



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



# **Simulation of Automobile Chassis Motion using a Gough-Stewart Platform**

Development of a Test Rig for Analysis of Adaptive Headlights Dynamics

CALLE ARKEVALL

CHRISTOFFER SANDLUND





MASTER'S THESIS 2017:SSYX04

# Simulation of Automobile Chassis Motion using a Gough-Stewart Platform

Development of a Test Rig for Analysis of Adaptive Headlights Dynamics

CALLE ARKEVALL  
CHRISTOFFER SANDLUND



Department of Electrical Engineering  
*Systems, Control and Mechatronics*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2017

# SIMULATION OF AUTOMOBILE CHASSIS MOTION USING A GOUGH-STEWART PLATFORM

Development of a Test Rig for Analysis of Adaptive Headlights Dynamics

CALLE ARKEVALL CHRISTOFFER SANDLUND

© CALLE ARKEVALL & CHRISTOFFER SANDLUND, 2017.

**Supervisor:** BALÁZS ADAM KULCSÁR, Department of Electrical Engineering, Chalmers University of Technology

**Supervisor:** VICTOR WAREBORN, the Volvo Car Corporation

**Supervisor:** MATTHIJS KLOMP, the Volvo Car Corporation

**Examiner:** BALÁZS ADAM KULCSÁR, Department of Electrical Engineering, Chalmers University of Technology

Master's Thesis 2017:SSYX04

Department of Electrical Engineering

*Systems, Control and Mechatronics*

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Cover:

The Gough-Stewart platform is visualised in the middle figure. The picture to the right depicts the headlight of an automobile, and the leftmost picture depicts a simulated driving scenario for testing of the headlights' illumination capacity. (The picture belongs to the Volvo Car Corporation.)

Typeset in L<sup>A</sup>T<sub>E</sub>X

Printed by Chalmers University of Technology

Gothenburg, Sweden 2017

# SIMULATION OF AUTOMOBILE CHASSIS MOTION USING A GOUGH-STEWART PLATFORM

Development of a Test Rig for Analysis of Adaptive Headlights Dynamics

CALLE ARKEVALL

CHRISTOFFER SANDLUND

Department of Electrical Engineering

*Systems, Control and Mechatronics*

Chalmers University of Technology

## Abstract

The primary aim of this thesis project is to enable a Gough-Stewart platform (a six legged robot) to mimic automobile chassis motion in real-time. More specifically, there are two automobile headlights attached atop of the Gough-Stewart platform, and the idea is for those headlights to imitate the behaviour of the headlights utilised in real, operated automobiles. Those automobiles are either actual vehicles that are driven on outdoor roads, or automobiles that are run in a simulation program. The Gough-Stewart platform is thus firstly fed motional reference data from an automobile chassis, and then the platform mimics those motions.

In order to achieve a sufficient imitation of chassis motion in real-time, a mathematical model of the Gough-Stewart platform is developed. This model calculates the motion of the Gough-Stewart platform's actuators based on how the automobile chassis is moving in three dimensional space. Hence, the Gough-Stewart platform is able to perform motion in six degrees of freedom, but in this application only three (heave, pitch, roll) are utilised. In addition, a control system that allows the Gough-Stewart platform to perform real-time imitation of chassis motion is implemented. The mathematical model is using forward kinematics to compute the motion of the Gough-Stewart platform, and the control design consists of several independent PID regulators – one for each of the Gough-Stewart platform's linear actuators. These regulators are governed by a centralised logic unit that receives the reference data which contains information about the chassis motion. This sensor data is processed through the mathematical model to obtain actuator length set-points, which are transmitted in a steady real-time stream from the logic unit to the PID regulators. The outcome of this thesis project demonstrates that a Gough-Stewart platform is applicable for real-time imitation of chassis motion. Moreover, it is concluded that several, rather simple, PID regulators – one for each movable actuator – together with a centralised logic unit that monitors those regulators, achieves sufficient real-time imitation of automobiles' chassis motion. It is also highlighted that there are no need to model any inverse kinematics of the Gough-Stewart platform. Instead, it is sufficient to combine forward kinematics of the Gough-Stewart platform with a simple but distributed control system design to achieve adequate imitation performance.

Keywords: Gough-Stewart platform, mimic chassis motion, real-time tracking, robotic control, forward kinematics



# Acknowledgements

This thesis project would not have seen today's light without the aid of our two instructors at the Volvo Car Corporation; Victor Wareborn and Matthijs Klomp. Prior to this thesis project, Victor and Matthijs have assembled the hardware that is utilised in this thesis, as well as set up a working communication between the Gough-Stewart platform and the actuator drivers. A great many thanks are directed towards them, at their patience with our struggles and inquiries, as well as their professional and academic level of knowledge. In addition, our supervisor and examiner at Chalmers University of Technology; Balázs Adam Kulcsár, deserves recognition for his assistance with both the kinematic modelling that is performed in this thesis project, as well as his openness towards discussions. In addition, gratitude are aimed towards both the companies Vector Informatik GmbH and NTI AG LinMot for their assistance with the hardware devices that are utilised in this thesis project. Lastly, the authors want to thank the splendid IT-department at the Volvo Car Corporation for their expert knowledge of the computer systems at use at the company.

Calle Arkevall & Christoffer Sandlund, Gothenburg, June 2017



# Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Aim and Purpose . . . . .	2
1.3 Scope . . . . .	2
1.4 Related Works . . . . .	3
1.5 Research Contributions . . . . .	3
1.6 Limitations . . . . .	4
<b>2 Technical Background</b>	<b>5</b>
2.1 Gough-Stewart Platform . . . . .	5
2.1.1 Actuators of the GSP . . . . .	7
2.1.2 Headlights Mounted on the GSP . . . . .	10
2.2 Peripheral Hardware . . . . .	11
2.2.1 Power Supply . . . . .	13
2.2.2 Network Hardware Interface . . . . .	13
2.2.3 Actuator Drivers . . . . .	14
2.2.4 Emergency Stop Button . . . . .	15
2.2.5 Safety Mat . . . . .	16
2.2.6 CAN bus . . . . .	18
2.3 CANopen . . . . .	18
2.4 Software . . . . .	19
2.4.1 CANoe . . . . .	20
2.4.2 Communication Access Programming Language . . . . .	20
2.4.3 IPG CarMaker . . . . .	21
2.4.4 LinMot-Talk . . . . .	23
<b>3 Modelling and Control System Design</b>	<b>25</b>
3.1 Transformation of Platform Motion to Actuator Motion . . . . .	25
3.1.1 Initial State of the GSP . . . . .	27
3.1.2 Motion Envelope Protection Algorithm . . . . .	27
3.1.3 Generalised Vector Function . . . . .	29
3.2 Handling of Discontinuities in Motion Input . . . . .	34
3.2.1 Set-point Calculation from Reference Data . . . . .	35

3.2.1.1	Finite State Automaton . . . . .	35
3.2.2	Control of Actuator Lengths . . . . .	37
3.3	Software Implementation . . . . .	38
3.3.1	CANopen Settings . . . . .	38
3.3.2	Implementation in CANoe and CAPL . . . . .	38
3.3.3	Graphical User Interface . . . . .	39
3.3.4	Realisation of Kinematics in ®MatLab . . . . .	41
<b>4</b>	<b>Results</b>	<b>43</b>
4.1	Evaluation of the Mathematical Model . . . . .	43
4.1.1	The Inverse Kinematics Complexity . . . . .	44
4.2	Imitation of Automobile Chassis Motion . . . . .	45
4.3	Vibrations of the Attached Headlights . . . . .	51
4.4	Hazard Analysis and CE Marking of the Test Rig . . . . .	52
<b>5</b>	<b>Conclusion</b>	<b>55</b>
5.1	Disadvantages of Inverse Kinematics . . . . .	55
5.2	Remarks on the Control System . . . . .	56
5.3	Performance of the Test Rig . . . . .	58
5.4	Environmental Impact . . . . .	59
5.5	Future Improvements of the Test Rig . . . . .	59
<b>A</b>	<b>CE Marking of the Test Rig</b>	<b>I</b>
<b>B</b>	<b>Hazard Analysis</b>	<b>XI</b>



# List of Figures

2.1	The Gough-Stewart Platform at use in the test rig. . . . .	6
2.2	The linked universal joints that are used to attach the actuators to both the upper and lower platform's. . . . .	6
2.3	A close up of the linear motor on one actuator. (The picture belongs to the company LinMot.) . . . . .	7
2.4	The retaining ring that constrains the rotor from completely sliding into the stator. . . . .	8
2.5	An actuator with its corresponding cylindrical, metal extension. . . .	9
2.6	The position of the attachment points on the upper platform are indicated by the three pairs of silver screws. . . . .	10
2.7	The fastening unit that a headlight is attached to. . . . .	11
2.8	The fully assembled test rig, with appurtenant headlights. . . . .	11
2.9	A flow chart that details how the VN8912 module and the actuator drivers are communicating with one another. . . . .	12
2.10	The electrical enclosure with all its interior units. . . . .	13
2.11	The network hardware interface – the VN8912 module – with an attached DIO card. . . . .	14
2.12	The structure of an example motion command. (The picture belongs to the company LinMot). . . . .	15
2.13	The command header, and its bit-wise partitioning. (The picture belongs to the LinMot company). . . . .	15
2.14	The emergency button that is attached to the door of the electrical enclosure is shown in red. The reset button with its embedded diode is displayed as well to the right of the emergency button. . . . .	16
2.15	The safety mat that is located in front of the electrical enclosure. . .	17
2.16	A simple example of an active node, a simulated node, and a CAN network node in the CANoe environment. . . . .	20
2.17	A simple example of an event trigger with a corresponding listener, which has got an additional internal event trigger. . . . .	21
2.18	An example simulation of an automobile in IPG CarMaker. . . . .	22
2.19	The control panel in LinMot-Talk, which displays information about one actuator driver. . . . .	23
3.1	A representation of how the coordinate frame is placed in an automobile. (The figure is generated from the automobile simulation program IPG CarMaker.) . . . . .	26

3.2	A flow chart that displays how input sensor data containing automobile chassis motion is converted into motion of the GSP. . . . .	27
3.3	The coordinate axes that are defined in the GSP system. . . . .	29
3.4	A graphical display of the vectors $u_i$ , $v(t)$ , and $w_i$ that are used to calculate the length of the $i$ th actuator. In addition, the angle between two adjacent actuators, $\alpha$ , is also depicted. . . . .	30
3.5	An example displaying the distances between a base frame, $O_0$ , and one additional frame, $O_1$ . (The picture is taken from the textbook <i>Modelling and Control of Robot Manipulators</i> [34]) . . . . .	32
3.6	The FSA that is embedded within the VN8912 module. . . . .	36
3.7	The CANoe environment with visible nodes that holds the implemented logic of all devices in the test rig. . . . .	39
3.8	The GUI that is used to select between the operative modes of the GSP. . . . .	40
3.9	The ®MatLab GUI that is displaying the kinematic behaviour of the GSP. . . . .	42
4.1	A display of when the motion envelope protection algorithm saturates infeasible heaving motion references. . . . .	44
4.2	A flow chart of the GSP's control system with known inverse kinematics and an outer control-loop. . . . .	45
4.3	The imitation capability of the GSP when the reference data is generated by an intense breaking scenario. . . . .	46
4.4	Enlargement of the imitation capability of the GSP's first actuator when the reference data is generated by an intense breaking scenario. . . . .	47
4.5	The error between the actual length of the first actuator and its calculated length during a driving scenario where an automobile accelerates and then breaks heavily. . . . .	47
4.6	The tracking delay of the first actuator. This delay does mostly originate from limitations of the CAN bus. . . . .	48
4.7	The imitation capability of the GSP with an inferior PID regulator tuning. . . . .	49
4.8	The effects of an inferior tuning on the GSP's imitation capability during an easier motion scenario. . . . .	51
4.9	A depiction of the GSP's imitation capability when a very poor PID regulator tuning is utilised. . . . .	52

# List of Tables

3.1	The initial values for each degree of freedom that are utilised in the ®MatLab GUI. . . . .	42
4.1	The PID regulator tuning that allows for the best imitation perfor- mance of the GSP. . . . .	46
4.2	An inferior tuning of the GSP's PID regulators that still results in a suitable imitation of chassis motion. . . . .	50
4.3	The inferior tuning used by the GSP's PID regulators during an easier motion scenario. . . . .	51
4.4	A PID regulator tuning of the GSP's actuators that is considered as very poor. . . . .	52



# 1

## Introduction

This introduction chapter serves as preface of the thesis. In this chapter, the background – as well as the aim, purpose and scope – of the thesis project are stated. Afterwards, a brief coverage of related works is presented, which is succeeded by a section that highlights the contributions to academia that this thesis brings forth. Lastly, two necessary limitations of the thesis project are stated.

### 1.1 Background

The Volvo Car Corporation [1] (Volvo Cars) is a Swedish company with roots that are reaching all the way back to the year 1927. It is headquartered in Gothenburg, and its focus is on developing and assembling fully functional automobiles. With production in the continents Europe, Asia, and also soon-to-be North America, and sales all over the world, Volvo Cars is one of the globe's most well-known automobile manufacturers [2]. Along with manufacturing, Volvo Cars is also conducting research within future automobile concepts.

The company is partitioned into several divisions, each targeted at its own research area or field of production. At the *Exterior Lighting Department* the outwards illumination of the automobiles is in focus in the sense that both development and research of automobiles' exterior light components are undertaken. Such exterior light components are a classification of all the lighting devices on an automobile, which includes, e.g., the headlights, the braking lights, and the turn signals. The intent with this research department is to optimise the operation of automobiles' light components, e.g. to maximise the field of illumination from the headlights without causing them to glare other road users or pedestrians.

The performance of new solutions are first tested purely in virtual simulation, and, if those tests turn out successfully, later in a real test vehicle that is operated on actual trafficked roads. There exist several issues with such real-life testing. Drawbacks are, e.g., that the operation of a non-finished automobile in real-life situations poses as a danger both to the operators, but also to everyday traffic [3]. Furthermore, to continuously conduct tests with a real automobile costs resources such as time and money [4]. Moreover, every test vehicle is emitting exhaust gases and fumes that are damaging to the environment. To avoid those drawbacks, the Exterior Lighting Department wants to establish a new, simpler, method of testing that allows for real-life verification of their headlights in an in house production environment.

In order to perform experiments that enable such testing and verification, a special test rig is utilised. This test rig consists of a six-legged robot, formally known as

a hexapod [5], upon which a horizontal bar is mounted. On this bar, two Volvo automobile headlights are cinched. The idea of this test rig is that it should work in tandem with an entirely virtual simulation of an automobile. When this automobile turns, tilts, or in some other way lurches, the headlights on the test rig should, in real-time, mimic the behaviour of the headlights corresponding to the automobile in the virtual simulation program.

The chosen type of hexapod for this project is a Gough-Stewart platform (GSP) [6], which is of common use in e.g. car or flight simulations [7]. Prior to the start of this thesis project the test rig is set up to the point such that a functional communication protocol between the GSP and a computer running the virtual simulation is established. The data that is acquired from the virtual automobile simulation consists of the heaving motion performed by the automobile's headlights, as well as the headlights' pitch and roll motion. Both the heaving and the angles are measured in relation to the road. Furthermore, it is possible to feed input commands to the GSP so as to move its actuators. However, there are currently no correlation between motion of the GSP and the data from the virtual simulation, and, moreover, there is no implemented control system that allows the GSP to mimic the chassis motion of the virtual automobile.

## 1.2 Aim and Purpose

The aim of the thesis project is to design a reliable control system that causes the GSP to mimic the chassis motion of a virtual automobile in real-time. This control system should make sure that the headlights mounted on the test rig adequately mimics the behaviour of the headlights corresponding to the automobile in virtual simulation.

In order to develop such a control strategy, the motions of an automobile's headlights in the virtual simulation must be translated into a mathematical model. This model, combined with both a model of the GSP itself, as well as with constraints that makes sure operation is only performed inside a feasible region, is then used in the design of a suitable control strategy.

## 1.3 Scope

The idea of the GSP is that it should be used to dynamically simulate an automobile's headlights in real-time. Thus, the primary scope of the thesis project encompasses the design and implementation of a control system. This control system should make sure that the GSP sufficiently mimics the behaviour of the chassis corresponding to an automobile in virtual simulation. To do this, reference data from the virtual simulation should be converted into motion commands, and then used to control the GSP's actuators (its legs). As long as the input commands that should be provided from the virtual simulation are not violating any constraints of the GSP, they should be used to calculate the next motion of the actuators.

## 1.4 Related Works

There exist several previous works which focuses on the development of a model that describes the dynamical behaviour of a GSP, as well as the implementation of a control system. Those works are primarily conducted in order to enable the GSP to track a specific motion pattern, or to mimic the motion of a real movable application. Y.X. Su et al. implemented high precision tracking in six degrees of freedom of a GSP that is subject to unknown noise in their paper *Disturbance-rejection high-precision motion control of a Stewart platform* [8]. In addition, both forward and inverse kinematics are used to model the behaviour of a GSP, and to develop a trajectory planning algorithm to track straight-line motion, in the work *Experimental study of motion control and trajectory planning for a Stewart Platform robot manipulator* [9] by C.C. Nguyen et al.

However, the focal point of this thesis project is not to investigate the tracking capability of the GSP per se, but rather to examine if units attached on top of the GSP can be utilised for entirely different purposes as the GSP is tracking a motion sequence in real-time. Thus, this thesis project brings forth an analysis regarding the suitability of utilising a GSP with attached components in order to investigate if those components are being impacted by the GSP during its operation.

## 1.5 Research Contributions

The development of a stable control system for the GSP brings forth several issues. Not only must the control design be able to handle swift and abrupt changes of input data (which might originate from quick and unexpected maneuvers of the automobile in virtual simulation), but it must also take several physical constraints into consideration. Those constraints are derived from the fact that the GSP is unable to tilt its roll and pitch angles too much due to the existence of the bar which the headlights are mounted upon. If the GSP obtains a sufficiently large roll or pitch angle, the headlights, or the bar itself (depending on the actual bar positioning of the headlights), collides with the floor. Furthermore, there exists limits in how large a heaving motion that the GSP can perform. This is due to the fact that the actuators cannot be extended or contracted indefinitely. Thus, to guarantee that neither collisions nor physically impossible heaving motion occur, there must exist constraints which guarantees that the GSP is operating inside a feasible region (a motion envelope protection zone) at all times.

Along with assessing that no constraints are violated, the control system must ensure that the GSP is able to mimic chassis motion in real-time, or as close to real-time as is physically possible. This requires a lightweight implementation so as to guarantee that the control system is not invoking any visible delays when the GSP mimics chassis motion in real-time. Thus, the control system must, based on the provided input data, compute new motion commands of the GSP, and make sure that they are not violating any constraints before transmitting them to the actuators, with a low enough time complexity. Furthermore, the motion of the GSP must be sufficiently smooth so as to not cause any damage to the hardware.

The GSP is able to move with six degrees of freedom [10]. This implies that the mathematical model of the GSP itself, as well as the control system, must take the entirety of three-dimensional space into consideration, despite that the simulation program is providing reference data in only three degrees of freedom. The system model must also, along with handling the reference data, be able to address the existing constraints, which are prohibiting the GSP's motion in certain patterns. It is therefore necessary to develop a model that covers all possible motions of the GSP in order for the control system to avoid deadlocks. Such a mathematical model includes several rotation matrices and translational motions that are used to map motion in different planes into one reference frame.

Furthermore, the plant model of the test rig is unknown, and it can thus not be utilised directly in a feedback loop of the control system. Therefore, a modelling workaround is required in order to design a control system that regulates the GSP sufficiently.

In addition, the test rig might be subject to either exterior disturbances (e.g. power outages, additional weight being placed on the GSP), sensor noise (e.g. insufficient actuator measurements), or inexplicit input commands due to failures in retrieving the simulation data. It is therefore important for the control system to be both robust and able to handle stochastic interferences.

### 1.6 Limitations

The mathematical model of the GSP does not take the GSP's actual dynamics into account. This is due to the fact that the dynamics are unknown and cannot be measured or in any other way easily determined by using a pure mathematical analysis of the GSP. However, the interior dynamics can be determined by utilising system identification. A design of experiments that allow for such an approach is to invoke several small vibrations in the GSP, and then measure the GSP's response from those vibrations. With such data, it is then possible to deduce a model of the GSP's dynamics. However, such a system identification technique is not used in this project.

In addition, inverse kinematics are not used in the design of the GSP's control system. Instead, a forward kinematics model is developed which allows for motion control of the GSP without utilising its actual dynamics in the design of the control system.



# 2

## Technical Background

Here, in the technical background chapter, an explanation is first given to what a GSP is, the way it functions, as well as how headlights are mounted upon it. This explanation includes an exposition of what kind of actuators that are utilised, and how they are able to move. Then, a thorough description of what type of hardware and software that is present in the test rig is provided. More specifically, it is explained how the entire test rig is fed power from an electrical enclosure, along with how a network hardware interface is used as a central logic system that, e.g., handles the data communication between devices of the test rig. There is also a description of how several hardware components are implemented as emergency measures in order to stop the motion of the GSP should it be necessary. Moreover, an overview of the communication interface that is utilised in order for all the hardware units to exchange data with one another is provided. Lastly, the software that is utilised to realise operation of the test rig is detailed.

### 2.1 Gough-Stewart Platform

The Gough-Stewart platform (GSP) consists of two platforms; one spatially fixed lower platform, and one movable upper platform. The platforms are connected to one another by six actuators. Those actuators are both extendable and contractible in length. The GSP is depicted in Figure 2.1. At both ends, the actuators are attached to the upper and lower platforms by two connected, universal joints. Such an attachment is illustrated in Figure 2.2. This allows for an almost free motion of the upper platform in three dimensional space. The 'almost' comes from the fact that the brackets of the connected universal joints implies physical constraints on the rotatability of the actuators. In other words, there is an upper limit on the maximal angle between a given actuator and the normal that is orthogonal to the floor. This upper limit has a magnitude of  $\frac{\pi}{4}$  radians. Thus, the upper platform is constrained from performing a too large surging or swaying motion (motion about the  $x$ - and  $y$ -axes, respectively). Note that surging and swaying motion refers to the upper platform moving in a plane that is parallel with the floor. However, this limitation is far larger than the motion that is feasible to expect during normal operation of the GSP. In addition to surging, swaying, and heaving motion (motion about the  $z$ -axis), the upper platform can also perform rotational motion in the angular domains roll ( $\theta$ ), pitch ( $\varphi$ ), and yaw ( $\psi$ ), which are defined as standard Euler angles [11].



**Figure 2.1:** The Gough-Stewart Platform at use in the test rig.



**Figure 2.2:** The linked universal joints that are used to attach the actuators to both the upper and lower platform's.

In its initial resting position, the centres of the lower and upper platforms' share the same  $x$ - and  $y$ -coordinate, implying that the upper platform resides completely within a plane that is parallel with the floor. Thus, the platforms are only separated

by a heaving motion, and this offset is caused by the length of the six actuators. As long as all the actuators are changing with the same length during the same interval of time, the GSP will only perform a heaving motion of the upper platform. In order to achieve motion in any other dimension, or if an angular motion is to be performed, (some of) the actuators must be altered differently from one another. An example of motion that requires different lengths of the actuators is a surging or swaying motion. Then, the actuators that are attached the most into the direction of motion must be shorter than the actuators that are located farther away. It is explained in more detail in Section 3.1 how the lengths of the actuators affects the position of the upper platform.

### 2.1.1 Actuators of the GSP

Each actuator consists of a linear motor [12]. A linear motor is an electric motor where the two components *rotor* and *stator* are producing a linear force along the length of the stator, instead of a rotational torque which is the case in normal electric motors [13]. This is accomplished by a design where the rotor is wrapped around the stator. The stator is in turn able to slide back and forth through a hole in the rotor. The linear motors are manufactured by the company LinMot [14], which is a global manufacturer of high quality tubular style linear motors. The type of linear motor used for the GSP is depicted in Figure 2.3.



**Figure 2.3:** A close up of the linear motor on one actuator. (The picture belongs to the company LinMot.)

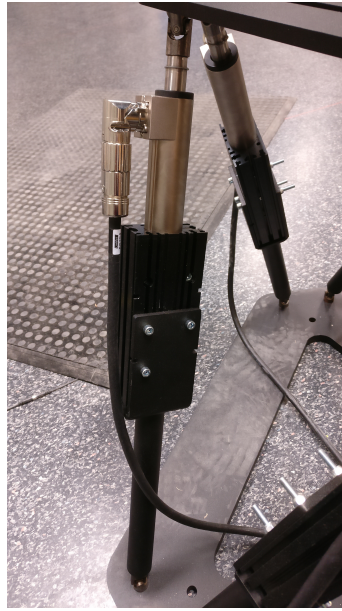
The rotor has got several permanent magnets inside of it. There is a magnetic field generated by the stator which produces a linear force along the length of the rotor, thus allowing it to slide through the stator. This makes it possible for the actuators to be both extended and contracted in length, depending on how the magnetic field is generated. In the test rig at Volco Cars, the maximum extendable length of the rotor is 480 millimetres, whereas its minimum length is -50 millimetres (this means that the tip of the rotor resides 50 millimetres inside the stator) [15]. However, there is a retaining ring attached to the upper end of the rotor, as is displayed in Figure 2.4. This retaining ring acts as a mechanical stop that disallows the rotor from contracting into the stator. Thus, the minimal length of the rotor is limited to be zero millimetres.



**Figure 2.4:** The retaining ring that constrains the rotor from completely sliding into the stator.

In order to be able to measure the current position of the rotor, the stators are equipped with sensors at an even distance from one another. Those sensors are able to measure the current position of the rotor with an accuracy of 0.1 micrometers. Each linear motor has also got an embedded PID regulator, which is used to make sure that the position of the rotor tracks a given reference value. In addition to the rotor's position, the PID regulators are also able to control with which velocity the rotor should reach its desired set-point, as well as which acceleration it should use. The logic of both the PID regulator and the sensors of a stator is implemented in a driver. There are six drivers in total – one for each linear motor. The drivers are discussed in detail in Section 2.2.3, and the implementation of the corresponding PID regulators is handled in Section 3.2.2.

Moreover, as is seen in Figure 2.5, a hollow metal cylinder is attached to the bottom end of each actuator. The metal cylinders are large enough for the rotors to fit inside of them. The purpose is to extend the total length of the actuators, so as to make the upper platform operate at a higher level. As is detailed in Section 2.1.2, this is desirable in order for the height of the headlights mounted on top of the upper platform to better match the height the headlights would have in a real automobile.

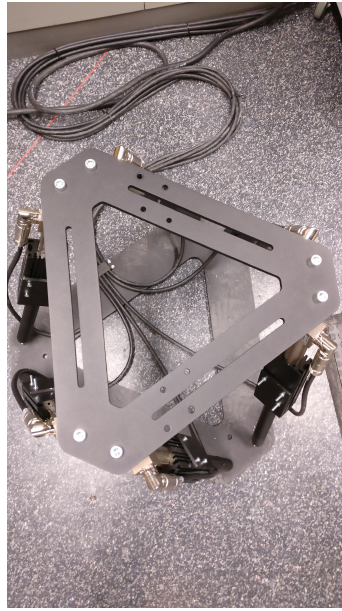


**Figure 2.5:** An actuator with its corresponding cylindrical, metal extension.

The minimum operating height of the upper platform is obtained when all the actuators are contracted as much as possible (thus, the metal cylinder is enclosed around a majority of the rotor). This places the upper platform at 675 millimetres above the floor. In the same manner, to reach maximum height, each rotor should be extended as much as possible, which raises the upper platform to a height of 910 millimetres. (The conversion between rotor extension and the height of the GSP is discussed in Section 3.1.3.) However, in order to obtain maximum output thrust of the linear motors, a much smaller region of operation is required. The motors are performing at maximum capacity when the rotors are extended between 50 and 330 millimetres [15]. Within this interval, each linear motor is able to produce a sustained force of 53 Newtons. This gives the GSP a total lifting force of 318 Newtons while operating at maximum capacity. During short intervals, a linear motor may reach a peak force of 203 Newtons, which greatly increases the GSP's ability to handle brief, unforeseen periods of additional strain. The GSP can operate at maximum capacity even when the actuators are of different lengths, as long as none of the rotor positions are outside the aforementioned interval.

The actuators are attached to both the lower and the upper platform in a circular fashion, despite the fact that both platforms have got more of a triangular appearance. A circle with origin in the middle of each platform is divided into three equally large sections, where each section contains two attachment points. The attachment points are located in each of the three 'corners' of the triangular platform. All attachment points on respective platform are sharing the same radius to that platform's centre. In Figure 2.6, the placement of the attachment points are shown on both platforms, as well as the appearance of the platforms themselves.





**Figure 2.6:** The position of the attachment points on the upper platform are indicated by the three pairs of silver screws.

### 2.1.2 Headlights Mounted on the GSP

When an automobile is driven on a real road its chassis is subject to a multitude of (often minor) shakes and trembles. Those disturbances usually arise from small cavities in the road, as well as from when the automobile is driven over debris of the less significant magnitude. In addition, a bump in the road will cause an even larger impact on the automobile's chassis motion. When the chassis of an automobile shakes all devices that are attached to it are affected, of which one are the headlights.

In order to mimic the behaviour of the headlights from an automobile in virtual simulation, real headlights are attached to the upper platform of the GSP. This is done by first attaching each headlight to an individual fastening unit, as is depicted in Figure 2.7. Then, each fastening unit is cinched to an aluminum bar that is horizontally attached to the GSP. This constitutes the complete test rig, and it is displayed in Figure 2.8. The distance between the headlights on the GSP is the same as the mutual distance the headlights have in a real automobile. Thus, by moving the upper platform of the GSP to the correct height it is possible for the headlights on the platform to be placed spatially the same as when they are used in a real automobile. By performing such a maneuver, the GSP can be used for an adaptive dynamics simulation of the headlights. By changing the mutual distance between the headlights on the horizontal bar, as well as the operating height of the upper platform, the test rig can mimic the behaviour of headlights from different automobiles. Thus, the test rig is adaptable in such a way that it can simulate several different automobiles' headlights. In its current state it is possible to use the GSP to imitate the headlights of all of the (private) automobiles that Volvo Cars is manufacturing.



**Figure 2.7:** The fastening unit that a headlight is attached to.

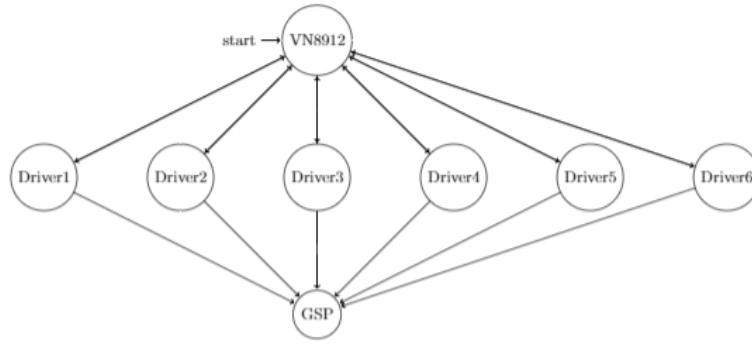


**Figure 2.8:** The fully assembled test rig, with appurtenant headlights.

The total weight of the entire fastening unit and the horizontal bar is approximately 8 kilograms, and the weight of the headlights is dependent on what type of automobile headlights that are used in the dynamic simulation. In general, though, a headlight that is used in one of Volvo Cars automobiles weigh around 5 kilograms. The combined sustained force of 318 Newtons that is produced by the six linear motors is thus able to lift and maneuver the upper platform together with all of its attachments.

## 2.2 Peripheral Hardware

Along with the GSP itself, there are several other hardware components that constitutes the test rig. As mentioned in Section 2.1.1, each actuator has got its corresponding driver. Those drivers are located in an electrical enclosure. The network hardware interface is also situated within this enclosure. A flow chart that details the communication between the VN8912 module and the actuator drivers is displayed in Figure 2.9.



**Figure 2.9:** A flow chart that details how the VN8912 module and the actuator drivers are communicating with one another.

Along with those, the enclosure is filled with several other devices, which are mainly used for handling the power supply of the test rig. Furthermore, there are some additional hardware units, such as two different emergency stop buttons, and a light that indicates if an emergency stop is active. This light also works as a reset button for the emergency stop. In addition, there exists a hardware bus that is used to handle the communication between the network hardware interface and the actuator drivers. Moreover, outside of the enclosure, there is also an ordinary desktop computer running an automobile simulation program, which generates references of chassis motion that the GSP should mimic. Those references must first be processed through a mathematical model in order to be comprehensible by the actuator drivers. The simulation software is detailed in Section 2.4.3, and the mathematical model is explained in Section 3.1.3. Figure 2.10 displays the electrical enclosure in all its completeness.





**Figure 2.10:** The electrical enclosure with all its interior units.

### 2.2.1 Power Supply

The electrical enclosure is powered by a three-phase cable connection, which is able to provide a maximum of 230 Volt. The three-phase cable is directly connected to the power supply of the enclosure. From this power supply, the input power is distributed to all other devices via cabling. In order for all hardware components to operate fully functional a voltage of 72.5 Volt is required. The three-phase connection thus provides the test rig with a sufficient level of power.

### 2.2.2 Network Hardware Interface

The network hardware interface is manufactured by a company called Vector that specialises on the development of automobile electronics [16]. More precisely, the hardware interface is a VN8912 module, which is a device that is able to handle data from different parallel bus channels in real-time [17]. Some of the supported communication protocols of the VN8912 module are CAN [18], LIN [19], and FlexRay [20]. However, in the test rig, the VN8912 module is utilised for CAN communication only. The purpose of the VN8912 module is to handle communication to and from the actuator drivers, which is of CANopen type. For a review of the CANopen protocol, and a description of how it is used in the test rig, see Section 2.3 and Section 3.3.1, respectively. There is also an additional plug-in unit connected to the VN8912 module; a DIO card. This DIO card allows the VN8912 module to acquire and generate digital signals. The VN8912 module with the attached DIO card is

displayed in Figure 2.11.



**Figure 2.11:** The network hardware interface – the VN8912 module – with an attached DIO card.

The entire logic of the test rig is implemented in the VN8912 module. This hardware interface is also fed input data from either an automobile simulation software, which is explained in Section 2.4.3, a log file containing collected driving data, or from commands in a GUI, which is outlined in Section 3.3.3. This input data must be processed in order for the GSP to move accordingly. The processing is done by applying the raw input data to a mathematical model, which then converts it to motion commands of the GSP. This model is detailed in Section 3.1. The purpose of the VN8912 module is to both handle the communication to and from the actuator drivers, and also to act as a central logic system that calculates set-points for the PID regulators of each actuator. As of such, the test rig is using a central logic system that is governing a distributed control system, which consists of the actuators' PID regulators. The derivation and implementation of the central logic system is presented in Section 3.2.1, and the distributed control system is outlined in Section 3.2.2.

The VN8912 module is a powerful interface that is running on a third generation (Ivy Bridge) Intel Core i7-3517U processor [17]. In order to simulate data traffic in real-time the VN8912 module utilises an 'extended real-time' component along with an embedded software called CANoe, which is addressed in Section 2.4.1. The VN8912 module also comes with an embedded programming language; the Communication Access Programming Language (CAPL), which is briefly introduced in Section 2.4.2. CAPL is used to implement the logic that is embedded in the VN8912 module.

### 2.2.3 Actuator Drivers

The drivers at use in the test rig are manufactured by the company LinMot. Each driver contains logic that allows it to receive and transmit data to and from the VN8912 module. Additionally, the logic of the PID regulator, which allows the connected linear motor to move its corresponding rotor, is also embedded in the driver. In order to move the rotor to a specified set-point, the PID regulator in the driver requires input commands from an exterior source. The VN8912 module (see Section 2.2.2) is used for this purpose. Such input commands informs the driver how the rotor should be adjusted. To perform motion of a rotor, its driver requires both a specification of what type of motion that is desired, and also the set-point value

that the rotor should move to. There exists several different commands that the drivers and the linear motors can handle, e.g. 'set target position', 'set target position with velocity', and 'set target position with velocity and acceleration'. The first command moves the rotor as fast as possible to the desired set-point, whereas the second command moves the rotor to the set-point with a specific velocity. The last command also takes a specified acceleration, in addition to position and velocity, into account when altering the length of the rotor.

The motion commands that are required by the drivers are represented by one word that contains the command ID, which is a header that informs the driver of which command to execute. In addition, along with the header, there can be up to 16 command parameter words, which are used to, e.g., specify a certain position, or velocity. In Figure 2.12, the structure of an arbitrary motion command is displayed.

Word	Description	Example of command
1.	Command Header with ID	Go To Absolute Position Immediate
2.	1. Command Parameter	Position
3.	2. Command Parameter	Maximal Speed
4.	3. Command Parameter	Acceleration
5.	4. Command Parameter	Deceleration
6.-16.	5.-15. Command Parameter	Not used

**Figure 2.12:** The structure of an example motion command. (The picture belongs to the company LinMot).

Furthermore, the command ID is itself divided into three parts; the master ID, the sub ID, and the command counter. Figure 2.13 displays the partitioning of the command ID. The master ID specifies which group of commands the motion command belongs to (e.g. motion with respect to position and velocity), whereas the purpose of the sub ID is to identify different commands within the same command group. The command counter is used as a checksum, and it makes sure that the motion command a driver receives is one that should be executed. The checksum works in such a way that a new motion command is only executed if the command counter has changed. Thus, since all the desired motion commands must change the command counter, the drivers are prohibited from performing any unwanted motion of the actuators. Such undesired motion can arise from, e.g., old motion commands that are still being relayed back and forth on the CAN bus.

Master ID								Sub ID				Command Count			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

**Figure 2.13:** The command header, and its bit-wise partitioning. (The picture belongs to the LinMot company).

## 2.2.4 Emergency Stop Button

Mounted on the door of the electrical enclosure is an emergency button, as is seen in Figure 2.14. This emergency button works such that it, once pressed, freezes the GSP in its current motion. At the right side of the emergency button there is also a reset button, with a built-in diode. This diode is lit when the emergency button

has been pressed, which indicates that the GSP is in an emergency state. In the emergency state the GSP cannot perform any motion at all, but merely hold its current position. As of such, power is still being supplied to the linear motors, but they are not able to alter the position of their respective rotor. The emergency state is modelled as a node in a finite state automation, which is further described in Section 3.2.1.1. The GSP is also limited from performing any motion as long as the emergency diode is lit. This is realised by constraining the VN8912 module from sending any motion commands to the drivers when the emergency state is active.



**Figure 2.14:** The emergency button that is attached to the door of the electrical enclosure is shown in red. The reset button with its embedded diode is displayed as well to the right of the emergency button.

In order for the GSP to leave the emergency state, the emergency diode must be turned off. For safety reasons the emergency diode requires a two-step process in order to be unlit. This two-step process is executed by first releasing the emergency button, and then pressing the emergency diode. This is done by either using the hardware buttons on the door of the electrical enclosure, or by pressing virtual buttons in a GUI, which is further detailed in Section 3.3.3.

Both the emergency button and its corresponding reset button are directly wired to the VN8912 module via an electrical circuit. This is to ensure that the emergency button is able to freeze the motion of the GSP with as little delay as possible.

### 2.2.5 Safety Mat

Along with the emergency button on the electrical enclosure, there is also a safety mat which is lying on the floor in front of the GSP. The safety mat works in the same manner as the emergency button; if somebody walks or steps on the safety mat, i.e. if somebody 'presses' the safety mat button, the GSP freezes its current motion. The safety mat is displayed in Figure 2.15, and the purpose of it is to create a region of space in front of the GSP where no person can stand during operation. Thus, if someone walks too close to the GSP, then it is not possible for it to perform any motion. This set up minimises the risk of someone being hit by the GSP during its operation.



**Figure 2.15:** The safety mat that is located in front of the electrical enclosure.

As with the emergency button described in Section 2.2.4, the safety mat is directly connected to the VN8912 module with an electrical circuit. This allows for as quick a freeze of the GSP as possible whenever the safety mat is walked upon.

### 2.2.6 CAN bus

The communication between the VN8912 module and the drivers is handled by the CANopen protocol, which is described in Section 2.3. The CANopen protocol requires a physical bus link that is able to handle CAN communication in order to transmit data. For this purpose a CAN bus is utilised. The CAN bus is able to transmit data packages – which are also known as frames – that contains an 11-bit ID, a bit that is used to send a remote transmission request (RTR), and zero to eight bytes of actual data. The RTR bit is used as a remote frame by a transmitting device to indicate to a receiver that the transmitter requests specific data. The receiver then responds with the desired information. [21]

In order to handle large up-times of data streams, the CAN bus must be terminated at both ends. This reduces the risk that signals, which have already reached their intended recipient, are being reflected back onto the CAN bus. If signals are reflected, they might interfere with the current data stream. [21]

Moreover, the CAN bus can handle data transmissions at different baud rates. The baud rate is a unit of modulation rate – how many symbols (or bits) that can be transmitted on the bus every second [22]. The standard baud rate for the CANopen protocol usually ranges from 10 kBit/s to 1 MBit/s [23].

## 2.3 CANopen

As is stated in Section 2.2.6, the CANopen protocol is used for the communication between the VN8912 module and the actuator drivers. CANopen is a protocol that uses the upper layers of the CAN protocol [24], which itself is defined to follow the OSI network model [25]. The OSI layers that are inherited by CANopen are the network layer, the transport layer, the session layer, the presentation layer, and the application layer.

One of the benefits with CANopen is that it enables data transmission between two or several hardware units among an even larger spectrum of connected devices [24]. Moreover, CANopen ensures a high quality and reliability of the data stream in such a way that a transmitted signal is guarded against exterior changes. This reliability originates from the fact that the CANopen protocol utilises the transport layer of the OSI network model, which allows for secure node-to-node communication. On an additional note, CANopen (and CAN in general) is often used in automobiles, as well as other vehicles, where the quality of the communication between different units is crucial in order to ensure a secure and operable device. [21]

All devices that are communicating with the CANopen protocol are connected in series, which implies that if one unit transmits a signal, then all other devices acts as receivers. The pre-defined ID, which is discussed in Section 2.2.6, is known as a COB-ID, and it is utilised by the receiver of a frame to deduce if a transmitted message is targeted towards that receiver. A receiver will only listen to a message if it contains the correct COB-ID, and ignore the message otherwise. The contents of the COB-ID is specified by the transmitting device. In order for a transmitter to target several receivers with the same message, the device must transmit the message once for every receiver. For each new transmission the COB-ID must then



be changed so that the message is directed to a receiver that has not yet been reached by the message. [24]

Furthermore, CANopen is a priority based communications protocol [21]. This implies that each data message that is transmitted on the CAN bus is sent with a corresponding priority, which tells all connected devices how urgent or important each message is. This priority consists of the first four bits of the COB-ID. A lower COB-ID implies that the message has got a higher priority. The highest possible priority for a message is obtained with a COB-ID of zero. If two devices are transmitting data on the CAN bus simultaneously, the message with the highest priority will get precedence. The lower priority message is thus postponed until its priority has become the highest, which occurs when there are no other messages with a lower COB-ID in queue to be transmitted on the CAN bus. This priority assessment of each message before transmission allows for the highest prioritised data signal to be transmitted without interruption. Thus, highly important data frames (such as time-critical safety messages) are equipped with as low a COB-ID as is possible.

Furthermore, CANopen has got an embedded protocol known as the Process Data Object (PDO) protocol, which is used to handle real-time data transmission between devices [26]. A PDO consists of one CAN frame. Thus, with one PDO, it is possible to communicate up to eight bytes of data to a receiver. There are two different kinds of PDOs that are utilised in a node-to-node communication network. Those are transmit PDO (TPDO) and receive PDO (RPDO). The TPDO is used by a node that transmits a frame, and the RPDO is used by the receiving node.

In a node-to-node communication with the CANopen protocol, the transmitter of a PDO and the intended receiver must agree upon the correct mapping of all PDO frames. If all devices are not using the same mapping, the receiver of a frame might not be able to decode what data it contains. This mapping is specified beforehand by every device in something that is called the Object Dictionary (OD) [27]. Each device has got a separate OD, which is used to configure the structure of the CANopen communication stream on a device specific level. The OD is a logical addressing scheme – a table – for accessing communication and application parameters, as well as functions and data, via the CAN bus.

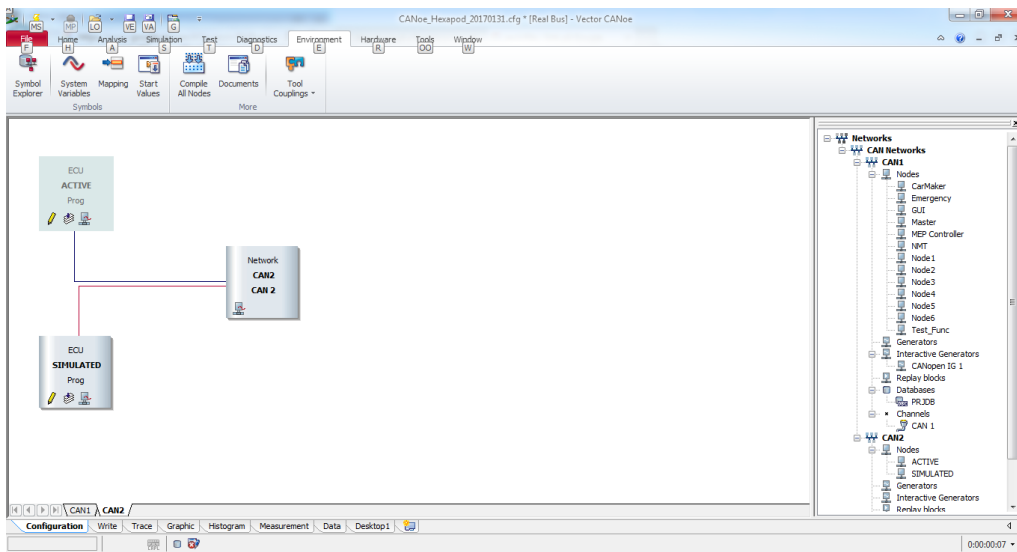
## 2.4 Software

In this Section the software environment that is embedded in the VN8912 module is first described. Then, the inner workings of this software environment's associated programming language is covered. In addition, the automobile simulation software that is used to generate reference values containing chassis motion is presented in detail as well. Lastly, a program that is able to analyse the current status and settings of the actuator drivers is briefly outlined.

### 2.4.1 CANoe

Aside from manufacturing the VN8912 module, Vector also distributes a software called CANoe [28]. This software is primarily designed for analysing, testing, and developing large networks with one or several Electronic Control Units (ECUs). As of such, CANoe is used as a backbone for the software implementation of the test rig. This is further described in Section 3.3.2.

The CANoe environment consists of several nodes, which correspond to real hardware components. Those nodes can be either active or simulated. When a node is active, the software that is implemented in that particular node is executed on a connected hardware component that corresponds to that node. This is the case for the VN8912 module and the actuator drivers; their nodes are set to active. A simulated node, on the other hand, does not require any physical hardware. The CANoe software itself will then execute the node logic on the computer that is running the CANoe program. In addition to functioning as the link between the CAN bus communication and the simulation environment, CANoe has also got several tools which allows for analyses of the CAN protocol transmissions. A CANoe environment with both an active and a simulated node, along with a CAN network node, is displayed in Figure 2.16.



**Figure 2.16:** A simple example of an active node, a simulated node, and a CAN network node in the CANoe environment.

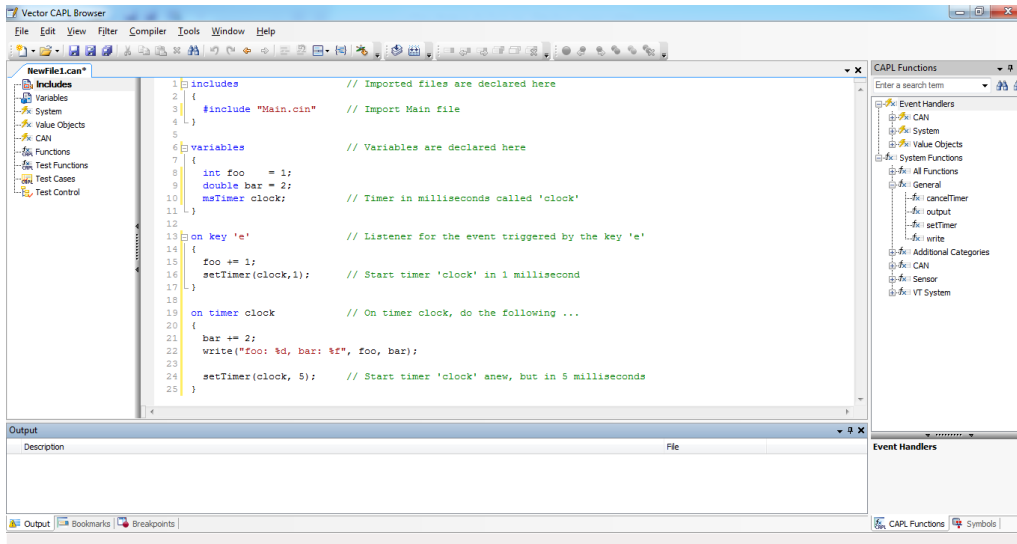
### 2.4.2 Communication Access Programming Language

The Communication Access Programming Language (CAPL), which is the built in programming language of CANoe, is a C like programming language [29]. However, one feature that distinguishes CAPL from the C programming language is that CAPL is event-based. This implies that a CAPL program will use listeners to detect whether a specific event has occurred. When an event takes place a corresponding event trigger is activated, and the program executes the software that is listening on that particular event. CANoe also comes with an embedded CAPL browser, which



is a compiler that allows for the CAPL code to be realised. In other words, the CAPL browser is used for implementing logic in the CANoe nodes.

In Figure 2.17, a short CAPL example program is displayed. There is a visible listener – the `on key 'e'` clause – which executes its logic once the key `e` is pressed on the keyboard. This clause triggers the timer `clock`, which has got its own internal listener; the `on timer clock` clause.



**Figure 2.17:** A simple example of an event trigger with a corresponding listener, which has got an additional internal event trigger.

### 2.4.3 IPG CarMaker

The reference data of an automobile's chassis motion that the GSP should mimic originates from a desktop computer running a program called IPG CarMaker. The IPG CarMaker software is a simulation platform for virtual driving of automobiles [30], and it is often used for testing purposes by large automobile manufacturers [31].

In IPG CarMaker, the operator has a plethora of possible driving settings to choose from. When the desired settings are selected, a simulation of an automobile appears in a graphical environment so as to visualise how the automobile behaves while driving. Possible settings range from selecting automobile specific parameters, such as the size, weight, and shape of the automobile, to how the road the automobile is driven upon appears, as well as what motion pattern the automobile should be driven in. More advanced driving scenarios, such as abrupt breaking or acceleration, jumping, or performing tight turns, are also possible to simulate in IPG CarMaker. An example simulation of an automobile in IPG CarMaker is depicted in Figure 2.18, where it is seen that the simulated vehicle is driven along a straight road.



**Figure 2.18:** An example simulation of an automobile in IPG CarMaker.

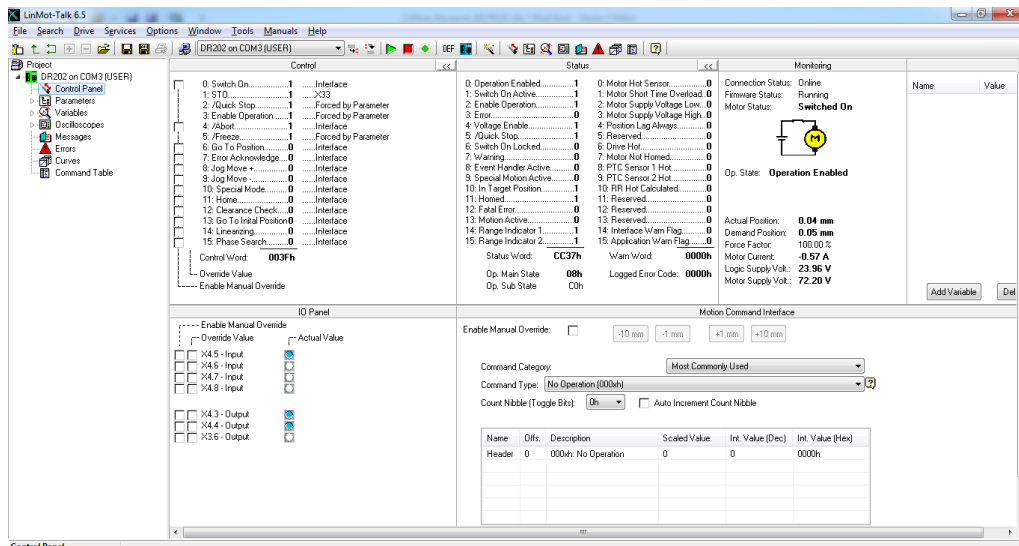
Furthermore, it is possible to place sensors anywhere in the automobile. This allows for extraction of motion data of specific parts or components of the automobile. A sensor placed on the automobile's chassis, and more precisely, in the grill, lies on the approximate same distance from both the roadway and the headlights as the centre of the upper platform of the GSP does. Thus, by utilising such a sensor, it is possible to extract references containing chassis motion that are consistent with real driving scenarios.

In the test rig, IPG CarMaker is used to provide the VN8912 module with reference data. This data contains an automobile's chassis motion, and it is either in the form of a real-time stream of reference values from an ongoing driving simulation, or it is provided to the VN8912 module as a log file that contains data points and time stamps of a previous driving session. This previous driving session data can either come from an earlier IPG CarMaker simulation, or it can originate from a chassis sensor in a real automobile that has been used to collect the driving data. In the case of a real-time IPC CarMaker simulation, the VN8912 module sends set-points to the actuator drivers continuously as the references are provided to the VN8912 module from IPG CarMaker. When a log file is used to provide reference data, the VN8912 module reads that log file in the same manner as it is interpreting real-time data from IPG CarMaker.

In order for the VN8912 module to send the correct set-points to the actuator drivers, the mathematical model derived in Section 3.1 is first applied to the reference data from IPG CarMaker. This is done in order to convert the data into useful set-points that the actuators' PID regulators are able to track. IPG CarMaker is able to send reference data to the VN8912 module every millisecond. As is stated in Section 2.2.6, the VN8912 module sends set-points to the actuator drivers every fifth millisecond. Therefore, the VN8912 module has approximately five milliseconds to process the data from IPG CarMaker before the set-points have to be transmitted to the drivers. How the VN8912 module is calculating those set-points is described in detail in Section 3.2.1.

### 2.4.4 LinMot-Talk

A software called LinMot-Talk [32], which is distributed by the company LinMot, is used during early stage development of the mathematical model described in Section 3.1, as well as during the design of the control system, which is addressed in Section 3.2. The LinMot-Talk software allows for tuning of the PID regulator parameters – the proportional, integral, and derivative gains. With LinMot-Talk it is also possible to directly connect a desktop computer to a specific actuator driver, so as to obtain information regarding, e.g., the current state of the driver, what motion command it has most recently handled, and how the CANopen message headers look. Moreover, driver specific changes regarding the CANopen communication, e.g. altering the baud rate at which the driver transmits and receives information, are adjustable as well. LinMot-Talk has in addition got an embedded control panel that is monitoring the current status of the connected driver. This control panel is shown in Figure 2.19.



**Figure 2.19:** The control panel in LinMot-Talk, which displays information about one actuator driver.



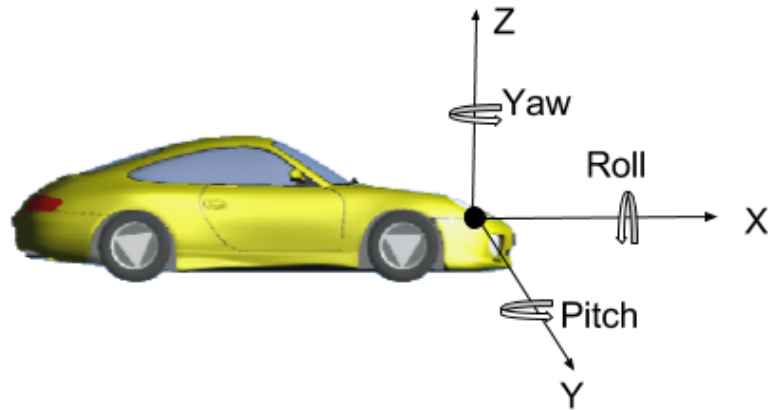
# 3

## Modelling and Control System Design

This chapter describes the method of work that is conducted in order to enable the GSP to mimic chassis motion in real-time. The exposition of this chapter contains a thorough coverage of the mathematical model of the GSP, as well as a description of how the control system of the test rig is implemented. Firstly, it is detailed how reference data that contains chassis motion is processed to obtain suitable set-points for the control system. It is also stated what measures that are undertaken in order to guarantee that the motion of the GSP is feasible. Afterwards, the design of the control system is outlined. Lastly, it is addressed how both the mathematical model and the control system are implemented in the hardware of the test rig.

### 3.1 Transformation of Platform Motion to Actuator Motion

In order to control the motion of the GSP – and thus enable it to mimic chassis motion in real-time – a mathematical model is required. The purpose of this model is to convert input data consisting of surging, swaying, and heaving motion, as well as motion in the angular domains roll, pitch, and yaw, into motion of the upper platform. As is addressed in Section 2.4.3, those input signals comes from motion of the chassis of an automobile that is driven in a virtual simulation program. Figure 3.1 depicts a coordinate system where all inputs are defined in an automobile. The surging, swaying, and heaving motions are originating from a centre point between the automobile's headlights. This centre point holds the location of a sensor that measures the automobile's chassis motion. Thus, when the VN8912 module receives reference data, it continuously calculates the required length of each actuator.

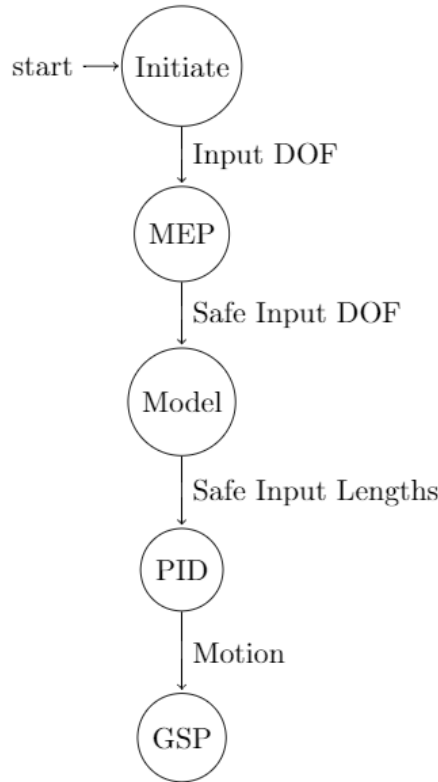


**Figure 3.1:** A representation of how the coordinate frame is placed in an automobile. (The figure is generated from the automobile simulation program IPG CarMaker.)

In the model, the lower platform is held fix, and the position of the upper platform is altered relative to the lower platform. Thus, the length of each actuator is only dependent on the actual position of the upper platform. When new reference data, consisting of spatial and angular motion of an automobile's chassis, is received by the VN8912 module, a kinematic model is utilised in order to calculate how the actuators changes in length. This kinematic model is explained in detail in Section 3.1.3.

Hence, the mathematical model utilises forward kinematics in order to calculate the length of each actuator. Forward kinematics is a common modelling technique for the calculation of a robot's end-effector, and it constitutes a convenient way of determining how a movable robotic system with multiple parts behaves [33].

In Figure 3.2, a flow chart is depicted which displays how input references containing automobile chassis motion are converted into actual motion of the GSP. In the figure, *DOF* is an abbreviation for degree of freedom, and it refers to the reference data of chassis motion in a specific degree of freedom. *MEP* corresponds to a motion envelope protection algorithm that is utilised to make sure that every motion that the GSP performs is feasible. Section 3.1.2 describes the workings of this algorithm in detail. Operation of the GSP is performed such that when reference data of chassis motion is provided to the VN8912 module, the data is first processed by the motion envelope protection algorithm. This algorithm saturates motion commands containing set-points that are physically out of reach for the GSP. Thus, the motion envelope protection algorithm serves as a means to guarantee that the GSP is not performing motion which would otherwise break it. Afterwards, a mathematical model that utilises forward kinematics to calculate the position of the upper platform, is applied in order to obtain set-points for the linear motors' PID regulators. Those set-points are transmitted to the actuator drivers, and the GSP is imitating the chassis motion. The following sections describes each step of the flow chart in more detail.



**Figure 3.2:** A flow chart that displays how input sensor data containing automobile chassis motion is converted into motion of the GSP.

### 3.1.1 Initial State of the GSP

An initial position of the upper platform is defined in order to both ease the modelling and the operation of the GSP. In this initial position, the centre of the upper platform is at rest with an offset from the centre of the lower platform. Both platforms' centres have performed zero surging or swaying motion, as well as zero roll, pitch, or yaw revolution. Thus, in the initial position, the upper platform is only separated from the lower platform by a coordinate bias vector that consists of a heaving motion of the upper platform. Note that this coordinate bias vector is expressed in units of millimetres. Furthermore, the angular revolutions are expressed in units of radians.

### 3.1.2 Motion Envelope Protection Algorithm

Several measures are undertaken in order to guarantee that motion of the GSP is physically feasible. Those measures are implemented as an algorithm that consists of three different assessments. Firstly, the upper platform is being constrained from moving in such a way that the GSP breaks. This is achieved through a saturation of new reference data in any of the six degrees of freedom that is received by the VN8912 module. This saturation implies that each reference value resides within a

pre-determined interval. The purpose of this interval is to make sure that the desired motion specified by the reference values is small enough to constrain the actuators from being extended more than what is physically possible. If the reference values are outside of the interval, they are saturated to the (closest) limit of the interval to obtain feasible set-points. As of such, too big motion in any domain is prohibited, and the GSP is only movable within a feasible region of operation. Furthermore, one intent with the saturation is also to allow for the upper platform to reach its set-point before new reference data is provided to the VN8912 module. This makes sure that the GSP always has time to reach its set-point before the next motion command is received.

As a second step of the algorithm, the reference data is applied in the generalised vector function discussed in Section 3.1.3, after it has been initially saturated. With this function, the length of each actuator that is required to obtain the desired position of the upper platform is calculated. Then, there exists an analysis that makes sure that those required lengths are neither shorter nor longer than the minimum and maximum possible, physical, lengths of an actuator. If they are, the required motion of the upper platform is deemed as infeasible and no motion commands are sent to the actuator drivers. In that case, the upper platform will hold its current position instead of updating it in accordance with the reference values. Hence, if any of the actuator lengths corresponding to a specific set-point are physically impossible to obtain, the GSP holds its current position and awaits new, feasible, reference data.

The last step of the motion envelope protection algorithm makes sure that the actuators of the GSP does not collide with one another during operation. This is achieved by first calculating the possible motion in each degree of freedom that the upper platform can perform, based on its current position. Then, the interval of motion for each degree of freedom is normalised as in Interval (3.1).

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \quad (3.1)$$

Here, -1 corresponds to the maximum negative motion along the  $x$ -,  $y$ -, or  $z$ -axes, or the maximum negative revolution of the Euler angles (depending on what degree of freedom that is normalised). Likewise, 1 correlate to the maximum positive motion or revolution. Furthermore, a zero within this interval represents no motion of the zero-valued degree of freedom. Thus, motion in all six degrees of freedom lies within an equally sized interval. To further simplify calculations, the absolute value of Interval (3.1) is used to generate the new, normalised Interval (3.2).

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \quad (3.2)$$

Possible motion of the upper platform is calculated as a percentage of the maximum movable capability of the upper platform. The movable capability is calculated as one minus the normalised interval of all degrees of freedom (the normalised interval that is being utilised is the absolute valued one from Equation (3.2)). An expression for the movable capability is depicted in Equation (3.3), where  $M_c$  denotes the movable capability, and the normalised degrees of freedom are characterised by the subscript *norm*. If a set-point requires the upper platform to move with a normalised motion that is larger than the movable capability (e.g. a surging motion



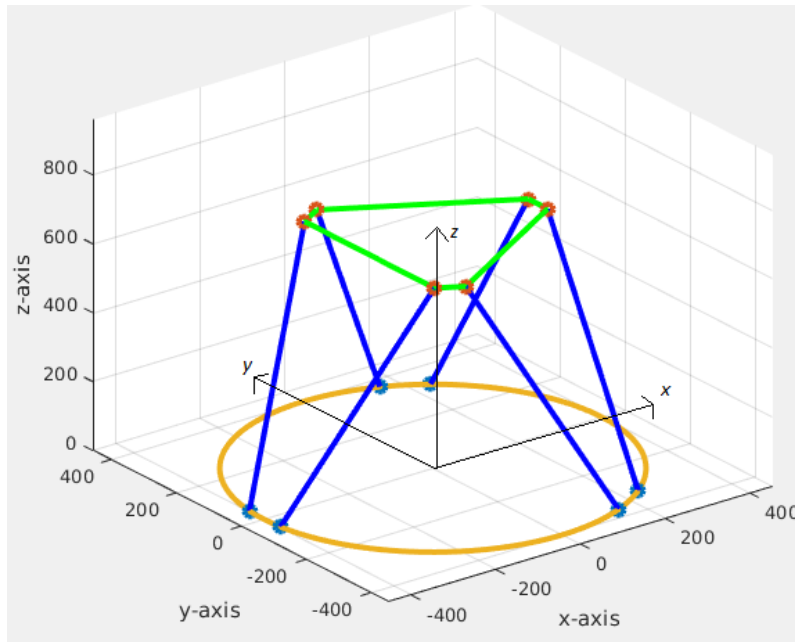
where  $x_{norm} > M_c$ ), then that motion is deemed as infeasible. A motion declared as infeasible is not performed by the GSP.

$$M_c = 1 - (x_{norm} + y_{norm} + z_{norm} + \theta_{norm} + \varphi_{norm} + \psi_{norm}) \quad (3.3)$$

This stage of the algorithm states that minimum motion of the GSP has occurred when the upper platform resides in its initial position. No deviation from the initial position implies that the movable capability is exactly one. Similarly, the maximum feasible motion of the upper platform can be obtained by moving in a plethora of different motion patterns; the only requirement is that the movable capability subtracts down to zero. Thus, for an arbitrary – feasible – motion in any degree of freedom, the algorithm bases the possibility to perform that particular motion on the current position of the upper platform, e.g. on its previous motions.

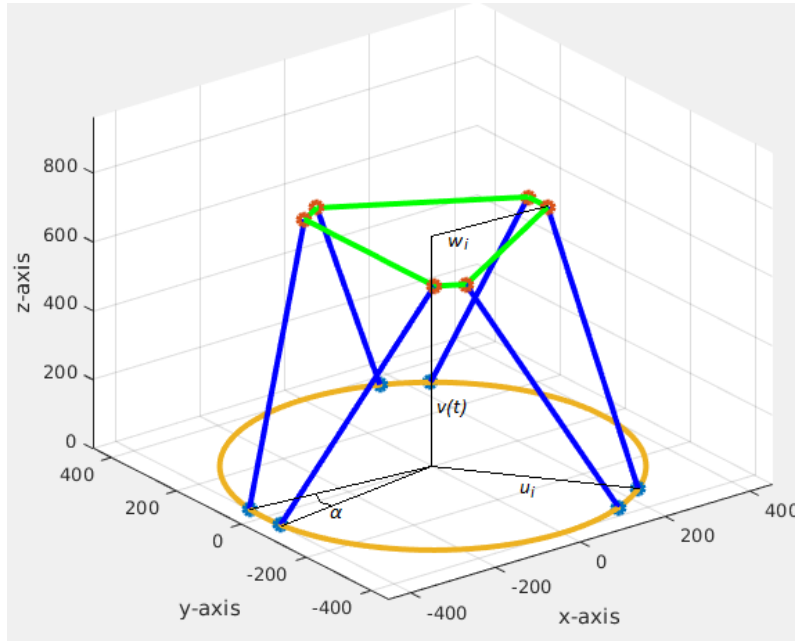
### 3.1.3 Generalised Vector Function

The lengths of all actuators are calculated by a forward kinematic function. This function is a generalised vector function since it is possible to obtain the lengths of all actuators by utilising this function. What differs between the six actuator calculations, though, are the actuators' attachment points on both the lower and the upper platform. An actuator is attached to each platform with two connected, universal joints, as is explained in Section 2.1.1. In order to obtain spatial expressions for the attachment points on both the lower and upper platforms, coordinate axes of the GSP system are first defined. The  $x$ - and  $y$ -axes are parallel with the floor, and the  $z$ -axis is defined as being orthogonal to the floor. In addition, all the three axes are going through the centre of the lower platform. The coordinate axes are depicted in Figure 3.3.



**Figure 3.3:** The coordinate axes that are defined in the GSP system.

Three vectors are introduced in order to calculate the length of each actuator. Those vectors models the position of an actuator's attachment points on both the platforms, and they are denominated  $u_i$ ,  $v(t)$ , and  $w_i$ . Figure 3.4 displays how those vectors are defined for an arbitrary actuator,  $i$ . There exist one  $u_i$  and  $w_i$  vector for each actuator. The vectors  $u_i$  and  $w_i$  originate from the centre of the lower and upper platforms, respectively, and their endpoints are the attachment points on the corresponding platform. The magnitude of both those vectors is fix - the distance between the attachment points and the centre of both platforms is not alterable. Moreover, since the centre points of the two platforms are shared by the actuators, the vector  $v(t)$  is used by all actuators to distinguish the offset between the platforms' centres. Note that the vector  $v(t)$  is dependent on time since it consists of the spatial reference coordinates that corresponds to an automobile's chassis motion. Those coordinates are continuously changing with time as the automobile is being operated.



**Figure 3.4:** A graphical display of the vectors  $u_i$ ,  $v(t)$ , and  $w_i$  that are used to calculate the length of the  $i$ th actuator. In addition, the angle between two adjacent actuators,  $\alpha$ , is also depicted.

In order to calculate the magnitude of the vector  $u_i$ , it is first observed that the attachment points on the lower platform are distinguished from one another both by the radius of the platform, as well as by a revolution about the  $z$ -axis. This is due to the fact that the actuators are attached to the lower platform in a symmetric pattern. (The attachment points on the upper platform are located symmetrically as well.) The attachment points on both platforms are divided into three pairs, where the centre distance of each pair is revoluted one third of a circle from the other two pairs' centre distance. By combining this with the offset angle  $\alpha$ , which is the angle between two attachment points in the same pair, an expression of the mutual

position of each attachment point on the lower platform is obtained. The angle  $\alpha$  is depicted in Figure 3.4. Equation (3.4) displays a vector,  $V_{low}$ , that contains the angular mapping of all actuators' attachment points on the lower platform.

$$V_{low} = \left[ \pi - \frac{\alpha}{2} \quad \frac{\pi}{3} + \frac{\alpha}{2} \quad \frac{\pi}{3} - \frac{\alpha}{2} \quad -\frac{\pi}{3} + \frac{\alpha}{2} \quad -\frac{\pi}{3} - \frac{\alpha}{2} \quad \pi + \frac{\alpha}{2} \right]^T \quad (3.4)$$

Moreover,  $\frac{\alpha}{2}$  is the angle between one attachment point and the centre distance of the pair that this attachment point belongs to. Thus,  $\frac{\alpha}{2}$  is negative in three cases, and positive in the other three scenarios.

To obtain the  $x$ - and  $y$ -coordinates for the attachment points on the lower platform, Equation (3.4) is used in a sinusoidal function, and then multiplied with the radius of the lower platform,  $R$ . This is displayed in Equation (3.5). Note that the attachment points on the lower platform are located at a height of 0, which means that they are defined to not have performed any heaving motion. Hence,  $x_l(i)$ ,  $y_l(i)$ , and  $z_l(i)$  are scalars corresponding to the coordinates of the lower attachment point of the  $i$ th actuator, where  $1 < i < 6$ .

$$x_l(i) = R \cdot \cos(V_{low}(i)), \quad y_l(i) = R \cdot \sin(V_{low}(i)), \quad z_l(i) = 0^{6 \times 1} \quad (3.5)$$

Thus, the vector  $u_i$  consists of the coordinates of the  $i$ th actuator's attachment points. An expression for  $u_i$  is depicted in Equation (3.6).

$$u_i = \begin{bmatrix} x_l(i) & y_l(i) & z_l(i) \end{bmatrix}^T \quad (3.6)$$

A calculation of the distance between the centre of the upper platform and the attachment points on that platform – the vector  $w_i$  – is performed in a similar manner. The difference, though, is that the entire upper platform is revolved  $\frac{\pi}{3}$  radians compared to the lower platform. This revolution must be taken into account when mapping the attachment points on the lower platform to those of the upper platform. A vector containing this angular mapping of the attachment points on the upper platform is thus obtained in accordance with Equation (3.7), where the subtraction of  $\frac{\pi}{3}$  is performed element-wise from  $V_{low}$ .

$$V_{up} = V_{low} - \frac{\pi}{3} \quad (3.7)$$

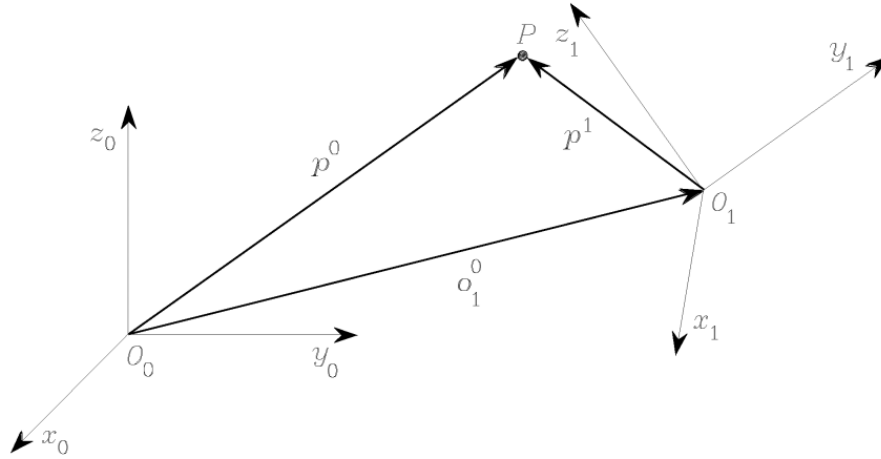
As with the attachment points on the lower platform,  $V_{up}$  is used in a sinusoidal function and then multiplied with the radius of the upper platform,  $r$ , to obtain an expression for the vector  $w_i$ . Equation (3.8) depicts the calculation of the coordinates for one arbitrary attachment point on the upper platform. Note that  $z_u$  is a vector with six elements, which all of them have the value zero. This is due to the fact that the GSP has not performed any heaving motion when it resides in its initial state.

$$x_u(i) = r \cdot \cos(V_{up}(i)), \quad y_u(i) = r \cdot \sin(V_{up}(i)), \quad z_u(i) = 0^{6 \times 1} \quad (3.8)$$

The upper platform coordinates of each attachment point is then used to derive an expression for  $w_i$ , likewise as the attachment points on the lower platform are utilised to calculate  $u_i$ . The vector  $w_i$  is displayed in Equation (3.9).

$$w_i = \begin{bmatrix} x_u(i) & y_u(i) & z_u(i) \end{bmatrix}^T \quad (3.9)$$

Frames are utilised to obtain expressions for the distances between the attachment points on both platforms. This is due to the fact that frames are capable of defining rotation and translation differences between two arbitrary, spatial points [34]. Figure 3.5 gives a small example of how frames can be used to model such differences. There, the distance between two frames; the base frame, which is denoted  $O_0$ , and another frame,  $O_1$ , is modelled. The relation between movable parts of a robot can be expressed easily by locating a frame in each component that can perform motion.



**Figure 3.5:** An example displaying the distances between a base frame,  $O_0$ , and one additional frame,  $O_1$ . (The picture is taken from the textbook *Modelling and Control of Robot Manipulators* [34])

In the modelling of the GSP, a base frame is located in the centre of the lower platform, and an additional frame is placed in the centre of the upper platform. The distance between the centres of the lower and upper platforms is characterised by the spatial offset of the upper platform compared to the lower one. This offset is modelled by the vector  $v(t)$ . The calculation of the vector  $v(t)$  is displayed in Equation (3.10).  $x(t)$ ,  $y(t)$ , and  $z(t)$  corresponds to the spatial motion references of an automobile's chassis.

$$v(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T \quad (3.10)$$

To calculate the distance between the actuator attachment points on both platforms, the distance between the base frame (the centre of the lower platform) and the attachment points on the upper platform is first worked out. In order to obtain such an expression, rotation matrices are used to calculate how the upper platform moves in relation to the lower one. There are three rotation matrices in total –  $R_\theta(t)$ ,  $R_\varphi(t)$ , and  $R_\psi(t)$  – that corresponds to a revolution of one Euler angle.  $R_\theta(t)$  is the rotation matrix for the roll angle,  $R_\varphi(t)$  corresponds to the rotation matrix for the pitch angle, and  $R_\psi(t)$  depicts the rotation matrix the yaw angle, respectively. Those rotation matrices are displayed in Equations (3.11)-(3.13). The rotation matrices

are dependent on time since each one of them is utilising one angular reference revolution that an automobile's chassis performs.

$$R_\theta(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta(t)) & -\sin(\theta(t)) \\ 0 & \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix} \quad (3.11)$$

$$R_\varphi(t) = \begin{bmatrix} \cos(\varphi(t)) & 0 & \sin(\varphi(t)) \\ 0 & 1 & 0 \\ -\sin(\varphi(t)) & 0 & \cos(\varphi(t)) \end{bmatrix} \quad (3.12)$$

$$R_\psi(t) = \begin{bmatrix} \cos(\psi(t)) & -\sin(\psi(t)) & 0 \\ \sin(\psi(t)) & \cos(\psi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

To obtain one rotation matrix that incorporates the revolution about all three coordinate axes,  $R_\theta(t)$ ,  $R_\varphi(t)$ , and  $R_\psi(t)$  are multiplied into one rotation matrix,  $R_{tot}(t)$ , which is displayed in Equation (3.14).

$$R_{tot}(t) = R_\psi(t) \cdot R_\varphi(t) \cdot R_\theta(t) \quad (3.14)$$

By multiplying  $R_{tot}(t)$  with  $w_i$ , an expression is obtained that describes how the rotational motion of the upper platform is affecting the relation between the attachment points on the upper platform and that platform's centre. Adding this expression with the vector  $v(t)$ , that is portraying the offset between the centres of both platforms, results in a vector that describes the distance between the centre of the lower platform and the attachment points on the upper platform. By subtracting this vector from  $u_i$ , a vector  $-P(t)$  between the attachment points on the lower and upper platforms is obtained. The calculation of  $P(t)$  is depicted in Equation (3.15).

$$P(t) = u_i - (v(t) + R_{tot}(t) \cdot w_i) \quad (3.15)$$

The length of an actuator,  $L(t)$ , is then obtained by calculating the Euclidean norm of the vector  $P(t)$ , as is seen in Equation (3.16). The Euclidean norm is utilised to obtain the one-dimensional, free-floating, length of an actuator from the three-dimensional vector  $P(t)$ .

$$L(t) = ||P(t)|| \quad (3.16)$$

The velocity that an actuator should be altered with is obtained by an interpolation of the calculated values of its length. The actuator length during the previous program iteration is subtracted from the calculated length during the current iteration, and the result is divided by the time it takes to perform one program iteration, i.e. the sampling time,  $T$ . This calculation of an actuator's velocity is depicted in Equation (3.17). Note that  $T$  consists of the processing time of both the VN8912 module and an actuator driver, as well as the delay that the CAN bus invokes on the communication.

$$V(t) = \frac{L(t) - L(t - T)}{T} \quad (3.17)$$

Likewise, the acceleration of an actuator is calculated by an interpolation of its velocity in the very same manner as its length is processed to obtain said velocity. The calculation of the acceleration is presented in Equation (3.18).

$$A(t) = \frac{V(t) - V(t - T)}{T} \quad (3.18)$$

The lengths that all the actuators should have during operation of the GSP, which are obtained from Equation 3.16, are calculated with the software environment @Mathematica [35]. This is done in such a way that an offline expression of the lengths is implemented as a function in the VN8912 module, where the spatial and angular reference data of an automobile's chassis motion are inputs. This function is then invoked each program iteration when the VN8912 module is handed new reference values. The velocity and acceleration, which are computationally less demanding to obtain, are calculated by the VN8912 module each iteration.

## 3.2 Handling of Discontinuities in Motion Input

Motion of the GSP is handled by two different subsystems; one distributed control system, and one centralised logic unit – the VN8912 module. The distributed control system consists of the six equal, albeit independent, PID regulators, which are embedded in their respective actuator driver. The technical workings of the PID regulators are detailed in Section 2.2.3, and Section 3.2.2 covers how those regulators are used to control the length of the actuators.

The VN8912 module is used to receive and process references that contains data of an automobile's chassis motion. This data is generated from either a sensor in a real automobile, a simulation program (which is outlined further in Section 2.4.3), a log file containing previously sampled driving data, or from a GUI that is implemented in CANoe. This GUI is described in Section 3.3.3.

Upon receiving reference data, the VN8912 module utilises the generalised vector function from Section 3.1.3 and the motion envelope protection algorithm discussed in Section 3.1.2 to calculate feasible set-points for the linear motors' PID regulators. Those set-points are sent along the CAN bus to the driver of each actuator. The embedded PID regulator then controls the length of the actuators in accordance with the calculated set-points.

### 3.2.1 Set-point Calculation from Reference Data

When the VN8912 module receives reference data consisting of an automobile's chassis motion, it is invoking the motion envelope protection algorithm that is detailed in Section 3.1.2 to guarantee that the reference data only results in feasible motion of the GSP. Afterwards, the VN8912 module is applying the generalised vector function that is explained in Section 3.1.3 to the reference data. With this function, the VN8912 module calculates the lengths that the GSP's actuators must have in order to perform the same motion as that of the chassis. Along with the calculated lengths of each actuator, the corresponding velocity and acceleration is also utilised so as to let the GSP mimic the chassis motion as closely as possible. With the lengths, velocities, and accelerations of each actuator, the VN8912 module calculates set-points for the linear motors' PID regulators. Those set-points are then transmitted to the actuator drivers. During operation, the VN8912 module is transmitting the set-points in a streaming fashion. The way that this streaming motion works is that the VN8912 module is calculating the required actuator lengths for the desired position of the upper platform each program iteration, and then it sends those set-points continuously to the drivers. Thus, the PID regulators are constantly obtaining new set-points. This results in a continuous – uninterrupted – alteration of the GSP's actuators, which enables real-time imitation of chassis motion.

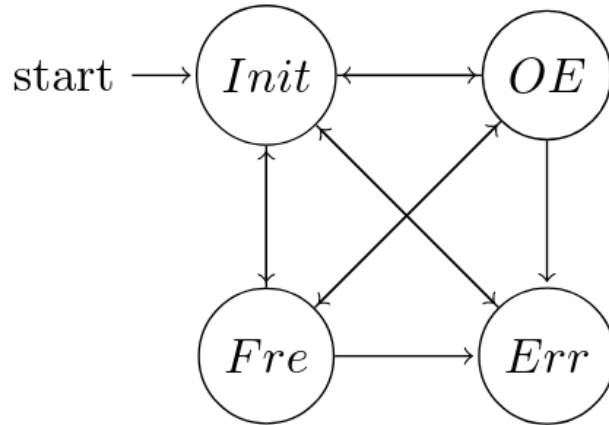
In addition, the velocity that the actuators can be altered with is saturated in order to not allow for too quick motion of the GSP. Thus, this saturation serves as an extra security measure on top of the motion envelope protection algorithm. The rate that this saturation is limited to is 50 micrometres per five milliseconds. Hence, the rate limitation will occur if the GSP is used to mimic chassis motion where the actuators must change with a faster velocity than what is allowed. Then, instead of imitating the exact motion sequence performed by the chassis, the GSP will reach the next feasible set-point as fast as the rate limitation allows. This results in a less exact imitation of the chassis motion in exchange for not damaging the hardware of the GSP.

However – as is detailed in Section 2.3 – due to limitations on the CAN bus, the VN8912 module can only transmit set-points to a specific actuator driver every fifth millisecond. This constraint implies that the transmission of set-points to the actuator drivers, as well as the imitation of real-time chassis motion of the GSP, are delayed by five milliseconds.

In order to implement the logic that the VN8912 module executes, the software environment CANoe is utilised. Section 3.3 further describes how this implementation is performed.

#### 3.2.1.1 Finite State Automaton

To ease the operation of the GSP, there is also an implemented Finite State Automaton (FSA) in the VN8912 module. The purpose of the FSA is to simplify the operation of the VN8912 module by introducing four different states that it can be in. Those four states are called *Initialise*, *Operation Enabled*, *Freezing*, and *Error*. A block diagram of the entire FSA is depicted in Figure 3.6. For brevity purposes, the four states in the FSA are abbreviated as *Init*, *OE*, *Fre*, and *Err*, respectively.



**Figure 3.6:** The FSA that is embedded within the VN8912 module.

In the Initialise state the VN8912 module is powered on and homed, which means that a calibration of the sensors in each actuator is performed. This is done to allow the VN8912 module to obtain known positions of each actuator's rotor, which is required for precise motion control of the upper platform. No motion command can be transmitted from the VN8912 module in this state. Initialise only serves as a start up state, or a state that is entered after an error has been solved to (re-)calibrate the sensors.

The Operation Enabled state is the state in which motion of the upper platform is permitted. In this state, the VN8912 module is able to both transmit and receive data on the CAN bus. It is also in this state that new driver set-points are calculated based on the reference data containing chassis motion that is fed to the VN8912 module. The Operation Enabled state is entered automatically once the homing sequence in the Initialise state has finished. The VN8912 module remains in the Operation Enabled state until either the power is switched off, the operator manually switches to the Initialise state through commands in the GUI, or an error occurs.

If an error is triggered the VN8912 module enters the Freezing state. Here, the ongoing motion of the GSP is immediately stopped and the actuators hold their current position. Thus, the GSP is frozen in place. Furthermore, the VN8912 module is constrained from transmitting any additional motion references to the actuator drivers. An error is triggered if any of the emergency measures are activated (the emergency button on the electrical enclosure – see Section 2.2.4 – is being pressed, or the safety mat outlined in Section 2.2.5 is walked upon). Thus, in the Freezing state, the actuator drivers are still supplied with power. This is necessary in order to produce a freezing behaviour as the drivers are being set to track the position which they currently have at the moment when the error occurs.

Moreover, the Error state is rarely used. This state is an error state implemented in the actuator drivers by the manufacturer of the linear motors. The problem, though, is that in the Error state power is dropped to the drivers, and thus they are not able to hold the current position of their rotors in this state. Due to that, this state poses as a big problem when it comes to fault-handling of the GSP, since



with the power being abruptly cut to the actuator drivers, the upper platform is falling down. This might potentially damage the GSP, its mounted headlights, or the floor upon which the test rig is placed. Therefore, the Error state of the FSA is only entered by the VN8912 module in case of a complete power loss.

The logic of the FSA is implemented with CAPL as an active node directly in the CANoe environment of the VN8912 module.

### 3.2.2 Control of Actuator Lengths

As is stated in Section 2.1.1, each actuator has got a corresponding PID regulator, which is implemented in the actuator driver. Those regulators are operating independently of one another. Set-points for the PID regulators are calculated by the VN8912 module, as is described in Section 3.2.1.

The tuning of the PID regulators is performed such that the power required by a driver to alter its corresponding rotor's position is kept at a minimum. As of such, the PID regulators are tuned to address a minimum energy optimal control problem every time the rotor changes its position. Since the drivers require merely a small amount of power for operation of the linear motors, it is straightforward to connect them with exterior devices, or embed them in larger circuits. In the test rig, this allows for an easy interaction with the other hardware components. Since the PID regulators in the distributed control system are of such a plug-and-play type, they allow for a simple way of controlling the GSP's upper platform. This is necessary in order for the GSP to be able to mimic chassis motion in real-time. The software LinMot-Talk, which is further described in Section 2.4.4, is utilised when tuning the PID regulators.

All the PID regulators of the six drivers are tuned with the same settings in order to make sure that the six actuators are behaving similarly to one another. Furthermore, each PID regulator is using feedforward to compensate for the behaviour of the plant, which in this scenario is the actual linear motor that correspond to that specific PID regulator. There are four current parameters that are being feedforwarded, which all are expressed in units of ampere. Those consists of the current that is used to compensate for a constant force control of the rotor, the current that is used to compensate for dry friction arising from the rotor sliding through the linear motor, a factor for compensation of viscous friction, as well as an acceleration factor. By using feedforward, the linear motors performs near-instant motion of the rotors once a motion command is received. Thus, by extending each PID regulator with such a feedforward technique, the GSP is operating as close to real-time as is possible.

In addition to utilising feedforward, each PID regulator is also equipped with a noise filter with a deadband interval. The reason for this is that the PID regulator setup influences the behaviour of that regulator's driver. Hence, the noise filter is used to filter out any disturbances that are present in the position feedback signal of the PID regulator. Thus, there are no oscillations or repeated active-deactive cycles present in the control of the rotors. The deadband interval that is used is set to 10 micrometers.

## 3.3 Software Implementation

This software implementation chapter first outlines how the CANopen protocol is set up for the communication between the VN8912 module and the actuator drivers. This is followed up by a description of how the logic in the VN8912 module is implemented inside the CANoe software environment. The chapter also explains the workings of a developed GUI, which functions as a tool that allows for operation of the GSP. In addition, it is briefly covered how the software environment @MatLab is utilised to understand the GSP's kinematics during early stage development of the mathematical model.

### 3.3.1 CANopen Settings

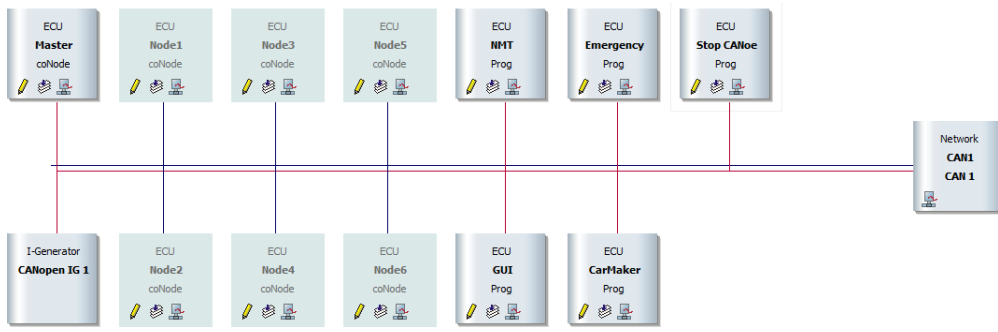
The PDOs utilised in the communication between the VN8912 module and the actuator drivers are being transmitted in an asynchronous manner. The PDOs are sent from the VN8912 module to the actuator drivers every fifth millisecond. However, the important aspect is that five milliseconds is only the average time it takes to send a PDO. In reality, the total cycle time that is required to send one PDO from the VN8912 module to a driver alternates within the interval  $[4.980 \ 5.002]$  milliseconds. The same delay is present when one of the drivers communicates with the VN8912 module.

Moreover, the PDOs are transmitted after an internal trigger in either the VN8912 module or the actuator drivers is set. On the VN8912 module side, the trigger is activated when new reference values are computed. This ensures that the VN8912 module is able to send new set-points to the actuator drivers as soon as it has managed to calculate them.

Furthermore, the baud rate at which CANopen is used to transmit the PDOs along the CAN bus between the VN8912 module and the actuator drivers is set to 1 Megabaud. The baud rate is set as high as possible to minimise the delay from data transmissions between the VN8912 module and the actuator drivers. Note that five milliseconds are required for one of the devices to transmit data to its receiver. In order for the VN8912 module and an actuator driver to exchange information between one another, a total time of 10 milliseconds is therefore required.

### 3.3.2 Implementation in CANoe and CAPL

The CANoe backbone consists of several nodes, as is seen in Figure 3.7. The node called Master corresponds to the VN8912 module, and it contains the logic that allows the hardware interface to transmit and receive data to and from the actuator drivers. This logic is implemented with the use of CAPL code.



**Figure 3.7:** The CANoe environment with visible nodes that holds the implemented logic of all devices in the test rig.

In the same manner as with the Master node, Node 1 to Node 6 contains the logic on the actuator side for handling the PDO communication. In addition, each respective linear motors' sensor logic, as well as its PID regulator implementation, is found in Node 1 to Node 6.

The logic of the VN8912 module is implemented in the node titled *VN8912*, and the Node called *GUI* holds the logic of a user interface. This user interface allows the operator to freely specify set-points for the upper platform, which then are processed to produce according motion of the GSP. The user interface is further described in Section 3.3.3. Moreover, the *Emergency* node holds the logic for handling of emergency stops of the GSP, and the *CarMaker* node enables the integration with the automobile simulation program. In addition, the *Stop CANoe* node contains routines that allow for a safe shutdown of the CANoe environment. Such a safe shutdown also incorporates a power off sequence of the GSP.

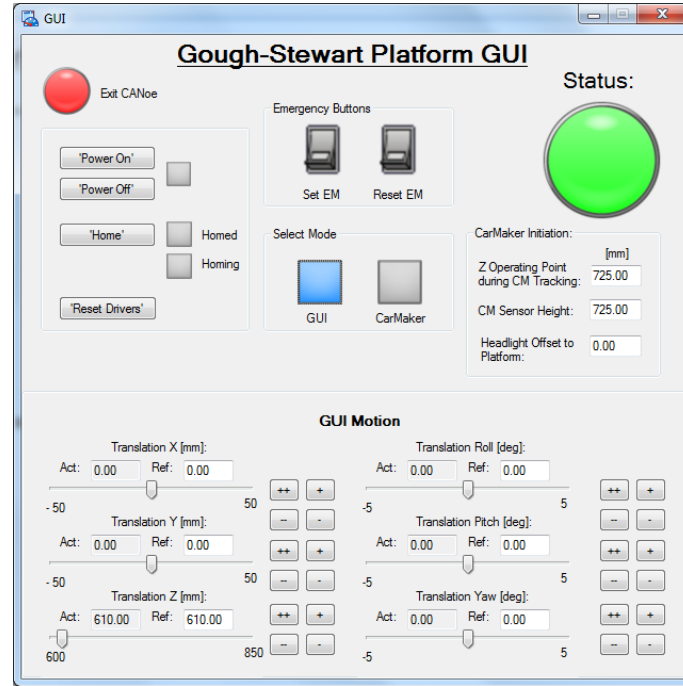
There is also a node named *NMT*, which is an abbreviation for network management. The logic in this node is required in order for the CANopen communication to work between the VN8912 module and the actuator drivers.

Furthermore, there exists two hidden nodes as well, which are not visible in the backbone of the CANoe environment displayed in Figure 3.7. Those nodes are called *Main* and *Universal Functions*, respectively. *Main* is merely a node that contains declarations for all the variables and timers that are utilised in the software implementation of the VN8912 module's entire logic. Moreover, *Universal Functions* contains, as the name might suggest, all the functions that are shared between different nodes. The reason as to why declarations and general functions are stored in their separate node is to simplify the implementation of the test rig's logic.

### 3.3.3 Graphical User Interface

Besides handling the data transmission between the VN8912 module and the actuator drivers, CANoe has also got an embedded design tool for a graphical user interface (GUI). The GUI serves as a way for the operator to easily select settings for the GSP, and also to invoke some pre-determined motion commands of the upper platform. Thus, it is possible to conduct – very simple – moves without imitating any chassis motion. Furthermore, the GUI provides a user-friendly way of displaying for the operator how the devices that communicates with the CANopen protocol

behaves. As is seen in Figure 3.8, the GUI contains, among other functions, a status lamp indicating if the operation of the GSP is run error-free, an exit CANoe button (upper left corner), a homing button, and buttons that allow for small spatial and angular offsets of the upper platform.



**Figure 3.8:** The GUI that is used to select between the operative modes of the GSP.

The status lamp in the upper right corner is lit green when operation is non-erroneous, and it turns to red when an error has occurred. As a safety precaution, the GSP is unable to perform any motion when the status lamp is red. The smaller red light in the upper left corner functions merely as an exit CANoe button. Once pressed, the GSP is powered down, all measurements ends, and the GUI closes. By pressing the *Home* button, the operator causes the GSP to perform a homing sequence. This homing sequence is further described in Section 3.2.1.1. The *Reset Drivers* button can be used to restart the actuator drivers in case that any of those have encountered a fault. In the upper middle of the GUI, the two emergency buttons *Set EM* and *Reset EM* are used to set and reset the emergency stop of the GSP, respectively. The text fields in the middle right of the GUI are used to specify actual distances between the headlights and the upper platform's position relative to the positioning of the headlights in an actual automobile. The sliders at the bottom of the GUI displays the GSP's current motion in each degree of freedom. Those sliders are updated in real-time as the GSP is being operated. The minimum and maximum bounds of motion in each degree of freedom are displayed under the sliders. Surging, swaying, and heaving motions are measured in millimetres, and the angular motions are specified in degrees. When the GSP is operational, the motion envelope protection algorithm described in Section 3.1.2 saturates motion that is

infeasible.

There are two operational modes in the GUI; *CarMaker* and *GUI*, which can be toggled by switching between the two buttons labelled CarMaker and GUI in the very middle of the GUI. In the CarMaker mode, the automobile simulation software that is discussed in Section 2.4.3 is used to generate references that contains chassis motion data. Those references are then passed through the VN8912 module, which utilises the mathematical model to convert them into set-points that are comprehensible by the distributed control system. Directly below the status lamp is a text field where the operator has to specify the offset between the headlights and the centre of the upper platform. The value of this offset depends on how the headlights are attached to the test rig. The GSP is then utilised to mimic the chassis motion of the automobile from the simulation program. On the other hand, the GUI mode allows the operator to steer the GSP by either using the buttons at the bottom of the GUI, or by entering a set-point in the text field besides those buttons. There are four buttons that correspond to each degree of freedom; two that invokes positive motion, and two that causes the GSP to conduct an equally sized negative motion. Small or large motion in each degree of freedom is specified by selecting the buttons with single or double signs. A button with a double sign correspond to a larger motion, and smaller motion is invoked by pressing the buttons with a singleton sign. Furthermore, the operator can freely enter a set-point for a specific degree of freedom in the text field on the right side above each slider. The text field on the left side is updated in real-time along with the slider, and it displays the GSP's current offset in that particular degree of freedom. Regardless of how the GSP's set-points are generated, invalid motion of the GSP is discarded by the implemented safety system, which is described thoroughly in Section 3.1.2.

If the CarMaker mode is selected for the generation of set-points, the buttons that are used during the GUI mode to change the position of the upper platform are disabled. Likewise, if the GUI mode is selected, the automobile simulation program cannot be used to generate reference values of chassis motion.

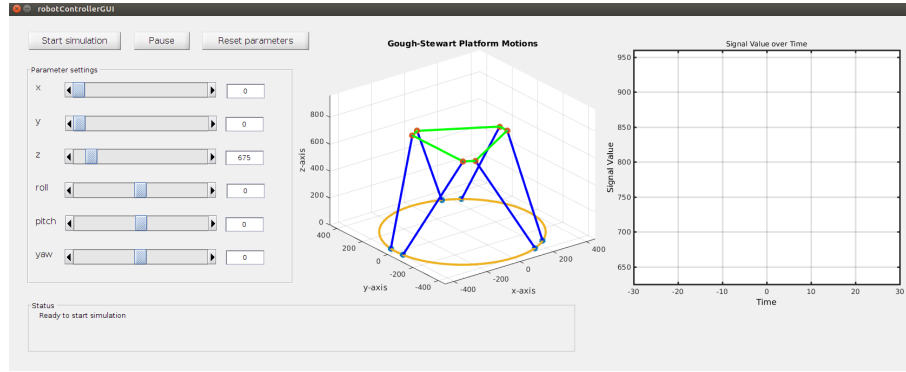
One benefit of developing the GUI with CANoe's embedded tool is that information about the data transmission is already incorporated within the CANoe environment – it does not have to be extracted from an exterior source. Therefore, the sliders and the text fields which displays the GSP's current offset in each degree of freedom can be updated in real-time, with minimal delay.

### 3.3.4 Realisation of Kinematics in @MatLab

A simple GUI that displays the motional behaviour of the GSP is produced in the @MatLab software environment, which is a matrix-based programming language [36]. The purpose of this GUI is mostly to serve as a means to grasp the impact of actuator kinematics, which is especially convenient during early stage modelling of the GSP. As is seen in Figure 3.9, the @MatLab GUI is divided into three components. The leftmost component displays sliders for each degree of freedom. By either moving the sliders, or writing a fix position- or angular value in the text box next to them, the corresponding degree of freedom is updated. In addition, the @MatLab GUI contains a graphical representation of the GSP, as well as a plot

### 3. Modelling and Control System Design

in the rightmost GUI component, which displays the length of each actuator. Both the virtual GSP and its actuator lengths are updated in real-time when the degrees of freedom are changed.



**Figure 3.9:** The  $\text{\textcircled{R}}$ MatLab GUI that is displaying the kinematic behaviour of the GSP.

Furthermore, the leftmost component also contains the three buttons; *Start simulation*, *Pause*, and *Reset parameters*. The first button either starts a new motion simulation of the GSP, or resumes a paused one. Moreover, the Pause button is used to halt an ongoing simulation, and the Reset parameters button simply sets the degrees of freedom to their initial values. Those initial values are displayed in Table 3.1. Note that the surging, swaying, and heaving motions are expressed in units of millimetres, and the angular motions roll, pitch, and yaw are expressed in radians.

$x$	$y$	$z$	$\theta$	$\varphi$	$\psi$
0	0	725	0	0	0

**Table 3.1:** The initial values for each degree of freedom that are utilised in the  $\text{\textcircled{R}}$ MatLab GUI.

# 4

## Results

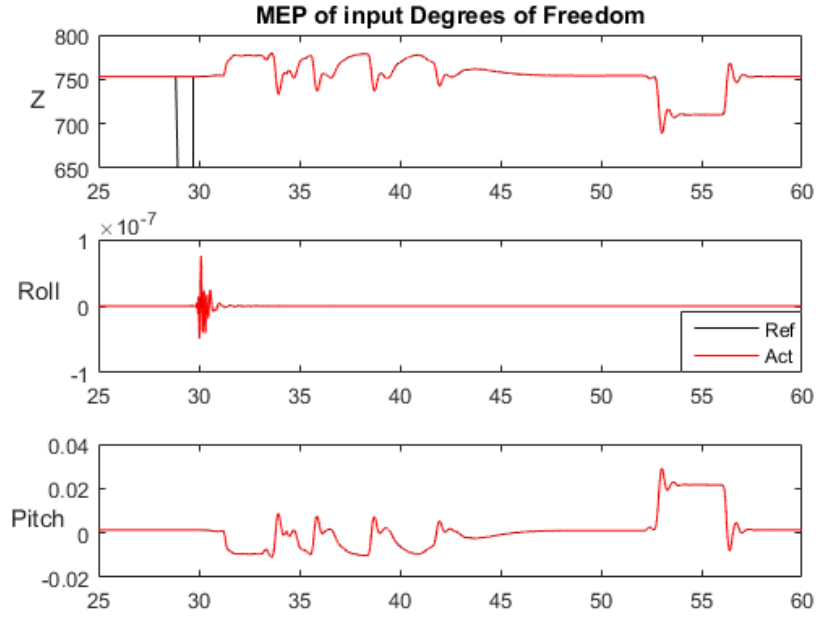
The GSP's ability to mimic chassis motion, as well as its suitability for use in the adaptive headlights dynamics simulations that are performed by Volvo Cars, are covered in this chapter. Firstly, an analysis of the chosen mathematical model is undertaken, where it is also highlighted why inverse kinematics are not utilised. Then, the behaviour of the control system is reviewed, and an exposition of how good the GSP is at imitating chassis motion is given. Lastly, a verdict is stated regarding if the GSP can be used for the adaptive headlights dynamics simulation that Volvo Cars want to perform.

### 4.1 Evaluation of the Mathematical Model

The mathematical model receives input reference data in the form of three spatial coordinates, as well as the revolutions around those axes. The input data is then processed in order to generate set-points that the control system of the GSP utilises to mimic the chassis motion of automobiles.

Furthermore, the motion envelope protection algorithm is implemented in order for the VN8912 module to discard reference data that results in infeasible motion of the GSP. Such infeasible data consists of, e.g., references that deviate too much from the current position. Starting at its initial position, the upper platform can perform surging and swaying motion of at most 50 millimetres in both positive and negative direction. In the same manner, the maximum heaving motion from the initial position has a magnitude of 125 millimetres. Angular revolutions around the coordinate axes are likewise constrained to the interval  $[-5, 5]$  degrees.

Figure 4.1 displays the behaviour of the motion envelope protection algorithm when it is applied to reference data from a sensor in an automobile's chassis. It is seen in the uppermost plot, which is displaying the imitation of the automobile's chassis heaving motion, that the too large motions occurring between 28 and 29 seconds are being saturated. These motions arises from a shake that occurs when a new driving scenario is initialised in the simulation program, and they are not part of the actual driving scenario itself. However, regardless of why such big deviations in the reference data are present, the motion envelope protection algorithm saturates those. The references of automobile chassis motion is generated by the simulation program discussed in Section 2.4.3.



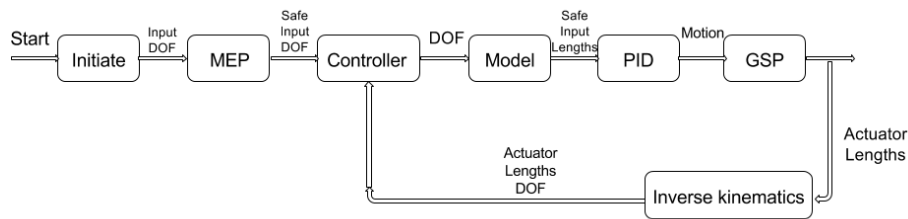
**Figure 4.1:** A display of when the motion envelope protection algorithm saturates infeasible heaving motion references.

#### 4.1.1 The Inverse Kinematics Complexity

As is evident from the modelling of the GSP, it is possible to convert the three spatial coordinates into the correct length of each actuator. The calculation of the actuator lengths are therefore possible by applying forward kinematics. However, it is not an easy calculation to go the other way around; namely starting with the lengths of all six actuators, and then calculating values for the different spatial coordinates – a process known as inverse kinematics [33]. This is due to the fact that the mathematical model of the GSP results in an underdetermined system of equations when inverse kinematics are invoked.

However, if inverse kinematics are feasible to utilise, they can be combined with an outer control-loop to strengthen the automobile chassis imitation performance of the GSP. This is possible by first feedbacking the actual lengths of each actuator to the VN8912 module. Then, the inverse kinematics are applied to the feedbacked lengths in order to calculate the spatial coordinates corresponding to each actuator's actual length. This enables the VN8912 module to house a centralised controller that incorporates the error between the actual and referred spatial coordinates into the calculation of the next actuator set-points. This gives, based on how the centralised controller is implemented, a higher control accuracy of the GSP compared to when the set-points for the actuator drivers are calculated only from the current reference values. A flow chart of such an outer control-loop is depicted in Figure 4.2.





**Figure 4.2:** A flow chart of the GSP’s control system with known inverse kinematics and an outer control-loop.

The reason for why inverse kinematics gives rise to an underdetermined system is due to the fact that each length of the actuators is calculated from the three spatial coordinates. Therefore, it is not possible to – in an easy way – reverse the process and distinguish what combination of those coordinates that are used to obtain a specific length.

Utilising inverse kinematics thus gives rise to an equation system with six known variables; the actuator lengths, and eighteen unknown variables; the spatial coordinates corresponding to each actuator’s length. Such a system of equations does not allow for an easy extraction of the spatial coordinates that correspond to a given actuator length.

## 4.2 Imitation of Automobile Chassis Motion

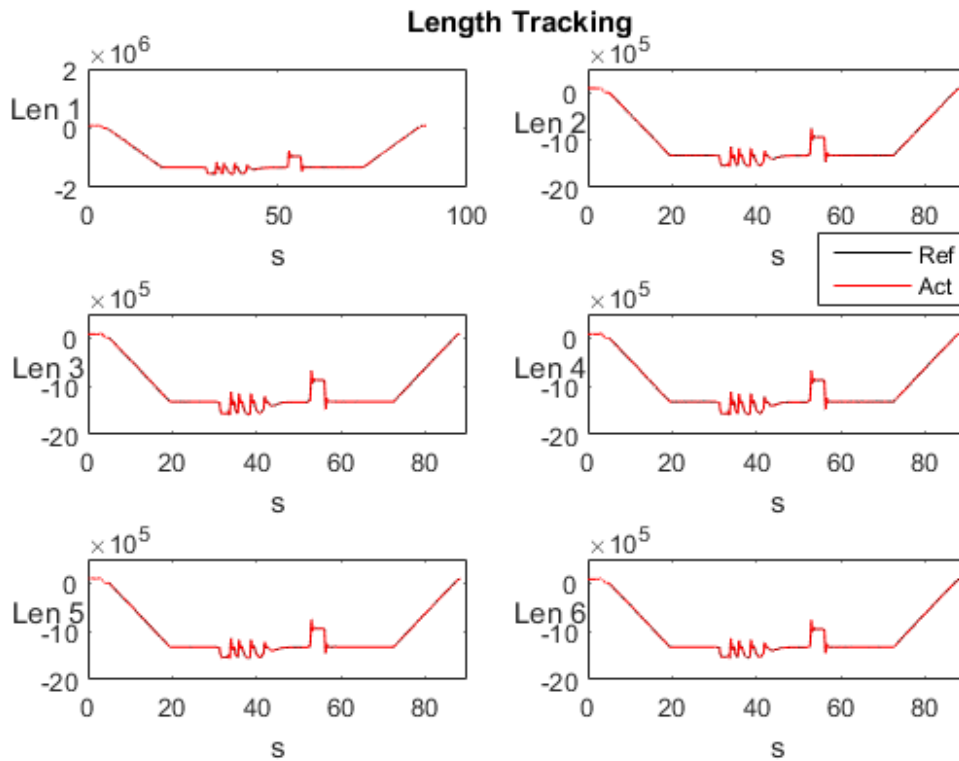
In order to assess the GSP’s automobile chassis imitation performance, the simulation program discussed in Section 2.4.3 is utilised to generate data from a driving scenario. During this simulation, the automobile starts stationary, and then accelerates up to 100 kilometres an hour. Afterwards, it breaks as hard as possible down to zero kilometers and hour. The simulation ends when the automobile is back to being stationary and its chassis has stopped trembling.

The real-time imitating capability of the GSP with attached headlights is sufficient in such a way that the centre of the upper platform is able to mimic the chassis motion of an operated automobile in real-time. In order to distinguish this, the embedded sensors in the linear motors are used to measure the current length of the actuators. Then, this current length is compared to the one that is being calculated by the mathematical model in the VN8912 module.

In Figure 4.3, it is visualised how the actual length of each actuator compares to the calculated length. Furthermore, Figure 4.4 displays an enlargement of the length comparison for the first motor (which is seen in the upper leftmost plot of Figure 4.3). In both figures, the  $x$ -axis is expressed in seconds, and the  $y$ -axis corresponds to the length of the actuators, which is represented in units of 0.1 micrometres. (What has to be noted is that the direction of alteration of the actuators’ lengths is inverted due to the linear motors being installed upside down. This is to make sure that the upwards extending part of the actuators consists of the rotors, and not the actual stators themselves. This guarantees that the lifting capacity is high enough in order for the upper platform to be movable.) As is seen in both figures, the actual

length coincides almost exactly with the calculated one. This is the case during the majority of the time the GSP mimics chassis motion.

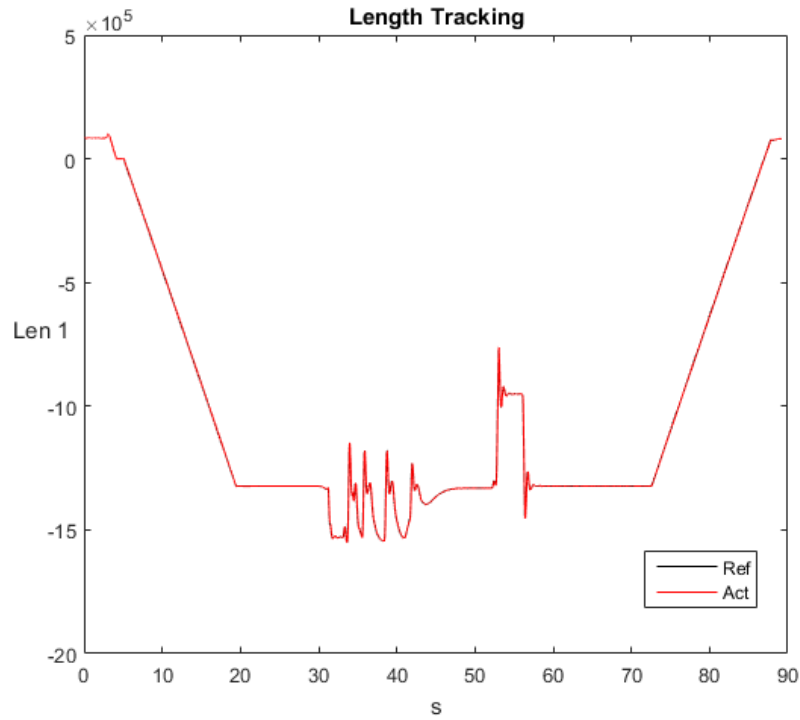
The maximum deviation that occurs between the actual length of the first actuator and its calculated length is 3.8 millimetres. Figure 4.5 depicts this error over time for the breaking scenario. The peaks that contains the largest errors arises from when the automobile is either shifting gears during high velocities, or rapidly performs an abrupt maneuver in an otherwise steady motion pattern – in this case heavy breaking. The gear shifts are seen between 30 and 45 seconds, and the breaking starts at 52 seconds. The smaller peak in error at 5 seconds originates from when the automobile is initiated in the simulation program. Then, the sensor in its chassis registers a small shake before the driving scenario is properly loaded. In order to generate those figures, the tuning of the PID regulators which gives the best imitation performance of the GSP is utilised. This tuning is presented in Table 4.1. In addition, the dead-band of the embedded noise filter is set to 0.01 millimeters during this measurement.



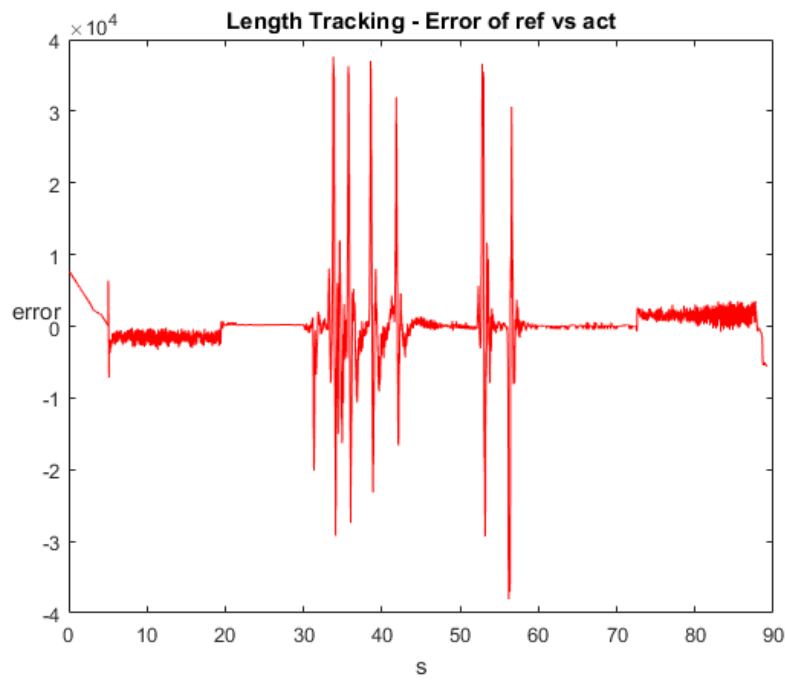
**Figure 4.3:** The imitation capability of the GSP when the reference data is generated by an intense breaking scenario.

P	I	D
11	20	10

**Table 4.1:** The PID regulator tuning that allows for the best imitation performance of the GSP.

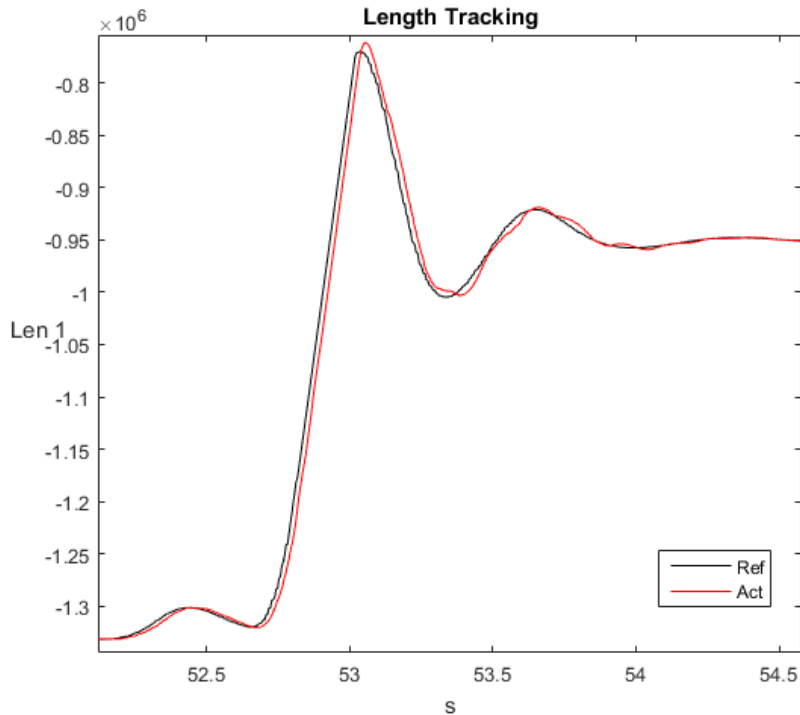


**Figure 4.4:** Enlargement of the imitation capability of the GSP's first actuator when the reference data is generated by an intense breaking scenario.



**Figure 4.5:** The error between the actual length of the first actuator and its calculated length during a driving scenario where an automobile accelerates and then breaks heavily.

Furthermore, the imitation capability of the GSP is delayed by approximately 10 milliseconds. This delay originates primarily from the limitation of transmission rate in the CAN bus. (Data transmission between the VN8912 module and the actuator drivers is delayed by 5 milliseconds each time one of the devices attempts to communicate with the other, as is described in Section 3.3.1.) In addition to the transmission time between the VN8912 module and the actuator drivers, a smaller delay is also originating from the need of an actuator's driver to process the set-point information once received, as well as the fact that the linear motor must physically move the rotor to this new set-point. In Figure 4.6, the delay of approximately 10 milliseconds is visible when an actuator is utilised to track a pre-determined motion pattern. This delay is the main reason for the maximum deviation of 3.8 millimetres that can occur when the GSP is utilised to mimic chassis motion. Since the actual length is being delayed, the calculated length is already at a new set-point when the actual length has reached its set-point. Thus, this deviation occurs.



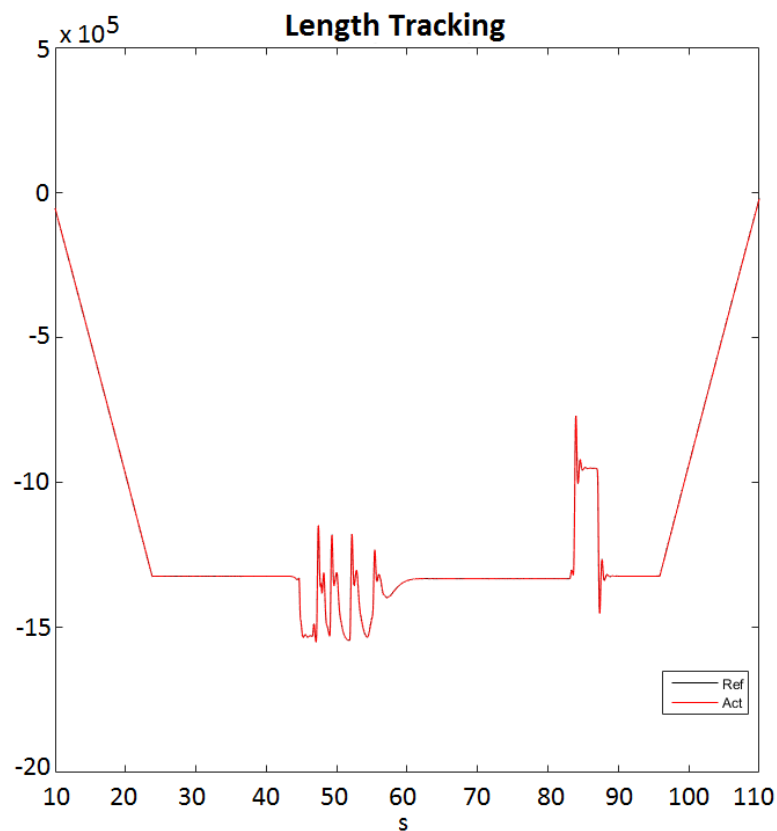
**Figure 4.6:** The tracking delay of the first actuator. This delay does mostly originate from limitations of the CAN bus.

As is evident from Figure 4.3 and Figure 4.4, the sensor in the automobile indicates that the chassis is first moving from a position above zero to a fix negative value. This motion is performed so that the centre of the upper platform is located at the same height as the sensor in the automobile's chassis. Then, the sensor is subject to two large trembles, which both are imitated by the GSP. The first tremble corresponds to when the automobile is accelerating up to 100 kilometres an hour. Then, it performs several gear shifts, which affects the chassis. Evidently, a gear shift

during operation of the automobile causes the chassis to shake. The second, and largest, tremble belongs to the actual breaking sequence of the automobile. During the simulation of the breaking entire scenario, the chassis heaves approximately 8 centimetres at most, which occurs during the intense breaking period.

The breaking scenario is considered as a harsh scenario in terms of chassis motion. This is due to the fact that, during normal operation, the automobile is performing turns and conducts small accelerations, but not intense breaking. Such regular turns and small accelerations do not invoke as large motion of the chassis as heavy breaking does. Hence, the breaking sequence is deemed as a feasible driving scenario for depicting what imitation level of chassis motion that is obtainable during difficult driving circumstances.

Different PID regulator tuning does – naturally – influence the behaviour of the GSP. However, the GSP's ability to mimic chassis motion is only marginally affected by the regulator tuning. When an inferior PID regulator tuning is utilised, the deviations between the actual lengths and the calculated lengths of the actuators are small enough to still allow for a good imitation of chassis motion. This is displayed in Figure 4.7. The inferior tuning of the PID regulators that is used to produce this figure is shown in Table 4.2. Furthermore, a noise filter with a dead-band of 0.02 millimetres is utilised during this experiment.



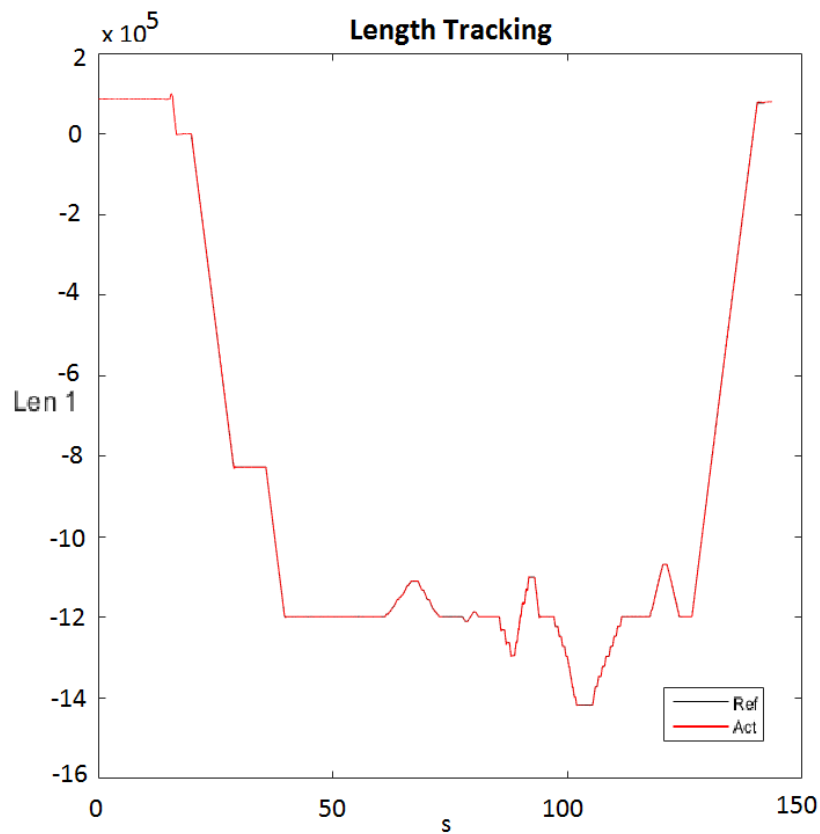
**Figure 4.7:** The imitation capability of the GSP with an inferior PID regulator tuning.

P	I	D
5	20	10

**Table 4.2:** An inferior tuning of the GSP’s PID regulators that still results in a suitable imitation of chassis motion.

Despite the fact that a different PID regulator tuning does not substantially affect the imitation performance, it does serverly influence the amount of vibrations that occurs in the test rig, as is addressed in Section 4.3. However, the GSP’s ability o mimic chassis motion is of course dependent on several factors. One of those are the settings of the PID regulators. Another factor that contributes to the performance is, e.g., what kind of automobile that is being utilised, and also what kind of driving sequence this automobile is operated in. The generated test data of all the previous cases were sampled when the GSP imitated the chassis of an automobile that performed an intense breaking scenario.

In Figure 4.8, the imitation of a simpler sequence is depicted. In this sequence, the upper platform of the GSP is first raised from its initial position at 600 millimetres above floor level to a height of 725 millimetres. Then, a swaying motion of 20 millimetres is performed, which is followed up by a negative sway to reach the point -20 millimetres. Afterwards, the upper platform returns to zero offset in the  $y$ -domain, and repeats the very same sequence, but with a surging motion instead. Lastly, the upper platform is lowered back to its initial position at 600 millimetres above the floor. What is observable from the figure is that the GSP sufficiently manages to mimic easier motion scenarios as well. During this testing sequence, an inferior tuning of the PID regulators is utilised in order to invoke a worse imitation capability. Nonetheless, the GSP does not deviate from the set-points that are provided by the VN8912 module. The inferior tuning is depicted in Table 4.3.



**Figure 4.8:** The effects of an inferior tuning on the GSP's imitation capability during an easier motion scenario.

P	I	D
15	20	10

**Table 4.3:** The inferior tuning used by the GSP's PID regulators during an easier motion scenario.

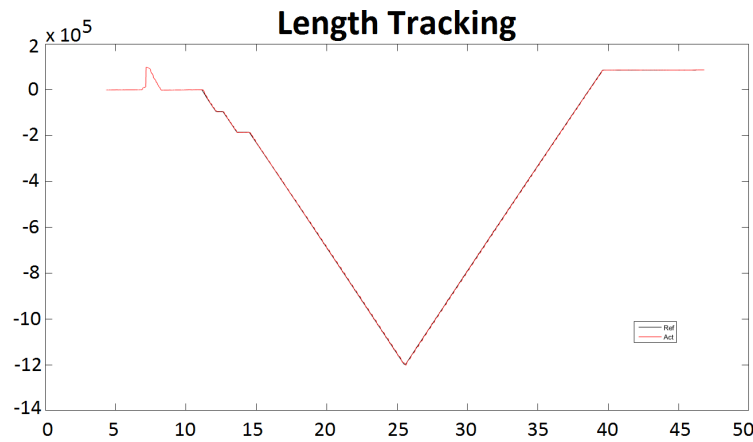
### 4.3 Vibrations of the Attached Headlights

When the GSP is performing motion, the mounted bar with its attached headlights is subject to vibrations. Those vibrations are more tangible at the very ends of the mounted bar – where the headlights are attached – compared to at the centre of the upper platform. Such vibrations are hard to remove only by tuning of the PID regulators. However, it is also observed that the tuning affects the frequency and magnitude of the vibrations to a small extent. With the best tuning the vibrations are rather small – but still present – compared to when a worse tuning is utilised. If the PID regulators are very poorly tuned the GSP is subject to vibrations of the extreme magnitude. Even an easy motion command such as the homing sequence is enough to cause the GSP to shake vigorously. This poor tuning is depicted in

Table 4.4. What is noticeable though, is that the imitation performance of the GSP is still sufficient, as is seen in Figure 4.9. This imitation was interrupted in advance due to the vibrations being deemed to intense to let the GSP be operable for a longer period of time.

P	I	D
5	15	7

**Table 4.4:** A PID regulator tuning of the GSP’s actuators that is considered as very poor.



**Figure 4.9:** A depiction of the GSP’s imitation capability when a very poor PID regulator tuning is utilised.

Despite the fact that the GSP is able to mimic chassis motion sufficiently in real-time, the vibrations of the mounted bar are considered to be too large. The vibrations causes the headlights to jolt too much in order for them to be used for the adaptive headlights dynamics analyses that Volvo Cars want to perform. Thus, the GSP cannot be utilised for real-time simulation of the adaptive headlights dynamics. However, if the GSP is imitating automobile chassis motion at a slower pace, e.g. at 20% of real-time, then the vibrations are non-existent. Hence, in order for Volvo Cars to perform their adaptive headlights dynamics analyses, the GSP must be operated slower than in real-time.

### 4.4 Hazard Analysis and CE Marking of the Test Rig

A hazard analysis is performed in order to guarantee that the test rig is both harmless and reliable in the sense of operator safety. The purpose of the hazard analysis is twofold; firstly, to prove that the test rig poses no danger to any human or property. Furthermore, it is also serving as a basis to CE mark the test rig. This allows Volvo Cars to utilise the GSP in analyses of the dynamics of their adaptive headlights,



which are to be commercially distributed at a later stage.

The hazard analysis relies on the European Commission's machinery directive [37] to determine the potential risks and dangers with the test rig. The outcome of the hazard analysis is that the test rig almost fulfils all fundamental health and security requirements in order to be classified as safe for the operator. However, in order to CE mark the test rig, there is a need to design a better safe region of operation around the GSP. The complete investigation regarding CE marking of the test rig is attached in Appendix A, whereas the actual hazard analysis is found in Appendix B.



# 5

## Conclusion

This chapter brings forth an analysis of all the achieved results, as well as a discussion of the methods with which those results are obtained. Firstly, an explanation is given as to why inverse kinematics are not utilised in the modelling of the GSP. Then is an evaluation of the control system design undertaken. This is followed by a section which addresses the GSP's suitability for use in such an adaptive headlights dynamics simulation that Volvo Cars want to perform. Afterwards are the benefits of the test rig in an environmental aspect covered. Two ideas for improvement of the test rig are finally provided.

### 5.1 Disadvantages of Inverse Kinematics

The chosen control design of the GSP is indeed sufficient for imitating automobile chassis motion. Nonetheless, a better imitation performance would most likely be obtainable if the VN8912 module is controlled in closed-loop. This is because the actual position of the linear motors' rotors are then feedbacked to the VN8912 module, and incorporated in the control design of the GSP. This would open up the possibility for more robust control techniques. One such technique is covered briefly in Section 5.2.

However, in order for the VN8912 module to operate in closed-loop, the dynamics of the GSP must be known. This is due to the fact that the would-be feedback consists of the lengths of the GSP's actuators. With those lengths it is possible to utilise inverse kinematics to calculate the spatial coordinates that corresponds to each actuator's length. But, since the dynamics of the GSP are not known, it is difficult to implement an exact closed-loop control design.

The dynamics of the GSP can, on the other hand, be estimated by utilising system identification. A possible design of experiments that allows for this is to, e.g., invoke small vibrations of the GSP and then measure them. By comparing the magnitude and frequency of the vibrations with how large input force that is used to initiate those, a numerical approximation of the GSP's dynamics can be determined. However, due to timing constraints, such an experiment is deemed to be outside the scope of this thesis project.

Nonetheless, the open-loop control design can be converted into a closed-loop control structure without knowledge of the GSP's dynamics. In such a case, a workaround is required to solve the underdetermined system of equations that arises from applying inverse kinematics. (This underdetermined system of equations emerges from the fact that three spatial coordinates are needed for the calculation of one actuator

length, as is described in Section 4.1.1.)

There exist - at least - two different approaches to handle such an inverse kinematic calculation. Either, the underdetermined system of equations can be solved numerically by using, .e.g, the Newton-Raphson method [38]. The downside with using a numerical solution of the the inverse kinematics is that the solution is not exact. This would imply that the feedbacked actuator lengths are approximations of the true actuator lengths, implying that the closed-loop design is utilising inexact data for control of the GSP. Even minor approximations poses as a severe problem since the sensors in the linear motors have got an accuracy of 0.1 micrometers. Thus, it is uncertain that such a closed-loop control design performs better than the open-loop approach that is being utilised. Furthermore, the inverse kinematics calculation is a computationally demanding operation if the underdetermined system of equations is solved numerically. This might severely limp the GSP's ability to operate in real-time, since the VN8912 module must perform its calculation of set-points in five milliseconds. Therefore, a numerical solution of the inverse kinematics is not implemented.

In order to ease the computational time complexity that is required by utilising a numerical solution, the inverse kinematics could be solved offline for all possible combinations of actuator lengths. Then, when the VN8912 module receives feedback data consisting of the actuators' lengths, the spatial coordinates corresponding to each length are retrieved from a look-up table. However, this approach implies that a numerical solution of the inverse kinematics for all combinations of actuator lengths must be solved beforehand any operation of the GSP is conceivable. Since the amount of such combinations is enormous, and the fact that the numerical solution still only provides approximations of the true actuator lengths, an offline calculation of the inverse kinematics is deemed infeasible as well.

If the dynamics of the GSP are known, a closed-loop controller that relies on inverse kinematics would most likely perform better compared to the currently implemented open-loop control design. However, since the inverse kinematics adds more difficulties than advantages to the test rig in its current state, the VN8912 module is controlling the GSP with an open-loop control design.

## 5.2 Remarks on the Control System

As is stated in Section 2.2.3, the linear motors have got embedded PID regulators. In the open-loop control design of the GSP, those PID regulators are handling the actual control of the upper platform. Control-wise, the VN8912 module merely serves as a means to produce the set-points for the PID regulators. Therefore, as long as the set-points are correctly calculated and transmitted to the drivers, the control performance of the GSP is determined by the linear motors ability to control their respective rotor. This poses as a problem because it is only the PID regulators' tuning parameters, as well as their embedded noise filter's deadband, that are modifiable. The interior logic of the PID regulators is unknown as it is considered a trade secret.

Therefore, the only way to change the performance of the PID regulators – and thus improve the control of the GSP – is through tuning of the regulator parameters. But

without knowing the exact details of how both the PID regulators' themselves are constructed, as well as how the noise filter is implemented, the best possible control of the GSP is hard to obtain. Hence, the PID regulators are tuned in a trial-and-error fashion, where the tuning that allowed for the best imitation capability of the GSP is selected.

There are two major flaws with the design of the GSP's control system. The first downside is that the dynamics are unknown. Secondly, since the GSP is supposed to mimic chassis motion in real-time, several control designs that are common in robotic motion systems are infeasible. Such techniques are mostly utilising a sliding control window, where both the current and future control inputs are used to calculate the GSP's motion. But, since the GSP operates in real-time, there is an impracticality with obtaining future control inputs. Only the current reference data from an automobile's chassis motion is feasible for use by the GSP's control system in order to guarantee as close to real-time imitation as possible.

However, another overhaul of the test rig that would allow for a better imitating performance of the GSP is to not use a distributed control system. If all the logic of the test rig is located in the VN8912 module, or in some other singleton device, the time it takes to transmit PDOs through the CAN bus is neglected. This implies that the majority of the delay that occurs during the GSP's real-time imitation of chassis motion is removed. With a less significant delay, a better imitation capability is achieved.

Relocating the logic of all the components of the test rig into the VN8912 module is not physically feasible. This is mostly due to the secrecy of the implementation of the linear motors' PID regulators. Instead, two other measures are undertaken in order to optimise the performance of the GSP's imitation capability. Firstly, in order to lower the time it takes to transmit PDOs via the CAN bus, it is being kept as short as possible; the CAN bus is around two metres long. A shorter CAN bus allows for faster transmissions through it. Additionally, to reduce the impact of the CAN bus even further, its baud rate is set as high as possible. With a transmission rate of 1 Megabaud, one transmission of PDOs between the VN8912 module and an actuator driver is possible every fifth millisecond. (This transmission rate is further addressed in Section 2.3.)

The motion of the GSP is currently controlled in a SISO fashion. Each actuator's driver receives set-points which are independent of the set-points that are transmitted to the other drivers. If, instead, inverse kinematics are feasible to utilise, a MIMO control approach can be implemented. This MIMO control design would alter the lengths of the actuators based on the spatial coordinates that correspond to the actual actuator lengths, and it does so in a non-independent manner. This allows for a control design that incorporates the behaviour of all actuators into the calculation of an actuator's set-points. Thus, the imitation capability of the GSP will, most likely, be more precise with a MIMO control design. But, as inverse kinematics are too difficult to implement – as described in Section 5.1 – a MIMO control design is not utilised.

### 5.3 Performance of the Test Rig

The vibrations of the GSP, and especially of the headlights at the very end of the attached bar, that arise during operation are a serious concern when it comes to performing analyses of the adaptive headlights dynamics. However, without knowledge of the GSP's dynamics, nor the ability to change the behaviour of the linear motors' PID regulators in a significant manner, there exists no easy way of dampening the vibrations. The lack of knowledge dynamics-wise is not the only factor as to why vibrations occur. The dynamics of the bar and the attachment units that are used to mount the headlights on the GSP are unknown as well. In addition, the fastening units of the two headlights are constructed asynchronously to one another, implying that they give rise to different types of vibrations. Furthermore, the bar and its fastening units does not fit perfectly together, which might lead to resonances in the test rig.

Moreover, the floor upon which the GSP is placed is not smooth. This allows for the GSP's lower platform to glide around slightly during operation, especially considering that it is not fixed to the floor. This is due to the fact that the room in which the GSP should operate is not fully built as of yet. When the lower platform glides over small cavities in the floor, the inclination of the GSP is altered. Thus, the conditions of the GSP's dynamics changes over time. Once the lower platform is cinched to the floor and the GSP becomes stationary, such gliding motion of the GSP during operation will – most likely – disappear.

On the mechanical side, the joints which are used as links between the actuators and the two platforms are sub-optimal. This is due to the fact that some joints have less friction than others, allowing the actuators to perform revolutions with a somewhat different inertia. Furthermore, the joints themselves are not assembled exactly in the same fashion on each actuator; some are a little more rotated or compressed compared to joints on the other actuators. Additionally, the actual rotors of the linear motors are subject to different friction when they slide into their respective stator, and the generated magnetic fields that affects the rotors inside of the actuators' extensions are not exactly the same across all actuators.

With such a plethora of potential error sources, the control system of the GSP is still functioning at a high standard. The platform itself is able to mimic chassis motion with extreme precision. The problem arises when the vibrations are taken into account. Those prevents the real-time analyses of adaptive headlights dynamics that Volvo Cars want to perform. However, by operating the GSP at a lower pace than real-time, the vibrations are much less significant. Therefore, in order for Volvo Cars to utilise the GSP for their intended purpose, the imitation of chassis motion must occur in a slow fashion.

To achieve better imitation of chassis motion of the GSP, its dynamics must be known. Therefore, one idea is to utilise system identification and a suitable design of experiments that allows for the construction of a model of the GSP's dynamics. Without knowledge of the dynamics, or the ability to perform considerable changes to the PID regulators, the GSP can not increase its current performance when it comes to imitating automobile chassis motion.

The test rig as a whole proves that the GSP is suitable for real-time imitation of

chassis motion. However, to remove the vibrations, and thus allow real-time tests to be performed, a better system design is required. One idea is to not use a distributed control system. It is also a requirement to have full knowledge of all the components of the system in order to design the most optimal GSP when it comes to imitating chassis motion.

## 5.4 Environmental Impact

The GSP can be used for analyses of adaptive headlights dynamics, but not in real-time. However, this still has a huge impact on the environment in the sense that Volvo Cars can reduce the number of times that real automobiles are operated for the purpose of testing their headlights. Naturally, reduced operation of real automobiles is convenient for the environment, as less exhaust fumes and emissions are generated. So, in the sense of impacting the environment the GSP is performing with grandeur.

## 5.5 Future Improvements of the Test Rig

The first thing that comes to mind regarding improvements of the test rig is to get full knowledge of all the parts of the system. To do this, the actuator drivers and the linear motors have to be replaced with corresponding devices where the PID regulator logic is not classified. If this is deemed infeasible, then complete insight of the logic in the current drivers and motors are required. Furthermore, by utilisation of a proper system identification experiment, the dynamics of the GSP can be obtained. With knowledge of the entire system, a better – imitation-wise – control design can be implemented.

Another idea is to attach sensors to both ends of the mounted bar, and have those sensors measure the vibrations. This sensor data should then be feedbacked to the VN8912 module, where it is utilised in an outer control-loop to calculate better fitted set-points for the PID regulators. The purpose of this outer control-loop would be to dampen the effect of the vibrations on the headlights, and thus potentially allow for the adaptive headlights dynamics simulation to be performed in real-time.

Additionally, in order to CE mark the entire test rig, a safe region of operation around the GSP must be present, according to the machinery directive. This region could, as a suggestion, consist of the safety mat in combination with fences. In such a scenario, the safety mat prohibits that someone enters the safety region without causing an emergency stop of the GSP, and the fences themselves completely prohibits anyone from standing too close to the GSP during operation. If the fences are to be part of the safety region's borders they must be bolted, or in some other way fixed, to the floor. The construction of a safety region is a recommended action to perform in order to be able to utilise the GSP for the analyses of adaptive headlights dynamics that Volvo Cars want to perform. The reason for this is that research and development of new concepts for automobile headlights is a commercial action, since the produced headlights are being sold in actual automobiles. Thus, a CE marking of the test rig allows for Volvo Cars to utilise the GSP as a stage in

## 5. Conclusion

---

their manufacturing chain of automobiles.



# A

# CE Marking of the Test Rig

## Hazard Analysis

FOR THE PURPOSE OF CE MARKING A  
GOUGH-STEWART PLATFORM

VOLVO CAR CORPORATION  
EXTERIOR LIGHTNING – 93561

A CHALMERS UNIVERSITY OF TECHNOLOGY  
MASTER'S THESIS DOCUMENTATION

*Authors:*  
Calle ARKEVALL  
Christoffer SANDLUND

June 7, 2017

# Fundamental Health and Security Requirements

This chapter outlines the fundamental requirements and measures of the Gough-Stewart platform (GSP) in regards to operator health and device security. The GSP is a six-legged robot - a hexapod - which consists of two platforms; one fix lower platform, and one movable upper ditto. The upper platform is able to move with six degrees of freedom, and its motion is dependent on how much each leg - actuator - is extended in length. The moving part of one actuator consists of a linear motor. This linear motor has got a corresponding driver that is used to govern its motion. The lower platform resides on the floor, and the upper platform together with the six actuators are attached to it.

The drivers of the linear motors are located within an electrical enclosure. There is also an emergency stop button on the door of this enclosure. Once the emergency stop button is pressed, motion of the GSP is stopped. Furthermore, a safety mat that is lying in front of the GSP is also cabled to the enclosure. If the safety mat is walked upon, motion of the GSP is stopped as well.

There is a horisontal bar attached on top of the GSP's upper platform. This bar is extending outwards from the GSP by half a metre at two opposing sides. At both ends of this bar, a Volvo Car Corporation automobile headlight is attached. The purpose of such an attachment on the GSP is for those headlights to mimic the behaviour of headlights corresponding to real driven automobiles, or automobiles that are being operated in a simulation software. To achieve this, the GSP is used to imitate automobile chassis motion.

In this report, a hazard analysis of the GSP is undertaken in order to determine if it is possible to CE mark the entire system. Firstly, a discussion with respect to the machinery directive is given regarding the use and operation of the GSP and its embedded components. Afterwards, a conclusion stating whether the system can be CE marked or not is given. Furthermore, the actual results of the hazard analysis are seen in the appendix of this documentation.

## THE GSP IN GENERALISED TERMS

This chapter covers, in a general fashion, the area of use of the GSP, as well as potential hazards that can arise during its operation.

### Intended Use of the GSP

The intended use of the GSP is for it to mimic chassis motion in real-time. This is done by feeding references containing sensor data of chassis motion to a network hardware interface – a VN8912 module. This VN8912 module is communicating with the drivers of each actuator in such a way that it is able to send motion commands to them. The VN8912 module can also order the drivers to stop the current motion of their linear motor. The emergency stop button as well as the safety mat are wired to the VN8912 module so as to indicate if an emergency stop is undertaken.

During intended use, the GSP is standing fix in its position. The motion it can perform is limited to a heaving motion, as well as the angular motions roll and pitch. Thus, the spatial position of the GSP in the room is not altered during operation.

### Hazard Sources

Potential hazards that can arise during operation of the GSP are mostly limited to scenarios where a person is being too close to the GSP during operation. Another cause for hazards is if the components in the electrical enclosure are subject to an electrical fault, which then might cause the operator to receive a shock.

Furthermore, if the GSP is to be moved, the person that moves it should do so by holding at the bottom of the actuators. This is due to the fact that the upper platform of the GSP is movable and does not allow for a solid grip when applying the force that is necessary to re-position the GSP.

### Potentially Hazardous Areas

Since the GSP is a robot that moves with an attached load, there comes a risk with standing too close to it during operation. Hence, a hazardous area during operation of the GSP is characterised by a safety region. Inside this region, a person might get hurt.

Additionally, one should not remain close to, or within, the electrical enclosure when the GSP system is supplied with power.

## **Operator Exposedness**

During intended use of the GSP, no person is exposed to any danger. The operator of the GSP is located in front of a computer at a desktop, which is placed on a safe distance from the GSP's region of operation.

Furthermore, a safety region should also be built around the GSP. The purpose of this region is to delimit people from standing too close to the GSP during operation. The safety region consists of one safety mat that is designed by the corporation ABB, which lies in front of the GSP, as well as fences. If a person applies pressure on the safety mat – e.g. walks upon it – the GSP is immediately freezing its current motion. Thus, no one can reach the GSP from the front without stopping its motion. In addition, the fences should be placed along the sides of the GSP, so as to guard against trespassing from any direction other than the frontal direction.

## **Reasonably Foreseeable Misuse**

If a person is deliberately standing too close to the GSP during operation, the person's safety can not be guaranteed. This is due to two reasons. Firstly, the GSP is performing motion with a large attachment unit, which carries the two headlights of an automobile. This load can potentially damage a person that is struck by it. Secondly, the actuators' of the GSP are continuously altering their lengths and positions, which implies that a person might get stuck between actuators if the boundaries of the safety region are violated.

Furthermore, if a person is trying to move the GSP by dragging it in the upper platform, the base platform of the GSP will remain stationary. Instead, the actuators and the upper platform will change their position, and the person who tries to move the GSP will be subject to an unexpectedly small resistance force. This might lead to the person falling, or even worse, getting stuck between the GSP's actuators.

## **Principles for Integration of Security**

This section covers the assembly of the GSP, and it highlights what measures that have been undertaken during manufacturing to reduce the hazardous impact that operation of the GSP has on humans in its vicinity.

### **Safe Construction of the GSP**

The GSP is constructed in such a way that it can cause as little harm as possible to humans that are too close to it. The linear motors are set to use as little force as is necessary in order to be able to move the upper platform and its attached

headlights. Apart from this, the upper platform, as well as the attached bar, are fabricated in aluminum in order to make the moving parts of the GSP as light as possible, in case that someone is hit by one of the GSP's components during operation.

The safety region should be guarding the GSP from outer influence, implying that a specific component of the GSP cannot be altered or affected by someone during operation. In addition, updates in the software of the GSP is only possible when no operation is conducted.

### **Hazards of Misuse**

There are three major sources of hazards should the GSP be misused. The first one is the fact that the GSP or its attached headlights might collide with a person during operation, if the person resides within the safety region. Secondly, if a person is trying to move the GSP by dragging it in the upper platform or too far up on any of the actuators, the lower platform will remain stationary. This results in a motion of the upper platform, but no actual movement of the entire GSP. Thirdly, if a person resides within the enclosure, there is a possibility that this person will be subject to electrical shocks.

## **CONTROL SYSTEM**

This chapter gives more in-depth details of how the control system of the GSP is functioning. The intent with this chapter is to give the reader knowledge of how the GSP is going to behave during different operating scenarios.

### **Safety and Reliability of the Control System**

The GSP is governed by a control system that only allows it to perform motion which is similar to that of an automobile's chassis during operation. The exception to this is if the emergency stop button is pressed; then, all motion is instead prohibited, and the GSP stops its current imitation of chassis motion. However, reference data that contains a too large chassis motion is discarded. The GSP is thus only performing motions that are small enough. Such small motions hardly make out any danger for a human which is standing close to the GSP.

The VN8912 module functions as the central logic unit of the GSP system. All the actuator drivers are connected to the VN8912 module, and their linear motors receive motion commands from the VN8912 module. The actual control system is implemented in the embedded software of the VN8912 module; CANoe. The purpose of the VN8912 module is to relay information to and from

the actuator drivers. With motion commands from the VN8912 module, the actuator drivers are utilising the linear motors to position the actuators according to sensor data from the VN8912 module.

If an error occurs, an error signal is transmitted to the VN8912 module. This error can originate either from the actuator drivers or from any of the emergency stop measures being activated. As long as an emergency is active the VN8912 module is not transmitting any new motion commands to the actuator drivers. Thus, motion of the GSP is not permitted.

In addition, the logic of the VN8912 module is implemented such that only one type of motion can be performed at a time. If the operator tries to execute several motion commands at once, only the first is performed by the GSP.

## **Emergency Stop Measures**

If an emergency stop measure is being activated, the GSP immediately discards any reference data, stops its current motion, and freezes in place. Furthermore, the VN8912 module is not transmitting any additional motion commands to the actuator drivers.

The emergency stop button on the door of the electrical enclosure is clearly visible from afar. Furthermore, the emergency stop button is easily accessible, and it requires no physical effort to press it. Once pressed, the motion of the GSP is immediately stopped, and the GSP freezes in its current position. Despite at what time or stage during operation that an emergency stop measure is activated, motion of the GSP is stopped.

An active emergency is reset via a two-step verification process. First, the actual emergency stop button has to be deactivated. Afterwards, a second button on the electrical enclosure must be pressed, which indicates that a reset of the emergency state is performed. When an emergency is reset, no motion of the GSP is performed. Instead, the GSP enters a state called Operation Enabled, which merely allows for new motion commands to be received by the GSP.

The safety region guarantees that no human is able to stand too close to the GSP during operation. However, there should be an emergency stop button located inside the safety region as well. As of such, a person who resides inside the safety region can either press this emergency button, or step on the safety mat, to stop the GSP's current motion.

## **Visibility from the Operator Perspective**

In order for the GSP to perform motion an operator must be located in front of a computer outside of the safety region. From this position, the operator have

full visual supervision of the GSP and the whole safety region. Thus, no human can reside too close to the GSP without the operators knowledge.

## **PROTECTION AGAINST MECHANICAL HAZARDS**

As the GSP consists of a large mechanical part, it is required to also analyse the system on a mechanical level. The components which might give rise to hazards are discussed in this chapter.

### **Moving Transmission Parts**

The GSP's actuators are the only components of the system that are able to perform motion. However, when motion of the actuators occur, the position of the GSP's upper platform is altered. Since the only movable components are part of the GSP itself, the safety region around it sufficiently shuts out humans from interfering with any potentially hazardous hardware during operation. (This holds only if the interior of the electrical enclosure is inaccessible during operation as well.)

### **Adding Permanent Protection to Parts of the GSP**

Attaching permanent covers, or equivalent cases for protection, on (parts of) the GSP, will cause the performance of the system to decrease. This is due to the fact that additional sheets of protection on, e.g., the actuators causes them to be larger, which results in a limitation of their area of operation. The VN8912 module is making sure that no actuators are colliding with one another during operation. Thicker actuators allow for a lesser amount of feasible motions.

Furthermore, adding extra hardware cover to the GSP also adds weight. This will limp the linear motors' ability to move the actuators, as the load has increased. Therefore, this kind of protection is not used.

### **Breakage of Mechanical Components**

One can never guarantee that a robot, or any mechanical device for that matter, will not break. Thus, there is a risk that components or parts of the GSP will break with time, causing unforeseen faults in the intended behaviour of the GSP. However, the safety region is still present, which discloses any human interaction from the GSP during operation.

## MAINTENANCE

To start the GSP, power must first be applied to the system. This is done by toggling a main switch, which is located on the side of the electrical enclosure. By enabling this switch, power is fed to all electronic devices inside the enclosure. Thus, the drivers and the VN8912 module are supplied with power, which allow them to start operating the GSP.

When the main switch is deactivated, no power is fed to the system. There is also possible to lock the main switch in an OFF state, which enables a safe maintenance environment for any operator that want to conduct hardware changes of the system.



# Missing Criteria in order to CE-Mark the GSP System

An adequate safety region around the GSP must be constructed. This safety region should be designed in such a way that no person can enter this region without stepping on the safety mat, and thus freezing any eventual ongoing motion of the GSP. This includes that the safety region, which is most likely going to consist of the safety mat and additional fences, must be robust against severe attempts to trespass too close to the GSP. Thus, it is recommended that the fences are bolted into the floor.

However, if fences are not an option, additional safety mats, or a laser, can be utilised. If a laser is chosen upon, it should be pointed at a reflector. If a person crosses the path of the laser, e.g. if the laser is no longer shining its light on the reflector, motion of the GSP should stop. As long as the GSP is protected against approaches from all directions the safety region fulfils its purpose. A design suggestion of a safety region is displayed in Figure 1, where both the safety mat and a laser are utilised in tandem with fences to ward against trespassers. In the figure, *EM* is an abbreviation for Emergency, and the two visible buttons displays the location of the Emergency Stop buttons of the GSP system.

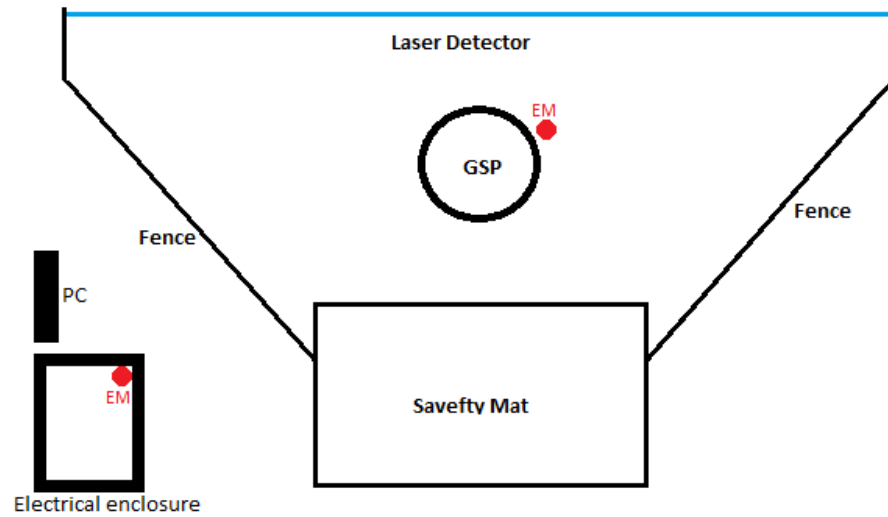


Figure 1: A suggestion for a design of a safety region around the GSP.

Besides the lack of a safety region, it is the writers impression that the GSP and its adjacent units fulfills the machinery directive in the sense of operator safety and robotic security aspects. Thus, with the design of a safety region that keeps all trespassers at a minimum distance from the GSP, the entire system can be CE marked.





Organisation VCC R&D - Exterior lighting	Document type <b>List of Hazards</b>		
Document name List of hazards Hexapod	Version 1	Date 2017-06-07	Page 1 (4)
Issuer (Dept. name, CDS-Id) 93651, Victor Wareborn, VWAREBOR	Reg. No.	Security class Add security class.	

List of hazards Hexapod

Safety analysis relating to design and construction of machinery

Machine/Equipment No.		Machine/Equipment Type	Test rig
Drawing/Layout No./Issue	Pärm	Produced part	-
Customer	PSS 170 Exterior Lighting	Manufacturer	International Heating Products
Project responsible	Victor Wareborn, VWAREBOR	Design responsible	Victor Wareborn, VWAREBOR
Total responsibility for complete machine: (CE marking to be made by customer or manufacturer)	Jan Gynning, JGYNNING	Participants	VWAREBOR, MKLOMP, HJARSS13, CARKEVA1, CSANDLUN
Assessment made during phase: (Planning/design/manufacturing)	Manufacturing	Ref. documents (e.g. previously made assessments)	-
Process in the machine/equipment	Motion of front light testing fixture		
Energy types and media present in the machine/equipment: (e.g. electricity, pneumatics, hydraulics, springs under suspension, gases, heat)			Electricity, magnetism
Comments/basic conditions: (e.g. operators, environmental influences, noise, process outlets, ergonomic problems)			

B  
Hazard Analysis



Organisation VCC R&D - Exterior lighting		Document type <b>List of Hazards</b>	
Document name List of hazards Hexapod	Version 1	Date 2017-06-07	Page 2 (4)
Issuer (Dept., name, CDS-id) 93651, Victor Wareborn, VWAREBOR	Reg. No.	Security class Add security class.	

No.	Hazard	Applicable	Location	Measure	Standards applied	Hazard under control	Residual hazard
<b>1</b>	<b>Mechanical hazards resulting in:</b>						
1.1	Crushing of limbs by moving parts of machine	Yes, Y	Attached Headlight, AH	Guard rails and safety mat / closed cabinet		Y	
1.2	Shearing of limbs by moving parts of machine	Y	AH	See 1.1		Y	
1.3	Cutting or severing by static or moving parts of machine	No, N					
1.4	Entanglement of limbs or hair in static or moving parts of machine	Yes, Y	Manipulator, M	See 1.1	See safety concept	Y	
1.5	Pulling or trapping limbs by moving parts of the machine	Y	M	See 1.1		Y	
1.6	Injury by impact with parts of the machine	Y	M	See 1.1		Y	
1.7	Injury by stabbing or puncturing by parts of machine	N					
1.8	Injury by friction or abrasive action by machine parts	N					
1.9	Injury caused by possible action of high pressure	Y	M	See 1.1		Y	
1.10	Injury from release of potential energy	N					
1.11	Injury from release of kinetic energy	Y	M	See 1.1		Y	
1.12	Injury resulting from loss of stability of the machine or machine parts	Y	M	See 1.1		Y	
<b>2</b>	<b>Electrical hazards resulting in:</b>						
2.1	Injury from contact with live electrical conductors	Y	Electrical cabinet, EC	Closed cabinet/ standard electrical safety measures	See electrical documentation	Y	
2.2	Injury from direct contact with parts which have become live under faulty conditions	Y	EC	See 2.1	See 2.1	Y	
2.3	Injury from electrostatic phenomena	N					
2.4	Injury from thermal radiation or molten practices as a result of short circuits or overloads, etc.	N					
2.5	Injury from external influences on electrical equipment	Y	EC	See 2.1	See 2.1	Y	
<b>3</b>	<b>Thermal hazards resulting in:</b>						
3.1	Burns, scalds and other injuries from contact with or heat radiation from objects or materials with extreme temperatures, or by flames or explosions	N					
3.2	Damage to health by hot or cold working environment	N					



Organisation VCC R&D - Exterior lighting	Document type <b>List of Hazards</b>		
Document name List of hazards Hexapod	Version 1	Date 2017-06-07	Page 3 (4)
Issuer (Dept., name, CDS-id) 93651, Victor Wareborn, VWAREBOR	Reg. No.	Security class Add security class.	

4	<b>Hazards caused by noise resulting in:</b>					
4.1	Hearing loss (deafness), other physiological disorders (e.g. loss of balance or awareness)	N				
4.2	Interference with spoken communication, acoustic signals, etc.	N				
5	<b>Hazard caused by vibration resulting in:</b>					
5.1	Use of hand-held machines resulting in a variety of neurological and vascular disorders	N				
5.2	Whole-body vibration, particularly when combined with poor posture	N				
6	<b>Hazards caused by radiation:</b>					
6.1	Low frequency, radio-frequency radiation; microwaves	N				
6.2	Infrared, visible and ultraviolet radiation	N				
6.3	X-and gamma rays	N				
6.4	Alpha, beta rays, electron or ion beams, neutrons	N				
6.5	Lasers	N				
7	<b>Hazards caused by materials and substances (and their constituent elements) processed or used by machinery</b>					
7.1	Hazards from contact with or inhalation of harmful fluids, gases, mists, fumes and dusts	N				
7.2	Fire or explosion hazard	N				
7.3	Biological or microbiological (viral or bacterial) hazards	N				
8	<b>Hazards caused by neglecting ergonomic principles in machinery design</b>					
8.1	Unhealthy postures or excessive effort	N				
8.2	Inadequate consideration of hand-arm or foot-leg anatomy	N				
8.3	Neglected use of personal protection equipment	N				
8.4	Inadequate local lighting	N				
8.5	Mental overload and underload, stress	N				
8.6	Human error, human behavior	Y	M / EC	See 1.1	See 1.4	Y
8.7	Inadequate design, location or identification of manual controls	N				
8.8	Inadequate design or location of visual display units	N				
9	<b>Combination of hazards</b>					
9.1		Y	M / EC	See 1.1	See 1.4	Y
10	<b>Unexpected start-up, unexpected overrun/overspeed (or any similar malfunction) from:</b>					
10.1	Failure/disorder of the control system	Y	M		See 1.4	Y



Organisation VCC R&D - Exterior lighting	Document type <b>List of Hazards</b>		
Document name List of hazards Hexapod	Version 1	Date 2017-06-07	Page 4 (4)
Issuer (Dept., name, CDS-id) 93651, Victor Wareborn, VWAREBOR	Reg. No.	Security class Add security class.	

10.2	Restoration of energy supply after and interruption	Y		M		See 1.4	Y	
10.3	External influences on electrical equipment	N						
10.4	Other external influences (gravity, wind, etc.)	N						
10.5	Software errors	Y		M		See 1.4	Y	
10.6	Errors made by the operator (due to incompatibility between machinery and human characteristics and abilities, see (8.6))	Y		M		See 1.4	Y	
11	<b>Impossibility of stopping the machine in the best possible conditions</b>	N						
12	Variations in tool rotational speed	N						
13	Failure of power supply	Y		M	See 1.1	See 1.4	Y	
14	Failure of control circuit	Y		M	See 1.1	See 1.4	Y	
15	Errors of fitting	N						
16	Break-down during operation	N						
17	Falling or rejected objects or fluids	N						
18	Loss of stability / overturning of machinery	N						
19	Person slip, trip and fall (related to machinery)	N						





# Glossary

CAPL	Communication Access Programming Language.
COB-ID	Communication OBject IDentifier.
DIO	Digital Input-Output.
ECU	Electronic Control Unit.
FSA	Finite State Automaton.
GSP	Gough-Stewart platform.
GUI	Graphical User Interface.
MIMO	Multiple-Input and Multiple-Output.
OD	Object Dictionary.
OSI	Open Systems Interconnection.
PDO	Process Data Object.
RPDO	Receive Process Data Object.
RTR	Remote Transmission Request.
SISO	Single-Input and Single-Output.
TPDO	Transmit Process Data Object.



# Bibliography

- [1] Volvo Cars AB. *This is Volvo Cars*. URL: <http://www.volvocars.com/us/about/our-company/this-is-volvo-cars> (visited on June 5, 2017).
- [2] Volvo Car Group. *Volvo Cars expands production in China and unveils new China strategy*. URL: <https://www.media.volvocars.com/global/en-gb/media/pressreleases/199915/volvo-cars-expands-production-in-china-and-unveils-new-china-strategy> (visited on June 5, 2017).
- [3] Michael W. Wilcox. *Recent Decisions: Products Liability: Strict Liability in Tort Applied to Both Automobile Manufacturer and Retailer*. Marquette University Law School, 1964.
- [4] Forbes. *New Products: More Costly and More Important*. URL: <http://www.forbes.com/sites/loracecere/2013/12/11/new-products-more-costly-and-more-important/#22b5b9fc797f> (visited on June 5, 2017).
- [5] RobotShop. *Hexapod Robots*. URL: <http://www.robotshop.com/en/hexapod-development-platforms.html> (visited on June 5, 2017).
- [6] D. Stewart. *A Platform with Six Degrees of Freedom*. Institution of Mechanical Engineers (UK), 1965-1966.
- [7] Bhaskar Dasgupta and T.S. Mruthyunjaya. *The Stewart Platform Manipulator: a Review*. Department of Mechanical Engineering, Indian Institute of Technology, Kanpur and Department of Mechanical Engineering, Indian Institute of Science, Bangalore, January 13, 1998.
- [8] C.H. Zheng Y.F. Zhang G.D. Chen Y.X. Su B.Y. Duan and J.W. Mi. *Disturbance-rejection high-precision motion control of a Stewart platform*. IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 12, NO. 3, May 2004.
- [9] Z.-L. Zhou C.C. Nguyen S.S. Antrazi and C.E. Campbell. *Experimental study of motion control and trajectory planning for a Stewart Platform robot manipulator*. proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, April 1991.
- [10] Layton C. Hale. *Principles and techniques for designing precision machines*. Lawrence Livermore National Laboratory, University of California, Livermore, California, February 1999.
- [11] Leonhard Euler. *Novi Commentarii academiae scientiarum Petropolitanae* 20. pp. 189-207 (E478), 1776.

- [12] J von der Heide and G.F. Papst. *Linear motor*. URL: <https://www.google.com/patents/US4151447> (visited on June 5, 2017).
- [13] Hydro One. *ELECTRIC MOTORS Energy Efficiency Reference Guide*. URL: [http://www.hydroone.com/MyBusiness/SaveEnergy/Documents/Electric\\_Motors\\_Reference\\_Guide.pdf](http://www.hydroone.com/MyBusiness/SaveEnergy/Documents/Electric_Motors_Reference_Guide.pdf) (visited on June 5, 2017).
- [14] LinMot. *Global manufacturer of high quality tubular style linear motors and linear motor systems*. URL: <http://www.linmot.com/> (visited on June 5, 2017).
- [15] LinMot. *Data sheet 0185-1003: Linear motors P01-37x240*. URL: <http://shop.linmot.com/E/ag1000.37.241/linear-motors-p01-37/stators-ps01-37x240/ps01-37x240f-c.htm> (visited on June 5, 2017).
- [16] Vector. *Capable partner in the development of automobile electronics*. URL: <http://www.vector.com/> (visited on June 5, 2017).
- [17] Vector Informatik GmbH. *VN8900 - Modular FlexRay/CAN FD/LIN/J1708/K-Line Network Interface with up to 8 channels*. URL: [https://vector.com/vi\\_vn8900\\_en.html](https://vector.com/vi_vn8900_en.html) (visited on June 5, 2017).
- [18] International Organization for Standardization. *ISO 11898-1:2015 Road vehicles – Controller area network (CAN) – Part 1*. URL: <https://www.iso.org/standard/63648.html> (visited on June 5, 2017).
- [19] International Organization for Standardization. *ISO 17987:2016 Road vehicles – Local Interconnect Network (LIN) – Part 1-7*. URL: <https://www.iso.org/search/x/query/17987> (visited on June 5, 2017).
- [20] International Organization for Standardization. *ISO 17458:2013 Road vehicles – FlexRay communications system – Part 1-5*. URL: <https://www.iso.org/search/x/query/17458> (visited on June 5, 2017).
- [21] National Instruments. *Controller Area Network (CAN) Overview*. URL: <http://www.ni.com/white-paper/2732/en/> (visited on June 5, 2017).
- [22] Bell System Data Communications. *Data Set 103F Interface Specification*. URL: [http://bitsavers.trailing-edge.com/pdf/westernElectric/telephony/modems/103F\\_Interface\\_Specification\\_May64.pdf](http://bitsavers.trailing-edge.com/pdf/westernElectric/telephony/modems/103F_Interface_Specification_May64.pdf) (visited on June 5, 2017).
- [23] CANopen Solutions. *Layer Setting Services (LSS)*. URL: [http://www.canopensolutions.com/english/about\\_canopen/lss.shtml](http://www.canopensolutions.com/english/about_canopen/lss.shtml) (visited on June 5, 2017).
- [24] CAN in Automation (CiA). *CANopen – The standardized embedded network*. URL: <https://www.can-cia.org/canopen/> (visited on June 5, 2017).
- [25] International Organization for Standardization. *ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. URL: <https://www.iso.org/standard/20269.html> (visited on June 5, 2017).

- 
- [26] CAN in Automation (CiA). *Process data object (PDO)*. URL: <https://www.can-cia.org/can-knowledge/canopen/pdo-protocol/> (visited on June 5, 2017).
- [27] CANopen Solutions. *Object Dictionary (OD) and Electronic Data Sheet (EDS)*. URL: [http://www.canopensolutions.com/english/about\\_canopen/communication.shtml](http://www.canopensolutions.com/english/about_canopen/communication.shtml) (visited on June 5, 2017).
- [28] Vector Informatik GmbH. *ECU Development & Test with CANoe*. URL: [https://vector.com/vi\\_canoe\\_en.html](https://vector.com/vi_canoe_en.html) (visited on June 5, 2017).
- [29] Vector Informatik GmbH. *Programming with CAPL*. Vector Informatik GmbH, First Printing, p. 1, 2004, December 14.
- [30] IPG Automotive. *IPG CarMaker*. URL: <https://ipg-automotive.com/> (visited on June 5, 2017).
- [31] IPG Automotive. *IPG Automotive: An excerpt from our list of clients*. URL: <https://ipg-automotive.com/company/references/#oems> (visited on June 5, 2017).
- [32] LinMot. *LinMot-Talk (Drive-configuration)*. URL: <http://www.linmot.com/download/linmot-talk-drive-configuration/> (visited on June 5, 2017).
- [33] Richard. P. Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. MIT Press, Cambridge, MA. ISBN 978-0-262-16082-7, 1981.
- [34] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators 2nd Ed. pp 21-22*. Springer, 2000.
- [35] Wolfram Research. *WOLFRAM MATHEMATICA The world's definitive system for modern technical computing*. URL: <https://www.wolfram.com/mathematica/> (visited on June 5, 2017).
- [36] MathWorks. *MATLAB The Language of Technical Computing*. URL: [https://se.mathworks.com/products/matlab.html?s\\_tid=hp\\_products\\_matlab](https://se.mathworks.com/products/matlab.html?s_tid=hp_products_matlab) (visited on June 5, 2017).
- [37] Industry-Entrepreneurship The European Commission GROWTH Internal market and SMEs. *Machinery*. URL: [http://ec.europa.eu/growth/sectors/mechanical-engineering/machinery\\_en](http://ec.europa.eu/growth/sectors/mechanical-engineering/machinery_en) (visited on June 5, 2017).
- [38] P. Deuffhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, Vol. 35. Springer, Berlin. ISBN 3-540-21099-7, 2004.