
CONTROLLER DESIGN FOR VEHICLE HEADING CONTROL

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & INSTRUMENTATION ENGINEERING

BY

SRINIT DAS (109EI0320)AND

UPASANA PRIYADARSINI PAL (109EI0332)

UNDER THE SUPERVISION OF

PROF. TARUN KUMAR DAN

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that the project work titled "**Controller Design for Vehicle Heading Control**" submitted by ***Srinit Das (109EI0320)*** and ***UpasanaPriyadarsini Pal (109EI0332)*** in the partial fulfilment of the requirements for the degree of**Bachelor of Technology** in **Electronics and Instrumentation Engineering** during the session 2012-2013at National Institute of Technology, Rourkelais anauthentic and bona fide work carried out by them under my supervision.

Date**:**

Prof. Tarun Kumar Dan, Associate Professor

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

We would like to express a deep sense of gratitude towardsour supervisor, Prof. Tarun Kumar Dan, for his excellent guidance, whole-heartedsupport and constant motivation. Without his cooperation, this project work could not have been successfully completed.

We would also like to thank all faculty members and staff of the Department of Electronics and Communication Engineering for their help and guidance. They have been the cornerstone of progressive learning and innovation throughout our stay in NIT Rourkela.

We extend our thanks to our family and friends whose valuable outlook and unwavering support would be an inspiration throughout our life.

Srinit Das (109EI3020)

UpasanaPriyadarsini Pal (109EI0332)

ABSTRACT

Unmanned ground vehicles have significant role to play in the fields of military, space exploration, accessing dangerous terrain, operating in hazardous industries and process stations. Control of the heading of the vehicle is critical to its safety and stability. Modelling the parameters and process is critical to developing a control strategy for it. There are multiple factors that are responsible for the heading of the vehicle which can lead to a complicated process model. However, the model has been simplified by suitable approximations. The objective of this research is to control the heading of a vehicle in the real world environment under the unpredictable and unstructured surroundings. The change of heading direction and the speed of the vehicle influence to the motion of a vehicle. The vehicle model that was assumed was a four wheeled electric motor driven car, whose steering system is modified to be controlled by a DC servo motor mounted at the steering wheel shaft, and is fitted with a compass sensor to measure the current heading direction of the vehicle. Various control strategies were simulated in Simulink to converge at the most appropriate one. The chief objective was to achieve minimum settling time and also reduce overshoot and steady state error. Strategies such as double loop controller, Internal Model Controller (IMC), Modified IMC and robust controller are used for the same process and the performance of each is presented.

CONTENTS

CHAPTER 1

INTRODUCTION

Introduction:

An autonomous robot is the robot that has ability to objectively perform the commanded tasks. There are various types of autonomous robots that can be easily found in many research projects and some applications also available in human daily life. Recent trends in autonomous robot focus on every type of mobile vehicles; ground, underwater, and also aerial vehicles. There are several types of control algorithms that are used to control robots which depend on the particular mission purposes.

To control the motion of a ground vehicle, the heading control problem is substantially concerned. The change of the desired heading direction and the change of the vehicle heading under the real environment are the problems. In addition, the speed of a vehicle is significantly affects the motion of the vehicle. The vehicle controller receives the desired heading direction and controls the current heading of the vehicle to move in the desired direction in order to arrive at predefined geographic coordinates. From this point, we concern the problem of controlling the heading direction of a vehicle to maintain its continuous motion along the desired trajectory. A four wheeled electric motor driven car with front wheel steering was considered as the sample vehicle to be modelled and controlled. This is the most common type of unmanned vehicle that is simple in principle and can be employed to perform a wide variety of tasks.The dynamic model of the vehicle's heading and steering are adopted from a document of experiment conducted in order to obtain the parameters. The transfer function was identified based on the parameters of the BAJA vehicle built by NIT Rourkela. In addition, the speed of a vehicle was taken into account for identifying the model of a tested vehicle. The parameters of a model were estimated by using Least Squares Method (LSM).

Initially, a double loop controller wasdesigned for the vehicle's heading control. The controller was decoupled into two cascaded loops. The inner loop was the position control implemented by PID control algorithm and the outer loop was the heading control implemented by PD control algorithm. The combination of PD-PID controller could improve the transient response of the vehicle headingwhen the desired heading changed abruptly. But it had a slight amount of overshoot and steady state error.

An Internal Model Controller (IMC) was, then, implemented to overcome the disadvantages of the double loop configuration. The transient response could be improved significantly with very desirable settling time by having a low filter time constant. Also the overshoot was eliminated. But robustness was compromised due to it. The mathematical model of the process had to be very accurate. Model mismatch could lead to bad transient performance. In order to make the controller more robust to model mismatch, a modified IMC scheme was implemented. However, it lead to poor transient response. Modified IMC has the advantage that it uses much less number of components or software processing. Further, a two degree of freedom IMC was implemented to increase the robustness of the system and it showed significant improvement in term of insensitivity to a substantial degree of model mismatch.

CHAPTER 2

MATERIALS AND METHODS

2.1: Vehicle steering

The alignment of the front wheels of a vehicle is responsible for steering the vehicle. As the wheel is turned, the vehicle teds to turn at the direction the front rims point to. But the direction the vehicle is heading is not quite the direction the wheels are pointing.

The angle at which the wheels are pointing is known as the steering angle. The angle at which the vehicle is turning is the heading angle. The difference between these two angles is known as the slip angle. This concept is illustrated in figure 1.

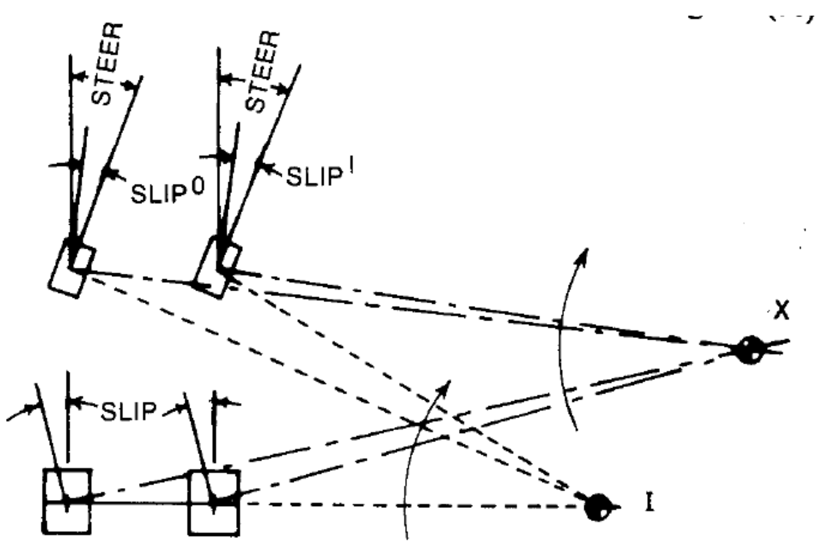

Figure 1: Steering angle, heading angle and Ackerman geometry

Cornering force is the centripetal force offered by the tyres to make the vehicle move in a curved path. Higher is the cornering force, tighter and faster the turn can be taken. The cornering force is a function of slip angle. The relationship between the slip angle and cornering force for a typical road tyre can be seen in figure 2.

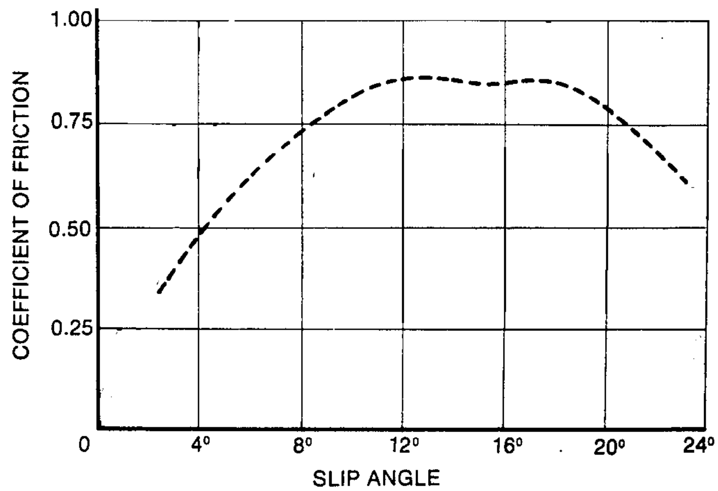

Figure 2: Relationship between slip angle and effective coefficient of friction

Apart from slip angle, other factors that affect the steering of a vehicle are caster angle, camber angle, King Pin Inclination, toe angle, Ackerman geometry, suspension stiffness, etc.

2.2: Proposed Hardware configuration

The hardware model proposed for this project is an unmanned ground vehicle with rack and pinion steering system. An electric motor or an IC engine may be used to propel the vehicle. The steering system here would be modified to facilitate automatic control. A DC servo motor installed at steering wheel shaft would control the steering angle to settle at the desired heading angle.

A compass sensor would be used to measure the current heading angle. The direction of travel would be determined by a set of predefined waypoints or as input from an external controller. We do not concern ourselves with the input of heading angle for now. Our main objective is to stabilize the heading direction with minimum overshoot and zero steady state error.The position of the steering wheel can be determined by a potentiometer or incremental encoder and the control signal would control the DC servo motor to change the steering wheel position.

2.3: System Identification

The vehicle's steering transfer function which was used for our simulations was based on a single track model. The dependence of this transfer function on physical parameters was identified through experiments as proposed in [2].The output of the system is the vehicle heading direction which can be measured by using an electronic compass. The system input is the steering angle that can be measured using an incremental encoder.

The model can be represented by the following state equation.

$$\dot{x} = Ax + Bu \quad (1)$$
$$y = Cx + D \quad (2)$$

This can be written as

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \delta \quad (3)$$

$$\psi = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \psi \end{bmatrix} \quad (4)$$

The state space representation can be converted to the transfer function form which indicates the relation between input steering angle and output heading angle by following equation.

$$\frac{\psi(s)}{\theta(s)} = C(sI - A)^{-1} B + D \tag{5}$$

The transfer function can be approximated as the second order system with one integrator and unknown coefficients as shown in the following equation.

$$\frac{\psi(s)}{\theta(s)} = \frac{K_{steer}(a_1 s + a_2)}{s(b_0 s^2 + b_1 s + b_2)} \tag{6}$$

The unknown coefficients a1, a2, b0, b1, and b2 were identified manually by experimentation in several conditions. The steering angle and the output heading angle information were measured by the steering encoder and compass sensor respectively. The set of collected data was fit and the parameters were estimated by using LSM algorithm [2].

$$\begin{aligned}
a_1 &= C_f C_r m U^2 \\
a_2 &= C_f C_r (l_f + l_r) U \\
b_0 &= I_z m U^2 \\
b_1 &= [C_f (I_z + l_f{}^2 m) + C_r (I_z + l_r{}^2 m)] U \\
b_2 &= C_f C_r (l_f + l_r)^2 + (C_r l_r - C_f l_f) m U^2
\end{aligned}$$

K_{steer} Steering ratio; ratio between the turn of the steering wheel and the angular displacement of the wheel.

C_f, C_r Tyres corner stiffness coefficients of front and rear wheels respectively.

l_f, l_r Distances from the center of gravity to the front and rear axles respectively.

M Mass of the vehicle.

U Longitudinal velocity.

I_z Moment of inertia around z-axis.

As can be seen from the equation, the only dynamic parameter in the system transfer function is the velocity of the vehicle. All the other parameters are fixed for a particular vehicle.

The input to the system, the steering angle is provided by a servo motor. The servo motor is assumed to have a second order transfer function of the form

$$G_s(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

2.4: Controller configurations:

There could be many control strategies to stabilize the process variable and to achieve low overshoot, low settling time and zero steady state error. Some control strategies considered for our process are:

- Single loop PID controller

- Cascaded double loop controller
- Internal Model Controller
- Modified Internal Model Controller
- Two degree of freedom Internal Model Controller

The above control strategies were studied in detail and implemented to stabilize the process and bring about faster transient response and minimise the steady state error. All simulations were done in Simulink and some of the graphs were plotted in MS Excel.

2.4.1: Proportional Integral Derivative Controller

A proportional-integral-derivative controller (PID controller) is a type of feedback loop controller very commonly used to control a wide variety of industrial processes. PID controller operates based on an error value which is the difference between a measured process variable and a desired setpoint. The controller tries to minimize the error value by suitably adjusting the control inputs for the process.

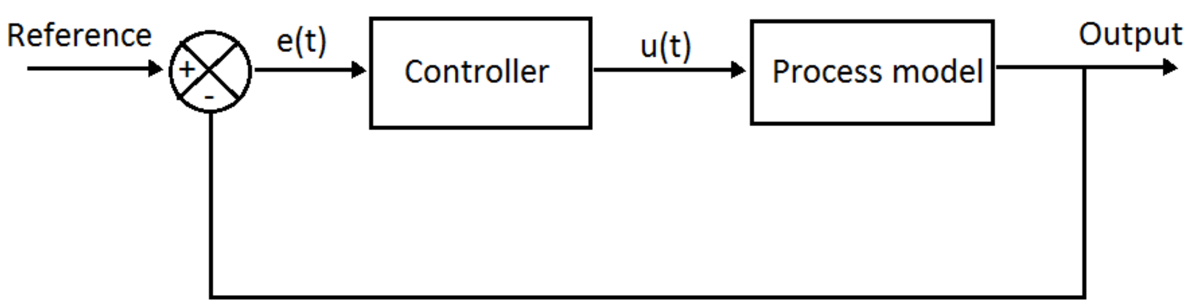

Figure 3: A classic unity feedback control configuration

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, dK + K_d \frac{de(t)}{dt}$$

$$U(s) = \left(K_p + \frac{K_i}{s} + K_d s\right) E(s)$$

The PID controller transfer function involves three separate constants as parameters:

- **Proportional term (K_p)** - The proportional term produces a controller output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, known as the proportional gain constant. A high proportional gain gives a large change in controller output for a given change in the error. If the proportional gain is very high, then the system can become unstable as the gain of the system exceeds the ultimate gain. On the other hand, a small proportional gain results in a small controller output for a large error value. If the proportional gain is too low, the control action may be too small to properly respond to system disturbances. The proportional term in the PID controller transfer function decreases the rise time, increases overshoot and decreases the steady state error. However, it does not significantly affect the settling time.
- **Integral term (K_i) -** The integral term adds to the controller output a value proportional to the integration of instantaneous error over time. The integral in a PID controller represents the accumulated error in the closed loop process that should be corrected. The accumulated error is multiplied by the integral gain (K_i) and added to the controller output. By adding this value to the controller output, it eliminates the steady state error by accelerating the movement of the process towards the setpoint. Increasing the Integral gain typically decreases rise time, increases overshoot and completely eliminates steady state error. However, it does not bring about much change to the settling time.

- **Derivative term (K_d)** - The slope of the error in time domain indicates whether the process variable is moving towards or away from the setpoint. In this way it is able to somewhat predict the movement of the error and modifies the controller output according to the derivative gain (K_d). Since derivative action is able to predict the behaviour of the error, it greatly improves the transient response. The derivative term in the PID controller decreases the overshoot and settling time, while not much affecting rise time and steady state error.

In the absence of knowledge of the underlying process, a PID controller is considered appropriate because it can be easily tuned to effectively control almost any kind of process. By tuning the three parameters in the PID controller, the controller can be effectively implemented for specific process requirements of most processes.

Some applications may not require all three actions. Where fast transient response is not necessary, a PI controller (K_d set to zero) may solve the purpose. Alternatively, a PD controller (K_i set to zero) gives fast response at the cost of steady state error. We might even use a P controller (K_i and K_d both set to zero) - because it is much simpler - for processes which require only marginal improvement.

A PID controller is more often represented in the form of the equation

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau + T_d\frac{de(t)}{dt}\right)$$

$$U(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)E(s)$$

Here, T_i and T_d are integral and derivative time constants respectively.

The PID action may not always operate on the error value. The derivative term can be based on the process variable itself to avoid the occurrence of spikes in the controller output if the setpoint changes abruptly.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau)\, d\tau + T_d \frac{dy(t)}{dt} \right)$$

2.4.2: Double Loop Controller

The feedback control mechanism may be decoupled into two loops for improved response. In this configuration we have two feedback loops. The inner loop works to stabilize the faster components of the process while the outer loop operates on the slower components. The controller chosen in either loops can be any of the PID configurations mentioned in the previous section. It is essential to have minimum error and fast response. Hence, for most processes, the inner loop is controlled by a PID controller and the outer loop is controlled most often by a PD controller which gives fast response.

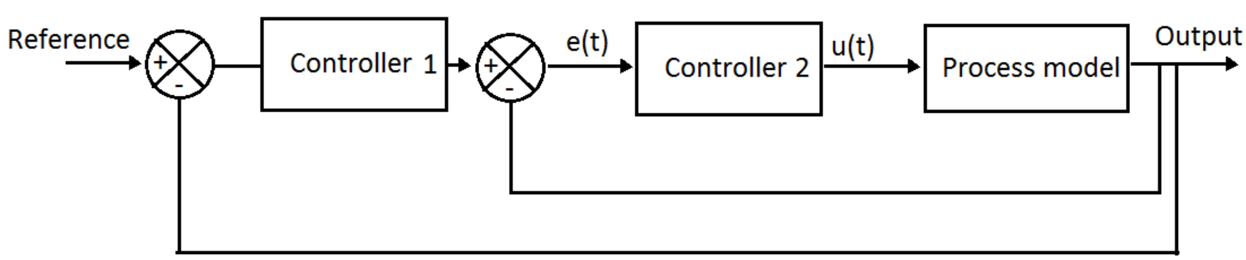

Figure 4: Double Loop Controller Configuration

2.4.3: Internal Model Controller

The theory of IMC states that "control can be achieved only if the control system encapsulates, either implicitly or explicitly, some representation of the process to be controlled". In particular, if the control scheme has been developed based on an exact model of the process, then perfect control is theoretically possible.

The Internal Model Controller is based on the inverse of the process model we are trying to control. If we cascade the process transfer function with a controller which is the exact inverse of the process, then effectively the gain becomes unity and we have perfect setpoint tracking.

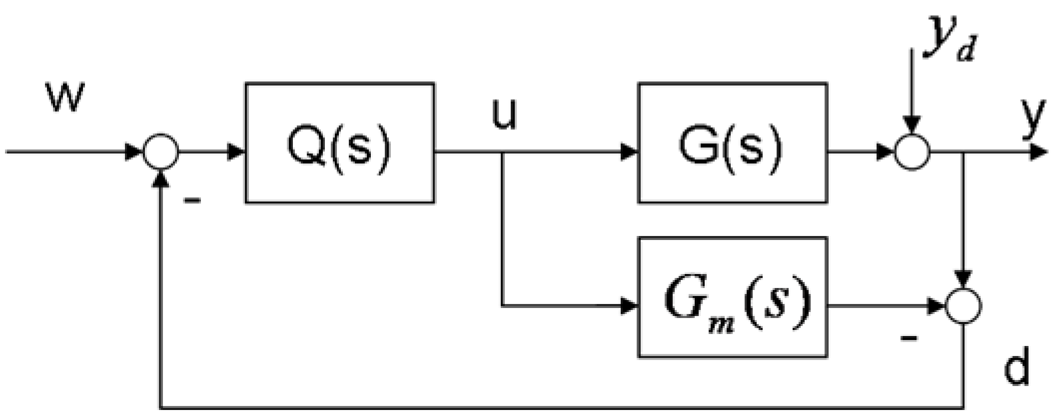

Figure 5: Block diagram of general IMC configuartion

The same input is given to the actual process and the process model. The output of both of them is compared and the difference generated. This difference, or the error produced is subtracted from the setpoint to bring about the desired tracking. The error signal also contains information about the disturbance which might have crept into the actual process. Now Q(s) must be designed such that it gives the fastest response.

Assume $G(s) = G_m(s)$ for perfect model match, $Q(s) = G^{-1}(s)$ and $y_d=0$ for zero disturbance.In this case the error d is zero. Hence, there is no feedback and it acts as an open loop control system. And since Q(s) is the inverse of the process model, the model exhibits perfect setpoint tracking.

However, this idealised form of IMC is never possible in real conditions. There are two chief reasons for this

- The mathematical model of the process can never be exactly same as the process. It is not possible to model a system with full accuracy.
- The inverse of the process cannot be always realisable. Not all functions are invertible.

Therefore in order to design the controller Q(s), we only take the inverse of those terms of $G_m(s)$ which are invertible. That is, the process model transfer function is represented by two components.

$$Gm(s) = G_{m+}(s)\, G_{m-}(s)$$

Where $G_{m+}(s)$ represents the non-invertible part and$G_{m-}(s)$ represents invertible part. Therefore,

$$Q(s) = G_{m-}^{-1}(s)$$

Even then, there is an additional criterion on the controller transfer function. It has to be a proper fraction, i.e., The order of the denominator must be greater than or equal to the order of the numerator. Hence, if the process transfer function has a higher order denominator, then the controller would become a improper fraction. Improper fractions are not physically perfectly realisable because of the need of a derivative term which cannot be perfectly duplicated. Hence, the controller transfer function is multiplied by a filter function f(s).

$$Q(s) = G_{m-}^{-1}(s)\ f(s)$$

Where f(s) is chosen such that the controller transfer function Q(s) becomes a proper fraction. The filter often has the form

$$f(s) = \frac{1}{(\lambda s+1)^n}$$

λ is the tuning parameter for this controller. A low value of λ gives fast response and high value of λ makes the system robust to model mismatch. Depending on the process, the filter function f(s) may be any different function as well having other tuning parameters.

2.4.4: Modified Internal Model Controller

The IMC theory presented in the preceding section requires the use of a large amount of hardware. The control system, if implemented in software, also would require complicated implementation. Hence, it is modified slightly to reduce the amount of hardware and software used. The model inverse is still used for computing the controller transfer function. The filter f(s) for controller Q(s) is designed as a second order transfer function. If the model is perfect, the process transfer function would be cancelled and we would have only a second order process in unity feedback configuration. We can now tune the parameters of the second order transfer function to show any desired response based on a set of performance criteria set down by the design requirement.

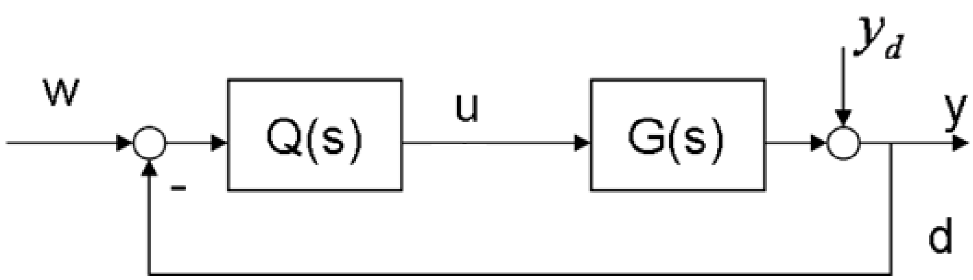

Figure 6: structure of Modified IMC. It is equivalent to a classic feedback system. Only the controller is dependent on the model in this case

$Q(s) = Gm_-^{-1}(s)\ f(s)$

Where, the filter function f(s) has a second order transfer function.

$f(s) = \frac{\omega_n^2}{s^2+2\xi\omega_n+\omega_n^2}$

now,settling time $T_s = \frac{4}{\xi\omega_n}$

and%Overshoot $OS\% = e^{\frac{\xi\pi}{\sqrt{1-\xi^2}}} * 100\%$

The values of T_s and OS% could be set based on performance characteristics. Based on those values, the value of ξ and ω_n can be calculated.

$$\xi = \sqrt{\frac{(\ln\left(\frac{OS\%}{100\%}\right))^2}{\pi^2 + (\ln\left(\frac{OS\%}{100\%}\right))^2}}$$

$$\omega_n = \frac{4}{\xi T_s}$$

2.4.5: Two Degree of Freedom Internal Model Controller

In the case of conventional IMC theory, the controller has one tuning parameter, i.e., λ.

Low value of λ gives fast response and high value of λ gives robustness to model mismatch. There is a trade-off between these performance criteria. However, the Standard IMC control strategy may be modified and incorporated with two separate controllers to take care of setpoint tracking and disturbance rejection separately.

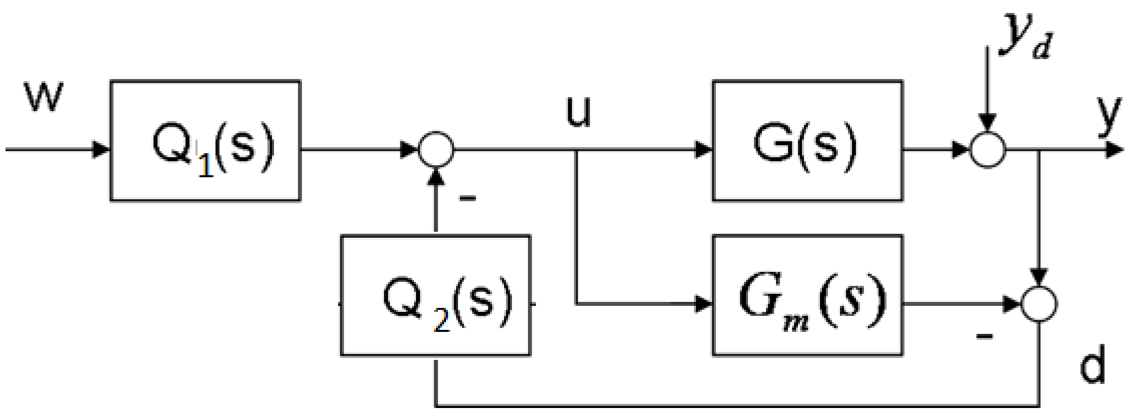

Figure 7: Two degree of freedom IMC control strategy

It can be seen from the equation below that both controllers are independent of each other at least for the nominal case of perfect model match.

$$y = \frac{G(s)Q_1(s)}{1 + Q_2(s)(G(s) - G_m(s))} w + \frac{1 - G_m(s)Q_2(s)}{1 + Q_2(s)(G(s) - G_m(s))} y_d$$

Assume $G_m(s) = G(s)$ for nominal case

Then $y = G_m(s)Q_1(s)w + (1 - G_m(s)Q_2(s))y_d$

Here, the controller $Q_1(s)$ takes care of setpoint tracking and controller $Q_2(s)$ takes care of disturbance rejection. Both of them can be designed with their specific performance criteria. Even though they dependent on each other when there is a model mismatch, they still exhibit a level of robustness in such cases.

CHAPTER 3

SIMULATIONS AND RESULTS

3.1: Vehicle Parameters and Process Transfer Function

The vehicle that was considered for modelling is the BAJA vehicle fabricated by team Black Mamba Racing from NIT Rourkela. The values of all the parameters that affect the process transfer function are:

$C_f = 0.12$

$C_r = 0.12$

$M = 390$

$I_f = 0.81$

$I_r = 0.66$

$I_z = 103$

The transfer function also depends on the speed of the vehicle U, which is the only parameter that can vary. Rest all of the parameters remain constant. Therefore the process transfer function can be represented as

$$\frac{\psi(s)}{\theta(s)} = \frac{513U^2 s + 20.32U}{s(7U^2 s^2 + 120Us + 41U^2 + 283)}$$

Different control strategies have been implemented to the process transfer function at four different speeds – 0.2m/s, 1m/s, 3m/s and 12m/s.

For a speed of 0.2m/s, the transfer function is

$$\left(\frac{\psi(s)}{\theta(s)}\right)_{u=0.2} = \frac{20.52s + 4.07}{s(0.28s^2 + 6s + 4.4)}$$

For a speed of 1m/s, the transfer function is

$$\left(\frac{\psi(s)}{\theta(s)}\right)_{u=1} = \frac{513s + 20.32}{s(7s^2 + 120s + 324)}$$

For a speed of 3m/s, the transfer function is

$$\left(\frac{\psi(s)}{\theta(s)}\right)_{u=3} = \frac{4852s + 60.32}{s(63 + 760s + 4652)}$$

For a speed of 12m/s, the transfer function is

$$\left(\frac{\psi(s)}{\theta(s)}\right)_{u=12} = \frac{7763s + 241.28}{s(1008s^2 + 8960s + 993720)}$$

The choice of servo motor model transfer function is trivial, because it is a simple second order transfer function. A typical servo motor transfer function was chosen is as follows.

$$G_s(s) = \frac{30}{s^2 + 20s + 30}$$

3.2: Single Loop PID Controller

The controller was tuned manually by trial and error to provide optimal results. The transient response for single loop PID control strategy is presented in figure 9.

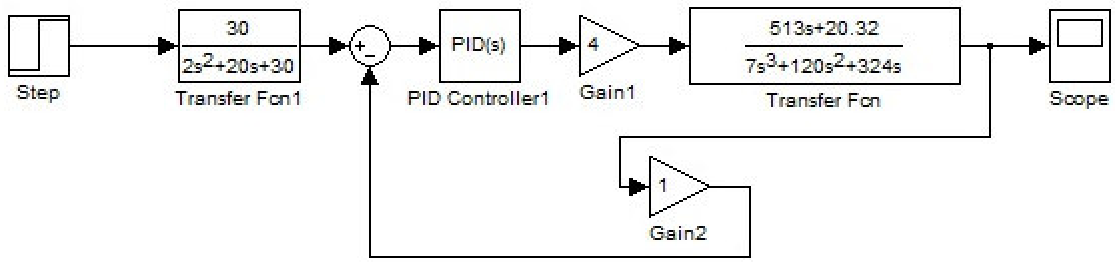

Figure 8: Single loop PID control structure

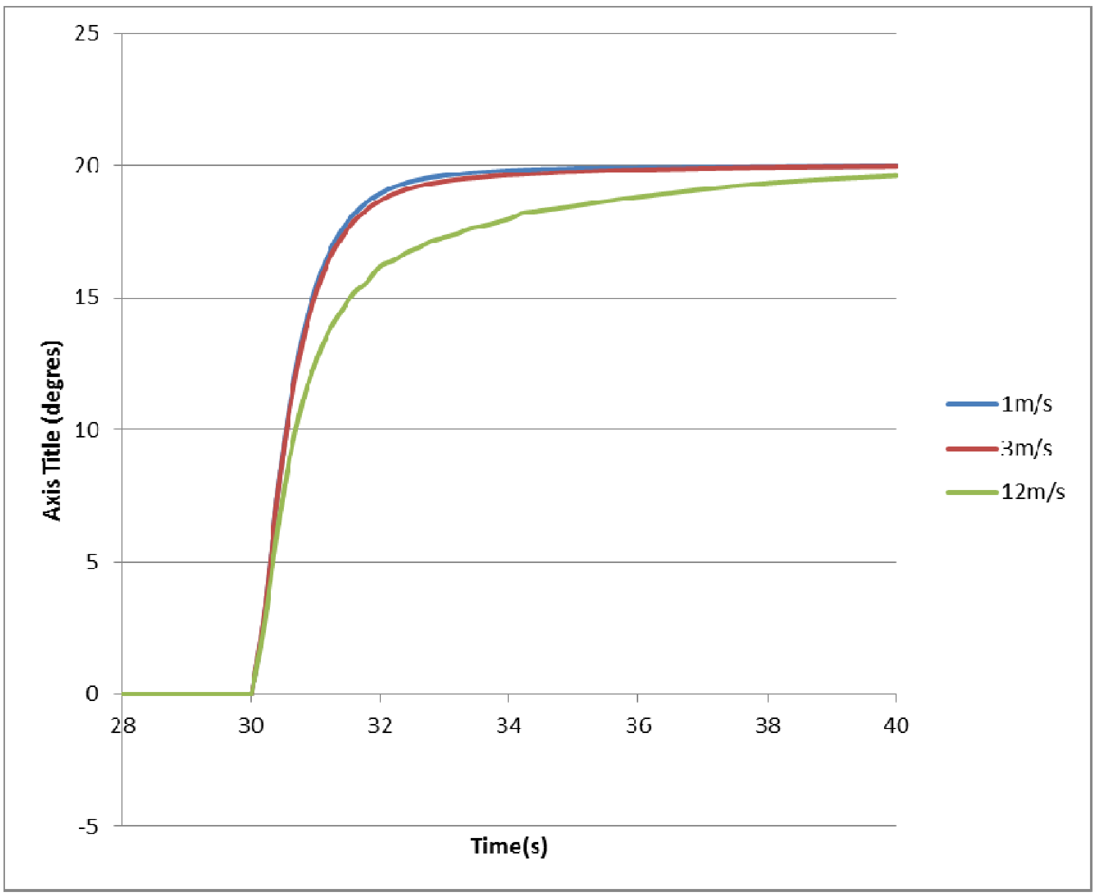

Figure 9: Transient response of the heading angle for single loop PID controller at different speeds

Table 1: Summary of transient response for single loop controller

Speed	Rise time	%Overshoot	Settling time	SS error
1 m/s	1.4 s	0 %	3 s	0
3 m/s	1.5 s	0 %	3.8 s	0
12 m/s	9.8 s	0 %	14 s	0

3.3: Double Loop PD-PID Controller

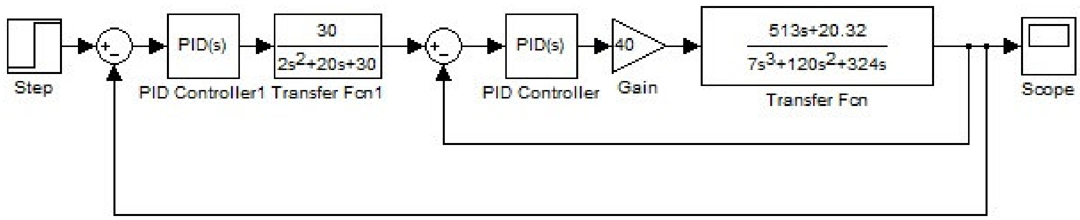

Figure 10: Double Loop Controller configuration

The transfer functions for the four different speeds mentioned previously were incorporated into the model for double loop control strategy. The block diagram of this strategy is presented in figure 10. The transient response at different speeds is presented in figure 11.

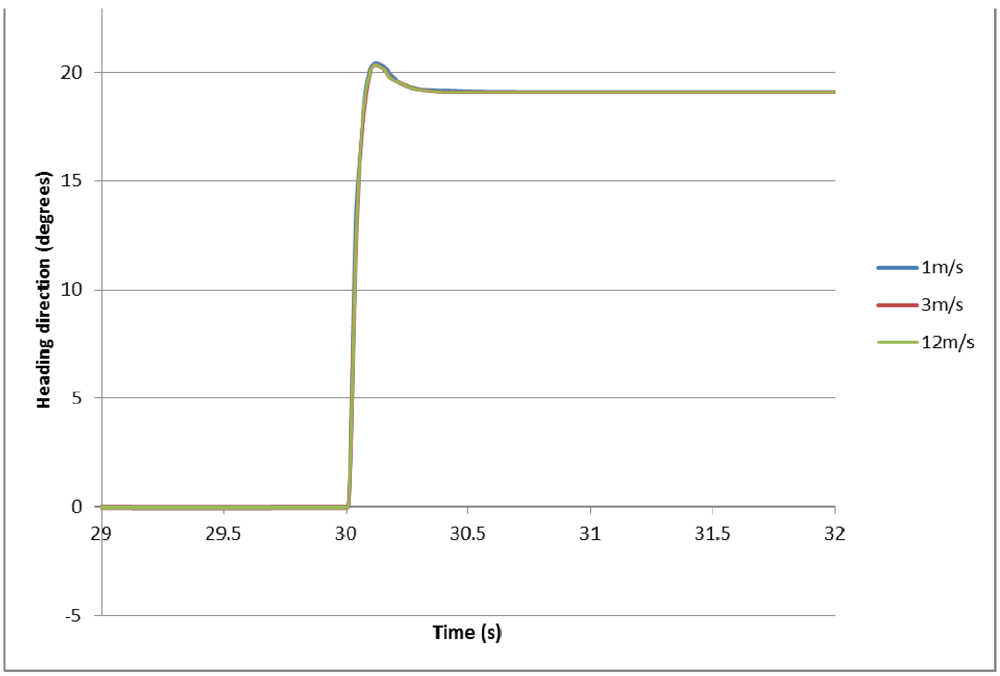

Figure 11: Transient response for double loop control configuration at three different speeds

Table 2: Summary of transient response for double loop controller

Speed	Rise time	%Overshoot	Settling time	SS error
1 m/s	0.25	7.26%	1.2 s	5 %
3 m/s	0.24	8.44 %	1.2 s	5 %
12 m/s	0.21	9.5%	1.2 s	5 %

3.4: Internal Model Controller

The process transfer functionsfor the four different speeds were simulated using the IMC design approach. For this case, perfect model matching has been assumed. It might not simulate the real world characteristics but it shall give us valuable insight to the performance of the IMC for this process. The control block diagram and the graph for transient response are represented in figures 12 and13y respectively.

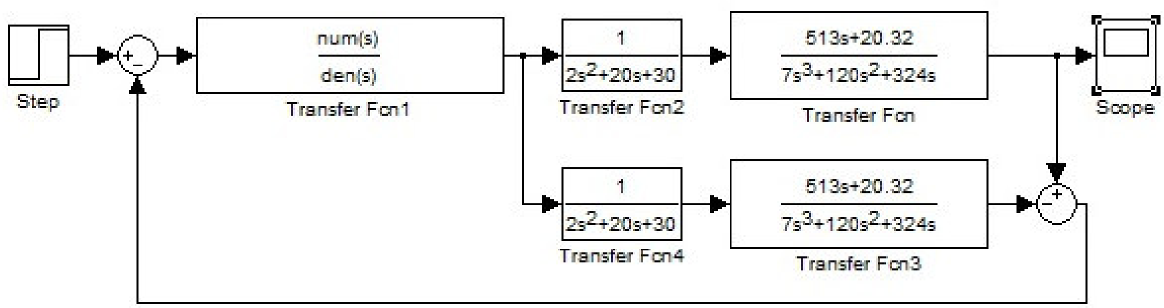

Figure 12: IMC with perfect model

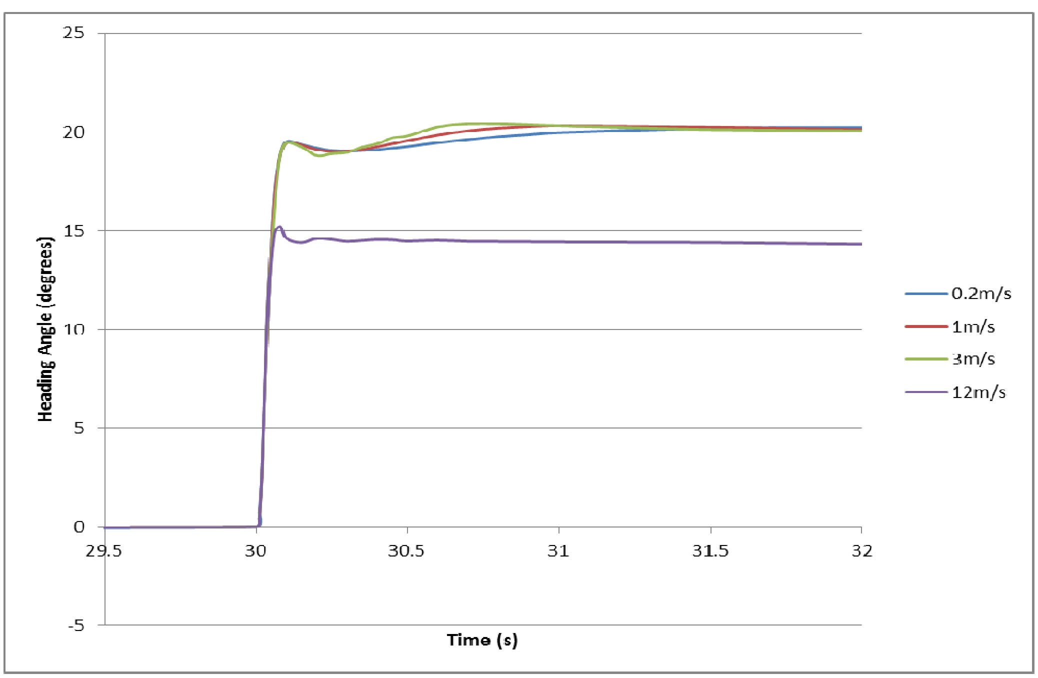

Figure 13: Transient response at different speeds for IMC configuration

Table 3: Table 2: Summary of transient response for IMC

Speed	Rise time	% Overshoot	Settling time	SS error
0.2 m/s	0.05	1.05 %	0.7 s	0
1 m/s	0.05	1.5 %	0.55 s	0
3 m/s	0.05	2 %	0.44	0
12 m/s	0.05	7.74 %	-	32 %

3.5: Modified Internal Model Controller

The process model was also incorporated with the modified IMC scheme to show its performance. The transient response and result is presented as follows.

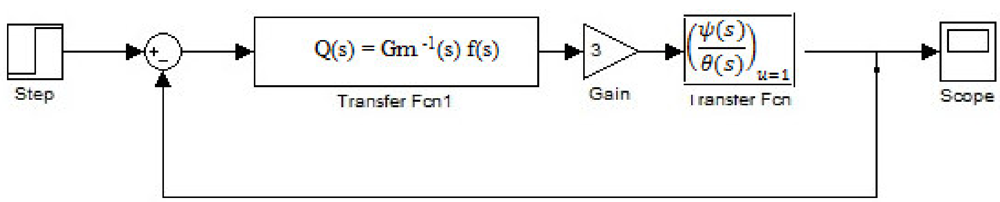

Figure 14: Modified IMC configuration

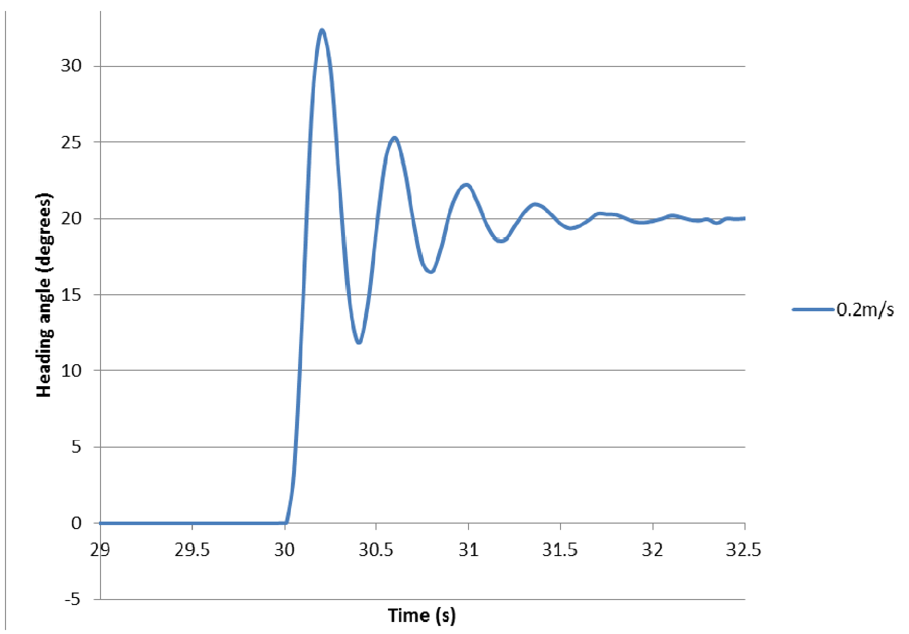

Figure 15: Transient response at 0.2m/s for Modified IMC configuration

Table 4: Summary of transient response for modified IMC

Speed	Rise time	%Overshoot	Settling time	SS error
3 m/s	0.46	38 %	2.5 s	0

3.6: Internal Model Controller with Model Mismatch

The mathematical model of the process cannot always truly represent the process in the practical case. Hence, some model mismatch was incorporated to the IMC design and the performance was evaluated. The model mismatch was a random choice of distorting the coefficients of the process transfer function while keeping in mind that the gain should not be changed. The model mismatch scheme was simulated for different forms and degrees of mismatch.

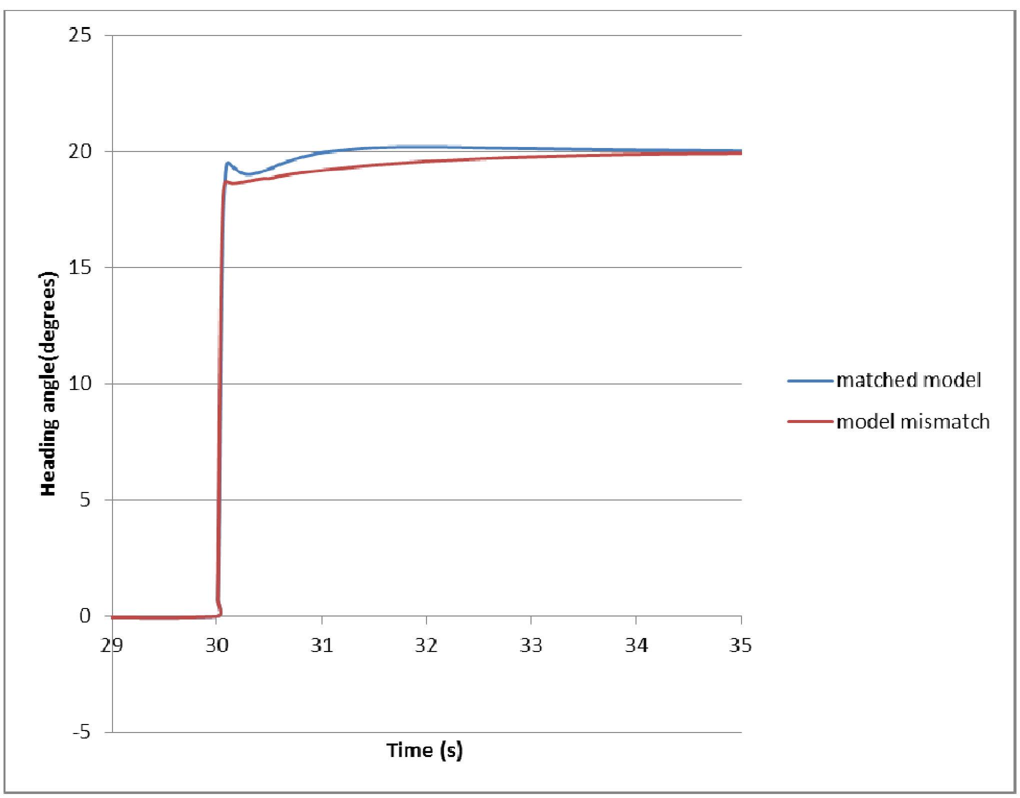

Figure 16: comparision of transient response for matched and mismatched models at 3m/s

Table 5: Comparision of transient response for matched and mismatched models at 3m/s

model	Rise time	%Overshoot	Settling time	SS error
matched	0.05	2 %	0.44 s	0
mismatched	0.08	2.5 %	0.6 s	0

3.7: Two Degree of Freedom Internal Model Controller

The two degree of freedom scheme was simulated for different speeds for matched and mismatched models. The results of these simulations follow.

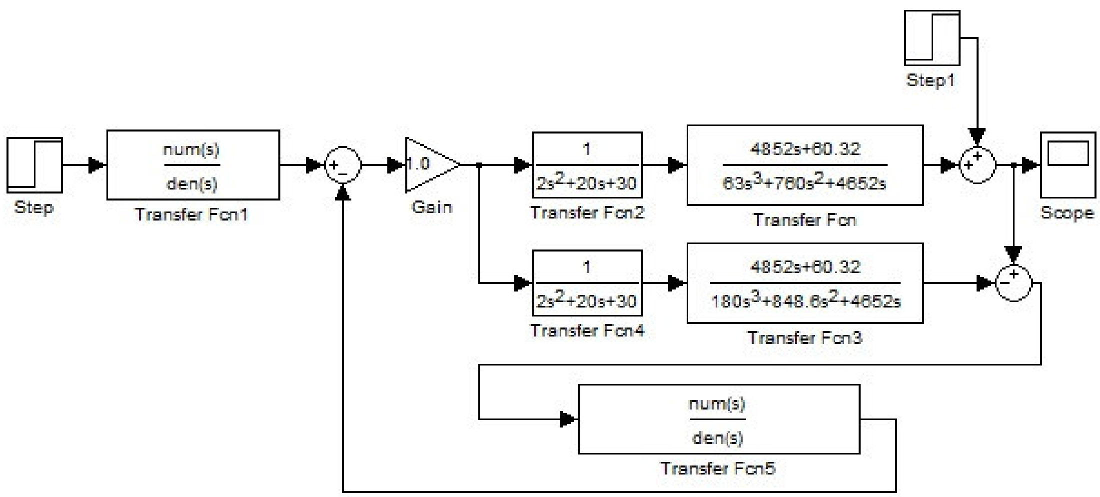

Figure 17: Block diagram for 2 dof IMC configuartion with model mismatch and step error

Here, both controllers have been assigned the same transfer function for simplicity

$$Q_1(s) = Q_2(s) = \frac{126s^5 + 2780s^4 + 24504s^3 + 115840s^2 + 139560s}{0.0000485s^5 + 0.019409s^4 + 2.911441s^3 + 194.1162s^2 + 4854.413s + 60.32}$$

Figure 18: Transient response for 2 dof IMC with severe model mismatch and noise as a step function at 3 m/s

Table 6: summary of 2 dof IMC transient performance

speed	Rise time	%Overshoot	Settling time	SS error	Error elimination
3 m/s	0.4s	12 %	1.5 s	0	0.5 s

CHAPTER 4

DISCUSSIONS

4.1: Brief Summary of Results

The study of all the responses from various types of controllers that were implemented led us to believe that there is some amount of trade off in any kind of control strategy we use. Depending on the requirement, we may choose to use any controller. But among all the configurations implemented, the double loop controller and the IMC configuration showed the most satisfactory results. Table 7 shows a comparision of transient response of all the control strategies at 3 m/s.

Table 7: comparision of transient response for different control strategies at 3 m/s

model	Rise time	%Overshoot	Settling time	SS error
Single loop	1.5 s	0 %	3.8 s	0
Double loop	0.08	2.5 %	0.6 s	5 %
IMC with mismatch	0.08	2.5 %	0.6 s	0
Modified IMC	0.46	38 %	2.5 s	0
2 dof IMC with severe mismatch	0.4s	12 %	1.5 s	0

As can be seen from table 7, the performance of double loop controller is very good, in fact, better than some of the more complex strategies. The IMC configuration shows preferable performance over double loop controller. It gives almost similar transient response, while also elimination the steady state error. It shows a good level of robustness to model mismatch and disturbances. The advantage of double loop controller is that it is much simpler to implement than the IMC both in terms of hardware as well as software. However, it is associated with some steady state error. The inner loop contains a PID controller and the outer loop contains a PD controller. The steady state error may be eliminated by using PID controller in the outer loop as wellbut it would make the response much slower than it is now.

4.2: Proposed controller design based on Internal Model Controller strategy

The model of the process changes with the speed of the vehicle. Since the IMC is a model based controller, its performance would degrade if it does not run at the modelled speed. One possible solution would be to tune the controller at the worst case scenario, i.e., the highest operating speed of the vehicle. In this way, the vehicle would be at its best configuration at the highest speeds and it is expected to give satisfactory performance at lower speeds because the process itself would be much more stable at lower speeds.

However, we believe that a much better performance can be achieved if the controller is made adaptive to speed changes. A simple strategy would be to make a look up table of the controller coefficients and incorporate them in the controller equation at the appropriate speeds. This provides a simple adaptive strategy which could be effective. The idea is to select the intervals of speed cantered around a set of speeds for which the controller is designed. Since the controller is seen to be robust for small changes in speed, it would not degrade it performance for a very wide range of speeds. The quantum of speed value for which the adaptive mechanism is built would determine the robustness. More is the number of discreet speeds for which the look up table is constructed, better will be the transient response.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Conclusion:

For any control application, the identification of the system model is the most critical to developing any strategy to it. While we can go to great lengths to approximate the model as accurately as possible, but the real world constraints and disturbances can never be fully represented mathematically. So the control strategy needs to be robust to handle the model error and disturbances. While generic controllers like classic feedback controllers can provide a high degree of robustness, they may not provide satisfactory behaiviour for fast and complicated processes. In such cases, the model based controllers are more useful to handle the complexity of the process. From the results presented in this document, it was seen that the model based controller could provide good transient response along with zero steady state error. It was also robust to a good degree. This model based strategy could be made more robust by incorporating some suitable adaptive strategy such as the one briefly described in the previous section.

Further work in continuation of this research would be to validate the adaptation strategy with simulations and quantifiable results, and to propose a more suitable strategy for adaptation of the controller. One may even look to design a static controller using the H-infinity methods which would require deep mathematical understanding. Use of fuzzy logic, neural networks and artificial intelligence would be a step further to revolutionise the field of control engineering. However, they have not yet been developed to be implemented in a simpler or smaller scale. With a high level of research devoted to develop robust general purpose controllers applicable to a wide variety of processes, we may see the most complex processes controlled using simple modifications.

BIBLIOGRAPHY

References:

[1] Ian Schworer, "Navigation and Control of an Autonomous Vehicle", Virginia Polytechnic Institute and State University Master thesis.

[2] HorwimanpornSuppachai, ChaiyapornSilawatchananai, ManukidParnichkun, ChairitWuthishuwong "Double Loop Controller Design for the Vehicle's Heading Control", Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics December 19 -23, 2009, Guilin, China.

[3] Carroll Smith, "Tune to Win", Aero Publishers, Inc. 1978

[4] S.Chaiyaporn. "System Identification of an Electrical Car for Steering and Speed Control." Asian Institute of Technology Master Thesis, Dec 2007.

[5] Srinit Das, "Preliminary Design Report, Black Mamba Racing", BAJA SAE India 2013.

[6] Marcel Bergerman, OmeadAmidi, James Ryan Miller, Nicholas Vallidis, Todd Dudek, "Cascaded Position and Heading Control of a Robotic Helicopter", Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29 - Nov 2, 2007

[7] GhassanMurad , Da-Wei Gu& Ian Postlethwaite (1997): On the design of robust internal model-based controllers, International Journal of Control, 68:3, 513-538

[8] Walid R. Issa, "A New Approach of Robust Internal Model Control for Nonlinear Pendulum System." Islamic University of Gaza master thesis.