

Motorcycle Electronics & Control Systems
M6X00417
School of Engineering
BEng Motorcycle Engineering
Philip Harris
P143447
April 2019



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE

Declaration of Authenticity

'This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.'

Signed:

Date:

Acknowledgments

The author would like to thank the following for their help and support during this project.

My family – Father Jim, Step Mother Jo, brothers Andrew and James, sister Lesley and my late mother Yvonne.

My friends – Anthony Collins, Chris Wilson, Luke Godden, Luke Jones and Sam Pontin. For all the highs, lows and everything else in between that we've encountered over the years.

Malcolm McDonald – Thank you for the time spent advising me throughout the project. Your knowledge and experience have been invaluable.

UWTSD – To all of the staff and lecturers who have helped me get this far from foundation year.

Camille Relet – Thank you for always believing in and supporting me during this time.

Contents

1.0 – Introduction.....	1
1.1 – Abstract.....	1
1.2 – Aims & Objectives.....	1
2.0 – Research	2
2.1 – Motorcycle Electronic Systems.....	2
2.1.1 – RBW (Ride by Wire).....	4
2.1.2 – Bosch MSC (Motorcycle Stability Control).....	7
2.1.2.1 – Wheel Speed Sensor	8
2.1.2.2 – IMU (Inertial Measurement Unit)	9
2.2 – Communication Signals.....	11
2.2.1 – Analog	12
2.2.2 – Digital	12
2.2.3 – Bus Interfaces.....	12
2.2.3.1 – Serial.....	12
2.2.3.2 – SPI (Serial Peripheral Interface).....	14
2.2.3.3 – I ² C (Inter-Integrated Circuit).....	15
2.2.3.4 – TWI (Two Wire Interface).....	16
2.2.3.5 – CAN (Controller Area Network).....	16
2.3 – Development Microcontroller	18
2.3.1 – Raspberry Pi	18
2.3.2 – Arduino	20
2.3.3 – Comparison & Selection	21
3.0 – Development	23
3.1 – Hardware	23
3.1.1 – Arduino Shield	23
3.1.2 – Throttle.....	23
3.1.3 – Servo Motor.....	24
3.1.4 – Inductive Proximity Sensor	26
3.1.5 – IMU (Inertial Measurement Unit)	28
3.1.6 – LCD Display.....	28
3.2 – Simulink.....	28
3.2.1 – Simulink Blocks	29
3.2.2 – Model	33

3.2.3 – Throttle Subsystem.....	33
3.2.4 – LCD Subsystem	40
3.2.5 – Warm-Up Subsystem.....	41
3.2.6 – Wheel Speed Subsystem.....	41
3.2.7 – Servo Output Subsystem	42
4.0 – Analysis & Further Work	43
4.1 – Simulink Model.....	43
4.2 – IMU.....	44
4.3 – Further Work	44
References.....	49
Appendix A	55
Appendix B	56
Appendix C	62
Appendix D.....	63
Appendix E.....	65
Appendix F.....	67
Appendix G	68

List of Figures

Figure 1 - 2010 Triumph Daytona 675 (AutoEvolution, 2019)	2
Figure 2 - 2019 BMW S1000RR (44 Teeth, 2018).....	2
Figure 3 - Triumph Daytona 675 Handlebar & Switches Parts Diagram (World Of Triumph, 2019)	4
Figure 4 - Triumph Daytona 675 Throttles, Injectors & Fuel Rail Parts Diagram (World Of Triumph, 2019)	5
Figure 5 - Domino XM2 Throttle with Interchangeable Cams (Bike Torque Racing, 2019)	5
Figure 6 - Demand Sensor (AF1 Racing, 2019)	6
Figure 7 – Exploded View, KTM Ride by Wire Throttle (eBay, 2019).....	6
Figure 8 - 2017 BMW S1000RR Throttle Bodies Parts Diagram (Max BMW Motorcycles, 2019).....	7
Figure 9 - 2014 KTM 1290 Super Duke R Throttle Maps (Motorcycle.com, 2013)	7
Figure 10 - Bosch MSC System Components (Bosch, 2019)	8
Figure 11 - Wheel Speed Sensor (Bosch, 2019)	8
Figure 12 - ABS Equipped Motorcycle (Motorcycle.com, 2019)	8
Figure 13 - Simulink Generated ABS Signal Example	9
Figure 14 – IMU Sensor (Bosch, 2019)	10
Figure 15 – IMU Measurements (BikesMedia, 2017)	10
Figure 16 - Acceleration Graph of a Motorcycle With and Without IMU (BikesMedia, 2017)	11
Figure 17 - Analog & Digital Signal Comparison (Chegg, 2019)	11
Figure 18 - 50%, 75% and 25% Duty Cycle Examples (SparkFun, 2019)	12
Figure 19 - Serial Connectivity (SparkFun, 2019)	13
Figure 20 - Serial Asynchronous Data Transfer (SparkFun, 2019)	14
Figure 21 - SPI Synchronous Data Transfer (SparkFun, 2019)	14
Figure 22 - SPI Master / Slave Interface (SparkFun, 2019)	15
Figure 23 - I ² C Master / Slave Interface (SparkFun, 2019).....	16
Figure 24 - CAN 2.0B Frame (CSS Electronics, 2018)	17
Figure 25 - Raspberry Pi 3 Model B+ (Raspberry Pi Foundation, 2019).....	19
Figure 26 - Arduino Mega 2560 Rev3 (Arduino, 2019)	20
Figure 27 - Raspberry Pi 3 Model B+ GPIO Layout (Raspberry Pi Foundation, 2019).....	21
Figure 28 - Raspberry Pi 3 Model B+ GPIO Pin Map (Raspberry Pi Foundation, 2019)	21
Figure 29 - Grove - Mega Shield V1.2 (Arduino, 2019)	23
Figure 30 - Servo Motor SG-90 (Components101, 2017)	24
Figure 31 - Servo Motor SG-90 Dimensions (Components101, 2017).....	24
Figure 32 - Servo Motor SG-90 PWM (Components101, 2017).....	25
Figure 33 - GKB-M0524NA Inductive Proximity Sensor Dimensions (Hugong Automation, 2019)	26
Figure 34 - MPU6050 IMU (Amazon, 2019)	28
Figure 35 - HD44780 16x2 LCD (Amazon, 2019)	28
Figure 36 - Initial Throttle Wiring Diagram	34
Figure 37 - Initial Simulink Model	34
Figure 38 - Throttle Signal 1, 0% Opening.....	35
Figure 39 - Throttle Signal 1, 100% Opening.....	35
Figure 40 - Throttle Signal 2, 0% Opening.....	36
Figure 41 - Throttle Signal 2, 100% Opening.....	36
Figure 42 - Signal 1 (Yellow) & Signal 2 (Blue) At 0% Opening With Median Filter	36
Figure 43 - Signal 1 (Yellow) & Signal 2 (Blue) At 100% Opening With Median Filter	36
Figure 44 - Signal 1 Lookup Values	37

Figure 45 - Signal 2 Lookup Values	37
Figure 46 - Excel Throttle Profile Calculator	37
Figure 47 - Excel Throttle Profile Calculator	38
Figure 48 - Throttle Profile Calculator Graph.....	39
Figure 49 - Polynomial Block.....	39
Figure 50 – LCD Wiring Diagram	40
Figure 51 - Throttle Signal 1 Floating Without Pull-Down Resistor	44
Figure 52 - Throttle Signal 1 Floating With Pull-Down Resistor	44
Figure 53 - 2010 Triumph Daytona 675 Throttle Position Signal	45
Figure 54 - 2010 Triumph Daytona 675 Gear Position Signal	45
Figure 55 - Simulink RPM Calculation	46
Figure 56 - Excel RPM Calculator	46
Figure 57 - Excel Gearing Calculator	47
Figure 58 - Load Cell Signal	48

List of Tables

Table 1 - 2009-2012 Triumph Daytona 675 Specifications (Triumph Motorcycles, 2009)	2
Table 2 - 2019 BMW S1000RR Specifications (BMW Motorrad, 2019)	3
Table 3 - CAN Message Components (CSS Electronics, 2018) (Texas Instruments, 2016)	18
Table 4 - Raspberry Pi 3 Model B+ Specifications (Raspberry Pi Foundation, 2019)	19
Table 5 - Arduino Mega 2560 Rev3 Specifications (Arduino, 2019) (Atmel, 2019)	20
Table 6 - Servo Motor SG-90 Specifications (Components101, 2017).....	25
Table 7 - GKB-M0524NA Specifications (Hugong Automation, 2019)	26
Table 8 - MATLAB Data Types (MathWorks, 2019).....	30

1.0 – Introduction

1.1 – Abstract

This report constitutes the research, design and evaluation of a test and development platform to enable the development of control strategies for motorcycles. Due to the nature of enhanced control, electronic systems are more important than ever to improve performance, efficiency and safety of both the vehicle and rider.

1.2 – Aims & Objectives

When looking at motorcycle electronics the control strategies are not freely available to evaluate, and the only documentation lies in their operation. While the principles of strategies such as ride by wire, gear shift assist, launch control and traction control are known the actual knowledge of what is happening within the hardware is a secret. The most open platform that can be investigated outside of an OEM are ECU manufacturers such as MoTeC or MecTronik, but these come at a substantial cost that is often not affordable for the user.

The aims of this project are:

- Research current available strategies and solutions
- Establish an affordable development platform that can be easily be replicated, enhanced and improved
- Develop RBW (ride by wire) conversion for a conventional motorcycle
- Develop control strategies to take advantage of the RBW function

2.0 – Research

2.1 – Motorcycle Electronic Systems



Figure 1 - 2010 Triumph Daytona 675 (AutoEvolution, 2019)

Type :	Liquid-Cooled, 12 valve, DOHC, in-line 3 cylinder
Displacement :	675 cc
Bore x Stroke :	74 x 52.3 mm
Compression Ratio :	12.7:1
Power :	94 kW (126 hp) at 12,600 rpm
Torque :	73 Nm (53 lb. Ft) at 11,750 rpm
Fuel System :	Multipoint sequential electronic fuel injection with forced air induction

Table 1 - 2009-2012 Triumph Daytona 675 Specifications (Triumph Motorcycles, 2009)

As the chosen development vehicle the 2010 Triumph Daytona 675 outside of the core systems to allow the engine to operate exhibits no additional electronic systems.

Since the production of this model from 2009-12 this is an area that has advanced greatly in motorcycles of all disciplines in the name of both safety and performance.



Figure 2 - 2019 BMW S1000RR (44 Teeth, 2018)

Type :	Water/oil-cooled 4-cylinder 4-stroke in-line engine, four titanium valves per cylinder, BMW ShiftCam
Displacement :	999 cc
Bore x Stroke :	80 x 49.7 mm
Compression Ratio :	13.3:1
Power :	152 kW (207 hp) at 13,500 rpm
Torque :	113 Nm (83 lb. Ft) at 11,000 rpm
Fuel System :	Electronic injection, variable intake

Table 2 - 2019 BMW S1000RR Specifications (BMW Motorrad, 2019)

In comparison the new for 2019 BMW S1000RR offers;

- Dynamic Traction Control (DTC)
 - Controls the amount of spin at the rear wheel under acceleration by cutting ignition, fuel or both
- Anti-lock Braking System (ABS)
 - User selectable modes to set the level of ABS, intrusiveness varies depending on lean angle (Cornering ABS)
- Riding Modes
 - 4 user selectable modes (Rain, Road, Dynamic and Race) that change throttle response, engine brake control, traction control, wheelie control and ABS
- Gear Shift Assist
 - Allows gear changes both up and down the gearbox without engaging the clutch
- Dynamic Damping Control (DDC)
 - Suspension damping if required is electronically adjusted in 10ms intervals to adapt to different riding conditions
- Hill Start Control (HSC)
 - Assesses if the vehicle is on an incline, once the rider has released the brakes the function holds the brakes for a predetermined amount of time
- Slide Control
 - The vehicle can determine yaw and allow a certain amount of spin to engage a controlled slide
- Wheelie Control

- The vehicle can determine pitch and adjust ignition and fuel to avoid excessive wheelies
- Pitlane Limiter
 - Trackday or race option. Ignition control to cut spark to maintain and not exceed a given velocity
- Launch Control
 - Race option. Uses available systems to assist the riders race start to maintain traction and minimise wheelies to ensure maximum acceleration
- Cruise Control
 - User inputted speed which the vehicle maintains until interrupted by the rider either engaging the clutch or brakes
- Tyre Pressure Control
 - Displays on the vehicle dashboard the current tyre pressures in the front and rear wheels and can alert the rider should they fall below a set threshold

(BMW Motorrad, 2019)

How is this possible? What differs between the two vehicles to allow the BMW to offer so many rider aids?

2.1.1 – RBW (Ride by Wire)

The throttle operation for example is handled very differently between the two, the Triumph uses a conventional cable operated throttle while the BMW uses ride by wire.

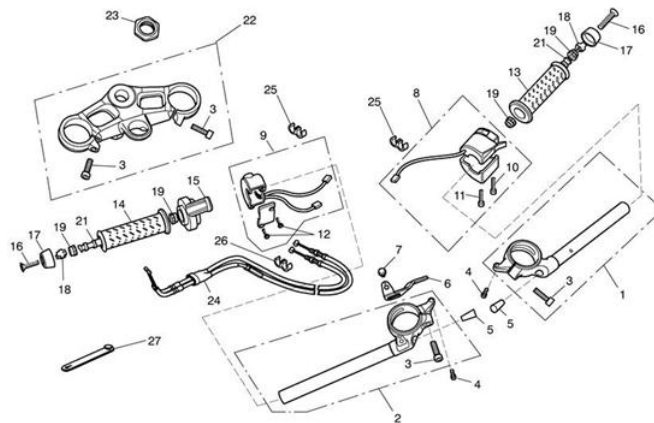


Figure 3 - Triumph Daytona 675 Handlebar & Switches Parts Diagram (World Of Triumph, 2019)

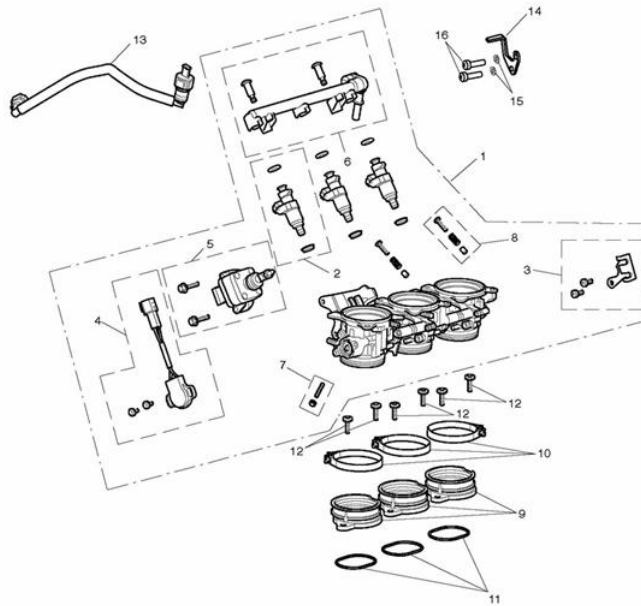


Figure 4 - Triumph Daytona 675 Throttles, Injectors & Fuel Rail Parts Diagram (World Of Triumph, 2019)

Figure 3 shows the conventional throttle (part 14), the twin cables (part 24) (pull cable and return cable) and in figure 4 the throttle body assembly which the opposite end of the cables connect to. The system operates on a cam pulley basis.



Figure 5 - Domino XM2 Throttle with Interchangeable Cams (Bike Torque Racing, 2019)

The throttle cam sizes determine the amount of turn in degrees required to go from 0 to 100% opening while the profile of the cam can shift the opening from linear to progressive to change the throttle input. As the throttle is turned and opened the pull cable rotates the opposing cam (while also compressing a return spring) on the throttle bodies in turn opening the butterfly valves to allow more air into the inlet port. This also rotates the sensor attached to the throttle bodies known as the throttle position sensor (TPS) (Figure 4, part 4) which sends a signal to the ECU informing it how much throttle the rider is requesting, the ECU cross references this value against engine RPM in a look up table to assess how much fuel is required to be injected into the port.

With RBW this link is broken, there is no longer a physical connection between the throttle and the throttle bodies. For RBW there are two popular throttle solutions, in one a traditional throttle and cables are still used but instead of being fed to the throttle bodies they are instead connected to a demand sensor.



Figure 6 - Demand Sensor (AF1 Racing, 2019)

Inside the demand sensor sits another cam on a return spring, because of this setup the amount of degrees that the throttle must be rotated to go from 0 to 100% opening can still be adjusted to suit the rider. The demand sensor acts as a potentiometer emitting two signals to the ECU, a system such as this cannot rely on just one signal alone hence the extra signal exists as verification and redundancy.

The other dismisses cables entirely containing everything within the throttle housing on the clip on with only the power, ground and signal lines protruding.



Figure 7 – Exploded View, KTM Ride by Wire Throttle (eBay, 2019)

The throttle bodies mostly remain unchanged from a conventional motorcycle apart from that instead of being cable operated it is now servo operated. The TPS is still retained at the side of the throttle bodies to verify the butterfly valve position.

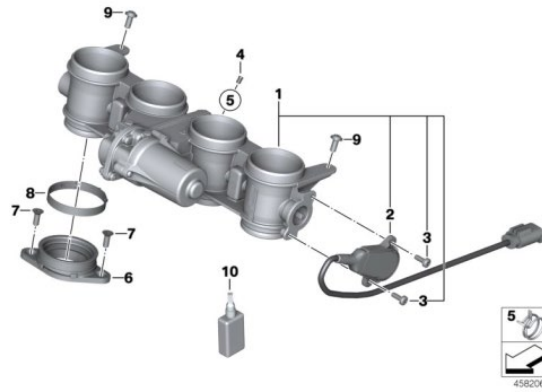


Figure 8 - 2017 BMW S1000RR Throttle Bodies Parts Diagram (Max BMW Motorcycles, 2019)

Because the ECU now has control over the butterfly valve actuation based on either inputs from the rider or sensors control strategies can now be applied.

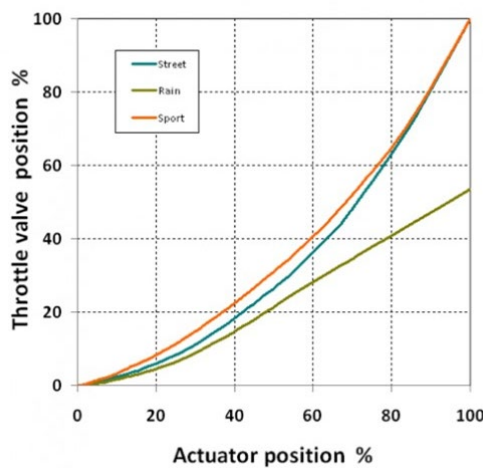


Figure 9 - 2014 KTM 1290 Super Duke R Throttle Maps (Motorcycle.com, 2013)

As seen in figure 9, in terms of percentage the output can differ more extremely with regards to the user input. Throttle maps can be tailored to given scenarios such as low grip (rain) where traction can be limited, here the initial opening of the butterfly valve is quite slow and will give a soft response as not to exceed the limits of grip. In this instance on the KTM 1290 due to opening the throttle valve just past 50% there is a limit to the maximum power available with this setting down to 130 hp from 177 hp. (KTM, 2019)

This is only a basic assessment due to more modern riding modes actively changing more parameters when moving between settings.

2.1.2 – Bosch MSC (Motorcycle Stability Control)

Often packages are not developed directly by the vehicle manufacturer but by specialist companies either independently or with manufacturer assistance. One such instance is Bosch who have developed their MSC platform and can now sell this product onto their

clients (Aprilia, BMW, KTM, etc). The benefit to the manufacturer is that it is easier and cheaper to purchase and fit a proven system for a vehicle line up rather than be impacted with increased time, research and development costs.

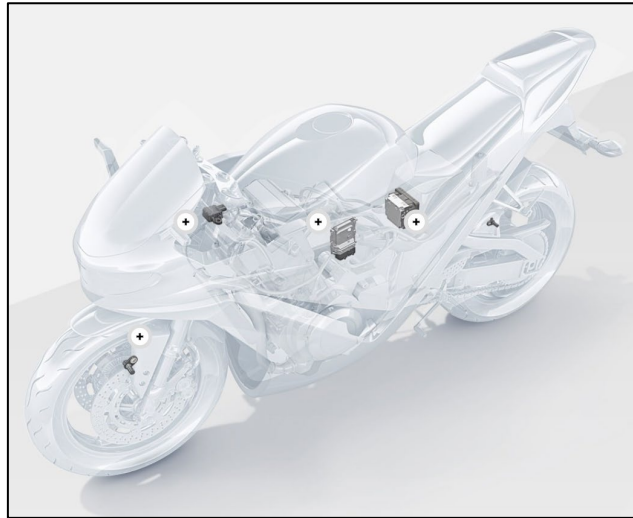


Figure 10 - Bosch MSC System Components (Bosch, 2019)

Bosch has developed MSC to assist the rider under acceleration and braking either upright or leant over in a corner. The system monitors and evaluates a number of important vehicle parameters such as torque, lean angle and acceleration and intervenes if required. (Bosch, 2019)

2.1.2.1 – Wheel Speed Sensor



Figure 11 - Wheel Speed Sensor (Bosch, 2019)



Figure 12 - ABS Equipped Motorcycle (Motorcycle.com, 2019)

Figure 12 shows the ABS ring positioned on the inside of the brake disc on the front wheel (there will also be another sensor and ring affixed to the rear wheel in a similar fashion), the sensor in a fixed position passes over the ring as the wheel rotates and generates a signal.

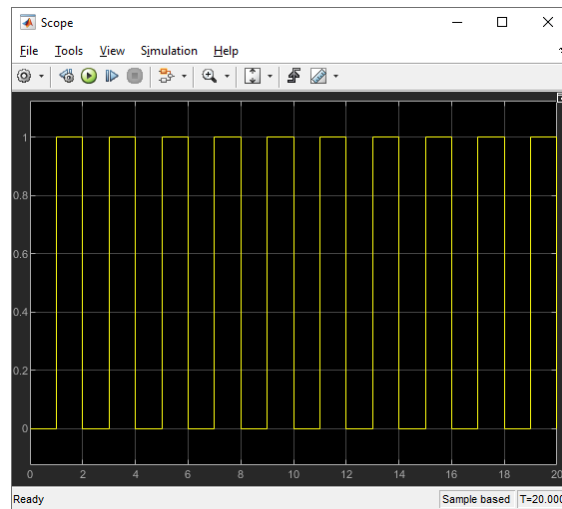


Figure 13 - Simulink Generated ABS Signal Example

Figure 13 demonstrates what an ABS signal would appear like, showing high and low voltage on the Y axis and time in seconds on the X axis. From this signal the ECU can calculate the velocity of each wheel, for example;

Tyre Circumference : 2 m

ABS Ring Trigger Points : 40

Each pulse would mean the wheel has turned 0.05 metres. Knowing this distance and then dividing it by the time between the pulses would give the velocity. The more trigger points that are present on the ABS ring then a higher resolution of data can be processed but a compromise has to be made as not to exceed the capabilities of the ECU.

In this instance if every time the signal goes high represents a pulse from a trigger on the ABS ring, should the signal stay low for a period of time this would indicate that the wheel has locked. With MSC, should it detect that the wheel is starting to lock the ECU can modulate the brake pressure to prevent the wheel from locking and stop the rider losing control.

In assessing the signals from the front and rear wheels and knowing individual wheel speeds the system can also engage traction control to regulate the amount of torque delivered to the rear wheel. (Bosch, 2019)

2.1.2.2 – IMU (Inertial Measurement Unit)

The IMU allows the reading of the vehicles acceleration and angular rate and can report to the ECU the dynamic status of the motorcycle.

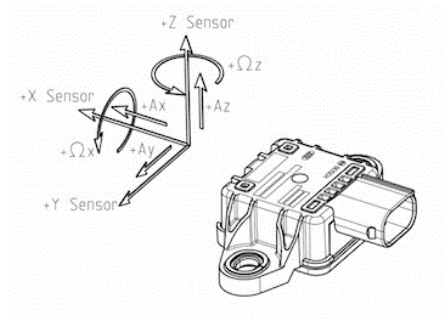


Figure 14 – IMU Sensor (Bosch, 2019)

The Bosch IMU uses a technology titled as MEMS (Microelectronic Mechanical Systems) consisting of accelerometers and gyroscopes.

The accelerometers work by using a mass suspended on a spring (seismic mass) with one for each direction (three axis; left-right, forward-backward and up-down). As the mass is known along with the spring rate by measuring how far the mass has moved from its static position the IMU can determine how fast the unit is accelerating in a given direction.



Figure 15 – IMU Measurements (BikesMedia, 2017)

The gyroscopes measure the angle of rotation up to three axis.

- Pitch, moving forwards or backwards
- Roll, leaning side to side
- Yaw, turning in relation to your initial direction

The MEMS IMU utilises a vibrating structure gyroscope with the theory that a mass which is vibrating will attempt to move in the same direction even when whatever it is attached to also moves. Similar to the seismic masses in the accelerometer this movement is measured to disclose the corresponding movement of the vehicle.

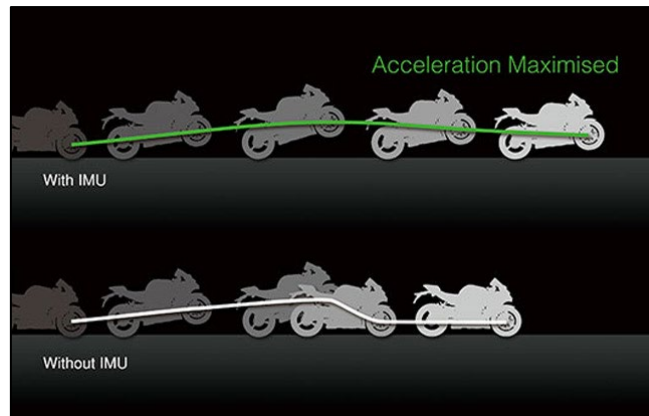


Figure 16 - Acceleration Graph of a Motorcycle With and Without IMU (BikesMedia, 2017)

The IMU can calculate these motions at over 100 times a second and allied with other sensor information allow the ECU to build a comprehensive picture of what the vehicle is doing. The following functions are can be achieved or enhanced with the correct usage of an IMU.

- Traction Control
- Wheelie Control
- Slide Control
- Launch Control
- Corner Sensitive ABS
- Suspension Damping Control

(Bennetts, 2018) (BikesMedia, 2017) (Bosch Mobility Solutions, 2016) (Bosch, 2019) (Visor Down, 2018)

2.2 – Communication Signals

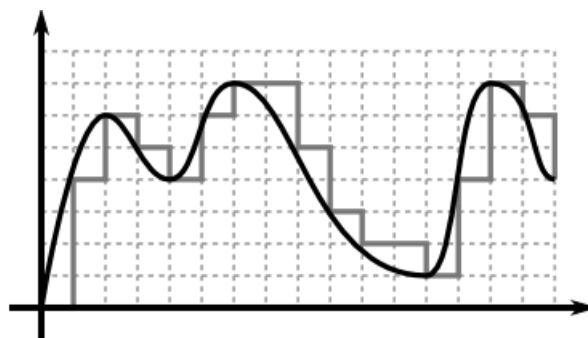


Figure 17 - Analog & Digital Signal Comparison (Chegg, 2019)

The graph in figure 17 demonstrates the analog signal pattern (represented as the curve) against a digital signal (represented as the discrete lines) with voltage (Y axis) against time (X axis).

2.2.1 – Analog

An analog signal can be defined as a continuous signal that has time varying quantities. In engineering the quantity that is time varying is typically the voltage and is subject to fluctuations. An analog signal can be used to measure change such as in a potentiometer. (SparkFun, 2019) (Chegg, 2019)

2.2.2 – Digital

Digital signals refer to a signal that is converted into a pattern of bits with a discrete value at each sampling point. Most commonly digital signals will be one of two values, 0 and 5V with timing graphs of these signal forming a square wave pattern. (SparkFun, 2019) (Chegg, 2019)

PWM (Pulse-Width Modulation) :

PWM is a type of digital signal and can be used in a variety of applications such as LED or servo control. As the signal can only be high (5V) or low (0V) at any given time, the proportion of time that the signal is in one of these states can be changed over a consistent time interval.

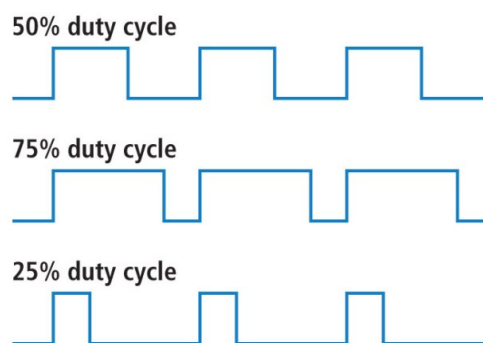


Figure 18 - 50%, 75% and 25% Duty Cycle Examples (SparkFun, 2019)

Figure 18 shows the concept as a duty cycle measured in percentage, if the signal is half on half off then it can be said that it has a duty cycle of 50% and resembles an ideal square wave. Whilst 100% would be the same as setting the voltage to a consistent 5V and 0% would be the same as grounding the signal. (SparkFun, 2019)

2.2.3 – Bus Interfaces

Various protocols and standards have been created to solve the problem of communication between devices.

2.2.3.1 – Serial

Many serial protocols have been created to meet particular requirements of the problems posed to them. USB and Ethernet are two well-known examples of this, SPI and I²C are also

examples of serial communication but these interfaces can be sorted into one of two groups. Synchronous and asynchronous.

SPI and I²C fall into the synchronous category as the data lines are paired with a clock signal so all devices share a common clock. This makes for a faster transfer of data but also requires at least one extra wire between communicating devices.

Asynchronous devices can transfer data without support from an external clock signal and helps minimising wires and I/O pins, but extra steps and rules are required when sending information back and forth to ensure data integrity.

- Data bits
- Synchronisation bits
- Parity bits
- Baud rate

The integral part of ensuring reliable communication is that both devices on a serial bus are configured to use the same protocols.

The baud rate determines how fast data is sent in units of bits per second (bps) and can be set to any value within reason although standard common rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200. In theory the rate could be set higher but if the limits of the hardware are exceeded then errors will occur as clocks and sampling periods cannot keep up.

A serial bus consists of two wires, RX, the receiver and TX, the transmitter. As opposed to other connections where either end is like for like, VCC to VCC or GND to GND the labels for serial are with respect to each device, thus RX from one device passes to TX and vice-versa.

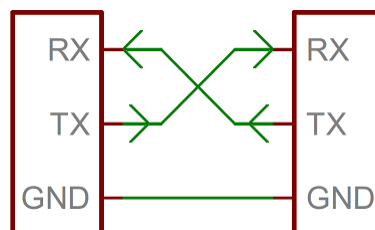


Figure 19 - Serial Connectivity (SparkFun, 2019)

The universal asynchronous receiver/transmitter (UART) is responsible for implementing serial communication, the UART creates the packets of data and controls the physical lines. When transmitting the UART creates the data packet and sends it down the TX line with timing set according to the specified baud rate. When receiving, the UART samples the RX

line at the specified baud rate, finds the synchronisation bits and then the data. If supported by the microcontroller, this data can then be passed onto a buffer where it stays until the microcontroller is ready to process it. UARTs will typically release this buffered data on a first in first out basis (FIFO).

Serial interfaces can be full duplex where devices can send and receive data simultaneously or half duplex where each device must take turns sending and receiving. (SparkFun, 2019)

2.2.3.2 – SPI (Serial Peripheral Interface)

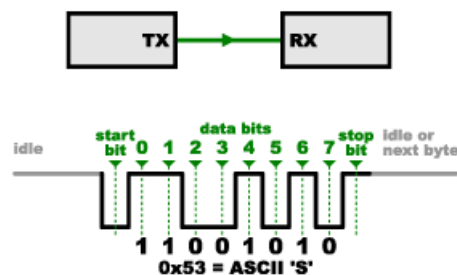


Figure 20 - Serial Asynchronous Data Transfer (SparkFun, 2019)

Figure 20 demonstrates the transfer sequence for a common serial communication and shows the extra overhead within each transfer as each device syncs at the beginning of each transmission compared to figure 21. If both devices are operating at different baud rates however the data will be compromised as the receiver is reading the data at specific times (the green arrows in figure 20).

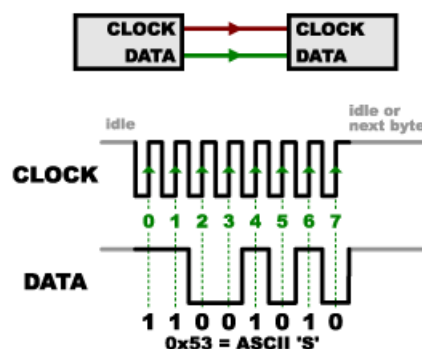


Figure 21 - SPI Synchronous Data Transfer (SparkFun, 2019)

SPI uses separate lines for data and a clock that keeps both sides in sync. The receiver is looking for the rising (low to high) or falling (high to low) edge of the clock signal and when it detects that edge it will look to the data line to read the next bit. Only one side generates the clock signal (CLK or SCK) and is referred to as the master which in the majority of cases will be the micro controller, anything on the other side will be known as the slave of which there can be multiple. When data is sent out from the master it is on a line known as the MOSI

(Master Out/Slave In) and if a response is required the slave will send data back on another line called MISO (Master In/Slave Out).

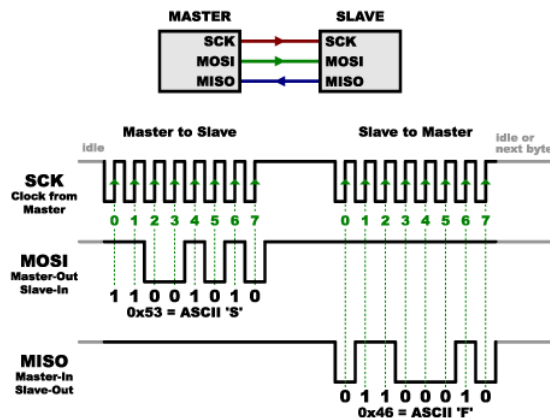


Figure 22 - SPI Master / Slave Interface (SparkFun, 2019)

In asynchronous serial, random amounts of data can be sent back and forth at any time, this is not possible with SPI but in practice does not usually present a problem as SPI is used to talk to sensors with a specific command structure.

SPI is also full duplex and can transmit and receive data at the same time, for example in requesting a new sensor reading while retrieving data from the previous reading. (SparkFun, 2019)

2.2.3.3 – I²C (Inter-Integrated Circuit)

I²C is designed to allow multiple slave devices to communicate with one or more master devices, similar to SPI it is intended for short distance communication and like asynchronous serial I²C only requires two signal wires to exchange data.

As asynchronous serial ports require no clock for every 8 bits of data sent there must be one start and one stop bit meaning that 10 bits are required for each transmission and when exchanging a lot of data eats into the data rate. Another issue is that while serial ports can support multiple devices, they are ideally suited for only two to communicate, any more than two and bus contention (where two devices try to send data down the same line at the same time) can become a persistent problem.

One downside to using SPI is the number of pins required, connecting a single master to a slave requires 4 lines and each additional slave requires an additional I/O pin on the master. When multiple sensors are needed or if PCB space is tight these extra lines are undesirable.

I²C however requires two lines similar to asynchronous serial but these two lines can support up to 1008 slave devices whilst allowing a multi-master system where more than one master

device can communicate with the slave devices over the bus, but it is limited as the master devices cannot communicate with one another and must take turns to use the bus lines.

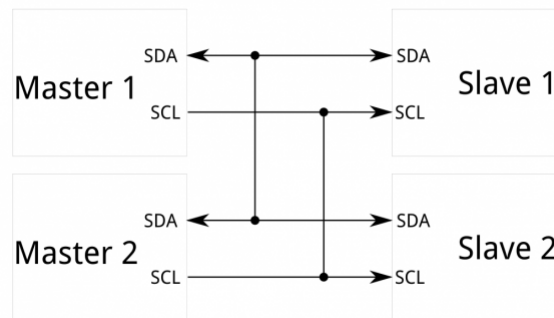


Figure 23 - I²C Master / Slave Interface (SparkFun, 2019)

Figure 23 shows the two lines for I²C, SCL is the clock signal which is always generated from the bus master and SDA which is the data signal. (SparkFun, 2019)

2.2.3.4 – TWI (Two Wire Interface)

Almost identical to I²C, the name was introduced by Atmel and other companies to avoid trademark issues. (I2C Bus, 2019)

2.2.3.5 – CAN (Controller Area Network)

Developed by Bosch and unveiled in 1986 with the first production vehicle (Mercedes-Benz W140) released in 1991 to use a CAN based wiring system. Since its introduction Bosch has revised CAN with CAN 2.0 published in 1991 being the latest although there have been amendments to extend the standard such as in 2012 with CAN FD 1.0 (CAN with Flexible Data-Rate) whilst maintaining compatibility with existing CAN devices. From 1993 CAN was adopted as an international standard, ISO 11898.

Pre CAN vehicles relied on complex point to point wiring, CAN was developed to simplify this and easily enable communication between multiple ECUs at a rate of 1Mbit/s. In an automotive application the vehicle could have as many as 70 for various subsystems, the engine control unit being the main ECU whilst others consist of the transmission, airbags, ABS and hybrid elements, the list is vast. Some form independent subsystems but communication amongst these is vital as systems control safety, economy and convenience features.

Examples of,

Stop/Start – Sensors (speed, steering angle, engine temperature, etc) around the vehicle are collated via the CAN bus to determine if the engine can be shut down when stationary to improve fuel economy and reduce emissions.

Collision Avoidance – Proximity data from parking sensors fed via CAN bus to driver assist systems to actuate brake by wire.

With the addition of more systems and increased functionality CAN 2.0 has limitations in the total load it can process at a rate of 1Mbit/s but has evolved with the addition of CAN FD 1.0. CAN FD provides two important improvements; data transmission is increased to 8Mbit/s and the data packet is increased to 64 bytes instead of 8, these relate to lower latency, higher bandwidth and increased real-time performance. (CSS Electronics, 2018) (Kvaser, October)

Looking back to CAN 2.0, it is split into two identifiers, CAN 2.0A and CAN 2.0B. CAN 2.0A is referred to as the standard format with an 11-bit identifier, while CAN 2.0B is the extended format with a 29-bit identifier. According to the CAN 2.0 Specification (Robert Bosch GmbH, 1991) both A and B follow the same basic concepts.

- Prioritisation of messages
- Guarantee of latency times
- Configuration flexibility
- Multicast reception with time synchronisation
- System wide data consistency
- Multimaster
- Error detection and signalling
- Automatic retransmission of corrupted messages as soon as the bus is idle
- Distinction between temporary errors and permanent failures of nodes and autonomous switching off of defect nodes

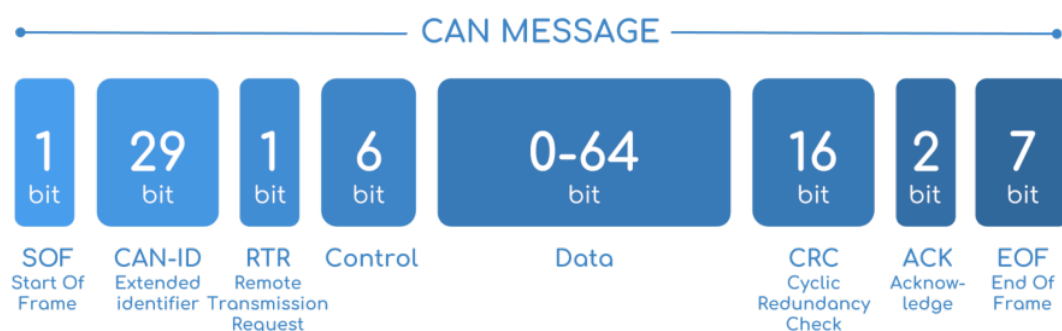


Figure 24 - CAN 2.0B Frame (CSS Electronics, 2018)

SOF : Start of Frame

The Start of Frame is dominant 0 to inform other ECUs that a message is coming and synchronise the nodes on a bus after being idle

CAN-ID : Extended Identifier	Contains the message identifier, lower values have higher priority
RTR : Remote Transmission Request	Allows ECUs to request messages from other ECUs
Control	Informs the length of Data in bytes
Data	Contains the actual data values
CRC : Cyclic Redundancy Check	Cyclic Redundancy Check is used to ensure data integrity
ACK : Acknowledge	Indicates if the CRC has been passed successfully
EOF : End of Frame	Marks the end of the CAN message

Table 3 - CAN Message Components (CSS Electronics, 2018) (Texas Instruments, 2016)

The CAN standard does not specify how to handle messages larger than 8 bytes, therefore a set of standards exist to further define how data is handled between devices, these include SAE J1939, CANOpen and OBD2.

2.3 – Development Microcontroller

There are various low-cost development boards available on the market that could suit the required application, but it is imperative to not only choose one that fulfils those requirements within a reasonable budget but also has a suitable support network. A popular development platform will bring with it other users experience and knowledge that can assist in trouble shooting thus speeding up development and avoiding common problems.

Whilst other solutions are available from the likes of Teensy, the two main options come from the Raspberry Pi Foundation and Arduino due to their cost, ease of access and knowledge base. Also, each has official support and integration from Mathworks with Simulink rather than relying on third party implementations. (MathWorks, 2019)

2.3.1 – Raspberry Pi

Raspberry Pi offers a variety of development platforms to suit various projects with their flagship currently being the Raspberry Pi 3 Model B+.

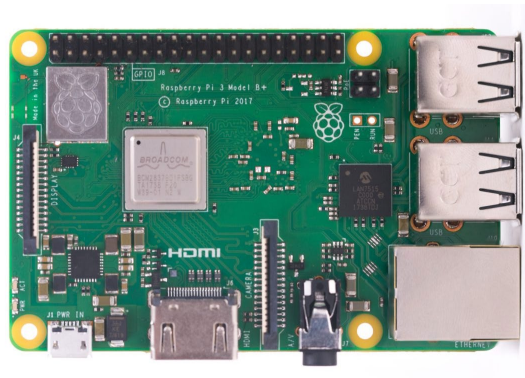


Figure 25 - Raspberry Pi 3 Model B+ (Raspberry Pi Foundation, 2019)

Processor :	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory :	1GB LPDDR2 SDRAM
Connectivity :	• 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE • Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) • 4 × USB 2.0 ports
Access :	Extended 40-pin GPIO header
Video & Sound :	• 1 × full size HDMI • MIPI DSI display port • MIPI CSI camera port • 4 pole stereo output and composite video port
Multimedia :	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD Card Support :	Micro SD format for loading operating system and data storage
Input Power :	• 5V/2.5A DC via micro USB connector • 5V DC via GPIO header • Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment :	Operating temperature, 0–50°C
Length :	85 mm
Width :	56 mm
Weight :	42 gms

Table 4 - Raspberry Pi 3 Model B+ Specifications (Raspberry Pi Foundation, 2019)

2.3.2 – Arduino

As with Raspberry Pi, Arduino also offers a variety of development boards depending on the scale and complexity of a project, these range from the diminutive Arduino Nano to the mid-range Arduino Uno and up the Arduino Mega 2560 Rev3.

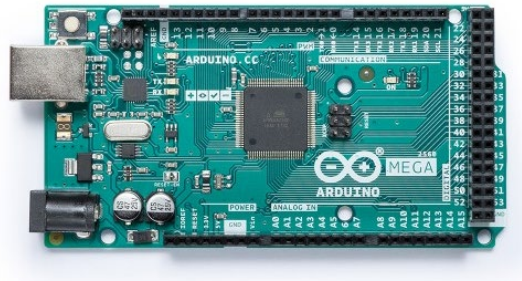


Figure 26 - Arduino Mega 2560 Rev3 (Arduino, 2019)

Microcontroller :	ATmega2560
Operating Voltage :	5V
Input Voltage (Recommended) :	7-12V
Input Voltage (Limit) :	6-20V
Digital I/O Pins :	54 (of which 15 provide PWM output)
Analog Input Pins :	16
DC Current per I/O Pin :	20 mA
DC Current for 3.3V Pin :	50 mA
Flash Memory :	256 KB of which 8 KB used by bootloader
SRAM :	8 KB
EEPROM :	4 KB
Clock Speed :	16 MHz
LED_BUILTIN :	13
Environment :	Operating temperature, 0–85°C
Length :	101.52 mm
Width :	53.3 mm
Weight :	37 gms

Table 5 - Arduino Mega 2560 Rev3 Specifications (Arduino, 2019) (Atmel, 2019)

Each of the 54 digital pins can be used as an input or output and operate at 5 volts. Each pin can provide or receive 20 mA and has an internal pull-up resistor. As like the Raspberry Pi certain pins have specialised functions.

- Serial
 - 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
- External Interrupts
 - 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level.
- PWM (pulse-width modulation)
 - 2 to 13 and 44 to 46. Provide 8-bit PWM output.
- SPI
 - 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).
- LED
 - 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- TWI
 - 20 (SDA) and 21 (SCL).

The Mega 2560 also consists of 16 analog inputs which provide 10 bits of resolution (1024 different values), by default these measure from ground (0V) up to 5 volts. (Arduino, 2019)

The Raspberry Pi is essentially a mini PC on a small PCB, it comes with a dedicated CPU, memory, GPU and visual outputs such as HDMI. The Arduino on the other hand is a fully-fledged micro controller rather than a PC as it lacks a full operating system that the Raspberry Pi requires and instead runs written code interpreted by its firmware. Due to this you lose functionality and tools that an operating system would provide but gain in performance as the code is executed directly with no operating system overhead.

Comparing the raw performance numbers the Raspberry Pi is vastly superior to the Arduino but it comes at the cost of complexity, it can do more but in this instance at the centre of this development platform it is not required. The simplicity of the Arduino is what sets it ahead for this application, with the attached sensors the code required is minimal with no interference from an operating system.

3.0 – Development

3.1 – Hardware

After comparison and selection of the Arduino Mega 2560 to be the base of the development platform the rest of the components can be assessed and selected.

3.1.1 – Arduino Shield

Shields are modular circuit boards that install into the Arduino and expand the base functionality of the controller. Various shields have been developed to tailor the Arduino controller to specific tasks such as wireless functionality, displays, motor drivers and so forth.

As versatile as the Arduino Mega is, it and other micro controllers suffer limitations on the board with the availability of VCC and GND pins which can make it difficult to connect other devices and sensors that require power.

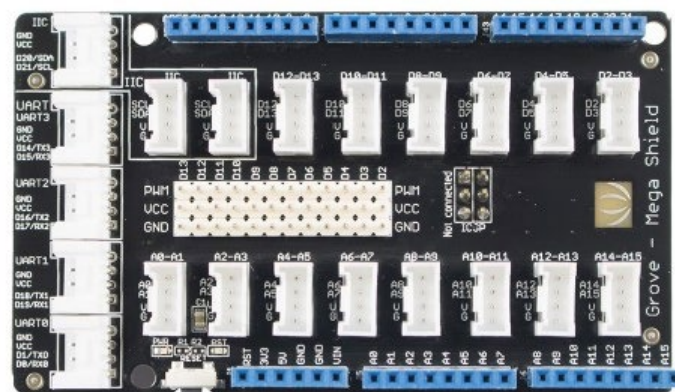


Figure 29 - Grove - Mega Shield V1.2 (Arduino, 2019)

On the base Arduino Mega 2560 there are only 3 5V and 5 GND pins available, with the addition of the Mega Shield V1.2 this is expanded upon to 15 5V and 18 GND pins with no loss of functionality.

3.1.2 – Throttle

Rather than attempt to implement a new design the decision was made to reverse engineer an existing component available in production to save time and reduce cost, the chosen part is from a 2018 KTM 1290 Super Duke R.

The throttle has six lines, 2x VCC (5V), 2 x GND and 2 x signal.

Signal 1 :

White w/red trace – Power

White w/black trace – Ground

Green w/black trace – Signal

Signal 2 :

Yellow w/red trace – Power

Yellow w/black trace – Ground

Green w/yellow trace – Signal

To be able to use this with the Arduino, the voltage for each signal line needs to be determined. Using the Arduino to power the throttle, a multi meter can be used to probe the ground and signal lines with readings being taken between 0 and 100% throttle opening.

Signal 1 – 1V to 4.4V

Signal 2 – 0.5V to 2.2V

The signal lines can therefore be connected to analog inputs on the Arduino where the voltage will be converted to a digital value between 0 and 1023 (10 bits of resolution) via the analog to digital converter (ADC). (Arduino, 2019)

3.1.3 – Servo Motor

For development a basic servo motor in the SG-90 can be used to replicate the principals of the output for a ride by wire system.



Figure 30 - Servo Motor SG-90 (Components101, 2017)

The servo has three lines, 1x VCC (5V), 1 x GND and 1 x signal.

Red – Power

Brown – Ground

Amber – Control Signal

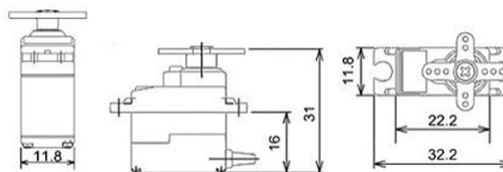


Figure 31 - Servo Motor SG-90 Dimensions (Components101, 2017)

Operating Voltage :	5V
Torque :	2.5 kg.cm
Operating Speed :	0.1s / 60°
Gear Type :	Plastic
Rotation :	0° - 180°
Weight :	9 gms

Table 6 - Servo Motor SG-90 Specifications (Components101, 2017)

The 2.5kg.cm torque of the SG-90 allows the motor to pull a load of 2.5kg at a distance of 1cm, this can be altered so that a 5kg load could be pulled at 0.5cm or a 1.25kg load at 2cm.

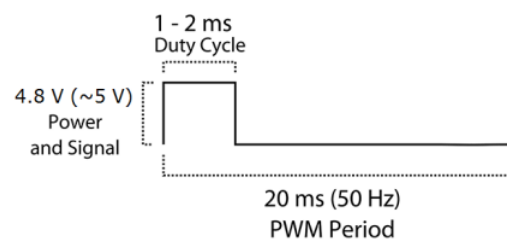


Figure 32 - Servo Motor SG-90 PWM (Components101, 2017)

As the servo can rotate 180° it has three distinct positions, 'position 0' (1.5ms pulse) which positions the arm in the middle, 'position 90' (2ms pulse) which positions the arm 90° clockwise from 'position 0' and 'position -90' which positions the arm 90° counter clockwise from 'position 0'.

3.1.4 – Inductive Proximity Sensor

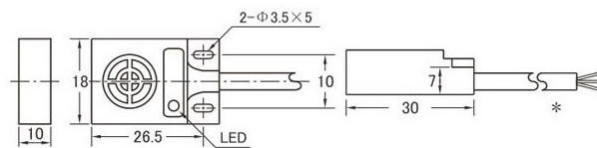


Figure 33 - GKB-M0524NA Inductive Proximity Sensor Dimensions (Hugong Automation, 2019)

Detection Distance :	5 mm $\pm$ 10%
Detection Object :	Magnetic metal (detection distance decreases when non-magnetic)
Response Frequency :	500 Hz
Supply Voltage :	5V
Temperature Range :	-25–60°C
Protection Level :	IP67

Table 7 - GKB-M0524NA Specifications (Hugong Automation, 2019)

As the chosen development vehicle does not have the facility to log individual wheel velocities a solution must be derived. In using an inductive proximity sensor, the heads of the brake disc bolts can be used as a trigger to create a signal that can be counted and knowing the circumference of the wheel with the tyre the velocity can be calculated.

The 2010 Triumph Daytona 675 uses five brake disc bolts on the front wheel and four on the rear, to specify a suitable sensor the amount of pulses per second needs to be calculated when the vehicle is at its theoretical maximum velocity.

Primary Gear – Driven	Primary Gear – Driver	Primary Gear Ratio (Driven/Driver) GR1
85	46	1.848

Gear	Driven	Driver	Gear Ratio (Driven/Driver) GR2
6 th	25	22	1.136

Front Sprocket (Teeth)	Rear Sprocket (Teeth)	Sprocket Ratio (Rear/Front) GR3
16	47	2.938

Overall Gear Ratio – 6th Gear = $GR1 \times GR2 \times GR3$

Overall Gear Ratio – 6th Gear = $1.848 \times 1.136 \times 2.938$

Overall Gear Ratio – 6th Gear = **6.168**

Rear Tyre Width (mm)	Height (%)	Rim Diameter (Inches)
180	55	17

$$\text{Tyre Circumference} = \left(\left(\frac{\text{Rim Diameter (Inches)}}{0.03937} \right) + 2(\text{Tyre Width (mm)} \times \text{Tyre Height (\%)}) \right) \times \pi$$

$$\text{Tyre Circumference} = \left(\left(\frac{17}{0.03937} \right) + 2(180 \times 55\%) \right) \times \pi$$

Tyre Circumference = **1.98 metres**

Max Power RPM	12,600
---------------	--------

$$\text{Theoretical Top Speed} = \frac{\text{Tyre Circumference (m)} \times 60 \times \text{Max Power RPM}}{1000 \times \text{Overall Gear Ratio}}$$

$$\text{Theoretical Top Speed} = \frac{1.98 \times 60 \times 12600}{1000 \times 6.168}$$

Theoretical Top Speed = **242.5 km/h**

Divide by 3.6 to convert km/h to m/s

$$\frac{242.5}{3.6} = \mathbf{67.36 \text{ m/s}}$$

Each rotation is equal to 1.98 metres, divide by 1.98 to calculate wheel revolutions per second

$$\frac{67.36}{1.98} = \mathbf{34.02 \text{ revs/s}}$$

Rear Wheel Pulses Per Second = 34.02×4

Rear Wheel Pulses Per Second = **136 Hz**

In standard settings from the factory the sensor more than complies with the requirements needed for this application with plenty of overhead should the tyre size or gearing be altered.

The inductive proximity sensor has three lines, 1x VCC (5V), 1 x GND and 1 x signal.

Brown – Power

Blue – Ground

Black – Signal

3.1.5 – IMU (Inertial Measurement Unit)

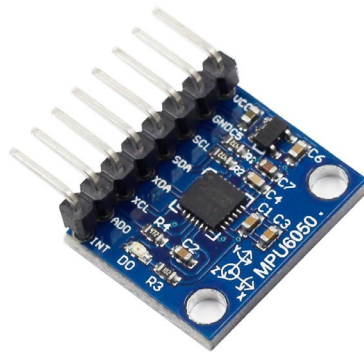


Figure 34 - MPU6050 IMU (Amazon, 2019)

Supporting six degrees of freedom the MPU6050 IMU is a well support IMU with Arduino development in mind. Combining a 3 axis gyroscope and 3 axis accelerometer with a DMP (digital motion processor) the MPU6050 is capable of processing the raw data captured and outputting the angles for yaw, pitch and roll reducing the calculating load on the corresponding controller. (TDK InvenSense, 2019) (TDK InvenSense, 2013)

3.1.6 – LCD Display

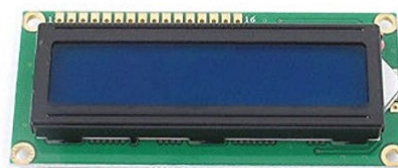


Figure 35 - HD44780 16x2 LCD (Amazon, 2019)

Even though relatively simple in providing 2 16-character lines, the HD44780 was specified due to the cost, documentation and available libraries to support the timescale of development that otherwise wouldn't have been possible with another unit. (Hitachi, 1998)

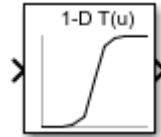
3.2 – Simulink

Developed by MathWorks, Simulink offers a graphical programming environment for modelling, simulating and analysing dynamic systems.

3.2.1 – Simulink Blocks

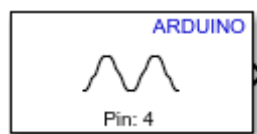
Summary of the Simulink blocks used within this model.

1-D Lookup Table :



Perform n-dimensional interpolated table lookup using index searches.

Arduino Analog Input :



Measures the voltage of the specified pin and outputs a digital value, the voltage is measured as 10-bit value ranging between 0 and 1023. An output of 0 indicates that the specified pin equals the ground voltage.

Arduino Digital Output :



Outputs the input signal on the specified digital pin.

Arduino Standard Servo Write :



Set the shaft position of the servo motor, for 0 input the shaft is set to 0 degrees.

Compare To Constant :



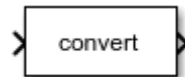
Determine how a signal compares to a constant.

Constant :



Outputs the specified constant value.

Data Type Conversion :



Convert the input to the data type and scaling of the output.

Name	Description
double	Double-precision floating point
single	Single-precision floating point
int8	Signed 8-bit integer
uint8	Unsigned 8-bit integer
int16	Signed 16-bit integer
uint16	Unsigned 16-bit integer
int32	Signed 32-bit integer
uint32	Unsigned 32-bit integer
int64	Signed 64-bit integer
uint64	Unsigned 64-bit integer
string	Text

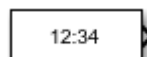
Table 8 - MATLAB Data Types (MathWorks, 2019)

Delay :



Delay the input signal by a specified amount of samples.

Digital Clock :



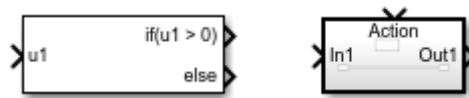
Outputs current simulation time at the specified sample rate.

Gain :



Element-wise gain.

If and If Action Subsystem :



IF expression, run the Action Subsystem connected to the 1st output port.

ELSEIF expression, run the Action Subsystem connected to the 2nd output port.

The number of ELSEIF output ports in the block is equal to the number of comma separated ELSEIF expressions entered into the dialog box.

ELSE, run the Action Subsystem connected to the last output port (can be enabled in the If block).

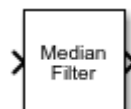
The output of the IF block must be connected to an If Action Subsystem.

Increment Stored Integer :



Increases the stored value of the input signal by 1.

Median Filter :



Calculates the median value over time using a sliding window, the window length is specified by the user.

Merge :



Merge the user defined number of input signals into a single output. If two signals attempt to pass through the simulation will fail.

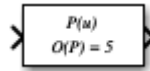
On Delay :



Apply a delay on the input signal when the input changes to True (1) or False (0).

When the input become True, the output becomes True after the specified time delay as long as the input is still True. If the input becomes False before the time delay has expired then the output stays False.

Polynomial :



Polynomial evaluation, calculates P given by the polynomial coefficient.

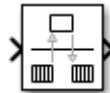
Pulse Generator :



Outputs a signal based on user defined parameters.

- Amplitude
- Period (number of samples)
- Pulse Width (number of samples)
- Phase Delay (number of samples)
- Sample Time

Rate Transition :



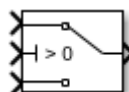
Amend the sample rate of a signal.

Scope :



Displays input signals in relation to simulation time.

Switch :



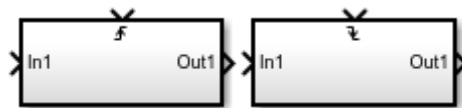
Pass through input 1 when input 2 satisfies the selected criteria, otherwise pass through input 3.

Criteria for passing first input.

- Input 2 $\geq$ Threshold
- Input 2 $>$ Threshold
- Input 2 $\sim$ Threshold

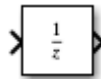
Threshold being a user defined value.

Triggered Subsystem :



Subsystems enabled by either a rising or falling signal.

Unit Delay :



Sample and hold the signal with a one sample period delay.

3.2.2 – Model

See Appendix A for Simulink model diagram.

At the time of writing the model is made up of

- Throttle Subsystem
- Warm-Up Subsystem
- LCD Subsystem
- Wheel Speed Subsystem
- Servo Output

3.2.3 – Throttle Subsystem

See Appendix B for Simulink model diagrams.

The throttle model development started as a simple model of a rotary potentiometer, Arduino and servo.

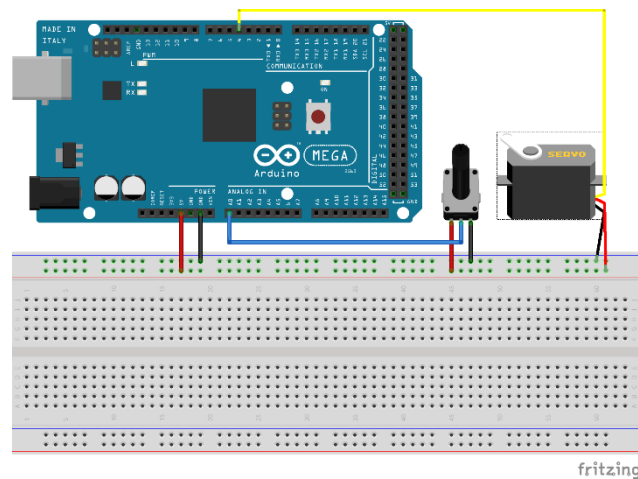


Figure 36 - Initial Throttle Wiring Diagram

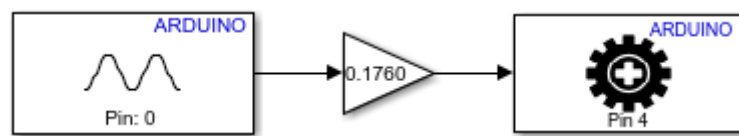


Figure 37 - Initial Simulink Model

The sample time is set within the Arduino Analog Input block (set to 0.01), this determines when during a simulation the block produces an output and updates its internal state. Unless a Rate Transition block is introduced then the sample rate will remain consistent until the signal is either terminated or reaches an output. (MathWorks, 2019)

The gain value is set to 0.1760, this is calculated by taking the maximum servo motor displacement in degrees and dividing it by the analog input resolution.

$$\text{Gain} = \frac{180}{1023}$$

$$\text{Gain} = \mathbf{0.1760}$$

At this point Simulink can be configured to compile and deploy the model to the Arduino board.

By selecting, *Tools > Run On Target Hardware > Prepare To Run*

The *Configuration Parameters* menu appears and under *Hardware Implementation* a dropdown menu titled *Hardware Board* is present and the Arduino Mega 2560 can be selected. Click *OK* and the menu is closed with the choices saved, *Deploy To Hardware* can be selected from the toolbar and Simulink will compile the model to run on the intended hardware. (MathWorks, 2019)

The next step is to remove the rotary potentiometer and interface with the KTM 1290 throttle. From the measurements previously taken calculations can be done to approximate what the expected ADC readings should be for each signal.

$$\text{Signal (1V)} = \frac{ADC}{\text{Voltage}}$$

$$\text{Signal (1V)} = \frac{1024}{5}$$

$$\text{Signal (1V)} = \mathbf{204.8}$$

Approximations of each signal;

Signal 1 :

0% Opening – 204.8

100% Opening – 901.1

Signal 2 :

0% Opening – 102.4

100% Opening – 450.56

Connecting the signal lines of throttle to the Arduino (Signal 1 to Pin 0, Signal 2 to Pin 2) and setting up a new model within Simulink the Scope function can be used to read the signal outputs. In this instance the Scope was set to read over 10 seconds.

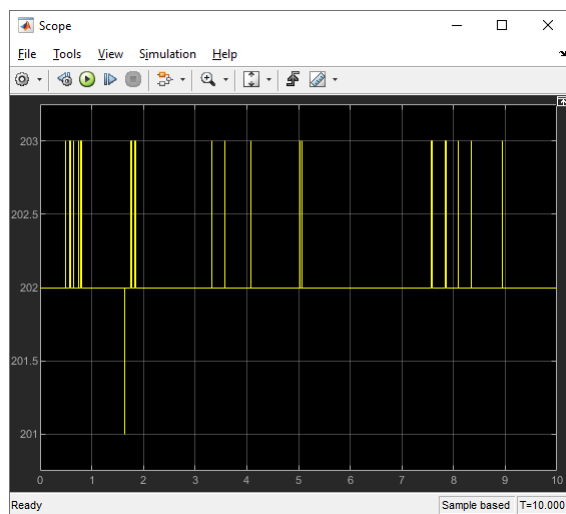


Figure 38 - Throttle Signal 1, 0% Opening

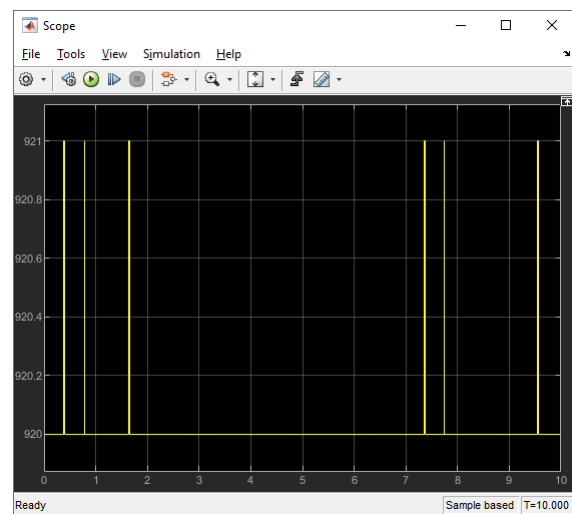


Figure 39 - Throttle Signal 1, 100% Opening

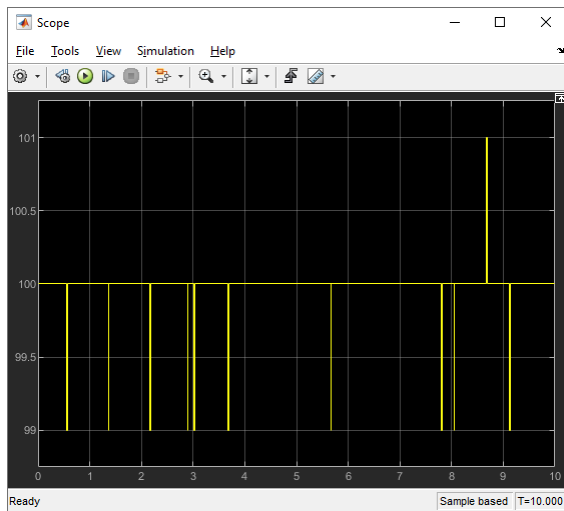


Figure 40 - Throttle Signal 2, 0% Opening

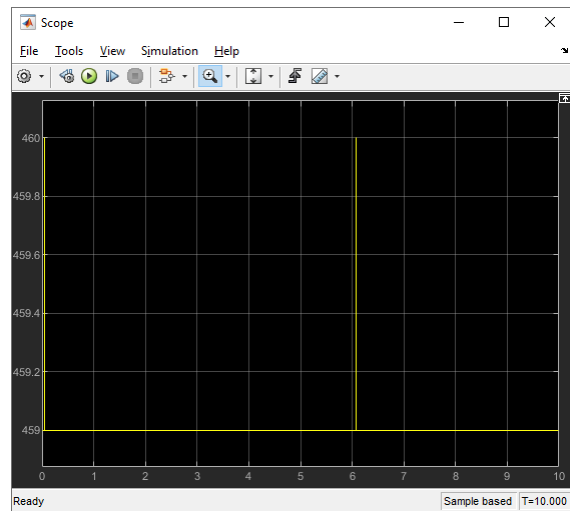


Figure 41 - Throttle Signal 2, 100% Opening

As shown in figures 38 through 41 each signal exhibits noise and does not retain a constant value. To combat this noise a software Median Filter can be added to analyse the signal and calculate an average over a sliding 10 sample window.

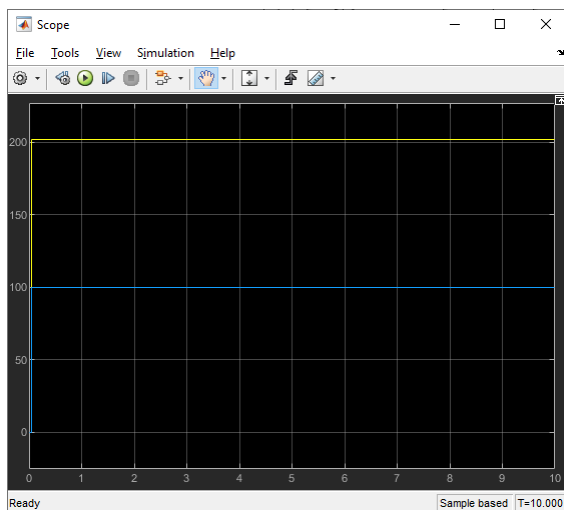


Figure 42 - Signal 1 (Yellow) & Signal 2 (Blue) At 0% Opening With Median Filter

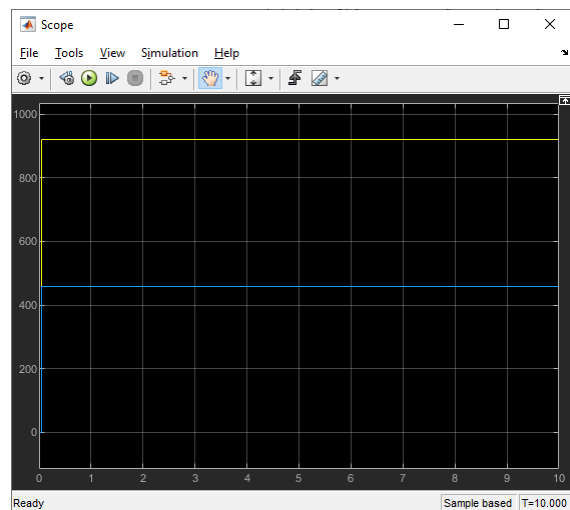


Figure 43 - Signal 1 (Yellow) & Signal 2 (Blue) At 100% Opening With Median Filter

With the filter in place;

Signal 1 :

0% Opening – 202

100% Opening – 920

Signal 2 :

0% Opening – 100

100% Opening - 459

For this test platform signal 1 will be passed through to control the output, signal 2 will be used as a backup line and the two together can form a control signal.

Each signal is to pass through a 1-D Lookup Table to amend its value as in the current form a value greater than zero will be passed onto the servo and set an undesired angle, at 0% throttle opening the servo is required to be in the zero position.

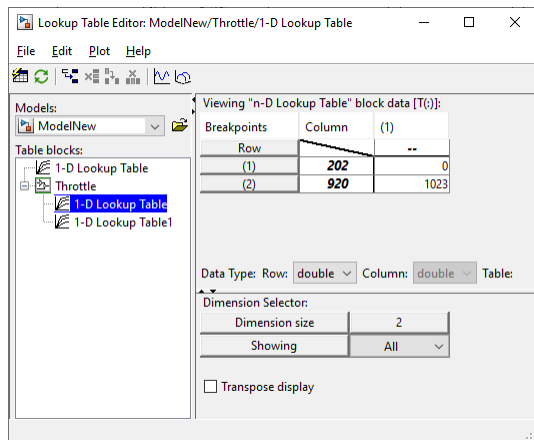


Figure 44 - Signal 1 Lookup Values

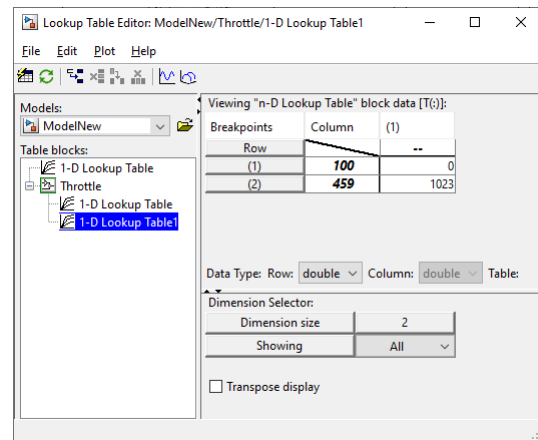


Figure 45 - Signal 2 Lookup Values

As the 1-D Lookup Table function supports interpolation only the start and end values are needed for each signal.

Because of the discrepancies between expected and real signal values tests were conducted to evaluate the accuracy of the Gain value from the initial model. According to the MathWorks documentation to move the servo arm 90° the Gain value would be equal to 90 over 1023 which is 0.088 but this proved to be inaccurate. Through further testing a value of 0.105 was found to be accurate.

In order to increase functionality a latching switch has been added to allow the user to select between two throttle maps. In the closed position the switch signal is sent to ground and reads zero whilst in the open position the signal is not constant but floating and reads greater than zero. For Map 1 the signal passes through the Gain block with the predefined value of 0.105, this retain a linear 1:1 relationship that a rider would experience with a conventional throttle but for Map 2 a polynomial is introduced. By introducing a polynomial the throttle map no longer has to be linear, this can now lead to fully customisable throttle maps to suit a lower power mode or to soften the initial roll on of the throttle.

1:1		New Profile				
Original	Lookup		Throttle %	Signal	%	Modified
202	0		0	0	0	0
203	1.425		0.139	1.425	50	0.712
204	2.850		0.279	2.850	50	1.425

Figure 46 - Excel Throttle Profile Calculator

A throttle profile calculator can be assembled in Excel to visualise the new throttle curvature and also attain the polynomial expression for the Polynomial block. Under 1:1 – Original,

values run down the column from 202 to 920 to represent the 0-100% opening of signal 1 and under 1:1 – Lookup, a formula is used to calculate the amended value from the 1-D Lookup Table. A constant is established by using the upper ADC value of 1023 (minus 1 because of zero) and dividing by 718 which is the amount of data points for signal 1.

1:1		New Profile				
Original	Lookup		Throttle %	Signal	%	Modified
918	1020.150		99.721	1020.150	100	1020.150
919	1021.575		99.861	1021.575	100	1021.575
920	1023.000		100.000	1023.000	100	1023.000

Figure 47 - Excel Throttle Profile Calculator

The constant is then multiplied by an ascending value in related to that entries position (0,1,2,3 up to 718) to give the final Lookup value.

For a new profile the original throttle position is calculated by dividing 100 by 1023 and multiplying by the signal value.

The % column is populated via a drop-down box using conditional formatting to allow the user to select a percentage between 0 and 200.

Finally in the Modified column an IF statement is used to calculate the new value.

`=IF(SUM(F4%*E4)>=1023,1023,SUM(F4%*E4))`

The new values can now be plotted onto a graph and using the trendline function of Excel a polynomial fit can be assessed. The polynomial order may need to be evaluated to determine the best fit for the curvature, if limitations are reached within Excel the data may be exported into the curve fitment tool of MATLAB.

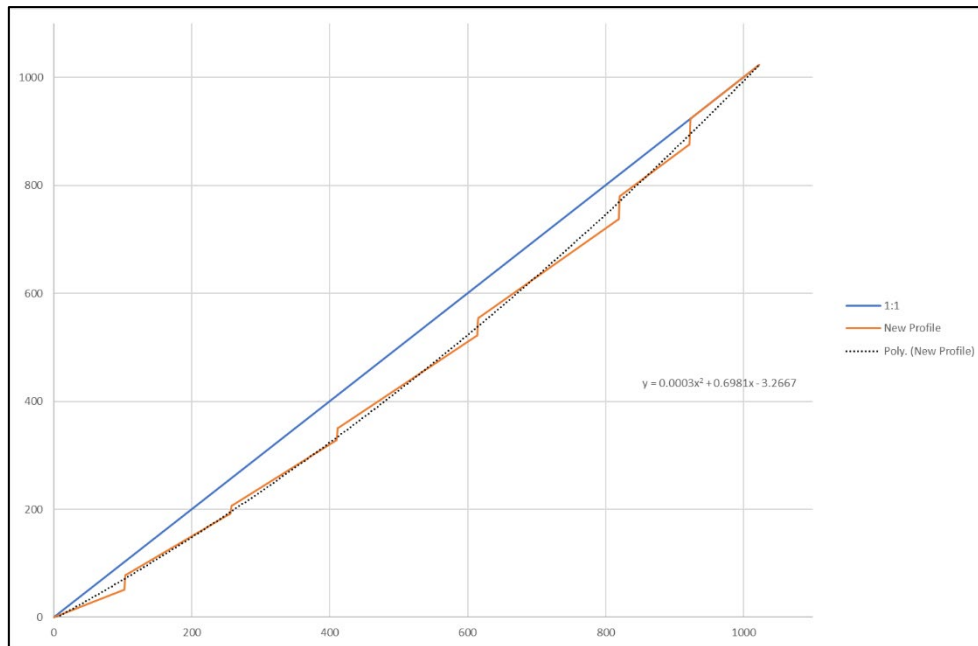


Figure 48 - Throttle Profile Calculator Graph

Once the desired expression has been attained it can be entered into the Polynomial block.

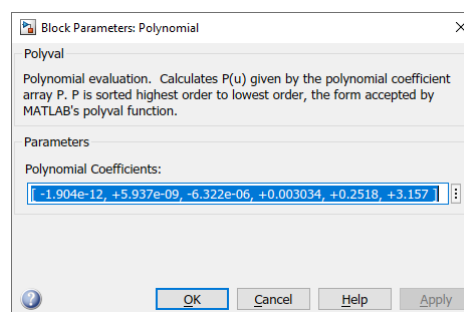


Figure 49 - Polynomial Block

While the input signal can be sent through to get a desired response from the servo safety and redundancy has to be thought of. A signal trace has been taken from the two input signals with signal 1 being divided by signal 2 to generate a control signal, at 0% opening the signal will read 2.02 and at 100% will read 2.00 so values around 2 will be considered within tolerance. It is then sent through a Median Filter to eliminate any potential anomalies that may occur and a Rate Transition block so that the operation is only carried out at 5 Hz to lessen the processing load on the controller before being split between the LCD subsystem and the LED subsystem.

The same signals are also passed through an IF block to determine if there is complete throttle failure rather than just one signal failing before passing through a Rate Transition block (5 Hz) and into the LCD and LED subsystems.

The LED subsystem exists as a visual warning alert system to gain the attention of the user that there is an issue with the throttle signals. The control signal is evaluated and if found to be under at 1.9 or over at 2.1 then using the Pulse Generator block a signal can be sent to the appropriate LED. This signal leads into an IF Action subsystem to also allow the signal to pass in the event of a complete throttle signal failure.

A subsequent set of IF and Switch blocks are set to evaluate signal 1 and 2 along with the control signal to ensure the relevant signal passes through in the event of failure.

The final signal output is passed out to the Servo Output subsystem while a trace of signal 1 is passed to the LCD subsystem.

3.2.4 – LCD Subsystem

See Appendix C for Simulink model diagram.

The LCD subsystem was built using the Arduino Drivers library. (RaphvK, 2018)

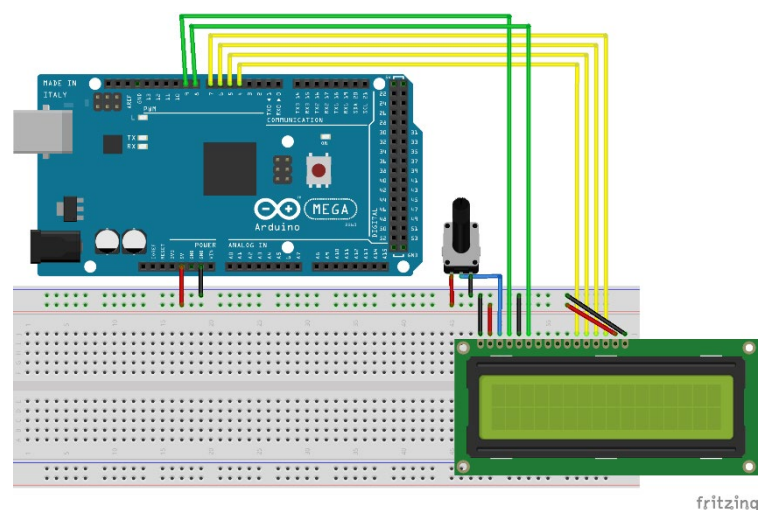


Figure 50 – LCD Wiring Diagram

Because of the throttle map selection and for safety considerations an LCD has been added, with this feature more information can be transmitted to the user

The LCD subsystem is setup to show an initial splash screen message to the user when the system first boots before defaulting to showing the throttle position in terms of percentage on the top line and which throttle map is selected on the bottom line.

If there is a failure to either signal line or a complete failure this is communicated to the user.

While the warm-up procedure is active this is also shown on the bottom line.

All signals into the subsystem have been set to 5 Hz to lessen the processing load on the controller.

3.2.5 – Warm-Up Subsystem

See Appendix D for Simulink model diagrams.

A user selectable function that can be enabled to automatically blip the servo a set amount of times at regular intervals when the control criteria is met. The criteria being that;

- The vehicle is in neutral
- The throttle is at 0% opening

The Warm-Up subsystem has three input signals from a momentary switch, gear position sensor and lastly the throttle output signal from the Throttle subsystem. The initial check is to ensure that the vehicle is in neutral before passing the input signal forwards to the On Delay and Compare To Constant blocks, here there is a hold of three seconds as a safety feature to make sure that the system cannot be enabled with a short press of the switch.

If the checks are passed then the signal generating subsystem is enabled, a Pulse Generator block becomes active generating a sine wave, the signal then triggers a counter which limits the total amount of waves to 100 if the system is not interrupted by a change of gear or the throttle signal.

3.2.6 – Wheel Speed Subsystem

See Appendix E for Simulink model diagrams.

The signal from the inductive proximity sensor is normally high and when triggered drops low as such the initial subsystem is set as a Falling Triggered Subsystem. A counter is enabled and emits a high signal every 4 pulses before returning low, this triggers a Rising Triggered Subsystem which passes through a time signal that splits into two. One passes uninterrupted into a Sum block, the other passes through a Unit Delay of one sample before joining at the Sum block. The signal then rises high from low as the time taken is calculated which then triggers the last Rising Triggered Subsystem that passes through the time signal. From here conversion calculations are carried out to determine RPM.

The first, distance over time where distance is the circumference of the tyre to attain the velocity.

The second, velocity multiplied by a constant of 16.667 to convert the velocity into metres/min.

The final divides this by the distance travelled per revolution to calculate the RPM of the wheel.

Using the theoretical top speed of 242.5 km/h.

$$242.5 \times 16.667 = 4041.6667 \text{ metres/min}$$

Divide 4041.6667 by the distance travelled per rotation, 1.98m.

$$\frac{4041.6667}{1.98} = 2041.2458$$

Multiply by the 4 pulses per rotation.

$$2041.2458 \times 4 = 8164.9832$$

Divide by 60

$$\frac{8164.9832}{60} = \mathbf{136 \text{ Hz}}$$

This corresponds and supports the calculations from 3.1.4.

3.2.7 – Servo Output Subsystem

See Appendix F for Simulink model diagram.

A simple passthrough subsystem, if the throttle signal exceeds 0.5 then it takes priority. If not then the signal from the Warm-Up subsystem passes which when not engaged and active is zero.

4.0 – Analysis & Further Work

4.1 – Simulink Model

The current model is still very much in its infancy in terms of development and requires further enhancement and refinement. Each system has been built independently and then integrated, this has created some overlap and needs to be addressed as with more features and systems added this will only get worse in time.

The current model is not without limitations.

- No TPS feedback to verify servo position
- Software Median Filter (introduces a small latency within the system)
- No error code reporting structure
- Wheel speed does not consider lean angle variable (tyre circumference decreases with increased lean angle)
- Warm-Up subsystem has no coolant temperature input
- LCD driven direct, could be updated to I²C and reduce wires from 12 to 4

As development proceeds these limitations would continue to be identified and phased out, some though such as the TPS could only be done via vehicle integration during real world testing.

The progress shown so far is promising but highlights the amount of time required for software development in designing, programming and testing to ensure that standards are met. It's often easy to make a system work but becomes rather difficult when attempting to make a system work safely. Testing even in the early stages is necessary to guarantee initial functionality but also intentionally breaking objects to understand how and what happens when a failure occurs and then how to react appropriately is vital. In this initial phase one particular issue persisted, when either signal 1 or 2 of the throttle would be disconnected to simulate a failure the signal would not return to zero but instead end up being open and float picking up interference from any nearby signals on the board.

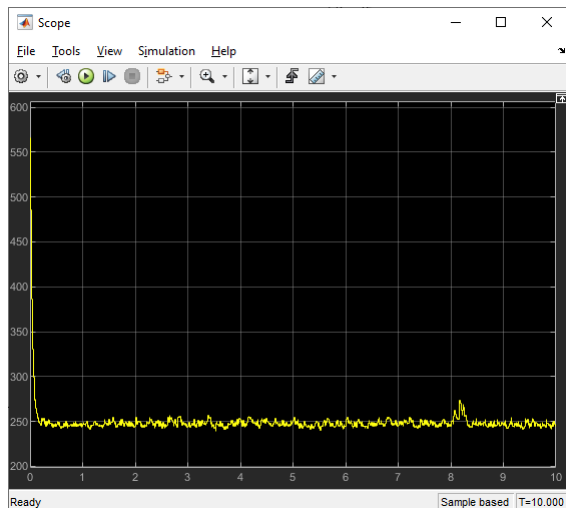


Figure 51 - Throttle Signal 1 Floating Without Pull-Down Resistor

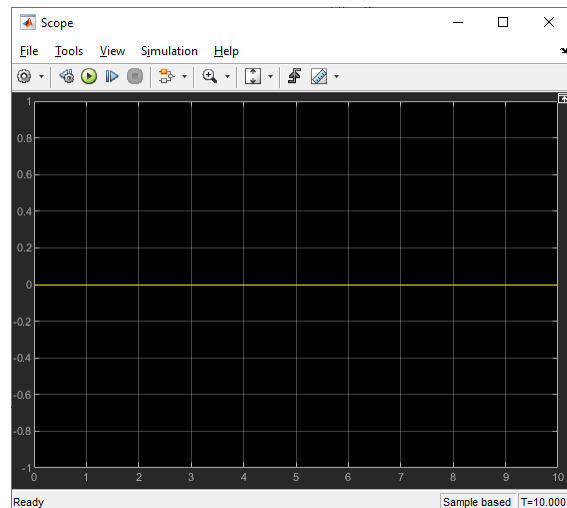


Figure 52 - Throttle Signal 1 Floating With Pull-Down Resistor

The solution was to introduce a pull-down resistor, the Arduino supports pull up resistors as a standard feature but a 10k Ω was attached to the pin with a wire to ground. For both signal 1 and signal 2 this resolved the floating issue.

While Simulink generates the machine code required from the model it would still be wise to evaluate the code after deployment.

4.2 – IMU

Simulink is a powerful tool to manage this kind of project in generating control strategies and the block format does come with strengths and weaknesses. Pre-configured blocks can speed up development if you're aware of what is required and if there is an applicable block available with the required settings. It's unreasonable to expect that every scenario can be covered.

Unfortunately this issue befell the inclusion of the IMU within the model, an applicable block does not exist for the MPU6050 within Simulink that can process the values from the DMP (raw values were available). Using the S-Function builder in Simulink the Arduino library could be successfully imported and the desired values extracted from the DMP using the Scope block but through various attempts for reasons I am unable to explain it was not possible to use these signals within the model. Due to time constraints this feature was postponed in order to proceed in other areas of development.

4.3 – Further Work

See Appendix G for Simulink model diagram.

The model shown in Appendix G is the current development model with each subsystem undergoing extensive refinement to support a proper error code management system, throttle signal management and gear shift assist.

The model can be further developed and will require integration with the signals sampled from the development vehicle.

Figures 53 and 54 show signal data attained from the Triumph.

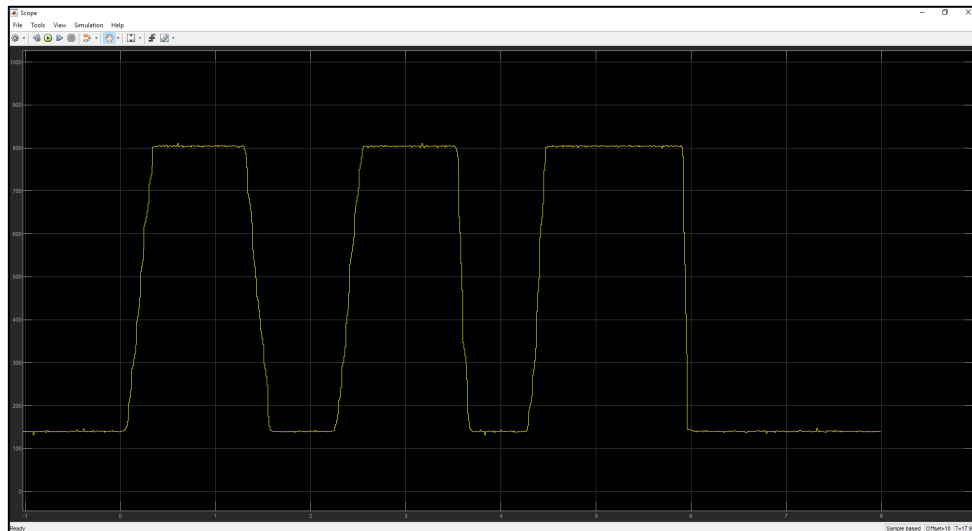


Figure 53 - 2010 Triumph Daytona 675 Throttle Position Signal

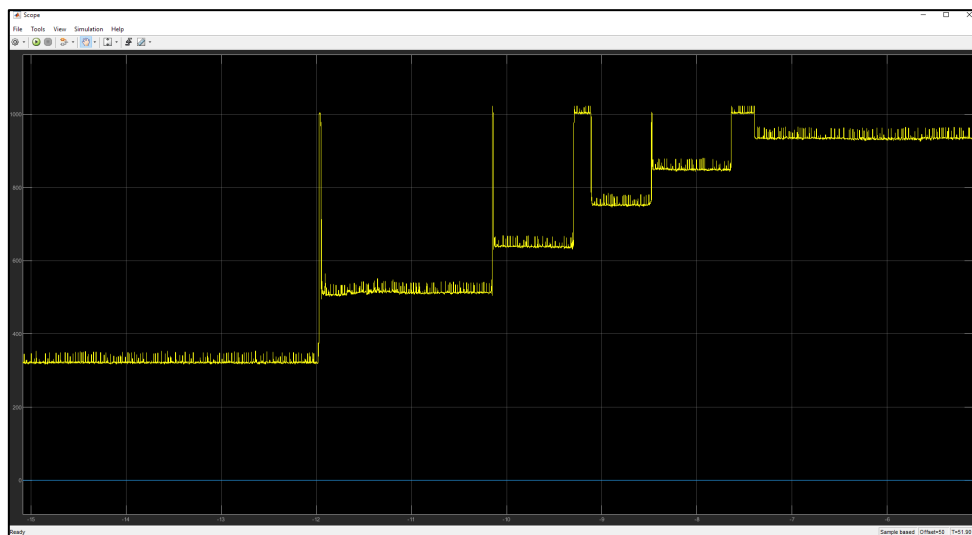


Figure 54 - 2010 Triumph Daytona 675 Gear Position Signal

The Arduino was unable to pick up a clean reliable RPM signal from the vehicle but for future testing purposes a solution has been derived.

$$RPM = \frac{Speed (MPH) \times Combined Gear Ratio \times 336}{Tyre Diameter (Inches)}$$

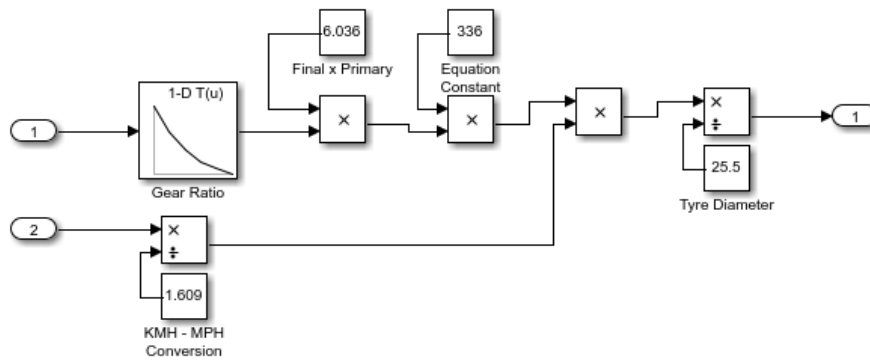


Figure 55 - Simulink RPM Calculation

Input 1 – Gear Position

Input 2 – Rear Wheel Velocity

As the primary and final gear ratios are fixed they are calculated and entered as a constant, a lookup table can then change the gearbox ratio depending on the output of the gear position sensor. Multiplied by an equation constant, then multiplied again by the velocity converted to MPH and finally divided by the tyre diameter in inches. While not ideal and limited by the fact that if the vehicle is stationary then no RPM is recorded Excel was used to verify the accuracy of this equation.

Gear				Ratio		MPH		RPM
1st	37	16		2.313		150.7		25628
2nd	39	21		1.857				20582
3rd	36	23		1.565		KMH		17347
4th	27	20		1.350		242.5		14961
5th	26	21		1.238				13721
6th	25	22		1.136				12594
Final	16	47		2.938		5.427989		
Primary	85	46		1.848				

Figure 56 - Excel RPM Calculator

Using the theoretical top speed used throughout the work of 242.5 km/h in the calculation gives an RPM value of 12594 which is within tolerance of the 12600 value quoted by the manufacturer and also aligns with the values of the gearing calculator.

Tyre	(180/55)	1.9786	0.6298		
RPM		12600			
Primary		46	85		
Ratios					Speed (km/h)
1st		16	37		119.17
2nd		21	39		148.39
3rd		23	36		176.06
4th		20	27		204.13
5th		21	26		222.58
6th		22	25		242.50
Final		16	47		

Figure 57 - Excel Gearing Calculator

While the initial calculations look promising this strategy would still need to be validated in a control test environment against the real RPM signal to ensure accuracy as at present it is unknown how much wheelspin or other unknown factors would affect the reading.

A load cell sensor was installed into the gear shift linkage to replace the conventional switch based sensor. A conventional switch only works in one direction with the gear linkage known as either Push or Pull depending on the configuration but a load cell (based on strain gauge) is not direction dependant to operate. The theory being that a signal could be used from the load cell to drive a gear assist control strategy, in its standard form when changing up through the gearbox the sensor allows the ECU to cut the ignition and allow the shift to pass without engaging the clutch. When investigating this signal it was discovered that the signal sits high and falls low when the sensor is engaged, unfortunately there is no way to distinguish if this is a change up or down the gearbox.

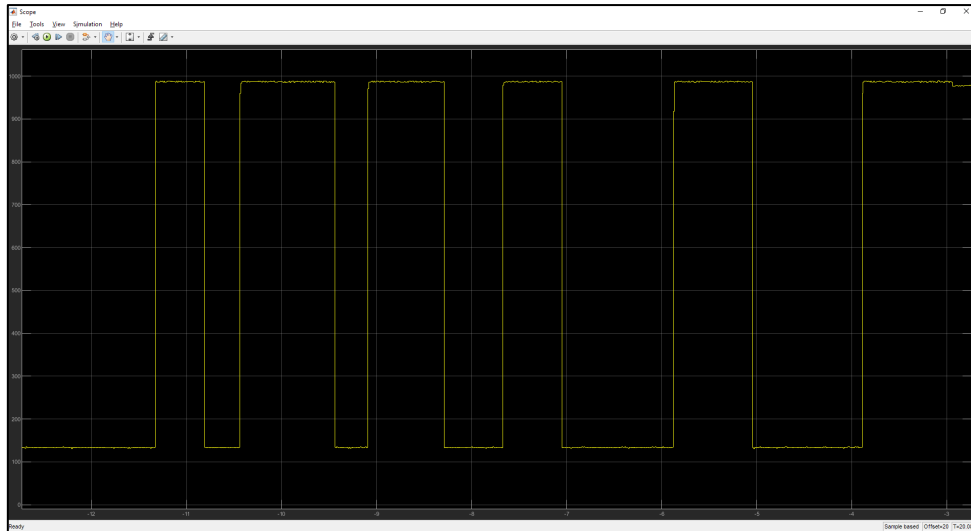


Figure 58 - Load Cell Signal

Assessments need to be made as to how the test platform could be fitted to the development vehicle with component placement and wiring runs. As the hardware is bespoke there are no obvious fitments for the additional sensors and parts will need to be either amended or redesigned to accommodate these changes.

Additional research to be carried out to analyse variable valve timing such as BMW ShiftCam.

The above summarises the lengthy time period that software development requires to prove effective and safe, for example CAN and CAN FD deployment taking between 7-8 years showcases how stringent a manufacturer has to be before a solution can be sold to either a client or the public.

References

- Raspberry Pi Foundation, 2019. *Raspberry Pi 3 Model B+*. [Online]
Available at: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>
[Accessed 24 April 2019].
- 44 Teeth, 2018. *All you need to know about the 2019 BMW S 1000 RR*. [Online]
Available at: <http://44teeth.com/wp-content/uploads/2018/11/P90327367.jpeg>
[Accessed 26 April 2019].
- AF1 Racing, 2019. *OEM Aprilia Demand Sensor*. [Online]
Available at: <https://www.af1racing.com/store/Scripts/prodView.asp?idproduct=13455>
[Accessed 26 April 2019].
- Amazon, 2019. *16x2 Blue LCD Module Controller HD44780*. [Online]
Available at: <https://www.amazon.com/Module-Controller-HD44780-Arduino-Corpc/dp/B01BB4VKIE>
[Accessed 29 April 2019].
- Amazon, 2019. *SunFounder MPU6050*. [Online]
Available at: <https://images-na.ssl-images-amazon.com/images/I/61GYgJyVv5L.SL1000.jpg>
[Accessed 27 April 2019].
- Arduino, 2019. *Analog Input*. [Online]
Available at: <https://www.arduino.cc/en/tutorial/AnalogInput>
[Accessed 25 April 2019].
- Arduino, 2019. *ARDUINO MEGA 2560 REV3*. [Online]
Available at: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/500x375/f8876a31b63532bbba4e781c30024a0a/a/0/a000067_front_1.jpg
[Accessed 24 April 2019].
- Arduino, 2019. *ARDUINO MEGA 2560 REV3 - Documentation*. [Online]
Available at: <https://store.arduino.cc/mega-2560-r3>
[Accessed 24 April 2019].
- Arduino, 2019. *ARDUINO MEGA 2560 REV3 - Tech Specs*. [Online]
Available at: <https://store.arduino.cc/mega-2560-r3>
[Accessed 24 April 2019].
- Arduino, 2019. *GROVE - MEGA SHIELD V1.2*. [Online]
Available at: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/500x375/f8876a31b63532bbba4e781c30024a0a/c/0/c000102_front.jpg
[Accessed 25 April 2019].
- Atmel, 2019. *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V*. [Online]
Available at: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR->

Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[Accessed 24 April 2019].

AutoEvolution, 2019. *TRIUMPH Daytona 675 Triple (2011 - 2012)*. [Online]

Available at: https://s1.cdn.autoevolution.com/images/moto_gallery/TRIUMPHDaytona675Triple-2870_2.jpg

[Accessed 26 April 2019].

Bennetts, 2018. *Meet the little black box that's controlling your next bike*. [Online]

Available at: <https://www.bennetts.co.uk/bikesocial/news-and-views/features/bikes/meet-the-little-black-box-thats-controlling-your-next-bike>

[Accessed 27 April 2019].

Bike Torque Racing, 2019. *DOMINO XM2 QUICK ACTION THROTTLE CONTROL*. [Online]

Available at:

https://www.biketorqueracing.co.uk/image.php?image_360=1&id=518203&size=large&var=p

[Accessed 26 April 2019].

BikesMedia, 2017. *Motorcycle Inertial Measurement Unit (IMU) Explained*. [Online]

Available at: <https://www.bikesmedia.in/uploads/image/reviews/2017/july/motorcycle-intertial-measurement-unit-imu.jpg>

[Accessed 27 April 2019].

BikesMedia, 2017. *Motorcycle Inertial Measurement Unit (IMU) Explained*. [Online]

Available at: <https://www.bikesmedia.in/uploads/image/reviews/2017/july/motorcycle-intertial-measurement-unit-imu-2.jpg>

[Accessed 27 April 2019].

BikesMedia, 2017. *Motorcycle Inertial Measurement Unit (IMU) Explained*. [Online]

Available at: <https://www.bikesmedia.in/reviews/motorcycle-intertial-measurement-unit-imu-explained.html>

[Accessed 27 April 2019].

BMW Motorrad, 2019. *Data & Equipment BMW S1000RR*. [Online]

Available at: <https://www.bmw-motorrad.co.uk/en/models/sport/s1000rr-2019/technicaldata.html>

[Accessed 26 April 2019].

BMW Motorrad, 2019. *The BMW S1000RR*. [Online]

Available at: <https://www.bmw-motorrad.co.uk/en/models/sport/s1000rr-2019.html>

[Accessed 26 April 2019].

Bosch Mobility Solutions, 2016. *Two-Wheeler & Powersports – Riding innovation*. [Online]

Available at: https://www.bosch-mobility-solutions.com/media/global/products-and-services/two-wheeler-and-powersports/riding-safety-systems/motorcycle-abs/folder_for_two_wheeler_powersports_riding_innovation.pdf

[Accessed 27 April 2019].

Bosch, 2019. *Motorcycle Stability Control*. [Online]

Available at: <https://www.bosch-mobility-solutions.com/en/products-and-services/two-wheeler-and-powersports/riding-safety-systems/motorcycle-stability-control/>

[Accessed 27 April 2019].

Chegg, 2019. *Analog Signal*. [Online]

Available at: <https://www.chegg.com/homework-help/definitions/analog-signal-4>
[Accessed 27 April 2019].

Chegg, 2019. *Digital Signal*. [Online]

Available at: <https://www.chegg.com/homework-help/definitions/digital-signal-4>
[Accessed 27 April 2019].

Chegg, 2019. *Digital Signal*. [Online]

Available at: <https://cramster-image.s3.amazonaws.com/definitions/elec-6-img-1.png>
[Accessed 27 April 2019].

Components101, 2017. *Servo Motor SG-90*. [Online]

Available at: https://components101.com/sites/default/files/component_pin/Servo-Motor-Wires.png
[Accessed 25 April 2019].

Components101, 2017. *Servo Motor SG-90*. [Online]

Available at: <https://components101.com/servo-motor-basics-pinout-datasheet>
[Accessed 25 April 2019].

Components101, 2017. *Servo Motor SG-90*. [Online]

Available at: <https://components101.com/sites/default/files/inline-images/servo-motor-dimensions.png>
[Accessed 25 April 2019].

Components101, 2017. *Servo Motor SG-90*. [Online]

Available at: <https://components101.com/sites/default/files/inline-images/how-to-use-servo-motor.png>
[Accessed 25 April 2019].

CSS Electronics, 2018. *CAN BUS EXPLAINED - A SIMPLE INTRO (2019)*. [Online]

Available at: <https://canlogger1000.csselectronics.com/img/CAN-Bus-Dummies-Simple-Message-Overview-SOF-RTR-CRC-ACK.png>
[Accessed 25 April 2019].

CSS Electronics, 2018. *CAN BUS EXPLAINED - A SIMPLE INTRO (2019)*. [Online]

Available at: <https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en#CAN-Bus-Message-Structure-Data-Logging-Explained>
[Accessed 25 April 2019].

eBay, 2019. *2015 KTM 1190 Adventure R ABS Throttle Grip Ride By Wire*. [Online]

Available at: <https://www.ebay.com/itm/253437579414>
[Accessed 26 April 2019].

Hitachi, 1998. *HD44780U (LCD-II)*. [Online]

Available at: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
[Accessed 29 April 2019].

Hugong Automation, 2019. *GKB-M0524*. [Online]

Available at: <http://hugong.cn/En/ProductShow.asp?ID=154>
[Accessed 25 April 2019].

I2C Bus, 2019. *I2C Bus*. [Online]

Available at: <https://www.i2c-bus.org/twi-bus/>

[Accessed 24 April 2019].

KTM, 2019. *KTM 1290 Super Duke R 2019 Software & Electronics*. [Online]

Available at: <https://www.ktm.com/gb/naked/1290-super-duke-r/software-electronics>

[Accessed 27 April 2019].

Kvaser, October. *Comparing CAN FD with Classical CAN*. [Online]

Available at: <https://www.kvaser.com/wp-content/uploads/2016/10/comparing-can-fd-with-classical-can.pdf>

[Accessed 25 April 2019].

MathWorks, 2019. *Arduino Support from Simulink*. [Online]

Available at: <https://uk.mathworks.com/hardware-support/arduino-simulink.html>

[Accessed 24 April 2019].

MathWorks, 2019. *Data Types Supported by Simulink*. [Online]

Available at: <https://uk.mathworks.com/help/simulink/ug/data-types-supported-by-simulink.html>

[Accessed 29 April 2019].

MathWorks, 2019. *Raspberry Pi Support from Simulink*. [Online]

Available at: <https://uk.mathworks.com/hardware-support/raspberry-pi-simulink.html>

[Accessed 24 April 2019].

MathWorks, 2019. *Servo Control*. [Online]

Available at: <https://uk.mathworks.com/help/supportpkg/arduino/examples/servo-control.html>

[Accessed 28 April 2019].

MathWorks, 2019. *What Is Sample Time?*. [Online]

Available at: <https://uk.mathworks.com/help/simulink/ug/what-is-sample-time.html>

[Accessed 28 April 2019].

Max BMW Motorcycles, 2019. *BMW Parts & Technical Diagrams S1000RR 17*. [Online]

Available at: <https://shop.maxbmw.com/fiche/DiagramsMain.aspx?vid=58389&rnd=07242017>

[Accessed 26 April 2019].

Motorcycle.com, 2013. *2014 KTM Super Duke R Review – First Ride*. [Online]

Available at: <https://www.motorcycle.com/manufacture/ktm/2014-ktm-super-duke-r-review-first-ride.html>

[Accessed 26 April 2019].

Motorcycle.com, 2019. *Motorcycle Safety Primer*. [Online]

Available at: <https://www.motorcycle.com/blog/wp-content/uploads/2013/06/Motorcycle-ABS-Brakes-0628-583x389.jpg>

[Accessed 27 April 2019].

RaphvK, 2018. *arduinoSimulink*. [Online]

Available at: <https://github.com/RaphvK/arduinoSimulink/wiki>

[Accessed 28 April 2019].

Raspberry Pi Foundation, 2019. *GPIO*. [Online]
Available at: <https://www.raspberrypi.org/documentation/usage/gpio/>
[Accessed 24 April 2019].

Raspberry Pi Foundation, 2019. *GPIO*. [Online]
Available at: <https://www.raspberrypi.org/documentation/usage/gpio/images/gpio-pins-pi2.jpg>
[Accessed 24 April 2019].

Raspberry Pi Foundation, 2019. *GPIO*. [Online]
Available at: <https://www.raspberrypi.org/documentation/usage/gpio/images/gpio-numbers-pi2.png>
[Accessed 24 April 2019].

Raspberry Pi Foundation, 2019. *Raspberry Pi 3 Model B+*. [Online]
Available at: <https://www.raspberrypi.org/app/uploads/2018/03/770A4970-1-1-1619x1080.jpg>
[Accessed 24 April 2019].

Robert Bosch GmbH, 1991. *CAN Specification Version 2.0*. [Online]
Available at: <http://esd.cs.ucr.edu/webres/can20.pdf>
[Accessed 25 April 2019].

SparkFun, 2019. *Analog vs. Digital*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/analog-vs-digital/all>
[Accessed 27 April 2019].

SparkFun, 2019. *I2C*. [Online]
Available at: <https://cdn.sparkfun.com/r/600-600/assets/3/d/1/b/6/51adfd8ce395f151b000000.png>
[Accessed 25 April 2019].

SparkFun, 2019. *I2C*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/i2c/all>
[Accessed 25 April 2019].

SparkFun, 2019. *Pulse Width Modulation*. [Online]
Available at: <https://cdn.sparkfun.com/assets/f/9/c/8/a/512e869bce395fbc64000002.JPG>
[Accessed 24 April 2019].

SparkFun, 2019. *Pulse Width Modulation*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Communication*. [Online]
Available at: <https://cdn.sparkfun.com/r/700-700/assets/2/5/c/4/5/50e1ce8bce395fb62b000000.png>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Communication*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/serial-communication/all>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Peripheral Interface (SPI)*. [Online]
Available at: <https://cdn.sparkfun.com/assets/f/c/6/2/4/52ddb2d5ce395f59658b4567.png>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Peripheral Interface (SPI)*. [Online]
Available at: <https://cdn.sparkfun.com/assets/d/6/b/f/9/52ddb2d8ce395fad638b4567.png>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Peripheral Interface (SPI)*. [Online]
Available at: <https://cdn.sparkfun.com/assets/f/f/e/5/8/52ddb2dbce395fae408b4568.png>
[Accessed 24 April 2019].

SparkFun, 2019. *Serial Peripheral Interface (SPI)*. [Online]
Available at: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>
[Accessed 24 April 2019].

TDK InvenSense, 2013. *MPU-6000 and MPU-6050 Product Specification Revision 3.4*. [Online]
Available at: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>
[Accessed 27 April 2019].

TDK InvenSense, 2019. *MPU-6050*. [Online]
Available at: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
[Accessed 27 April 2019].

Texas Instruments, 2016. *Introduction To The Controller Area Network (CAN)*. [Online]
Available at: <https://www.ti.com/lit/an/sloa101b/sloa101b.pdf>
[Accessed 25 April 2019].

Triumph Motorcycles, 2009. *Triumph Daytona 675 Specifications*. [Online]
Available at:
https://web.archive.org/web/20090430012350/http://www.triumph.co.uk/uk/2009%20Triumph%20Daytona%20675_Specifications.aspx
[Accessed 26 April 2019].

Visor Down, 2018. *HOW IT WORKS: IMU*. [Online]
Available at: <https://www.visordown.com/features/workshop/how-it-works-imu>
[Accessed 27 April 2019].

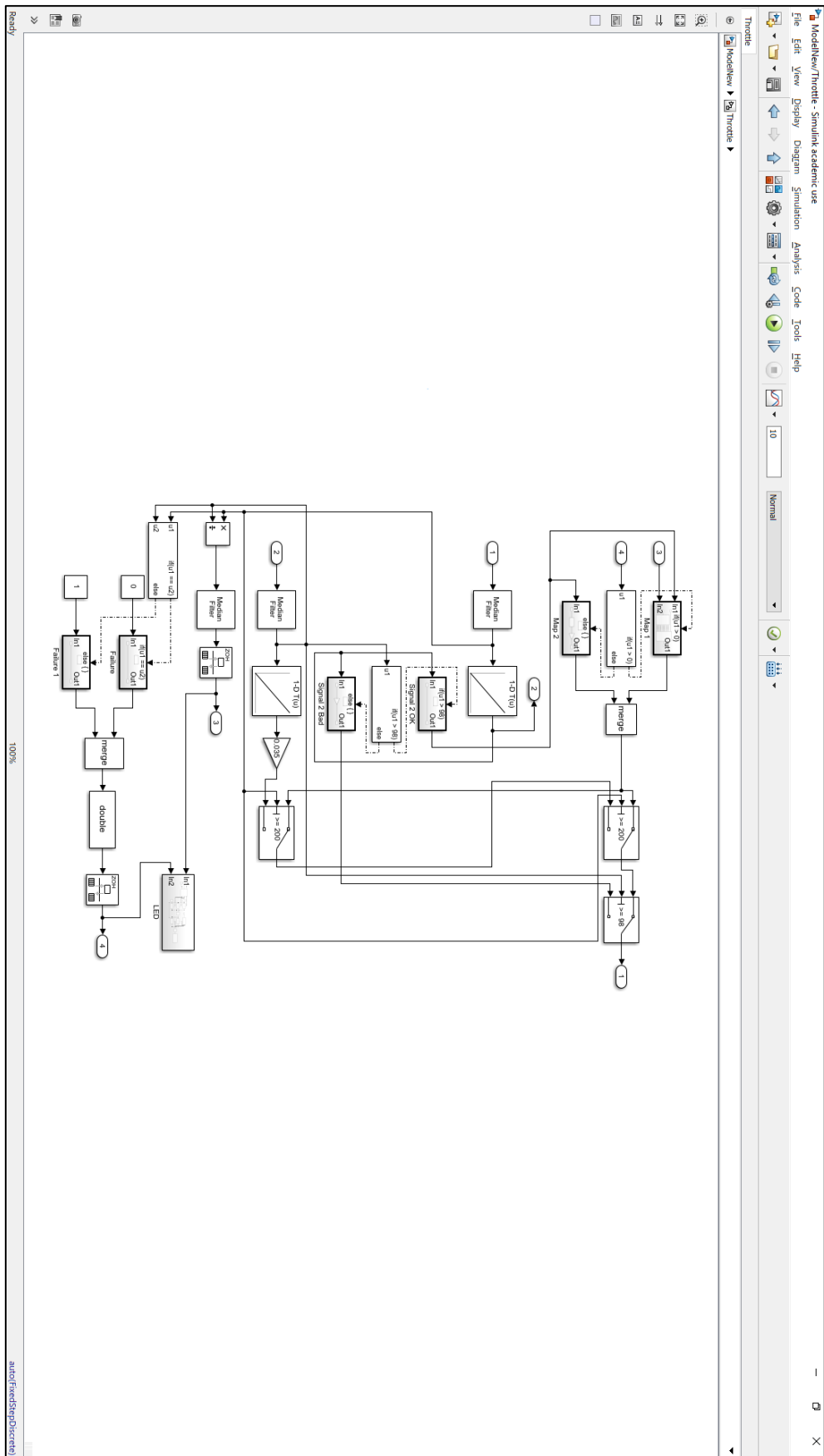
World Of Triumph, 2019. *Handlebars and Switches*. [Online]
Available at:
<https://www.worldoftriumph.com/partscatalog/catalog/listing/catalog/triumphmc/modelid/610414/block/100062758-0-2/>
[Accessed 26 April 2019].

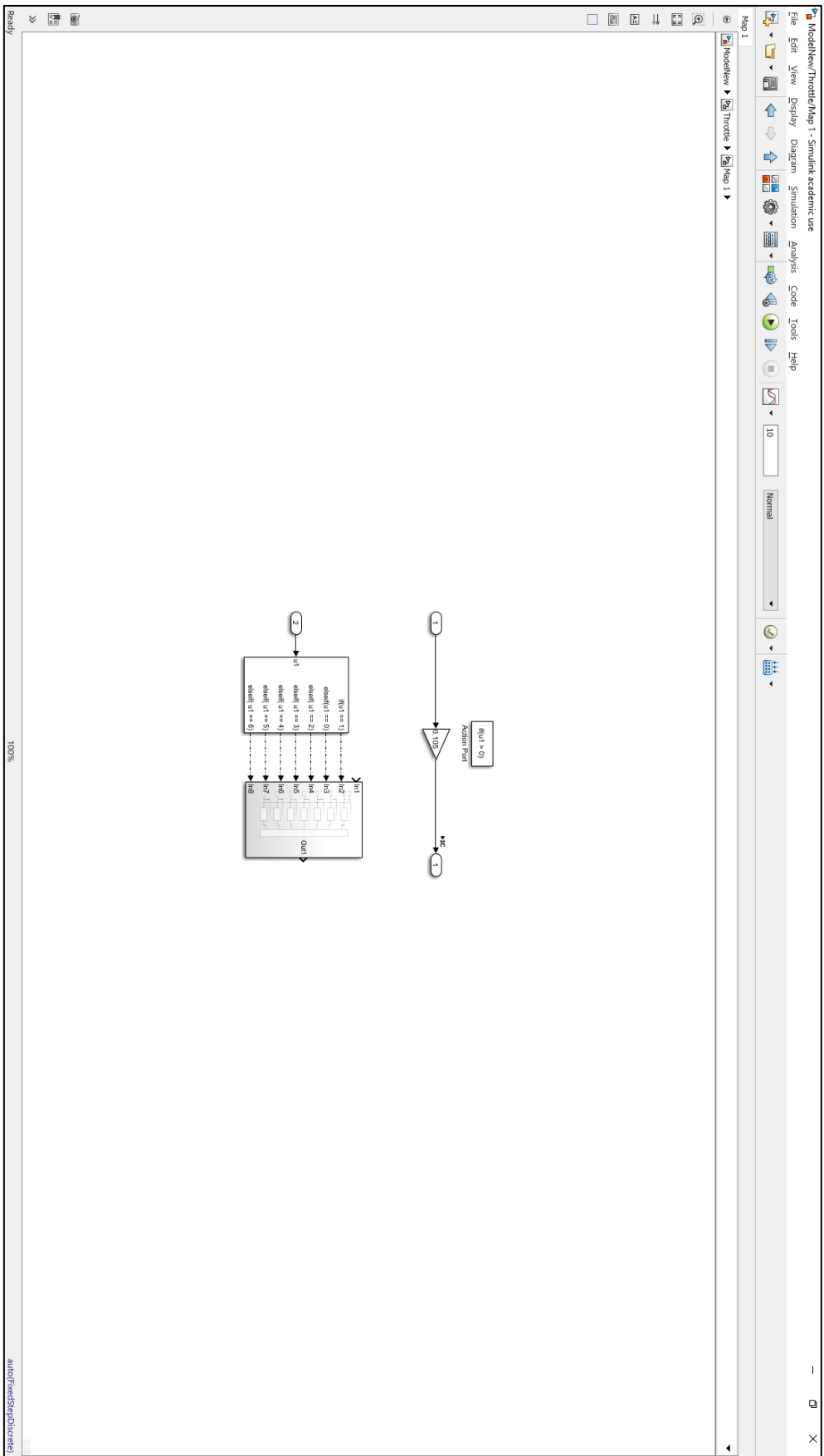
World Of Triumph, 2019. *Throttles Injectors and Fuel Rail*. [Online]
Available at:
<https://www.worldoftriumph.com/partscatalog/catalog/listing/catalog/triumphmc/modelid/610414/block/100062750-0-2/>
[Accessed 26 April 2019].

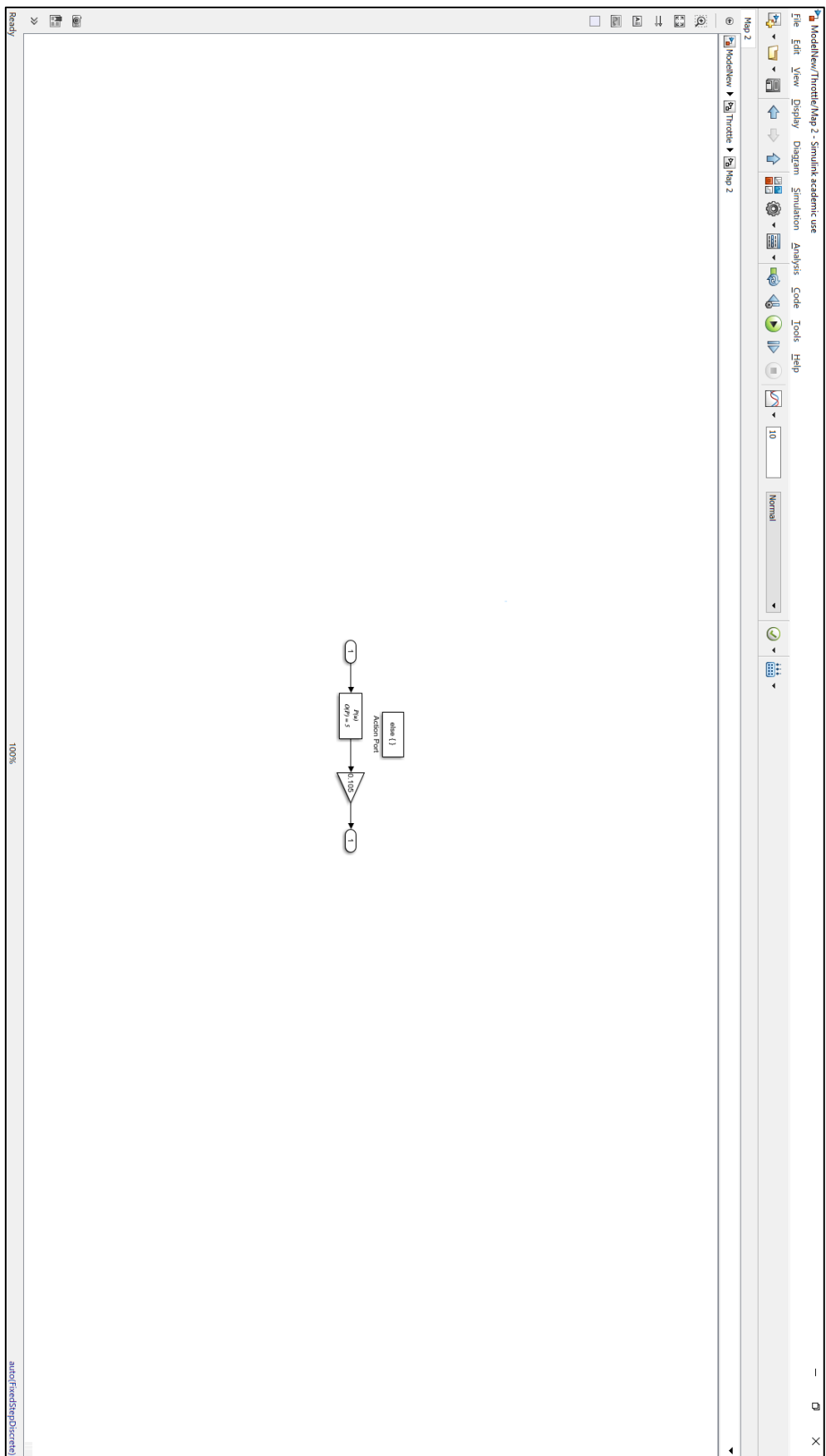
Page | 55

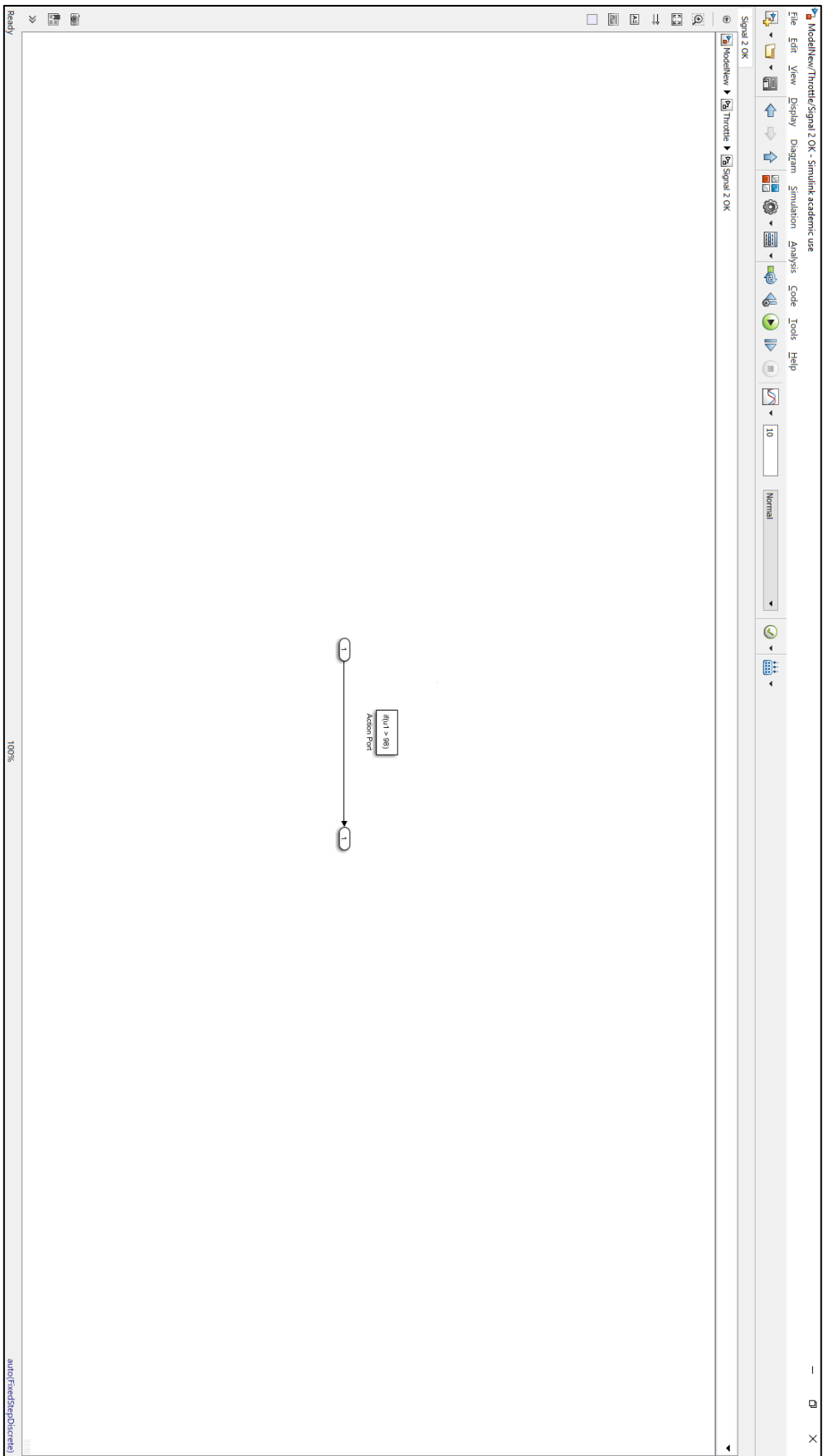


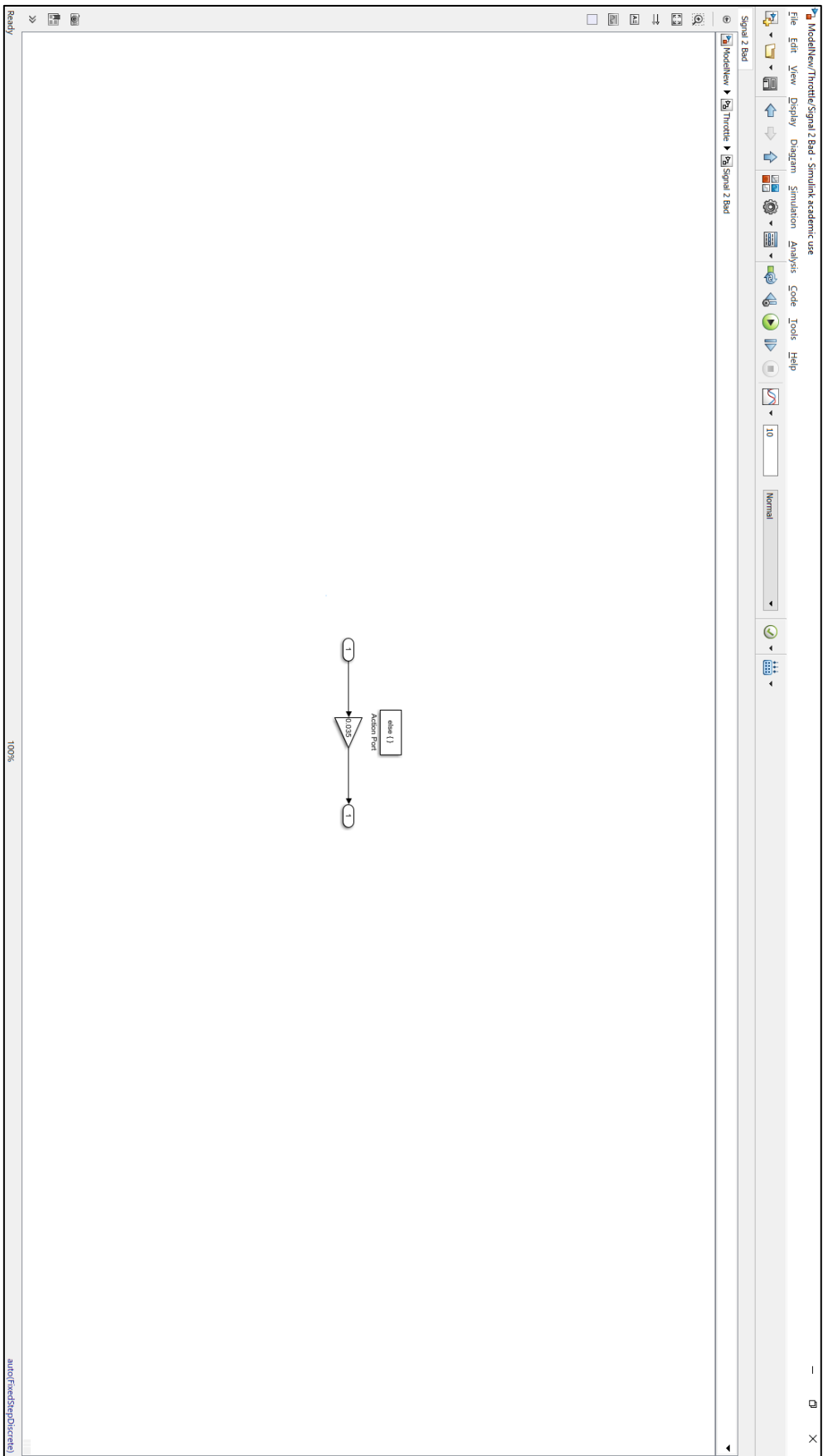
Appendix B

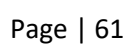




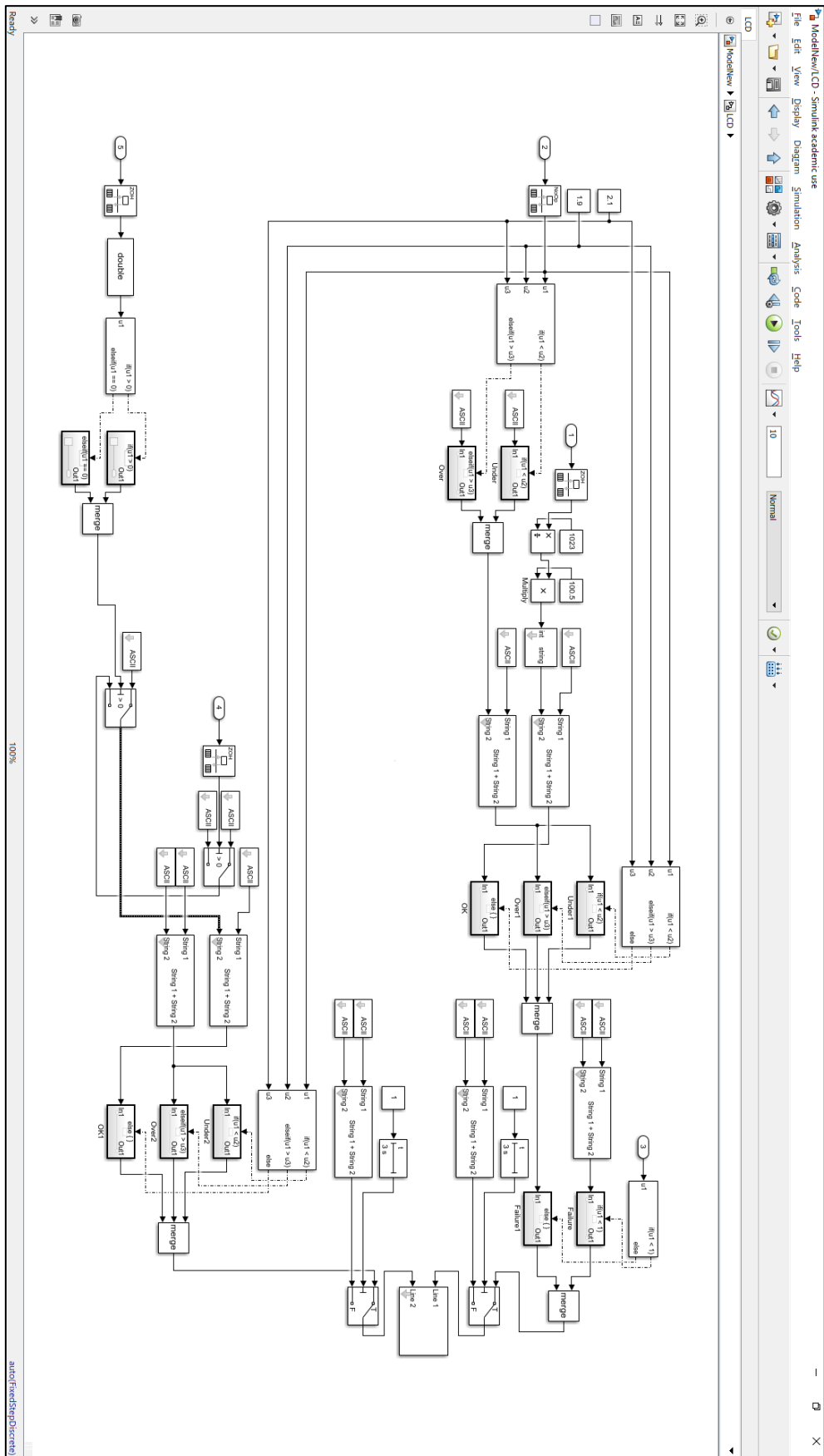




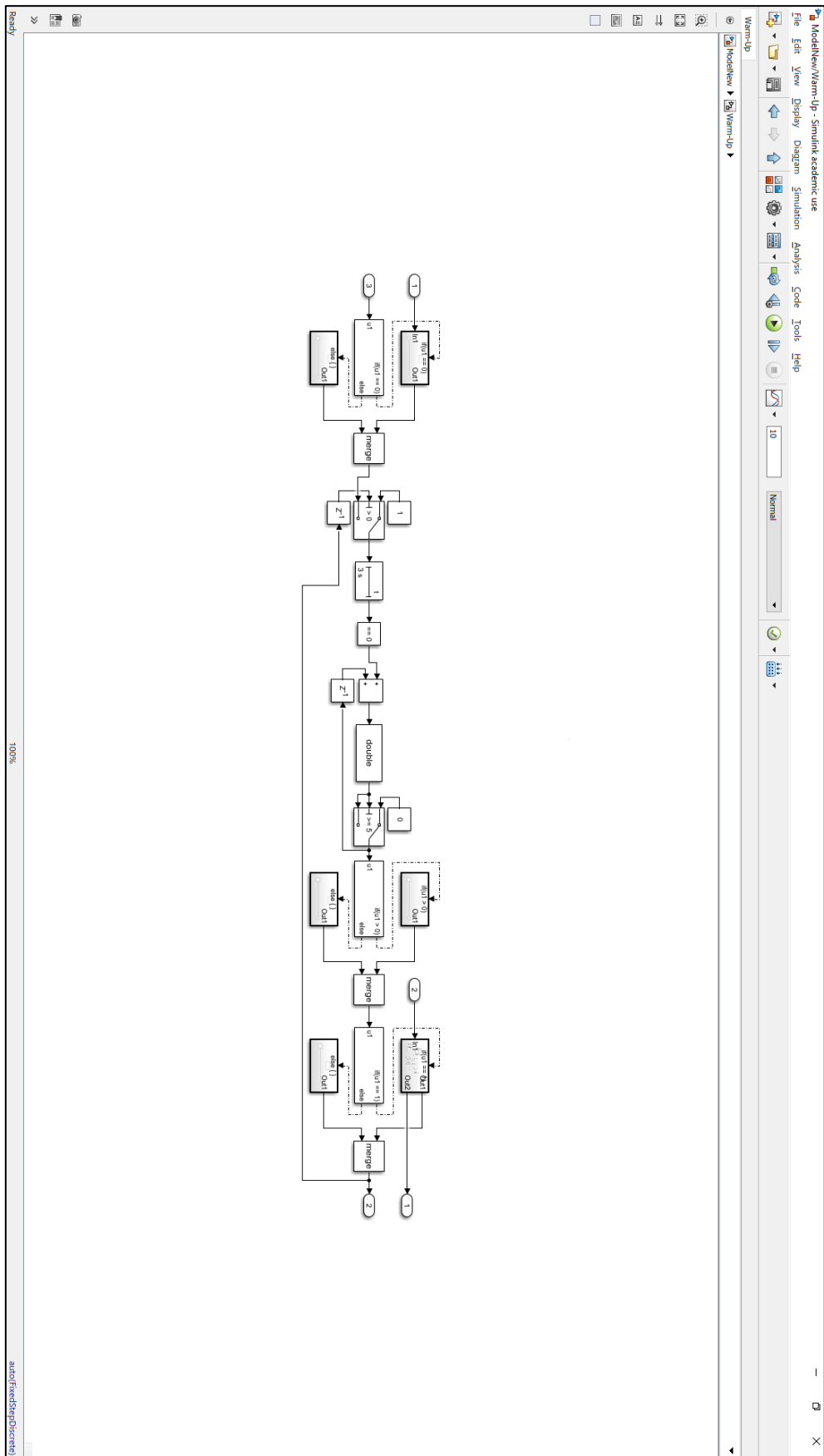


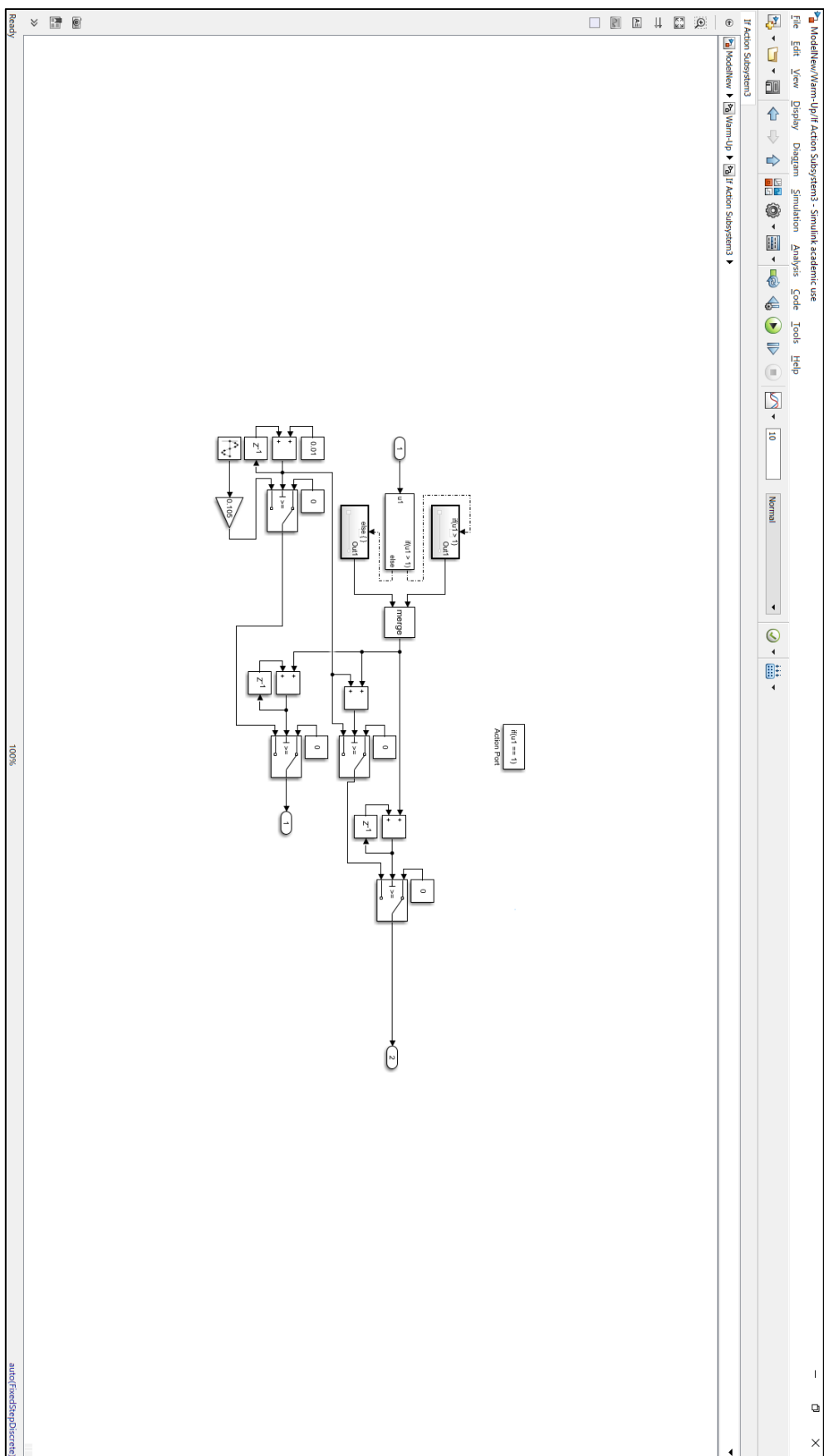


Appendix C

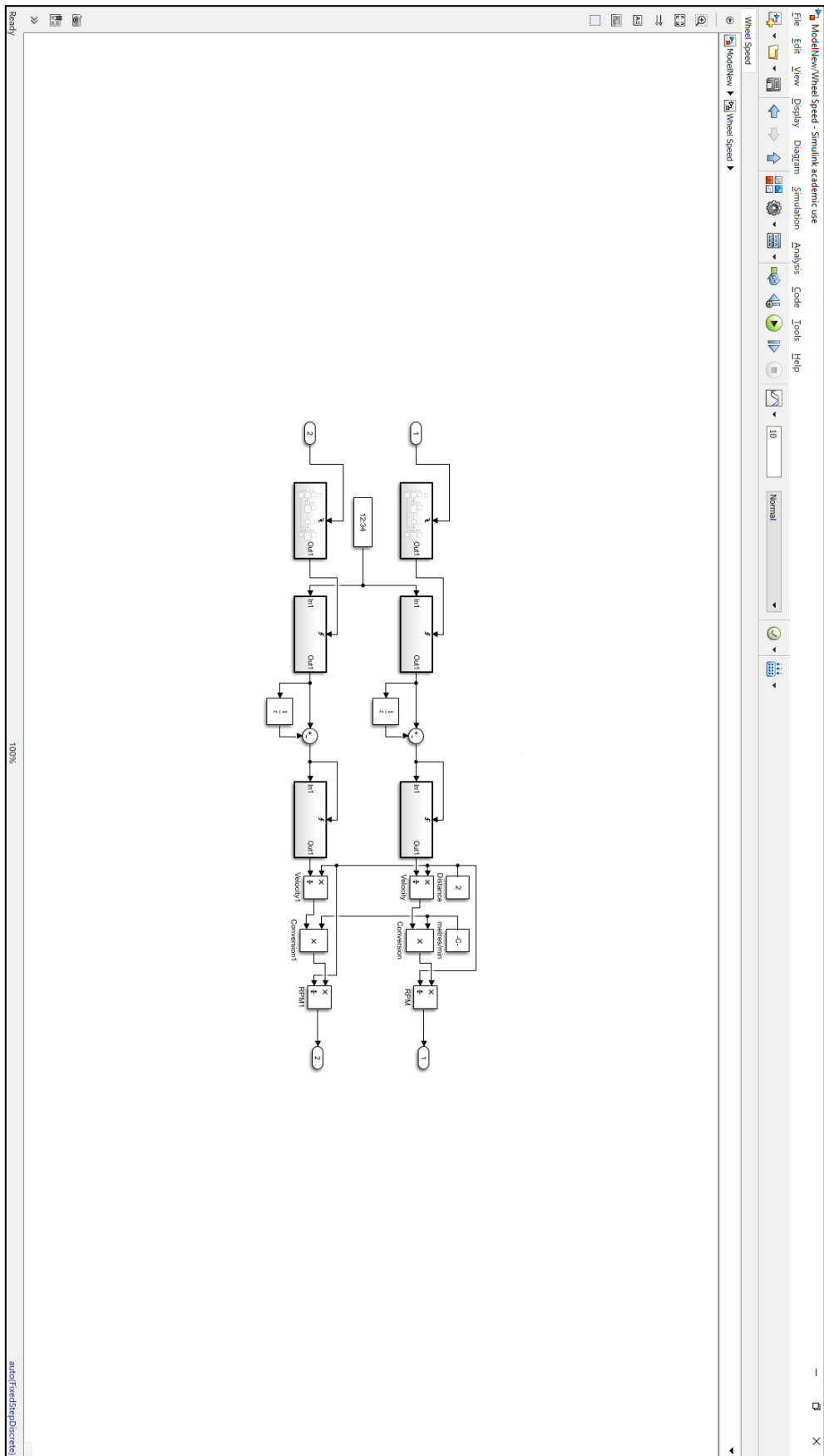


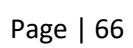
Appendix D



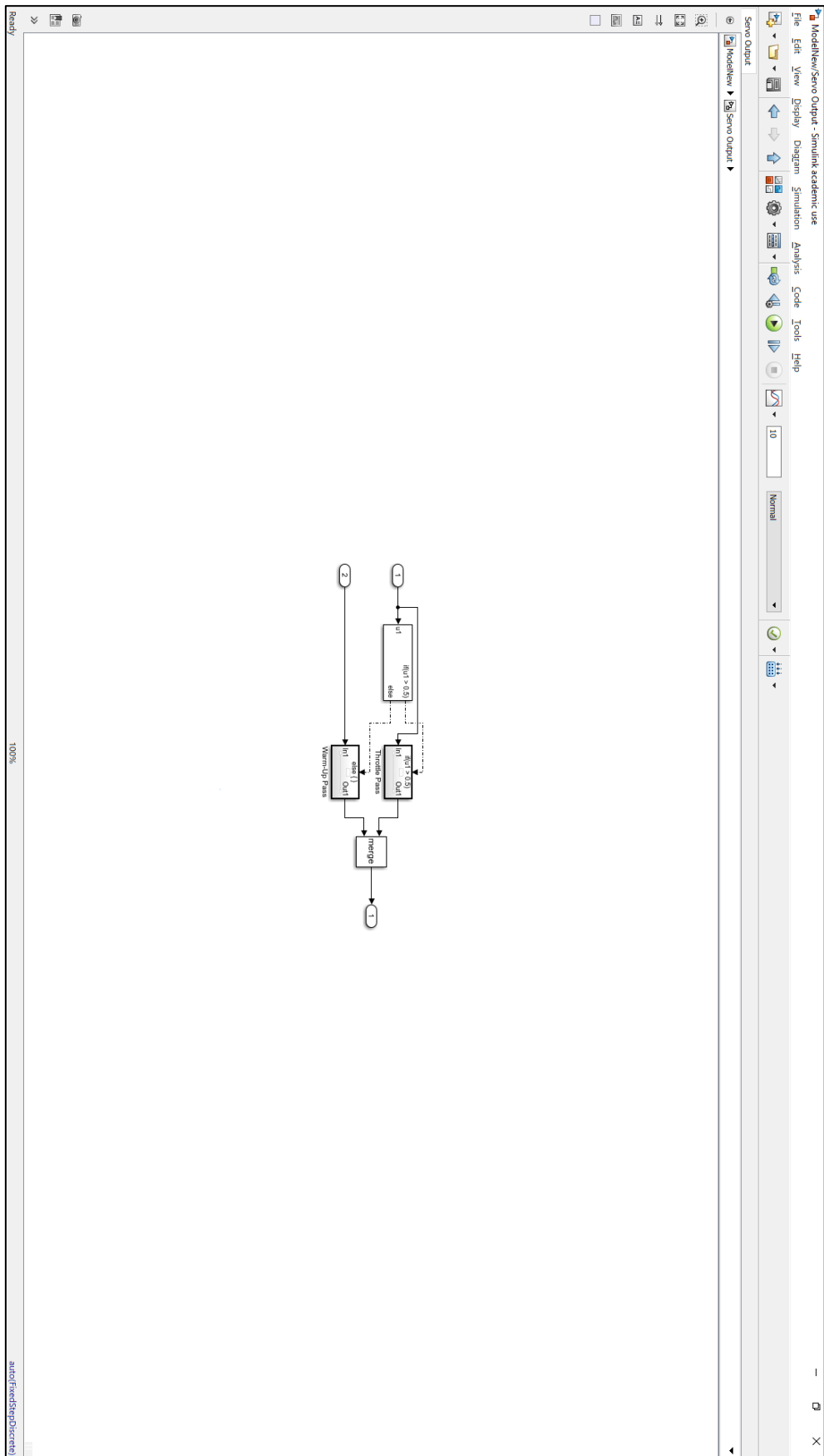


Appendix E





Appendix F



Page | 68

