Prototype Development and Testing of Active Yaw Moment Control System Using Vehicle Momentum Wheel

by

ANKUR ARORA

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Applied Science
in

Faculty of Engineering and Applied Science
Department of Automotive, Mechanical and Manufacturing Engineering

University of Ontario Institute of Technology

January 2015

I dedicate this thesis to my beloved parents who have always been very supportive and encouraging.

Abstract:

Vehicle stability is the essential key to evade accidents and drive without harm. It is in actual an extremely dynamic and active research topic nowadays. Active yaw moment control to improve vehicle's handling and stability has been chosen as the control strategy. This technique uses a momentum wheel control (MWC) independent of the tire/road interaction to generate the corrective yaw moment. A fully functional prototype of the control system on an RC model car with onboard processing capabilities is designed and developed. Vehicle dynamics analysis and the test results are discussed.

Control system is designed using the Linear-Quadratic Regulator (LQR) theory, calculates the necessary torque required to apply to the vehicle chassis in order to generate the corrective yaw moment. Kalman filter is used to supress the sensors noise and any irregularities in the system model design. Motor controller board provides enough current to generate the required torque. All the calculations are done using Advanced RISC Machines (ARM) microcontroller. System dynamics are analysed on the computer using LabVIEW software.

Acknowledgment:

I would like to express my deepest appreciation to my supervisor Professor Ebrahim Esmailzadeh for his guidance and providing me the opportunity to work with his research team.

In addition, I am very thankful to Dr. Jing Ren for helping me out with the problems that I had using the Kalman filter.

I wish to express my sincere thanks to Dr. Fereydoon Diba for his helpful suggestions and assistance given to me throughout my research work.

I am indebted to Mr. Dave Mackay and Durham College to lend me the measuring equipment for the testing and configuration of the developed prototype.

I would like express my eternal appreciation to the Manager of Technical Services, Mr. Hidayat Shahid, and the UOIT machine shop members for the technical support and co-operation.

Last but not least, I greatly appreciate the research funding and support provided by the Natural Science and Engineering Research Council of Canada (NSERC) to pursue this research work.

TABLE OF CONTENTS

List of Figures:

List of Tables:

Nomenclature:

A	Amplifier/Ampere
ACK	Acknowledge Signal
AD	Slave Address
a	Distance Between Front Axle And COG
B	Magnetic Flux Density
b	Distance Between Rear Axle And COG
c_f	Front Tire Stiffness
c_r	Rear Tire Stiffness
$DATA$	Transmit/Receive Data
DIN	Input Data
$DOUT$	Output Data
F_{xrr}	Longitudinal Force At Rear Right Tire
F_{yrr}	Lateral Force At Rear Right Tire
F_{xrl}	Longitudinal Force At Rear Left Tire
F_{yrl}	Lateral Force At Rear Right Tire
H	Magnetic Field
Hz	Hertz
I_d	Momentum Disc's Moment Of Inertia
I	Vehicle's Moment Of Inertia
I_c	Current

IMU_{mag}	Magnitude Of Motion
IMU_{thr}	User Defined Threshold To detect Motion
k	Time Step
K_v	Lateral Velocity Gain
K_r	Yaw Rate Gain
$K_{r'}$	Momentum Disc Velocity Gain
K_δ	Steering Angle Gain
K_{us}	Understeer Gradient
l	Wheel Base
M	Mass Of Vehicle
$NACK$	Not Acknowledge Signal
Nm	Newton Meter
p	Pitch
q	Roll
R	Read Command
RA	Master Address
R_H	Hall Effect Co-efficient
r	Yaw Rate
r'	Momentum Disc Angular Rate
r_d	Desired Yaw Rate
r_y	Yaw
S	Feed Forward Gain Matrix / Start Condition

T	Torque Required
T_d	Momentum Disc's Torque
T_{ON}	Time While Signal Is High
T_{OFF}	Time While Signal Is Low
T_r	Distance Between Left And Right Tire
t	Thickness
V_{AVG}	Average Output Voltage
V_d	Desired Lateral Velocity
V_H	Hall Voltage
v_y	Lateral Velocity
$\hat{v}_x$	Longitudinal Velocity Estimate From Principle Observer
$\hat{v}_y$	Lateral Velocity Estimate From Principle Observer
$v_{x,int}$	Longitudinal Velocity From Pseudo Integration
$v_{y,int}$	Lateral Velocity From Pseudo Integration
$\hat{v}_{x,int}$	Longitudinal Velocity Estimate From Pseudo Integration
$\hat{v}_{y,int}$	Lateral Velocity Estimate From Pseudo Integration
W	Write Command
w_1	Weighing Factor For Vehicle Yaw Rate
w_2	Weighing Factor For Lateral Velocity
w_3	Weighing Factor For Steering Angle
w_4	Weighing Factor for Momentum Disc Yaw Rate
X	x-Axis

Y	y-Axis
Z	z-Axis
δ	Steering Angle
δ_m	Measured Steering Angle
δ_d	Desired Steering Angle
δ_e	Error Steering Angle
δ_c	Corrected Steering Angle
ϕ	Roll Angle
θ	Pitch Angle
β	Side-Slip Angle
Y	y-Axis
Z	z-Axis
$\emptyset$	Diameter
“	Inches (Unit Of Length)

Abbreviations:

4WAS	Four Wheel Active Steering
AFS	Active Front Steering
ATD	Active Torque Distribution
ADB	Active Differential Breaking
ABS	Anti-lock Breaking System
ARM	Advanced RISC Machines
ADC	Analog to Digital Converter
AEC	Automotive Electronics Council
AMP	Amplifier
ASC	Active Steering Control
ARS	Active Rear Steering
ASCII	American Standard Code for Information Interchange
AWD	All-Wheel Drive
BEC	Battery Eliminator Circuit
BPS	Bits Per Second
CMOS	Complementary Metal Oxide Semiconductor
CTS	Clear to Send
CS	Chip Select
COM	Component Object Model
DYC	Direct Yaw Moment Control

DI	Dynamic Inversion
DOF	Degree Of Freedom
DMA	Direct Memory Access
DC	Direct Current
DMP	Digital Motion Processor
DLPF	Digital Low Pass Filter
ECU	Electronic Control Unit
ECS	Electronic Control System
ESC	Electronic Speed Control
EKF	Extended Kalman Filter
FIFO	First In First Out
FET	Field effect Transistor
GPS	Global Positioning System
GND	Ground terminal
GUI	Graphical User Interface
HIGH	High logic
HEX	Hexadecimal code
IMU	Inertial Measurement Unit
I/O	Input/ Output
I2C	Inter IC serial protocol
INT	Interrupt
IC	Integrated Chip

IDE	Integrated Development Environment
LQR	Linear Quadratic Regulator
LDO	Low DropOut regulator
LSB	Least significant Bit
LOS	Line Of Sight
LED	Light Emitting Diode
LPF	Low Pass Filter
LOW	Low logic
MPC	Model Predictive Control
MPU	Micro-Processor Unit
MEMS	Micro-Electro Mechanical System
MSB	Most Significant Bit
MOI	Moment Of Inertia
MWC	Momentum Wheel Controller
NIMH	Nickel-Metal Hydride
PCB	Printed Circuit Board
PID	Proportional-Integral-Differential
PWM	Pulse Width Modulation
PPR	Pulse Per Revolution
PC	Personal Computer
QFN	Quad Flat No-leads package
RAD	Rear Active differential

RPM	Revolutions Per Minute
RC	Radio Controlled/ Resistor Capacitor
RTS	Ready To Send
RF	Radio Frequency
RSSI	Received Signal Strength Indication
SMC	Sliding Mode Control
SPI	Serial Protocol Interface
SDL	Serial Data Line
SCL	Serial Clock Line
SOC	State Of Charge
TCS	Traction Control System
USB	Universal Serial Bus
UART	Universal Asynchronous Serial Bus
UKF	Unscented Kalman filter
UT	Unscented Transform
WSN	Wireless Sensor Network

Chapter 1: **Introduction**

1.1. PREAMBLE

Handling performance and stability are the two most important factors that influence the safety of vehicle especially at high speed. Topics to improve both the handling and stability of vehicle are leading focus in the automotive research industry. Numerous technologies with variety of control systems have been proposed to increase stability and improve the handling of the vehicle and new technologies are being developed every day.

Core objective of the vehicle stability control is to keep the vehicle on the road on the desired path. So far yaw moment control proves to be the most effectual way to heighten the vehicle stability. Conversely, yaw moment control comprises large number of uncertainties and nonlinear behavior because of the steering angle/steering angle rate, tire slip angle, friction forces, and tire lateral forces. Most of the above mentioned factors depend on the road conditions and bounds the control system performance. According to these facts, this thesis considered the enhancement of the vehicle stability using yaw moment control system which is independent from the tire/road interaction. The momentum wheel system as an independent yaw moment control system has previously been introduced and the performance of this system has been investigated using computer simulations [1]. In this thesis, the performance of the momentum wheel system has been studied and evaluated using experimental testing.

In this chapter diverse yaw moment controls strategies are discussed in brief and then compared with the momentum wheel control systems.

1.2. LITERATURE REVIEW

Different methods and control techniques for yaw moment stabilization of the vehicle are reviewed. Brief discussion is presented in the following section.

1.2.1. Yaw Moment Control Methods

A brief review on the different yaw moment control techniques is presented in this section. Generally, there are two basic yaw moment control method, direct and indirect as shown in Figure 1.1. The corrective yaw moment can be applied using one of two basic methods or both methods.

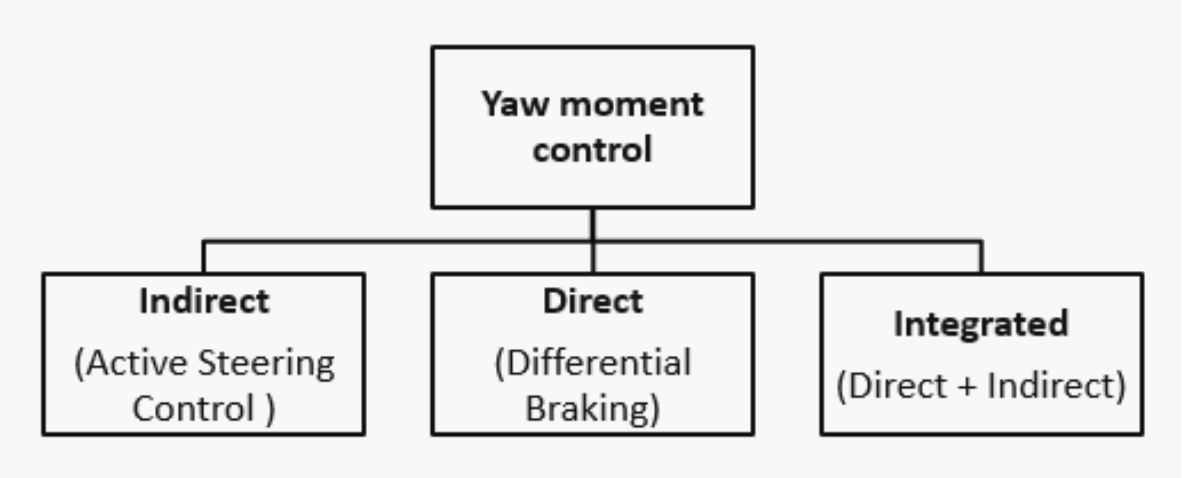

Figure 1.1: Yaw moment control methods.

Indirect method of yaw moment control modifies the vehicle steering angle in order to improve the vehicle's total yaw moment. Whereas, in direct method break force is applied directly to suitable combination of wheel to avert the vehicle form losing its stability. In the last scheme

combination of the both can be used, which integrates advantages of both the techniques and counter balance the downsides of each other and hence improve system performance. Result is significant improvement in system response and wide range of operation [2].

1.2.1.1. Active Steering Control (ASC)

Conventionally, the only way to control lateral force of the vehicle was steering wheel directed by the driver [3]. Traditional steering systems did not have variable steering ratio as shown in Equation 1.1, which means driver need to apply more force to turn the steering wheel at lower speed than when the vehicle is at high speed. Furthermore, vehicle becomes unstable at the high speeds due to this phenomenon [4].

$$Steering\ Ratio = \frac{Angle\ of\ steering\ wheel}{Angle\ of\ wheel} \tag{1.1}$$

In ASC control technique relation between the steering wheel angle and the vehicle steering angle is no more fixed. ASC continuously changes the steering ratio to improve the vehicle yaw moment. This improves safety, performance and, vehicle handling especially in emergencies and obstacle avoidance [4, 5].

Another advantage of the ASC control technique is that it modifies the actual vehicle steering angle without moving the steering wheel position, which is less intrusive for the driver since it does not affect the longitudinal vehicle dynamics [6].

Brief detail about the control structure for ASC is explained below in Figure 1.2.

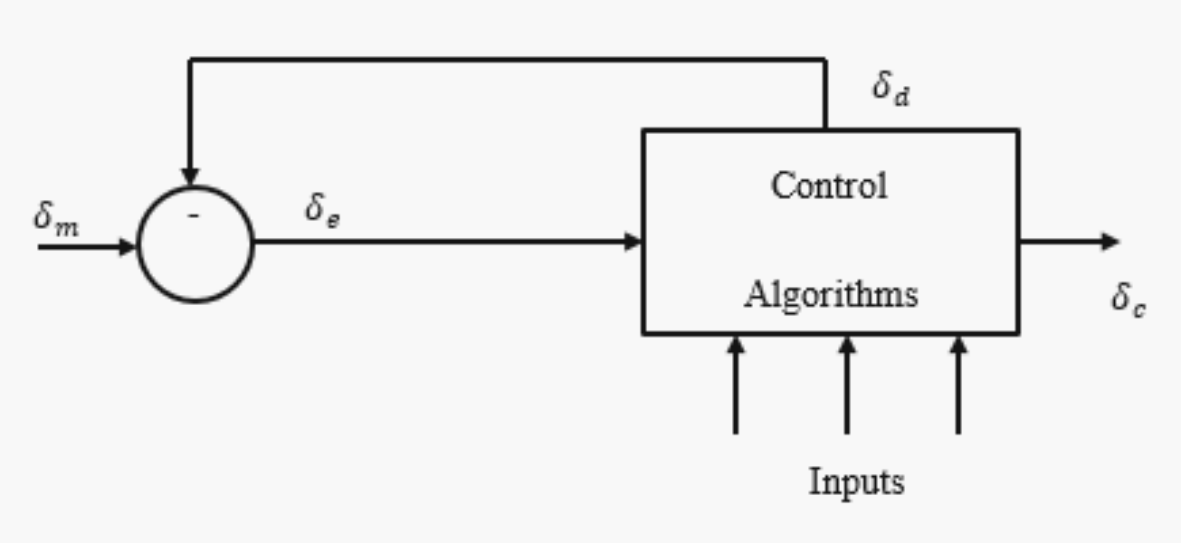

Figure 1.2: Generalizes ASC control system

Where, δ_m is measured angle, δ_d is desired angle, δ_e is error angle and, δ_c is corrected angle. Based on the input variables and measured steering angle (δ_m), control algorithm calculated the desired angle (δ_d) and then feed the error signal (δ_e) into the controller. Controller calculated the corrected angle (δ_c) that is then applied to the steering wheel.

ASC has been implemented constantly to improve the system handling and stability performance as stated in [7-9]. In general, ASC can be implemented using three techniques as shown in the Figure 1.3. Active front steering (AFS) [4-6, 10] delivers great performance during the steady state conditions whereas, Active rear steering (ARS) [11-13] improves system response in the low speed cornering maneuvers. Combining AFS and ARS increases the system stability at high speed and manoeuvrability at low speeds [14]. Results published in [15-17] with 4WAS control strategy shows a significant improvement in the system performance.

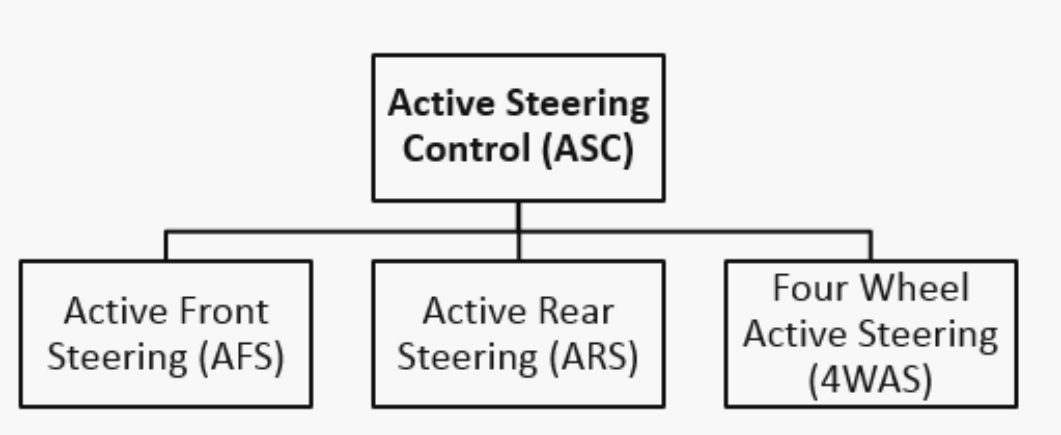

Figure 1.3: Active steering control methods

Superposition of the active steering angle changes the vehicle's lateral tire forces in order to control the yaw moment and works very well during the linear dynamic conditions or normal driving conditions. In case of high lateral acceleration maneuvers, when slip angles go beyond the saturation limits, steering input has no more effect on the tire lateral forces [11]. Also, there are constraints on steering angle and steering rate with using this technique to control the yaw moment of the vehicle, which limits the system performance into a finite area and system cannot provide major performance in all the situations [18].

This condition makes ASC control system unfeasible and forces us to use second method to control the yaw moment.

1.2.1.2. Direct Yaw Moment Control (DYC)

Direct yaw moment control (DYC) method modifies vehicle's yaw rate directly instead of improving other parameters that in turn affects vehicle's yaw rate. DYC can be implemented using two methods. First method is to apply required breaking forces on tires in order to make vehicle stable on the road, this method is called Active breaking [19-22] or by using the active differential

torque distribution technique [14]. DYC improves vehicle stability and performance by altering the longitudinal forces of the left and right wheel to make sure that vehicle produces correct yaw moment and lateral motion [23]. Use of DYC system ensures that vehicle track a reference yaw rate.

In [24] vehicle fitted with a rear active differential (RAD) device with a feed forward-feedback control scheme is considered, which raise the system behaviour beyond the saturation limit of yaw moment provided by RAD.

Use of all wheel drive (AWD) torque distribution provides stability under the conditions where acceleration approaches the limit of stability. In [25] author proposed a technique in which distribution of front–rear torque is considered as a function of yaw rate and left-right torque distribution is a function of lateral acceleration. While, [26] presents how torque distribution control system can further be improved by considering the effect of coefficients of road surface every instant.

There are constraints on the achievable yaw moment because of the allowable breaking torque and tire slip ratio. At high speeds value of steady state yaw rate decreases and adds burden of the driver in cornering and put serious constraints in the vehicle performance [27]. This leads us to discuss the third method.

1.2.1.3. Integrated (ASC + DYC)

ASC provides comfortable and reliable performance during the low speed operations while its performance degrades at higher speed. On the other hand, DYC does not perform efficiently at low speed due to the lower yaw rate, AFS can counter balance this limitation of DYC. Combining both techniques can improve system performance and safety [2]. In [28] active torque distribution

(ATD) is used with active differential breaking (ADB). In [12] ARC and Active breaking has been used with the model matching control theory, which assures stable system performance during the nonlinear region of the tire friction circle.

Author in [27] uses a fuzzy logic approach together with AFS and active breaking to control the yaw rate and Side-slip angle synchronously and hence increase the transient and steady response of vehicle.

1.2.2. Controller Design

Several control methodologies can be applied to control the yaw moment of vehicle. Some of the relevant control methodologies are reviewed in this section.

Classical control methods does not provide enough system performance and robustness due to the assumptions in the system model design [29]. Now a days advanced control approaches can be applied due to the development in the electronic control Units (ECU), as well as abilities and handiness of vast variety of sensors (GPS, accelerometer, and gyrometer).

Fuzzy control has been widely used in the active control system of vehicle such as Anti-lock Brake System (ABS), Traction Control System (TCS) and, so on [30]. The complex nonlinear systems with uncertainty can be easily described using fuzzy control. As the dynamics of the vehicle is a highly nonlinear problem, fuzzy logic control has been used widely to modify and control the dynamic behaviour of the vehicle [29]. In Ref. [30] Fuzzy control method integrated with AFS and DYC is used to improve the yaw stability. System takes steering angle, vehicle velocity as an inputs and yaw rate, side-slip as feedback variables. In Ref. [27] Fuzzy controller ensures side-slip and yaw rate follow their desired value by monitoring steering angle and velocity.

Model Predictive Control (MPC) techniques has been used in [6, 18, 31]. MPC includes multiple constrained actuators to increase overall system stability with respect to a predefined performance criterion. MPC controllers can be time varying [32], hybrid [33] and, real-time[34]. In all of the mentioned MPC systems the controller is assumed to be the only element to actuate the steering. In Ref. [18] MPC technique with driver-assist steering has been discussed, where the vehicle dynamics are generated by the combined effect of driver action and AFS action.

In Ref [35] author presented the advantages of using robust H_∞ controller for DYC system. Using the proposed method, results showed improvement in the robustness of the system. The H_∞ Worked effectively because it considered uncertainties such as parameter perturbation and model perturbation, which affects the vehicle handling and stability. Another method that uses the combination of H_∞ /H_2 controller with AFS and DYC claims better performance than with H_∞ and increased robustness stability resisting outside noise [23].

Sliding Mode Control (SMC) technique with AFS and DYC has been discussed in Ref. [3, 4] Considering system parameter uncertainty and using variable structure control vehicle stability is further improved. In Ref [36] downsides of SMC are eradicated using a second order SMC. Second order SMC produces continuous control action, consequently removing chattering problem. Yet, still maintain the accurateness and robustness of SMC.

Another controller design using Mu-controller with AFC, incorporating control system errors is presented in [37]. Weighting functions in Mu-controller allows to filter the measurements and decrease the effects of external disturbances. Further, objective function is optimized using minimization of these errors hence, providing great yaw rate tracking performance and high rejection ratio to external disturbances.

A new control technique using logic threshold control and PID control is used in Ref. [38]. Combining fast characteristics of logic threshold method and simple structure of PID control improves system performance.

Another emerging control theory has been discussed in Ref. [39] which is called Dynamic Inversion (DI), a type of nonlinear control technique. DI can be adopted in the systems which operate in different environment conditions without any gain scheduling. DI improves the system performance by replacing plant dynamics into user defined dynamics.

1.3. THESIS OUTLINE

Chapter-2 includes the details about the considered vehicle model, design method of the LQR controller used to calculate required torque, brief data regarding the electronic modules integrated into the developed prototype and, various steps involved in development and modification process of the vehicle prototype.

Chapter-3 holds thorough information on each hardware module included in the developed prototype. Every module is explained in different section with the information about the circuit details, working principle, hardware specifications, connection diagram, and module configuration (if any) and, communication protocol used.

Chapter-4 explains the software used in the thesis. Embedded programmer section explains how to program the embedded controller, Wireless transceiver section has the steps to configure the wireless module according to the user requirement, On-line and post processing section gives brief about the LabView and includes the GUI of both the developed programs.

Chapter-5 discusses the importance, design and implementation of the Kalman filter together with pseudo integral technique that is used to estimate the lateral and longitudinal velocities. This chapter also includes the LQR implementation and the calculation of LQR gains. PI controller as momentum wheel controller is explained.

Chapter-6 describes the testing criteria and the experimental results.

Chapter-7 contains the conclusions and the future work suggestions.

Chapter 2: **Proposed Yaw Moment Safety System**

Performance of all the yaw moment control methods that have been discussed in the previous section, highly depends on the tire and road interaction. For example, DYC performs very well when there is equal tire-road frictions coefficients on both side of the vehicle but the performance degrades on μ-split conditions and reduces the ride comfort by applying sudden forces [31]. Similarly, AFS works effectively only in the low lateral accelerations [40]. Also, both the techniques have some constraints on the control actions.

In all the discussed techniques, there is no control which is independent of the road conditions. Producing correct yaw moment through the external source would be a more effective way to control the dynamic behavior of the vehicle. Techniques using three independent controlled moving mass have previously been applied to satellites for the three-axis stabilization of the satellites. Independent controlled masses allows satellites to change orientation manoeuvers. Some of the methods are discussed in [41, 42]. Possibility of importing this technique to modify the vehicle dynamics have been previously studied and introduced [43] [44].

In this thesis, the yaw moment control system using momentum wheel system has been studied and the effectiveness of this technology has been experimentally investigated. In order to test the momentum wheel system, a test prototype has been developed and implemented based on the system design and control method proposed in Ref. [1]. Following section provides more in depth details about the prototype development and controller design.

The control system's diagram of the developed prototype is shown in Figure 2.1. System comprises of mainly four major sections. Noise filter in the system is a Kalman filter that takes steering angle (δ) and momentum disc velocity (r') as an inputs, which filters out the measurement errors and prepare the signal to be used in the next stage. Because it's not viable to measure lateral velocity of the vehicle directly, estimator is used for this purpose [45]. Input to the estimator is longitudinal acceleration (a_x), lateral acceleration (a_y), vertical acceleration (a_z), pitch (p), roll (q), yaw (r), gravity constant (g) and, longitudinal velocity (v_x). After processing all the variables next step is to calculate the torque required to control the yaw moment of the vehicle. Linear-Quadratic Regulator (LQR) controller is selected to calculate the required torque [46]. Last block is momentum wheel controller (MWC) which uses PI controller to interrelate the required torque (T) to the required DC motor voltage. Each section is described comprehensively in this section.

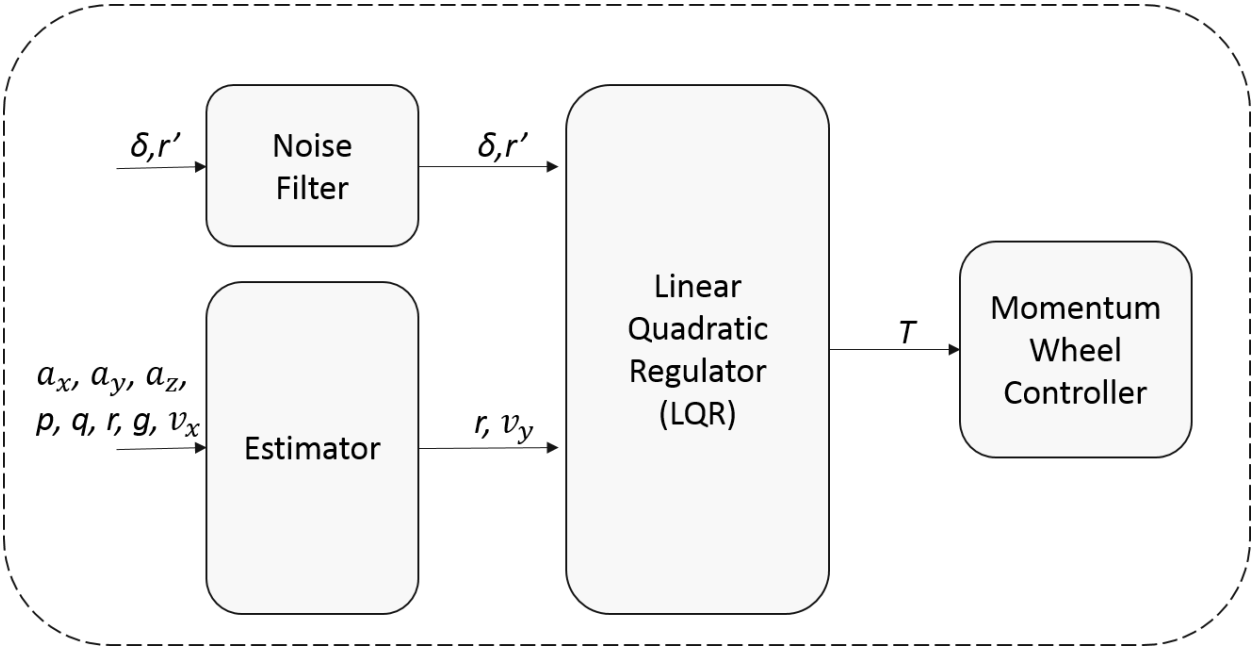

Figure 2.1: Control system diagram

Chapter-2 has four sections, in the first section vehicle model considered to be used for the design of LQR controller has been discussed. In second section, controller design theory and calculations (LQR) to calculate gains are explained in detail. Third section provides a brief about the electronic design and the core modules used to acquire vehicle dynamics. Last section talk over the developing and modifying process of the RC model car.

2.1. VEHICLE MODEL

Control system design is developed based on famous linear bicycle model, to make the calculations less complex.

Linear vehicle model with 3 DOF is developed by adding a momentum wheel in the rear end of vehicle is modeled as shown in Figure 2.2. The degree of freedom of this model are vehicle yaw rate (*r)*, vehicle lateral velocity (v_y) and, momentum wheel angular rate (*r'*).

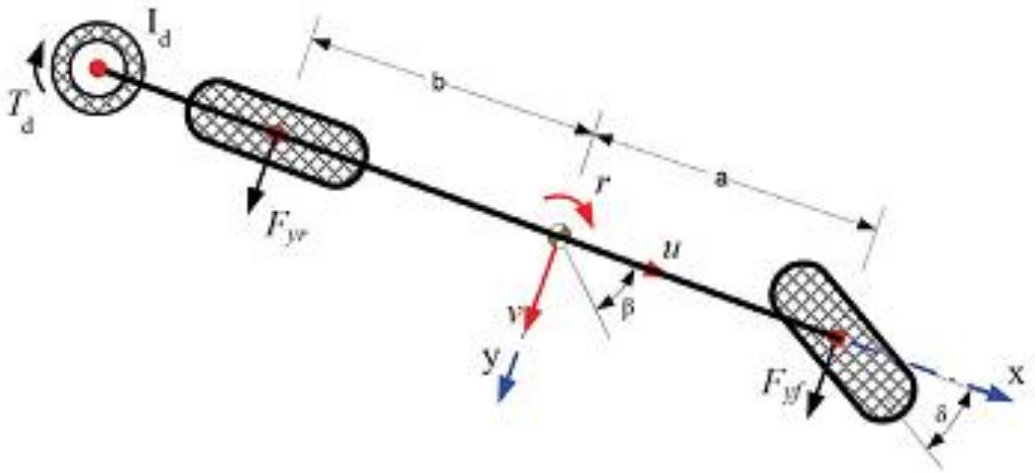

Figure 2.2: Linear vehicle model

Momentum wheel system consist of a disk with moment of inertia (I_d) and a rotary actuator. The dynamics of the rotary actuator are neglected in the system calculations. The torque (T_d) generated by the rotary actuator would be applied to the vehicle's body in order to produce corrective yaw moment. It is controller's responsibility to calculate this actuator torque. Steering angle (δ) is considered as an external disturbance (control input signal).

2.2. CONTROLLER DESIGN

The reaction torque of the momentum wheel actuator will generate the corrective yaw moment. Linear-Quadratic regulator (LQR) control technique is applied to calculate the rotary actuator torque *T*. The block diagram of the control algorithm is shown in Figure 2.3, where, *r* is yaw rate, *r'* is rotational speed of momentum wheel, δ is steering angle, $v_y$is lateral velocity of the vehicle and, v_x is the longitudinal velocity of the vehicle.

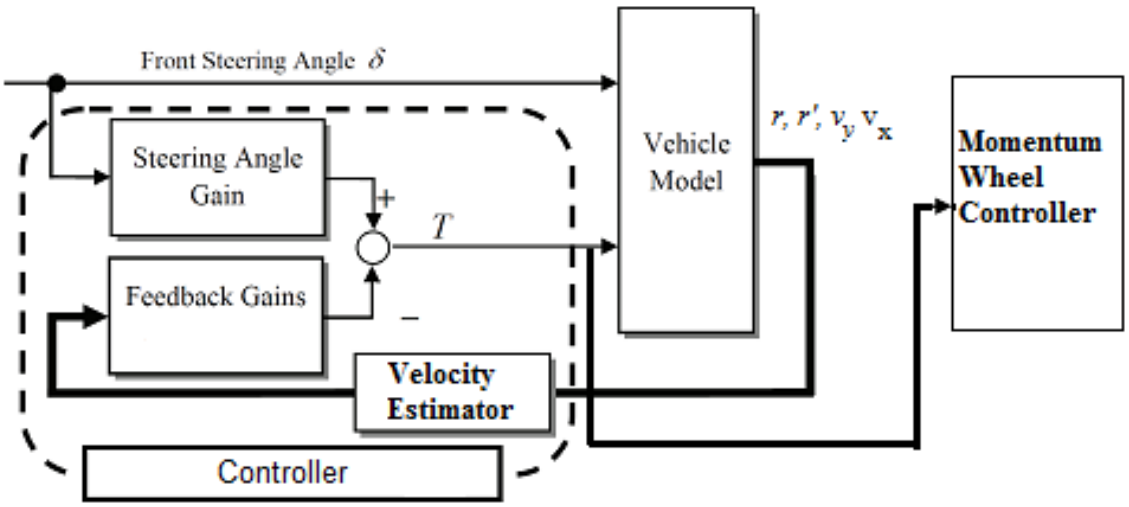

Figure 2.3: Block diagram of control algorithm [44].

Above mentioned variables except v_y and v_x, can be easily measured directly using the standard sensors. To measure the lateral and longitudinal velocities, an online estimator is paired with the control design. Controller consists of the feed forward signals and feedback variables. Feed forward signal related to the input steering angle. Whereas, three feedback signals are yaw rate, lateral velocity and, momentum wheel speed. Desired torque is calculated using Equation 2.1 [1].

$$T = K_v.v_y. + K_r.r + K_{r'}.r' + K_\delta.\delta \tag{2.1}$$

where, K_v, K_r, $K_{r'}$ and, K_δ are the controller gains used to calculate the resulting torque. These gains are optimized by the LQR controller. Main objective of the controller is to minimize the difference between desired and actual control signal. The objective can be formulated using the performance index of the controller, which is a minimizing function [1] shown in equation 2.2. Angular velocity of the momentum wheel is assumed to be zero in the performance index function to make system energy efficient.

$$I_y = \frac{1}{2}\int_0^{t_f} [w_1(r - r_d)^2 + w_2(v - v_d)^2 + w_3T^2 + w_4r'^2]dt \tag{2.2}$$

where, w_i is the weighting factors, r_d is the desired yaw rate and v_d is the desired lateral velocity. Weighting factors w_i indicates the relative significance of each term calculated by trial/error method and can be tuned to get the optimal performance (weighing factors are different for different vehicle models). First two terms in Equation 2.2 improves handling and safety of the vehicle, third term minimizes the control effort of momentum wheel and last term minimizes the angular speed of the momentum wheel.

Because, the desired side-slip angle is presumed to be zero [19] therefore $v_d = 0$ (lateral velocity is not desired) and r_d can be written as shown in equation 2.3 [47, 48].

$$r_d = \frac{v_x}{l + K_{us} v_x^2} \delta \tag{2.3}$$

where v_x, l, and K_{us} are the longitudinal velocity, wheel-base and understeer gradient respectively. Controller gains are calculated using LQR control theory. To obtain the analytical solution for the control system in the form of LQR problem, Equation 2.3 can be written as following Equation 2.4.

$$I_y = \frac{1}{2} \int_0^{t_f} [U^T R U + (X_d - X)^T Q (X_d - X)] dt \tag{2.4}$$

where,

$$X = \begin{bmatrix} v_y \\ r \\ r' \end{bmatrix}$$

$$X_d = \begin{bmatrix} v_d \\ r_d \\ 0 \end{bmatrix}$$

$$U = [T_d]$$

$$R = [w_3]$$

$$\begin{bmatrix} w_2 & 0 & 0 \\ 0 & w_1 & 0 \\ 0 & 0 & w_4 \end{bmatrix}$$

At last, applying the standard LQR approach, optimal control problem can be written as following Equation 2.5 [49].

$$U = -R^{-1}B^{T}(KX + S) \tag{2.5}$$

In Equation 2.6 *K* is the feedback gain matrix and *S* is the feed forward vector. First part of the equation is Riccati equation used to calculate the feedback gains and second part is solved as shown in equation 2.7 to obtain the feedforward gains.

$$\begin{cases} A^{T}K + KA - KBR^{-1}B^{T}K + Q = 0 \\ (A^{T} - KBR^{-1}B^{T})S + KE\delta - QX_{d} = 0 \end{cases} \tag{2.6}$$

$$S = -(A^{T} - KBRB^{T})^{-1} \times (KE\delta - QX_{d}) \tag{2.7}$$

At last equation 2.5 can be written as

$$S = N \,.\, \delta \tag{2.8}$$

where,

$$N = -(A^{T} - KBRB^{T})^{-1} \times (KE - QA_{d}^{-1}B_{d}) \tag{2.9}$$

Finally controller gains can be found as shown in equation 2.10-2.13.

$$K_v = \frac{b_{11}k_{11} + b_{21}k_{21} + b_{31}k_{31}}{w_3} \tag{2.10}$$

$$K_r = \frac{b_{11}k_{12} + b_{21}k_{22} + b_{31}k_{23}}{w_3} \tag{2.11}$$

$$K_{r'} = \frac{b_{11}k_{13} + b_{21}k_{23} + b_{31}k_{33}}{w_3} \tag{2.12}$$

$$K_\delta = n_{11} \tag{2.13}$$

where, b_{ij}, k_{ij} and, n_{ij} are the elements of matrix *B*, *K* and, *N* respectively.

2.3. ELECTRONIC SYSTEM STRUCTURE

Figure 2.4 shows the structure of the electronic control system for the developed prototype which consist of all the used hardware modules.

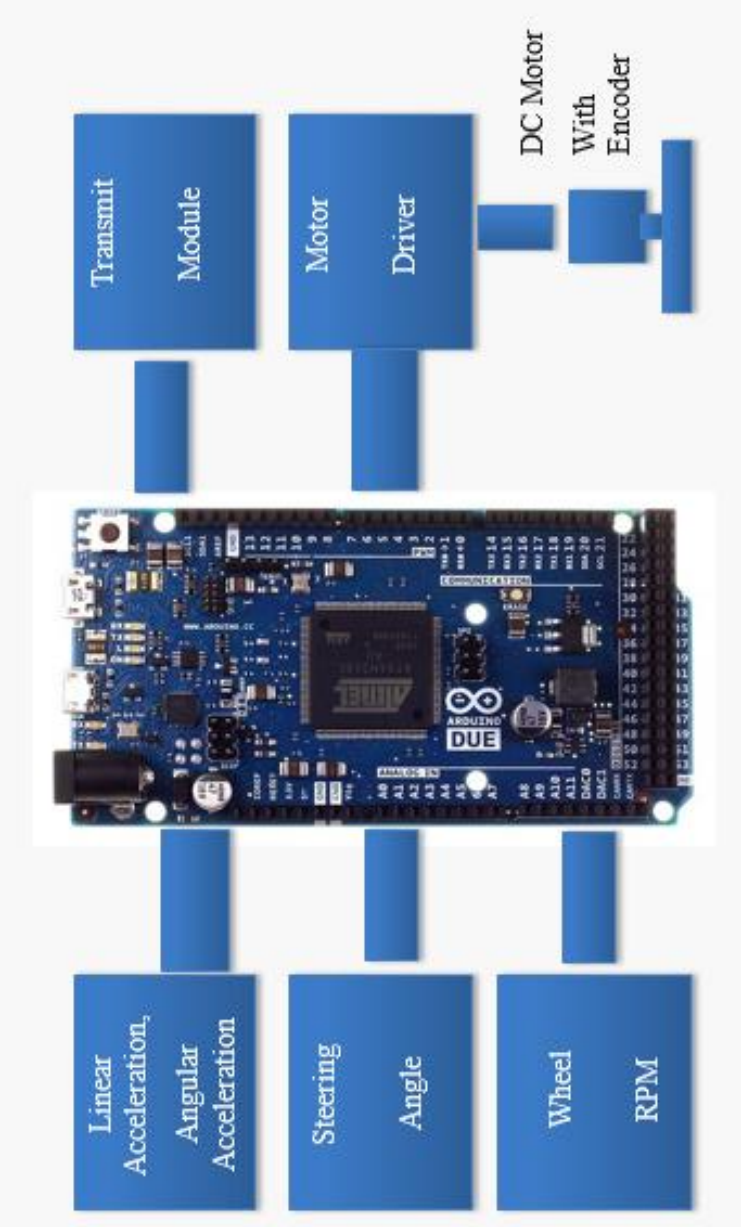

Figure 2.4: Electronic system block diagram

Table 2.1 summarizes the used hardware. More detail on each module is provided in Chapter 3.

Table 2.1: Hardware module used

Measured/Control Parameter	Device/Circuit	Interface Type	Communication Protocol
Acceleration (all three-axis)	MPU-6050	Digital	I2C
Pitch, Roll, Yaw	MPU-6050	Digital	I2C
Longitudinal Velocity	Hall Circuit	Digital	-
Steering Angle	RC filter	Analog	-
Motor Control	VNH5019	Digital	PWM
Motor Speed	Built-in Speed Encoder	Digital	-
Wireless Communication	ZigBee	Digital	UART

2.4. VEHICLE PROTOTYPE DEVELOPMENT

To decrease the prototype development time, already available scaled (1:10) radio controlled (RC) model car as shown in Figure 2.8 is used to perform all the experiments (which has the same geometric similarities) and observe vehicle dynamics using the above mentioned sensors and components. Reason to use the scaled model car is, safe to perform the tests and model can be easily modified as required without any major cost. A custom designed metallic platform holds all the electronic circuitry and the momentum wheel control structure all together with the DC motor on top of the RC model car's chassis as shown in Figure 2.5.

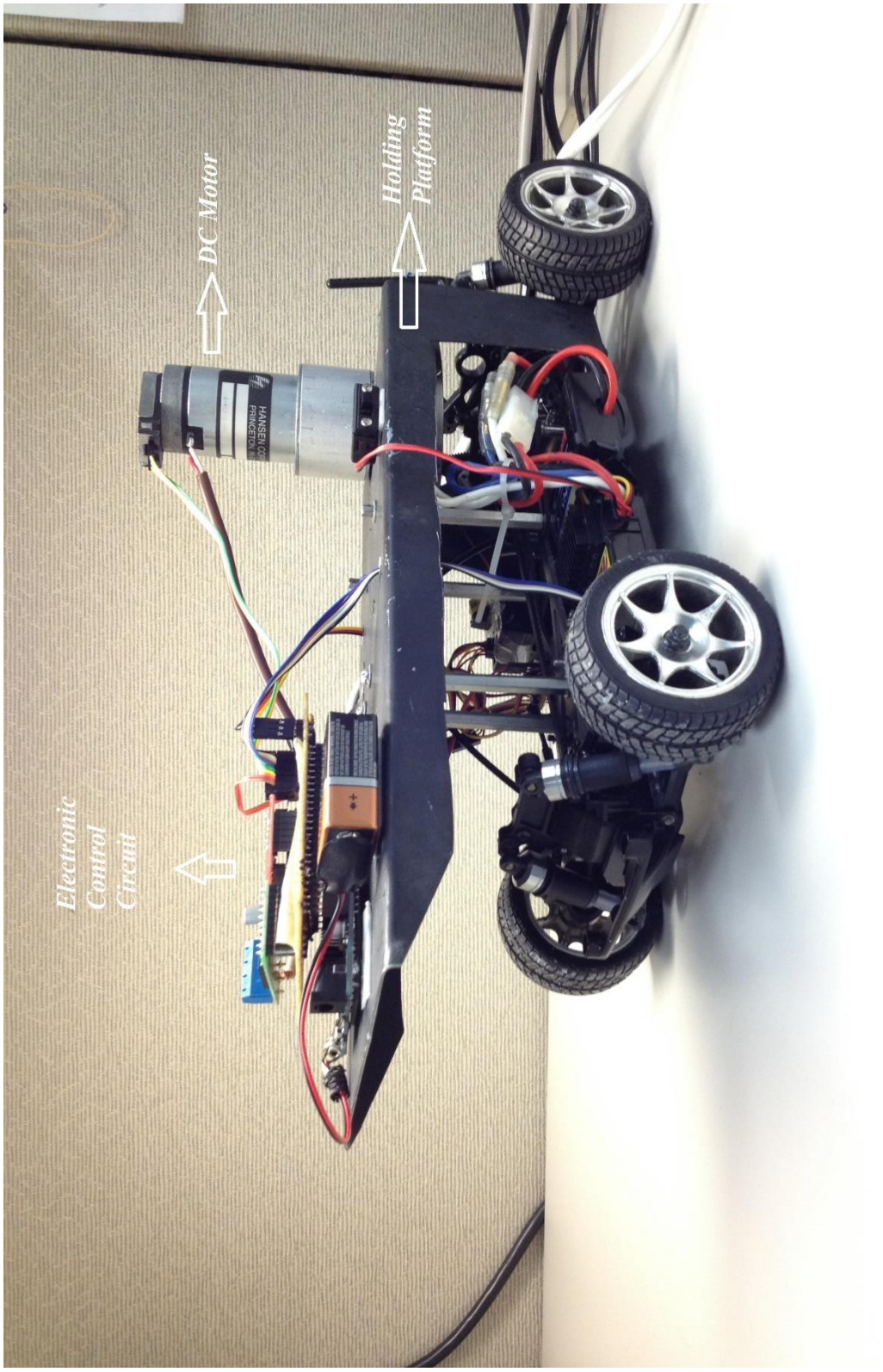

Figure 2.5: Developed prototype

Vehicle's dimensions are provided in the Table 2.2, Figure 2.6 and Figure 2.7.

Table 2.2: Vehicle prototype dimensions

Parameter	Value	Remarks
Height	21.5 cm	With motor
Width	20 cm	End to end
Length	38.8 cm	Including holding platform
Wheelbase	26 cm	-
Ground clearance	1.35 cm	-
Tire circumference	21.6 cm	-
Total weight	3.215 kilograms	-

2.4.1. Scaled RC model car

RC model car used to develop a prototype of the proposed system as shown in Figure 2.8. RC model car uses separate radio transmitter and receiver module than the measurement and control system. It operates on 72.5 MHz and can control up to two channels. One channel is used for the acceleration and one for the steering control as shown in Figure 2.9 of the RC model car.

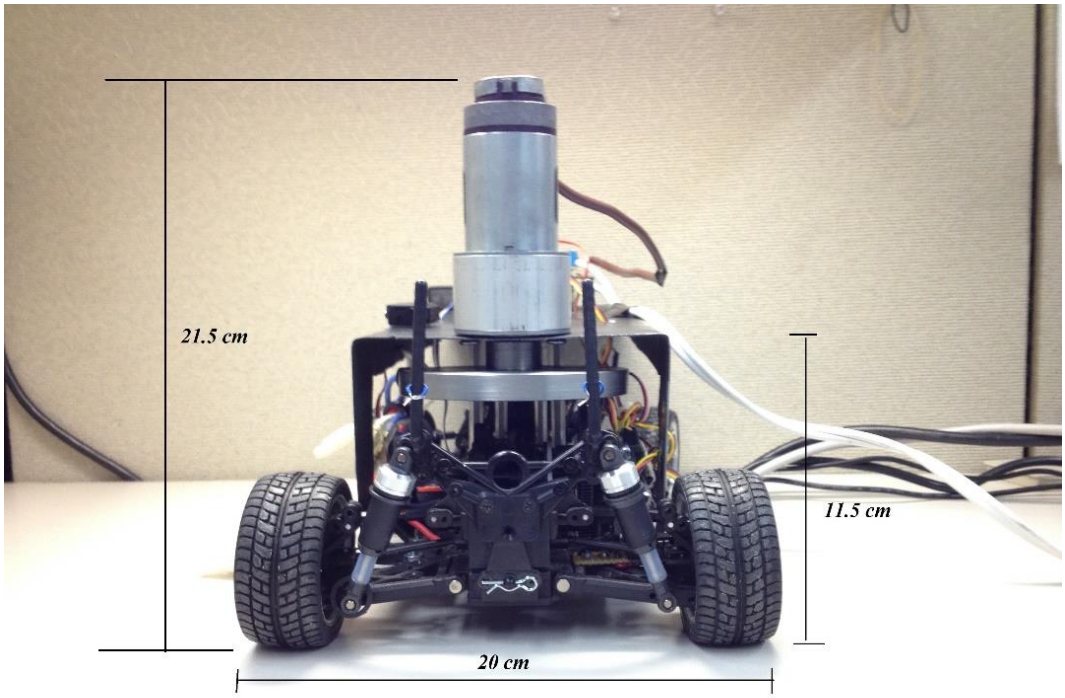

Figure 2.6: Prototype dimensions

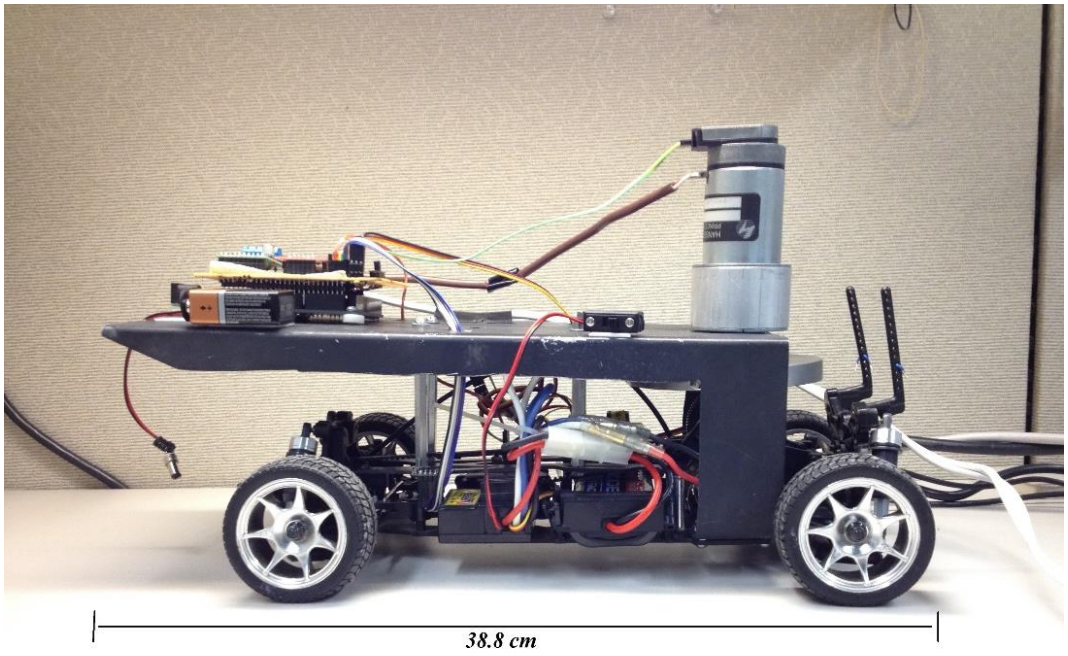

Figure 2.7: Prototype dimensions

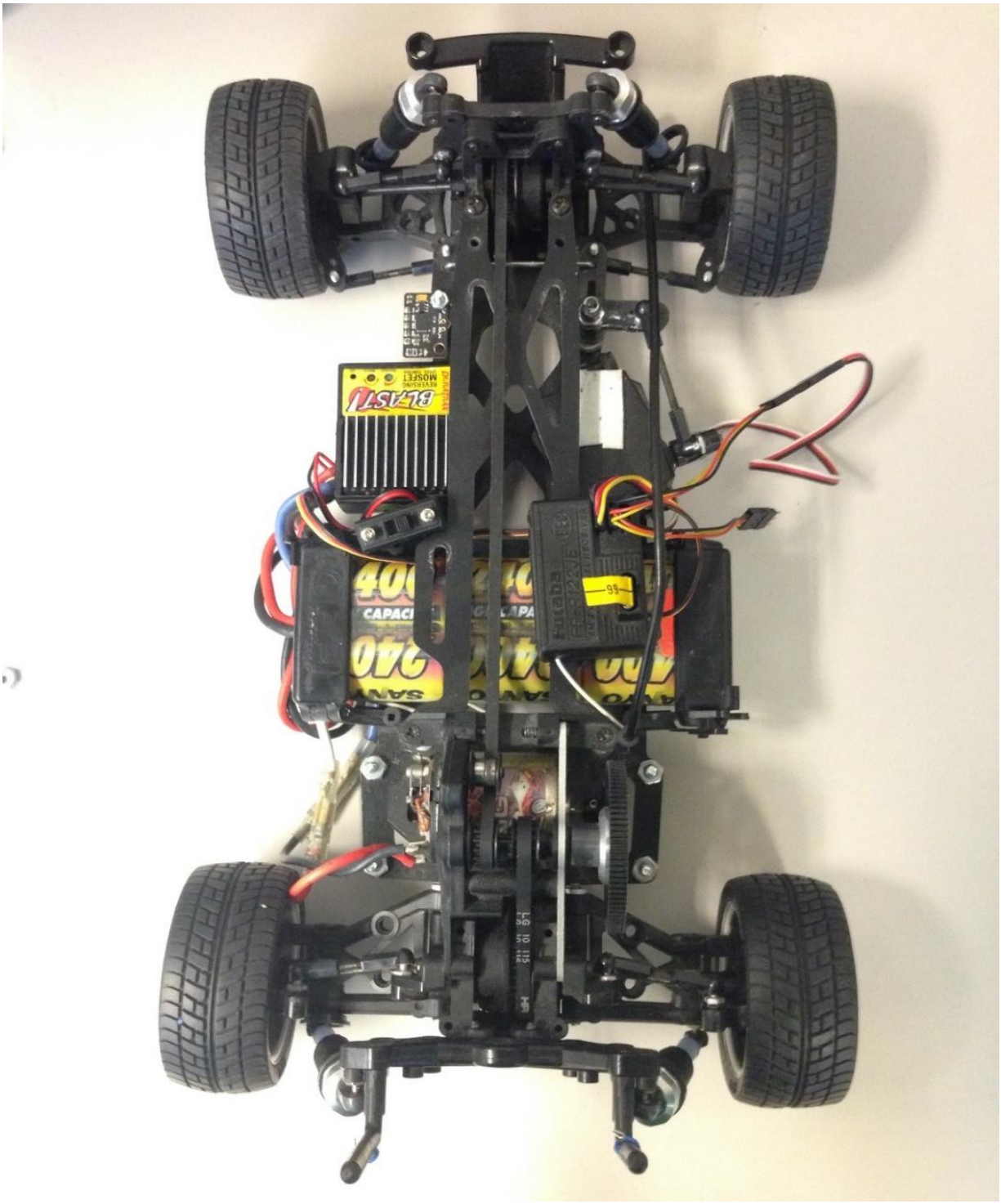

Figure 2.8: RC model car platform

Main parts of the RC model car are:

- Transmitter
- Receiver
- Electronic speed control (ESC)
- Servo Motor
- DC motor
- Battery pack

2.4.1.1. Transmitter

Futaba radio transmitter provides reliable and long range transmission. It transmits control signals over the 75.5 MHz frequency, with the transmitter user is able to control the direction and the speed (in both the directions) of the RC model car. RC model car transmitter is shown in the Figure 2.9. Transmitter has its own power source which provides enough power to drive the control circuit and radio waves. It works on eight AA batteries and has the following features:

- Throttle and Steering Trim
- Throttle and Steering Reverse
- Throttle End Point Adjustment
- Built-in antenna
- Low battery warning

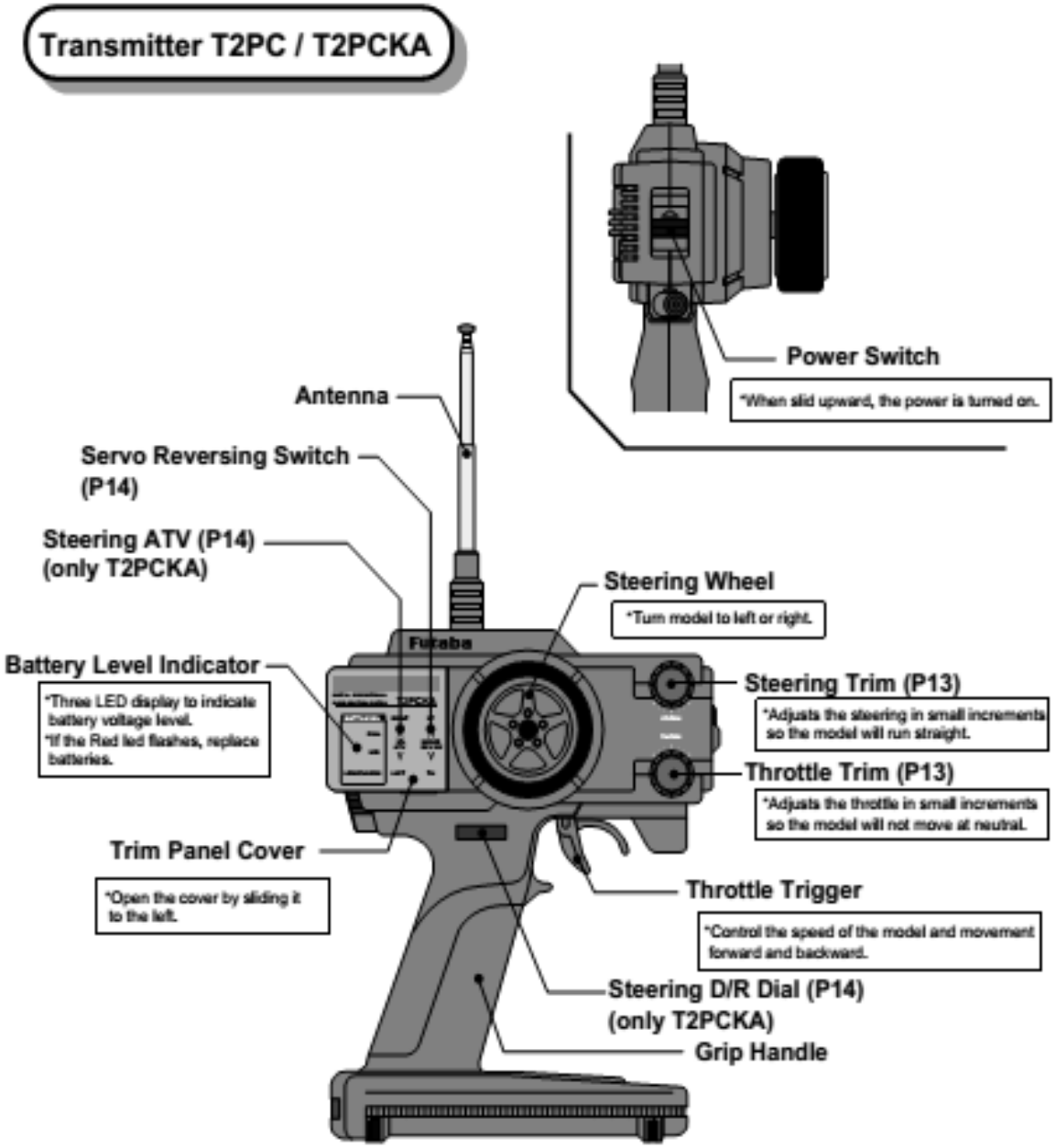

Figure 2.9: RC model car transmitter

2.4.1.2. Receiver

Two channel receiver operating at 72.5 MHz is as shown below in Figure 2.10. It receives signal from the transmitter and performs appropriate actions. Channel-1 (CH-1) is fed directly to the

steering servo motor and channel-2 (CH-2) is applied to the throttle servo through the electronic speed control (ESC).

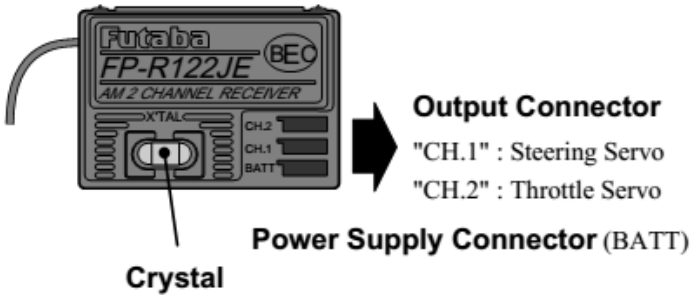

Figure 2.10: RC model car receiver

2.4.1.3. Electronics Speed Control

Speed of the RC model car is controlled using an Electronic Speed Control (ESC) module manufactured by Duratrex as shown in Figure 2.11.

Figure 2.11: Duratrex electronic speed control (ESC) module

Specifications of the ESC are given in Table 2.3.

Table 2.3 ESC specifications

Parameter	Value	
	Forward	**Reverse**
Maximum constant current	140 amps	60 amps
Battery size	5 to 7 Cells	
Case size	1.57” x 1.81” x 0.79”	
Weight	73 grams	
Battery eliminator circuit (BEC) voltage	5.6 volts	

ESC controls only the DC motor. Servo motor which control the steering angle is directly connected to the receiver shown in Figure 2.12.

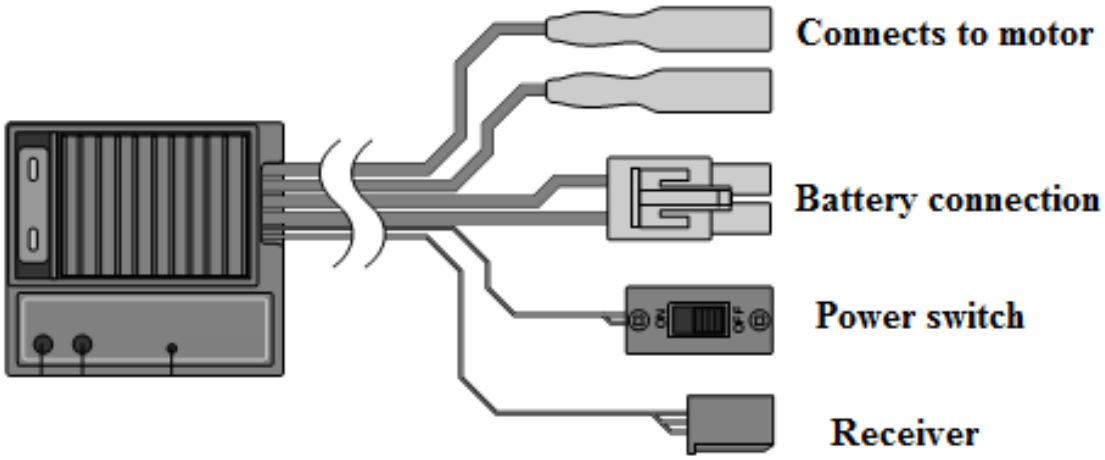

Figure 2.12: ESC leads connection

2.4.1.4. Servo Motor

Servo motors are compact in size and are highly energy efficient. These features makes the servo motor a perfect choice in various kind of RC toys. Servo motor controls the steering angle of the RC model car. Servo motor consist of a DC motor, potentiometer and control circuit. An electric signal (PWM) applied to the position able shaft close-fitted with the gears controls the movement of motor. Servo motor's specification are given below [50]:

- Speed: 0.23 sec/60° @ 4.8V
- Torque: 44 oz-in (3.2 kg-cm) @ 4.8V
- Dimensions: 1.6 x 0.8 x 1.4" (1-9/16 x 13/16 x 1-7/16") (40 x 20 x 36mm)
- Weight: 1.3oz (1-5/16oz) (37g)

2.4.1.5. DC Motor

DC motor together with the ESC provides the control over the speed of the vehicle. Based on the signal received by the receiver, ESC controls the current and the direction of the DC motor.

2.4.1.6. Battery

RC model car is operated by the 7.2V, 3000 mAh's, and NiMH battery, which supplies the current to the Servo, DC motor and the ESC. Battery weighs 350 grams and shown in Figure 2.13.

Figure 2.13: RC model car battery

2.4.2. Momentum Wheel

Metal disc shown in Figure 2.14 acts as momentum wheel to generate the required corrective torque (A 3D design and dimensions (in mm), of the custom designed metal disk, are provided in the Figure 2.17 below.

2.4.2.1. Drawing Details

Disc wieghs 670 grams and has a Momentun of intertia (MOI) I_d = of $8 * 10^{-4} {Kg}/{m^2}$ as calculated by software. The disc is attached to the DC motor and mounted at the rear of RC model car, where the centre of gravity lies. Dimensions selected for the momentum wheel were limited by the physical dimension of the RC model car and can be optimized to provide maximum moment of inertia with lowest possible mass.

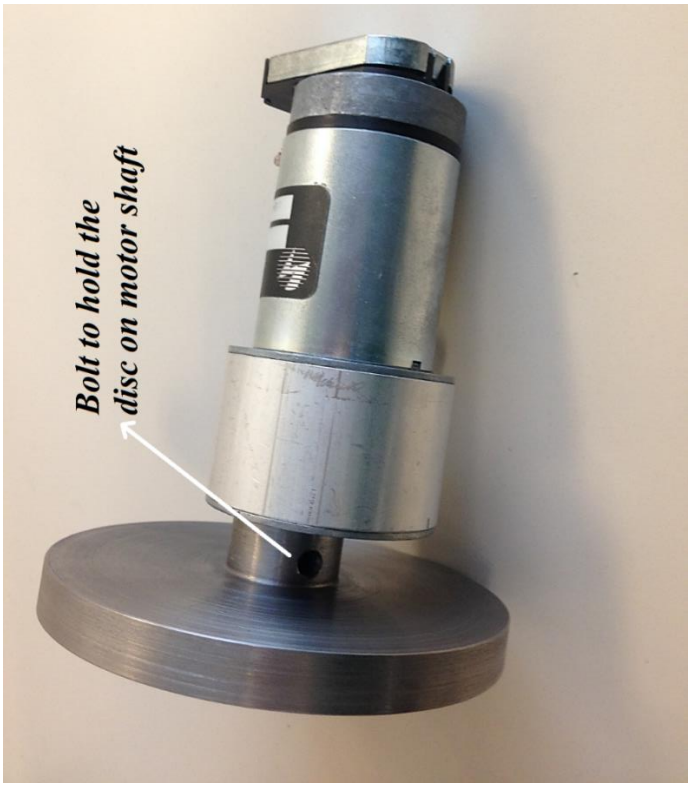

Figure 2.14: Momentum disc attached with motor

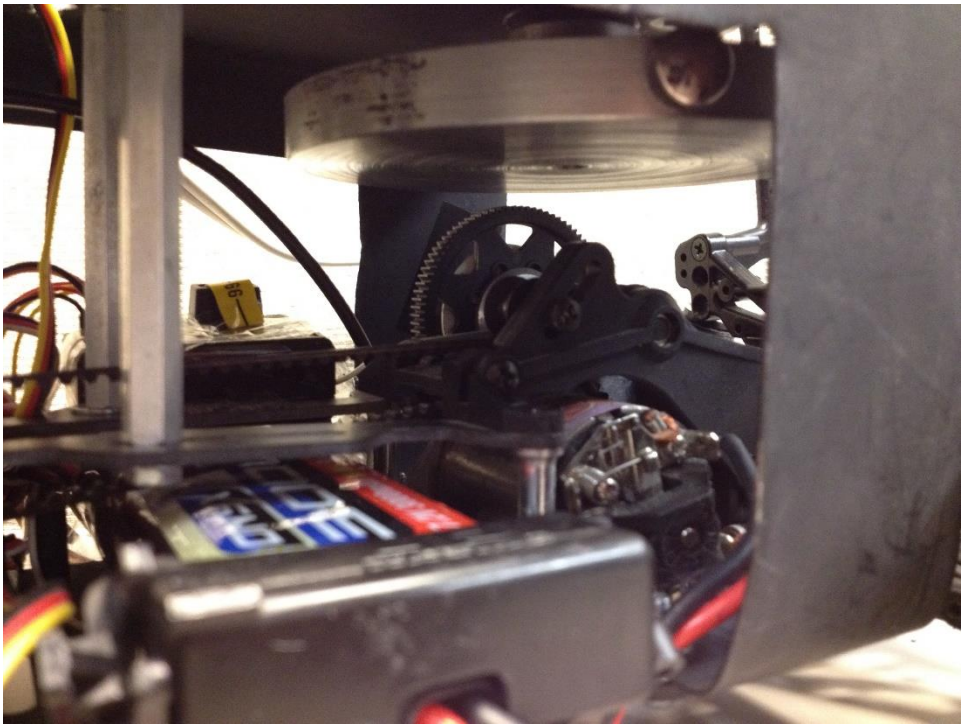

Figure 2.15: Momentum disc

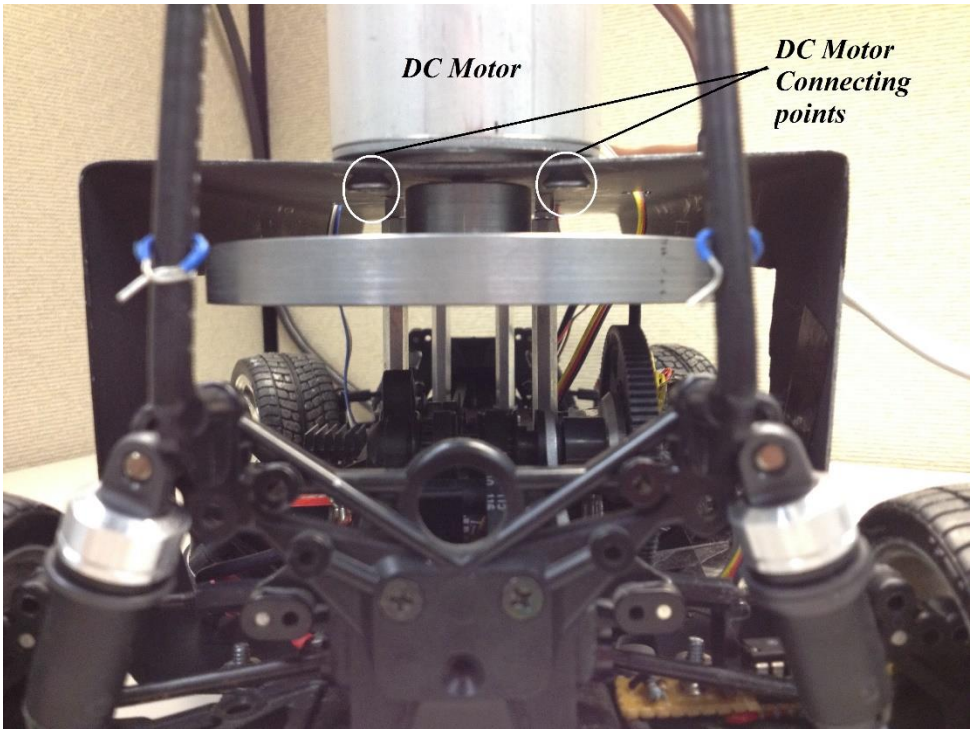

Figure 2.16: Momentum disc as seen from the rear of the RC model car

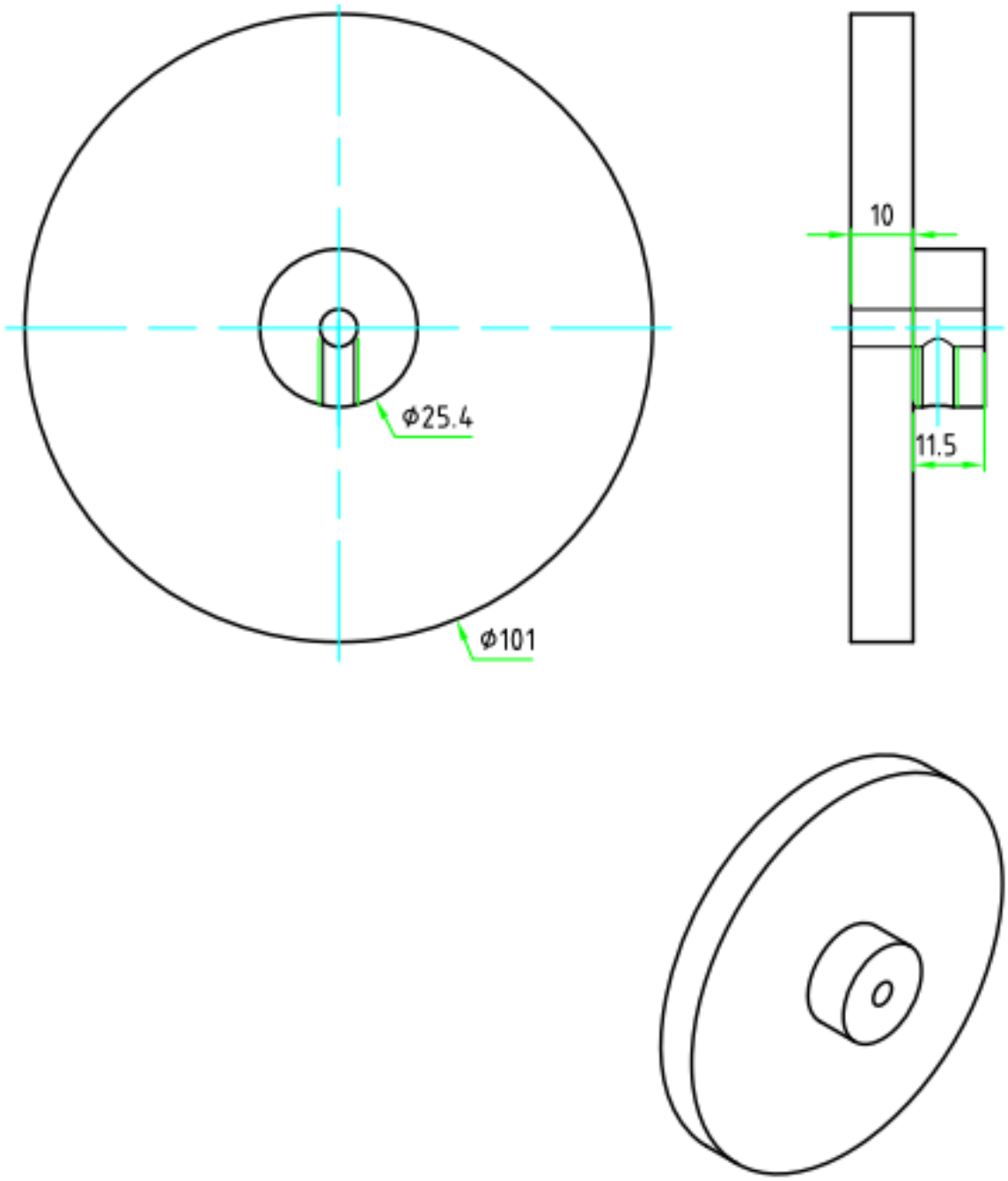

Figure 2.17: Momentum disc dimensions

2.4.3. Holding Platform

Holding platform is made out of the metal sheet. It hold motor, electronic circuitry and, momentum wheel together on top of the RC model car's chassis as shown in Figure 2.18. Other than providing space for all the mechanical components and electronic circuits, holding platform has one more important function i.e. transfer the torque generated by the motor to the RC model car's chassis. To transfer the maximum torque, holding platform was connected to chassis with eight connecting points. From which two connecting points located in front Figure 2.18, two in the middle Figure 2.20and, four at the rear of the chassis shown in Figure 2.21(two on the left side, two on the right side). More connecting points were used in the rear to hold the weight of the momentum disc and the DC motor also, to reduce the effect of vibrations due to the DC motor operation.

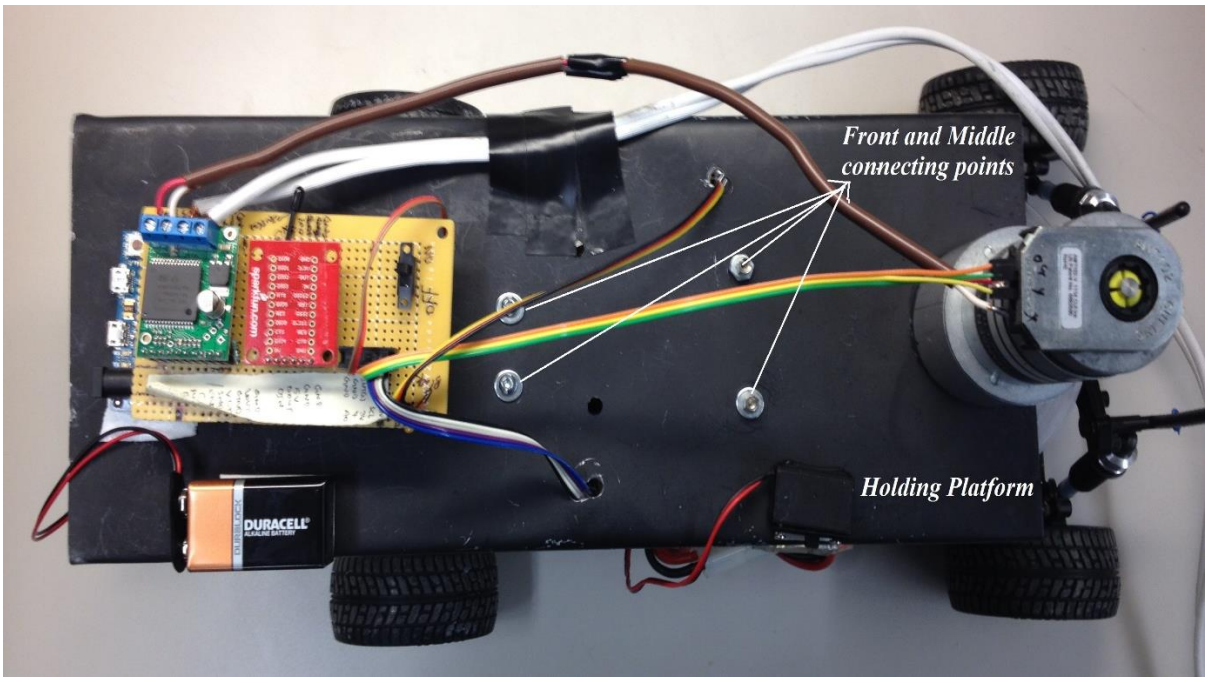

Figure 2.18: Holding platform

Four metal rods as shown in Figure 2.19 connects the holding platform to the RC model car's chassis as shown in Figure 2.20.

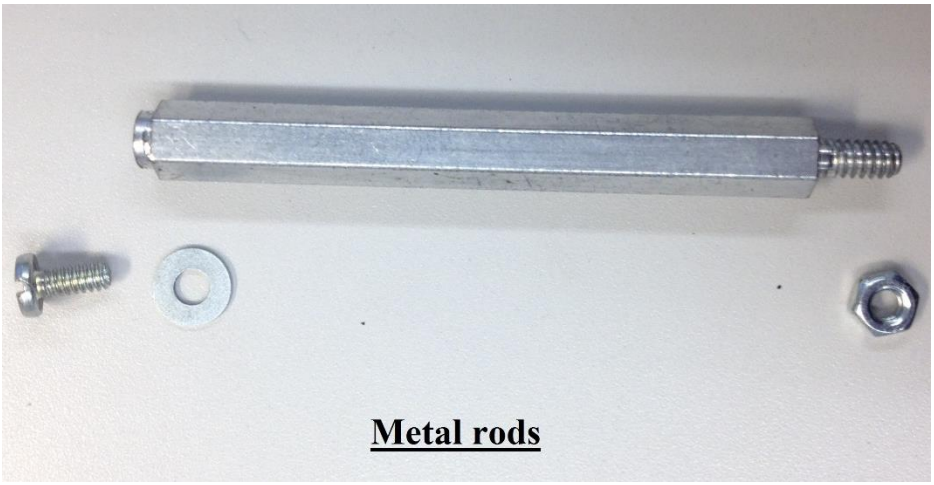

Figure 2.19: Metal rods

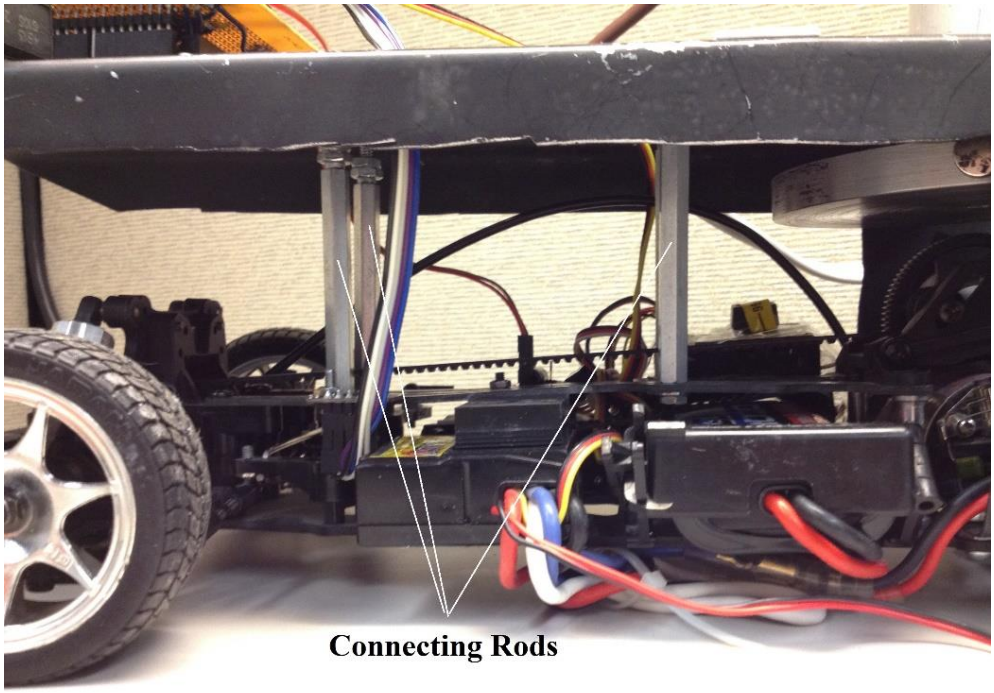

Figure 2.20: Front and rear metal rods

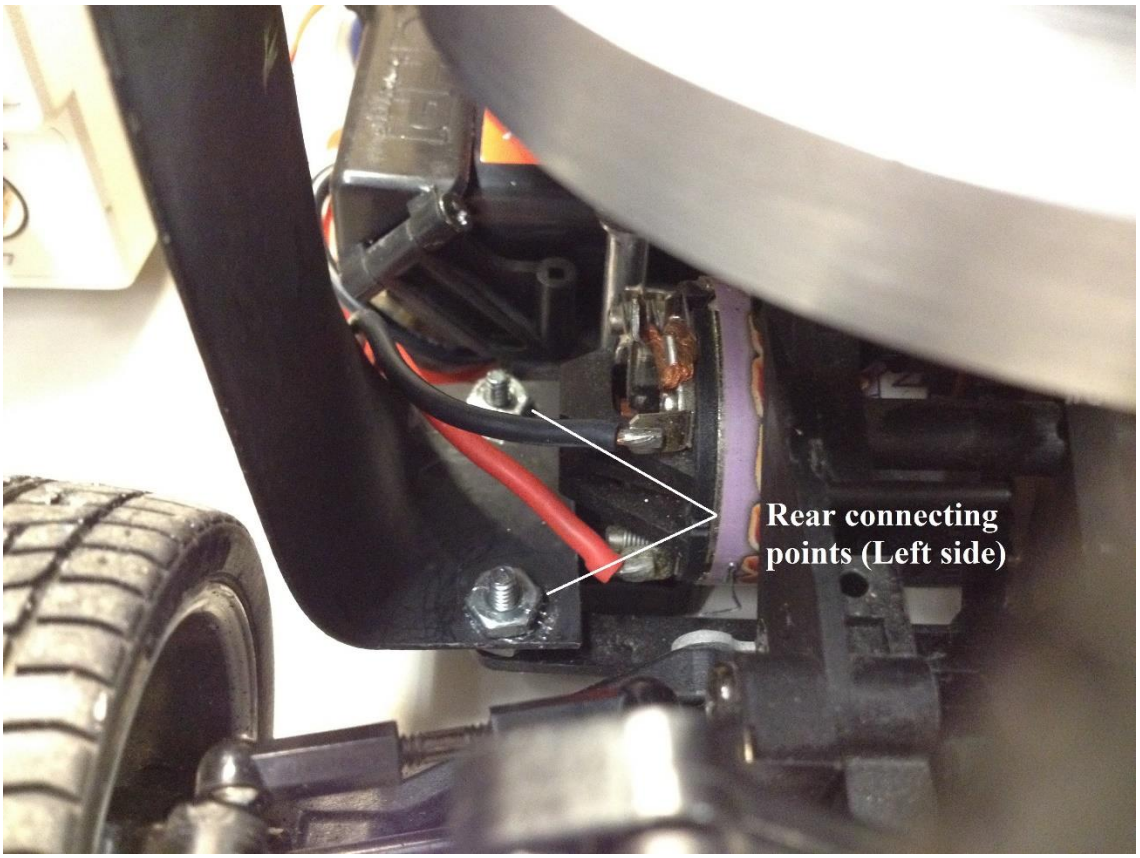

Figure 2.21: Rear connecting points

Chapter 3: **Hardware Description**

In this chapter in depth detail about the each electronic module has been provided. Steps on how to program and configure a hardware module before using it in the integrated system are discussed.

3.1. PARTS LIST

Major components and modules used in the development of prototype are tabulated in Table 3.1.

Table 3.1: Parts list

Number	Name	Manufacturer	Supplier	Quantity	Additional information
1	Microcontroller Board	Arduino	Robotshop	1	Uses 32-bit ARM processor
2	6-DOF Inertial Measurement Unit (IMU)	Invensense	Robotshop	1	Measures linear and angular acceleration
3	ZigBee	Digi	Robotshop	2	Wireless transceiver
4	ZigBee explorer	Robotshop	Robotshop	1	Connects computer with wireless transceiver
5	Motor Driver	ST Microelectronics	Pololu	1	Generates enough current to rum motor.

Table 3.2: Table 3.1 continued

6	RPM Counter	Custom Hall Circuit	-	1	Generates pulses together with hall sensor circuit
7	PWM to Analog	Custom LPF	-	1	Calculates steering angle.
8	Model Car	-	-	1	Used to observe dynamics of the vehicle.
9	DC Motor	Hansen	Hansen	1	Generates torque.
10	Battery	Duracell	-	1	Powers electronic circuit.
11	Holding Platform	Custom Metal Structure	-	1	Hold electronic and hardware structure all together.
12	Metal Rods	-	-	4	Holds metallic platform to the chassis.
13	Switch	-	Robotshop	1	Enable/Disable Momentum wheel controller
14	10 Pin Connector	-	Robotshop	1	Connects various sensors to the microcontroller board
15	General Purpose Printed Circuit Board (PCB)	-	Robotshop	1	Used to solder motor driver, ZigBee module
16	Female Header Pins	-	Robotshop	-	Used to connect motor driver, ZigBee on the general purpose PCB
17	Male Header Pins	-	Robotshop	-	Connects PCB on top of the microcontroller board

3.2. ARDUINO DUE:

Arduino DUE is a microcontroller board supporting 32-bit ARM microcontroller, designed by Arduino. Microcontroller board acquires data from different sensors and then process it as per the algorithm. Functions performed by the microcontroller board are:

- Noise filter: Filters steering angle and momentum wheel velocity
- Kalman filter: Lateral and longitudinal velocity
- LQR: Calculates required torque
- PI controller: Controls motor

3.2.1. Overview

Arduino Due is the ARM-based development board used as the brain of control system design, shown in Figure 3.1, supporting a powerful 32-bit CortexM3 core. Arduino DUE outperforms other 8-bit and 16-bit microcontroller boards when it comes to float calculations. ARM architecture with high clock speed makes sensor data acquisition from multiple sensors quick and reliable.

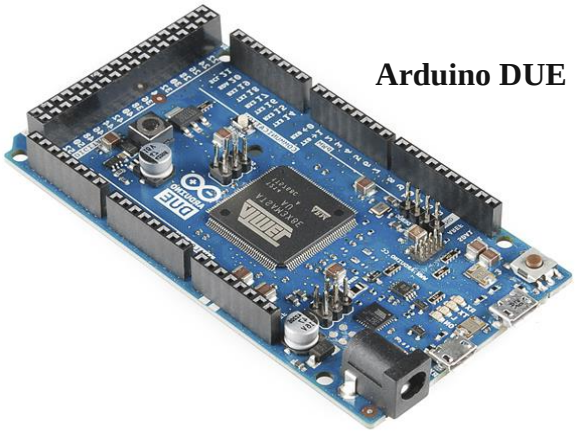

Figure 3.1: Arduino DUE microcontroller board.

Microcontroller board has two micro USB ports, a programming and communications port and other lets microcontroller board to act as a client or host, permitting it to perform as or utilize a USB keyboard or mouse .Arduino DUE board has many advantages over the other microcontroller boards, which are compiled below: -

- Microcontroller: AT91SAM3X8E
- Operating Voltage: 3.3V
- Recommended Input Voltage: 7-12V
- Min-Max Input Voltage: 6-20V
- Digital I/O Pins: 54 (of which 12 provide PWM output)
- Analog Input Pins: 12
- Total DC Output Current on all I/O lines: 130 mA
- DC Current for 3.3V Pin: 800 mA
- DC Current for 5V Pin: 800 mA
- Flash Memory: 512 KB all available for the user applications
- SRAM: 96 KB (two banks: 64KB and 32KB)
- Clock Speed: 84 MHz
- Direct memory access (DMA) controller
- 9- channel 32-bit Timer/Counter
- 8-channel 16-bit PWM

3.2.2. Power Distribution

Arduino Due board operates at 3.3V and can be powered from a Universal Serial Bus (USB) cable or with the external DC adaptor. Board supports 2.1mm center-positive power jack. Also, external battery or DC source (6V-20V) can be connected directly to the VIN and GND pins headers of the board. It is recommended to use the board between 7v to 12V. Maximum current draw from the onboard regulator is 800 mA. Pin description for the power section has been tabulated in Table 3.3

Table 3.3: Power section pin description

Pin name	Description
VIN	Input voltage to the board using external power source.
5 V	Regulated 5V output
3.3 V	Regulated 3.3V output
GND	Ground
IOREF	Operating voltage reference for external devices

Microcontroller board supports two level of voltages. 5V section is to power up the external peripherals and 3.3V is used for the microcontroller itself. Microcontroller board does not supports more than 3.3V at the input/output pins. Input voltage to the microcontroller board ranges between 6V to 20V. More than 20V at the input can damage the board. Schematic diagram for the power distribution section (5V and 3.3V) is shown in the Figure 3.2and Figure 3.3 respectively.

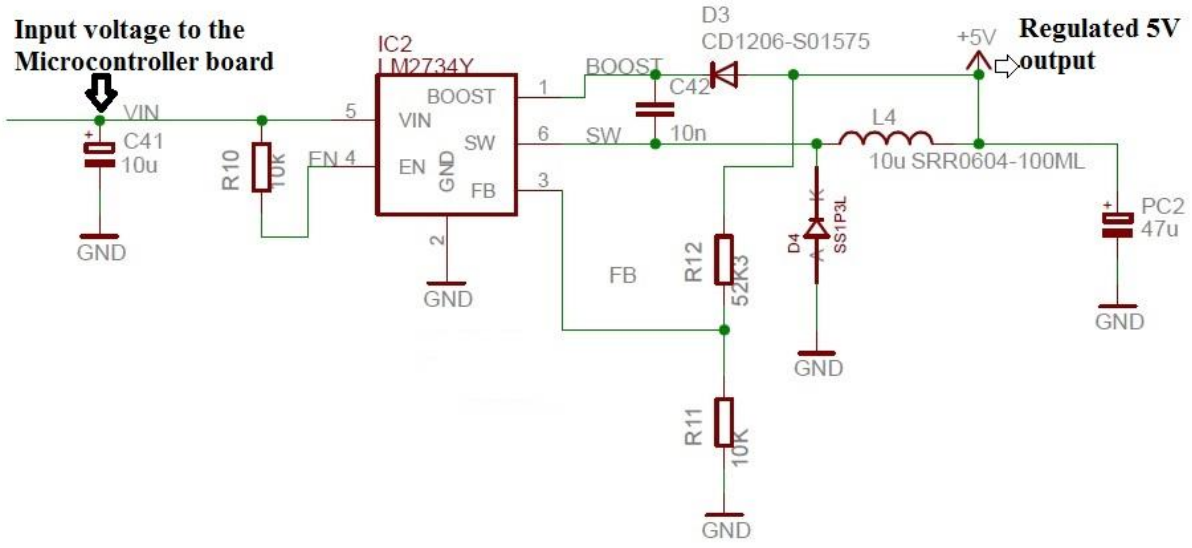

Figure 3.2: Power distribution 5V.

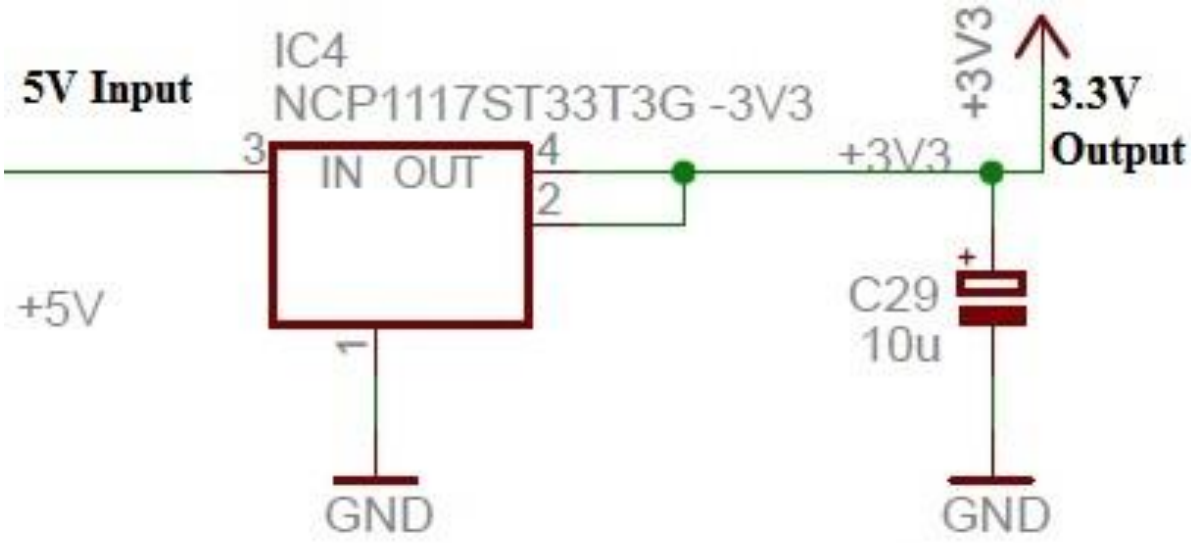

Figure 3.3: Power distribution 3.3 V.

3.3. INERTIAL MEASUREMENT UNIT (MPU-6050)

MPU-6050 is a 6-Degree of freedom (DOF) inertial measurement unit (IMU) that provides the linear and angular accelerations of the vehicle. Before discussing the MPU-6050, basic working principles of accelerometer and gyrometer has been explained in the following sections.

3.3.1. Accelerometer

Accelerometer is used to measure rate of change of velocity i.e. acceleration, which can be in one axis, two axes or three orthogonal axes. Accelerometer can be used to measure velocity, position and orientation of the object it is attached to. It also has wide variety of applications as a vibration sensor.

Accelerometers are sensitive to both linear acceleration and the local gravitational field. At rest accelerometer will measure 1g (9.81m/s^2) due to the earth's gravitational pull, which is called static acceleration. To measure the dynamic acceleration due to the motion but not because of the gravity, gravitational offset should be subtracted from the measured value. Some 'g' reference points shown in Table 3.4.

Accelerometers are available in different types based on their sensing element and principle of operation. An accelerometer can be divided into main two categories mechanical and solid state. Because accelerometer based on Micro electromechanical systems (MEMS) technique is used in the prototype development, MEMS accelerometer has been discussed in brief.

Table 3.4: g-Reference values.

g-level	Description
1g	Earth's gravity
1g	Passenger car in corner
2g	Bumps in road
3g	Indy car driver in corner
5g	Bobsled rider in corner
7g	Human unconsciousness
10g	Space shuttle

Based on the sensing element accelerometer can be of the following type.

- Capacitive
- Piezoelectric
- Piezoresistive
- Hall Effect
- Magnetoresistive
- Heat transfer

Solid state accelerometers are compact in size and highly sensitive compared to the conventional mechanical accelerometers. MEMS technology has enabled these devices to operate on very low power and provide robust sensing. First MEMS accelerometer was developed at Stanford

University in 1979 [51]. Most of the MEMS accelerometer works on the principle of capacitive sensing [52].

3.3.2. Gyrometer

Gyroscopes are used to measure the angular rate of an object (how quickly an object turns), typically measured in degrees per second (^{0}/s) in reference to one of the three axes. In simple words it measures variations in rotational angle per unit of time. Common applications includes tracking movements, Navigation device, vehicle stability control, and image stabilization. Gyroscope structure can be divided into main three categories:

- Rotary or Mechanical gyroscopes
- Vibrating structure gyroscopes
- Optical gyroscopes

3.3.3. MPU-6050 Overview

MPU-6050 is micro-electro mechanical system (MEMS) based device integrating 3-axis gyroscope and 3-axis accelerometer with a Digital Motion Processor (DMP) on a single chip. MPU-6050 uses I2C communication protocols shown in Figure 3.4. Some of the main features of the device are given below [53]: -

- Motion fusion by the on chip DMP
- Supply voltage range 2.375 – 3.46V
- Minimal cross axis sensitivity between accelerometer and gyroscope
- 10,000 g shock tolerant
- Small and thin QFN package: 4x4x0.9mm

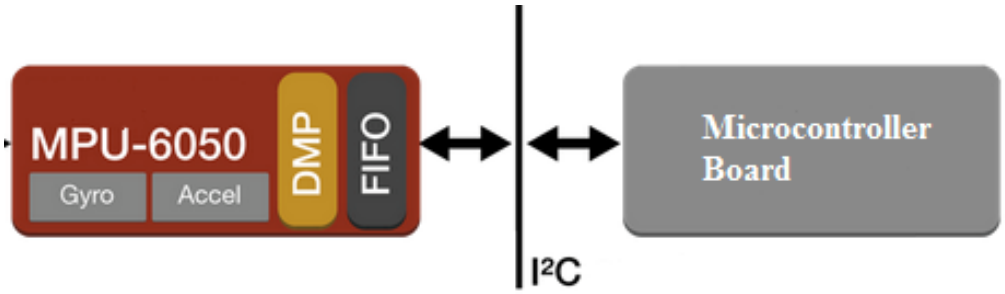

Figure 3.4: MPU-6050 system design.

3.3.3.1. Gyroscope Features

- Full scale range of ±250, ±500, ±1000, ±2000 degrees per second
- Integrated 16-bit Analog to Digital converters (ADC)
- Enhanced bias and sensitivity temperature stability
- Improved low-frequency noise performance
- Digital programmable low pass filter

3.3.3.2. Accelerometer Features

- Full scale range of ±2g, ±4g, ±8g, ±16g
- Orientation and tap detection
- Integrated 16-bit ADC's
- Low power mode

3.3.4. Block Diagram

MPU-6050 is comprised of the following main blocks, see Figure 3.5 [53].

- Three-axis gyroscope with 16-bit ADCs and signal conditioning
- Three-axis accelerometer with 16-bit ADCs and signal conditioning
- Digital motion processor (DMP)
- Primary I2C
- Auxiliary I2C
- Clocking
- Sensor data register
- First in first out (FIFO)
- Interrupts
- Digital temperature sensor
- Self-test circuit
- Charge pump
- Bias and LDO (low dropout regulator)

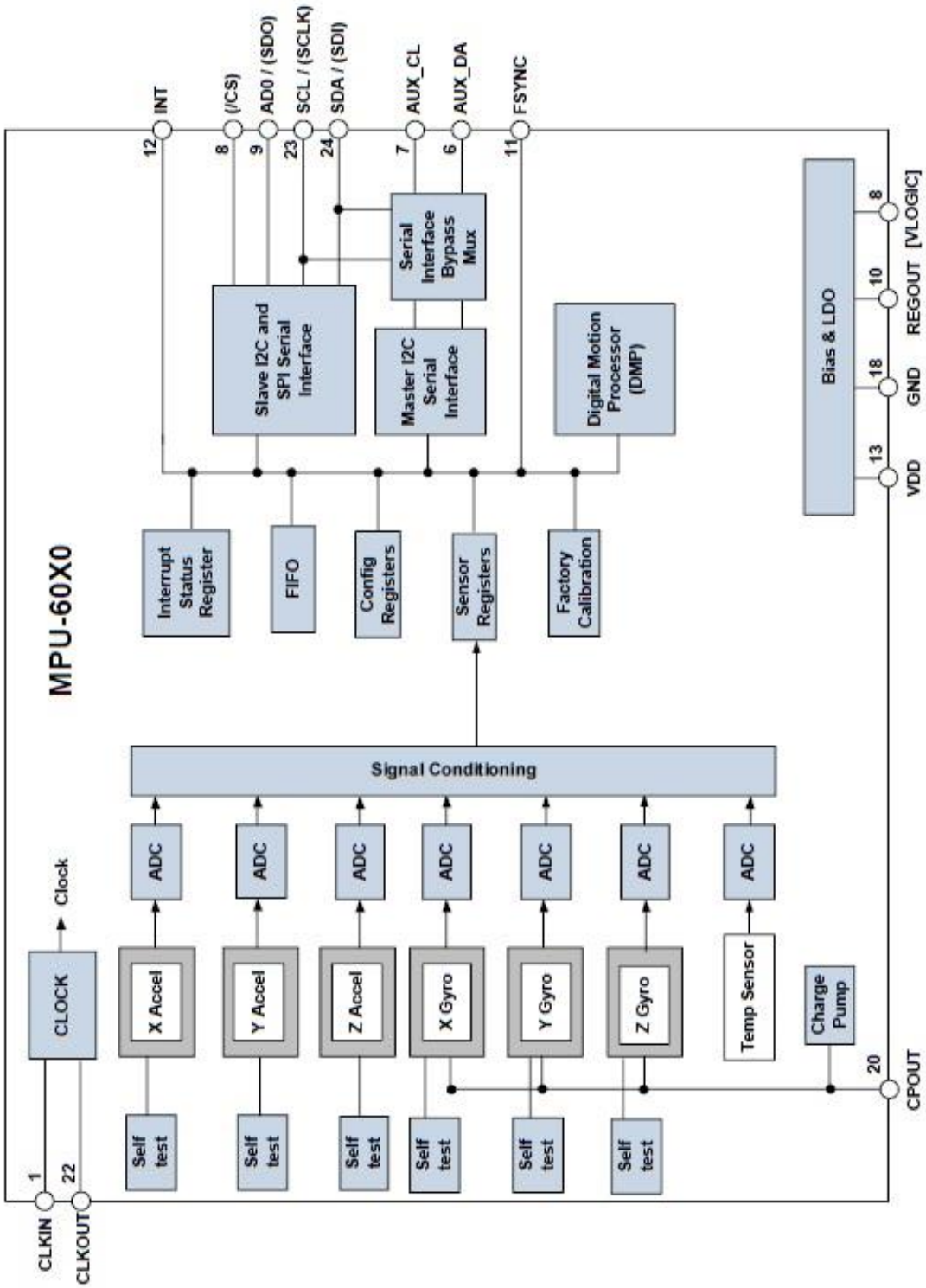

Figure 3.5: MPU-6050 Block diagram.

3.3.4.1. Three-axis gyroscope with 16-bit ADCs and signal conditioning

MPU-6050 contains three independent MEMS gyroscopes. Each gyroscope detects rotation about the X-axis, Y-axis, and Z-axis respectively. Rotation about any of the axis causes a vibration due to the Coriolis Effect that is detected by a capacitive pickoff.

The signal is then amplified, demodulated, and filtered to produce a voltage that is proportional to the sensed angular rate. Analog voltage is converted in to digital using individual on chip 16-bit ADCs. The ADC sample rate is programmable from 8,000 samples per second, down to 3.9 samples per second, and user-selectable low pass filters enable a wide range of cut-off frequencies.

3.3.4.2. Three-axis accelerometer with 16-bit ADCs and signal conditioning

MPU-6050 uses capacitive sensing technique to detect the acceleration. It contains three proof masses, one each for three-axis. Displacement generated by the acceleration along a particular axis is detected using capacitive sensors. Voltage generated by the sensors is then converted in to the digital signal using the 16-bit sigma-delta ADC. Advanced MEMS architecture reduces the thermal drift and provides minimal cross axis sensitivity.

3.3.4.3. Digital Motion Processor (DMP)

Digital Motion processor embedded in the MPU-6050 allows to do the complex fusion algorithms on the chip itself. This offloads computational algorithms from the host microcontroller and saves processing power. Using the DMP, motion processing algorithms can run at a higher rate than the conventional controller algorithms. For example motion processing algorithm should run at 200Hz for the accurate results, on the other hand update rate can be much smaller than the 200Hz.

3.3.4.4. Primary I2C

MPU-6050 uses I2C serial protocol to communicate with the host microcontroller (explained in next section). It is always in the slave mode while communication. Pin 9 (AD0) of the MPU-6050 is used to set the least significant bit (LSB) of the I2C slave address. Default Slave address is 0X68 (by pulling the Pin-9 low permanently) on the MPU-6050 breakout board manufactured by robotshop.

3.3.4.5. Auxiliary I2C

MPU-6050 offers integration of third party off the chip 3-axis digital magnetometers in to the fusion data hence provide better results with low latency. Auxiliary I2C can be used in the two modes. In the master mode MPU-6050 acts as master to the external connected devices whereas in pass through mode it can communicate directly with the external devices.

3.3.4.6. Self-Test

Self-Test allows to test the electrical and mechanical components of the chip. When enabled causes the sensor to be actuated (move the proof mass to certain distance which should generate pre-defined Coriolis force) and generate a signal. Generated signal is then used to observe the self-test response using Equation 3.1. If the response is within the limits provided in the specification, the part has passed self-test otherwise not.

$$Self-Test\ Response = (Output\ when\ self\ test\ enabled - Output\ when\ self\ test\ disabled) \quad (3.1)$$

3.3.4.7. Clock

MPU-6050 provides flexibility in the allowable inputs for the clock generation. An internal relaxation oscillator, any of the X, Y, or Z gyros (MEMS oscillator) can be used as the clock source. External clock signals of fixed frequency of 32.768 kHz square wave, 19.2 MHz square wave are also supported.

On start-up MPU-6050 uses its internal clock generation module until programmed to operate from another source. Z-gyro has been used for the clock generation.

3.3.4.8. Sensor Data Registers

Sensor data registers contains measurement data from accelerometer, gyrometer, and temperature sensor. These registers can be read using the I2C serial protocol and are only readable.

3.3.4.9. First In First Out (FIFO)

FIFO register can contain the data from accelerometer, gyrometer, temperature sensor or an auxiliary sensor, which can be configured using the FIFO configuration register. FIFO counter keeps track of the valid data bytes in the FIFO register. Size of the FIFO register is 1024-bytes and it supports both burst read and interrupt function.

3.3.4.10. Interrupts

Interrupts eliminates the need of pooling. This saves the time as host microcontroller does not have to spend time in putting request and waiting for the available data. Use of interrupts guarantees to respond to the request on time and reduce the risk of losing data. Following Table 3.5 shows the sources of interrupt generation in MPU-6050.

Table 3.5: Interrupt sources of MPU-6050.

Interrupt Name	Description
MOT_INT	Motion detection interrupt
FIFO_OFLOW_INT	FIFO buffer overflow
I2C_MST_INT	I2C master interrupt
DATA_RDY_INT	Measurement data ready

Only FIFO buffer overflow and Measurement data ready has been used. More details on interrupts is provided in the appendices.

3.3.4.11. Digital Temperature Sensor

MPU-6050 contains an on chip digital temperature sensor that measures the die temperature. Temperature measurement data can be read either from FIFO or sensor data register. This data is also incorporate in the sensor fusion algorithms by DMP.

3.3.4.12. Bias and Low dropout (LDO)

This section generates the supply voltage, reference voltages, and required current for the MPU-6050.

3.3.4.13. Charge pump

High voltage required by the MEMS oscillator is generated by the charge pump section.

3.3.5. Communication

Communication behavior between the microcontroller board and MPU-6050 is explained in this section.

3.3.5.1. Overview

MPU-6050 supports only I2C protocol for communication with the host microcontroller. I2C was developed in 1982 by Philips initially supporting 7-bit address and 100 KHz clock. In 1992 protocol was revised, adding 400 KHz fast-mode and an expanded 10-bit address format. Some of the features of I2C protocols are listed below [54].

- Speed mode: 100 KHz or 400 KHz
- I2C requires only two wires
- Addresses up to 1008 slave devices
- Support multi-master system
- Only one extra bit of Meta data

3.3.5.2. Terms Involved

Signal terms used during the communication are tabulated below in Table 3.6, which includes the start and stop conditions to initiate/end the communication, information about the response (acknowledge and no acknowledge) signals and explanation about selecting the appropriate device address for the MPU-6050.

Table 3.6Signal terms.

Interrupt Name	Description
S	Start
AD	Slave address
W	Write (0)
R	Read (1)
ACK	Acknowledge
NACK	Not acknowledge
RA	Master address (MPU-6050)
DATA	Transmit or Receive data
P	Stop

3.3.5.3. Signals

I2C bus consist of two signal lines Serial data line (SDL) and serial clock line (SCL). These lines are bidirectional and attached devices be a master or a slave. If a master device want to talk to a slave device, master will put the slave address on the bus, and the device with the matching address acknowledges the master.

MPU-6050 always act as a slave device. SDL and SCL lines needs pull-up resistors to VCC. Resistor value of 4.7k has been used in the MPU-6050 board as shown in the Figure 3.6. The slave

address of MPU-6050 is 7 bits long. Last bit (seventh) of the address is determined by the logic level on pin 9 of the chip, which allows user to connect two devices on the same bus Figure 3.6.

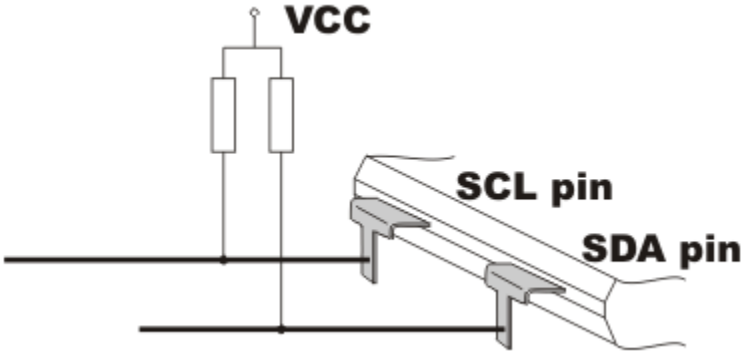

Figure 3.6: Pull-up resistor.

Slave address selection scheme is as shown in Table 3.7.

Table 3.7: Slave address selection.

MSB						**LSB**
1	1	0	1	0	0	**X**

X	**Address**
0	0X68
1	0X69

3.3.5.4. Protocol

I2C data bytes are 8-bits long and there is no restriction on the number of bytes transmitted per data transfer. Every byte should be followed by an acknowledge signal. I2C communication starts with START (S) condition and stops with a STOP (P) condition, explained in Table 3.8. Master

device is responsible to generate both the signals. Although, Slave device can put masters in hold condition sometimes.

Table 3.8: Start and Stop conditions.

Condition	SDA	SCL
Start	HIGH to LOW transition	HIGH
Stop	LOW to HIGH transition	HIGH

Signal illustration of start and stop condition is given in Figure 3.7

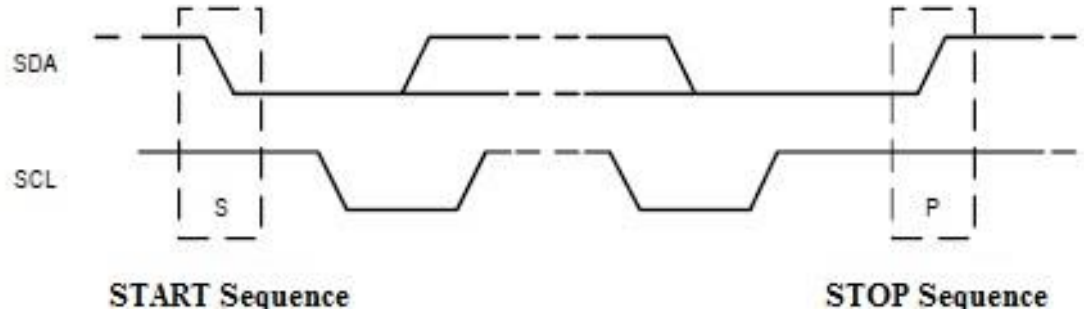

Figure 3.7: Strat and Stop signal illustration.

Acknowledge signal (ACK) is generated by the slave device while the master generates clock for acknowledge signal. Slave can generate an ACK signal by pulling down the data line (SDA) and holding it low during the HIGH portion of the clock pulse. Whole process is explained in the Figure 3.8.

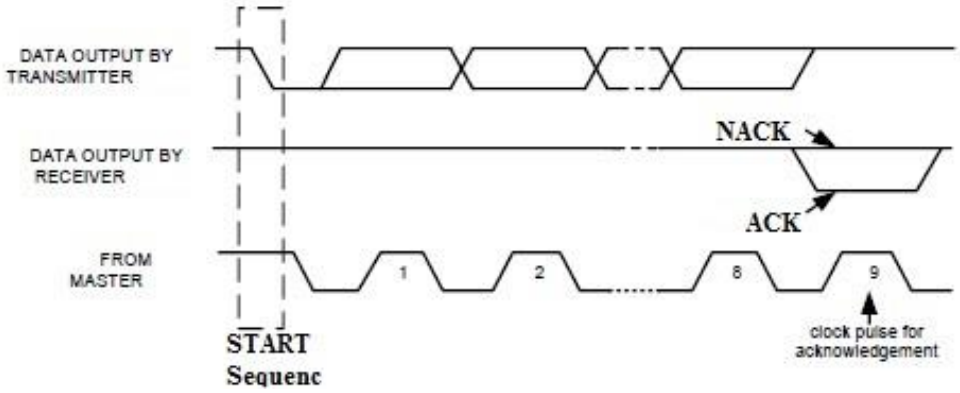

Figure 3.8: Acknowledge/Not Acknowledge signal.

The data is followed by the 7-bit address and a single Read/Write (R/W) bit. At the ninth clock MPU-6050 generates an acknowledge signal. Then the master sends register address (RA) and MPU-6050 acknowledges it, at last master puts register data on the bus. This is followed by the ACK signal generated by the MPU-6050, and the transfer can be ended by the stop condition (P). Burst Write/Read can also be done, sequence for single-byte Read/Write and burst Read/Write is shown in the Figure 3.9 and Figure 3.10 [55].

Single byte Write mode

Master	S	AD+W		RA		DATA		P
Slave			ACK		ACK		ACK	

Burst byte Write mode

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

Figure 3.9: I2C write sequence

Single byte Read mode

Master	S	AD+W		RA		S	AD+R			NACK	P
Slave			ACK		ACK			ACK	DATA		

Burst byte Read mode

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

Figure 3.10: I2C read sequence

3.3.6. Circuit Diagram

MPU-6050 (Figure 3.11) communicates to the host controller using the I2C protocol at 400 kHz clock speed. MPU-6050 generates an interrupt signal every 25mS to the host controller indicating sampled data is now available to read and then microcontroller reads data from the MPU-6050 after every interrupt it receives. Sampling rate of the Accelerometer and gyrometer is set to 1000Hz and DMP interrupt frequency can be programmed as required from 1Hz – 200Hz.

Figure 3.11: MPU-6050

Pin description of the MPU-6050 board is given in the following Table 3.9.

Table 3.9: MPU-6050 pin description

Pin name	Description
VIN	Supply voltage (3.3V – 5V)
GND	Ground
SDA	Serial data line
SCL	Clock input
INT	Interrupt pin

R3 and R4 are used to pull-up the data and clock line for reliable communication. Auxiliary compass can be connected to the pins named AUX_DA (data pin) and AUX_CL (clock) as shown in Figure 3.12. These pins has been left unused because no external compass has been used.

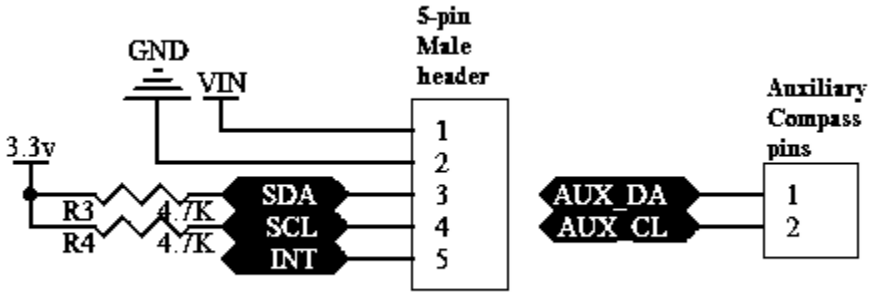

Figure 3.12: MPU-6050 auxiliary compass connection

As shown in the following Figure 3.13 Pin 9 (AD0) is used to select last bit of the MPU-6050 address. Here it is pulled low so, the last bit becomes LOW and thus the address is 0X68.

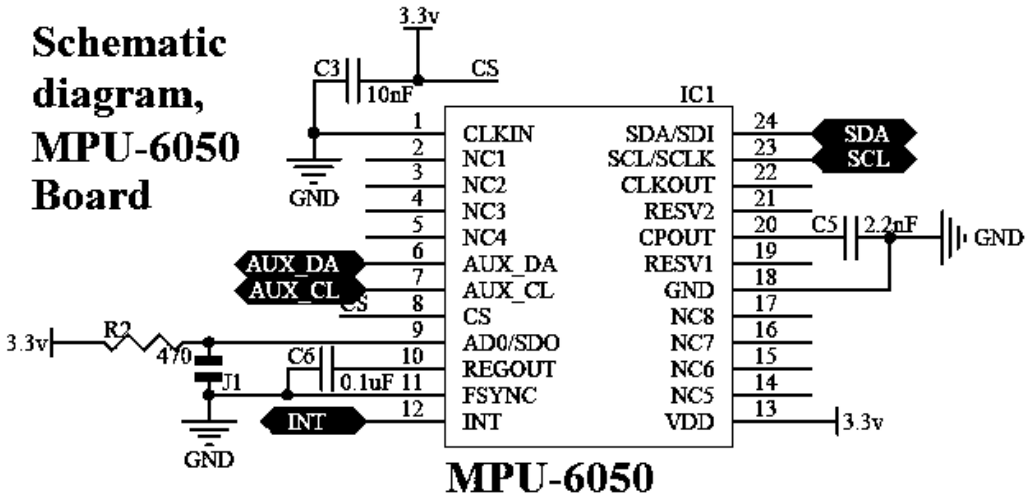

Figure 3.13: MPU-6050 pin connections.

Connecting MPU-6050 with the host microcontroller takes only 3-pins excluding the supply pins as shown in the Figure 3.14. Connection details with pin numbers is provided in the Table 3.10.

Table 3.10: Connection details.

MPU-6050		**Arduino Due**	
Pin name	**Pin #**	**Pin name**	**Pin #**
SCL	**23**	**SCL**	**21**
SDA	**24**	**SDA**	**20**
INT	**12**	**PWM**	**2**

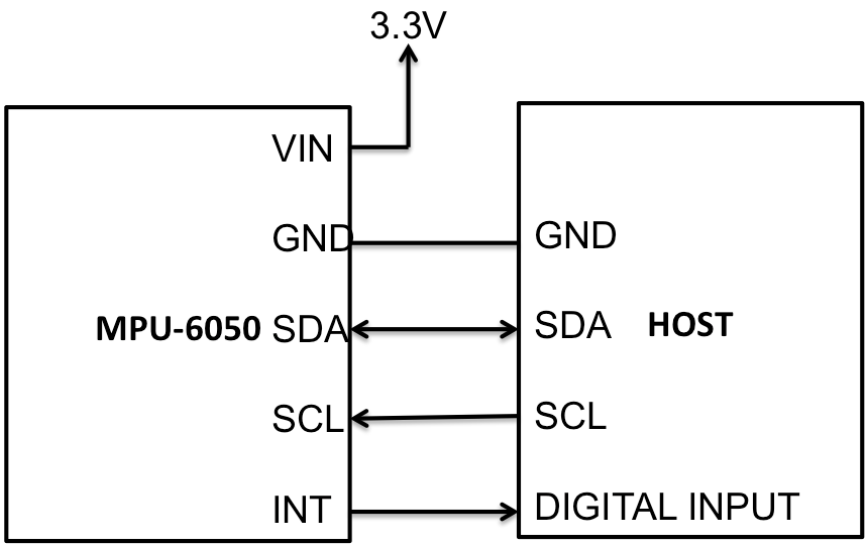

Figure 3.14: MPU-6050 connection with Arduino DUE.

3.3.7. Configuration

Following are the steps performed (Figure 3.15) in the software before reading the data from MPU-6050.

3.3.7.1. Test Connections

Host microcontroller send the request to the MPU-6050 for the device ID, which is 0X34 by default. Comparing it with stored value of 0X34 in the memory, controller knows if the device connected is MPU-6050 or not. If the returned data from the slave device doesn't match with the stored value, no communication occurs. Flow chart for the testing the connection is given below in Figure 3.16.

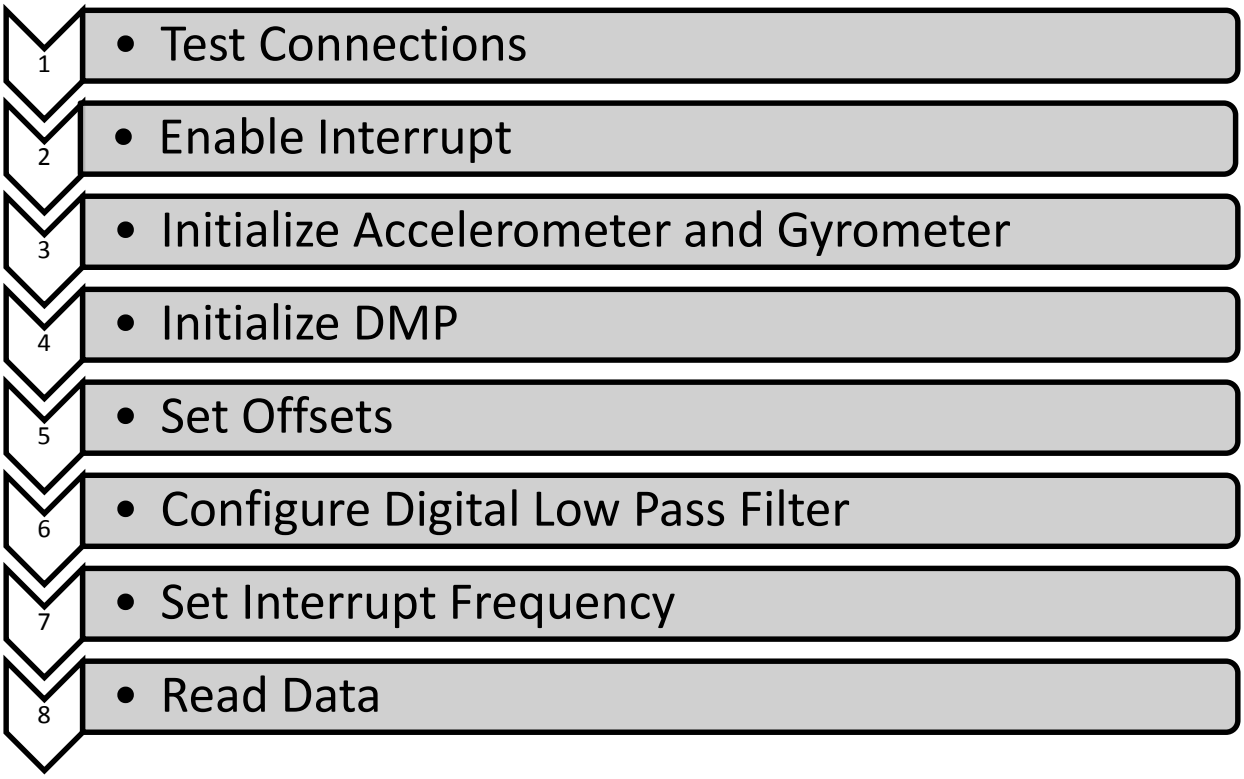

Figure 3.15: MPU-6050 configuration.

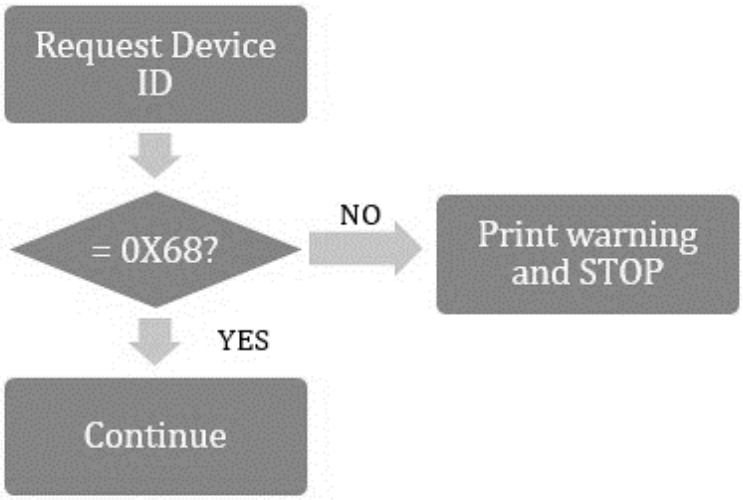

Figure 3.16: Test connection flow chart.

3.3.7.2. Enable Interrupt

MPU-6050 generates an interrupt whenever DMP data is available to read. A digital pin-2 (user-selectable) on Arduino board is used as an input to detect this interrupt. MPU-6050's INT pin changes its state from LOW to HIGH as an indication of an interrupt therefore RISING edge detection interrupt on pin-2 is used.

3.3.7.3. Initialize Accelerometer and Gyrometer

Power on and prepare for general usage. Activate the device and take it out of sleep mode (which must be done after start-up). Set both the accelerometer and the gyroscope to their most sensitive settings, namely ± 2g and ±250 degrees/sec, and sets the clock source to use the X-axis Gyro for reference, which is slightly better than the default internal clock source. Register details to configure accelerometer and gyrometer are discussed below.

Gyrometer is configured using the GYRO_CONFIG register of MPU-6050 as shown in Figure 3.17 [55]. This register is used to trigger gyroscope self-test and set the full scale range. Full scale range can be selected by changing the FS_SEL value provided in Table 3.11.

GYRO_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Figure 3.17: GYRO_CONFIG register.

Table 3.11: Gyrometer full scale range selection

FS_SEL	Full Scale Range
0	$\pm 250\ ^{0}/s$
1	$\pm 500\ ^{0}/s$
2	$\pm 1000\ ^{0}/s$
3	$\pm 2000\ ^{0}/s$

Accelerometer is configured by ACCEL_CONFIG register explained below in Figure 3.18 [55]

ACCEL_CONFIG

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Figure 3.18: ACCEL_CONFIG register.

Selectivity bits are shown in Table 3.12.

Table 3.12: Accelerometer full scale selection

AFS_SEL	Full Scale Range
0	$\pm 2g$
1	$\pm 4g$
2	$\pm 8g$
3	$\pm 16g$

3.3.7.4. Initialize DMP

Digital Motion Processor performs data fusion algorithm on the data measured form accelerometer, gyrometer, and temperature. On chip calculations with the help of lightning fast hardware provides accurate and reliable results. After testing the connections with MPU-6050 next step is to initialize the DMP, this includes writing data into the configuration registers of MPU-6050.

3.3.7.5. Set Offset

It refers to the accelerometer output when there is no motion or gravity is acting on the device. It is also called as the 0g output. In case of gyrometer it is the output when there is no circular acceleration.

Offset can be easily determined by placing the device straight on a rigid body and taking the average of the measured value collected over a certain period of time. After calculations we can write these values for every axis in to the respective offset registers of the MPU-6050. Value in the offset register should have the opposite sign to the calculated average value.

3.3.7.6. Configure Digital Low Pass Filter

MPU-6050 provides an on chip configurable Digital Low Pass Filter (DLPF). The DLPF is configured by DLPF_CFG register shown in Table 3.13[55]. The accelerometer and gyrometer are filtered according to the value of DLPF_CFG as shown in the table below. For the smooth transitions and to avoid any noise in the measurements DLPF_CFG has been set to 3, which provides sampling rate of 1 kHz and bandwidth of 44Hz and 42Hz for accelerometer and gyrometer respectively.

Table 3.13: DLPF_CFG register

AFS_SEL	Accelerometer ($F_s = 1kHz$)		Gyroscope		
	Bandwidth (Hz)	Delay (ms)	Bandwidth (Hz)	Delay (ms)	F_s (kHz)
0	260	0	256	0.98	8
1	184	2.0	188	1.9	1
2	94	3.0	98	2.8	1
3	44	4.9	42	4.8	1
4	21	8.5	20	8.3	1
5	10	13.8	10	13.4	1
6	5	19.0	5	18.6	1
7	RESERVED		RESERVED		8

3.3.7.7. Set Interrupt Frequency

DMP output frequency is calculated easily using the Equation 3.2.

$$Interrupt\ Frequency = \frac{200}{(1 + Value)} Hz \tag{3.2}$$

Value = number in the register.

Going faster than 100Hz (value = 0x00 = 200Hz) tends to result in very noisy data. It is important to make sure the host processor can keep up with reading and processing the FIFO output at the desired rate. Putting value = 4 gives output rate of 40Hz.

3.3.7.8. Read Data

Last step is to read data from the MPU-6050. After receiving an interrupt host microcontroller wait for the correct available data length and then get data from the FIFO buffer. Default FIFO packet length is 42 bytes. Host microcontroller continuously check the FIFO count value to prevent the overflow. If an overflow does occur, FIFO is reset to continue safely.

3.3.7.9. Calculation of Yaw Rate

Yaw rate (z-axis gyrometer) of the radio controlled (RC) model car is calculated using the MPU-6050 module. MPU-6050 is configures to most sensitive full scale range i.e. $\pm$ 250 °/sec using the FS_SEL register as discussed in the section 3.3.7.3. 16-bit ADC provides the yaw rate in Two's compliment form. Sensitivity scale depends on the selection of full scale range, relation between both parameter is given in Table 3.14.

Table 3.14: Gyrometer full scale range and sensitivity scale factor calculation.

FS_SEL	Full Scale range (°/sec)	Sensitivity scale factor (LSB/(°/sec))	Measured Data length (bits)
0	$\pm$250	131	16
1	$\pm$500	65.5	16
2	$\pm$1000	32.8	16
3	$\pm$2000	16.4	16

Final value is calculated in °/sec, by dividing the 16-bit measured data with the sensitivity scale factor, shown in the Equation 3.3.

$$Yaw\ rate = \frac{Measured\ Data}{Sensitvity\ scale\ factor} \tag{3.3}$$

Example Calculation, Considering FS_SEL = 0 gives us full scale range of ±500 °/sec and data length is fixed for all the value i.e. 16-bit. MSB of the measured data tells sign whereas rest of the 15-bit reflects the magnitude. Maximum value a 15-bit number can hold is $2^{15} = 32768$. By plugging in the maximum value of measurement and yaw rate in to the Equation 3.3, sensitivity scale factor obtained in Equation 3.4 can be verified with Table 3.14.

$$Sensitivity\ scale\ factor = \frac{\pm 32768}{\pm 250} = 131 \tag{3.4}$$

If the measurement data is 12450 then the calculated yaw rate can be calculated as shown in Equation 3.5 below,

$$Yaw\ rate = \frac{12450}{131} = 95^0/\text{sec} \tag{3.5}$$

Same procedure is followed to calculate the pitch and roll of the vehicle. In the last, obtained result is converted into radians/sec by multiplying it with the conversion factor of 0.0174532925.

3.3.8. Velocity Measurement

Measurement of velocity is done using the data from both the accelerometer (X-axis) and the wheel RPM measurement circuit. Measured data is then combined by implementing Kalman filter, which calculates the optimal measurements. Kalman filter is discussed in details in the following section. In this section RPM measurement and acceleration measurement is deliberated.

3.3.8.1. RPM Measurement

First approach used to count the wheel RPM was using a quadrature wheel encoder, which gives 128 Pulses per revolution (PPR). Microcontroller checks how many pulses it has received every second and then calculates RPM using this data. Considering the highest RPM of RC model car i.e. 4000, generates 8533 pulses per second. Other than calculating RPM, microcontroller also performs data acquisition from other sensors, Kalman filter calculations, motor control, PID calculations, and sending data to the host computer. All these calculations along with the RPM count decreased the performance of Kalman filter because of the high interrupt frequency generated by wheel encoder. Also high resolution was not required at the high RPM.

To solve this problem a simple Hall circuit with three magnets on a rear right rotating wheel of the RC model car (Figure 3.19) is used to measure the RPM of vehicle. This setup produces three pulses per rotation. At the maximum RPM (4000) this circuit produces only 200 pulses, which is significantly lower than what wheel encoder provided with the 128 Pulses per revolution (PPR). Applying this approach did not affect the other calculations. Pulses generated by the circuit is then given to the Arduino microcontroller at pin#7 (configured as interrupt input) shown in Figure 3.20.

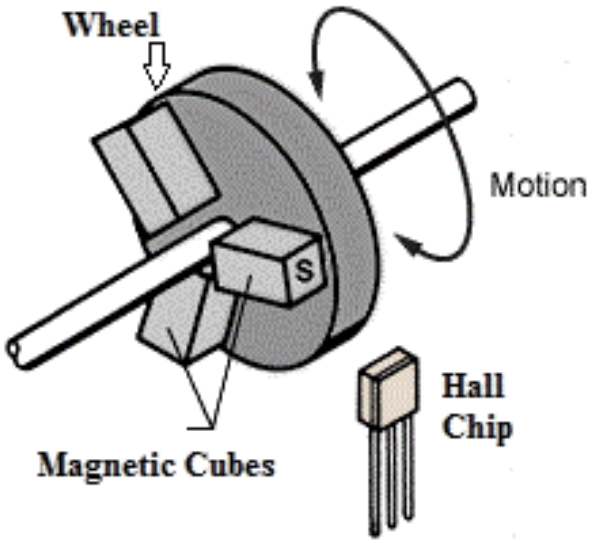

Figure 3.19: RPM count scheme.

Pulses received by Arduino microcontroller generates an interrupt at pin#7. Microcontroller keep count of these interrupts and calculate frequency based on the count. Frequency is calculated by dividing this count by 3 as shown in Equation 3.6 and multiplying the result with tire circumference (0.216 meter) gives us the speed in m/s, as shown in the Equation 3.7 below.

$$RPS = \left(\frac{Count}{3}\right) \tag{3.6}$$

$$v_x = RPS * Tire\ circumfrence \tag{3.7}$$

where, RPS is revolution per second.

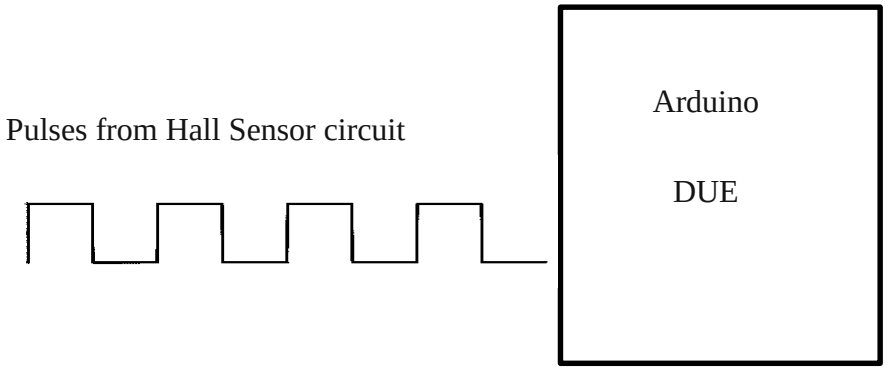

Figure 3.20: RPM count.

Flow chart of the microcontroller to calculate RPM is shown in the Figure 3.21 below.

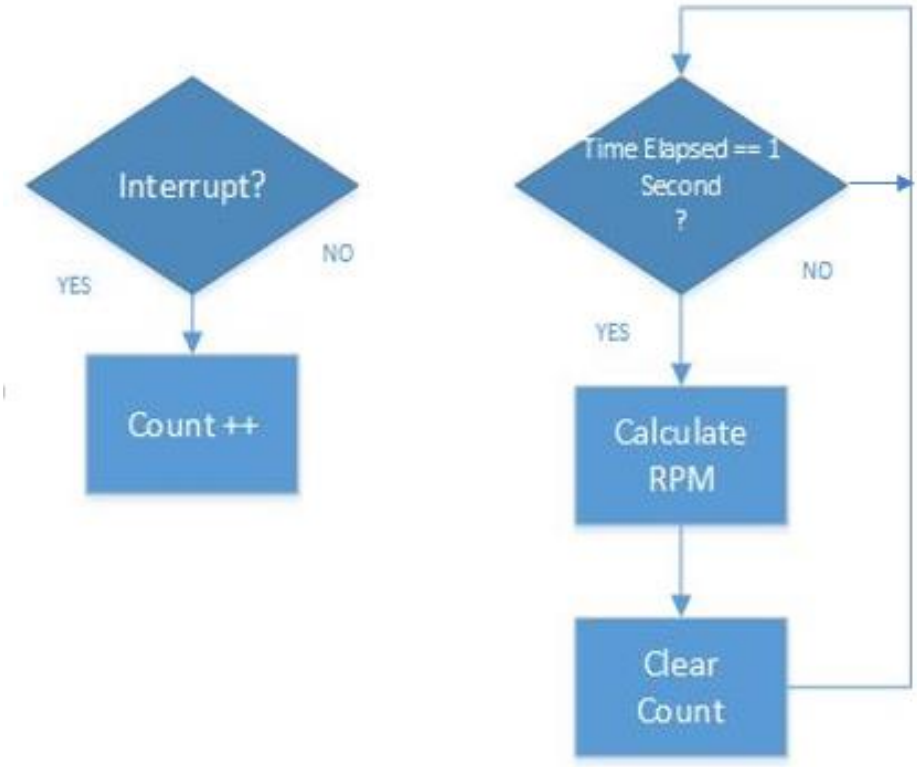

Figure 3.21: Flow chart to calculate wheel RPM.

Acceleration measured by the accelerometer does not only varies with the motion of the object but also with the change in orientation or gravity. To get the linear acceleration (only due to motion) these effects should be subtracted from the actual readings. To calculate the gravity, rotation of the object around the axis should be known all the times. There are several methods to calculate orientation and rotation of the object, discussion on topic is beyond the scope of this thesis. MPU-6050's DMP engine provides an accurate estimate of the orientation using data fusion on Accelerometer and Gyrometer data. Orientation is measured in quaternion set and then used to measure the real acceleration (gravity compensated). Steps to calculate the real acceleration are shown in Figure 3.22.

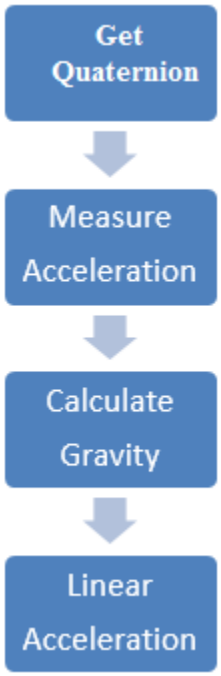

Figure 3.22: Acceleration flow chart.

First, orientation is calculated using the quaternion data and second step is to measure the acceleration. Third step is to measure the gravity vector using quaternion and in the last step gravity vector (calculated in third step) is subtracted from the acceleration measured in the second step in order to get the real acceleration.

Kalman filter (explained in Chapter-5) is used to estimate the vehicle's lateral and longitudinal velocity using the measured acceleration and wheel RPM. For longitudinal velocity estimation both the acceleration (x-axis) and the wheel RPM are used, whereas only acceleration (y-axis) along with the roll, pitch and, yaw are used to calculate the lateral velocity as shown in Figure 3.23.

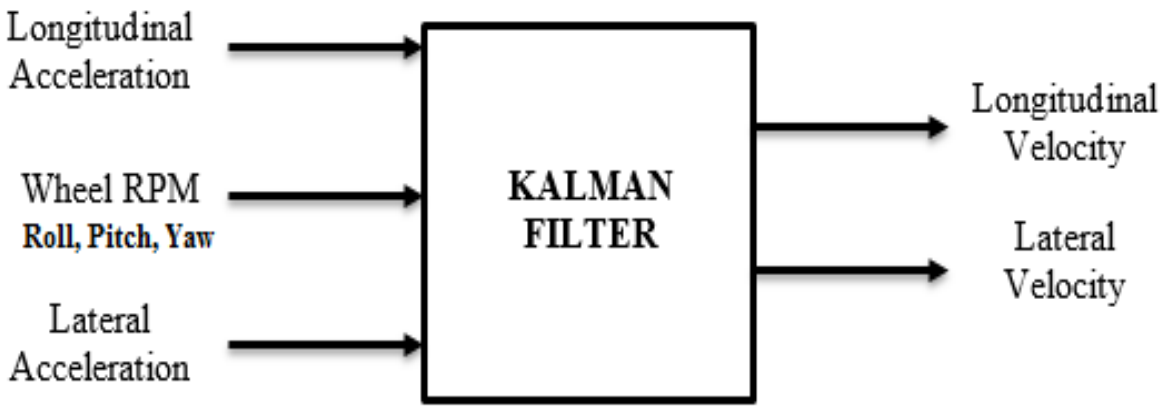

Figure 3.23: Velocity calculation

3.4. ZIGBEE

Figure 3.24: ZigBee module

ZigBee modules are low power and low cost device used to deliver data between remote devices as shown in Figure 3.24 [56]. It uses 2.4 GHz frequency and follows the ZigBee protocols to communicate with each other. In general, these devices are mainly used in the Wireless sensor networks (WSN) for the reliable data transfers and can act as both receiver and transmitter. In the project two dedicated Zigbee modules are used, one as a transmitter and other as receiver. Some of the main reasons to use ZigBee modules are: -

- Line of sight (LOS) range up to 1600m
- Low transmit power < 50mW
- Support many networking topologies
- Small form factor
- Extensive command set
- Fully configurable

Communication between Host computer and microcontroller conducted with ZigBee module. It uses simple UART (universal asynchronous receiver transmitter) interface to connect with microcontroller board. No external components are required for connections. Any device with a UART interface can connect directly to the pins of ZigBee module as shown in the Figure 3.25.

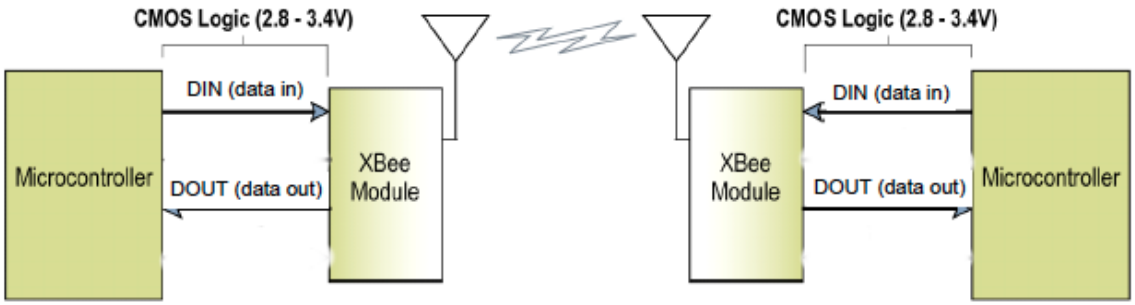

Figure 3.25: Connection diagram of ZigBee module.

3.4.1. Serial Data

Data enters into the UART module through the DIN (pin 3) as an asynchronous signal. The signal should be high when no data is being transmitted. Each data bytes consist of a start bit (LOW), stop bit (HIGH), and 8 bit data as shown below in Figure 3.26. For the reliable communication parity checking and timing tasks are also performed by the UART module.

Figure 3.26: I2C serial data structure.

3.4.2. Serial Buffers

ZigBee module has buffers to collect received serial data. Function of a receive buffer is to collect the incoming data and hold it until it can be processed. If a large amount of data has to interchanged, CTS/RTS flow control can be used to avoid any loss of data. On the other hand, transmit buffer collect data via the RF link that will be transmitted out the UART. Data flow diagram is illustrated in Figure 3.27.

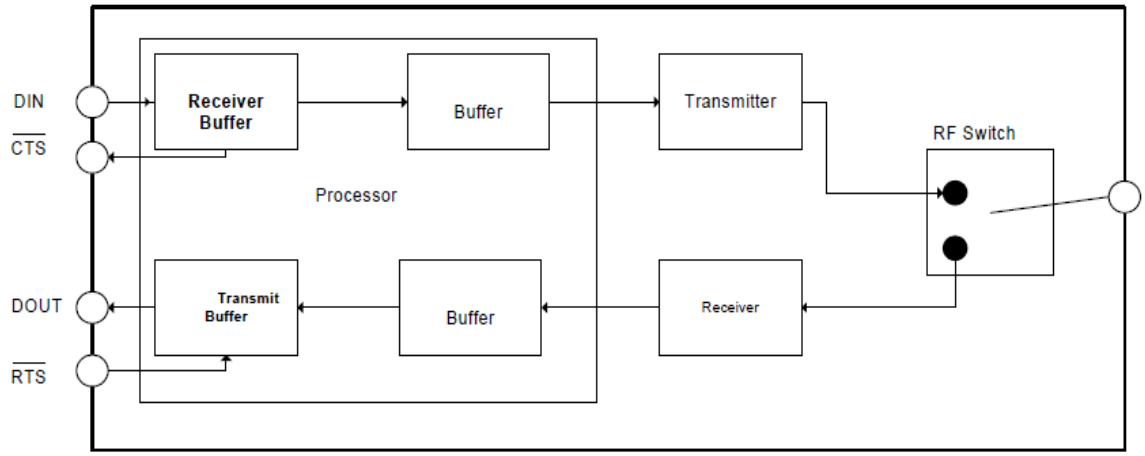

Figure 3.27: UART data flow.

3.4.3. Connections

Microcontroller uses 115200bps (bits per second) baud rate to communicate with ZigBee module, connection diagram is shown in Figure 3.28. Data send by the microcontroller is received by another ZigBee module connected to the computer using the same baud rate i.e. 115200bps.

The ZigBee Explorer shown in Figure 3.29 accepts 5V and takes care of the 3.3V regulation required by the Zigbee module to operate. Board also includes signal conditioning circuit, and

basic activity indicators such as Power, RSSI and DIN/DOUT activity LEDs.

Computer connects with ZigBee module by using USB ZigBee explorer board, shown in Figure 3.30. This module has a FT232 chip that act as a signal translator between the Computer's USB and ZigBee's UART, converting USB port to serial port. Drivers are needed to be installed on the computer in order to use the USB ZigBee explorer.

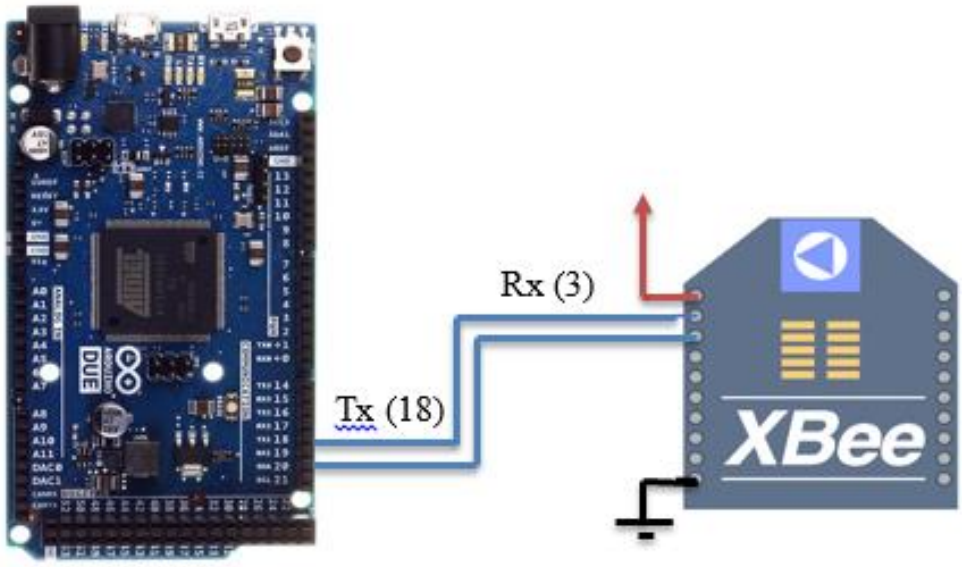

Figure 3.28: ZigBee connection diagram.

Figure 3.29: ZigBee explorer.

Received data from microcontroller LabView has been used to design the GUI (graphical user interface). X-CTU software provided by Digi international (free of cost) can be used to change the baud rate of ZigBee module, modify configuration parameters and update the firmware version (explained in Chapter 4).

Figure 3.30: ZigBee connection with computer.

Zigbee module (upside down) on the general purpose PCB is shown in Figure 3.31.

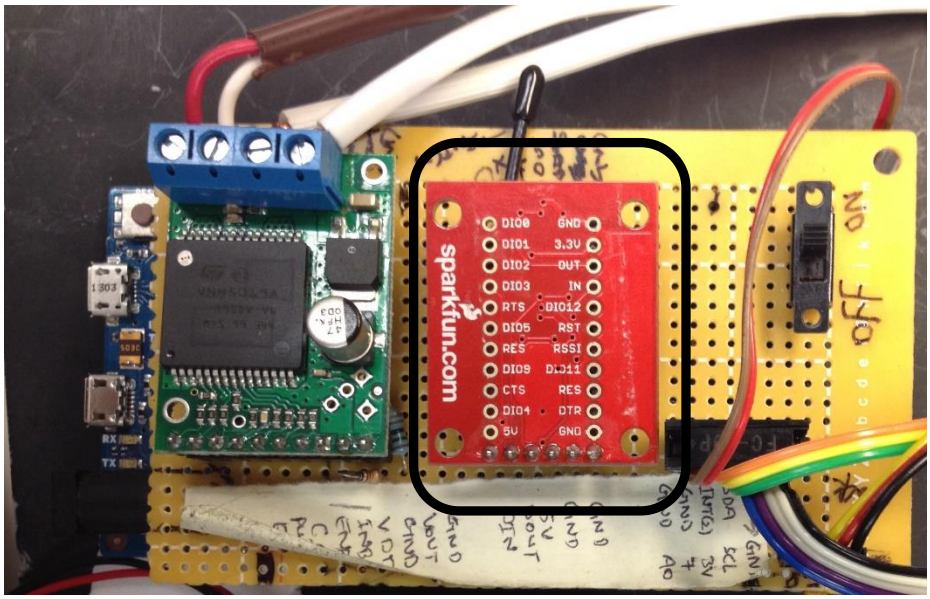

Figure 3.31: Zigbee module on PCB

3.5. MOTOR DRIVER

Microcontroller does not provide enough current at the output pins to drive the motor. The motor draws a much higher current. Connecting directly will result in not working motor and destroying the microcontroller due to high currents. Therefore, motor driver board is used to serve the purpose. Motor driver amplifies the power levels and can control both the speed and rotation of the motor.

Motor driver board includes ST's high-power VNH5019 motor driver IC, a fully integrated H-bridge configuration used for bidirectional speed control of a single brushed DC motor. The board incorporates most of the required components for a typical application, including pull-up resistors, current-limiting resistors, and a field effect transistor (FET) for reverse battery protection.

3.5.1. Over Voltage Protection

Overvoltage protection of the motor driver board is set at 27 V, sometimes it can trigger at voltages as low as 24 V, Two batteries 12V each, in series are used to run the DC motor.

Figure 3.32: Motor driver board.

3.5.2. Features

VNH5019A shown in Figure 3.32 [57] is a full bridge single motor driver with the following features:

- Automotive Grade: compliance with automotive electronics council (AEC) guidelines
- Maximum Output Current: 30 Amperes
- Maximum battery voltage: -16 V to 41 V
- PWM operation up to 20 kHz
- CMOS compatible inputs
- Under voltage and over voltage protection
- PWM up to 20 kHz
- Current sense output
- Low standby power consumption
- Operating temperature: -40 C to 150 C

3.5.3. Pin Details

Pin out diagram for the motor driver board is shown in the Figure 3.33 [57].

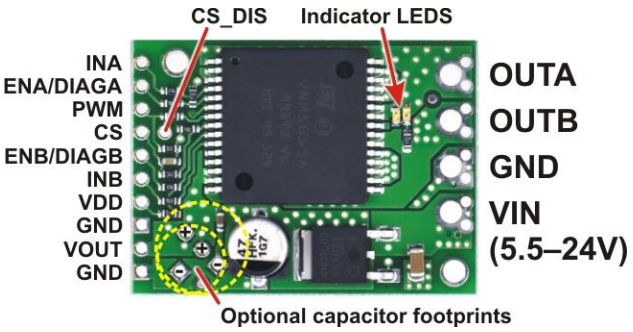

Figure 3.33: Motor driver pin details.

Only a few of the pins are used and their functions are descripted in the Table 3.15.

Table 3.15: Motor driver pin description.

Pin Name	Description
IN A	Motor direction control input
IN B	Motor direction control input
PWM	Speed control input
VDD	Board supply voltage
GND	Ground connection
VIN	Motor power supply voltage
OUTA	Motor input
OUTB	Motor input
CS	Current Sense output
CS_DIS	Current Sense Enable/Disable pin

3.5.4. Schematic Diagram

Schematic diagram of the motor controller board is as shown in the Figure 3.34.

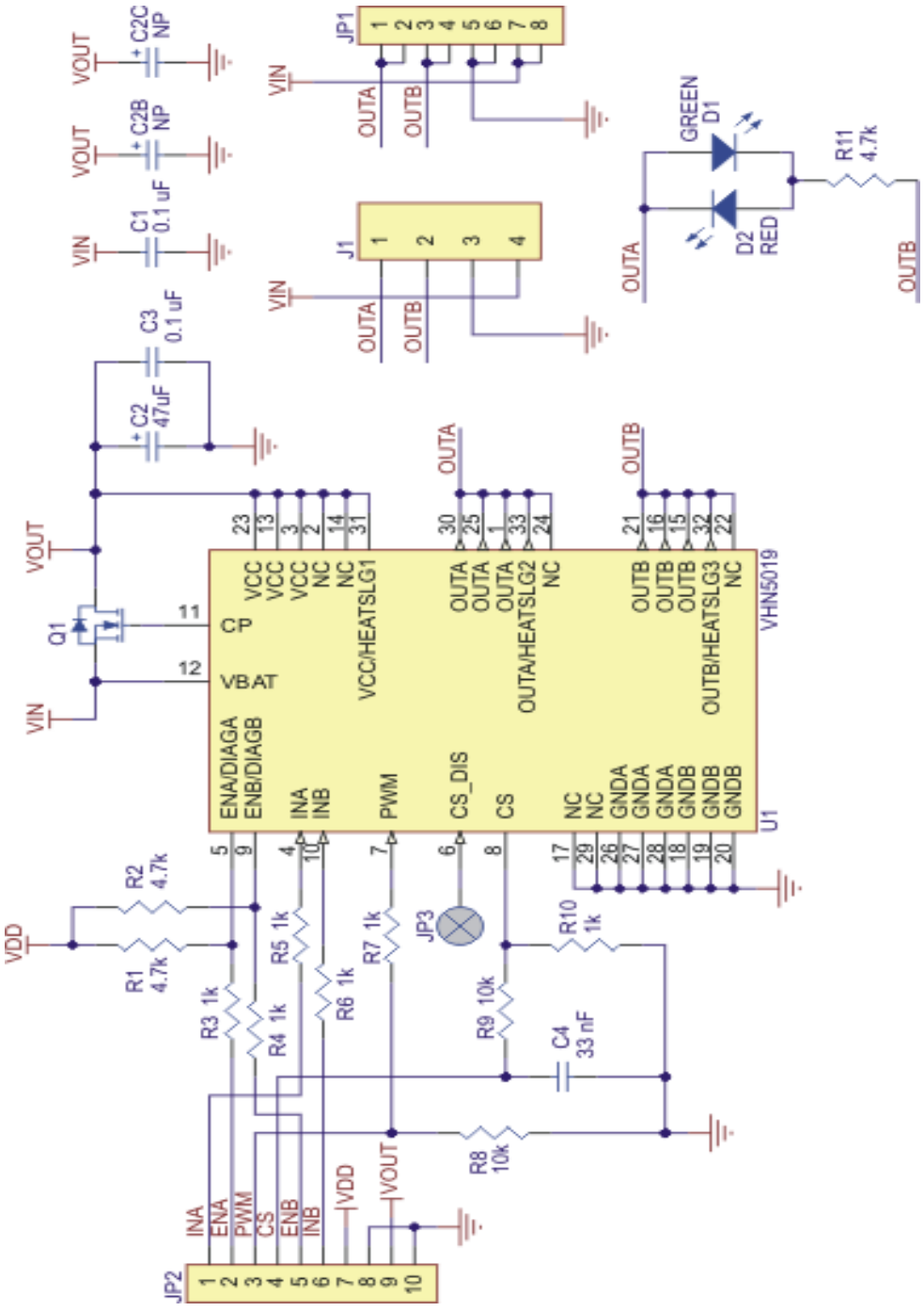

Figure 3.34: Motor driver schematic diagram.

3.5.5. Motor Control

Motor control lines are on one side of the board and can be easily connected to the microcontroller board as shown in Figure 3.35 using the female jumper wires. Board supports non-interrupt able operation up to 24 V of operation. Although, using supply voltage above 27 V may reset the motor controller board. Logic input voltage VDD can be in between 2.5 – 5 V, this makes it compatible with to use with ARM controller that operates at 3.3 V. Connection diagram of the motor driver board and the microcontroller is shown in the following Figure 3.35 [57].

Input A (INA) and Input B (INB) controls the direction of the motor. These two lines are connected to the digital output pin-4 and pin-5 of Arduino DUE microcontroller board. Changing the logic level on these two lines controls the direction of the motor. Truth table for the motor operation is shown below in Table 3.16.

Table 3.16: Motor direction control.

Input A	Input B	Motor Direction
HIGH	LOW	Clockwise
LOW	HIGH	Anti-clockwise
HIGH	HIGH	Stop
LOW	LOW	Stop

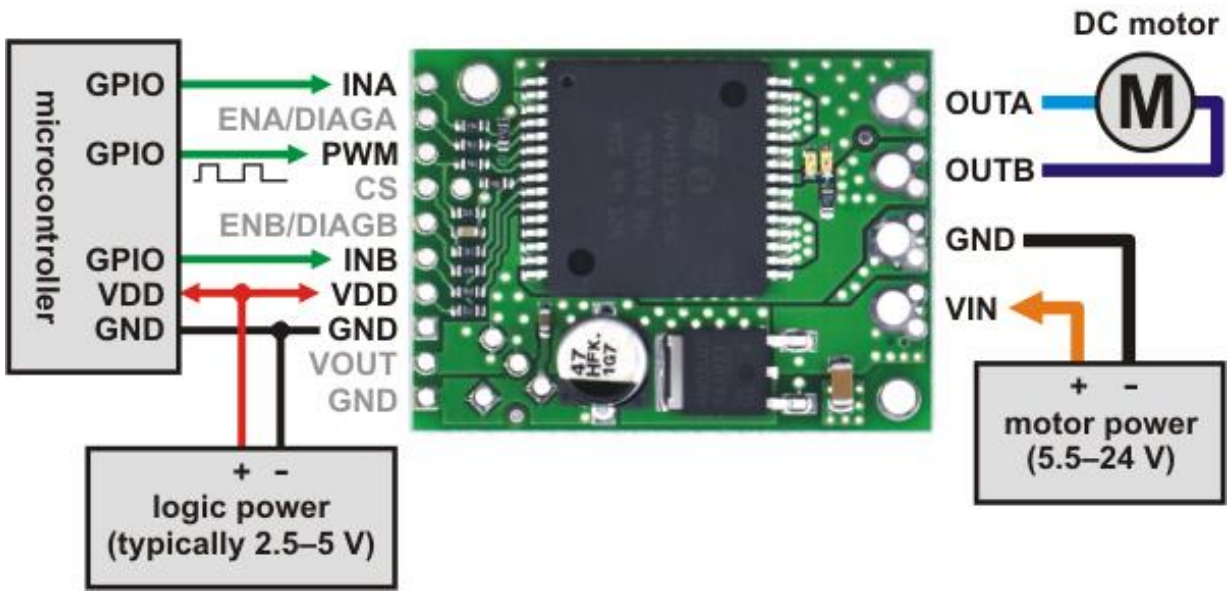

Figure 3.35: Motor driver connection layout.

Using only two digital lines of microcontroller, connected to the INA and INB pins motor direction can be controlled. Varying the duty cycle of PWM signal generated by the microcontroller on pin-6, controls speed of the motor (explained in section 3.5.5.1). In order to avoid any unwanted motion (during the absence of PWM control signal) the PWM pin is pulled low on the board by default. Pulling the CS_DIS pin low enables current sensing capability of the board. A 1K resistor on the chip select (CS) pin provides 140mV/amperes and a 10K resistor in series with the output limits the current flowing in to the microcontroller.

3.5.5.1. Pulse Width Modulation (PWM)

Speed of the motor is controlled using the inbuilt PWM generation module of the microcontroller. Microcontroller generates PWM of 20 KHz frequency with 16-bit of resolution. Which means, duty cycle of the wave is varied from 0% to 100% by varying the register value from 0 to 65536. Thus providing resolution of 0.0015%. Some basics of the PWM:

Microcontroller is only able to generate digital signals High (VDD) and LOW (GND, 0 volts). PWM technique provides analog results with digital means, signal switched between HIGH and LOW simulates voltages in between VDD and 0 Volts by varying the time signal spend ON versus the OFF (duty %) as shown in Figure 3.36.

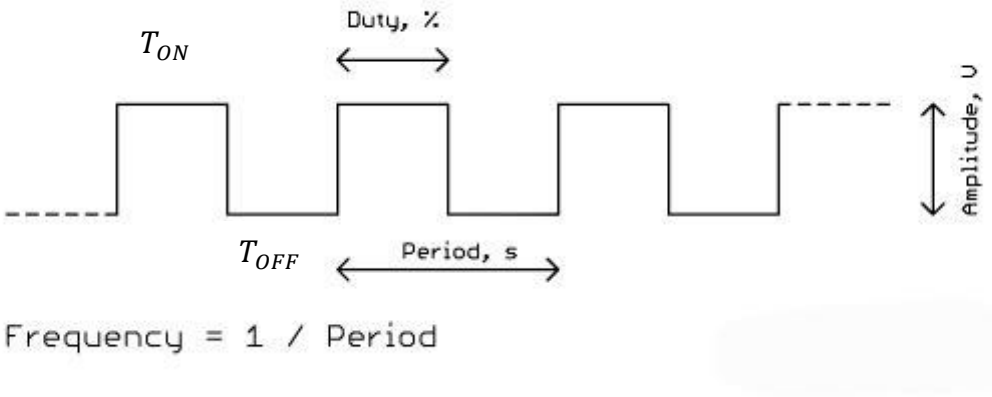

Figure 3.36: Pulse width modulation (PWM) wave.

More time spend when the signal is high the more will be the duty cycle. Duty cycle of a PWM wave can be calculated using Equation 3.8. Where, the time duration while the signal is HIGH is T_{ON} and T_{OFF} is the time duration when the signal is LOW.

$$Duty\ Cycle = \left(\frac{T_{ON}}{T_{ON} + T_{OFF}}\right) * 100 \tag{3.8}$$

Varying the duty cycle also vary the average output voltage delivered to the load, which can be calculated using Equation 3.9. This way by just changing the T_{ON} period of the PWM wave output voltage is varied and the motor speed is controlled by changing the duty cycle of the PWM wave.

$$V_{AVG} = Duty\ cycle * V \tag{3.9}$$

where, amplitude of the PWM wave is V and V_{AVG} is the average output voltage of PWM wave as shown in Figure 3.36.

Motor driver board connected to the microcontroller is shown in the Figure 3.37.

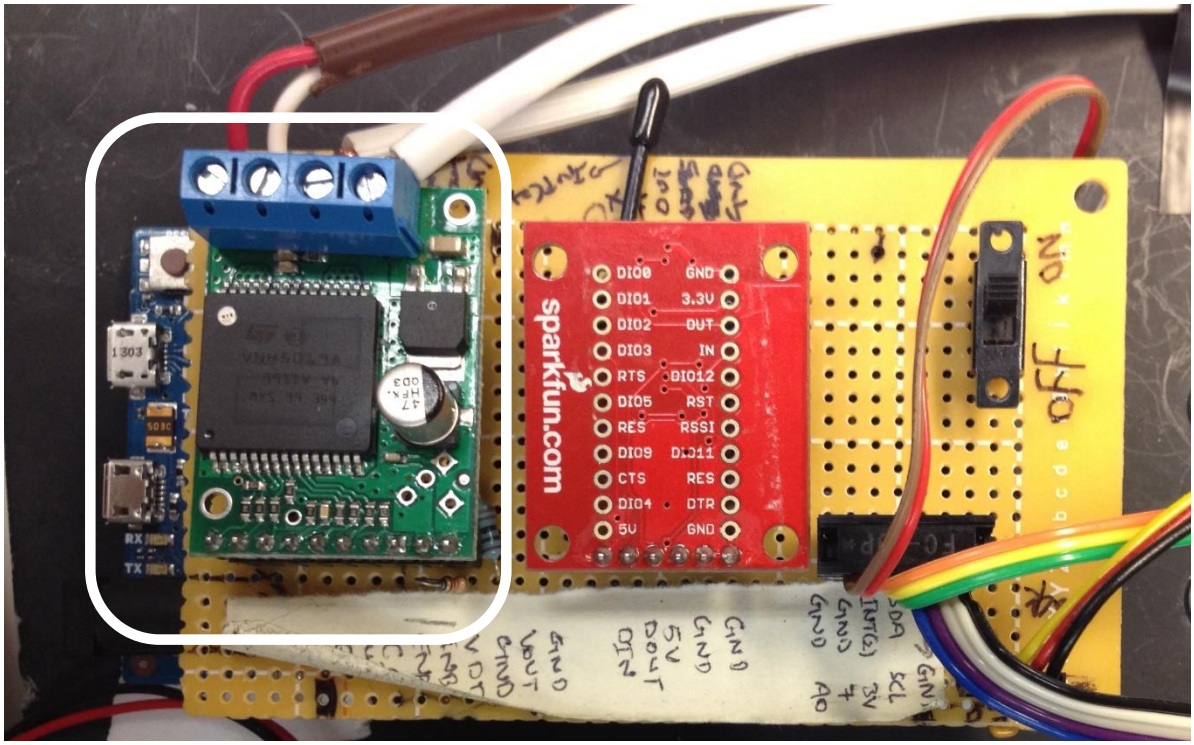

Figure 3.37: Motor driver board

3.6. HALL SENSOR

Unipolar digital output Hall Sensor is used to count the RC model car's wheel RPM. It generates pulses which is then provided to the Microcontroller to count the wheel RPM.

3.6.1. Hall Effect

Hall Effect was discovered by E.H. Hall in 1879. Electrical current flowing through a conductor in the presence of a magnetic field produces a measureable voltage between the two sides of the conductor. Generation of this transverse voltage is called Hall voltage as shown in Figure 3.38.

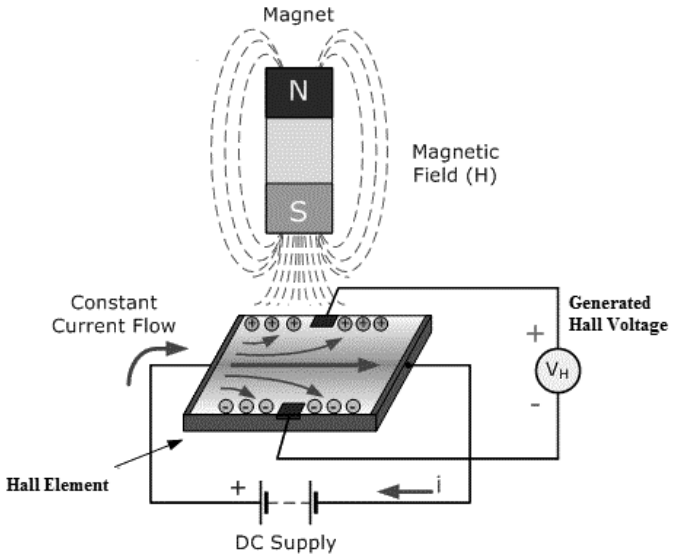

Figure 3.38: Hall Effect.

Magnetic field acting on a constant current carrying semiconductor causes to push the charge carriers, electrons and holes, to either side of the semiconductor. This deflection of the charge carriers is due to the involvement of magnetic force they experience passing through the semiconductor. DC supply is to generate constant current across the semiconductor chip. Hall voltage is denoted by V_H.

3.6.2. Hall Effect Sensor

Hall Effect sensors produces an output voltage (V_H) proportional to the strength of the magnetic field passing through the semiconductor. The output voltage is usually very small, only a few microvolts even under the impact of strong magnetic field. Therefore most of the commercial available devices comprises of built-in amplifiers, voltage regulators, logic switching circuit to improve the sensitivity and output voltage magnitude. A typical Hall Effect sensor is shown in Figure 3.39.

Figure 3.39: Hall Sensor.

3.6.3. Types of Hall Sensor

Hall sensors deliver either analog or digital output. Output of the analog Hall sensor varies linearly with the magnetic field, whereas digital Hall sensors outputs a HIGH (VDD) signal when the magnetic field around the semiconductor crosses a certain threshold value and LOW (GND) otherwise [58].

3.6.3.1. Analog

Output of the analog sensor is taken directly from the amplifier placed after the Hall chip shown in Figure 3.40. Increase in the magnetic field increases the output of amplifier until it starts to saturate, limited by the power supply. Reaching the saturation means amplifier's output will not increase with the further increase in magnetic field as illustrated in Figure 3.41 [59].

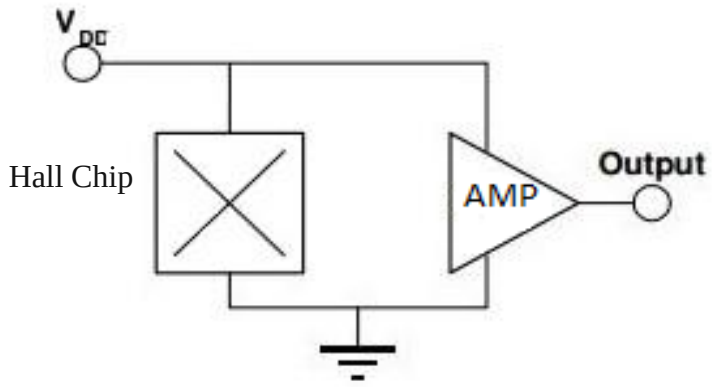

Figure 3.40: Analog Hall Sensor.

Equation relating the output Hall voltage with magnetic field, current and semiconductor thickness is given in the following Equation 3.9. Although there are more terms involved while calculating the hall sensor output voltage but this is the basic equation.

$$V_H = R_H \left(\frac{I_c}{t} * B\right) \tag{3.10}$$

Where, R_H is Hall Effect co-efficient, I_c is the current flow through the sensor in amps, t is the thickness of sensor in mm, B is the magnetic flux density in tesla.

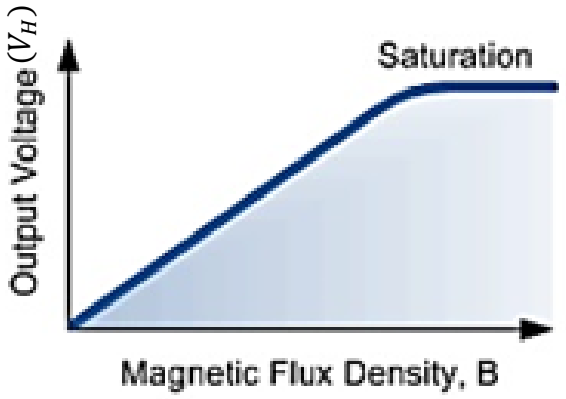

Figure 3.41: Hall voltage saturation.

3.6.3.2. Digital

Digital sensors outputs either HIGH or LOW signal depending upon the strength of the magnetic field. It consist of a few more components than the analog type to make it work like a digital sensor shown in the Figure 3.42 [59].

Accumulation of a Schmitt-trigger threshold detector with built-in hysteresis, offers the Hall Effect circuit digital output abilities. Triggering circuit provides a smooth transition from LOW to HIGH when the applied magnetic flux density exceeds a certain threshold. Built-in hysteresis cancels out oscillations (spurious switching of the output) by defining a magnetic dead zone in which switching action is deactivated after the threshold value is crossed. Last stage of the circuit is output transistor that gives the switch digital logic compatibility. Output transistor is capable to drive many loads, including relays, LED's and other digital circuitry.

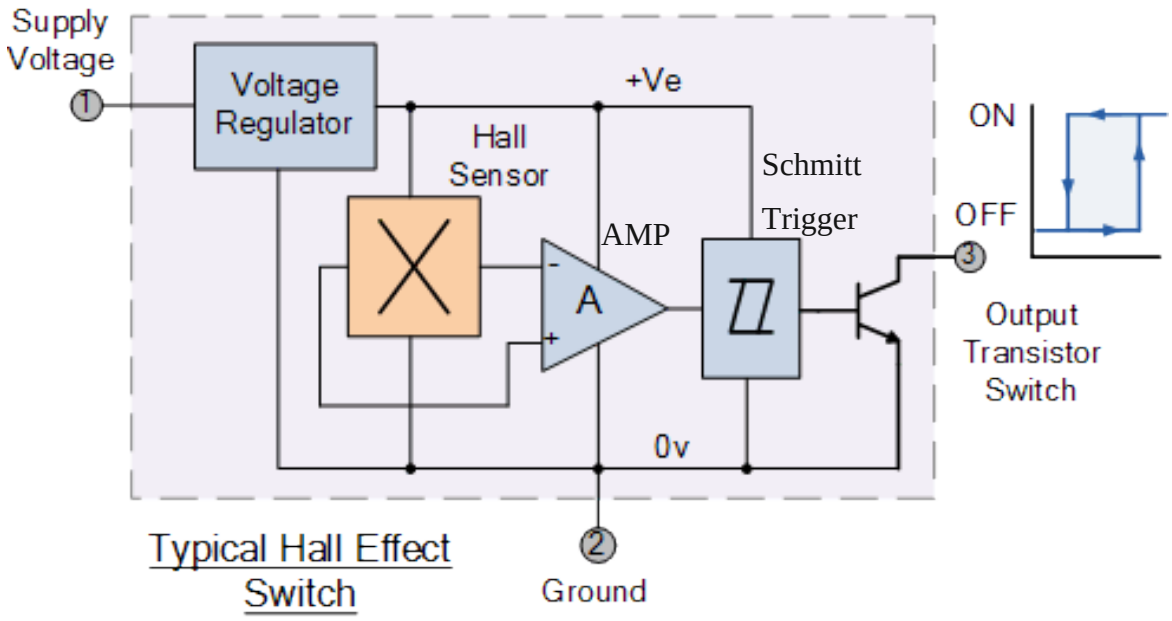

Figure 3.42: Typical Hall Effect switch.

3.6.4. Circuit Diagram

Hall sensor require just as many as one component to make it operational. Only one pull-up resistor at the output pin is required as shown in Figure 3.43. Signal obtained from the Hall sensor is then supplied to the buffer (LM-324) before applying it to the microcontroller.

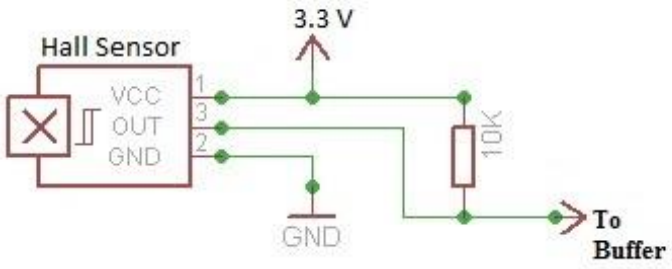

Figure 3.43: Hall Sensor connections.

RPM count circuit attached to the RC model car is shown in the Figure 3.44 below.

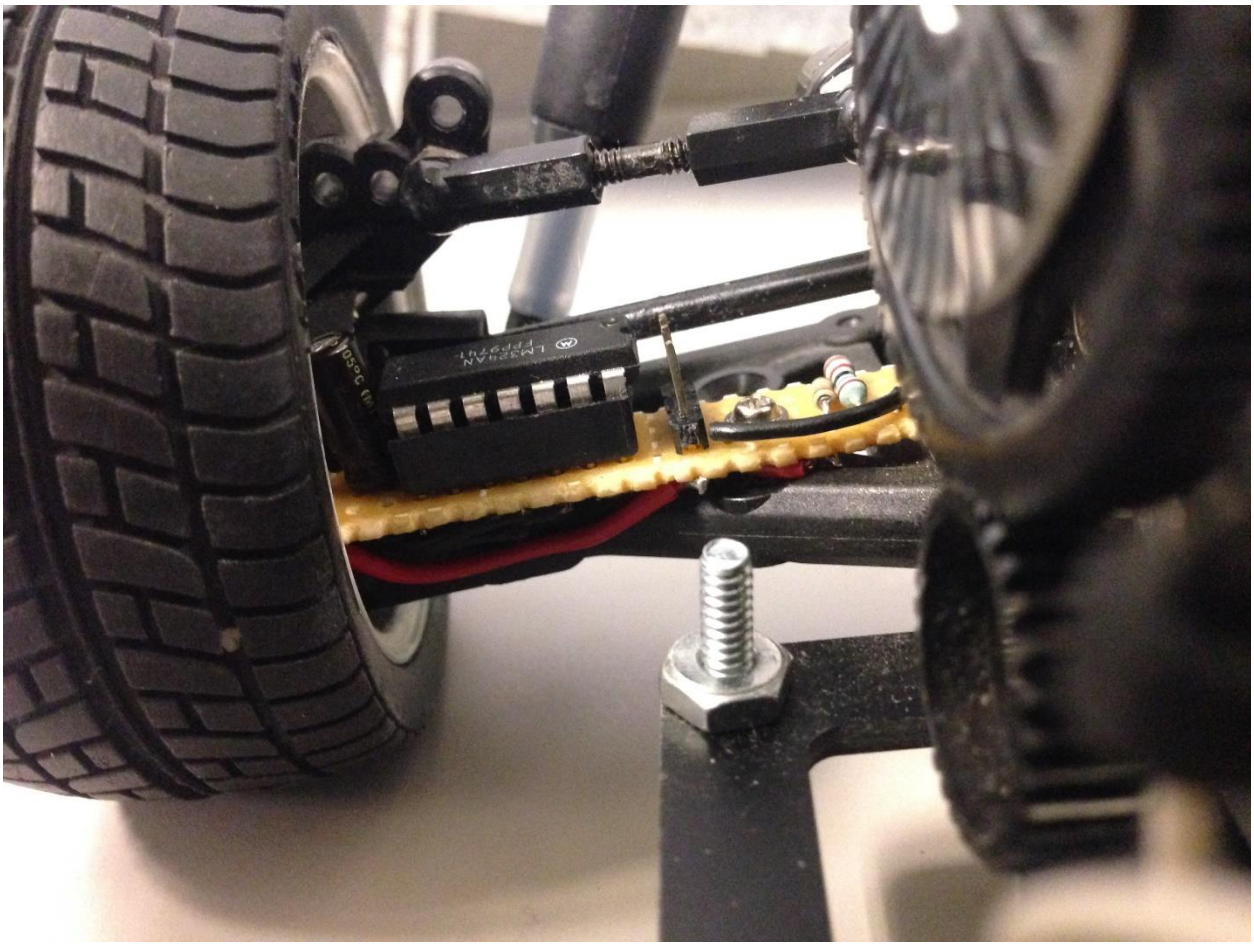

Figure 3.44: RPM measurement circuit attached to the rear right wheel mechanism

3.7. PWM TO ANALOG

Custom made simple passive RC filter is designed to measure the vehicle's steering angle. This circuit takes PWM signal as an input from the RF receiver of model car and converts it into the smooth analog voltage. Microcontroller converts this analog signal in to the digital. Converted voltage is proportional to the duty cycle of the incoming PWM.

3.7.1. Steering Angle Acquisition

Position of servo motor is controlled by varying the duty cycle of the PWM wave applied to the servo motor. In other words, to control the position of servo motor, pulse of varying length is applied which correspond to the angle of the motor. Time period of the control pulse is 20ms as shown in Figure 3.45. Pulse width is varied from 1mS to 2mS to sweep through -90 degrees to +90 degrees, which is total of 180degrees Figure 3.46.

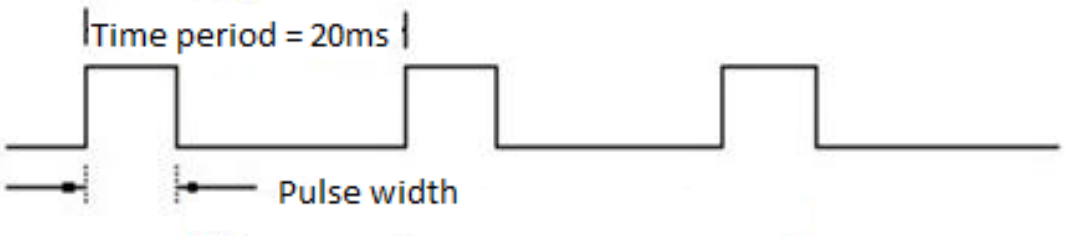

Figure 3.45: Time period of control pulse.

RC model car receiver generates a constant 52Hz (19.2ms) frequency with varying duty cycle (which is controlled using the transmitter (2.4.1.1) to control the steering angle of the RC model car. Because the receiver used is bit old, it generates 52Hz instead of 50Hz (checked on oscilloscope). This pulse is applied to the servo motor directly and is enough to drive the motor without the need of a motor driver board. Angle of the servo motor in accordance with the pulse width applied is shown in the Figure 3.46.

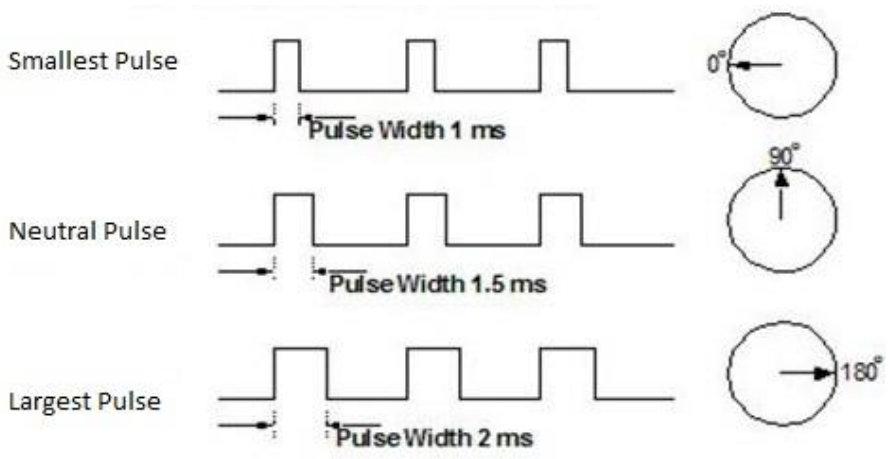

Figure 3.46: Servo motor position control.

Connection of the servo motor with the RC model car receiver is shown in Figure 2.10. Servo motor is further connected to the steering mechanism shown in Figure 3.47, which generates the required steering angle shown in the Table 3.17.

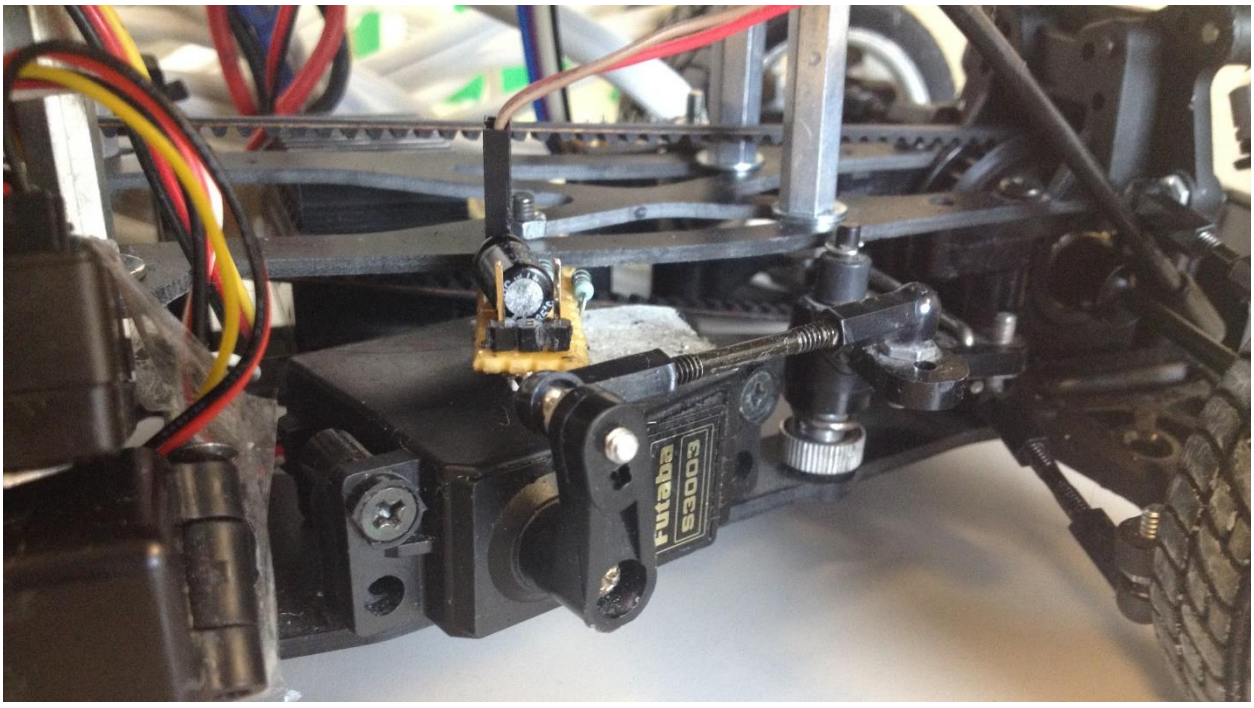

Figure 3.47: Steering Mechanism

Table 3.17: Pulse width and steering angle of RC model car.

Direction	Pulse Width	Steering Angle (degrees)
Right	1.2mS	-25
Neutral	1.6mS	0
Left	2.0mS	+25

Relation between the pulse width and the angle achieved is linear (verified by the oscilloscope). Also analog voltage generated by the passive RC filter and pulse width possess linear relationship. A relation between the applied pulse widths and the angle of the motor is derived using the line equation. System diagram to calculate steering angle of the vehicle is shown in Figure 3.48.

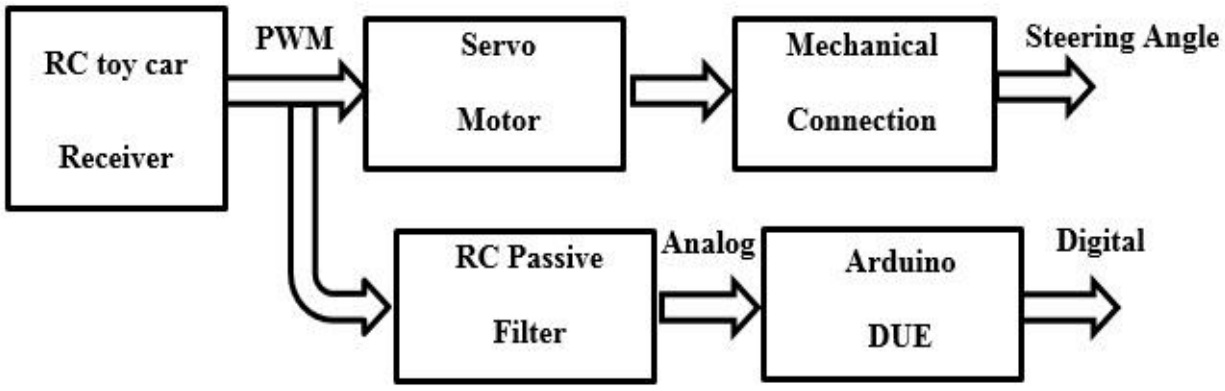

Figure 3.48: Block diagram of steering angle acquisition.

3.7.2. RC Filter Design

RC low pass filter (LPF) is used to convert pulses of waves into the analog voltage, which is then used as a reference to the steering angle of the vehicle.

3.8. DC MOTOR

High torque DC motor as shown in Figure 3.49 with a circular disk as a load provides adequate torque that is required by the system to overcome the external disturbances and maintain vehicle's stability. DC motor has a gear ratio of 48:1 and a stall torque of 1Nm. Some of the main features are given below.

- High starting torque
- Ball bearings
- Stainless steel output shaft
- Spur gears
- Reversible

Connecting DC motor with the motor driver board requires only two connections as shown in Figure 3.35 (section 3.5.5).

Figure 3.49: High torque DC motor.

Figure 3.50 shows the physical structure and measurement of the DC Motor.

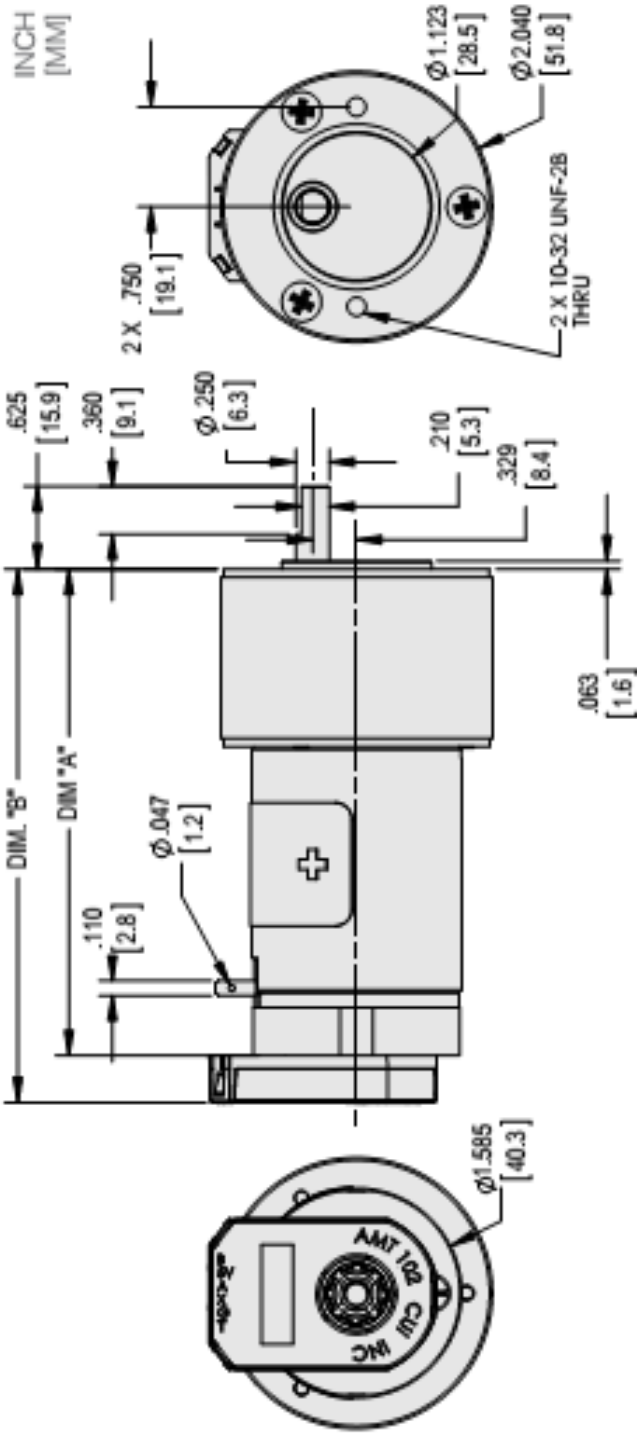

Figure 3.50: DC motor dimensions.

DC Motor assembled on the model car is shown in Figure 3.51. DC Motor has a digital encoder which is used to measure the velocity of momentum wheel. Encoder provides 100 PPR, because motor has gear ratio of 48:1, calculated pulses in one second is divided by the product of PPR and gear ratio to get the momentum disc velocity. Obtained count is then multiplied by 60 to convert it into RPM as shown in Equation 3.10 and Equation 3.11.

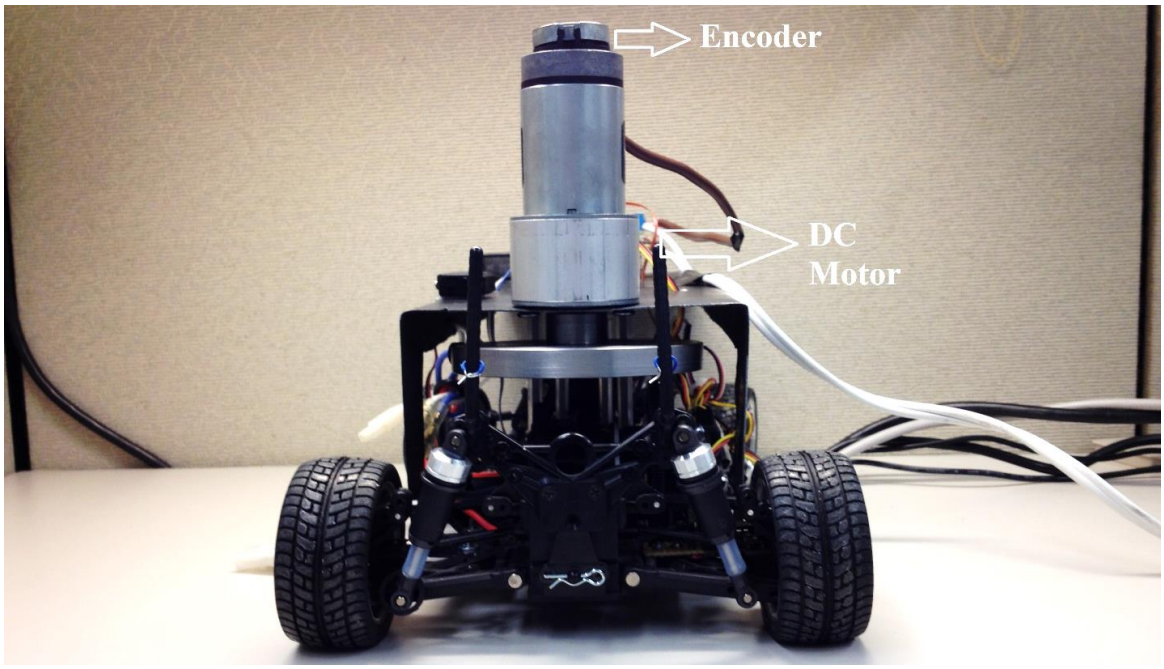

Figure 3.51: DC motor

$$Disc\ RPM = \left(\frac{Pulse\ count\ in\ one\ second}{PPR * gear\ ratio} * 60\right) \tag{3.11}$$

$$Angular\ velocity\ (r') = (Disc\ RPM * 0.104719755)\ \text{rad/s} \tag{3.12}$$

3.9. POWER SUPPLY

Firstly, two 12 V NI-MH (Nickel-Metal hydride) batteries were considered to be used to power the DC motor. Both connected in serial configuration generates a total voltage of 24 V. During the testing procedure it was noted that batteries weight (450 grams each) made RC model car slow and unstable during the turns, that was because the metal rods shown in Figure 2.19 were not able to support that much of weight and hence the whole holding structure was performing as a flexible joint. Therefore an external power supply shown in Figure 3.52 with maximum voltage of 30V and current 3A was considered to power the DC motor.

Figure 3.52: Power supply for DC motor

Chapter 4: **Software Assembly**

This chapter explains computer software used to program microcontroller (Arduino), configure wireless transceiver (X-CTU) and, receive and analyse data using online and post processing program respectively (LabView).

4.1. EMBEDDED PROGRAMMER

Arduino integrated development environment (IDE), based on Java is used to program the Arduino microcontroller in C++. Main window has the following sections shown in Figure 4.1. Toolbar has many common functions, text editor to write a code and, a message area for debugging the code. Steps to connect Arduino board with computer and to upload a program are shown below.

1. Download Arduino IDE
2. Connect the board
3. Install drivers
4. Launch Arduino IDE
5. Open any example
6. Select your Board
7. Select serial port
8. Upload the program

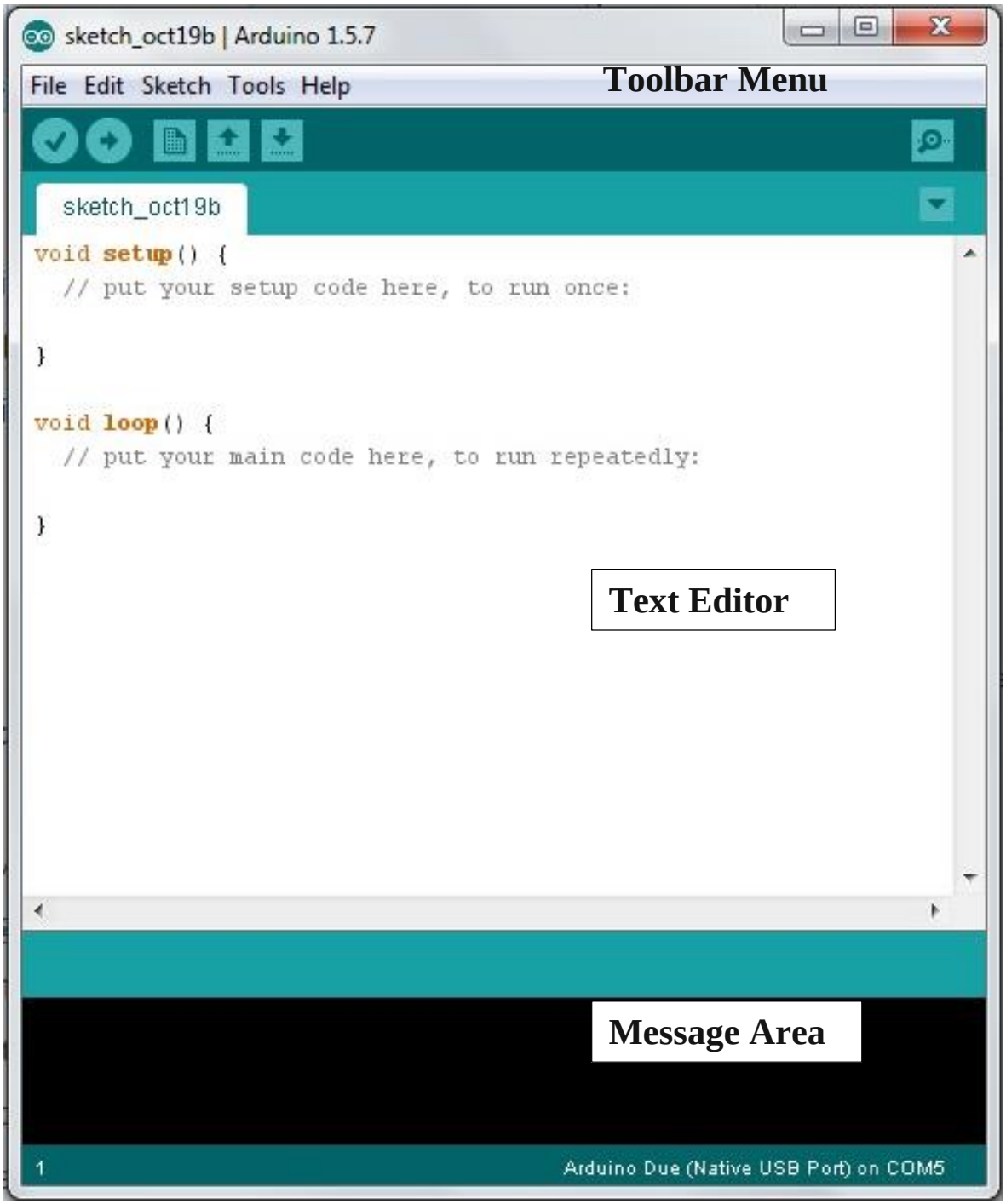

Figure 4.1: Arduino IDE main window.

4.1.1. Connecting to Arduino

After the installation of Arduino IDE on the computer, particular board can be selected by selecting the Tools➔ Board (Figure 4.2).

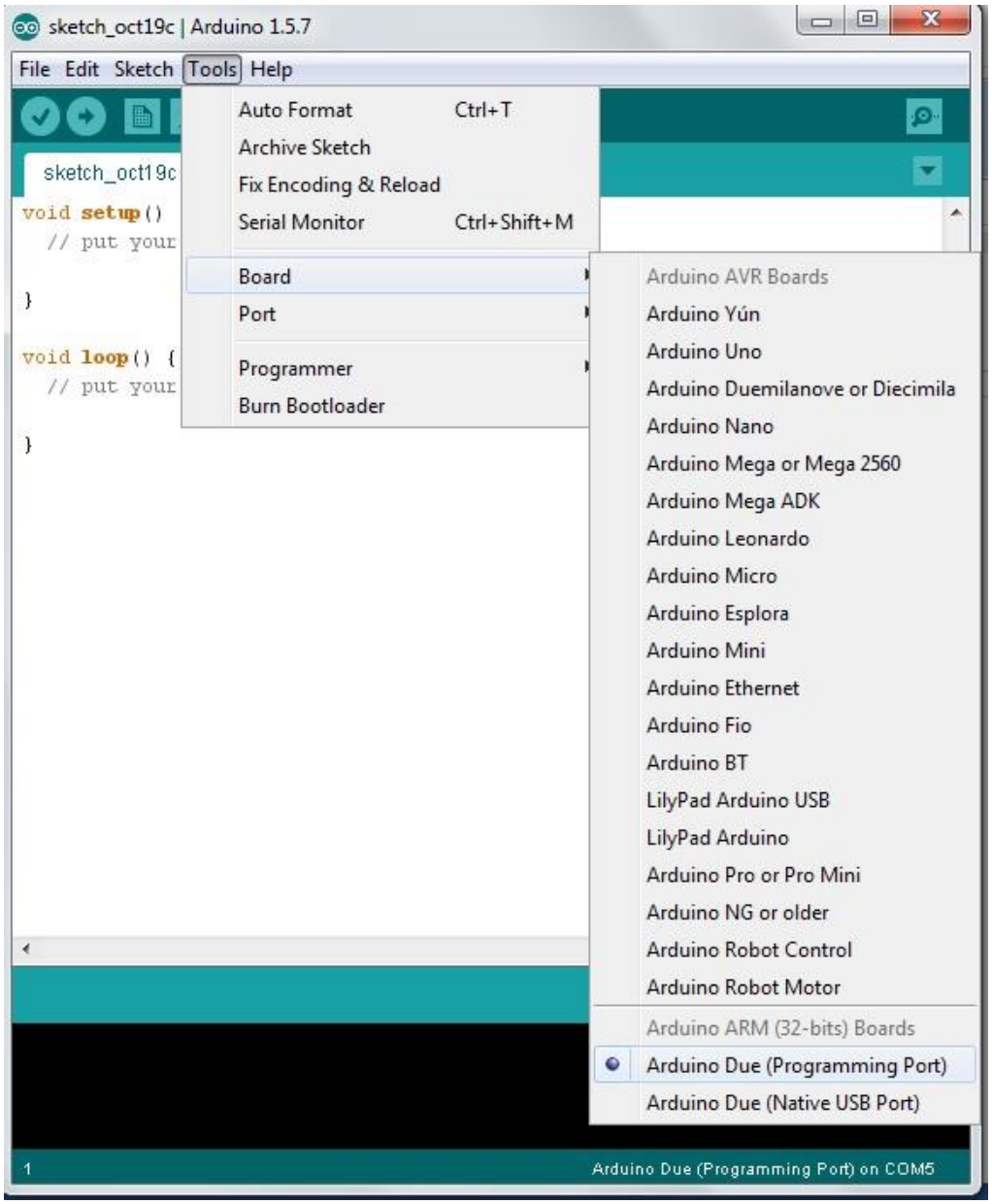

Figure 4.2: Selecting Arduino board.

When an Arduino board is connected to the computer, it appears under the Tools➔Port and can be selected from this menu option shown in Figure 4.3.

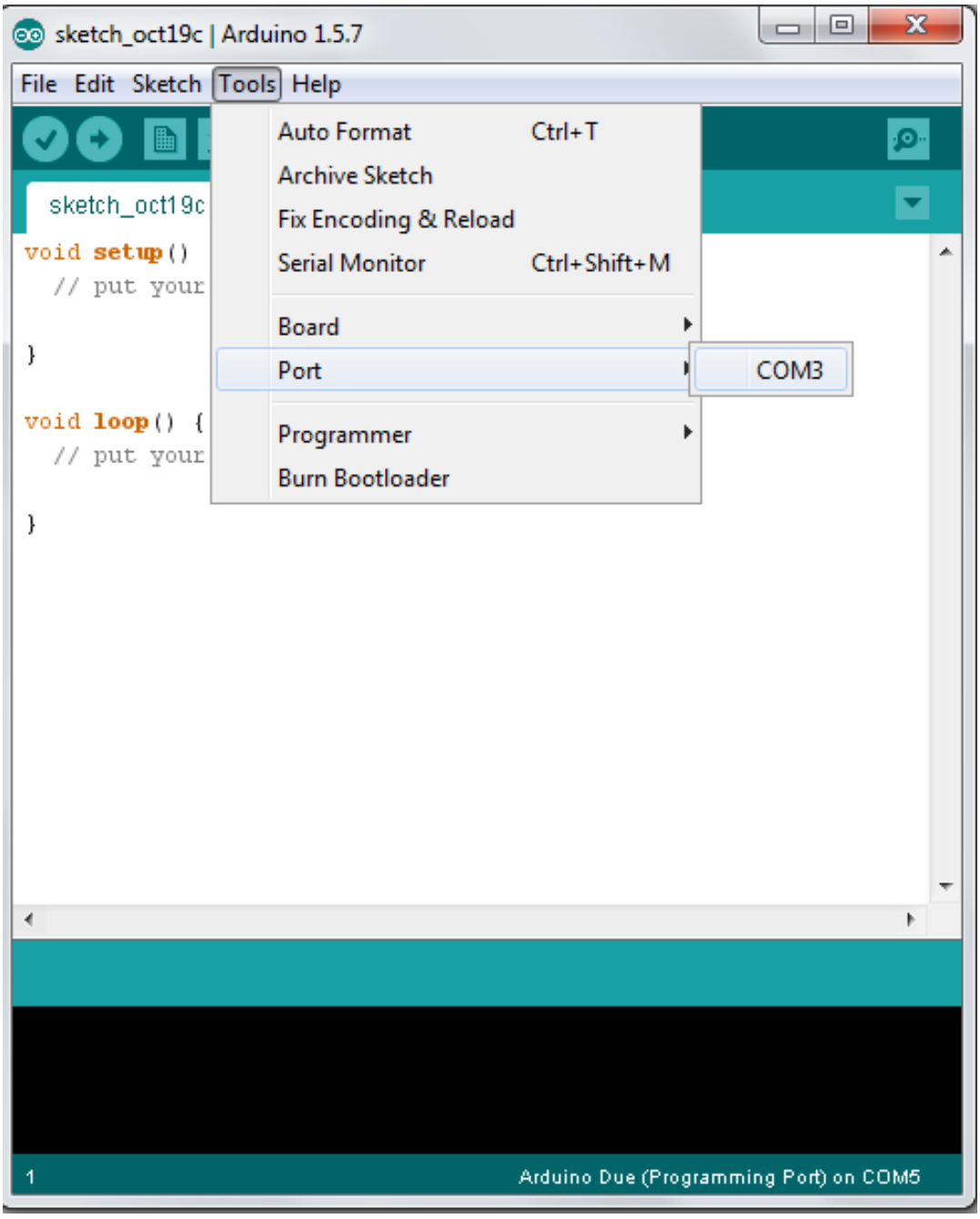

Figure 4.3: Arduino port selection

There is not any option other than the COM3 because at the time screenshot was taken, no Arduino board was connected to the computer.

Code written in the Arduino environment is called sketch and has an extension of '.ino'. Toolbar menu has a few options that allow different functionalities. Only the menu bar functions (from left to right) as shown in Figure 4.4 are explained.

Figure 4.4: Arduino IDE menu bar functions.

4.1.2. Menu Buttons

4.1.2.1. Verify

Checks the written code for any sort of errors that can stop it from compiling. All the errors are displayed in the message area.

4.1.2.2. Upload

Selecting this option will start compiling and write the code in to the microcontroller's memory. Arduino code goes through following steps by choosing this option.

- Check for any errors.
- Compilation
- Linkage

4.1.2.3. New

Creates and open a new sketch.

4.1.2.4. Open

This option allows to open and edit any previously existing sketch.

4.1.2.5. Save

Saves the sketch.

4.2. WIRELESS TRANCEIVER CONFIGURATOR

X-CTU is a software utility program designed to configure and interact with the firmware files found on ZigBee modules. It provides a simple to use graphical user interface (GUI). Software can be downloaded free of cost from the Digi's website. There are only four main tabs in the main window Figure 4.5. Only the functions used to test connectivity and configure the UART settings are explained in this section.

Main window has the following tabs:

- PC setting
- Range Test
- Terminal
- Modem configuration

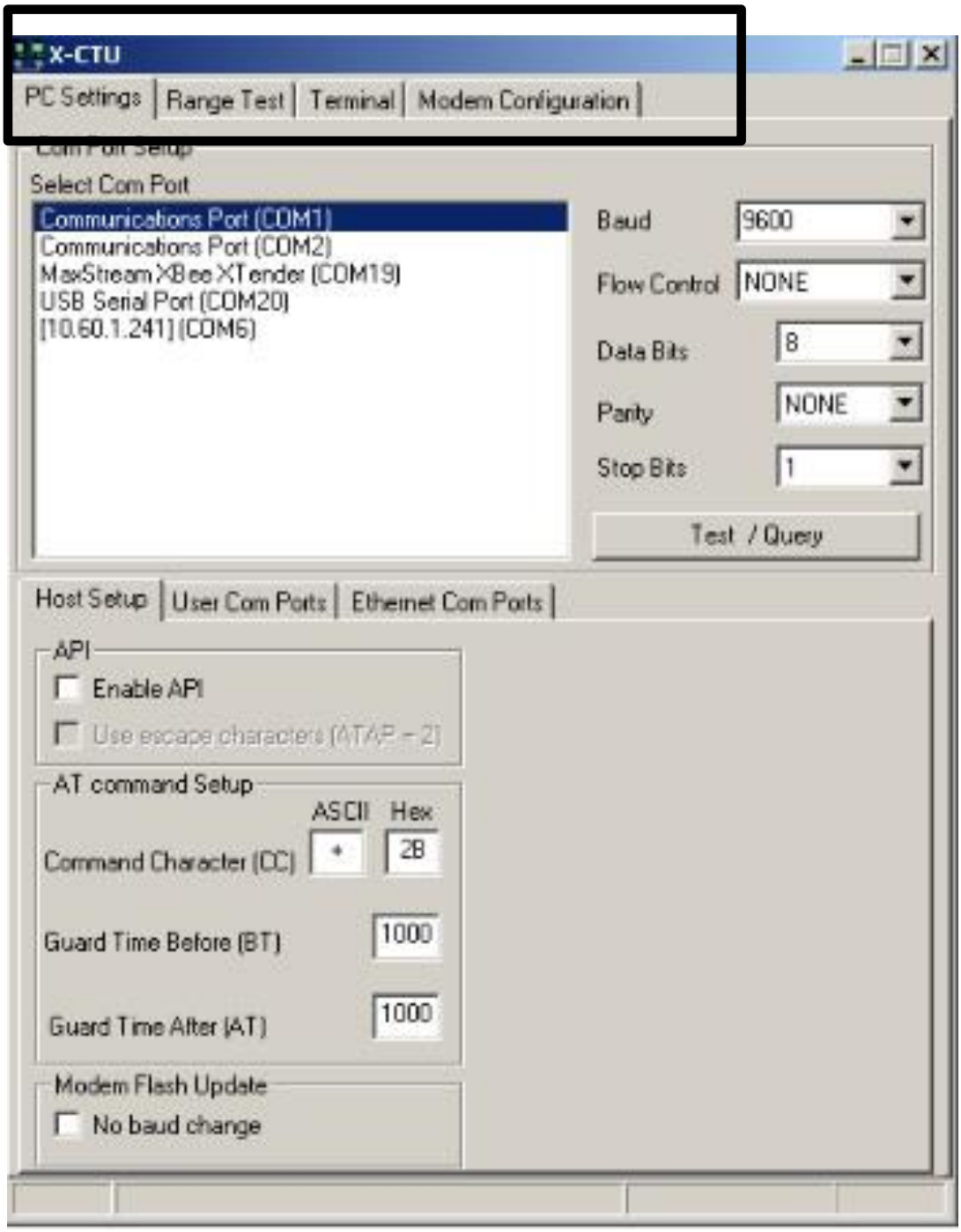

Figure 4.5: X-CTU main window.

4.2.1. PC Settings

PC settings tab is the default select when the X-CTU application is launched. This allows user to select the COM port and configure the settings (Table 4.1) by selecting the options from the drop down menu. Some of the settings include:

Table 4.1: X-CTU com port setup.

Setting	Description
Baud rate	Standard/Non standard
Flow control	Hardware, Software, None
Data bits	4,5,6,7, and 8 data bits
Parity	None, Odd, Even, Mark and Space
Stop bit	1, 1.5, and 2

After the settings has been selected, user can hit the *TEST/QUERY* button to test the connectivity between the transceiver and personal computer (PC). User will get a message if the settings and COM port are correct as shown in Figure 4.6.

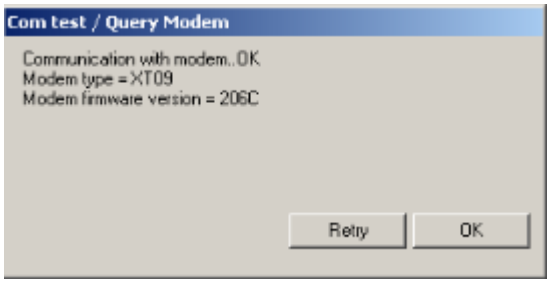

Figure 4.6: X-CTU com test.

4.2.2. Terminal Tab

Data sent and received by the transceiver module is displayed under this tab. Main functions of the terminal tab are:

- Terminal emulator
- Ability to send and receive predefined data packs
- Send and receive data in HEX and ASCII formats

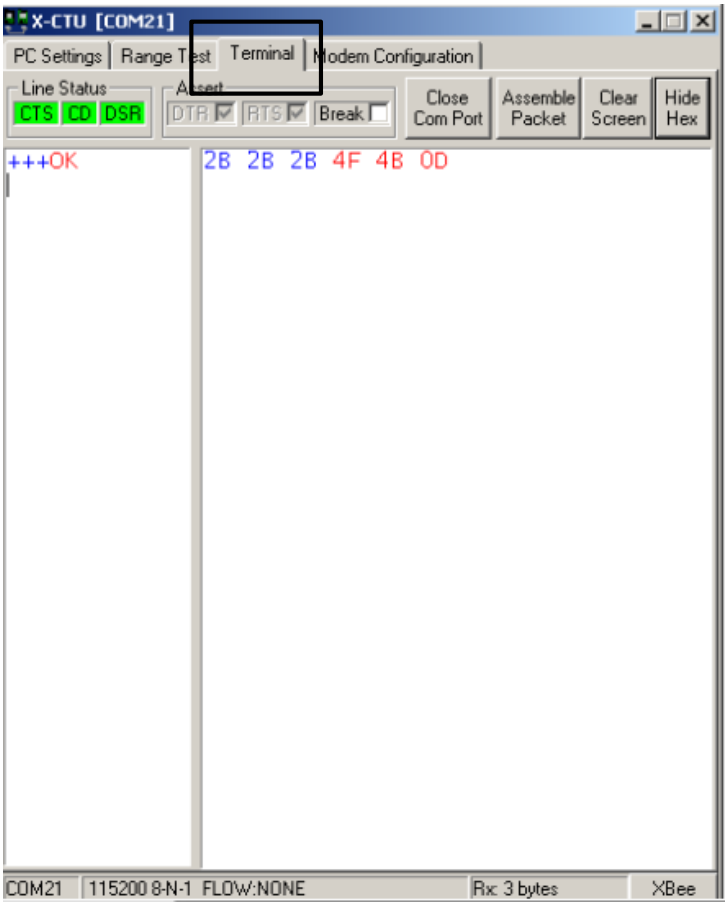

Figure 4.7: X-CTU terminal tab.

All the communication occur, displays under this tab. Blue text is what has been send out to the radio module while red text is the incoming data from the radio module as shown in Figure 4.7.

4.2.3. Modem configuration

Using this tab user can change the baud rate of the transceiver module. Baud rate used in the prototype development is 115200 bps. Modem configuration tab as shown in Figure 4.8 has three basic functions:

- Provides graphical user interface with a radios firmware
- Read and write firmware to radio's microcontroller
- Download an update firmware

4.2.3.1. Reading Firmware

1. Connect the radio module
2. Set the PC settings tab to the default
3. Select Modem configuration➔Modem parameters and Firmware➔Read

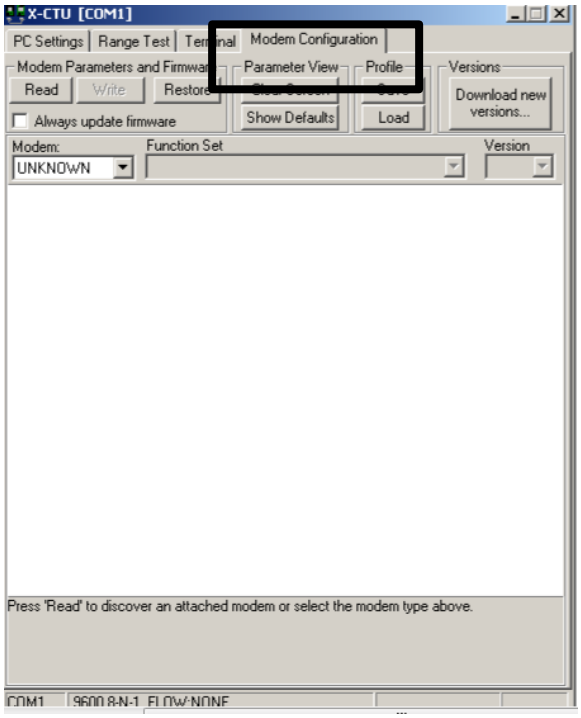

Figure 4.8: X-CTU Modem configuration.

4.2.3.2. Making Changes

After read function has been performed, user can change the settings as per the requirement. Settings are displayed I three colors:

- Black: read-only
- Green: Default
- Blue: User specified

To modify any of the user-selectable parameter, click on the command and enter the new value or select the new value from the drop down menu. Single click on the Write button will write the parameter changes to the modem's non-volatile memory.

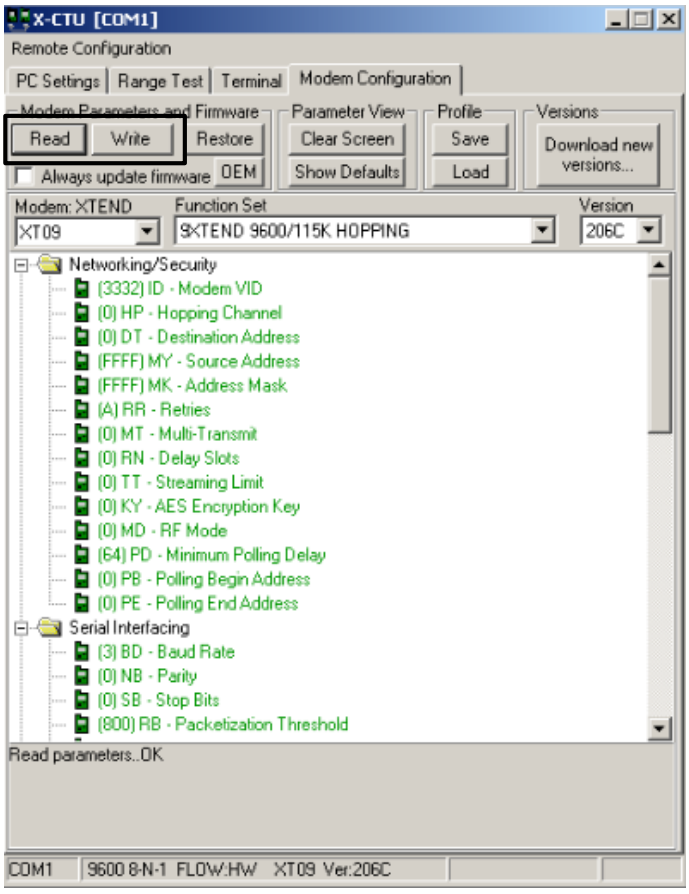

Figure 4.9: X-CTU Writing firmware.

4.3. ON-LINE AND POST PROCESSING

LabView (Laboratory Virtual Instrument Engineering workbench) is a graphical programming language developed by National Instruments. Simple to use graphical structure makes it ideal for automation, control, data acquisition and analysis. LabView saves the files with extension '.VI'. A VI consist of a front panel, a block diagram and, an icon that represents the program. Evaluation version of LabView can be downloaded from NI's website. Full development system provided by the university has been used to develop the data acquisition program.

4.3.1. Front Panel

Figure 4.10 illustrates the front panel of the LabView. Front panel is used to display all the controls and displays. It also handles the function input and output. Example front panel contains a knob, digital display, selection control and, a button.

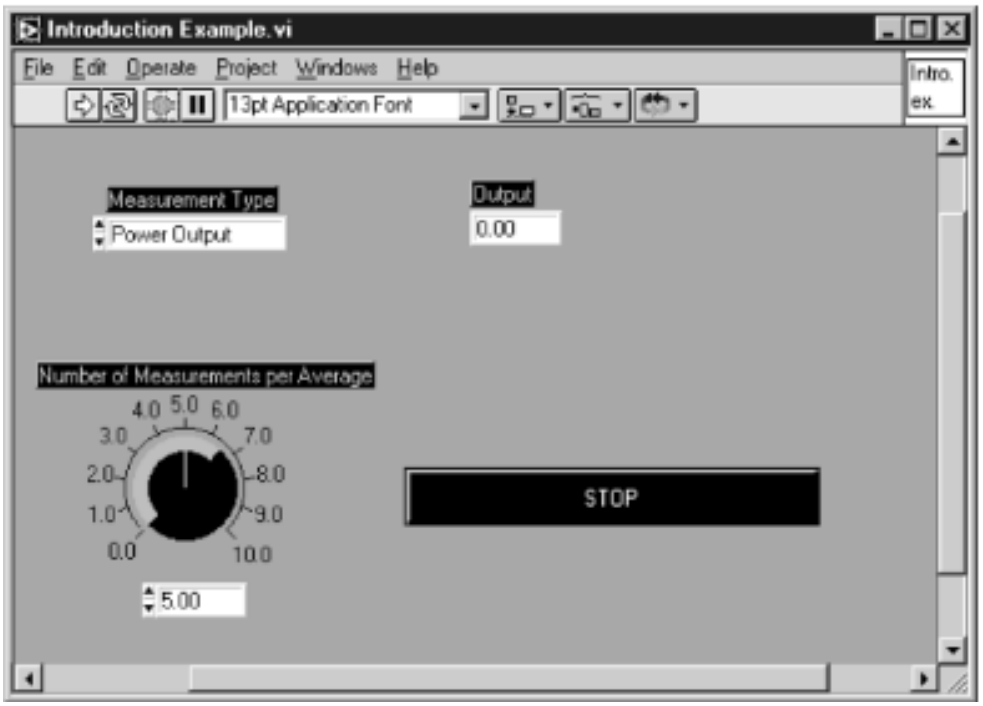

Figure 4.10: LabVIEW front panel.

4.3.2. Block Diagram

Block diagram contains the actual code for the VI. Figure 4.11 depicts the sample block diagram for the above shown block diagram. Outer rectangular border represents a while loop, and one inside the while loop is a case structure. Number of measurements and type of measurement is user-selectable. While loop stops execution when STOP button is pressed.

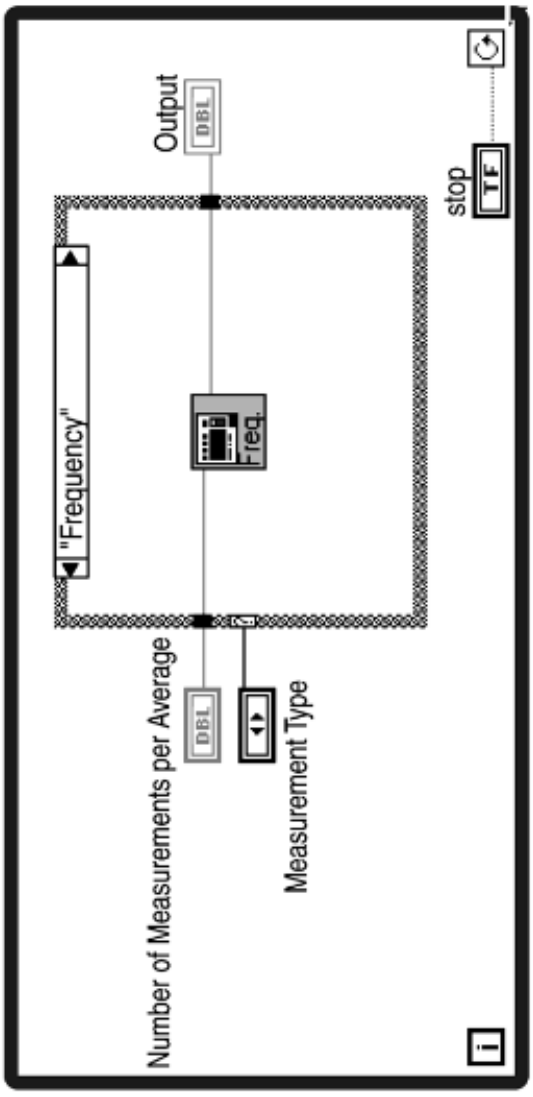

Figure 4.11: LabVIEW block diagram.

4.3.3. Developed LabView Program

Two programs (on-line processing and post processing) have been developed in the process of prototype development.

4.3.3.1. On-line Processing

This program obtains data transmitted by the microcontroller every 25 mSec, data received is then sorted in to the respective arrays as per the predefined string format and displayed on the GUI. An option to write the received data into measurement file has been included in the LabView program to analyse the data while offline. Data received from the microcontroller is:

- Longitudinal acceleration (a_x)
- Longitudinal velocity ($\hat{v}_x$)
- Lateral acceleration (a_y)
- Lateral velocity ($\hat{v}_y$)
- Yaw rate (r)
- Steering angle (δ)
- Slip ratio (β)
- Momentum disc acceleration (r'')
- Momentum disc velocity (r')
- Required torque form the motor (T)

Whereas, current torque of the motor is calculated within the LabView.

Front panel of the developed program is shown in Figure 4.12.

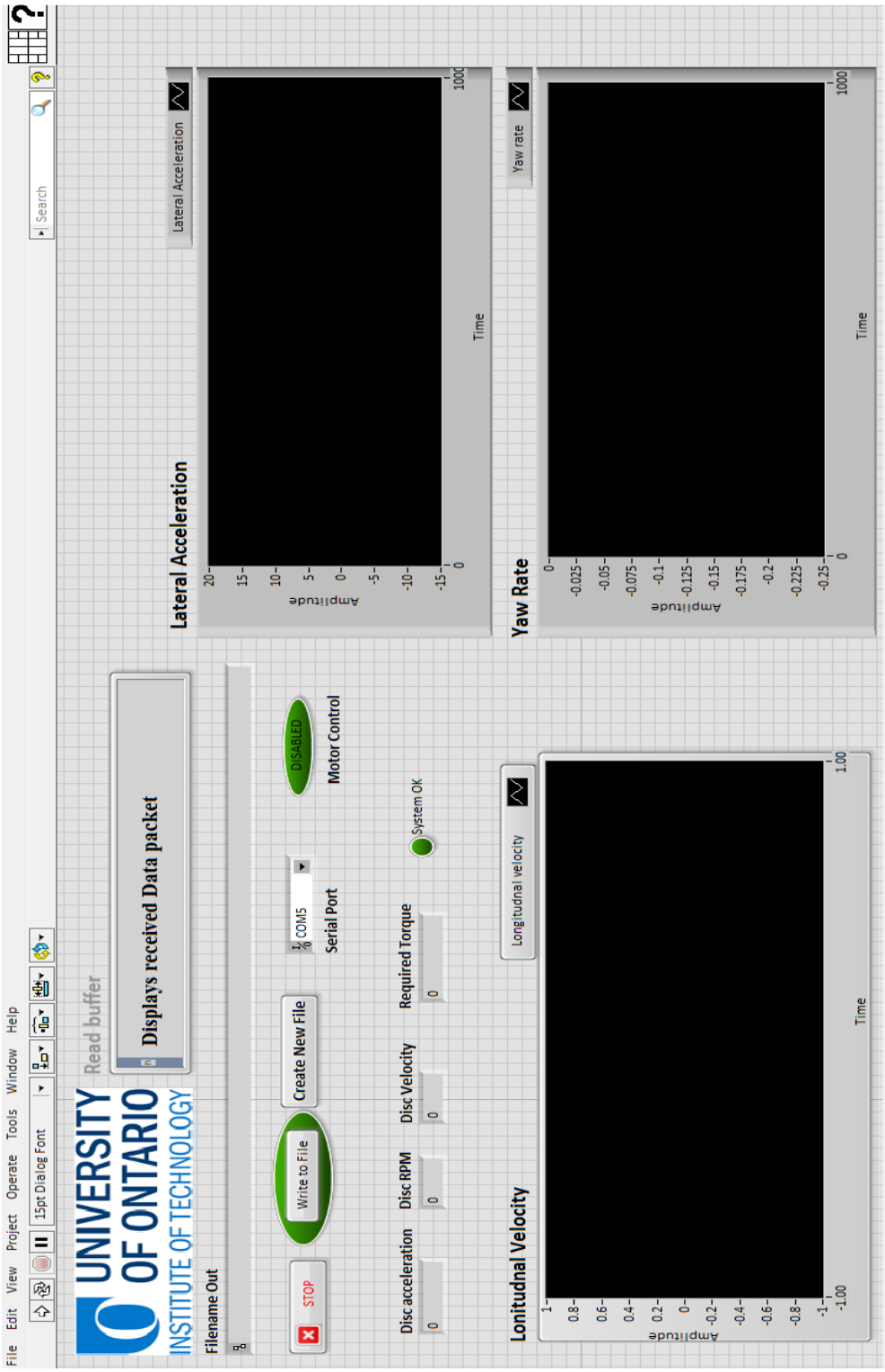

Figure 4.12: Front panel of On-line processing program

4.3.3.2. Post Processing

Post processing program is used to analyse and compare the vehicle dynamics performance with and without the momentum wheel. Front panel of the program is shown in the Figure 4.13.

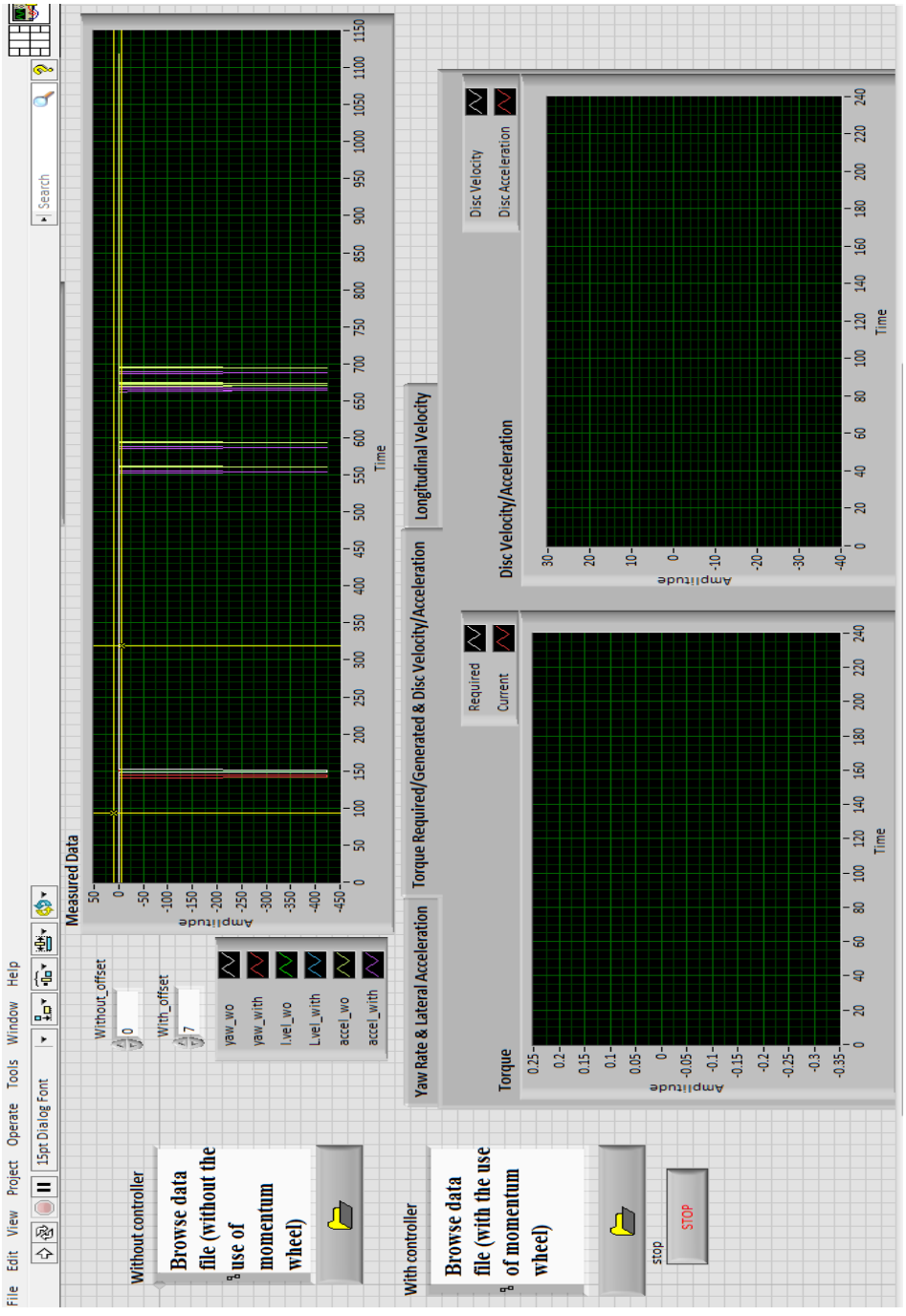

Figure 4.13: Front panel

Chapter 5: **Velocity Estimator and Torque Controller**

Kalman filter is named after Rudolf E. Kálmán and is more than 50 years old. It is very popular because of its recursive properties and minimum computational requirements [60]. Kalman filter can be use d as filter to remove noise from the measurements or to estimate the hidden system states. Kalman filter fundamentally consists of a set of mathematical equations (linear-quadratic) that implement a predictor-corrector type estimator that is optimal in a way that it minimizes the estimated error covariance, when some presumed assumptions are met. Kalman filter applies recursive operation on the measurements to provide an estimate of the current system state [61]. Kalman filters are often implemented in embedded control systems because in order to control a process, you first need an accurate estimate of the process variables.

5.1.1. Why Kalman Filter

Inertial Measurement Units (IMU) are very sensitive to the drifts and bias. They also suffers from the external environmental noises, which are added into the measurements and reduces system performance. Kalman filter has the ability to model the effect of these unwanted noise signal on the measurements and system states. There are three basic reasons why deterministic system and control theories are not accurate means for system design in some cases:

1. No mathematical system model is perfect: There are many sources of uncertainty in any mathematical model of a system that is not determined absolutely.

2. Not all the quantities can be measured directly using the sensors. Also, they always introduce unwanted noise and biases to the overall system.
3. The deterministic dynamic models are driven not only by our control inputs but also by disturbances which cannot be modelled deterministically.

As can be seen from the proceeding discussion to assume the perfect model for the complex control system models is not feasible. So, we need to find a model that gives an accurate estimate of the true state even though we cannot directly measure it and it should satisfy the following conditions:

1. The average value of our estimate state to be equal to the average value of the true state i.e. expected value of the estimate should be equal to the expected value of the state.
2. The state estimate that varies from the true state as little as possible, i.e. we want to find the estimate with the smallest possible error variance.

5.1.1.1. Assumptions

There are some assumptions about the noise and the system state that as to be made before designing the Kalman filter.

- The model you use to predict the state needs to be a linear function of the measurement.
- The process error w_k and the measurement error v_k (noise) are uncorrelated, zero-mean white-noise processes with known covariance matrices as shown in Equation 5.1-5.3.
- The initial state, x_0 is a random vector uncorrelated to system and measurement noise processes.
- Initial system state has known mean and covariance matrix.

Reason to use the Gaussian distribution for the process and measurement noise is because most of the random process that occur in nature are close to the Gaussian distribution.

$$E[w_k \ w_k^T] = \begin{cases} Q_k & k = l \\ 0 & otherwise, \end{cases} \tag{5.1}$$

$$E[v_k \ v_k^T] = \begin{cases} R_k & k = l \\ 0 & otherwise, \end{cases} \tag{5.2}$$

$$E[w_k \ w_k^T] = 0 \ for \ all \ k = l \tag{5.3}$$

where, $Q_k \ and \ R_k$ are symmetric positive semi-definite matrices.

5.1.1.2. Gaussians

Type of noise considered during the design process of the kalman filter is gaussian white noise. The term "white" states that noise has equal power per unit bandwidth, which results in a flat power spectrum across the frequency of interest. The term "gaussian" refer to the probability density function of the amplitude of noise signal. Area under a gaussian always adds up to one. Gaussian distribution is shown in Figure 5.1.

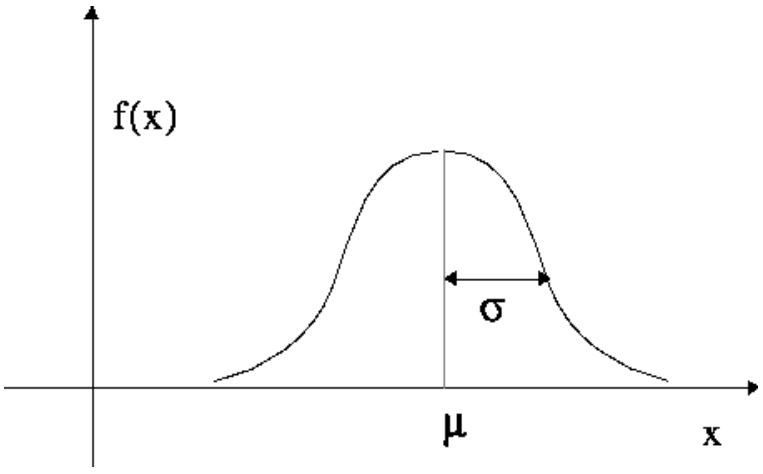

Figure 5.1: Gaussian distribution

where, μ is the mean of the signal and, σ width of the gaussian called the variance (often written as quadratic variable, σ^2). Any gaussian in one-dimensional space is characterized by (μ, σ^2). Formula for the gaussian function is given in Equation 5.4.

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{1}{2}\frac{(X - \mu)^2}{\sigma^2}] \tag{5.4}$$

Kalman filter's task is to maintain the μ and σ^2 that produces the best estimate of the states considered in the design. To do this kalman filter iterates through two steps: measurement update and time update which are explained later in the section.

5.1.2. Applications

Kalman filter has applications in range of areas. The most famous one is being the guidance for the trajectory for the Apollo-11. Also, it is used for the estimation in the autonomous driving vehicle. Some of the applications of the Kalman Filter are:

- Tracking objects (e.g., missiles, faces, heads, hands)
- Fitting Bezier patches to (noisy, moving, ...) point data
- Economics
- Navigation and Traffic management
- Digital signal processing
- Stabilizing depth measurements
- Feature tracking and cluster tracking
- Fusing data from radar, laser scanner and stereo-cameras for depth and velocity measurements

- Battery state of charge (SoC) estimation
- Motor control
- Orbital estimation
- Estimation

Before going in to the design process of Kalman filter description of parameters involved in the Kalman filter calculations are provided in Table 5.1.

Table 5.1: Kalman filter parameters

Parameter	Parameter Identification	Description
x_k	System state vector	Contains terms of interest
u_k	Control input vector	-
F_k	Transition matrix	Relates state parameters at *k-1* with state parameters at time *k.*
B_k	Control input matrix	Applies any effect of control input on the state vector
w_k	Process noise vector	Zero-mean distribution with covariance given by the covariance matrix Q_k
z_k	Measurement vector	Noisy measurements

Table 5.1: continued

H_k	Transformation matrix	Maps state vector parameters into the measurement domain
v_k	Measurement noise vector	Contains measurement noise term for each observation in the measurement vector. It is assumed to be zero-mean Gaussian white-noise with covariance R_k
K_k	Kalman gain matrix	-
P_k	Covariance matrix	-
$\tilde{y}_k$	Innovation at- time *k*	Error between measurement and state estimate
S_k	Innovation covariance matrix	Contains innovation variables
Q	Process noise covariance matrix	Contains process noise variable
R	Measurement noise covariance matrix	Contains measurement noise variable

5.1.3. Filter theory

Main application of the Kalman filter is to smooth the noisy measurement and produce estimate of the hidden information in the measurements. Kalman filter can be applied to any model in which state space of the latent variables are continuous and observed variables have a gaussian

distribution. The Kalman filter is a recursive estimator. This means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. Kalman filter undertakes the system states at time *k* evolved from the prior state at time *k-1*, shown in Equation 5.4. Measurement of the system can be performed using the Equation 5.5.

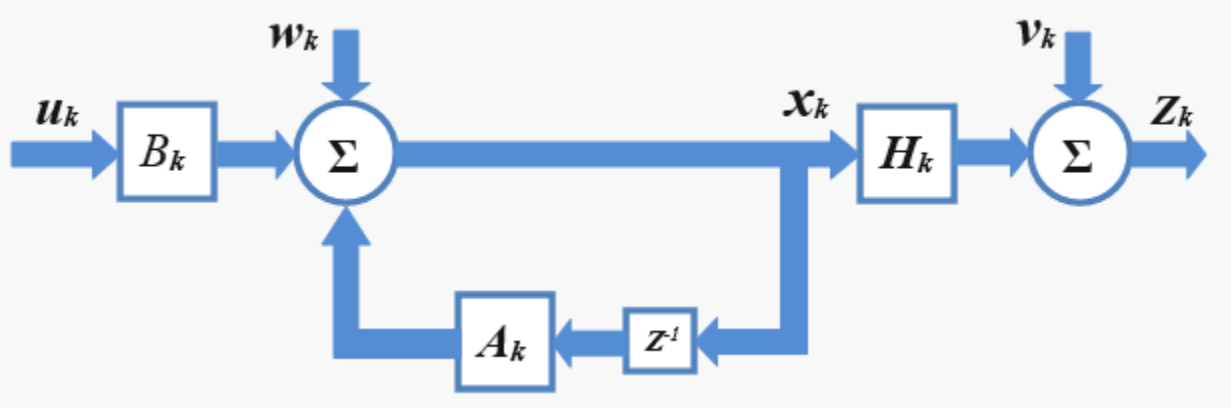

Figure 5.2: Kalman filter model

$$x_k = A_k x_{k-1} + B_k u_k + w_k \tag{5.5}$$

It means that each x (signal value) may be evaluated by using a linear stochastic equations. Any x_k is a linear combination of its previous value plus a control signal u_k and a process noise w_k. Remember that, most of the time, there's no control signal u_k.

$$z_k = H_k x_k + v_k \tag{5.6}$$

Kalman Filtering Algorithm tries to converge into correct estimations, even if the Gaussian noise parameters are poorly estimated.

5.1.3.2. Matrices dimensions

Dimensions of matrices involved in the designing of the Kalman filter are dependent on the system states and the number of measurement sensors [62], and can be calculated as below:

- The transition matrix A_k relates the system state at the previous time step *k-1* to the state at the current step k. *F* has the dimensions of *n x n* where, *n* is the number of variables we are keeping track of. For example it could be position, velocity and, acceleration.
- Measurement matrix Z_k is just the output of the sensor. If a measurement system has *m* sensors then size of the matrix will be *m*. Number of sensors considered can be different from the *n.*
- Output transformation matrix H_k is of size *m x n.* Because when we multiply H_k with X_k as per Equation 5.5, it gives us matrix Z_k which is of size *m*.
- P_k, is the covariance matrix of vector X_k, therefore dimension of P_k is *n x n*.
- R_k, is a covariance matrix of variances and covariance of the sensor measurement. Dimension of R_k depends on matrix Z_k therefore, its size is *m x m*.
- The *n x l* matrix *B* relates the optional control input to the state *x*.

5.1.4. Kalman filter Algorithm

The system can be solved by following the simple set of equations provided in Table 5.2. When performing the actual calculations for the filter, the state estimate and co-variances are coded into matrices to handle the multiple dimensions involved in a single set of calculations. This allows for representation of linear relationships between different state variables (such as position, velocity, and acceleration) in any of the transition models or co-variances.

Structure of Kalman involves only one input (measurement,Z_k) and returns one output (estimate,$\hat{X}_k$). The Kalman filter estimates a process by using a feedback control: the filter estimates the state at some time and then calculates feedback measurements. As such, the equations for the Kalman filter fall into two groups [63] as shown in Figure 5.3 and Table 5.2.

1. Time update equations.
2. Measurement update equations.

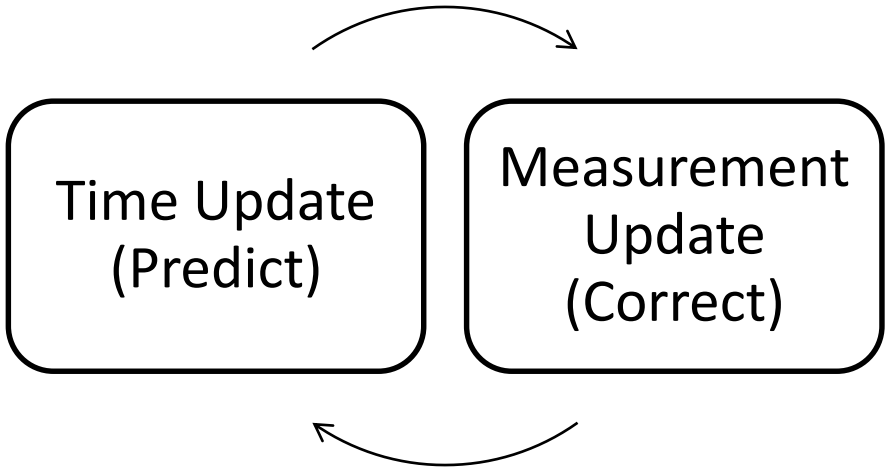

Figure 5.3: Kalman filter cycle

Table 5.2: Kalman filter equations.

Time Update	Predicted state estimation	$x_{predicted} = Ax_{k-1} + Bu_{k-1}$	(5.7)
	Predicted estimate covariance	$P_{predicted} = AP_{k-1}A^T + Q$	(5.8)
Measurement Update	Innovation	$\tilde{y} = z_k - Hx_{predicted}$	(5.9)
	Innovation covariance	$S = HP_{predicted}H^T + R$	(5.10)
	Optimal gain	$K = P_{predicted}H^T s^{-1}$	(5.11)
	State update	$x_k = x_{predicted} + K\tilde{y}$	(5.12)
	Covariance update	$P_k = (I - KH)P_{predicted}$	(5.13)

5.1.4.2. Time Update

The time update equations are accountable for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step.

5.1.4.3. Measurement Update

The measurement update equations are responsible for the feedback i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

5.1.5. Types of Kalman Filter

We have discussed about the Kalman filter for the linear system. But, most of the problems may not be defined by the linear Kalman filter. Thus, we have non-linear type Kalman filter. Figure 5.4 depicts the types of Kalman filter.

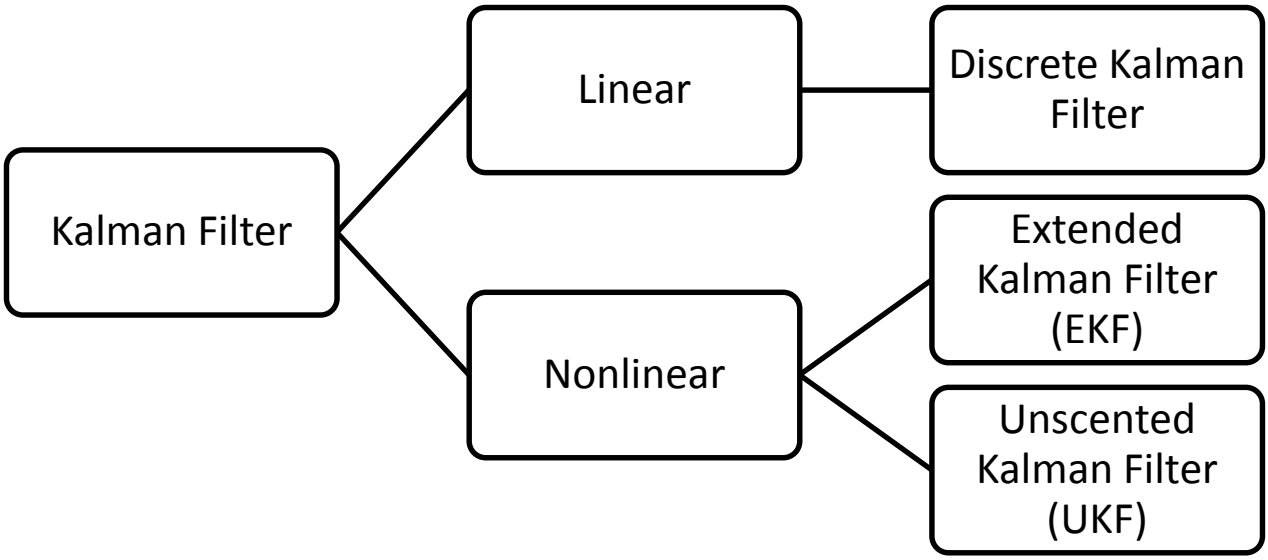

Figure 5.4: Kalman filter types.

5.1.5.1. Extended Kalman Filter (EKF)

Most real life problem are not linear and therefore needs to be linearized before Kalman filter can be used for the estimation. Extended Kalman Filter (EKF) is one approach to solving this problem.

To apply the Kalman filter it is necessary to describe the real world by a set of non-linear differential equations, as described below:

$$x = f(x) + w \tag{5.14}$$

$$z = h(x) + v \tag{5.15}$$

Where, *x* is a vector of the system state, *f(x)* is a nonlinear function of those states, *w* is a random zero-mean process which is given by $Q = E(ww^2)$ and, *v* is the zero-mean random process described by the measurement noise *R* as $R = E(vv^2)$.

The extended Kalman filter solves this problem by finding the Jacobian of *f* and *h* around the estimation state and gives the final model function centered on the state. The extended Kalman filter works almost like a regular Kalman filter, except for *F* and *H*. which are related to the nonlinear system and measurement equations according to the following equations.

$$\boldsymbol{F} = \frac{\boldsymbol{\partial f(x)}}{\boldsymbol{\partial x}} \tag{5.16}$$

$$H = \frac{\partial h(x)}{\partial x} \tag{5.17}$$

The relationship is as shown in Figure 5.5.

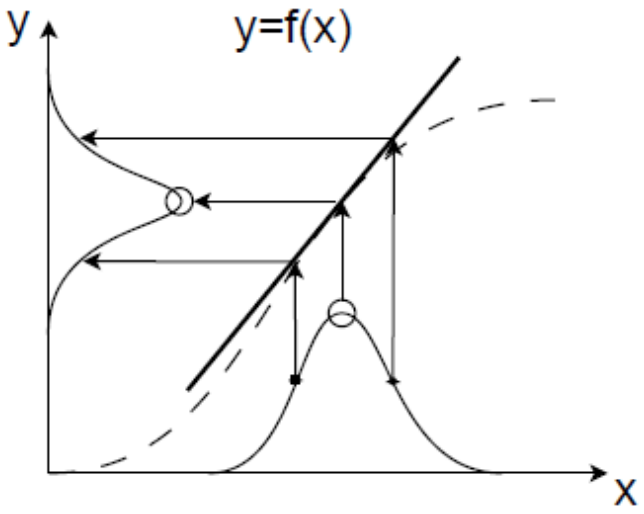

Figure 5.5: Illustration how EKF linearize a nonlinear function around Gaussian distribution.

Calculations required to perform EKF are provided in Table 5.3.

Table 5.3: EKF calculations.

Time Update Equations	Measurement Update Equations
$\hat{x}_{k\|k-1} = f(\hat{x}_{k-1\|k-1}, u_k, 0)$ $P_{k\|k-1} = F_k P_{k-1\|k-1} {F_k}^T + Q_k$	$K_k = P_{k\|k-1} {H_k}^T (H_k P_{k-1\|k-1} {H_k}^T + R_k)^{-1}$ $\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k \left(z_k - h\left(\hat{x}_{k\|k-1}, 0\right)\right)$ $P_{k\|k-1} = (I - K_k H_k) P_{k\|k-1}$

5.1.5.2. Unscented Kalman Filter (UKF)

For a highly non-linear functions f and h the extended Kalman filter can give particularly poor performance. The problem for estimating Gaussian random variables can also be approached by the algorithm called unscented transform. The idea is to produce several sampling points (Sigma-points) around the current state estimate based on its covariance and then, propagating these points through the nonlinear map to get more accurate estimation of the mean and covariance of the mapping results. In this way, it avoids the need to calculate the Jacobian, hence incurs only the similar computation load as the EKF.

The unscented transform (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transform. This method is significantly different from the other sampling methods, which require orders of magnitude more sample points to propagate accurate distribution state.

The unscented transform also has another advantage that noise can be treated as non-Gaussian fashion. The strategy for doing so involves propagation of noise through the functions by first augmenting the state vector to also include noise sources.

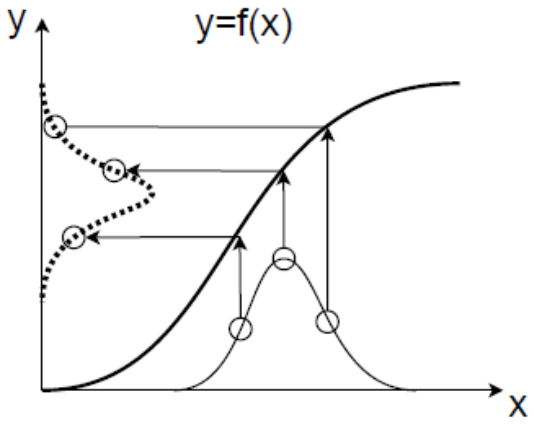

Figure 5.6: Sampling of sigma-points through a nonlinear function using unscented transform.

5.1.6. Implementation and Design

In this section design and implementation method of the Kalman filter used has been described. Momentum wheel controller requires steering angle, momentum disc velocity, yaw rate and, lateral velocity to calculate the required torque signal. All three except the last parameter can be easily calculated. To calculate the lateral and longitudinal velocity of the vehicle, pseudo integral approach together with the Kalman filter has been used. Control system structure is shown in the Figure 5.7.

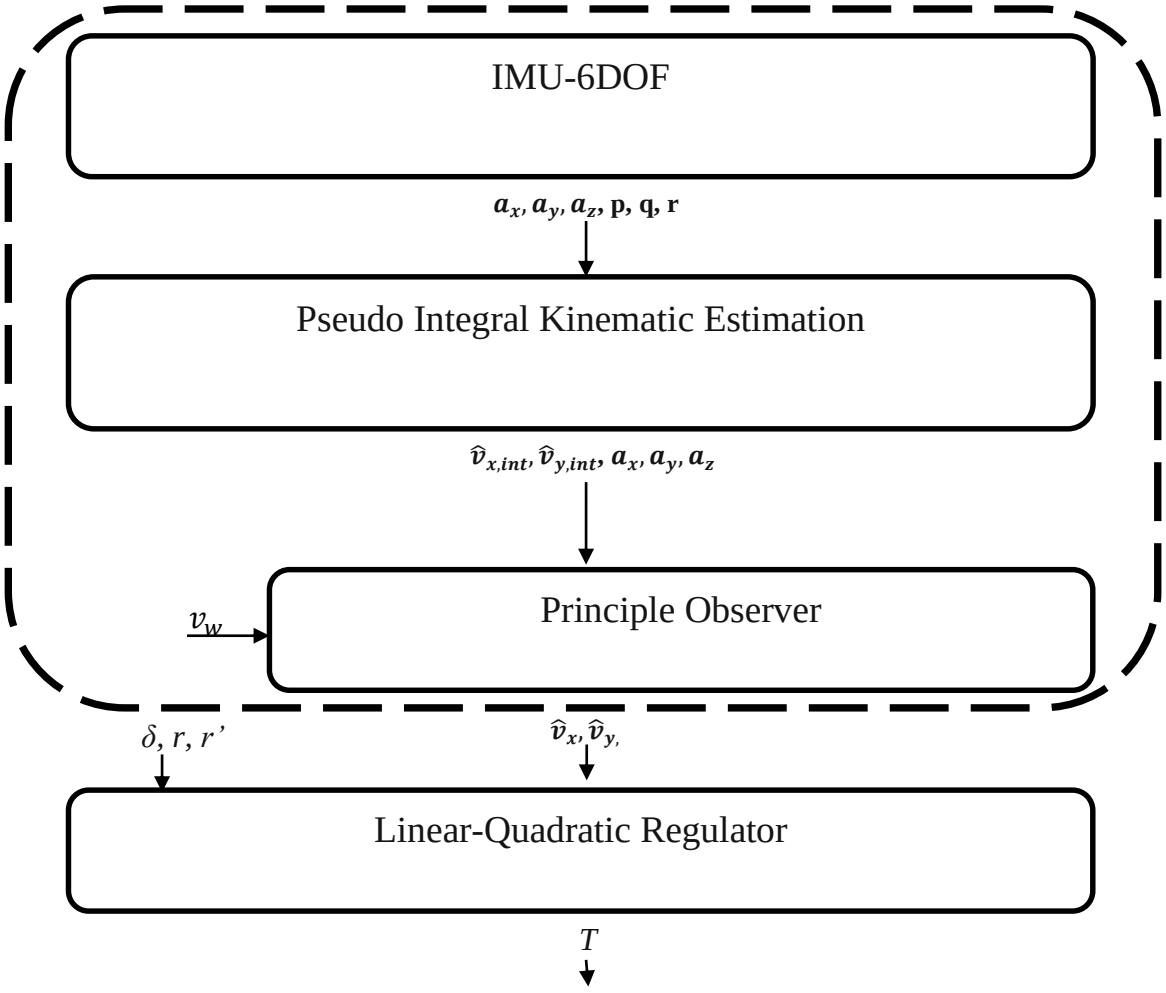

Figure 5.7: Control system structure

5.1.6.1. Pseudo Integral Kinematic Estimation

Lateral velocity of the vehicle is calculated in terms of longitudinal and lateral velocity and yaw rate as shown in the Equation 5.17 below.

$$a_y = \dot{v}_y + r v_x \tag{5.18}$$

Using the above equation to calculate lateral velocity is not an accurate approach because sensor measurements are affected by the vehicle orientation and road angle. By taking these influences in to account acceleration can be written as below,

$$a_y = a_{y,m} + (-g \sin\phi \cos\theta) \tag{5.19}$$

where, $a_{y,m}$ is the acceleration measured by the IMU module, ϕ is the roll angle and, θ is the pitch angle. Both roll and pitch angles are obtained directly from the DMP unit of MPU-6050 module. IMU modules are highly sensitive to the sensor errors, drift and this error gets incorporated in the calculations while taking the integration. To remove this problem pseudo integration is used, which reduces the sensor drift by suppressing the integration process when not required. Integration process is forcefully stopped if the vehicle is at complete rest. Vehicle's rest state is determined by using the measurement data from the IMU module as shown in the following Equation 5.19 below.

$$IMU_{mag} = \left| \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2} - g \right| \tag{5.20}$$

Where, IMU_{mag} represents motion of the vehicle. If vehicle is at rest value of IMU_{mag} is equal to 'g'. This value is then compared with the threshold value IMU_{thr} (determined experimentally) and appropriate action is taken as shown in the Equation 5.20 and, final value is obtained by multiplying the *gain* with the estimated velocity.

$$gain = \begin{cases} 1, & IMU_{mag} < IMU_{thr} \\ 0, & otherwise \end{cases} \tag{5.21}$$

Kinematic estimator has the following vehicle dynamics model

$$\dot{v}_x = a_x + r.v_y - q.v_y + g.\sin\theta \tag{5.22}$$

$$\dot{v}_y = a_y - r.v_x + p.v_z - g.\sin\phi \ .\cos\theta \tag{5.23}$$

$$\dot{v}_z = a_z + q.v_x - p.v_y - g.\cos\phi \ .\cos\theta \tag{5.24}$$

Above equations can be written in the form of state space equations of the vehicle kinematics as shown below in Equation 5.24.

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} . \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + g \begin{bmatrix} \sin\theta \\ \sin\phi \ . \cos\theta \\ \cos\phi \ . \cos\theta \end{bmatrix} \quad (5.25)$$

First row in the above model that calculates the time derivative of the longitudinal velocity (measured directly using the wheel s sensor) is not considered in the kinematic model calculations. Taking later two equations following kinematic expressions can be written as shown in the Equation 5.25.

$$\begin{cases} v_{y,int} = \ a_y - \ r.v_y + p.\hat{v}_{z,int} - g.\sin\phi \ . \cos\theta \\ v_{z,int} = \ a_z + \ q.v_y - p.\hat{v}_{y,int} - g.\cos\phi \ . \cos\theta \\ \hat{v}_{y,int} = gain \, . \, v_{y,int} \\ \hat{v}_{z,int} = gain \, . \, v_{z,int} \end{cases} \quad (5.26)$$

$\hat{v}_{y,int}$ and $\hat{v}_{z,int}$ are the lateral and vertical velocities estimated by the pseudo integration method v_y is the longitudinal speed measured by the wheel hall sensor. Although integration of $\hat{v}_{y,int}$ and $\hat{v}_{z,int}$ can lead to the drifts is the estimated velocities but, estimate quickly returns and stabilize to zero to correct the drift.

5.1.6.2. Principle Observer

Principle observer is a Kalman filter that calculates vehicle velocity estimates, along with the reference lateral velocity $\hat{v}_{x,int}$ obtained from the pseudo integral estimator. Purpose of the principle observer is to further improve the longitudinal velocity estimate $\hat{v}_x$ by not only depending on the wheel sensor data but by considering IMU data as well. Similarly, accuracy of $\hat{v}_y$ is improved by eliminating drift and noise.

Using Equation 5.4 and Equation 5.5 system state model and measurement model can be described as,

$$x_k = \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{v}_z \end{bmatrix} \tag{5.27}$$

$$F_k = \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \tag{5.28}$$

$$B_k = \begin{bmatrix} a_x + g.\sin\theta \\ a_y - \text{g}.\sin\phi\ .\cos\theta \\ a_z - \text{g}.\cos\phi\ .\cos\theta \end{bmatrix} \tag{5.29}$$

$$H_k = \begin{bmatrix} v_w \\ \hat{v}_{y,int} \\ 0 \end{bmatrix} \tag{5.30}$$

R_k is designed by referring to the datasheet [53] provided by the manufacturer and is calculated as shown below in Equation 5.31[64]. P_k, and Q_k were obtained by performing experimental procedures. Kalman filter was implemented in the scalar form using the microcontroller.

$$Total\ Noise = \left(Noise\ density * \sqrt{bandwidth}\right) \tag{5.31}$$

5.1.7. Torque Controller

Torque controller is designed using the LQR theory as explained in Section 2.2 (because design method for the controller was already provided in [1], which decreased the development time). Microcontroller is programed to calculate the system gains, as shown in Equation 2.1, every time it receives data from the IMU unit. Apart from v_y, r, r' and, δ LQR takes some other parameters to calculate the respective gains for the before mentioned parameters. These parameters are,

- Mass of the vehicle (M)
- Moment of inertia of vehicle (I)
- Moment of inertia of momentum disc (I_d)
- Distance between front tire and COG (a)
- Distance between rear tire and COG (b)
- Front tire stiffness (c_f)
- Rear tire stiffness (c_r)
- Longitudinal velocity (v_x)

Analytically torque controller can be divided into two sub controllers as shown in Figure 5.8, one part is responsible to calculate the system gains and other part determine the required torque.

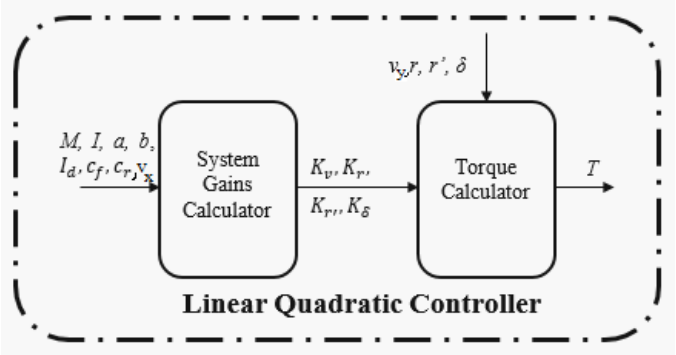

Figure 5.8: LQR block diagram

Using the equations provided in Section 2.2 system gains $K_v, K_r, K_{r\prime}, K_\delta$ are calculated over the range of longitudinal velocity v_x and shown below in Figure 5.9, Figure 5.10, Figure 5.11and, Figure 5.12 respectively.

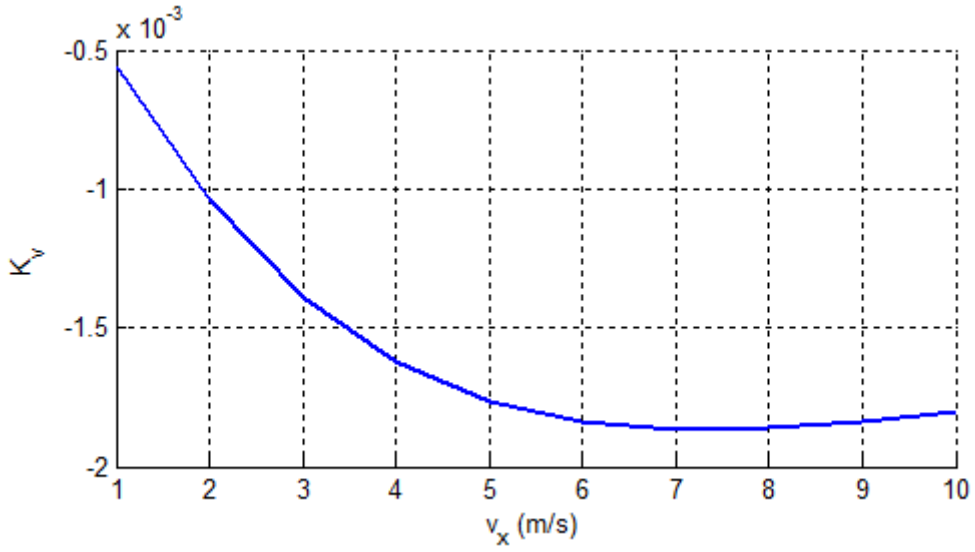

Figure 5.9: Gain for lateral velocity

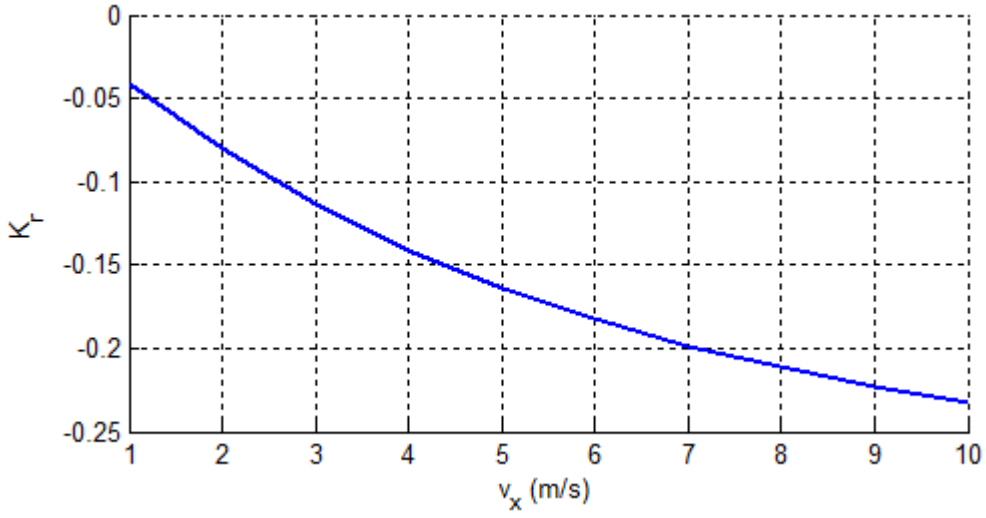

Figure 5.10: Gain for yaw rate

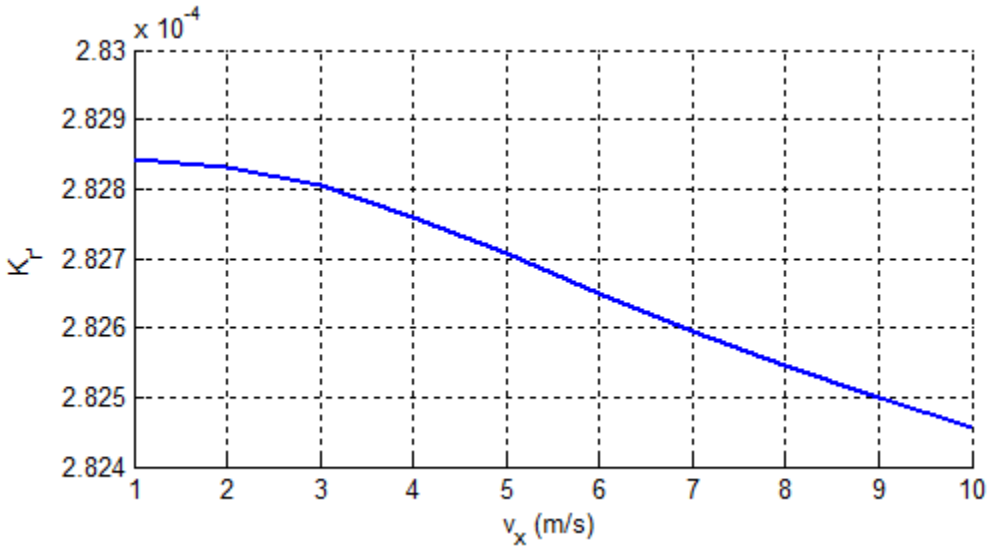

Figure 5.11: Gain for momentum disc angular velocity

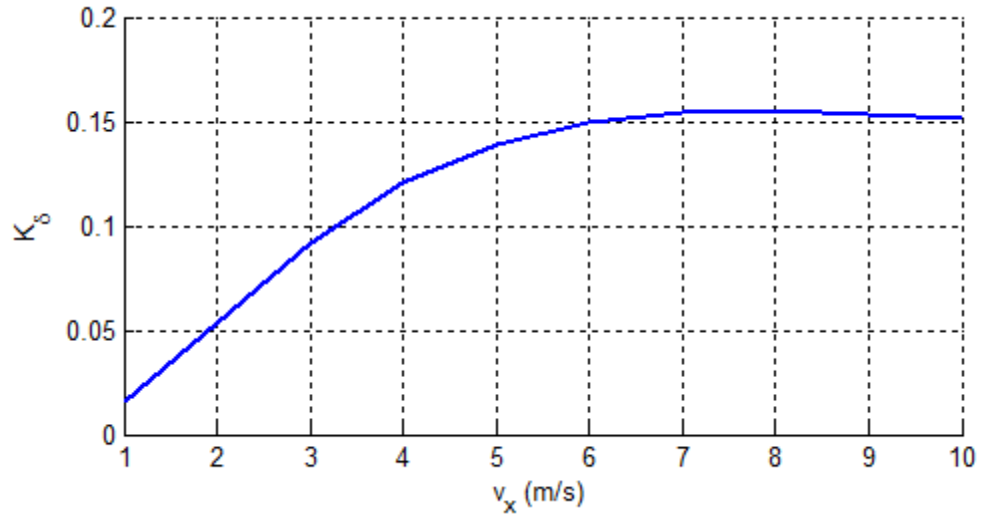

Figure 5.12: Gain for steering angle

Using the calculated gains and the Equation 2.1, required to control the vehicle yaw rate can then be calculated.

5.1.8. Momentum Wheel Controller

Required torque to generate is calculated by the torque controller i.e. LQR and then fed to the momentum wheel controller (MWC). Upon receiving the required torque signal, MWC (implemented in microcontroller as a PI controller) performs certain calculations as shown in the Figure 5.13. Calculate required acceleration using Equation 5.31, using Equation 5.32 determine required velocity to achieve acceleration calculated in the last step, and the preceding step is to generate respective PWM signal by changing the duty cycle of the motor control signal. Duty cycle of the PWM wave is calculated using the PI controller.

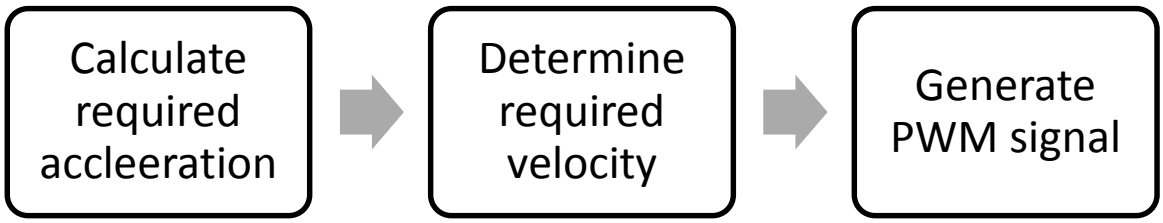

Figure 5.13: Steps involved to calculate duty cycle for PWM wave

$$Acceleration = (Torque/\ Moment\ of\ inertia) \tag{5.32}$$

$$\int_{t_i}^{t_f} a(t) = \ v(f) - v(i) \tag{5.33}$$

where, *v(f)* is the required velocity, *v(i)* is the current velocity.

Almost linear relation between the momentum disc velocity and applied duty cycle is observed and implemented using a third order polynomial equation. After calculating the required duty cycle, signal is applied to the PI controller as shown in the Figure 5.14.

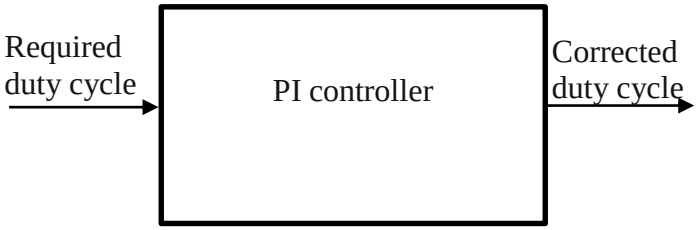

Figure 5.14: Duty cycle control using PI controller

A 20 kHz PWM wave is generated using the microcontroller. Resolution of the PWM wave is 16-bit which means, duty cycle of the PWM wave can be changed from 0% - 100% by varying the value from 0 - 2^{16} (0 - 65535). Calculated duty cycle is then applied (every 700 ms) to the PWM pin of the motor controller board as shown in the Figure 3.33, which in turn controls the velocity of the momentum wheel.

Microcontroller calculates required duty cycle every 25 ms but actually updates it every 700 ms using a 16-bit timer that generates an interrupt every 700 ms.

Chapter 6: **Experimental Results and Discussion**

Main objective of the prototype development is to improve the vehicle dynamics. To do so number of tests has been performed on the prototype. Required data (provided in 4.3.3.1) is saved into the measurement file each time of the test with the file name *"test_YY_MM_DD_HHMM.tdms"*. where, *YY* is year, *MM* is month, *DD* is day, *HH* is hour and, *MM* is minute. Saved file has an extension of *.tdms*, which can be later opened in LabView.

In order to analyse and see improvement in the yaw rate, pair of test is done. One without the MWC, one with the MWC and both are compared in the last. Due to the fact that the developed prototype was not able to follow the same maneuver every single time the test was performed, a testing criteria was developed to make sure that the testing and measurement conditions as same as possible in both the cases. Before discussing the testing results, first, testing criteria has been explained in the following section. Testing setup requires following components as shown in Figure 6.1.

- Developed prototype
- Power supply for DC motor
- Remote control for RC model car.
- Laptop to view and analyse the acquired data (with wireless communication).
- Battery to power up the electronic circuitry.

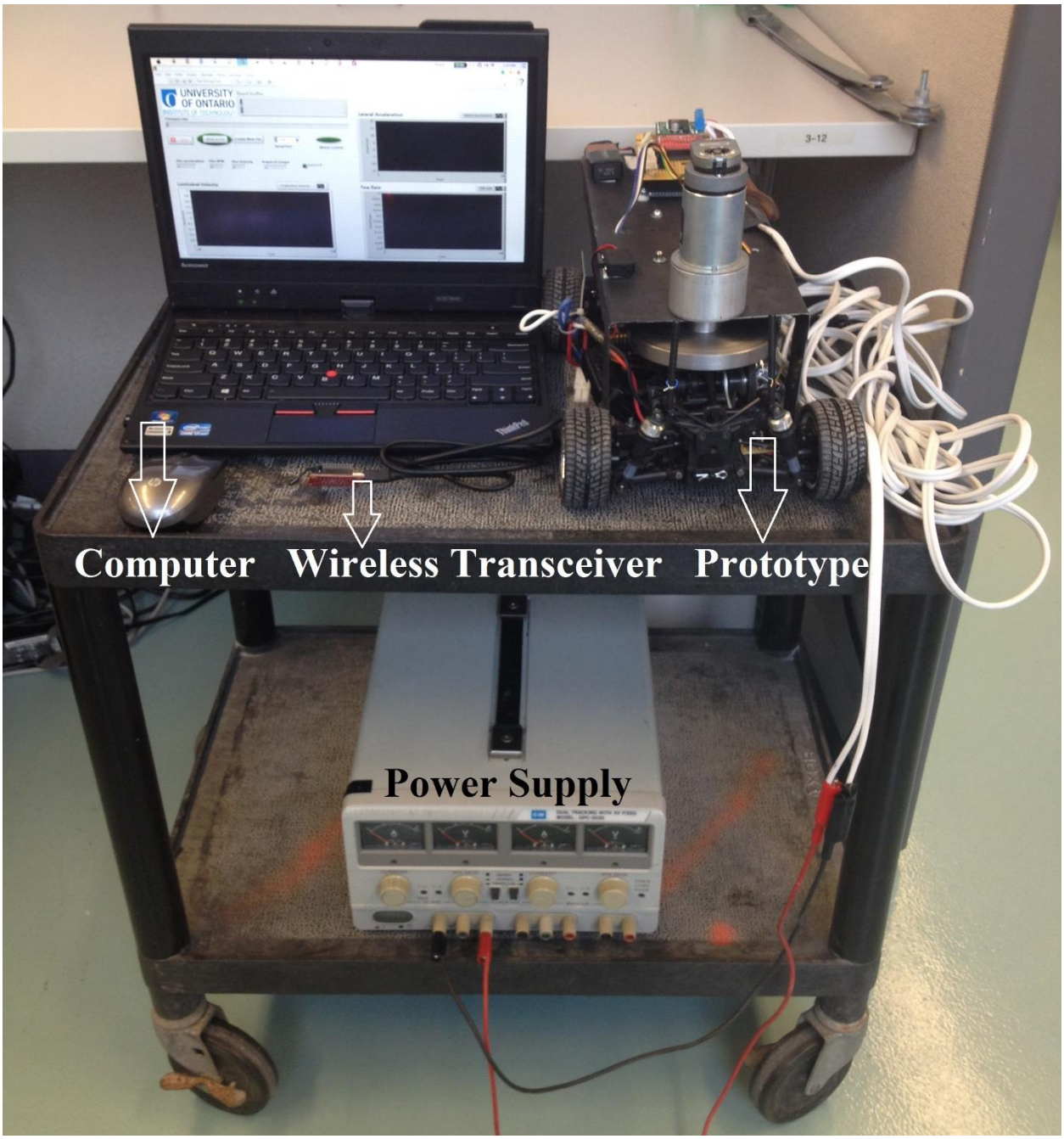

Figure 6.1: Experimental setup

6.1. TESTING CRITERIA

While performing test on the developed prototype, it was made sure that following conditions were met all the time.

- Every test starts from the same starting point.
- During each test three circular turns are made continuously as shown in Figure 6.2.
- Surface conditions are similar in both the tests.
- Vehicle's tires are alike.
- Steering angle of the vehicle is fixed.
- Relative longitudinal speed (in both cases) should be within $\pm 5\%$.
- Measure of handling is considered to be the vehicle yaw rate. (Equation 2.2)
- Measure of the safety is considered to be side-slip angle. (Equation 2.2)

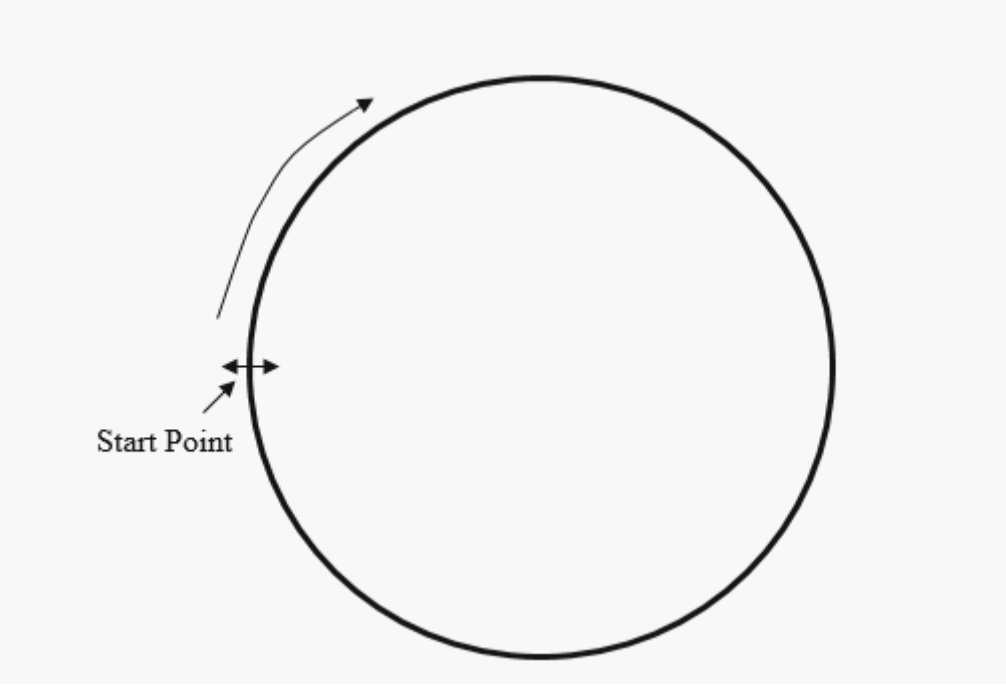

Figure 6.2: Test path

6.2. RESULTS

Tests were performed in three different conditions to observe vehicle's safety and handling measures. To start a test, RC model car is placed at the starting point with prefixed steering angle and longitudinal velocity. MWC is disabled using the switch shown in Figure 6.3. RC model car is then turned ON (using the RC model car remote) and allowed to run for three circular turns and the measurement data is collected via LabView. Same procedure is repeated by enabling the MWC, and the final results were compared to find the improvement in yaw rate, lateral acceleration and, side-slip angle.

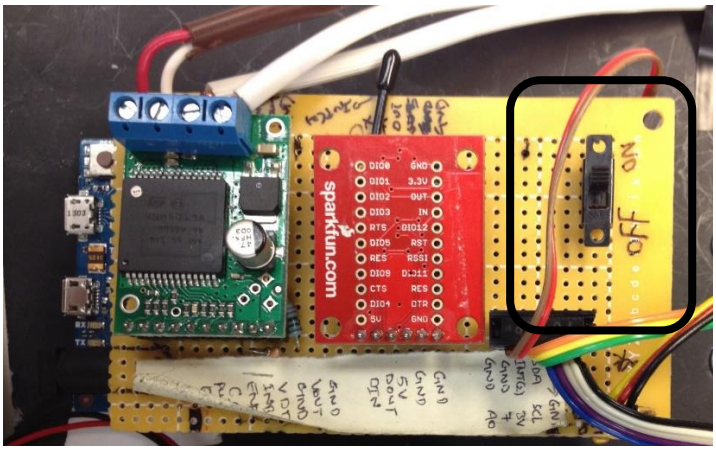

Figure 6.3: MWC enable/disable switch

After performing tests on different surfaces and with different tire conditions it was observed that the bare tire provided the highest friction, tire covered with shining tape provided moderate friction and, tire with solid lubricant has the lowest friction. Tests results for the above mentioned conditions are discussed in the following section.

Initially tests were performed by using continuous MWC which did not perform as per the requirements. So, discrete MWC was chosen to be used.

6.2.1. Low Friction Case

For this test RC model car's tires were covered with solid lubricant (soap) as shown in Figure 6.4 to decrease the tire-road friction. Steering angle was also fixed at a value of -0.436 radians. Average results for this type of test are presented in Table 6.1.

Table 6.1: Low friction test results

Test conditions v_x = 2.5 m/s (relative error = 3.03 %) δ = -0.436 radians			
Parameter	**Without MWC (rms)**	**With MWC (rms)**	**Relative Improvement (%)**
Yaw rate (rad/s)	0.1376	0.1218	11.47
Sideslip angle (degrees)	3.80	3.44	9.28
Lateral acceleration (m/s^2)	0.8239	0.7173	12.94

Figure 6.4: Tire covered with solid lubricant

Using MWC yaw rate is improved by 11.47 % as shown in Figure 6.5.

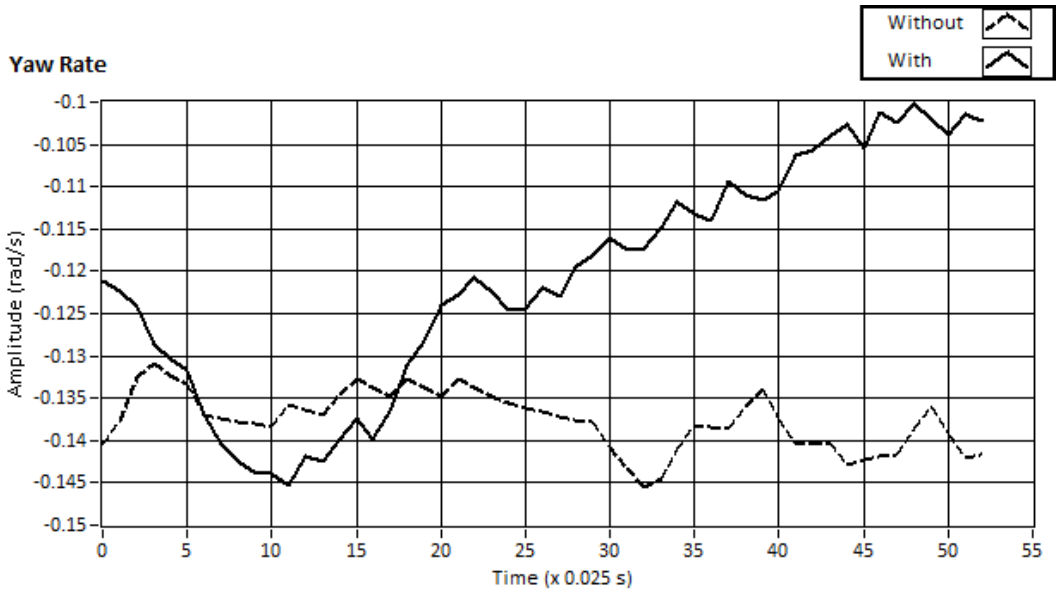

Figure 6.5: Yaw rate results with low friction

Vehicle side-slip angle is improved by 9.28% as shown in Figure 6.6.

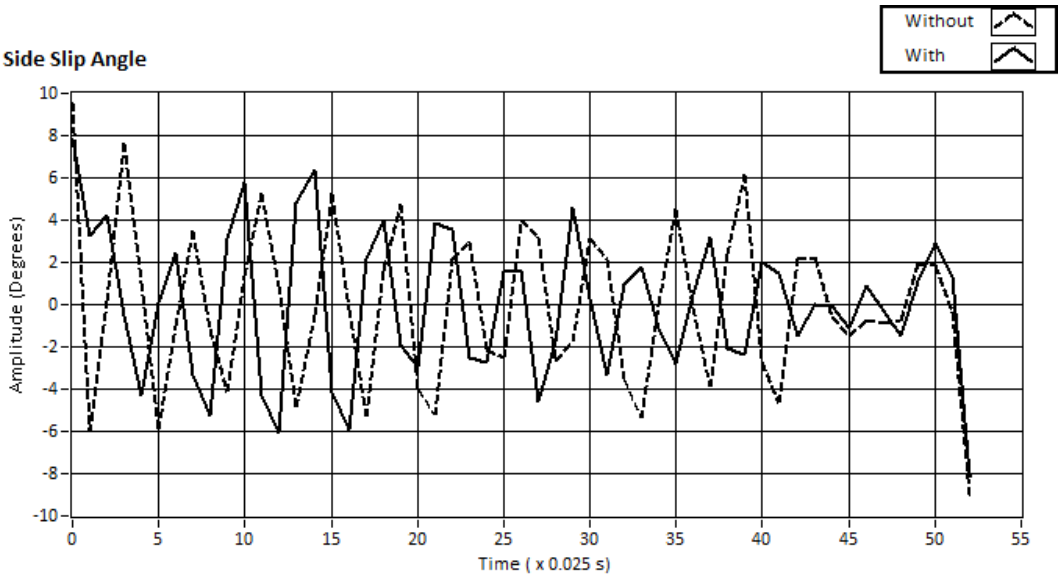

Figure 6.6: Side-slip results with low friction

Lateral acceleration of the vehicle with MWC remains always lower than without MWC. An improvement of 12.92% is achieved as shown in Figure 6.7.

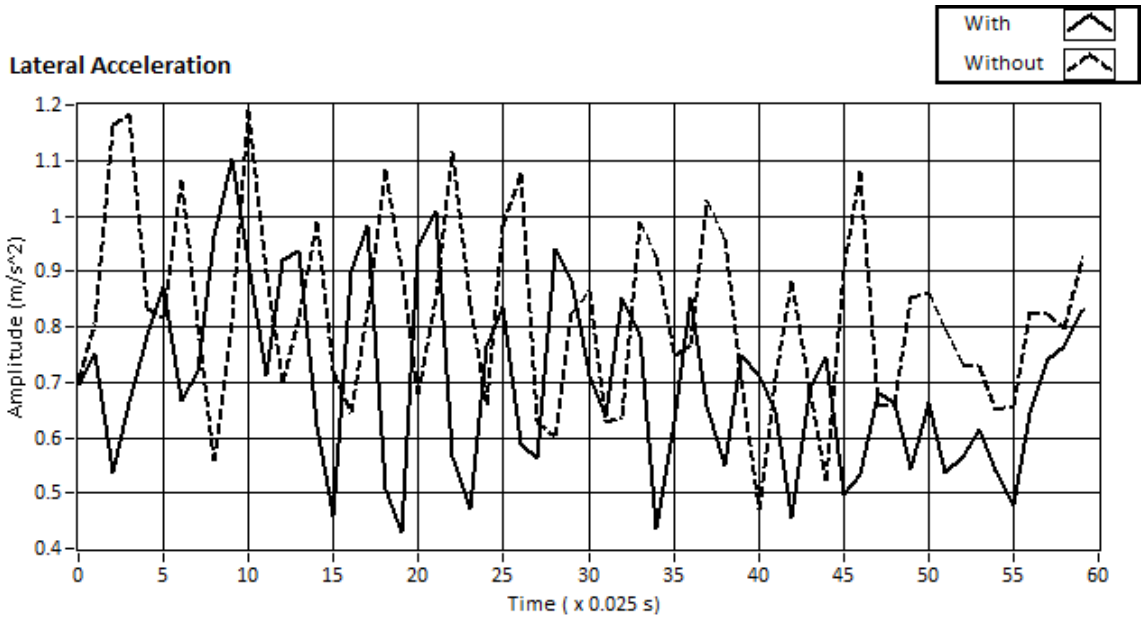

Figure 6.7: Lateral acceleration results with low friction

Curves for torque required and applied are shown in Figure 6.8.

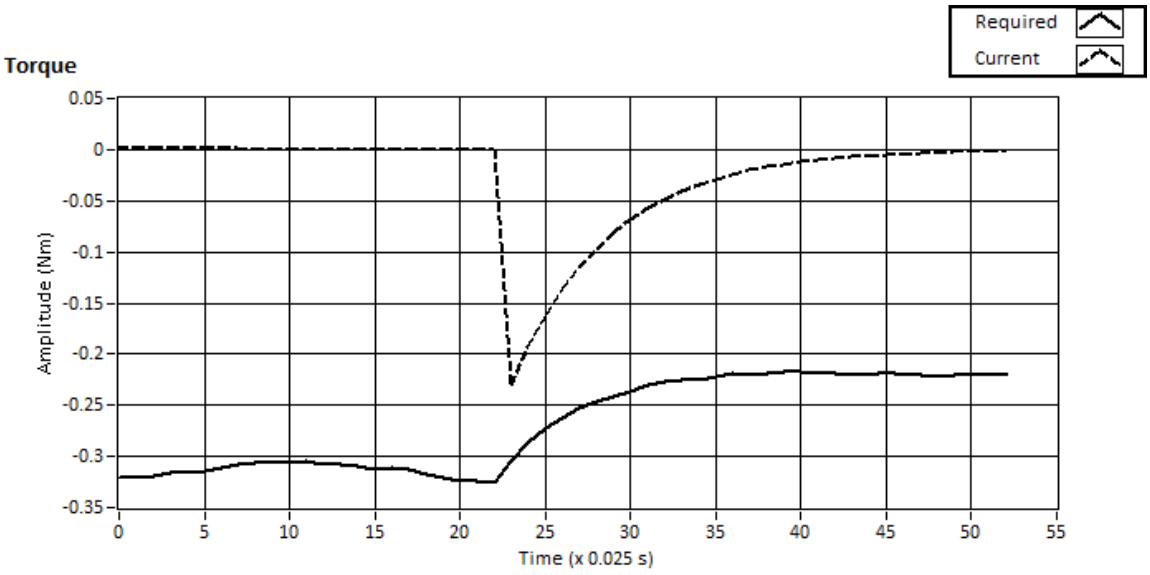

Figure 6.8: Torque applied/required

6.2.2. Medium Friction Case

Vehicle tires are covered using shining tape as shown in Figure 6.9 to perform test with moderate friction. Lower yaw rate than what was achieved using the solid lubricant was detected in the results. Steering angle was fixed at a value of -0.436. Results are tabulated in Table 6.2.

Table 6.2: Medium friction test results

Test conditions v_x = 2.17 m/s (relative error = 4.1%) δ = -0.436 radians			
Parameter	**Without MWC (rms)**	**With MWC (rms)**	**Relative Improvement (%)**
Yaw rate (r)	0.1692	0.1573	7.06
Sideslip angle (β)	4.40	3.55	11.42
Lateral acceleration (a_y)	0.9207	0.8153	11.44

Figure 6.9: Tire covered with shining tape

Improvement of 7.06% in the vehicle yaw rate is attained with MWC as shown in Figure 6.10.

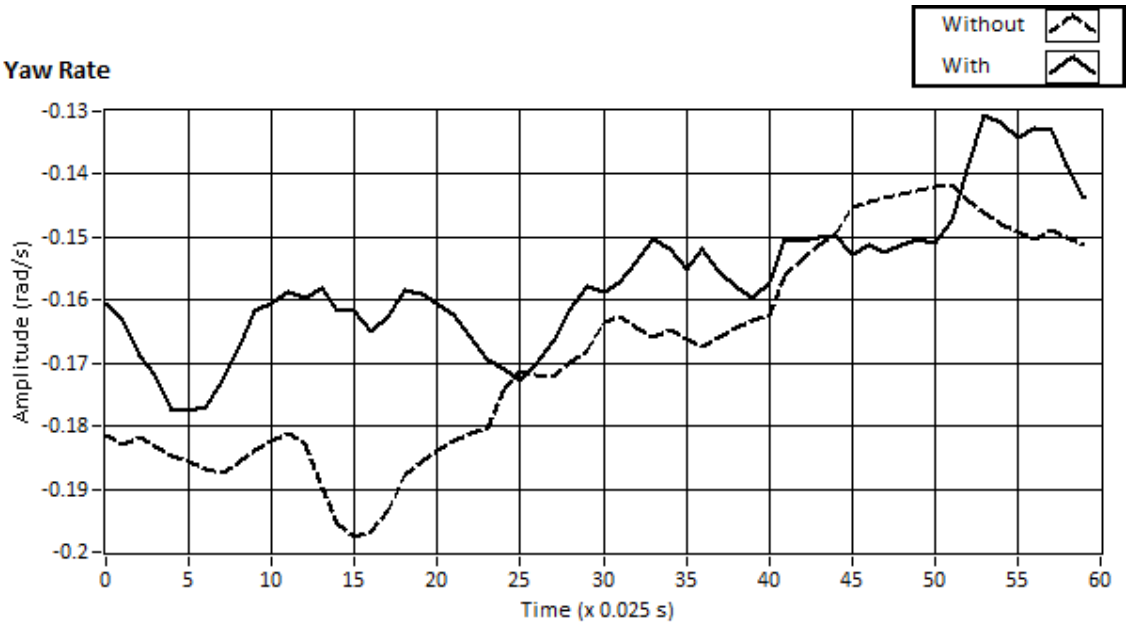

Figure 6.10: Yaw rate results for medium friction

Vehicle side-slip is improved by 11.42% as shown in Figure 6.11.

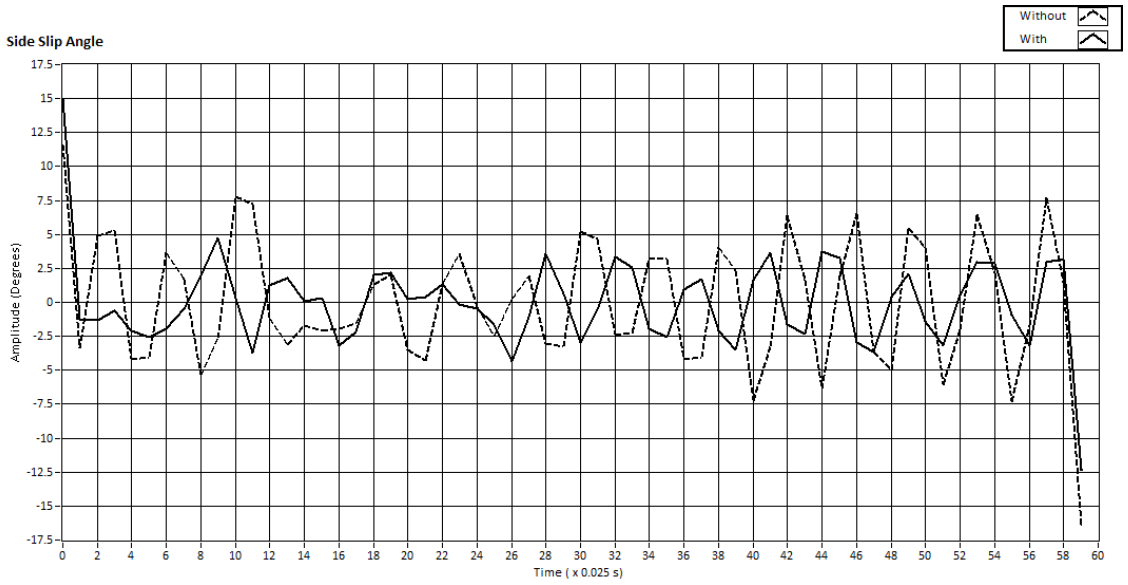

Figure 6.11: Side-slip results for medium friction

Lateral acceleration is noted to be enhanced by 11.44% as shown in Figure 6.12.

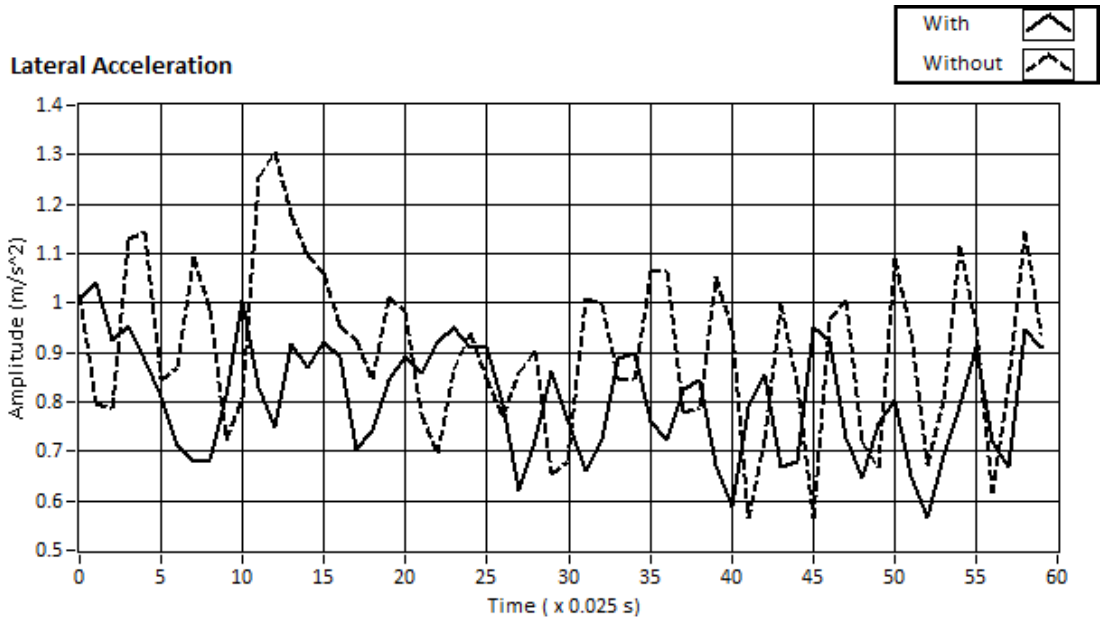

Figure 6.12: Lateral acceleration results for medium friction

Torque required and applied to the motor is shown in Figure 6.13.

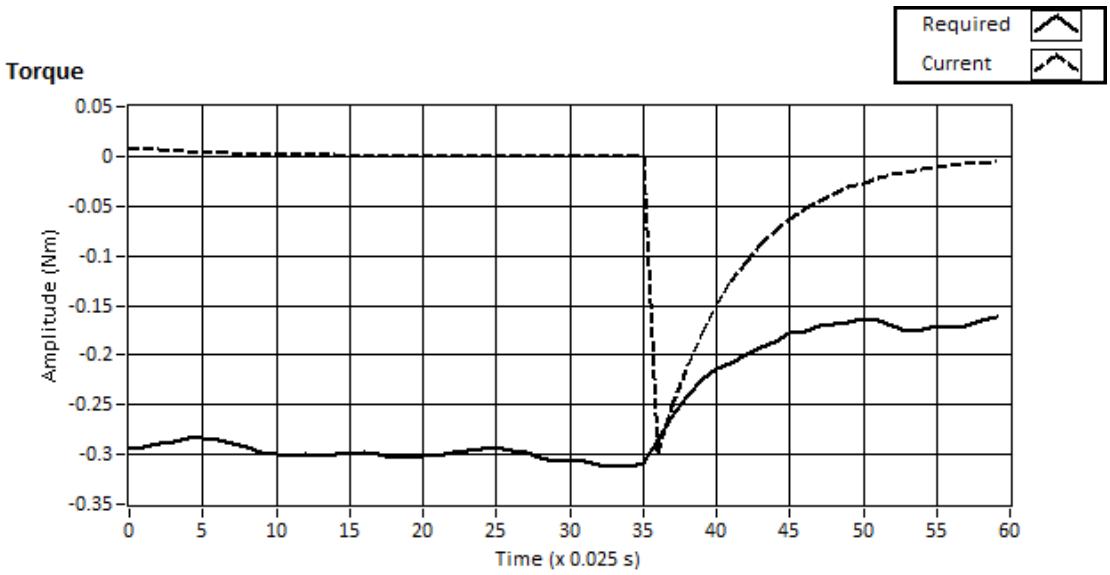

Figure 6.13: Torque applied/required

6.2.3. High Friction Case

During this test situation vehicle tires are left bare to attain high friction conditions. During the tests it was witnessed that with the bare tires RC model car had highest yaw rate among all three test conditions. Results are formulated in Table 6.3.

Table 6.3: High friction test results

Test conditions v_x = 2.15 m/s (relative error = 4.83 %) δ = -0.436 radians			
Parameter	**Without MWC (rms)**	**With MWC (rms)**	**Relative Improvement (%)**
Yaw rate (rad/s)	0.2349	0.2199	6.38
Sideslip angle (degrees)	10.87	9.83	9.53
Lateral acceleration (m/s^2)	1.33	1.20	10.04

Tire used in this test condition is shown in Figure 6.14.

Figure 6.14: Bare tire

Comparatively less improvement of 6.38% in the yaw rate is observed as shown in Figure 6.15.

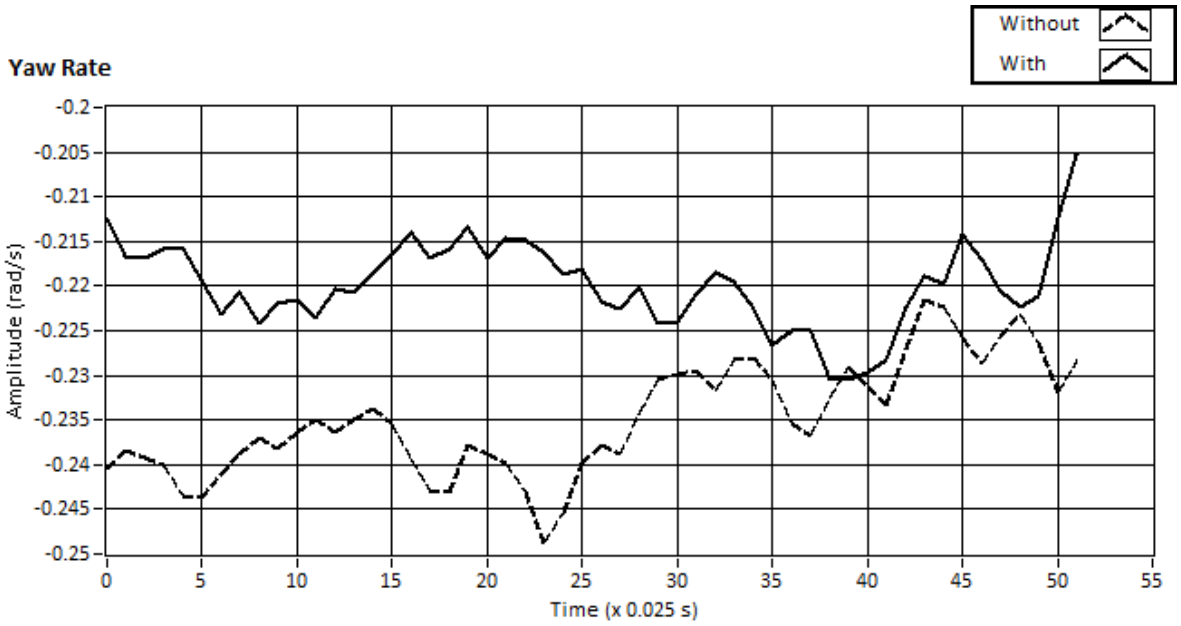

Figure 6.15: Yaw rate results for high friction

Safety measure of the vehicle i.e. side-slip is improved by 9.53%, shown in Figure 6.16.

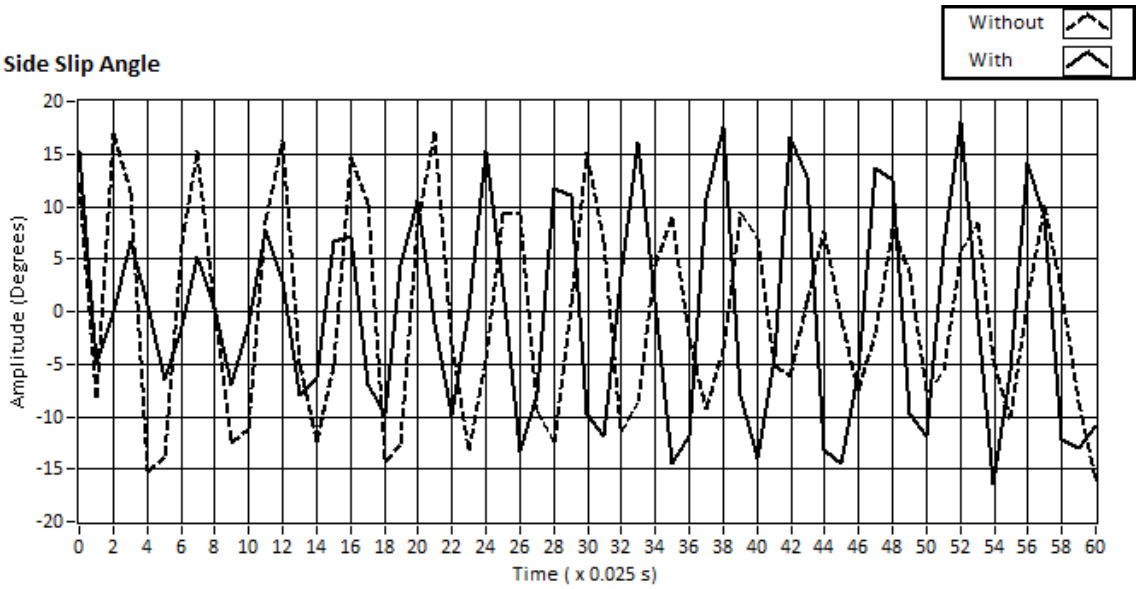

Figure 6.16: Side-slip results for high friction

Lateral acceleration is found to be enhanced by 10.04% when the used with MWC as shown in Figure 6.17.

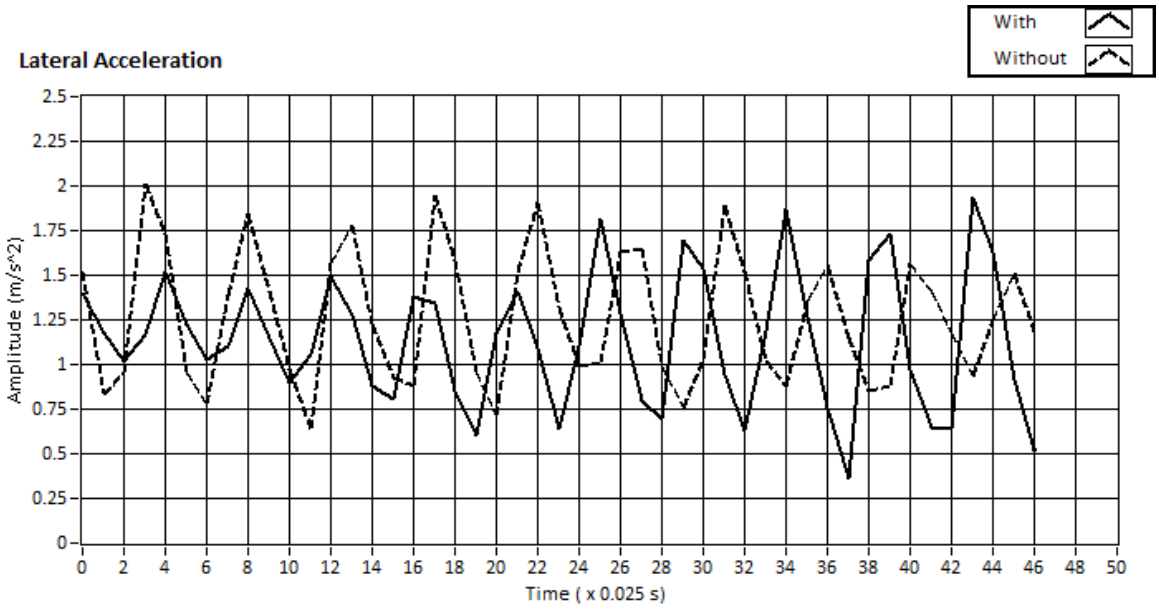

Figure 6.17: Lateral acceleration results for high friction

Torque required from the motor and torque applied to the motor is shown in Figure 6.18.

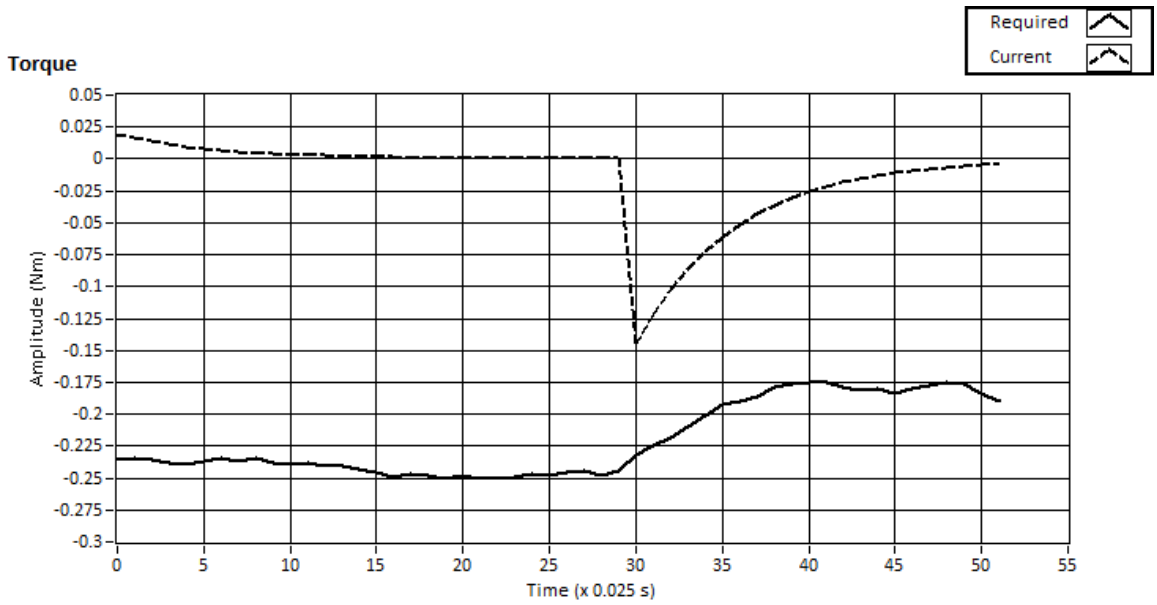

Figure 6.18: Torque required/applied

Chapter 7: **Conclusions and Future Work**

This chapter discusses the conclusions and some suggestions for the future work.

7.1. CONCLUSIONS

Popular yaw rate control methods and controller design are studied in the area of vehicle active stability. Yaw rate control method using the active momentum wheel was selected to be validated. Vehicle prototype was developed to perform experiments, analyze vehicle dynamics and confirm the effectiveness of the proposed control method.

RC model car's body was taken off to install all the electronic components and the momentum wheel. A few modifications were made to the RC model car's chassis according to the requirements. A holding structure made out of the metal sheet was used to put all the electronic circuitry and hold the DC motor on it. Holding structure was then fixed to the RC model car's chassis using the metal rods.

Momentum wheel was designed and manufactured to generate the required torque. Momentum wheel was constructed using a soft steel rod and developed in the university's mechanical workshop. Next, DC motor was installed with the developed momentum wheel and fixed on to the holding platform. Extra support was added at the rear of the RC model car to hold the weight of the DC motor and momentum wheel.

Pseudo integral kinematic estimation together with the Kalman filter was used to calculate the lateral and longitudinal velocity of the RC model car. Before mentioned techniques not only reduced the sensor measurement noise but also improved the vehicle state estimations. LQR controller was considered as the candidate to calculate the required torque. Required torque was then interpolated by the MWC in the form of duty cycle required to apply to the DC motor. MWC is a simple PI controller that improves the systems settling time and steady state error.

All the electronic circuitry is designed to support the data acquisition from sensors and to implement the control algorithms. ARM microcontroller was used as an embedded controller to implement the pseudo integral method, Kalman filter, LQR controller and, PI controller. To perform the before said functions microcontroller used its internal module such as PWM, 16-bit timers, ADC, I2C and, interrupts. IMU with 6-DOF provided the required linear and angular velocities. Two wireless modules were used for the communication between the RC model car and data acquisition PC.

Arduino IDE was used to program the microcontroller. X-CTU software provided by Digi Corporation used to configure the wireless module at a baud rate of 115200 bps. LabView was used to develop the on-line and post processing programs. On-line program acquires all the incoming data from the microcontroller and then put them in to their respective arrays. Acquired data is displayed on the graphs and saved into the measurement file for the post processing of the data. Post processing program allows to compare two different files of the test and compare the results.

Tests performed on the developed prototype shows an improvement in the vehicle handling (yaw rate) and safety (side-slip angle) measures, which verifies the considered control theory.

7.2. SUGGESTIONS FOR FUTURE WORK

Due to the fact that the developed prototype was not able to follow a certain maneuver, tests only in the extreme conditions (over steer case) were performed and validated. In the future a dedicate controller for the vehicle motor to follow a fixed maneuver can be used to perform test with standard lane change maneuvers. By applying this method acquired data can be compared easily.

As seen from the results, DC motor used to generate torque was not able to generate demanded torque every time. Reason for before mentioned failing is due to the low speed and torque ratio. A DC motor which can achieve high acceleration will further improve the system safety and handling measures.

Vehicle prototype can additionally be upgraded using the lighter build material. Also, metal rods used to connect holding platform to the vehicle's chassis can be replaced by the fixed holding structure attached to the chassis. Which will escalate the torque transfer from DC motor to the vehicle's chassis.

During the test and experimental work friction coefficient were unknown. In future work, vehicle parameters can be estimate using the system states and variables and included in the control design part.

Microcontroller used to acquire data and implement all the control algorithms discussed in this thesis, runs at clock of 84 MHz. Upgrading microcontroller with higher clock will further decrease the errors involved in the integration and differentiation equations. Also, system response time will improve drastically.

References：

1. F. Diba and E. Esmailzadeh, 2013, Integrated momentum wheel and differential braking control to improve vehicle dynamic performance. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of systems and control engineering*, **227**(7): pp. 563-576.
2. M. Nagai*et al.*, 2002, Study on integrated control of active front steer angle and direct yaw moment. *JSAE review*, **23**(3): pp. 309-315.
3. N. Hamzah*et al.*, 2012, Yaw stability improvement for four-wheel active steering vehicle using sliding mode control. in *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*. IEEE127-132.
4. A. Hu and F. He, 2011, Variable structure control for active front steering and direct yaw moment. in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*. IEEE3587-3590.
5. W. Klier*et al.*, 2004, Concept and functionality of the active front steering system. *SAE paper 20042212*.
6. Y. Ji*et al.*, 2014, Integrated control of active front steering and direct yaw moment based on model predictive control. in *Control and Decision Conference (2014 CCDC), The 26th Chinese*. IEEE2044-2049.
7. Y. Ikeda, 2010, Active steering control of vehicle by sliding mode control-switching function design using SDRE. in *Control Applications (CCA), 2010 IEEE International Conference on*. IEEE1660-1665.

8. S. Mammar and D. Koenig, 2002, Vehicle handling improvement by active steering. *Vehicle system dynamics*, **38**(3): pp. 211-242.

9. P. Falcone*et al.*, 2008, Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International journal of robust and nonlinear control*, **18**(8): pp. 862-875.

10. P. Falcone*et al.*, 2007, Predictive active steering control for autonomous vehicle systems. *Control Systems Technology, IEEE Transactions on*, **15**(3): pp. 566-580.

11. M. Nagai*et al.*, 1997, Integrated control of active rear wheel steering and direct yaw moment control. *Vehicle System Dynamics*, **27**(5-6): pp. 357-370.

12. M. Nagai*et al.*, 1999, Integrated control of active rear wheel steering and yaw moment control using braking forces. *JSME international journal. Series C, Mechanical systems, machine elements and manufacturing*, **42**(2): pp. 301-308.

13. Y. Hirano and K. Fukatani, 1997, Development of robust active rear steering control for automobile. *JSME International Journal. Series C, dynamics, control, robotics, design and manufacturing*, **40**(2): pp. 231-238.

14. M. Aripin*et al.*, 2014, A Review of Active Yaw Control System for Vehicle Handling and Stability Enhancement. *International Journal of Vehicular Technology*, **2014**.

15. T. Hiraoka*et al.*, 2009, Automatic path-tracking controller of a four-wheel steering vehicle. *Vehicle System Dynamics*, **47**(10): pp. 1205-1227.

16. G.-D. Yin*et al.*, 2011, A study on μ-synthesis control for four-wheel steering system to enhance vehicle lateral stability. *Journal of dynamic systems, measurement, and control*, **133**(1): pp. 011002.

17. R. Marino*et al.*, 2007, Nonlinear PI front and rear steering control in four wheel steering vehicles. *Vehicle System Dynamics*, **45**(12): pp. 1149-1168.

18. S. Di Cairano and H. Tsengz, 2010, Driver-assist steering by active front steering and differential braking: design, implementation and experimental evaluation of a switched model predictive control approach. in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE2886-2891.

19. B. Boada*et al.*, 2005, Fuzzy-logic applied to yaw moment control for vehicle stability. *Vehicle System Dynamics*, **43**(10): pp. 753-770.

20. M. Mirzaei, 2010, A new strategy for minimum usage of external yaw moment in vehicle dynamic control system. *Transportation Research Part C: Emerging Technologies*, **18**(2): pp. 213-224.

21. V. Cerone*et al.*, 2009, Yaw stability control design through a mixed-sensitivity approach. *Control Systems Technology, IEEE Transactions on*, **17**(5): pp. 1096-1104.

22. E. Esmailzadeh*et al.*, 2003, Optimal yaw moment control law for improved vehicle handling. *Mechatronics*, **13**(7): pp. 659-675.

23. A. Hu and B. Lv, 2010, Study on mixed robust control for integrated active front steering and direct yaw moment. in *Mechatronics and Automation (ICMA), 2010 International Conference on*. IEEE29-33.

24. M. Canale*et al.*, 2007, Robust vehicle yaw control using an active differential and IMC techniques. *Control Engineering Practice*, **15**(8): pp. 923-941.

25. R.P. Osborn and T. Shim, 2006, Independent control of all-wheel-drive torque distribution. *Vehicle system dynamics*, **44**(7): pp. 529-546.

26. K. Matsuno, *Driving torque distribution control system for vehicle and the method thereof*. 1998, Google Patents.

27. W. Yihu*et al.*, 2007, A fuzzy control method to improve vehicle yaw stability based on integrated yaw moment control and active front steering. in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. IEEE1508-1512.

28. S. Zhang*et al.*, 2009, Vehicle stability control strategy based on active torque distribution and differential braking. in *Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on*. IEEE922-925.

29. A.H. Niasar*et al.*, 2003, Yaw moment control via emotional adaptive neuro-fuzzy controller for independent rear wheel drives of an electric vehicle. in *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on*. IEEE380-385.

30. L. Junwei and Y. Huafang, 2009, Fuzzy logic applied to yaw moment control for vehicle stability. in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. IEEE386-390.

31. L. Gang*et al.*, 2011, Vehicle active front steering and yaw moment integrated control. in *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*. IEEE787-790.

32. P. Falcone*et al.*, 2008, MPC-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, **46**(S1): pp. 611-628.

33. D. Bernardini*et al.*, 2009, Drive-by-wire vehicle stabilization and yaw regulation: A hybrid model predictive control design. in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE7621-7626.

34. S. Di Cairano*et al.*, 2010, Steering vehicle control by switched model predictive control. in *Advances in Automotive Control*. 1-6.

35. H. Yu*et al.*, 2013, Direct Yaw-Moment Hinfinity Control of Motor-Wheel Driving Electric Vehicle. in *Vehicle Power and Propulsion Conference (VPPC), 2013 IEEE*. IEEE1-5.

36. N. Hamzah*et al.*, 2012, Vehicle stability enhancement based on second order sliding mode control. in *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*. IEEE580-585.

37. Y. Chao-chun*et al.*, 2010, Robust active front steering control based on the mu control theory. in *Electrical and Control Engineering (ICECE), 2010 International Conference on*. IEEE1827-1829.

38. S. Xuan*et al.*, 2012, Active yaw-moment control based on logic threshold and PID control. in *Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on*. IET338-341.

39. I. Yang*et al.*, 2013, Integrated control systems of active front steering and direct yaw moment control using dynamic inversion. in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE1303-1306.

40. M. Mirzaei*et al.*, 2008, An optimal approach to non-linear control of vehicle yaw dynamics. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, **222**(4): pp. 217-229.

41. X.-S. Ge and L.-Q. Chen, 2004, Attitude control of a rigid spacecraft with two momentum wheel actuators using genetic algorithm. *Acta Astronautica*, **55**(1): pp. 3-8.

42. M. Scharfe*et al.*, 2001, Design and development of a compact magnetic bearing momentum wheel for micro and small satellites.

43. A. Goodarzi*et al.*, 2014, Innovative Active Vehicle Safety using Integrated Stabilizer Pendulum and Direct Yaw Moment Control. *Journal of Dynamic Systems, Measurement, and Control*.

44. F. Diba and E. Esmailzadeh, 2012, Dynamic performance enhancement of vehicles with controlled momentum wheel system. in *American Control Conference (ACC), 2012*. IEEE6539-6544.

45. J.J. Oh and S.B. Choi, 2012, Vehicle velocity observer design using 6-D IMU and multiple-observer approach. *Intelligent Transportation Systems, IEEE Transactions on*, **13**(4): pp. 1865-1879.

46. K. Zhou*et al.*, *Robust and optimal control*. Vol. 40. 1996: Prentice Hall New Jersey.

47. A. Goodarzi and E. Esmailzadeh, 2007, Design of a VDC system for all-wheel independent drive vehicles. *Mechatronics, IEEE/ASME Transactions on*, **12**(6): pp. 632-639.

48. Y.A. Ghoneim*et al.*, 2000, Integrated chassis control system to enhance vehicle stability. *International Journal of Vehicle Design*, **23**(1): pp. 124-144.

49. B.D. Anderson and J.B. Moore, *Optimal control: linear quadratic methods*. 2007: Courier Dover Publications.

50. Futaba. *Servo motor*. http://www.gpdealera.com/cgi-bin/wgainf100p.pgm?I=FUTM0031. Accessed: 2014/12/05;

51. I. Lee*et al.*, 2005, Development and analysis of the vertical capacitive accelerometer. *Sensors and Actuators A: Physical*, **119**(1): pp. 8-18.

52. S. Beeby*et al.*, *Chapter 5 in "MEMS Mechanical Sensors,"*. 2004, Boston: Artech House Academic Press.

53. I. Inc. *MPU6050 Specifications*. 52, Data Sheet. http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf.

54. N. Semiconductors, 2000, I2C-bus specification and user manual, UM10204.

55. I. Inc *MPU6050 register map*. 2012. 47, Data Sheet. http://invensense.com/mems/gyro/documents/RM-MPU-6000A.pdf.

56. Robotshop.com. *ZigBee module*. [Image] http://www.robotshop.com/ca/en/xbee-2mw-wire-antenna-series2-zigbee-mesh.html. 2014 Accessed: 2014/12/10;

57. pololu.com. *VNH5019*. [Image] http://www.pololu.com/product/1451/pictures. Accessed: 2014/12/10;

58. L. Allegro Microsystems, 2013, Hall-Effect IC Applications Guide, 27701-AN, Rev. 2.

59. electronics-tutorial.ws. *Hall Sensor*. [Image] http://www.electronics-tutorials.ws/electromagnetism/hall-effect.html. Accessed: 2014/10/18;

60. R. Faragher, 2012, Understanding the basis of the Kalman filter via a simple and intuitive derivation. *IEEE Signal Processing Magazine*, **29**(5): pp. 128-132.

61. M.S. Grewal and A.P. Andrews, *Kalman filtering: theory and practice using MATLAB*. 2011: John Wiley & Sons.

62. Udacity. *Kalman Matrices*. https://www.udacity.com/wiki/cs373/kalman-filter-matrices. Accessed: 2014/09/08; Matrices Dimensions].

63. G. Welch and G. Bishop, *An introduction to the Kalman filter*. 1995.

64. A. Devices. *Accelerometer specifications*. [Definations] http://www.analog.com/en/content/td_accelerometer_specifications_definitions/fca.html. Accessed: 2014/12/08;