Ignacio Perez Mahiques

Improvement of the rudder force simulation in the flight simulator of TU Graz

Bachelor Thesis

to achieve the university degree of

Bachelor of Science

Bachelor's degree programme: Industrial Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl-Ing. Dr.techn. Reinhard Braunstingl

Institute für Mechanik
Head: Univ.-Prof. Dipl-Ing. habil. Katrin Ellerman

Graz, June 2019

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

The project consists in the creation of a device that simulates the rudder force feedback of a flight simulator. In a real airplane, a force is felt by the pilot when pressing the pedals in order to turn the rudder. This device consists of electronic components, which are used to calculate the necessary input voltage of the motor controller that will create a force feedback on the pedals. The inputs of the device are the position of the pedals and two parameters sent by the flight simulator software, a gain and a displacement.

The device that is explained during the project improves the fidelity of the simulation and introduces a new security mechanism compared to the device made by Thomas Krauss in which this project is based.

Contents

List of Figures

1 Introduction

1.1 Flight simulators

Flight simulators reproduce the situations that can happen in a real aircraft and are used to develop navigation, maneuver and maintenance skills in pilots (Villamil Rico, Avella Rodríguez, and Tenorio Melo, 2018). Since the pilots need to feel the same in a flight simulator than in a real flight, fidelity is critical. Hence, the hardware and software of the simulator should adapt to the circumstances of the flight.

According to the Federal Aviation Administration (FAA) (2016), flight simulators are classified according to their level of qualification in Aviation Training Device (ATD), Flight Training Devices (FTD) and Full Flight Simulators (FFS).

- The ATD are training simulators that resemble the cockpit and the software from an aircraft and they provide a platform to train in flight tasks. They are classified in Basic ATD (BATD) and Advanced ATD (AATD) depending on the characteristics of the simulator.
- The FTD are simulators that recreate the characteristics of a specific aircraft. A motion system is not required but they meet the requirements for most certificates. They are classified from level 4 to level 7, depending mostly on the aerodynamic model of the simulator.
- The FFS are the most sophisticated flight simulators in the classification by the FAA. They include a motion and visual system and they are classified depending on the characteristics of these systems in levels A-D, being D the highest level (Federal Aviation Administration (FAA), 2016).

Figure 1.1: Airbus 320 FTP (Robinson, Mania, and Perey, 2004)

Figure 1.2: Full Flight Simulator (Robinson, Mania, and Perey, 2004)

Nowadays, low fidelity flight simulators, such as the one shown in the figure 1.1, are being developed to recreate firmly the avionic display. However, the interaction between the user and the device is synthetic and unnatural (Robinson, Mania, and Perey, 2004). High fidelity flight simulators are a really important part of the training of future pilots, and knowing exactly how the airplane behaves to each action of the pilot is critical.

Therefore, it is really important to improve high fidelity simulators to be identical in aspect and simulate the external factors such as pressure, air temperature and air speed. The physiological reactions in a flight can also be critical in making a decision. Consequently, motion, touch, vision and hearing should be taken into account (Robinson, Mania, and Perey, 2004). Figure 1.2 shows a high fidelity flight simulator, which has a motion and visual system to resemble as much as possible a real airplane.

When pilots in a real flight control the rudder with the pedals, a force feedback is felt by the feet and this should be experienced in a flight simulator before flying an airplane. This feedback depends on the type of airplane being controlled and the circumstances of the flight, such as the speed of the wind and the altitude.

1.2 Motivation

Robinson, Mania, and Perey (2004) emphasize the importance of kinesthesis (motion and touch) in the fidelity of a flight simulator. The prototype made by Thomas Krauss managed to simulate the rudder force feedback of the pedals with a digital electronic device. The first prototype stopped working correctly as a result of reading bad inputs that launched the security mechanism in normal circumstances. A feedback output to read the important variables of the device would be useful to quickly discover which part is not working. Hence, the motivation of this project is to increase the fidelity of the flight simulator by improving the rudder force feedback and solving the security issues that affect the old device.

1.3 Objectives

The device in which this project is based is an improvement of the first prototype made by Thomas Krauss. The focus of the project lies on the addition of more reliable input sensors, a security mechanism that sends the error and alerts to the user and increases the fidelity of the device. Therefore, the objectives of the project are:

- Developing a functional device that simulates correctly the rudder force feedback on the pedals.
- Creating an output of information with the data that is being managed by the microprocessor and the reason of a possible error on the system.
- Improving the fidelity of the device by improving the reading of the input from the pedals and the flight simulator software.
- Increasing the accuracy of the processing of the signal, which is composed of analog to digital conversions and calculations by the microprocessor.

2 Background Information

2.1 Simulation of the rudder force

The simulation of the rudder force feedback on the pedals in a flight is really important for the training of a pilot in order to learn how to press the pedals to achieve the desired movement of the aircraft. Firstly, the rudder force depends of the characteristics of the aircraft. Hess (2005) compares the rudder force from three transport aircrafts and a helicopter. Figure 2.1 shows the difference gain and displacement of a transport aircraft and the helicopter. Moreover, between two similar current airplanes from different manufacturers the difference is also quite significant. One airplane has 80 lbf of maximum force and 3.6 inches of maximum displacement compared to 125 lbf and 4.0 inches of the other airplane (Hess, 2005).

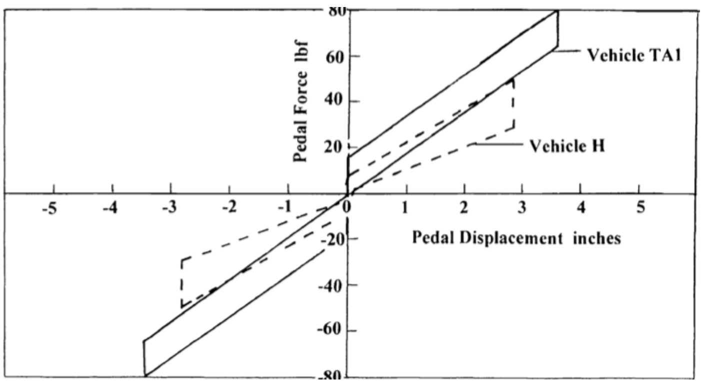

Figure 2.1: Pedal force/feel characteristics between aircrafts TA1 (transport airplane) and H (helicopter) (Hess, 2005)

In a specific aircraft, the rudder force depends on the velocity flow in the rudder. The velocity flow is the quantity of air that hits the rudder, which prevents it from turning. Since the higher speed of the aircraft causes more wind to hit the aircraft, the velocity flow increases with the speed. The altitude of the airplane also affects the rudder force because the density of the air increases with the altitude. Therefore, the higher altitude an airplane reaches, the higher maximum rudder force is created. The velocity flow and the characteristics of the aircraft define a gain parameter. Figure 2.2 shows the variation of this parameter.

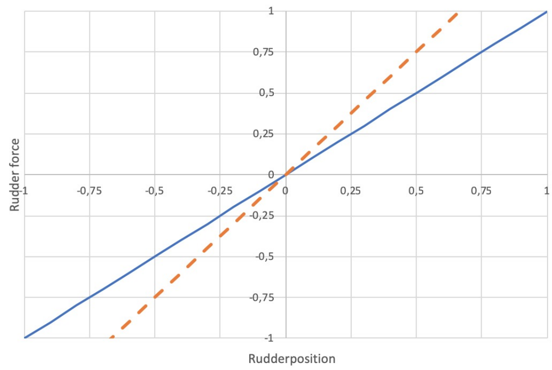

Figure 2.2: Modification of the gain of the rudder force

The displacement of the rudder force, which is the deviation of the rudder force in the X-axis, is caused by the trim position of the pedals. The pilot can set a constant force to the pedals without pressing the pedals. A real situation when the trim position is used is in case of an engine failure. When an engine from one side of the aircraft fails, the pilot must counter the difference of force from one side of the aircraft with the rudder. The pedals are set in a position where the aircraft is balanced. Figure 2.3 shows how the rudder changes when the trim position is set to another value.

Once the mathematical model with the two parameters, gain and displacement, is set, the position of the pedals defines the rudder force that will be applied on them. Since the pedals control the angle of the rudder, the force will be highest when the pedals are pressed the maximum and null when the pedals are in the initial position if there is any displacement.

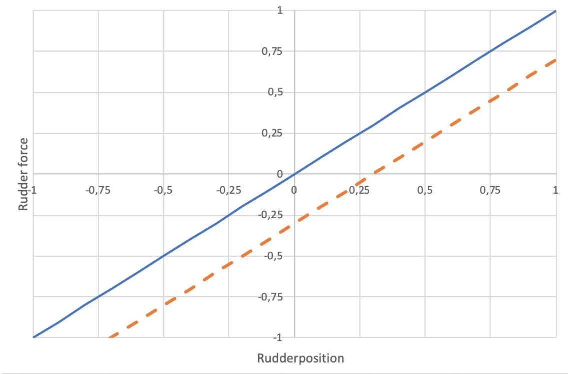

Figure 2.3: Modification of the displacement of the rudder force

2.2 Control theory

A control environment is a system whose function is to read the inputs of the system and respond to these inputs accordingly. Each system has different inputs and should react differently to them. Therefore, a mathematical model should be designed for each system, in order to give the right output.

There are two main types of control systems: open-loop systems and closed-loop systems. As shown in figure 2.4 a basic open-loop system has two inputs: a control input and a disturbance input. The two inputs modify the summing signal that arrives to the plant, element in which the system is based. The problem in this system is the difficulty to know how the plant should react to give the right output, for example, maintain the output variable constant (Burns, 2001).

This problem is solved in a closed-loop system, where a sensor is measuring the output value to detect the error of the output in order to permit a controller to correct it as shown in figure 2.5. The controller is designed from the mathematical model that adapts to the system. The measured value is returned to the beginning of the system to compare it with the designed value. The disturbance input, which causes instability in the output of the open-loop system, is minimized by the controller of the system.

The device that simulates the rudder force is a closed-loop system in which the measured value is the input that comes from the pedals, which is made

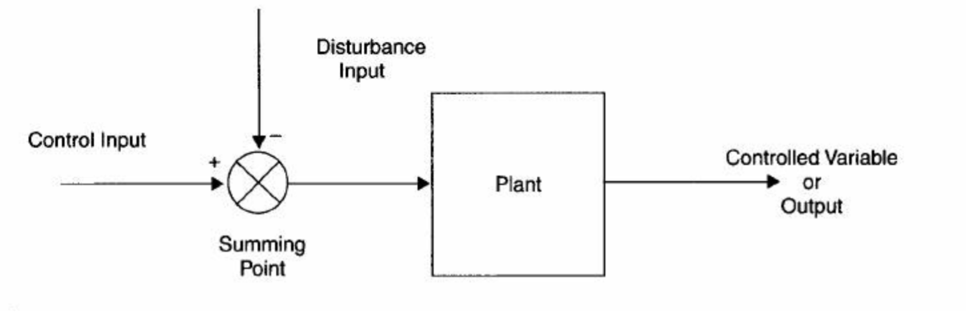

Figure 2.4: Open-loop control system

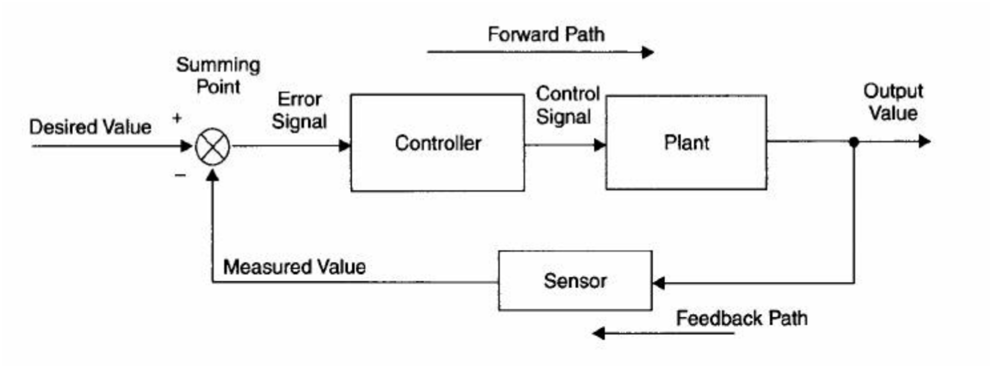

Figure 2.5: Closed-loop control system

by the pilot to change the rudder angle of the aircraft. The desired value or reference is given by the flight simulator. With the two inputs the device creates a response, which is transferred to the pedals as a force feedback.

The mathematical model, which is explained in section 2.1 could be seen as a P controller, since it has a proportional relation. This relation depends on the gain and the diversion of the starting point, which are given as an input of the flight simulator. The output would be the one that is calculated by the mathematical model. This process runs in a loop to maintain the force feedback at the desired value that the flight simulator is sending to the device.

Figure 2.6 shows a block diagram of a control system created by Crowe (2005). In the simulation device, the noise signal $N(s)$ and the disturbance input D(s) are null, the composite process signal $Gp(s)$ is 1 and k_c from equations 2.2 and 2.3 is the proportional value that defines the P controller and changes with the input. In this case, the mathematical expressions are:

$$E(s) = R(s) - Y(s) \tag{2.1}$$

$$U_c(s) = k_c E(s) \tag{2.2}$$

$$Y(s) = U_c(s) = k_c(R(s) - Y(s)) \tag{2.3}$$

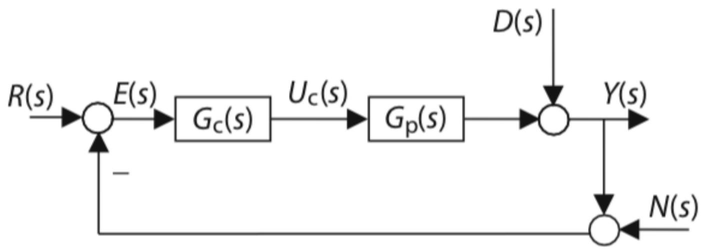

Figure 2.6: Block diagram control system

2.3 Electronic devices information

The project is based in the subject of electronics because multiple electronic devices are used in order to achieve the digital simulation of the rudder force. In this section these devices will be introduced and explained:

2.3.1 Potentiometer

The term potentiometer describes a variable resistor that changes its resistance from ground (0 Volts) to the maximum resistance depending on a linear or rotary movement. As shown in figure 2.7, a potentiometer is generally composed of three terminals. The first terminal is connected to ground, the second terminal is the output voltage of the potentiometer and the third terminal is connected to the maximum voltage that is wished as output.

A constant electric current goes through the resistor producing a voltage, for example, if the resistance is maximum the output voltage will be the same voltage of the third terminal. In this application, a high precision potentiometer is used to create a voltage, which defines the movement of the rotary shaft moved by the pedals.

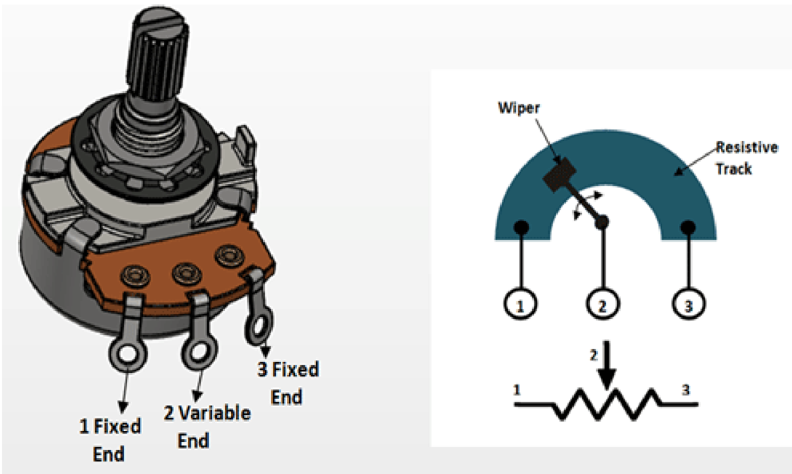

Figure 2.7: Diagram of a potentiometer

2.3.2 Operational Amplifier

An operational amplifier is an amplifier circuit that is used in analogue circuits in many applications. This simple building block has two inputs, the inverting input and the non-inverted input and normally one output. The operation amplifier has two power supply connections for the positive and negative rail, which are normally connected to a positive and negative voltage, even though one can be connected to ground.

The main principle in an ideal amplifier is that there is no current flow at the input terminals; consequently, the voltage of the inputs is equal. However, a real amplifier has small currents at its input terminal. Moreover, an ideal amplifier has infinite open-loop gain and input impedance, while these characteristics in a real amplifier are large but not infinite (Clayton, 2003). An operational amplifier can act as an inverter, an integrator, a current-to-voltage converter or a subtractor, depending on the basic configuration that is being used.

One limitation of the operational amplifier is the saturation, as shown in figure 2.8, an operational amplifier cannot have a larger output than the supply connections (Clayton, 2003). For example, an operational amplifier that is connected to 5 Volts and ground, it will reach saturation if the output is negative or bigger than approximately 90% of 5 Volts (4.5 Volts). In this case, the two input voltage will change in order to avoid the impossible output value. This could lead to permanent damage in the operational amplifier.

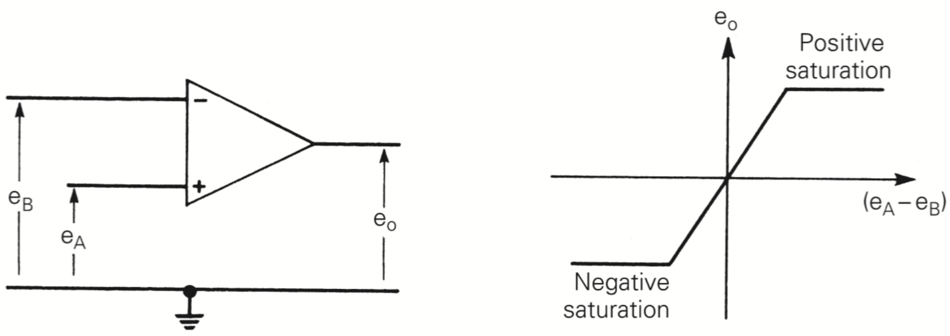

Figure 2.8: Idealized transfer curve in an operational amplifier (Clayton, 2003)

2.3.3 Zener diode

A Zener diode is a diode with a highly doped p-h junction. This diode allows the current to flow from the anode to the cathod if the voltage is bigger than V_f and in the opposite direction if the voltage is more negative than $-V_r$. Therefore, when a voltage in one side of the Zener gets near to its V_f or $-V_r$, the current starts to flow limiting the voltage since that current decreases the voltage. This characteristic is used in the device to protect the single board computer. A resistor should be connected before the Zener diode in series in order to limit the current flow through the diode.

2.3.4 Analog to Digital Converter (ADC)

An analog to digital converter is an integrated circuit that converts an analogue voltage into digital data. The most important specifications are resolution, accuracy, conversion time and sampling frequency. The resolution is the number of discrete steps in which the input voltage is divided. The resolution depends on the number of bits of the converter and the input voltage.

$$Resolution = \frac{Vout}{(2^n - 1)} \tag{2.4}$$

Moreover, the resolution is the input voltage required to modify the Least Significant Bit (LSB) in the output code. This can be seen in figure 2.9, where the resolution is the quantum of analog input voltage that produces a change in the output.

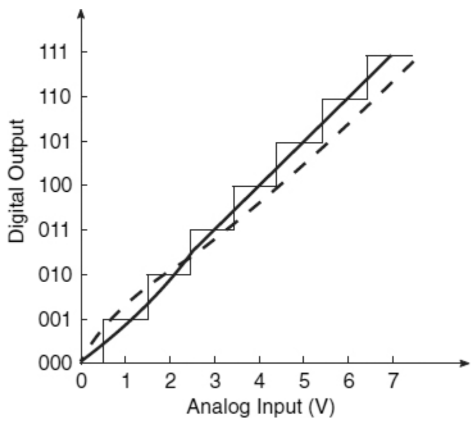

Figure 2.9: Transfer characteristics of a 3-bit Converter (Maini, 2007)

Accuracy is the error between the actual output and the expected output when an analog input is transformed into a digital output. The error is formed by the gain error, the offset error, the quantization error and the nonlinearity error. In figure 2.9 the nonlinearity error is represented. The conversion time is the time period that has elapsed for the ADC to convert the analogue input into a complete digital data (Maini, 2007).

In ADC, sampling frequency is the rate at which the analogue input is sampled. This parameter is important because it should be higher than twice the highest frequency in the analogue signal to avoid aliasing. According to the Shannon-Nyquist theorem, the digital output will be different as expected if the sampling frequency is not right, as some information is being lost in the conversion (Maini, 2007). An anti-analyzing filter is recommended to improve the accuracy of the converter.

2.3.5 Single board computer (SBC)

A single board computer (SBC) is a computer that has all the components of a computer in a small board. The components are usually soldered directly into the board.

The most important component of the board is a system on a chip (SoC). A SoC is an integrated circuit that contains the main modules of a computer in one component (Clark, 2019). Moreover, an integrated circuit is a chip that integrates a circuit of several electronic components into a solid block (Kumar and Umashankar, 2008). Since a SBC is a small size computer, the SBC is connected to other components by digital inputs and outputs. Analog signals can be read and sent by the Arduino using analog to digital and digital to analog converters, which are in some SBC included in the board.

2.3.6 Arduino

The software used in the single board computer (SBC) is Arduino. Arduino is a platform that provides hardware products, such as single board computers and a development environment, which is open source and could work with third-party product as well as its products. The Arduino integrated development environment is an intuitive editor based in a subset of C, more user-friendly but with the same standard functions (Perea, 2015).

The library used to transmit the digital signal from the single board computer to the DAC is Serial Peripheral Interface (SPI). SPI is a serial data protocol used for communication between a microprocessor and other devices in short distances (*Arduino - SPI*, 2015). The main device, in this case a microprocessor and the peripheral devices, have normally four lines:

- MISO, Master In Slave Out, in which the slave line sends data to the master.
- MOSI, Master Out Slave In, in which the master sends data to the peripherals.
- SCK, Serial Clock, the clock pulses that define data transmission
- SS, Slave Select, a line specific to each device that is used by the master to enable and disable each peripheral device.

When the SS pin is high, data is transmitted from the master and when the pin is low, the master signal is ignored. This allows to send different data to multiple peripheral devices (*Arduino - SPI*, 2015).

2.3.7 Digital to Analog Converter (DAC)

"D/A converter takes digital data at its input and converts them into analogue voltage or current that is proportional to the weighted sum of digital inputs." (Maini, 2007). The most important specifications are resolution, accuracy and conversion speed. The resolution is the number of discrete steps in which the maximum output is divided. The resolution in percentage depends only in the number of bits of the converter.

$$Resolution(\%) = \frac{1}{(2^n - 1)}100 \tag{2.5}$$

However, in a real system the resolution can be calculated by multiplying the output range. For example, with a 5V output range and a 16 bit DAC, the resolution would be 76 μV.

$$Resolution = \frac{Vout}{(2^n - 1)} \tag{2.6}$$

Accuracy is the error between the actual output and the expected output when a digital input is transformed into a analog output. It can be calculated with the resolution.

$$Accuracy = \pm\frac{Resolution}{2} = \pm\frac{Vout}{2(2^n - 1)} \tag{2.7}$$

The conversion speed or settling time is the time period that takes the DAC to receive the digital input and the analog output to reach its final value (Maini, 2007).

2.3.8 Controller of the motor

A servo drive is used to provide speed and torque control for the AC servo motor that controls the feedback on the pedals. The function of this mechanism is to receive the incoming signal from the DAC, to amplify the signal to the requirement of the motor and to transmit an electric current to the motor. There are two types of servo motors: digital and analog. The difference between them is that the digital motor analyses the input signal in order to adjust the output current to the motor requirement and improve the performance of the motor.

2.4 Current status of the flight simulator of TU Graz

The flight simulator in TU Graz is fixed base and uses the hull of a DC 10 simulator. The vision system consists of a 7 by 3 meters screen, which reflects a horizontal of 180 degrees. Figure 2.10 shows the interior of the flight simulator, which recreates an airplane cockpit with highly innovative avionics. The avionics software used is the same as the one in some Full Flight Simulators (FFS) and the simulator software simulates a flight dynamic, engines and a radio navigation system (*TU Graz Flight Simulator*, n.d.).

Figure 2.10: Interior of the flight simulator (*TU Graz Flight Simulator*, n.d.)

Before 2015, a mechanical device was used to simulate the rudder force in the flight simulator of TU Graz. This mechanism was formed by springs that made a constant feedback force when the pedal was pressed. This mechanism was totally inaccurate in some situations, for example, at take-off where there is almost no force by the wind and consequently, no feedback force should be made by the pedals. Hence, a digital simulation would increase the fidelity of the feedback of the pedals.

The first prototype made by Thomas Krauss was composed by a potentiometer, which converted the rotary movement of the pedals shaft to a voltage. Therefore, the output voltage would be maximum when the pedal was pushed at full force. Before converting the signal to digital, an operational amplifier was used to increase the output voltage of the potentiometer and consequently improve the accuracy of the analog to digital converter.

The Analog to Digital Converter (ADC) converts the voltage in a binary code, which is read by the microprocessor. The microprocessor used was the Arduino UNO and received the information from the flight simulator software with a RS-232 cable. The microprocessor gave an output depending on the position of the pedals and the air force.

The digital output sent by the microprocessor arrived to the Digital to Analog Converter (DAC), which transformed the output in a voltage, which was multiplied in an operational amplifier in order to adapt to the requirement of the motor. The operational amplifier was connected to the controller of the motor, which moves the motor to give the right feedback in the pedals.

At first the device created by Thomas Krauss, which is shown is figure 2.11, worked well. Currently, in some situations the device stops working properly, probably because the safety mechanism goes off unnecessarily. One probable reason is that the potentiometer gives wrong measurements. Since the device does not have an interface or an output of data, the problem cannot be specified. For this reason, the easiest way of solving the problem was creating from scratch a device that would do the same function.

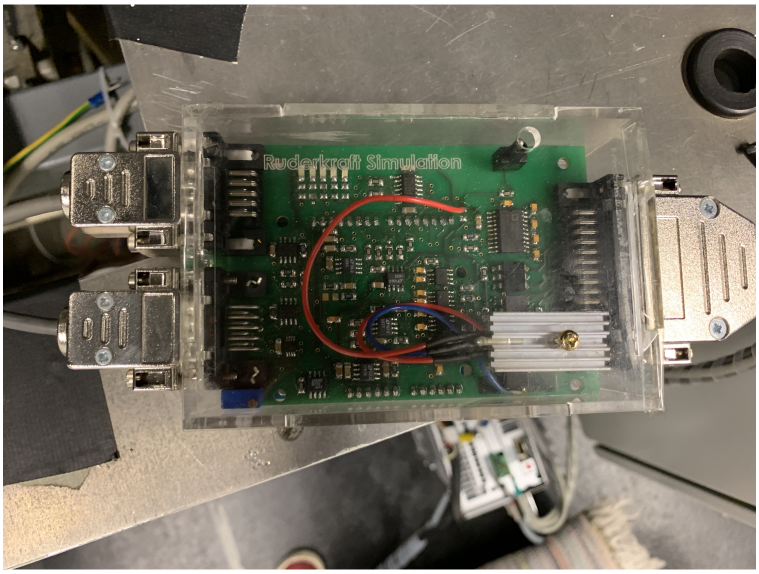

Figure 2.11: Ruder force simulation by Thomas Krauss

3 Proposed system

Figure 3.1 shows the proposed system of the project, which gathers all the components that are necessary for the creation of a device that simulates the rudder force feedback in the pedals. Each component is described in more detail in this chapter.

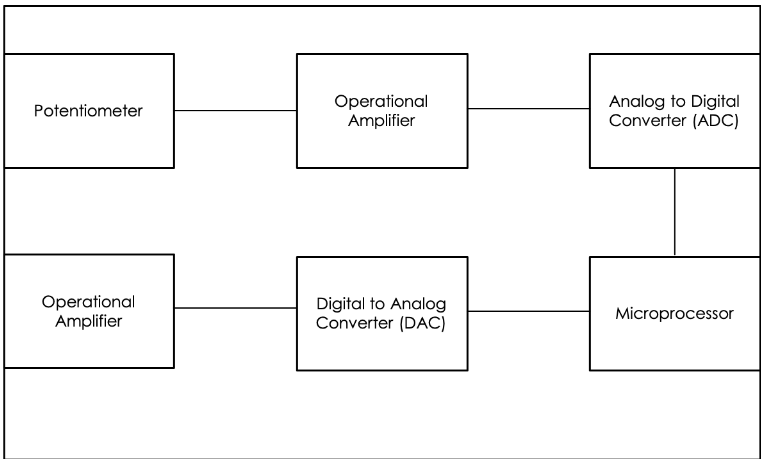

Figure 3.1: Components of the rudder force simulation device

3.1 Potentiometer

As explained in chapter 2.3.1, the potentiometer is used to detect the rotary movement of the shaft that is controlled by the pedals. The variable resistor is moved by the shaft changing the resistance of the potentiometer. Since the most common situation will be the absence of force in the pedals, the potentiometer should stay in the stand-by position without error. In this application a high precision potentiometer, which is analog and single-turn is used.

The input terminals of the potentiometer are connected to $U_0 = 5$ Volts and to ground. Since the potentiometer starts in a middle position and the shaft can rotate a maximum value of 30 degree en each direction, the maximum shaft rotation is $\Delta\phi = 60°$. The maximum angle that the potentiometer can detect is $\phi_{max} = 340°$. Therefore, the maximum voltage output of the potentiometer is $U_{pot} = 0.88$ Volts.

$$U_{pot} = U_0 \frac{\Delta\phi}{\phi_{max}} \tag{3.1}$$

3.2 Operational Amplifier

In order to maximize the accuracy of the analog to digital converter by using its full resolution, the output voltage of the operational amplifier is set to 5 Volts, maximum input voltage of the analog to digital converter. Since the U_{pot} is 0.88 Volts the multiplying factor of the operational amplifier should be $\nu = 5.68$. The operational amplifier is used in a simple non-inverting amplifier circuit shown in figure 3.2. The multiplying factor of the circuit is shown in equation 3.2. The chosen value of the resistor R_1 is $100k\Omega$ and by applying the equation 3.2 the resulting value of R_2 to get a maximum voltage close to 5 Volts is $470k\Omega$. As explained in section 2.3.2 the highest voltage that the operational amplifier can output is 90% of the supply voltage. Therefore, the operational amplifier is connected to ± 12 Volts to support a maximum voltage of 5 Volts. The voltage ± 12 Volts is chosen because this voltage is needed in the second operational amplifier and the controller has two pins that can output this voltage.

$$\nu = 1 + \frac{R_2}{R_1} \tag{3.2}$$

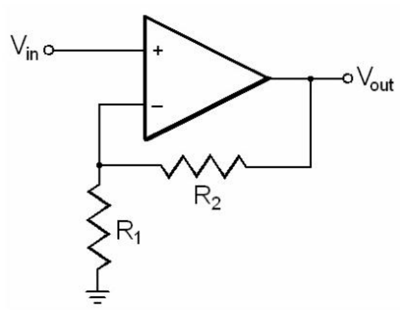

Figure 3.2: Non-inverting amplifier

3.3 Zener diode

Since the pedals shaft could only rotate less than $\Delta\phi = 60^\circ$ the operational amplifier was used to amplify the signal from the potentiometer. In this case, the maximum value of the input of the ADC would be 5 Volts. However, in the case of a broken cable or a wrong read of the potentiometer the input could be higher than 5 Volts until almost 12 Volts, the supply voltage of the operational amplifier. In order to prevent a fatal accident that could destroy the Intel Galileo board, a diode Zener is used as a safety component.

This component is situated after a $1k\Omega$ resistor from the input voltage of the Intel Galileo to ground in the opposite direction (cathode to anode). In this position, it limits the voltage that arrives to the Intel Galileo board from $-V_f$ to V_r, two characteristics of the Zener diode. Firstly, it was thought to use a Zener diode with $V_r = 5.1$ Volts but the error in the voltages close to 5 Volts was too big for a precision device. Since it would only act in unusual occasions and the Intel Galileo can support until 5.5 Volts, a Zener with $V_r = 5.6 Volts$ and $V_f = 0.7 Volts$ was selected.

3.4 Analog to digital converter (ADC)

The analog to digital converter (ADC) used in the device is the AD7298, which is included in the single board computer (SBC) Intel Galileo. The AD7298 is a 12 bit and 8 channel converter. However, in this case just one channel is used. The resolution can be calculated with the equation 2.4 and is equal to 12.2 mV. Hence, a variation of 12.2 mV would change the least significant bit (LSB). Figure 3.3 shows the conversion from the analog input and the binary code of the AD7298 and the relation of resolution with the LSB.

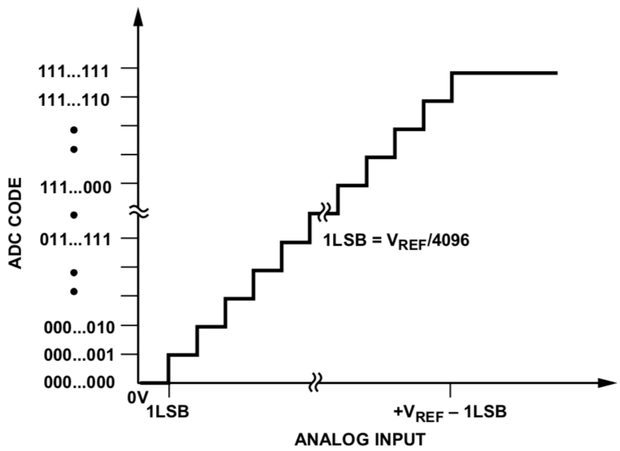

Figure 3.3: Binary Transfer Characteristics (Analog Devices, 2011)

The output code is transmitted to the microprocessor of the SBC with the SPI library. The Arduino code to read the analog signal consists of a function that changes the resolution to 12-bit as the function Analogread() is by default 10-bit and a function Analogread() to read the analog value. Since the resolution is 12-bit the analog value will be a number between 0 and 4095.

In figure 3.4 the timing of the transfer of the code to the microprocessor is explained. The falling edge of the CS signal starts the data transfer. The transfer takes 16 SCLK cycles, which are for one signal t_{SCLK} = 50 ns each plus the SCLK set time $t_2 = 10$ ns. Therefore, the $t_{conv} = 810$ ns (Analog Devices, 2011). The conversion time is really important because the frequency of the input signal should be higher than $2f_{conv} = 1.23$ MHz to avoid aliasing as explained in the section 2.3.4.

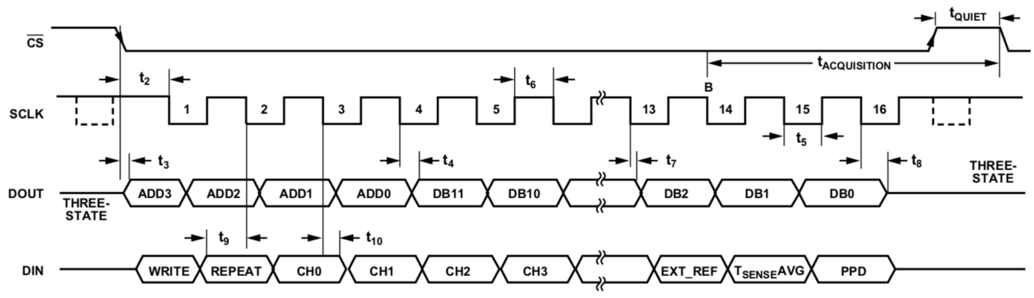

Figure 3.4: Serial Interface Timing Diagram (Analog Devices, 2011)

3.5 Single board computer (SBC)

The SBC used for the simulation of the rudder force was the Intel Galileo Gen 1. The three main reasons why this SBC was chosen were that it includes an ADC converter of 12 bits as explained in section 3.4, it has an Ethernet input and it is compatible with the Arduino software. The Intel Galileo Gen 1 uses an Intel Quark SoC X1000 processor as a system on a chip (SoC) to carry out the instructions that are given to it by the programming and the inputs. The connections of the Intel Galileo Gen 1 are:

- 14 digital input/output pins, which 6 are PWM compatible ports
- 6 analog inputs, which measure from ground to 5 Volts
- I^2C bus with SCA and SCL pins
- UART serial port
- ICSP header with SPI communication
- Ethernet port
- RS-232 serial port
- Micro-SD slot
- USB Client and USB Host port
- mini-PCI Express slot

The ethernet port is used to exchange information between the flight simulator software and the SBC. In the prototype by Thomas Krauss a RS-232 cable was used as an input of variables of the flight simulator. One of the motivations of this project is having an output of important variables in order to see what information is going in the SBC and if the SBC has processed it and sent the right output. This feedback should improve the life of the device as it is easier to detect if a component is broken.

3.6 Control algorithm in Arduino

The programming of the rudder force in Arduino is based in two variables, gain and displacement, which were explained in section 2.1. In the first version of the program, these variables can be changed inside the program and when the device is connected to the flight simulator, these values will be given by the software by an ethernet. The security output, whose function is to access the data of the device in case of failure, is the gain, displacement, the input of the potentiometer, 12 bit numerical value, and the output the device, 16 bit numerical value. All of them are displayed in the Arduino monitor of the personal computer connected to the Intel Galileo by an USB cable and when the device is connected to the flight simulator, the values will be sent by ethernet.

The input from the potentiometer is read with the function AnalogRead() and converted into a 16 bit number. With this number, calculations are made with the two variables, gain and displacement in order to output the correct value to the digital to analog converter. The transfer of this value is done in two parts, the most significant byte is transferred first and then the least significant byte is transferred. Therefore, the output value is divided in two values, one byte each, and then both are transferred to the converter.

As shown in section 4.2.2, the program achieves a rudder force simulation according to the values of the two variables. The safety mechanism of the software controls that the input of the controller is never out of the range of the controller of ± 10 Volts.

3.7 Digital to analog converter (DAC)

In order to transform the digital output of the SBC to an analog input of the motor controller a LTC1655 converter was used. The LTC1655 has a 3-wire serial interface compatible with the SPI library (Linear, 1998). In figure 3.5 a basic code in Arduino is shown, whose purpose is to connect and to transfer the digital code from the SBC to the DAC.

Firstly, variables are declared and the SPI transfer started. Secondly, the slave is set to low to active it and the most significant byte and least significant byte, consisting both of them of 8 bits, are calculated and transferred. Lastly, the slave is deactivated by setting it to high.

```
#include <SPI.h>

const int slaveSelect = 10;
unsigned int Dig_value;         //Dig_value 0-65536
unsigned int MSB;
unsigned int LSB;

void setup() {
  SPI.begin();
  SPI.setDataMode (SPI_MODE0);  //Mode 0: CPOL=0, CPHA=0 / (0,0)
  pinMode(slaveSelect, OUTPUT);
}

void loop() {
  MSB = highByte(Dig_value);
  LSB = lowByte(Dig_value);
  digitalWrite(slaveSelect,LOW); // Slave activation
  SPI.transfer(MSB);
  SPI.transfer(LSB);
  digitalWrite(slaveSelect,HIGH); // Slave desactivation
  delay(10);
}
```

Figure 3.5: Connection of the DAC with the SBC by the SPI library

The connections used in the SBC were the digital pin 13 as serial clock, pin 11 as master out slave in and pin 10 as slave select. In this order, these pins were connected to pin 1, 2 and 3 of the LTC1655 converter. The timing diagram in figure 3.6 shows how the three pins act during the transfer of data and the timing of the output signal. The first 8 changes of the clock the most significant byte of the code is transferred while the last 8 changes of the clock the least significant byte is transferred.

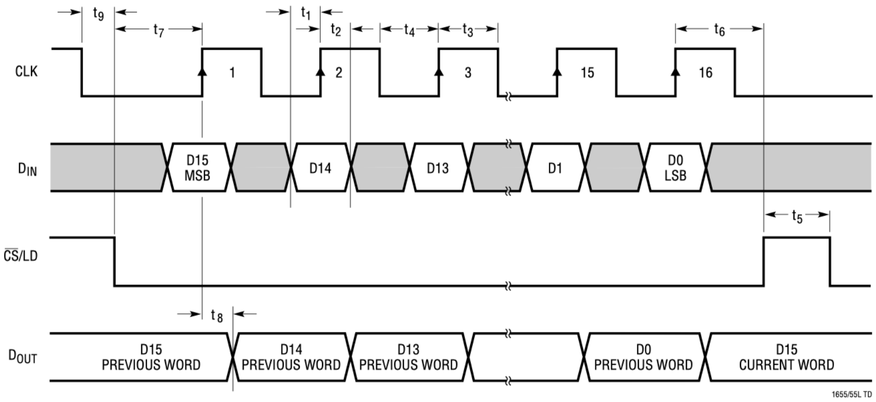

Figure 3.6: DAC Serial Interface Timing Diagram (Linear, 1998)

The LTC1655 has a reference voltage $V_{ref} = 2.048$ Volts and the output of the DAC varies from 0 Volts to 4.096 Volts. However, the input voltage of the motor controller must be ± 10 Volts. In order to achieve this output using the reference voltage, a typical application circuit, which is explained in the data sheet of the DAC, was modified. This circuit is shown in figure 3.7 and it was modified the value of R_2 and R_4 to $470k\Omega$ since the output of the circuit of the figure is ± 4.096 Volts. The power supply of the operational amplifier was changed from ± 5 Volts to ± 12 Volts to permit an output higher than ± 5 Volts.

3.8 Circuit board

Figure 3.8 shows the layout of the complete circuit board, which can be printed in a printed circuit board (PCB) in order to make a final version more reliable than the stripboard used in the last experiment. This experiment is explained in section 4.1.2. The design has capacitors to stabilize the input voltages and the power supplies. The input of the device is the voltage of the potentiometer, which is named U_{pot} and the output, which is named U_{out}, is the input of the motor controller. The connections of the device with the motor controller are explained in section 5. The bipolar 12 Volts voltage

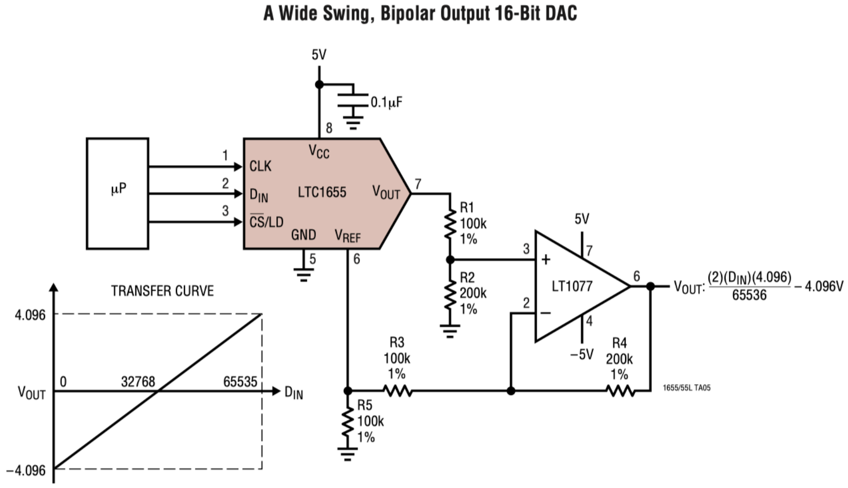

Figure 3.7: Typical Application of the LTC1655 (Linear, 1998)

can be supplied by the motor controller and the 5 Volts power supply can be supplied by the Intel Galileo board.

3.9 Motor controller

The flight simulator has a XtrapulsCD1-a-400/30 servo drive to provide speed control to the shaft of the pedals. This digital controller shown on figure 3.9, has a 400 Volts power supply and an adjustable maximum motor speed of 25000 rpm (Infranor, 2013b). The maximum peak current for 1 second is 30 A while the maximum load current is 15 A (operating in normal circumstances). This current is limited digitally to 9.77 A to match the motor stall current, one of the characteristics of the motor shown in table 3.1.

The connection is made with the X2 connector to the circuit board and with the X5 connector, which are positioned in the controller as figure 3.10 shows. In figures 3.11 and 3.12 the configuration of pins of the connectors is shown.

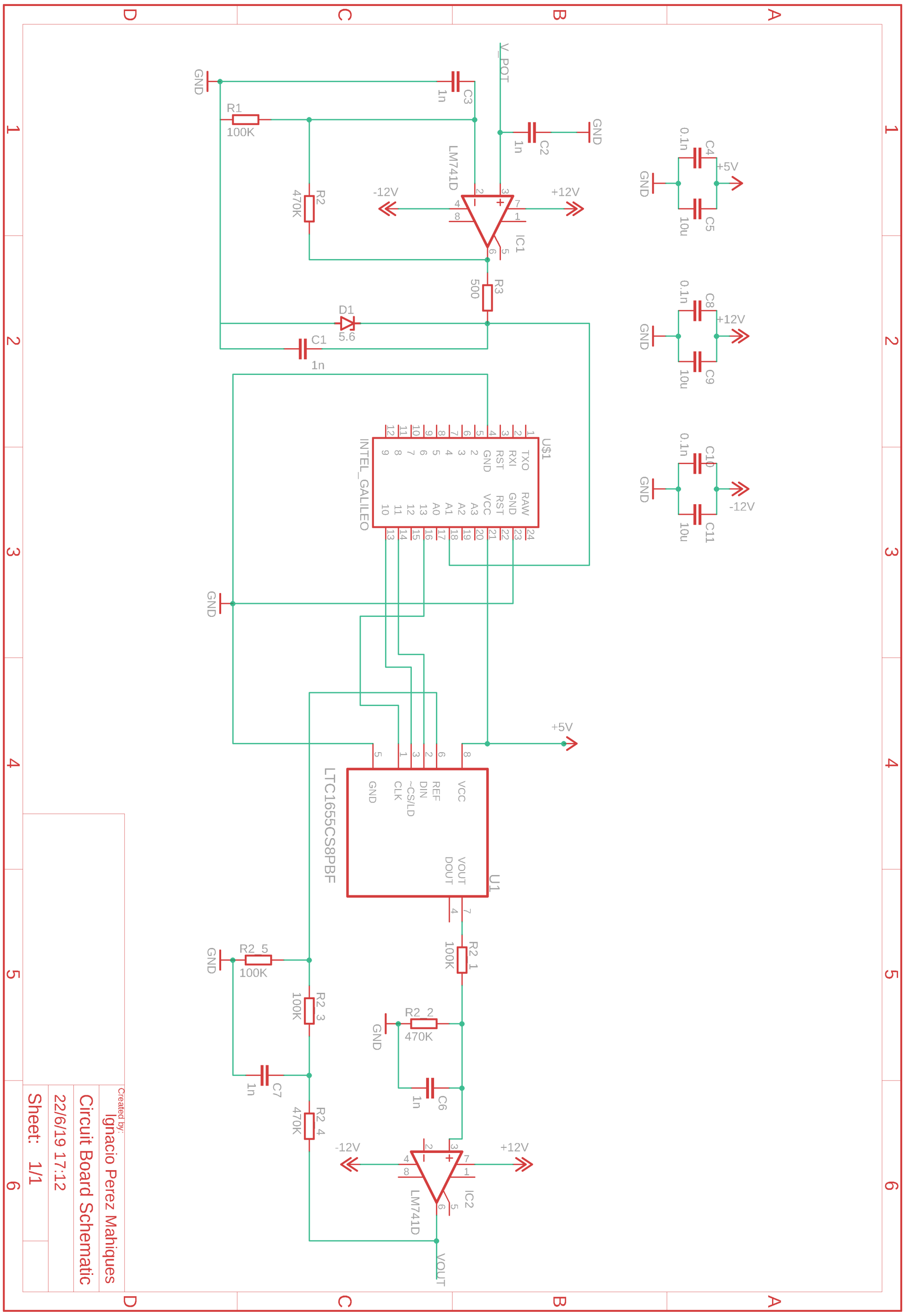

Figure 3.8: Schematics of the circuit board

Figure 3.9: Digital servo drive CD1-a-400 (Infranor, 2013b)

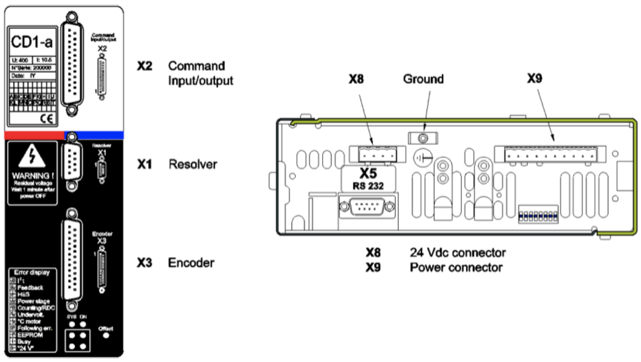

Figure 3.10: Available connectors of the CD1-a-400 drive (Infranor, 2013b)

3 – X2: COMMAND CONNECTOR, LOGIC INPUTS-OUTPUTS AND ENCODER (Sub D 25 pins male)

Same connector for both 230 V and 400 V ranges.

Pin	Function	I / O	REMARKS
1	FC+: Limit switch +	I	Positive logic, optocoupled input, galvanic insulation
14	FC-: Limit switch -	I	Positive logic, optocoupled input, galvanic insulation
24	Ref. inputs	I	Supply reference of the galvanic insulated logic inputs
20	ENABLE	I	Positive logic, optocoupled input, galvanic insulation
23	Ref. inputs	I	Supply reference of the galvanic insulated logic inputs
2	Current command CI	I	Positive logic, optocoupled input, galvanic insulation
10	CV0 Zero speed / Motor phasing input command	I	Positive logic, optocoupled input, galvanic insulation
25	GND	I	GND reference of the grounded drive
13	RESET	I	Positive logic, optocoupled input, galvanic insulation Inhibition of the faults memory stored in the drive
12	Ref. inputs (0 Volt)	I	Supply reference of the galvanic insulated logic inputs.
17	CV+ Input command CV +	I	± 10 V speed input command for max. speed
16	CV- Input command CV -	I	or ± 10 V current input command for Imax with "CI" input active
15	GND	I	GND reference of the earthed drive
3	I limit current limitation	I	Analog input for external max. current limitation 0 to 10 V for 100 % to 0 % of Imax
11	Analog output	O	+/-10 V, resolution: 8 bit, load: 10 mA, linearity: 2 %, low pass filter: 170 Hz, programmable output signal: Channel 1 of the digital oscilloscope (current, speed or position) or Phasing OK output (from 0 V to 10 V when the motor phasing is OK for an incremental encoder without HES)
18, 19	AOK: drive ready	O	Relay contact: closed if drive OK, open if fault. Protection against overvoltages by bidirectional TRANSIL Pmax = 10 W with Umax = 50 V or Imax = 100 mA
21	+ 12 Volts	O	Output impedance: 9 Ohms. Max. 150 mA available(1)
22	- 12 Volts	O	Output impedance: 47 Ohms. Max. 50 mA available
4	Z/	O	Differential output of Z/ encoder marker pulse (max. 5 V, 20 mA)
5	Z	O	Differential output of Z encoder marker pulse (max. 5 V, 20 mA)
6	A/	O	Differential output of encoder A/ channel (max. 5 V, 20 mA)
7	A	O	Differential output of encoder A channel (max. 5 V, 20 mA)
8	B/	O	Differential output of encoder B/ channel (max. 5 V, 20 mA)
9	B	O	Differential output of encoder B channel (max. 5 V, 20 mA)

(1): The sum of the currents consumed by both X2 connector, pin 21, and X3 connector, pin 10, must not exceed 150 mA.

Figure 3.11: Pins information of the X2 connector (Infranor, 2013b)

5 - X5 SERIAL LINK (Sub D 9 pins male)

Same connector for both 230 V and 400 V ranges.

PIN	FUNCTION	REMARKS
5	0 Volt	GND (connection of the shield if no "360°" connection on the connector)
3	TXD	Transmit data RS-232
2	RXD	Receive data RS-232
6	TXH	Transmit data RS-422
7	TXL	Transmit data RS-422
8	RXL	Receive data RS-422
9	RXH	Receive data RS-422

Figure 3.12: Pins information of the X5 connector (Infranor, 2013b)

Characteristics	Value	Units
Max. Speed (±10%)	4600	rpm
Stall Torque (±10%)	13.2	Nm
Stall Current (±10%)	9.77	A
Nominal Torque at 3000 rpm	10.75	Nm
Peak Torque (±10%)	79.2	Nm
Torque constant (±5%)	1.35	Nm/A
Rotor inertia (J)	2.85	$kgm^2 10^{-3}$
Axial Force	200	N
Radial Force	685	N
Weight	9.6	kg

Table 3.1: Main characteristics of the FP-1311 at 400V (Infranor, 2013a)

3.10 Servo motor

The servo motor FP-1311, shown in figure 3.13, is used to move the shaft that provides the feedback force on the pedals. It provides a maximum speed at 400 Volts of 4600 rpm, a stall torque of 13.2 Nm and a stall current of 9.77 A. The torque can increased until the peak value of 79.2 Nm (Infranor, 2013a). Other important characteristics of the motor are shown in table 3.1.

Figure 3.13: Motor FP-1311 (Infranor, 2013a)

4 Evaluation

4.1 Execution of the experiments

The proposed system explained in chapter 3 was checked in two types of experiments. The first experiment was putting together the circuit in a breadboard and simulating the movement of the pedals with a potentiometer and the second one was soldering the components into a stripboard and controlling the output voltage by changing the gain and the displacement.

4.1.1 Breadboard experiment

The first step after defying the components of the system was to test them with the simplest method in order to adjust the components and make sure to have a fully working system before testing it in the flight simulator. The first component to test was the operational amplifier with the potentiometer. As explained in chapter 3.3, a Zener diode was needed to ensure that the voltage would not go over 5 Volts if a cable from the potentiometer would break or the signal was wrong. The second component to test was the DAC with the ± 10 Volts amplified output. The DAC received the signal from the Intel Galileo board by the SPI connection and the desired output voltage was achieved with the help of the second operational amplifier of the system.

When both parts of the circuit worked properly, the whole system was tested with a more complex Arduino program. The objective of the experiment was to achieve a ± 10 Volts output depending on the position of the potentiometer. Figure 4.1 shows the whole setup for the experiment with all the connections between components. In figure 4.2 is shown how the Intel Galileo board was connected to the breadboard with 5 cables, the analog input, the ground cable and the 3 cables that permit the digital output to the DAC.

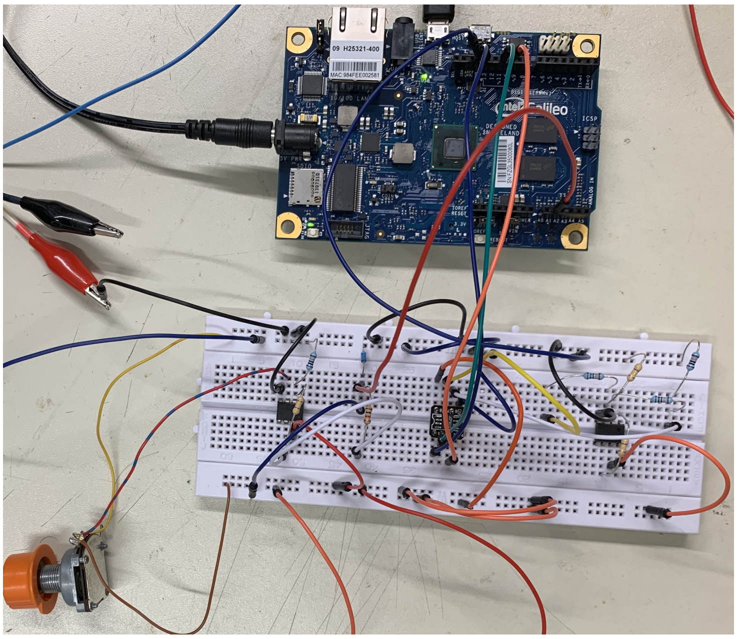

Figure 4.1: Test of the system in the breadboard

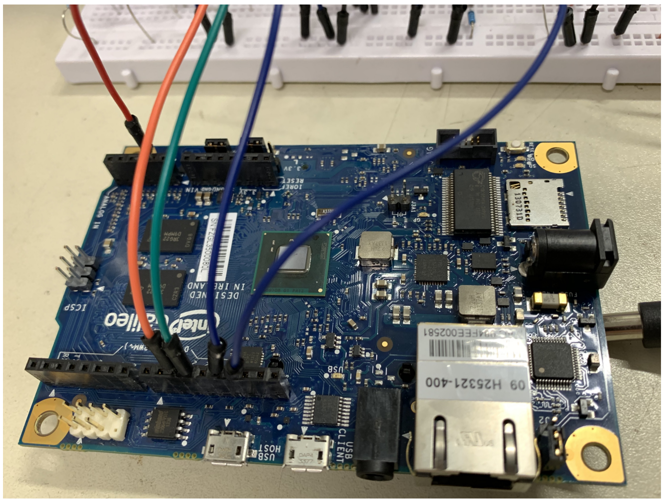

Figure 4.2: Close-up in the connections of Intel Galileo

4.1.2 Stripboard experiment

After checking the right functioning of the system in the breadboard, all the components were soldered into a stripboard. A stripboard is a rectangular grid of holes with strips of copper in both directions of the board but only in the rear side. In the front side, the components are placed and the pins of the components and the cables are soldered in the rear side (Bishop, 2001). This type of board is used to prototype circuits, which should stay permanently assembled, before a final version in a printed circuit board (PCB) is made. Figures 4.3 and 4.4 show the connections of the spring board.

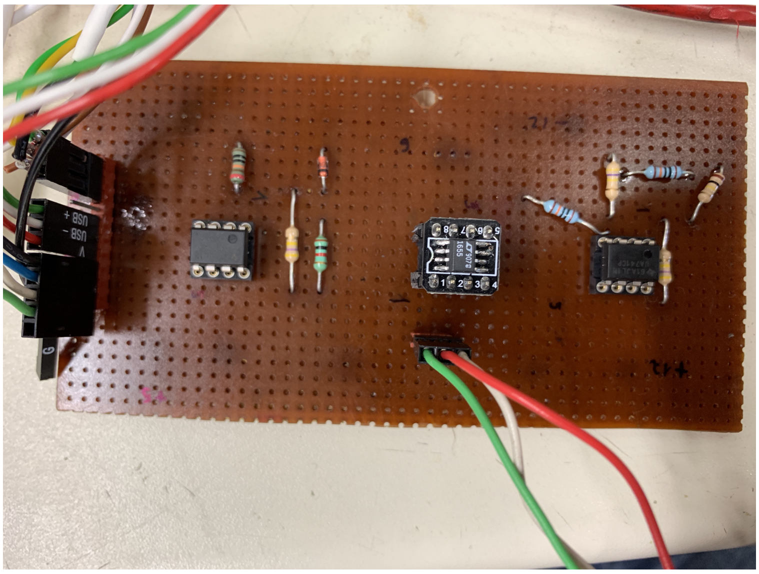

Figure 4.3: Front side of the stripboard

When the connections were checked, an experiment with the stripboard and the full program on the Arduino was made. This experiment tested the variable gain and displacement of the rudder force explained in section 2.1. It also checked the safety mechanism by simulating a bad input of the potentiometer. In this experiment the maximum current of the power supply was measured in order to assure the supply from the X2 connector of the controller. The Intel Galileo was used to output the 5 Volts supply for the potentiometer and the DAC. A external power supply was used to power the operational amplifiers with a voltage of ±12 Volts.

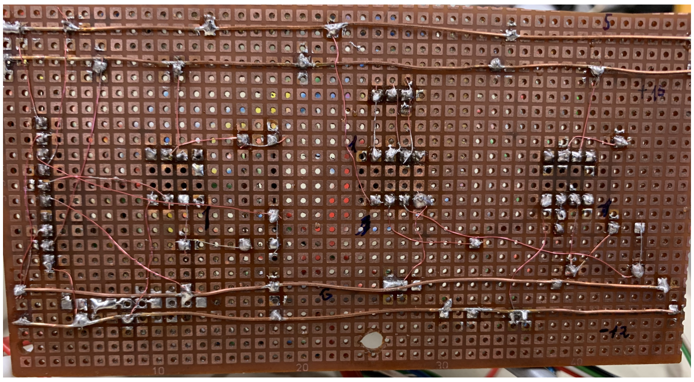

Figure 4.4: Rear side of the stripboard

4.2 Results

4.2.1 Breadboard experiment

Figure 4.5 shows the results of the breadboard experiment. The blue line is the expected lineal output of ± 10 Volts while the red points are the experimental values, which were measured in the output of the device. The experiment consisted in turning 10 degree of the potentiometer each time and reading the value that will be connected to the controller. The experimental value is due to the approximation of the standard value of the resistors, the error in the processing of the signal and in the power consumption of the resistor that is necessary for the connection of the Zener diode.

4.2.2 Stripboard experiment

Four different tests were conducted in order to check the correct functioning of the final circuit and of the program done in the Arduino IDE, which is explained in section 3.6. Since the experiment is simulating the pressing of one pedal, the plots take the middle position of the potentiometer (30 degrees) as starting position (0 degrees). Therefore, the maximum angle of rotation of the shaft is 30 degrees.

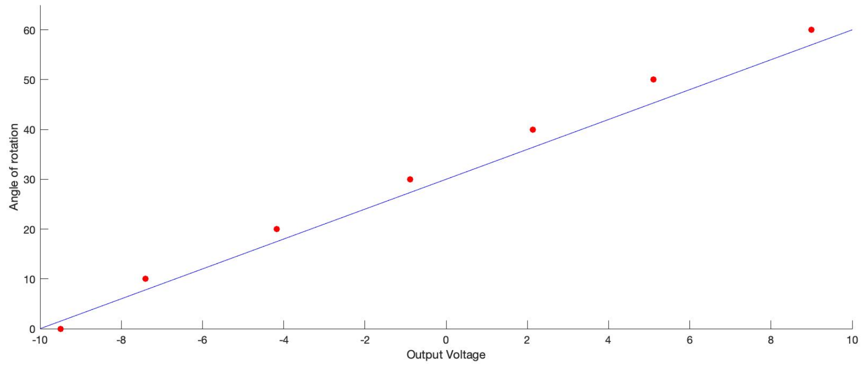

Figure 4.5: Plot of the results of the experiment in the breadboard

Two experiments were done to prove the change of the steepness when the gain varied. Figure 4.6 shows the comparison of a system with a gain of 50% and a system with a gain of 80%, both without displacement.

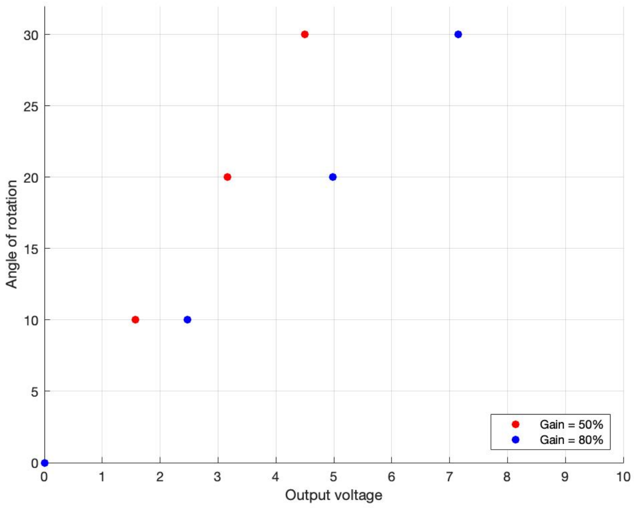

Figure 4.6: Plot of the modification of the gain experiments

In the third experiment a displacement of 25% was added. Figure 4.7 shows the comparison of this experiment with the first experiment, since both experiments had a gain of 50%. The comparison reflects a deviation of the X-axis of the experiment with displacement. Moreover, the linearity of the results is consistent in both experiments.

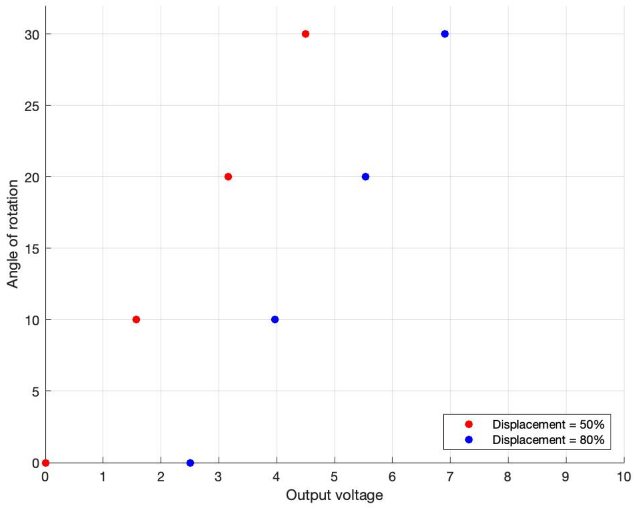

Figure 4.7: Plot of the modification of the displacement experiments

The last experiment checked the safety mechanism, which makes sure that the input of the controller is never bigger than 10 Volts. Figure 4.8 shows the results of increasing the angle of rotation of the potentiometer more than the maximum normal value, 30 degree. When the potentiometer indicates 35 degrees, it shows a different value than with 30 degrees. However, this value is not significant, since it will only appear in unusual circumstances. When the potentiometer reads values bigger than 40, the diode Zener protects the Intel Galileo board and the Arduino program cannot output a numerical value that could cause damage to the controller. The result will be equal if pressing the other pedal. Therefore, the maximum negative value of the controller will not be output by the device.

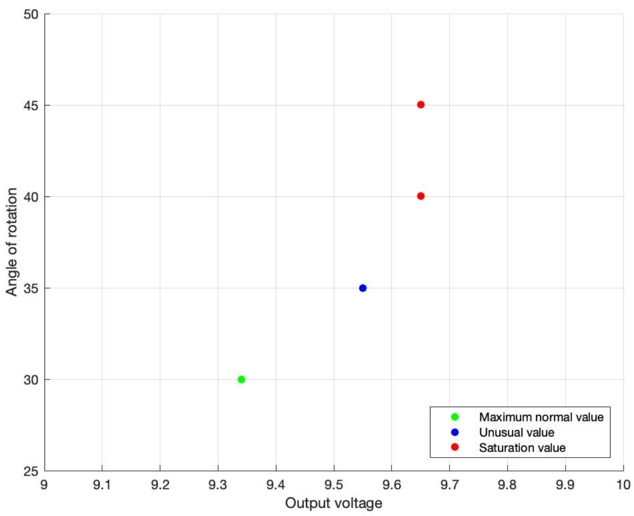

Figure 4.8: Plot of the results of the saturation experiment

5 Outlook

The objective of the project was to design a device that could recreate with fidelity the rudder force in a flight simulator. A working proof of concept, which accomplishes all the requirements of the project, was created and tested. However, the lack of time did not permit to connect it to the flight simulator.

In order to connect this device to the flight simulator controller, the X2 connector can be used. Using the information from the datasheet of the controller, which is shown in figure 3.11, the powering of the operational amplifiers in the stripboard is supplied by the pins 21, 22 of the X2 controller. The powering of the operational amplifiers can be supplied with the external power supply used in the old device. The X2 controller can be also used to connect the ground (GND) and the input command to the pins 25 and 17 respectively. The Intel Galileo allows an output of 5 Volts supply for the potentiometer and the DAC. The grounds of the controller, Intel Galileo and the board should be connected together. A printed circuit board (PCB) should be designed to create a more secure connection.

The two parameters used in the programming of the Intel Galileo, gain and displacement, should be read by an Ethernet connection. The serial output, which is now sent to the personal computer connected to the Intel Galileo, would be sent to the flight simulator software using the Universal Data Package (UDP) of the Ethernet connection.

An improvement of the device can be to change the potentiometer for a higher precision potentiometer or for an optical rotary encoder. This could improve the quality of the input signal, which is the most important input parameter of the device. Improving the safety could be done by creating a safety voltage supply for the controller in case of a failure or by creating a similar device that would make the same calculations and stop the simulation when the two devices output different values.

All in all, during the project, all the components from the device by Thomas Krauss were examined in order to improve their performance. The objective of increasing fidelity is achieved by improving the components and by creating a new software program. The security output of information will facilitate the understanding of the functioning of the device and will help to determine the broken component in case of a failure. Moreover, the new device accomplishes the requirements of the motor controller to simulate the rudder force feedback.

Bibliography

Analog Devices (2011). "8-Channel, 1 MSPS, 12-Bit ADC AD7298 Data sheet." In: (cit. on pp. 22, 23).

Arduino - SPI (2015). URL: https://www.arduino.cc/en/reference/SPI (visited on 02/05/2019) (cit. on p. 14).

Bishop, Owen (2001). "Understand Electronics." In: ed. by Owen B T - Understand Electronics (Second Edition) Bishop, pp. 144–153 (cit. on p. 35).

Burns, Roland (2001). *Advanced Control Engineering [electronic resource].* Burlington: Burlington : Elsevier Science, 2001. (cit. on p. 7).

Clark, Liz (2019). *Practical Tinker Board [electronic resource] : Getting Started and Building Projects with the ASUS Single-Board Computer*. Berkeley, CA : Apress : Imprint: Apress, 2019. (cit. on p. 13).

Clayton, G B (George Burbridge) (2003). *Operational amplifiers [electronic resource]*. Ed. by Steve Winder. 5th ed.. Oxford (cit. on p. 11).

Crowe, J (2005). *PID Control [electronic resource] : New Identification and Design Methods*. Ed. by K K Tan et al. London : Springer London, 2005. (cit. on p. 8).

Federal Aviation Administration (FAA) (2016). "14 CFR Part 60: Flight Simulation Training Device Inıtial and Continuing Qualification and Use." In: Volume 60 (cit. on p. 1).

Hess, Ronald A. (2005). "Rudder Control Strategies and Force/Feel System Designs in Transport Aircraft." In: *Journal of Guidance, Control, and Dynamics* 28.6, pp. 1251–1262. ISSN: 0731-5090. DOI: 10.2514/1.12632 (cit. on p. 5).

Infranor (2013a). "SERVICE MANUAL XtraforsPrime." In: (cit. on p. 31).

Infranor (2013b). "XtrapulsCD1-a Digital drive for AC sinusoidal brushless motors." In: (cit. on pp. 27, 29, 30).

Kumar, K Udaya and B S Umashankar (2008). *8085 Microprocessor : Architecture, Programming and Interfacing*. Noida: Noida: Pearson India (cit. on p. 13).

Linear (1998). "LTC1655/LTC1655L - 16-Bit Rail-to-Rail Micropower DACs inSO-8 Package." In: pp. 1–16. URL: http://www.linear.com/product/LTC1655 (cit. on pp. 25, 27).

Maini, A K (2007). *Digital Electronics: Principles, Devices and Applications*. Wiley. ISBN: 9780470032145 (cit. on pp. 12–15).

Perea, Francis (2015). *Arduino essentials : enter the world of Arduino and its peripherals and start creating interesting projects*. Birmingham, England : Packt Publishing Ltd, 2015. (cit. on p. 13).

Robinson, Andrew, Katerina Mania, and Philippe Perey (2004). "Flight simulation: research challenges and user assessments of fidelity." In: pp. 261–268 (cit. on pp. 2, 3).

TU Graz Flight Simulator (n.d.). URL: https://www.tugraz.at/en/kooperationen/flugsimulation/simulators/widebody-aircraft/ (visited on 04/06/2019) (cit. on pp. 15, 16).

Villamil Rico, Luis Carlos, Edna Joydeth Avella Rodríguez, and Jorge Antonio Tenorio Melo (2018). "Simuladores de vuelo: una revisión." In: *Ciencia y Poder Aéreo* 13.2, pp. 138–149. ISSN: 1909-7050. DOI: 10.18667/cienciaypoderaereo.606 (cit. on p. 1).