

Control and Optimization of a Scaled-down Narrow Tilting Vehicle

Dissertation presented by
Sébastien DE LONGUEVILLE

for obtaining the Master's degree in
Electro-mechanical Engineering

Supervisor(s)
Paul Fiset, Julien Hendrickx

Reader(s)
Quentin Docquier, Bruno Dehez

Academic year 2017-2018

Abstract

Narrow Tilting Vehicles (NTV) offer a solution to the increasing traffic congestion as well as an optimization of the energy footprint of vehicles. During a previous work, an electro-mechanical demonstrator was designed and partially realized [20] to optimize the behavior of such a moving vehicle. The present report focuses on the electro-mechanical update, control and validation of the demonstrator, in order to deliver a fully functional embedded controlled system with transmission in real time of the measured and controlled parameters (speed, acceleration, odometry, tilt, torque). It started from the actual status of the demonstrator in the previous work and, based on the feedback from experience, considers the optimization of the vehicle's functionalities to allow the optimal development of the control.

The three axes of development are detailed and the choices are discussed : the mechanical design, the electronics including the nano-computer Raspberry Pi 3B, the transducers, particularly the strain gauges to measure the torque, the data management and the control, the low level integration through the CANopen/TMLCAN and the high level DTC and STC controls. Finally, an experimental validation of the demonstrator is performed and the results obtained are presented and discussed. Various future prospects for the project are proposed before concluding on the validation of the demonstrator.

Acknowledgements

I would like to thank all the people who supported me during this master thesis.

My most sincere gratitude goes to Professor Paul Fisette, for his availability, his comments and advices, numerous and very helpful, throughout the year, and to Professor Julien Hendrickx, my co-promoter.

I would like to thank Quentin Docquier for his commitment into the project and his precious inputs and support all over the year.

Thanks to Sébastien Timmermans for the various talks we had about the evolution of the project.

Many thanks to the following people for their time and involvement during the whole year : Thierry Darras in electronics, for his support in hardware debugging, André Lengelé and Axel Jottard for their expertise and tips in machining, Alex Bertholet for his expertise in strain gauges design, Bruno Dehez and Benoît Herman for the constructive discussions we had on mechanical design and sensors.

I would also like to thank Didier Vanherck, from Technosoft support team, for his help to implement applications that were outside of the manufacturer standard procedures.

Finally, I want to thank my family and friends for their support and patience during the realization of this master thesis, in particular, my mother, Dominique de Longueville, and my uncle, Bernard de Longueville, for their careful review and precious suggestions.

Table of contents

Abstract	i
Acknowledgements	iii
List of symbols, abbreviations and patented references	vii
List of Figures	ix
List of Tables	xii
Introduction	1
Context	1
Research incentives	1
Specific features of Narrow Tilting Vehicles	1
Pre-existing demonstrator overview	2
Objectives	5
Chapter 1 : Electro-mechanical upgrades - Technical description	7
1.1 Mechanics	7
1.1.1 Frame Stiffening and backlash reduction	7
1.1.2 Anti-tipping mechanism	8
1.1.3 Rear gear ratio modification	8
1.2 Torque sensor	9
1.2.1 Background - Justification of the choice	9
1.2.2 Mechanical design	11
1.2.3 Signal processing	15
1.2.4 Calibration	15
1.3 Tilt sensor	16
1.3.1 Background - Justification of the choice	16
1.3.2 Mechanical design	17
1.3.3 Signal processing	17
1.4 Inertial unit and magnetometer	18
1.4.1 Background	18
1.4.2 Signal processing	19
1.5 Power managment	20
1.5.1 Battery charging	20
1.5.2 Protection circuit updated dimensioning	20
Chapter 2 : Data processing and communication architecture	21
2.1 Hardware components	21
2.1.1 Nano computer - Raspberry Pi 3B	21
2.1.2 CAN physical layer	22

2.1.3	Driver interface board	23
2.1.4	Main interface Board	24
2.1.5	Remote control	25
2.2	Software programming	27
2.2.1	Brain configuration	27
2.2.2	ROS	27
2.2.3	CAN/CANopen communication	30
2.2.4	CANopen main features	31
2.2.5	Robotran interface for ROS	32
2.2.6	Remote control	33
Chapter 3 : Updated demonstrator configuration		35
3.1	Mechanical configuration	35
3.2	Global electric architecture	37
3.3	Global ROS Architecture	39
Chapter 4 : Vehicle control		41
4.1	Low-level control	41
4.1.1	Propulsion motor	41
4.1.2	Steering motor	42
4.1.3	Tilting motor	43
4.2	High-level control	45
4.2.1	DTC control	45
4.2.2	STC control	45
4.2.3	Protection mechanism	46
Chapter 5 : Experimental Validation		47
5.1	Untilted vehicle	47
5.1.1	Straight line Acceleration	47
5.1.2	Cornering vehicle	48
5.2	DTC controlled vehicle	49
5.2.1	Steady-state behavior	50
5.2.2	Trajectory variation	50
5.3	STC controlled vehicle	52
5.4	Localization	53
5.5	Feedback from the experimental results	53
5.5.1	Rear wheel misalignment	54
5.5.2	Tilting actuation	55
5.5.3	CAN communication	55
5.6	Conclusions of the experimental validation	56
5.7	Future prospects	56
Conclusion		59
Bibliography		61
Appendices		a
Appendix A : Torque sensor		c
Appendix B : Vehicle's electrical configuration		g
Appendix C : Printed Circuit Boards		k
Appendix D : Robotran/Simulink Interface		m

List of symbols, abbreviations and patented references

Abbreviation

ADC	Analog to Digital Converter
CAD	Computer Aided Design
CAN	Controller Area Network : digital communication bus
CANopen	Application layer developed for can bus
CiA402	CAN in Automation : CANopen profile for motion control
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DTC	Direct Tilting Control
EKF	Extended Kalman Filter
ESTRN	Equivalent strain (in Solidworks environment)
GPIO	General Purpose Inputs and Outputs
IMU	Inertial Measurement Unit
MBS	MultiBody System
NTV	Narrow Tilting Vehicle
NMT	Network Management Tool
PCB	Printed Control Board
PID	Proportional Integral Derivative (controller)
PWM	Pulse Width Modulation
ROS	Robot Operating System
SDO	Service Data Object
SDTC	Steering-Direct Tilting Control (hybrid control mode)
SPF	Single Point of Failure
SPI	Serial Peripheral Interface (communication protocol)

SSH	Secure Shell (remote control protocol)
STC	Steering Tilting Control
PDO	Process Data Object
TCP	Transmission Control Protocol
TMLCAN	Technosoft Motion Library for CAN
UART	Universal Asynchronous Receiver Transmitter (hardware device for communication)
URES	Equivalent deformation (in Solidworks environment)
VGA	Video Graphics Array : Display standard (connectors and protocol)

Symbols and parameters

A	Cross section [m^2]
δ	Steering angle [$^\circ$]
E	Young's modulus [$kgm^{-1}s^{-2}$]
ε	Strain [/]
f	Frequency [Hz]
F	Force [N]
F_{res}	Resulting force [N]
G_F	Gauge Factor's [/]
G_{Rx}	Receiving gain [dBi]
G_{Tx}	Transmission gain [dBi]
i	Current [A]
i_{mes}	Measured current [A]
i_{ref}	Reference current [A]
L	Wheelbase of the NTV [m]
m	Mass [kg]
P_{Tx}	Emission power [dBm]
R	Resistance [Ω]
r	Curvature radius of the NTV [m]
σ	Stress [$kgm^{-1}s^{-2}$]
σ_y	Yield point [$kgm^{-1}s^{-2}$]
S_{Rx}	Receiver sensitivity [dBm]
T	Torque [Nm]

T_{mes}	Measured torque [Nm]
T_{ref}	Reference torque [Nm]
ψ	Yaw angle [°]
$\dot{\psi}$	Yaw rate [°/s]
θ	Tilting angle [°]
θ_{ref}	Reference angle [°]
V	Tangential speed of the NTV [m/s]

Hardwares and softwares

Gazebo	ROS multibody simulator
Linux	Open source operating system
Robotran	Software to model and analyze multibody systems
Odroid Xu4	Single board computer from Hardkernel Co
Pixhawk Px4	Flight auto-pilot
Raspberry Pi	Nano-computer
Solidworks	CAD software
TranCalc	Simulation software

List of Figures

Figure 1	Narrow Tilting Vehicle illustration [4]	1
Figure 2	Illustration of Narrow Tilting Vehicles	2
Figure 3	Initial scaled model [23]	3
Figure 4	Actuation axes of the vehicle [23]	3
Figure 5	Front and rear suspensions of the initial scaled model [23]	3
Figure 1.1	T-pendulum within the steering suspension	8
Figure 1.2	New steering backing (in blue on the vehicle) [23]	8
Figure 1.3	Anti-tipping mechanism implementation	9
Figure 1.4	Modified gear ratio	9
Figure 1.5	Torque sensor implementation (left) and CAD design (right)	11
Figure 1.6	Sensor integration in the front suspension (in blue, the solidary parts of the strain gauge)	12
Figure 1.7	Strain Gauges positioning	12
Figure 1.8	Calibration set-up of the torque sensor	13
Figure 1.9	Calibration set-up of the torque sensor	16
Figure 1.10	Experimental characterization of the torque sensor	16
Figure 1.11	Tilt sensor implementation (left) and CAD design (right)	18
Figure 1.12	Position and orientation of the different frames : Vehicle's frame (red) and Pixhawk Px4's frame (blue)	19
Figure 1.13	Pixhawk Px4's frame positioning	19
Figure 1.14	Circuit diagram of the power supply and warning system of the 12 and 24V batteries	20
Figure 2.1	SPI-CAN interface board [5]	22
Figure 2.2	Main interface board of the scaled model	24
Figure 2.3	Outlines of the main interface board	25
Figure 2.4	Level shifter circuit	25
Figure 2.5	Simplified software architecture	28
Figure 2.6	ROS communication architecture [15]	29
Figure 2.7	CAN Software interface of Linux kernel	31
Figure 2.8	CANopen layers structure	32
Figure 2.9	Robotran embedded implementation	33
Figure 2.10	Web interface of the vehicle, allowing remote control (left) and telemetry (right)	34
Figure 3.1	2017 Vehicle Assembly (upper view)	35
Figure 3.2	2018 Vehicle Assembly with main updates (upper view)	35
Figure 3.3	2017 Vehicle Assembly (isometric view)	36
Figure 3.4	2018 Vehicle Assembly with main updates (isometric view)	36
Figure 3.5	Electrical scheme of the prototype (adapted from [23]). 2017 initial electric configuration of the vehicle and updates are given in Appendix B	37

Figure 3.6	Global ROS architecture	39
Figure 4.1	Schematic representation of the speed control of the propulsion motor M .	42
Figure 4.2	Schematics of the position control of the steering motor M in DTC control mode	42
Figure 4.3	Schematics of the torque control of the steering motor M in STC control mode	43
Figure 4.4	Schematic representation of the position control of the titling motor M .	43
Figure 4.5	Schematic representation of the torque regulation of the titling motor M .	44
Figure 4.6	Torque control response of the tilting actuator, for a reference torque of 1.1 Nm	46
Figure 4.7	Mechanical limits of the front suspension, function of the steering (δ) and tilting (θ) of the vehicle[23]	46
Figure 5.1	Time variation of the vehicle's speed and its set point, for a straight line trajectory	48
Figure 5.2	Vehicle behavior (tilting torque, tilting angle and tangential speed) time evolution for various cornering speeds, for $\delta_{FD} = 28^\circ$ and $\theta = 26^\circ$	48
Figure 5.3	Vehicle's behavior evolution for different actuated tilting angles	49
Figure 5.4	Vehicle's behavior for DTC control, with $V=1\text{m/s}$ and $r=3\text{m}$	50
Figure 5.5	Vehicle's behavior for an tilted cornering entry, with $V=1\text{m/s}$ and $r=3\text{m}$.	51
Figure 5.6	Corner entry	52
Figure 5.7	Corner exit	52
Figure 5.8	Vehicle behavior (tilting torque, tilting angle and tangential speed) time evolution for various curvature radii, for $\delta_{FD} = 28^\circ$ and $\theta = 26^\circ$	53
Figure 5.9	Theodolite measurement set-up (right) and prism location on the NTV (left)	54
Figure 5.10	Vehicle position measurements using the theodolite	54
Figure 5.11	Vehicle velocity computation using the theodolite	54
Figure 5.12	Vehicle rattle apparition	55
Figure 5.13	Real wheel tensioning and alignment system [23]	55
Figure 5.14	Cornering scaled model	60
Figure A.1	Evolution of the design of the T-pendulum	c
Figure A.2	Full Wheatstone bridge configuration of the strain gauges	d
Figure A.3	Binoculaire design simulations using TransCalc	d
Figure B.1	Former electrical scheme of the prototype [23]	g
Figure B.2	Former electrical scheme of the prototype [23]	h
Figure B.3	Former electrical scheme of the prototype [23]	i
Figure C.1	Modified driver interface board, 2017-2018	k
Figure C.2	Interface board for sensors, Raspberry Pi and Pixhawk Px4, 2018	k
Figure C.3	Instrumentation amplification board of the torque sensor - UCL, 2012 . .	l
Figure C.4	Excitation and Powering board of the torque sensor - UCL, 2012	l
Figure D.1	Simulink implementation of the kalman filtre	m
Figure D.2	Theoretical kart pendulum	n
Figure D.3	Robotran implementation of the kart pendulum	n
Figure D.4	Linearized matrices computed analytically	o
Figure D.5	Linearized matrices computed using Robotran	o

List of Tables

Table 1.1	Pros and cons of decoupling methods	10
Table 1.2	Deformations simulations of the T-pendulum under the nominal bending torque (11Nm) and under a torsion load of 1.1Nm at its upper tip	14
Table 1.3	Update protection threshold values	20
Table 2.1	Pros and cons of remote control solutions	26
Table 2.2	Remote communication experimental characterization	26
Table 3.1	Electronic components of the demonstrator, adapted from [23]	38
Table A.1	Simulations of the non-scooped T-pendulum bending and torsion deformation	e

Introduction

Context

Research incentives

Nowadays urban traffic skyrockets and, still, cars are barely ever used at full load capacity. Transportation and road infrastructures as well as commuters' behaviours are only slightly improving to face this challenge of traffic congestion that has obviously multiple implications. Indeed, economics, politics or ecology are to be taken into consideration to "write the equation", in addition to the natural personal reluctance to modify a consumer behaviour.

It is against this background that a CEREM team is conducting research at the Université Catholique de Louvain (UCL) in the field of the narrow tilting vehicles, aiming at optimizing the energy footprint of vehicles.

The validation of the theoretical results is based on a tailor made scaled-down Narrow Tilting Vehicle (further referred as NTV).

During previous research works, an electromechanical demonstrator was designed and realized [23] to optimize the behavior of the moving vehicle. Its performance was assessed, partially validated and documented.

The present work focuses on electromechanical update, control and validation of the vehicle. It started from the actual status of the demonstrator and, based on the feedback from experience, considered the optimization of its functionalities to allow the optimal development of the control.

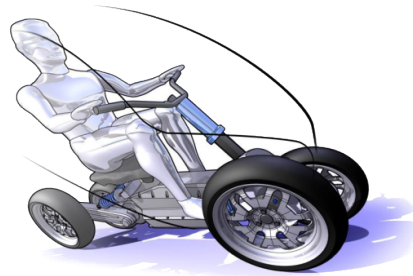


Figure 1: Narrow Tilting Vehicle illustration [4]

Specific features of Narrow Tilting Vehicles

Both motorbikes and NTV have an important weight and a high centre of mass. Both vehicles require a specific training to be driven. In the case of the NTV, the automatization of its tilting enables to reduce this training and allows the driver to use his skills of regular car driver to operate a NTV.

The NTV includes basically the suspension of a regular motorbike on which one additional tilting degree of freedom is added between the vehicle's frame and its suspension.

Two specific control methods can be applied to enable the control of the vehicle in an optimized way: the Direct Torque Control (DTC) and the Steering Tilting Control (STC).

The **DTC** (Direct Torque Control) considers that the wheels are mechanically linked to the steering wheel through a steering fork and that the motion is instantaneously transmitted by the pilot of the vehicle. The tilt of the vehicle is then automatically controlled by an actuator which tilts the frame towards the inner side of the curve, in order to counteract the centrifugal effect. The DTC is well adapted for low speed but can become very unstable at high speed in addition to require a lot of energy to operate.

For the **STC** (Steering Tilting Control), the wheels are steered independently from the steering wheel. Indeed, the steering is controlled by the vehicle which first counter-steers¹ to benefit from lateral dynamics of a cornering vehicle in order to induce a variation of its tilting angle. Once the vehicle is properly tilted², the controller will start steering in the direction dictated by the steering wheel. The STC is stable at high speed because those kinds of controllers require tilting self-stability of the vehicle (which is not achieved at low speed). Figure 1 below shows a NTV in two situations : standstill and moving as well as the corresponding simplified free-body diagrams.

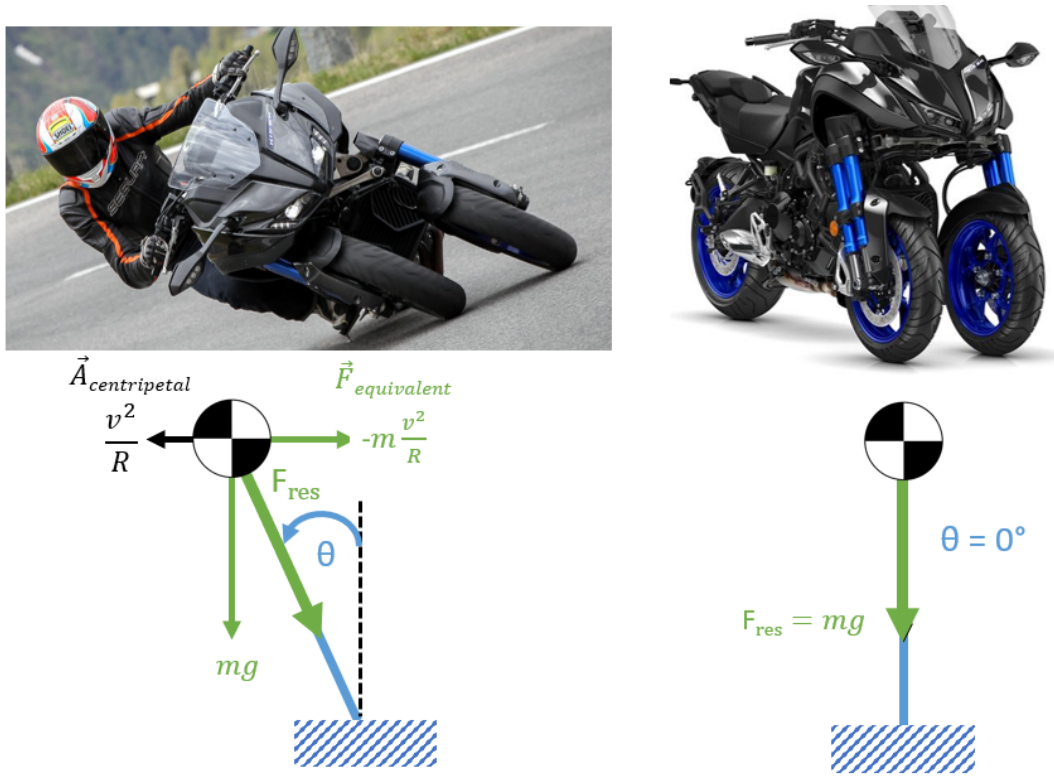


Figure 2: Illustration of Narrow Tilting Vehicles

The DTC and STC controls are therefore adapted to a given working speed range and will be the topics of the present project, both controls being implemented and the DTC control being experimentally validated.

Pre-existing demonstrator overview

The initial scaled model [23] has two adjustable front wheels and one propulsive rear wheel. It is made of a front double wishbones suspension without damping system, one unsprung propulsive

¹according to the expected steering to follow the cornering target provided by the steering wheel

²with respect to the target cornering radius and the travel speed of the system and thus the centripetal acceleration in such a configuration

rear wheel and a frame, constituted of folded metallic plates, welded at their ends.

The front suspension lay-out can be modified according to the needs by adjusting the following parameters : camber angle, toe-in angle, caster angle, trim angle, ground offset, ground clearance,... In addition to that, three of its freedom degrees are actuated : the steering, the frame tilting and the propulsion.

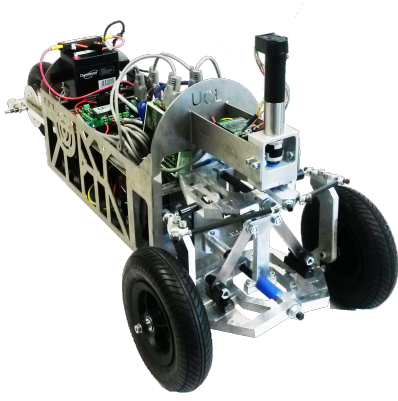


Figure 3: Initial scaled model [23]

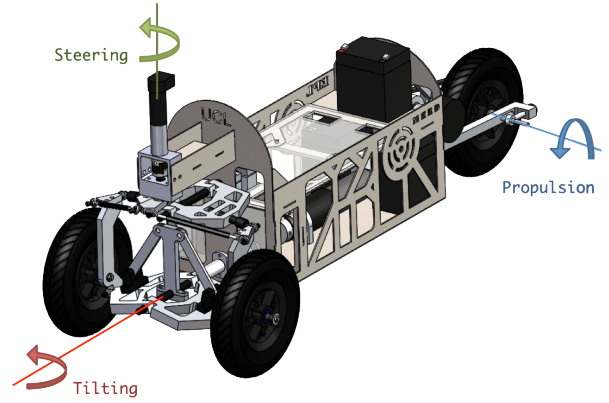


Figure 4: Actuation axes of the vehicle [23]

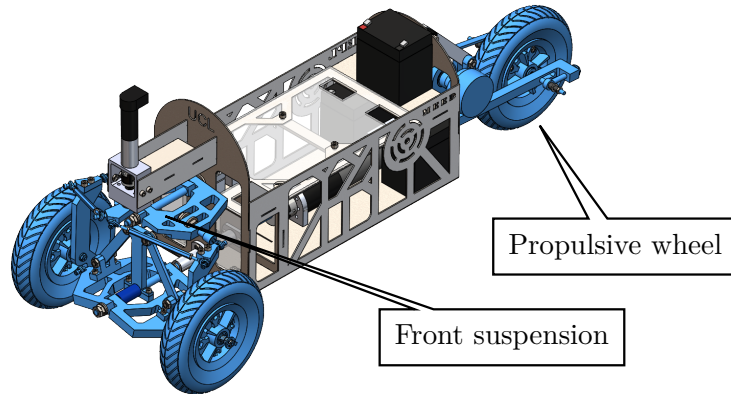


Figure 5: Front and rear suspensions of the initial scaled model [23]

From an electrical point of view, the vehicle contains a power supply, as well as safety systems (over-voltage and high currents protection and emergency stop). It has three motors at each freedom degree and their associated drivers as well as the set of sensors for their low level control and the motherboard interconnecting them.

Finally, two nano-computers were available for the control of the vehicle without being connected to the electrical architecture of the NTV. Given this, the mass of the initial demonstrator was 21,5 kg including the electronic boards, the cables and the batteries.

The status of the demonstrator at the beginning of the study allows to identify the functional existing devices as well as some optimizations to be implemented :

The already implemented items [23] were the followings :

- the mechanical design and assembly.
- the model of the NTV dynamics through multibody modelling.

- the power electronics : power supply from batteries and voltage converter.
- the steering and propulsive individual motors controlled from an external computer.

As far as we are concerned, the following items were to be updated to allow the vehicle to be autonomous, controllable and robust:

1. Mechanics

- Some backlash was observed as well as a lack of stiffness, elastic and plastic deformation of some parts of the vehicle, impacting its tilting capability. This concerned the frame and the front suspension.
- The actual gear ratio on the initial demonstrator did not allow a sufficient high vehicle's acceleration at start-up.

2. Electronics

- A functional interface between the motor and its tilting controller was not implemented, leading to an inoperative tilting control.
- No functional CAN interface exists. The motors could therefore exclusively be activated one by one, using serial connections.
- A CAN conversion source is missing for the embedded computer of the vehicle, as well as a high level CANopen and TMLCAN interface³. Therefore, no centralized motor control could be performed directly on the vehicle.
- No sensors for the torque and the *absolute* position of *tilting* and *steering* were implemented. Therefore the control of the vehicle required a complex manual calibration at each start-up.
- An embedded power supply (battery) for the embedded computer and the Pixhawk Px4 auto-pilot was not available. As no communication between them existed (see above), no link between them could therefore be established and, consequently, the Px4 was unusable.
- A battery charging device was missing for the vehicle. Thus, a time-consuming procedure was necessary to remove and charge the batteries, limiting the availability of the vehicle.
- A remote communication device was not implemented. Therefore, the control of the freely moving vehicle was not possible.
- The emergency stop was not functional on the demonstrator. As a consequence, the safety of the moving vehicle could not be guaranteed.
- The programming of a central unit was not implemented. Therefore, no control or sensors data export was possible.

3. Control

- The programming of the nano-computers to be embedded in the vehicle had still to be developed. Thus the communication with the motors and their control as well as the external communication with the end-user and the peripherals (sensors, interfaces) was not possible.

³Those two protocols are applicative protocols used to control remotely motors through a CAN communication bus

- The DTC and STC high level controls were not implemented. The control of the vehicle was not feasible.
- No disengagement of the tilting motor from the rest of the vehicle (for STC control) was implemented. Therefore, the vehicle could not be set in a physical configuration allowing STC control.

From these inputs, the objectives of the present work can be settled.

Objectives

Following on from the last year's Master Thesis [23], the main objectives of this thesis are to implement the following upgrades and developments, in order to deliver a fully functional **embedded controlled** system.

The current work focuses on three axes :

1. Mechanical design
2. Electronics including the communication, the transducers and the data management.
3. Control, including the low level through the CANopen/TMLCAN and the high level DTC and STC controls.

Mechanics

Mechanical design - In order to increase the closeness between the demonstrator and the simulated multibody model, the mechanical backlash must be reduced and the frame's stiffness increased.

Gear ratio - The gear ratio will be modified to allow a higher vehicle's acceleration at start-up to comply with the need of the study.

Electronics

Communication - A dialogue will be established between the central architecture of the vehicle, the drivers, the motors as well as the remote control and the end user.

Sensors - Sensors will be implemented to provide position, torque and acceleration information about several degrees of freedom of the vehicle, with the aim of developing the most exhaustive representation of the vehicle's state, and therefore allowing for the development of more complex controllers, user-friendly start-up procedures and dynamic models close to the real vehicle.

Power management - Two main improvements of power management will be achieved. On one hand, the modification of the power protection board will be carried out in order to comply with the end-of-charge properties of the batteries instead of their nominal characteristics. On the other hand, a charging interface for the batteries and a deep discharge warning system to protect them will equip to the demonstrator.

Control

The development and programming of the controls, the communication protocols and the data acquisition systems of the vehicle will be performed. The objective is to gather all relevant information and to use them for direct (DTC) and indirect (STC) controls.

Report structure

The report is divided into four main sections :

First, a technical description of the electro-mechanical upgrades that were implemented is presented. These upgrades concern the vehicle's frame itself, the torque and the tilting sensor, the inertial unit, the magnetometer and the power management. A synthesis shows the final mechanical design, as well as the global electric architecture of the studied demonstrator.

The second section deals with the data processing and communication architecture. It details and justifies the choices that were made for the hardware components : the nano-computer, the CAN, the driver interface and the main interface boards. The software configuration and programming that were retained and developed for the project are detailed : the brain configuration, ROS, the CAN/CANopen communication, the Robotran interface for ROS and the remote control.

The third part of the report addresses the vehicle control and its specificities. The low level control of the motors is firstly discussed, introducing the different control modes setup for each actuator. Then, the high level control of the vehicle is detailed, including both DTC and STC controls, as well as the protection mechanism.

The fourth section addresses the experimental validation of the demonstrator. This is performed starting from unitary tests on the demonstrator to assess the motor control, the CANopen communication, the remote control and the data processing, before focusing on the DTC and STC controls.

Finally, the results obtained are presented and discussed, leading to define future prospects for the project and draw the general conclusion of the present work.

Chapter 1

Electro-mechanical upgrades - Technical description

In this chapter, the electro-mechanical upgrades of the vehicle are described and structured around two main axes : the mechanical components and the sensors.

Maintaining the stiffness and the integrity of the assembly becomes an issue due to the overall dimensions of the scaled vehicle (1.015m long, 0.41m wide and 0.452m high [23]) and its mass (28kg in its current version, compared to 21.5kg previously). On the other hand, reaching high enough acceleration to support efficiently the validation tests is also an issue. The update of the mechanical design is presented in this chapter.

On top of that, sensors were developed and implemented on the demonstrator to provide position and torque information about several degrees of freedom of the vehicle to allow for more complex controls and dynamic models close to the real vehicle. The autonomy of the vehicle was increased by implementing the charging system of its batteries as well as the protection circuits. The second part of the chapter details the different devices and their justification.

1.1 Mechanics

1.1.1 Frame Stiffening and backlash reduction

Some components were replaced and thickened based on a revised design from the previous work: the T-pendulum and the steering backing.

The T-pendulum is the "T" shaped load strut of the front suspension, responsible for the torque transmission between the vehicle and the frame, as detailed in Section 1.2.2. The T-pendulum's bending is used to measure the torque through strain gauges, as detailed later in the report. The T-pendulum of the current demonstrator is shown in Figure 1.1.

The modified steering backing of the demonstrator can be seen in Figure 1.2.

In addition to these modifications, repair and reinforcement of the main frame by welding were performed to stop the crack propagation and to guaranty an optimized stiffness of the whole body in motion.

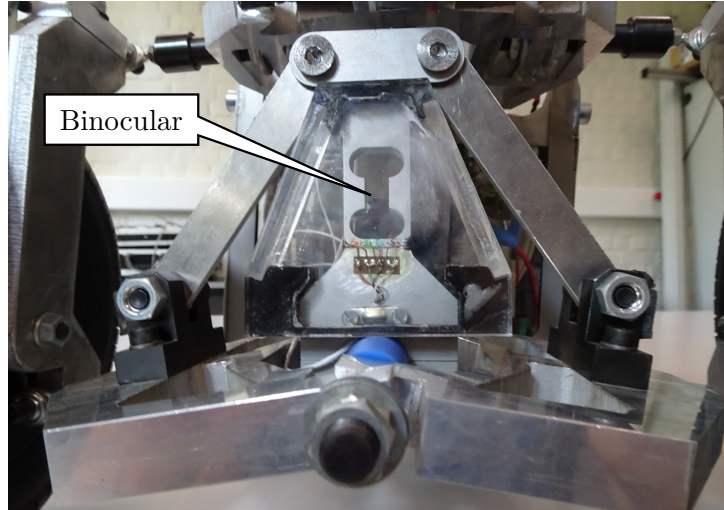


Figure 1.1: T-pendulum within the steering suspension

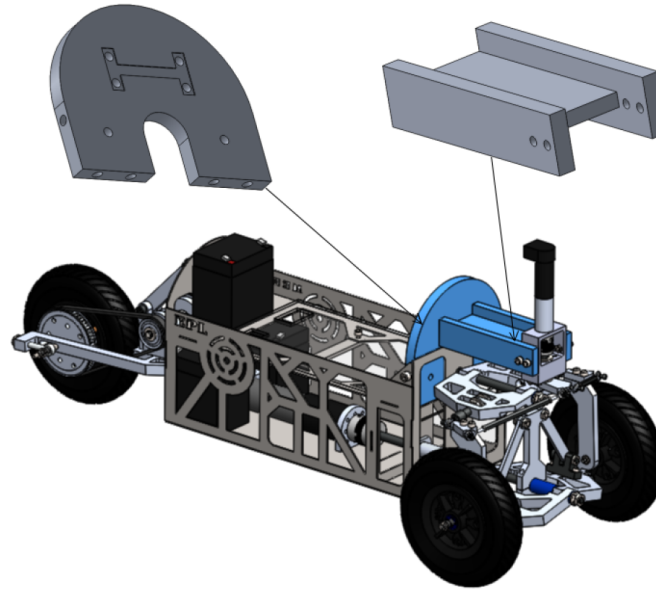


Figure 1.2: New steering backing (in blue on the vehicle) [23]

1.1.2 Anti-tipping mechanism

A very simple device, based on polystyrene cylinders, was built to protect and retain the vehicle in case of unexpected unballasting during the test phase, see Figure 1.3.

1.1.3 Rear gear ratio modification

Given its limited propulsion torque, the initial demonstrator could not reach sufficient speed to carry-out properly most tests, 1 to 3 m/s being indeed needed. In order to increase the rear wheel torque and therefore to increase the maximal acceleration at start-up, the gear ratio was increased from 60:24, (i.e. 2.5) to 60:14, (i.e. 4.2) which is the maximum reduction ratio attainable without redesigning the rear suspension, see Figure 1.4.

Remark : As it was not possible for this practical reason to increase even more this ratio, an

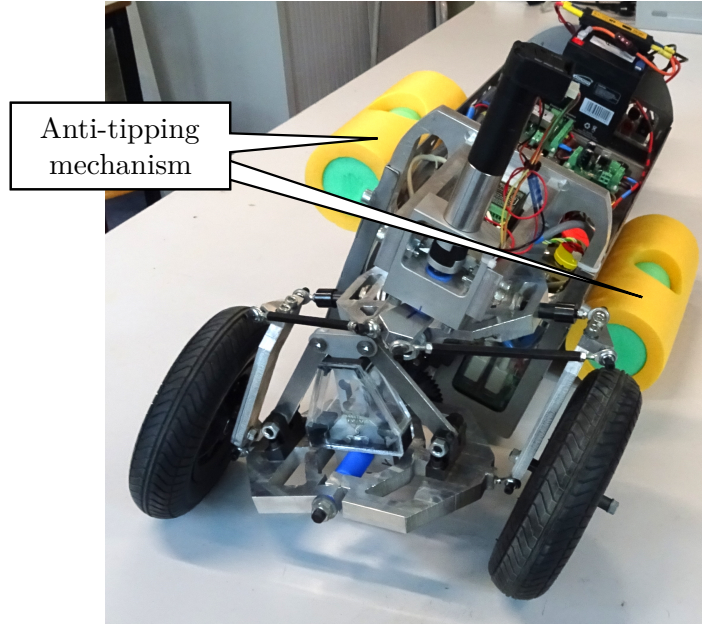


Figure 1.3: Anti-tipping mechanism implementation

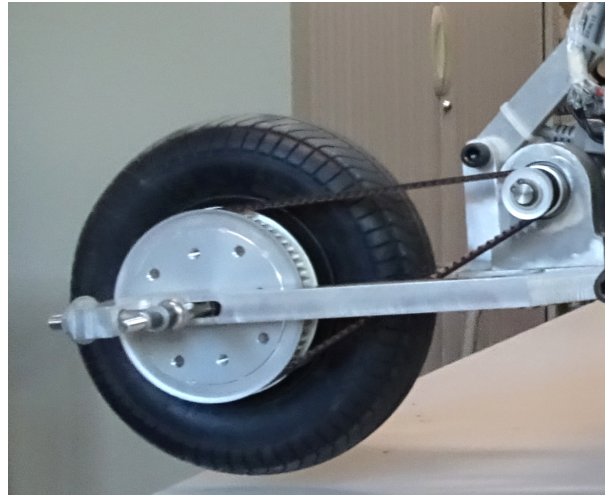


Figure 1.4: Modified gear ratio

additional action was taken through the control board to increase the maximum instantaneous current authorized for the propulsive motor.

1.2 Torque sensor

1.2.1 Background - Justification of the choice

Considering the high friction torque induced by the powerful tilting motor and its gearbox, one of the major challenges of this thesis was to implement the STC control of the vehicle as well as the possible switch between the different control modes of the demonstrator (DTC and STC).

The advantage of such an automatic switch would be to support the future implementation of the hybrid mode SDTC.

The theoretical hypothesis that the frame is disconnected from its actuator in tilting is far from the reality due to the high friction induced by the powerful tilting motor and its gearbox. Indeed this friction is high enough to counteract every tilting motion induced by the lateral dynamic of the NTV blocking completely the STC normal operation.

Removing or decreasing this resistive torque applied by the motor on the frame can be achieved through four different approaches :

Mechanics By physically disconnecting the motor's drive shaft from the rest of the scale model. This can be carried out in a purely manual/mechanical way, whilst a more sophisticated alternative could involve the use of an electro-mechanical device such as a permanent magnet clutch or brake.

Modelling By accurately characterizing the electro-mechanical and friction models of the tilting motor and its gearbox in order that simulations allow to quantify the impact of this actuator on the model and thus to counterweight it.

Control By measuring the resistive torque developed by the motor on the frame and using it as a state feedback information for the zero torque control of the motor, in order to control it and cancel the impact of the torque on the frame.

Motor replacement By replacing the largely oversized tilting motor with a less powerful one and reducing its gearbox ratio, with the aim of decreasing the friction caused by those two components. This cutback should, in turn, enabling the steering control to induce effective tilting variation, which is not possible with the present motor, for which a too large friction avoids any unactuated tilting.

The pros and cons of these different decoupling methods are listed in Table 1.1.

	Advantages	Drawbacks
Mechanics (Clutch/Brake)	• Simple implementation • Affordable component's price – Electro-magnetic : $\pm 400\text{€}$ – Mechanical : $\pm 150\text{€}$	• Requires lots of mechanical adjustments ¹ • Complex transition between both controls
Modelling	• No need for additional devices	• Time consuming (complex identification for accurate characterization) • Increased complexity of multibody simulations
Control	• Affordable homemade sensor (300€) • Easy mechanical implementation • Simplifies hybrid (SDTC) control implementation • Provides real torque information (for validation,...)	• High prices (1200€) of the industrial sensors
Motor replacement	• Simple implementation • Reduces motor friction	• Friction is physically unavoidable

Table 1.1: Pros and cons of decoupling methods

¹given the restricted internal volume of the vehicle

Given the above, it was decided to work on the optimization of the control features and implement strain gauges on the NTV (see Section 1.2.2), based on the additional following reasoning :

Strain gauges allow :

- to carry out a control compatible to the multibody simulation;
- an automatized transition between STC and DTC controls;
- to benefit from a complementary information related to the torque really applied on the frame, which is used as a input data for the NTV state in the multibody study and therefore is very valuable;
- to avoid large modification of the NVT as only the T-pendulum has to be modified, compared, for example, to the addition of a clutch requiring deep modifications of the NTV.

Finally, the most significant disadvantages of the sensor (price and workload associated) are compensated by the relevant UCL expertise in the field, allowing for availability of dedicated electronic circuits as well as an effective technical support for the design of the sensor.

1.2.2 Mechanical design

The main purpose of this strain gauge is to provide a torque information associated with the actuated tilting degree of freedom of the vehicle. This torque information will thereafter be used for control and validation purposes. These objectives will be achieved namely by allowing the motor driver to perform a torque regulation. based on the real tilting torque applied on the demonstrator's frame ², or to prove steady-state self-stability of the controlled vehicle³.

As shown in Figures 1.5 and 1.6, this sensor was designed on the existing "T" shaped load strut of the vehicle (further referred as the "T-pendulum"). This pendulum transmitting directly the tilting torque to the vehicle's frame, a simple way to measure this torque is to measure the bending of this pendulum using strain gauges on a proof specimen.

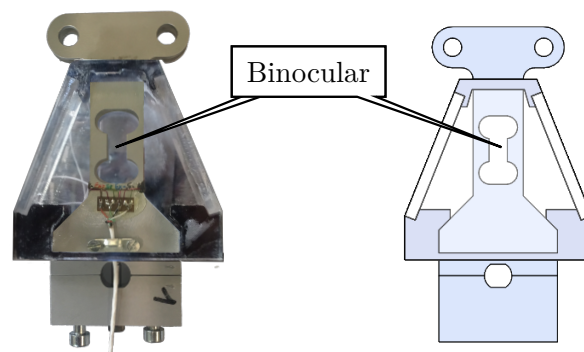


Figure 1.5: Torque sensor implementation (left) and CAD design (right)

²This measurement considers thus implicitly non-idealities of the inclination drive chain such as the internal friction of the drive chain and is thus similar to the ideal "joint forces" computed by Robotran

³which is the case when the resulting inclination torque measured on the frame is zero

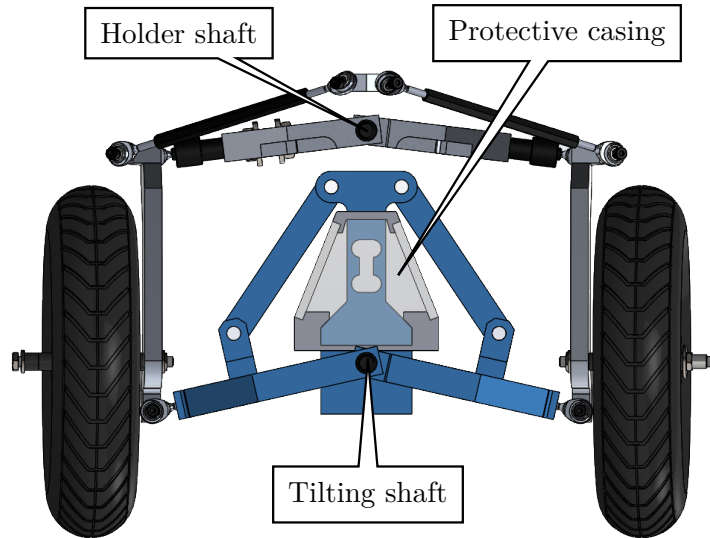


Figure 1.6: Sensor integration in the front suspension (in blue, the solidary parts of the strain gauge)

The implementation was carried out as follows :

1. **Dimensioning** - Sizing of the binocular, with the aim of maximizing the deformation of the proof specimen at the strain gauges and consequently the sensitivity of the torque sensor. The positioning of the strain gauges on the binocular can be seen in Figure 1.5
2. **Production** - Machining of the deformation element and its binocular, gauges sticking and data acquisition system roll-out.
3. **Calibration** - Tuning of the measurement and amplification unit of the sensor to match the required specifications (Gain, Zero-offset and Range).

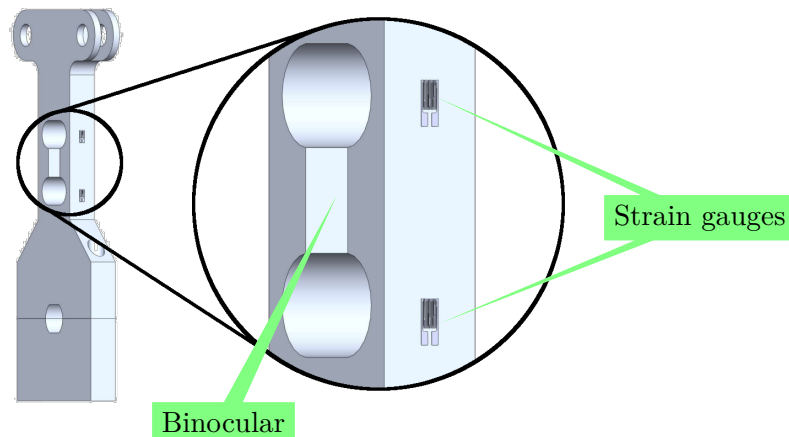


Figure 1.7: Strain Gauges positioning

Protective casing

This shield (see Figure 1.6) was developed to prevent the load cells from being slightly moved around because of a shock with an element coming from the sensor's environment, which would

lead to an irreversible drift of the sensors measurement (deformation variation, unbalancing and thus irreversible drift).

Test specimen design

The purpose is to maximize the deformation of the proof specimen at the gauge's level, for an applied bending moment. On the other hand, it was also critical to ensure that this deformation would not lead to a plastic deformation of the T-pendulum or the strain gauges, which would irreversibly lead to an accuracy drift of the sensor, or even worse, to a deformation body breakage.

The following simulations were thus carried-out :

First, the required measurement range of the sensor was determined, based on the permissible maximum centripetal acceleration of the vehicle and its corresponding unballasting tilting torque (in DTC control mode, [23, p28-29]), which leads to bounds of $\pm 11\text{Nm}$ and therefore to a sensing range of 22Nm . This torque should be theoretically the highest applied to the demonstrator's frame (measured via the pendulum), given that STC control does not imply any tilting actuation of the frame.

Computations were then carried-out using TransCalc simulation software (see Appendix A.3) to determine the binocular dimensions and gauges relative positioning, with the aim of getting as close as possible from the nominal gauge strain. To this aim, the T-pendulum element was approximated as a perfect beam and the maximal applied torque (11Nm) was replaced by an equivalent force ($\frac{11\text{Nm}}{0.1\text{m}} = 110\text{N}$) applied at the span end of the beam.

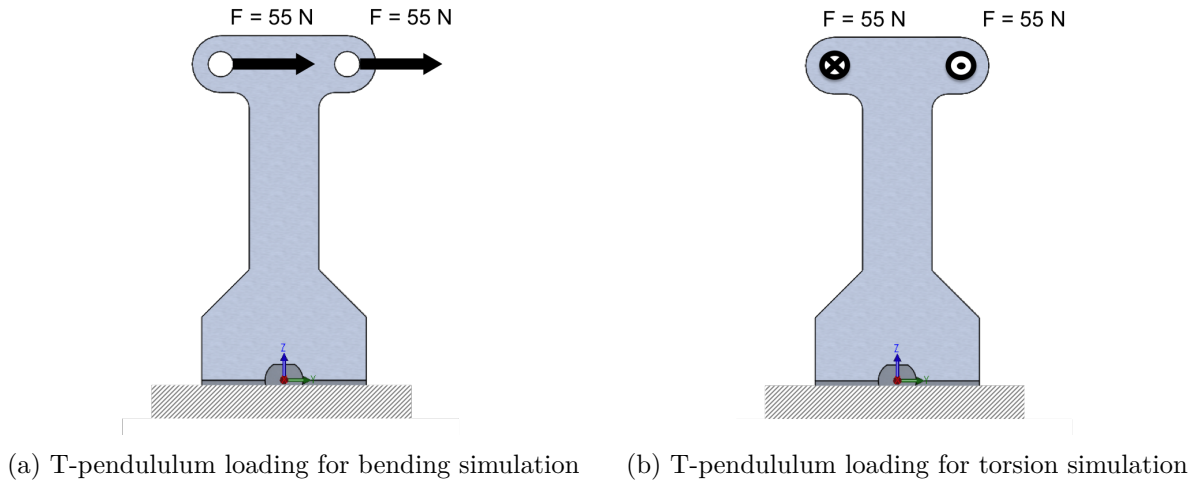


Figure 1.8: Calibration set-up of the torque sensor

Lastly, internal stress and strain simulations were realized using the of Solidworks CAD software on the T-pendulum (including its binocular). Those results (Table 1.2 and Appendix A) were used to validate the beam approximation of the pendulum and to determine the highest Von Mises equivalent stress endured by the distorted pendulum. This equivalent stress was then compared to the yield strength of the component material [6] [16].

The bending modelling was performed by applying this maximal load (perpendicular to the sticking plane of the gauges) corresponding to the maximum working load of the assembly at the upper bore holes). Those simulations led to deformation results close to the ones previously computed ($630\text{ }\mu\text{S}$ obtained by Solidworks simulation to be compared to the $755\text{ }\mu\text{S}$ obtained

using TransCalc⁴). From that, the computed deformation allows to validate the sizing of the binoculars and the results obtained using a beam approximation for the pendulum.

In addition, the maximum internal stress resulting from this simulated loading ($\sigma_{eq} = 59MPa$) enables to validate that the deformation of the test specimen for the given material (Al 7075) remains in the elastic domain as this constraint is well below the yield point of the material (420 MPa).

The resulting design evolution of the T-pendulum is shown in Appendix A.1.

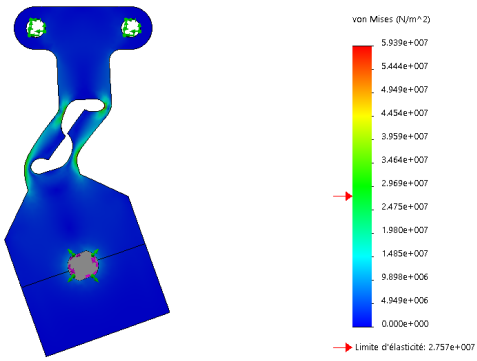
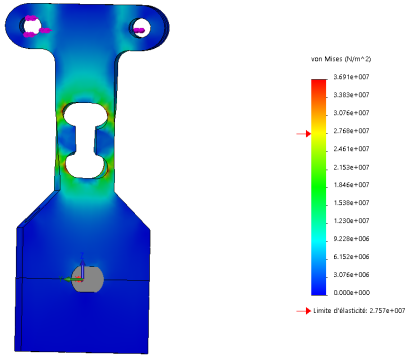
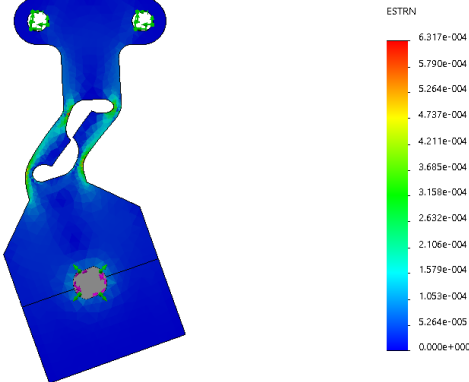
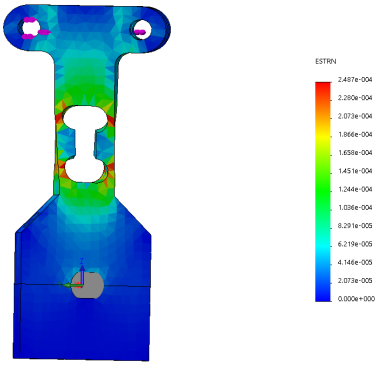
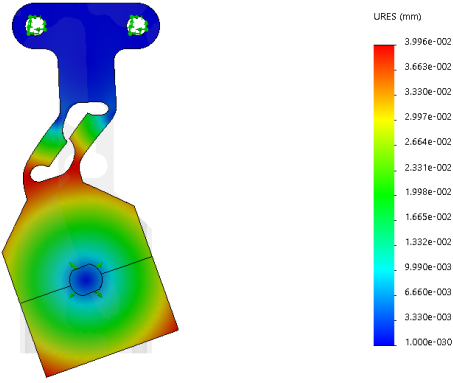
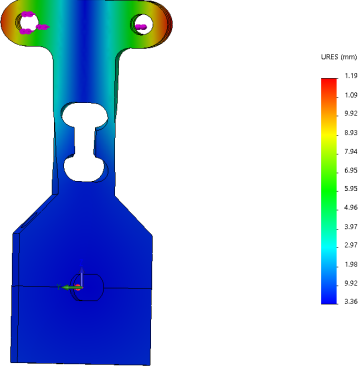
	Bending	Torsion
Constraint		
Resulting deformation		
Resulting displacement		

Table 1.2: Deformations simulations of the T-pendulum under the nominal bending torque (11Nm) and under a torsion load of 1.1Nm at its upper tip⁵

⁴This slight difference can be explained by the beam approximation made in TransCalc simulations while Solidworks' finite element solver based on a mesh generated from the real "T-pendulum" assembly

⁵ESTRN represents the equivalent 3D deformation of the body (expressed in Strain), while URES represents its equivalent displacement.

1.2.3 Signal processing

The gauges are assembled in a full Wheatstone bridge configuration (see Appendix A.2), which allows a high sensitivity of the assembly to bending strain while fully decoupling it from axial strains of the sensor and compensating for strain gauges thermal sensitivity. In this configuration, the resulting strain can be deduced from the delta of resistance unbalancing the Wheatstone bridge.

This resistance variation is due to the section decrease of a strained bar which, in turn, leads to an increase of the electrical resistance of this same bar.

Mathematically, this resistance variation can be linked to the material strain, using elastic deformation law ($\sigma = \varepsilon E$), Gauge Factor's definition ($G_F = \frac{\Delta R}{R_G \frac{1}{\varepsilon}}$) and stress-load relation ($\sigma = \frac{F}{A}$),

The resistance variation (ΔR) of a deformed strain gauge can be directly re-expressed as follows:

$$\Delta R = \frac{G_F R_G}{AE} F$$

which links the relative variation of the gauge's resistance (ΔR) to an applied external load (F) on a cross section (A), for an elastically deformed material, containing gauge with an unstained resistance defined as R_G .

The voltage between the two bridge midpoints (also referred as the bridge unbalancing) goes through an instrumentation amplifier in order to impose the output voltage range for the sensor.

The generic transfer function of the instrumentation amplifiers is then used to impose the internal resistors' values of the amplifier and therefore the amplification gain.

$$\frac{V_{out}}{V_{Wheatstone}} \left(1 + 2 \frac{R_1}{R_{gain}} \right) \frac{R_3}{R_2}$$

In this case, the sensor was designed to generate a signal of $\pm 10V$ for a full-scale measurement of ± 11 Nm, therefore requiring an amplification gain of 500, which was achieved using the following resistor's value.

- $R_1 = R_2 = R_3 = 50k\Omega$
- $R_{gain} = 100 \Omega$

1.2.4 Calibration

The sensor was then characterized and calibrated by applying multiple times different bending loads, from 0 to 11 kg⁶ (by steps of 2kg, up to 10kg, and 11kg) on the experimental set-up shown in Figure 1.9. Those measurements allowed to determine the main characteristics of the sensors (gain, linearity and repeatability).

This calibration procedure allows to identify the following sensor characteristics

- sensitivity : 909 [mV/Nm]
- measurement range : 22 [Nm]
- non-linearity : 0.5%

⁶which corresponds to a bending torque of 0 to 10.8Nm

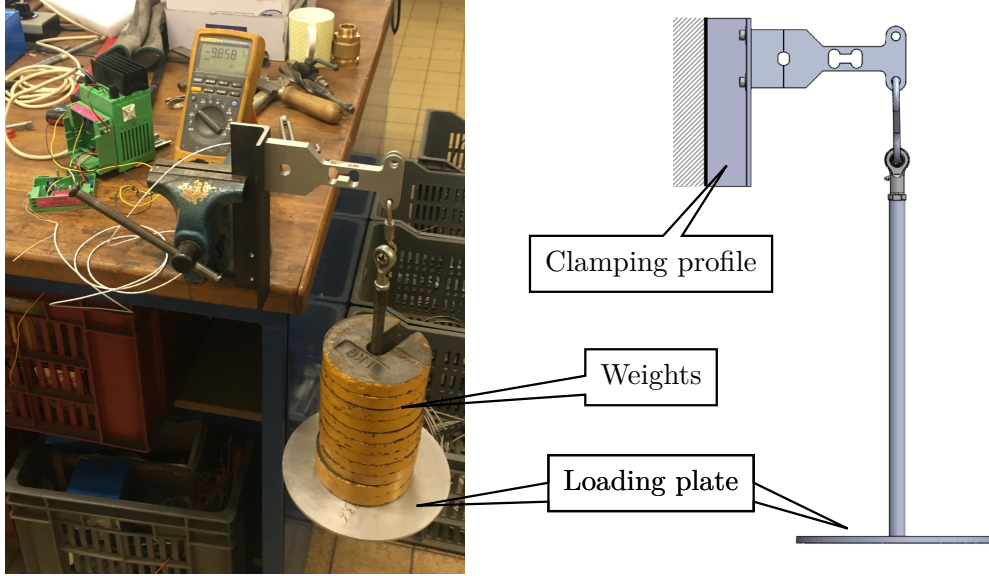


Figure 1.9: Calibration set-up of the torque sensor

The transfer function of the designed sensor is shown in Figure 1.10.

Remark : The torque reference of the sensor corresponds to a positive torque in the clockwise direction around the roll axis of the vehicle, viewed from the driver position.

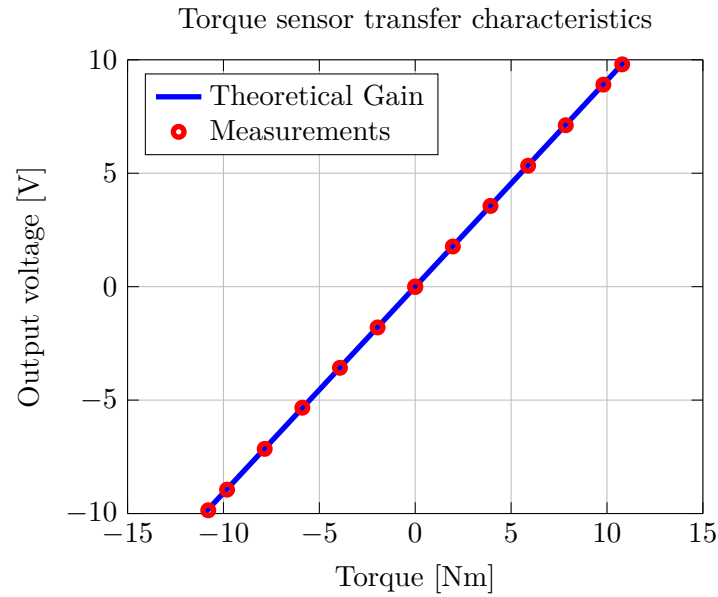


Figure 1.10: Experimental characterization of the torque sensor

1.3 Tilt sensor

1.3.1 Background - Justification of the choice

Given the fact that the motor's encoders can only measure the relative displacement of the motor starting from its initialization (corresponding to a random motor's position), it was necessary

to find a way to obtain this absolute information (i.e. the real tilting of the NTV) in order to control the vehicle.

Different types of sensors enable getting an absolute measurement :

Magnetic sensor Measures the relative angle between this moving sensor and a permanent magnet positioned in front of it. This kind of sensors is relatively cheap (30-60€ for the whole assembly) but their positioning on the tilting connection shaft is delicate.

Absolute encoder Those kind of sensors measure directly the positioning of their output shaft through different technologies most often based on optics or magnetics. They are the most accurate ones, with a 12 to 16-bit resolution and an accuracy of 0.004 to 0.002°. But their significant price (more than 300€ for the present application) and the way to integrate them in the NTV make them a less attractive solution.

Calibration The motors are firstly actuated up to a limit position⁷ then, motors displacement measurement are reset to zero, allowing to consider this reset measurement as an absolute positioning information, directly related to this homing position. The calibration mechanism requests time to be integrated in the assembly (due to the restricted access), as well as a start-up procedure. In addition to that, a re-calibration is systematically needed in case of problem. No hot re-start is thus possible.

Potentiometer This device allows to provide a voltage information related to its angular position. It is cheap but not highly accurate (less than an absolute encoder) which is actually not critical for the present application. Its accuracy (being around 0.5°) is sufficient for the current need.

Given the above and taking into account its easy mechanical implementation, its price and robustness, the choice of the potentiometer is the optimal choice for the present application.

1.3.2 Mechanical design

This sensor is designed to provide an absolute value of the tilt angle of the frame of the demonstrator (further referred as the tilt or inclination angle). The need for such a measurement is due to the fact that the actuator's encoders of the demonstrator only provide relative angular information about those motors. This means that those measurements depend on the start-up position of the actuator (which is not known without an absolute sensor as the designed one), making this sensor useful for the high-level control of the vehicle.

This position is thus measured directly on the tilting actuator, by means of a gear ratio, transmitting this motion to a potentiometer (using a multiplication ratio of 4 to maximize its stroke with respect to the inclination range of motion of the frame which, in turn, maximizes the sensor's sensitivity). The angular information can then be directly derived from the output voltage of this variable resistor, which is directly linked to its position.

1.3.3 Signal processing

The sensor is powered by a 5V voltage supply acting as tilting-dependant voltage divider. On the other hand, the potentiometer has a motion amplitude of 300°, allowing it to track a maximum tilting variation of 75°.

This information allows to identify the following sensors characteristics :

⁷determined by the triggering of a limit switch or simply by the detection of a mechanical end stop

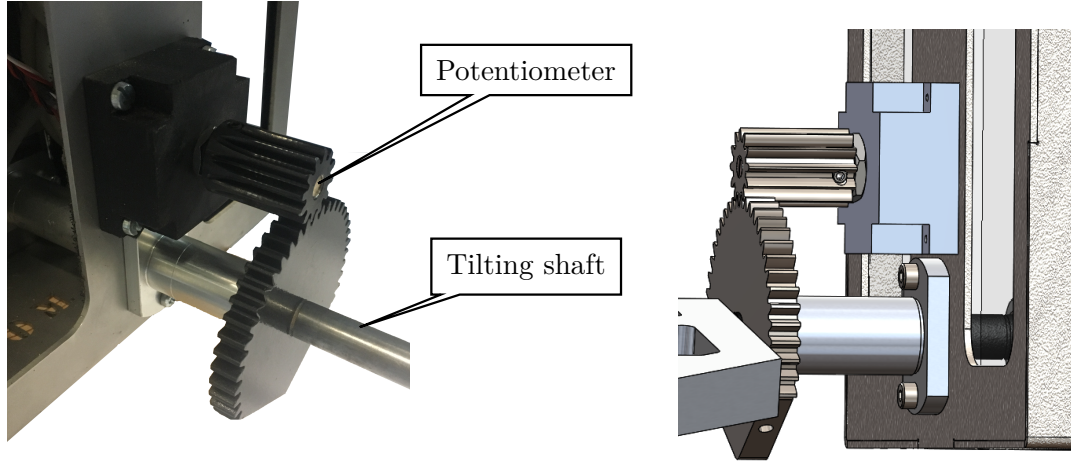


Figure 1.11: Tilt sensor implementation (left) and CAD design (right)

- sensitivity : 67 [mV/°]
- measurement range : 75°
- non-linearity : 2%

1.4 Inertial unit and magnetometer

1.4.1 Background

In order to get extra information about the vehicle's state, orientation and acceleration, it was decided to use the Pixhawk Px4⁸ auto-pilot [1] and all its embedded sensors. The Px4 was bought last year for control purpose but led to troubles mentioned in Section 2.1.1. In the present work, it was decided to use its native computing resources in order to merge the information from its different sensors (accelerometer, gyroscope, magnetometer and pressure sensor) to deduce the absolute position and orientation of the vehicle.

The device and its related features, described in detail below, are implemented and fully operational. It computes the vehicle's state by merging the following information coming from the Px4 :

- accelerometer: position⁹
- gyroscope : orientation
- magnetometer : heading

As preliminary, it is important to note that the reference frames of the vehicle and the Pixhawk Px4 are different. Both frames are given in Figure1.12 and Figure1.13.

Those data are then used for the position and orientation measurement, by integrating twice the acceleration of the vehicle in the Px4 reference frame (see Figure 1.13), to compute the vehicle displacement in this frame, with respect to its initial position. Afterwards, the gyroscope computes orientation information about the vehicle in the Px4 reference frame (pitch, roll, yaw). This orientation information is then refined, using a magnetometer to get heading information,

⁸a flight control dedicated hardware board, containing various sensors for positioning and having computational capabilities

⁹by doubly integrating acceleration measurements, leading to quadratic error

based on magnetic field measurements. Those measurements are next merged by the Px4, to compute data about the NTV's odometry (position, orientation, velocity and covariance¹⁰) of those different information.

1.4.2 Signal processing

The resulting information is gathered by integrating measured data coming from all those sensors. The main drawback of such a kind of positioning method is the fact that, depending whether those data are only integrated once (linear integral error, e.g. gyroscope) or twice (quadratic integral error, e.g. accelerometer), it induces respectively a linear or quadratic integral error, leading to increasing sensors inaccuracy with the time.

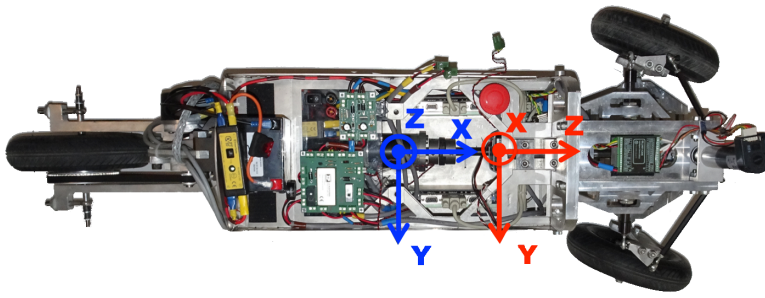


Figure 1.12: Position and orientation of the different frames : Vehicle's frame (red) and Pixhawk Px4's frame (blue)

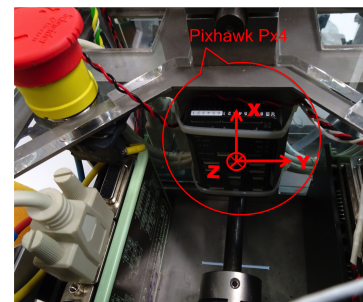


Figure 1.13: Pixhawk Px4's frame positioning

It would therefore be suitable to combine these data with other position information. For this reason, a non-linear filter (called extended Kalman filter) was implemented on the Raspberry Pi [25] allowing to use odometry and to fuse it with the Px4 pre-computed information. The Px4 measurements are thus initially transmitted to the onboard computer through an UART connection (see Section 2.1.4), using the ROS (Robot Operating System) node "mavros"¹¹, allowing it to get so-called "odometry" information (pose¹², speed, acceleration and covariance of the different sensors of the Px4). These data are then directly fed to the "robot-pose-ekf" ROS node, which simply uses odometry information from these different sensors (correspondingly the value from the Px4 and the real speed information from the actuated joints of the robot). Finally, those merged information coordinates are transferred from the Px4 reference frame to the vehicle reference frame (see Figure 1.12 and Figure 1.13).

The most appropriate mean to tackle this integral error propagation would be the addition of an absolute position information (for which the measurement accuracy does not vary with time) in the Kalman filter. It was experimented that the use of a theodolite can provide such information (see Section 5.4). Today, this system is currently used for validation purposes, independently from the vehicle's sensors. Nevertheless, this theodolite could be programmed to communicate with the vehicle and therefore to allow real time positioning of this one [18].

Remark : in addition to those sensors, the Px4 is also equipped with a thermometer. This temperature information is currently not used but could be taken into account to re-calibrate the

¹⁰which is directly linked to the covariance of the different sensors and to the merging filter that has been used by the Px4 to compute this odometry

¹¹This mavros link allows to get directly a pre-processed information about the vehicles state (position, orientation, speed and acceleration).

¹²Pose represents both position and orientation information.

torque sensor, which is highly sensitive to temperature variations, due to strain gauge thermal distortion of metals.

1.5 Power managment

1.5.1 Battery charging

Complementary modules (see Figure 1.14) were connected to the power electronics of the demonstrator, in order to improve the lifespan and the ease of use of the scale model and of its batteries. They allow the batteries to be charged from the electric network and send an acoustic warning in case of excessive discharge of the batteries.

This warning device was setup, to preventing one of the discharged electrical circuit (motor or control devices supply) from being suddenly cut off, which could lead to an unforeseeable behavior of the vehicle caused by the operation in degraded mode.

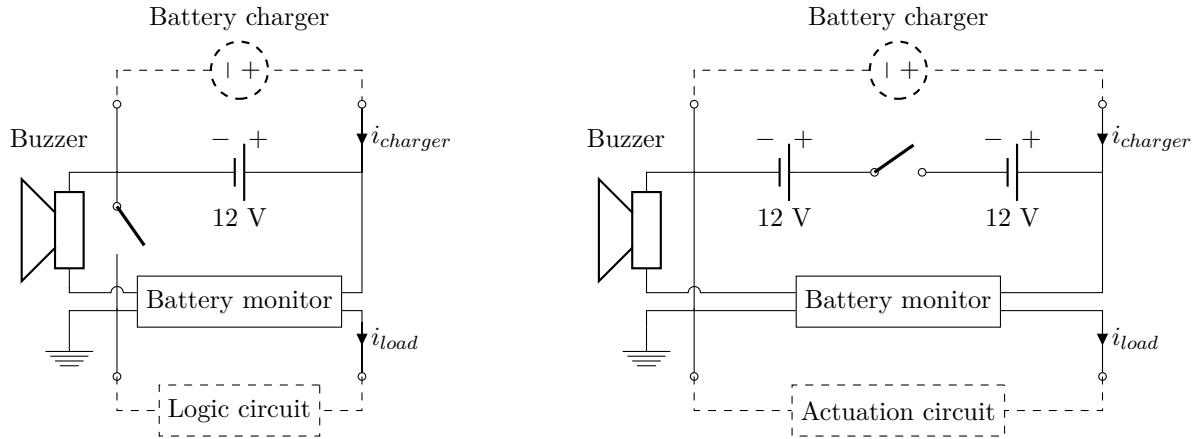


Figure 1.14: Circuit diagram of the power supply and warning system of the 12 and 24V batteries

1.5.2 Protection circuit updated dimensioning

The breaking values of some components were increased in order to allow the emergency stop to be fully operational at any time and to prevent an over-voltage break in normal operating conditions. Such breaks were caused by the under-sizing of the protection diodes compared to the effective characteristics of the circuit, i.e. 14V output voltage for a fully charged 12V battery.

In addition to that, an increase of the fuse's limit current allows the motors to get a current two times larger as previously and therefore to develop a higher power peak.

The evolution of the breaking values of all those protection circuits is summarized in Table 1.3.

	Former threshold	New threshold
Over-voltage	14V	17V
	27V	30V
Emergency protection	30V	40V
Over-current	10A	20A

Table 1.3: Update protection threshold values

Chapter 2

Data processing and communication architecture

This second chapter addresses the specificities of the data processing and communication architecture. The hardware components retained for the project are presented and the choices are justified. It concerns principally the embedded computers of the vehicle, the CAN communication bus, the driver interface and the main interface boards.

Besides, the software configuration and programming are discussed and their structure and operation are detailed : the brain configuration, the ROS middleware, the CAN/CANopen communication, the Robotran multibody software implementation for ROS and the remote control.

2.1 Hardware components

2.1.1 Nano computer - Raspberry Pi 3B

The main controller previously available for the vehicle control was an Odroid Xu4. Based on the following argumentation, it has been replaced by a Raspberry Pi nano-computer, mainly for accessibility and flexibility reasons, among which the provided wireless connection.

From a computational point of view, the nominal technical characteristics of the Raspberry Pi appear slightly less performant than those of the Odroid Xu4 : a quad-core ¹ instead of an octo-core for the Odroid ². Nevertheless, the advanced technology of the Raspberry Pi (64-bits processor using deeper branch prediction) allows for a large part to compensate this difference, due to the lower cycles per instruction of the Raspberry Pi. Especially for the present application, there is no negative impact in deploying the Raspberry Pi.

The user-friendly design of the Raspberry Pi constitutes one major advantage of the device: it contains Wi-Fi and Bluetooth antennas allowing an easy wireless connection with the moving NTV (instead of a single Ethernet connectivity for the Odroid) and normalized European connectors (instead of non standardized connectors for the Xu4).

Due to the very large users' community, the exchanges, the developments and the widely spread support on internet, the Raspberry Pi has evolved to a device very simple to operate contrary to the Odroid that remains less user-friendly.

¹each one running at 1.2GHz

²clocked at 1GHz for the half of them and at 2GHz for the four remaining cores

Remark : Since compiling on laptops processors can be up to 10 times faster than on a Raspberry Pi, the project has been designed to be cross compiled on a remote laptop before sending automatically the executable files to the Raspberry Pi. This has been done by modifying the catkin toolchain.³

2.1.2 CAN physical layer

Developing an embedded application for the demonstrator could not have been achieved without a fully operational communication protocol between the deciding authority of the vehicle (the Raspberry Pi) and all its peripherals (mainly the motor drivers). This is the reason why it was necessary to adapt the pre-existing electronic architecture of the demonstrator, which only allowed to control a single motor at a time, based on a serial connection with an external laptop.

This required to adapt the electronics in the following manner :

- adding a CAN interface for the Raspberry-Pi, enabling it to communicate using CAN protocol;
- operationalizing the existing connections of the motors' drivers to the CAN bus;
- patching the steering motor's driver to standardize the language (CAN protocol instead of TMLCAN⁴).

Raspberry Pi-CAN interface board

The transmission of CAN frames from and to the Raspberry Pi is carried-out through an existing board (Figure 2.1), using an SPI ⁵ connection and a mcp2515 microchip to translate the instructions from the Raspberry Pi into CAN messages than can be read by all the motor drivers of the demonstrator.

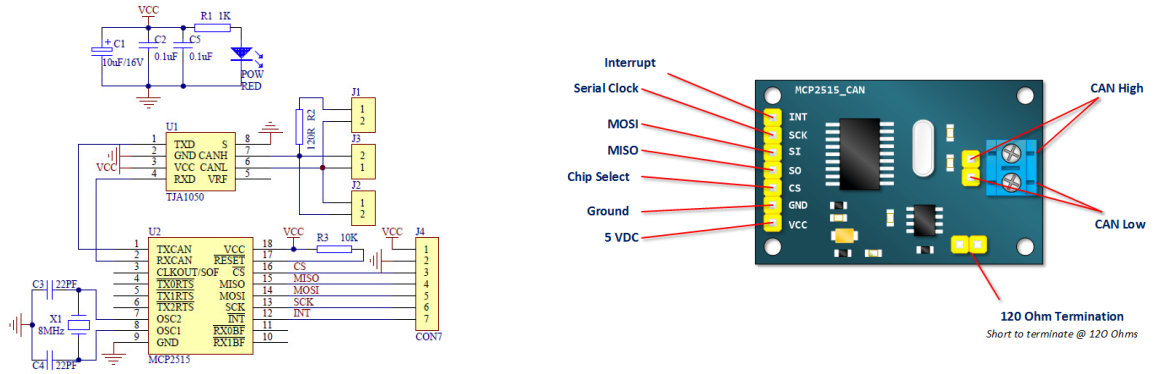


Figure 2.1: SPI-CAN interface board [5]

As it will be developed later in the report, (see Section 2.2.3), kernel additions enable the embedded computer's operating system (Linux) to handle natively the mcp2515 driver, which has lots of advantages.

³However, this does not prevent from compiling directly on the Raspberry Pi, even if that may take few minutes more than cross compiling

⁴which is a manufacturer specific communication protocol, more complex and less widespread than the CANopen applicative protocol, for which lot of open-sources libraries already exists, making it simpler to set-up and use.

⁵Serial Peripheral Interface is one the many serial communication protocols held by the Raspberry Pi, in a similar way as USB, UART, Ethernet,...

Indeed it allows this CAN translator to behave as if it was directly integrated on the micro-computer and thus to send/receive automatically CAN messages by simply providing connections information about this additional board to the operating system.

Another key strength of this pre-designed circuit was its relatively low price, compared to other pre-existing or homemade solutions.

CAN-Motor drivers interface board

The aim of this board is to establish the link between the motor controllers' connectors and the CAN translator to allow them to share properly their information with each other.

Initially some mismatches in the connection specifications of the different drivers prevented any CAN communication from working. This board was thus reassembled with different connectors while in addition to that, the previous interconnection cables were replaced by monitoring DB9 cables, ensuring a fully decoupled transmission of the essential analog signals (the CAN bus twisted pair, the ground and the power supply)

Both corrective actions enable to operationalize fully the CAN communication between the motor drivers and the nano-computer.

Remark : The previously used serial connection to the different drivers has been kept, enabling unitary debug of the drivers in case of failure. This way of working allows to have at disposal a very robust⁶ diagnostic entry whatever the external conditions.

Hardware communication protocol

In its former state, two motor controllers (the propulsive and tilting ones) were communicating using CANopen protocol while the third one (the steering driver) was using TMLCAN messages to transmit information. The main drawback of this configuration is that it requires to implement two fully independent complex high level communication layers (CANopen and TMLCAN).

Given that both protocols rely on the same physical layer (CAN⁷), it has been decided to amend the steering driver firmware so that it can interpret CANopen messages instead of TMLCAN ones. This non-referenced firmware adjustment was carried out with the assistance of the manufacturer, Technosoft, and allowed to reduce significantly the programming and communication complexity of the motor control chain.

2.1.3 Driver interface board

The purpose of the driver interface board is to allow the motor drivers to import the state information of their actuators from their respective encoders and Hall effect sensors.

An essential upgrade of the board has to be performed. Indeed, in the previous implementation, the critical signals were prevented to be delivered to the motor drivers of the vehicle due to an erroneous board design, preventing the operation of the motors.

The board and its cable interfaces were therefore deeply modified, enabling to operationalize fully the control of the tilting and the propulsive motor. (See new schematic of the board in Appendix C)

⁶compared to CAN connections for which technical requirements are more constraining, leading to a higher risk of defect

⁷which enables them to communicate on the same data buses

2.1.4 Main interface Board

In addition to the pre-existing electronic design of the vehicle (see Appendix B), a fully autonomous operation of the demonstrator could not have been achieved without undertaking the following features :

- boarded power supply of the controllers and sensors (Raspberry Pi, Pixhawk Px4, CAN interface,...);
- communication between the Raspberry Pi and all its peripherals (Px4 autopilot, CAN module and digital IO's);
- analog signals shift, conversion and import to the Raspberry Pi, enabling it to access tilt and torque measurements from the front suspension's sensors;
- connection port for the control of the power relay that manages the electrical supply of the vehicle.

To answer those needs, a new Printed Circuit Board (PCB) has been developed. The actual board is shown in Figure 2.2. Furthermore, the outline schematic of this board is depicted in Figure 2.3, while the actual schematic of the implemented board can be accessed in Appendix C.

This PCB uses a SPI connection to link physically the Raspberry Pi to its CAN module (used for CAN frames translation). Another SPI bus is used to link it with the Analog to Digital Converter (ADC) which reads the tilt and torque measurements and exports them to the Raspberry Pi. Besides, an UART connection enables to import measurements from the Px4 autopilot.

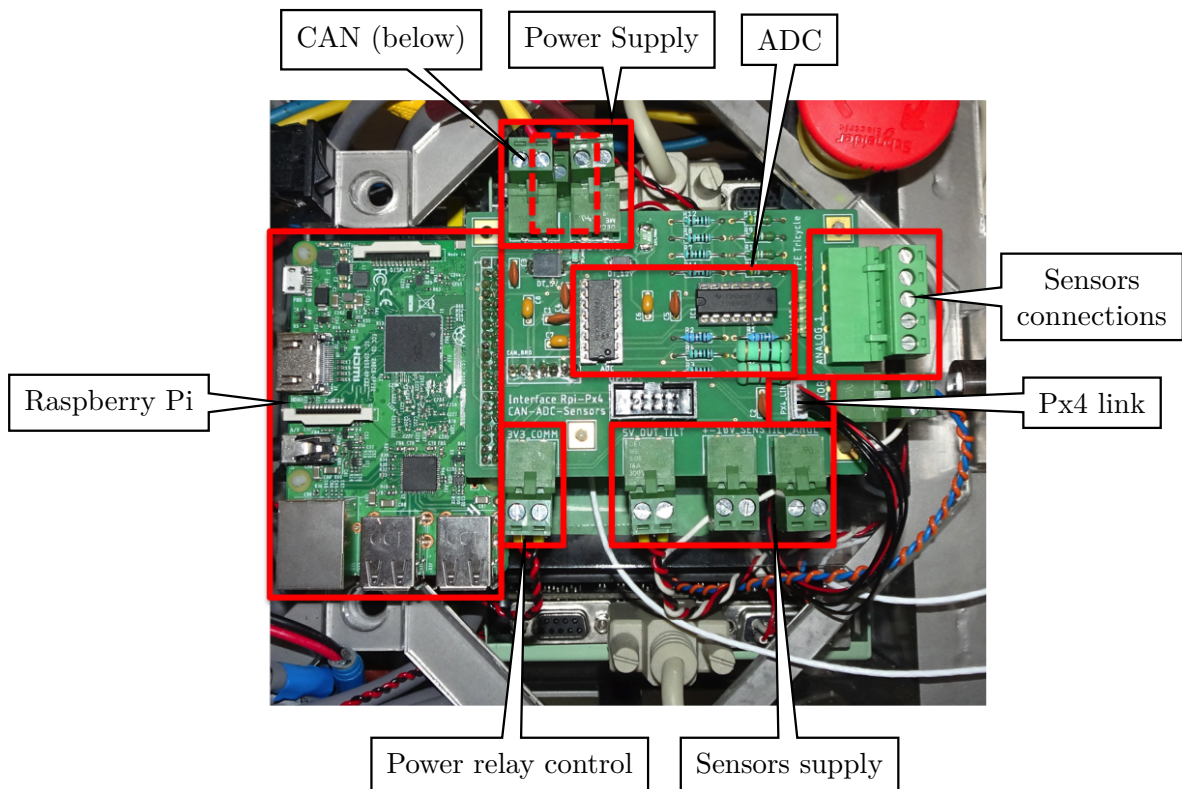


Figure 2.2: Main interface board of the scaled model

In addition to that, a level shifter circuit (Figure 2.4) is responsible for the conversion of $[-10V, 10V]$ measurements coming from the torque sensor into $[0V, 5V]$ analog signals⁸, because of the restricted measurement range of the ADC converter ($[0V, 5V]$). It can be noticed that this measurements shift induces an additional non-linearity of 0.3% and an offset of 10 mV in the input data due to the non-idealities of the amplifiers.

This PCB links also the power supply of all those devices together so that they can be fed from a unique connection.

At last, this circuit gives access to some General Purpose Inputs and Outputs (GPIO), which are used by the Raspberry Pi to control directly the power electronics of the demonstrator. Those ones leave also room for plugging potential future hardwares to the micro-computer.

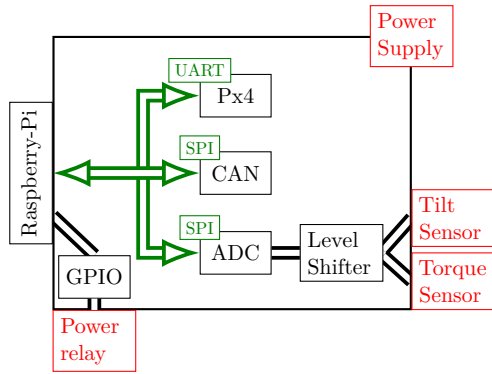


Figure 2.3: Outlines of the main interface board

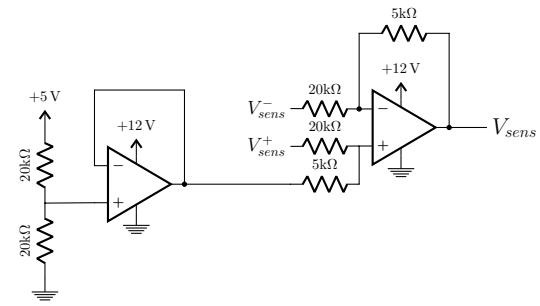


Figure 2.4: Level shifter circuit

2.1.5 Remote control

The main purpose of this communication is to establish a link between the vehicle and a remote user in order to allow the vehicle control and optionally to send him real time data about the NTV behaviour.

Three options for establishing a remote control based on radio waves were identified :

Wi-Fi antenna and connection hosting (see Section 2.2.6), enabling any device connected to this network to control the vehicle. The solution is modular as it allows to change the antenna and therefore the emission properties (antenna's gain, emission power and therefore transmission range).

3DR telemetry module of Pixhawk Px4 (see also Section 1.4) to set-up a remote communication with an external computer equipped with the same antenna.

Radio control transmitter, as used in modelism, has been considered during the previous work. It has the same range performance but less modularity than the Wi-Fi solution.

The pros and cons of these different options are presented in Table 2.1.

⁸with the following transfer function $V_{RPI} = 2.5 + \frac{V_{sensor}}{4}$

	Advantages	Drawbacks
Wi-Fi antenna	- Modularity - emission range (configurable antenna's gain,...) - controls edition (programmable GUI) - Easy connection (USB, coaxial,...) - Allows real time data display and processing	- Antenna's fragility - Requires implementation time (interface and network programming)
3DR telemetry module	- Easy connection on the Px4 - Data transmission to computers	- Limited communication range (4m)⁹ - Limited amount of achievable controls (due to Px4 pre-configuration)
Radio control	- Similar performance to Wi-Fi - User-friendly interface	- Complex receiver implementation (for the Raspberry Pi) - Limited modularity - Does not allow for applying telemetry

Table 2.1: Pros and cons of remote control solutions

Given all that, it was evaluated that the Wi-Fi solution was the most efficient and flexible tool to allow the remote communication between the vehicle and a given user, as it enables control as well as telemetry transmission, possibly at long range, see calculation below.

Two Wi-Fi options can be used for the remote access and control of the Raspberry Pi:

- a short range Wi-Fi hotspot generation using the Raspberry's native antenna, allowing reliable and fast wireless access (no configuration needed), e.g. for vehicle reprogramming and static testing;
- a long range Wi-Fi hotspot generation using an external Wi-Fi range extender (AWUS036 Wi-Fi dongle), enabling to keep contact on a wide area with the moving vehicle.

These options are characterized by their communication delay (i.e. the ping), working range and bandwidth, see Table 2.2, making them suitable for various applications.

	Ping [ms]	Range ¹⁰ [m]	Bandpass [Mbps]
Native WiFi	3	35	5
AWUS036 AC	7	»200	20

Table 2.2: Remote communication experimental characterization

Remark : Aside from the Wi-Fi solution, the Ethernet cable remains available to connect the Raspberry Pi for configurations requiring high speed communication data transfer, internet

⁹The feed-back from other UCL projects confirmed this limited range, the ease of use of this system for measurements transmission and the restricted number of controls (mainly related to flying drones systems) which is not suitable for an application like the current one.

¹⁰This range is strongly linked to the sensitivity of the receiving antenna, and therefore to the remote computer performance.

access¹¹, ...

The wireless system that has been retained for our application is the Alfa AWUS036 long range Wi-Fi dongle, having an emission power of 32dBm, mounted with with additional 12dBi antennas¹², providing a realistic transmission range of more than 200m.

2.2 Software programming

2.2.1 Brain configuration

The brain configuration is based on three key points, shortly listed below. In addition to them, several software applications were implemented according to the needs.

- The operating system is Linux as its flexibility allows for easy in depth management of the computer.
- ROS Lunar, as a robotics middleware specialized in embedded applications, control and robotics is chosen for the project.
- The Robotran software for embedded multibody simulation is applied for the current NTV model. It can manage both real time and predictive simulations.

2.2.2 ROS

ROS is a middleware dedicated to robotics applications and embedded control. It allows to emulate the behaviour of multi-trade systems through the use of real-time mechanisms. Its purpose is to replace one unique large program by multiple small modules (called nodes) communicating together through local socket connections¹³. This allows for modularity and flexibility, increasing therefore the robustness of the system by limiting the risk of Single Point of Failure (SPF). In addition to that, a large professional community uses ROS with an open-source mind sharing lot of softwares and debugging tools for mainstream applications (CANopen communication, sensors' measurements, ...).

Justification for choice

The strengths of ROS are the followings :

- thanks to its **modularity** and **flexibility**, ROS enables to perform directed *multi-tasking*, optimizing the tasks execution by the cores (in a similar way to multi-trading);
- for applications requiring a high computation power, it allows to easily perform the different tasks on *multiple devices*, increasing drastically the available processing power. For the current application, this could be the case if complex controllers are implemented;
- finally, ROS is designed to allow nodes written in *different languages* to easily communicate amongst themselves.

¹¹In addition to that, a NAT allows to share the internet connection to all the devices connected to the Raspberry Pi Wi-Fi network.

¹²Those characteristics can be compared to the ones of the Raspberry Pi : an antenna's gain of 1dBi and a maximal emission power of 20 dBm, allowing to understand the transmission range difference of both antennas.

¹³This data transportation protocol connects physically the callers together, very similarly to old-fashioned (phone) call. It has the advantage of providing instantaneous communication between all those nodes, once the communication has been established

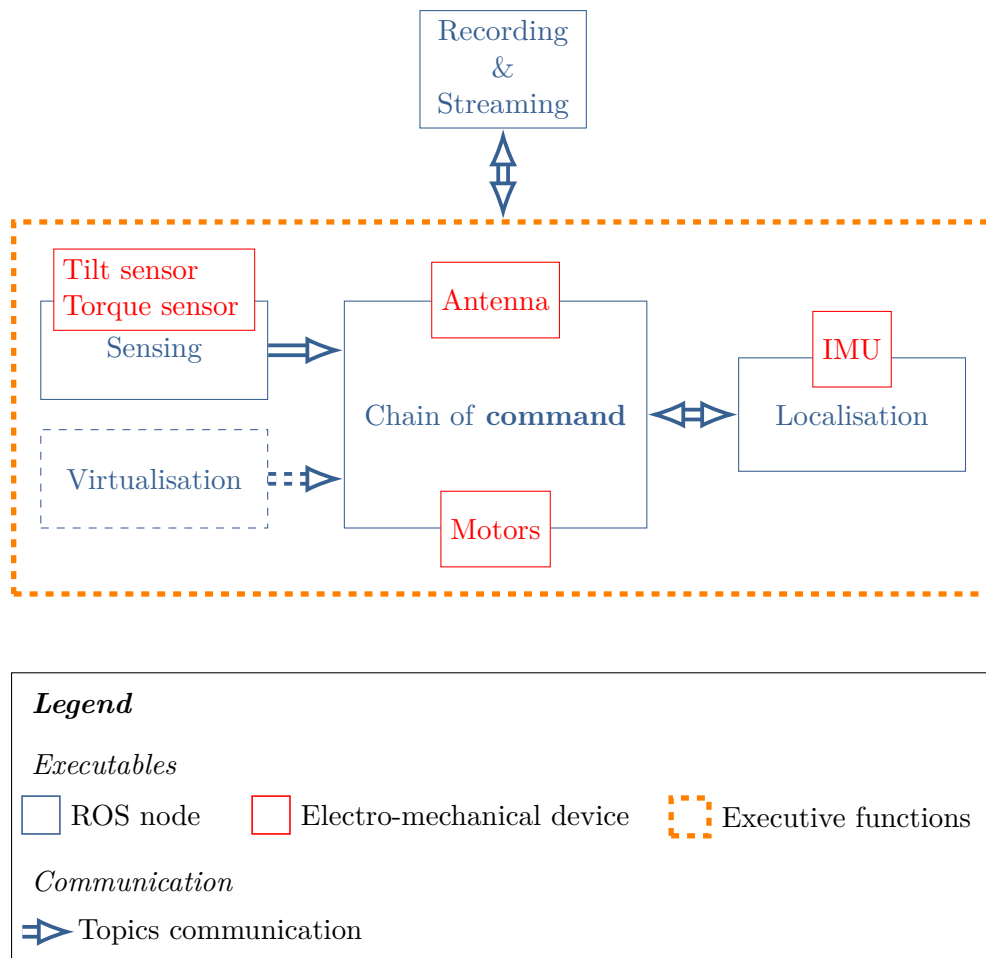


Figure 2.5: Simplified software architecture

ROS users are part of an **extended professional community** developing lot of open-sources applications (ROS packages). Large institutes as Google, MIT or Stanford University develop those major packages accessible to ROS users for mainstream applications (data processing of sensors, data filtering, communication, ...). Related documentation and training information related to these applications are available within the community.

Aside from these advantages, it is important to keep in mind that ROS requires specific training to manage its features (topic communication, simultaneous initialization of several nodes,...) and to remain watchful when importing the packages.

Finally, nodes and topics initialization lead to a slight start-up overhead of a few milliseconds, compared to monolithic programs. Nevertheless, given the flexibility brought by the system, those disadvantages become negligible. Figure 2.6 shows ROS communication architecture.

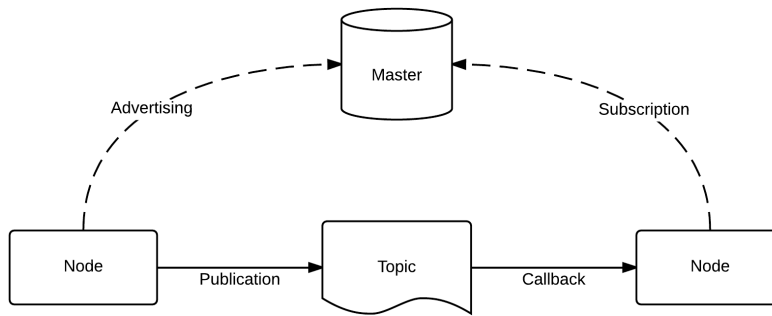


Figure 2.6: ROS communication architecture [15]

In the framework of the present study, the nodes of interest are the followings :

Remark : For the sake of clarity, all the ROS nodes developed for the present study, are underlined in the following section.

VEHICLE CONTROL

[NTV-CANopen-master](#) Low level motor control.

Manages the communication between the Raspberry Pi and all the drivers to control every actuator and import their position, speed and acceleration and send those measurements to the vehicle's main controller (NTV-controller node) **ros-kit-canopen**[25]. It also handles the vehicle's power supply by means of the Raspberry Pi's GPIO and Wiring-Pi protocol.

[NTV-controller](#) Vehicle's main controller for DTC and STC modes.

Computes motor target of the different drivers and send them to the CANopen master node. Those targets depend on the information measured by the vehicle's sensors and the user controls (Speed, curvature radius and controller type). The operation of those different controllers is detailed in Section 4.2.

LOCALIZATION

[Robotran-runner](#) The interface allows to execute Robotran as a ROS node and therefore to benefit from the advantages of ROS in Robotran's executable (topics communication,

modularity, pseudo multi-threading,...) through the use of dependency injection and linker between C and C++ compiling¹⁴.

Odometry It allows to re-expression the position of the vehicle, in order to feed it to the extended kalamn filtre

Robot-pose-ekf Extended Kalman Filter implementation for ROS.

Remark : an theoretical explanation about the Kalman filter and theoritical simulations of its behavior on the vehicle can be found in Appendix D.3.

SENSING

Mavros-node [25] ROS node for PX4 measurements processing and information transfer between the Raspberry and the Pixhawk Px4 through Mavlink protocol and serial connection.

ADC-sensing ADC sensor measurements pooling (Tilt sensor and Torque sensor).

REMOTE CONTROL

Rosbridge-Webserver [25] Websocket interface for ROS, enabling remote control of ROS through the web browser of any device connected on the same network as the server (in this case, the Raspberry Pi), as explained in Subsection 2.2.6.

DATA PROCESSING

Rosbag [25] Records in real time all the ROS topics in a .bag file, allowing to work in an equivalent way to a black box saving exclusively data¹⁵ from the last 15 minutes of use, constituting therefore an efficient data-recording system, without overflowing the Raspberry Pi with numerous useless files. These files can be easily transferred to a computer, for data post-processing purpose. They can also be directly used on the on-board computer, to re-run a recorded defective behavior of the vehicle for debugging purpose.

Plot-Juggler [25] Streams in real time all the topics and allows to plot them. Those plots can also be generated from a .bag file. In addition, it enables publishing a given pattern on a specific topic, for simulation purpose.

Figure 2.5 schematizes the organization of the software architecture. A more detailed representation (showing independently each node's interactions) can be viewed in Figure 3.6.

2.2.3 CAN/CANopen communication

Linux and ROS benefit from built-in librairies allowing to communicate directly respectively through the CAN bus at low level for Linux and, to enable ROS to format high level CANopen frames in order to transmit them to the operating system's kernel (socketCAN), responsible for the transmission to the CAN bus. The main librairies are :

- can-utils CAN subsystem implementation for Linux (also called SocketCAN). This so-called “network socket” (see Figure 2.7) is a single data collection point allowing all softwares requiring a CAN communication to perform it through this generic interface, without

¹⁴ROS multiple machine but increased delays, as explained in Table 2.2

¹⁵namely all the published ROS topics, providing therefore information about the vehicle and its software behavior (sensors measurements, controller targets and references evolution,...)

having to implement themselves communication to the CAN translation device, whatever it is, provided that the socket has been pre-configured for this task.

- ROS-kit-canopen CANopen library for ROS, communicating directly through socketCAN interface, allowing to manage automatically CANopen communication through the data bus, as soon as its network socket has been configured¹⁶. In addition to that, dictionaries have been setup to map and translate automatically the various requests into the internal language of the target device, complying with its internal memory specificities.

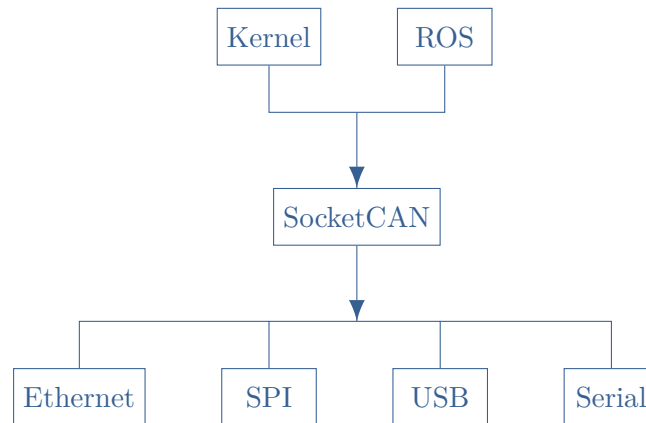


Figure 2.7: CAN Software interface of Linux kernel

2.2.4 CANopen main features

The vehicle's communication is based on a CAN physical and data link layers, enabling to transmit raw information between all connected computing devices.

On top of those layers lies the CANopen protocol, covering the network to applicative OSI layers, allowing to perform sophisticated operations such as motor controller set-up, using complex time varying references¹⁷. Figure 2.8 below shows the structure of the CANopen layers.

In addition to that, the CANopen communication tool is based on a master/slave principle, allowing a single master¹⁸ to control all other devices connected to the CAN Network¹⁹.

Such a complex high level protocol is thus relatively difficult to establish given its complexity. But it has also the advantage of providing lots of tools to handle automatically the key features of high level communication (data routing, clients management, ...).

The following key functionalities of CANopen protocol are implemented or considered for further development :

NMT - Network Managment Tool Used by the CANopen master to manage the activation status of all the CAN nodes connected to the networks. All those slave nodes can thus be remotely turned on/off while they can also be used to track regularly failures into the communication network, using heartbeat requests to detect and correct them automatically.

¹⁶This allows to connect any device to the CAN bus without extra configuration, which can be useful for debugging purposes.

¹⁷for position, speed or torque control

¹⁸the Raspberry Pi, in the framework of this project

¹⁹including the CAN-bus and all its connected devices

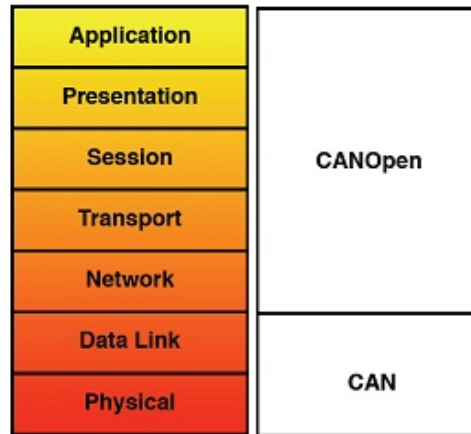


Figure 2.8: CANopen layers structure

SDO - Service Data Object This protocol enables the motors motion by allowing direct access and editing the internal memory of CANopen nodes. For example, it enables to modify the target value of a controller or to get measurements from a sensor that belongs to a CANopen node (e.g. an encoder).

EMCY - Emergency management It allows to trigger automatic actions whenever an error is flagged by one of the peripherals connected to the network. This is currently used to stop automatically the vehicle when the motors drivers report an error.

PDO - Process Data Object This protocol sends shorter messages than the SDO in an automatic way to broadcast information (motor position, control mode, current consumption,...) to the whole network. It is currently not implemented but could be of interest to limit the traffic on the CAN bus.

The advantage of using this communication protocol is the automatic network management to check the functionality of the motors at any time and enables an immediate action in case of flaw detection, as well as the automatic formatting of the data in order to translate them in operational orders to the motors. In addition to that, CANopen protocol was also selected because of the transparency of its specifications, compared to the other high level CAN based protocols (TMLCAN,...) compatible with the existing drivers.

This is why it was decided to set up a complex high level applicative protocol (CANopen) instead of simply implementing and using the physical communication protocol (CAN).

2.2.5 Robotran interface for ROS

This program is currently used to locate the vehicle using odometry and to linearize the dynamic equations of the vehicle around its current state in order to derive the coefficients to be transmitted to its controllers, mainly depending on the speed of the vehicle.

It is worthwhile noticing that ROS operates its own multi-body simulator (Gazebo²⁰). This one has been replaced by Robotran for the following reasons :

- the dynamic analysis of the demonstrator has already been performed at UCL for several years, using Robotran. A working Robotran simulation was therefore available for redesign.

²⁰which can be configured in a way similar to Robotran, relying on XML formatted *.urdf* multibody description files, similar to Robotran's *.mbs* file

- Gazebo is a robotics/control oriented simulator more than a purely multibody simulator. It focuses therefore on features that are less useful for our application (sensing simulation,...), whilst it does not include other functions (like equilibrium or modal analysis), that could become handy for the future development of complex controllers.

It appears to be easier to compute the odometry with Robotran instead of using the direct dynamics of the vehicle for this purpose, given that the kinematics of the NTV will probably be used in any case to carry out more complex control technique, like Predictive Control.

The odometry computation as well as the way Robotran interfacing with ROS is schematized in Figure 2.9.

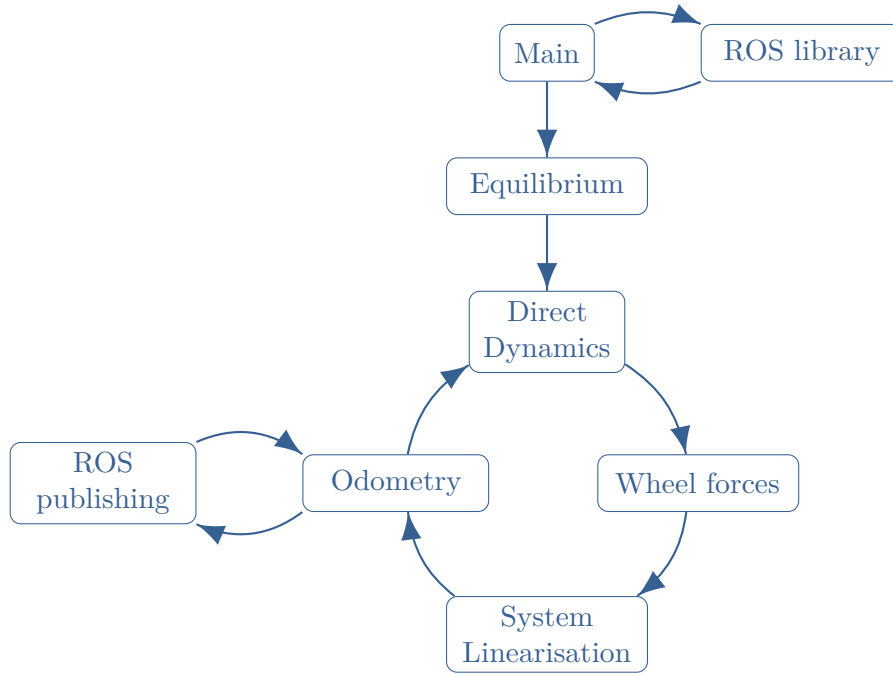


Figure 2.9: Robotran embedded implementation

Remark : The appropriate tilting angle of the vehicle for a given cornering configuration is currently pre-computed offline. To compute those parameters directly online through Robotran should allow to use a pre-computed look-up table and the related approximations due to the finite size of the table.

2.2.6 Remote control

The remote control of the vehicle is based on interactive files served from the Raspberry Pi, in such a way that the vehicle can easily be controlled by editing those files through a web browser (as simply as using a Graphical Interface), by means of any device that supports an internet connection. In addition to this control interface, this tool provides real time plot of display of requested data (e.g. the vehicle's acceleration, tilting,...). A screenshot of this interface is given in Figure 2.10.

Such a configuration allows for any device (laptop, smartphone,...) to get access to the interface without prior need for configuration due to the fact that everything is managed by the Raspberry Pi (Wi-Fi hotspot generation, files access and network management, ROS interfacing, domain name resolution,...).

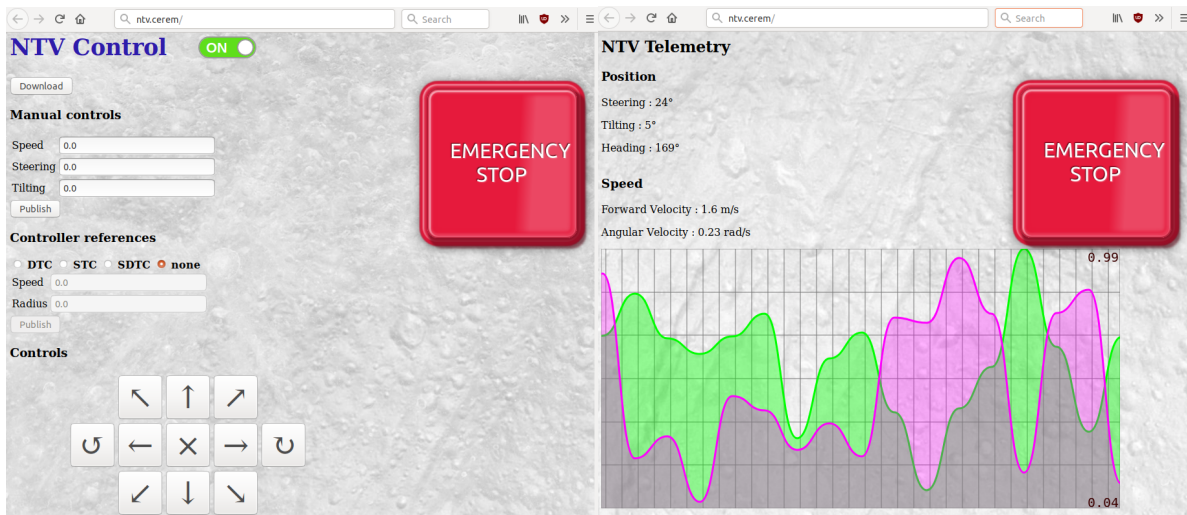


Figure 2.10: Web interface of the vehicle, allowing remote control (left) and telemetry (right)

In addition to that, a remote shell access (SSH) allows in depth re-programming of the vehicle.

Chapter 3

Updated demonstrator configuration

This chapter summarizes the update configuration of the demonstrator, for mechanics, electronics and softwares.

3.1 Mechanical configuration

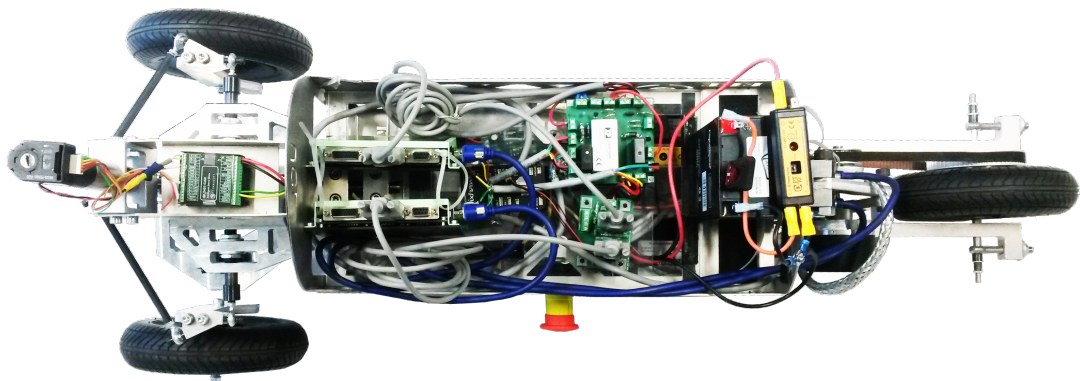


Figure 3.1: 2017 Vehicle Assembly (upper view)

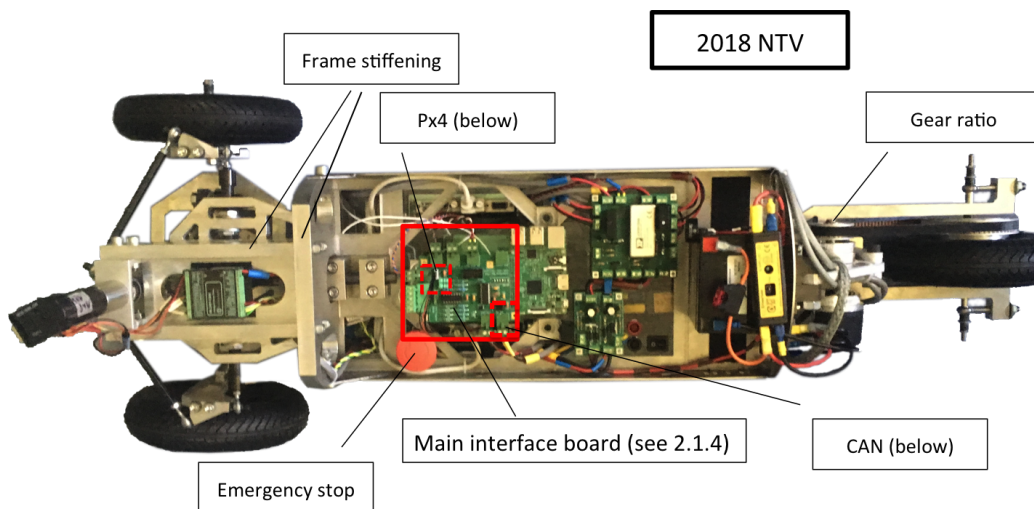


Figure 3.2: 2018 Vehicle Assembly with main updates (upper view)

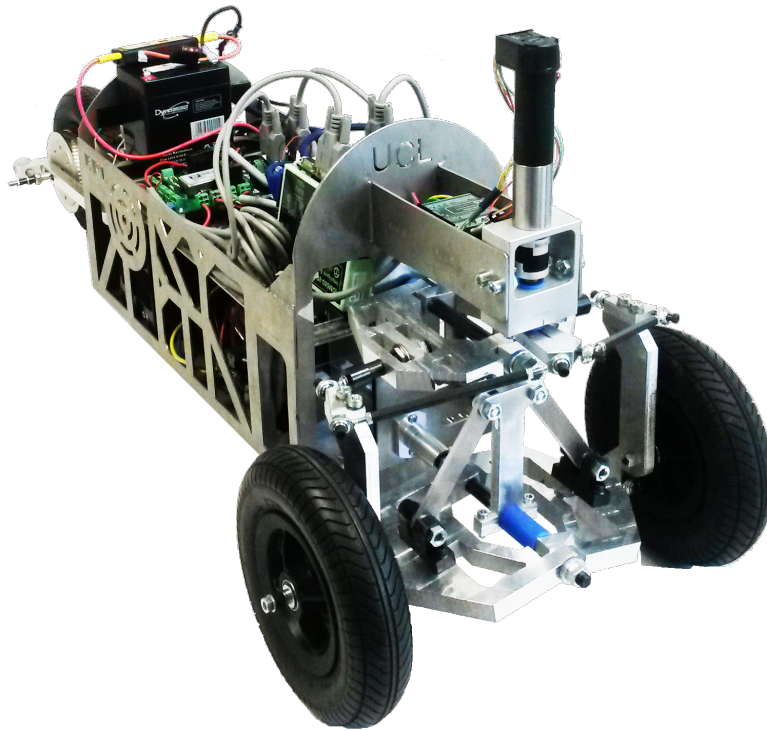


Figure 3.3: 2017 Vehicle Assembly (isometric view)

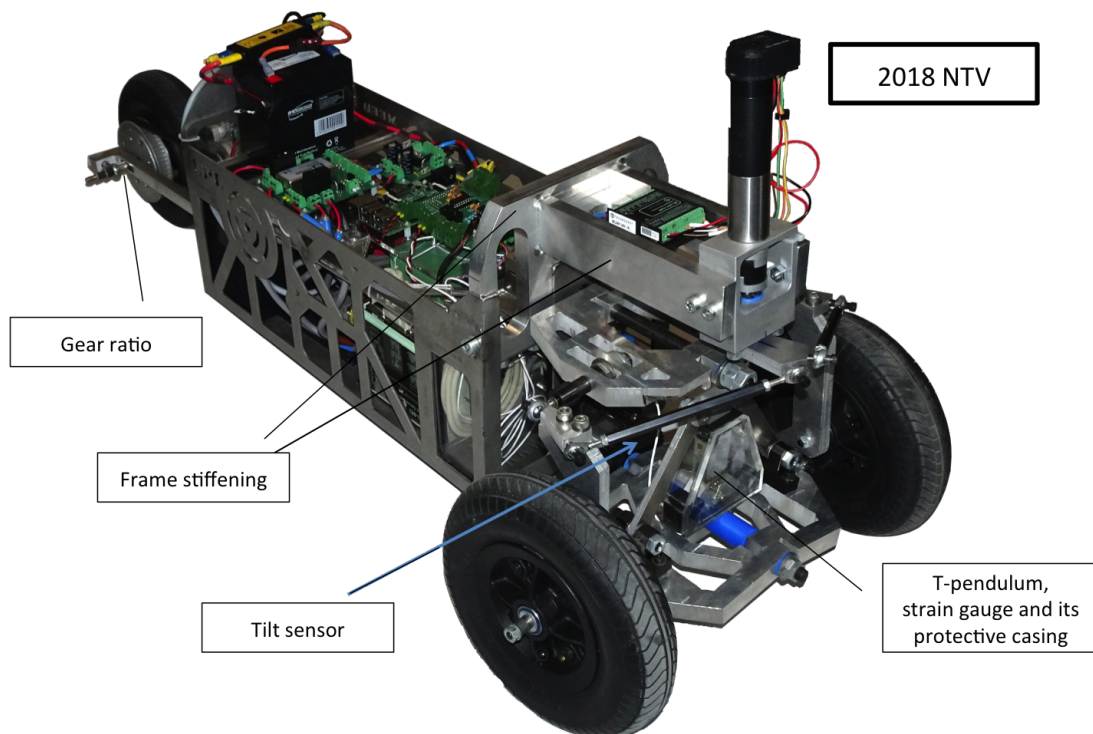


Figure 3.4: 2018 Vehicle Assembly with main updates (isometric view)

3.2 Global electric architecture

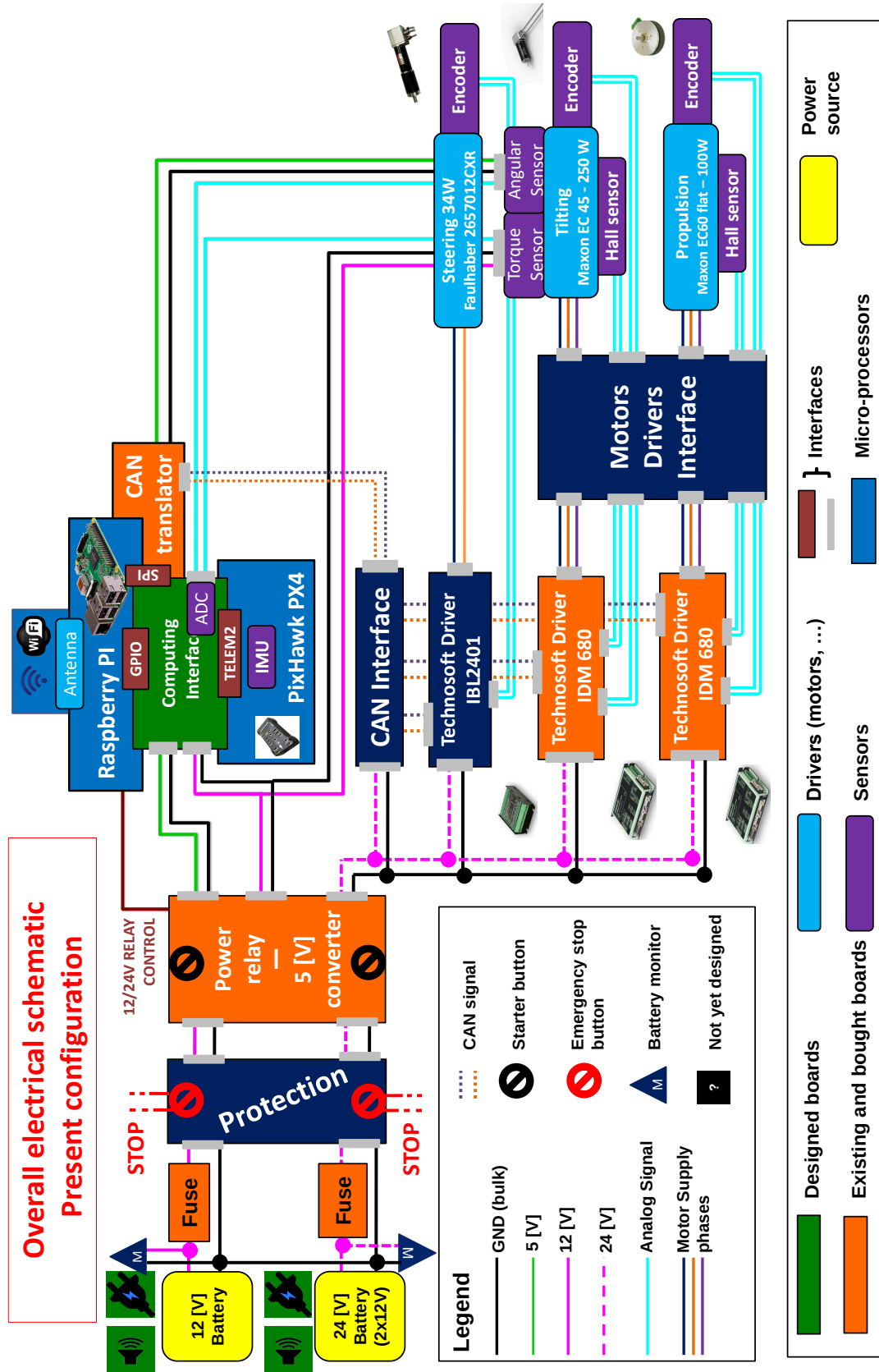
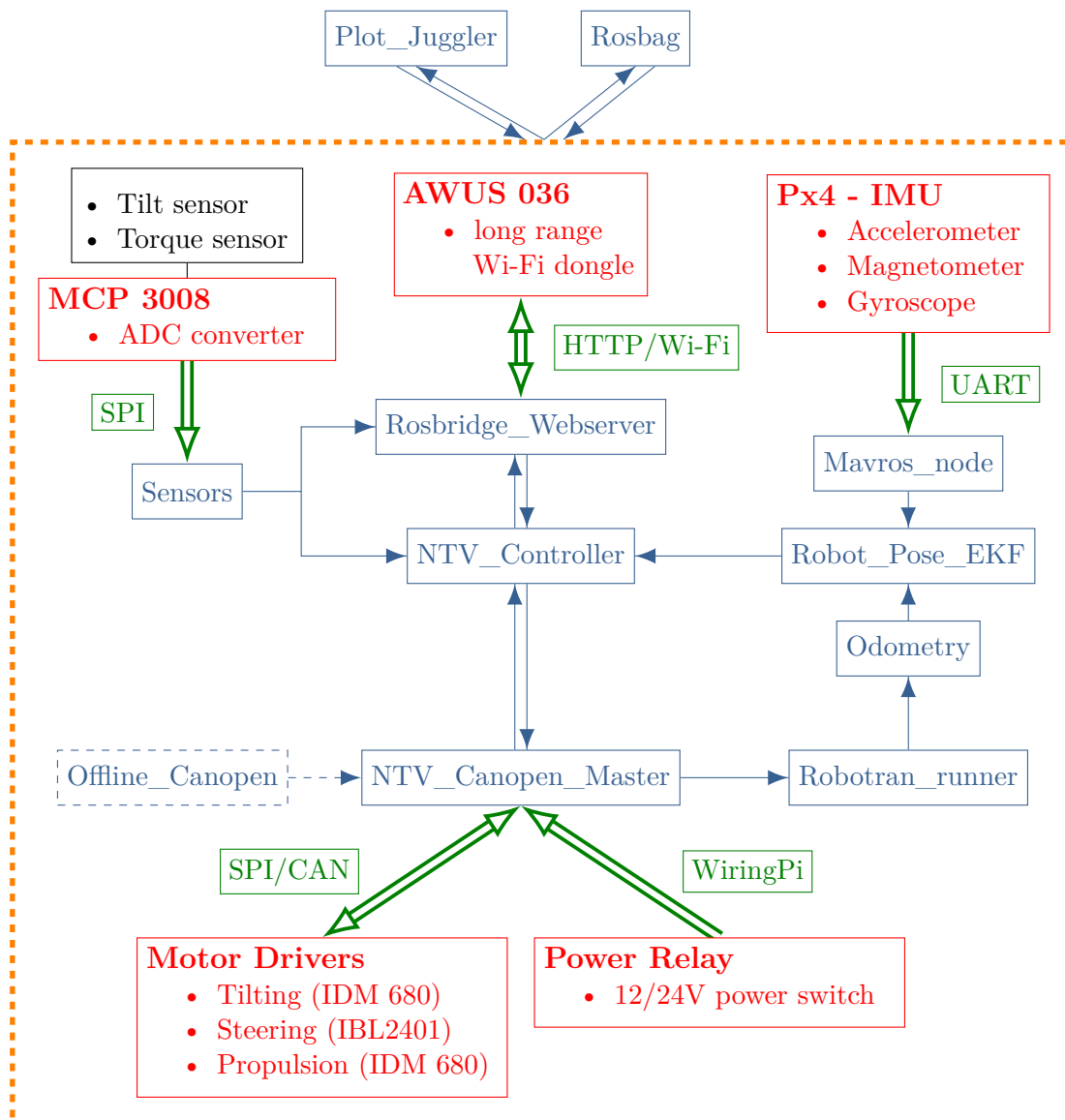


Figure 3.5: Electrical scheme of the prototype (adapted from [23]). 2017 initial electric configuration of the vehicle and updates are given in Appendix B

Type	Component	Commercial name	Specifications	Reference
Motors	Propulsion	Maxon EC 60 flat	24 [V] - 100 [W]	[12] p. 236
	Tilting	Maxon EC45	24 [V] - 250 [W]	[12] p. 195
	Steering	Faulhaber 2657	12 [V] - 34 [W]	[10]
Encoders	Propulsion	MILE 512-4096 CPT	550 tics/c	[12] p.309
	Tilting	HEDL 9140-500 CPT	550 tics/c	[12] p.331
	Steering	HEDS 5500	500 tics/c	[3]
Controllers	Central brain	Raspberry Pi 3B		[11]
	Propulsion	IDM680-8BI	640 [W] servo drive	[20]
	Tilting			
	Steering	IBL2401-CANopen	25 [W] servo drive	[19]
IMU (in Px4)	Gyroscope	ST Micro L3GD20H	16-bit	[1]
	Accelerometer/Magnetometer	ST Micro LSM303D	14-bit	
	Accelerometer/Gyroscope	Invensense MPU 6000	3-axis	
	Barometer	MEAS MS5611	barometer	
Interfaces	CANbus module	MCP2515 & TJA1050	500 [kbps]	[5]
	Wi-Fi Antenna	AWUS036 AC	1[W], 12 [dBi], 800 [Mbps]	[13]
	Battery charger and discharge alarm	PowerTector PT10	12-24 [V]	[7]
Sensors	ADC converter	MCP3008	10-bit, 8 channels	[5]
	Tilt sensor	Vishay PE30	15 [mV/°]	[9]
	Torque Sensor	UCL	± 11 [Nm], 909 [mV/Nm]	
Batteries	Power supply	Lead-Acid 5 [Ah]	3×12 [V]	[7]

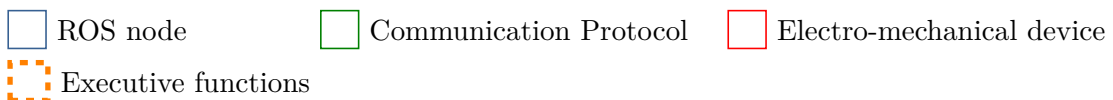
Table 3.1: Electronic components of the demonstrator, adapted from [23]

3.3 Global ROS Architecture



Legend

Executables



Communication



Figure 3.6: Global ROS architecture

Chapter 4

Vehicle control

To comply with the design condition that the NTV has to be usable by people with limited training, its automated controls become an essential part of its development.

This automation is carried out at two different levels.

The first one, called "low-level" control, is used as an independent regulation of each actuator of the vehicle, ensuring their functional reference tracking. This control is performed independently by each motor's driver and can involve a position, speed or torque reference, depending on the application.

The second one, generally designated as "high-level" control is, in contrast, designed to address more complex operation situations. It is responsible for the translation of inexperienced users instructions (the desired curvature radius and longitudinal speed of the vehicle) into effective references for the different low-level controllers, while taking into account the physical constraints of the vehicle as well as its dynamic behaviour.

4.1 Low-level control

Depending on the speed of the vehicle and its expected behaviour, the type of control requested for the different motors will vary, and can be either a control of torque or speed.

The different types of controls applied to the different motors are described below.

4.1.1 Propulsion motor

The propulsion motor is controlled, in any case, in speed, by a current regulation loop included in a speed regulation loop. This motor acts thus as the vehicle's speed regulator.

The schematics of the speed control carried out on the propulsion motor is given in Figure 4.1.

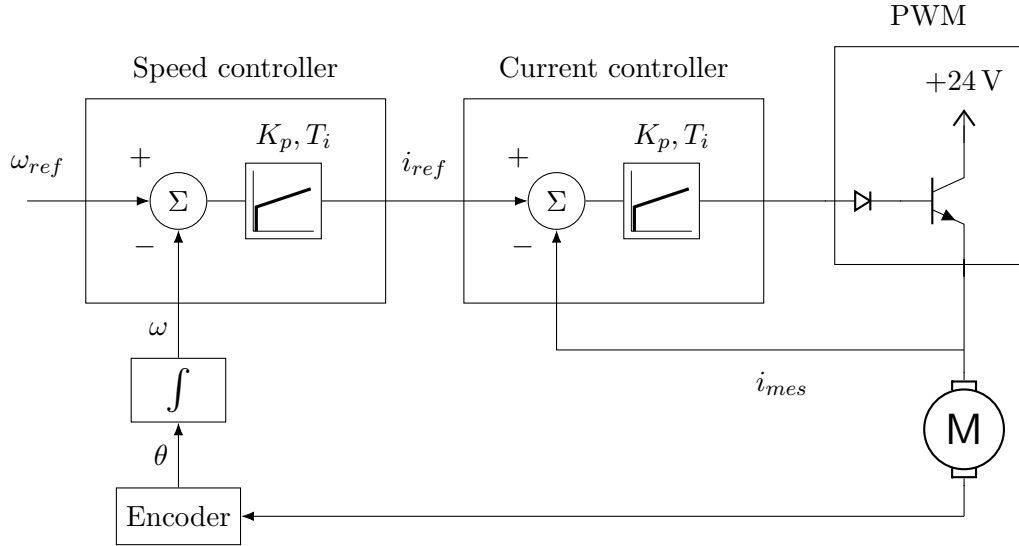


Figure 4.1: Schematic representation of the speed control of the propulsion motor M

4.1.2 Steering motor

In the low-level control, the steering motor is controlled in position or in torque, the Raspberry Pi allowing to switch from one configuration to the other.

Both controls are carried out as follows :

- **Position control** - In DTC control mode, the steering motor is controlled in position, by a current regulation loop included in a speed regulation loop itself included in a position control loop, see Figure 4.2.
- **Torque control** - In STC control mode, the steering motor is controlled in torque, using a single current control loop, see Figure 4.3.

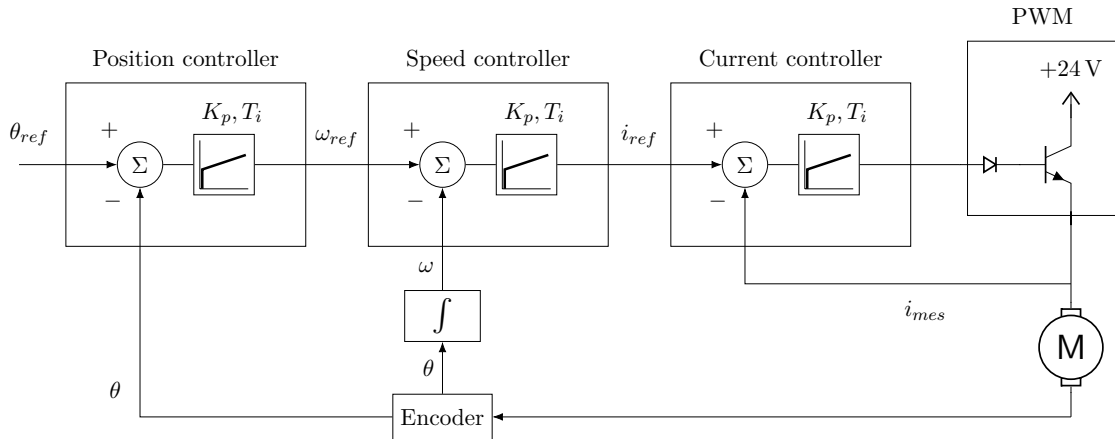


Figure 4.2: Schematics of the position control of the steering motor M in DTC control mode

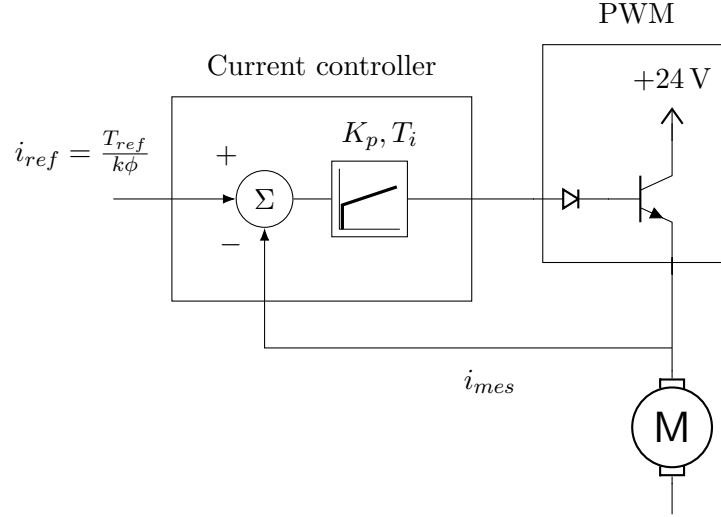


Figure 4.3: Schematics of the torque control of the steering motor M in STC control mode

4.1.3 Tilting motor

The control mode of the tilting motor is highly dependent on the vehicle speed, through the high level control used on the vehicle (DTC or STC). Indeed, at low vehicle speed (for DTC control), this motor is controlled using position regulation to impose directly the tilting of the vehicle. Whereas at high vehicle speed, the motor control switches to torque regulation.

Both motor control modes are presented hereafter :

- **Position control** - At low speed, for DTC control, the tilting motor is controlled in position, by a current regulation loop included in a position regulation loop, see Figure 4.4.

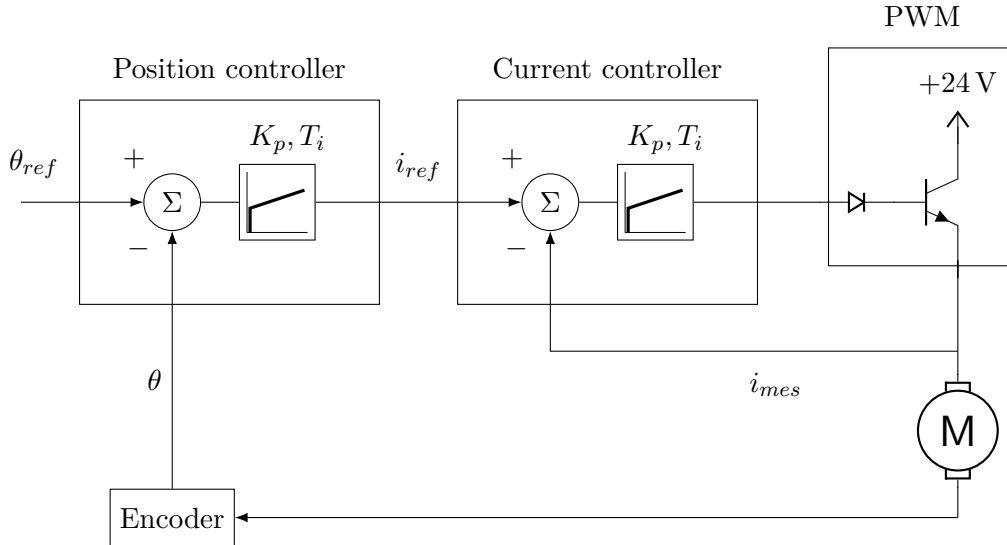


Figure 4.4: Schematic representation of the position control of the tilting motor M

- **Torque control** - At high speed, the tilting motor is controlled in torque, by a current regulation loop included in a torque regulation loop, see Figure 4.5.

The implementation of the torque controller is made on the NTV but needs to be fine-tuned to become effective, due to high level of vibrations induced in the system. Therefore, at high

speed, the motor is uncoupled from the NTV by removing the transmission link of the Oldham coupling.

This torque control is carried out based on the torque measurements from the developed sensor; this allows to control the effective torque induced by the motor on the frame instead of the calculation based on the motor consumption that does not take into account the internal friction due to the motor and its gearbox. This control is therefore not only based on a simple current regulation loop¹ but includes the real torque.

The torque controller allows instantaneous decoupling of the tilting actuator and therefore makes possible the automatic switch between the different control modes. This zero torque control would also enables to carry-out an "hybrid" SDTC control, by controlling at the same time the steering of the vehicle and the tilting torque applied on the vehicle's frame, to reach a given inclination angle.

Instead of a simple current loop for which the torque from the motor is approximated as $T_{util} = k\phi i_a$, neglecting therefore all the transmission internal losses and the electrical problems due to the current form (PWM)², this torque controller considers an additional control loop relying on a torque reference and a feedback information from the torque sensor, enslaving the first one.

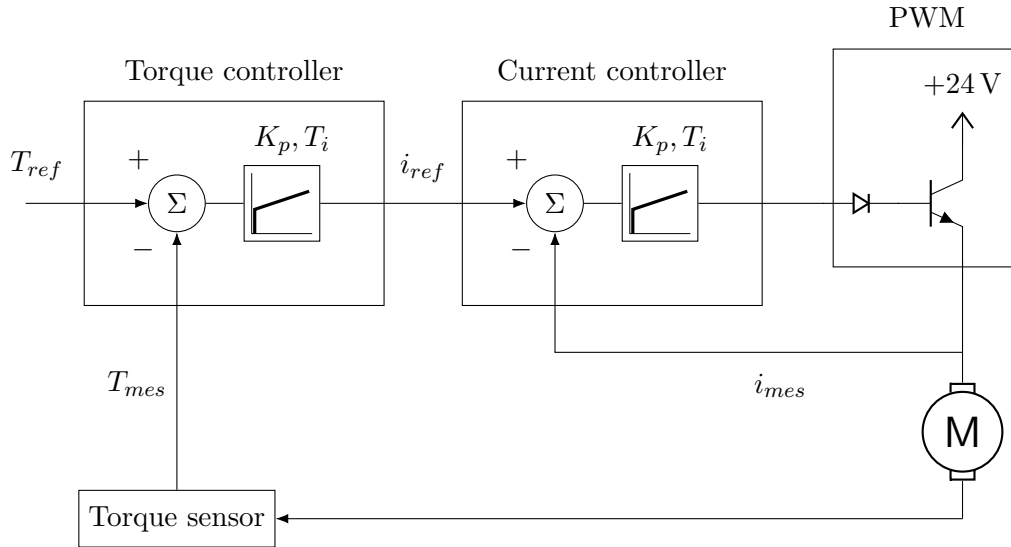


Figure 4.5: Schematic representation of the torque regulation of the titling motor M

The controller is basically not designed to manage such a torque control which is therefore carried out by deceiving the existing system and using the sensor output voltage to perform the operation. The torque conversion is then recovered by applying the transfer function of the sensor.

As this additional control loop is performed directly on the driver, the processing becomes quite faster. This trick allows avoiding communication delays between two control loops operating on two separate devices and reducing processing time.

¹as it is the case for most torque controllers

²The pulsed current requires an electronic filtering and does not allow for a fast enough answer.

4.2 High-level control

4.2.1 DTC control

In this configuration, the steering angle of the vehicle is controlled to ensure a given curvature radius of the vehicle's trajectory. Besides that, the tilting of the vehicle is directly controlled in position through its corresponding actuator while the propulsion speed of the rear wheel is controlled in order to ensure a given vehicle's tangential speed.

The tilting angle of the frame is therefore imposed to compensate the lateral dynamic effects of the vehicle, which should theoretically lead to a self-stability of the tilted vehicle in cornering (and thus to a zero roll torque on the vehicle's frame).

Those two set points (namely δ , the steering and θ , the tilting) are computed as follows :

$$\delta = \text{atan}\left(\frac{L}{r}\right)$$
$$\theta = \text{atan}\left(\frac{V^2}{gr}\right)$$

with L being the wheelbase of the scaled model, r the curvature radius of its trajectory and V the tangential speed of the vehicle

Based on the same principle, a second DTC controller has been developed, to use offline pre-computed dynamic equilibria of the vehicle to determine the tilting and steering set points to apply to the vehicle. Those reference values are computed using multibody simulations to determine the configuration in which zero tilting and steering torque should theoretically be obtained (for a given steering radius and vehicle's speed). This second controller is currently not used, given the limited reliability of the computed equilibria.

Those results are then stored in two separated Look-Up Tables³ used by the controller to retrieve the most suitable controls to apply during vehicle operation.

4.2.2 STC control

For this second controller, the tilting actuator is physically disconnected from the rest of the vehicle (see Section 4.1.3). The tilting angle is then controlled by means of the steering actuator through a simple torque PID controller, using tilting references and feedback information from the actual tilting angle. Finally, the vehicle's speed is controlled by the rear wheel actuator of the demonstrator.

The target inclination angle used by the controller is then computed using the same theoretical tilting equilibrium $\theta_{ref} = \text{atan}\left(\frac{V^2}{rg}\right)$ as for the DTC controller.

This STC controller thus carries out a cornering entry in the following manner : first, it induces a counter-steering, using a first target reference until the vehicle reaches its auto-stable tilting-angle for the selected curve and, secondly, it induces a steering in the opposite direction in order that the vehicle heads to the desired path.

³a first one for the steering and the second one for the tilting

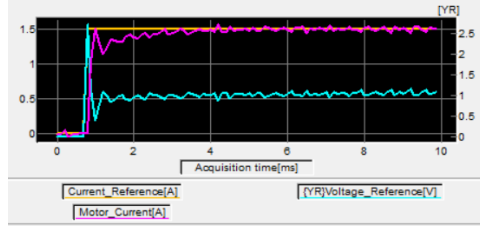


Figure 4.6: Torque control response of the tilting actuator, for a reference torque of 1.1 Nm

4.2.3 Protection mechanism

A supervision of the control data is carried out, in order to protect the vehicle and regardless of the vehicle control mode, DTC or STC.

As the computed set points include an arctangent, the resulting values could potentially be $[-90^\circ; 90^\circ]^4$; therefore the obtained values are compared to the envelop values of the front suspension (Figure 4.7) and are limited to these ones in order to avoid damaging the suspension.

In addition to that, a warning is sent to the end-user to alert him on the fact that the chosen configuration cannot be reached given the mechanical limits of the vehicle (measured in 2017 [23]).

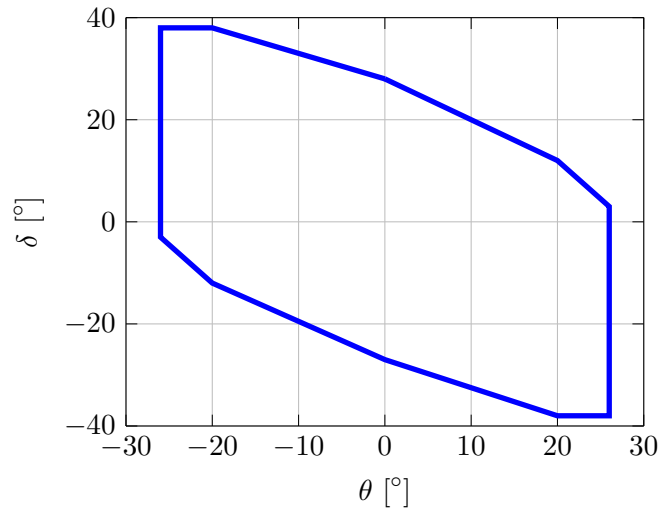


Figure 4.7: Mechanical limits of the front suspension, function of the steering (δ) and tilting (θ) of the vehicle[23]

Too small radii are mechanically not feasible (i.e. $R < 0.5\text{m}$) and are therefore interpreted as straight lines orders to avoid considering meaningless commands.

Finally, the controller's tilting acceleration is limited to ensure that a brutal torque control will not lead to the vehicle's unballasting.

⁴A negative radius will thus lead to a counter-clockwise cornering of the vehicle.

Chapter 5

Experimental Validation

This chapter presents the tests performed on the demonstrator to assess the various controls involved in the vehicle's automation.

It describes the tests carried out and explains the results obtained in order to conclude on the validity of the update of the original device including its newly implemented sensors to perform the controls and, possibly, to deduce future improvements.

It also aims at analyzing the vehicle's torque sensor measurements, which are a good indication of how close from unballasting is the vehicle.

It should be noticed that the vehicle's positioning is validated through the use of an absolute position system from the civil construction laboratory.

5.1 Untilted vehicle

In order to characterize the vehicle behavior and validate the implemented controls, a first batch of tests was carried out with various NTV trajectories, from linear to curved and complex ones (curvature entering or exiting).

5.1.1 Straight line Acceleration

The vehicle is first controlled to go into a straight line trajectory with a given speed configuration, before applying steps to the speed reference of the vehicle. This reference is then compared to the measured speed¹ in order to characterize the speed control of the demonstrator's propulsive wheel.

Figure 5.1 shows a time constant of 4 seconds for the speed controller. This is relatively slow but appears to be coherent with the mass/power ratio of the vehicle (28kg for a 100W propulsion actuator).

¹computed by merging information from odometry and the IMU with the extended Kalman filter

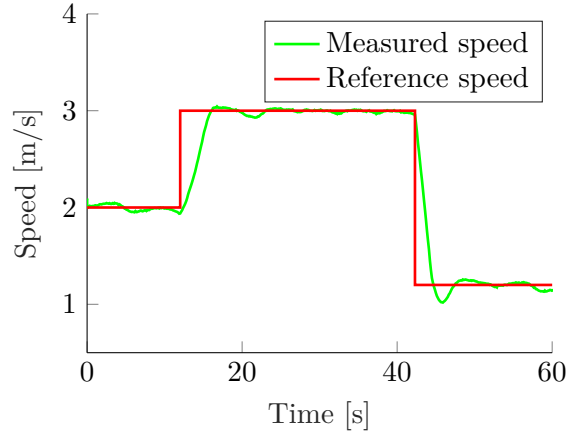


Figure 5.1: Time variation of the vehicle's speed and its set point, for a straight line trajectory

5.1.2 Cornering vehicle

Speed effect

The vehicle was then studied in an untilted cornering configuration for a curvature radius (r) of 4m, and therefore a steering angle (δ) of 11.6° . The tilting angle is then maintained at 0° by its actuator and a speed reference is applied between 0.5 m/s and 2 m/s, by steps of 0.5 m/s (Figure 5.2).

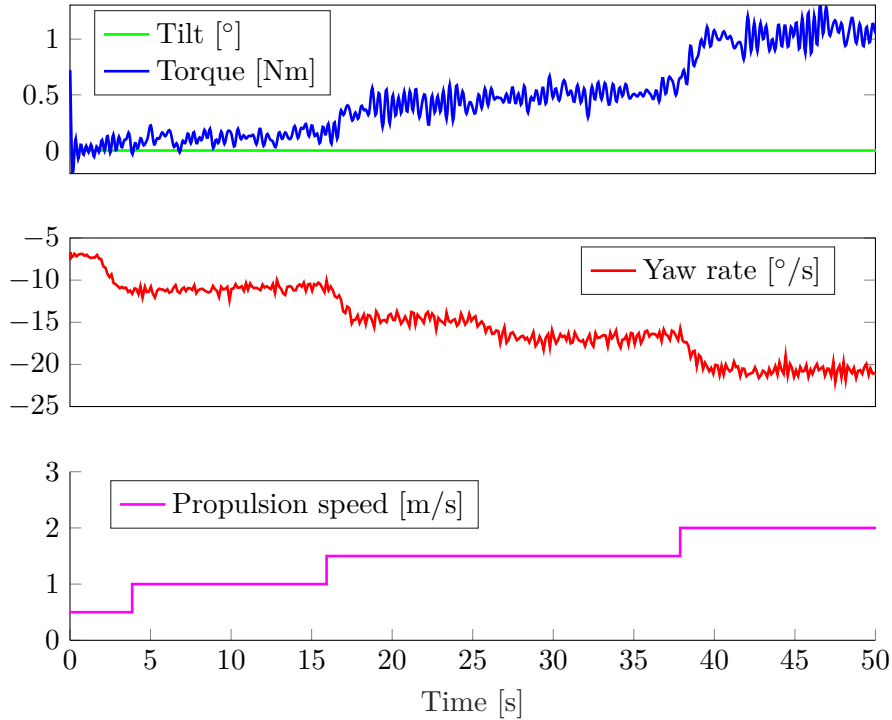


Figure 5.2: Vehicle behavior (tilting torque, tilting angle and tangential speed) time evolution for various cornering speeds, for $\delta_{FD} = 28^\circ$ and $\theta = 26^\circ$

The results allow to observe the evolution of the torque sensor's measurements depending on the vehicle's speed. It can be noticed that, for a given curvature radius of the vehicle, the torque measurements depend directly on the vehicle speed. As expected, the torque variation appears to be linked to the squared velocity of the vehicle, confirming the strong relation between the

centripetal acceleration of the vehicle and the measured T-pendulum torque. More specifically, a speed increment of 1 m/s between 0.5 and 1.5 m/s results in the same torque increase as a speed increment of 0.5 m/s between 1.5 and 2 m/s.

Tilting angle variation

The impact of the demonstrator's tilting angle during cornering is then observed by running the vehicle at 3 m/s, with a curvature radius of 4m (and thus a steering $\delta = 11.6^\circ$). In this configuration, the theoretical tilting angle is $\theta = \text{atan}\left(\frac{9}{4*9.81}\right) = 12.9^\circ$.

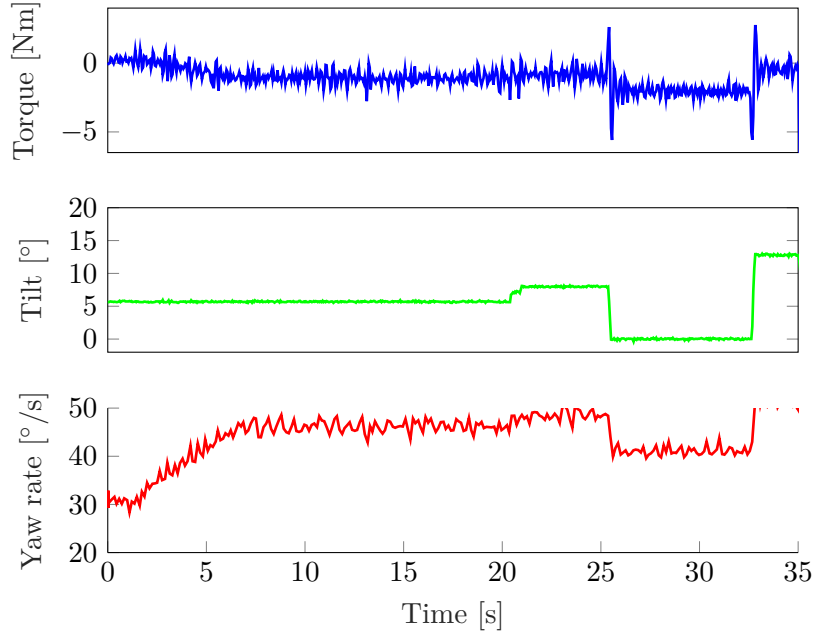


Figure 5.3: Vehicle's behavior evolution for different actuated tilting angles

The inclination angle is successively imposed to $6^\circ, 7^\circ, 8^\circ, 12.7^\circ$ and 14° . It can be observed from the graphs of Figure 5.3 that an increase of the tilting angle leads to a decrease of the frame's tilting torque while in contrast, increasing this tilting angle (and therefore inclining the vehicle inwards its curved trajectory) leads to a growth of the tilting torque. In addition to that, the results show that zero torque is achieved for a tilting angle of 12.6° , which is close to the theoretical tilting angle computed to achieve zero tilting torque.

5.2 DTC controlled vehicle

Taking into account the influence of the different vehicle's state variables on its behavior, the vehicle was tested using its DTC controller, which actuates the vehicle as follows : $\delta_{ref} = \text{atan}\left(\frac{L}{r}\right)$ and $\theta_{ref} = \text{atan}\left(\frac{V^2}{rg}\right)$ (see Chapter 4). Monitoring the position of those two angles allowed then to stress the reference of the different actuators. In addition to that, an analysis of the torque measured on the vehicle frame and the vehicle's yaw rate allows to characterize the steering and tilting controls of the vehicle. Indeed, torque measurement can be used to determine whether the tilting reference is adapted to the vehicle's state, allowing to compensate perfectly for the centrifugal effect on the running demonstrator which should thus lead to a zero torque measurement. This torque measurement can also be used to characterize the torque effort applied by the tilting actuator to incline the vehicle.

Finally, given that the yaw rate of the vehicle ($\dot{\psi}$) is directly linked to the speed of the vehicle and the curvature radius of its trajectory ($\dot{\psi} = \frac{V}{r}$), measuring the vehicle speed and its yaw rate allows to confirm that the reference radius of the DTC controller matches the actual steering radius of the DTC controlled vehicle.

5.2.1 Steady-state behavior

The vehicle is then controlled in DTC, with a constant reference radius of 4m and a target speed of 0.5m/s, leading to a steering reference angle $\delta_{ref} = \text{atan}\left(\frac{L}{R}\right) = 14^\circ$ and a tilting reference angle $\theta_{ref} = \text{atan}\left(\frac{V^2}{Rg}\right) = 1.9^\circ$ (Figure 5.4).

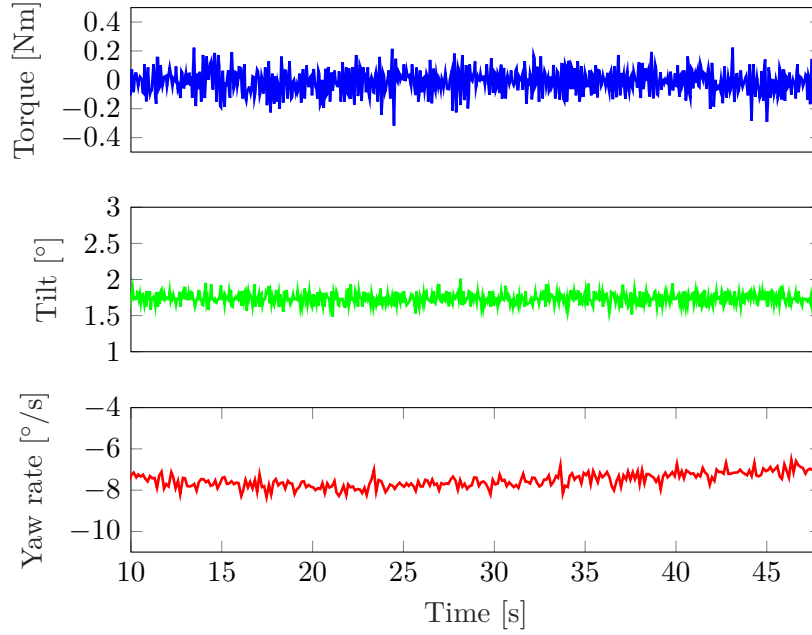


Figure 5.4: Vehicle's behavior for DTC control, with $V=1\text{m/s}$ and $r=3\text{m}$

It is observed that, for a tilted vehicle, the measured torque oscillates around zero. On the contrary, for an untilted vehicle, a non-zero torque value is measured, due to the centrifugal effect, see also Figures 5.6(a) and 5.7(b).

5.2.2 Trajectory variation

Corner entry

The behavior of the vehicle entering a curve ($r = 3\text{m}$) is assessed.

An increase of the torque on the T-pendulum can be seen in the graphs for the non tilted vehicle entering a curve, see Figure 5.6(a).

On the contrary, for the tilted vehicle, due to the position control of the tilting motor, this torque is significantly lower than for the untilted vehicle, see Figure 5.6(b).

In all configurations, a torque overshoot is observed that could be explained by a lack of fluidity in the transition orders.

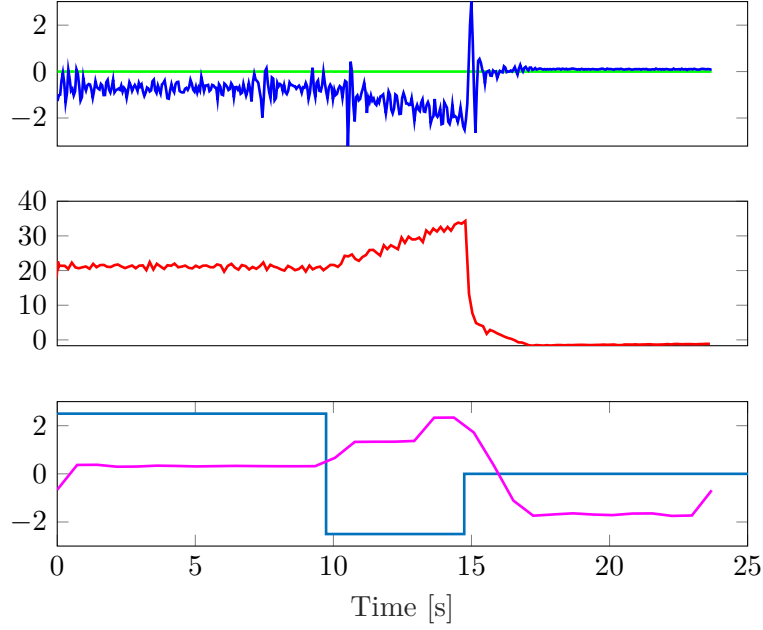


Figure 5.5: Vehicle's behavior for an tilted cornering entry, with $V=1\text{m/s}$ and $r=3\text{m}$

Corner exit

The behavior of the vehicle exiting a curve ($r = 3\text{m}$) is assessed.

A decrease of the torque on the T-pendulum can be seen on the graphs for the non tilted vehicle exiting a curve, see Figure 5.7(a).

On the contrary, for the tilted vehicle, due to the position control of the tilting motor, a lower torque is observed except during the transient where the overshoot is much larger than for the untilted vehicle, see Figure 5.7(b).

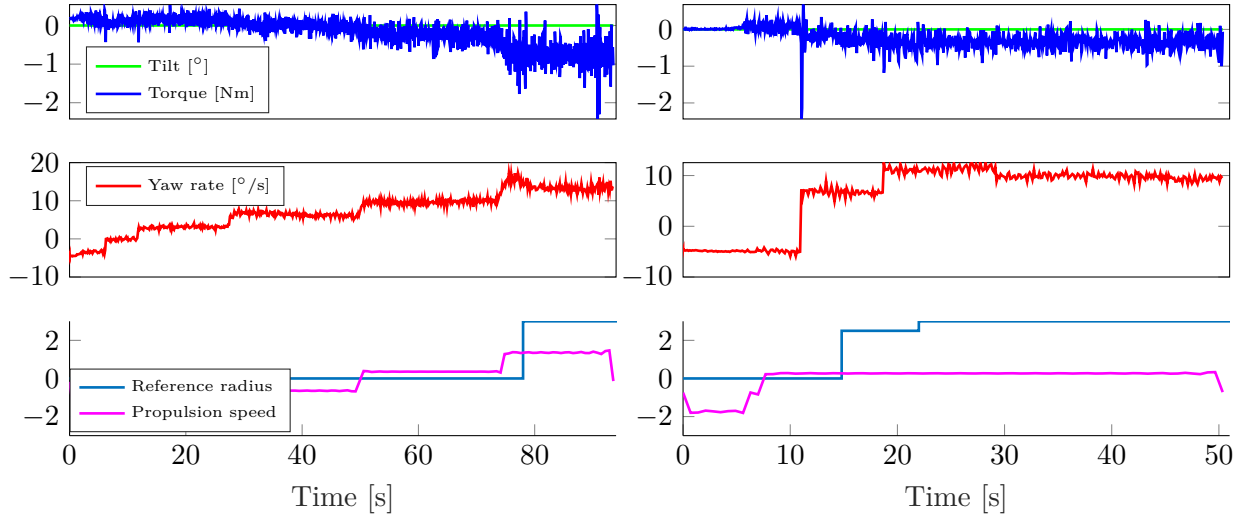
As for the corner entry, a torque overshoot is observed that could be explained by a lack of fluidity in the transition orders.

Curvature radius modification

More complex trajectories are then performed with the vehicle. For example, a clockwise curve exit is carried out, followed by a short straight line trajectory before steering in the opposite direction, to perform a counter-clockwise trajectory. In addition to that, a second transition is executed, from counter-clockwise to clockwise curve trajectory, see Figure 5.8.

During all those maneuvers, the vehicle is kept at a constant speed of 1 m/s while the curvature radius set point of the controller was alternated between 0m (straight line) and $\pm 2.5\text{m}$. A resulting tilting angle should therefore be 0° or $\pm 2.3^\circ$, depending on the DTC controller's reference. Furthermore, those trajectories should lead to a measured yaw rate of $11.5^\circ/\text{s}$. The resulting evolution of the tilting angle and the vehicle's yaw rate confirm the correct reference tracking of the vehicle, according to those expected set points.

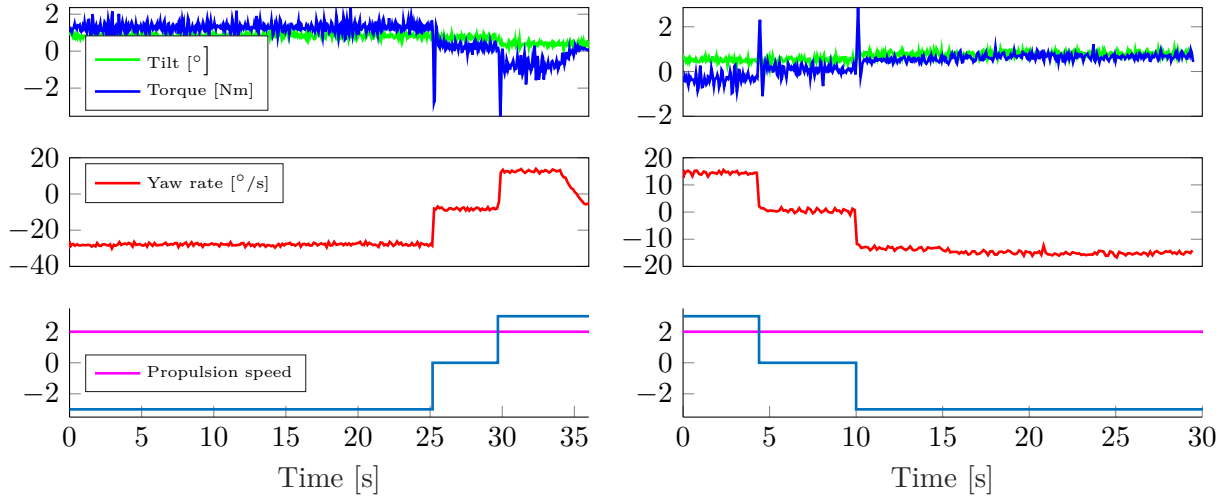
In addition to that, the T-pendulum torque measurements show that the torque increase due to tilting actuation during those routing changes remains far below the expected unballasting torque of the vehicle (11Nm). This confirms that the vehicle's stability is maintained on such a trajectory.



(a) Untilted vehicle during cornering entry

(b) Tilted vehicle during cornering entry

Figure 5.6: Corner entry



(a) Untilted vehicle during cornering exit

(b) Tilted vehicle during cornering exit

Figure 5.7: Corner exit

5.3 STC controlled vehicle

Auto-stability test

The vehicle is tested by disconnecting the tilting actuator from the rest of the vehicle and by making it move on a straight trajectory to determine the auto-stability speed from which it will stand up alone without any control need. This speed is measured as being 2.6m/s. This also represents the speed from which the STC control is functional.

STC testing

It should be mentioned that the STC control tests intrinsically needs the vehicle moving at high speed to be performed. Such high speed tests require therefore a large and strictly flat surface to be feasible. On such a surface, it appears that, in its present configuration, the vehicle can not

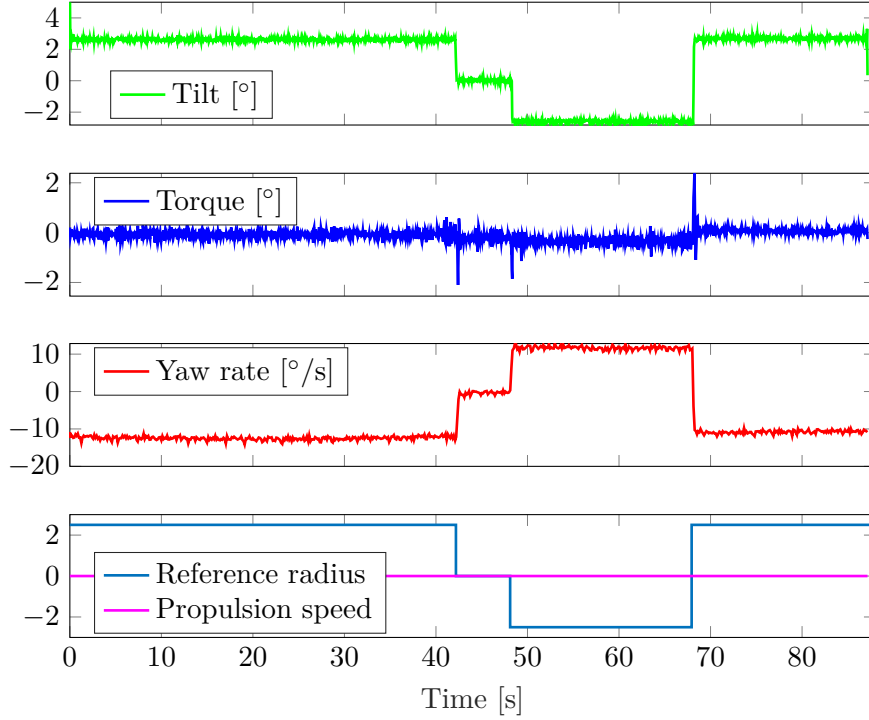


Figure 5.8: Vehicle behavior (tilting torque, tilting angle and tangential speed) time evolution for various curvature radii, for $\delta_{FD} = 28^\circ$ and $\theta = 26^\circ$

move faster than 3.3m/s which is a limitation for the STC control. Taking into account the fact that the optimization of the gear ratio is already maximized (see Chapter 1), this means that the motor power could be an issue for the future improvement of the STC control.

5.4 Localization

In addition to the embedded position measurement on the vehicle, a external position measurement is carried out with a theodolite enabling to provide an absolute measurement of the position of the vehicle (without drift in time) which is much more accurate than the first one. The measurement set-up is based on a theodolite tracking automatically the position of a prism, fixed on the vehicle, to provide the (x, y, z) coordinates of the prism in function of the time.

These measurements require to be post-processed to enable the coordinates transfer from the prism reference to the vehicle's center of mass reference, which is highly influenced by the tilt of the vehicle. This being done, the measurements obtained with the theodolite can easily be compared with those coming from the other sensors.

Figure 5.9 shows the measurement set-up (on the right side) and the assembly of prism on the vehicle (on the left side). An example of position measurements using the theodolite is given in Figure 5.10 and an example of the resulting velocity computation can be seen in Figure 5.11.

5.5 Feedback from the experimental results

In addition to those simulations of the moving vehicles, other important information about the vehicle's behavior can be derived from those running tests. Indeed, the tests results allow to

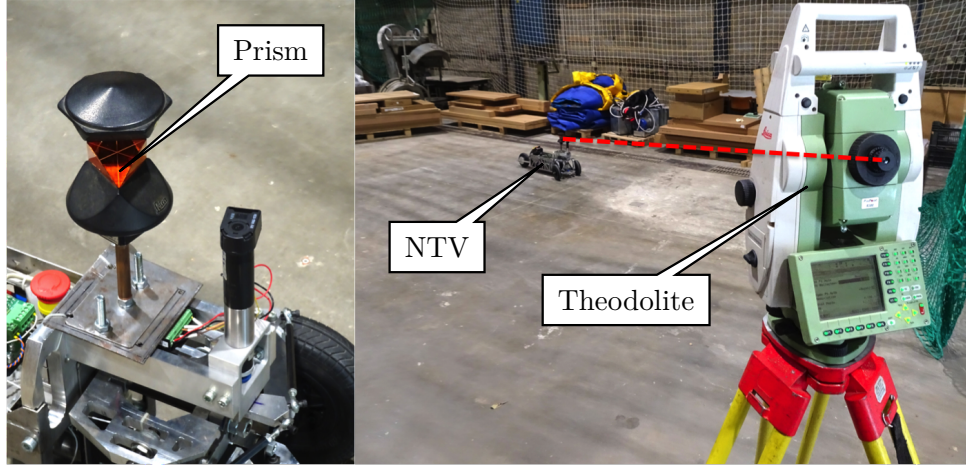


Figure 5.9: Theodolite measurement set-up (right) and prism location on the NTV (left)

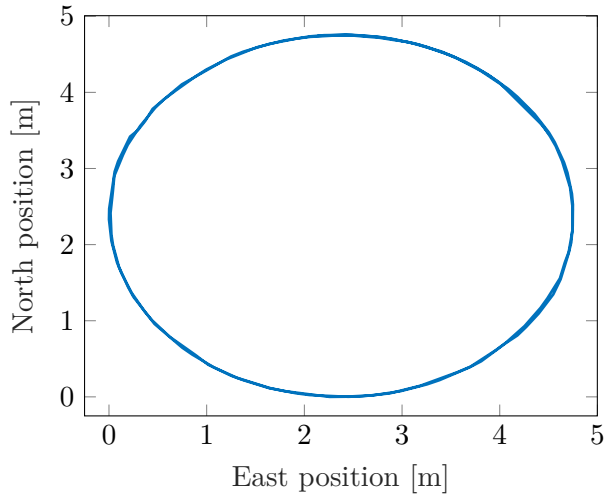


Figure 5.10: Vehicle position measurements using the theodolite

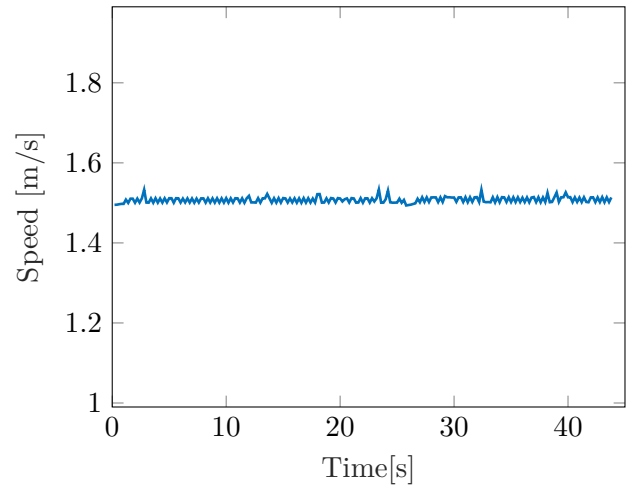


Figure 5.11: Vehicle velocity computation using the theodolite

characterize the maximal update frequency of the controller (which depends highly on the CAN bus performances). Besides, those trial runs enable to detect the occurrence of strong oscillations in the demonstrator's rear-end, above a critical speed.

5.5.1 Rear wheel misalignment

As it is observed in Figure 5.12, at the maximum speed (3.3m/s) the vehicle begins to experiment oscillations coming from the rear wheel during motion on an imperfect concrete surface (drone zone). These oscillations could be due to the bent rear wheel and the misalignment of its rotation axis. A manual mechanical adjustment of the rear wheel allows to decrease the phenomenon when the vehicle is in motion on a perfect surface (in-door sport room). Nevertheless, it should be useful to foresee some dedicated tool to adjust the alignment of the rear wheel along the vehicle axis, in a more accurate way.

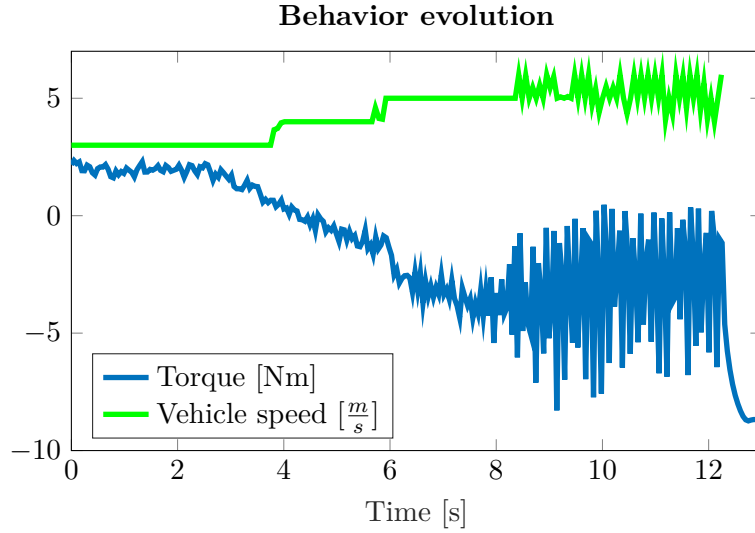


Figure 5.12: Vehicle rattle apparition

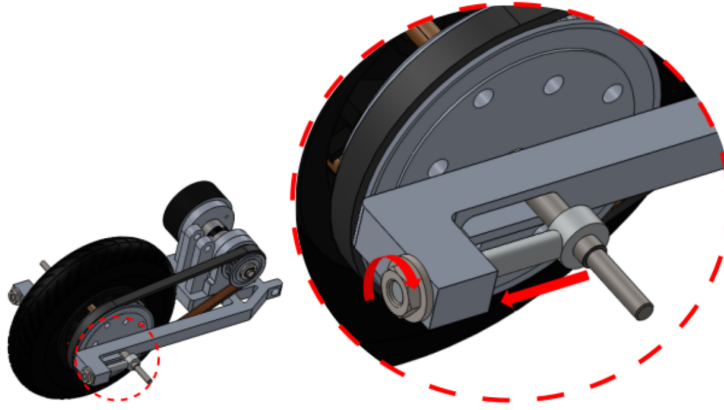


Figure 5.13: Real wheel tensioning and alignment system [23]

5.5.2 Tilting actuation

As it can be seen on the graph, the tilting motor, which is widely oversized, operates a too rough position control of the vehicle. As a consequence, torque peaks can be measured on the vehicle frame during the actuation phases of the motor. More precisely, for a reference value of the control that would be very different from the actual vehicle configuration, the rough functioning of the controller could leads to unballasting. Smoothing the behavior of the controller or limiting the motor output torque should be considered to improve the control safety of the vehicle.

This unballasting was observed for step of the tilting set point larger than 15° .

5.5.3 CAN communication

The CAN communication of the vehicle appears to be the bottleneck of the control architecture of the vehicle. Indeed, given the maximal baud rate of the implemented communication bus (500kbps) and the high volume of information that it transfers, it does not allow to perform an information² refreshment rate higher than 200 Hz, limiting accordingly the working frequency of

²For the information that is transferred through the CAN bus like the motors positions, accelerations,...

the main controller of the onboard computer. It is not a limiting factor for the DTC control, but this should be taken into consideration for the implementation of higher level controllers.

5.6 Conclusions of the experimental validation

Numerous tests were carried out to assess the vehicle's behavior during various controls involved in the vehicle's automation.

The experimental validation shows that the vehicle operates efficiently when automatically controlled. It shows that the electro-mechanical and communication updates that were implemented during the present work are functional and relevant. Indeed they allow to fulfill the objectives of the work, i.e. the full control of the vehicle.

In addition, it can be observed that the behavior of the vehicle during the experimental validation is close to the expected theoretical behavior. For the future, it would be useful to fully validate the STC controller and to compare all the experiments results to multibody simulations to characterize the relationship between practical and theoretical results.

5.7 Future prospects

The feedback from the tasks achieved during the update of the demonstrator and from the behavior of the vehicle during the experimental tests allow to identify various future prospects for the evolution of the NTV.

They can be organized in two main categories

1. Electro-mechanics

- Due to the observed bending of the rear wheel, considering implementing an accurate wheel adjusting system to avoid vibrations and perturbations at high speed would be of interest.
- Increasing the propulsive motor power could be considered to allow the vehicle to reach higher speeds.
- The theodolite could be used to communicate in real time with the vehicle, providing therefore an absolute (undrifted) positioning information that can be used by the vehicle controllers.

2. Control and simulations

3. Characterizing the operation of the implemented STC controller should be considered. In addition to that, improving the torque control of the tilting actuator would allow to perform an "hybrid" SDTC control. Finally, more complex controllers, like optimal controllers or predictive controllers could be implemented on the vehicle, to handle in an adequate way highly non-linear systems like a running NTV.
4. The multibody simulator of the vehicle could also be refined by modifying the simulated controllers, to be as close as possible to the real ones. Likewise, modelling the vehicle's actuators using their internal characteristics could be foreseen.
5. Various parametric studies could be carried-out on this finally running scaled model, allowing to analyze the impact of different suspension configurations on the dynamic behavior of the vehicle. Those studies could also be compared to the multibody simulations

of the vehicle, in order to establish the link between the theory and the observed behavior of the vehicle.

Conclusion

Narrow Tilting Vehicles (NTV) offer a practical solution to the increasing traffic congestion as well as an optimization of the energy footprint of vehicles.

The present work focuses on the electro-mechanical update, control and validation of a previously designed and realized NTV[23], in order to deliver a fully functional embedded controlled system with transmission in real time of the measured and controlled parameters (speed, acceleration, odometry, tilt, torque, ...)

The project started from the actual status of the demonstrator. Based on the feedback from the previous work and on the observation of the vehicle's behavior at the beginning of the present thesis, it considers the optimization of the vehicle's functionalities to allow the optimal development of the control.

The multi-disciplinary aspect of the project gives this work an unique dimension. Numerous achievements are made, in electro-mechanics, communication, sensing and control, to improve the demonstrator and to enable its automated control in motion, complying thereby with the design condition that the NTV has to be usable by people with limited training.

The three axes of development are detailed and the choices are discussed : the mechanical design, the electronics including the Raspberry Pi 3B nano-computer, the transducers, the data management, the low level control integration through the CANopen communication and the high level DTC and STC controls.

The experimental validation shows that the vehicle operates efficiently when automatically controlled. The feedback from the tests shows the relevance of the retained solutions, in particular, the use of the strain gauges for the torque sensor, the implemented measurements and control modules and the Wi-Fi communication. The experimental validation confirms that the current demonstrator's operation fulfills its objectives and allow the full control of the vehicle.

The integration of the various modules in the whole system proves to be successful, allowing therefore to constitute a robust basis for future work on the subject. Indeed, the validation leads to identify various possible tracks for evolution of the Narrow Tilted Vehicle, making this work a step forward in the more general context of the optimization of transportation means to answer the needs of the modern society.

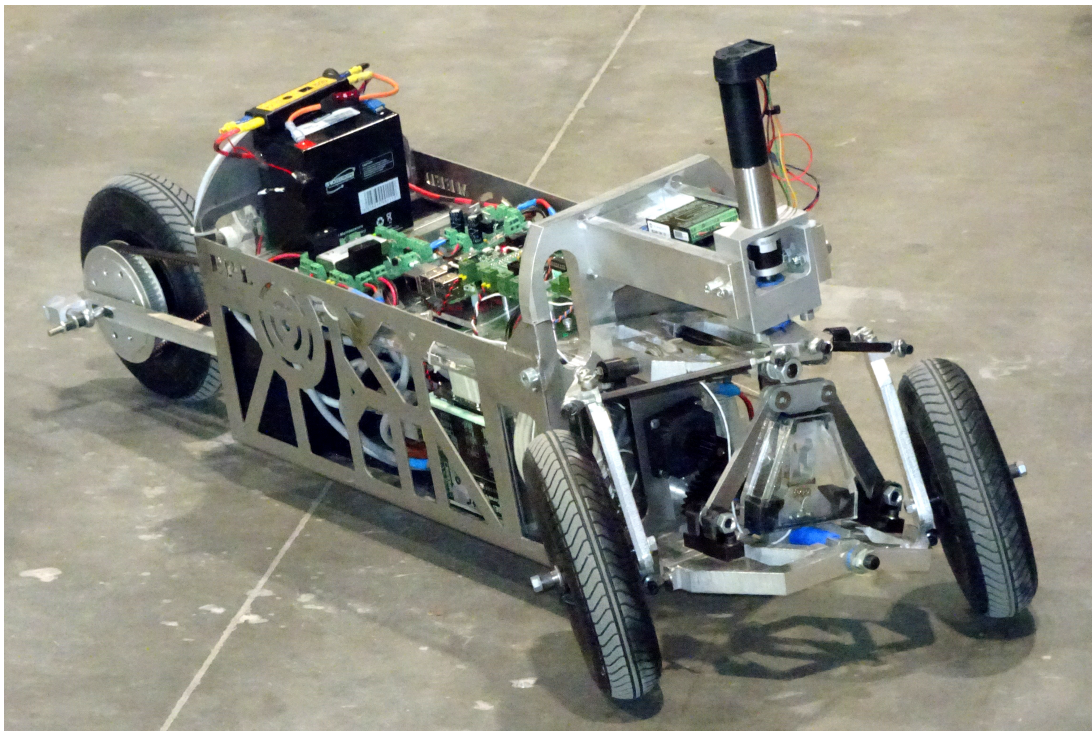


Figure 5.14: Cornering scaled model

Bibliography

- [1] Px4 autopilot team developpers. Pixhawk autopilot. <https://pixhawk.org/modules/pixhawk>, 2017. Accessed: 19-04-2017.
- [2] Px4 autopilot team developpers. Pixhawk flight controller hardware project. <https://pixhawk.org/>, 2017. Accessed: 14-04-2017.
- [3] Avago. Heds-5500 a06. <https://www.broadcom.com/products/motion-control-encoders/incremental-encoders/transmissive-encoders/heds-5500a06>, 2014. Accessed: 18-05-2017.
- [4] JJ Chong, James Marco, and David Greenwood. Modelling and simulations of a narrow track tilting vehicle. *Exchanges: the Warwick Research Journal*, 4(1):86–105, 2016.
- [5] Raspberry community. Can bus on raspberry pi with mcp2515. <https://www.raspberrypi.org/forums/viewtopic.php?f=44&t=141052>, 2016.
- [6] Joseph R Davis. *Alloying: understanding the basics*, chapter Aluminium and aluminium alloys, pages 355–356. ASM international, 2001. <https://materialsdata.nist.gov/bitstream/handle/11115/173/Aluminum%20and%20Aluminum%20Alloys%20Davis.pdf?sequence=3&isAllowed=y>.
- [7] Battery Discount. Vente aux particuliers et professionnels. <http://www.batterydiscount.be/>, 2017. Accessed: 14-04-2017.
- [8] Quentin Docquier, Aubin Verlé, and Paul Fisette. Narrow tilting vehicle (ntv) control through mbs modelling : Comparison of direct and steering tilt control, September 2016.
- [9] Vishay electronics. Fully sealed potentiometer professional grade. <https://www.vishay.com/docs/51037/pe30.pdf>, 2017.
- [10] Faulhaber. Micromoteurs c.c. série 2657...cxr. https://fmcc.faulhaber.com/details/overview/PGR_3890_13818/PGR_13818_13813/fr/GLOBAL/, 2017. Accessed: 12-01-2017.
- [11] The Raspberry Pi foundation. Product: Raspberry pi 3b. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>, 2016. Accessed: 22-04-2018.
- [12] Maxon motors. Program 2014/15 high precision drives and systems. <http://pdf.directindustry.com/pdf/maxon-motor/program-2014-15/7173-600371.html>, 2015. Accessed: 18-05-2017.
- [13] ALFA Network. Alfa network long range wifi dongle awus036ac. <https://www.alfa.com.tw/WiFi%20USB%20Antenna.html>, 2017. Accessed: 14-03-2017.
- [14] University of Michigan. Inverted pendulum: System modeling. <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>, 2014.
- [15] Wikipedia online. Robot operating system, an operating system for robots. https://fr.m.wikipedia.org/wiki/Robot_Operating_System, January 2016.

- [16] Deville rectification. 7075 aluminium alloy. http://www.intersteel.co.uk/_layout/pdf/7075.pdf, 2011.
- [17] Jihan Ryu and J Christian Gerdes. Estimation of vehicle roll and road bank angle. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2110–2115. IEEE, 2004.
- [18] Nicolas Sorensen, Rémy Vermeiren, and Pierre Latteur. Development of a guidance and automatic positioning system for builder drones with a theodolite. <https://dial.uclouvain.be/memoire/ucl/fr/search/site/theodolite>, 2018.
- [19] Technosoft. Ibl2401 - intelligent drives and motors. <http://www.technosoftmotion.com/en/intelligent-drives-and-motors/other-drives/closed-frame/ibl2401>, 2017. Accessed: 18-04-2017.
- [20] Technosoft. Idm680 - intelligent drives and motors. <http://www.technosoftmotion.com/en/intelligent-drives-and-motors/other-drives/closed-frame/idm680>, 2017. Accessed: 18-04-2017.
- [21] Inc. The MathWorks. Estimate states of nonlinear system with multiple, multirate sensors. <https://nl.mathworks.com/help/control/ug/multirate-nonlinear-state-estimation-in-simulink.html>, 2017.
- [22] Inc. The MathWorks. State estimation using time-varying kalman filter. <https://nl.mathworks.com/help/control/getstart/estimating-states-of-time-varying-systems-using-kalman-filters.html>, 2017.
- [23] Sébastien Timmermans and Ghijselings Pierre. Conception et validation d’un véhicule étroit inclinable à échelle réduite. Master’s thesis, Ecole Polytechnique de Louvain, Belgique, 2017.
- [24] Sébastien Timmermans and Ghijselings Pierre. Mode d’emploi d’un prototype ntv, 2017.
- [25] ROS Wiki. Documentation-ros wiki. URL: <http://wiki.ros.org>, 2018.

Appendices

Appendix A

Torque sensor

A.1 T-pendulum design evolution

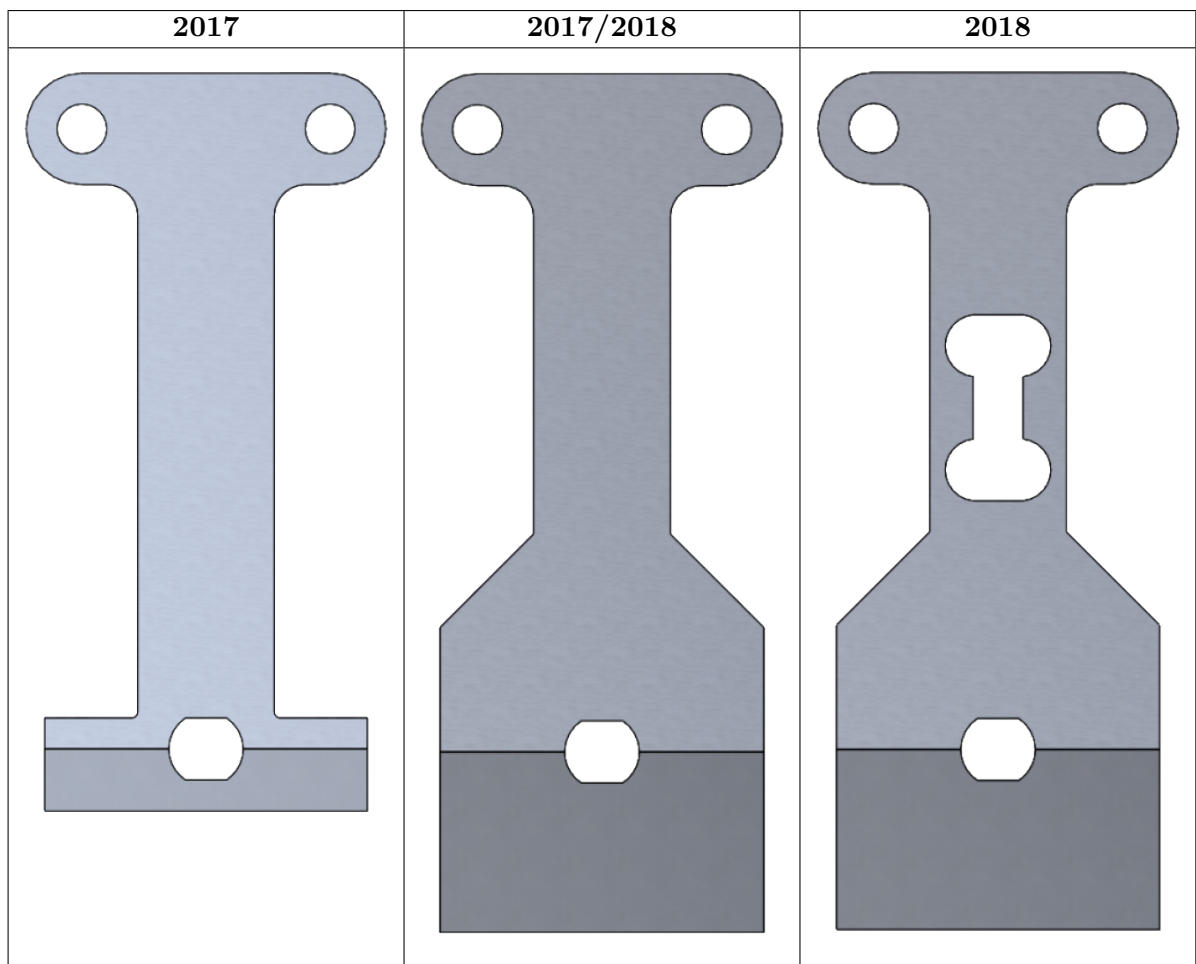


Figure A.1: Evolution of the design of the T-pendulum

A.2 Wheatstone measurement bridge configuration

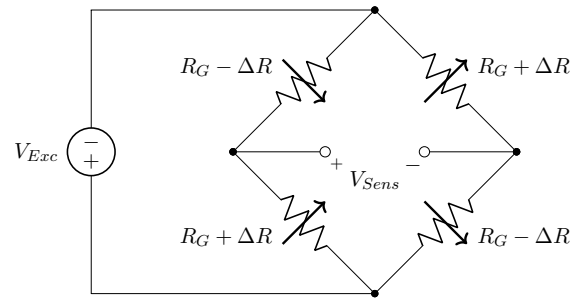


Figure A.2: Full Wheatstone bridge configuration of the strain gauges

A.3 Test specimen additional simulations

The screenshot shows the 'Bending - Binocular' software interface. It includes input fields for various parameters and a section for calculated values.

Parameter	Value	Unit
Applied Force, F	110	N
	11.217	kgf
Beam Parameters		
Distance Between Hole CLs, L	20	mm
Radius, r	5	mm
Beam Width, w	10	mm
Beam Height, h	22	mm
Minimum Thickness, t	2.5	mm
Modulus of Elasticity	70	GPa
Gage Parameters		
Gage Length	2.1	mm
Gage Factor	2.5	
Calculated Values		
Recommended Distance Between Gage Centerlines, Z	21.19	mm
Nominal Gage Strain	755	$\mu\epsilon$
Strain Variation	8.9	%
Span at Applied Force	1.887	mV/V

At the bottom of the window are buttons for 'Compute', 'OK', 'Cancel', 'Help', and 'Print'.

Figure A.3: Binoculaire design simulations using TransCalc

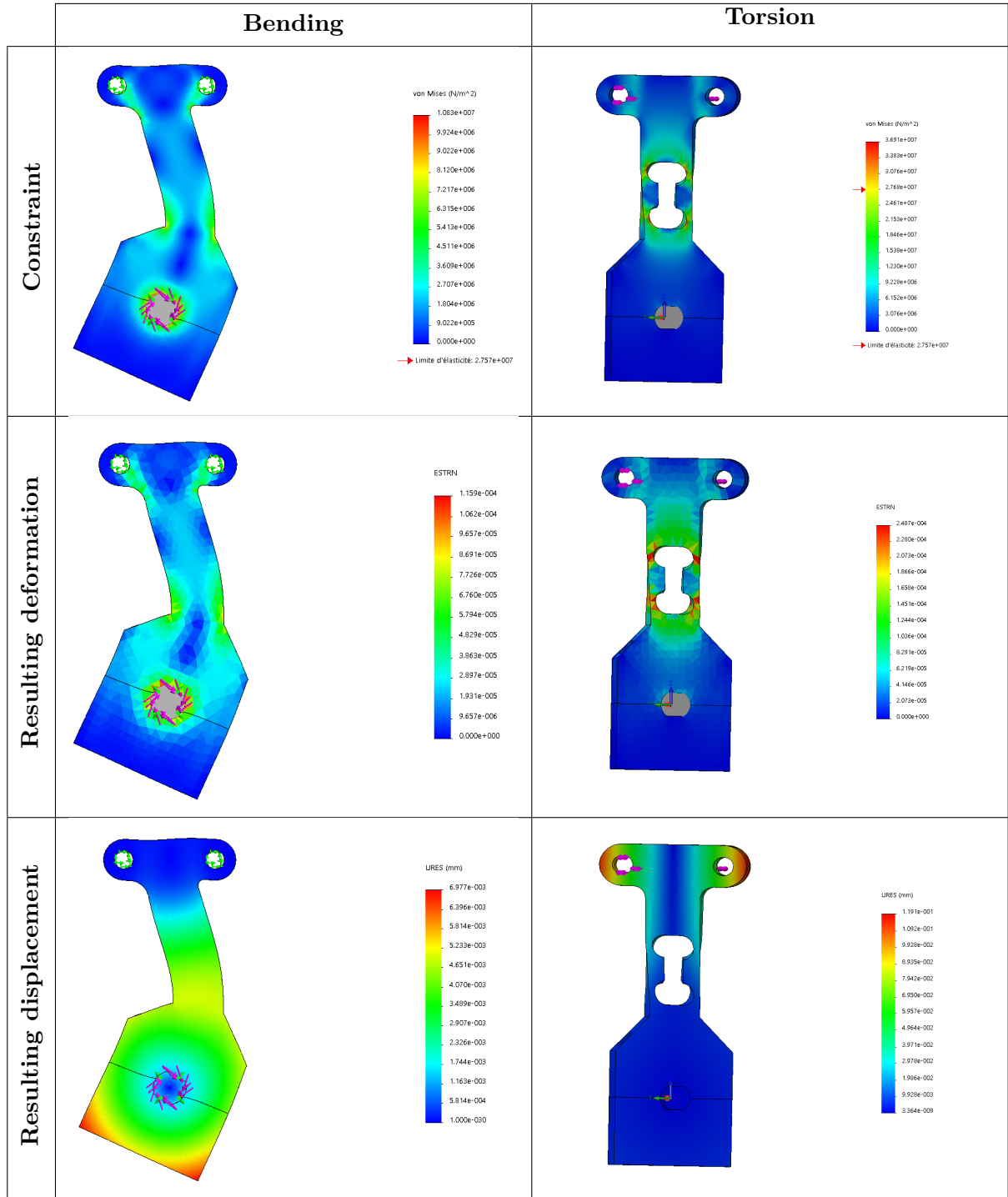


Table A.1: Simulations of the non-scooped T-pendulum bending and torsion deformation

Appendix B

Vehicle's electrical configuration

B.1 Electrical schemes (2017)

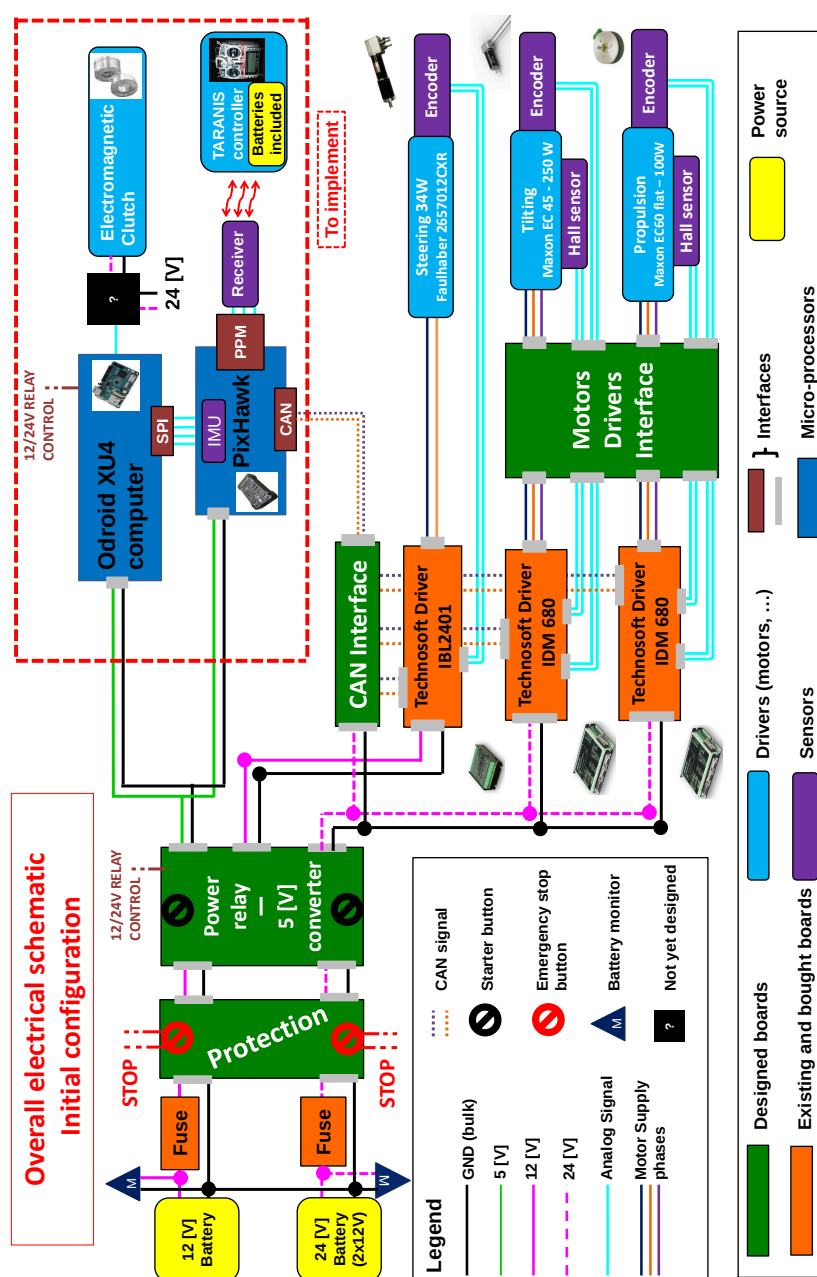


Figure B.1: Former electrical scheme of the prototype [23]

B.2 Electrical schemes (modifications)

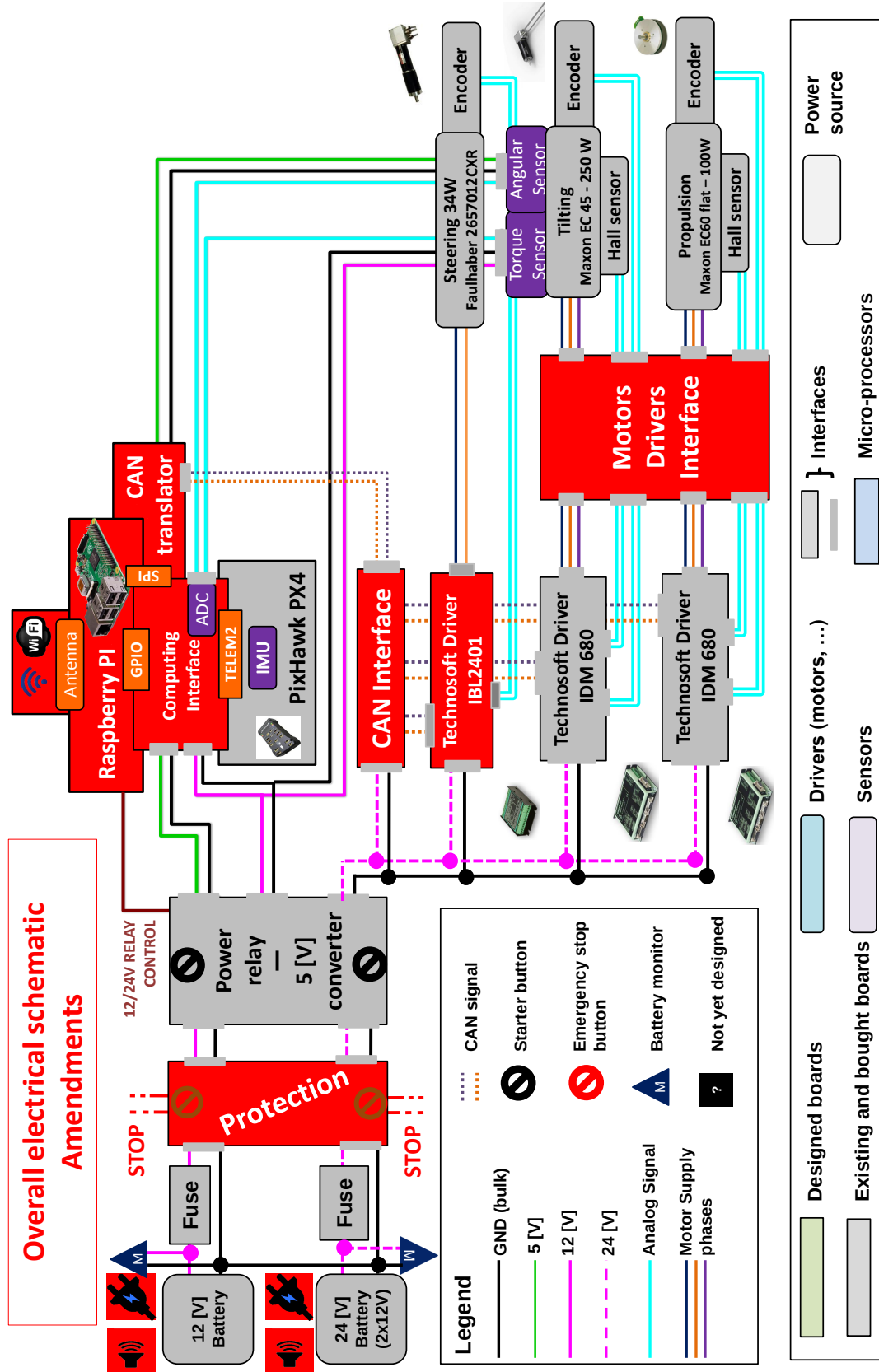


Figure B.2: Former electrical scheme of the prototype [23]

B.3 Electrical schemes (2018)

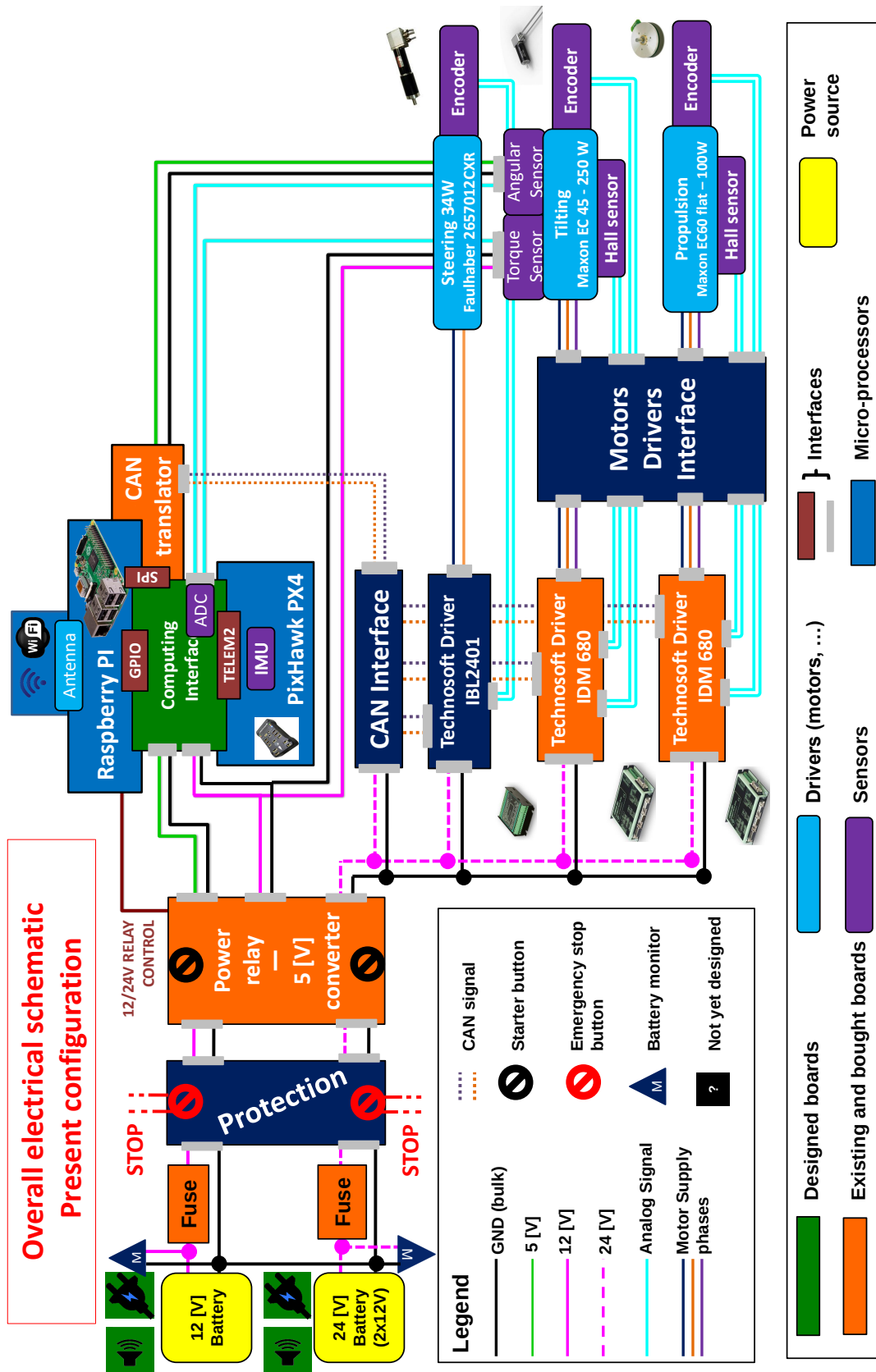


Figure B.3: Former electrical scheme of the prototype [23]

Printed Circuit Boards

Revised driver's interface board

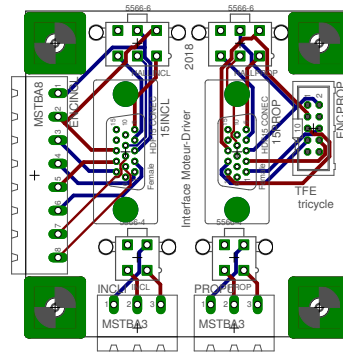


Figure C.1: Modified driver interface board, 2017-2018

Raspberry Pi's interface board

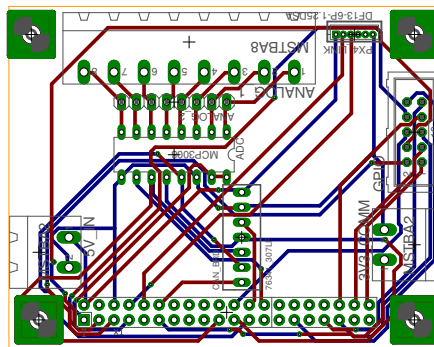


Figure C.2: Interface board for sensors, Raspberry Pi and Pixhawk Px4, 2018

Tilting sensor measurement board

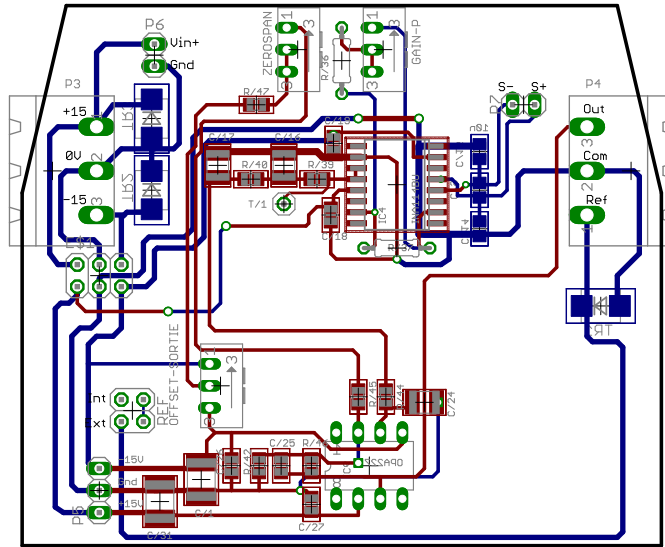


Figure C.3: Instrumentation amplification board of the torque sensor - UCL, 2012

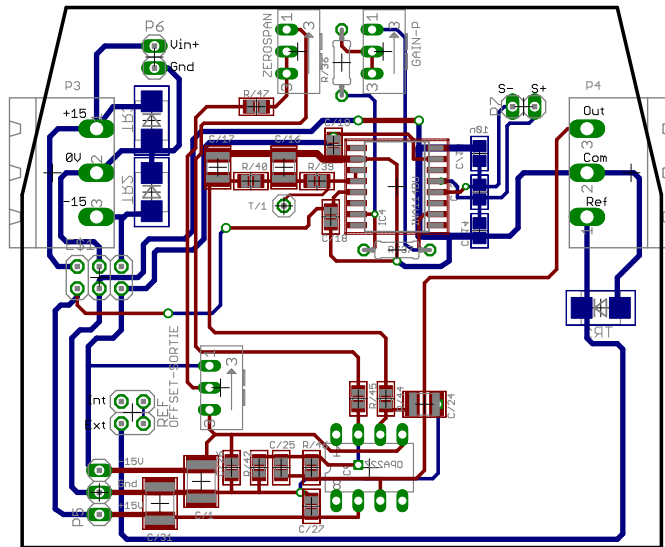


Figure C.4: Excitation and Powering board of the torque sensor - UCL, 2012

Appendix D

Interface Robotran/Simulink

D.1 Linéarisation du système dynamique

Le schéma global du système d'estimation d'état peut être observé à la Figure D.1

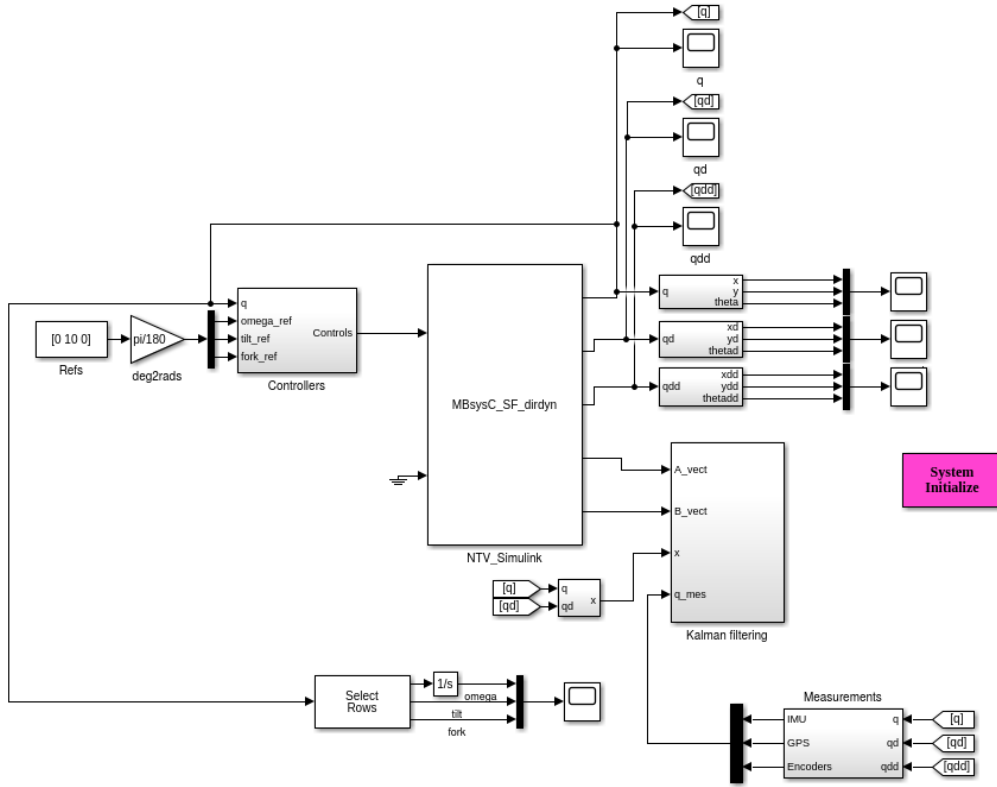


Figure D.1: Implémentation simulink du filtre de Kalman

Linéarisation du modèle

Comme décrit dans [8], l'expression analytique des matrices d'état du système dynamique peut être obtenue en partant des équations dynamiques réduites :

$$M_r(q_u)\ddot{q}_u + F_r(\dot{q}_u, q_u, Q_u) = 0$$

Et en choisissant l'expression du vecteur d'état comme valant $x = (\dot{q}_u, q_u)$

$$M_r(q_u^*)\Delta\ddot{q}_u + \underbrace{\frac{\partial F_r}{\partial \dot{q}_u}(*)}_{G_r} \Delta\dot{q}_u + \underbrace{\left(\frac{\partial M_r}{\partial q_u}(*)\ddot{q}_u^* + \frac{\partial F_r}{\partial q_u}(*)\right)}_{K_r} \Delta q_u - \Delta Q_u = 0$$

$$\underbrace{\begin{pmatrix} \Delta\ddot{q}_u \\ \Delta\dot{q}_u \end{pmatrix}}_{\Delta\dot{x}} = \underbrace{\begin{pmatrix} M_r^{-1}G_r & M_r^{-1}K_r \\ I & 0 \end{pmatrix}}_A \underbrace{\begin{pmatrix} \Delta\dot{q}_u \\ \Delta q_u \end{pmatrix}}_{\Delta x} + \underbrace{\begin{pmatrix} M_r^{-1} \\ 0 \end{pmatrix}}_B \Delta Q_u$$

Pour le modèle linéarisé du système, nous pouvons faire l'hypothèse que $\Delta x \simeq x$, pour de faibles variations d'état. Nous obtenons donc les matrices d'état linéarisées suivantes :

$$A = \begin{pmatrix} M_r^{-1}G_r & M_r^{-1}K_r \\ I & 0 \end{pmatrix} \in \mathbb{R}^{2n_x \times 2n_x}$$

$$B = \begin{pmatrix} M_r^{-1} \\ 0 \end{pmatrix} \in \mathbb{R}^{2n_x \times n_{qa}}$$

D.2 Validation de l'outil de linéarisation

Les équations d'état obtenues à l'aide du module de linéarisation implémenté dans Robotran ont ensuite été validées en implémentant un chariot à pendule inversé sur Robotran et en comparant les résultats numériques obtenus avec les résultats analytiques obtenus par [14], pour $\phi = 0$.

Modèle multi-corps

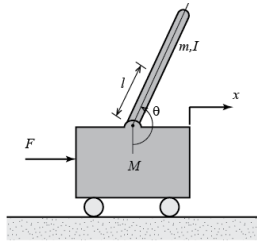


Figure D.2: Chariot pendulaire théorique

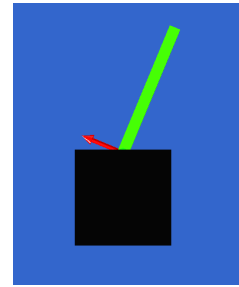


Figure D.3: Implémentation Robotran du chariot pendulaire

Validation des resultats

```

linsys =

a =
      x_dot  phi_dot      x      phi
x_dot      -0.1818      0      0      -2.673
phi_dot      0.4545      0      0      31.18
x              1      0      0      0
phi              0      1      0      0

b =
      u
x_dot      1.818
phi_dot     -4.545
x              0
phi              0

```

Figure D.4: Matrices linéarisées calculées analytiquement

```

A
|-0.181818 -0.000000 -0.000000 -2.672947 |
|0.454545 0.000000 0.000000 31.184382 |
|1.000000 0.000000 0.000000 0.000000 |
|0.000000 1.000000 0.000000 0.000000 |

B
|1.818182 |
|-4.545455 |
|0.000000 |
|0.000000 |

```

Figure D.5: Matrices linéarisées calculées par Robotran

Les matrices calculées à l'aide de ces deux méthodes étant identiques, nous pouvons donc valider ce module de calcul.

Remarque : l'expression du vecteur d'état choisie $(\dot{x}, \dot{\phi}, x, \psi)$ n'étant pas la même que celle choisie par [14] $(x, \dot{x}, \phi, \dot{\phi})$, il est nécessaire de "ré-ordonner" les matrices d'état provenant de [14] pour obtenir des résultats comparables à ceux provenant de Robotran. Ces choix de convention sont à garder à l'esprit, afin d'éviter toute incohérence lors de l'implémentation du filtre de Kalman.

D.3 Filtre de Kalman

Le principe du filtre de Kalman est de permettre de traiter plusieurs signaux bruités pour pouvoir en déduire une estimation plus précise de l'état réel dans lequel le système se trouve. Cela nécessite de connaître plusieurs paramètres caractéristiques du système :

Équations d'état (bruitées)

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t) \quad \mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(t))$$

Équations d'observation (bruitées)

$$\mathbf{y}(t) = h(\mathbf{x}(t)) + \mathbf{v}(t) \quad \mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t))$$

où $\mathbf{v}$ et $\mathbf{w}$ sont respectivement les bruits gaussiens de processus et de mesure, c'est-à-dire les bruits qui apparaissent sur la commande ($\mathbf{u}$) et sur les mesures d'état ($\mathbf{x}$), et où $\mathbf{Q}$ et $\mathbf{R}$ sont les matrices de covariance de ces bruits (caractéristiques du modèle et des capteurs utilisés)

Nous pouvons également écrire que :

$$f(\mathbf{x}(t), \mathbf{u}(t)) \simeq \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

L'estimation d'état du filtre de Kalman se fait ensuite en plusieurs étapes :

1. Initialisation (importation de l'état initial)

$$\hat{\mathbf{x}}(t_0) = E[\mathbf{x}(t_0)], \mathbf{P}(t_0) = Var[\mathbf{x}(t_0)]$$

2. Prédiction de l'évolution de l'état (sur base des commandes appliquées)

obtenue en résolvant :

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = f(\hat{\mathbf{x}}(t), \mathbf{u}(t)) \\ \dot{\mathbf{P}}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^\top + \mathbf{Q}(t) \end{cases}$$

Où les matrices $\mathbf{P}(t)$, $\mathbf{Q}(t)$ et $\mathbf{R}(t)$ sont :

- $\mathbf{P}(t)$ Covariance de l'erreur estimée à *priori*
- $\mathbf{Q}(t)$ Covariance du bruit du processus/système
- $\mathbf{R}(t)$ Covariance du bruit de mesure

3. Mise à jour de l'état (sur base de nouvelles mesures)

$$\begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= f(\hat{\mathbf{x}}(t), \mathbf{u}(t)) + \mathbf{K}(t)(\mathbf{z}(t) - h(\hat{\mathbf{x}}(t))) \\ \dot{\mathbf{P}}(t) &= \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}(t)^\top - \mathbf{K}(t)\mathbf{H}(t)\mathbf{P}(t) + \mathbf{Q}(t) \\ \mathbf{K}(t) &= \mathbf{P}(t)\mathbf{H}(t)^\top \mathbf{R}(t)^{-1} \end{aligned}$$

où les différentes variables sont :

- $\mathbf{K}(t)$: gain du filtre de Kalman
- $\mathbf{z}$ - état mesuré
- $\tilde{\mathbf{x}}$ - état estimé

Il peut également être remarqué que les filtres de Kalman variant temporellement permettent de faire varier ces matrices de covariance du système à chaque pas de temps. Ceux-ci ne permettent néanmoins pas de faire varier les matrices d'état du système linéarisé (hautement variables, dans notre cas). Ce genre de filtre ne convient donc pas complètement à notre application.

Il a donc été décidé d'utiliser un filtre de Kalman Étendu (spécialement conçu pour des systèmes comme le nôtre, hautement non-linéaires, dont les équations d'état linéarisées varient fortement en fonction de l'état actuel du modèle). Ce genre de filtre permet de calculer l'expression des équations d'état à chaque itération. Ces filtres sont généralement prévus pour effectuer ces calculs sur base de l'expression analytique de ces équations. Il est cependant possible de remplacer ces équations par leur forme linéarisée, calculée à chaque pas de temps, même si cela est un peu moins efficace.

Dans le cas du filtre de Kalman étendu, les matrices de transition et d'observation ($\mathbf{F}(t)$ et $\mathbf{H}(t)$) sont donc calculées à chaque itération, en évaluant les jacobiniennes des fonctions d'état et d'observation autour de l'état actuel.

$$\begin{aligned} \mathbf{F}(t) &= \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}(t), \mathbf{u}(t)} \\ \mathbf{H}(t) &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}(t)} \end{aligned}$$

Dans notre cas, ces jacobienues sont donc simplement les matrices A et B calculées à chaque pas de temps par Robotran.

Il est par contre important de noter qu'au contraire du filtre de Kalman, "simple", le filtre de Kalman étendu n'est pas basé sur une méthode optimale. Celui-ci converge donc vers une solution locale qui peut, dans certains cas, converger vers un état erroné, en fonction des conditions initiales choisies.

