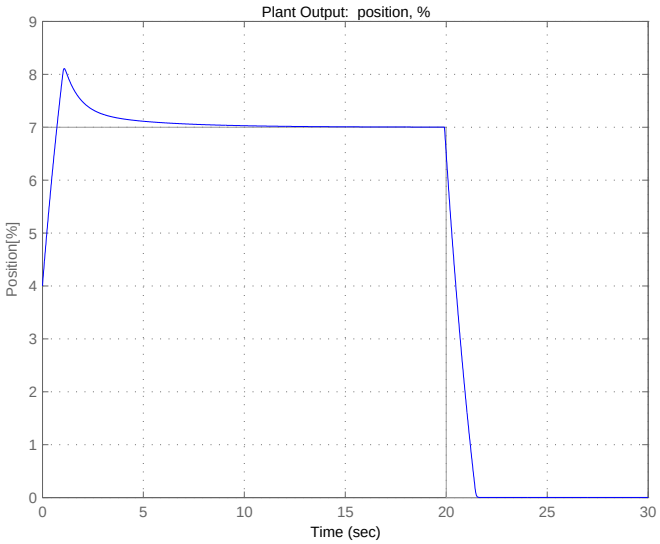
The responses above show satisfactory speed of 0.025 seconds and zero steady state error.

5.4 Power Supply (Actuator) Saturation

Imposing power supply saturation on the closed loops gives rise to the windup problem. This is experienced both in PI control and unconstrained MPC as shown in the plots in Figure 5.4.



(a) PI windup problem



(b) Windup in unconstrained MPC

Figure 5.4: Windup problem in PI control of steering system basic block and unconstrained MPC. Overshoot is seen at the output of the closed loop under these two control methods. Anti-windup techniques will be investigated to deal with this behavior.

5.5 Disturbance Sensitivity Analysis

The response of the closed loop to unexpected disturbances that occur at low frequencies is investigated in this section by simulation to make sure the MPC, rearranged PI and EG controllers designed handle disturbances in an acceptable manner.

5.5.1 PI Control:

Analysis of the PI controller shows that the output disturbance sensitivity of the closed loop is given by,

$$\frac{0.025s}{0.025s+1} \tag{5.5-1}$$

which rapidly approaches zero as frequency decreases. This is a good disturbance rejection behavior. The plot showing the disturbance response for a step output disturbance of 12% is given in Figure 5.5.

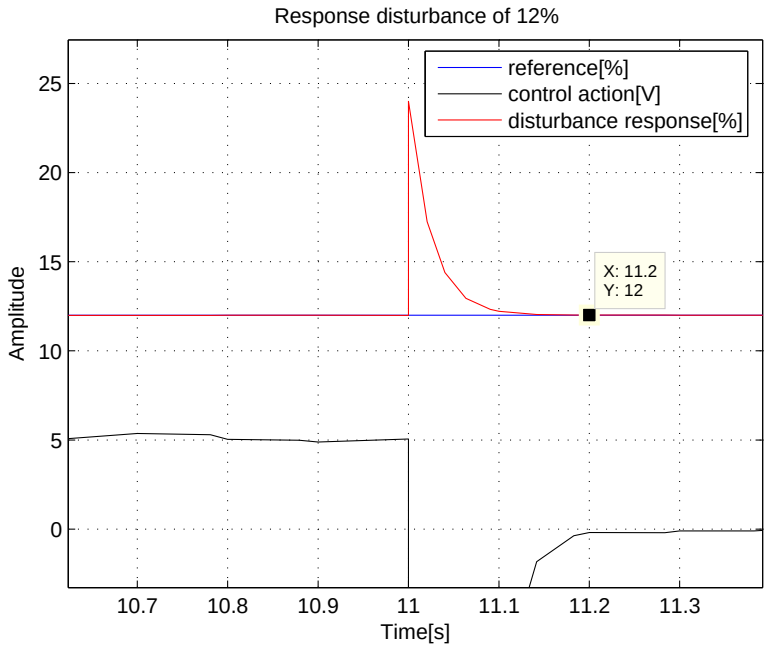


Figure 5.5: The figure shows the PI response to a 12% step output disturbance. The effects of disturbances on the output response die off completely in 0.25s

5.5.2 Model Predictive Control:

Simulations performed with the synthesized MPC controller are done in the presence of disturbances and the plot shown in Figure 5.6 shows the response for step output disturbances of magnitude 12%.

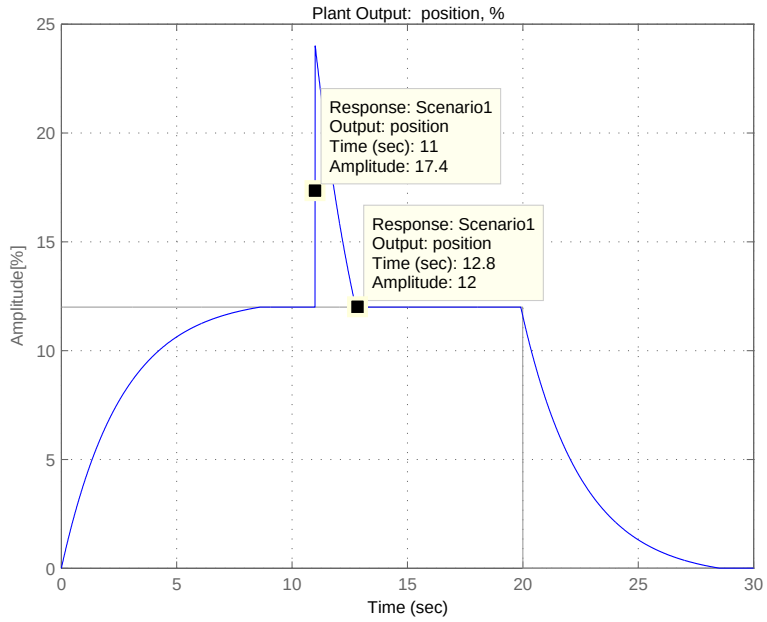


Figure 5.6: MPC response with output disturbance is shown in the plot and the response due to disturbances dies off in 1.8s

5.6 Analysis of Sensitivity to Parameter Changes

Due to the inaccuracies in the plant model, the closed loop can go unstable due to changes in either the DC gain or the time constant of the plant. The discussion in this section is dedicated to the analysis of the closed loop sensitivity to plant parameter changes.

5.6.1 DC Gain Variation

PI control

The variation in the DC gain of the plant causes a change in output operating range of the steering system. Apart from the shrinking or expanding of the output operating region, the closed loop becomes fast or slow with decreasing or increasing DC gain respectively. Moreover, these changes have a direct impact on the closed loop bandwidth.

For changes that speed-up the closed loop, the sampling rate has to be large enough to avoid instability in the discrete implementation. The treatment below illustrates the effect of the DC gain variation on closed loop poles.

The operating point closed loop pole is at $s_p = -40.002.$ and the operating point DC gain is $G_{DC} = 2.49$.

Taking the partial derivative of the closed loop characteristic equation Φ_{cl} with respect to the DC gain G_{dc} gives:

$$\frac{\partial \Phi_{cl}}{\partial G_{dc}} = \frac{\partial}{\partial G_{dc}}(2.6s + 41.77G_{dc}) = 2.6\frac{\partial s}{\partial G_{dc}} + 41.77 = 0 \tag{5.6-1}$$

$$\Rightarrow \frac{\partial s}{\partial G_{dc}} = \frac{-41.77}{2.6} = -16.065 \tag{5.6-2}$$

$\therefore \pm$ unit change in $G_{dc} \Rightarrow$ pole change of ∓ 16.065

Thus allowing a variation of the DC gain by ± 1 unit will result in new pole positions at $s = -56.067$ or $s = -23.937$ respectively. It can be seen here that for a negative change in the DC gain of more than 2.49 units, the closed loop will become unstable. Otherwise, only the speed of the closed loop is affected.

MPC

Simulations performed using the synthesized MPC controller with plant models with

different DC gains shows that the closed loop speed is the one that is affected by the parameter change as shown by the plot in Figure 5.7 below. It should be noted that a gain below zero gives rise to instability. The arguments given for Anti-Windup also apply here.

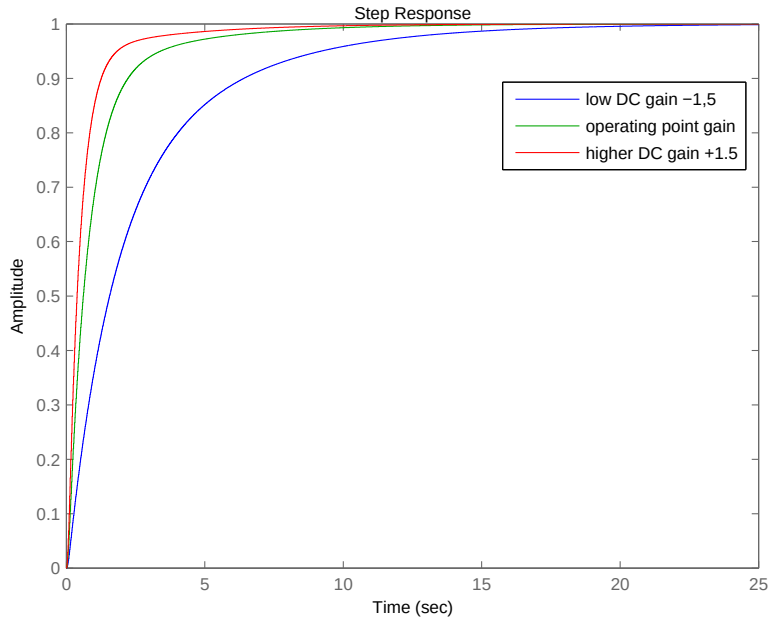


Figure 5.7: Variation of DC gain in plant model on MPC closed loop shows a change in closed loop speed. This behavior is acceptable and has no dangerous effects

5.6.2 Time Constant Variation

The change in the time constant also has an impact on the closed loop speed. This effect propagates to impact the sampling rate. Moreover, severe changes may cause overshoots and instability. The time constant of the steering system is not expected to change much as it is dominated by the power supply response as obtained experimentally [50]. This is the case both in PI and Model Predictive Control. The plots in Figure 5.8 show the effect of varying the plant time constant on MPC. Similar results are obtained for PI control, except the response under PI controller does not develop overshoots with changes in the time-constant. It is noted however, that the time constant can not be zero or below.

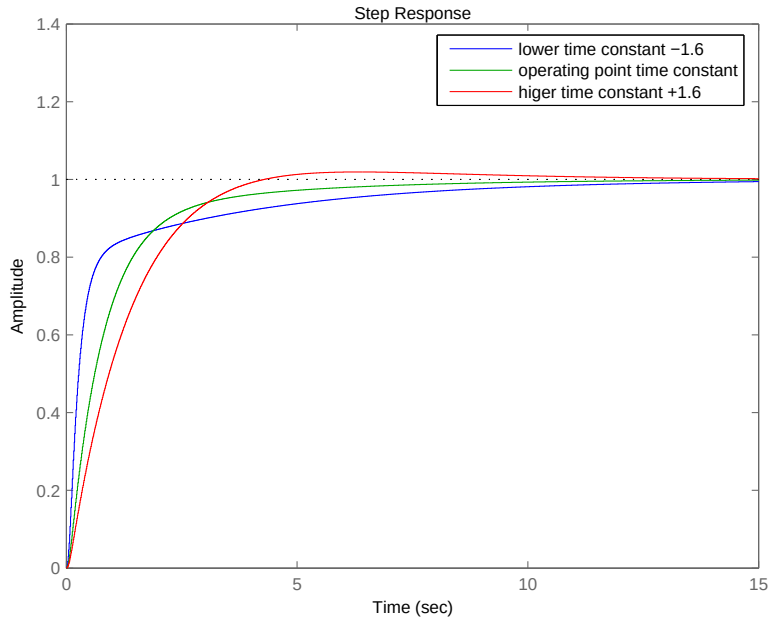


Figure 5.8: MPC closed loop response with variations in plant time constant shows change in closed loop speed and possible overshoots. This behavior is not good since it makes the proton beam bounce on the walls of the containment, and produce unwanted energy modes

Chapter 6

Proton Beam Steering Windup Simulations

6.1 Introduction

The PI, and MPC controllers designed in the previous chapter are used here with anti-windup modification in order to observe the input constraints. These modified controllers are simulated in MATLAB to compare their performance.

6.2 Anti-Windup Investigation

6.2.1 Simulations

Different control design approaches can be used to come up with the controller to tune the closed loop. In this section the continuous time PI controller was chosen for simulation purposes because it offered relatively equivalent performance to the discrete time pole

placement controller.

PI control design Continuous time analysis of the plant is done and it is seen that the gain margin of the plant is infinity and the phase margin is 113.7^o. For the control of this system Proportional-only control is dismissed due to the fact that, it requires a very high gain to overcome steady state error [53] which in turn requires a high sampling rates.

A Proportional plus Integral (PI) controller is used to ensure asymptotic set-point tracking. The Proportional gain for the controller is tuned to be 41.77 (see Chapter 5), which requires a sampling frequency about 200Hz according to the Nyquist sampling law [36]. The controller and the plant are then converted to their discrete time equivalent models at the same sampling rate (0.005s as stated earlier) and applied in a unity feedback configuration. The PI controller is given in Equation (6.2-1). The Simulink model used to simulate the closed loop under PI control uses the configurations of PI shown in Figure A.1 in the Appendix for windup and anti-windup behavior.

$$PI = \frac{41.77(2.6s + 1)}{2.6s(s + 24.44)} \tag{6.2-1}$$

The discrete form of this controller is obtained by directly digitizing the components of its rearrangement in the block diagram. It is worth noting that the non dominant pole $s = -24.44$ is eliminated to make the controller inverse proper.

The plot in Figure 6.1, below shows the step response of the steering system with wind-up behavior. Overshoot behavior is evident on the system output and if this is the output of Steerer1, this overshoot will go through as the reference to Steerer2 causing undesired behavior.

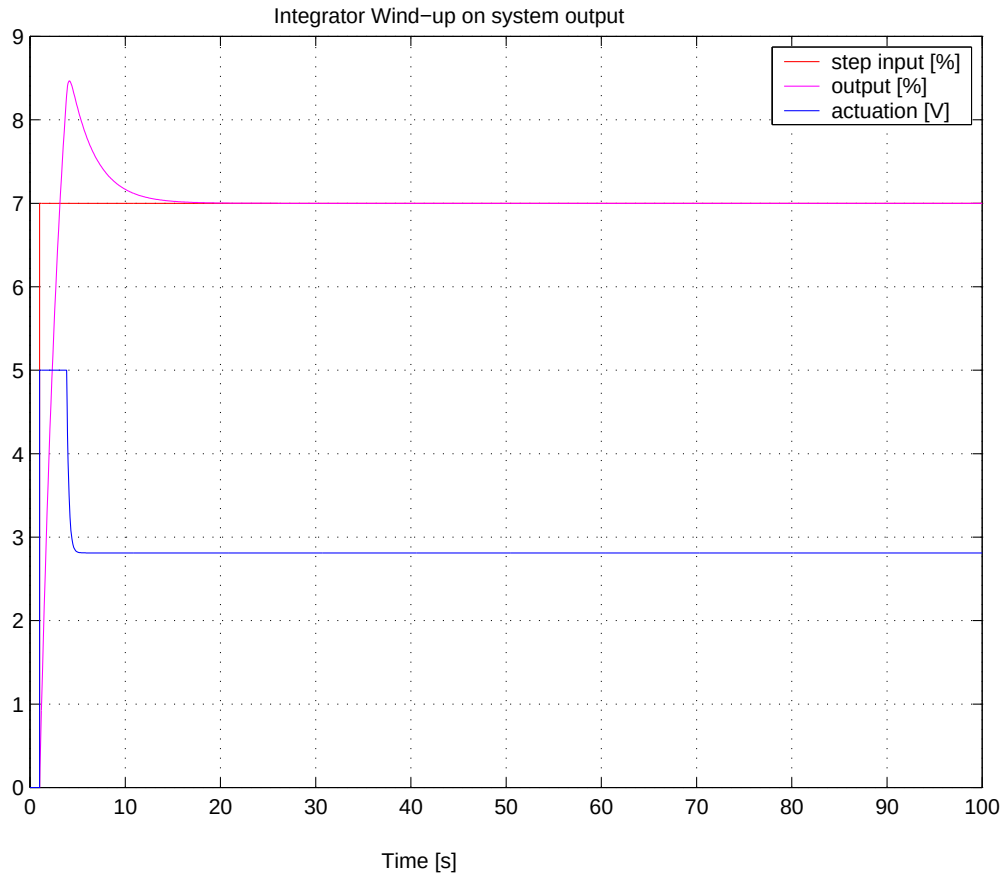
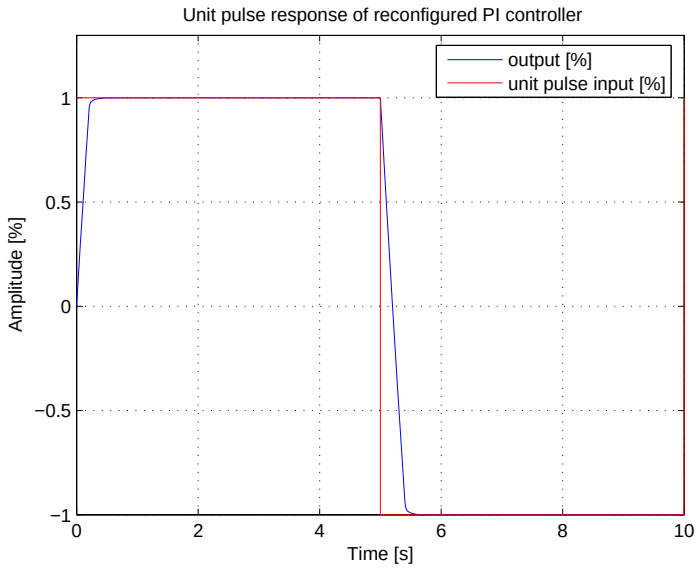


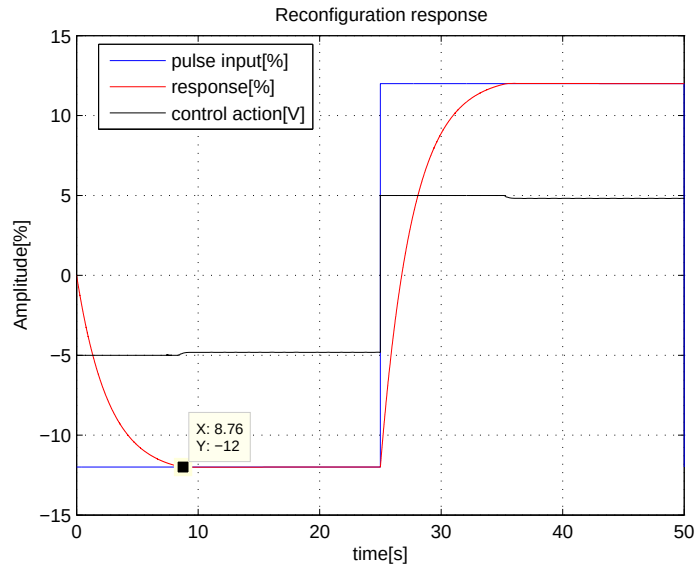
Figure 6.1: Step response of basic block with windup: the output shows an overshoot

The configuration in the Simulink block diagram in Figure A.2 in the Appendix is used to simulate the response of the system again.

The results shown in Figure 6.2 below show that the anti-windup problem has been solved, and the system no longer has overshoots. With this configuration the closed loop can be made sufficiently fast without any windup problem.



(a) Unit pulse input response



(b) 12% step pulse response

Figure 6.2: Closed loop response with a controller reconfigured reconfigured for anti-windup with unit pulse input and maximum allowed pulse input of 12%

It is noted from simulation that closed loop 100% settling time of about 0.25 seconds is achievable for small non-saturating inputs. However, with larger input steps, the response deteriorates to settling times around 8.75 seconds.

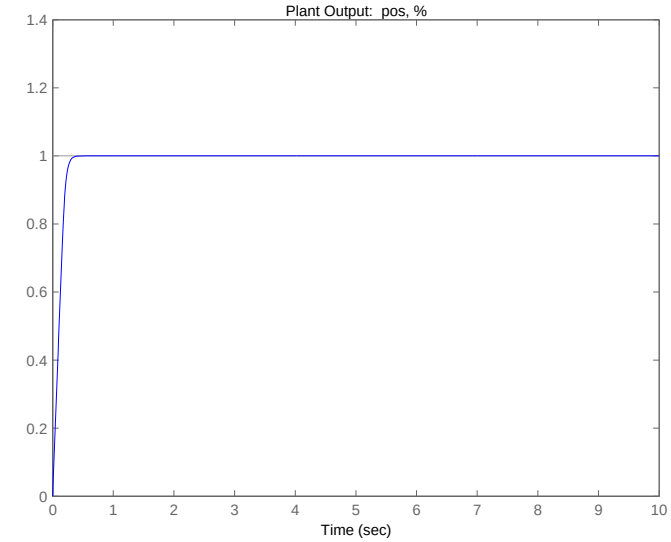
6.3 Model Predictive Control Investigation

6.3.1 Simulations

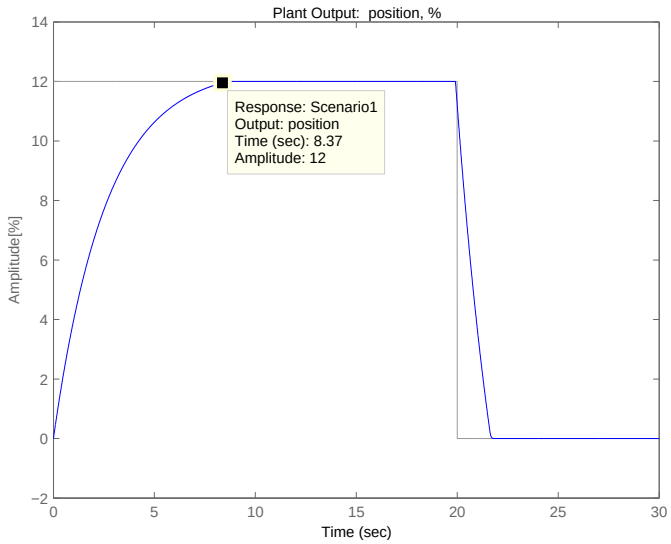
The simulations performed in this section were done using MATLAB. The steering system plant model was used to synthesize an MPC controller that will handle the input constraints as well as ensuring setpoint tracking. The controller given in equation (6.3-1)below was extracted at 0.01s sampling rate.

$$mpcController = \frac{-0.41z^3 + 0.4088z^2 + 6.651e-017z + 8.356e-019}{z^3 - 1.762z^2 + 0.8373z - 0.07486} \tag{6.3-1}$$

This controller is simulated on the steering system to give the response given in the plots in Figure 6.3 below.



(a) Response to unit step reference change



(b) Response to 12% step reference change

Figure 6.3: MPC control of the steering system's basic block with constraints enforced: the output has no overshoot.

The simulation results above show similar results to those obtained by the anti-windup method in both speed of response and remedy to the hang-up problem caused by the input constraints. However the resulting MPC controller is more complex than the PI controller.

6.4 Error Governor Investigation

Problem formulation: The controller used for this simulation is given by the state-space model,

$$\dot{\mathbf{x}}(\mathbf{t}) = \begin{bmatrix} 0 & 19.85 \\ 0 & -24.44 \end{bmatrix} \mathbf{x}(\mathbf{t}) + \begin{bmatrix} 0 \\ 10.83 \end{bmatrix} \mathbf{u}(\mathbf{t}) \tag{6.4-1}$$

$$y = \begin{bmatrix} 0.2099 & 10.83 \end{bmatrix} \mathbf{x}(\mathbf{t}) \tag{6.4-2}$$

and is naturally stable with $\lambda_1(A) = 0$ having a distinct eigenvector and $\lambda_2(A) = -24.44$.

$$g(x(t)) = \| \, c^T e^{At} x_0(t) \, \|_\infty \tag{6.4-3}$$

and the constraint set χ is,

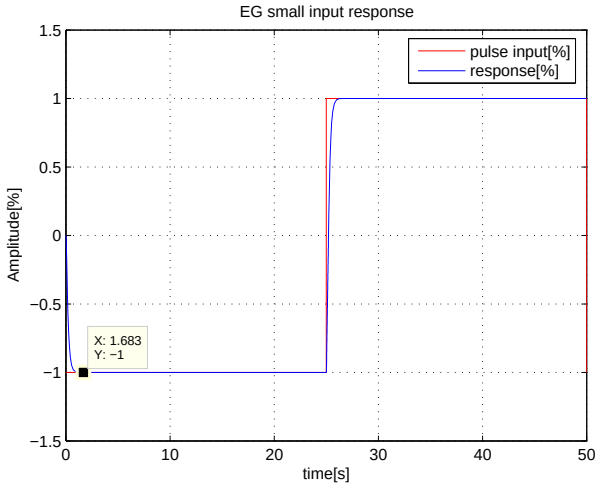
$$\chi = \{x(t) \in \mathbb{R}^2 | -0.4617 < x_2(t) < 0.4617; x_1(t) = 0; g(x(t)) \leq |5|\} \tag{6.4-4}$$

A slight mathematical approach difference was used to get the state vector from the controller output using Moore-Penrose pseudo-inversion [23] as:

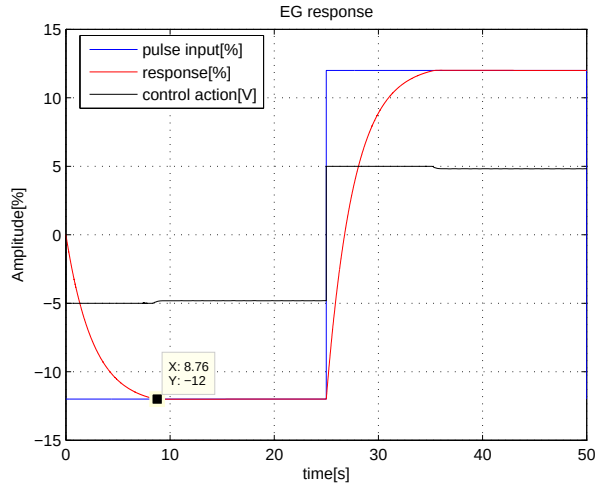
$$x = C^{\#} y = [C^T C]^{-1} C^T y \tag{6.4-5}$$

6.4.1 Simulations

The EG simulations are done with the steering system model and the obtained results are published in Figure 6.4 below.



(a) non-saturating input response



(b) saturating input response

Figure 6.4: Closed loop response of steering system's basic block with EG anti-windup compensator shows good results: the output does not overshoot

The results obtained in Figure 6.4 are similar to the previously obtained results from the other two approaches for large inputs, and they deteriorate for small inputs.

The block diagram given in Figure A.3 in the Appendix was used for these simulations. The MATLAB function used to implement the EG logic is given in Appendix A.1.

6.5 Cascaded Steering System in One Steering Direction

In order to test the designed steering control system for both centering of the beam and keeping it parallel to the beamline, a cascaded form of the control system in 1-dimension is used. From the simulation of this system the centering ability of the system is determined from the ability of the cascaded steering outputs to follow the set reference value. The parallelism of the beam is achieved whenever the outputs of the cascaded steering stages are the same.

6.5.1 PI Control and Reconfiguration

A SIMULINK model used for simulating this cascaded behavior is shown in Figure A.5 in the Appendix. It is worth noting the presence of the coupling from the control input of the first steering stage to the output of the second steering stage.

A first analysis of the cascaded model aims at diagonalizing the transfer function matrix (4.2-3) of the cascaded steering system. This is done by eliminating the off-diagonal term as follows [12, 35].

$$G_{tr} * K = \begin{bmatrix} g_{11} & 0 \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ k_{21} & 1 \end{bmatrix} = \begin{bmatrix} g_{11} & 0 \\ g_{21} + g_{22}k_{21} & g_{22} \end{bmatrix} \tag{6.5-1}$$

Eliminating the off-diagonal therefore means that,

$$g_{21} + g_{22}k_{21} = 0 \tag{6.5-2}$$

$$\therefore k_{21} = \frac{-g_{21}}{g_{22}} \tag{6.5-3}$$

However, the results obtained from this simulation are very bad as shown in the plot in Figure 6.5.

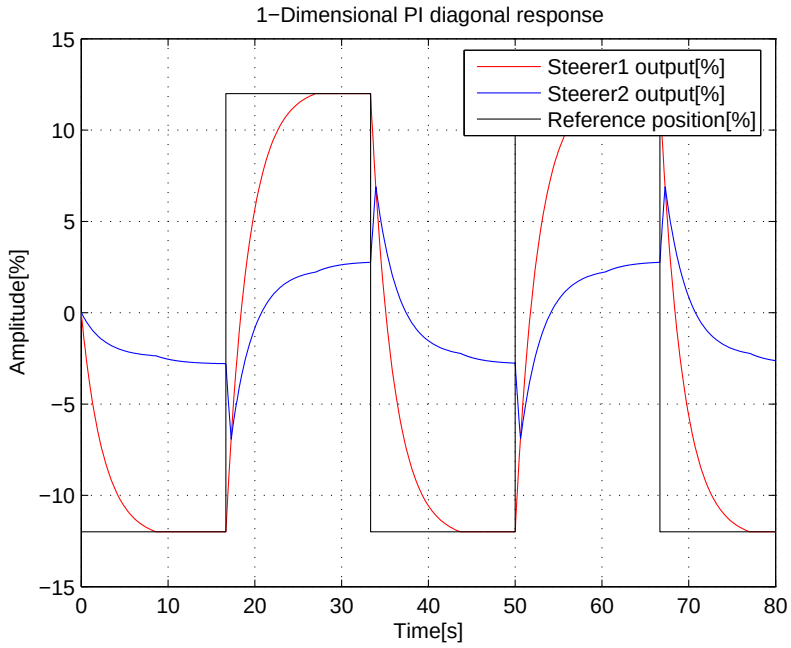


Figure 6.5: Application of diagonalization and the designed PI controller gives undesirable response for the cascaded steering system in one steering direction.

Simulations are then done without the diagonalization and the results obtained from the SISO control of the individual steering stages are shown in Figure 6.6. These results show accurate centering and parallelism of the beam. It should be noted that the broad type of plot used for the output of Steerer2 is to discriminate the response of the second steering stage, and not because the output in question is noisy.

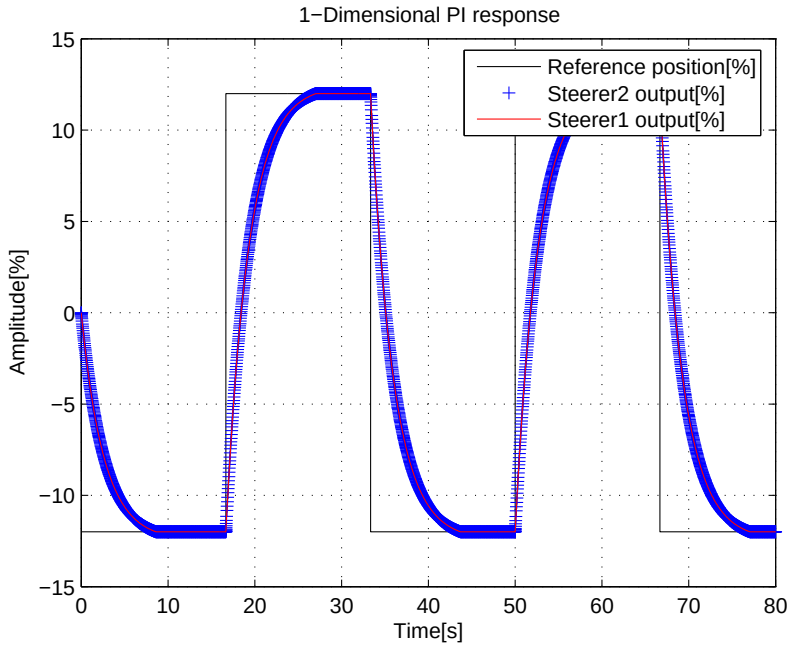


Figure 6.6: Response of SISO control of cascaded Steering system without diagonalization. This gives good results, in the sense that the both beam centeredness and parallelism are achieved. Note that the markers of Steerer2 are overlaid on the curve for Steerer1

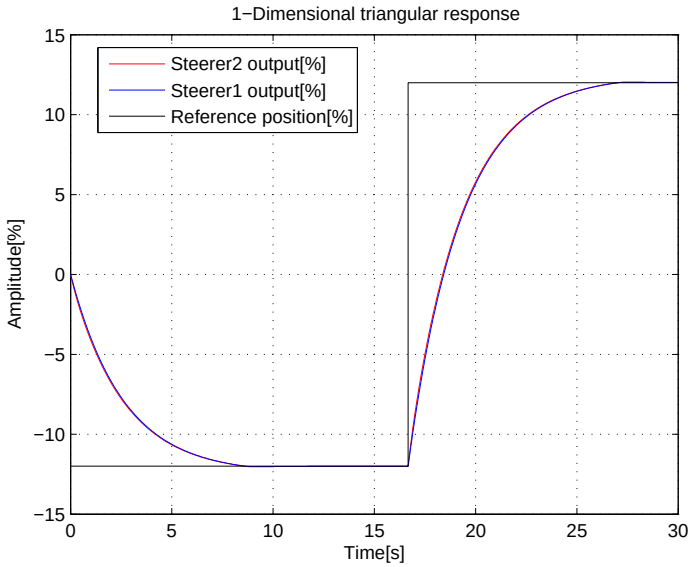
.

6.5.2 PI Control with EG Anti-Windup compensator

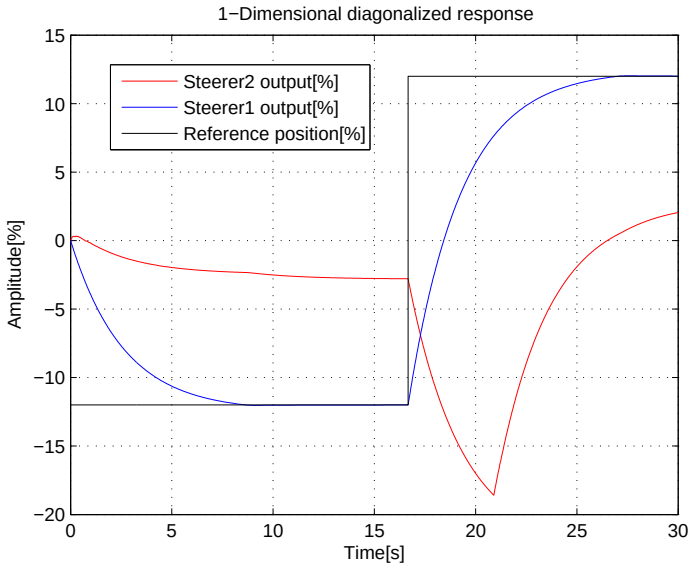
Simulations done with the EG anti-windup compensator showed similar results with those observed when using the reconfiguration of the PI controller. These results are published in the plots in Figure 6.7

6.5.3 Model Predictive Control

Simulations done with MPC showed poor results both with diagonalization and without diagonalization. This is shown in the loss of beam parallelism during the transient response of the system, when the system is not diagonalized. This problem becomes worse for small reference changes. Furthermore, the response of the system fails to give zero steady state error and hence parallelism when diagonalization is used for large reference inputs. However, these conditions are satisfied at steady state, when the reference is small but not during the transient response. The plots in Figure 6.8 show the simulation results obtained with MPC.

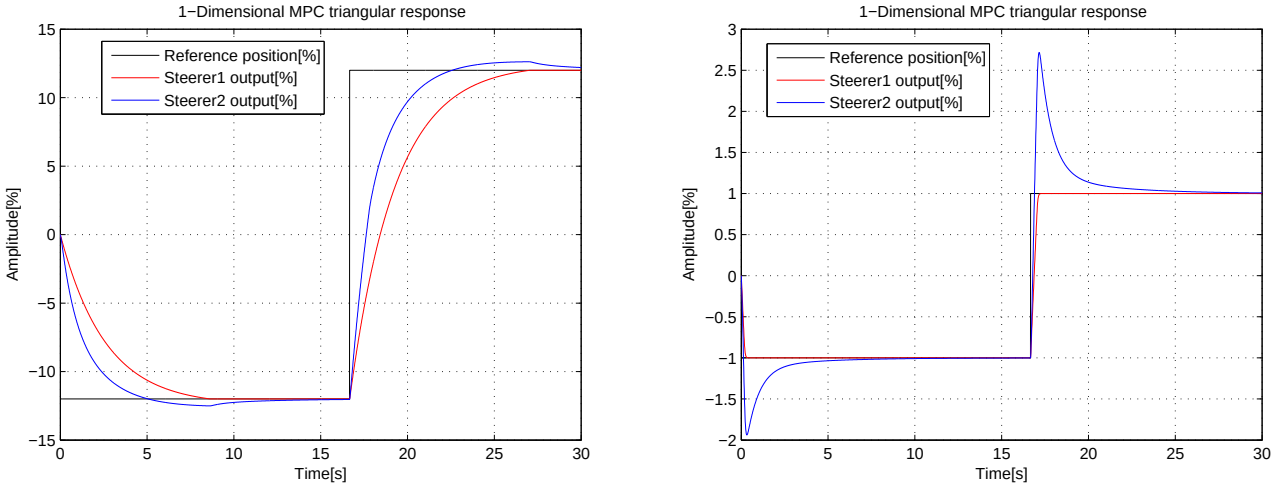


(a) Response of the lower triangular cascaded steering system for one steering direction. This shows good parallelism as shown by similar beam position as measured by both beam position sensors, and good beam positioning showed by the outputs of the steerers following the reference asymptotically

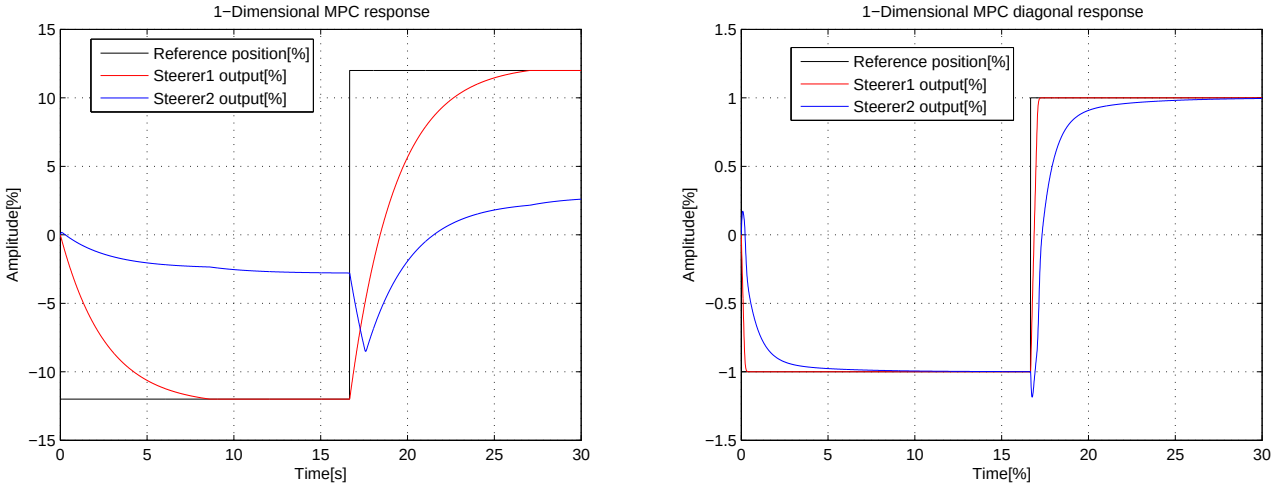


(b) The response shows the response of the diagonalized cascaded steering system in one steering direction. It has undesired response

Figure 6.7: Response of the Steering system with and without diagonalization in one steering direction. The results are good for a lower-triangular structure in (a). However, they become bad when the structure is diagonalized, showing steady state error.



(c) Response of undiagonalized lower triangular steering system under MPC



(f) Response of diagonalized steering system under MPC

Figure 6.8: Response plots of cascaded steering system response under MPC in one steering direction. The results are undesirable; showing errors in beam parallelism during transient state (outputs do not coincide), steady state error and overshoots are observed on the output of Steerer 2.

Chapter 7

Control Implementation

7.1 Implementation Platform

The choice of implementation technology for control systems is dictated by several factors, some of which are; performance requirements, physical space availability, budgetary limitations and even skills availability. However, the budget is the one that dictates most decisions made on the choice of implementation technology [33]

Nowadays, personal computers offer high speeds and are available at low costs. That therefore makes them the best choice of implementing reliable inexpensive control systems, which are able to support hardware and software products from a large number of vendors [46], [30].

Furthermore, the implementation of controllers on computers allows updating of the controller software without having to suspend the operation of industrial plants if the software is designed using good practices.

7.2 Operating System Choice

Many of the large experimental physics systems are mission-critical in nature and often require real-time control. The choice of operating systems to be used on the computers implementing the control system is thus influenced by real-time requirements. Apart from real-time operation, the choice of operating system is also influenced by financial aspects, in which case free open source operating systems are favorable. Other factors that are determined by the control system architecture, such as middleware in distributed systems, also play a role in the choice of operating systems. It would also be considered that the UNIX/LINUX family of operating systems are open source. This allows the recompiling of their kernel to remove the unnecessary modules, and in doing so reduce initialization times and load on the processor. A survey of a number of operating systems' operation in realtime control was conducted in [19] for the implementation of control systems and real-time operation. Based on the metrics mentioned here, RTLinux seems to have all the benefits discussed and is hence the operating system of choice.

7.3 Programming Languages

The choice of programming language depends on a number of factors that are determined by the standards of a company, the type of application to be developed, and the processing speed required. For control applications, it is often necessary to deliver real-rime operation. This therefore, requires a programming language that can perform algorithms and mathematical computations fast. Furthermore, some of the software modules are implemented to control the hardware for measurements and control purposes. Due to these requirements and others, C is the *de facto* language for the task [14]. GNU tools and other third party tools used with the C language can ensure good and safe programming that conforms to standards, and can therefore be regarded as the right choice for linux platforms.

However, some of the control software components are for non-critical aspects of control (an example being display modules). For these components flexibility is allowed and languages like Visual Basic, QT3 and others can be used depending on the platform for which those modules are intended [19].

7.4 Real-Time Systems and Programming

Real-time systems can be grouped into two categories, These are the *hard* real-time systems, in which failure of an event to occur at a specific time may have serious consequences and hence the system has to be stopped for safety or other reasons. On the other hand, *soft* real-time systems can afford to miss deadlines without causing damage or danger within the system. Therefore, in this case the system can continue running. However the samples of data, that were to be used in the missed deadline have to be discarded and not passed onto the next stage [24].

7.5 Software Implementation of Control

Like most other software developed for industrial applications, it is desirable for the software to be written in accepted and easy to understand styles, and be well documented to assist understanding for software developers that may work on it at later stages.

Furthermore, due to the distributed nature of experimental physics control systems, the architecture employed for the software is of major importance. The software should be architected into components that achieve sub-tasks of the control of problem and can be implemented on different computing nodes. Moreover, the components should be designed to be reusable through the object oriented programming paradigm [1]. Well defined interfaces should be defined to handle requests of service from other components

of the system, such as user interfaces and data management systems.

While the main aim of control systems is to implement mathematical algorithms that achieve the control of industrial plants, and much effort should focus on designing optimal algorithms, and the objects and modules and components that achieve the control task, it is also required that well defined interfaces and objects are defined to cater for other components of the control software. Groups of software objects were suggested in [37] that apply to a large number of of industrial control systems, and these are listed below.

1. Graphical User Interface Objects.
2. System Data Display Objects.
3. Data Acquisition (DAQ) Objects.
4. Data Management and Storage Objects.
5. Control Algorithm Implementation Objects.

In addition to the above software objects, some control systems have safety and security issues like interlocks. There need to be objects that check and ensure the safety requirements of the control systems. Moreover, it is time saving to design object oriented software for control purposes, to enable re-use and ease of modifying the software implementation [7].

7.6 Steering System Software

The software modules are implemented by heavy use of objects using the CCLASS methodology [25]. There is no direct access to the functionality and data of this module, and use is made of well defined interface functions.

7.6.1 Modularization

1. Control Module

   This module implements the control algorithm for controlling the individual dipole magnets. It is intended for implementing the most suitable control algorithm for the steering task, however other algorithms are also implemented to provide choice.

2. DAQ Module

   This module implements the input-output functions to the fieldbus used for control and reading the sensors for control purposes. It provides an interface to external programs through which they can create and use DAQ objects.

3. Safety Module

   This module implements the interlock (*Permissive contacts*) checking and manipulating functionality of the steering system for safety and fail safety behavior in the beam steering system.

7.6.2 Execution of the Control Tasks

The flow chart in Figure 7.1 shows the flow of operation tasks in the steering system. It should be noted that lines with the same colour represent a single thread of execution.

Under automatic control the control task follows the normal cycle of scanning the inputs (reference and sensors), then applying the control law, and finally writing the control action to the actuators (PSU). However in manual mode, the software just reads the inputs from the operator and writes them as they are to the PSU.

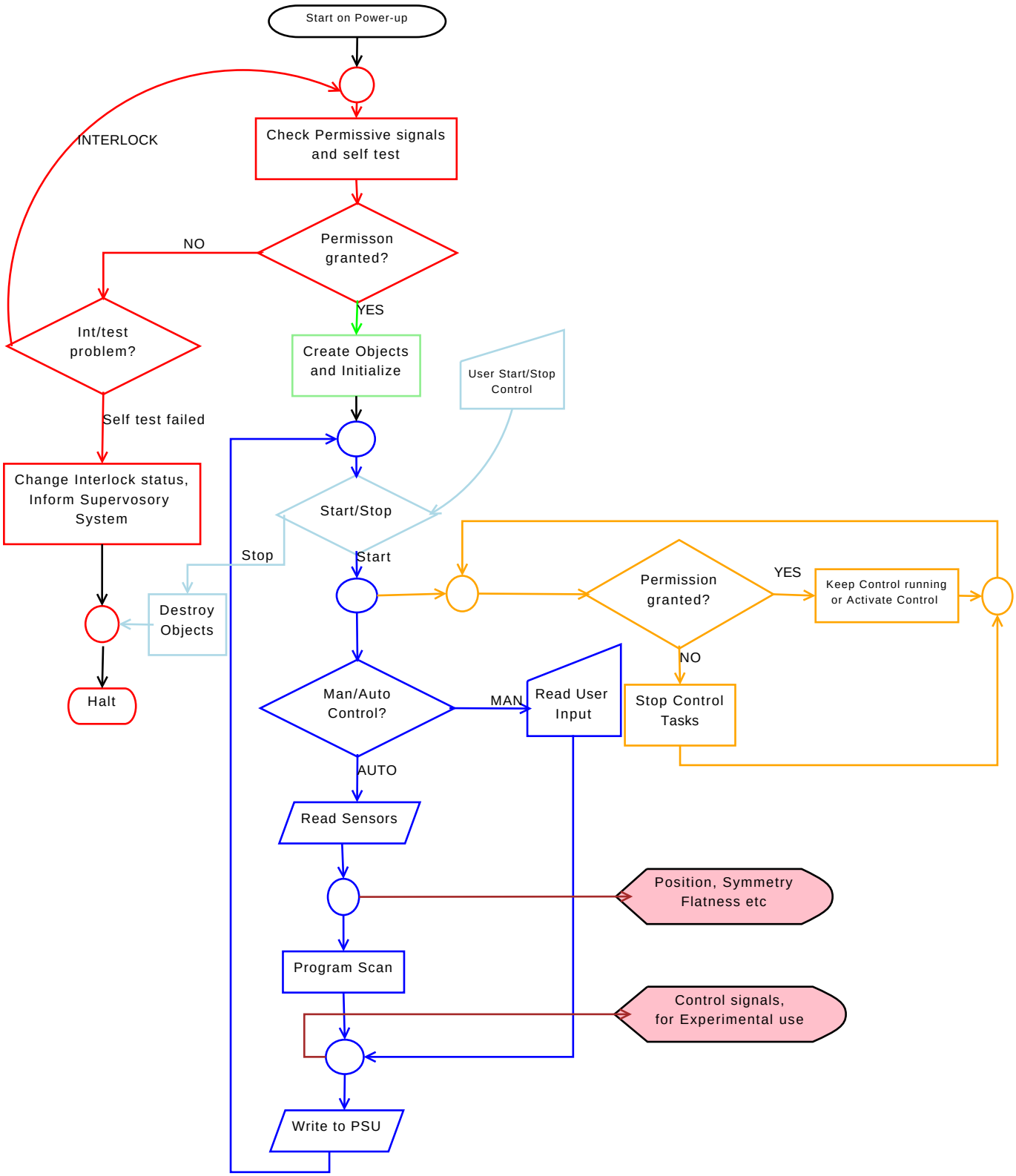


Figure 7.1: Flow of tasks for control

Chapter 8

Experimental Results

Experimental testing of the control system is not performed due to the unavailability of proper infrastructure. The Second Beamline project, to which the Proton Beam Steering Control System belongs has been stopped for administrative reasons. However some parts of the project were requested to continue to the point of experimental testing which will only be done once the proper infrastructure is in place. The work on the Proton Beam Steering System is therefore done up to the point where it awaits experimental testing.

The currently available Proton Therapy infrastructure can not be used for experimentation due to a architectural differences between it and the envisaged second beamline Beam Steering Control System . These differences are discussed in the sections that follow.

8.1 Objectives of the Control System

On the current Proton Beam facility the beam steering done is aimed at keeping the beam centered relative to a measurement sensor (MWIC) placed in the beamline. This is different from the goal of the work done in this thesis. The aim of the Control System

developed here is to keep the proton beam centered relative to the beamline axis, and to keep the beam parallel to the central axis.

8.2 Control Dimensionality

The above mentioned difference in objectives result in different dimensionality and structures of control. This is due to the fact that the goal of the steering system in the Proton therapy facility currently in-place is achieved using one pair of steering dipole magnets and a single beam position sensor, while the objectives of the work in this thesis require two pairs of steering dipole magnets and two beam position sensors. This control task is two-dimensional and this introduces certain MIMO aspects like coupling, results of which can not be demonstrated on the control infrastructure in-place.

8.3 Communications

The communication fieldbus chosen for control tasks in the Second Beamline project is the RS485 differential bus, whose delays are smaller than reasonable control sampling times. On the other hand, on the current infrastructure, control has to be done over Ethernet which has quite significant non-deterministic delays and the risk of communication breakdown. This introduces output delays which are seen as deadtimes in the plant model and these call for a completely different control architecture from that which is used for the Second Beamline Beam Steering Control System.

Given the above challenges, it was decided that experimental tests not be done on the current infrastructure as it provides insufficient sensors and actuators, and can not test the full capabilities of the control system developed in this work. Furthermore, the different communication system transforms the control problem into a different one (with

deadtimes) which calls for architectures like the Smith Predictor control, and hence the results of one problems can not be used to directly evaluate the performance of the other.

Chapter 9

Results and Discussion

9.1 Steering System Decomposition

9.1.1 System Structure

The complete steering system is a 4×4 block-diagonal system which can be decomposed into two 2×2 subsystems. These subsystems represent steering in the x and y directions. Furthermore, within each subsystem, the individual steering stages are cascaded in a manner in which the output of the first steering stage is the reference to the second stage. Moreover, the control input of the first stage is coupled onto the output of the first stage. This gives rise to a cascaded 2×2 MIMO system with a lower triangular structure as shown by (4.2-3). The triangular structures are controlled by Proportional Integral control and MPC. This is done without decoupling the remaining off-diagonal transfer function for reasons discussed in the next subsection.

9.1.2 Diagonalization

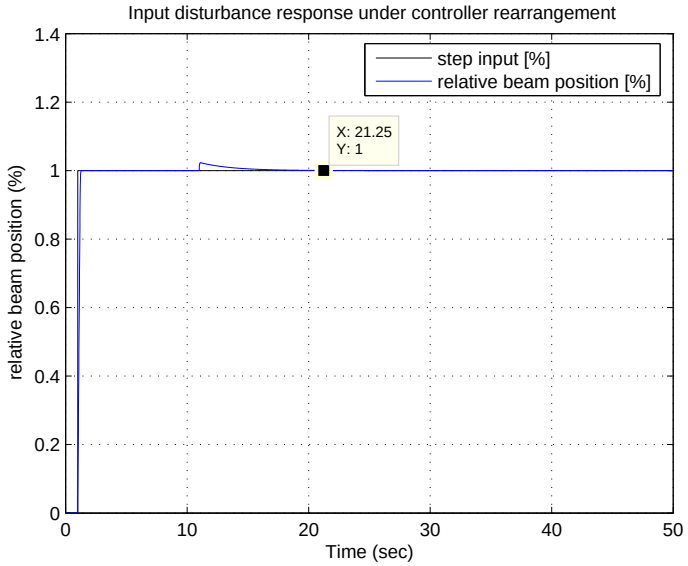
This system is mathematically diagonalizable by matrix manipulations as done in (6.5-1). The resulting structure should be controllable by two separate SISO controllers [12, 35]. However, the designed controllers for the diagonal transfer functions representing Steerer1 and Steerer2 yield undesirable results when put in cascade to control the diagonalized steering system. The results published are shown in Figure 6.5 for the re-configured PI controller, and Figure 6.7(b) for EG control. MPC also showed undesirable results. However, for MPC there seems to be several causes of bad results, since the results are bad even for the undiagonalized case, and an error in beam parallelism is observed in Figure 6.8.

9.2 Steering Controller

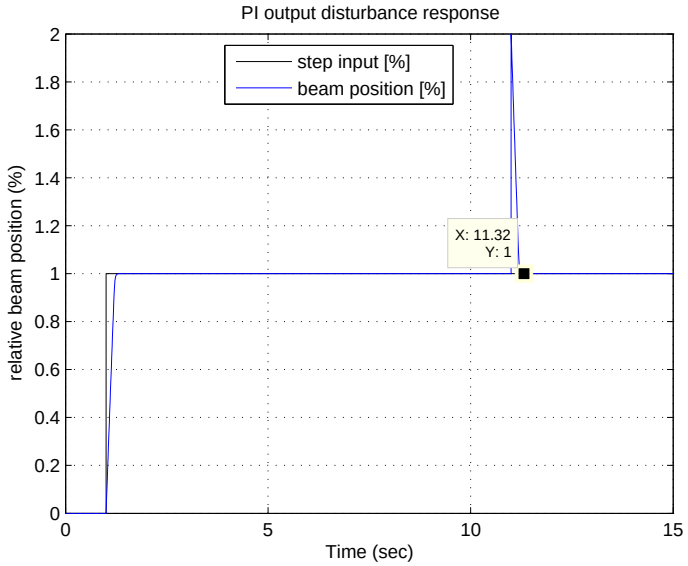
As a result of the diagonalization and the generic type of model governing the building blocks of the steering system, a PI controller (re-configured and with EG compensator) is found to offer the required performance. It gives the required closed loop speed-up, which can be tuned to give any required speed-up factor, ensures asymptotic set-point tracking through its integrating pole and the necessary disturbance rejection as demonstrated in plots shown in Figures 6.2 and 6.4. The PI controller used in both EG and re-configuration is tuned by the Affine parameterization method and the closed loop is tested for internal stability, which is the main concern in tuning methods that use pole-zero cancellation [22]. A non-dominant pole is added to the PI controller to adapt it for implementation with the EG compensator. The PI controller is given in (6.2-1). The model predictive controller also showed good response and asymptotic set-point tracking. The simulation results of this controller are given in Figure 6.3.

9.2.1 Performance of Controllers

All the controllers were tuned to give a a 0.25 seconds (100%) settling time. The reason for tuning the controllers the same is to be able to compare them on common grounds. The disturbance rejections of all the controllers are shown on the plots in Figures 9.1, 9.2 and 9.3.

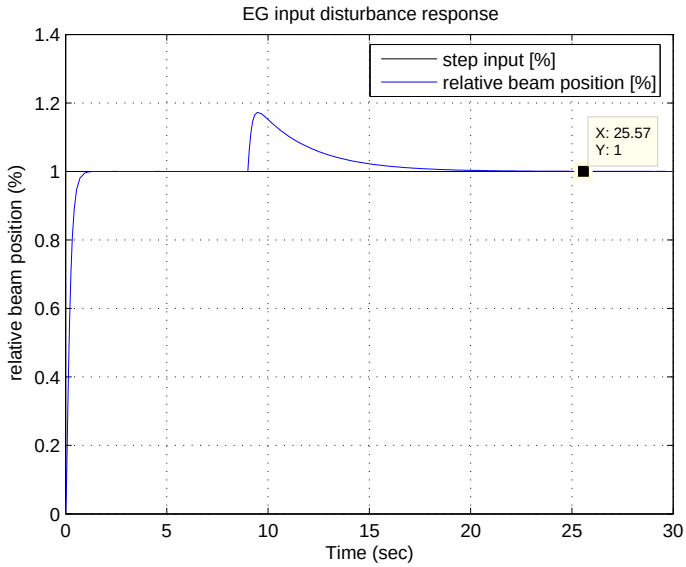


(a) Input disturbance

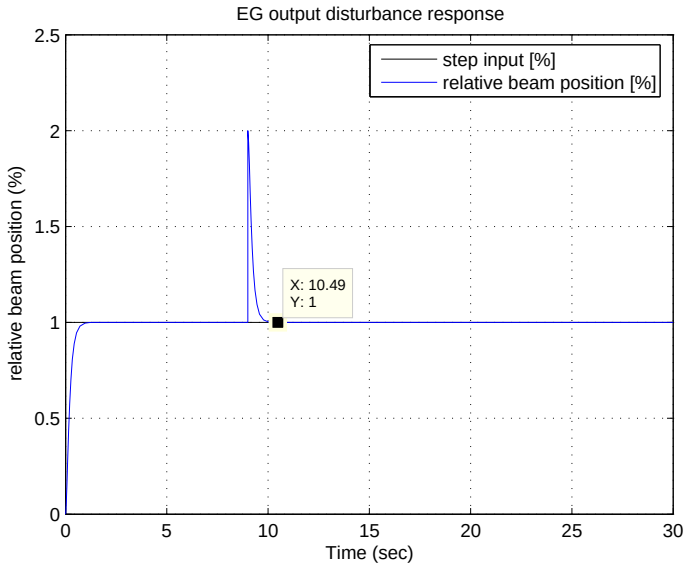


(b) Output disturbance

Figure 9.1: Unit step disturbance rejection of the reconfigured PI controller. The results shown on the plots show a good system response. This is because the system is able to recover from step disturbances within a reasonable time.

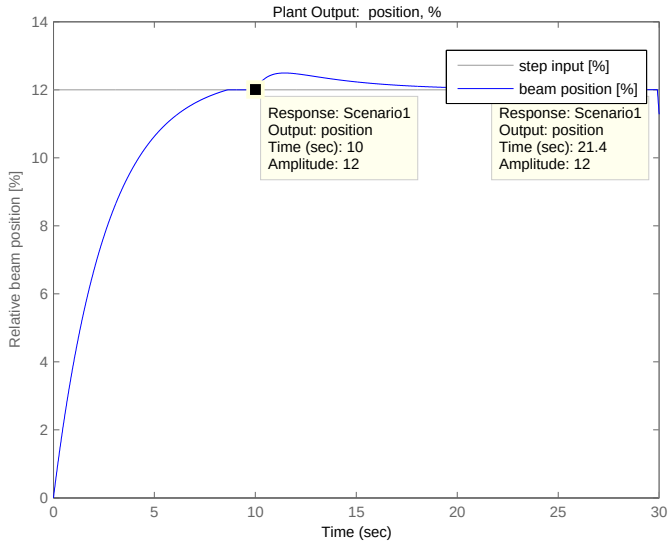


(a) Input disturbance

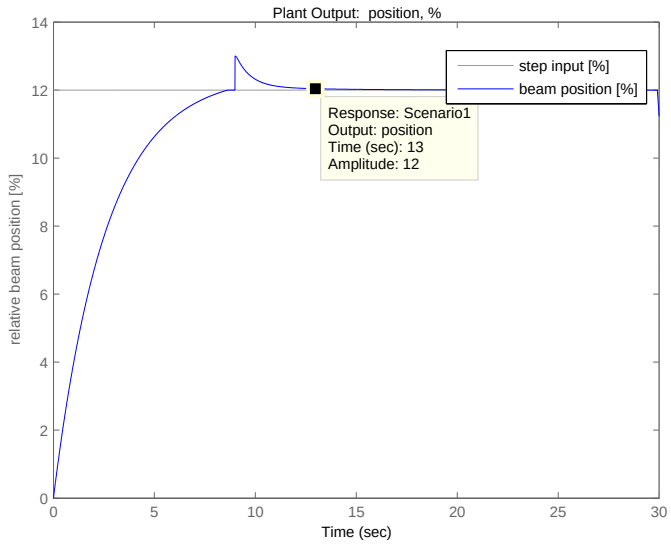


(b) Output disturbance

Figure 9.2: Unit disturbance rejection of PI controller with EG compensator. The results shown on the plots are good and comparable to those of a reconfigured PI controller. The system is able to recover from step disturbances within a reasonable time



(a) Input disturbance



(b) Output disturbance

Figure 9.3: Unit disturbance rejection of the MPC. The results shown on the plots are good

The tables below show the comparative times taken for the response of the steering system to settle (100%) back to its steady state value after occurrence of either an input or output unit step disturbance. The results are taken from the plots of the responses of the steering system under each one of the three control methods.

Table 9.1: Disturbance Recovery Time (s) of the Steering Control System

Disturbance injection point	PI re-configuration	EG Control	MPC
Output	0.32s	1.49s	3s
Input	10.24s	16.57s	11.4s

Table 9.2: Disturbance Attenuation of the Steering Control System

Disturbance injection point	PI re-configuration	EG Control	MPC
Output	0.04	0.18	1
Input	1	1	1

9.3 Windup Alleviation

All of the methods investigated are successfull at alleviating windup problems in actuators. However, each one of them has its advantages and disadvantages, and the methods are compared in the subsections that follow.

9.3.1 Controller Reconfiguration

This control scheme and anti-windup method showed superior results to the other two control methods.

Moreover, the control scheme is quite intuitive, and would not involve extra degrees of freedom in the controller, which would require explicit optimization to deal with windup problems. As a result of its simplicity, its computational needs are minimal compared to the other two methods. However, like all other methods this approach suffered poor

performance when the steering system structure was diagonalized forcing the control of the structure as a lower-triangular one. Possible reasons for this behavior are given in section 9.5. Simulation results of the cascaded steering system control are given in Figure 6.5 and Figure 6.6.

9.3.2 Error Governor Control

This approach employed the PI controller used for the previously discussed method. However, instead of rearranging the controller, a nonlinear anti-windup compensator was added to prevent actuator saturation. This method showed comparable results to the previous method, with a slightly poorer performance for small inputs. However, the source of that could be numerical inaccuracies. The EG method is more involved than the rearrangement of the controller and hence more susceptible to mistakes and numerical inaccuracies. Furthermore, this method involves optimization and its performance can be limited by the high computational needs of optimizers.

The simulation results of this control scheme are published in Figure 6.7. They show satisfactory control of the cascaded lower triangular structure of the steering system. However, the control of the diagonalized structure also fails as with the rearranged PI controllers.

9.3.3 Model Predictive Control

MPC generally gave problems for the control of the cascaded steering structure, both diagonalized and lower-triangular. However, this control method does eliminate windup behavior on a single steering basic building block. Furthermore, this method is also optimization based, and it is more computationally demanding. The results of this approach are shown in Figure 6.8 for the cascaded steering structure, and Figure 6.3

for the steering system basic building block.

The most important problem of MPC is the unavailability of methods to analyze its robust behavior, which may not be good for mission-critical applications.

9.3.4 Comparison and Summary

Based on the comparisons of the merits and demerits of the three control schemes investigated in earlier subsections, re-configured PI controll is preferable over the two other approaches and MPC is not recommended unless further work is done to rectify its poor performance. This is also supported by the disturbance rejection data given in Tables 9.1 and 9.2. The EG method is good but it is more involved and computationally demanding than the PI re-configuration.

9.4 Beam Centering and Parallelism Performance

Simulations done with the three control methods give different performances both in keeping the beam centered and parallel to the beamline axis. These are compared in the subsections that follow.

9.4.1 Beam Centering

Beam centering in the context of the work done here requires that the proton beam be both parallel to the beamline axis, and pass through the center of one of the sensors. Or in simpler terms, the beam has to pass through the centers of both the beam position sensors.

From the data obtained through simulation of the steering system under the different control methods, it can be seen that this may be achieved with the rearranged PI control and EG control. This is shown on the plots in Figures 6.6 and 6.7(a) for the lower-triangular beam steering subsystems. The lower-triangular steering system response under MPC showed very poor transient response characterized by overshoots and parallelism errors. This control method is not recommended for the steering system unless more work is done on it. The plots of the performance of MPC are given in Figure 6.8. However, all control methods have good steady state responce on centering the beam.

9.4.2 Beam Parallelism

The requirement for parallelism is that the two beam position sensors should read the same beam position values, regardless of whether the beam is centered on not. The values of the worst case errors in keeping the beam parallel, under the three control approaches are quoted below, as extracted from the responses quoted in the previous subsection. These are given in terms of the sensor reading difference, which is proportional to the angle of deviation of the beam from the parallel axis to the beamline.

- re-configured PI controller maximum error = 0.01%
- EG controll maximum error = 0.06%
- MPC maximum error = 4%

These figures show that the re-configured controller performs better than the EG controller which in turn outperforms MPC.

9.5 Diagonalization and IMC

Due to the similarity of the characteristic equations of the three transfer functions of the beam steering control system, the controllers designed for the steering PSU-magnet pairs contain the generating function of the offdiagonal transfer function of the system [22]. This has the effect of canceling the effect of the coupling due to this transfer function, according to IMP. This works especially for the steering system due to the fact that the coupled output (interpreted as an output disturbance) occurs at the same time as a change in the set-point of Steerer2.

For the above reason, any attempt to diagonalize the transfer function matrix of the steering system results in a double effort and gives undesirable results at the output. Therefore because of the control structure, the steering system feedback diagonalization is not done.

Chapter 10

CONCLUSION

10.1 MIMO Structural Decomposition

The Proton Beam Steering Control System structure has a 4×4 block diagonal structure which can be decomposed and controlled as two 2×2 control tasks which handle beam steering in the x and y direction separately.

The configuration that was initially proposed for the beam steering system at iThemba LABS [54] (see Figure 2.3), results in a very complicated system. This is because of the position of the steering magnet set between the two beam position sensors which makes it difficult to establish if the beam is parallel to the beamline axis. An alternative configuration is proposed in Figure 4.1. On the new configuration, the 2×2 subsystems give fully coupled system structures of the form given in Equation (4.2-1).

These fully coupled structures require complex control techniques. However, by altering the physical configuration of the beam steering components such that the first beam position sensor is sufficiently close to the second set of steering magnets as in Figure 4.4, the dominance of the corresponding off-diagonal transfer function is sufficiently reduced

resulting in a lower-triangular structure of the form given in Equation (4.2-3). Based on how critical the control is, this system structure can be diagonalized or controlled as is by SISO controllers. IMC was simulated on the lower-triangular structure, and yields good results.

10.2 Beam Centering and Parallelism

In order to simplify the control task, a configuration is suggested as shown in Figure 4.5, in which the second steering stage (Steerer 2) takes the output of the first steering stage (Steerer 1) as a reference input. In this configuration, the first steering stage ensures that the beam is centered by following an external reference input, while the second steering stage ensures that the beam is parallel to the beamline at all times by following the output of the first stage. Simulations of this configuration show good performance in centering the beam and keeping it parallel to the beamline axis at all times under re-configured PI control and EG control.

10.3 Windup Remedy

The three investigated anti-windup schemes show an improvement in control in that they eliminate saturation in actuators leading to windup. However, as stated in [5], the closed loop has to be designed to be significantly fast, because the anti-windup schemes slow it down sufficiently. A comparison of the simulation results of the three methods shows that for SISO control, controller rearrangement is more preferable than EG control and MPC. This is due to its simplicity and low computational demands. However, when one is faced by a strictly MIMO control task, it may be preferable to resort to other two since the concern is not only on alleviating windup, but also to preserve the direction of the vector of control inputs.

10.4 Achieved Steering Performance

Simulation results of the rearranged PI controller and EG control show relatively good results on keeping the beam centered and parallel relative to the beamline axis. Numerical performance indicators are quoted in Subsection 9.2.1. MPC showed poor results in keeping the beam parallel to the beamline as shown in the plots in Figure 6.8.

10.5 Future Work

Future work is needed, to investigate formal approaches in analyzing the robust behavior of MPC controllers. This is especially important for the MPC using the IMC algorithm, as IMC is a technique based on pole/zero cancellations that most engineers do not like, due to the internal stability issue. This will increase the confidence of engineers to use this powerful control method on more critical control tasks.

Bibliography

[1] Arruat, M., Jansson, A., di Maio, F., Hemelsoet, G.H., Lindroos, M., Tungesvik, O. 1997. Generic Automated Beam Steering and Beam Shaping Programs with an Object Oriented Approach. *Accelerator and Large Experimental Physics Control Systems*, Science Press.

[2] Åström, K.J., Verbruggen, H.B. 1989. Towards Intelligent PID control. *Artificial Intelligence and Real-Time Control*, Pergamon Press, New York.

[3] Åström, K.J., Verbruggen, H.B. 1989. Artificial Intelligence and Feedback Control. *Artificial Intelligence and Real-Time Control*, Pergamon Press, New York.

[4] Åström, K.J. 2003. Respect the Unstable. *IEEE Control Systems Magazine*, IEEE. pp. 12-25.

[5] Bemporad, A. 1997. Reference Governors: On-Line Set-Point Optimization Techniques for Constraint Fulfillment. [PhD Thesis]. Universit'a di Firenze, Italy.

[6] Bemporad, A. 2002. L_2 Anti-Windup Via Receding Horizon Optimal Control. *Proc. of the American Control Conference*, Anchorage. pp. 639-644.

[7] Bille, F., Pugliese, R., Abrami. A., Svensson, O. 1997. Applying Intelligent System Concepts to Automatic Beamline Alignment. *Accelerator and Large Experimental Physics Control Systems*, Science Press.

[8] Bozoki, E.S., Huang, J., Bittner, J.W. 1989. A Noniterative Method for Calculating Beam Position From Induced Electric Signals. *Particle Accelerator Conference*, IEEE. pp. 1519-1522.

[9] Braae, M. 1994. *Control Theory for Electrical Engineers*. UCT Press, Cape Town.

[10] Braae, M. 2004. *Control Engineering II*. UCT Press, Cape Town.

[11] Braae, M. 2006. Adaptive Control. [Lecture Notes], University of Cape Town.

[12] Braae, M. 2006. MIMO Control System Design. [Lecture Notes], University of Cape Town.

[13] Carlos, E. et al. 1989. Model Predictive Control: Theory and Practice = A Survey. *Automatica*. **25**. pp 335-348.

[14] Carstens, C. 2006. The Second Beam Line Project: A High Level Software Design, First Draft. [Interim release], iThemba LABS.

[15] Charpak, G., Rahm, D., Steiner, H. 1972. Some developments in the operation of multiwire proportional chambers. *Nuclear Instruments and Methods*. **80**. pp. 13-34.

[16] Chen, Y. 1996. Control of transverse motion caused by chromatic aberration and misalignments in linear accelerators. *Nunclear Instruments and Methods*. **398**. pp. 139-146.

[17] Conradie, L. 2006. Steering Magnets used at iThemba LABS. [personal Communication].

[18] Cottrell, R.L.A., Logg, C.A., Browne, M.J. 1979. A Feedback System for Steering and Correcting the Energy of the SLAC Beam in the Beam Switchyard. *Nuclear Instruments and Methods*. **164**. pp. 405-409.

[19] Duval, P. 1997. The Use of PCs in Controlling DESY Accelerators. *Accelerator and Large Experimental Physics Control Systems*, Science Press. pp. 162-165.

[20] Gilbert, E.G., Kolmanovsky, I., Tin Tan, K. 1995. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *Int Journal of Robust and Nonlinear Control*, **5(5)**, pp 487-504.

[21] Gill, K.F., Schwarzenbach, J. 1978. *System Modeling and Control*. Edward Arnold Publishers Ltd. London.

[22] Goodwin, G.C., Graebe, S.F., Salgado, M.E. (2001). *Control Systems Design*. Prentice Hall, Upper Saddle River, New Jersey.

[23] Greene, J. 2007. Neural, Fuzzy and Evolving Systems,[Lecture Notes]. UCT, Cape Town.

[24] Herman, Bruyninckx et. al 2002. Real-Time and Embedded Guide. [Online]. Leuven, Belgium

[25] Jongarius, J. 2002-2005. Writing Bug-free C Code. [Online]. Pearson Education Inc.

[26] Kapasouris, P., Athans, M. and Stein, G. 1988. Design of feedback control systems for stable plants with saturating actuators. *Proc. 27th IEEE Conf. on Decision and Control*, pages 469479, Austin, Texas, U.S.A.

[27] Keil, J., Dietrich, J., Mohos, I. 1999. Control and Data Processing of the Distributed 500 MHz Narrowband Beam Position Monitor System of ELSA. *Proceedings of the 1999 Particle Accelerator Conference*, New York. pp. 2054-2056.

[28] Kessler, C. 1958. Das Symmtrische Optimum. *Regelungstechnik*, **6**, pp.395-400, 432-436.

[29] Kohler, I., Cronje, F. 2006. Developers of the Current steering system at iThemba LABS. [Personal Communication].

[30] Kohler, I.H., Weehuizen, H.F., Hogan, M.E., Theron, P.J. 1997. The PC Based Control System of the NAC. *Accelerator and Large Experimental Physics Control Systems*, Science Press, Beijing, China. pp. 175-176.

[31] Loron, L. 1997. Tuning PID Controllers by the Non-Symmetrical Optimum Method, *Automatica*, **33**, pp.103-107.

[32] Lussi, C. 2007. Bipolar Power Suppliers used at iThemba Labs. [personal Communication].

[33] Ma, L., Shi, P., Lu, H.J., Cheng., Ye, K.R. 1997. Using PCs in the BEPC Beam Diagnostic Instrumentation System. *Accelerator and Large Experimental Physics Control Systems*, Science Press.

[34] Ma, L., Nawrocky, R.J., Rarback, H.M., Singh, O.V., Yu, L.H. 1987. An Automatic Beam Steering System for the NSLS X-17T Beam Line Using Closed Orbit Feedback. *Particle Accelerator Conference*, IEEE. pp. 512-514.

[35] Maciejowski, J.M. 1989. *Multivariable Feedback Design*. Addison-Wesley Publishing Co. Carlifonia.

[36] Mastoscusa, E.J. (11,April,2007). Exploring Classical Control Systems. [Online]

[37] Morris, J.T., Abola, A.C. 1997. What Objects Do Controls Applications Need?. *Accelerator and Large Experimental Physics Control Systems*, Science Press. pp. 368-371.

[38] Nikolau, M. (May 2007) Model Predictive Controllers: Acritical Synthesis of Theory and Industrial Needs. Chemical Engineering Dept, University of Houston, Houston. [Online].

[39] Peterson, D.G. 1972. Proton Beam Positioning System with Computer Control. *Nuclear Instruments and Methods*. **104**. pp 451-454.

[40] Petit, E., Gudewicz, P. 1988. Capacitive Beam Position Monitors and Automatic Beam Centering in the Transfer Lines of GANIL. *Particle Accelerator Physics Conference*, IEEE. pp. 1142-1144.

[41] Pillai, C., Oothoudt, M., Zubro, M. 1998. Automatic Beam Positioning Control at LASREF. *Particle Accelerator Conference*, IEEE. pp. 2064-2066.

[42] Proctor, F.M., Shackleford, W.P. 2002. Embedded Real-Time Linux for Cable Robot Control. *The Computers and Information in Engineering*, ASME. pp. 1-6.

[43] Quinn, B., Beaudoin, S. et al. 2003. Design and Testing of a Fast Beam Position Monitor. *Particle Accelerator Conference*, IEEE. pp. 2571-2573.

[44] Sakildien, M. Measurement of Dipole Magnetic Field. [personal conversation], iThemba LABS.

[45] Sastry, S., Bodson, M. 1994. *Adaptive Control: Stability, Convergence, and Robustness*, Prentice-Hall Advanced Reference Series (Engineering).

[46] Sei, N., Yamazaki, T., Toyokawa, H., Ohgaki, H., Suzuki, R., Sugiyama, S., Mikado, T., Yamada, K., Ohdaira, T. 1997. PC-based Control System in Storage Ring TERAS. *Accelerator and Large Experimental Physics Control Systems*, Science Press.

[47] Stovall, J.E., Gray, E.R., et al. 1996. Alignment and Steering Scenarios for the APT LINAC. *Particle Accelerator Conference*, IEEE. pp. 686-688.

[48] Takako, M., Hirohiko, S., Sato, Y., Irie, Y. 2003. Beam Position Measurement using LINAC Microstructure at the KEK Booster Synchrotron. *Particle Accelerator Conference*, IEEE. pp. 2509-2511.

[49] Tham, M.T. 2002. Internal Model Control,[Online Lecture Notes]. Chemical and Process Engineering, University of Newcastle Upon Tyne, http://lorien.ncl.ac.uc/ming/imc.pdf, (19 September 2007).

[50] Tśoeu, M.S. 2007. System Identification Report, [Interim report], iThemba LABS.

[51] Tśoeu, M.S. 2007. Current Control System Performance Evaluation Report, [Interim report], iThemba LABS.

[52] Tśoeu, M.S. 2006. Conceptual Design of the Proton Beam Steering Control System. [Interim Release], iThemba LABS.

[53] Tśoeu, M.S. 2006. Steering Control Analysis. [Interim Release], iThemba LABS.

[54] van Tubbergh, C., De Kock, E.A. 2006, Conceptual design of the proton beam treatment facility at iThemba LABS. [Interim release], iThemba Labs.

[55] van Der Merve, J. 2007. Interfacing the to the Power Supply. [personal Communication].

[56] Wiedemann, H. (1993). *Particle Accelerator Physics*: *Basic Principles and Linear Dynamics*. Springer-Verlag, USA.

[57] www.tlabs.ac.za, 27 February 2007. [Online].

[58] Ziegler, J.G., Nichols, N.B. 1942. Optimum Setting for Automatic Controllers. *ASME Trans*. **64**, pp.759-768.

[59] Zoltan, 2007. Sensor Data Acquisition. [personal Communication].

Appendix A

A.1 Error Governor M-file

```
function lambda = errorGov(y)
%
%       lambda = errorGov(C,y,lim,u)
%
%       This function computes the states of a LTI system from its output
%       and C matrix. It assumes that D = 0 in y = Cx + Dx. It uses
%       Moore-Perose pseudo inversion to achieve that. The function then
%       determines if the states are within the interior of the constraint
%       set. It then selects a multiplier lambda(t) that will ensure no
%       saturation, according to the algorithm proposed by Bemporad1988
%       et al

C = [0.2099 10.83];
x = C\y        %compute the states

%compute the interior of the constraint set X and determine if x is an
%element of that interior.
```

```
xlim = [0 0.4617];


cond = abs(x(2))<xlim(2);
if cond == 1
    lambda = 1;
else
    lambda = abs((24.44*0.4168)/(12*10.83+0.01)); %prevent division by zero
end
```

A.2 Simulink Models Used for Simulations

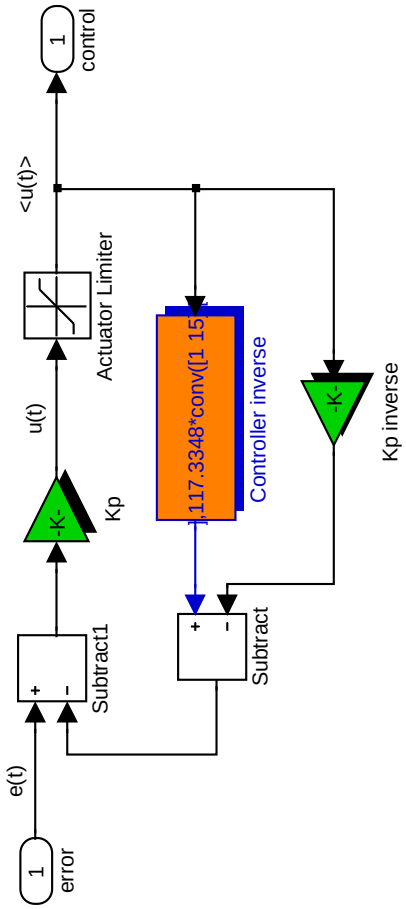


Figure A.1: PI controller rearrangement

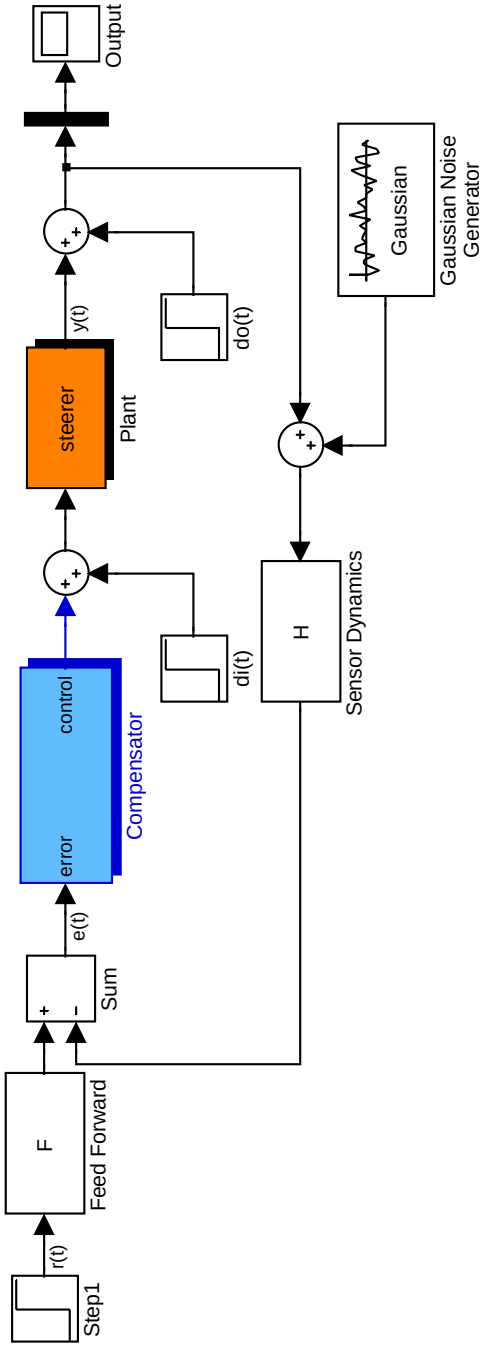


Figure A.2: Closed loop Simulink model used to implement anti-windup. The rearranged controller in Figure A.1 is used

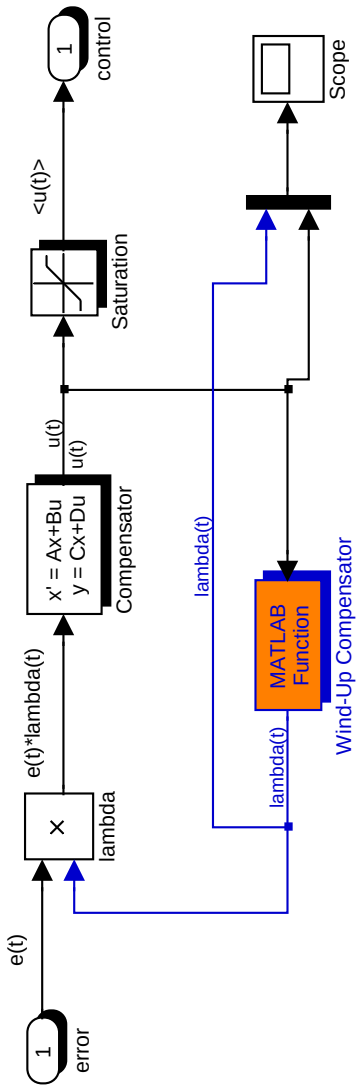


Figure A.3: Controller Simulink model for EG simulation

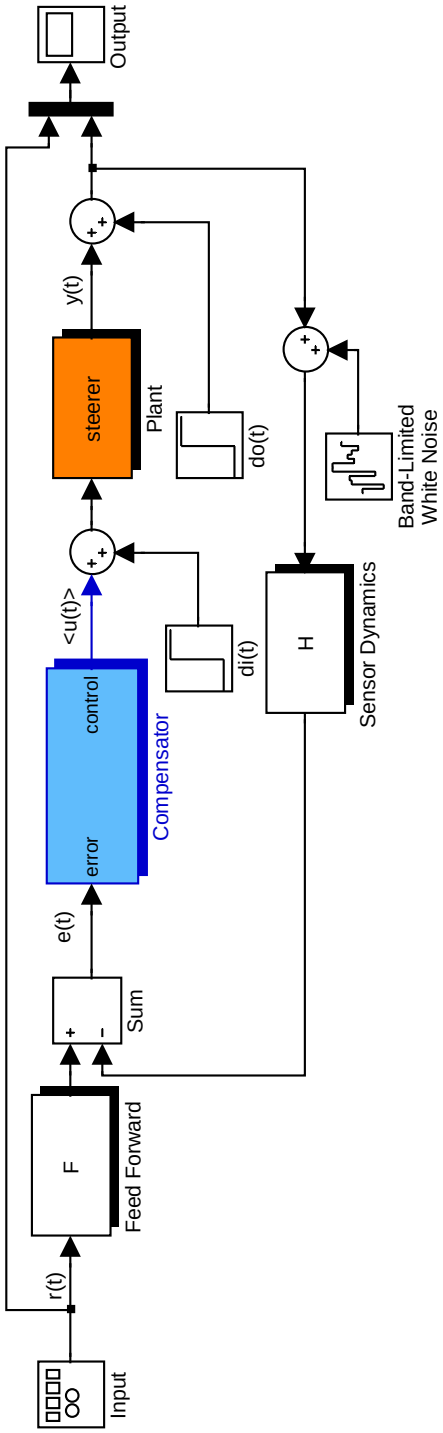


Figure A.4: Closed loop Simulink model for EG simulation

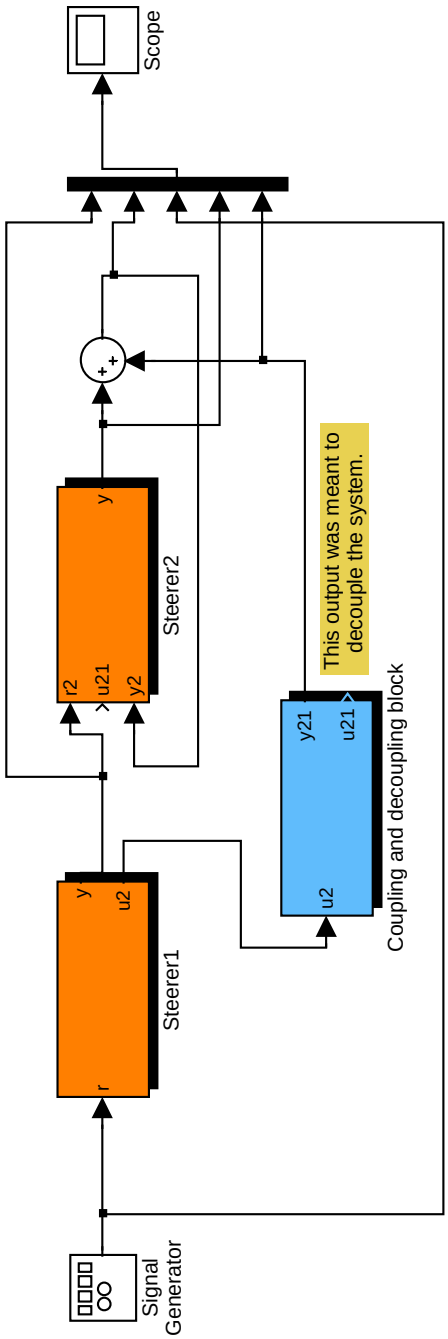


Figure A.5: Simulink model of the cascaded one direction Steering system: Steerer2 reference is the output of Steerer1, and the input of Steerer1 is coupled onto the output of Steerer2 through g_{21}. A means of decoupling the output of Steerer2 is provided as u21