

POLITECNICO DI TORINO

Master's Degree in Automotive Engineering



**Politecnico
di Torino**

Master's Degree Thesis

A.Y. 2023/2024

Control Development and Experimental Validation of a 1:5 Scaled Vehicle

Supervisors
Prof. Alessandro VIGLIANI
Eng. Angelo Domenico VELLA

Candidate
Shuning LIU

March 2024

Contents

Abstract	1
List of Figures	2
List of Tables	3
Introduction	4
1 Hardware Devices	9
1.1 Motors	9
1.1.1 Electrified Drive Line	9
1.1.2 Steering Servo Motor	20
1.2 Control Systems	21
1.2.1 E-Motor Controller	22
1.2.2 Radio Controller	25
1.2.3 Speedgoat Baseline Real-Time Target Machine	26
1.2.4 Router	29
1.3 Power Supply System	30
1.3.1 24 V Main Battery	30
1.3.2 7.4 V Secondary Battery	31
2 Controller Configuration and Design Software	33
2.1 ESCON Studio	33
2.2 MATLAB and Simulink	33
3 Experimental Tests	38
3.1 Data Acquisition System	38
3.1.1 Potentiometer	38
3.1.2 GPS Antenna	42
3.1.3 IMU	44
3.1.4 Scadas XS	44
3.2 Control Modalities	45
3.2.1 Control with Dashboard Blocks	46
3.2.2 Control with Radio Controller	49
3.2.3 Control with Predefined Vectors	53
3.3 Field Tests	59
3.3.1 Data Post Processing	59
3.3.2 Results and Discussions	61
Conclusion	81
References	83
A Appendix	85

Abstract

This thesis work focuses on the multiple control modalities development and experimental validation of a 1:5 scaled vehicle using a real-time target machine.

The 1:5 scaled vehicle is the Losi 5IVE-T 2.0, which is a 4WD, radio-controlled vehicle that has been upgraded from equipping with a 2-stroke internal combustion engine (ICE) to an electrified version using an e-motor, an e-motor controller, and a battery.

The real-time target machine is the Speedgoat Baseline, which is a compact and efficient system for this thesis work. It integrates MATLAB and Simulink workflows seamlessly through Simulink Real-Time and supports real-time testing of control modalities by utilizing supported I/O modules.

After emphasizing the motivation and benefits of using scaled vehicles and real-time target machines in automotive research and development, this thesis work introduces the 1:5 scaled vehicle. Information on all hardware and software devices (such as motors, controllers, Speedgoat, a router, batteries, etc.) required to establish a ready-to-run setup for the electrified 1:5 scaled vehicle is provided in the text. Additionally, devices of the real-time data acquisition system (such as potentiometer, GPS antenna, IMU, Scadas XS, etc.) and control models built with MATLAB and Simulink workflows that enable the ready-to-test ability of the 1:5 scaled vehicle are followed. Data post processing, results shown in figures, and some discussions of the field tests that use predefined trajectories to control the 1:5 scaled vehicle are listed before concluding this thesis work with potential future exploration and development.

List of Figures

1	Original Losi 5IVE-T 2.0: (a) with cover; (b) without cover.	6
2	Throttle servo motor of Losi 5IVE-T 2.0.	6
3	Part of Ackermann adjustable system of Losi 5IVE-T 2.0.	7
4	Rear suspension system of Losi 5IVE-T 2.0: (a) before modification; (b) after modification.	7
5	Views of electrified Losi 5IVE-T 2.0 assembly modelled in SOLIDWORKS.	10
6	Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox. . .	12
7	Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox. 12	
8	Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 25 teeth pinion gear.	13
9	Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 25 teeth pinion gear.	13
10	Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 30 teeth pinion gear.	14
11	Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 30 teeth pinion gear.	14
12	Drive line with a flexible motor shaft coupler.	18
13	Drive line with a rigid motor shaft coupler.	19
14	Final supporting structure of drive line frame.	19
15	RS PRO Brushless DC e-motor.	20
16	S900S 1/5 Scale Metal Gear Steering Servo and linkage.	21
17	Maxon Motor ESCON 50/5 e-motor controller.	22
18	Maxon Motor ESCON 50/5 e-motor controller interfaces.	23
19	Maxon Motor ESCON 50/5 e-motor controller.	24
20	Spektrum radio controller of Losi 5IVE-T 2.0.	26
21	Speedgoat Baseline real-time target machine interfaces.	27
22	IO397 I/O Module terminal board and pin mapping.	28
23	TP-LINK TL-MR3020 router interfaces.	30
24	7S4P 30000 mAh 24 V battery.	31
25	2S 4000 mAh 7.4 V 50C LiPo battery.	32
26	Two different positions of 2S 4000 mAh 7.4 V 50C LiPo battery.	32
27	ESCON Studio interface with the startup wizard window opened.	34
28	ESCON Studio interface with controller monitor section activated.	34
29	IO397 I/O Module driver blocks.	37
30	100 kOhm single-turn rotary potentiometer.	39
31	Views of two connection attempts of potentiometer and steering servo motor linkage. 39	
32	Three limit positions of steering servo motor linkage.	41
33	Steering angle measurement system of the Losi 5IVE-T 2.0.	41
34	GPS antenna and Scadas XS triggering cable connection.	43
35	IMU cable connection and Scadas XS interfaces.	43
36	Control with Dashboard Blocks: complete model.	45
37	Control with Dashboard Blocks: vehicle control subsystem.	46
38	Control with Dashboard Blocks: motion direction selection subsystem.	47
39	Control with Dashboard Blocks: complete wiring.	48

40	Control with Radio Controller: simplified model.	50
41	Control with Radio Controller: throttle filter.	50
42	Control with Radio Controller: complete model.	51
43	Control with Radio Controller: throttle DC_trigger to DC_controller DC % subsystem.	51
44	Control with Radio Controller: receiver wiring.	52
45	Control with Radio Controller: e-motor controller wiring.	52
46	8 shape trajectory example.	54
47	Control with Predefined Vectors: control part.	55
48	Control with Predefined Vectors: vehicle control subsystem.	56
49	Control with Predefined Vectors: steering path subsystem.	56
50	Control with Predefined Vectors: steering angle measurement subsystem.	57
51	Control with Predefined Vectors: trajectory selection part.	57
52	Control with Predefined Vectors: potentiometer wiring.	58
53	Control with Predefined Vectors: GPS antenna, IMU, and Scadas Xs wiring.	58
54	Losi 5IVE-T 2.0 on the ground of Piazzetta Manlio Brosio.	60
55	Acceleration-Deceleration Trajectory plots: 4 km/h.	64
56	Acceleration-Deceleration Trajectory plots: 6 km/h.	65
57	Circle Trajectory plots: 4 km/h; 30deg.	66
58	Circle Trajectory plots: 4 km/h; -30deg.	67
59	Circle Trajectory plots: 6 km/h; 30 deg.	70
60	Circle Trajectory example: 4 km/h; 30deg	71
61	Spiral Trajectory plots: 4 km/h and 6 km/h; from 30 deg to 0 deg.	73
62	Spiral Trajectory example: 4 km/h; from 30 deg to 0 deg.	74
63	8-Shape Trajectory plots: 4 km/h; 30 deg and -30 deg.	75
64	8-Shape Trajectory example: 4 km/h; 30 deg and -30 deg.	76
65	Sweep Trajectory plots: 4 km/h and 6 km/h; 30 deg.	77
66	Sweep Trajectory time domain plots: 4 km/h; 30 deg, 20 deg, and 10 deg.	78
67	Sweep Trajectory frequency domain plots: 4 km/h and 6 km/h; 30 deg.	79
68	Sweep Trajectory frequency domain plots: 4 km/h; 20 deg, and 10 deg.	80

List of Tables

1	Diameters of 1.5 module motor pinion gears.	10
2	Parameters of the startup version of the electrified Losi 5IVE-T 2.0.	15
3	Longitudinal performance indicators of the vehicle using three new drive line options.	16
4	Lateral performance indicator of the vehicle using startup version of drive line and three new drive line options.	17
5	Components of final drive line of the electrified Losi 5IVE-T 2.0	20
6	Components of steering angle measurement system of the electrified Losi 5IVE-T 2.0	40

Introduction

The goal of this thesis work is to develop and test multiple control modalities for a 1:5 scaled vehicle using a real-time target machine. Before starting to focus on specific details, preliminary knowledge of the motivation and benefits of using scaled vehicles and real-time target machines should be acquired.

Scaled Vehicle Scaled vehicles typically refer to model vehicles or radio-controlled vehicles that are proportionally smaller than real vehicles. These vehicles can be static display models or toy race cars that can be controlled remotely.

In the process of automotive research and development, small-scaled vehicles with structures similar to those of actual vehicles are used to develop and test various vehicle systems for the following reasons: (i) Firstly, it can reduce the investment in various resources. For example, the simulation of all kinds of driving scenarios and the debugging and calibration of the functions of various vehicle systems can be completed at relatively small experimental sites. This can significantly reduce research and development costs; (ii) In addition, automotive systems are simplified to a certain extent. This helps engineers better and more effectively understand the interactions between the systems, thus helping them to optimise the functional systems. It improves the efficiency of research and development and ultimately enhances the performance and safety of the actual vehicles.

Numerous studies have been conducted in the area of utilising scaled vehicles to facilitate automotive research and development. A lab-scale testbed with many scaled vehicles that have similar longitudinal and powertrain dynamics to those of a full-size high-mobility multipurpose wheeled vehicle can be developed to validate decision and control algorithms for intelligent transportation systems (ITS) applications, as stated in [1], thanks to the high reliability of the scaling approach based on the hardware-in-the-loop (HIL) simulation demonstrated. Apart from longitudinal dynamics, the scaled vehicles are also used to study lateral dynamics. As stated in [2], lateral control designs can be tested by placing a scaled vehicle on a standard exercise treadmill, which simulates the motion of a full-size vehicle on a paved road. In particular, a scaled vehicle can also be used to evaluate an anti-lock braking system (ABS) controller, as stated in [3], and investigate the influence of various properties on rollover propensity, as stated in [4]. As researchers focus more and more on autonomous driving technologies, scaled vehicles are also used to validate control algorithms designed for autonomous driving tasks such as adaptive cruise control, lane keeping, and lane following, as stated in [5].

Real-Time Target Machine In the process of technology development in various fields, real-time target machines are mainly used in two ways: (i) One is Rapid Control Prototyping (RCP). The real-time target machine runs control algorithms to control the real controlled objects (such as switches, solenoid valves, motors, engines, etc.) and performs real-time data acquisition, processing, and analysis to quickly verify and test the feasibility of the control algorithms; (ii) The other is the Hardware-in-the-Loop simulation (HIL). The real controllers control the models of controlled objects run by the real-time target machine in order to test the stability and reliability of the controllers. The combination of these two methods for the verification of entire control systems can provide key data support and technical assurance for the research and development process.

In the automotive field, many application use cases, as can be found in [6], have approved that using real-time target machines is a sufficiently efficient solution to improve the performances of both the control algorithms and electronic control units (ECU).

Losi 5IVE-T 2.0 The scaled vehicle for this thesis work is selected to be Losi 5IVE-T 2.0. Isometric views of the original Losi 5IVE-T 2.0, with and without the cover, can be seen in Figure 1.

The original Losi 5IVE-T 2.0, as stated in [7], is a 1:5-scale 4WD radio-controlled vehicle equipped with a Zenoah 32cc gasoline pull-start engine, which is able to provide a maximum net power of 2.4 kW at 13000 rpm and a maximum net torque of 2.0 Nm at 9000 rpm, as stated in [8]. There is an electric servo motor, which is shown in Figure 2, in charge of controlling the throttle valve to regulate the speed of the Zenoah engine. In addition, since the engine generates a significant amount of torque, the throttle servo brace is designed to eliminate radio tray flex while braking, providing added rigidity to the servo. This results in a more precise and consistent brake feel while also reducing wear and tear.

As for the steering function of the Losi 5IVE-T 2.0, another electric servo motor is used to regulate the steering angle of the Losi 5IVE-T 2.0 by controlling the rotation of the steering linkage set, which belongs to an Ackermann adjustable system. The Ackermann adjustable system also has four rod sets in charge of the camber angles of all four tyres and two rod sets responsible for the toe angles of the two front tires. A top zoomed view of the Losi 5IVE-T 2.0 to show the two rod sets of the front left tyre can be seen in Figure 3.

Another key piece of information is about the suspension system. One of the reasons why Losi 5IVE-T 2.0 is selected to investigate the control of a scaled vehicle is that it guarantees the suspension system's resemblance to those of full-sized vehicles, thus offering more accurate technical support and trustworthy insights derived from lateral and longitudinal dynamic experiments. The suspension system is made up of springs, dampers, and anti-roll bars for both the front and rear axles. The original springs used are blue ones, with a 11.6 lb rate for front springs and an 8.0 lb rate for rear springs, and the original dampers used are 45 WT ones.

In order to carry the Speedgoat baseline real-time target machine, an additional bar frame is designed that also carries devices other than the Speedgoat. The second floor results in a significant reduction in terms of ground clearance. Fortunately, due to the complete ability to adjust these parts as needed, parameters such as spring stiffness and damping response may all be changed. After modification, for the whole suspension system, the springs used in the final setup of Losi 5IVE-T 2.0 are orange ones, with a 14.2 lb rate for front springs and an 10.7 lb rate for rear springs, and the dampers used are 80 WT ones. In theory and in practice, higher stiffness and damping work better than before. As an example, the rear suspension system before and after modification is shown in Figure 4.

Back to the throttle function of the Losi 5IVE-T 2.0, considering the need for more precise steering and throttle control based on electrical signals managed by multiple controllers, such as the e-motor controller and Speedgoat, there are also modifications to electrify the drive line. All the detailed information is in the following chapter.



(a)



(b)

Figure 1: Original Losi 5IVE-T 2.0: (a) with cover; (b) without cover.



Figure 2: Throttle servo motor of Losi 5IVE-T 2.0.



Figure 3: Part of Ackermann adjustable system of Losi 5IVE-T 2.0.



(a)



(b)

Figure 4: Rear suspension system of Losi 5IVE-T 2.0: (a) before modification; (b) after modification.

The thesis is structured as follows:

- **Hardware Devices:** the selection of drive line alternatives with various motor pinion gear teeth is where this first chapter begins in order to raise the Losi 5IVE-T 2.0's maximum speed. Following that, other devices (such as the e-motor, steering servo motor, e-motor controller, radio controller, Speedgoat baseline real-time target machine, router, batteries, etc.) utilised in the electrification of the Losi 5IVE-T 2.0 are presented along with their features, installation into the setup of the Losi 5IVE-T 2.0, and wiring with other components;
- **Software:** ESCON Studio, the software program used to setup the e-motor controller, is initially introduced in this chapter. In particular, a detailed explanation of the configuring process is provided. Secondly, how does Simulink Real-Time integrates MATLAB and Simulink workflows is introduced. More specifically, the capabilities of several groups of Simulink blocks are listed. In addition, the rough correlation between the electrical signals and the physical pins is presented;
- **Experimental Tests:** in order to prepare for the experimental tests on multiple control modalities, the primary aspect that should be focused on is establishing a data acquisition system. Characteristics, integration into the setup of the Losi 5IVE-T 2.0, and wiring with other participants of the hardware devices used to support data collection (such as the potentiometer, GPS antenna, IMU, Scadas XS, etc.) are introduced. Next, details about codes, models, and relationships between the electrical signals and the physical pins of various control modalities are explained.
- **Conclusion:** is the last chapter that provides an overview of the entire thesis research, with a focus on the benefits and drawbacks of using different control modalities. With regard to the above, the conclusion offers some potential fields for further research and expansion.

1 Hardware Devices

This chapter begins with the selection of drive line options that include a variety of motor pinion gear teeth. The goal of that section is to increase the maximum speed of the Losi 5IVE-T 2.0. Other devices that are utilised in the electrification of the Losi 5IVE-T 2.0 will be presented after that. These devices include the e-motor, steering servo motor, e-motor controller, radio controller, Speedgoat baseline real-time target machine, router, batteries, and so on. Additionally, their features, installation into the setup of the Losi 5IVE-T 2.0, and wiring with other components will be discussed.

1.1 Motors

The human race has developed a wide range of transportation options in order to shorten the amount of time spent on travel and to make travel more comfortable. Furthermore, the propulsion mechanism that propels the vehicle ahead is the most essential component of these alternatives to the human walking method. The propulsion system of the cars, which consisted of internal combustion engines like the one found in the first Losi 5IVE-T 2.0, is the heart of the vehicles for a considerable amount of time in the past. However, in recent years, as a result of the development and expansion of technology, electric vehicles have made their way into people's minds. Electric vehicles, in comparison to conventional fuel vehicles, play a key role in the achievement of energy sustainability, the protection of the environment, and the reduction of dependence on fossil fuels. Regarding the trend of the automotive industry, one of the most essential reasons for changing the Losi 5IVE-T 2.0 from its previous version to a new electric power system is this important reason.

1.1.1 Electrified Drive Line

For the purpose of providing the operator with fundamental control over the vehicle on a mechanical level, the original version of the Losi 5IVE-T 2.0 makes use of conventional transmitter and receiver devices. In this scenario, the specific and actual speed of the vehicle, as well as the turning movements, are determined by the operator's judgement and estimated expectations regarding the status of the vehicle. To put it another way, the operator's control of the movement is very rough. Therefore, converting the initial version of the Losi 5IVE-T 2.0 to an electric power system is a very advantageous and logical step to facilitate the later utilisation of electronic components for the purpose of precisely controlling vehicle dynamics.

When it comes to the throttle function of the first edition of the Losi 5IVE-T 2.0, Luca Biando designed an electric drive line that included an e-motor and a gearbox to replace the Zenoah 32cc gasoline pull-start engine. This allowed the vehicle to achieve a maximum speed of 2.65 km/h, as stated in [9]. In the following chapter, a number of different alternatives will be outlined and examined in order to enhance the maximum speed that the electrified Losi 5IVE-T 2.0 is capable of achieving.

Motor Pinion Gear The startup layout of the electrified Losi 5IVE-T 2.0 can be found in Figure 5a. Due to the fact that the goal is to enhance the maximum speed of this vehicle, the drive line frame, which is depicted in Figure 5b, is the crucial component that required attention.

Taking into consideration the fact that replacing the stock LOS352001 62Teeth 1.5Module Spur

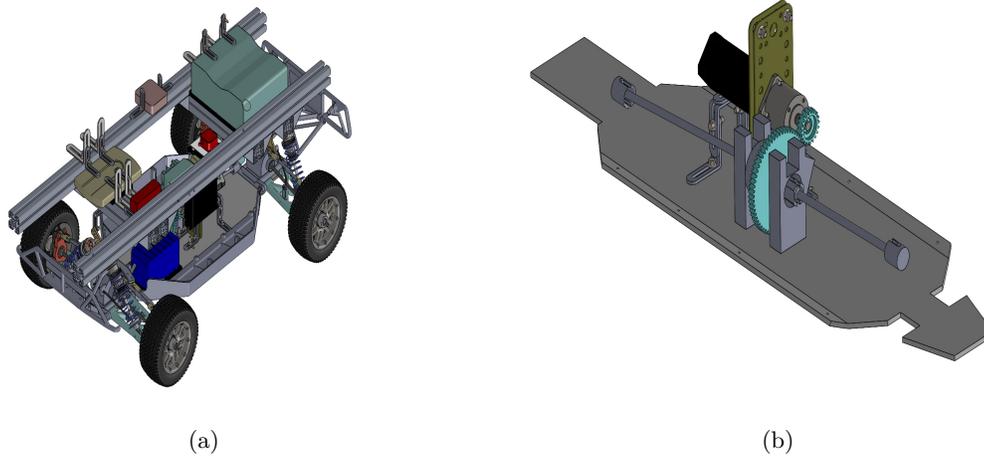


Figure 5: Views of electrified Losi 5IVE-T 2.0 assembly modelled in SOLIDWORKS.

Gear is a complicated process, as mentioned in [7], there are three different methods that might be implemented in order to enhance the maximum speed of the vehicle:

- Remove the 5:1 gearbox to decrease the gear ratio of gearbox from 5 to 1.
- Increase the teeth number of the motor pinion gear to decrease the gear ratio of transmission from 3.1 to a smaller value.
- Increase the motor maximum rotational speed.

The third alternative was the one that was studied last because it is the one that saves the most money. Before acquiring components and fitting them to the startup version of the drive line, three possibilities were built using SOLIDWORKS and compared using relevant performance indicators based on the other two solutions and the motor pinion gears that are available. These three new drive line frames of the electrified Losi 5IVE-T 2.0 were modelled in SOLIDWORKS using the diameters of the available 1.5 module motor pinion gears with varied tooth numbers offered by the manufacturer, as indicated in [10]. These dimensions are displayed in Table 1. In this context, the tooth number of the motor pinion gears is denoted by T_p .

Table 1: Diameters of 1.5 module motor pinion gears.

T_p	Diameter
teeth	mm
20	33
25	40.5
30	48

Drive Line Options

1. Remove the 5:1 gearbox only:

In order to begin the process of updating the drive line frame CAD assembly, the gearbox was initially removed. A motor shaft coupler was then attached to the motor in order to link the motor shaft, which has a diameter of 5 mm, to a motor shaft extension rod, which has a diameter of 8 mm. This was done in order to maintain the location of the motor and the structures that support it in its initial configuration. In the end, the original LOSB5045 20Teeth 1.5Module Pinion Gear is replaced with a new 20 teeth one and properly connected to the stock central differential spur gear. A new 20 teeth gear was needed because of the information provided in [9], which specifies that the gearbox output shaft can not be detached. Figure 6 serves as a visual representation of the entire drive line frame.

In the front view of the assembly, after the new motor pinion gear is correctly attached to the stock central differential spur gear, there is no interference between the bolts that are used to fix the motor supporting structure in its position and the chassis plate, as shown in Figure 7, which is a positive development. As a result, this choice is the one that can be used with the least amount of difficulty.

2. Remove the 5:1 gearbox and use a 25 teeth motor pinion gear:

After replacing the factory motor pinion gear with a new one with 25 teeth, the new gear was then linked to the stock central differential spur gear in the correct manner. The complete drive line frame is shown in Figure 8.

After correctly attaching the new motor pinion gear to the stock central differential spur gear from the front view of the assembly, there is an interference between the bolts that are used to fix the motor supporting structure in its position and the chassis plate, as shown in Figure 9. This interference can be considered undesirable. On the other hand, the issue might be willingly resolved by extending the length of the existing slots. As a result, this alternative constitutes a potential solution.

3. Remove the 5:1 gearbox and use a 30 teeth motor pinion gear:

This alternative is very similar to the second one in many respects. An illustration of the whole drive line frame can be found in Figure 10. There is an obvious representation of the interference in Figure 11.

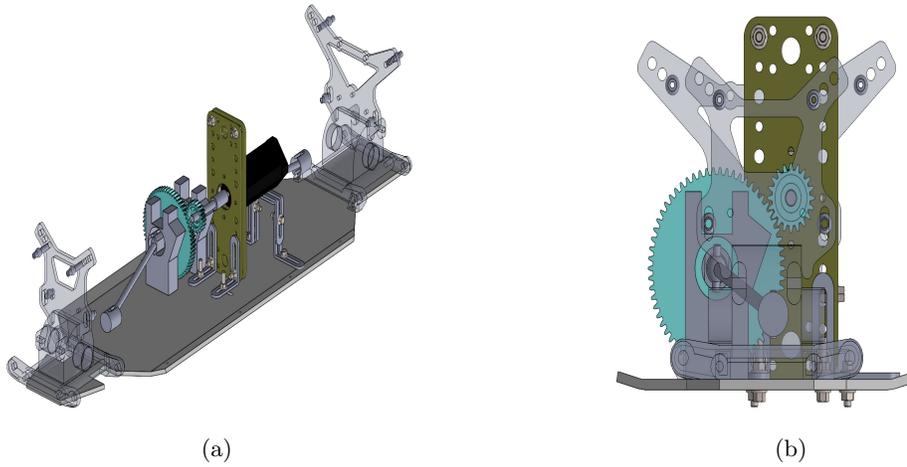


Figure 6: Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox.

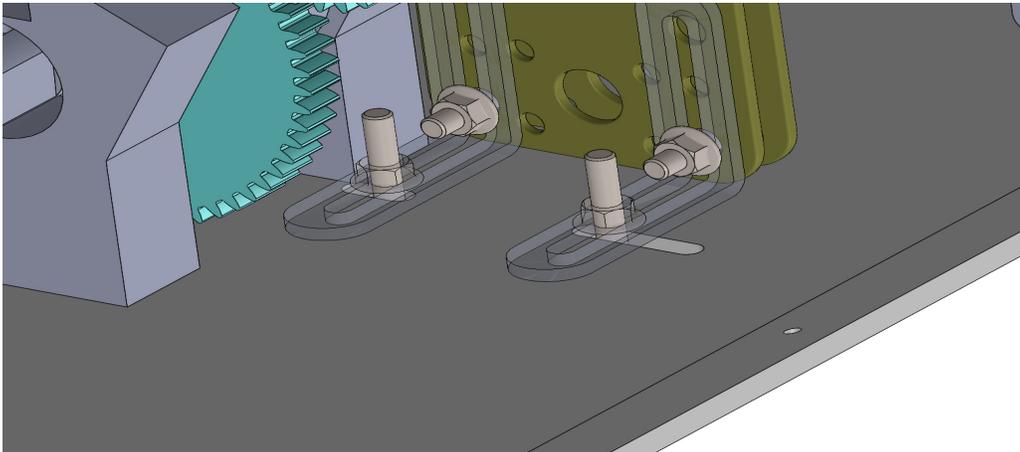


Figure 7: Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox.

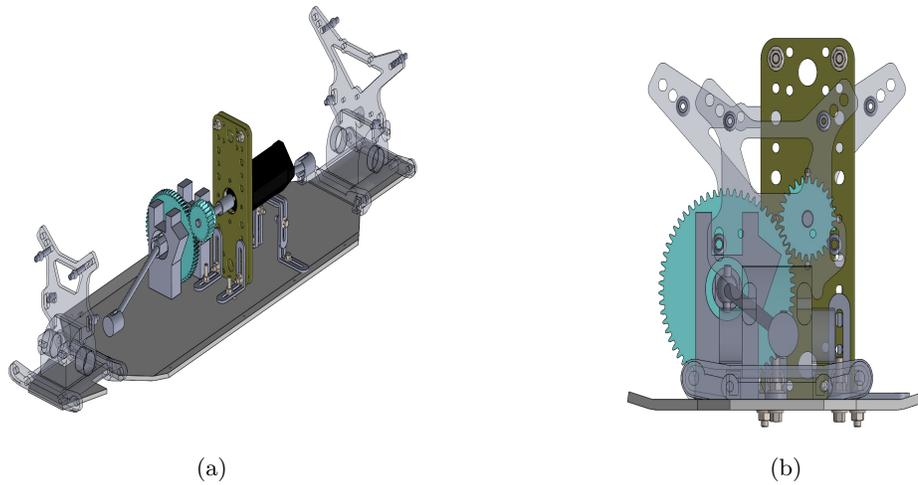


Figure 8: Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 25 teeth pinion gear.

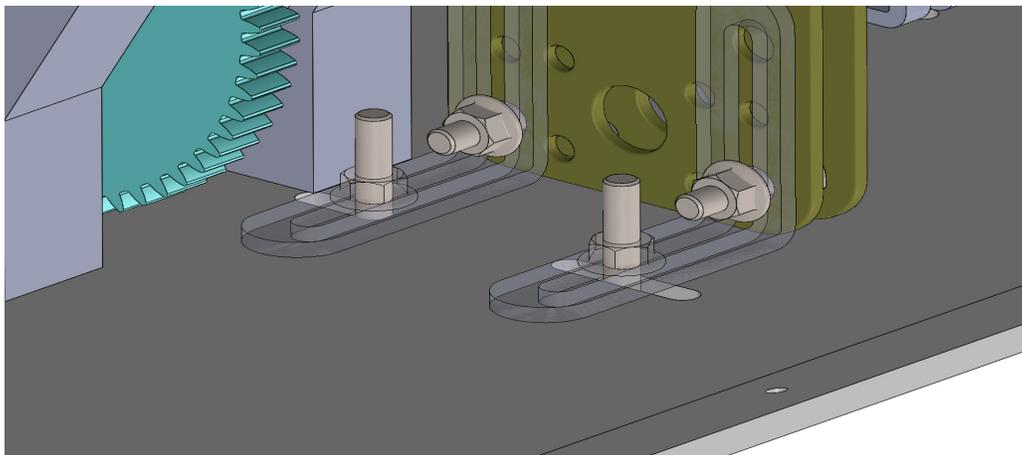


Figure 9: Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 25 teeth pinion gear.

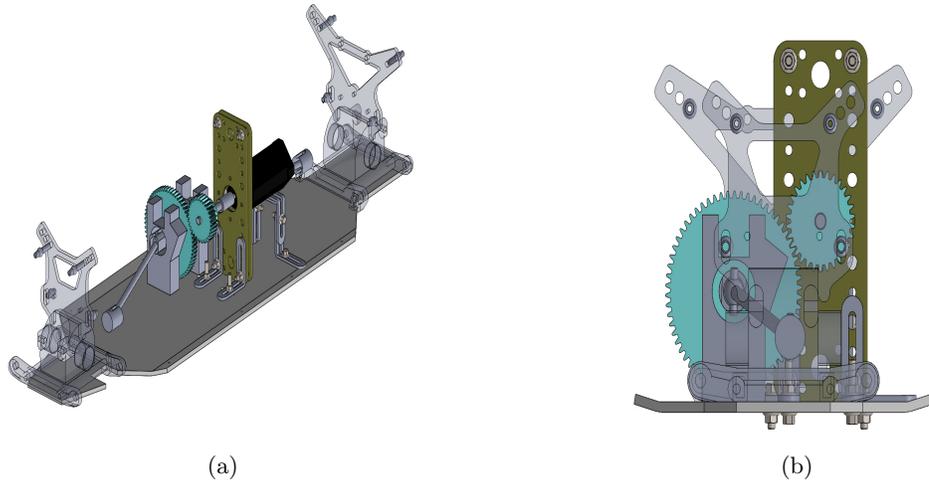


Figure 10: Views of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 30 teeth pinion gear.

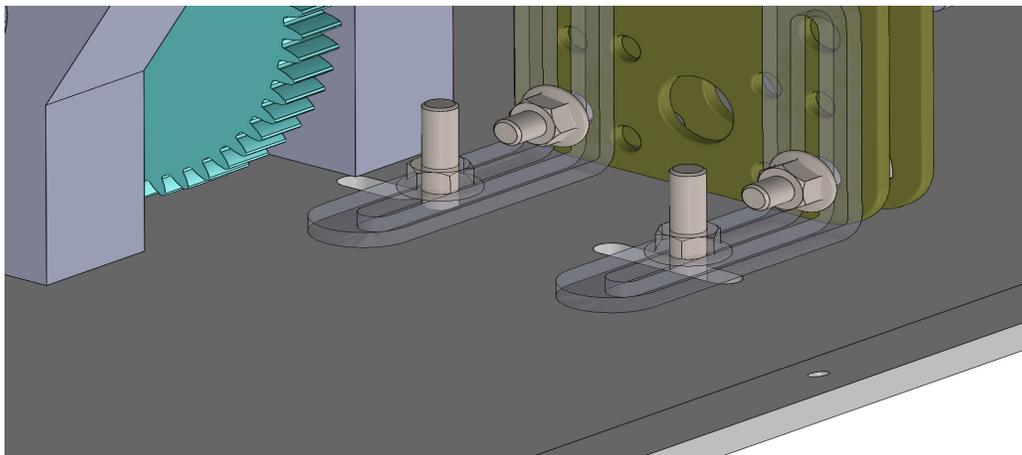


Figure 11: Zoomed view of drive line frame of the electrified Losi 5IVE-T 2.0, without 5:1 gearbox and with a 30 teeth pinion gear.

Performance Comparison among Options In order to compute performance indicators, the parameters of the vehicle's startup version are specified in Table 2. These parameters can be found in [11–13] as well.

Table 2: Parameters of the startup version of the electrified Losi 5IVE-T 2.0.

Notation	Value	Unit	Name
M	25	kg	Total Mass of Vehicle
ω_m	4000	rpm	E-Motor Rated Rotational Speed
T_m	0.25	Nm	E-Motor Rated Torque
τ_g	5	/	Gear Ratio of Gearbox
τ_d	3.31	/	Gear Ratio of Differential
R_t	90	mm	Tire Radius
L	0.61	m	Wheel Base of Vehicle
δ	30	deg	Maximum Steering Angle of Vehicle

Where the total mass of the vehicle M is an approximate value **needed to be updated to the accurate one**, and the tire radius R_t , the wheel base L and the maximum steering angle δ of vehicle were measured by hand.

- Longitudinal Behaviour Comparison:

The gear ratio of transmission τ_t and the total gear ratio τ can be calculated by Equation (1) and Equation (2). Where T_o is the tooth number of the central differential spur gear, while T_i is the tooth number of the motor pinion gear.

$$\tau_t = \frac{T_o}{T_i} \quad (1)$$

$$\tau = \tau_g * \tau_t * \tau_d \quad (2)$$

When the motor is at its rated rotational speed ω_m , the maximum speed of the vehicle V_{max} in km/h can be calculated by Equation (3) and Equation (4). Where ω_t is the tire rotational speed at the maximum vehicle speed in rpm.

$$\omega_t = \frac{\omega_m * 2\pi}{\tau * 60} \quad (3)$$

$$V_{max} = \omega_t * R_t * 3.6 \quad (4)$$

When the motor is at its rated torque T_m , the longitudinal acceleration of the vehicle on a level road a_x in m/s^2 can be calculated by Equation (5) and Equation (6). Where T_t is the torque at tire level in Nm when the DC motor is at its rated rotational speed T_m .

$$T_t = T_m * \tau * \eta_t \quad (5)$$

$$a_x = \frac{\frac{T_t}{R_t} - F_r}{M} \quad (6)$$

In order to simplify the formulas that would be used to compare the performances of various drive line options, the total transmission efficiency η_t , which should be a positive value lower than 1, was assumed to be 1. Similarly, the total resistance force F_r , which should be a positive value higher than 0 N, was assumed to be 0 N. Without making any assumptions, the actual value of the longitudinal acceleration should be smaller.

When the motor is at its rated torque T_m , the acceleration time to reach the maximum speed of the vehicle on a level road t in s can be calculated by Equation (7). Where the starting speed of the vehicle V_0 was assumed to be 0 km/h.

$$t = \frac{V_{max} - V_0}{a_x} \quad (7)$$

This formula makes the assumption that the longitudinal acceleration remains constant throughout the acceleration phase. This assumption is made in order to simplify the calculation, taking into consideration the fact that the reason this indicator was introduced was simply to get a general idea of the value of acceleration time. The value of acceleration time shouldn't be too large, for example, tens of seconds, to avoid wasting too much time to get to the startup point, enough speed of the vehicle for future field tests.

When the motor is at its rated torque T_m , the maximum slope angle on a gradient θ_{max} in deg can be calculated by Equation (8). Where g is the acceleration due to gravity and was assumed to be 9.81 m/s^2 .

$$\theta_{max} = \sin^{-1} \left(\frac{\frac{T_t}{R_t} - F_r}{M * g} \right) \quad (8)$$

Due to the fact that all three drive line alternatives require the removal of the 5:1 gearbox, the τ_g value is reduced to 1. In accordance with the formulas that related to longitudinal behaviour that are discussed before, the longitudinal performance indicators of the three choices can be computed, and the results are presented in Table 3.

Table 3: Longitudinal performance indicators of the vehicle using three new drive line options.

T_p	τ_t	τ	V_{max}	a_x	t	θ_{max}
teeth	/	/	km/h	m/s^2	s	deg
20	3.1	10.261	13.226	1.14	3.22	6.674
25	2.48	8.2088	16.533	0.912	5.04	5.335
30	2.067	6.8407	19.837	0.76	7.25	4.431

The vehicle is able to achieve a higher speed as the tooth number of the motor pinion gear grows; however, the amount of time it takes to reach the maximum speed increases, and the gradeability of the vehicle also decreases. The highest possible speed that can be reached is

sufficient. Among these three possibilities, the maximum acceleration time computed is less than ten seconds, which is acceptable. On the other hand, because the gearbox is removed, the torque at tyre level is divided by five, which results in the slope angle decreasing from 35.53 degrees to a few degrees. This results in a significant reduction in the vehicle's ability to climb gradients.

- Lateral Behaviour Comparison:

When the motor is at its rated rotational speed ω_m , the lateral acceleration of the vehicle a_y in m/s^2 can be calculated by Equation (9), with the maximum steering angle of the vehicle δ .

$$a_y = \frac{V_{max}^2}{L} * \tan(\delta) \quad (9)$$

Within the framework of this formula, it is presumed that the vehicle exhibits a kinematic behaviour that is pure, ignoring issues such as tyre slip and compliance. This behaviour is suited for manoeuvres that take place at lower speeds.

Using the formula that was shown before, it is possible to calculate the lateral acceleration of the vehicle by employing the startup version of the drive line in addition to three new drive line alternatives. The results of this calculation are presented in Table 4.

Table 4: Lateral performance indicator of the vehicle using startup version of drive line and three new drive line options.

Drive Line Option		Lateral Acceleration
Gearbox	T_p	a_y
/	teeth	m/s^2
With Gearbox	20	0.51
Without Gearbox	20	12.775
Without Gearbox	25	19.962
Without Gearbox	30	28.738

The lateral acceleration of the drive line option without gearbox is approximately 25 times that of the startup version of the drive line, which is reasonable because there is a five-fold gap between those two drive lines in terms of the maximum speed of the vehicle because of the removed gearbox, considering that the formula used is suitable for low-speed manoeuvres, for example, the one using the startup version of the drive line. For the three new options, as the tooth number of the motor pinion gear increases, the maximum speed of the vehicle increases. With the same steering angle required, the lateral acceleration increases, which is reasonable. And it means that the vehicle body will be more prone to tilt during field tests of the new three drive lines compared to the startup version, which will make the motion and trajectory of the vehicle inconsistent with expectations because of assumptions.

Final Drive Line To avoid excessive lateral acceleration while ensuring that the maximum speed of the vehicle provides more possibilities for following research, the drive line option with a 20 teeth motor pinion gear is selected.

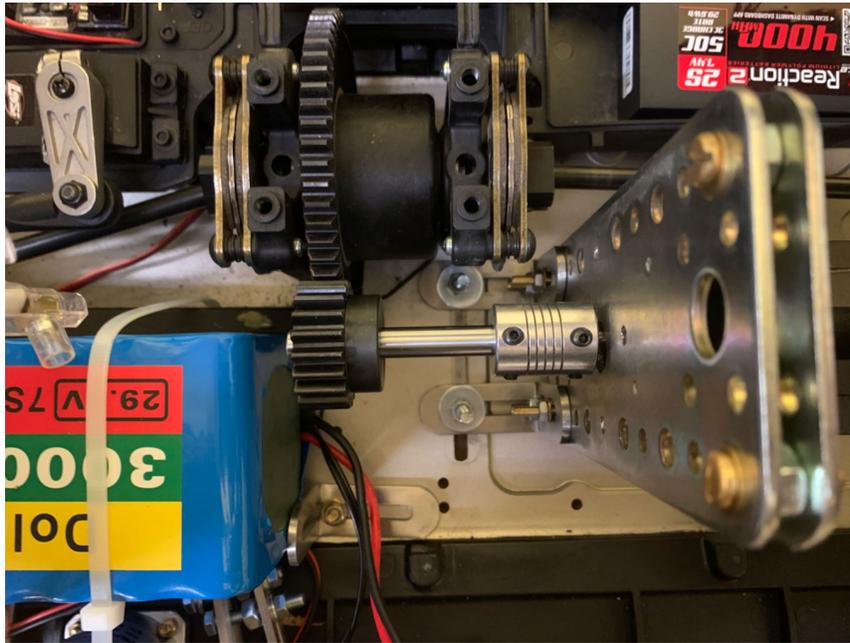


Figure 12: Drive line with a flexible motor shaft coupler.

The motor shaft coupler that was purchased is a flexible one, and it was used for the initial testing of the new drive line. In Figure 12, the new motor pinion gear with 20 teeth, the motor shaft extension rod, and the engine shaft coupler are placed in sequential order from left to right. When all the parts were assembled on the Losi 5IVE-T 2.0 and the speed of the e-motor was controlled with Simulink dashboard blocks to test the new drive line, it was discovered that even if the speed was only a few hundred rpm, the 20 teeth motor pinion gear would be vibrant in the radial direction, accompanied by a noise. Furthermore, after increasing the speed, the 20 teeth motor pinion gear with the motor shaft coupler would even move away from the e-motor in the axial direction.

To solve the above, the final purchased motor shaft coupler is a simple and rigid one, which is the bright blue part in Figure 13. Once again, using Simulink dashboard blocks to control the speed of the e-motor to test the new drive line, it was found that only a minor radial vibration occurred when the speed increased to close to the maximum value that the e-motor could output. The final electrified drive line for follow-up testing can be obtained by securing the inner L-shape mount in the holes near the exterior of the vehicle, as shown in Figure 14a. Also, remove the e-motor supporting structure that was under the e-motor, as shown in Figure 14b. This will enable the entire new drive line assembly to be moved to the interior and front of the vehicle. Through these processes, the motor pinion gear with 20 teeth is able to reduce the distance to the motor shaft coupler while maintaining the appropriate contact with the central differential spur gear. This modification helps to lessen the vibration that is caused by the 20 teeth motor pinion gear when it is in working condition, which ultimately results in an improvement in the overall stability of the new drive line assembly.

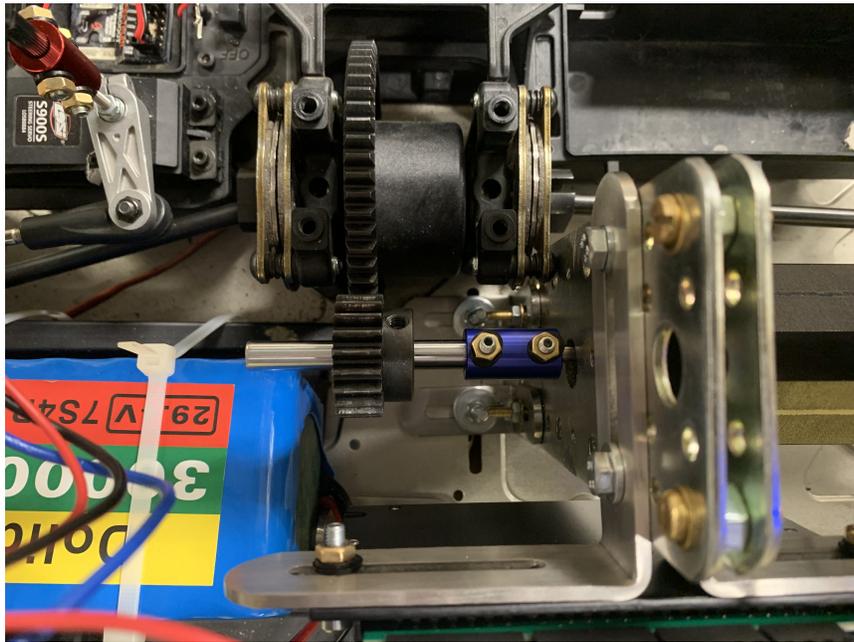


Figure 13: Drive line with a rigid motor shaft coupler.



(a)



(b)

Figure 14: Final supporting structure of drive line frame.

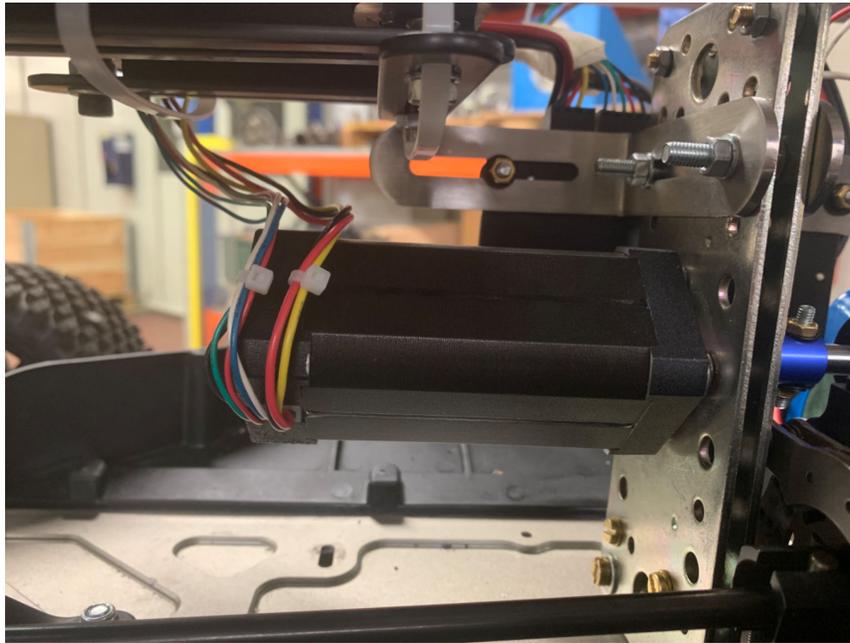


Figure 15: RS PRO Brushless DC e-motor.

The purchased components can be found in the links which are listed in Table 5. The motor shaft extension rod is cut to be 70 mm to avoid engagement with the 24 V main battery.

Table 5: Components of final drive line of the electrified Losi 5IVE-T 2.0 .

Component	Link
20 Teeth 1.5 Module Motor Pinion Gear with 8 mm Hub Diameter	[14]
8 mm Motor Shaft Extension Rod	[15]
5 mm to 8 mm Motor Shaft Coupler	[16]

E-Motor The electric motor used in the final setup of the Losi 5IVE-T 2.0 is the RS PRO Brushless DC e-motor, which can be seen in Figure 15. It is a three phase 24 V e-motor, with 4000 rpm rated speed, 0.25 Nm rated torque and 0.75 Nm maximum peak torque, as indicated in [11]. In addition, it has a hall sensor installed to support a closed loop control strategy, which will be introduced in detail in the following chapters. The wiring colour scheme used to connect the e-motor with the e-motor controller can be found in [11] and will be stated later.

1.1.2 Steering Servo Motor

In contrast to the thorough electrification of the Losi 5IVE-T 2.0 power system, the original version of the steering system, which consisted of the radio controller receiver and the steering servo motor, did not undergo significant changes. In this chapter, the steering servo motor is the main point of

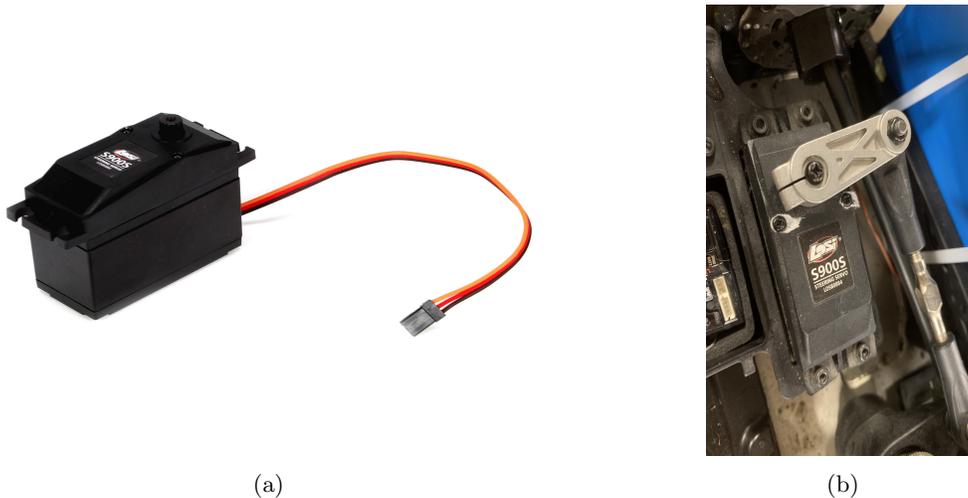


Figure 16: S900S 1/5 Scale Metal Gear Steering Servo and linkage.

discussion. Information on the receiver will be presented in the chapter that is dedicated to the radio controller.

The 6 V servo motor that is used for the Losi 5IVE-T 2.0 is the S900S 1/5 Scale Metal Gear Steering Servo. This servo motor is shown in Figure 16a. It is used to perform rotary movements of the steering servo motor linkage, which is the silver part that is placed on the servo motor shown in Figure 16b. As a result, it is able to control the steering angles at the wheel level through a set of rods. According to [17], the maximum speed of the servo motor is 0.21 seconds per 60 degrees while operating at a voltage of 6 V. In the chapter that is dedicated to the potentiometer used for data acquisition, there are figures to show the relationship between rotary angles at the linkage level and steering angles at the wheel level.

The steering servo motor is connected to the steering port of the receiver in the original version of the Losi 5IVE-T 2.0. This connection is made using an easy-to-use three-pin connector. This allows the receiver to receive steering actions that are conducted on the knob at the transmitter level. In order to facilitate more research on the steering control of the Losi 5IVE-T 2.0 through a real-time target machine, the basic three-pin connector is divided into three distinct wires. These wires are then plugged into particular ports of the IO397 I/O Module terminal board in order to receive control messages from the real-time target machine. Check out the following text for a more in-depth explanation of the connections.

1.2 Control Systems

In the previous chapter, information was provided on the motors that are directly connected to the mechanical structure of the Losi 5IVE-T 2.0. The subsequent chapter, on the other hand, mostly describes the components that make up the higher-level control system. For example, the e-motor controller, which is responsible for controlling the direction and speed of rotation; the radio

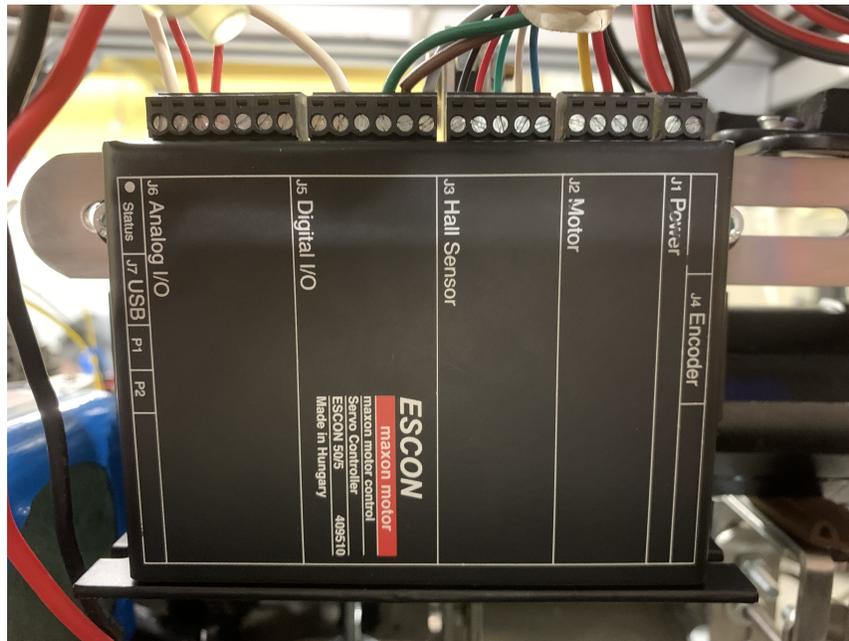


Figure 17: Maxon Motor ESCON 50/5 e-motor controller.

controller system, in which the receiver receives wireless signals from the transmitter for the purpose of controlling the operating state of the Losi 5IVE-T 2.0; the Speedgoat Baseline Real-time Target Machine, which is responsible for executing the Simulink real-time models; and the Wi-Fi router, which allows for wireless control without the need to connect the host PC to the Speedgoat.

1.2.1 E-Motor Controller

Introduction An e-motor controller is a device used to control the operation of an electric motor, mainly by adjusting the current, voltage, phase, and other parameters of the motor to achieve control over the start, stop, speed regulation, forward and reverse rotation, and other states of the motor. As mentioned earlier, the Losi 5IVE-T 2.0 is equipped with a three-phase 24 V brushless DC motor that includes a built-in hall sensor, making the Maxon Motor ESCON 50/5 e-motor controller, as is shown in Figure 17, an optimal option.

The ESCON 50/5, as stated in [18], is a compact and robust 4-quadrant PWM servo controller designed for precise control of permanent magnet-activated brushed DC motors or brushless EC motors with a power output of up to around 250 Watts. The ESCON 50/5 is designed to be controlled by an analog set value and includes extensive analog and digital I/O capabilities, which can be seen in Figure 18a. The operating modes that are available, including closed-loop speed control, open-loop speed control, and current control, satisfy the most demanding requirements.

This thesis focuses more on taking PWM signals, which have duty cycle limits from 10 % to 90 %, as the reference to command the e-motor, and thus to enable the speed control. Thanks to the

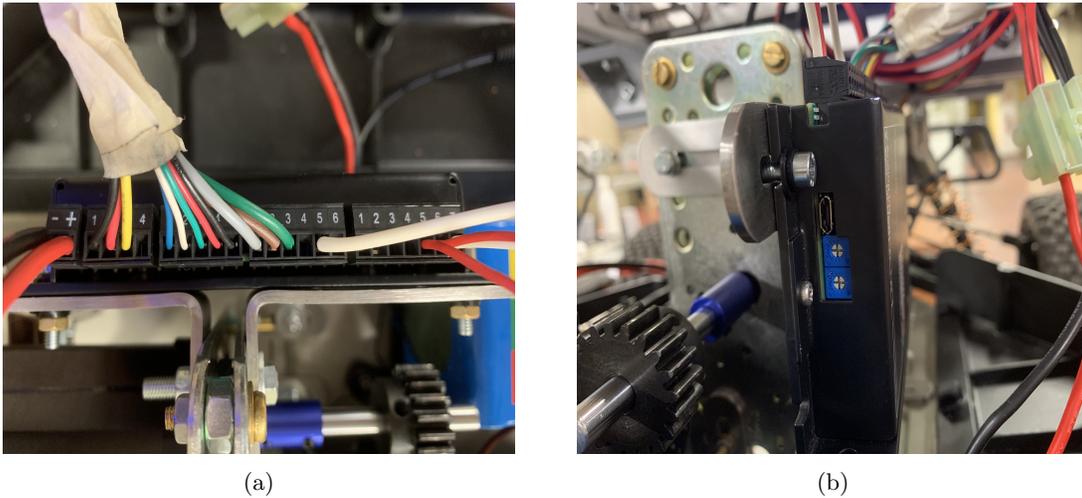


Figure 18: Maxon Motor ESCON 50/5 e-motor controller interfaces.

built-in hall sensor of the e-motor, closed-loop speed control was designed and achieved to solve the problem that occurred in the radio controller control modality.

Moreover, the ESCON 50/5 can be configured through a USB interface, as is shown in Figure 18b, using the graphical user interface, built for Windows PCs, called ESCON Studio. Detailed information about the configuration procedure will be presented in the following chapter.

Connections and Wiring The ESCON 50/5 e-motor controller has a distinct advantage over other available controllers because of the pluggable screw-type multi-pole terminal blocks with 3.5 mm pitch. Each block is printed with numbers, which can be seen in Figure 17, to facilitate dealing with lots of cables, correlating cables to signals, and helping to group signals into various categories that correspond to the type of connection printed on the controller surface, as is shown in Figure 18. In this thesis work, six out of seven connections are used, no matter what type of control modality is activated.

J1 Power This 2-pole block is supposed to be linked to a power supply unit. In this Losi 5IVE-T 2.0 case, considering the required voltage of the e-motor, a 24 V battery is used. Pin 1 of the J1 block is the ground of the supply voltage, which should be connected to the ground (black) wire of the battery. Pin 2 is for power supply voltage, which should be connected to the positive (red) wire of the battery.

In addition, to ensure the safety of the Losi 5IVE-T 2.0 when it's not doing any tests, unplugging this J1 block is recommended.

J2 Motor Since the e-motor selected is a brushless DC (EC) motor, the 4-pole J2 block is to be connected with the three windings of the e-motor. Based on what is stated in [11] and [18], the

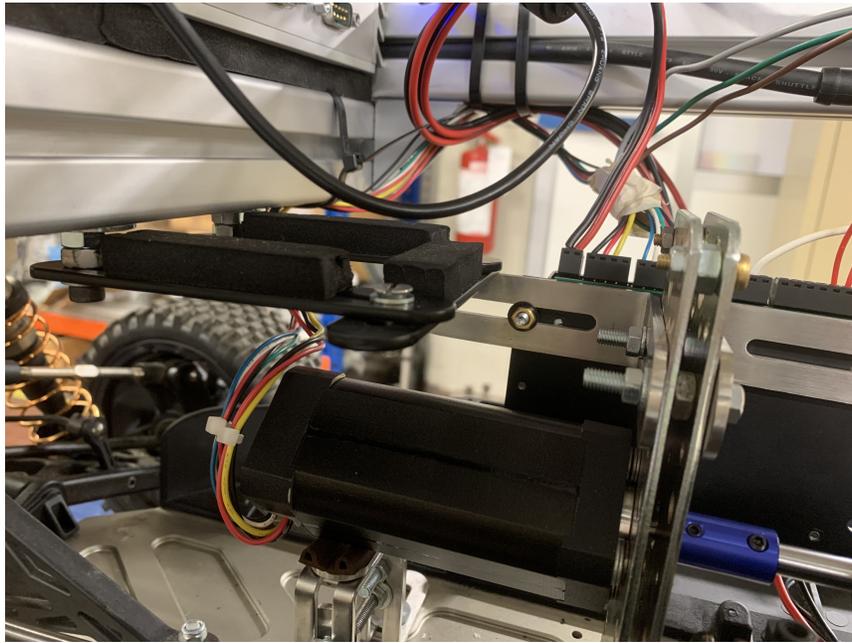


Figure 19: Maxon Motor ESCON 50/5 e-motor controller.

black, red, and yellow wires are plugged into Pin 1, 2, and 3 separately. Pin 4 is left unplugged because there is no motor shield signal routed from the e-motor.

J3 Hall Sensor This 5-pole block is intended to allow the e-motor hall sensor to assist with developing the closed-loop speed control scheme. Based on what is stated in [11] and [18], the blue, white, and green wires are plugged into Pin 1, 2, and 3 separately to correctly link the hall sensor inputs to the controller. And Pin 4 and 5 provide the supply voltage and ground for the hall sensor.

In Figure 19, two sets of wires routed from e-motor and then plugged into dedicated e-motor controller blocks can be found.

J5 Digital I/O There are 6 poles of this block. Pin 1 is the only pin capable of processing any PWM signal with a frequency between 10 Hz and 5 kHz and a duty cycle between 10% and 90%. Thus, in this project, it's used to receive the regulated throttle duty cycle signal (grey) generated by Speedgoat to manage the rotating speed of the e-motor, thereby controlling the speed of the Losi 5IVE-T 2.0. In regard to the e-motor's rotation direction, Pin 2 is employed to manage the regulated digital signal associated with clockwise rotation (brown), with logic 0 values generally falling below 1.0 V and logic 1 values generally exceeding 2.4 V. On the other hand, Pin 3 is used for counter-clockwise rotation of the e-motor (green). It means that if the logic value of Pin 2 is 0 and that of Pin 3 is 1, the Losi 5IVE-T 2.0 is in the forward mode. As for Pin 5, it's mandatory to link it to the ground (white) from Speedgoat, since the connection blocks of the controller don't share the same ground.

Pin 1 and Pin 2 can only handle input digital signals, while Pin 3 and Pin 4 can also output digital signals. And Pin 6 is able to output an auxiliary voltage. However, these capabilities are not interested in this thesis study.

J6 Analog I/O This 7-pole block has two sets of pins. The first four pins are able to translate two analog signals from other electronic components to two digital signals using a 12-bit A/D converter and are left unplugged because of the purpose of this project.

The fifth and sixth pins are able to output two analog signals, between -4 VDC and +4 VDC, transferred from digital signals inside the controller by a 12-bit D/A converter. In the closed-loop speed control strategy, these two analog signals are selected in the ESCON Studio software to be the averaged actual current and speed of the e-motor to be used as real-time feedback references in the speed control logic subsystem in the Simulink model. Same as previously discussed, the last pin should be linked to a ground (white) from Speedgoat.

J7 USB The connection port of J7 USB is a USB-type micro-B female, different from the above connections. It is used to configure the e-motor controller in the initial phase. Once the configuration file is uploaded to the controller, this port can be left unplugged.

1.2.2 Radio Controller

Introduction The conventional radio controller system, as is shown in Figure 20, is consist of the Spektrum SPM5000 DX5R 5CH DSMR Transmitter and the Spektrum SR6200A 6CH DSMR AVC Receiver.

The Spektrum SPM5000 transmitter has a physical trigger to control the throttle and brake functions of the Losi 5IVE-T 2.0 and a physical knob to control the steering function. Also, it has a display screen on the top of it. The most relevant information is the status of the transmitter and receiver battery levels in this project to ensure successful communication between two devices. More attention is paid to the Spektrum SR6200A receiver since it takes on the responsibility of transferring command signals from the user to the Losi 5IVE-T 2.0.

Moreover, thanks to the work done by Luca Biando, as stated in [9], the steering and throttle PWM signals coming out of the receiver have a frequency of 91 Hz. The PWM duty cycle of the steering signal is between 10 % and 17.2 % and that of the throttle signal is between 8.4 % and 19 % with the rest positions both at 13.6 %.

Connections and Wiring In the control modality that uses the radio controller system, the components to be indirectly controlled are the S900S 1/5 Scale Metal Gear Steering Servo and the RS PRO Brushless DC e-motor. The reason why the control is indirect is because the goal of this thesis work is to utilise a real-time target machine but not the receiver of the radio controller to directly send control commands to the servo motors. In other words, for the original setup of the Losi 5IVE-T 2.0, the receiver of the radio controller is to receive control commands from the transmitter and transfer them directly to the steering and throttle valve servo motors. While for the updated electrification version of the Losi 5IVE-T 2.0, the commands is transferred by the receiver to the Speedgoat Baseline real-time target machine.

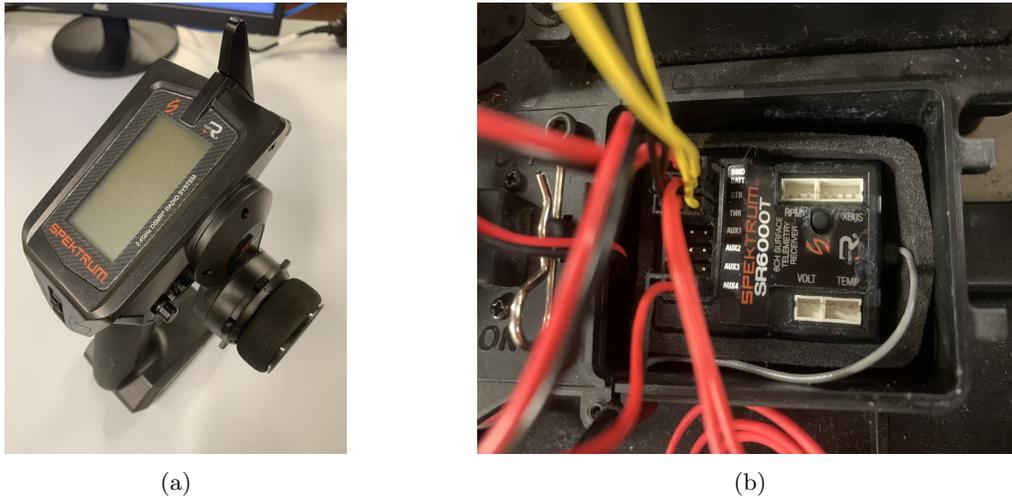


Figure 20: Spektrum radio controller of Losi 5IVE-T 2.0.

Therefore, the previous connections between receiver and servo motors should be updated to a new version. In Figure 20b, there are seven three-pin ports dedicated to different purposes. The first one, with 2 pins utilised, is aimed at being connected to a power supply unit, which is a 2S 4000 mAh 7.4 V 50C LiPo battery in this project, to obtain the required voltage for working. The next two ports are used to send steering and throttle commands received from the transmitter to Speedgoat through specifically assigned pins on the Speedgoat IO397 I/O Module terminal board. In detail, the black wires should be plugged into the ground ports, and the yellow wires carry the command signals. The red wires are left unplugged because the first three-pin port is already connected to an external battery. Pin numbers used on the Speedgoat IO397 I/O Module terminal board will be stated in the chapter that is dedicated to various control modalities.

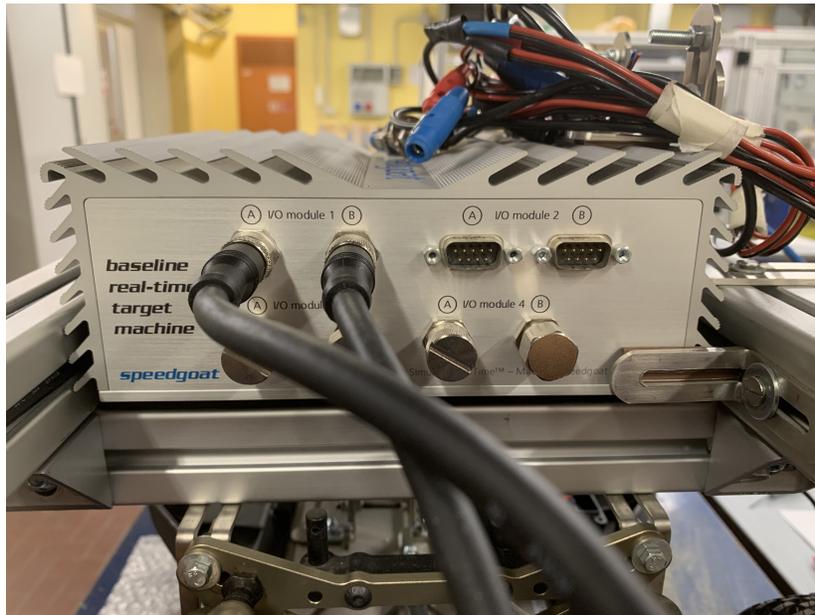
1.2.3 Speedgoat Baseline Real-Time Target Machine

Introduction The real-time target machine used in this project is the Speedgoat Baseline. It has a number of advantages over other real-time target machines that are now available. Upon initial inspection, it appears to be a compact system and is built to handle high levels of vibration and shock, which is exactly the case when installing it on a moving scaled vehicle. Additionally, it is equipped with a passive heatsink, which allows for completely silent fanless operation, as stated in [19].

The Speedgoat Baseline also demonstrates efficient performance. It smoothly combines MATLAB and Simulink workflows using Simulink Real-Time and allows real-time testing of control methods using various supported I/O modules. In particular, it supports real-time parameter adjustment and real-time signal viewing. This makes it appropriate for rapid control prototyping (RCP) on desks and in the field, as well as for smaller-scale desktop hardware-in-the-loop (HIL) simulations.

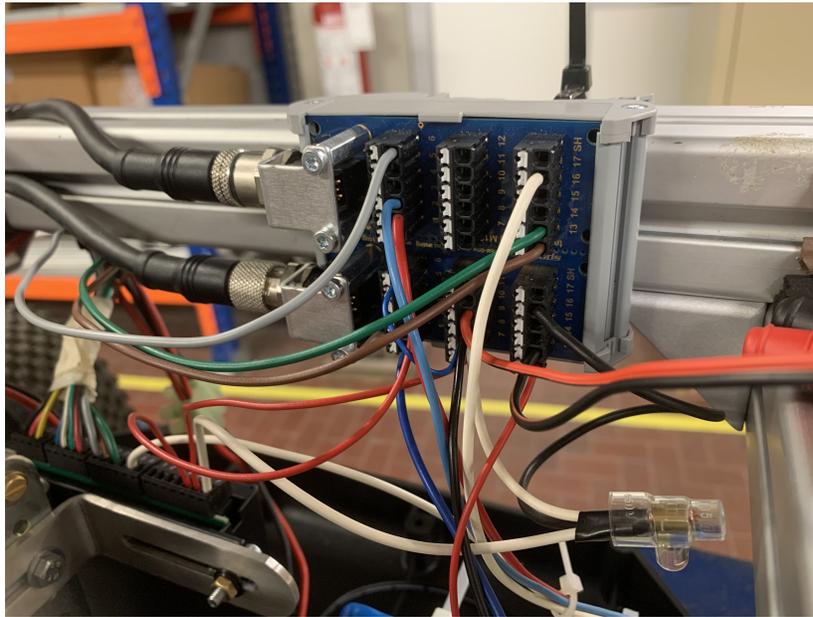


(a)



(b)

Figure 21: Speedgoat Baseline real-time target machine interfaces.



(a)

Block Parameters: Setup

setup_fpga_3 (mask) (link)

Speedgoat Driver Block

IO397-50k - fpga_setup_IO397-50k

© 2007 - 2023, Speedgoat GmbH, www.speedgoat.com

Module setup: IO397-50k Pull Resistor

IO3xx I/O Module: IO397-50k

IO3xx I/O Module ID: 1

Configuration File (Bitstream): speedgoat_IO397_50k_RCP.mat

Revision: 5.000000

Timestamp Configuration File: 01/03/23 15:48:28

FPGA Frequency: 75 MHz

Tick Time: 13.33 ns

PCI Slot (-1: autosearch): -1

Buttons: OK, Cancel, Help, Apply

Pinmapping speedgoat_IO397_50k_RCP

Pin	Digital I/O	Code	Module	Functionality	Direction	Transceiver	Port	Pull Resistors
1b				0V				pull-up 3.3 VDC
2b				5V				
3b	1	1		PWM - A	OUT	TTL		
4b	2	1		PWM - B	OUT	TTL		
5b	3	1		PWM - Trigger	OUT	TTL		
6b	4	2		PWM - A	OUT	TTL		
7b	5	2		PWM - B	OUT	TTL		
8b	6	3		PWM - A	OUT	TTL		
9b	7	3		PWM - B	OUT	TTL		
10b	8	1		QAD - A	IN	TTL		
11b	9	1		QAD - B	IN	TTL		
12b	10	1		QAD - C/Index	IN	TTL		
13b	11	11		DIO	IN/OUT	TTL		
14b	12	12		DIO	IN/OUT	TTL		
15b	13	13		DIO	IN/OUT	TTL		
16b	14	1		Interrupt input	IN	TTL		
17b				GND				

Terminal board A: analog I/O

Pin	Functionality	Analog converter
1a	Analog Input 01 (+)	ADC
2a	Analog Input 01 (-)	
3a	Analog Input 02 (+)	
4a	Analog Input 02 (-)	
5a	Analog Input 03 (+)	
6a	Analog Input 03 (-)	
7a	Analog Input 04 (+)	
8a	Analog Input 04 (-)	
9a	Analog Output 01	DAC
10a	Analog Output 02	
11a	Analog Output 03	
12a	Analog Output 04	
13a	GND	
14a	GND	
15a	0V	
16a	5 VDC	
17a	GND	

(b)

Figure 22: IO397 I/O Module terminal board and pin mapping.

In this thesis study, the IO397 I/O Module plays an important role. The IO397 I/O module is a mPCIe-compatible, configurable and Simulink-programmable FPGA I/O module, offering 50k logic cells, 4 ADC input channels, 4 DAC output channels, 14 ESD-protected TTL Digital I/O lines, as stated in [20].

The high-performance FPGA boards, like the IO397 I/O Module, offer a convenient and efficient solution for FPGA programming supported by Speedgoat configuration files and the Speedgoat I/O Blockset. Users create algorithms using classic Simulink modelling, use HDL Coder to convert the Simulink model into Verilog and VHDL code, and then implement the code on FPGA devices. No programming knowledge is necessary for the entire process. This system simplifies hardware language programming with Simulink modelling, offering a user-friendly approach with minimal complexity and optimal performance.

There are three configuration files in the IO397 configuration package, including Rapid Control Prototyping (RCP) configuration, Hardware in-the-Loop (HIL) configuration, and Communication configuration. In this thesis study, the RCP configuration file is used because it has three PWM channels which can be used for controlling the e-motor and the steering servo motor.

Connections and Wiring In Figure 21a, two ports are connected to the left interface of the Speedgoat Baseline real-time target machine. The one on the right is aimed at obtaining the required voltage, between 8 VDC and 36 VDC, from an external power supply unit, which is a 7S4P 30000 mAh 24 V battery in this project, through a D-Sub 2 power-way female connector. The left one is used to connect the Speedgoat Baseline to a host personal computer through an Ethernet cable to upload Simulink models.

On the right interface of the Speedgoat Baseline, there are four groups of I/O module connection ports, which are shown in Figure 21b. The first group is used to be connected to the I/O397 I/O Module terminal board, which is shown in Figure 22a, through two M12 17-pos cables.

The pin mapping of the RCP configuration file can be seen in Figure 22b. Depending on the control modality used, connections between the wires of other components and the pins of the IO397 I/O Module terminal board can be different because different channels are activated for different purposes of signals. Therefore, wire connections shown in Figure 22a are not correct for all the control modalities. Detailed correlations between cables and pins will be stated in the chapter dedicated to control modalities.

1.2.4 Router

For the control modality with the conventional radio controller system, thanks to the wireless communication between the transmitter and the receiver that is installed on the Losi 5IVE-T 2.0 and the ability of the Speedgoat Baseline to download and run a defined Simulink model in real-time as the startup application, a real-time wireless control can be done. However, for control modalities developed with dashboard blocks or predefined vectors, it's necessary to manually adjust some inputs in the Simulink model. That means the Ethernet cable cannot be disconnected during the whole simulation time period, which limits the capability of the Losi 5IVE-T 2.0 to perform tests in the field and the possibility of further study on it.



Figure 23: TP-LINK TL-MR3020 router interfaces.

In order to solve this problem, a wireless connection between the Speedgoat Baseline real-time target machine and the host personal computer is developed by introducing a router. The router used is the TP-LINK TL-MR3020 router, as is shown in Figure 23a and stated in [21], which is a really small and lightweight but efficient device. The Ethernet cable is linking the target machine and the router instead of the host personal computer. And the host is connected to the router through WiFi to enable the wireless connection, in which the router can be seen as an intermediate interface thanks to the access point operating mode selected, which can be seen in Figure 23b.

The router is powered on by one of the USB ports on the Speedgoat Baseline. In addition, the router is placed on top of the right bar to ensure the strength of the WiFi signal.

1.3 Power Supply System

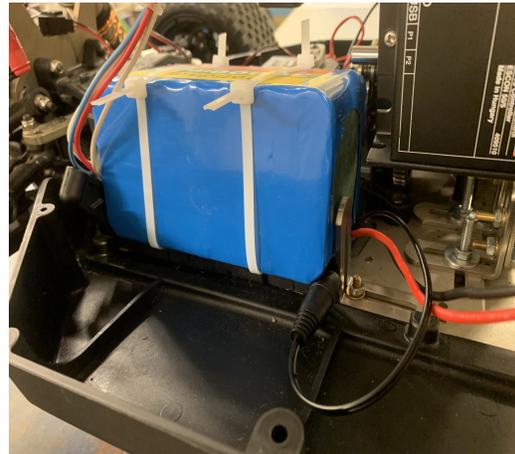
Within the final setup of the Losi 5IVE-T 2.0, there are two distinct types of batteries that are utilised to provide the necessary voltages to various electronic components. These components include the e-motor controller, Speedgoat, receiver of the radio controller system, and IMU. The following chapter will go over their specifications, installation, and wiring information.

1.3.1 24 V Main Battery

To satisfy the voltage requirement of the e-motor, which has been described in the previous chapter, the available space after removing the Zenoah 32cc gasoline pull-start engine, the acceptable weight with respect to the total weight of the Losi 5IVE-T 2.0, and the sufficient capacity to carry out all the required tests, the 7S4P 30000 mAh 24 V battery with 130x80x66 mm dimensions is selected. According to Figure 24a, it is initially equipped with a charging connector that is utilised to establish a connection with the 29.4V 2A charger. Additionally, it has a single set of positive (red colour) and ground (black colour) wires that is used to supply voltages to various other components.



(a)



(b)

Figure 24: 7S4P 30000 mAh 24 V battery.

In this thesis work, the battery is supposed to provide power to the e-motor controller as well as the Speedgoat Baseline real-time target machine. In consideration of this, the initial set of wires is divided into two distinct sets. The first one is connected to the power plug J1, and the second one is linked to a D-Sub 2 power-way female connector. Also, based on the positions of the two components the 24v battery should power, the relative position of it is shown in Figure 24b.

1.3.2 7.4 V Secondary Battery

Apart from the 24 V main battery, a 2S 4000 mAh 7.4 V 50C LiPo battery, which can be seen in Figure 25, is used in the setup of the Losi 5IVE-T 2.0 to supply voltage. According to the use of this 7.4 V battery, control modes can be divided into three categories, which will be discussed in the following chapter that is devoted to a variety of control modalities.

For the modality that make use of dashboard blocks constructed in the Simulink model to control the steering and throttle functions of the Losi 5IVE-T 2.0, the 7.4 V secondary battery is not necessary to be used. For the modality that uses the conventional radio controller, the 7.4 V secondary battery is mandatory to provide voltage required by the Spektrum SR6200A receiver to receive control signals from the Spektrum SPM5000 transmitter. For the modality that uses predefined trajectory vectors generated in MATLAB scripts, the 7.4 V secondary battery is connected to the IMU by means of an EC5 battery serial connector in order to provide an acceptable voltage, considering that the input voltage range of the IMU is from 5 V to 36 V as indicated in [22] .

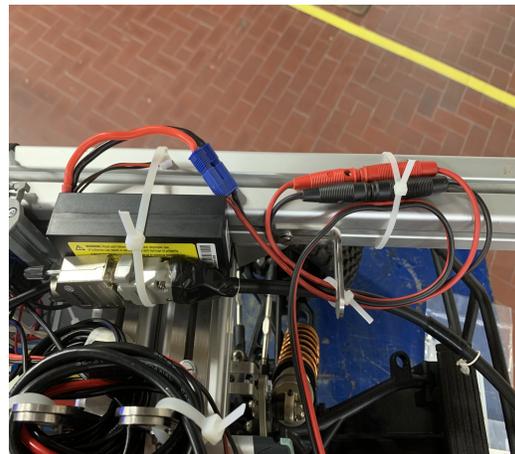
Furthermore, the locations of the 7.4 V battery in the last two control modalities are not the same. For the radio controller modality, the secondary battery is placed in a dedicated box closed to the receiver, which can be seen in Figure 26a. While for the predefined vector modality, it is tightened on the left longitudinal bar, as is shown in Figure 26b. This is to guarantee that the other sensitive cables of the entire data acquisition system are not curved to an excessive degree.



Figure 25: 2S 4000 mAh 7.4 V 50C LiPo battery.



(a)



(b)

Figure 26: Two different positions of 2S 4000 mAh 7.4 V 50C LiPo battery.

2 Controller Configuration and Design Software

2.1 ESCON Studio

The ESCON Studio software is provided by Maxon to help users configure their e-motor controllers using personal computers, thanks to the J7 USB port. After opening the software, the useful startup wizard tool window immediately appears on the software interface, which is shown in Figure 27.

Instead of directly configuring the controller, it's suggested to use a virtual ESCON 50/5 IPC2 controller to generate the configuration file and then upload the tuned version to the e-motor controller. After selecting and adding lots of parameters during the entire configuration procedure, the regulation tuning tool window will show up. The auto-tune feature is recommended. It can automatically detect parameters that are not correctly defined and add them to the configuration based on the plant identification.

Finally, the controller monitor section, as can be seen in Figure 28, provides an overview of digital/analog inputs/outputs, controller status, and motor/sensors status in real-time. It also provides a summary of all properties configured and the possibility to change all the settings. Some important parameters of the final version of the configuration file uploaded to the actual ESCON 50/5 e-motor controller are listed here:

- "Motor" type chosen is "maxon EC motor", although the e-motor used in the Losi 5IVE-T 2.0 is named a brushless DC one.
- "Speed Sensor" type chosen is "Available Hall Sensor" to enable the closed-loop speed control strategy.
- "Mode of Operation" is "Speed Controller (Closed Loop)".
- "Enable" type should be "Enable CW & CCW", with CW enabled from digital input 2 and CCW enabled from digital input 3, which correspond to the wiring of the J5 block previously discussed.
- "Set Value" type chosen is "PWM Set Value" to enable the possibility of cooperating with Speedgoat.
- "Digital Inputs/Outputs" are selected based on the wiring of the J5 block.
- "Analog Outputs" are selected based on the wiring of the J6 block.

To save the final version of the configuration file in ".edc", click "File" and "Upload Parameters". When it's needed to configure the controller, first make sure that the controller is properly connected to the personal computer through the USB port, then click "File" and "Download Parameters" to upload the configuration file to the controller.

2.2 MATLAB and Simulink

To support Speedgoat Baseline real-time target machine, there are many Mathworks products and Speedgoat products should be downloaded and installed (such as MATLAB, Simulink, MATLAB Coder, Simulink Coder, Simulink Real-Time, HDL Coder, Fixed-Point Designer, Speedgoat I/O

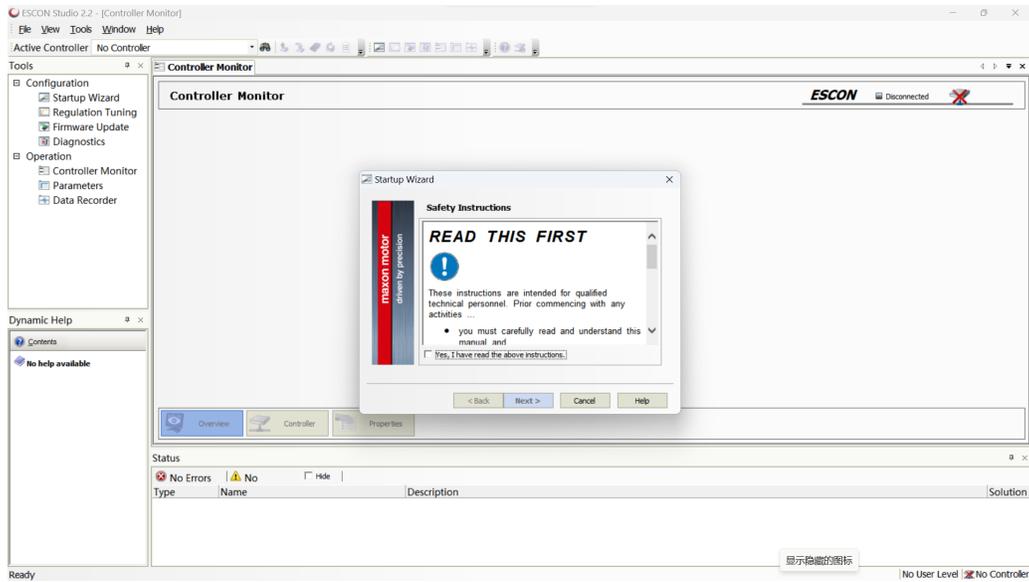


Figure 27: ESCON Studio interface with the startup wizard window opened.

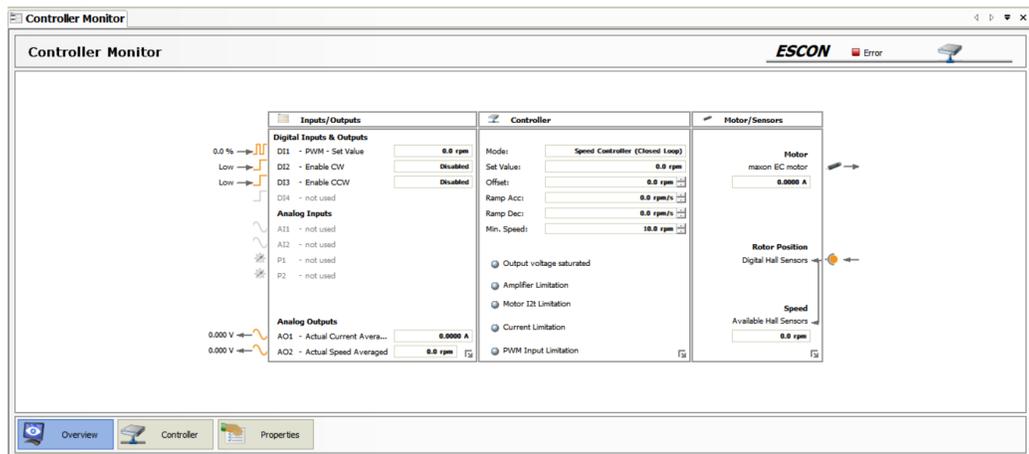


Figure 28: ESCON Studio interface with controller monitor section activated.

Blockset, Speedgoat HDL Coder Integration Package, etc.). In particular, to generate real-time applications, a valid compiler which is included in the Simulink Real-Time Target Support Package, should be installed on the host personal computer.

The first step to take after installing all the necessary products is to establish communication between the host personal computer and the Speedgoat Baseline. The key point is that the host must have a static IP address that can be recognised by the target machine. Therefore, open the Windows "Network and Sharing Center" and configure the "Internet Protocol Version 4 Properties" by manually defining the IP address and subnet mask for the host. Then, open the "Simulink Real-Time Explorer", select the "Target Configuration" section, and manually define the IP address for the target. Click the "Disconnected" button on the left upper corner. If the button changes to "Connected", means that the communication has been successfully established. Furthermore, after installing MATLAB product updates or new versions of the Speedgoat I/O Blockset, or after changing to a different MATLAB version, the "Update Software" button in the "Target Configuration" section should be clicked to perform the software update of the target machine.

Thanks to the various and dedicated Simulink Speedgoat driver blocks provided by the Simulink Real-Time product, Simulink real-time models can be easily established to enable communication between the models and the I/O modules installed in the target machine. Here are some steps to create a real-time model from a regular Simulink model:

1. Open the "Library Browser", find the "Simulink Real-Time: Speedgoat I/O Blockset" library, and open the subset of the I/O module that is activated;
2. Drag and drop the needed driver blocks into the regular model;
3. Connect the I/O ports to the regular blocks that are able to support the code generation of the real-time applications in the model (such as "Scope", "1-D Lookup Table", "MATLAB Function", etc.);
4. Configure I/O settings in the dialog fields of the driver blocks, taking into consideration the cable and wire connections of the I/O modules.

In this thesis study, the IO397 I/O Module is used. So, some of the driver blocks from the IO3xx subset of the "Simulink Real-Time: Speedgoat I/O Blockset" library used to create three main real-time applications that match the control modalities, which are shown in Figure 29, are listed here:

- "Setup" block is the starting point of building a Simulink real-time model. One and only one "Setup" block must be contained in the model for one configurable I/O module. In the dialog field of this block, after selecting the right type of module, one of the available configuration files should be activated, depending on the operating purpose of the module. In this project, the "speedgoat_IO397_50k_RCP.mat" RCP configuration file is activated. By clicking the "Pin Mapping" button, a window showing an overview of the functionality and channel number of each pin on the I/O module terminal board will be opened to help users understand the correct correlation between signals and physical pins, as is shown in Figure 22b.
- "PWM generation (5)" block is aimed at generating PWM signals and outputting them to the other consuming components through the B part of the IO397 terminal board. In this project,

two PWM signals used to control the e-motor and the steering servo motor are needed. So, two single-input "PWM generation (5)" blocks are employed separately for the throttle and steering design pathways. In particular, the "Sample time" parameter in the dialog field of this block should be carefully defined, considering the PWM frequency ranges supported by the consuming components. After calibrating, the final PWM frequency parameter to define the "Sample time" is set to be 100 Hz.

- "Digital Output" block is to output logic values, 0 or 1, to control the e-motor's rotation direction in this thesis study. Two blocks are used to be linked to Pin 2 and Pin 3 of the J5 Digital I/O block of the e-motor controller separately to manage the activation of clockwise and counter-clockwise rotations of the e-motor. The relationship between the logic values and the moving direction of the Losi 5IVE-T 2.0 will be stated in detail in the following chapter.
- "IO397 Analog Input" block is aimed at reading analog signals. In this project, the signals could be the averaged actual current and speed of the e-motor selected in the ESCON Studio software and the instant potentiometer voltage. One and only one "IO397 Analog Input" block can be contained in the model and the number of output ports for one configurable I/O module is limited by four considering the I/O module used is the IO397, which has 4 ADC input channels. Based on the voltage range of the signals, the "Analog input range" parameter can be freely selected among seven options.
- "IO397 Analog Output" is aimed at writing analog signals. In this project, the signals could be the feeding voltage designed for the potentiometer and the trigger voltage designed for the Scadas XS, [27]. One and only one "IO397 Analog Output" block can be contained in the model and the number of output ports for one configurable I/O module is limited by four considering the I/O module used is the IO397, which has 4 DAC output channels. Based on the voltage range of the signals, the "Analog output range" parameter can be freely selected among six options.

In the dialog fields of the last four blocks, there is a "Channel vector" parameter that should be correctly defined for different control models, taking the actual wire connections into account. Check out the chapter that is dedicated to the explanation of Simulink real-time models developed for different control modalities.

Once a Simulink real-time model has been run on the target machine and the "External" indicator appears at the left bottom corner of the Simulink interface by clicking the "Run on Target" button, the real-time application generated from the model is automatically uploaded to the file system of the target machine. One can load many real-time models on the target, but only one of them can be run at a time. Also, the target machine supports recognising one application as the startup one. This function benefits the control modality that uses the conventional radio controller.

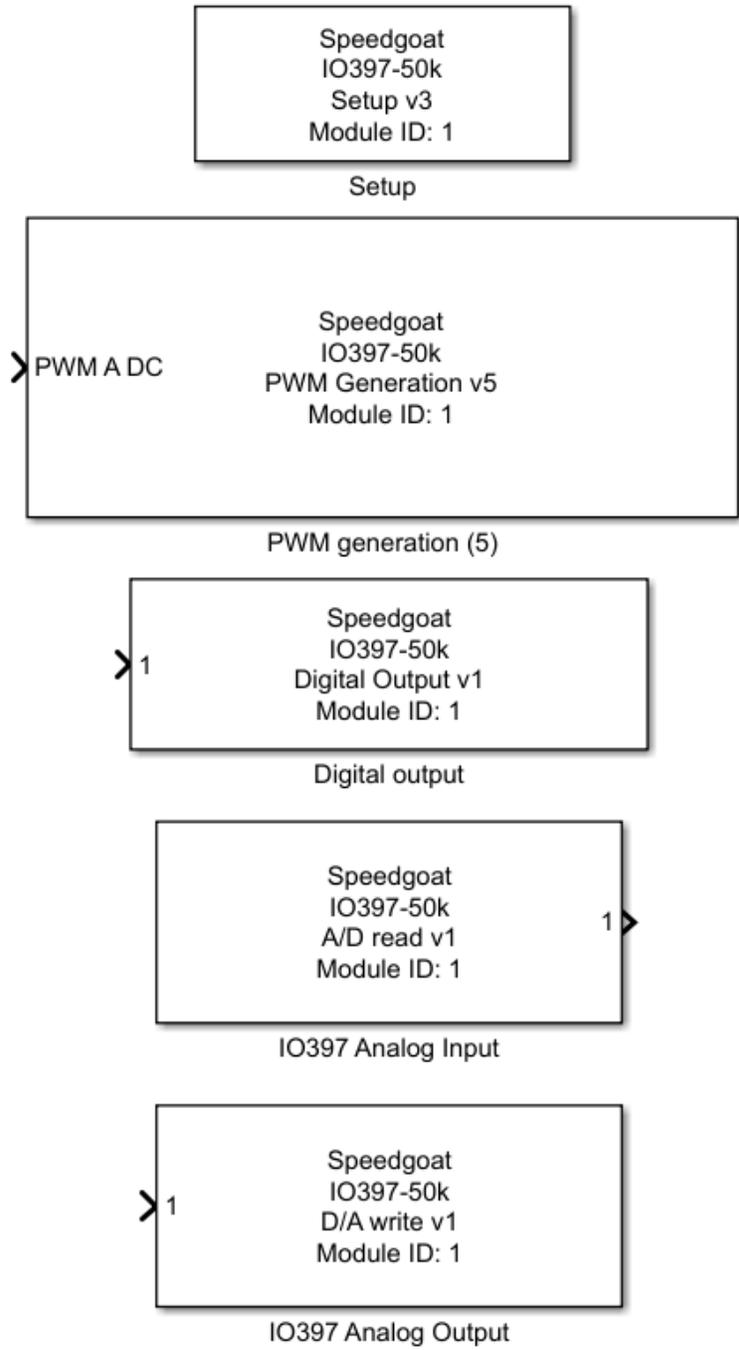


Figure 29: IO397 I/O Module driver blocks.

3 Experimental Tests

In this chapter, the main points are the design of the control modalities using MATLAB and Simulink workflows and the experiments that are performed in the field to validate the control modality that uses the predefined vectors to control the steering and throttle functions of the Losi 5IVE-T 2.0. To assist the experiments carried out in the field, the data acquisition system, including a potentiometer, a GPS antenna, an IMU, and a Scadas XS, is first introduced.

3.1 Data Acquisition System

Thanks to the "To Workspace" block provided in Simulink, it's possible to record several signals in real-time to help with the data post processing task. However, the number of available feedback signals from Simulink is not enough. Thus, several external dedicated sensors, which will be explained in the following text, are installed on the Losi 5IVE-T 2.0 to present a relatively complete overview of its dynamic behaviours.

3.1.1 Potentiometer

Introduction The input steering angle signal can be easily recorded by using the "To Workspace" block, while the actual steering angles at wheel level can be easily affected by the uneven field ground, knowing that the e-motor rated torque is a very low value, 0.25 Nm, and the gradeability is poor after removing the gearbox. That means the actual steering angle signal at wheel level could be very different from the desired one, and the other dynamic statuses are possibly affected to some extent. Thus, a system dedicated to the measurement of the real steering angles should be designed.

As stated in the previous chapters, the steering angles performed at the two front wheels are controlled by the S900S 1/5 Scale Metal Gear Steering Servo through the movement transmission system, which contains the linkage linked to the steering servo motor and the connection rods from the linkage to the wheels. Because the IO397 I/O Module can read analog signals like voltages by using an "IO397 Analog Input" block in the control model, a simple 100 kOhm potentiometer with one rotary turn, as is shown in Figure 30a, is chosen to make a steering angle measurement system that can be installed on the Losi 5IVE-T 2.0 and to allow getting the actual real-time steering angles at the wheel level.

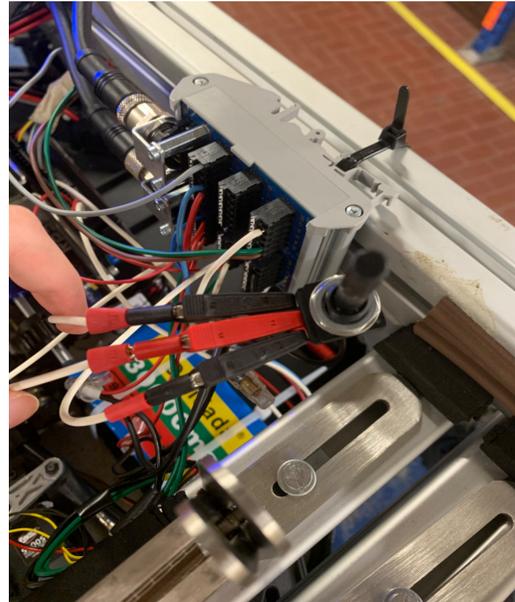
The main idea is to feed the potentiometer with 5 V by using the IO397 module and then route the output voltage from the potentiometer back to the IO397 module. The output voltage is varying with the mechanical rotation of the potentiometer shaft, which should be coaxial with the steering servo motor side hole of the linkage. As a result, once there is a steering action at the wheel level, the linkage will consequently rotate at a certain angle. Then, the potentiometer shaft will also simultaneously rotate at the same angle, making the output voltage increase or decrease depending on the installation of the potentiometer system.

Connection Attempts

1. Use two flange couplers: The original screw used to link the linkage to the steering servo motor is a cap-head M3 screw with a 10 mm length. In addition, the diameter of the potentiometer



(a)



(b)

Figure 30: 100 kOhm single-turn rotary potentiometer.



(a)



(b)

Figure 31: Views of two connection attempts of potentiometer and steering servo motor linkage.

rotary shaft is 6 mm. So, two flange couplers with 3 mm and 6 mm bore diameters should be found separately. The holes in the flat flanges of the two couplers should ideally be the same distance from the centre lines. This is so that the four longer screws used to connect the couplers, as shown in Figure 31a, can be put in the right way to make sure that the two flange couplers are coaxial. Then, the potentiometer body is fixed on an adjustable fixing bracket by tightening the thin hexagon nut, and the adjustable fixing bracket is fixed on the right longitudinal bar through two L-shape mounts.

This design attempt relies a lot on the accurate positions of the holes in the flat flanges. Therefore, the second attempt is tried.

2. Use a simple rigid coupler: Thanks to the availability of a simple rigid 3 mm to 6 mm coupler, as can be seen in Figure 31b, this second attempt is built and applied to the final setup of the Losi 5IVE-T 2.0. Compared to the previous attempt using flange couplers, the second one assures the coaxiality of the new 40 mm M3 screw (without cap) for the steering servo motor linkage and the potentiometer rotary shaft. Also, it makes the steering angle measurement system more compact and require fewer components to be repaired.

The purchased components can be found in the links which are listed in Table 6.

Table 6: Components of steering angle measurement system of the electrified Losi 5IVE-T 2.0 .

Component	Link
100 kOhm Sigle-Turn Rotary Potentiometer	[23]
3 mm to 6 mm Potentiometer Shaft Coupler	[24]
40 mm M3 Screw (without cap)	[25]
Adjustable Fixing Bracket	[26]

Wiring and Calibration A simple test is done to calibrate the functionality of the three potentiometer pins before welding the wires (which have different colours to indicate the most possible functionality of each wire to other researchers to do further study on this project) to the pins by using wires with crocodile connectors, as is shown in Figure 30b and a simple Simulink model.

The model includes a "Setup" block, a "PWM generation (5)" block to control the steering servo motor with three manually selected steering duty cycle limit values, as stated above, an "IO397 Analog Output" block to feed the potentiometer with 5 V, an "IO397 Analog Input" block to read the feedback potentiometer voltages, and some other basic blocks to complete the model building.

Thanks to the model, the correlation of the steering angle range at the wheel level, the feedback voltage range of the potentiometer, and the PWM duty cycle range of the steering servo motor can be found considering the three limit positions of the steering servo motor linkage, which can be seen in Figure 32. From left to right, the figures show the right most steering (-32.5 deg), the rest position without any steering action (0 deg), and the left most steering (32.5 deg) of the Losi 5IVE-T 2.0. Maximum values at two limits are approximate ones averaged by several measurements using an electronic compass application.

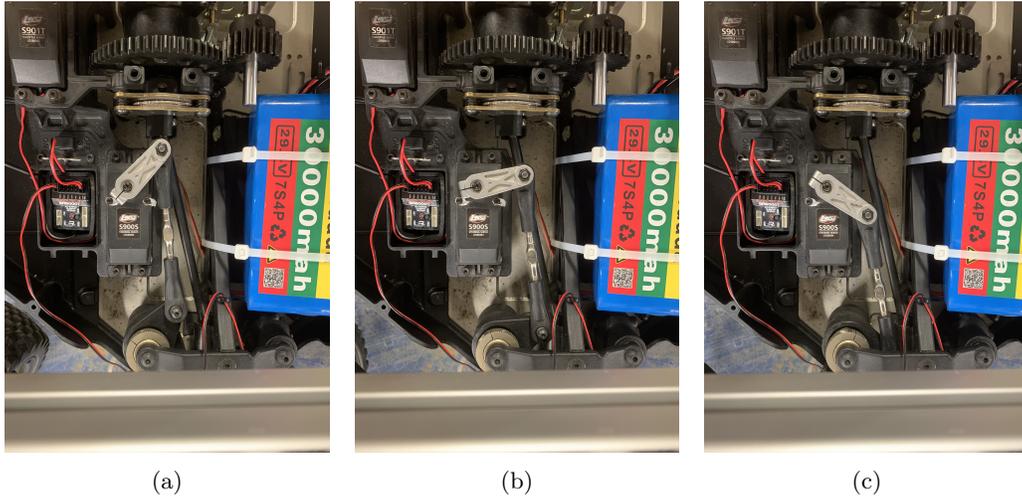


Figure 32: Three limit positions of steering servo motor linkage.

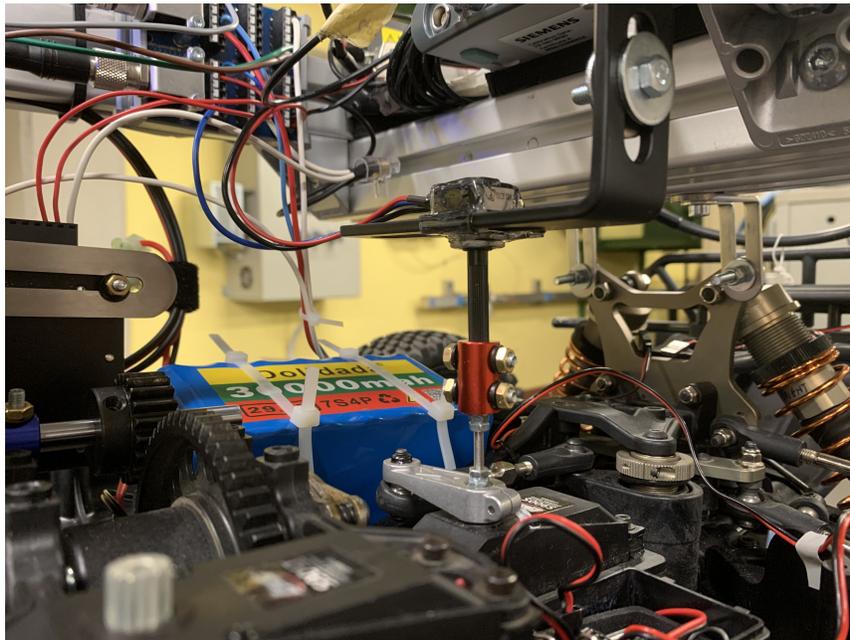


Figure 33: Steering angle measurement system of the Losi 5IVE-T 2.0.

In addition, the left pin is found to be connected to the ground, the middle one is for the output voltage, and the right one should be fed with an input voltage. After welding the wires with corresponding colours, the steering angle measurement system is installed as can be seen in Figure 33.

Some problems occurred when testing the final structure and functionality of the steering angle measurement system using the control modality with predefined steering and throttle vectors.

The first one is that the original M4 screws provided with the coupler are only 10 mm in length, which is too short. It means that there is no available length for washers and nuts to hold the screws tight to limit the unwanted slip between the potentiometer shaft and the coupler found after checking the "Scope" block added for the comparison between the input steering angle values and the feedback ones obtained from the potentiometer. The solution is to use the new 20 mm ones, as shown in Figure 33.

Second issue is caused by the residual torque that builds up when the 40 mm M3 screw is tightened onto the linkage. This causes the feedback steering angle values calculated by the feedback potentiometer voltage to drift by 0.014 degrees per second, even when the steering servo motor is at rest. To compensate for this drift, a "Gain" block with -0.014 gain is added to the control model. Check out the details explained in the chapter dedicated to explaining the control models.

3.1.2 GPS Antenna

The GPS system is an elaborate satellite navigation system designed and funded by the United States Department of Defence and consists of 24 satellites capable of continuously transmitting geographic position and time signals, which are equally distributed in six orbits. Typically, a GPS receiver on the ground receives 5 to 12 satellite signals, and at least four satellite signals are required in order to obtain location coordinates on the ground. Three are used to determine the latitude, longitude, and altitude of the GPS receiver, while the fourth provides synchronised correction time.

There are three main categories of data that a GPS receiver can provide to the user. The first category is position information, which includes longitude, latitude, and altitude. The unit quantity for longitude and latitude is geographic degrees, and the unit is deg, while the unit quantity for altitude is position, and the unit is m. The second category is speed information, which includes speed, east velocity, north velocity, and up velocity. The unit for speed is km/h, and the units for east velocity, north velocity, and up velocity are m/s. The last category includes time and the number of satellite signals received in real time.

The GPS receiver is eventually magnetically attached to one of the two e-motor assembly supporting plates and connected via a data cable to the GPS antenna interface of a data acquisition system, the Scadas XS used in this thesis work. In Figure 34, the upper thinner black cable routed from the GPS receiver to the Scadas XS can be seen. The Scadas XS is responsible for storing the data output from the GPS receiver on its micro-SD card. By properly processing the acquired data, it is possible to obtain the in-plane velocity and real-time location in Cartesian coordinates of the Losi 5IVE-T 2.0. Details of the data processing can be found in the subsequent sections.

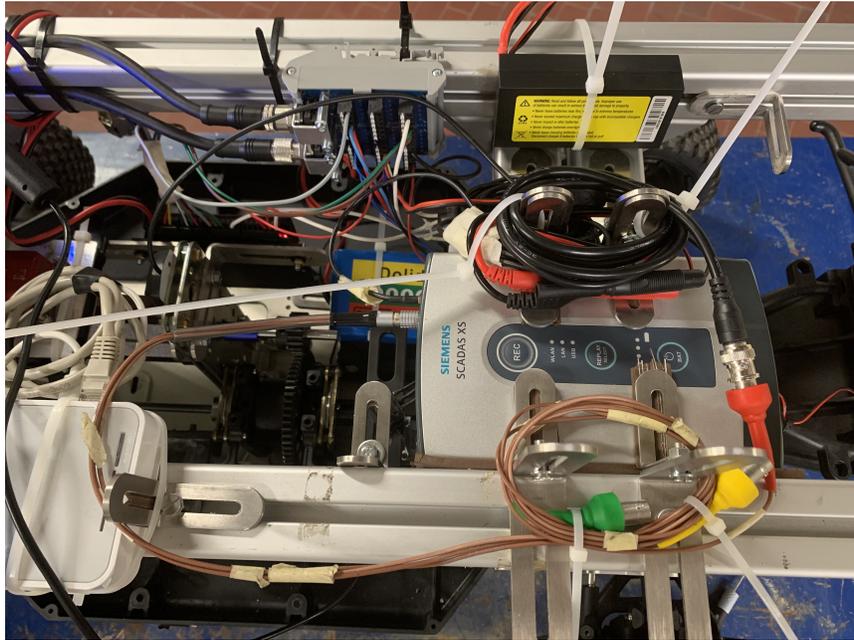


Figure 34: GPS antenna and Scadas XS triggering cable connection.



(a)



(b)

Figure 35: IMU cable connection and Scadas XS interfaces.

3.1.3 IMU

Apart from the in-plane velocity and real-time location in Cartesian coordinates, the linear accelerations and the angular velocities are the other useful data that needs to be recorded and post-processed to study the longitudinal and the lateral dynamic behaviours of the Losi 5IVE-T 2.0. Thanks to an Inertial Measurement Unit (IMU), this information could be available.

The selected device to provide the above information is the SBG Ellipse-A, a high-performance MEMS-based inertial system that is, in particular, compact enough for this project. As stated in [22], the SBG Ellipse-A has two main functional parts. The first part is the IMU, which contains three accelerometers, three gyroscopes, and three internal magnetometers. These sensors enable the ability of the SBG Ellipse-A to output the linear accelerations along the three axes of the IMU reference system printed on the top surface, as is shown in Figure 35a, the angular velocities around the three axes, and the Euler angles. The unit for the linear accelerations is m/s^2 , the unit for the angular velocities is rad/s , and the unit for the Euler angles is rad . The sampling rate of the accelerometers is 4 kHz, and that of the gyroscopes is 10 kHz. Both of those two sensors are equipped with an advanced anti-aliasing FIR filter. The second part includes an on-board enhanced Extended Kalman Filter (EKF) to perform the sensor data fusion and outputs data with a 200 Hz frequency through the CAN protocol.

The SBG Ellipse-A is finally placed close to the Center of Gravity (CoG) estimated by the mass property tool in the SOLIDWORKS. The original cable to be plugged into the main connector, as can be seen in Figure 35a, is split into two. One has red and black banana connectors to be linked to the EC5 battery serial connector of the 7.4 V 50C LiPo battery through a banana-to-EC5 battery serial cable to receive supply voltage. The other has a DB9 connector to be linked to the DB9 side of the CAN communication cable to send data to the Scadas XS through the CAN port, which is the right black connector in Figure 35b. Also for the IMU, the Scadas XS is responsible for recording data in order to support the data post processing. Scripts to process the read data will be explained in the subsequent sections.

3.1.4 Scadas XS

As stated, the Scadas XS, which can be seen in Figure 34, is the device used to receive and record real-time data from the GPS receiver and the SBG Ellipse-A IMU. Apart from the connections introduced in the previous sections, there is some other information about this device.

To utilise the Scadas XS for this project, one more cable connection should be developed. The entire data acquisition system is built for the control modality using predefined vectors. Because the start of each trajectory needs a mouse click on the "Slider Switch" block built in the model and the start of the data recording needs a manual click on the "Start" button displayed on the tablet screen, it's hard to ensure the exact same timing. Therefore, a trigger signal should be introduced to check at which time the simulation is started and at which time it's ended when post processing the data. So, an analog signal is generated by the "IO397 Analog Output" block and routed to one of the three channels of the 1ABC port. In Figure 34, the thicker brown cable routed from the A part of the IO397 terminal board to the Scadas XS can be seen. Detailed wire connections will be shown in the next sections

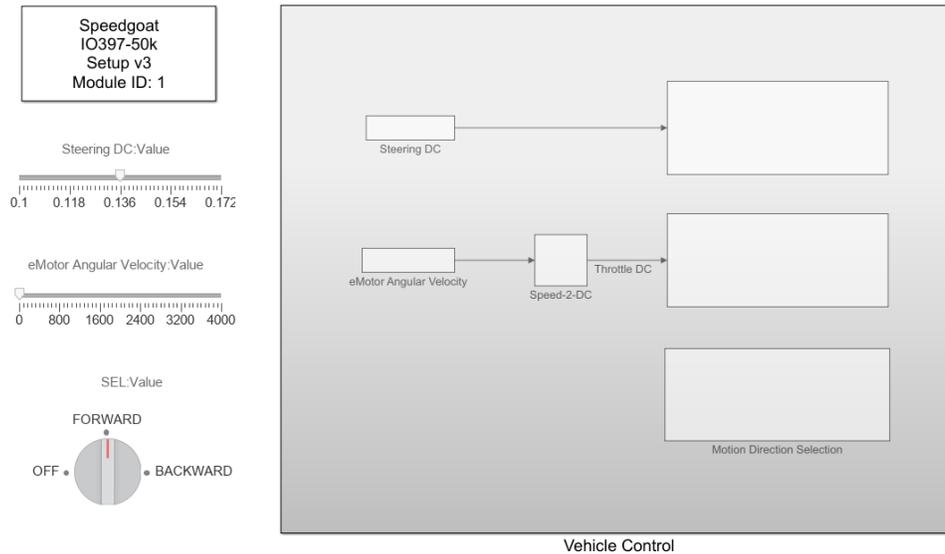


Figure 36: Control with Dashboard Blocks: complete model.

In the preparation phase, an important action to take is to create a dedicated measurement template that includes all the necessary settings, such as enabling all the needed channels, selecting the correct signals in the "CAN" section considering the type of IMU used, and checking the static storage sample rate in the "Measurement" section, through the LMS Smart Scope tablet application.

After performing all the tests, take out the micro-SD card from the Scadas XS and save all the ".xtrp" files stored in the created template. Each file corresponds to one run. Then, the ".xtrp" files can be converted to ".mat" files, which can be easily processed by using MATLAB to benefit the discussion phase of this thesis work. In the following text, there will be comprehensive explanations about the scripts used to do the data post processing task.

3.2 Control Modalities

The goal of this thesis work is to utilise the Speedgoat Baseline real-time target machine and the installed IO397 I/O Module to control the steering and throttle functions of the electrified version of the Losi 5IVE-T 2.0 in different control modalities. To this end, the configuration file used is the rapid control prototyping (RCP) one, which can be checked again in the left window in the Figure 22b. Depending on the control modality used, wiring between the IO397 I/O Module terminal board and other devices, such as the S900S 1/5 Scale Metal Gear Steering Servo, the Maxon Motor ESCON 50/5 e-motor controller, and the Scadas XS, can be different because, although the configuration file is always the same, different channels of the terminal board are activated for different purposes of signals. All the details about building the dedicated real-time applications and wire connections will be explained in the following text.

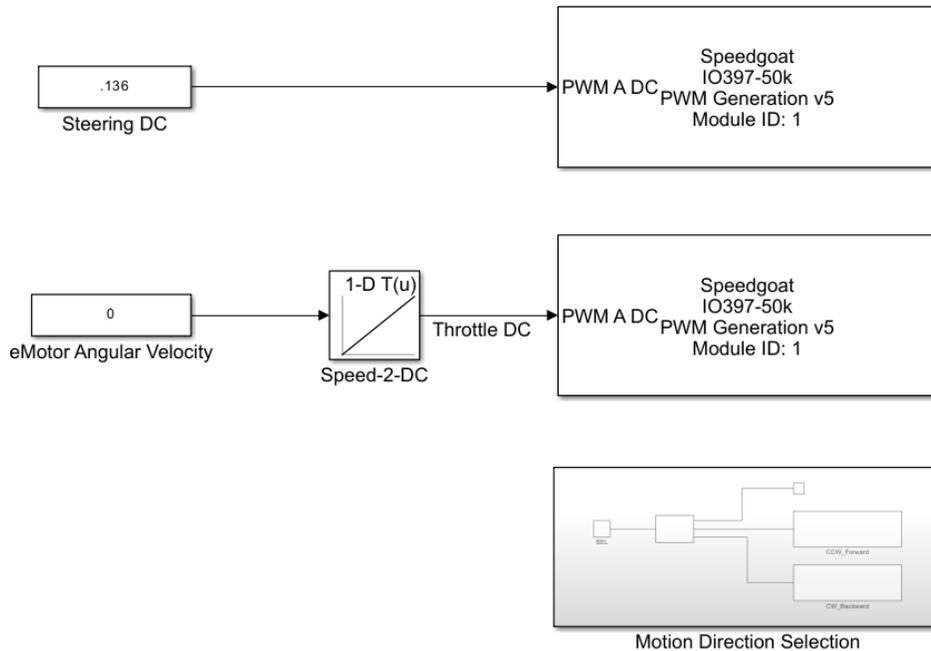


Figure 37: Control with Dashboard Blocks: vehicle control subsystem.

3.2.1 Control with Dashboard Blocks

Introduction The simplest way to integrate the Speedgoat Baseline into the setup of the Losi 5IVE-T 2.0 is to directly link the steering servo motor and the e-motor controller to the IO397 I/O Module in order to receive the steering and throttle PWM signals generated by the "PWM generation (5)" blocks that are included in the Simulink real-time model. Two "Slider" blocks are used to output various instant values, which should respect the limitations of different kinds of signals, to the "PWM generation (5)" blocks. And one "Rotary Switch" block is used to select the motion direction of the Losi 5IVE-T 2.0. Therefore, this control modality is named "Control with Dashboard Blocks".

Simulink Model and Module Wiring The development of the Simulink real-time model is not a completely standalone process, because the wiring between the IO397 I/O module and other components determine the channel numbers of the signals, which should be correctly defined in the dialog fields of the IO3xx driver blocks.

First of all, the "Setup" block, as can be seen in Figure 36, is always the first driver block to be added in order to build a Simulink real-time model. The module selected is IO397-50k, and the configuration file selected is "speedgoat_IO397_50k_RCP.mat". The right window in Figure 22b will be opened if the "Pin Mapping" button shown in the left window is clicked. This file provided by

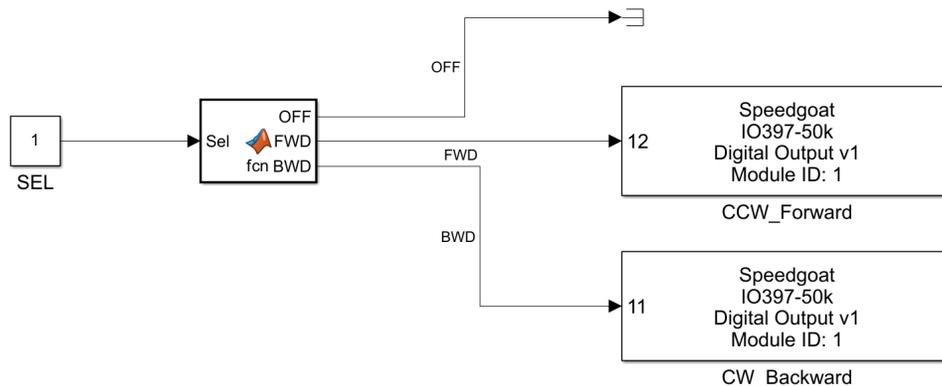


Figure 38: Control with Dashboard Blocks: motion direction selection subsystem.

the RCP configuration package is an essential reference for connecting all the desired wires to send and receive digital and analog signals from the module to external devices or vice versa.

As explained in the previous chapter, one PWM signal with a duty cycle ranging from 10% to 17.2% should be outputted from the B part of the IO397 terminal board to the steering servo motor. A "Slider" block, which has a minimum value set to be 0.1 and a maximum value set to be 0.172, is linked to a "Constant" block, named "Steering DC", in the "Vehicle Control" subsystem, which can be seen in Figure 37. And a "PWM generation (5)" block is connected to the output port of the "Constant" block. To generate a PWM signal with an acceptable frequency from the duty cycle values, the "Sample time" parameter in the dialog field of the block is set to be the reciprocal of the "PWM_freq" variable defined as 100 Hz in the "Model Workspace". The "Channel vector" parameter is set to be "[1]", which means the positive (red) wire of the steering servo motor should be connected to Pin 2b to receive a supply voltage from the module, the signal (orange-blue) wire should be plugged into Pin 3b, and the ground (white) wire should be linked to Pin 17b to complete the PWM communication between the steering servo motor and the Speedgoat Baseline.

Another PWM signal with a duty cycle ranging from 10% to 90% is responsible for controlling the e-motor controller. Apart from the blocks used for the steering pathway, an "1-D Lookup Table" block is added to map the rotating speed of the e-motor ranging from 0 rpm to 4000 rpm into the throttle duty cycle ranging from 0.1 to 0.9. And the "Channel vector" parameter is set to be "[2]". The PWM signal (grey) wire should link Pin 1 of the J5 Digital I/O block of the e-motor controller to Pin 6b of the module. And another ground (white) wire routed from Pin 17b should be linked to Pin 5 of the controller to complete the PWM communication between the e-motor controller and the Speedgoat Baseline.

In addition, to determine the motion direction of the Losi 5IVE-T 2.0, the third pathway from a "Rotary Switch" block to two "Digital Output" blocks should be built. The "Rotary Switch" block has three states, "OFF", "FORWARD", and "BACKWARD", corresponding to three state

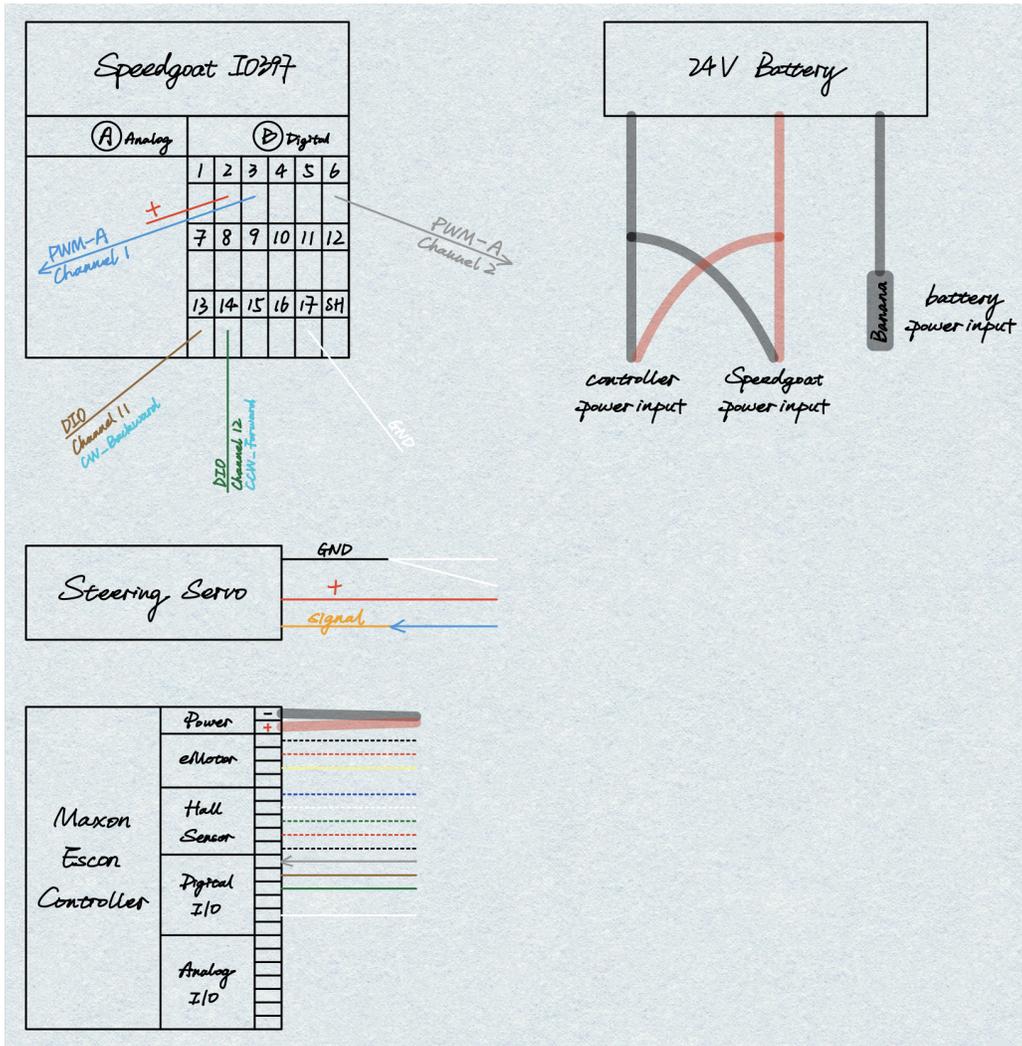


Figure 39: Control with Dashboard Blocks: complete wiring.

values, "0", "1", and "2", outputted to a "Constant" block, named "SEL". The conversion from the desired motion states to the two digital signals outputting logic values, 0 or 1, to the J6 Analog I/O block of the e-motor controller to control the e-motor's rotation direction is done by a "MATLAB Function" block, which can be seen in Figure 38, containing codes shown in the appendix.

The "Channel vector" parameter is set to be "[11]" or "[12]" to activate the clockwise or counter-clockwise rotation of the e-motor, which corresponds the backward or forward motion. One digital signal (brown) wire should link Pin 2 of the J6 Analog I/O block of the e-motor controller to Pin 13b of the module. And another digital signal (green) wire routed from Pin 14b should be linked to Pin 3 of the controller to complete this digital communication. Check out the complete wiring related to the control modality using dashboard blocks, including the connections about the 7S4P 30000 mAh 24 V battery, in Figure 39.

3.2.2 Control with Radio Controller

Introduction The first control modality requires manual clicking or sliding actions on the various dashboard blocks by using a mouse. The second one utilises the stock radio controller. The steering and throttle actions performed at the transmitter level are received by the receiver wirelessly and then transferred to the IO397 I/O Module. The real-time application carried by the Speedgoat Baseline is meant to process the original signals and then output the control signals to various external devices, such as the steering servo motor and the e-motor controller.

Simulink Model and Module Wiring Before establishing the complete model to send processed signals to the steering servo motor and the e-motor controller, a simplified version of it, which is shown in Figure 40, should be used to know what the actual throttle duty cycle limits are, which should be defined in the "Model Workspace" and then used by other blocks. Apart from the "Setup" block that should be added in order to create a new Simulink real-time model, an "IO397 Analog Input" block should be used to read the analog commands passed from the receiver throttle port to the I/O module. A subsystem named "Throttle Filter", as can be seen in Figure 41, is responsible for normalising the received voltage by adding a "Gain" block and translating the voltage from the PWM form to the duty cycle values by using a "Transfer Fcn" block, which has a 5 Hz cut-off frequency. Finally, a "To Workspace" block is used to save the throttle duty cycle signals in the "Structure With Time" format and write them in the MATLAB base workspace.

Three ".mat" files are obtained separately to show the corresponding duty cycle values of the three limit positions of the transmitter physical trigger. The script to post process the ".mat" files to obtain the actual throttle duty cycle limits is shown in the appendix. The actual throttle duty cycle limits calculated and used in several subsystems of the complete control model are 0.082613, 0.13744, and 0.1923.

To control the Losi 5IVE-T 2.0 by using the stock radio controller instead of the dashboard blocks, the "IO397 Analog Input" block is the core part. Command signals generated by the transmitter are received wirelessly by the receiver and then transferred to the IO397 module. Two dedicated subsystems containing the "Gain" and "Transfer Fcn" blocks are linked to the output ports of the "IO397 Analog Input" block to output the steering and throttle duty cycle values to the next functional blocks and subsystems. Finally, the processed duty cycle values are routed to the "PWM

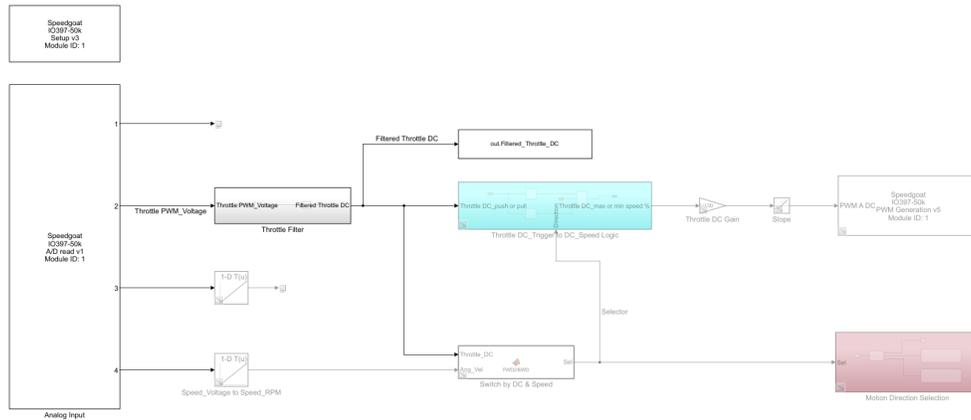


Figure 40: Control with Radio Controller: simplified model.



Figure 41: Control with Radio Controller: throttle filter.

generation (5)” blocks as previously explained. The complete control model is shown in Figure 42.

For the steering pathway, there is a ”MATLAB Function” block, containing codes shown in the appendix, named ”Steering Startup Limitations”, used to apply an additional noise tolerance to avoid abrupt movements at the physical knob level, which will possibly overcome the defined steering duty cycle limits. Where ”Steering_DC_Filt” is the input of this function. The ”noise_tol_perc”, ”Steer_lim”, and ”Rest_DC” are fixed parameters which that can be fine-tuned in the ”Model Workspace”.

The throttle pathway consists of two interacting parts. The first part is to determine the rotating speed of the e-motor by translating the duty cycle values received from the transmitter to the duty cycle values that should be sent to the e-motor controller. In the subsystem named ”Throttle DC.Trigger to DC.Controller DC %”, which can be seen in Figure 43, there are two inputs. If the duty cycle values received from the transmitter are in the throttle forward range, from 0.13744 to 0.1923, thanks to the second input determined by the second part of the throttle pathway, the duty cycle values sent to the e-motor controller will make the Losi 5IVE-T 2.0 move at the right speed. The ”Rate Limiter” block named ”Slope” is added to limit the rising and falling rates of the PWM signal. The rates can be freely tuned in the ”Model Workspace”.

The second part of the throttle pathway is to determine the rotating direction of the e-motor not only using the duty cycle values received from the transmitter but also the feedback rotating speed of the e-motor from the e-motor controller J6 Analog I/O block. The feedback rotating speed of the e-motor can also be read by the ”IO397 Analog Input” block. A ”1-D Lookup Table” block

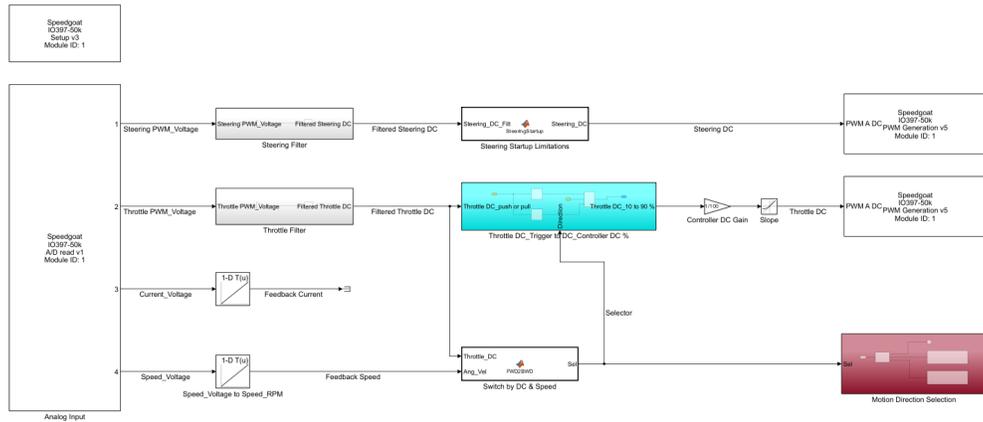


Figure 42: Control with Radio Controller: complete model.

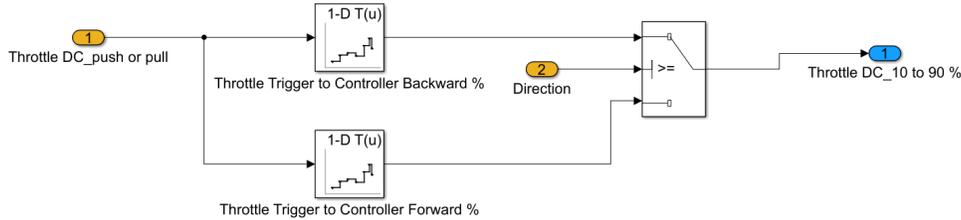


Figure 43: Control with Radio Controller: throttle DC_trigger to DC.controller DC % subsystem.

is used to map the voltage ranging from 0 V to 4 V into the total speed range of the e-motor, from -4000 rpm to 4000 rpm. Then, another "MATLAB Function" block, containing codes shown in the appendix, named "Switch by DC & Speed", is used to receive both the converted angular velocity and the actual throttle duty cycle value at the same time to determine the next instant "Sel" outputted to the "Throttle DC_Trigger to DC_Controller DC %" and the "Motion Direction Selection" subsystems. The reason for developing this updated version compared to the previous function used in the first control modality is to smooth the abrupt behaviour of the e-motor when a change from one moving direction to the other moving direction of the Losi 5IVE-T 2.0 is performed at the transmitter level by pulling or pushing the physical trigger over the rest position of it.

Apart from the connections that have been explained in the first control modality, the wiring related to the stock receiver can be found in Figure 44. The battery port of the receiver should be connected to the 2S 4000 mAh 7.4 V 50C LiPo battery through a pair of EC5 battery serial connectors. The command signal (yellow-green) wires are linked to Pin 1a (Analog Input 01+) and Pin 3a (Analog Input 02+) separately to send the steering and throttle commands from the transmitter to the IO397 module. Pin 2a (Analog Input 01-) and Pin 13a (GND) of the module are connected to the steering ground (black) of the receiver. And Pin 4a (Analog Input 02-) and

However, it cannot assure the 100% accuracy of the signal transmission. And taking evidence of the noise tolerances introduced and the throttle duty cycle limits updated, the multiple times of the format transformation of the command signals increase the uncertainty of signal routing.

3.2.3 Control with Predefined Vectors

Introduction The third control modality developed replaces the dashboard blocks used in the first control modality with five predefined trajectories, such as "Acceleration-Deceleration", "Circle", "Spiral", "8-Shape", and "Sweep". The vectors containing information about the steering and throttle duty cycle variations are written in five dedicated MATLAB scripts, which should be called by a main script before running the developed Simulink real-time model on the target machine and performing tests in the field.

MATLAB Scripts In this section, all the scripts used to perform the field tests will be shown and explained.

Acceleration-Deceleration Trajectory This trajectory is designed to test the longitudinal behaviour of the Losi 5IVE-T 2.0. The definitions of the steering duty cycle vector, named "steer.posacceldecel", and the throttle one, named "thr.posacceldecel", are simple. Considering all the factors, such as the maximum testing speed, 6 km/h, of the Losi 5IVE-T 2.0, the ground conditions, and the available area of the field selected, the time period of the theoretical maximum speed is set to be 15 seconds. The complete script of this trajectory is shown in the appendix.

Circle/Spiral/8-Shape/Sweep Trajectories These trajectories are designed to test the lateral behaviour of the Losi 5IVE-T 2.0. The definitions of the steering and the throttle duty cycle vectors, are more complex compared to the previous one. Since the corporation between two vectors is necessary.

Compared to the script used to define the "Acceleration-Deceleration" trajectory, the "Circle" script stresses the importance of the steering function. So, another input, the desired steering angle, is introduced. Check out the complete script for this trajectory in the appendix. The difference between the script designed for the "Circle" trajectory and that of the "Spiral" script is the "Steering Path" part, which is also shown in the appendix.

Also for the "Sweep" trajectory, the part that should be updated is the "Steering Path". The function used is the "chirp" function provided by MATLAB, with start and end frequencies set to 0 Hz and 5 Hz, respectively. Therefore, different from the logic of designing the "Circle" steering vector that is fixing a constant steering angle during the "t_maxspeed" and that of the "Spiral" that is increasing the steering angle, the main part of the steering vector designed for the "Sweep" is meant to vary the steering angle from 0 Hz frequency to 5 Hz frequency. As a result, the "Steering Part" section written for the "Sweep" trajectory is shown in the appendix.

The "8-Shape" trajectory is a very distinct one because the desired 8 shape, which should have two straight lines at the intersection of the 8 shape and the rest of it should be two three-quarter circles, must be strictly defined by an essential parameter, the radius of the three-quarter circle. The radius of the three-quarter circle depends on many factors. Before doing field tests, the estimated

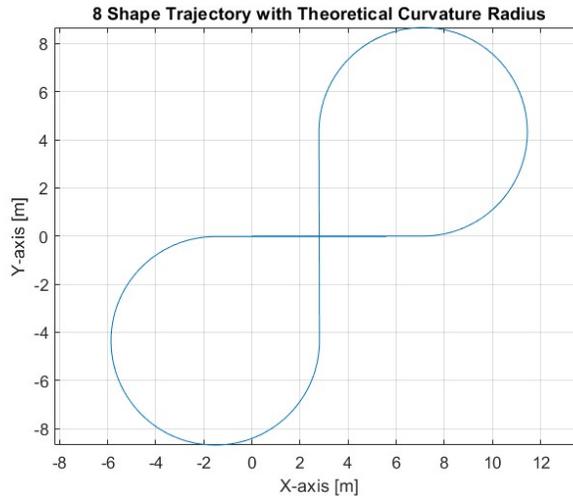


Figure 46: 8 shape trajectory example.

one considering the physical data and the desired speed, 4 km/h, of the Losi 5IVE-T 2.0 can be approximately calculated and used to design the first attempt of the "8-Shape" trajectory. Check out the complete script for this trajectory in the appendix. Also, to visualise the trajectory shape, another script followed the complete one, which can be seen in the appendix.

As displayed by the codes, the calculated curvature radius is 4.3314 m and the trajectory shape, which can be seen in Figure 46, is an 8 shape. However, after performing a "Circle" trajectory with a 4 km/h maximum speed and a 30 deg steering angle, the averaged curvature radius measured by hand is 1.18 m. So, the "8-Shape" script is updated to the second version with "R" manually defined as 1.18 m. After testing the Losi 5IVE-T 2.0 with the new script, it's found out that the steps defined to complete a three-quarter circle portion are not enough. After several tests, the curvature radius is set to be 1.5 m in the final version of the "8-Shape" trajectory script, which will be used to provide the steering and throttle vectors required by the "1-D Lookup Table" blocks in the Simulink control model.

In addition, all the steering and throttle vectors, except for the ones of the "8-Shape" trajectory, are normalised to benefit the development of the Simulink model and the flexibility of field tests.

Starting Script Apart from the scripts to define the five trajectories, a main starting script, which is shown in the appendix, containing all the necessary parameters to call the trajectories is needed. Specifying the rest positions of the two throttle ranges is necessary. Because the forward range and the backward range of throttling don't have the same length. In addition, although the left turning range and the right turning range have the same length, the rest positions of the two steering ranges are suggested to be specified. Remember to run this script before running the control model on the Speedgoat Baseline target machine.

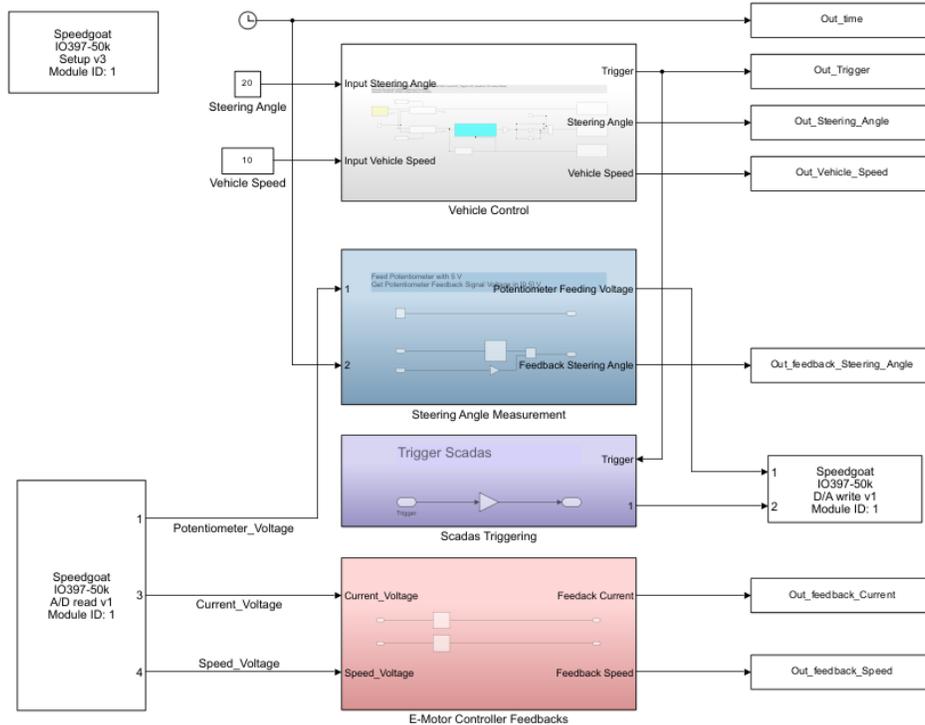


Figure 47: Control with Predefined Vectors: control part.

Simulink Model and Module Wiring To build the complete model for the control modality using the predefined steering and throttle vectors, there are two main parts.

The first part of the control model is to control the Losi 5IVE-T 2.0 and to output several useful signals to the MATLAB base workspace to support the data post processing task, which can be seen in Figure 47. There are four subsystems developed.

The "Vehicle Control" subsystem receives the desired inputs, the steering angle and the vehicle speed, passes through the steering and throttle pathways, outputs two PWM signals to the steering servo motor and the e-motor controller separately to control the real-time steering angle and vehicle speed, and outputs two digital signals to the e-motor controller to determine the motion direction of the Losi 5IVE-T 2.0, which is shown in Figure 48.

In detail, the "Path Triggering" subsystem is used to manage the clock logic to achieve a new start for each triggering event performed in the second part of the model. The output "Time" of this block is not the actual simulation time but the local time in terms of every triggered trajectory. For the steering pathway, there is only one subsystem, named "Steering Path", which can be seen in Figure 49, containing five "1-D Lookup Table" blocks. Each block is used to carry one predefined

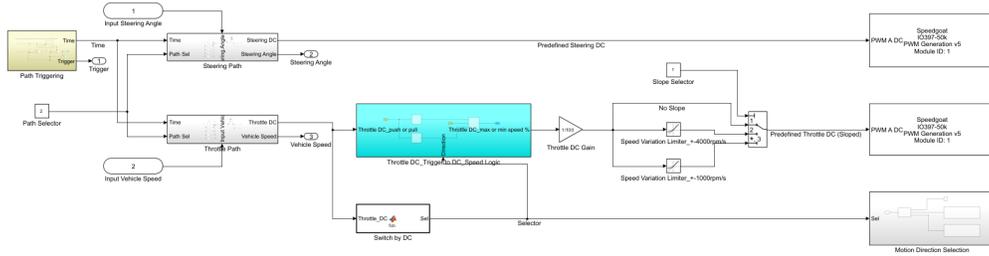


Figure 48: Control with Predefined Vectors: vehicle control subsystem.

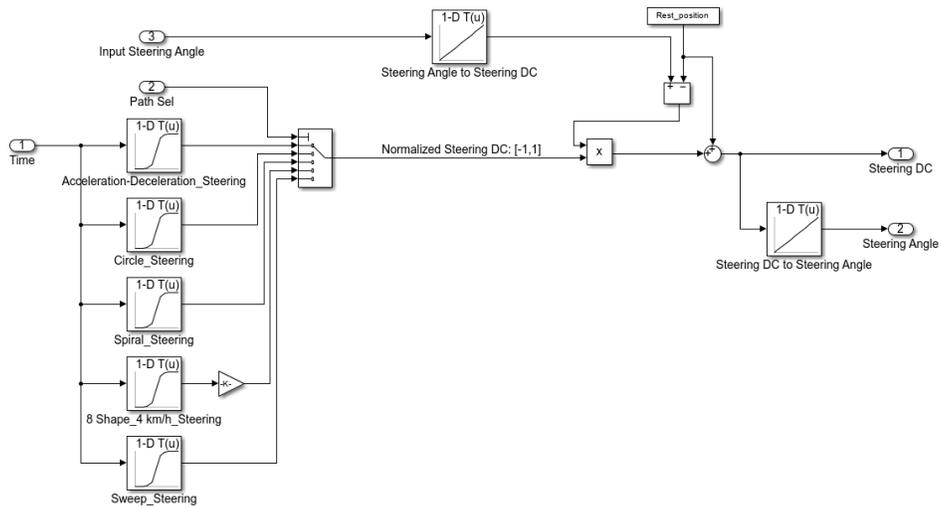


Figure 49: Control with Predefined Vectors: steering path subsystem.

steering vector. Once the trigger is activated, the local time "Time" starts to increase, making the "steer.pos" vector designed for the trajectory selected by the "Path Sel" start to be outputted in a normalised form, from -1 to 1. The third input is routed, passing through some additional blocks, to the "Normalised Steering DC" to ensure that the instant steering PWM duty cycle values outputted to the steering servo motor can be easily adjustable in terms of both the quantity and the timing during the whole simulation. By using another "1-D Lookup Table" block, also the instant steering angle desired at the wheel level can be outputted.

The "Throttle Path" subsystem has a similar structure. The input steering angle should be switched to the input vehicle speed. The normalised "thr.pos" vector is ranging from 0 to 1, knowing that all the designed trajectories don't have the intention to make the Losi 5IVE-T 2.0 move backward. And the second output is the vehicle speed that should be achieved at each moment. There are some small differences compared to the previous models. First, the manually triggered "Motion Direction

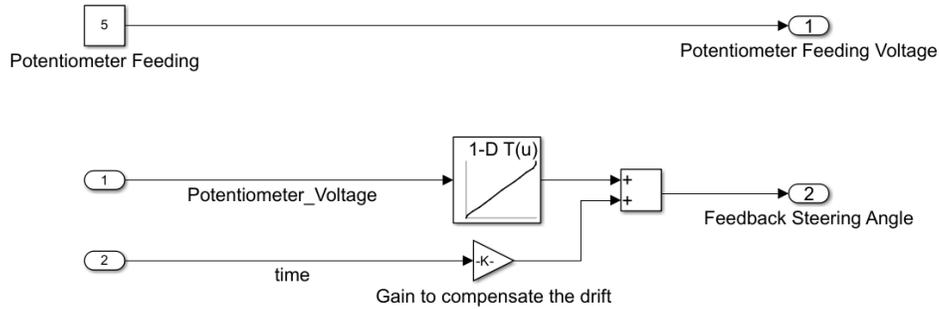


Figure 50: Control with Predefined Vectors: steering angle measurement subsystem.

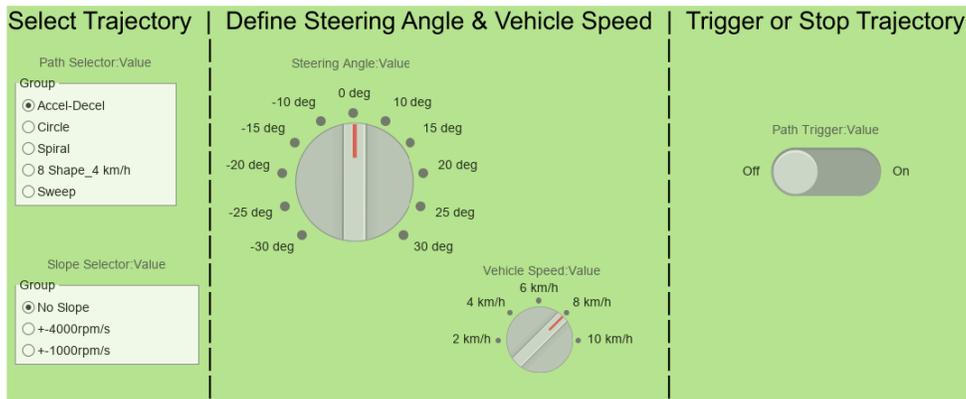


Figure 51: Control with Predefined Vectors: trajectory selection part.

Selection” subsystem in the first control modality is updated to a simpler version compared to that of the second modality, which not only reduces one step to do compared to the first modality but also maintains the possibility of future study in the case of backward motion. Second, a ”Slope Selector” is added to enable the possibility of switching among three different rate limiters, which can be useful in the testing phase of the drive line.

The ”Steering Angle Measurement” subsystem, as shown in Figure 50. sends out the feeding voltage to the potentiometer via the ”Digital Output” block, receives the instant feedback potentiometer voltage via the ”IO397 Analog Input” block, and outputs the feedback steering angle after passing through the ”1-D Lookup Table”. To compensate for the feedback voltage drift found and explained in the chapter dedicated to the potentiometer, a ”Gain” block is added to the global simulation time signal. Then the result is summed up with the mapped steering angle to obtain the final feedback steering angle. The ”Scads Triggering” subsystem and the ”E-Motor Controller Feedbacks” subsystem are well explained in the above text.

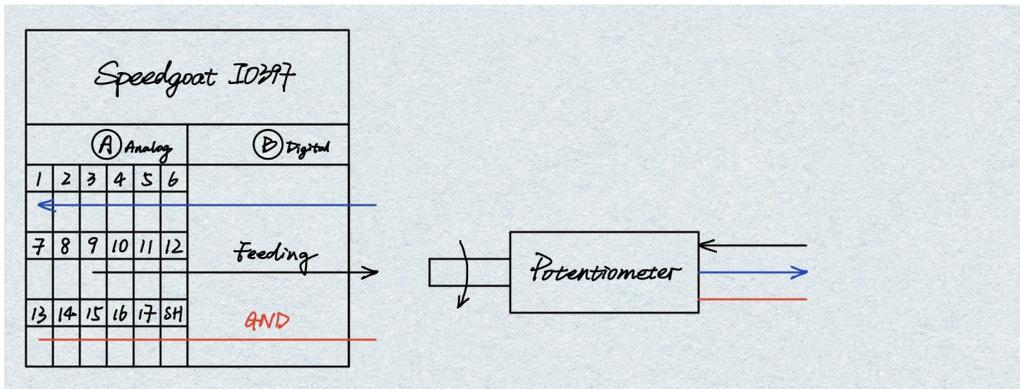


Figure 52: Control with Predefined Vectors: potentiometer wiring.

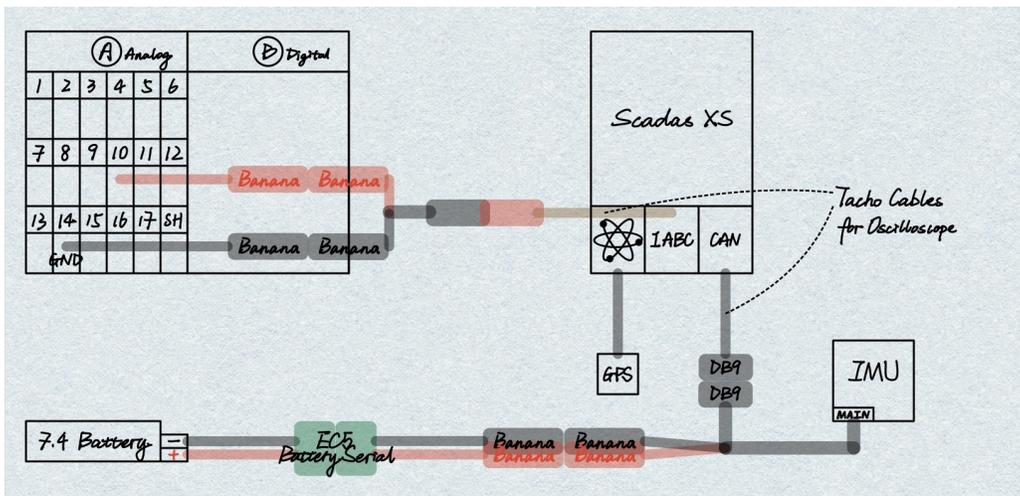


Figure 53: Control with Predefined Vectors: GPS antenna, IMU, and Scadas Xs wiring.

To conclude and emphasise, apart from controlling the Losi 5IVE-T 2.0, this first part of the model also outputs seven signals to the MATLAB base workspace, such as the global simulation time, the trajectory triggering indicator, the instant desired steering angle and vehicle speed, the instant feedback steering angle and vehicle speed, and the feedback e-motor current.

The other part of the complete model is to activate different trajectories required to do the field tests, which is shown in Figure 51 . There are three steps to starting a defined trajectory with preferred settings. First, select the trajectory that is going to be performed with a certain slope of the speed variation. Then, choose the input steering angle and the vehicle speed. Pay attention to the "8-Shape" trajectory, which is the unique one that must have 30 deg as the input steering angle and 4 km/h as the input vehicle speed. Otherwise, the outputted instant desired values

cannot represent the correct values. Finally, after running this model on the Speedgoat Baseline and checking the "External" indicator at the left bottom corner of the Simulink interface, click the "On" button to start the selected and defined trajectory. It's free to change to a new input steering angle or a new input vehicle speed during the local trajectory time. Even changing to a completely new trajectory is operable.

The wiring of the 24 V battery, the B part of the IO397 module, the steering servo motor, and the J1, J2, J3, J5, and J6 blocks of the e-motor controller is the same with the first two control modalities. For the potentiometer, as indicated in Figure 52, the ground (red) wire is connected to Pin 13a of the IO module, the positive (black) wire should be linked to Pin 9a to receive feeding voltage generated by the "IO397 Analog Output" block, and the signal (blue) wire should be plugged into Pin 1a to know the instant feedback voltage thanks to the "IO397 Analog input" block. To develop the "Scadas triggering" subsystem, a set of positive (red) and ground (black) wires with banana connectors should be plugged into Pin 10a and Pin 14a of the I/O module and linked to the Scadas XS. Pin 10a will be able to send a 3.3 V voltage, which is activated by the "Trigger" signal and generated by the "IO397 Analog Output" block, to the Scadas XS to support the data post processing task. The complete wiring related to the GPS, the IMU, and the Scadas XS can be checked in Figure 53.

3.3 Field Tests

To verify three control modalities used to control the Losi 5IVE-T 2.0, two kinds of tests are performed. First, in the laboratory, with the four wheels free to run, the functionality of the drive line assembly, the steering assembly, and the steering angle measurement assembly are verified for each modality. Second, in order to verify the third control modality, which uses predefined trajectories, field tests with the wheels attached to the ground are necessary.

The field selected is Piazzetta Manlio Brosio, 10129 Torino TO. Not all of this area is suitable for testing the vehicle. Moreover, the utilised portion of it is uneven, which challenges the gradeability of the vehicle. In Figure 54, the Losi 5IVE-T 2.0 is placed on one of the local maximum points.

All five designed trajectories are tested with the wireless control mode, which uses WiFi communication between the personal computer and the router. Furthermore, thanks to the closed-loop strategy of the e-motor controller and the data acquisition system developed, dozens of signals can be post processed to verify the control modality and study the longitudinal and lateral behaviours of the scaled vehicle.

3.3.1 Data Post Processing

There are two data categories based on the data recording device used. Thanks to the Speedgoat Baseline and the Simulink real-time model, seven signals can be recorded, which can also be checked in Figure 47:

- "Out_time" is the global simulation time.
- "Out_Trigger" indicates the starts and ends of trajectories.
- "Out_Vehicle_Speed" is the desired vehicle speed.



Figure 54: Losi 5IVE-T 2.0 on the ground of Piazzetta Manlio Brosio.

- "Out_Steering_Angle" is the desired steering angle.
- "Out_feedback_Steering_Angle" is the real steering angle based on the potentiometer.
- "Out_feedback_Current" is the real e-motor current based on the e-motor controller.
- "Out_feedback_Speed" is the real e-motor speed based on the e-motor controller.

Apart from the direct display of each signal, the comparison between the desired signal and the real ones should be discussed. And for the "Sweep" trajectory, the "tfestimate" function provided by MATLAB is used to show how the steering changing frequency affects the magnitude and phase responses of the steering system. The sampling frequency should be consistent with that of the Scadas XS, 200 Hz. Based on the frequency settings of the chirp function used, only frequencies ranging from 0 Hz to 5 Hz should be reasonably displayed in the figures.

Thanks to the GPS, the IMU, and the Scadas XS, three groups of signals are able to be obtained and post processed based on:

- Speedgoat Baseline: a trigger signal is used to synchronise the timing system of the Simulink real-time model and the Scadas XS.
- GPS: longitude, latitude, altitude, east velocity, and north velocity are the important signals to show the trajectory and calculate the in-plane velocity of the vehicle.
- IMU: Euler angles, angular velocities, and linear accelerations are used to analyse the dynamic performance of the vehicle.

Some proper unit conversions should be applied to some signals by using the developed MATLAB data post processing scripts. For longitude, latitude, altitude, Euler angles, and angular velocities,

the unit is converted from rad to deg. In particular, in order to plot the trajectories, the "lla2enu" function provided by MATLAB is used to transform the geodetic coordinates to the local Cartesian coordinates using the "ellipsoid" earth approximation. In addition, the unit of in-plane velocity after calculating is updated to km/h to be consistent with the desired and real velocities displayed based on the control model. Meanwhile, in order to reduce the noise found after displaying the original signals, several low-pass filters with tuned sampling frequencies are applied to some of the signals, such as the angular velocities and the linear accelerations.

For the longitudinal behaviour analysis, the feedback vehicle speed calculated from the feedback e-motor speed is a useful reference to be discussed with the linear acceleration along the x axis. While, for the lateral behaviour analysis, the feedback steering angle based on the potentiometer is used as a reference when showing the yaw rate, the roll rate, and the linear acceleration along the y axis.

3.3.2 Results and Discussions

Acceleration-Deceleration Trajectory For the "Acceleration-Deceleration" trajectory, tests with different desired maximum vehicle speeds (4 km/h and 6 km/h) are performed with the "No slope" speed variation limiter activated. The relevant signals are plotted in Figure 55 and Figure 56.

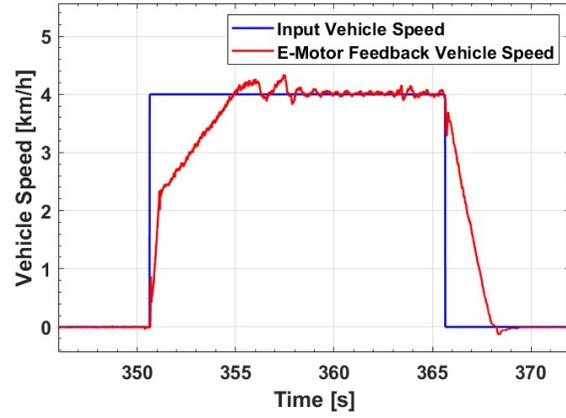
Focusing on the 4 km/h one, it's clear to see that there is a climbing phase for the vehicle to reach the desired speed after the steep increase in less than 0.5 s. During this phase, the e-motor feedback current is constant at its maximum, and the altitude is helping the vehicle. While, after reaching the maximum speed, the vehicle has to climb a slope, which makes the longitudinal acceleration start to decrease to negative values. Thanks to the throttle command sent to the e-motor, it starts to recover the current (torque) to its maximum, making the vehicle speed and the acceleration along x axis fluctuate. In Figure 56, it can be seen that if the vehicle is still heading to the desired speed and there is no slope to climb, the feedback current can be properly maintained at its maximum. The main reason is the poor gradeability of the vehicle, considering that the rated torque of the e-motor is only 0.25 Nm. Another piece of evidence can be seen in the rest of the figures, showing that the trajectory is not a perfect straight line, the feedback steering angle is increasing a bit, and the yaw rate is not a noisy signal fluctuating around 0 deg/s. It means that not only the longitudinal behaviour will be affected by the uneven ground but also the lateral one.

One more thing to note is that both at the start and the end of the simulation, the trajectory plotted by the GPS longitude and latitude signals has abnormal recorded points. The reason could be related to the poor accuracy of the GPS antenna when the speed of the vehicle has a large variation.

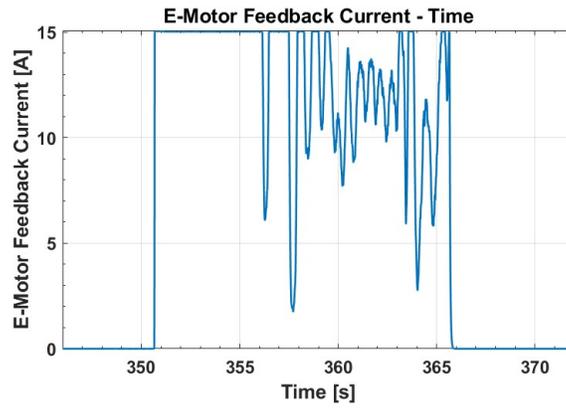
Circle Trajectory For the "Circle" trajectory, tests with different desired steering angles (30 deg and -30 deg) and maximum vehicle speeds (4 km/h and 6 km/h) are performed.

The first set of figures, as can be seen in Figure 57 and Figure 58, is to show that the designed steering vectors can perform the right steering actions at 4 km/h with an approximate 30 or -30 deg. In detail, if a 30 deg (right) steer is required, the yaw rate will have negative values because

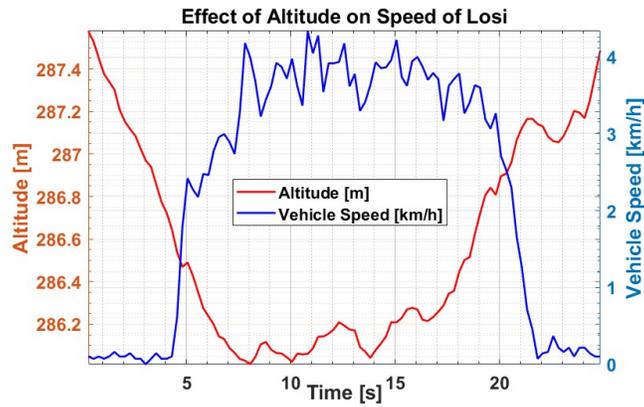
Comparison of Vehicle Speed based on E-Motor Feedback



(a)

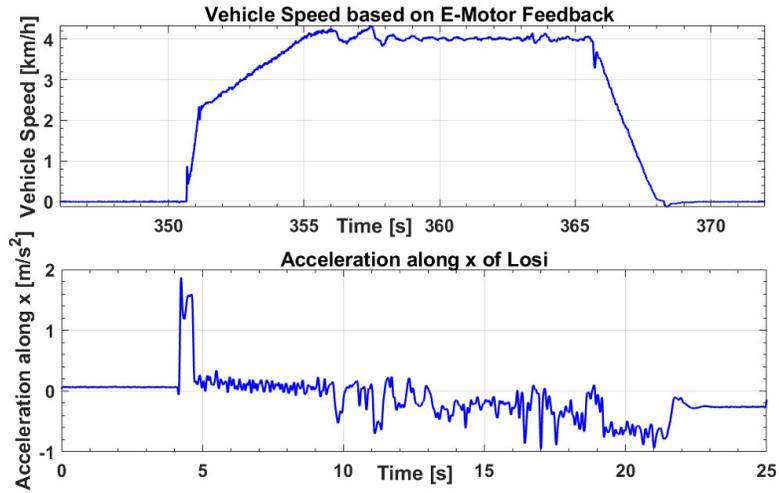


(b)



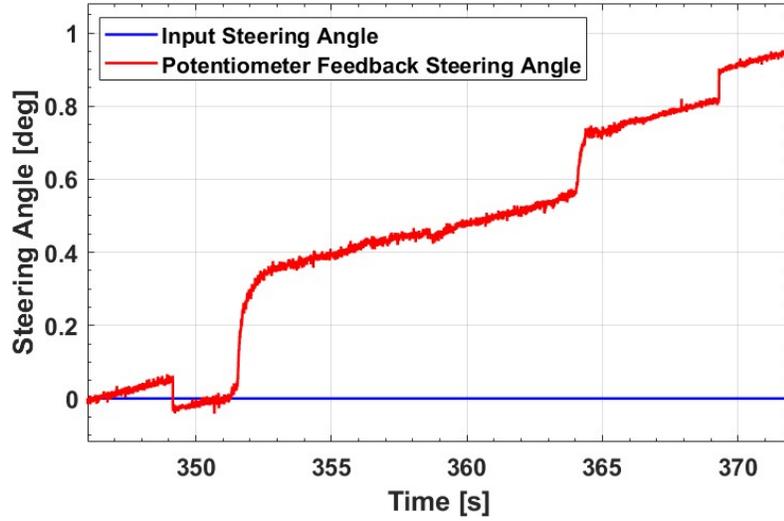
(c)

Relationship between Vehicle Speed based on E-Motor Feedback and Acceleration along x of Losi



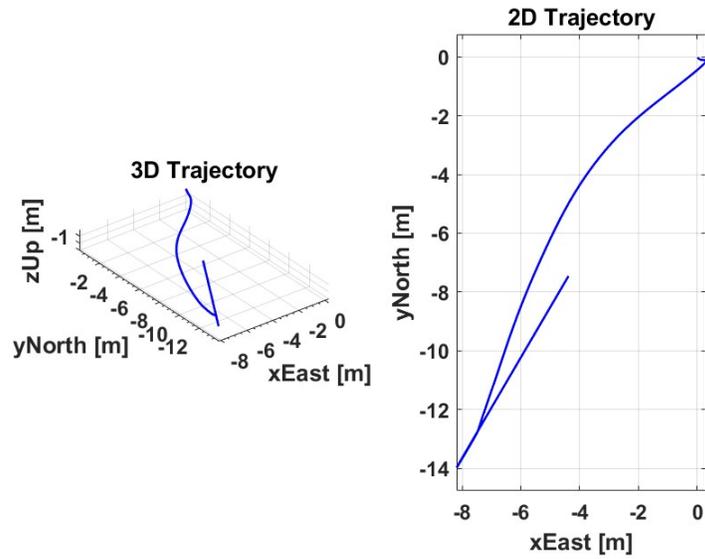
(d)

Comparison of Steering Angle

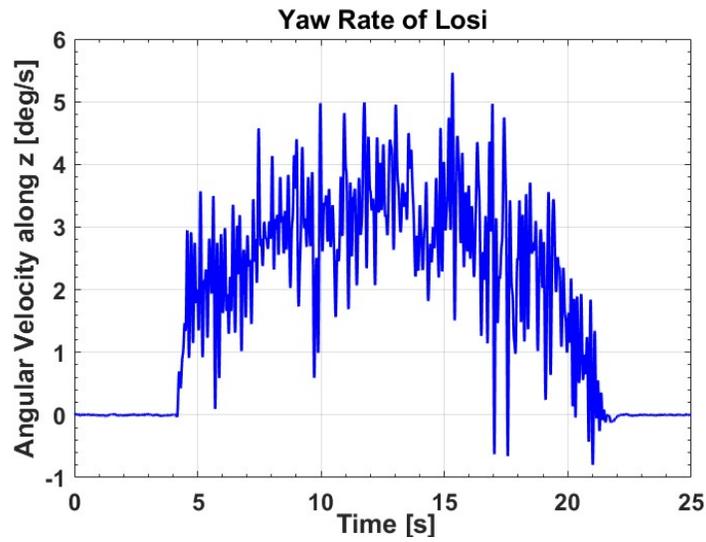


(e)

Trajectory of Losi



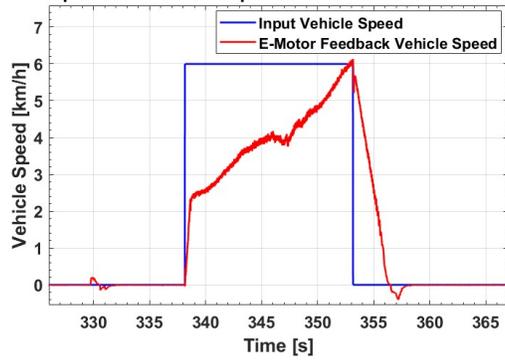
(f)



(g)

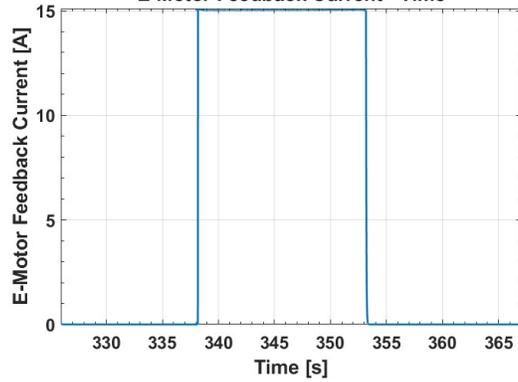
Figure 55: Acceleration-Deceleration Trajectory plots: 4 km/h.

Comparison of Vehicle Speed based on E-Motor Feedback



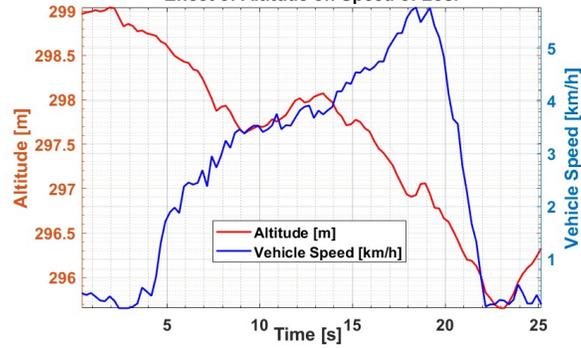
(a)

E-Motor Feedback Current - Time



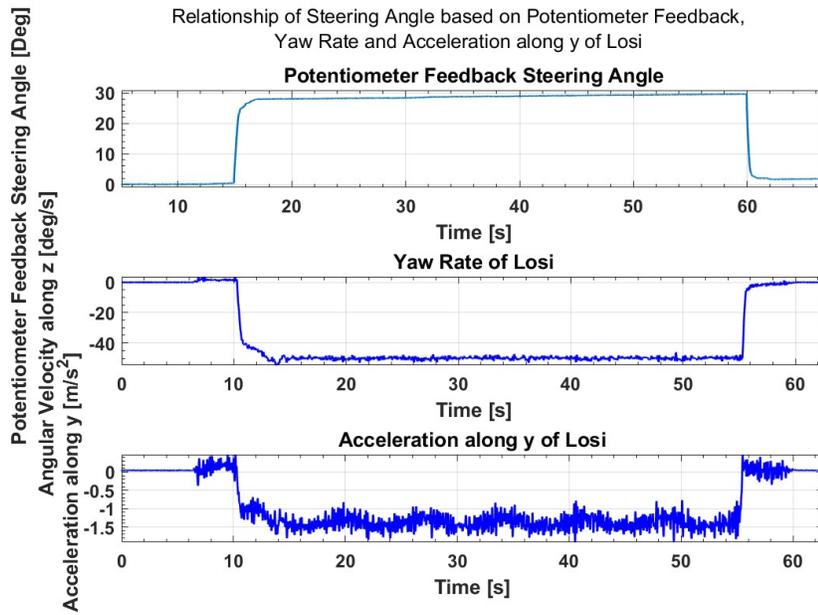
(b)

Effect of Altitude on Speed of Losi

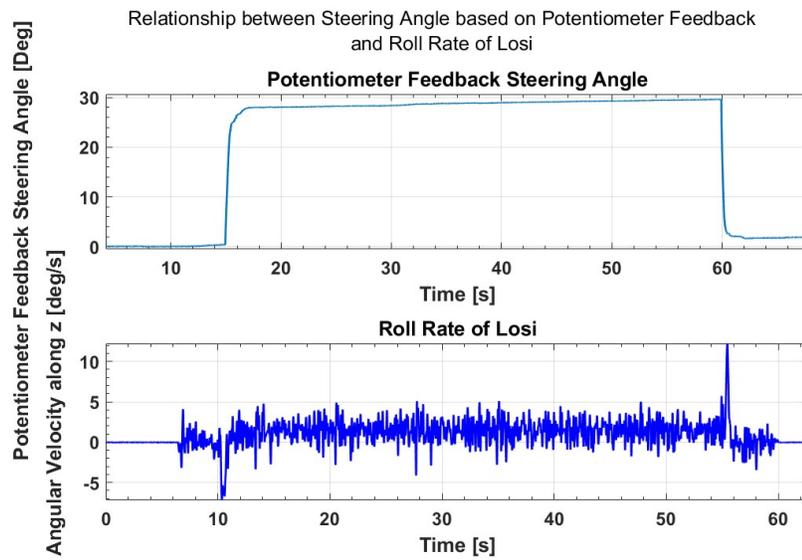


(c)

Figure 56: Acceleration-Deceleration Trajectory plots: 6 km/h.

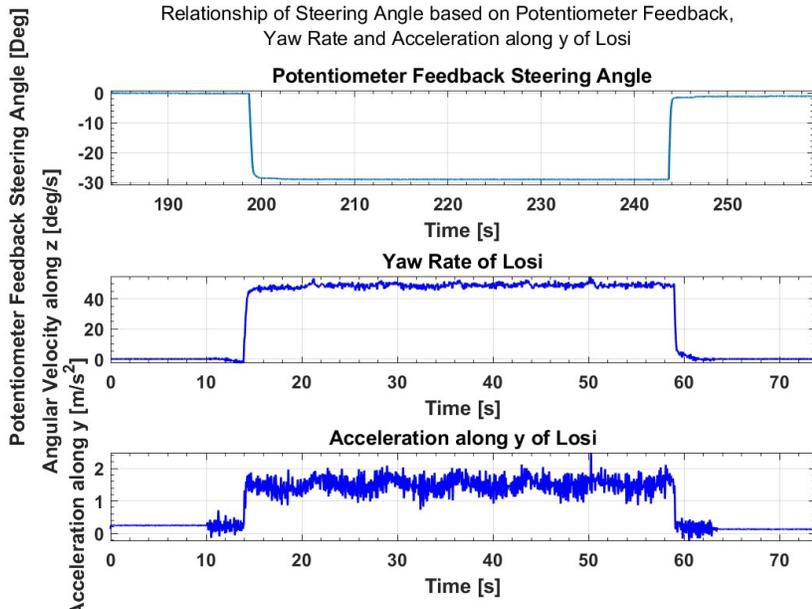


(a)

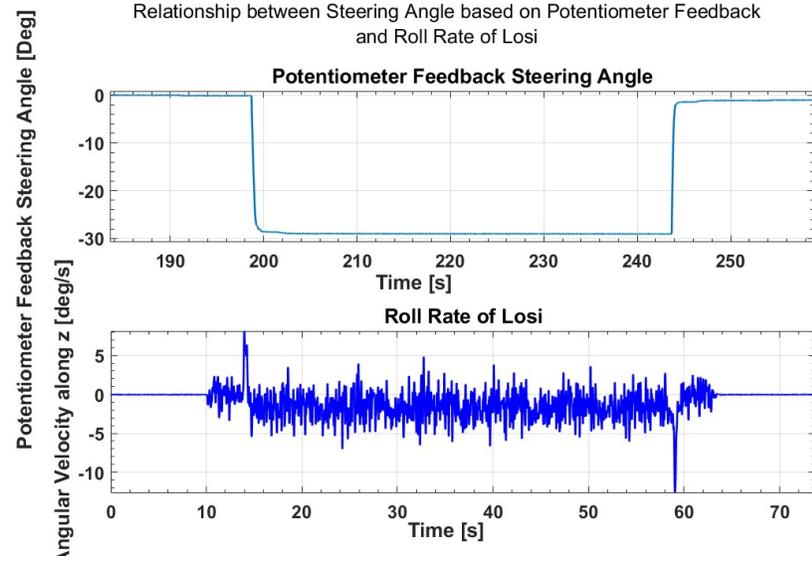


(b)

Figure 57: Circle Trajectory plots: 4 km/h; 30deg.



(a)



(b)

Figure 58: Circle Trajectory plots: 4 km/h; -30deg.

the IMU local z axis is transferred from the original one (pointing to the ground) to the new one (pointing to the sky). While the reason for the negative lateral acceleration values is that the lateral acceleration should be pointing out of the curvature and the local y axis is pointing to the right of the vehicle. The roll rate signal always has two opposite values at the start and the end of the steering phase. In the case of right steering, the vehicle tends to lean to the left due to inertia. So, at the start of the steering phase, the roll rate should have a steep increase to the negative values.

The second group, which is shown in Figure 59, shows that although the input steering angle is set to be 30 deg and the vehicle should have a constant 6 km/h speed, the real steering angle performed by the vehicle and its real speed have negative effects on each other. In other words, if the speed can be reached, the steering angle will be smaller than wanted, which is shown in the first page of Figure 59. And if the steering angle can be acceptable achieved, the speed will be lower, which is shown in the second page of Figure 59. The similar problem also occurs when testing the "Spiral" trajectory.

In addition, an example of this trajectory displayed using GPS data is shown in Figure 60. Compared to what can be seen in the field (all the circles performed have nearly the same center position), the reason could again be the limited accuracy of the GPS antenna.

Spiral Trajectory For the "Spiral" trajectory, tests with desired steering angles (from 30 deg to 0 deg) and maximum vehicle speeds (4 km/h and 6 km/h) are performed.

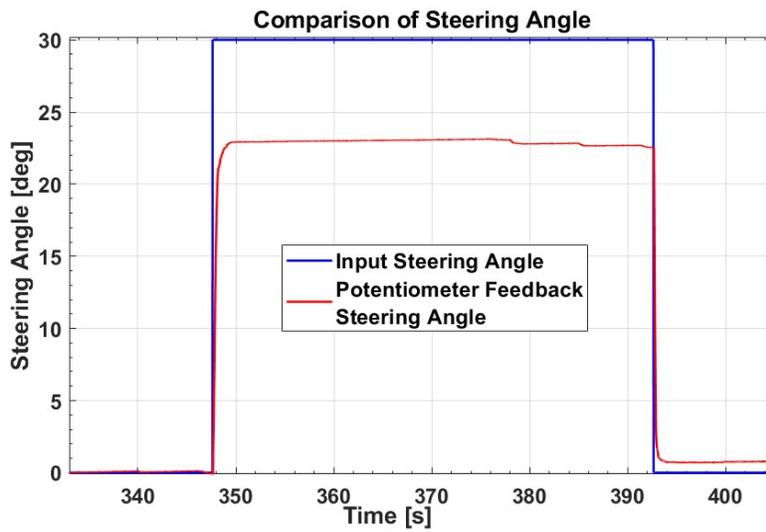
Figure 61 shows that when the maximum speed is set to be and achieved at 4 km/h, the maximum steering angle achieved is about 26 deg. While if the maximum speed is set to be 6 km/h and achieved at 4 km/h, the steering angle is only 24.5 deg. Proving once again that the real speed can affect the real steering angle performed. Moreover, by looking at Figure 59c and Figure 61d, it's clear to see that the maximum speed phase has fluctuations of different magnitudes, indicating that the trajectories performing higher steering angles will have a more unstable behaviour in terms of the vehicle speed.

In addition, an example of this trajectory displayed using GPS data is shown in Figure 62. Note that both at the start and the end of the simulation, there are abnormal recorded points. The reason could again be related to the GPS antenna.

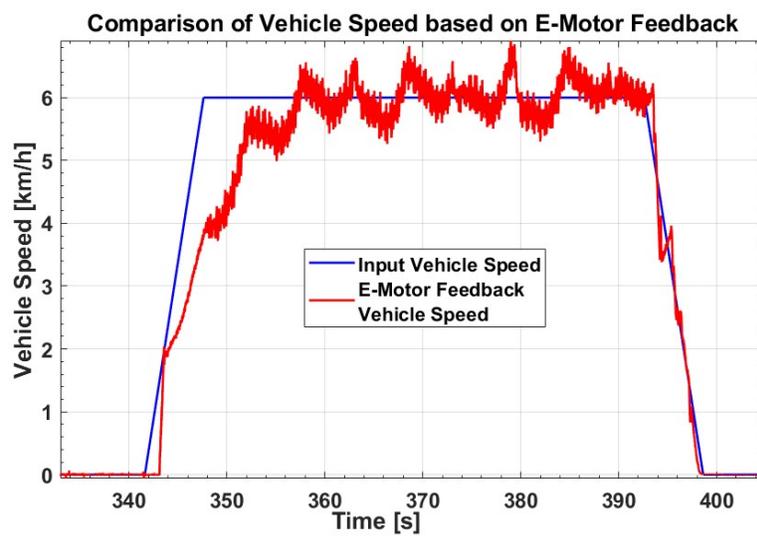
8-Shape Trajectory For the "8-Shape" trajectory, tests with strictly defined trajectories are performed.

Figure 63 shows that the designed steering vectors can be approximately performed and proves again that higher steering angles (30 deg or -30 deg) will make the vehicle speed have fluctuations of higher magnitudes. Again, an example of this trajectory displayed using GPS data is shown in Figure 64. Note that the straight lines designed to form the intersection phase are in parallel but do not coincide as designed, and the distance between them is much larger than what can be seen in the field.

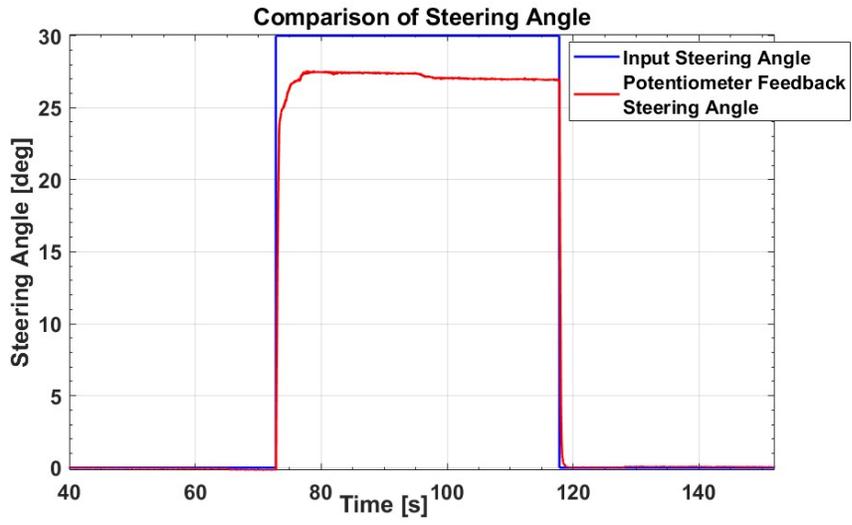
Sweep Trajectory For the "Sweep" trajectory, tests with different desired steering angles (30 deg, 20 deg, and 10 deg) and maximum vehicle speeds (4 km/h and 6 km/h) are performed.



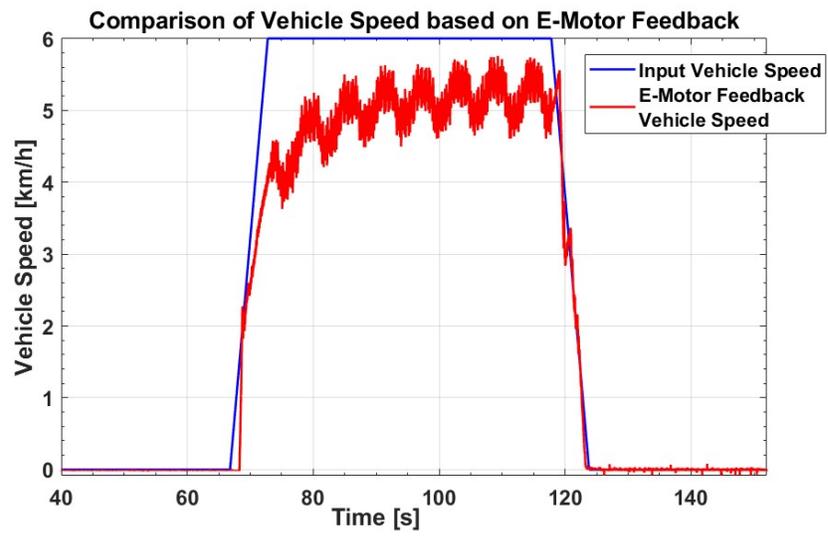
(a)



(b)



(c)



(d)

Figure 59: Circle Trajectory plots: 6 km/h; 30 deg.

Trajectory of Losi

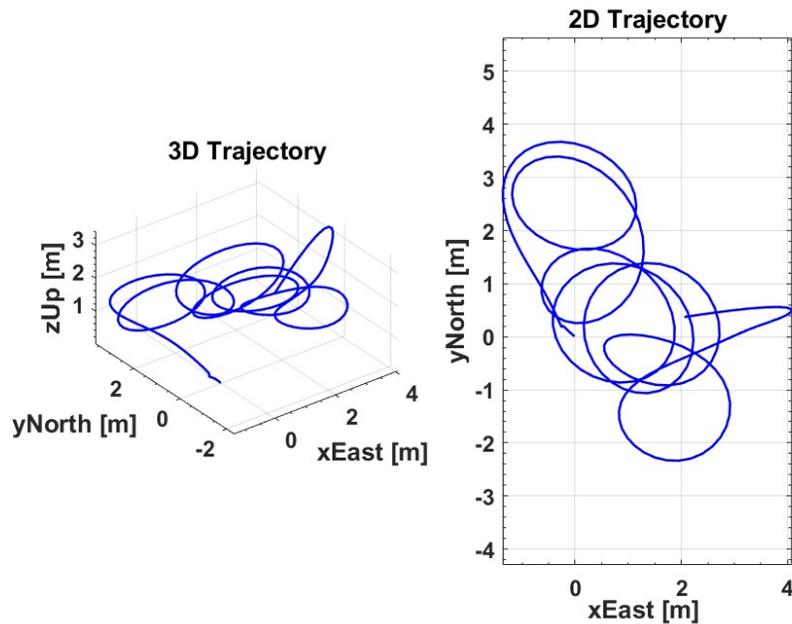
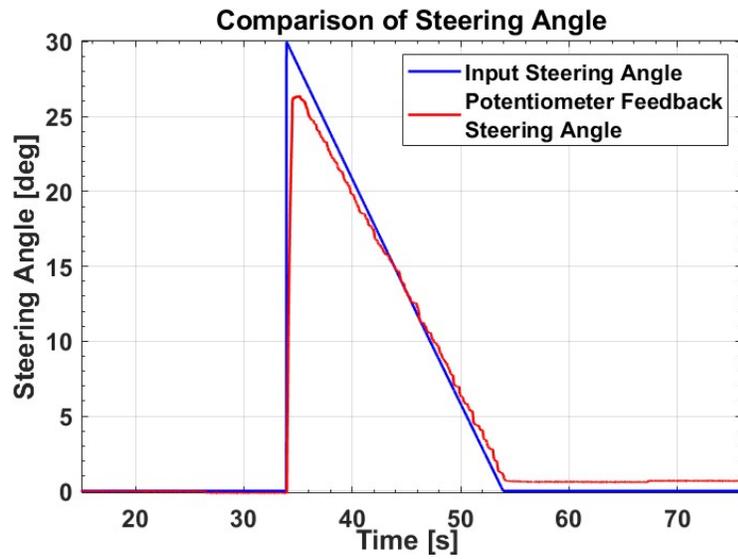


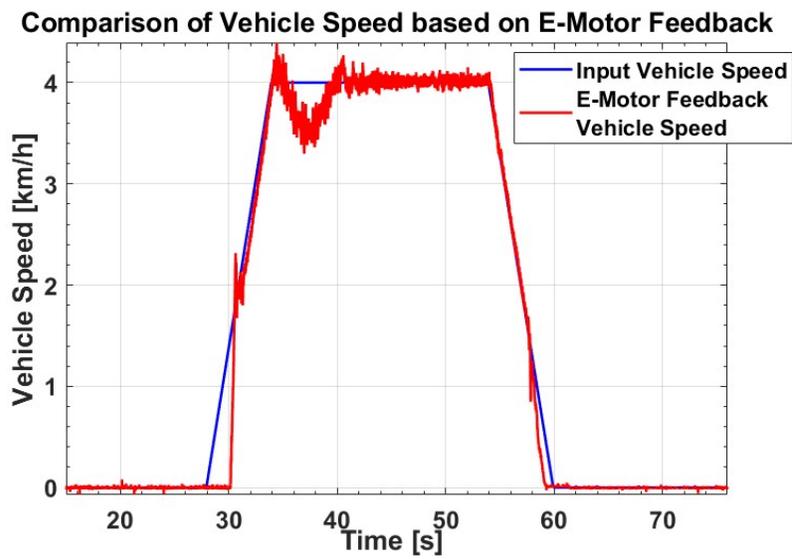
Figure 60: Circle Trajectory example: 4 km/h; 30deg

Comparison between two trajectories with different desired vehicle speeds is shown in Figure 65. It's clear to see that the input speed has negligible effect on the feedback steering angle or the yaw rate maximum magnitudes. However, the maximum magnitude of the lateral acceleration signal is increased.

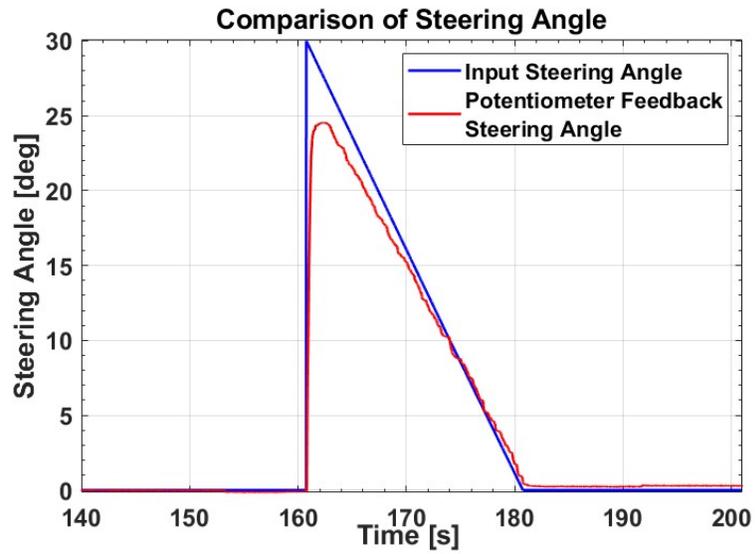
Considering that this trajectory is designed using the "chirp" function provided by MATLAB, apart from plotting the two steering angle signals in the time domain, which can be seen in Figure 66, it's useful to plot signals in the frequency domain. Therefore, several Bode plots are listed in the following to show how the steering angle changing frequency (from 0 Hz to 5 Hz) affects the magnitude and phase responses of the vehicle steering system. As is shown in Figure 67a, with the increasing frequency, the magnitude and the phase have decreasing trends. It means that the actual steering angle will be more shrunk and delayed with respect to the input one as the frequency increases, which is consistent with the feedback steering angle signal shown in Figure 66a. By focusing on Figure 67, it can be known that increasing the input vehicle speed from 4 km/h to 6 km/h has negligible effects on the magnitude and phase responses. The differences in the signals may come from the noise of the steering system. However, considering also the plots in Figure 68, it's clear that decreasing the input steering angle will make the absolute value of the magnitude response smaller. It means that, with the same instant steering angle changing frequency, the actual steering angle will be more affected in terms of magnitude if the input steering angle is large. Also means that, with the same time period, the actual steering angle will have less and slower shrinkage



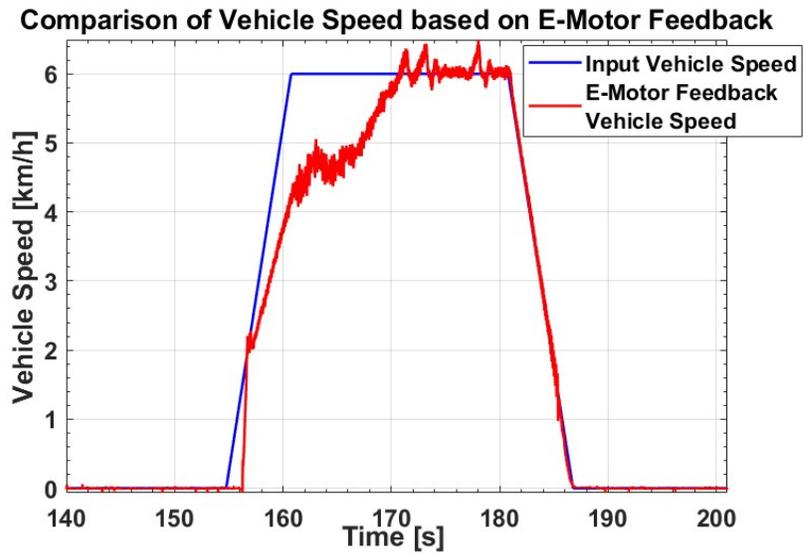
(a)



(b)



(c)



(d)

Figure 61: Spiral Trajectory plots: 4 km/h and 6 km/h; from 30 deg to 0 deg.

Trajectory of Losi

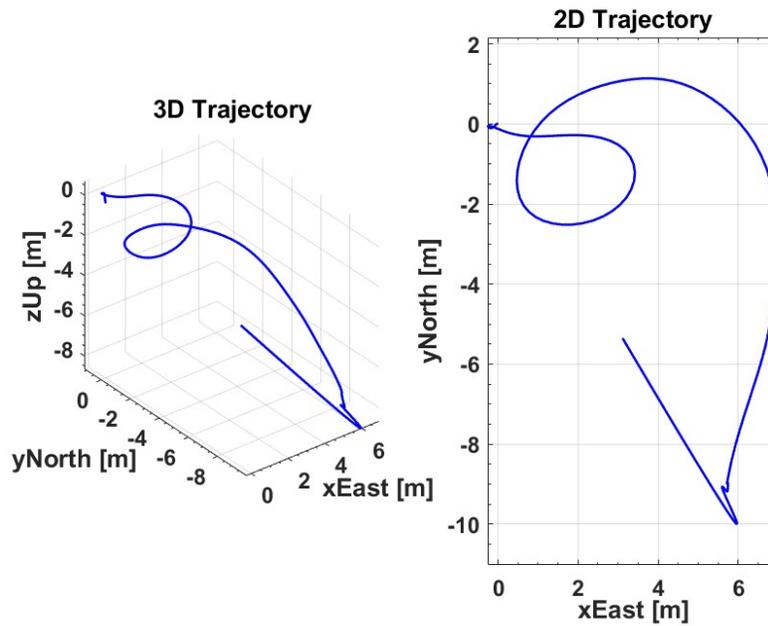
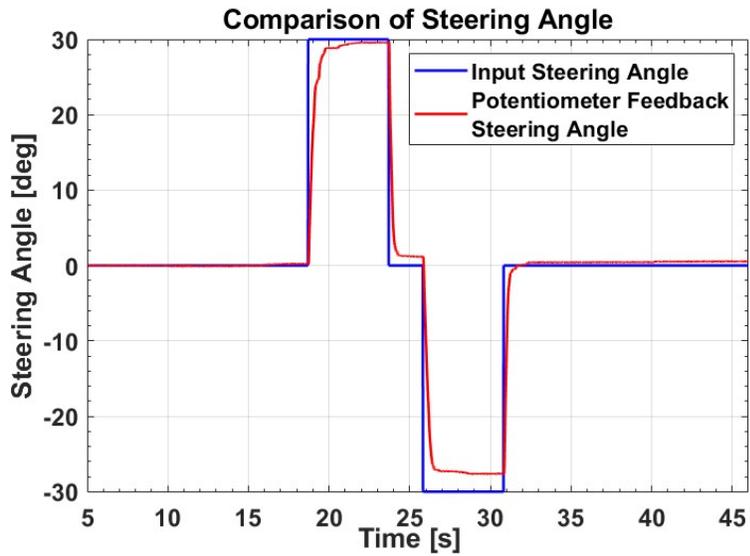


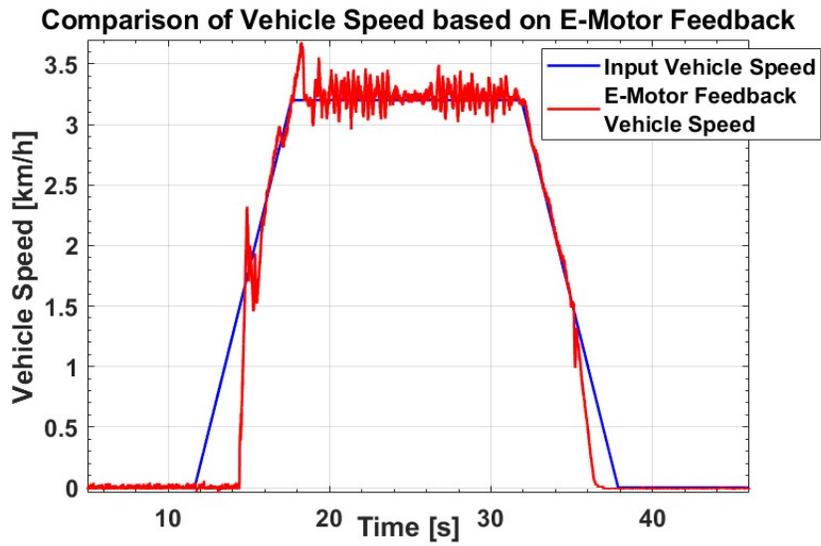
Figure 62: Spiral Trajectory example: 4 km/h; from 30 deg to 0 deg.

if the input steering angle is small, which is consistent with the feedback steering angle signals shown in Figure 66.

In summary, the five trajectories designed with different predefined steering and throttle vectors are able to be performed by the electrified Losi 5IVE-T 2.0 in wireless mode. And the data acquisition system developed can successfully receive and record real-time data to support post processing.



(a)



(b)

Figure 63: 8-Shape Trajectory plots: 4 km/h; 30 deg and -30 deg.

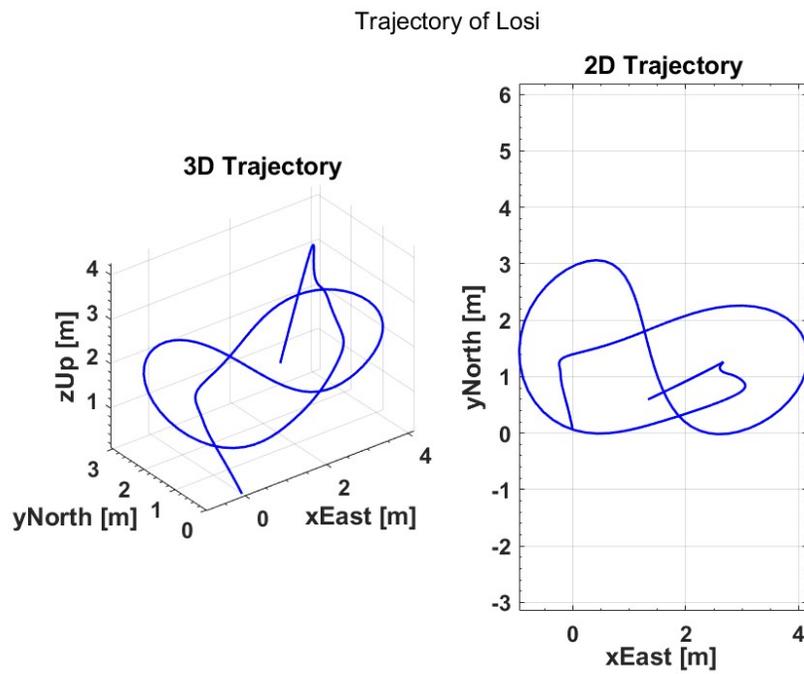
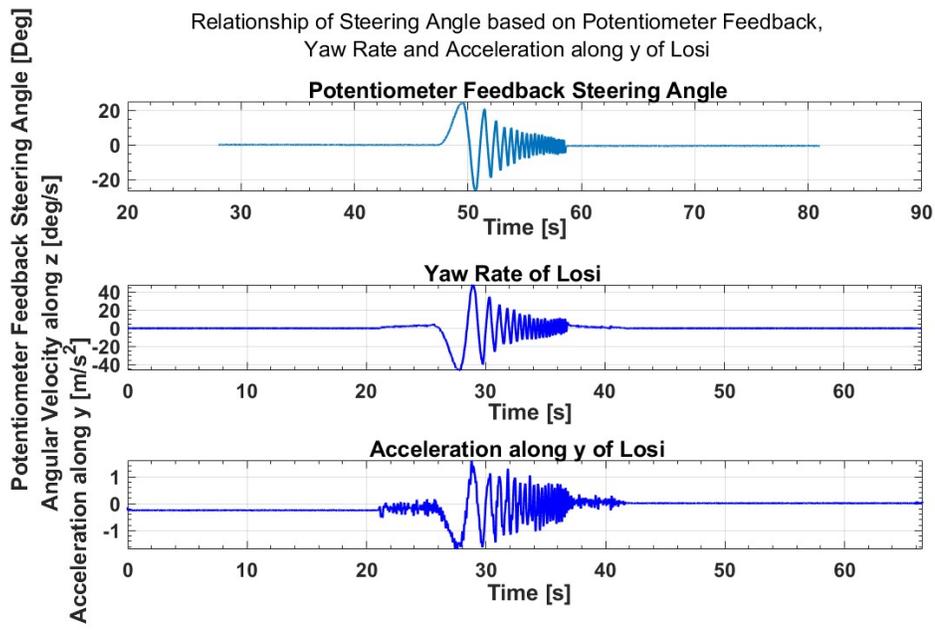
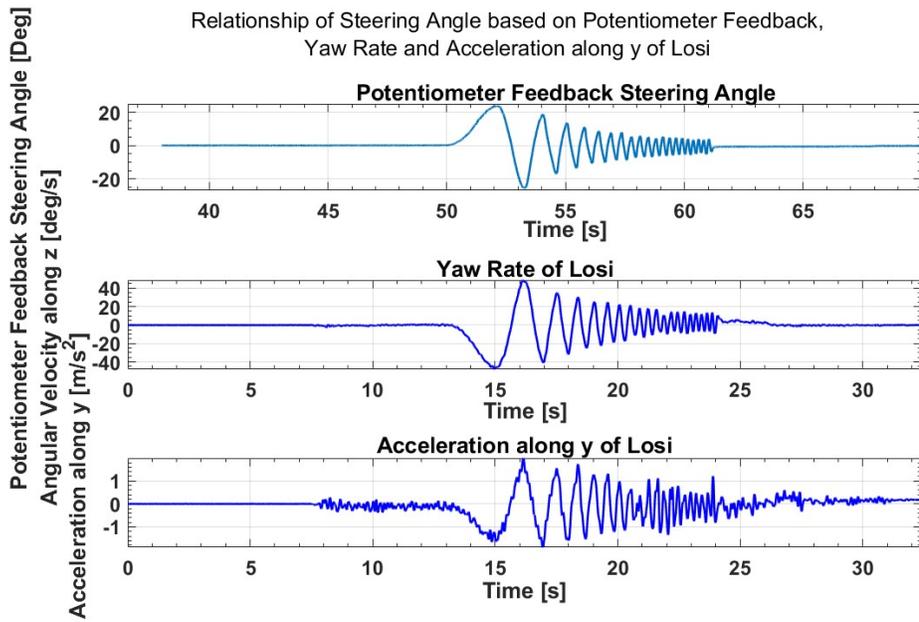


Figure 64: 8-Shape Trajectory example: 4 km/h; 30 deg and -30 deg.

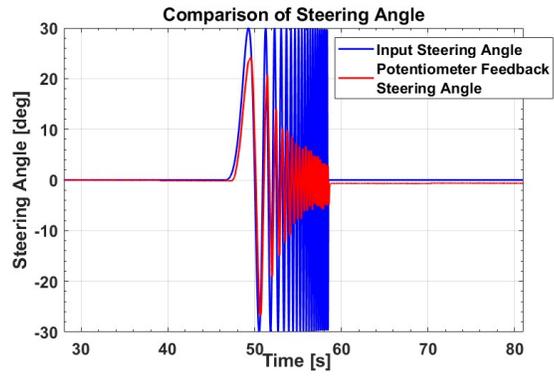


(a)

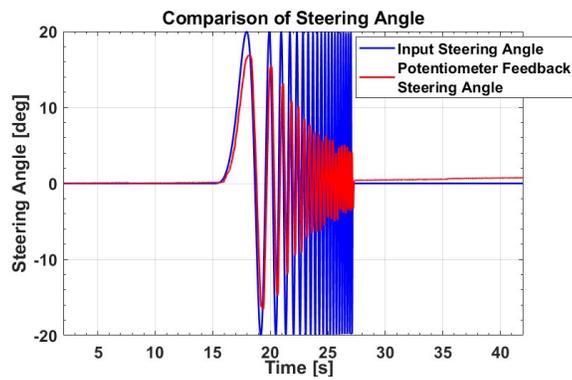


(b)

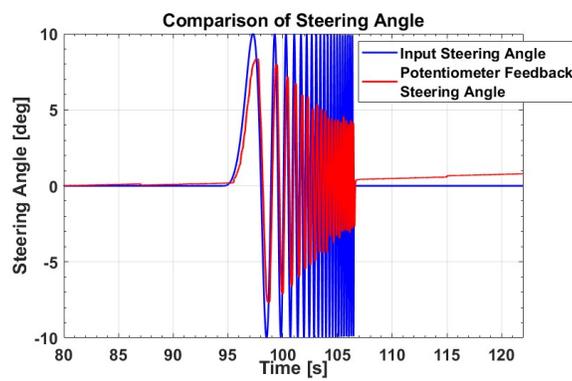
Figure 65: Sweep Trajectory plots: 4 km/h and 6 km/h; 30 deg.



(a)

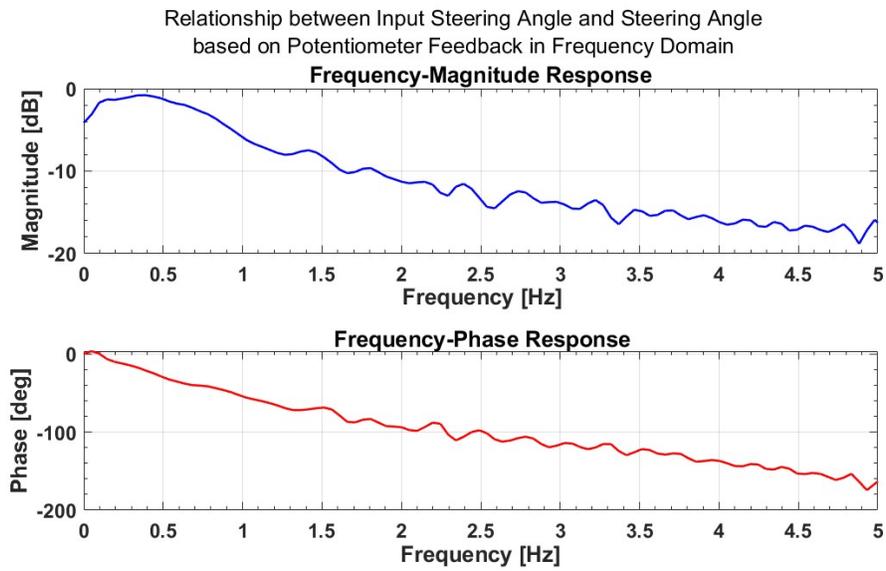


(b)

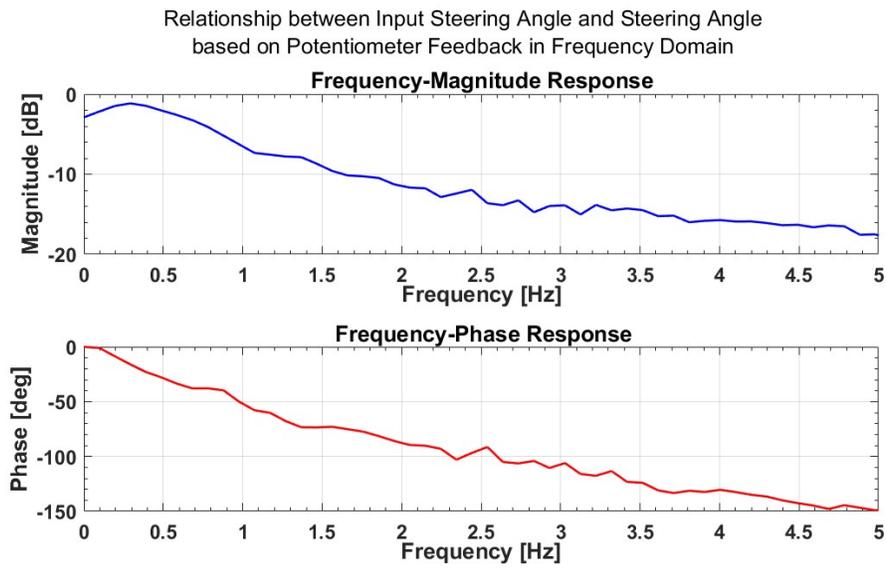


(c)

Figure 66: Sweep Trajectory time domain plots: 4 km/h; 30 deg, 20 deg, and 10 deg.



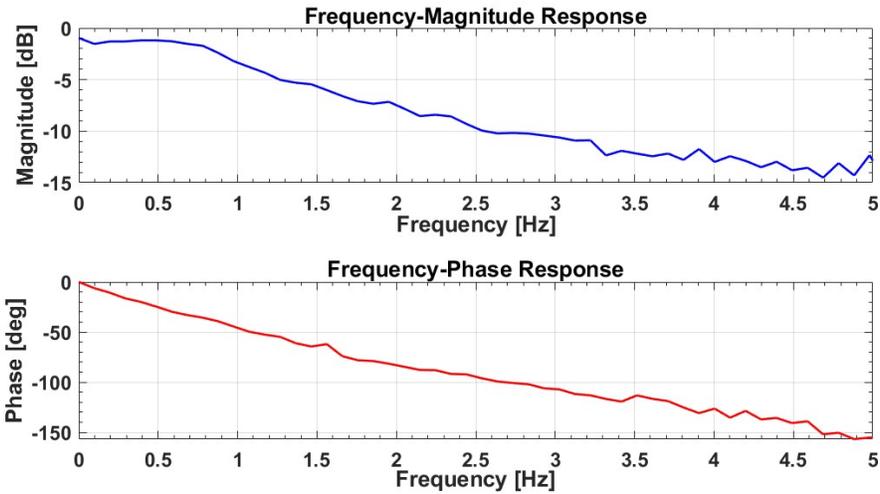
(a)



(b)

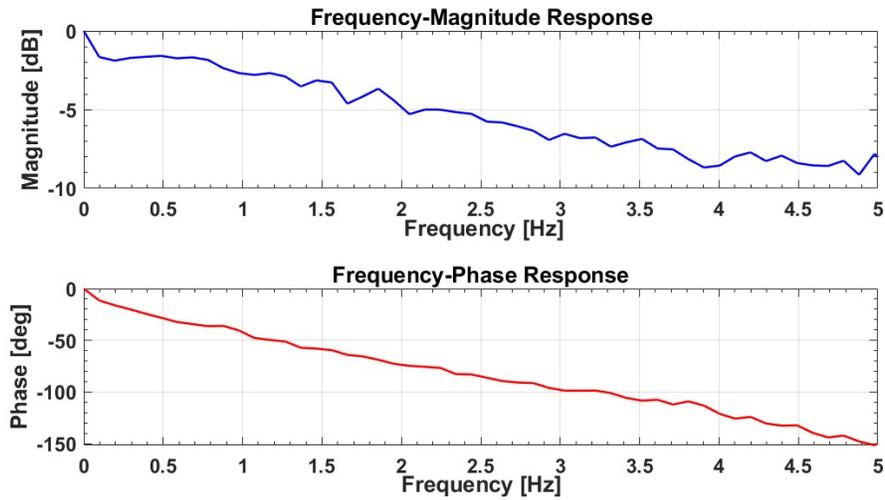
Figure 67: Sweep Trajectory frequency domain plots: 4 km/h and 6 km/h; 30 deg.

Relationship between Input Steering Angle and Steering Angle based on Potentiometer Feedback in Frequency Domain



(a)

Relationship between Input Steering Angle and Steering Angle based on Potentiometer Feedback in Frequency Domain



(b)

Figure 68: Sweep Trajectory frequency domain plots: 4 km/h; 20 deg, and 10 deg.

Conclusion

This thesis focuses on the feasibility of using the Speedgoat Baseline real-time target machine for precise control of the drive line and steering systems of the electrified version of the Losi 5IVE-T 2.0. Three major stages are described in detail, from the modification of the original Losi 5IVE-T 2.0 to the design of control modalities to the experimental validation in the field.

First of all, the hardware devices installed on the final setup of the Losi 5IVE-T 2.0, such as the e-motor and its controller to enable the electrified drive line, the stock steering servo motor to enable the steering function, the stock radio controller and the Speedgoat Baseline to provide command signals, the router to support wireless mode, and the batteries to provide supply voltages for some components, are introduced and explained in terms of their basic information and characteristics, functionalities, relationship with other components, and installation details. In particular, the selection among optional drive line designs based on some simplified formulas to predict the longitudinal and lateral behaviours of the vehicle is analysed and determined. Now, the vehicle is ready to move physically.

To manage and monitor the electronic signals routed to and from the e-motor, the e-motor controller is used. The purpose of the ESCON Studio software is to configure the controller based on the type, the specifications, and the required functionalities of the e-motor. Basic information about MATLAB and Simulink workflows used to build different real-time models based on different control modalities is followed. Details, such as MATLAB scripts written to define trajectories and Simulink real-time applications developed to control the vehicle, are explained in the chapter dedicated to the control modalities.

In order to validate the performance of the control modality using predefined vectors, a complete data acquisition system with multiple sensors is developed. The potentiometer is used to translate the steering angles performed at the wheel level into analog voltage values routed to the IO397 module. The GPS and the IMU are able to generate multiple signals that indicate position, speed, angular velocities, and linear accelerations. And the Scadas XS is used to receive and save those signals on its microSD card to support dynamic analysis of the vehicle. Thanks to the unit conversions and a low-pass filtering method introduced in the data post processing chapter, the results of the performed trajectories and the discussion about them are listed.

It turned out that the Losi 5IVE-T 2.0 is successfully upgraded to an electrified version with at least 6 km/h speed achieved. And the three control modalities work well with the installed components, including the data acquisition system. However, there are, for sure, further studies and efforts that could be done to explore the ability of this Losi 5IVE-T 2.0-based project:

- The control modality using dashboard blocks has the problem that one cannot simultaneously control both the steering and the throttle pathways as desired because each of them needs an action of the mouse. While the control modality using the radio controller is able to solve this problem, the signals are noisy and unstable due to various format conversions. Therefore, a possible solution could be introducing another external device, for example, a joystick, directly connected to the personal computer to replace the mouse.
- As mentioned in the chapter dedicated to the results and discussions, the uneven field ground

challenges and affects a lot the dynamic behaviour of the vehicle due to the poor rated torque provided by the selected e-motor. Therefore, it's suggested to switch to a new e-motor with higher available torque to improve the gradeability of the vehicle and obtain more reliable data from the field tests.

- The control models developed are open-loop ones. Knowing that there are driver blocks provided by the Speedgoat Real-Time Blockset package that support CAN communication between the Speedgoat Baseline and the other devices and that the SBG Ellipse-A IMU outputs signals through CAN, it's possible to directly send angular velocities and linear accelerations to the Speedgoat to participate in a closed-loop control strategy. The same effort should be made for the position and speed data outputted by the GPS. Finally, the Scadas XS is no longer needed because the IMU and the GPS are integrated and linked to the Speedgoat, and the data processing task is transferred from a post one to a real-time one. In addition, pay attention to the available sampling frequency of the signals.
- This thesis work introduces a few sensors, such as a simple potentiometer, a GPS antenna, and an IMU. For further study, new sensors, such as cameras, radars, and lidars, could be explored and implemented to enable the vehicle to perform autonomous driving tasks like lane keeping.

Overall, this thesis offers a valuable contribution to the topic of utilising a real-time target machine to control an electrified 1:5 scaled vehicle by presenting three distinct control modalities and suggesting potential interests for further development.

References

- [1] R. Verma, D. Del Vecchio, and H. K. Fathy, "Development of a scaled vehicle with longitudinal dynamics of an hmv for an its testbed," *IEEE/ASME transactions on mechatronics*, 2008.
- [2] R. T. O'Brien, J. A. Piepmeier, P. C. Hoblet, S. R. Burns, and C. E. George, "Scale-model vehicle analysis using an off-the-shelf scale-model testing apparatus," in *Proceedings of the 2004 American Control Conference*, IEEE, 2004.
- [3] R. G. Longoria, A. Al-Sharif, and C. B. Patil, "Scaled vehicle system dynamics and control: a case study in anti-lock braking," *International journal of vehicle autonomous systems*, 2004.
- [4] W. E. Travis, R. J. Whitehead, D. M. Bevly, and G. T. Flowers, "Using scaled vehicles to investigate the influence of various properties on rollover propensity," in *Proceedings of the 2004 American Control Conference*, IEEE, 2004.
- [5] A. Bulsara, A. Raman, S. Kamarajugadda, M. Schmid, and V. N. Krovi, "Obstacle avoidance using model predictive control: An implementation and validation study using scaled vehicles," *SAE Technical Paper Series*, 2020.
- [6] "Frequent application use cases using real-time target machine in automotive field."
<https://www.speedgoat.com/solutions/industries/automotive>.
- [7] "Losi 5ive-t 2.0 manual."
https://www.losi.com/on/demandware.static/-/Sites-horizon-master/default/dw98af01e0/Manuals/LOS05014V2_Manual.pdf.
- [8] "Zenoah g320rc manual."
<https://www.ddmracing.com/site/zenoah-g320rc-owners-manual-ddm.pdf>.
- [9] "Electrification and control of a scale vehicle, luca biondo."
<https://webthesis.biblio.polito.it/27478/>.
- [10] "1.5m 12t - 80t spur gear with step motor pinion gears product parameters table."
<https://www.ebay.co.uk/itm/125796344558>.
- [11] "Motor brushless rs 536-6030 data sheet."
<https://docs.rs-online.com/3b10/0900766b816e097d.pdf>.
- [12] "Nema 17 planetary gearbox eg17-g5 data sheet."
<https://www.ebay.it/itm/203650783418>.
- [13] "Vertigo performance losi 5ive gear chart."
<https://www.facebook.com/485810191429757/photos/losi-5ive-gear-chart-for-the-se-using-the-monster-locker-spur-gear-and-gear-hub/1154585514552218/>.
- [14] "20 teeth 1.5 module motor pinion gear with 8 mm hub diameter."
<https://www.ebay.co.uk/itm/125796344558>.
- [15] "8 mm motor shaft extension rod."
<https://www.amazon.com/8mmX100mm-Linear-Motion-Shaft-Diameter/dp/B08HYK9FJS?th=1>.

- [16] "5 mm to 8 mm motor shaft coupler."
<https://www.amazon.it/raccordi-passo-passo-connettore-alluminio-stampante/dp/B07P99778Q>.
- [17] "S900s 1/5 scale metal gear steering servo."
<https://www.lo.si.com/product/s900s-1-5-scale-digital-metal-gear-servo/LOSB0884.html>.
- [18] "Maxon motor escon 50/5 e-motor controller."
https://www.maxongroup.com/medias/sys_master/root/8834332262430/409510-ESCON-50-5-Hardware-Reference-En.pdf.
- [19] "Speedgoat baseline real-time target machine."
<https://www.speedgoat.com/products-services/real-time-target-machines/baseline-real-time-target-machine>.
- [20] "Io397 i/o module."
<https://www.speedgoat.com/products/simulink-programmable-fpgas-fpga-i-o-modules-io397>.
- [21] "Tp-link tl-mr3020 router."
<https://www.amazon.de/-/en/TP-Link-TL-MR3020-300Mbit-portable-repeater/dp/B0634PLTW>.
- [22] "Sbg ellipse-a imu."
<https://support.sbg-systems.com/sc/el/files/latest/44477063/44477047/1/1696266958828/Ellipse+3+-+Hardware+Manual.pdf>.
- [23] "100 kohm sigle-turn rotary potentiometer."
<https://www.radio741.com/69606-100kohm-100k-linear-plastic-potentiometer-piher.html>.
- [24] "3 mm to 6 mm potentiometer shaft coupler."
https://www.amazon.it/gp/product/B0C7ZGRXM9/ref=ox_sc_act_title_3?smid=A2QM2TRGDFFZ00&psc=1.
- [25] "40 mm m3 screw (without cap)."
<https://www.amazon.it/filettate-esagono-incassato-cupola-acciaio/dp/B098BMPH9Y>.
- [26] "Adjustable fixing bracket."
<https://www.amazon.it/fissaggio-regolabili-allungati-connettore-110x50x24/dp/B0C8LTH85Y>.
- [27] "Simcenter scadas xs."
<https://community.sw.siemens.com/s/article/simcenter-scadas-xs-everything-you-need-to-know#ref>.

A Appendix

Control with Dashboard Blocks codes

The following codes is contained in the "MATLAB Function" block.

```
1 function [OFF,FWD,BWD] = fcn(Sel)
2 switch Sel
3     case 0
4         OFF=1;
5         FWD=0;
6         BWD=0;
7     case 1
8         OFF=0;
9         FWD=1;
10        BWD=0;
11    otherwise 2
12        OFF=0;
13        FWD=0;
14        BWD=1;
15 end
```

Control with Radio Controller codes

The following script is used to obtain the actual throttle duty cycle limits:

```
1 %% Throttle DC Limits
2
3 Backward1 = 0.084;
4 RestPosition = 0.136;
5 Forward2 = 0.192;
6
7 %% Rest Position
8 load Filtered_Throttle_DC_RestPosition.mat
9
10 f1=figure('Name', 'Filtered Throttle DC at Rest Position');
11 plot(Filtered_Throttle_DC.time, Filtered_Throttle_DC.signals.values)
12 title('Filtered Throttle DC at Rest Position')
13 xlabel('Time [s]'), ylabel('Duty Cycle [0,1]')
14 set(gca, 'YLim', [0.13 0.142], 'Ytick', 0.13:0.002:0.142, 'XLim', [0
15     12.5])
16 grid on
17 % only focus on steady state
18 idx1=find(Filtered_Throttle_DC.time==6);
19 idx2=find(Filtered_Throttle_DC.time==8);
20
21 RestPosition_actual=mean(Filtered_Throttle_DC.signals.values(idx1:
22     idx2));
```

```

22 disp(['Actual Rest Position = ' num2str(RestPosition_actual)])
23
24 %% Backward(1)
25 load Filtered_Throttle_DC_Backward1.mat
26
27 f1=figure('Name', 'Filtered Throttle DC at Backward(1)');
28 plot(Filtered_Throttle_DC.time, Filtered_Throttle_DC.signals.values)
29 title('Filtered Throttle DC at Backward(1)')
30 xlabel('Time [s]'), ylabel('Duty Cycle [0,1]')
31 set(gca, 'YLim', [0.08 0.145], 'Ytick', 0.08:0.002:0.145, 'XLim', [0
    18.5])
32 grid on
33
34 % only focus on steady state
35 idx1=find(Filtered_Throttle_DC.time==12);
36 idx2=find(Filtered_Throttle_DC.time==14);
37
38 Forward2_actual=mean(Filtered_Throttle_DC.signals.values(idx1:idx2))
    ;
39 disp(['Actual Backward(1) = ' num2str(Forward2_actual)])
40
41 %% Forward(2)
42 load Filtered_Throttle_DC_Forward2.mat
43
44 f1=figure('Name', 'Filtered Throttle DC at Forward(2)');
45 plot(Filtered_Throttle_DC.time, Filtered_Throttle_DC.signals.values)
46 title('Filtered Throttle DC at Forward(2)')
47 xlabel('Time [s]'), ylabel('Duty Cycle [0,1]')
48 set(gca, 'YLim', [0.13 0.195], 'Ytick', 0.13:0.002:0.195, 'XLim', [0
    19])
49 grid on
50
51 % only focus on steady state
52 idx1=find(Filtered_Throttle_DC.time==12);
53 idx2=find(Filtered_Throttle_DC.time==14);
54
55 Forward2_actual=mean(Filtered_Throttle_DC.signals.values(idx1:idx2))
    ;
56 disp(['Actual Forward(2) = ' num2str(Forward2_actual)])

```

The following codes is contained in the "MATLAB Function" block, named "Steering Startup Limitations":

```

1 function Steering_DC= SteeringStartup (Steering_DC_Filt,
    noise_tol_perc, Steer_lim, Rest_DC)
2
3 %% Tolerances Calculation

```

```

4
5 % additional noise tolerance
6 noise_tol = (noise_tol_perc/100)*Rest_DC; % Noise tolerance equal to
   7% of the rest value (check the model workspace)
7
8 %% Startup
9
10 % avoiding abrupt movements overcoming the servo motor PWM signal
   duty cycle limits
11
12 if Steering_DC_Filt > Steer_lim(2) + noise_tol
13     Steering_DC = Rest_DC;
14 elseif Steering_DC_Filt < Steer_lim(1) - noise_tol
15     Steering_DC = Rest_DC;
16 else
17     Steering_DC = Steering_DC_Filt;
18 end
19
20 end

```

The following codes is contained in the "MATLAB Function" block, named "Switch by DC & Speed":

```

1 function Sel = FWD2BWD(Throttle_DC, Ang_Vel, Forward, Backward,
   Rest_DC, noise_tol_perc, speed_tol)
2
3 % INPUT
4 %     Throttle_DC     Filtered duty cycle range: (0,1)
5 %     Ang_Vel         Angular velocity read by the motor
   controller [rpm]
6 %     Forward         Duty cycle range in forward mode with
   measured tolerance: [0.13744 0.1923]
7 %     Backward        Duty cycle range in backward mode with
   measured tolerance: [0.082613 0.13744]
8 %     Rest_DC         Duty cycle at rest position: 0.136
9 %     noise_tol_perc  Tolerance percentage value for noise
   compensation: 7
10 %     speed_tol       Tolerance for smoothing abrupt behaviours
   when changing mode [rpm]: 50
11 % OUTPUT
12 %     Sel             Variable for selecting the direction of
   motion
13 %
14 %                 Sel = 0, Rest position
15 %                 Sel = 1, Forward motion
16 %                 Sel = 2, Backward motion
17 %% Tolerances Calculation

```

```

18
19 % additional noise tolerance
20 noise_tol = (noise_tol_perc/100)*Rest_DC; % Noise tolerance equal to
      7% of the rest value (check the model workspace)
21
22 %% Startup
23
24 % avoiding abrupt movements overcoming the servo motor PWM signal
      duty cycle limits
25
26 if Throttle_DC > Forward(2) + noise_tol
27     Throttle_DC = Rest_DC;
28 end
29 if Throttle_DC < Backward(1) - noise_tol
30     Throttle_DC = Rest_DC;
31 end
32
33 %% Logic
34
35 if (Throttle_DC > Rest_DC+noise_tol && Ang_Vel > speed_tol) || (
      Throttle_DC < Rest_DC+noise_tol && Ang_Vel > speed_tol) ...
36     || (Throttle_DC > Rest_DC+noise_tol && Ang_Vel < speed_tol
          && Ang_Vel > -speed_tol)
37     Sel = 1;
38 elseif (Throttle_DC < Rest_DC-noise_tol && Ang_Vel < -speed_tol) ||
      (Throttle_DC > Rest_DC-noise_tol && Ang_Vel < -speed_tol) ...
39     || (Throttle_DC < Rest_DC-noise_tol && Ang_Vel < speed_tol
          && Ang_Vel > -speed_tol)
40     Sel = 2;
41 else
42     Sel = 0;
43 end
44
45 end

```

Control with Predefined Vectors

The complete "Acceleration-Deceleration" trajectory script:

```

1 %% Inputs
2
3 speed = 13.226; % [km/h] Vehicle Speed, max = 13.226
4 t_maxspeed = 15; % [s] Time Period of the MaxSpeed Phase
5
6 disp(' ')
7 disp(['Input Vehicle Speed: ' num2str(speed) 'km/h'])
8 disp(['Input Max Speed Time: ' num2str(t_maxspeed) 's'])

```

```

9
10 %% Vehicle Data
11
12 R_tyre = 90/1000; % [m] Tire Radius
13 tau_total = 1*(62/20)*3.31; % Total Gear Ratio
14
15 %% Speed Data
16
17 % Speed
18 w_eMot = (speed*tau_total/(3.6*R_tyre))*60/(2*pi); % [rpm] eMotor
    speed
19 throttle_DC = interp1(speed_range, throttle_range, w_eMot); %
    throttle DC
20 if isnan(throttle_DC)
21     disp ("Speed out of range [0 13.226] [km/h]")
22 end
23
24
25 %% Trajectory Data
26
27 % times
28 t_rest = 1.5; % [s] Time before starting the maneuver
29 t_total = 2*t_rest+t_maxspeed; % [s] Time to complete the whole test
30
31 % steps computation
32 timestep = 0.01; % [s] Timestep
33 rest_steps = t_rest/timestep; % Steps at steady state
34 maxspeed_steps = t_maxspeed/timestep; % Steps related to the chirp
    path
35 steps=t_total/timestep; % Total steps number
36
37 %% Steering Path
38
39 % Steering
40 steer.timeacceldecel=linspace(0,t_total,steps);
41 steer.posacceldecel=(((ones(1,steps)*steering_rest)...
42     -steering_rest)/((steer_range(3)-steer_range(1))/2));
43     % -steering_rest)/((steer_range(3)-steer_range(1))/2)
44     % is the normalization procedure
45
46 %% Throttle Path
47
48 % Trajectory to be followed - Throttle
49 thr.timeacceldecel=steer.timeacceldecel;
50 thr.posacceldecel=(((ones(1,rest_steps)*throttle_rest,ones(1,
    maxspeed_steps)*throttle_DC,...

```

```

51     ones(1,rest_steps)*throttle_rest)]...
52     -throttle_rest)/(throttle_range(3)-throttle_rest));
53     % -throttle_rest)/(throttle_range(3)-throttle_rest)
54     % is the normalization procedure based on FWD direction
55     % (emphasizing FWD direction is because the FWD range and the
56     %   BWD range
57     % of throttling don't have the same length, 0.136-0.084=0.052
58     %   for BWD
59     % and 0.19-0.136=0.054 for FWD)

```

The complete "Circle" trajectory script:

```

1  %% Inputs
2
3  theta_deg = 32.5; % [deg] Steering Angle, max = 32.5
4  theta_rad = deg2rad(theta_deg);
5  speed = 13.226; % [km/h] Vehicle Speed, max = 13.226
6  t_maxspeed = 45; % [s] Time Period of the Circle Phase
7
8  disp(' ')
9  disp(['Input Steering Angle: ' num2str(theta_deg) 'deg'])
10 disp(['Input Vehicle Speed: ' num2str(speed) 'km/h'])
11 disp(['Input Circle Time: ' num2str(t_maxspeed) 's'])
12
13 %% Vehicle Data
14
15 R_tyre = 90/1000; % [m] Tire Radius
16 tau_total = 1*(62/20)*3.31; % Total Gear Ratio
17
18 %% Steering Data
19
20 % Angle
21 steer_DC = interp1(angle_range, steer_range, theta_deg); % Steering
22   DC
23 if isnan(steer_DC)
24     disp ("Angle out of range [-32.5 32.5] [deg]")
25 end
26
27 %% Speed Data
28
29 % Speed
30 w_eMot = (speed*tau_total/(3.6*R_tyre))*60/(2*pi); % [rpm] eMotor
31   Speed
32 throttle_DC = interp1(speed_range, throttle_range, w_eMot); %
33   Throttle DC
34 if isnan(throttle_DC)
35     disp ("Speed out of range [0 13.226] [km/h]")

```

```

33 end
34
35 %% Trajectory Data
36
37 % times
38 t_rest = 1.5; % [s] time before starting the maneuver
39 t_acc = 6; % [s] time to reach the test speed
40 t_motion = t_maxspeed+2*t_acc; % [s] time in which the vehicle
    moves = acceleration and test phases
41 t_total = 2*t_rest+t_motion; % [s] Time to complete the whole test
42
43 % steps computation
44 timestep = 0.01; % [s] Timestep
45 rest_steps = t_rest/timestep; % Steps at steady state
46 acc_steps = t_acc/timestep; % Steps for the acceleration phases
47 maxspeed_steps = t_maxspeed/timestep; % Steps related to the
    circular path
48 motion_steps = t_motion/timestep; % Steps related to the motion path
49 steps=t_total/timestep; % Total steps number
50
51 %% Steering Path
52
53 % Trajectory to be followed - Steering
54 steer.timecirc=linspace(0,t_total,steps);
55 steer.poscirc=(((ones(1,rest_steps+acc_steps)*steering_rest,ones(1,
    maxspeed_steps)*steer_DC,...
56     ones(1,acc_steps+rest_steps)*steering_rest])-steering_rest)/((
    steer_range(3)-steer_range(1))/2));
57
58 %% Throttle Path
59
60 % Trajectory to be followed - Throttle
61 thr.timecirc=steer.timecirc;
62 thr.poscirc=(((ones(1,rest_steps)*throttle_rest, linspace(
    throttle_rest,throttle_DC,acc_steps),...
63     ones(1,maxspeed_steps)*throttle_DC,linspace(throttle_DC,
    throttle_rest,acc_steps),...
64     ones(1,rest_steps)*throttle_rest))-throttle_rest)/(
    throttle_range(3)-throttle_rest));

```

The "Steering Path" part of the "Spiral" script:

```

1 %% Steering Path
2
3 % Trajectory to be followed - Steering
4 steer.timespir2=linspace(0,t_total,steps);
5 steer.posspir2=(((ones(1,rest_steps+acc_steps)*steering_rest, ...

```

```

6     linspace(steer_DC,steering_rest,maxspeed_steps), ...
7     ones(1,acc_steps+rest_steps)*steering_rest]-steering_rest)/((
        steer_range(3)-steer_range(1))/2));

```

The "Steering Path" part of the "Sweep" script:

```

1 %% Steering Path
2
3 % Trajectory to be followed - Steering
4 steer.timechirp=linspace(0,t_total,steps);
5 steer.poschirp=(((ones(1,rest_steps+acc_steps)*steering_rest,...
6     (-chirp(t0,freq0,t1,freq1,'quadratic',90)...
7     *((steer_DC-steering_rest)*100)+steering_rest*100)/100,...
8     ones(1,rest_steps+acc_steps)*steering_rest)]...
9     -steering_rest)/((steer_range(3)-steer_range(1))/2));
10 % chirp phi = 90 to have a smooth increase from 0 to max value
    of y axis
11 % *((steer_DC-steering_rest)*100)+steering_rest*100
12 % is the chirp normalization procedure

```

The complete "8-Shape" trajectory script:

```

1 %% Inputs
2
3 theta_deg = 30; % [deg] Steering Angle desired, max = 32.5
4 theta_rad = deg2rad(theta_deg);
5 speed = 4; % [km/h] Vehicle Speed desired, max = 13.226
6 t_maxspeed = 52.36; % [s] Time Period of completing one 8 shape =
    (2*straightline_steps+2*unfullcircle_steps)*timestep
7
8 %% Vehicle Data
9
10 M = 25; % [kg] Vehicle Mass _ estimated
11 R_tyre = 90/1000; % [m] Tire Radius
12 T_eM = 0.25; % [Nm] eMotor Rated Torque
13 tau_total = 1*(62/20)*3.31; % Total Gear Ratio
14 eta = 1; % Total Transimission Efficiency assumed to be 1
15 F_r = 0; % Total Resistance Force assumed to be 0
16
17 T_t = T_eM*tau_total*eta; % [Nm] Total Torque at Tire Level _
    estimated
18 a_x = (T_t/R_tyre-F_r)/M; % [m/s2] Vehicle Longitudinal Acceleration
    at eMotor Rated Rotational Speed _ estimated
19 t_acc_min = (speed/3.6)/a_x; % [t] Minimum Acceleration Time to
    achieve Desired Vehicle Speed _ estimated assuming constant
    acceleration motion
20 R = 2*M*R_tyre*(speed/3.6)^2/(T_t*sin(theta_rad)); % [m] Instant
    Curvature Radius _ estimated

```

```

21 L_circle = 2*pi*R; % [m] Peripheral of Circle with R
22 t_8 = 2*(L_circle/4*3/(speed/3.6))+4*(R/(speed/3.6)); % [s] Time to
    complete an 8 shape
23 num_8 = t_maxspeed/t_8; % Numbers of completed 8 shapes
24
25 disp(' ')
26 disp(['Minimum acceleration time to achieve desired vehicle speed: '
    num2str(t_acc_min) 's'])
27 disp(['Curvature radius: ' num2str(R) 'm'])
28 disp(['Time needed to complete a single 8 shape: ' num2str(t_8) 's'
    ])
29 disp(['Numbers of completed 8 shapes: ' num2str(num_8)])
30
31 %% Steering Data
32
33 % Angle
34 steer_DC = interp1(angle_range, steer_range, theta_deg); % Steering
    DC
35 steer_DC_opposed = interp1(angle_range, steer_range, -theta_deg); %
    Steering DC corresponds to opposite Steering Angle
36 % Steering angles are symmetric to 0 deg, while steering DC are not
37 % symmetric. So cannot simply use -steer_DC in steer.pos
38 if isnan(steer_DC)
39     disp ("Angle out of range [-32.5 32.5] [deg]")
40 end
41
42 %% Speed Data
43
44 % Speed
45 w_eMot = (speed*tau_total/(3.6*R_tyre))*60/(2*pi); % [rpm] eMotor
    Speed
46 throttle_DC = interp1(speed_range, throttle_range, w_eMot); %
    Throttle DC
47 if isnan(throttle_DC)
48     disp ("Speed out of range [0 13.226] [km/h]")
49 end
50
51 %% Trajectory Data
52
53 % times
54 t_rest = 1.5; % [s] Time before starting the maneuver
55 t_acc = 6; % [s] Time to reach the test speed
56 t_straightline = 2*(R/(speed/3.6)); % [s] Time Period defined to
    complete one whole straight line at the intersection of 8 shape
57 t_unfullcircle = L_circle/4*3/(speed/3.6); % [s] Time Period defined
    to complete 270 deg of one circle peripheral

```

```

58 t_motion = t_maxspeed+2*t_acc; % [s] Time in which the vehicle moves
    = acceleration and maxspeed phases
59 t_total = 2*t_rest+t_motion; % [s] Time to complete the whole test
60
61 % steps computation
62 timestep = 0.01; % [s] Timestep
63 rest_steps = t_rest/timestep; % Steps at steady state
64 acc_steps = t_acc/timestep; % Steps for the acceleration phases
65 straightline_steps = ceil(t_straightline/timestep); % Steps related
    to the one whole straight line at the intersection of 8 shape
66 unfullcircle_steps = ceil(t_unfullcircle/timestep); % Steps related
    to the 270 deg of one circle peripheral
67 maxspeed_steps = t_maxspeed/timestep; % Steps of completing two 8
    shapes
68 motion_steps = t_motion/timestep; % Steps related to the motion path
69 steps=t_total/timestep; % Total steps number
70
71 %% Steering Path
72
73 % Trajectory to be followed - Steering
74 steer.time8_4=linspace(0,t_total,steps);
75 steer.pos8_4=(((ones(1,rest_steps+acc_steps)*steering_rest,...
76     ... 8 shape start
77     ones(1,straightline_steps/2)*steering_rest,...
78     ones(1,unfullcircle_steps)*steer_DC, ...
79     ones(1,straightline_steps)*steering_rest,...
80     ones(1,unfullcircle_steps)*steer_DC_opposed, ...
81     ones(1,straightline_steps/2)*steering_rest,...
82     ... 8 shape end
83     ones(1,acc_steps+rest_steps)*steering_rest]) ...
84     -steering_rest)/((steer_range(3)-steer_range(1))/2));
85
86 %% Throttle Path
87
88 % Trajectory to be followed - Throttle
89 thr.time8_4=steer.time8_4;
90 thr.pos8_4=(((ones(1,rest_steps)*throttle_rest, ...
91     linspace(throttle_rest,throttle_DC,acc_steps),...
92     ones(1,maxspeed_steps)*throttle_DC, ...
93     linspace(throttle_DC,throttle_rest, acc_steps)...
94     ones(1,rest_steps)*throttle_rest])-throttle_rest)/(
    throttle_range(3)-throttle_rest));

```

The script used to visualise the "8-Shape" trajectory:

```

1 %% Trajectory Plot
2

```

```

3 % Single 8 Shape
4 t_maxspeed_single = 52.36; % [s] Time Period of completing one 8
   shape = (2*straightline_steps+2*unfullcircle_steps)*timestep
5 maxspeed_single_steps = t_maxspeed_single/timestep; % Steps of
   completing one 8 shape
6
7 initial_pose = [0; 0; 0]; % Initial Position [x, y, theta]
8 velocity_vector = ((([ones(1,rest_steps)*throttle_rest, ...
9   linspace(throttle_rest,throttle_DC,acc_steps),...
10  ones(1,maxspeed_single_steps)*throttle_DC, ...
11  linspace(throttle_DC,throttle_rest, acc_steps)...
12  ones(1,rest_steps)*throttle_rest])-throttle_rest)/(
   throttle_range(3)-throttle_rest))*13.226/3.6; % [m/s] Vehicle
   Speed Vector, max = 13.226/3.6
13 steering_angle_vector = ((([ones(1,rest_steps+acc_steps)*
   steering_rest,...
14   ... 8 shape start
15   ones(1,straightline_steps/2)*steering_rest,...
16   ones(1,unfullcircle_steps)*steer_DC, ...
17   ones(1,straightline_steps)*steering_rest,...
18   ones(1,unfullcircle_steps)*steer_DC_opposed, ...
19   ones(1,straightline_steps/2)*steering_rest,...
20   ... 8 shape end
21   ones(1,acc_steps+rest_steps)*steering_rest]) ...
22   -steering_rest)/((steer_range(3)-steer_range(1))/2))*32.5; % [
   deg] Steering Angle Vector, max = 32.5
23
24 % Save updated pose to trajectory
25 num_steps = length(steering_angle_vector);
26 trajectory = zeros(num_steps, 3);
27 trajectory(1, :) = initial_pose;
28
29 for i = 2:num_steps
30   delta_time = timestep;
31   pose = trajectory(i - 1, :);
32   velocity = velocity_vector(i);
33   steering_angle = steering_angle_vector(i);
34   if steering_angle == 0
35     trajectory(i, :) = update_pose(pose, velocity, delta_time);
36   elseif steering_angle > 0
37     trajectory(i, :) = update_pose_steer_right(pose, velocity,
   delta_time);
38   else
39     trajectory(i, :) = update_pose_steer_left(pose, velocity,
   delta_time);
40   end

```

```

41 end
42
43 figure;
44 plot(trajectory(:, 1), trajectory(:, 2));
45 title('8 Shape Trajectory with Theoretical Curvature Radius');
46 xlabel('X-axis [m]');
47 ylabel('Y-axis [m]');
48 axis equal;
49 grid on;
50
51 function new_pose = update_pose(pose, velocity, delta_time)
52     x = pose(1);
53     y = pose(2);
54     theta = pose(3);
55     % Vehicle dynamics
56     x_dot = velocity * cos(theta);
57     y_dot = velocity * sin(theta);
58     % Update pose using dynamics
59     x = x + x_dot * delta_time;
60     y = y + y_dot * delta_time;
61     theta = theta;
62     new_pose = [x, y, theta];
63 end
64
65 function new_pose = update_pose_steer_right(pose, velocity,
delta_time)
66     x = pose(1);
67     y = pose(2);
68     theta = pose(3);
69     % Vehicle parameters
70     R = 4.3314;
71     % Vehicle dynamics
72     x_dot = velocity * cos(theta);
73     y_dot = velocity * sin(theta);
74     theta_dot = velocity / R;
75     % Update pose using dynamics
76     x = x + x_dot * delta_time;
77     y = y + y_dot * delta_time;
78     theta = theta + theta_dot * delta_time;
79     new_pose = [x, y, theta];
80 end
81
82 function new_pose = update_pose_steer_left(pose, velocity,
delta_time)
83     x = pose(1);
84     y = pose(2);

```

```

85     theta = pose(3);
86     % Vehicle parameters
87     R = 4.3314;
88     % Vehicle dynamics
89     x_dot = velocity * cos(theta);
90     y_dot = velocity * sin(theta);
91     theta_dot = velocity / R;
92     % Update pose using dynamics
93     x = x + x_dot * delta_time;
94     y = y + y_dot * delta_time;
95     theta = theta - theta_dot * delta_time;
96     new_pose = [x, y, theta];
97 end

```

The starting script:

```

1 %% Parameters
2
3 % throttle parameters
4 speed_range = [-4000,0,4000]; % [rpm] Input eMotor Angular Speed
   Range
5 throttle_range = [8.4,13.6,19]/100; % Throttle DC Range
6 throttle_rest = 13.6/100; % Throttle Rest Position
7
8 % steering parameters
9 angle_range = [-32.5,0,32.5]; % [deg] Input Steering Angle Range
10 steer_range = [10,13.6,17.2]/100; % Steering DC Range
11 steering_rest = 13.6/100; % Steering Rest Position
12
13 %% Acceleration-Deceleration Trajectory
14 disp(' ')
15 disp ("Acceleration-Deceleration Trajectory Parameters")
16 Accel_Decel_Script_SL
17
18 %% Circle Trajectory
19 disp(' ')
20 disp ("Circle Trajectory Parameters")
21 Trajectory_Circle_Script_SL
22
23 %% Spiral Trajectory
24 disp(' ')
25 disp ("Spiral Trajectory Parameters")
26 Trajectory_Spiral_Script_SL
27
28 %% 8 Shape Trajectory_constant 4 km/h
29 disp(' ')
30 disp ("8 Shaped Trajectory_constant 4 km/h Parameters")

```

```
31 Trajectory_8_Shape_4_Script_SL
32
33 %% Sweep Trajectory
34 disp(' ')
35 disp ("Chirp Trajectory Parameters")
36 Steer_Chirp_Script_SL
```