Universidade de Aveiro Departamento de Engenharia Mecânica
2010

Emanuel da Silva Ávila

Servopiloto para condução autónoma

Servo-pilot for autonomous driving

Universidade de Aveiro Departamento de Engenharia Mecânica
2010

Emanuel da Silva Ávila

Servopiloto para condução autónoma

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor Vítor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro e do Doutor Jorge Augusto Fernandes Ferreira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Universidade de Aveiro Departamento de Engenharia Mecânica
2010

Emanuel da Silva Ávila

Servo-pilot for autonomous driving

Dissertation presented to University of Aveiro in fulfillment of the requirements for the of Degree Masters of Science in Mechanical Engineering, with Ph.D. Vítor Manuel Ferreira dos Santos, Associate Professor with the Mechanical Engineering Department at University of Aveiro and Ph.D. Jorge Augusto Fernandes Ferreira, Assistant Professor with the Mechanical Engineering Department at University of Aveiro as scientific advisors.

Palavras-chave

Condução autónoma, *micro-electromechanical systems* (MEMS), modularidade, sistemas inerciais.

Resumo

Este projecto enquadra-se numa actividade que tem como objectivo o desenvolvimento de meios que permitam a um carro ter percepção das acções do condutor sobre os comandos do veículo, e posteriormente controlar o veículo em situações de navegação. A primeira fase deste projecto irá permitir a monitorização dos comandos do veículo, como os pedais, e será suficientemente flexível para que possa ser instalado e removido facilmente. Não necessitará de modificações no veículo para ser instalado, e poderá ser utilizado em aplicações que necessitem de medições semelhantes, como medições antropométricas durante terapia por electro-estimulação ou monitorização de movimento e equilíbrio em robôs humanóides. A escolha tecnológica direccionou-se para sistemas baseados em sensores inerciais como acelerómetros e giroscópios, devido ao seu preço atractivo, dimensões reduzidas e ausência de partes móveis, que melhoram a sua durabilidade. A novidade proposta consiste na utilização de vários sensores ligados entre si que permitam medições diferenciais para desacopolar movimentos locais da dinâmica global do veículo. As dimensões dos dispositivo desenvolvidos são muito inferiores às de todas as soluções comerciais. Por isso o projecto baseia-se em módulos que contêm acelerómetros e giroscópios MEMS ligados entre si por CANbus numa rede *master multi-slave* para a análise de movimento. Para actuação o projecto recorre a um PLC industrial ligado a reguladores de pressão electrónicos e cilindros pneumáticos para emular o comportamento humano. A rede de sensores provou funcionar bem em condições estacionárias, onde as acelerações dinâmicas são despresáveis, mas requer melhoramentos de *software* para lidar com situações de movimento.

Keywords Autonomous driving, micro-electromechanical systems (MEMS), modularity, inertial systems.

Abstract

This project is part of a broader activity that aims to develop a setup to allow a car to have perception about the driver's actions on the vehicle's controls and latter to control the vehicle in navigation situations. The first part of the project will allow monitoring the vehicle controls, such as pedals, and the setup will be flexible enough to be installed and removed easily. Also, it will not require vehicle modifications and can be used for other applications that require similar types of measurements, such as antropometry during electro-stimulation therapy or to monitor humanoid robot motion and balance. The technological choice has been directed to inertial-based sensors such as accelerometers and gyroscopes due to their attractive cost, small packaging size and absence of moving parts, which improves durability. One novelty on the proposed setup is that many inertial sensors can be connected in the same network and differential measurements can be done to decouple the local motion of parts from the vehicle global dynamics. Also, the size of the devices developed is much smaller and lighter than all commercial solutions found. For this purpose, the project relies on MEMS accelerometer and gyroscope modules connected in a master multi-slave network using CANbus to support the communications for motion analysis. On the other hand, the project includes an industrial PLC connected to a combination of pneumatic cylinders and electric pressure controllers to emulate the human foot actuation on the car control pedals. The network of inertial sensors proved to work well on near-stationary situations where the dynamic acceleration applied to all the modules is negligible. When subject to higher dynamic perturbations, the system requires further software improvements so it can perform adequately.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The ATLAS project

1.1.1 Overview

The ATLAS project was created to develop autonomous driving robots for competition. After the success achieved, the research group focused in applying the acquired knowledge in real scale vehicles, to start research and development in passive and active security systems. Bound by several restrictions, there was the need to interface a conventional car with software commands, as well as monitor human driver actions for active security purposes.

1.1.2 Framework

The ATLAS team develops its code based in the CARMEN robot navigation toolkit, from Carnegie Mellon university. All robot software modules are running simultaneously and communicate among them using IPC[15]. Currently, the AtlasCar1 platform (fig 1.1) is equipped with: one 200A alternator; one 3000W DC/AC inverter; one UPS; two desktop computers; various cameras; two LIDARs; and one GPS unit. All the data acquired by the system is saved to log files for later off-line analysis and can also be played in simulation mode to test new algorithms.

At this stage, the main issues concern the ability to control the vehicle using software instructions, monitoring the driver actions on the vehicle controls and monitor the driver awareness about the surrounding environment.

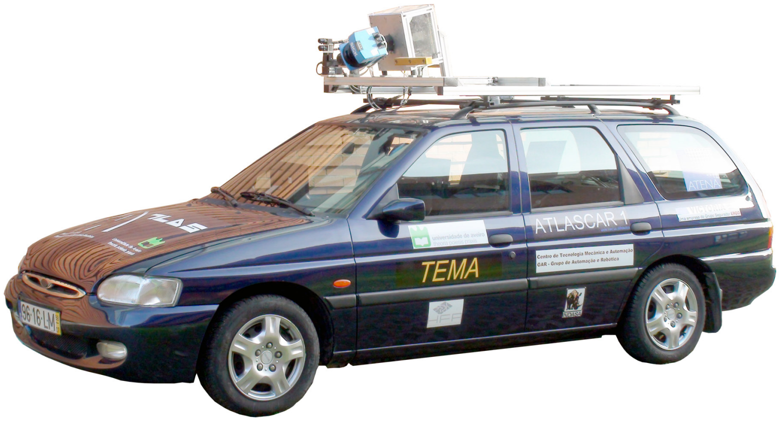


Figure 1.1: AtlasCar1

1.2 Challenges

A servo-pilot is a system responsible for maintaining an output condition specified by input parameters, using error sensing feedback to correct eventual errors between the input required condition and the actual output[18]. This system needs then to have feedback about the vehicle controls assigned to it, so it can compute the errors associated to each output. Monitoring the vehicle controls such as pedals, gear lever, wheel and others, for this project will require that no modifications are preformed to the vehicle structure, and for driver monitoring functions, it has to be able to operate separately from the servo-pilot remaining systems, and be discrete enough to have no interference with the human driver. The servo-pilot actuation unit has to be able to control the vehicle main controls such as pedals and steering wheel in the first stage and later others like gear lever, hand brake and turn indicators. This unit needs to be suited to easy installation and removal, so the vehicle can be driven to the test local and back, and has to be installed without any vehicle structural modifications, that would revogue the vehicle circulation license according to Portuguese legislation.

1.3 Project overview

The requirements for this project are: limited cost; easy to assemble, install and remove from the vehicle; should require no vehicle modifications.

For this project several approaches were considered:

For the perception, solutions based in encoders, potentiometers and inertial sensors like accelerometers and gyroscopes were analyzed. As both encoders and potentiometers require anchorage points in the moving part and in the vehicle structure, the options were discarded.

Inertial modules offer less resolution than the other methods but this can be compensated with redundancy; they can also be fixed to the vehicle without modifications. With that in consideration, an inertial module was developed based in ST microelectronics MEMS accelerometers and gyroscopes; taking advantage of its small size, the outcome was significantly smaller than all market available similar devices. For this application there is a particular interest in measuring the position of the vehicle pedals, wheel, gear lever and parking brake, among others. In first stage, and also due to the number of sensors available, only the pedals will be measured. The geometry of this vehicle allows to secure the sensors in the back of the pedals inside the grip rubber cover. For the following stages this setup will be extended to the gear lever, hand brake and the wheel. To decouple the component's motion from the vehicle's, one sensor must be fixed to the vehicle mainframe to serve as a reference and all other sensors are attached to the moving parts. Data is then compared with the fixed reference sensor, so differential measurement can filter out undesired global dynamic data.

Concerning the actuation, several types were studied such as electric linear, electric rotational and pneumatic. For this application, and given the space constraints, the only two options were the electric linear and pneumatic, as rotational types require structure modifications to attach to the vehicle. Given that pedals have spring elements responsible for their return to home position , force control is the simplest to implement, because the force applied by a spring is linearly related to its displacement and such control can be preformed by a regular PID controller which can act similarly to an human driver.

1.4 Mathematical principles

To obtain differential measures vector computing techniques are applied. Starting with a simplification in which equal acceleration modules are measured by the different measuring devices, the angle between them can be calculated by the equation 1.5, where v_1 and v_2 are two 3D vectors.

$$\vec{v_1} \cdot \vec{v_2} = |\vec{v_1}| \cdot |\vec{v_2}| \cdot \cos\beta \tag{1.1}$$

$$\vec{v_1} \cdot \vec{v_2} = v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} \tag{1.2}$$

$$|\vec{v_1}| = \sqrt{v_{1x}^2 + v_{1y}^2 + v_{1z}^2} \tag{1.3}$$

$$\cos\beta = \frac{\vec{v_1} \cdot \vec{v_2}}{|\vec{v_1}| \cdot |\vec{v_2}|} \tag{1.4}$$

$$\beta = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{|\vec{v_1}| \cdot |\vec{v_2}|}\right) \tag{1.5}$$

1.5 State-of-the-Art

A variety of inertial sensors and modules can be found in the market. Typically with three, six or nine degrees of freedom (DOF), as they contain accelerometer, gyroscope or magnetometer, as well as combination of these units up to three. However most of them are not compatible with our application due to size constraints.

There are some systems available in the market that can interface a standard car with software controls, but usually these devices have complex control protocol and require permanent vehicle modification.

Description for these systems can be found in the next section.

1.5.1 Inertial measurement systems

Analog devices ADIS16405

This unit produced by Analog Devices (fig 1.2) has nine DOF being three accelerations, three angular rates and three magnetic readings. The main features include: SPI interface and embedded

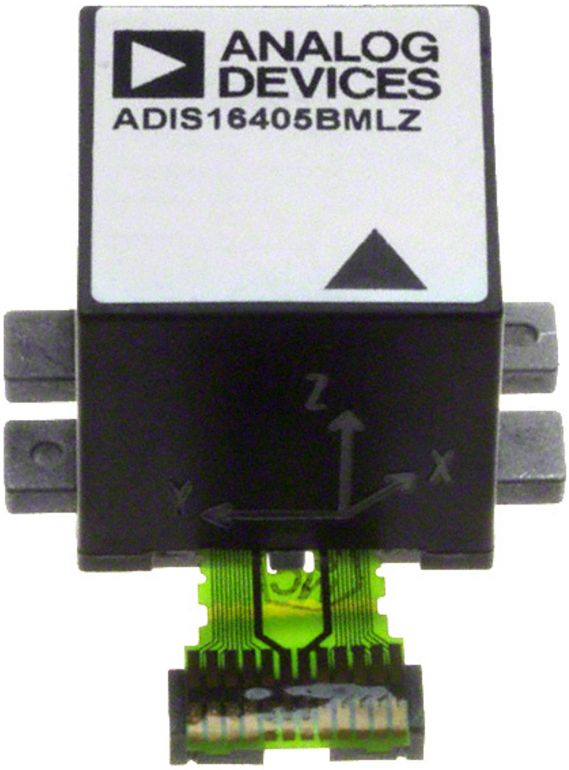


Figure 1.2: Analog devices ADIS16405

temperature sensor for auto compensation, accelerometer scale up to 18 g, angular rate up to 350°/sec and magnetometer up to 2.5 Gauss[6]. This device is presented in 23mm cubic housing[6] and costs around 450€[7].

Xsens MTi

This unit produced by Xsens (fig 1.3) is similar to the ADIS16405, previously referred, being the main difference the use of embedded DSP capable of processing the sensor data and generating 3D orientation output as well as raw data from each sensor. The main features include: RS-232, RS-485 and RS-422 interface, accelerometer scale up to 5 g, angular rate up to 300°/sec and magnetometer up to 0.75 Gauss[1]. This device is presented in 58x58x22mm housing[1] and costs around 3000€.

Figure 1.3: Xsens MTi

Figure 1.4: MEMSense nIMU

MEMSense nIMU

MEMSense produces one of the smallest IMUs available in the market (fig 1.4). It has 9 DOF, similarly to the previous devices and comes with asynchronous serial interface and I^{2}C™ as well. It can read accelerations up to 10 g, angular rate up to 300°/sec and magnetometer up to 1.9 Gauss[11]. This device is presented in 46x23x12mm housing[11] and costs around 1100€[12].

MicroStrain® 3DM-GX3™-25

This device (fig 1.5) combines most of the features from the Xsens MTi in a package almost as small as the MEMSense nIMU. Main features include: RS-232, USB 2.0 and TTL serial interface, temperature compensated over entire operational range, data rate up to 1 kHz, outputs Euler angles, rotation matrix, delta angle, delta velocity, acceleration up to 5 g, angular rate up to 300°/sec and magnetic field up to 1.9 Gauss[9]. This device is presented in 38x24x12mm housing[9] and costs around 1500€[10].

3DM-GX3™-25-OEM
Attitude Heading Reference System
#6223-5460.
Y
X
Z
MicroStrain® www.microstrain.com

Figure 1.5: MicroStrain® 3DM-GX3™-25

Inertial measurement systems summary

Table 1.1 shows a comparison of the systems referred before.

Table 1.1: Comparison of different IMUs

Device Spec	ADIS16405	Xsens MTi	MEMSense	3DM-GX3	Unit
Gyro scale	±350	±300	±300	±300	°/sec
Gyro linearity	0.1	0.1	0.1	0.2	% of FS
Accelerometer scale	±18	±5	±5	±5	g
Accelerometer linearity	0.1	0.2	0.4	0.2	% of FS
Magnetometer scale	±3.5	±0.75	±1.9	±2.5	Gauss
Magnetometer linearity	0.5	0.2	0.5	0.4	% of FS
Package	23x23x23	58x58x22	46x23x12	38x24x12	mm
Interfaces	SPI	RS-232 -485 -422	I2C USART	USB 2.0 RS-232	
Price	450	3000	1100	1500	€

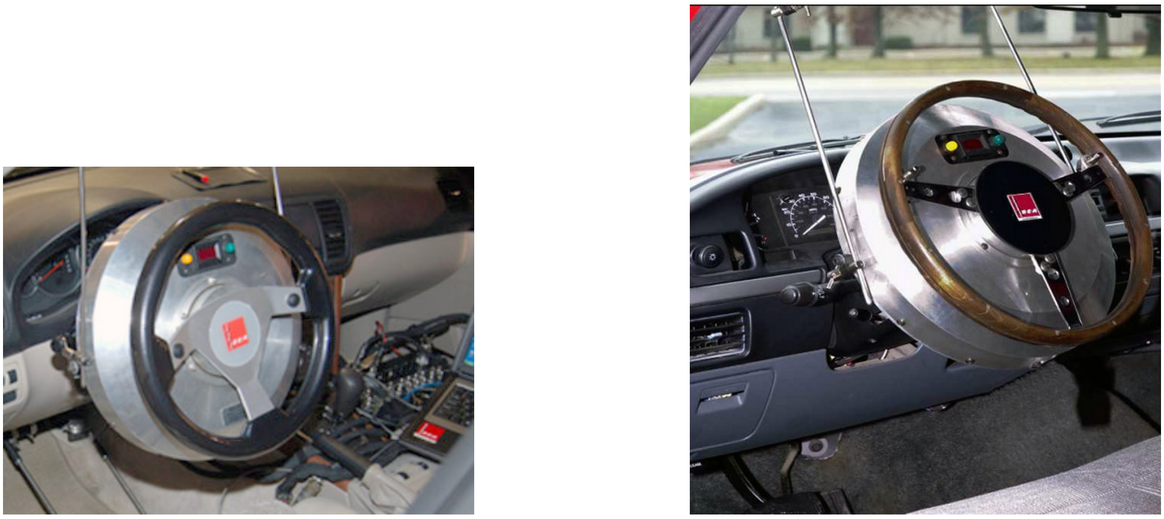
These systems can preform the measurements needed for this project. However their inability to be connected in bus, their package size and their price are not compatible with this project. This is why a new sensor that has these features is required, and since it does not exist in the market, it will be designed for this project.

1.5.2 Actuation systems

SEA, Ltd. ASC

The SEA, Ltd. automated steering controller as seen in fig 1.6a and fig1.6b is a robot that can be programmed to execute a variety of steering maneuvers[13], and also record human input for later reproduction. It is only capable of open-loop actuation [5]. It is mounted in the vehicle by replacing it's original steering wheel and fixing two vacuum pads to the windshield to prevent the entire system rotation.

It has the advantage of allowing the operator to keep hands on the wheel while operating, but is not suitable for diary assembly and disassembly, and once again can't preform close-loop operation.



(a)

(b)

Figure 1.6: SEA, Ltd. Automated Steering Controller system

ATI/Heitz Sprint 3

This robot (fig 1.7) is very similar to SEA ASC, being the main diferences the maximum torque it can achieve and its programming is done by downloading the data from the computer to one special EEPROM which is then inserted into the unit.

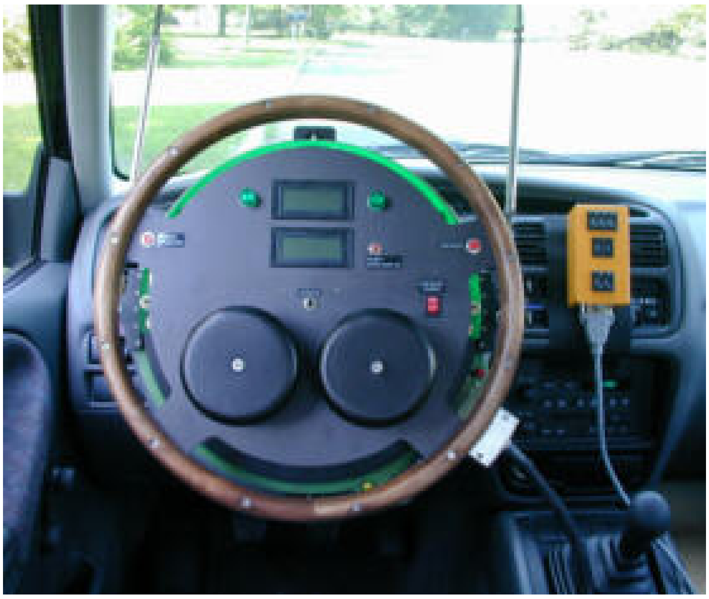


Figure 1.7: ATI/Heitz Sprint 3 steering controller

Stähle robot systems

Stähle GmbH has some driving robots that are mounted in the driver seat and actuate both the steering wheel, the pedals and the gear lever (fig1.8). It can operate in open-loop and close-loop control, has TCP/IP, CANbus and RS-232 communication protocols, and various analog and digital inputs and outputs. This would be the ideal system for this project, but it costs around 135.000€.

Kairos Autonomi® Pronto4™

Kairos Autonomi® has developed Pronto4™ (fig1.9), an agnostic autonomy system which means that it can convert almost every ground vehicle or surface vessel that uses a steering wheel or even skid steering into an tele-operated or semi-autonomous vehicle[8]. This system was used by Team Juggernaut and by University of Utah in the DARPA Urban Challenge. While having some advantages, this system is not suitable for this project, since it requires vehicle modification and takes about four hours to install.

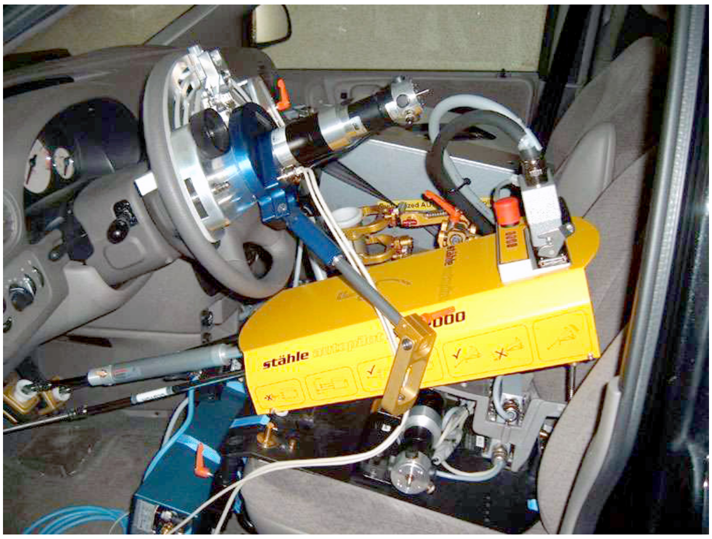


Figure 1.8: Stähle SPF2000FF robot driver

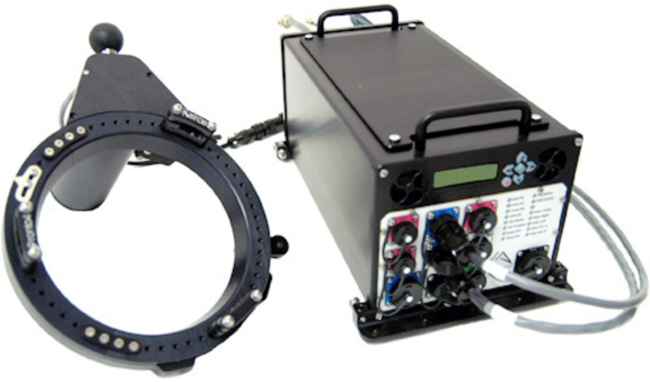
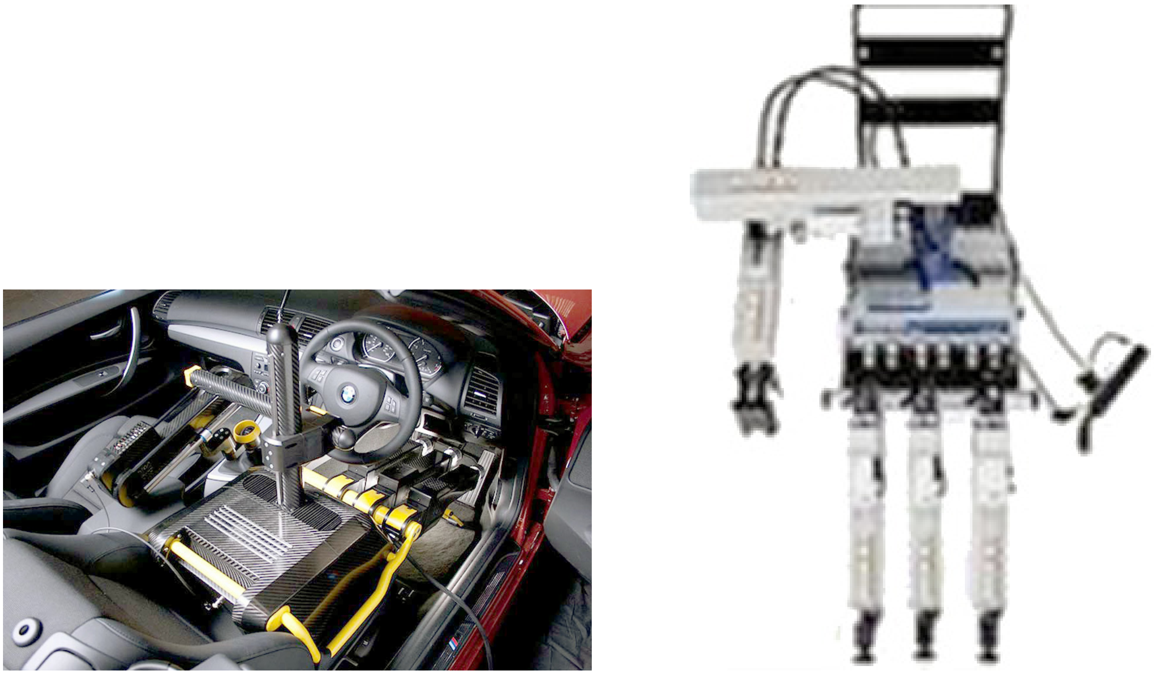


Figure 1.9: Kairos Autonomi® Pronto4™ agnostic autonomy system

1.5.3 Other systems

Other market solutions can be found, like the Froude Hofmann Robot Driver (fig 1.10a) or the Horiba ADS-7000 - Automatic Driving System (fig1.10b). But these systems are only suitable for dynamometer tests.



(a)

(b)

Figure 1.10: Froude Hofmann Robot Driver (a) and Horiba ADS-7000 (b) systems

Chapter 2

Inertial sensor network

This chapter explains the sensor network architecture, the hardware project and assembly, as well as the software and communication protocols.

2.1 Architecture

The network has the typical master, multi-slave structure (fig 2.1), where it uses Controller Area Network bus (CANbus) to establish communication between the master and the slaves. Each slave module is composed by one Microchip PIC18LF2580-I/ML microcontroller, one STMicroelectronics LPR503AL analog two-axis gyroscope, one STMicroelectronics LIS331DLH digital three axis accelerometer, one Texas Instruments SN65H233EP CANbus transceiver and several accessory passive components. These sensors are cutting edge technology and just recently started to spread in the electronics market. Few companies offer this type of solutions and those who offer do it at significant costs and with very specific applications. This systems brings then cutting edge technology at affordable costs and flexible to all applications.

On startup, the slave modules load some configuration parameters from their EEPROM and initiate the digital accelerometer configuration over an I^2C^{TM} bus communication, with the microcontroller as the master and the accelerometer as the slave. After configuration, the accelerometer generates one interrupt signal which triggers the microcontroller software to immediately initiate the data reading process from the digital accelerometer. After this acquisition, the microcontroller initiates the analog to digital conversion to read the analog gyroscope data, and then sends all the data to the master module that is listening for data in the CANbus.

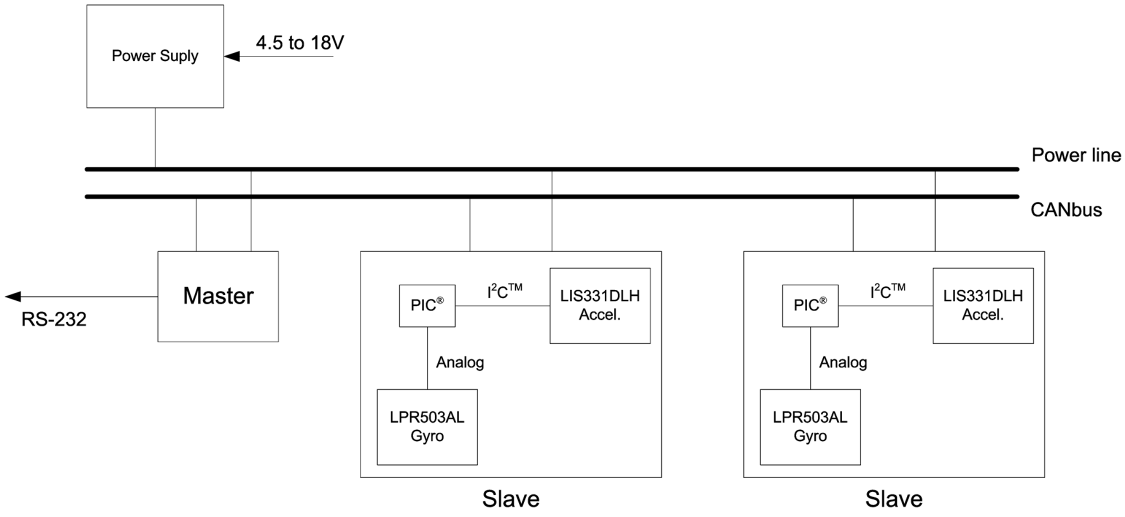


Figure 2.1: Sensor network architecture

The master module, similarly to slave modules, loads configuration parameters from its EEPROM on startup, creates a database for storing the latest data from each module and then starts the main cycle, waiting for data arriving by CANbus. After data has arrived, an interrupt is triggered to execute the procedure to read the data and store it in the database. One of the EEPROM parameters is the time interval at which the master device sends the data from the active modules to one remote computer by RS-232 communication protocol.

The communication between devices is carried out by four wires: two for the CANbus and two for the power supply. The two CANbus wires have to be terminated with one 120Ω resistor at each end to comply with CANbus specifications.

Before going to small scale, the entire setup was implemented and tested using breadboard (fig 2.2) and standard components, where the software was developed and debugged.

The power is supplied by a buck regulator (fig 2.3, 2.4a, 2.4b & 2.5),based in Texas Instruments TPS54356PWP integrated circuit and CSD16321Q5 power MOSFET. This buck regulator can supply up to 3A at 3.3V with an input from 4.5 to 18V.

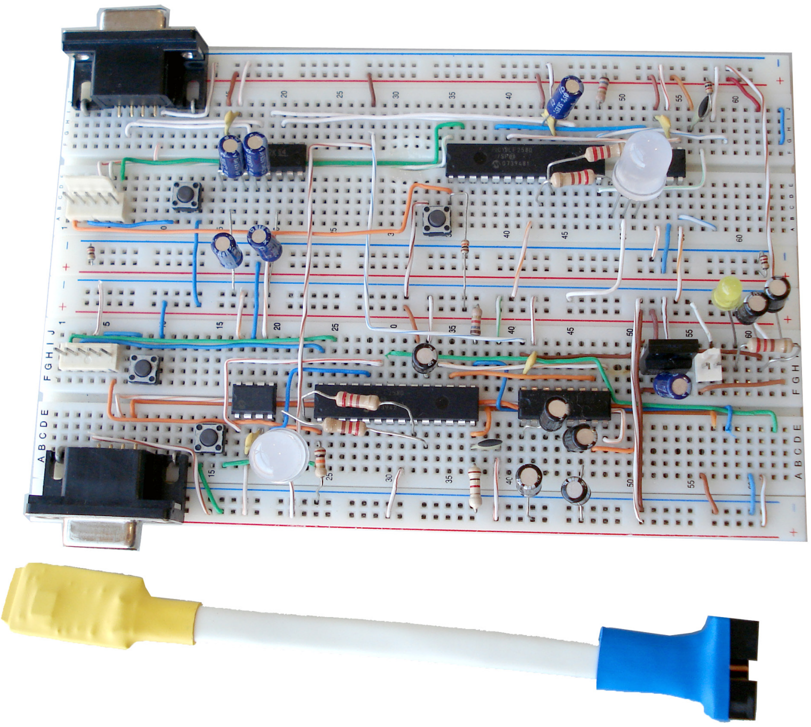


Figure 2.2: Breadboard setup for prototyping the circuit

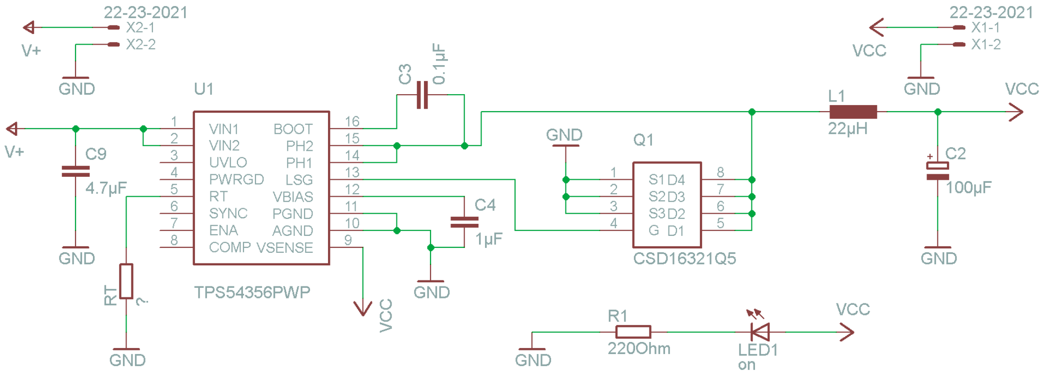
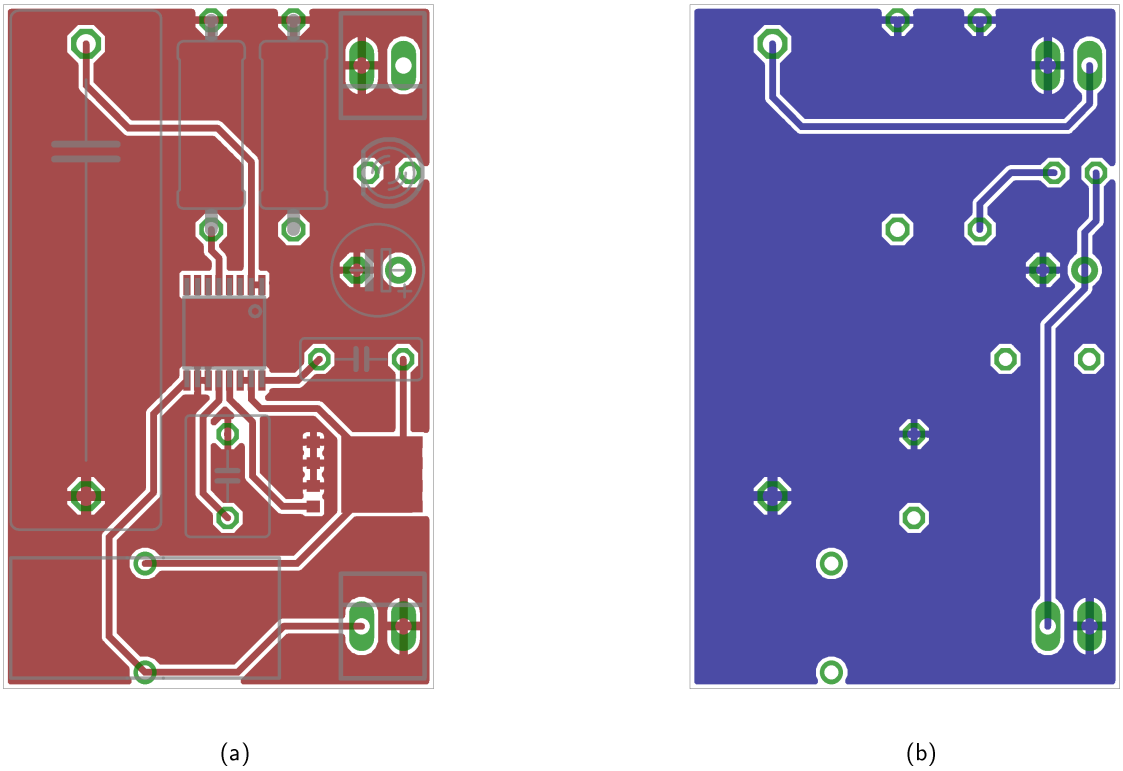


Figure 2.3: Power supply PCB schematic



(a)

(b)

Figure 2.4: Power supply PCB layout: top (a) and bottom (c)

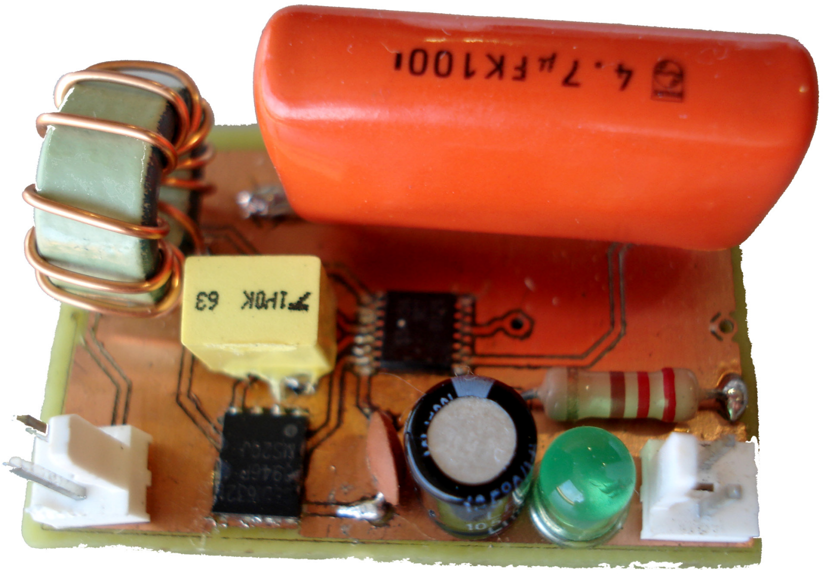
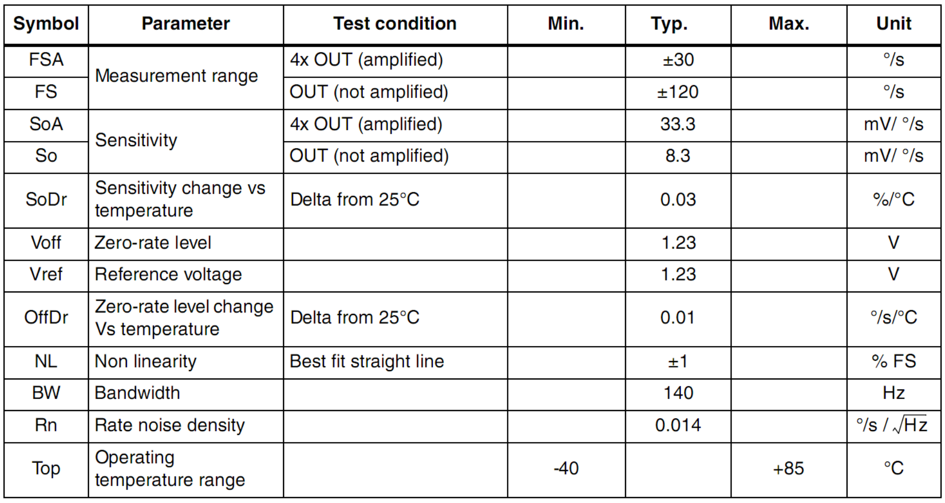


Figure 2.5: Power supply for the sensors network

2.2 Sensor specification

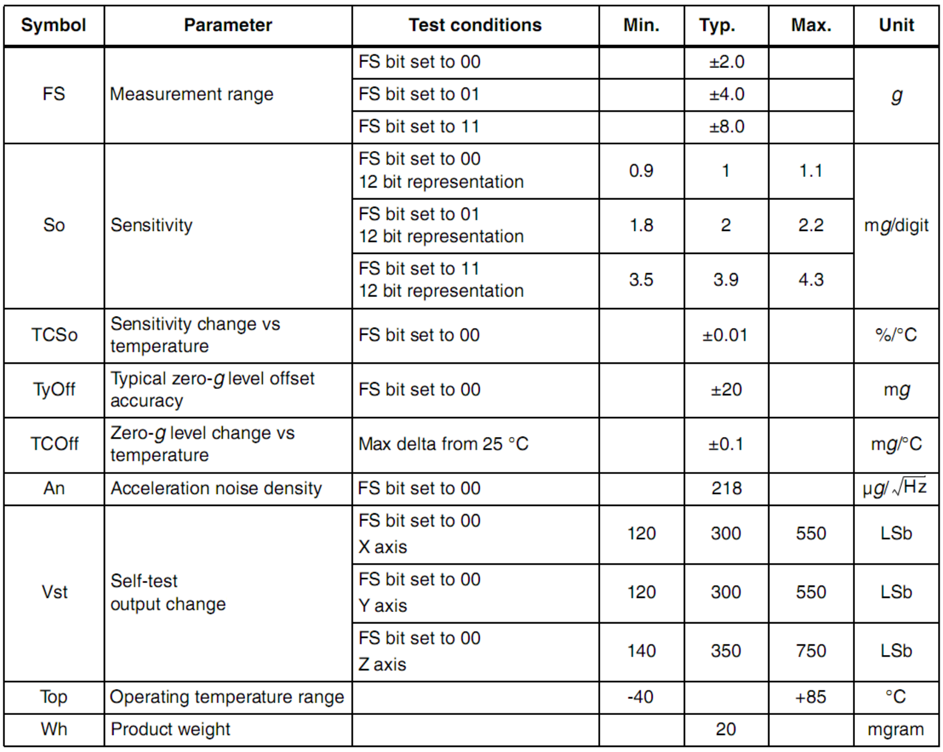
The vendor specifications of the sensors used are shown in tables 2.1 & 2.2

Table 2.1: LPR503AL gyroscope mechanical characteristics[17]



Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
FSA	Measurement range	4x OUT (amplified)		±30		°/s
FS		OUT (not amplified)		±120		°/s
SoA	Sensitivity	4x OUT (amplified)		33.3		mV/ °/s
So		OUT (not amplified)		8.3		mV/ °/s
SoDr	Sensitivity change vs temperature	Delta from 25°C		0.03		%/°C
Voff	Zero-rate level			1.23		V
Vref	Reference voltage			1.23		V
OffDr	Zero-rate level change Vs temperature	Delta from 25°C		0.01		°/s/°C
NL	Non linearity	Best fit straight line		±1		% FS
BW	Bandwidth			140		Hz
Rn	Rate noise density			0.014		°/s / $\sqrt{\text{Hz}}$
Top	Operating temperature range		-40		+85	°C

Table 2.2: LIS331DLH accelerometer mechanical characteristics[16]



Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
FS	Measurement range	FS bit set to 00		±2.0		*g*
		FS bit set to 01		±4.0		
		FS bit set to 11		±8.0		
So	Sensitivity	FS bit set to 00 12 bit representation	0.9	1	1.1	m*g*/digit
		FS bit set to 01 12 bit representation	1.8	2	2.2	
		FS bit set to 11 12 bit representation	3.5	3.9	4.3	
TCSo	Sensitivity change vs temperature	FS bit set to 00		±0.01		%/°C
TyOff	Typical zero-*g* level offset accuracy	FS bit set to 00		±20		m*g*
TCOff	Zero-*g* level change vs temperature	Max delta from 25 °C		±0.1		m*g*/°C
An	Acceleration noise density	FS bit set to 00		218		μ*g*/$\sqrt{\text{Hz}}$
Vst	Self-test output change	FS bit set to 00 X axis	120	300	550	LSb
		FS bit set to 00 Y axis	120	300	550	LSb
		FS bit set to 00 Z axis	140	350	750	LSb
Top	Operating temperature range		-40		+85	°C
Wh	Product weight			20		mgram

2.3 Slave

2.3.1 Hardware

The entire hardware setup was developed around the integrated circuits needed for this device, trying to keep it as compact as possible (fig 2.8). All components, except the CAN transceiver are presented in surface mount packages for space optimization and the number of components was reduced to the minimum. Besides the integrated circuits, there are only two voltage stabilization capacitors, two $I^2C^{\text{™}}$ bus pull-up resistors, one low pass filter and one high pass filter for each analog gyroscope axis and two status LEDs as shown in figures 2.6 , 2.9 and 2.10.

To keep the device as small as possible a third layer was added to the PCB to carry power, bus lines and device programming lines (fig 2.7b). The programming port was implemented based in the USB type A connectors, applying the male layout contacts to the PCB, which makes the connection as simple as inserting the tip of the module inside the modified USB type A female which is connected to the programming hardware as seen in figures 2.11a and 2.11b.

The assembling of the modules was accomplished using hot air from a rework station and solder paste; some difficulties where encountered due to the very small size of the components, and sometimes a preheating of the PCB was required to ensure the correct amount of solder paste, and avoid short-circuit between component pads.

This sensor module dimensions are 21x12x7mm as seen in figure 2.7a. It is considerably smaller than all market available solutions and this allows it to be used in places until now unreachable to this kind of sensors.

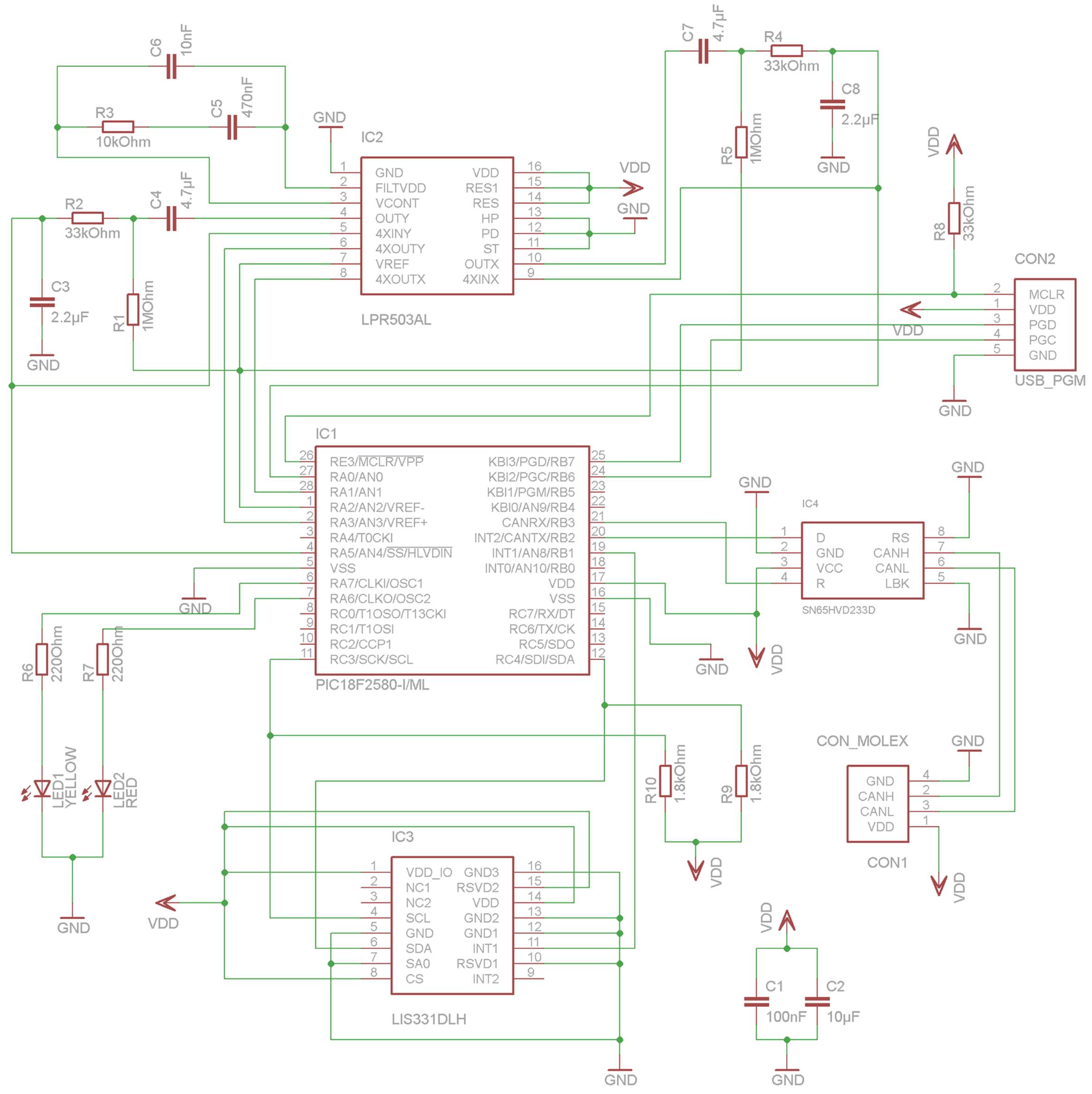
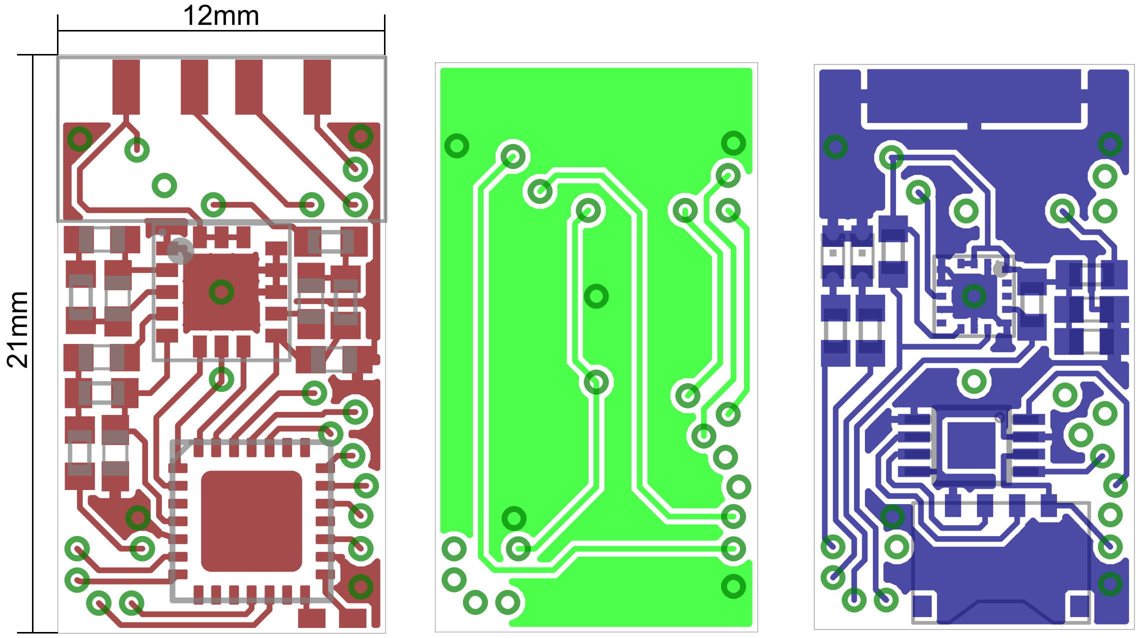


Figure 2.6: Sensor PCB schematic



(a)

(b)

(c)

Figure 2.7: Sensor PCB layout: top (a), middle (b) and bottom (c)

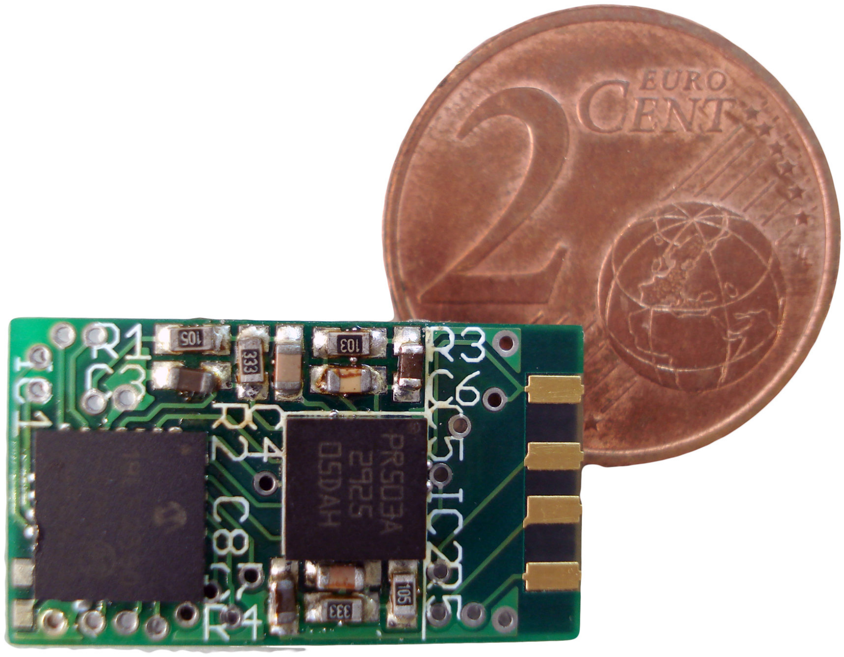


Figure 2.8: Slave size compared to coin

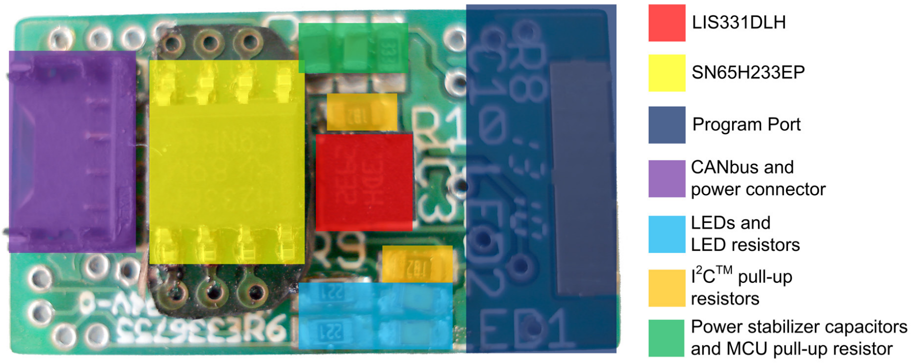

🟥 LIS331DLH
🟨 SN65H233EP
🟦 Program Port
🟪 CANbus and power connector
🟦 LEDs and LED resistors
🟧 I^2C^{TM} pull-up resistors
🟩 Power stabilizer capacitors and MCU pull-up resistor

Figure 2.9: Slave front view

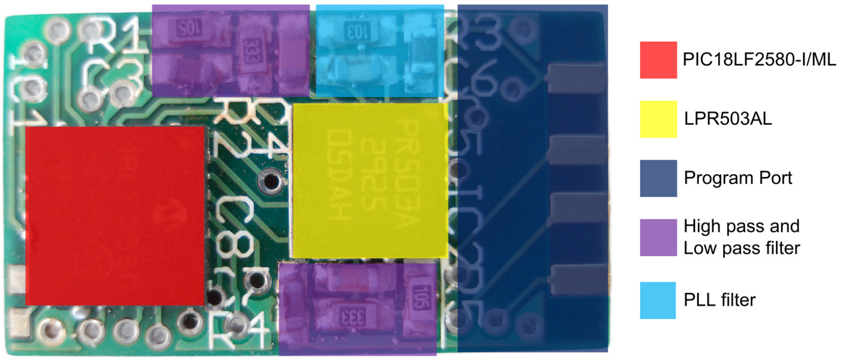


Figure 2.10: Slave back view

(a)

(b)

Figure 2.11: Programming port

2.3.2 Software

The slave module software (fig 2.12) was developed in C language and compiled with MPLAB® C18 C Compiler Lite in MPLAB® IDE environment. The software has one main routine which configures the I^{2}C™ communications and peripheral device writing configuration bytes to the accelerometer. It also configures the device hardware for interrupts, analog to digital conversion and IO digital pins. After that, the main routine enters an infinite loop responsible for the status LEDs handling. The yellow LED blinks every second and the red LED is on when the system is in error mode, that can be caused by unsuccessful CANbus communication or lack of data from the accelerometer. While in the infinite loop, the accelerometer triggers an interrupt every time it has a new set of data available for reading. The device then jumps to the interrupt service routine which calls for I^{2}C™ handling functions to read the six bytes of acceleration data form the accelerometer, then calls for analog to digital handling functions to read the data from the analog gyroscope, and after the collection of all the data it sends it to the CANbus for the master to collect it, store and then send to a remote computer.

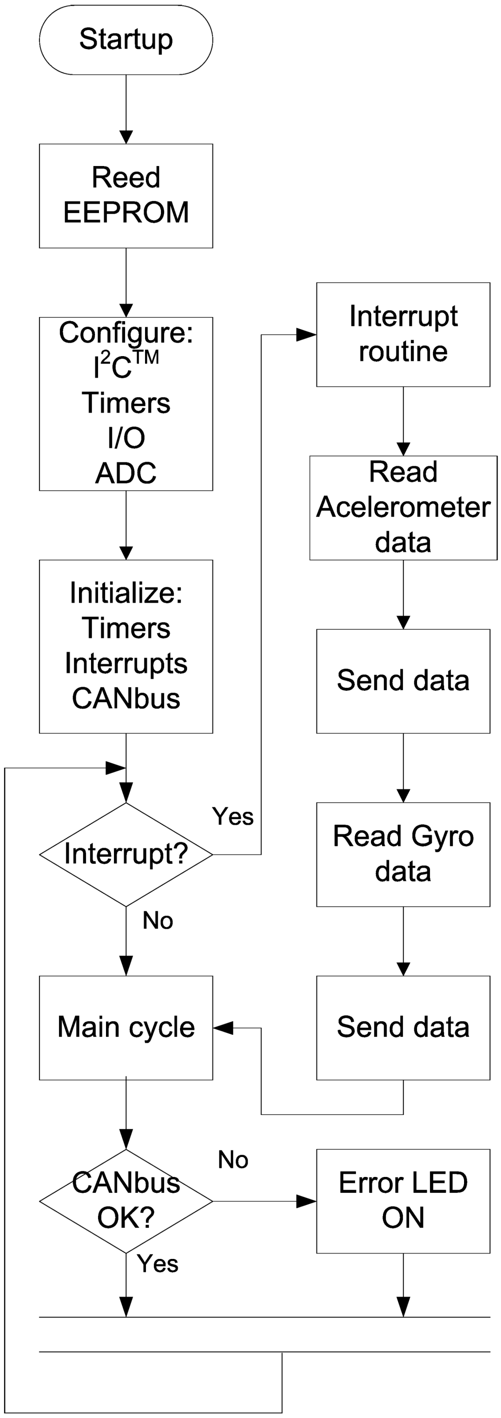


Figure 2.12: Slave module software flowchart

2.4 Master

2.4.1 Hardware

The master device hardware (fig 2.15a & 2.15b) is based in the same PCB as the slave device except it does not include inertial sensor or filters, and it is attached to one interface PCB (fig 2.14a & 2.14b), which contains the DB9 connector and the Maxim MAX3233ECWP TTL to RS-232 transceiver (fig 2.13).

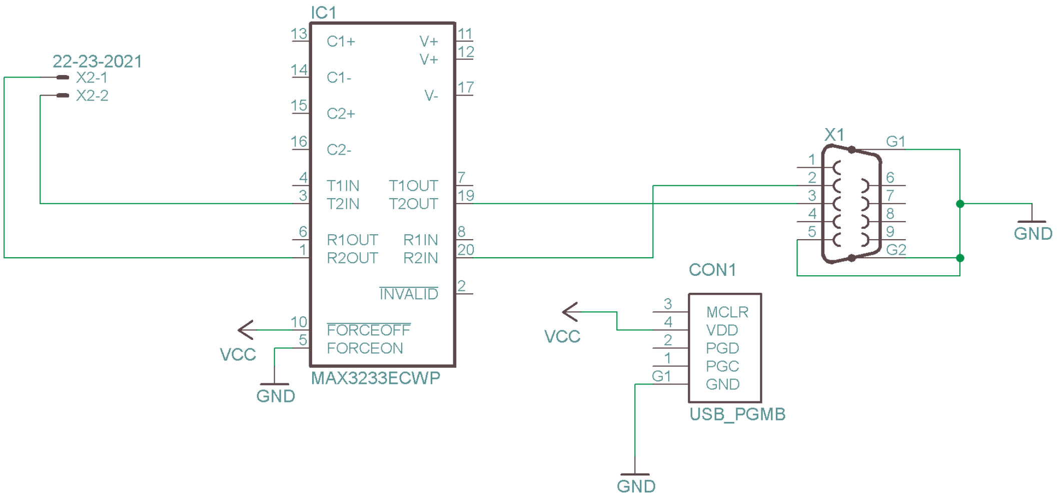
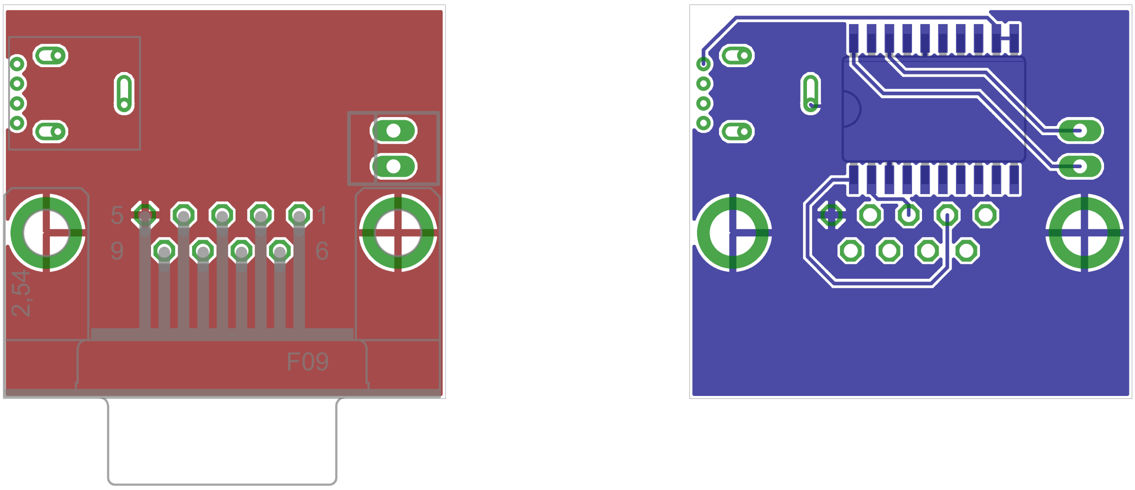


Figure 2.13: Master PCB schematic

2.4.2 Software

Similarly to the slave software, the master software (fig 2.16) was also developed in C language and compiled with MPLAB® C18 C Compiler Lite in MPLAB® IDE environment. The software has one main routine which creates the database in the device second and third memory bank with capacity to store 32 simultaneous slave devices, and clears the memory, setting all the values to zero. In sequence, EEPROM values are loaded to global variables to define the rate at which the device sends all the data stored in memory to one remote computer, and then configures IO pins, timers, CAN module and interrupts, before entering the infinite loop responsible for status LEDs actuation, sending the data over RS-232 (fig 2.17) and cleaning the database from expired entries caused by modules who have been disconnected. When data is received from CANbus one

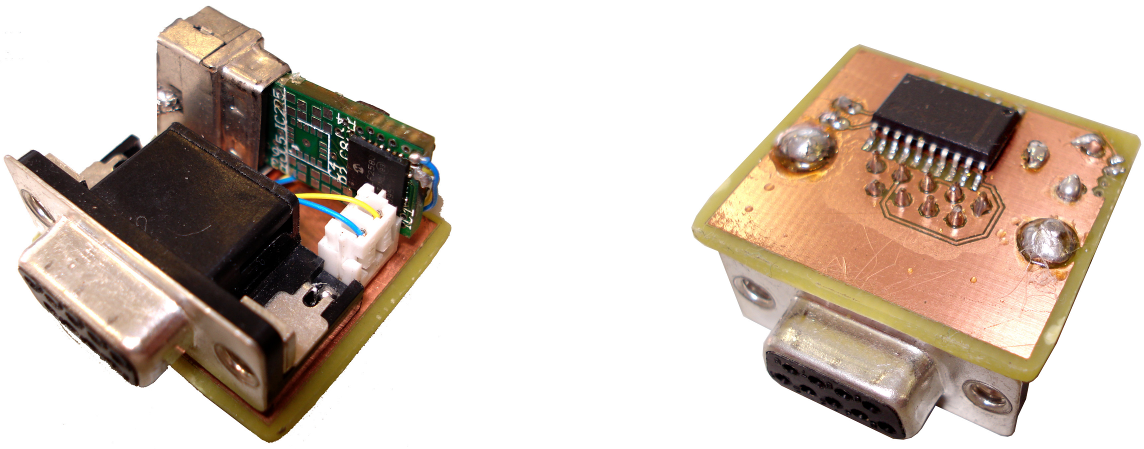


(a)

(b)

Figure 2.14: Master PCB layout: top (a) and bottom (c)

interrupt is generated and the device executes the interrupt service routine. This routine reads the new data; searches the table to find the entry which this new data came from; if the entry already exists the data is updated; if it's from new module, the routine searches for empty database entries to store the new module data. If the database is full, the master module enters error mode, but still handling the remaining slave modules.



(a)

(b)

Figure 2.15: Master module top (a) and bottom (b) view

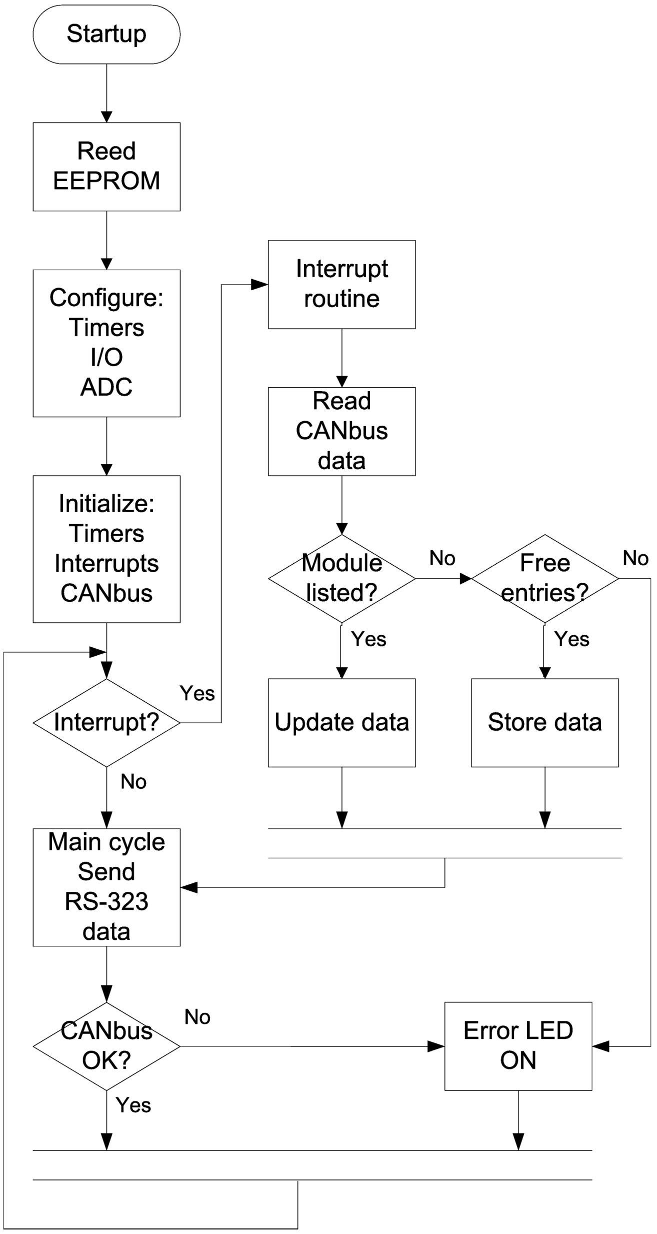


Figure 2.16: Master module software flowchart

2.5 RS-232 communication Protocol

This protocol (fig 2.17) is used to send the data from the master device database to any remote device. It works as a data stream, and the send timing can be modified by changing the 0x00 address in the device EEPROM memory.

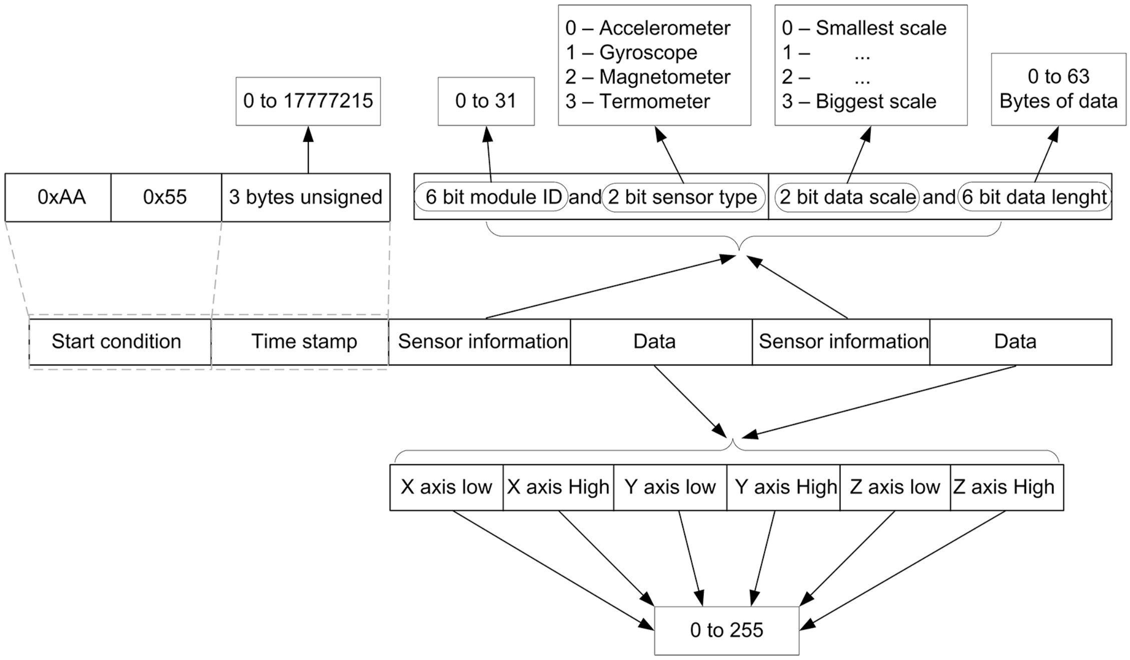


Figure 2.17: Master module RS-232 protocol

2.6 CAN communication Protocol

This protocol (fig 2.18) is used between slave devices and the master module. Given that CAN protocol requires at least eleven bit addressing, and to keep the messages as short as possible, some data is sent in the available address bits, followed by the six data bits containing the sensor measures values.

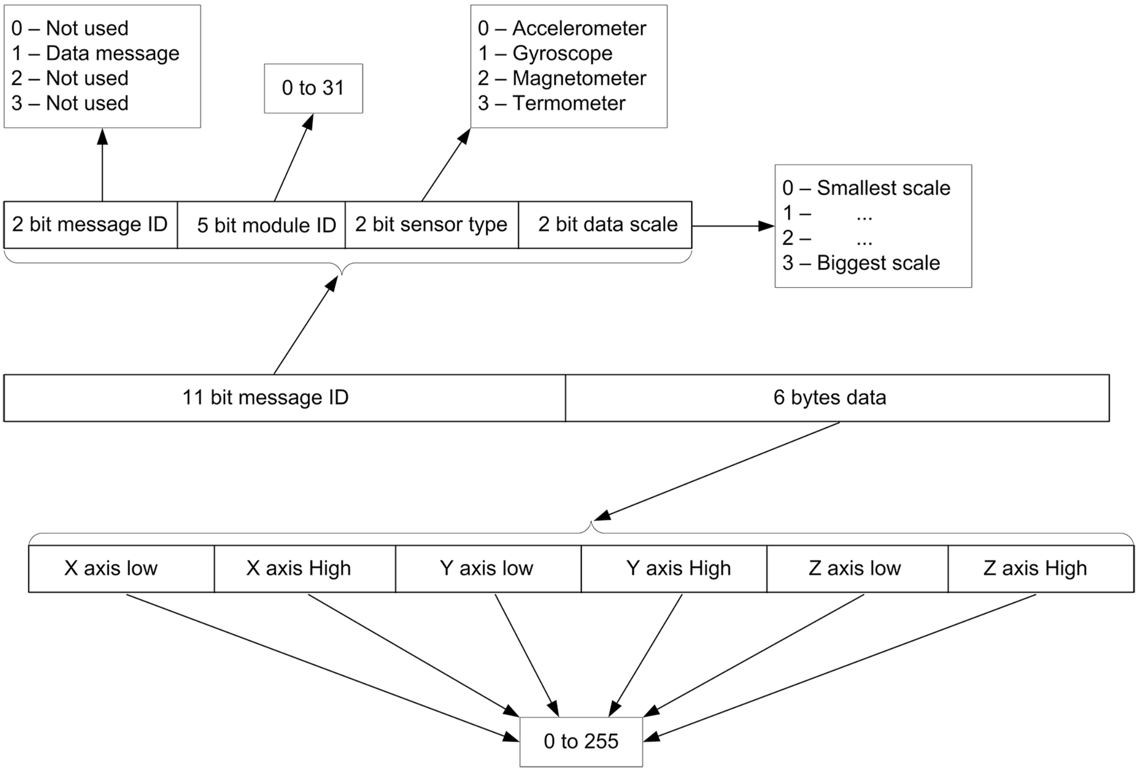


Figure 2.18: Sensor network CAN protocol

Chapter 3

Actuation system

3.1 Overview

This chapter describes the design stage of the pedal actuation unit: frame design, pneumatic components selection and design as well as manufacture of the custom parts, that link the pneumatic cylinders to the vehicle pedals.

3.2 Design

The pedal actuation unit (fig 3.1) project was desegned based in simple principles.

- The unit has to be compact;
- Easy and quick to install;
- Easy to operate;
- Simple to remove, so the vehicle can be driven to the test site, the systems are installed in few minutes, and after the tests the unit is detached from the vehicle and it can be driven back again.

To accomplish this goal, the unit frame was developed using Bosch Rexroth aluminium framing, and pneumatic actuators were used to provide compliance and compact design. The actuators are controlled by electric pressure regulators to simulate human force control. To ensure system robustness the actuation unit central control will be assigned to an industrial PLC.

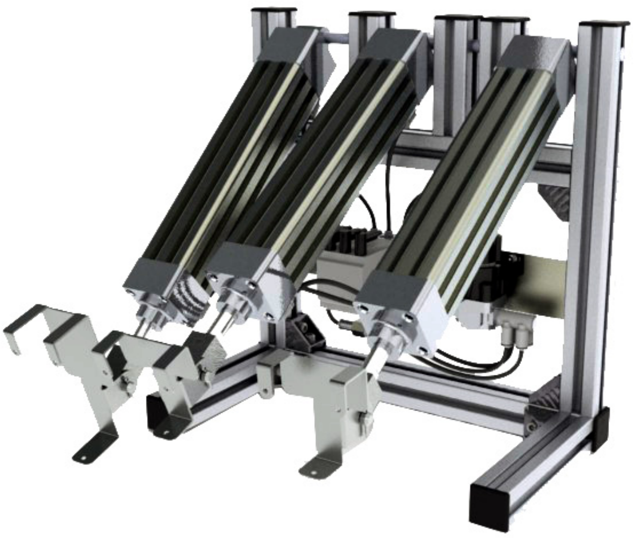


Figure 3.1: Pedal actuator unit

For designing purposes, the force needed to actuate the vehicle controls was measured using a load cell. The maximum values registered were: 80N for accelerator, 375N for brake, 205N for clutch and a torque of 7.5N/m for the steering wheel.

3.3 Frame

For assembly ease, the frame (fig 3.2) is designed out of Bosch Rexroth 30x30mm aluminium framing, and fits in front of the vehicle driver seat. It is designed to be light, and easy to install, remove and store. It provides support to all pneumatic actuation components.

3.4 Pneumatic components

With the base values from the force tests the following pneumatic equipment was selected:

- Cylinders: SMC CP95SDC32-200;
- Electric pressure regulators: SMC ITV0050-L (fig 3.3a & 3.3b);
- Manual pressure regulator: SMC ARX21-01B;
- Tube: $\varnothing 4mm$

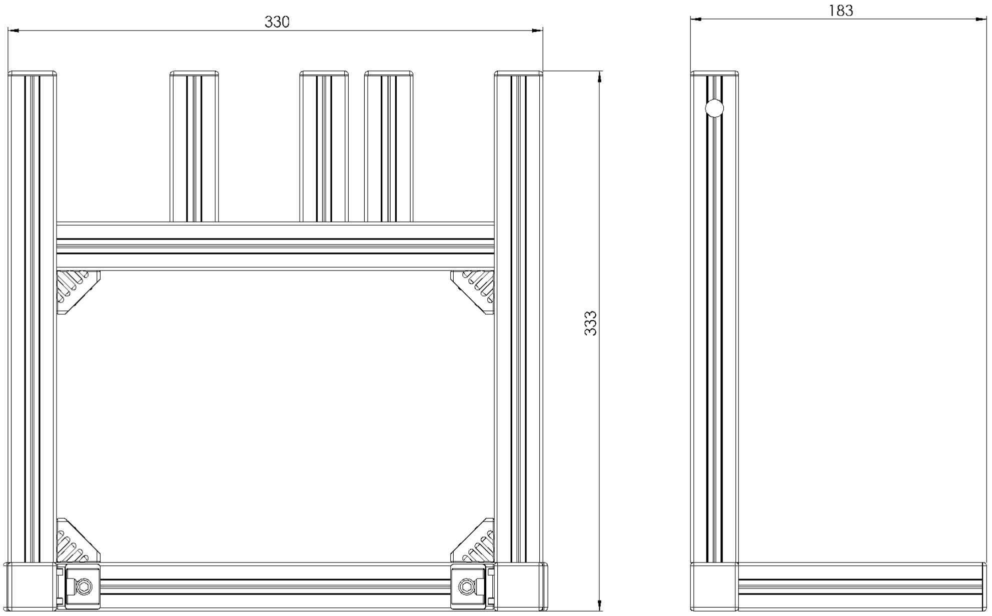
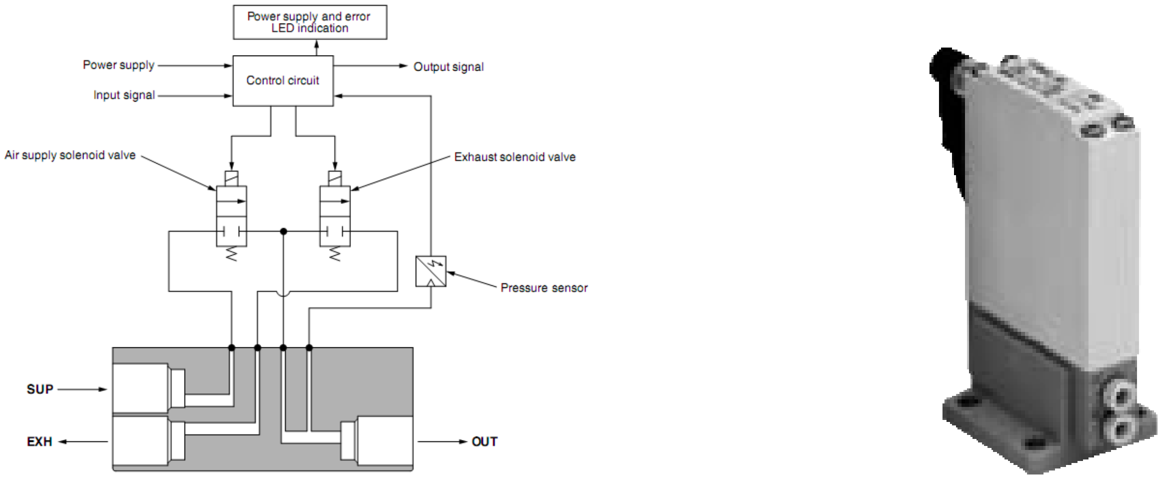


Figure 3.2: Frame dimensions

- Tube accessories: SMC KQ2 series
- Silencer: SMC AN A1-01

3.5 Custom parts

Custom parts were designed to fix the pneumatic cylinders to the vehicle pedals. To comply with the required movement degrees of freedom, the attachment has two parts. One, as seen in figures 3.4 and 3.5 has one M10 thread which is attached to the pneumatic cylinder rod and was machined from a block or aluminum. The other, as seen in figures 3.6 and 3.7 is shaped so it fits each of the ATLASCAR1 pedals and locks into place by the spring effect from the shape itself and was cut and fold from steel sheet metal.



(a)

(b)

Figure 3.3: SMC ITV00 working principle[2] (a) and package (b)

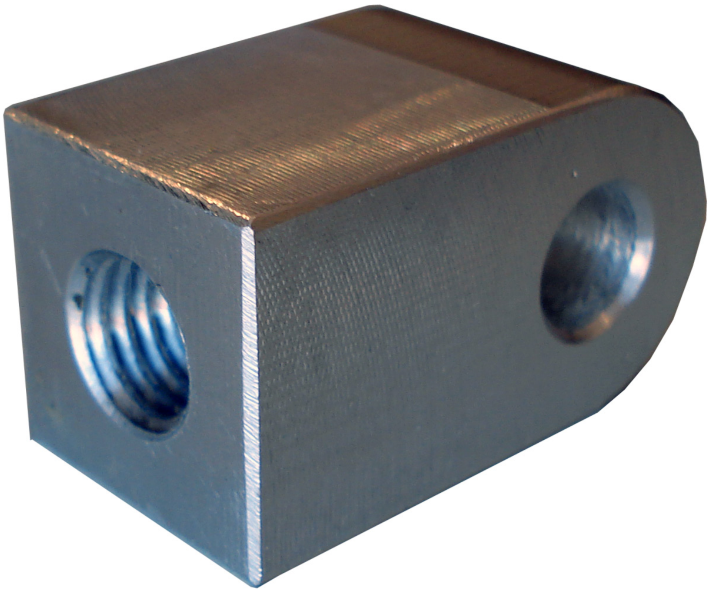


Figure 3.4: Cylinder rod attachment

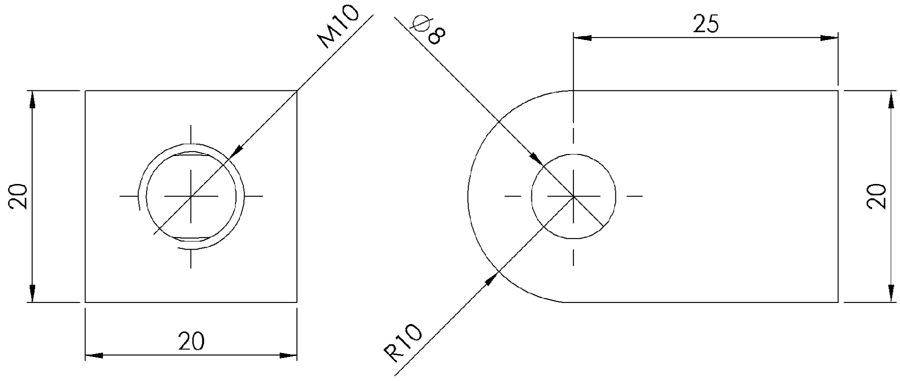


Figure 3.5: Cylinder rod attachment drawing

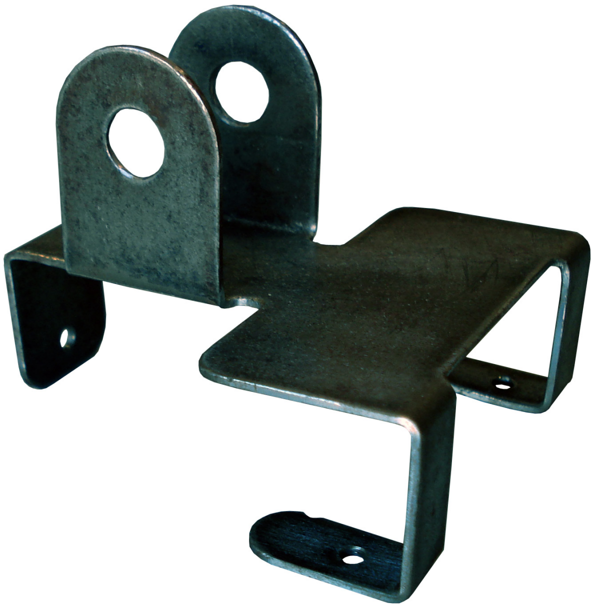


Figure 3.6: Clutch and brake pedals attachment

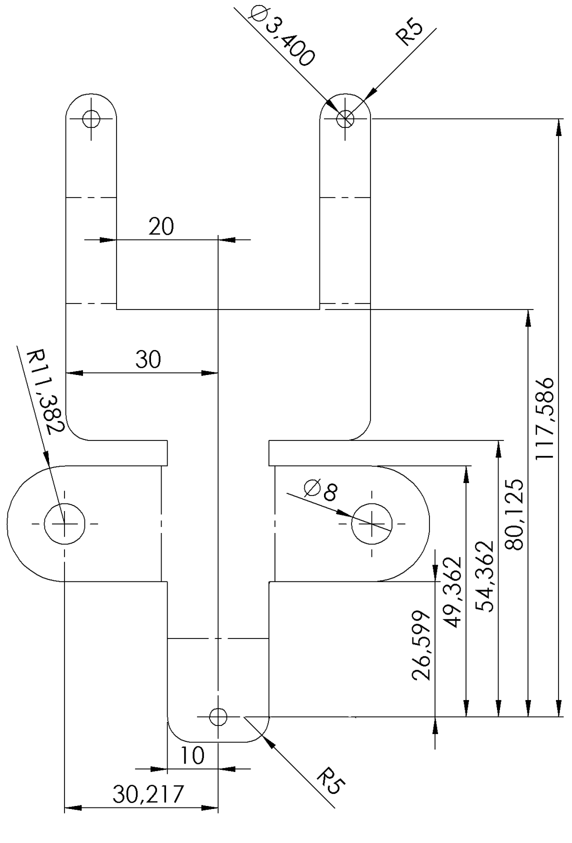


Figure 3.7: Clutch and brake pedals attachment drawing

Chapter 4

Results

4.1 Overview

This chapter describes the sensor network tests and results. For testing purposes, two sensors were fixed to the pedal setup from a standard game controller with thermal glue. One was fixed to the frame and the other to the accelerator pedal. The cable connections were also glued to prevent the detachment from the sensors. The system was tested in two different situations. Static situations, which mean that the only acceleration experienced by the setup is gravity, and dynamic situations where the setup experiences other accelerations due to its motion.

4.2 Static tests

For the initial tests, data was collected independently from each sensor in static horizontal position (fig 4.7a) and processed by a MATLAB® script that receives the data and displays it without scale conversion . The test results are shown in figures 4.1 & 4.2.

For differential measurement tests, the MATLAB® script was adapted to receive data from two modules and the compute the angle between two 3D acceleration vectors as explained in section 1.4. The results are shown in figure 4.3.

Then, the accelerator pedal was pressed as shown in figure 4.4 and the data from figure 4.5 was collected.

To validate the differential approach to the problem, the setup was then positioned statically once again, but now in vertical position (fig 4.7b) and data shown in figure 4.6 was recorded.

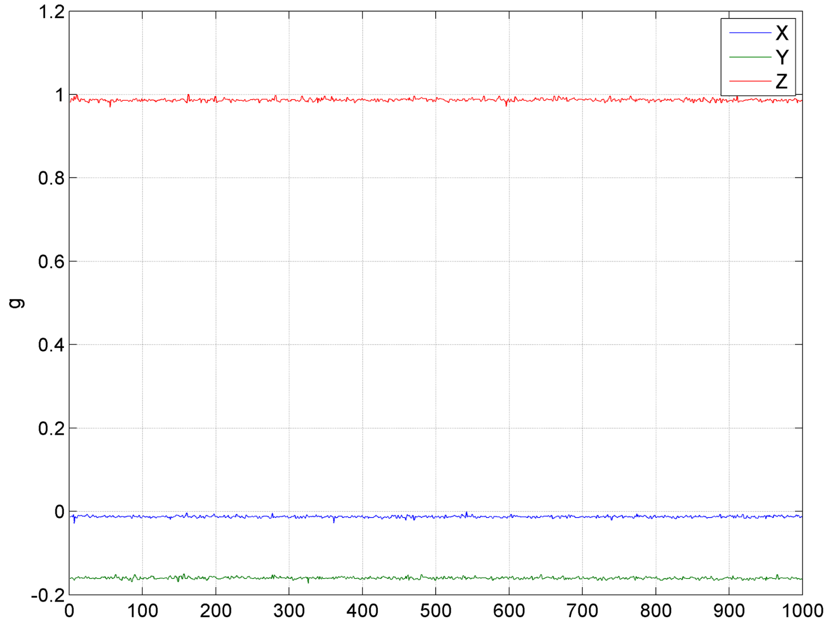


Figure 4.1: Three axis raw data from accelerometer

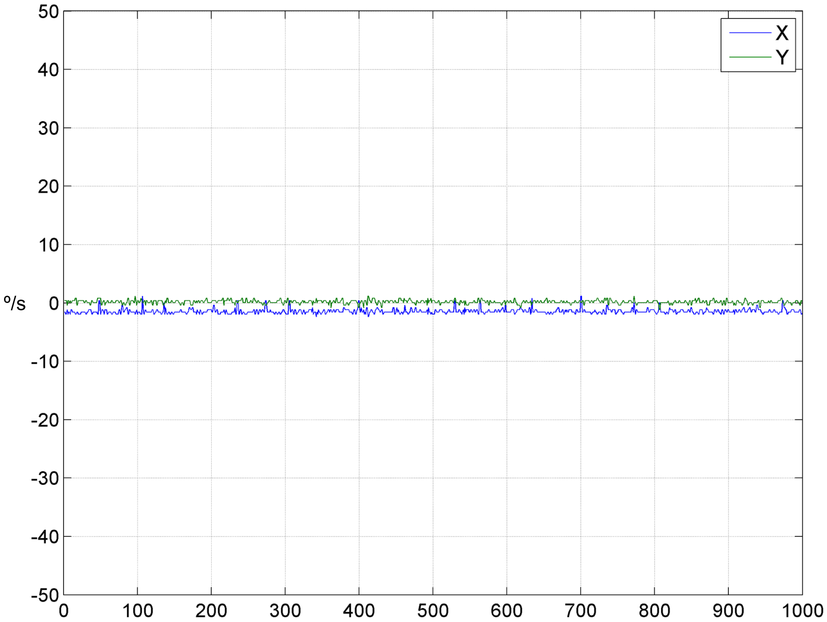


Figure 4.2: Two axis raw data from gyroscope

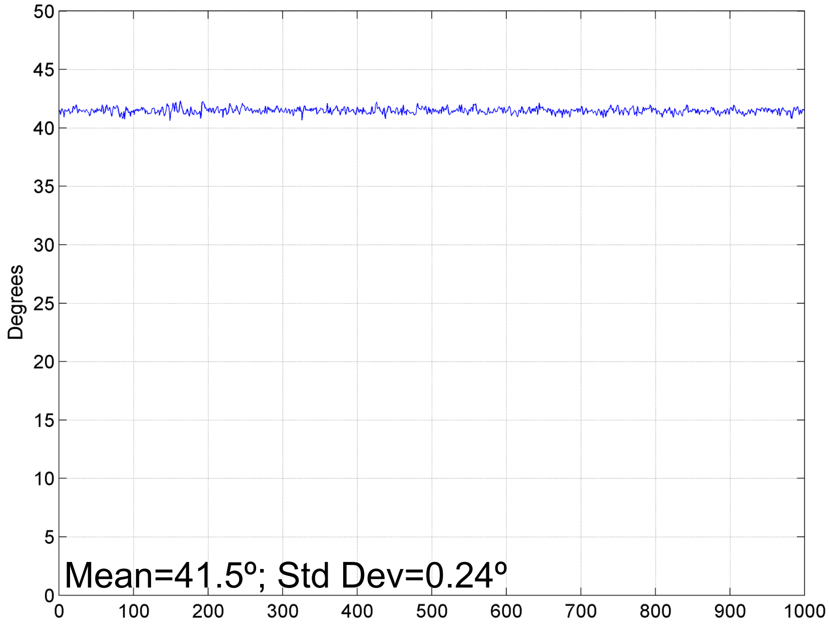


Figure 4.3: Differential angle measurement with the setup in horizontal position

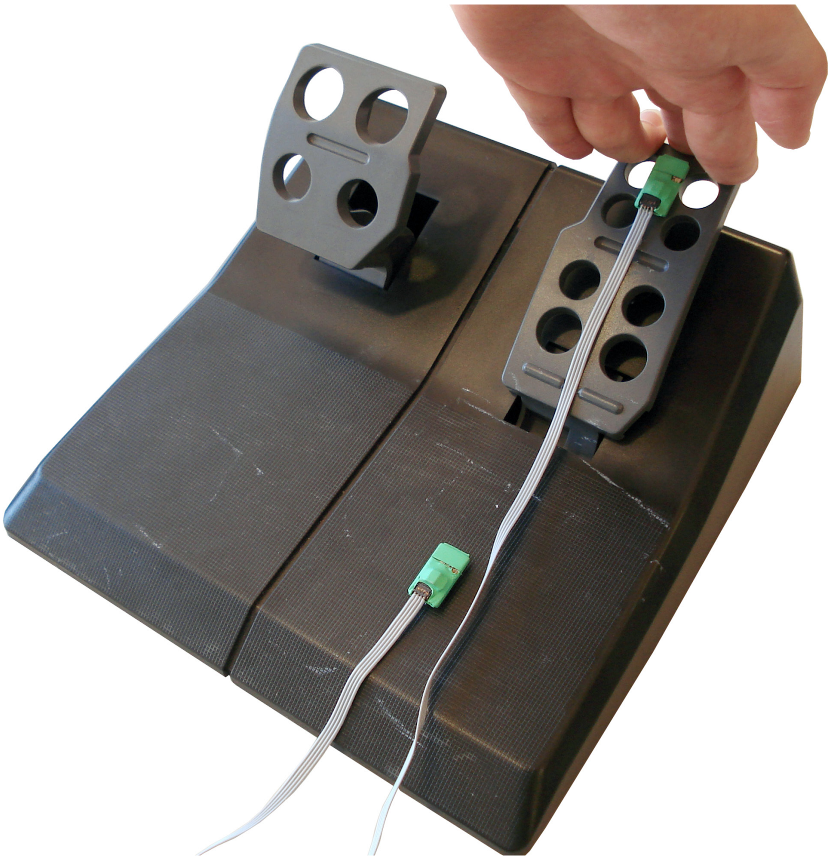


Figure 4.4: Setup in horizontal position with accelerator pedal pressed

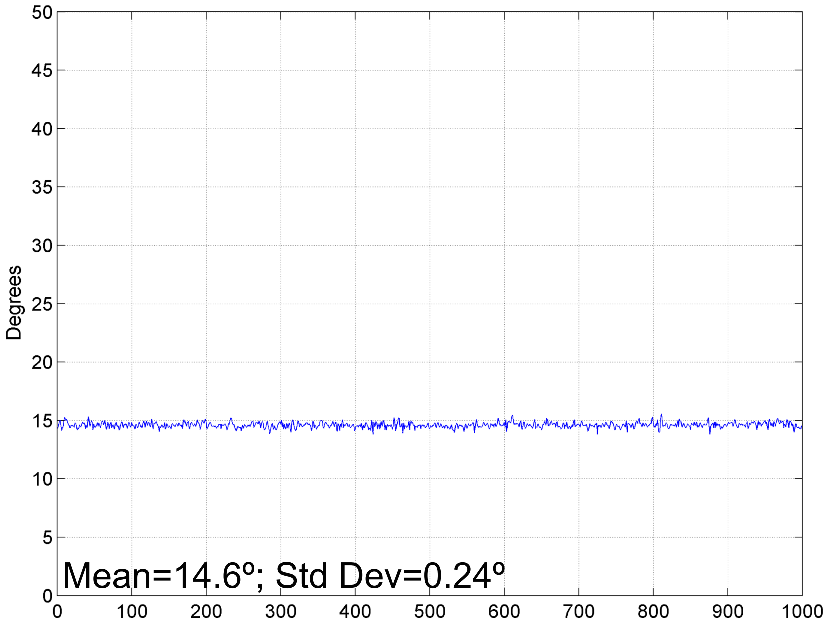


Figure 4.5: Differential angle measurement with the setup in horizontal position with accelerator pedal pressed

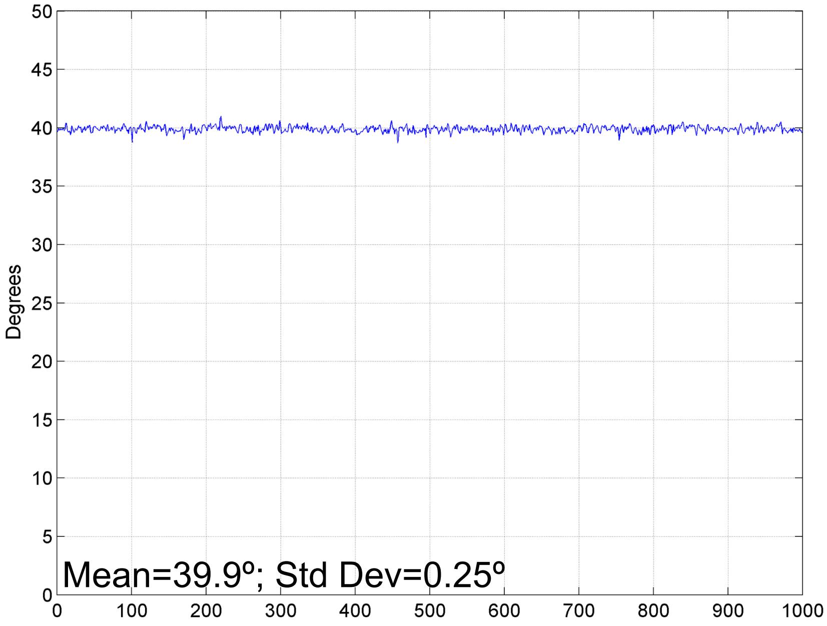
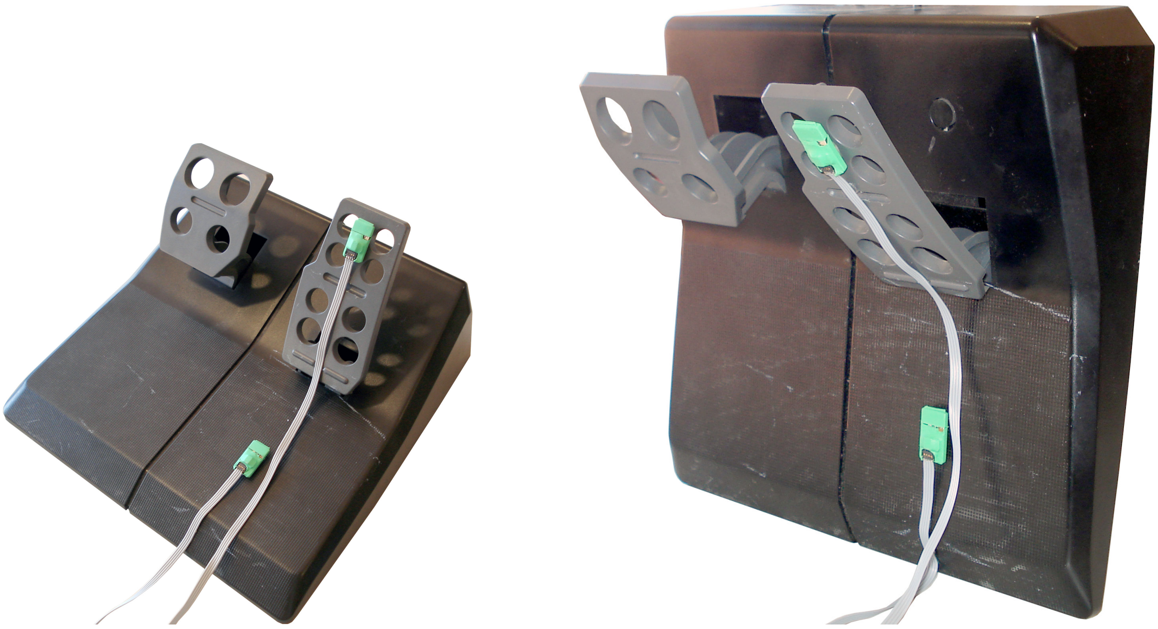


Figure 4.6: Differential angle measurement with the setup in vertical position



(a)

(b)

Figure 4.7: Setup in horizontal (a) and vertical(b) static test position

To understand how the setup roll angle affects the measurement, the setup was attached to the FANUC M-6iB industrial robotic manipulator and rolled slowly to avoid other accelerations than the gravity. Both the roll and the measurement were registered as shown in figure 4.8. The gyroscope data collected during this experience is shown in figure 4.9.

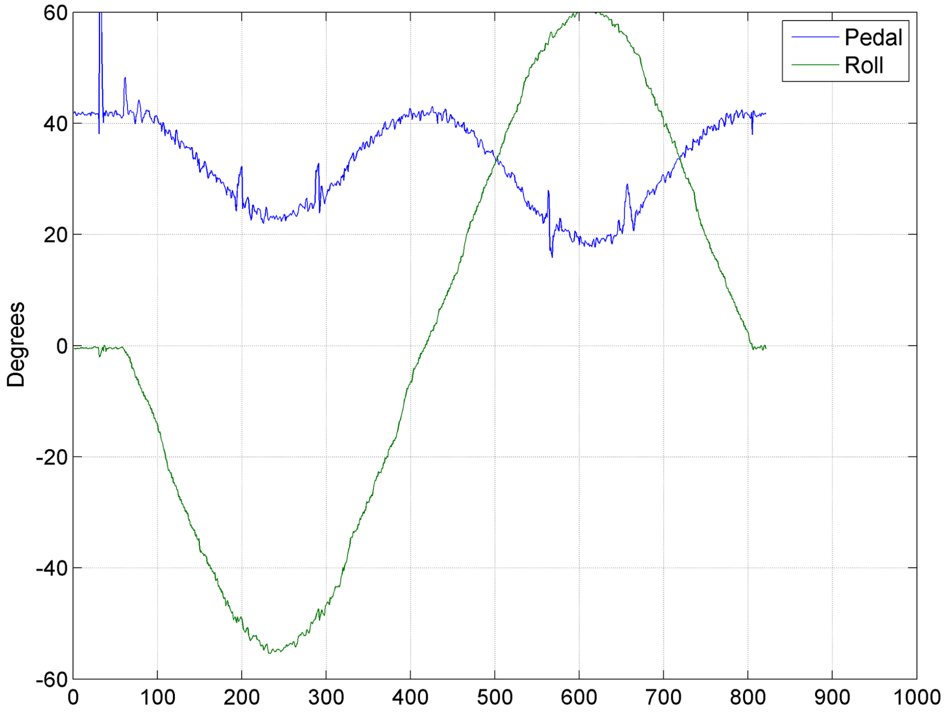


Figure 4.8: Pedal angle measurement versus setup roll

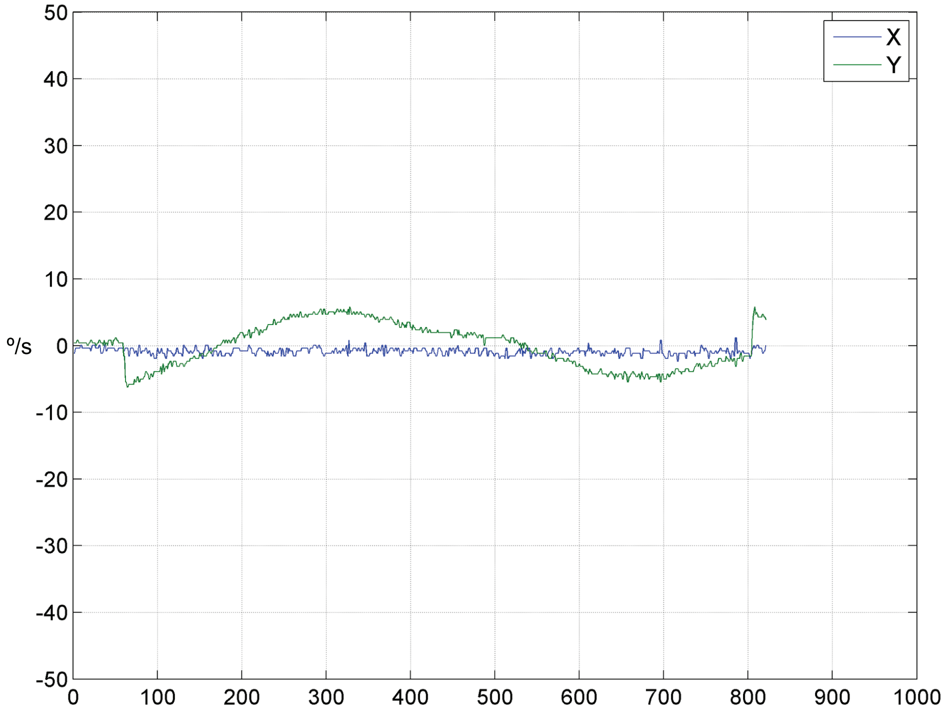
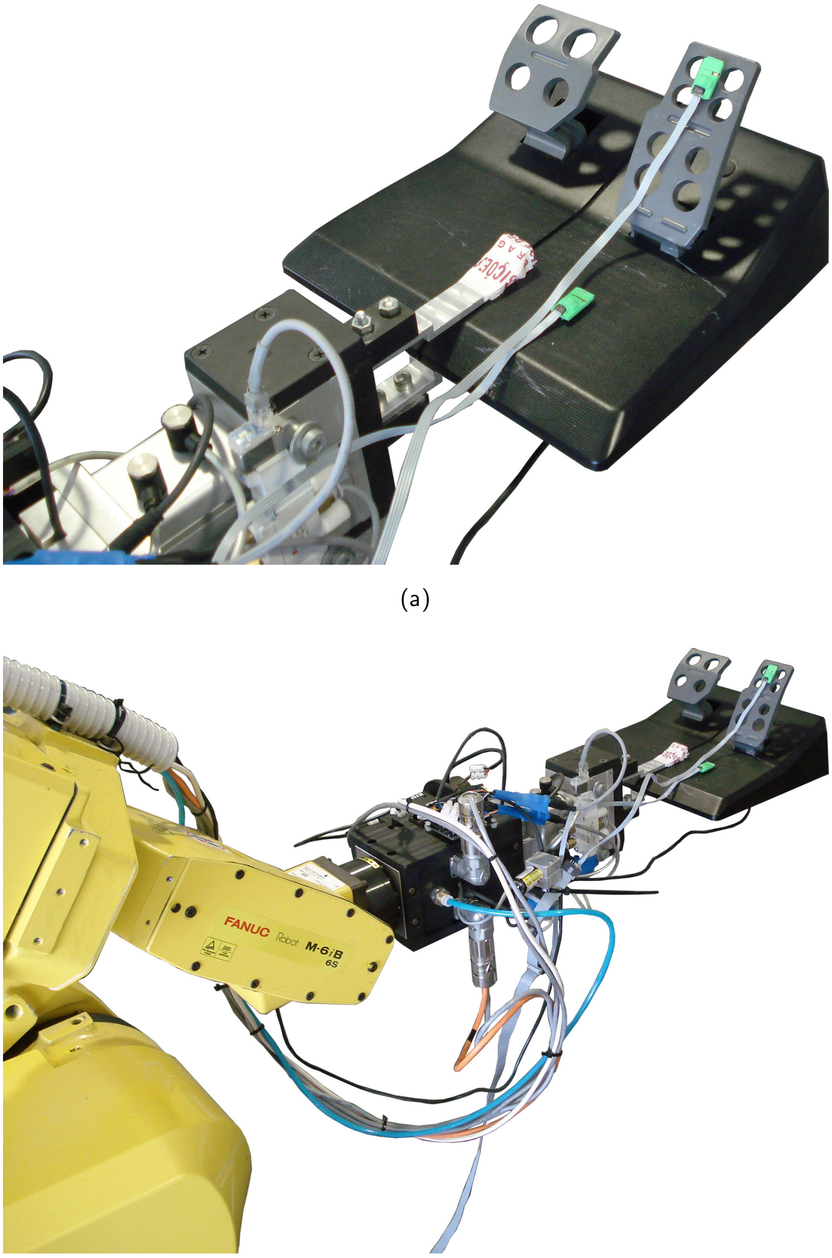


Figure 4.9: Gyroscope data for the slow setup roll test

4.3 Dynamic tests

For motion tests the setup was secured to one FANUC M-6iB industrial robotic manipulator to preform repeated movements capable of being executed afterwards when necessary (fig 4.10a & 4.10b). Several different movements were preformed at a linear speed of 250mm/s.



(a)

(b)

Figure 4.10: Setup attached to FANUC M-6iB robot

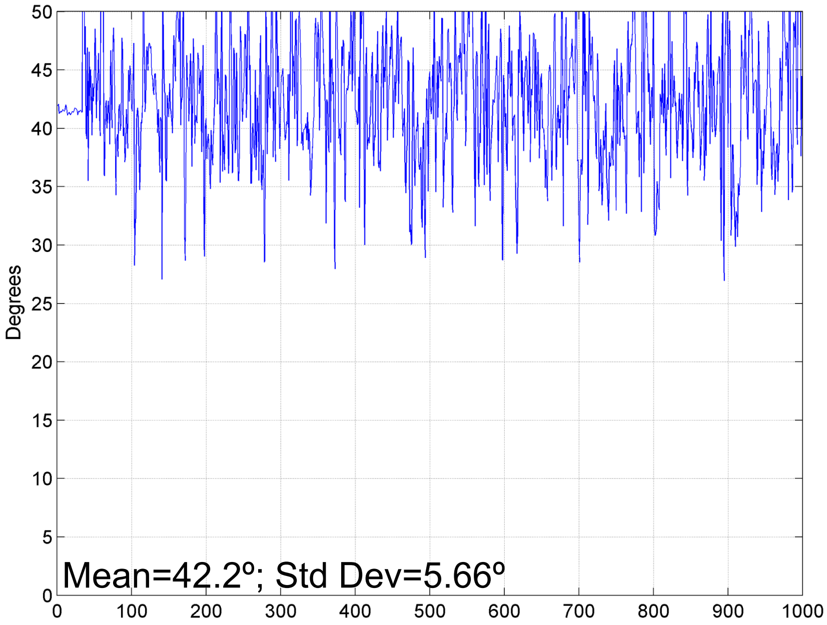


Figure 4.11: Horizontal plane elliptical movement

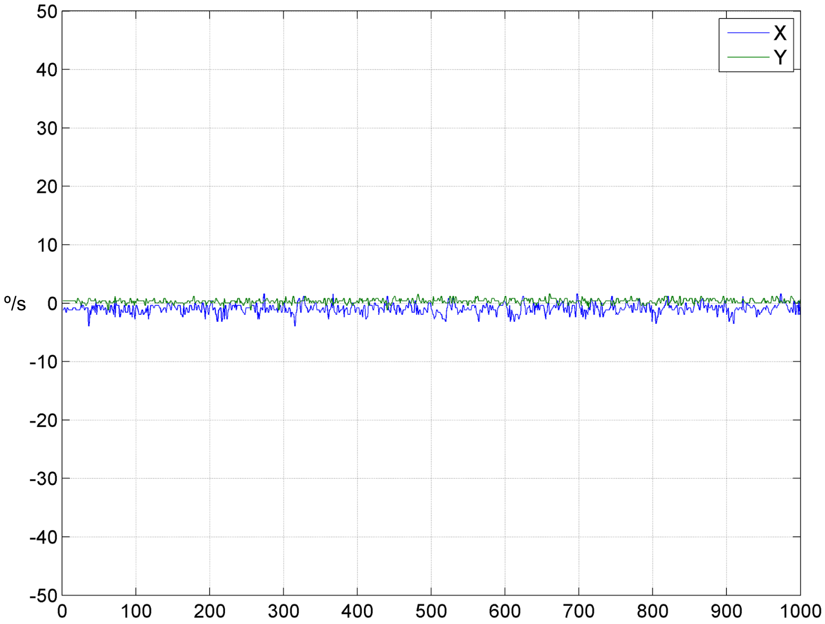


Figure 4.12: Horizontal plane elliptical movement gyro data

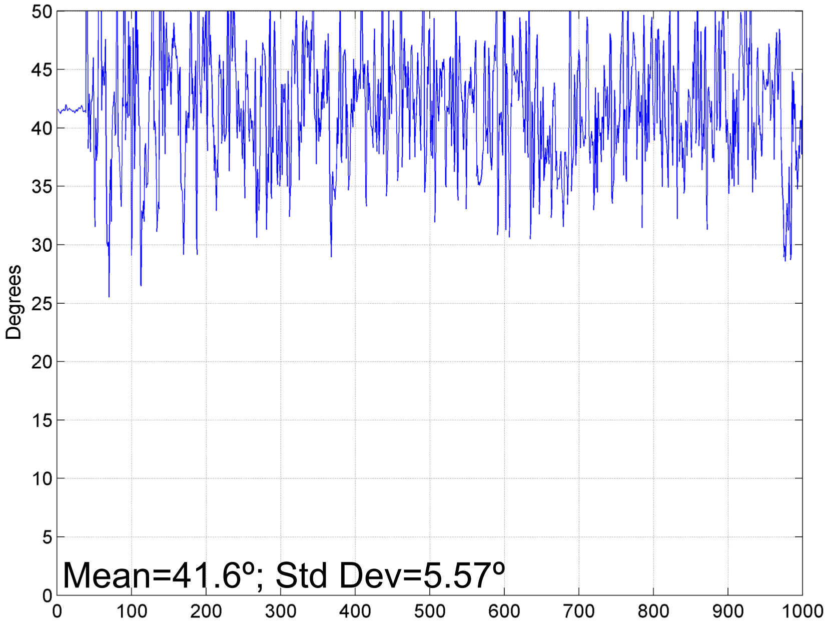


Figure 4.13: Horizontal plane elliptical with setup roll movement

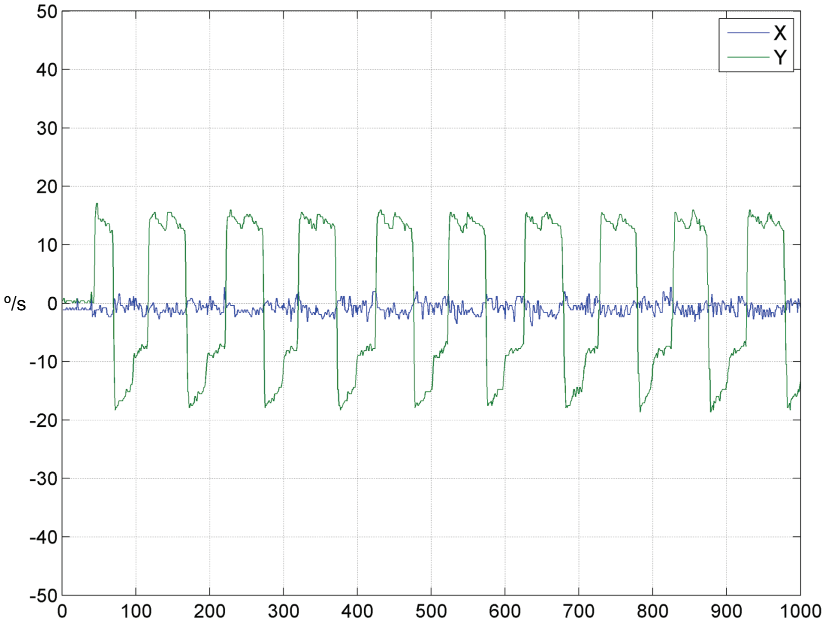


Figure 4.14: Horizontal plane elliptical with setup roll movement gyro data

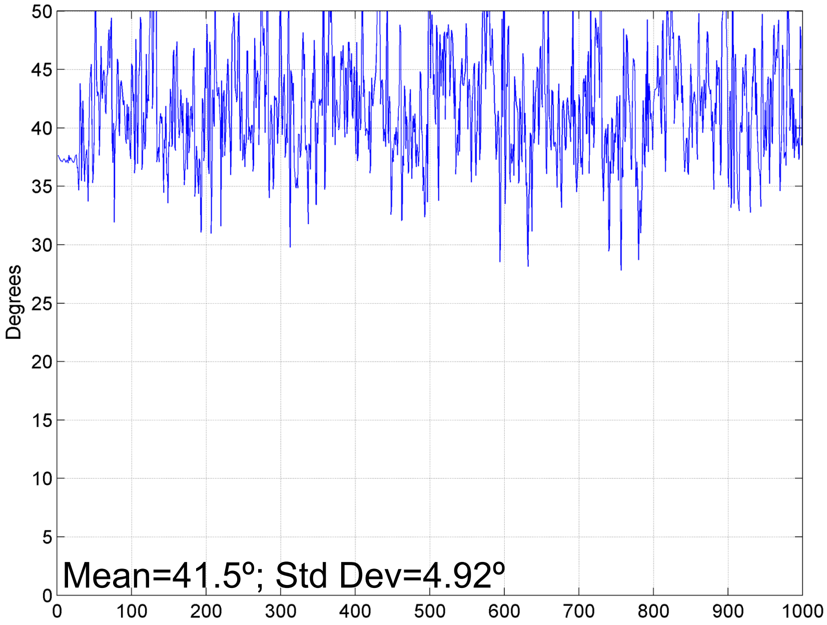


Figure 4.15: Horizontal plane linear with setup roll movement

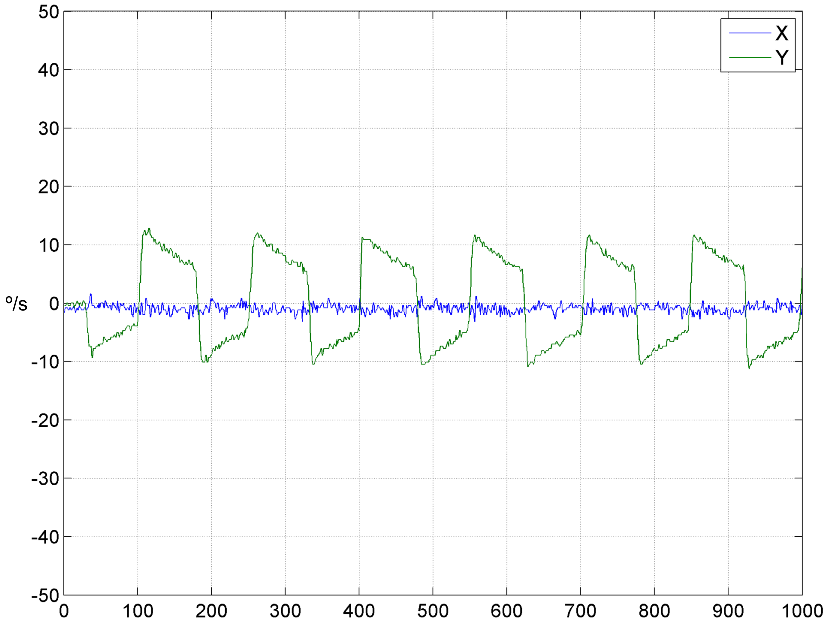


Figure 4.16: Horizontal plane linear with setup roll movement gyro data

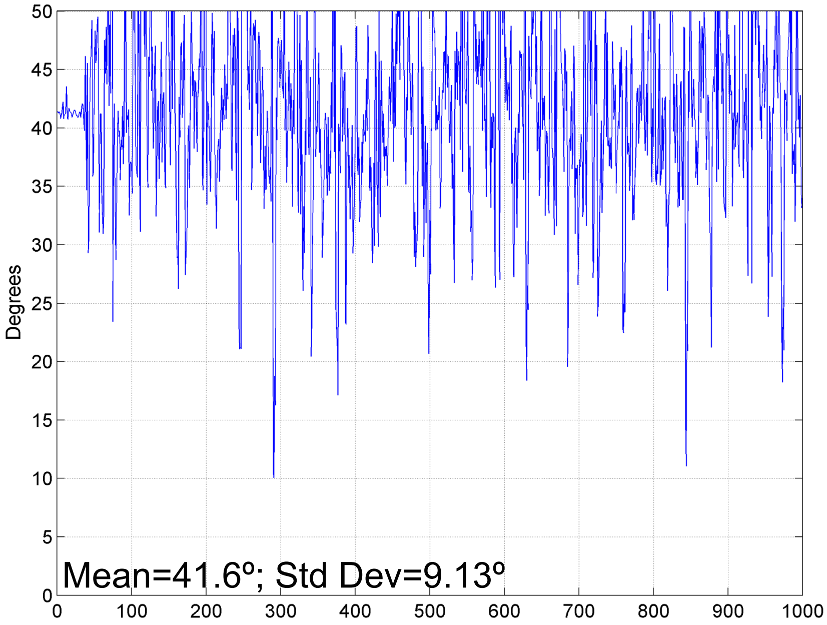


Figure 4.17: X axis rotation movement

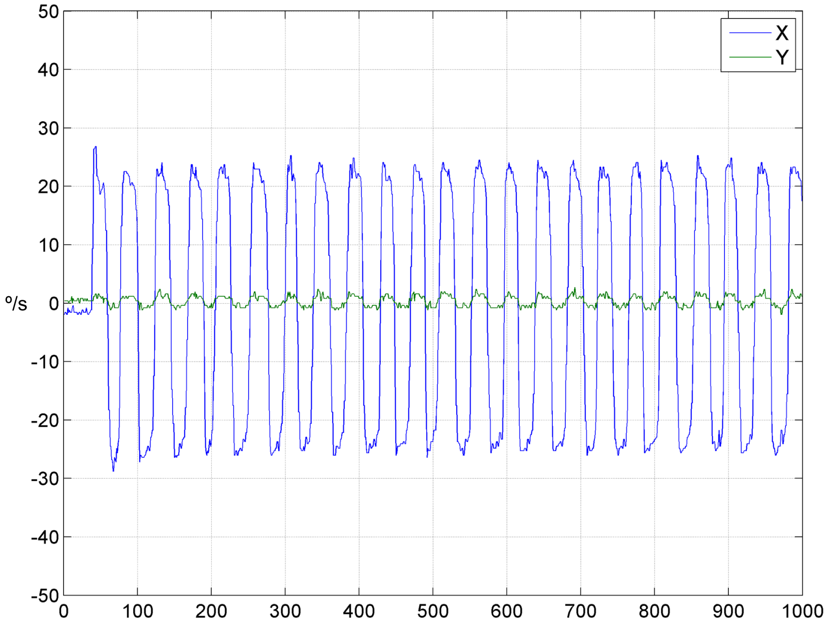


Figure 4.18: X axis rotation movement gyro data

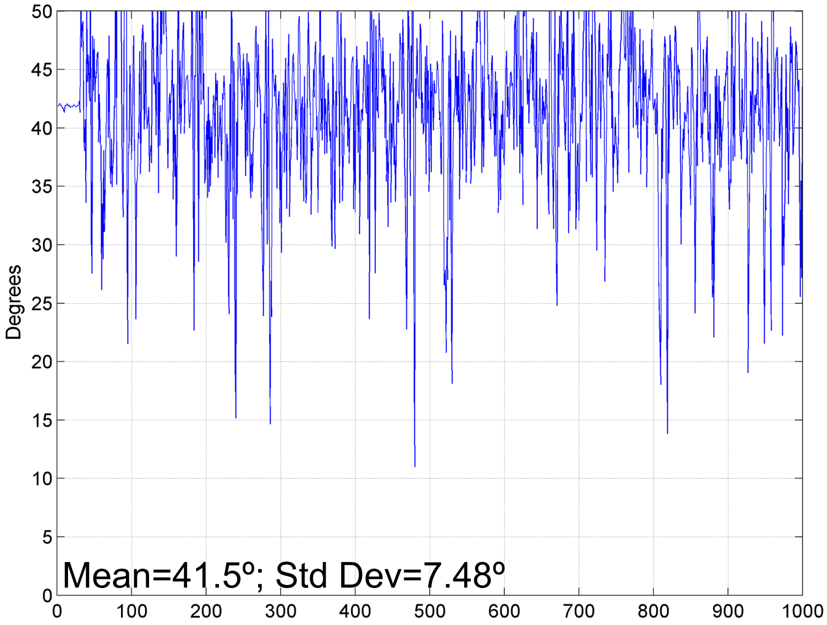


Figure 4.19: X axis rotation and vertical translation movement

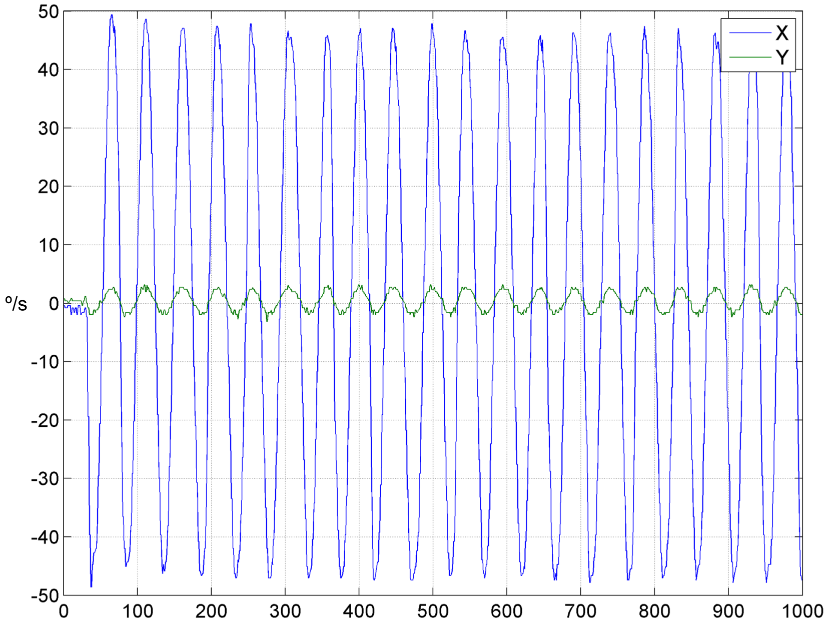


Figure 4.20: X axis rotation and vertical translation movement gyro data

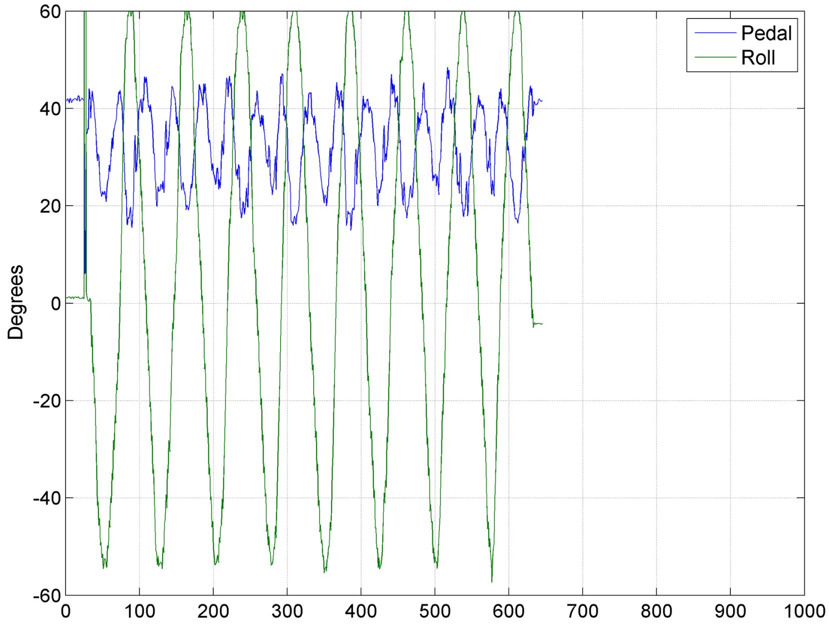


Figure 4.21: Fast roll movement

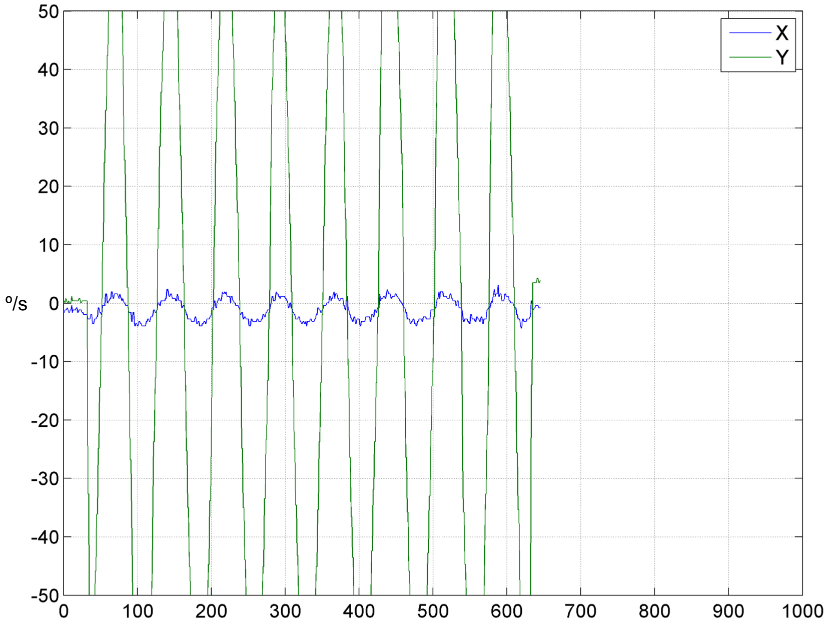


Figure 4.22: Fast roll movement gyro data

4.4 Analysis of results

From the data analysis the system accuracy while static is about 0.25° corresponding to 0.8% of measuring range. In static situations the difference between mean values are within standard deviation values so the differential measurement approach is validated. Due to the actual angle computing method, the system is very sensitive to the setup roll angle as seen in figure 4.8. If the setup is submitted to roll positions over ±25° from gravity vector line, the measured angle accuracy lowers to 10%. This is explained by the fact the the angle is computed by the equation 1.5, which uses the three axis data to compute the angle between the two acceleration. As the X axis component becomes more significant compared with Y and Z axis and both sensors X axis are parallel, the angle value starts to decrease all the way to zero, which would happen when the acceleration component in the X axis reaches 1g. After this analysis the computing method was adapted to discard the X axis acceleration component and compute the angle using only Y and Z axis components. The results of this test are shown in figure 4.23. By the analysis of these results the system can now be subjected to roll angles up to ±40° and still preform accurate measurements. For roll angles over ±40° the acceleration components in Y and Z become too small to allow accurate measurements.

When the setup is submitted to global movement, the standard deviation values increase significantly and minor changes are also noted in the mean value. This happens because the acceleration vector module is not the same to all the sensors, this fact voids the simplification used to obtain this differential measures.

After motion tests it became clear that such simplification was not valid and more complex dynamic laws have to be considered. The general movement law is complex to compute[4] , however if the units share one parallel axis, which is also parallel to the pedal rotation axis, there may be the possibility to keep some simplifications on the system. As the geometry of the system is known, a reference unit can be used to compute the differences in the acceleration applied to the other sensors[3]. Having one parallel axis common to the units, rotation movements are most likely to cause different modules in the acceleration applied to each unit. Knowing the vector that links each unit origin with the reference, the angular velocity data from the gyroscopes ω can be used with the circular trajectory radius r in equation 4.1 to compute the expected difference in the acceleration between sensors, which will be used to correct the previous method.

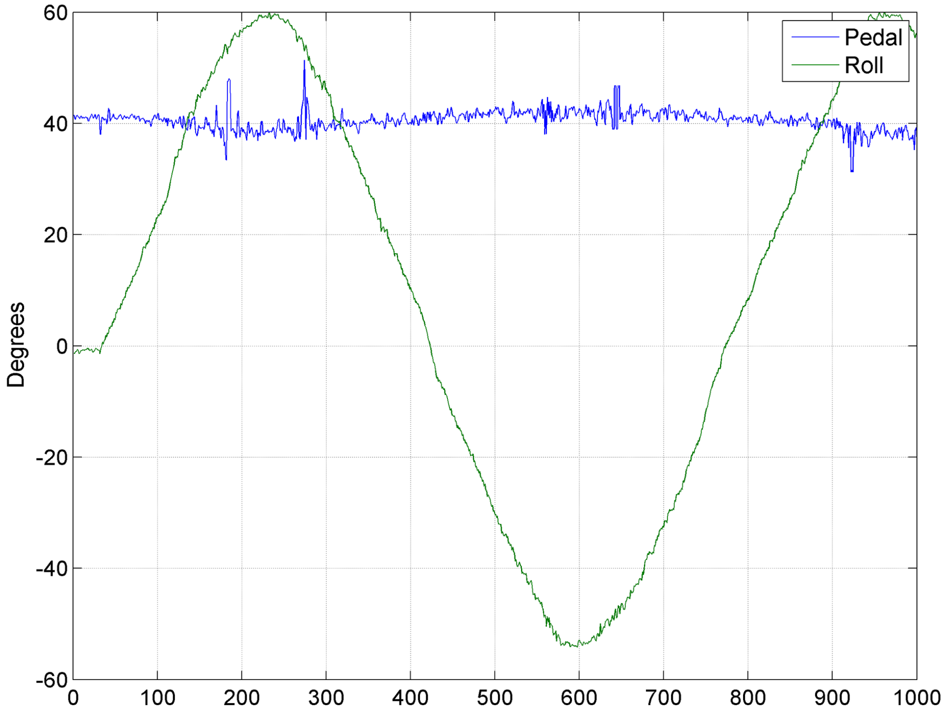


Figure 4.23: Pedal angle measurement versus setup roll discarding acceleration X component

$$a_c = \omega^2 r \tag{4.1}$$

After computing the difference in the centripetal acceleration between the reference and the sensor the resultant value will be subtracted to the sensor global acceleration and the result should have approximately the same modulus as the reference unit. If this method is successfully applied for global dynamics but still fail if the pedal is in motion relative to the vehicle, (which happens when the angular velocity measured by the reference is different from the one measured by the pedal sensor), more variables have to be considered. For this situation, the difference between angular velocities can be used with the vector that links the pedal rotation axis to the pedal sensor origin, to compute the acceleration caused by the relative movement, that will also be subtracted from the global acceleration measured by the pedal sensor. These methods only consider centripetal acceleration. If after testing, this is not enough, the angular velocity measured by the gyroscopes can be derived to obtain the angular acceleration and compute the tangential acceleration, and therefore the global acceleration caused by the rotation movement (eq 4.4 & 4.5). Where a_c is the centripetal acceleration, a_t is the tangential acceleration, ω is the angular velocity, r is the circular trajectory radius and α is the angular acceleration. With these updates, the system should be able to preform in most real situations experienced by any urban cruising car.

$$a_t = \frac{\Delta v_t}{\Delta t} \tag{4.2}$$

$$a_t = r\frac{\Delta \omega_t}{\Delta t} = r\alpha \tag{4.3}$$

$$a_{total} = \sqrt{a_c^2 + a_t^2} \tag{4.4}$$

$$a_{total} = \sqrt{\omega^4 r^2 + \alpha^2 r^2} \tag{4.5}$$

Chapter 5

Conclusions

5.1 Summary

The market available sensor systems can not be used in this project due to their size, specificity and price. For that reason a new sensor setup had to be designed and built to fit this project needs. The sensor network fulfilled its purpose as a easy to install and reduced cost solution, each module can be built costing around 30€ in this prototype stage, for bigger series the units cost would have a significant reduction. With overall dimensions of 21x12x7mm it is the smallest system of its type and smaller than all market solutions, the network can be expanded up to 32 sensor modules and the CANbus line can have lengths up to 100m. This system can also be used for many other applications with or without differential measurement, like humanoid robots balance, measure handicapped people response to electro-stimulation or simply to measure accelerations and rotation velocities on any body and so on. At this time the system differential measurements works for conditions where the sensor modules have their X axis parallel, the gravity component measured in the Y and Z axis are significant enough to allow accurate angle measurement and don't involve fast movements, inertial accelerations or setup roll angles over ±40°. For improved motion differential measurements, the mathematical algorithm needs to be updated, so acceleration differences between sensors can be predicted and more accurate measurements can be achieved.

5.2 Suggestions for future work

There are many things that can be done to improve and complete this work:

- Improve the mathematical algorithm, to handle general dynamic situations;
- Continue the actuation unit development and construction;
- Update the module design when geomagnetic and digital three axis gyroscope become available from STMicroelectronics;
- Add one temperature sensor to the sensor module, to compensate the sensor offset caused by temperature differences;
- Implement a CANbus boot-loader into the sensor modules to avoid the need of a separate programming port and reduce the module size.

This is only the beginning of this project and there's lot of work to be done, the existing system reliability needs be improved and new components have to be developed.

References

[1] Xsens Technologies B.V. Mti leaflet, March 2010.

[2] SMC Corporation. Itv series catalog, December 2009.

[3] E. Foxlin. Head-tracking relative to a moving vehicle or simulator platform using differential inertial sensors. In *PROC SPIE INT SOC OPT ENG*, volume 4021, pages 133-144. Citeseer, 2000.

[4] Eric Foxlin, Yury Altshuler, Leonid Naimark, and Mike Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In *ISMAR '04: Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 212-221, Washington, DC, USA, 2004. IEEE Computer Society.

[5] G. Heydinger. *PORTABLE AUTOMATED DRIVER FOR UNIVERSAL ROAD VEHICLE DYNAMICS TESTING*. PhD thesis, The Ohio State University, 2008.

[6] Analog Devices Inc. Adis16405 datasheet, July 2009.

[7] Analog Devices Inc. Analog devices adis16405 webpage. http://www.analog.com/en/mems/imu/adis16405/products/product.html, online on June 2010.

[8] Kairos Autonomi Inc. Pronto4™ datasheet, September 2007.

[9] MicroStrain Inc. 3dm-gx3-25 datasheet, June 2009.

[10] MicroStrain Inc. Microstrain 3dm-gx3-25 webpage. http://www.microstrain.com/3dm-gx3-25.aspx, online on June 2010.

[11] MEMSense LLC. Nano imu datasheet, December 2009.

[12] MEMSense LLC. Memsens nimu webpage. http://www.memsense.com/index.php/Product-Pages/uimu-miniature-light-weight-3d-digital-output-sensor.html, online on June 2010.

[13] SEA Ltd. Vehicle dynamics brochure, October 2006.

[14] G. Mao and Q. Gu. Design and implementation of microminiature inertial measurementsystem and GPS integration. In *National Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000*, pages 333-338, 2000.

[15] M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (CARMEN) toolkit. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 2436-2441, 2003.

[16] STMicroelectronics. Lis331dlh datasheet, July 2009.

[17] STMicroelectronics. Lpr503al datasheet, July 2009.

[18] G. W Younkin. *Industrial servo control systems: fundamentals and applications*. CRC, 2003.