DEVELOPMENT OF A FLEXIBLE TEST PLATFORM UTILIZING GEARBOX SIMULATORS THROUGH PROGRAMMING

Bachelor Degree Project in Automation Engineering
30 ECTS
Spring term 2016

Beñat Unibaso Eguzkitza
Ismael Ismail Dobón

Supervisor: Sunith Bandaru
Examiner: Anna Syberfeldt
Abstract

A gearbox simulator is developed as platform for testing and demonstrating purposes. For that, a rig composed by a mechanical system and electronic equipment for controlling two servomotors is used. The objective of this equipment is to simulate the forces that the gearbox would transmit to the gear lever when the gear change operation is being carried-out. To reach this goal, a program is developed in LabVIEW to command the servomotors, emulating the forces by controlling the output torque and transmitting them to the gear stick as it would be in a real gearbox, taking into account real force-angle curves. Also, a graphical user interface is developed in order to monitor the simulator performance and ease the way the data is chosen and introduced into the software. As seen in the experiment results, the graphs present similarities in shape and magnitude, which is important in regards of feeling; a better performance could be reach suppressing some system constraints.

Keywords: Gearbox, shifter, servomotor, simulator, LabVIEW
Contents

List of Figures IV
List of Tables VI
Abbreviations VII
Acknowledgements VIII

1 Introduction 1
   1.1 Background .................................. 1
   1.2 Goals and Objectives .......................... 2
   1.3 Methodology .................................. 4
   1.4 Organization of the Report ..................... 5

2 Literature Review 6
   2.1 Shifting and Synchronization System .......... 6
   2.2 Equipment Analysis and Specification ......... 7
   2.3 Programming ................................ 10

3 Description of the Setup and Reverse Engineering 12
   3.1 General Overview ................................ 12
   3.2 Mechanical Shifting, Gear Lever and Connection Cables .......... 12
   3.3 Servomotors and Control System ................. 14
   3.4 Electronic Equipment and Signal Wiring ........... 16
   3.5 Adjustment of the Components ................... 20

4 Implementation & Experiments 26
   4.1 Program ..................................... 26
   4.2 Experiments .................................. 41
CONTENTS

5 Conclusions 44
  5.1 Conclusions About the Results and Future Work 44

A Appendix: Equipment technical data 46
  A.1 Servomotor technical data 46
  A.2 National Instruments DAQ devices specifications 47

B Appendix: Experiment results 50

Bibliography 53
## List of Figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Figure Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Shifting movements</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Schematic of the system</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Electric motors classification</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Planetary gears</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Gear stick</td>
<td>13</td>
</tr>
<tr>
<td>3.2</td>
<td>Servomotor model DSM5.32</td>
<td>14</td>
</tr>
<tr>
<td>3.3</td>
<td>M23 connectors</td>
<td>15</td>
</tr>
<tr>
<td>3.4</td>
<td>D-subminiature connectors</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>Electronic components and data flow schematic</td>
<td>18</td>
</tr>
<tr>
<td>3.6</td>
<td>Cémios board</td>
<td>21</td>
</tr>
<tr>
<td>3.7</td>
<td>Shunt jumper</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>Cémios board close-up</td>
<td>22</td>
</tr>
<tr>
<td>3.9</td>
<td>Software drivers</td>
<td>23</td>
</tr>
<tr>
<td>3.10</td>
<td>Inputs/outputs configuration</td>
<td>24</td>
</tr>
<tr>
<td>3.11</td>
<td>Encoder/resolver configuration</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>&quot;Select&quot; graph</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Program Structure</td>
<td>28</td>
</tr>
<tr>
<td>4.3</td>
<td>Initialization function block</td>
<td>28</td>
</tr>
<tr>
<td>4.4</td>
<td>Program help schemes</td>
<td>29</td>
</tr>
<tr>
<td>4.5</td>
<td>Help dialogue</td>
<td>30</td>
</tr>
<tr>
<td>4.6</td>
<td>Data selection dialogue</td>
<td>30</td>
</tr>
<tr>
<td>4.7</td>
<td>Data selection dialogue buttons</td>
<td>31</td>
</tr>
<tr>
<td>4.8</td>
<td>Data acquisition function block</td>
<td>31</td>
</tr>
<tr>
<td>4.9</td>
<td>Data acquisition code</td>
<td>31</td>
</tr>
<tr>
<td>4.10</td>
<td>Read Excel function block</td>
<td>31</td>
</tr>
<tr>
<td>4.11</td>
<td>Read Excel code</td>
<td>32</td>
</tr>
<tr>
<td>4.12</td>
<td>Data processing function block</td>
<td>32</td>
</tr>
<tr>
<td>4.13</td>
<td>Data transformation</td>
<td>32</td>
</tr>
<tr>
<td>4.14</td>
<td>Delete row</td>
<td>33</td>
</tr>
</tbody>
</table>
List of Figures

4.15 Data processing code .................................................. 33
4.16 Plotted data ............................................................... 34
4.17 Monitoring ................................................................. 34
4.18 Continuous Execution function block .............................. 35
4.19 Continuous Execution function block .............................. 36
4.20 Angle Transformation function block .............................. 36
4.21 Shift position block ..................................................... 37
4.22 Data Choosing function block ....................................... 37
4.23 "Shift" Data Selection ................................................. 37
4.24 Data Choosing ............................................................. 38
4.25 Force Selection function block ....................................... 38
4.26 Force Selection Code .................................................... 39
4.27 Force Algorithm function block ..................................... 39
4.28 Force algorithm code .................................................... 40
4.29 Force algorithm code .................................................... 40
4.30 Force sensor positioning .............................................. 41
4.31 Force sensor positioning .............................................. 42
4.32 Select movement comparison curve ............................... 43
4.33 First and second gears comparison curve ....................... 43
A.1 PCI 6220/6221 ............................................................ 47
A.2 PCI 6220 Pinout .......................................................... 48
A.3 PCI 6221 Pinout .......................................................... 49
B.1 Select comparison curve ............................................... 50
B.2 First and second gears comparison curve ....................... 51
B.3 Third and fourth gears comparison curve ....................... 51
B.4 Fifth gear comparison curve ......................................... 52
B.5 R gear comparison curve .............................................. 52
List of Tables

3.1 Real-time system inputs (NI PCI-6221) . . . . . . . . . . . . . . 19
3.2 Real-time system outputs (NI PCI-6221) . . . . . . . . . . . . . 19
3.3 Supervision system inputs (NI PCI-6220) . . . . . . . . . . . . . 20
3.4 Supervision system outputs (NI PCI-6220) . . . . . . . . . . . . 20
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDC</td>
<td>Brushless DC Motors</td>
</tr>
<tr>
<td>DAQ</td>
<td>Data Acquisition</td>
</tr>
<tr>
<td>DTE</td>
<td>Data Terminal Equipment</td>
</tr>
<tr>
<td>GPC</td>
<td>General-Purpose Computer</td>
</tr>
<tr>
<td>GPIB</td>
<td>General Purpose Industrial Bus</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>KA</td>
<td>Kongsberg Automotive</td>
</tr>
<tr>
<td>NI</td>
<td>National Instruments</td>
</tr>
<tr>
<td>NVH</td>
<td>Noise, Vibration, and Harshness</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>PMSM</td>
<td>Permanent Magnet Synchronous Motors</td>
</tr>
<tr>
<td>RTOS</td>
<td>real-time operating system</td>
</tr>
<tr>
<td>VI</td>
<td>Virtual Instrument</td>
</tr>
</tbody>
</table>
Acknowledgements

The realization and the completion of this project has supposed months of hard work and big effort; this is why we want to thank the people that have support us during this period.

First of all, we want to thank our supervisors, Sunith and Albin. Even their responsibilities have been different in some ways –the academic and the professional points of view– we are very glad to have work with both of you. Thank you again.

On the other hand, we would like to express our gratitude to our families. Without their support this would not have been possible.

Last but not least, we have to thank all of friends that have shared with months with us. Help, support, and friendship with the best colleagues that we could have. We desire the best for you.
Chapter 1

Introduction

The aim of this chapter is to introduce the reader to the subject of the present project. The chapter contains the background of the thesis and the main requirements of the company that specified out the project. Finally, the chapter also presents the organization of the project report.

1.1 Background

Kongsberg Automotive is a company that is specialized in vehicle element testing such as seats or shifting systems for worldwide automotive companies. In particular, gearbox and shifting tests are done to obtain a concrete assessment of the system. These tests help to better understand and improve the shifting quality by the creation of optimal resources and improvement of shifter systems.

The reason why Kongsberg Automotive is giving importance to the shift quality, is because it is one of the most important aspects to take into account when maximizing comfort for the driver, which directly impacts in the driving experience with manual transmissions. What the driver feels while using the shifting system is a mixture of different interactions between the shifter and the transmission, arising from the shift and select movements.

According to Deleener et al. (2015), this experience can be explained with four different aspects that define the quality of the movements in the shifting system. Those are shift effort, exactness, Noise, Vibration, and Harshness (NVH), and shift comfort. In the case of the shift effort and [NVH] a deeper understanding of the shifting system is required. However, often the transmission is the only component that is studied, usually focusing on the synchronization system to be optimized (Kim et al.; 2002).
The forces that the synchronization applies on the shift stick can be resolved into two different force directions, namely, shift and select, as seen in the Figure 1.1. These are the reaction forces that the manual transmission will apply on the user’s hand. In summary, each one of the forces is applied in one direction of the gear selection. While the shift is actuated when trying to select a specific gear, the select movement decides which pair of gears will be selected.

The motivation of Kongsberg Automotive is to continue improving the quality of the shifters, making reliability tests, as well as being able to show the progress achieved in demonstrations. But the cost of these gearboxes makes it difficult to dispose off a high variety of them, which leads to a reduced flexibility of the system. The solution is to create a system that can simulate the behaviour of the forces applied by any kind of gearbox, thus saving time and money, and giving the user a chance to feel the difference between different gearboxes.

An overview of the employed setup can be seen in the Figure 1.2. The system will be composed of two sinusoidal synchronous AC servomotors that will transmit the theoretical forces of a real gearbox through the shift and select levers. These servomotors are going to be controlled from an industrial bench composed of two industrial motor controllers and two computers, one with a real-time system installed, and another with a General-Purpose Computer (GPC) working as a supervisory computer.

### 1.2 Goals and Objectives

The aim of this project is to program an already existing servomotor system to simulate the influence of a manual transmission on a shifter, focusing on the forces, mainly friction, that are transmitted to the gear stick from the synchronizer; the intention is to get a real shifting experience for the driver avoiding the use of the current gearboxes but employing a smaller and more versatile system. Additionally, a simulation system would be a
way to improve the reliability testing of gearboxes as they are driven by Kongsberg Automotive because it would ease the change between different models of gearboxes and their transportation and space requirements. The existing system will be provided with a Graphic User Interface (GUI) that will show in real time the information about the system, and let the user interact with the system, easily changing gearbox configurations. These configurations will be easily introduced to the program in a standardized way.

In general, a high level of expertise is required to fully understand those movements and forces that are at work inside the gearbox. In that sense, this project only covers the forces that affect the two movements that a gear stick user experiences, and the gearbox itself is considered as a black box that disrupts the stick movement. The raw data that represents these forces for those movements is provided by the company, and two servomotors controlled by a control software will simulate these forces, permitting an easier way to relocate the test place or even a demonstration station, and allowing to change the gearbox model whenever it is required.

A complete list of the goals of the project is as follows:

1. Control system
   - Analyze the way servomotors can simulate different movements of the gearbox
   - Build a main program that will control the servomotor system in LabVIEW

2. Hardware
   - Determine if the available hardware is valid for the simulator
CHAPTER 1. INTRODUCTION

• Suggest different alternatives to compare the viability

3. Programming

• Build the function models that will represent the movements at the shifter
• Develop a standardized form of data collection
• Build a stable program that will be able to simulate the gearbox behaviour as required by the company. It should be able to accept models of different gearboxes from different manufacturers

4. GUI

• Build a GUI which is user-friendly and flexible for changes and further development
• The program will have the option of choosing between different types of gearboxes depending on the information loaded

5. Documentation

• All the information collected to do the previous work has to be written as documentation for further development, so anyone with engineering knowledge will be able to understand how the equipment works and how the changes should be done, also in the programming part covered in this thesis

1.3 Methodology

The methodology adopted in this project can be divided into different stages. There are some distinct steps depending on the status of the project itself, as the research and evaluation of the hardware are well differentiated from the execution of the project, and these stages have been defined to optimize the working time and to plan all the steps needed for the project fulfilment. These stages will be described in detail in the following chapters.

• System analysis
• Theoretical research about the topic (literature survey)
• Hardware analysis at Kongsberg Automotive
• Hardware testing
CHAPTER 1. INTRODUCTION

- Hardware validation
- Programming
- Program implementation
- Conclusions

1.4 Organization of the Report

Contents

The present report is organized in chapters and sections presented in a logical manner to reach the end goals. These chapters are:

- *Chapter 1: Introduction*, where the background and the goals are described.

- *Chapter 2: Literature review*, where the elements and aspects of the project are analyzed and researched to be presented in a theoretical way.

- *Chapter 3: Methodology and approach*, where the initial work in the system setup is described as well as the preparation of the work for the development stage following the method described.

- *Chapter 4: Development and implementation*, where the programming stage of the project is explained as it is carried out.

- *Chapter 5: Conclusions*, where a description of the work carried out and the final conclusions are provided.
Chapter 2

Literature Review

This chapter describes the theoretical aspects with which the problem is approached. This covers a survey of the literature on the type of hardware and motors used in the system setup, specification of the controller system in charge of the motors, as well as the software and communication type. Also, information regarding the shifter and the synchronizer operation inside the gearbox is presented to introduce the construction of the simulator.

2.1 Shifting and Synchronization System

As mentioned previously, the shifting operation is important in the driving experience not only due to the comfort aspect, but also in regards to the fuel consumption of the vehicle (Bo et al.; 2015). As mentioned by Kim et al. (2004) “the shift feeling has been evaluated traditionally in a subjective manner”, taking into account factors as easiness, clash, harness, etc. The shift feeling can be considered as a result of the interaction between the shift stick, the linkage, the synchronizer and the drivetrain. Regarding to the complexity of this setup, which is composed by multiple elements of different nature and behaviour, it is difficult to evaluate the feeling in an objective and quantitative manner. Due to this reason, complex dynamic models are required to calculate the parameters that indicate whether the shifting operation is comfortable or not (Kim et al.; 2004).

Synchronization forces are one of the main features that have to be simulated in order to achieve the real feeling of the gear stick through the select and shift movements. These movements appear in the gear stick as vertical and horizontal movements, and are transmitted as two different movements by the links through the shifting system, as established by Lechner et al. (1999). External and internal linkages and the driveline are also impor-
tant for getting an accurate model of the forces transmitted to the driver’s hand (Kunal et al. 2010). According to Bencker et al. (2005), the synchronization sequence can be divided into five phases that are reflected in the gearshift effort profile at the gear lever. This gear shift profile is what the driver feels at the gear stick.

For this purpose, different type of emulation systems can be developed using electric motors. As explained before, from the driver point of view, the synchronization system consists of two different types of movement that can be simulated using electric actuators. These actuators are intended to simulate the forces like friction and resistance that the transmission drives from the synchronizer to the shifter and gear stick. Through the movement of two computer-controlled servomotors this goal can be achieved, by providing position information and torque output following a force-curve for a specific gearing operation.

2.2 Equipment Analysis and Specification

A study of the general system is carried out in order to obtain a basic understanding that will help when designing different aspects in regards of the way the system works. Within this study, it will be important to get information about the operation of the synchronizer in the gearboxes and the connexion between it and the shifter.

The existing hardware is assessed to determine its functionality and its limitations; different system optimization studies are also required for discovering other features that the hardware could have. A study will be performed about advantages and disadvantages of servomotors, as well as a study about the type of motors that are specified for industry.

Servomotor Classification and Study

As a wide variety of electric motors are available in the market, the choice of the motor depends on the necessities of the system. Almost all the electric motors have the same basic principles of working. The conversion of electric energy into mechanical energy is done by the interaction between the magnetic fields in the stator and the current inducted in the windings of the motor rotor. Even the typical classification is done between the direct current (DC) and alternating current (AC) motors, they follow the same mode of working and is the power supply what differs from one type to the other (Aydin 2012). AC motors can be divided into single-phase and poly-phase type motors. Even though, the power output of the motor
cannot tell enough information about the motor performance itself, as other
parameters that will be treated later are also important (Gottlieb 1997).

A popular way of classifying motors is according to the type of magnets
inside the motor. They can be of field winding excitation or permanent
magnets, and under this group there are a variety of alternatives. These
permanent magnets can be placed on the surface of the rotor or embedded
in the rotor (Aydin 2012). The nature of the magnetic flux is also a way
of categorization. The two main types are the Permanent Magnet Syn-
chronous Motors (PMSM), working with an AC supply, and the Brushless
DC Motors (BLDC). The type of motor installed in the bench where the
project is executed is the former. Despite sharing a few properties, PMSM
tend to be more rigid than BLDC, giving the setup a reliable advantage
in the structure. Also, PMSM are preferred if flux-weakening operation is
implemented in the control system (Pillay and Krishnan 1991).

A detailed tree of the electric motor classification can be seen in the
Figure 2.2. Of course, each type of motor has its own characteristics that
makes them useful for different applications. Some important character-
istics are the torque, speed and position control. In summary, commonly
used motors have a high speed control but lower torque and they do not
have a position control.

Two main types of motors are suitable for the purpose of simulation.
Stepper motors are useful because they exactly control the position of the
rotor, which enables them to precisely select a specific position, but the
maximum torque is low. Finally, the servomotors are high torque motors
that can also control the position and speed of the rotor thanks to the
closed-loop feedback provided by any kind of position measurement, i.e.
a encoder or a resolver. For this application, where forces of each step in the
synchronization process of a gearbox are going to be simulated, the control
of both the torque and the position is required, and servomotors appear as
the best alternative.

One important aspect of servomotors is the planetary reduction gear, or
planetary gearbox, intended to shift rotational speed. The most common
configuration is the planetary, i.e. epicycloid gearing, where the centre
of the planet gear spins around the centre of the sun gear with different
configurations. A simple internal scheme of a planetary gear can be seen
in the Figure 2.1. In general, planetary gearboxes are used in conjunction
with servomotors to increase torque, decrease motor speed and balance
rotational inertia, as it provides a robust mechanical interface (Antony and
Pantelides 2006).
The gearbox also influences other important parameters with regards to the dynamics of the motor. The elasticity or wind-up of the components under load can affect the positioning accuracy (Antony and Pantelides; 2006). On the other hand, this inertia, added by the gearhead, increases the torque needed to accelerate and decelerate the motor. This leads to an insufficient smooth operation for the purpose of this project, as the high inertia hinders the possibility to emulate the loose movements of a real gear stick (Kim et al.; 2002).

Electronic Equipment and Control Hardware

Kongsberg Automotive disposes of a servomotor system control equipment that will be used to create the simulator. A deep study has to be made in order to discover and reinforce its limitations, and get the knowledge about its characteristics. Through the study it is established whether the
use of this system is enough for the application or if another type of device is required for the system to function effectively.

The system is based on two digital drives that will be able to control the servomotors in charge with the simulation of the behaviour of the gearbox. These drives are used for controlling sinusoidal synchronous AC motors, which fit perfectly with the type of servomotors of the system (Transtechnik Servomécanismes 2012, 2013).

The Data Acquisition (DAQ) of the system is managed with two different computers. These computers are responsible for the control, force and position data acquisition, and behaviour of the entire setup. One of these computers has installed a real-time operating system featured in the NI LabVIEW Real-Time Module, which will be described in later sections. This computer uses two National Instruments PCI-6221 boards for general DAQ tasks, each of them to communicate with one of the controllers. This computer is named internally Controller. On the other hand, the Supervisor is equipped with two National Instruments PCI-6220 boards and is in charge of the general function of the system, which includes the remote control of the PC Controller and the activation of the rig.

These data acquisition boards are part of the National Instruments (NI) M Series of low-cost/multifunction DAQ devices. Even though the boards cannot reach the performance of other devices, e.g. Agilent equipment as seen mentioned by Szabo et al. (2010), they can carry out all of the required tasks in a satisfactory way. For the scope of this project, the boards series absolutely fulfil these requirements.

2.3 Programming

The system is controlled via National Instruments LabVIEW software through two computers also connected to the servo controllers installed in the setup bench. LabVIEW stands for “Laboratory Virtual Instrumentation Engineering Workbench”, and it is a widely used graphical programming language when automation control, and communication between a computer and hardware is established with different interfaces, as General Purpose Industrial Bus (GPIB) or RS-232 communication (Elliott et al. 2007). As said in National Instruments (2013c), ”LabVIEW is a highly productive development environment for creating custom applications that interact with real-world data or signals in fields such as science and engineering”. In LabVIEW the code is not written, but constructed as connections between

---

1PCI stands for Peripheral Component Interconnect and is a computer local bus for connecting devices to the motherboard.
function blocks, called Virtual Instrument (VI), with wires carrying signals and executed inside control structures. This graphical code is translated into executable machine, produced by a compiler included in the software, and later on the executable runs helped by the LabVIEW run-time engine (Sumathi and Surekha 2007).

The graphical programming features of NI LabVIEW play a very important role for virtual instrumentation. Virtual instrumentation is a “interdisciplinary field that merges sensing, hardware and software technologies in order to create flexible and sophisticated instruments for control and monitoring applications” (Sumathi and Surekha 2007). Virtual instruments are powerful software-based applications and specific hardware that can perform as traditional industry instruments. These characteristics permit more suitability and flexibility than fixed-function instruments when building specific systems (National Instruments 2013a). Other advantages are the lower cost versus the more expensive cost of the traditional measurement equipment, the application-oriented nature versus the the function-specific, stand-alone of the traditional ones, and the user-defined characteristics (Sumathi and Surekha 2007). In terms of flexibility, and as treated in the previous sections, different communication protocols and devices can be used with NI LabVIEW, and can be modified depending on the needs.

Moreover, [GUI] is an aspect that prevails in LabVIEW, perfectly integrated with an environment of control and [DAQ] by computer software. In case of simulation and sometimes communication and control by software, MATLAB is a well known software with a widely use due to the big variety of the tasks that can be developed into it. Disregarding the option of employing both programs, LabVIEW presents a better performance in [DAQ] and industrial communication, as in case of the goal of this thesis, controlling a servomotor using an industrial controller and a computer (Tašner et al. 2012). For example, as said by Elliott et al. (2007), “functions can have multiple continuous while loops where one loop is acquiring data rapidly and the other loop processes the data at a much slower rate”. LabVIEW provides powerful tools to run bench applications and simulations, and always struggling with data communication and real-time systems.

LabVIEW has also advantages in regards of real-time operation. Unlike general purpose operating systems, real-time operating systems can perform operations respecting deadlines and critical times for specified duties (National Instruments 2013b). National Instruments provides software that runs specific embedded hardware devices and third-party computers, and features these characteristics for critical timing and high reliability in the programming (National Instruments n.d.a).
Chapter 3

Description of the Setup and Reverse Engineering

In this chapter the description of the hardware and the resources employed in the development of the project are explained, as well as the approach and work of back engineering with the purpose of getting it fully defined and studied for the proper functioning. Also, carrying out this analysis permits the validation of the available hardware.

3.1 General Overview

The setup that conforms the bench where the control program will be implemented can be divided in three different sub-systems. Basically, the mechanical parts of this rig are custom-made elements working as a gearbox without load, and part of the electrical components are also custom made boards and connections merged with proprietary DAQ and control hardware. Then, to servomotors work as the mechanical-electrical interface between the mechanical pieces and the control hardware. The final step are the computers where the programming is carried out.

3.2 Mechanical Shifting, Gear Lever and Connection Cables

The mechanical components of the rig are based on a real manual shifting system, and it is composed by the shifting links and shifting box that are intended to perform as a real shifting system does. While the first part of the shifting system is directly connected to the hand of the driver and will
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

command all the movements done in the system, the end of this sub-system is connected to a shifting box as the intermediate point where the transmission of the forces will be done during the simulation. All the elements of this part are somehow represented as a real car gearing transmission – the gear lever and the links to the gearing levers, previous to the synchronization stage – and their way of performing is based on a real system setup, but it is not a real gearbox, but a “dummy” one.

There are no further details about the mechanical performance of the setup, as it has little impact from the point of view in the development of the simulator. The gear stick consists in an uncovered, simple manual stick connected with thin wire ropes – one for each of the movement – to the gear levers. A picture of the gear stick can be seen in the Figure 3.1. This gear stick permits the movement in any direction, but only horizontal and vertical is considered at one time: left-to-right movement will cause the select operation and up-to-down will cause the shifting operation. But although being considered two independent movements in regards of the direction, both operations are done simultaneously and have impact in each other: the shifting operation will be reflected in the levers in different positions depending on select action is being done in that particular moment. The forces transmitted through the wire ropes cause the rotation of the internal levers, that are constructed in the similar shape as the mechanical pieces of a real gearbox.

Then, this levers transmit another time these movements with metal links connected to the servomotors. This linkage connects the rod of the levering stage to the a connection rod installed in the servomotor shaft. Due to the transmission nature, where a rotatory movement is transmitted through a straight metal link and the dependency between both movements, is necessary to use a spherical bearing in the connection to the servomotor rod.
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

3.3 Servomotors and Control System

The purpose of this stage of the setup is to recreate the behaviour of the gearbox; for this purpose, the system disposes of two identical servomotors that will transmit the required force to the shift and select levers. The servomotors are controlled by two servomotor controllers, each for one of them; this stage is directly connected to the control bench.

The motor model used is a AC brushless Servomotor DSM5.32, manufactured by Sangalli Servomotori. This motor is a high-torque self cooling PMSM constructed with 8 poles and equipped with a brake. This servomotor has a resolver for providing position feedback to control its motion and final position; a resolver is a analog device that calculates the mechanical revolutions of the rotor with two sinusoidal signals provided by two windings attached to the rotor and the stator of the motor. The resulting sinusoidal signal, part of the closed-loop feedback of the servomotor, can be processed by software to obtain either the angular position or rotational speed. Further information about the technical data of the motors and the...
resolvers can be found in Appendix A.1.

Each servomotor is also equipped with a 40:1 planetary reduction gearbox manufactured by APEX Dynamics USA. With this gearbox the par will be increased 40 times, and in the same way, the speed will be 40 times lower. As previously said in the Section 2.2 an attached gearbox with high transformation ratio permits more output torque but inhibits the smooth movements performed by the gearbox.

In the Figure 3.2 can be seen a general description of the servomotor construction. The main part is the electric motor itself (a.) where the stator, rotor, and electronics are located. Attached to the motor shaft (b.) is located the the connecting rod, or crank, which rotates together with the motor rotor. In the end of the crank (c.) can be found the mechanical link with a spherical bearing to transfer the forces to the next stage in the setup. For security reasons, two metal blocks are fixed to the structure to stop risky movements to the system. Finally, attached to the head of the motor (e.) there is the planetary reduction gearbox with the purpose of transforming the torque and speed output.

The servomotors have the connection sockets in the rear of the external armour. The interface used in both sockets is a circular connector type M23 in two different configurations, as they can be seen in the figure 3.3. The one in the left side is used for power and the one in the right carries the information of the resolver.

Figure 3.3: Circular M23 connectors present in the servomotors

The model of the servomotor controllers is a ServoPac TT-230, manufactured by Transtechnik Servomécanismes, prepared for command and work as an interface with a peripheral device as the AC synchronous motors installed in the rig (Transtechnik 2013). This stand-alone devices have the capacity of control the motor in different modes, i.e. analog speed mode or analog torque mode, as well as reading the absolute position of the motor shaft and measuring the total current drawn by the motor. These parameters can be read with a proprietary software intended to test and configure each of the motors, and also the configure the communication between the controllers and the peripheral acquisition boards installed on the rig, i.e. remote activation of the motors.

1Also called digital drives.
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

Each controller has four different D-subminiature connection interfaces\footnote{“D-subminiature” is the denomination for the electrical connector characterized by its metal shield with a D shape. Male connectors are called plugs and are the ones with metal pins, and female connectors are called sockets.} in its rear: a DA-15 female connector for the resolver signals, a DA-26 female connector for the encoder signals, a DA-26 male connector for the general input and output interface, and a DE-9 male connector for configuring the controller with the software using both RS-232 or CANopen communication. Graphic description of these D-subminiature connectors can be seen in the Figure 3.4. Moreover, the controller has two EtherCAT ports –in appearance are identical to Ethernet, as it is based on that protocol– in the upper part of the rear.

The signals carried out in these ports have internal denominations as they can be found in the controller rear. Thereby, the resolver signals are called $X_1$, the encoder signals $X_2$, the general input and output bus is called $X_2$, the RS-232 port is called $X_5$, and the EtherCAT ports are called $X_6$ and $X_8$.

3.4 Electronic Equipment and Signal Wiring

Almost the whole electronic rig is located in a custom build bench, constructed by Cémios as an industrial 19-inch rack, with the mission of working as the heart of the system; this bench includes the necessary electrical connections and the different electronic elements and devices that conforms the equipment. The existing setup is almost custom-build, but includes some proprietary hardware parts. Of course, all the electronic equipment is connected to two computers that will manage the software and the data.

Figure 3.4: D-subminiature connectors present in the ServoPac TT-230 front
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

Computer hardware and control devices

As said before, two computers can be found as part of the system. One of this computers is a [GPC] located outside of the bench. It works with Windows OS and, among the common programs, has LabVIEW 11 installed in it for system-design purposes. This computer is internally denominated supervisor; and is in charge of commanding the full system, as the behaviour of the whole setup and the programming will be done within it. The supervisor computer is also in charge of management of some system signals. The other computer, located inside the rack, is prepared with the LabVIEW Real-time Module 2011 as real-time operating system (RTOS). This computer has the internal denomination of controller, as the vast majority of the data acquisition operation are done with it and is directly connected to the servomotor controllers. In this sense, the supervisor computer will work as the master device in the setup. The controller computer will work as a slave, and the communication between both computers is done by an Ethernet cable connected to them.

Attached to the rack can also be found the two servomotor controllers ServoPac TT-230, installed in the front part of the bench. Despite having its own device driver as said in previous sections, these devices require no direct control from any software, as they provide a continuous signal flow to the data acquisition boards installed present in the setup.

Two communication cables are connected directly from the servomotor to the controllers. These can be seen in the Figure 3.5 as the orange and green cables; the green one carry the information from the resolver and the orange one general inputs and outputs. This data is processed by the controller and then distributed to the electronic data acquisition hardware.

Electronic and data acquisition equipment

The system has the capacity for acquisition of analog and digital inputs and outputs. Also, it is prepared for acquisition of temperature parameters and force measurements through serial port connection, but these inputs are dismissed for the purpose of the project.

The first part of this electronic equipment is a custom-build printed circuit board, manufactured by Cémios, whose purpose is working as an interface between the other stages of the system. This board is where all the inputs and outputs coming from the servomotor controllers merge, and then are sent to the computers via the data acquisition boards, i.e. is connected to the interface of the PCI boards installed in both computers. As can be seen in the Figure 3.5, a multicolour bus of wires represent the
communication of data coming out from the Cémios DAQ board to the National Instruments PCI cards.

As briefly mentioned previously in the Section 2.2, two National Instruments PCI DAQ expansion cards are present in the computers to acquire the data and translate it to the software. The first of them, installed in the control computer, is a NI PCI-6221 low-cost multifunction M-series DAQ board. In particular, the signals acquired by the control computer through this board can be seen in the Tables 3.1. These two signals are directly used to read the position with the servomotors, provided by the resolvers installed in them: these signals come from the resolvers to the servomotor controller, and then are sent again to the data acquisition board in the form of pulses; these pulses are later transformed in the program to angular position. In the other hand, the specification of the outputs can be seen in the table 3.2. Four digital outputs are available to start the servomotors:
two of them for deactivating the STO\(^3\) state, and the other two for direct enabling the motors. Moreover, two analogical output channels command the reference value for the desired torque provided by the servomotors.

**Table 3.1:** Real-time system inputs (NI PCI-6221)

<table>
<thead>
<tr>
<th>Input Port</th>
<th>Name</th>
<th>Signal</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctr0</td>
<td>Shift Position Receiver</td>
<td>Pulses</td>
<td>Resolver</td>
</tr>
<tr>
<td>Ctr1</td>
<td>Select Position Receiver</td>
<td>Pulses</td>
<td>Resolver</td>
</tr>
</tbody>
</table>

**Table 3.2:** Real-time system outputs (NI PCI-6221)

<table>
<thead>
<tr>
<th>Output Port</th>
<th>Name</th>
<th>Signal</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0.0</td>
<td>Shift Torque Control</td>
<td>-10/10 Volts</td>
<td>Reference</td>
</tr>
<tr>
<td>A0.1</td>
<td>Shift Torque Control</td>
<td>-10/10 Volts</td>
<td>Reference</td>
</tr>
<tr>
<td>P2.0</td>
<td>Shift Drive ON</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P2.1</td>
<td>Select Drive ON</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P2.3</td>
<td>Shift STO disabling</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P2.4</td>
<td>Select STO disabling</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
</tbody>
</table>

The second board one is a NI PCI-6220, with similar characteristics as the former one, and is installed in the supervisor computer. As well as the first PCI board, this board also has the capacity of reading the resolver pulses of the servomotors, and thereby calculate the rotor angular position. As outputs, digital signals for enabling the power of the electronic equipment are controlled from the supervisor computer too. In addition, two reserved digital channels can be found at it. More information about the inputs and outputs of this PCI board is listed in the Tables 3.3 and 3.4 respectively.

Although both NI Peripheral Component Interconnect (PCI) cards are almost identical in regards of data acquisition capacity, the main difference between them is that the NI 6221 is prepared with two analog output channels; further information of the technical characteristics and the pinout of both expansion cards can be found in the table found in Section A.2 of the Appendix A. Also, the way they are installed in the system differs in some aspect. The NI 6221 installed in the controller computer uses the Cémios DAQ board as physical interface, connected to it with an industrial

\(^3\)STO stands for Safe Torque OFF, and is a feature of the ServoPac TT-230. It consist of a safety function to prevent an unexpected start-up.
bus. On the other hand, the NI 6220 installed in the supervisor computer uses a NI TBX-68 termination accessory, equipped with 68 screw terminals; here is where the electrical wiring is done, instead of coming directly from the Cémios board. The connection between the Cémios board and the NI device is done through a heavy duty connector rectangular connector.

<table>
<thead>
<tr>
<th>Input port</th>
<th>Name</th>
<th>Signal</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctr0</td>
<td>Shift Position Receiver</td>
<td>Pulses</td>
<td>Resolver</td>
</tr>
<tr>
<td>Ctr1</td>
<td>Select Position Receiver</td>
<td>Pulses</td>
<td>Resolver</td>
</tr>
</tbody>
</table>

Table 3.4: Supervision system outputs (NI PCI-6220)

<table>
<thead>
<tr>
<th>Output port</th>
<th>Name</th>
<th>Signal</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0.0</td>
<td>Enabling Bench</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P0.1</td>
<td>Enabling Power</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P0.2</td>
<td>Reserve var Shift</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
<tr>
<td>P0.3</td>
<td>Reserve var Select</td>
<td>ON/OFF</td>
<td>Digital</td>
</tr>
</tbody>
</table>

3.5 Adjustment of the Components

Although a complete description of the system was carried out thanks to the analysis of the available documentation and observation of the setup, more reverse engineering work was necessary for tuning up the system and preparing it for the purpose brought up by this thesis. In particular, some crucial changes in the electronics and configuration in the software were required.

Initially, further test in the DAQ interfaces was done in order to check the actual reliability of the connections. Inputs and outputs were checked with the software that National Instruments provides for this purpose, the Measurement & Automation Explorer; it is installed automatically with the National Instruments software and provides direct access to the National Instruments DAQ devices [National Instruments, 2013d].

In the first scenario, reading of the inputs (servomotor position sensors) could be done directly. On the other hand, some adjustments were required for some system outputs.
Figure 3.6: Overview of the Cémios board. As can be seen, almost the left part of the board is occupied by the jumper pins. The shunt jumpers in the upper left side are placed in an odd position.

Electrical wiring

The Cémios board, among the electrical connections, power supply, and signal wiring, has a set of jumper blocks for configuring some outputs of the system; these jumper blocks are commonly found in printed circuit boards of all kind for setting-up purposes. This arrangement of jumper pins appears in the Figure 3.6 as can be seen in the picture, different red shunt jumpers are connected for the available green signal ports.

In regards of the signalling, in the upper left of the corner can be seen the labelled wires for the digital output signals 2.4, 2.5, 2.6, and 2.7, previously listed in the Table 3.2. The jumpers immediately under this socket are the responsible for the configuration of this signals connection. The ones for the signals 2.4 and 2.5 were connected in the right position, while the rest were disabled. Only the change of the position was necessary to enable them, as the shunt jumper permitted the correct power supply connection; this correct con-
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

Figure 3.8: Cémios board close-up with the correct configuration

Controller configuration

In respect of the device drivers, setting up a few parameters was also required. The ServoPac TT is widely configurable, including configuration regarding servo-loop, motor and sensor, communication, and input and output parameters (Transtechnik 2013). As referred in that manual, the configuration parameters keep stored into the drive non-volatile memory.

As said in the Section 3.3, these devices can be controlled both with CANopen or RS-232 communication. The latter was the preferred for this purpose, as it is a more spread protocol. Because every connected device is considered a Data Terminal Equipment (DTE), the communication between them has to be done using a RS-232 crossover cable and directly; this type of communication is known as null modem (Lammert Bies 2015). For configuring the device was used the proprietary software provided by the manufacturer, called GemStudio, which can be seen in the Figure 3.9.

At the moment of setting-up the electronic rig only one of the digital drives was not configured. The configuration was taken from the other device, copying the parameters.

Inputs and outputs

The output in charge of enabling the second servomotor was internally disabled in the digital drive. Through the software both inputs were declared and enabled, allowing the remote powering of the motors. In the Figure
3.10 appear the window where the inputs configuration is done. There also appears a further list of the configurable inputs, even though only a few or them are necessary; in particular, the Enable and Error Reset are the important ones regarding the functionality.

Resolver output

The signal coming from the resolver was not prepared and a reconfiguration was also required. Initially, this signal is computed inside the drive itself, and then converted into pulses to be transmitted to the DAQ devices to be post-processed in LabVIEW; these pulses are created with the intent of emulating what an encoder would transmit the position information, as resolver signals are harder to use. In the Figure 3.11 can be seen both configuration dialogues: in the left window physical lines regarding the encoder
CHAPTER 3. DESCRIPTION OF THE SETUP AND REVERSE ENGINEERING

Figure 3.10: Inputs configuration. In the left side can be seen the advanced configuration, and the basic configuration in the right side.

Figure 3.11: Encoder and resolver configuration.

pulses had to be selected as outputs, while in the resolver configuration window just the dialogue box had to be checked.

Output mode

The digital drive software permits several profiles for the motor performance, i.e. profile position or velocity modes [Transtechnik, 2013]. Among all of them, analog torque output is the current mode chosen for the purpose
of this project, since it permits a direct conversion of the output torque controlling the current flowing into the motor power supply; later on this will be explained in the methodology followed to simulate the required forces that have to be transmitted to the system.

The configuration is established writing in the control word $0x6060$ the value $-5$, as indicated in [Transtechnik 2013]. This keeps stored in the drive memory for further use.
Chapter 4

Implementation & Experiments

In this chapter, the main functionalities of the program is going to be explained, going through all the function blocks of the program. Also, the experiments done to test the behaviour of the system are going to be explained.

4.1 Program

Basically, the objective of the whole program is to read some data from a document, interpret that information, and send it through commands so the control system can perform the simulation. For this purpose, the program manages to read the data from the document, transform it, and send the required and properly processed information to the system.

The document containing the data consists of a Microsoft Excel file composed by four or five sheets, depending on the gearbox. Each one of the tables present in the file represent a different movement of the gear stick: One for the select movement, and the others for the shift movements on each one of the different select positions –also called gates–, i.e. the different gears. Each data table is going to be available in one different sheet in this excel file.

As said before, the data appears represented in tables, formed by 10 columns and a large number of rows containing the measurements; this depends on the gearbox tester. Each one of the columns will represent a particular variable about the gearbox testing. Although all the columns are representative for the testing, only few of them are used in the program. In this particular case, the used data is stored in the first and third columns,
because they contain the angle of the test and its corresponding tested force. In summary, the system should act with the force corresponding to the angle in which it was measured. The rest of the data is being kept in the document for further requirements of the project development. An example of this data is showed as a plot in the Figure 4.1.

![Graph showing relationship between force and angular position](image)

**Figure 4.1: “Select” graph with the angle and force of the data**

The program for controlling all the system is divided into two different parts, each one running in one of the sub-systems as seen in Figure 4.2. The functions and duties of each program are set regarding the type of system they are running into; in this sense, the program running in the real-time system is used where a fast response and processing is required, as well as accuracy in the timing. On the other hand, the supervision system is the host for the main user interface and code that does not require high accuracy. As said before in previous chapters, the relation between the supervisor and controller computer follows the model of communication of master/slave.

**Supervision Program**

The first step in the development of the program is reading and processing the data stored in the Excel files to get them adapted to the requirements of the system. The program is divided into different function blocks in
Figure 4.2: The structure of the program with the two sub-systems

charge of solving different types of problems during the execution, and also
arranging the information in the proper way. All the main function blocks
and their functionality are explained in this section.

Initialization

Figure 4.3: Initialization.vi function block, as seen inside the program,
with its inputs and outputs

The purpose of the Initialization.vi (Figure 4.3) function block is
to work as the interface between the user and the program, and manage
the tasks related with this. The user directly interacts with the program, inserting the data document path in a dialogue box, and selecting through the GUI which data will be used. The user has the choice of reading the data form a new path or just using the data that is already stored in the program, if there is any. Basically, the code initializes the system, enabling the bottoms and opening needed files, and then it enters a loop that will be cycling until the user makes a decision.

One part of this loop opens a dialogue box that will contain a help instructions for the operation of the GUI as shown in the Figure 4.4. Once the “help” button is pressed, a help dialogue box is displayed with all the instructions required for the proper control of the GUI as can be seen in Figure 4.5.

![Figure 4.4: Help function code](image)

On the other hand, the second part reads the document file through the path that the user have inserted into the dialogue box, with the purpose of displaying the different available sheets to the user, so that the proper one can be chosen. These sheets are displayed in drop-down list placed on the top of all the graphs, just as shown in Figure 4.6, and the user has to choose one of the data sheets to use it for that particular movement.

Once the user has made the choice for the data in each graph with its corresponding drop-down list, the values contained in the sheets are sent to the next function block, and depending on the user decision, the function will read new data or just use the data that was already stored in the program. The selection is made using the buttons seen in the Figure 4.7.

**Data acquisition**

The second stage is based on the Data Acquisition.vi (Figure 4.8) function block that consists on an automated data-processing function, where
the user does not take any part directly. As seen in the Figure 4.8, the input to this block is the information selected by the user previously, and the output is a cluster of this data in form of a graph; this function block consists of other different blocks that are in charge of extracting the data from the Excel document, as can be seen in the Figure 4.9.

Read Excel

The first function block inside Data_Acquisition.vi is the Read_Data.vi block (Figure 4.10), where the Excel document is opened, and consequently the containing data extracted. This step is done by using the “report gen-
First, the document is opened from the previously specified data path, and once it is opened, another block selects the correct data sheet of the document, taking into account the user choice in the previous step, in the drop-down list dialogue. Once the sheet is selected, another block reads the data inside it and sends it in an array; this process is done for all the graph selection algorithm. Depending on the gearbox model and the user selection, four or five different data tables are extracted, which will
represent each one of the linear movements of the shifter in the physical system. All of this data is put together in a cluster to be sent to the next stage, where that data is processed for further use.

Data Processing

In the Data_Processing.vi function block (Figure 4.12), all the data tables exported in the previous stage are processed individually following the same process:

- Transform the data from string or text to numeric as shown in the Figure 4.13

- Delete the first row, which contains the title and useless information, as shown in the Figure 4.14
The highest and lowest values of the data are found, and the contained information between these values is put inside another table. Finally, all the data is separated in two different sub-tables. The algorithm of this process can be seen in the Figure 4.15.

After those two data arrays are plotted the user will be able to see how the appearance of the read data. At the same time, both of the arrays are sent to a cluster with the other data arrays to the real-time module, where the data is finally processed. Once the data is sent, the program begins a loop cycle where the user has the control again, as is able to command the system remotely. The user will send the commands to start and stop the program, and information about the physical system will be received. This information is plotted in a chart, as seen in the Figure 4.16.

Main program

The third stage of the program is a cyclic loop that manages all the commands and information exchange with the real time system. This loop will be running until the user stops the program or until any error occurs. This block is the second part of the GUI, the monitoring, where the relevant information is displayed instantaneously.
Figure 4.16: Plotted data

Figure 4.17: Monitoring graphical user interface

As seen in the Figure 4.17, the GUI in this page consists of “Start” and “Stop” buttons that enable all the system and let it begin working, and a few visual displays that show the state of the system, i.e., gear number, force applied, or the plots of the movement data. Also, a help button is included to assist the user in the proper use of the GUI.
Internally, the system is composed of basic function blocks that perform a determined tasks in the program. In resume, inside loop values are read and written in the real-time module to interact with the physical system. The program first opens the input/output ports to be able to read and write them, and then, depending on the user interaction with the GUI, it writes the values on the correct output.

**Real-time program**

The real-time system interacts with both, supervision and physical systems. It exchanges information with the supervision system, as the commands of the user come directly from it; the system state information is also sent to the supervision system to be displayed at the GUI, and contains the instantaneous force, angle, and information about the gear that the system is in that particular moment. This information is sent using a shared variable in LabVIEW that allows the use of it by both systems; as said in previous sections, the communication between them is established by EtherCAT protocol.

On the other hand, the state of the physical system is checked, the *enable* signal is read from a digital input in the DAQ board and the position of each one of the servomotors is acquired and transformed in the software. For the interaction with the the other components this system is based on different function blocks that will manage the program running.

**Continuous execution**

![Continuous Execution.vi function block](image)

**Figure 4.18: Continuous Execution.vi function block**

The program is based on a real-time execution loop that interacts with the physical system to make it behave in the proper way. The inputs of the *Continuous Execution.vi* function block (Figure 4.18) are the information coming from the supervision system, and the angle information that is measured in the same real-time program with a data acquisition module. On the other hand, the outputs of the system are the signals to command the motors, that are force and angle; these are sent to the supervision system,
as well as the gear position information. Inside this block, other function
block manages to transform this input information into the outputs, as seen
in the Figure 4.19.

![Figure 4.19: Continuous_Execution.vi code, composed by different function blocks](image)

**Figure 4.19:** Continuous_Execution.vi code, composed by different function blocks

**Angle Transformation**

![Figure 4.20: Angle_Transformation.vi function block](image)

**Figure 4.20:** Angle_Transformation.vi function block

The first part of the program is the Angle_Transformation.vi function
block (Figure 4.20). Here, the transformation of the angles and another
needed measurements is carried out. This angle is based on the resolver
signal coming from the motor and then transformed inside the servo drive
to an encoder-type signal, but due to the mechanical joints and linkages,
this angle has to be transformed. For that, an approximated angle transfor-
mation is applied to the measured angle to convert it to the angle measured
in the levers in the shifting box. That transformation is based on a math-
ematical equation that is applied to the initial angle value.

As shown in the Figure 4.21, while reading the data, the correct position
of the Shift movement is checked and stored. The same procedure is applied
in the Select angle, and taking into account the two positions, a different
data array, i.e. data table is used, so it is important to check the correct
current position of both movements.
CHAPTER 4. IMPLEMENTATION & EXPERIMENTS

37

Figure 4.21: Angle transformation and shift position

Data Choosing

Figure 4.22: Data Choosing.vi function block

The next stage is to choose the data with the Data Choosing.vi function block (Figure 4.22), which is used to perform the simulation of the gearbox’ behaviour. First, the correct Shift data is chosen. Depending on the Select position, one of the four data arrays is selected to be used as the reference, as seen in the Figure 4.23.

Figure 4.23: Selection of the "Shift" Data depending on the "Select" position

As explained above, this data table is divided into two sub-arrays, each one representing a direction of the movement – increasing and decreasing of the angle value— so the next step is to detect whether the angle of the servomotors is increasing or decreasing. Depending on this value, shift angle
and select angle, the correct data sub-array is going to be independently selected from both data arrays as shown in Figure 4.24.

![Figure 4.24: Selection of the correct data depending on the direction of the angle change](image)

These two data arrays are sent together with the angle data in a cluster to the next function block, where the information is properly extracted and processed.

**Force Selection**

![Figure 4.25: Force_Selection.vi function block](image)

The Force_Selection.vi (Figure 4.25) function block chooses the actual force value that has to be applied to the system. For this purpose, an algorithm based in the k-NN algorithm\(^1\) is employed. This algorithm finds the nearest value of the angle that has been measured, detecting which is the correct value of force that has to be send to the system as shown in the Figure 4.26.

The code finds the nearest value of the input angle, and consequently the closest value in the “angle column” table, in order to find out the location of the required force, that is correlated to the angle value by the row number. The column of the angle is extracted from the main table and the current angle value subtracted. The output is a table composed by

---

\(^1\)K-nearest neighbour algorithm, shortened k-NN, is a method used for classification and regression.
remainders resulting from the different subtraction operations; the lowest value will be the closest to the actual value. In the program, the minimum of the absolute values in the column of the table is found with a particular LabVIEW block, which outputs the row number of this minimum value. When the row number is known for this minimum value, the required force for that particular angle is found. This procedure is applied for both Select and Shift forces, and are sent together with the angle values to the next and last function block.

**Force algorithm**

![Force Algorithm VI](image)

**Figure 4.27: Force Algorithm.vi function block**

The next and final function block is called Force Algorithm.vi (Figure 4.27), where the force value is transformed into signals to be sent to the servo drives. The motors need a particular torque value to work, and the force values given in the Excel files are measured directly from the shifting levers. Because of that, these forces have to be transformed. This transformation takes into account the algorithm in regards of the mechanical parts of the setup, and it is given by the company: this is a formula that has to be implemented in the software. Two different transformation algorithms have been used for both movements, as shown in the Figure 4.28.

Apart from that algorithm, another factor has to be taken into account. As explained before, the servomotors do not apply force but torque, so the output force of the algorithm has to be transformed to torque. Even though that transformation is not a simple one, due to a small angular displacement, the company agrees to despise this value and to transform
Figure 4.28: The programing code transforming the input force with a formula

\[ T = F \cdot r \]

the force into torque in the conventional way, by multiplying force and length

Finally, this torque has to be transformed to an electrical signal. As seen in the Appendix A, the motors have a torque constant of 0.91 Nm/A; this constant permits a direct transformation from torque to current. In addition, the planetary gearbox attached to the servomotor helps it to increase while decreasing the speed. As previously said in the Section 3.3, the particular planetary gearbox of this application has a 40:1 transformation ratio, and it has to be taken into account in the transformation to be able to have the correct output in the system. All this information is implemented in the code of the function block, as seen in Figure 4.29.

Figure 4.29: The programing code transforming the input force into the signal for the motors
4.2 Experiments

Experiment Description

For validating the project, some experiments have been performed to test the system and to ensure the approximation to a real gearbox. The most important experiment within them has been testing the different forces of the shift stick in the simulator to establish a correct comparison with the real system. This test consists on the measurement of the Shift and Select forces in a shifting cycle. The test shows the applied forces by the user’s hand to make the movements. The same test is done in the real gearbox, with the same gear lever system, to compare both and check the differences.

With this purpose, a test equipment provided by the company has been used in the real gearbox and in the simulator. This equipment consists on a force sensor and a distance sensor that are placed on top of the gear stick. Using a specific software, the applied force in each one of the positions is measured and plotted. This graphs represent the behaviour of the shifting movements, and are used to compare both, the simulated gearbox and the real one. As seen in the Figure 4.30, the force sensor is placed into the shifter lever to detect and measure the force that the user is applying.

![Force sensor positioning](image)

Figure 4.30: Force sensor positioning

Apart from that, a position sensor is connected to the force sensor, to detect the displacement when the movements are done in the stick, as shown in Figure 4.31.
The experiment was done with all the movements of the real gearbox, which are select, shift gears 1-2, shift gears 3-4, shift gear 5 and shift gear R.

Experiment results
For comparing the results obtained in the experiment, the measurement data graphs from the simulator testing and the real gearbox testing have been plotted in order to establish a comparison. The graphs show similarities between the real gearbox shifting and the simulator. As example, two experiment results are shown as plots in the Figures 4.32 and 4.33. In these graphs, the test results obtained in both systems – the real gearbox and the simulated one – are plotted overlap together to compare similarities. In general, the graphs show the change of the force depending on the displacement of the shifting lever, i.e. the similarity in the shape of the curves show that the behaviour is quite close when different positions are reached with the gear stick.

The analysis of the graphs shows that there are some offsets in the curves of the simulation and the real system. Apart from that, both graphs in the two movement show a similar performance on the user. The complete set of graphs can be seen in the Appendix B.
CHAPTER 4. IMPLEMENTATION & EXPERIMENTS

Figure 4.32: Select movement comparison curves

Figure 4.33: Comparison between real first and second gears shifting forces and simulated one
Chapter 5

Conclusions

In this chapter the results of the developed work are commented, as well as some future work is described in order to present different ways for further development of the system.

5.1 Conclusions About the Results and Future Work

The goals of the project have been completed successfully, in terms of the simulation of the gearbox. The programming of the test platform is now a reality. The experiments and test done in the setup show a correct functioning in regards of the system operation, that approaches reasonably well to the real behaviour, but there are some points of optimization. The mechanical joints all over the system and the servomotors obstruct the possibility of a good performance and a perfect transmission of the forces, due to friction, inertial forces, joint clearance, and the weight of the mechanical links among others. In summary, a better mechanical installation would get rid of some inaccuracies in the behaviour, and improve the general performance of the system.

For example, the possibility of the servomotor to provide a higher torque incapacitates the motor to rotate without offering resistance to the movement due to the planetary gearbox inner friction, as commented in previous chapters; a reduction gearbox with a smaller transformation ratio –the one present in the system has transformation ratio of $40:1$, which is really high– would permit a better and more realistic performance. On the other hand, and as an alternative for the improvement of the simulator in this sense, a strategy similar to the ones used for power steering system could be used: a position and torque control that pretends the absence of the inertia added
by the planetary reduction gear, making the rotation of the motor shaft smoother. This would be a good solution, maintaining the high torque in the output, but further and deep software development would be required.

Regarding the program, a lot of different future improvements could be developed. The whole program itself has been developed for a basic functioning, but more advanced features can be added. For example, the blocking of the select movement when a gear is selected can be implemented, approaching the simulator to a more realistic behaviour; due to limitations in the hardware, this is not possible now, but it can be developed in further works, as most of the servomotor models, including the one used in this project, dispose of a braking system that would perform in the desired way to develop those features. Also, the program could be simplified so that more samples could be done in the same time, improving the program response to the system.
## Appendix A

### Equipment technical data

#### A.1 Servomotor technical data

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Units</th>
<th>Model: DSM5.32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stall Torque</td>
<td>$M_0$</td>
<td>Nm</td>
<td>2.9</td>
</tr>
<tr>
<td>Max Voltage</td>
<td>$U_{\text{MAX}}$</td>
<td>V</td>
<td>230</td>
</tr>
<tr>
<td>Stall Current</td>
<td>$I_0$</td>
<td>A</td>
<td>3.2</td>
</tr>
<tr>
<td>Nominal Current</td>
<td>$I_N$</td>
<td>A</td>
<td>2.7</td>
</tr>
<tr>
<td>Nominal Torque</td>
<td>$M_N$</td>
<td>Nm</td>
<td>2.4</td>
</tr>
<tr>
<td>Nominal Power</td>
<td>$P_N$</td>
<td>W</td>
<td>750</td>
</tr>
<tr>
<td>Nominal Speed</td>
<td>$N_N$</td>
<td>min$^{-1}$</td>
<td>3000</td>
</tr>
<tr>
<td>Max Speed</td>
<td>$N_{\text{MAX}}$</td>
<td>min$^{-1}$</td>
<td>4000</td>
</tr>
<tr>
<td>Peak Current</td>
<td>$I_{\text{MAX}}$</td>
<td>A</td>
<td>13</td>
</tr>
<tr>
<td>Voltage Constant</td>
<td>$K_E$</td>
<td>V/Krpm</td>
<td>55</td>
</tr>
<tr>
<td>Torque Constant</td>
<td>$K_T$</td>
<td>Nm/A</td>
<td>0.91</td>
</tr>
<tr>
<td>Rotor Inertia</td>
<td>$J_R$</td>
<td>Kg·cm$^2$</td>
<td>1.72</td>
</tr>
<tr>
<td>Resist. 20°C</td>
<td>$R_{U-V}$</td>
<td>Ω</td>
<td>3.4</td>
</tr>
<tr>
<td>Induct. 1 KHz</td>
<td>$L_{U-V}$</td>
<td>mH</td>
<td>7</td>
</tr>
<tr>
<td>Mass</td>
<td>$m$</td>
<td>Kg</td>
<td>3.5</td>
</tr>
</tbody>
</table>
### A.2 National Instruments DAQ devices specifications

**Figure A.1:** National Instruments PCI 6220/6221. Taken from National Instruments (n.d.b).

<table>
<thead>
<tr>
<th>Family</th>
<th>NI 6220</th>
<th>NI 6221</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>PCI, PXI</td>
<td>PCI, PXI, USB</td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>AI Resolution (bits)</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Analog Outputs</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>AO Resolution (bits)</td>
<td>-</td>
<td>16</td>
</tr>
<tr>
<td>Max. AO Rate (kS/s)</td>
<td>-</td>
<td>833</td>
</tr>
<tr>
<td>AO Range (V)</td>
<td>-</td>
<td>±10</td>
</tr>
<tr>
<td>Digital I/O</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Correlated (clocked) DIO</td>
<td>8, up to 1 MHz</td>
<td>8, up to 1 MHz</td>
</tr>
</tbody>
</table>
Figure A.2: National Instruments PCI 6220 Pinout. Taken from National Instruments (n.d.b).
Figure A.3: National Instruments PCI 6221 Pinout. Taken from National Instruments (n.d.b).
Appendix B

Experiment results

Figure B.1: Comparison between real select movement graph and the simulated one
Figure B.2: Comparison between real third and fourth gears shifting forces and simulated one

Figure B.3: Comparison between real third and fourth gears shifting forces and simulated one
Figure B.4: Comparison between real fifth gear shifting forces and simulated one

Figure B.5: Comparison between real R gear shifting forces and simulated one
Bibliography


URL: http://dx.doi.org/10.1007/BF02996112


Transtechnik Servomécanismes (2013). *ServoPac TT Application Templates*, 1.8 edn, Transtechnik Servomécanismes.